

**Владимирский государственный университет**

**Ю. М. МОНАХОВ**

**ПРИКЛАДНЫЕ АЛГОРИТМЫ**

**Алгоритмы на графах и сетях**

**Учебное пособие**

**Владимир 2023**

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Ю. М. МОНАХОВ

# ПРИКЛАДНЫЕ АЛГОРИТМЫ

## Алгоритмы на графах и сетях

Учебное пособие

*Электронное издание*



Владимир 2023

ISBN 978-5-9984-1740-5

© ВлГУ, 2023

© Монахов Ю. М., 2023

УДК 004.056

ББК 32.97

Рецензенты:

Доктор педагогических наук, профессор  
зав. кафедрой математики, информатики и методики обучения  
Шуйского филиала Ивановского государственного университета

*С. А. Зайцева*

Кандидат технических наук, доцент  
доцент кафедры информатики и защиты информации  
Владимирского государственного университета  
имени Александра Григорьевича и Николая Григорьевича Столетовых

*А. В. Тельный*

**Монахов, Ю. М.**

**ПРИКЛАДНЫЕ АЛГОРИТМЫ.** Алгоритмы на графах и сетях [Электронный ресурс] : учеб. пособие / Ю. М. Монахов ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2023. – 335 с. – ISBN 978-5-9984-1740-5. – Электрон. дан. (6,83 Мб). – 1 электрон. опт. диск (CD-ROM). – Системные требования: Intel от 1,3 ГГц ; Windows XP/7/8/10 ; Adobe Reader ; дисковод CD-ROM. – Загл. с титул. экрана.

Представлен систематизированный материал по курсу «Прикладные алгоритмы», посвященный применению теории графов в практических задачах.

Предназначено для студентов вузов направлений подготовки 10.04.01 «Информационная безопасность», 10.05.04 «Информационно-аналитические системы безопасности», 09.04.02 «Информационные системы и технологии», 09.04.04 «Программная инженерия» всех форм обучения, а также для широкого круга специалистов по информационным технологиям.

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Табл. 38. Ил. 148. Библиогр.: 230 назв.

ISBN 978-5-9984-1740-5

© ВлГУ, 2023  
© Монахов Ю. М., 2023

=====

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	4
Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ.....	6
1.1. Обход графов в ширину и глубину .....	6
Вопросы и задания .....	16
1.2. Алгоритм Тэрри.....	29
Вопросы и задания .....	35
1.3. Матроиды, жадные алгоритмы, алгоритм Краскала .....	39
Вопросы и задания .....	48
1.4. Сетевые алгоритмы. Выбор кратчайшего пути .....	50
Вопросы и задания .....	70
1.5. Алгоритм Форда – Фалкерсона .....	97
Вопросы и задания .....	106
Список рекомендуемых источников .....	114
Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ .....	120
2.1. Использование графовых моделей в задаче оценки живучести распределенных информационных систем .....	120
Вопросы и задания .....	247
Список рекомендуемых источников .....	248
2.2. Использование теории графов для моделирования распространения информации в крупномасштабных социальных сетях .....	257
Вопросы и задания .....	299
Список рекомендуемых источников .....	300
ЗАКЛЮЧЕНИЕ .....	316
ПРИЛОЖЕНИЕ .....	317

=====

## ВВЕДЕНИЕ

Растущий интерес к дискретной математике вполне понятен. Это молодая, стремительно развивающаяся отрасль математического знания. С одной стороны, дискретная математика создает фундаментальные предпосылки для развития идей алгоритмизации, программирования, информатизации. С другой стороны, развитие вычислительной техники, информационных сетей и технологий постоянно приводит к появлению целого ряда новых задач в дискретной математике. Круг таких задач настолько широк и разнообразен, что в рамках дискретной математики уже сейчас можно выделить множество разделов: комбинаторика, теория множеств, теория графов, теория конечных автоматов и управляемых процессов, теория информации и кодирования, теория алгоритмов, теория сложности алгоритмов. Каждый из перечисленных разделов заслуживает самостоятельного учебного курса, одно учебное пособие не может их заменить.

В настоящем пособии рассмотрены практические приложения теории графов и орграфов в широком круге задач. Задачи о кратчайшем или наиболее выгодном пути с определенной точки зрения возникают при решении вопросов безопасности и эффективности работы информационных, транспортных, экономических, производственных сетей и сообщений. Множество более сложных задач можно свести к повторительным схемам, в которых работают алгоритмически решенные задачи теории графов.

Автор не ставил целью сделать изложение материала учебного пособия полным и математически обоснованным. Наибольшее внимание уделено матричным способам представления графов, постановкам поисковых задач на графах и методам их алгоритмической реализации, т. е. тем прикладным вопросам, с которыми приходится сталкиваться любому программисту.

## ВВЕДЕНИЕ

=====

В первой главе приводятся прикладные алгоритмы теории графов: алгоритмы обхода графов в ширину и глубину, получения ориентированного графа, поиска маршрута в связном графе (алгоритм Тэрри); рассматриваются матроиды, алгоритм Краскала, задача о нахождении кратчайшего пути (алгоритм Уоршалла, алгоритм топологической сортировки, алгоритм Беллмана – Форда, алгоритм Дейкстра, алгоритм Флойда, алгоритм Флойда – Уоршалла), задача о нахождении максимального потока в сети (алгоритм Форда – Фалкерсона). Алгоритмы рассматриваются на примерах.

Вторая глава посвящена вопросам применения графовых моделей в практических задачах, в частности в задаче оценки живучести распределенных информационных систем, и для моделирования распространения информации в крупномасштабных социальных сетях.

Пособие содержит варианты заданий. В приложении даются ответы на каждое задание.

=====

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

### 1.1. Обход графов в ширину и глубину

#### *Описание алгоритма*

Пусть нам необходимо обойти граф  $G(V, E)$ , который представлен списком смежности  $\Gamma$ . Обход графа подразумевает некоторое систематическое перечисление его вершин. Для этого используются следующие вспомогательные структуры данных:

1) *Структура данных  $T$*  является своего рода вспомогательным буфером, в который временно помещаются обойденные вершины (это необходимо для обхода смежных с ними вершин). Данная структура может являться стеком (в случае поиска в глубину) или очередью (в случае поиска в ширину).

*Стек* – это структура данных, в которой первый помещенный в нее элемент извлекается последним.

*Очередь* – это структура данных, в которой первый помещенный в нее элемент извлекается первым.

2) *Массив  $X$* , длина которого равна числу вершин, содержит данные о том, была ли отмечена (пройдена) вершина. Каждый элемент массива соответствует одной вершине графа и может принимать два значения:

1 – вершина отмечена (пройдена);

0 – вершина не отмечена.

Рассмотрим, как осуществляется обход поэтапно:

1. обнуление массива  $X$ . До начала обхода все вершины являются неотмеченными;
2. выбирается некоторая произвольная вершина  $v$ , с которой будет начинаться обход;
3. вершина  $v$  помещается в структуру  $T$  и отмечается в массиве  $X$  как пройденная ( $X[v] = 1$ );

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

4. из структуры  $T$  извлекается вершина (обозначим её  $u$ ). Эта вершина является пройденной. Таким образом, именно на этом этапе мы выделяем обойденные вершины;
5. по списку смежности  $\Gamma$  поочередно выбираются все вершины смежные с  $u$  (обозначим вершину смежную с  $u$  –  $w$ ), и если они не были ранее отмечены (то есть, если  $X[w] = 0$ ), то они помещаются в структуру  $T$  и отмечаются.

Если в структуре  $T$  находятся какие-либо вершины, то осуществляется переход к пункту 4. Если же нет, то обход графа  $G(V, E)$  закончен.

### Пример 1.1.1.

#### *Обход графа в глубину*

Рассмотрим на примере обход графа  $G(V, E)$  (Рисунок 1.1.1).

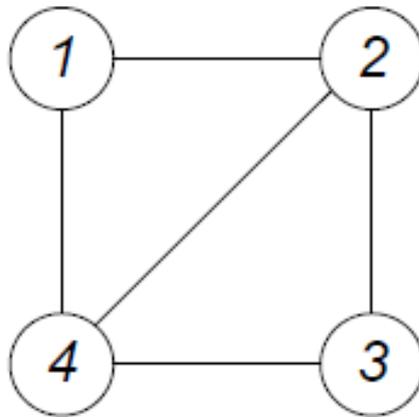


Рисунок 1.1.1 – Исходный граф примера 1.1.1

Список смежности ( $\Gamma$ )

1 – 2, 4

2 – 1, 3, 4

3 – 2, 4

4 – 1, 2, 3

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

1) Обнулим массив  $X$ .

$X$ :

1	2	3	4
0	0	0	0

2) Начнем обход с первой вершины ( $v = 1$ ).

3) Поместим вершину ( $v = 1$ ) в структуру  $T$  и отметим ее в массиве  $X$  как пройденную.

$T$ :

1			
---	--	--	--

$X$ :

1	2	3	4
1	0	0	0

4) Извлекаем из структуры  $T$  вершину 1 ( $u = 1$ ), т.е. вершина 1 пройдена.

$T$ :

--	--	--	--

5) Затем помещаем вершины смежные с первой ( $u = 1$ ) и еще не отмеченные в массиве  $X$  в структуру  $T$  и отмечаем их в массиве  $X$ .

$T$ :

2	4		
---	---	--	--

$X$ :

1	2	3	4
1	1	0	1

6) Извлекаем из структуры  $T$  вершину 4 ( $u = 4$ ), т.е. вершина 4 пройдена.

$T$ :

2			
---	--	--	--

7) Затем помещаем вершины смежные с четвертой ( $u = 4$ ) и еще не отмеченные в массиве  $X$  (2 и 3) в структуру  $T$  и отмечаем их в массиве  $X$ .

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

$T$ : 

2	3		
---	---	--	--

$X$ : 

1	2	3	4
1	1	1	1

8) Извлекаем из структуры  $T$  вершину 3 ( $u = 3$ ), т.е. вершина 3 пройдена.

$T$ : 

2			
---	--	--	--

9) Все вершины смежные с третьей уже отмечены в массиве  $X$ , поэтому ничего не помещаем в структуру  $T$ .

10) Извлекаем из структуры  $T$  вершину 2 ( $u = 2$ ), т.е. вершина 2 пройдена.

$T$ : 

--	--	--	--

11) Все вершины смежные со второй уже отмечены в массиве  $X$ , поэтому ничего не помещаем в структуру  $T$ .

12) Структура  $T$  пуста. Обход вершин графа закончен: 1 – 4 – 3 – 2.

Пример 1.1.2

***Обход графа в ширину***

Рассмотрим на примере обход графа  $G(V, E)$  (Рисунок 1.1.2).

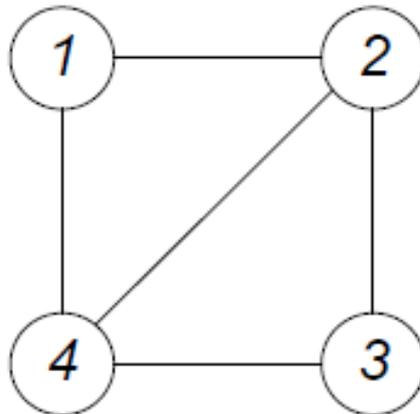


Рисунок 1.1.2 – Исходный граф примера

Список смежности ( $G$ )

1 – 2, 4

2 – 1, 3, 4

3 – 2, 4

5 – 1, 2, 3

1) Обнулим массив  $X$ .

$X$ :

1	2	3	4
0	0	0	0

2) Начнем обход с первой вершины ( $v=1$ ).

3) Поместим 1 вершину ( $v = 1$ ) в структуру  $T$  и отметим ее в массиве  $X$  как пройденную.

$T$ :

1			
---	--	--	--

$X$ :

1	2	3	4
1	0	0	0

4) Извлекаем из структуры  $T$  вершину 1 ( $u = 1$ ), т.е. вершина 1 пройдена.

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

$T$ : 

--	--	--	--

5) Затем помещаем вершины смежные с первой ( $u=1$ ) и еще не отмеченные в массиве  $X$  в структуру  $T$  и отмечаем их в массиве  $X$ .

$T$ : 

2	4		
---	---	--	--

$X$ : 

1	2	3	4
1	1	0	1

6) Извлекаем из структуры  $T$  вершину 2 ( $u=2$ ), т.е. вершина 2 пройдена.

$T$ : 

4			
---	--	--	--

7) Затем помещаем вершины смежные со второй ( $u=2$ ) и еще не отмеченные в массиве  $X$  (3) в структуру  $T$  и отмечаем их в массиве  $X$ .

$T$ : 

4	3		
---	---	--	--

$X$ : 

1	2	3	4
1	1	1	1

8) Извлекаем из структуры  $T$  вершину 4 ( $u=4$ ), т.е. вершина 4 пройдена.

$T$ : 

3			
---	--	--	--

9) Все вершины смежные с четвертой уже отмечены в массиве  $X$ , поэтому ничего не помещаем в структуру  $T$ .

10) Извлекаем из структуры  $T$  вершину 3 ( $u=3$ ), т.е. вершина 3 пройдена.

=====

$T$ : 

--	--	--	--

- 11) Все вершины смежные с третьей уже отмечены в массиве  $X$ , поэтому ничего не помещаем в структуру  $T$ .
- 12) Структура  $T$  пуста. Обход вершин графа закончен: 1 – 2 – 4 – 3.

На практике обход в глубину можно использовать в частности для получения ориентированного дерева из неориентированного. Рассмотрим это на примере.

Пример 1.1.3

*Получения ориентированного дерева*

Исходное дерево представлено на рисунке 1.1.3.

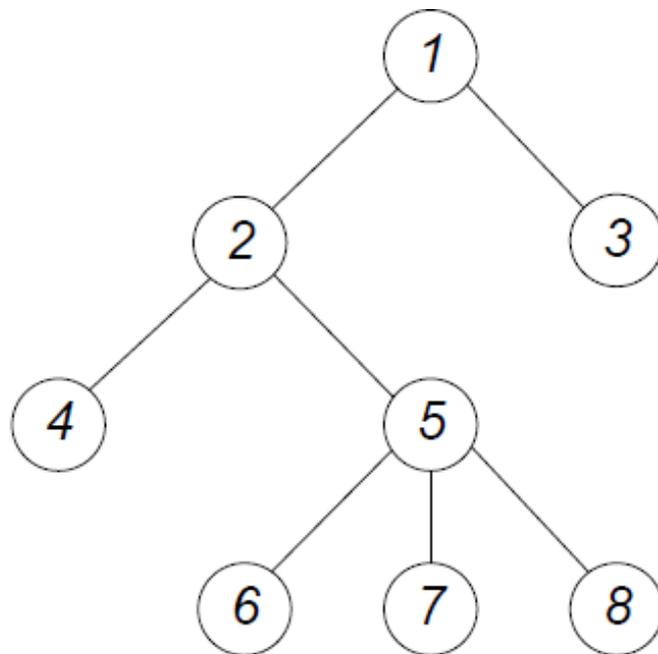


Рисунок 1.1.3 - Исходный граф примера

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

Список смежности ( $T$ )

1 – 2, 3

2 – 1, 4, 5

3 – 1

4 – 2

5 – 2, 6, 7, 8

6 – 5

7 – 5

8 – 5

Наша цель – посредством обхода исходного дерева в глубину, получить ориентированное дерево с корнем  $v$  и списком смежности  $T'$ .

1) Обнулим массив  $X$ .

X:	1	2	3	4	5	6	7	8
	0	0	0	0	0	0	0	0

2) Пусть корнем будет вершина 5 ( $v = 5$ ). Поместим вершину 5 в структуру  $T$  и пометим ее как пройденную в массиве  $X$ .

T:	5							
----	---	--	--	--	--	--	--	--

X:	1	2	3	4	5	6	7	8
	0	0	0	0	1	0	0	0

3) Затем извлекаем из структуры  $T$  вершину ( $u = 5$ ) и помещаем смежные с ней вершины (ранее не пройденные) в  $T$ .

T:	2	6	7	8				
----	---	---	---	---	--	--	--	--

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

$X:$	1	2	3	4	5	6	7	8
	0	1	0	0	1	1	1	1

4) Начинаем формировать список смежности ориентированного дерева ( $\Gamma'$ ). Для этого берем вершину  $u$  и выписываем из списка смежности исходного графа ( $\Gamma$ ) вершины, совпадающие с вершинами, находящимися в структуре  $T$  (это будут вершины смежные с  $u$ ), то есть

$$\Gamma': 5 - 2, 6, 7, 8$$

5) Затем извлекаем из структуры  $T$  вершину 8 ( $u = 8$ ). Дополняем список смежности  $\Gamma'$  восьмой вершиной. У нее нет смежных вершин.

$T:$	2	6	7					
------	---	---	---	--	--	--	--	--

$$\Gamma': 5 - 2, 6, 7, 8; 8 -;$$

6) Затем извлекаем из структуры  $T$  вершину 7 ( $u = 7$ ). Дополняем список смежности  $\Gamma'$ .

$T:$	2	6						
------	---	---	--	--	--	--	--	--

$$\Gamma': 5 - 2, 6, 7, 8; 8 -; 7 -;$$

7) Затем извлекаем из структуры  $T$  вершину 6 ( $u = 6$ ). Дополняем список смежности  $\Gamma'$ .

$T:$	2							
------	---	--	--	--	--	--	--	--

$$\Gamma': 5 - 2, 6, 7, 8; 8 -; 7 -; 6 -$$

8) Затем извлекаем из структуры  $T$  вершину 2 ( $u = 2$ ). Дополняем список смежности  $\Gamma'$ .

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

$T$ : 

1	4						
---	---	--	--	--	--	--	--

$X$ : 

1	2	3	4	5	6	7	8
1	1	0	1	1	1	1	1

$\Gamma'$ : 5 – 2, 6, 7, 8; 8 –; 7 –; 6 –; 2 – 1, 4;

9) Извлекаем из структуры  $T$  вершину 4 ( $u = 4$ ). Дополняем список смежности  $\Gamma'$ .

$T$ : 

1							
---	--	--	--	--	--	--	--

$\Gamma'$ : 5 – 2, 6, 7, 8; 8 –; 7 –; 6 –; 2 – 1, 4; 4 –;

10) Извлекаем из структуры  $T$  вершину 1 ( $u = 1$ ). Дополняем список смежности  $\Gamma'$ . Помещаем смежные с ней вершины (ранее не пройденные) в  $T$ .

$T$ : 

3							
---	--	--	--	--	--	--	--

$X$ : 

1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1

$\Gamma'$ : 5 – 2, 6, 7, 8; 8 –; 7 –; 6 –; 2 – 1, 4; 4 –; 1 – 3;

11) Извлекаем из структуры  $T$  вершину 3 ( $u = 3$ ). Дополняем список смежности  $\Gamma'$ .

$T$ : 

--	--	--	--	--	--	--	--

$\Gamma'$ : 5 – 2, 6, 7, 8; 8 –; 7 –; 6 –; 2 – 1, 4; 4 –; 1 – 3; 3 –;

=====

12) Структура  $T$  пуста. Обход графа закончен. Получили следующий список смежности ориентированного дерева ( $\Gamma'$ ):

- 1 – 3
- 2 – 1, 4
- 3 –
- 4 –
- 5 – 2, 6, 7, 8
- 6 –
- 7 –
- 8 –

**Вопросы и задания**

1. Какие вспомогательные структуры данных используются в алгоритме «Обход графов в ширину и глубину»?
2. Какие этапы содержит алгоритм «Обход графов в ширину и глубину»?
3. Построить из дерева ориентированное методом обхода в глубину из первой вершины. Варианты 1-6 представлены на рисунках 1.1.4 – 1.1.9.

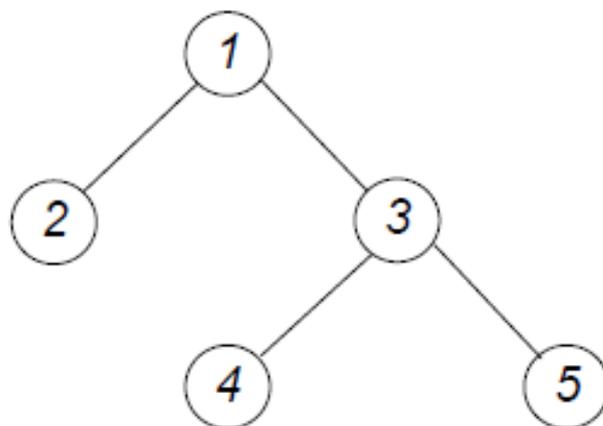


Рисунок 1.1.4 - Исходный граф варианта 1

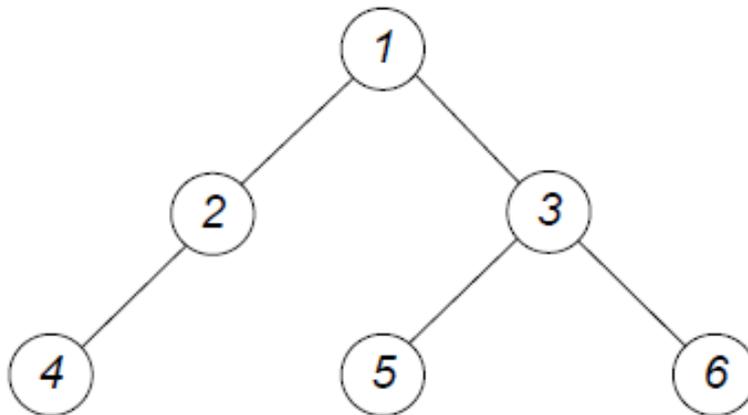


Рисунок 1.1.5 - Исходный граф варианта 2

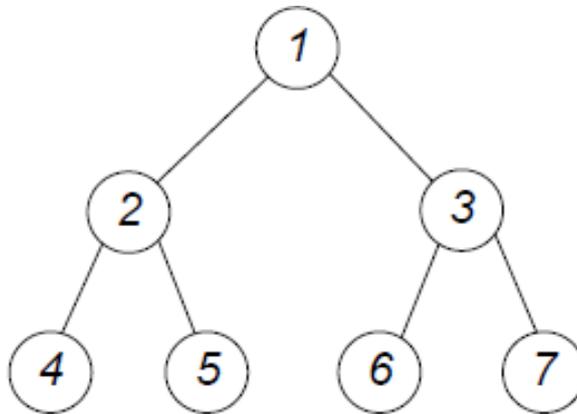


Рисунок 1.1.6 - Исходный граф варианта 3

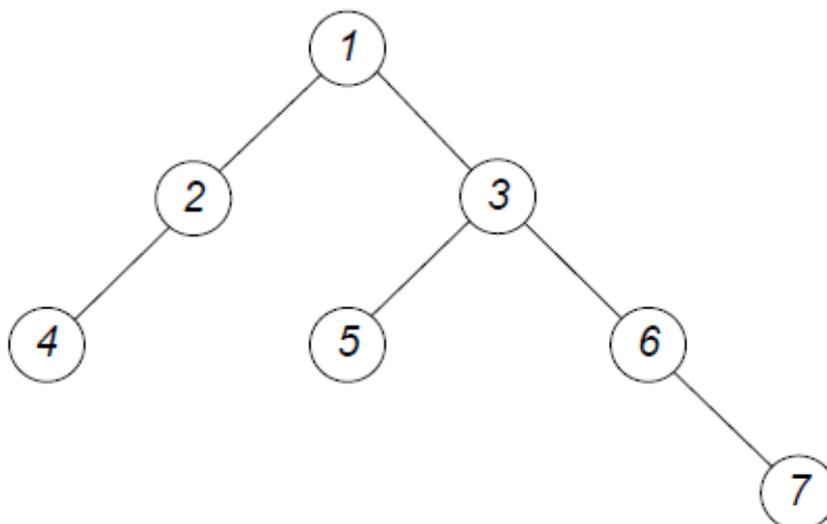


Рисунок 1.1.7 - Исходный граф варианта 4

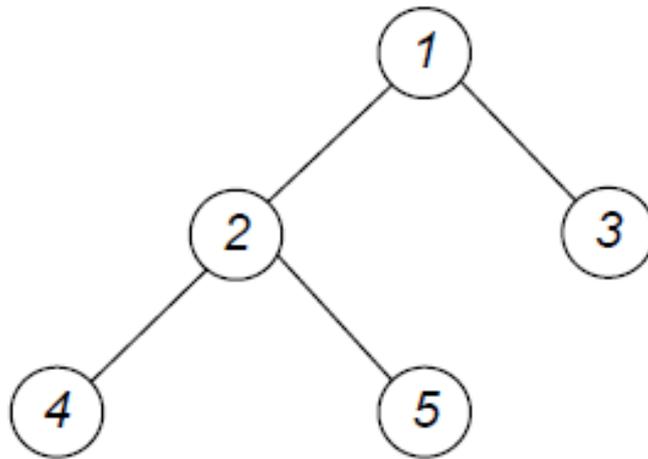


Рисунок 1.1.8 - Исходный граф варианта 5

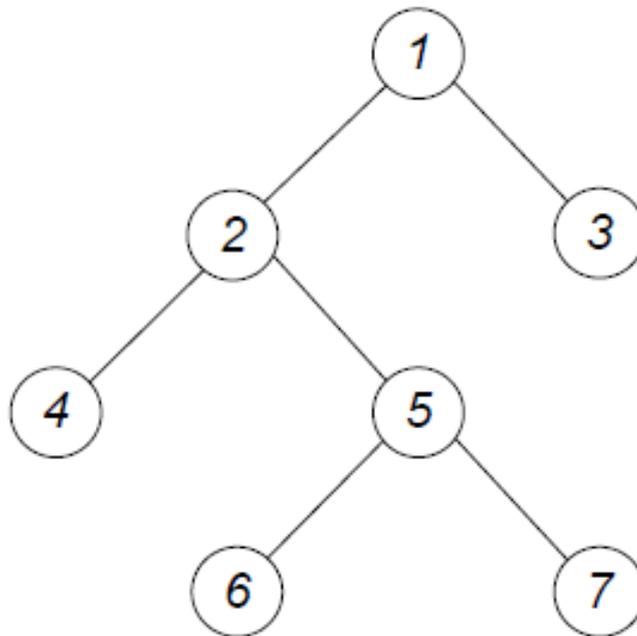


Рисунок 1.1.9 - Исходный граф варианта 6

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

4. Построить из дерева ориентированное методом обхода в глубину из третьей вершины. Варианты 7 - 12 представлены на рисунках 1.1.10 – 1.1.15.

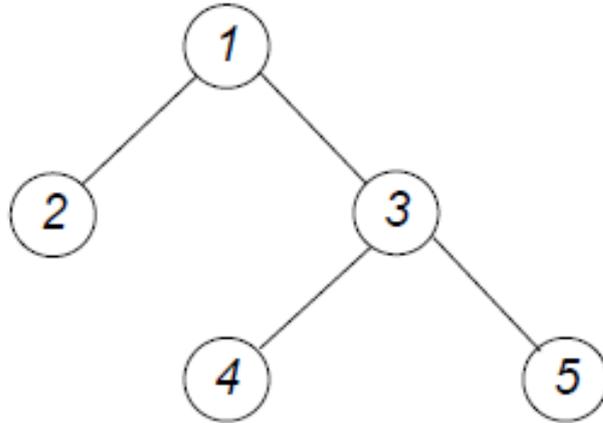


Рисунок 1.1.10 - Исходный граф варианта 7

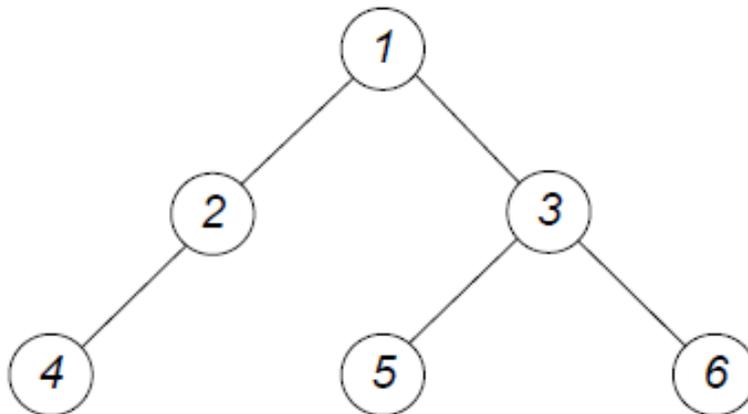


Рисунок 1.1.11 - Исходный граф варианта 8

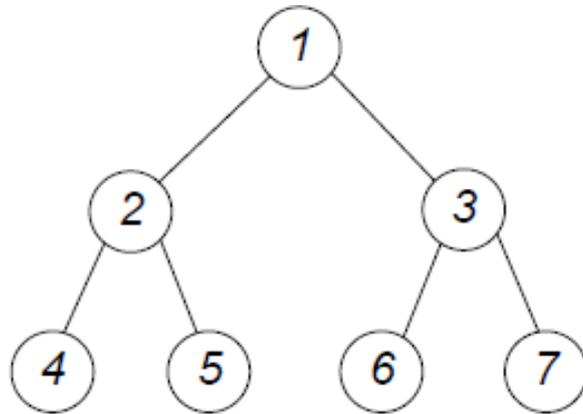


Рисунок 1.1.12 - Исходный граф варианта 9

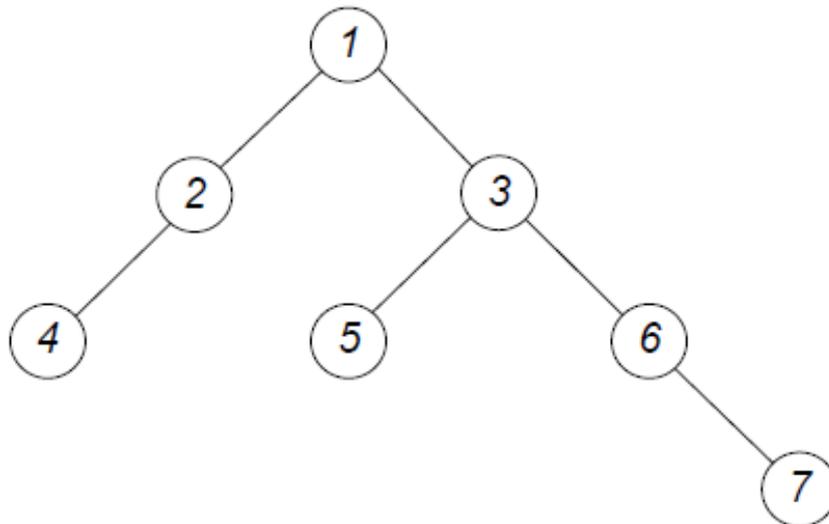


Рисунок 1.1.13 - Исходный граф варианта 10

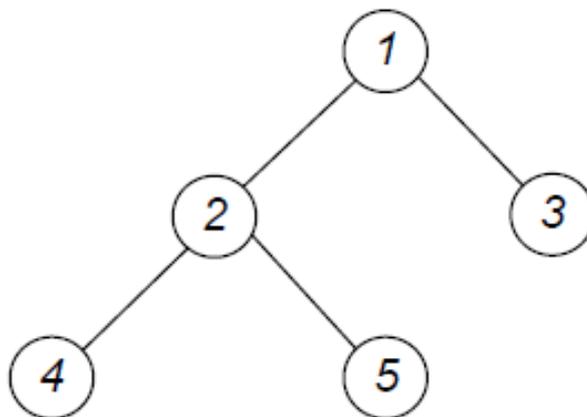


Рисунок 1.1.14 - Исходный граф варианта 11

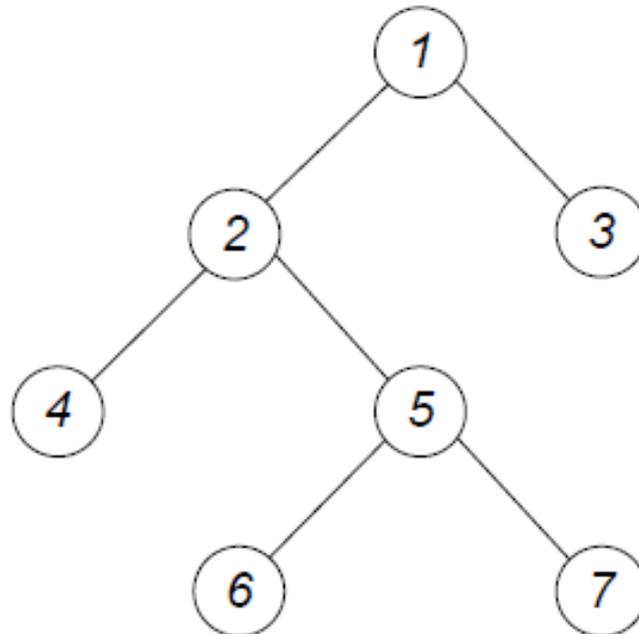


Рисунок 1.1.15 - Исходный граф варианта 12

5. Построить из дерева ориентированное методом обхода в глубину из второй вершины. Варианты 13 - 18 представлены на рисунках 1.1.16 – 1.1.21.

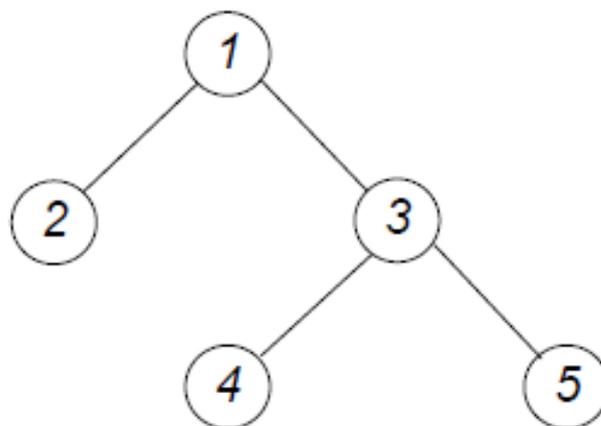


Рисунок 1.1.16 - Исходный граф варианта 13

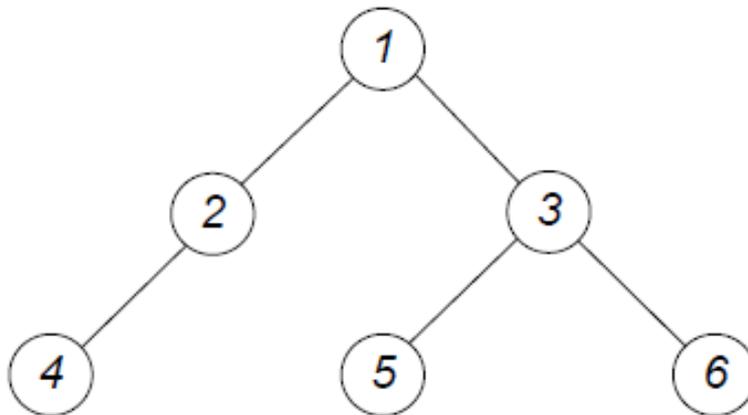


Рисунок 1.1.17 - Исходный граф варианта 14

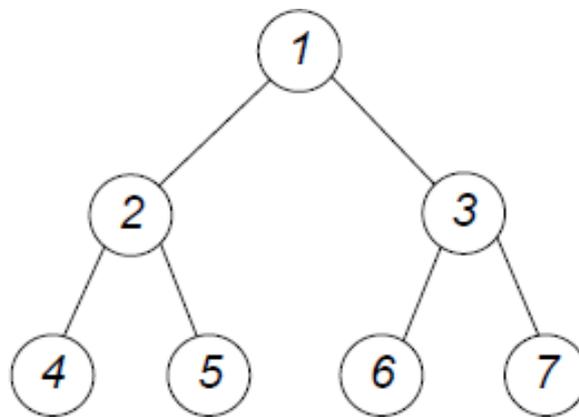


Рисунок 1.1.18 - Исходный граф варианта 15

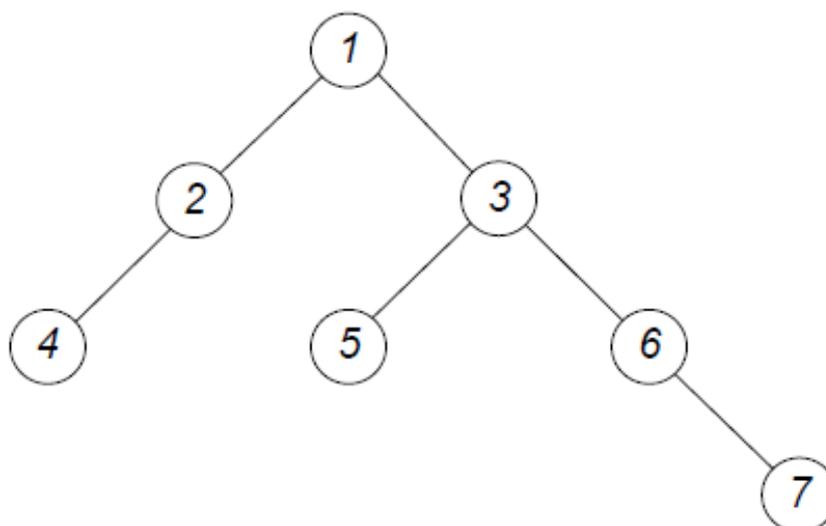


Рисунок 1.1.19 - Исходный граф варианта 16

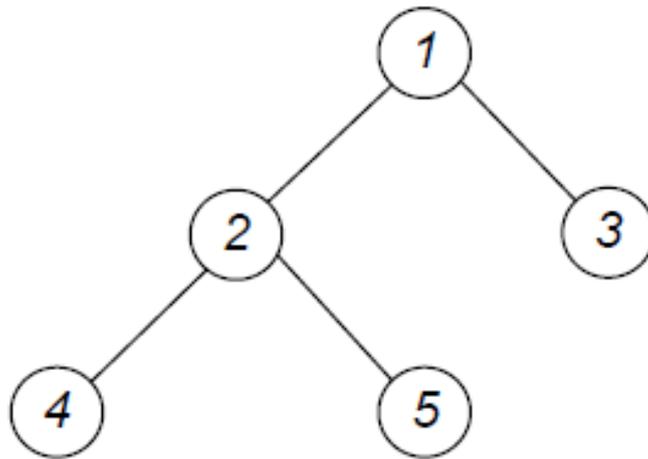


Рисунок 1.1.20 - Исходный граф варианта 17

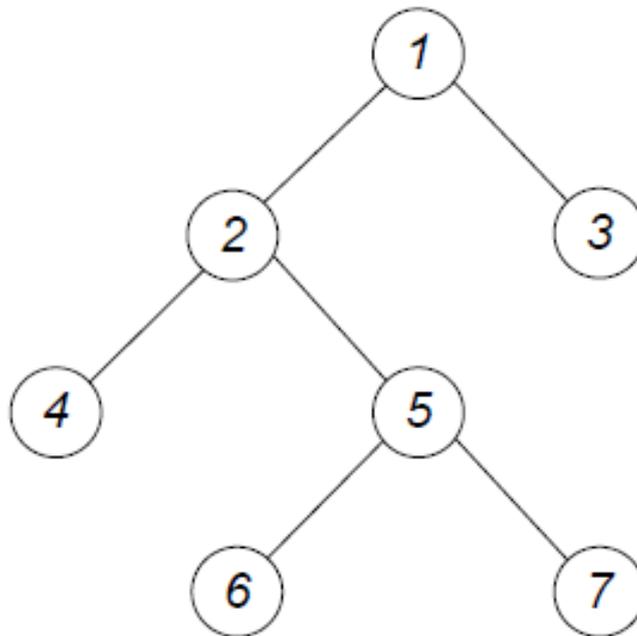


Рисунок 1.1.21 - Исходный граф варианта 18

6. Построить из дерева ориентированное методом обхода в глубину из четвертой вершины. Варианты 19 - 24 представлены на рисунках 1.1.22 – 1.1.27.

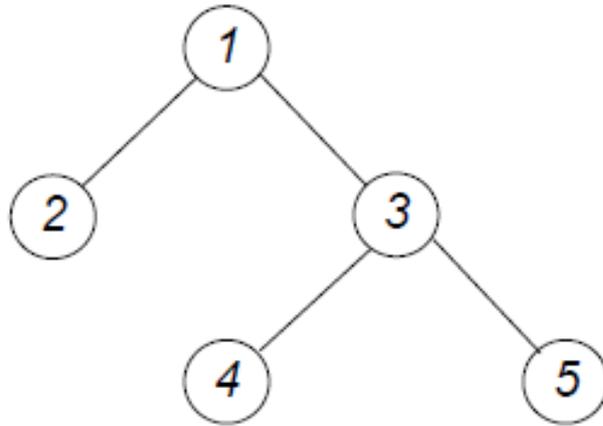


Рисунок 1.1.22 - Исходный граф варианта 19

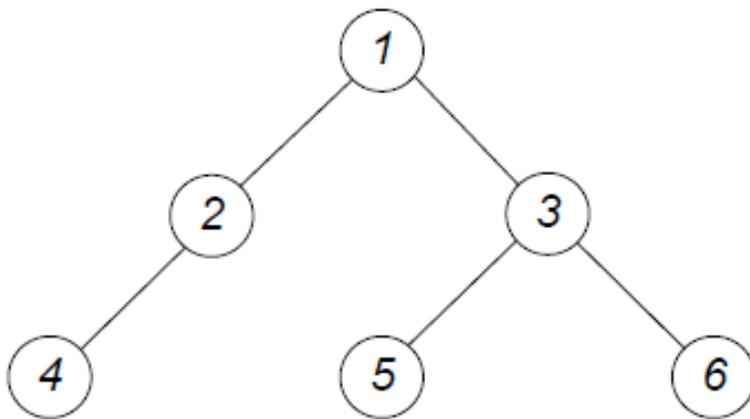


Рисунок 1.1.23 - Исходный граф варианта 20

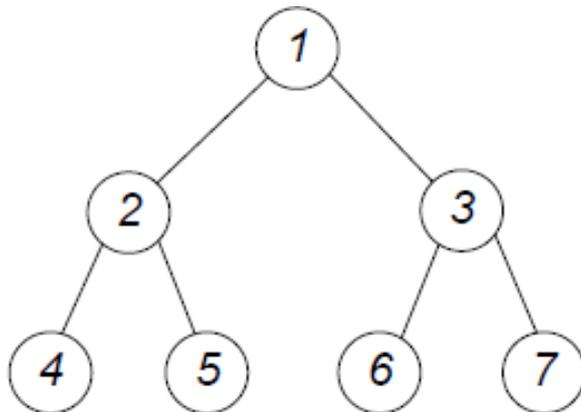


Рисунок 1.1.24 - Исходный граф варианта 21

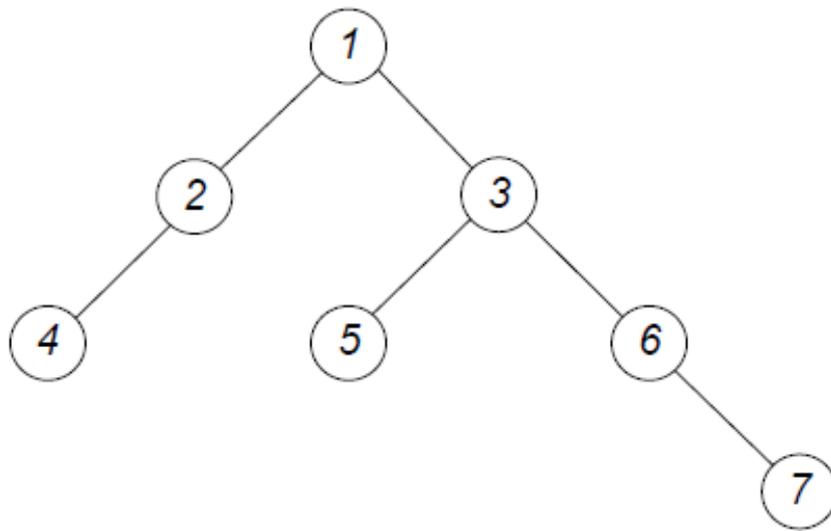


Рисунок 1.1.25 - Исходный граф варианта 22

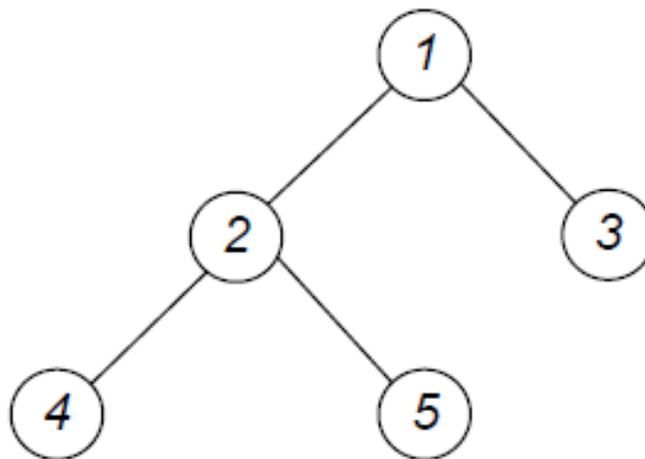


Рисунок 1.1.26 - Исходный граф варианта 23

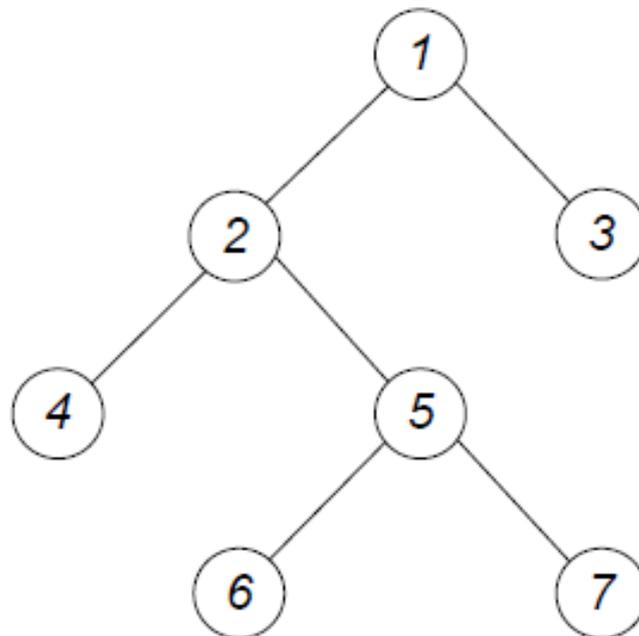


Рисунок 1.1.27 - Исходный граф варианта 24

7. Построить из дерева ориентированное методом обхода в глубину из первой вершины. Варианты 25 - 30 представлены на рисунках 1.1.28 – 1.1.33.

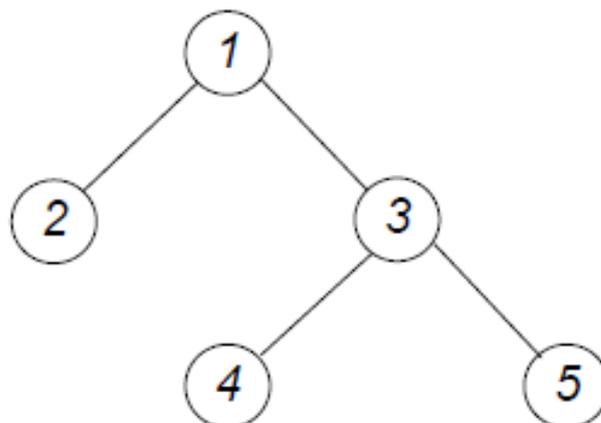


Рисунок 1.1.28 - Исходный граф варианта 25

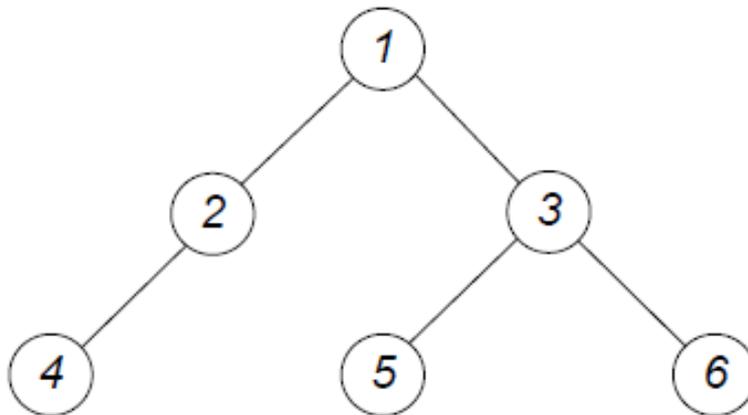


Рисунок 1.1.29 - Исходный граф варианта 26

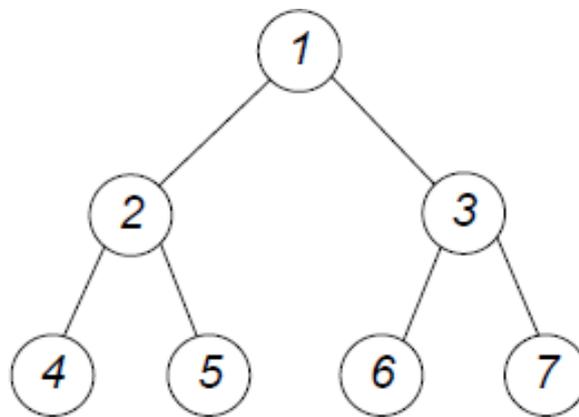


Рисунок 1.1.30 - Исходный граф варианта 27

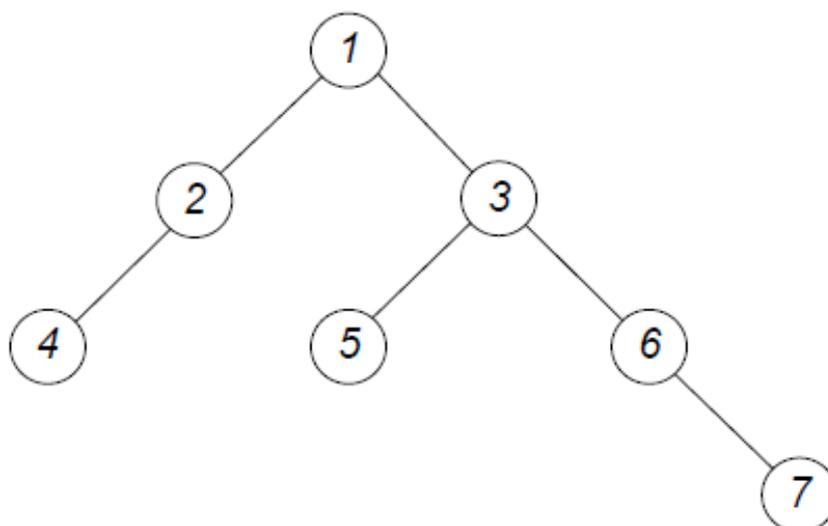


Рисунок 1.1.31 - Исходный граф варианта 28

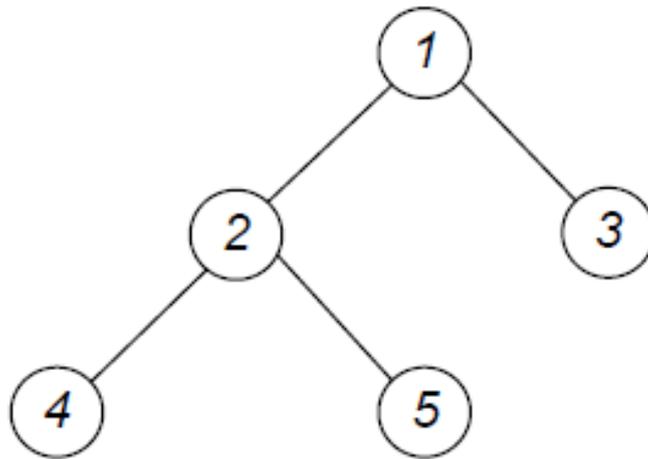
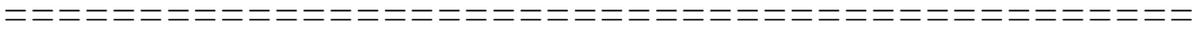


Рисунок 1.1.32 - Исходный граф варианта 29

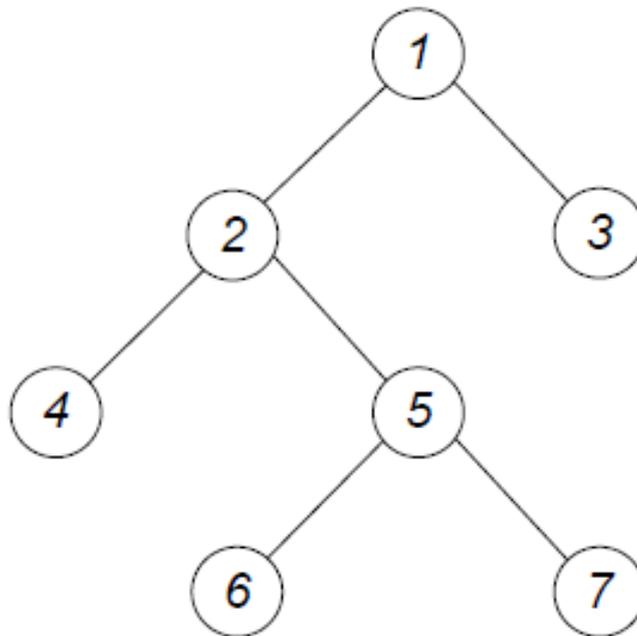


Рисунок 1.1.33 - Исходный граф варианта 30

=====

## 1.2. Алгоритм Тэрри

### *Описание алгоритма*

Алгоритм Тэрри – алгоритм поиска маршрута в связном графе  $G(V, E)$ , соединяющий заданные вершины  $v$  и  $w$ . Исходя из вершины  $v$  и осуществляя последовательный переход от каждой достигнутой вершины к смежной ей вершине, всегда можно найти маршрут в связном графе  $G(V, E)$ , соединяющий заданные вершины  $v$  и  $w$ . Переход от вершины к вершине согласно алгоритму, осуществляется по следующим правилам:

1. при проходе ребра необходимо всякий раз отмечать направление, в котором оно было пройдено;
2. исходя из некоторой вершины  $v'$  нужно всегда следовать по тому ребру, которое не было пройдено или было пройдено в противоположном направлении;
3. для всякой вершины  $v'$ , отличной от  $v$ , необходимо отмечать пометкой «х» первое заходящее ребро, если вершина  $v'$  встречается первый раз;
4. исходя из вершины  $v'$ , отличной от  $v$ , по первому заходящему в  $v'$  ребру нужно идти лишь тогда, когда нет других возможностей.

#### Пример 1.2.1

### *Поиск маршрута между вершинами*

Рассмотрим алгоритм Тэрри на примере поиска маршрута между вершинами 1 и 4 графа  $G(V, E)$  (Рисунок 1.2.1). Если это не противоречит правилам алгоритма, то переходить от одной вершины будем к смежной ей с меньшим номером.

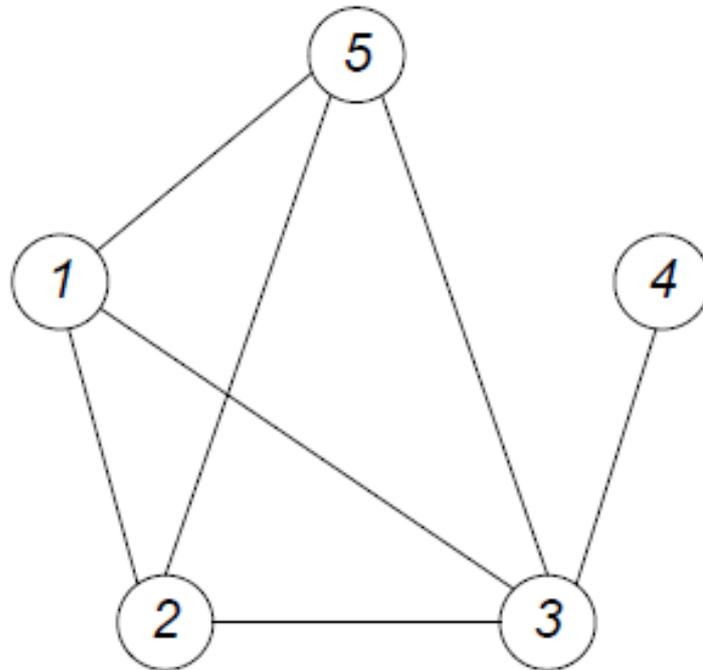


Рисунок 1.2.1 – Исходный граф примера 1.2.1

1. Переходим из вершины 1 в вершину 2. отмечаем направление прохода ребра (Рисунок 1.2.2).

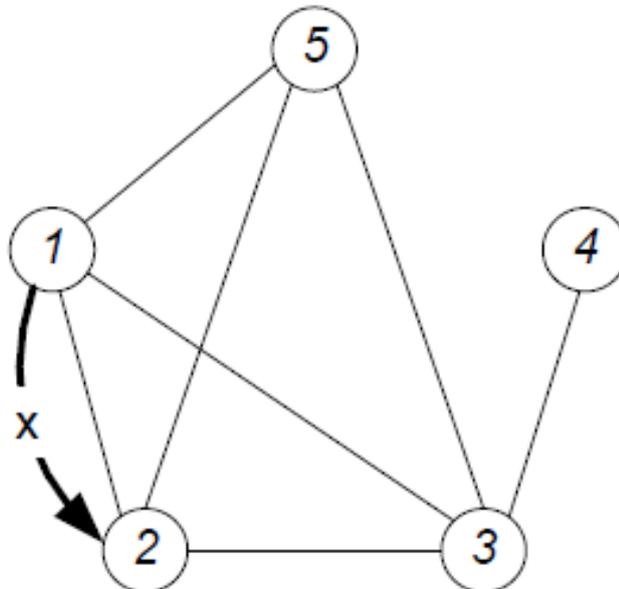


Рисунок 1.2.2 – Текущий граф примера 1.2.1

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

- =====
2. Из вершины 2 можем перейти к вершинам 3 и 5, а также к вершине 1, но лишь в том случае, если нет других вариантов. Переходим к вершине 3. Отмечаем направление перехода (Рисунок 1.2.3).

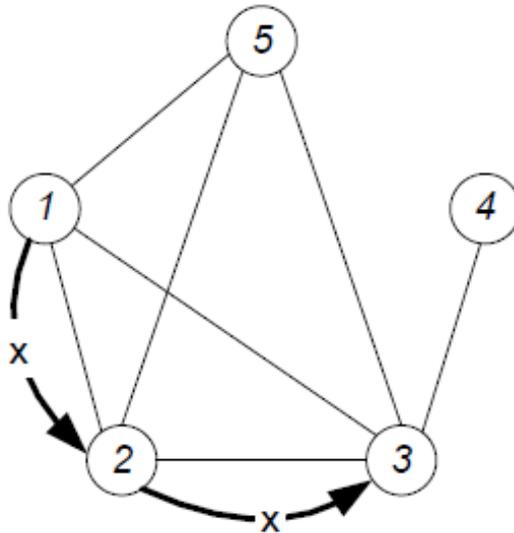


Рисунок 1.2.3 – Текущий граф примера 1.2.1

3. Из третьей вершины переходим в первую, так как она имеет наименьший номер среди всех возможных вариантов перехода (Рисунок 1.2.4).

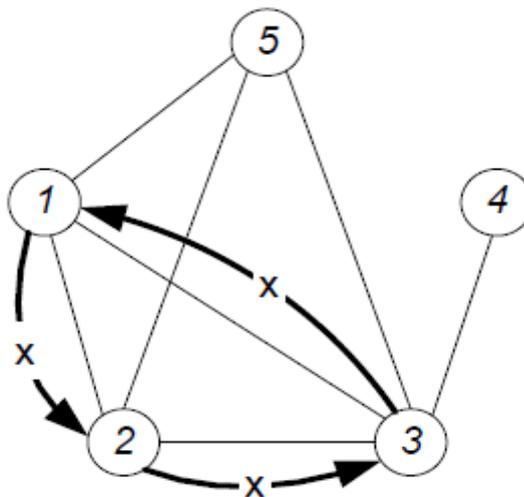


Рисунок 1.2.4 – Текущий граф примера 1.2.1

4. Первая вершина смежна с вершинами 2, 3 и 5. В вершину 2 мы не можем осуществить переход, так как это направление уже отмечено. В вершину 3 осуществлять переход не будем, так как ребро 1-3 уже отмечено как заходящее в вершину 1 и существует альтернатива перехода к вершине 5. Следовательно, переходим к вершине 5. Отмечаем направление перехода (Рисунок 1.2.5).

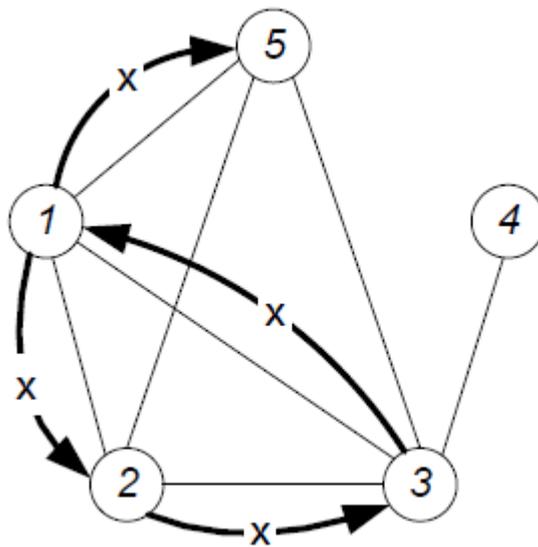


Рисунок 1.2.5 – Текущий граф примера 1.2.1

5. Из вершины 5 переходим к вершине 2, так как она имеет наименьший номер среди всех возможных вариантов перехода (Рисунок 1.2.6).

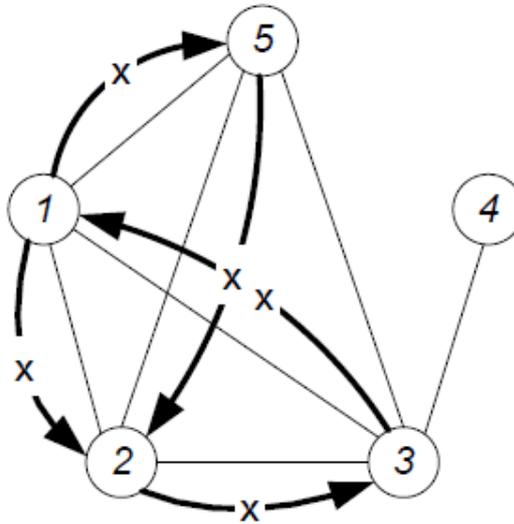


Рисунок 1.2.6 – Текущий граф примера 1.2.1

6. Из вершины 2 переходим к вершине 1, так как ребро, связывающее эти вершины, отмечено как заходящее в вершину 2, а не отмеченных ребер при рассматриваемой вершине нет. Отмечаем направление перехода (Рисунок 1.2.7).

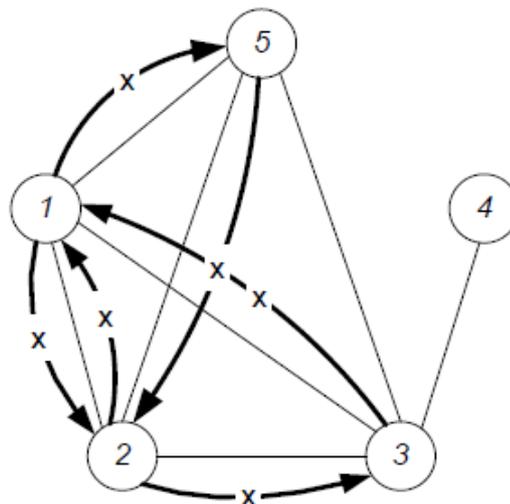


Рисунок 1.2.7 – Текущий граф примера 1.2.1

7. Из вершины 1 можно перейти только к вершине 3 (Рисунок 1.2.8).

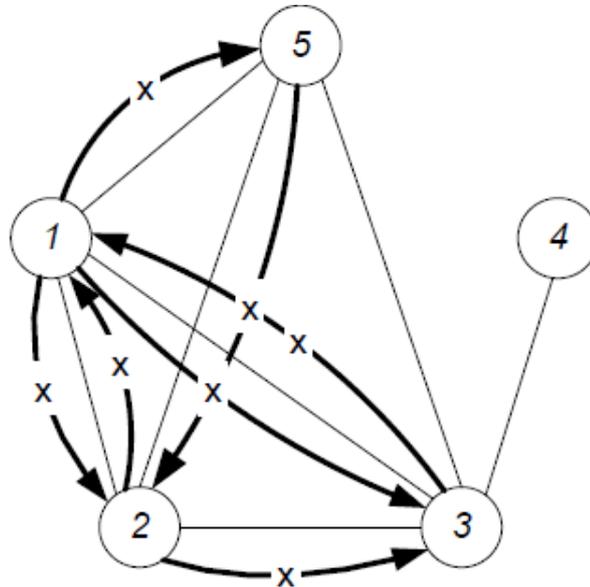


Рисунок 1.2.8 – Текущий граф примера 1.2.1

8. Из вершины 3 осуществляем переход к вершине 4, так как она имеет наименьший номер среди всех альтернативных вариантов (Рисунок 1.2.9). Поиск маршрута окончен.

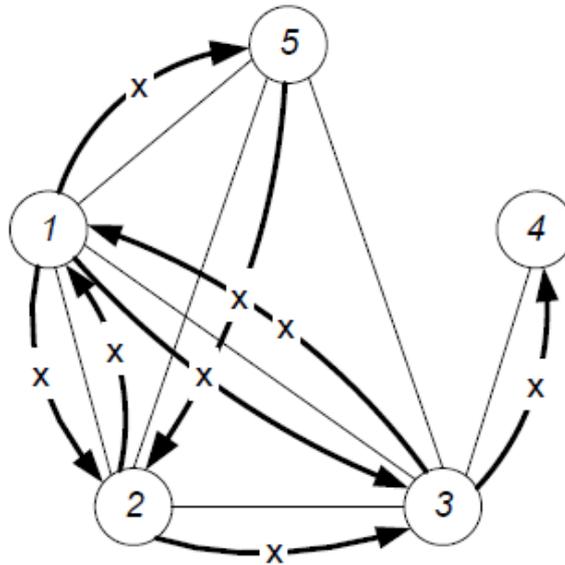


Рисунок 1.2.9 – Окончательный граф примера 1.2.1

Маршрут из вершины 1 в вершину 4 в соответствии с алгоритмом Тэрри имеет вид:

**1 – 2 – 3 – 1 – 5 – 2 – 1 – 3 – 4**

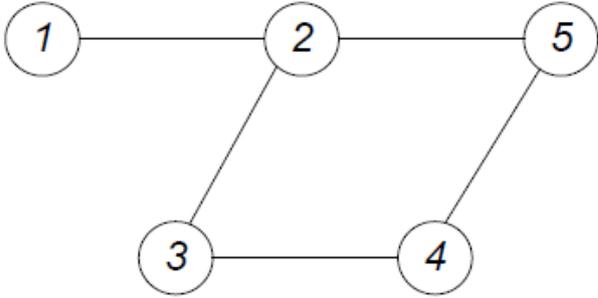
### Вопросы и задания

1. В чем состоит основная идея алгоритма Тэрри?
2. Какие правила перехода от вершины к вершине согласно алгоритму Тэрри вы знаете?
3. Определить путь обхода графов в соответствии с таблицей 1.2.1.

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

Таблица 1.2.1 – Варианты заданий

Но- мер вари- анта	Задание	Рисунок графа
1	обход в ши- рину из точки 3	
2	обход в ши- рину из точки 1	
3	обход в ши- рину из точки 5	
4	обход в глу- бину из точки 1	
5	обход в глу- бину из точки 2	

# Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.2.1

6	обход в ширину из точки 2		
7	обход в ширину из точки 4		
8	обход в глубину из точки 1		
9	обход в глубину из точки 3		
10	обход в глубину из точки 5		
11	обход в ширину из точки 1		
12	обход в ширину из точки 4		
13	обход в ширину из точки 5		
14	обход в глубину из точки 2		
15	обход в глубину из точки 6		

# Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.2.1

16	обход в ширину из точки 2	
17	обход в ширину из точки 6	
18	обход в глубину из точки 3	
19	обход в глубину из точки 5	
20	обход в глубину из точки 1	
21	обход в ширину из точки 1	
22	обход в ширину из точки 3	
23	обход в ширину из точки 4	
24	обход в глубину из точки 6	
25	обход в глубину из точки 2	

26	обход в ширину из точки 2	
27	обход в ширину из точки 5	
28	обход в глубину из точки 1	
29	обход в глубину из точки 3	
30	обход в глубину из точки 4	

### 1.3. Матроиды, жадные алгоритмы, алгоритм Краскала

#### Описание алгоритма

Матроидом  $M = \langle E, \varepsilon \rangle$  называется конечное множество  $E$ , число элементов которого равно  $n$  ( $|E| = n$ ), семейство его подмножеств  $\varepsilon$ , которое содержится в множестве всех подмножеств множества  $E$  ( $\varepsilon \subset 2^E$ ) такое, что выполняются следующие три аксиомы:

1.  $M_1$ : пустое множество принадлежит подмножеству  $\varepsilon$  ( $\emptyset \in \varepsilon$ );
2.  $M_2$ : если множество  $A$  принадлежит подмножеству  $\varepsilon$ , то и множество  $B$ , которое содержится в множестве  $A$ , тоже принадлежит подмножеству  $\varepsilon$ .

$$(A \in \varepsilon \ \& \ B \subset A \Rightarrow B \in \varepsilon);$$

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

3.  $M_3$ : если множества  $A$  и  $B$  принадлежат подмножеству  $\varepsilon$  и количество элементов множества  $A$  на один элемент меньше чем в множестве  $B$ , то существует элемент принадлежащий разности множеств  $B$  и  $A$  такой, что объединение его с множеством  $A$  будет принадлежать подмножеству  $\varepsilon$ .

$$(A, B \in \varepsilon \ \& \ |B| = |A| + 1 \Rightarrow \exists e \in B \setminus A \ A \cup \{e\} \in \varepsilon);$$

Элементы множества  $\varepsilon$  называются независимыми, а остальные подмножества  $E$  ( $2^E \setminus \varepsilon$ ) – зависимыми. Пусть множество  $X$  (произвольное множество) содержится в множестве  $E$ . Максимальным независимым подмножеством множества  $X$  называется множество  $Y$ , такое что если множество  $Y$ , принадлежащие независимому множеству  $\varepsilon$ , содержится в множестве  $X$  и  $Z$ , принадлежащем независимому множеству  $\varepsilon$ , содержится в множестве  $X$ , то множество  $Z$  содержится в множестве

□

$$Y(Y \subset X \ \& \ Y \in \varepsilon \ \& \ Z \subset X \Rightarrow Z \subset Y).$$

Множество максимальных независимых подмножеств множества  $X$  обозначим  $\underline{X}$ .

Рассмотрим следующее утверждение, справедливое для любого  $X$ :

$M_4$ : максимальные независимые подмножества данного множества равномощны

$$(Y \in \underline{X} \ \& \ Z \in \underline{X} \Rightarrow |Y| = |Z|).$$

Пусть  $M = \langle E, \varepsilon \rangle$  и выполнены аксиомы  $M_1$  и  $M_2$ . Тогда аксиома  $M_3$  и утверждение  $M_4$  эквивалентны, то есть  $M_1, M_2, M_4$  – эквивалентная система аксиом матроида.

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Пусть имеются конечное множество  $E$ , весовая функция  $\omega$  и семейство  $\varepsilon \subset 2^E$ . Необходимо выбрать в указанном семействе  $\varepsilon$  подмножество  $X$  наибольшего веса. Для решения этой задачи будем считать, что множество  $E$  упорядочено в порядке убывания весов элементов. В начале множество  $X$  пусто. Затем проверяем каждый элемент множества  $E$ , начиная с первого, если объединение его с множеством  $X$  принадлежит указанному семейству  $\varepsilon$ , то добавляем его в множество  $X$ , если не принадлежит семейству  $\varepsilon$ , то рассматриваем следующий элемент, и так до  $n$ -ого.

Алгоритм такого типа называется *жадным*. Очевидно, что жадный алгоритм является очень эффективным, он за наименьшее количество шагов находит искомое подмножество. Жадные алгоритмы и их свойства исследованы сравнительно недавно, но их значение в практике программирования чрезвычайно велико. Если удастся свести конкретную экстремальную задачу к такой постановке, где множество допустимых вариантов (из которых необходимо выбрать наилучший) является матроидом, то в большинстве случаев следует сразу применять жадный алгоритм, поскольку он достаточно эффективен в практическом смысле.

Если же наоборот, оказывается, что множество допустимых вариантов не образует матроида, то это «плохой признак». Скорее всего, данная задача окажется труднорешаемой. В этом случае целесообразно тщательно исследовать задачу для предварительного получения теоретических оценок сложности, чтобы избежать бесплодных попыток «изобрести» эффективный алгоритм там, где это на самом деле невозможно.

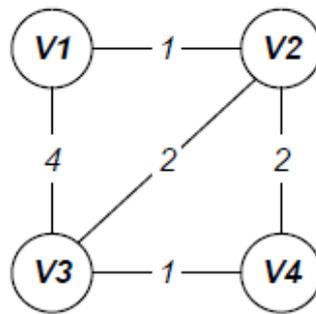
Другими словами, если  $M = \langle E, \varepsilon \rangle$  - матроид, то для любой весовой функции  $\omega$  жадный алгоритм находит независимое множество  $X$  с наибольшим весом; если же  $M = \langle E, \varepsilon \rangle$  не является матроидом, то существует такая функция  $\omega$ , что множество  $X$ , найденное жадным алгоритмом, не будет максимальным.

Пусть  $G(V, E)$  – граф. Основной подграф  $G(V, E)$  – это подграф, содержащий все вершины. Основной подграф являющийся деревом,

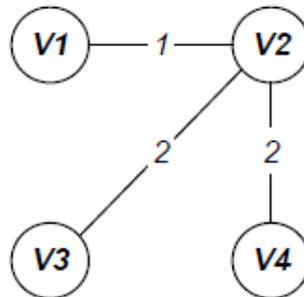
## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

называется *остовом*. Несвязный граф не имеет остова, связный граф может иметь много остовов. Если задать длины рёбер, то можно поставить задачу нахождения кратчайшего остова. Существует множество различных способов найти какой-то остов графа. Множество кратчайших путей из заданной вершины ко всем остальным тоже образует остов. Однако этот остов может не быть кратчайшим. На рисунке 1.3.1 а), б), в), г) показаны диаграммы графа, дерево кратчайших путей из вершины один с суммарным весом 5 и два кратчайших остова этого графа.

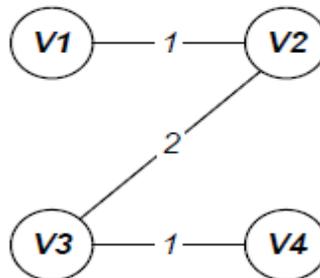
Одним из способов нахождения кратчайшего остова в связном графе является *алгоритм Краскала*. Он является частным случаем жадного алгоритма. Заметим, что множество подмножеств множества рёбер, не содержащих циклов, образует матроид.



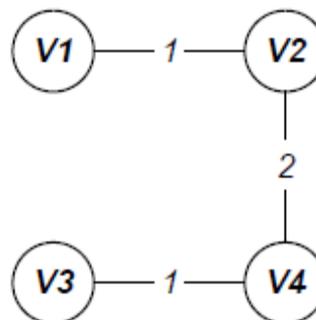
а)



б)

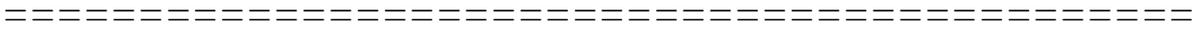


в)



г)

Рисунок 1.3.1 - Диаграммы графа, дерево кратчайших путей из вершины один с суммарным весом 5 и два кратчайших остова этого графа



*Алгоритм Краскала*

Алгоритм Краскала основан на выборе ребер графа, составляющих его наикратчайший остов. Изначально известен список ребер графа с их длинами. Выбор происходит по двум критериям:

Цикл проходит  $n = v - 1$  раз, где  $v$  – количество вершин графа.

Таким образом, при совершении нужного количества вышеуказанных действий мы будем иметь множество ребер, составляющих наикратчайший остов графа. Рассмотрим этот алгоритм на примере.

Пример 1.3.1

*Нахождение минимального остова графа*

Задан граф (Рисунок 1.3.2). Необходимо найти его минимальный остов при помощи алгоритма Краскала.

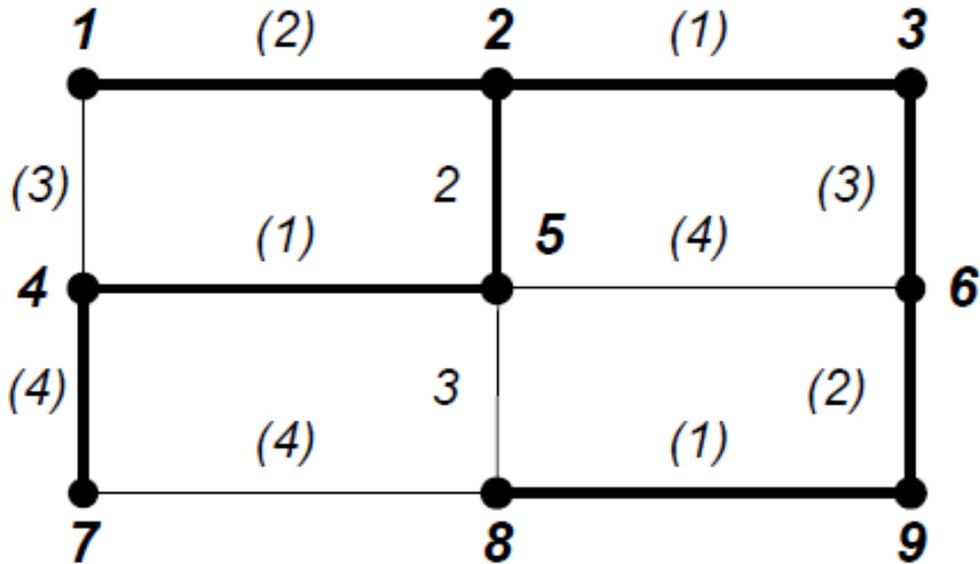


Рисунок 1.3.2 – Исходный граф

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

### Решение

Представим список ребер исходного графа с указанием длины каждого из них. Затем отсортируем ребра в порядке возрастания их длин.

Исходный список	Отсортированный список
(1,2) - 2	(2,3) - 1
(1,4) - 3	(4,5) - 1
(2,3) - 1	(8,9) - 1
(2,5) - 2	(1,2) - 2
(3,6) - 3	(2,5) - 2
(4,5) - 1	(6,9) - 2
(4,7) - 4	(1,4) - 3
(5,6) - 4	(3,6) - 3
(5,8) - 3	(5,8) - 3
(6,9) - 2	(4,7) - 4
(7,8) - 4	(5,6) - 4
(8,9) - 1	(7,8) - 4

Для решения задачи с помощью алгоритма Краскала составим таблицу (Рисунок 1.3.3), отражающую шаги выполнения алгоритма для рассматриваемого примера.

Шаг	Массив ребер	Компонента связности

Рисунок 1.3.3 – Таблица, отражающая шаги выполнения алгоритма

В втором столбце формируем массив ребер по принципу:

Шаг 1. Помещаем в первый столбец первое ребро отсортированного списка. Компонента связности данного ребра будет соответствовать вершинам, которые данное ребро связывает.

Шаг 2...Шаг N. Рассматриваем n-ое ребро отсортированного списка и:

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

1) добавляем ребро в массив, если оно удовлетворяет условию ацикличности;

2) пропускаем ребро, в противном случае.

И так далее до конца списка ребер.

Проверка ацикличности формируемого массива производится при помощи третьего столбца таблицы, в который помещаются компоненты связности «леса», разрастающегося во втором столбце.

Компоненты связности формируются следующим образом:

1) если вершины добавляемого ребра не входят ни в одну из уже существующих компонент связности, то формируется новая компонента связности из двух вершин (Рисунок 1.3.4).

Шаг	Массив ребер	Компонента связности
1	(2,3)	<span style="border: 1px solid black; padding: 2px;">2,3</span>
2	(2,3); (4,5)	<span style="border: 1px solid black; padding: 2px;">2,3</span> <span style="border: 1px solid black; padding: 2px;">4,5</span>
3	(2,3); (4,5); (8,9)	<span style="border: 1px solid black; padding: 2px;">2,3</span> <span style="border: 1px solid black; padding: 2px;">4,5</span> <span style="border: 1px solid black; padding: 2px;">8,9</span>

Рисунок 1.3.4 - Новая компонента связности из двух вершин

2) если одна из вершин добавляемого ребра входит в одну из имеющихся компонент связности, то ребро заносится в массив, соответствующий компоненте связности, а вторая вершина добавляемого ребра присоединяется к имеющейся компоненте (Рисунок 1.3.5).

Шаг	Массив ребер	Компонента связности
4	(2,3); (4,5); (8,9); (1,2)	<span style="border: 1px solid black; padding: 2px;">2, 3, 1</span> <span style="border: 1px solid black; padding: 2px;">4,5</span> <span style="border: 1px solid black; padding: 2px;">8,9</span>

Рисунок 1.3.5 - Ребро заносится в массив, соответствующий компоненте связности, а вторая вершина добавляемого ребра присоединяется к имеющейся компоненте

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

3) если одна вершина рассматриваемого ребра входит в одну из имеющихся компонент связности, а вторая – в другую, то компоненты связности и массивы ребер объединяются (Рисунок 1.3.6).

Шаг	Массив ребер	Компонента связности	
5	(2,3); (4,5); (8,9); (1,2); (2,5)	2, 3, 1, 4, 5	8,9

Рисунок 1.3.6 – Изменения в таблице алгоритма

4) если обе вершины рассматриваемого ребра принадлежат одной компоненте связности, то ребро исключается из рассмотрения и в таблицу изменения не вносятся. В рассматриваемом примере такими ребрами являются (1,4), (5,8), (5,6) и (7,8).

Продолжим решение задачи при помощи алгоритма Краскала, учитывая перечисленные условия (Рисунок 1.3.7).

Шаг	Массив ребер	Компонента связности	
6	(2,3); (4,5); (8,9); (1,2); (2,5); (6,9)	2, 3, 1, 4, 5	8, 9, 6
7	(2,3); (4,5); (8,9); (1,2); (2,5); (6,9) [(1,4)]	2, 3, 1, 4, 5	8, 9, 6
8	(2,3); (4,5); (8,9); (1,2); (2,5); (6,9); (3,6)	2, 3, 1, 4, 5, 8, 9, 6	
9	(2,3); (4,5); (8,9); (1,2); (2,5); (6,9); (3,6) [(5,8)]	2, 3, 1, 4, 5, 8, 9, 6	
10	(2,3); (4,5); (8,9); (1,2); (2,5); (6,9); (3,6); (4,7)	2, 3, 1, 4, 5, 8, 9, 6, 7	
11	(2,3); (4,5); (8,9); (1,2); (2,5); (6,9); (3,6); (4,7) [(5,6)]	2, 3, 1, 4, 5, 8, 9, 6, 7	
12	(2,3); (4,5); (8,9); (1,2); (2,5); (6,9); (3,6); (4,7) [(7,8)]	2, 3, 1, 4, 5, 8, 9, 6, 7	

Рисунок 1.3.7 - Изменения в таблице алгоритма Краскала

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

Примечание: в квадратных скобках указано рассматриваемое на данном шаге ребро, но не включаемое в массив ребер в силу выполнения условия 4).

В результате рассмотрения всех ребер отсортированного списка получим строку таблицы, определяющую минимальный остов графа (шаг 12).

Таким образом, минимальным остовом исходного графа является остов, состоящий из ребер (2,3); (1,2); (2,5); (4,5); (8,9); (6,9); (3,6); (4,7).

### Вопросы и задания

1. Что такое «Матроид»?
2. Какие алгоритмы называют «жадными»?
3. Что такое «остов»?
4. Какой способ нахождения кратчайшего остова в связном графе вы знаете?
5. На чем основан алгоритм Краскала?
6. Найти минимальный остов графа, заданного списком ребер, при помощи алгоритма Краскала:
  1. (1,2) – 3; (1,3) – 6; (1,4) – 2; (1,5) – 5; (2,3) – 4; (2,4) – 3; (2,5) – 1; (3,4) – 2; (3,5) – 5; (4,5) – 2;
  2. (1,2) – 6; (1,6) – 5; (1,8) – 5; (2,3) – 4; (3,4) – 1; (3,8) – 3; (4,5) – 1; (4,7) – 2; (5,6) – 2; (6,7) – 4; (7,8) – 1;
  3. (1,2) – 5; (1,3) – 3; (1,6) – 4; (2,3) – 6; (2,6) – 1; (3,4) – 2; (3,5) – 5; (4,5) – 1; (4,6) – 1; (5,6) – 4;
  4. (1,2) – 1; (1,3) – 7; (1,6) – 5; (2,3) – 4; (2,6) – 2; (3,4) – 3; (3,5) – 6; (3,6) – 1; (4,5) – 2; (4,6) – 5; (5,6) – 3;
  5. (1,2) – 1; (1,8) – 2; (2,3) – 2; (2,4) – 7; (2,8) – 3; (3,4) – 5; (4,5) – 5; (4,6) – 6; (5,6) – 3; (6,7) – 4; (6,8) – 1; (7,8) – 1;
  6. (1,2) – 4; (1,3) – 5; (1,4) – 2; (1,5) – 3; (2,3) – 1; (2,4) – 2; (2,5) – 3; (3,4) – 5; (3,5) – 6; (4,5) – 7;
  7. (1,2) – 4; (1,6) – 1; (1,8) – 6; (2,3) – 5; (3,4) – 2; (3,8) – 7; (4,5) – 2; (4,7) – 3; (5,6) – 3; (6,7) – 5; (7,8) – 5;

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

8.  $(1,2) - 1; (1,3) - 7; (1,6) - 5; (2,3) - 4; (2,6) - 2; (3,4) - 3; (3,5) - 6; (4,5) - 2; (4,6) - 5; (5,6) - 3;$
9.  $(1,2) - 5; (1,3) - 3; (1,6) - 4; (2,3) - 6; (2,6) - 1; (3,4) - 2; (3,5) - 5; (3,6) - 2; (4,5) - 1; (4,6) - 1; (5,6) - 4;$
10.  $(1,2) - 5; (1,8) - 1; (2,3) - 1; (2,4) - 3; (2,8) - 2; (3,4) - 1; (4,5) - 4; (4,6) - 5; (5,6) - 4; (6,7) - 6; (6,8) - 2; (7,8) - 2;$
11.  $(1,2) - 5; (1,3) - 3; (1,4) - 7; (1,5) - 2; (2,3) - 2; (2,4) - 1; (2,5) - 2; (3,4) - 4; (3,5) - 5; (4,5) - 2;$
12.  $(1,2) - 5; (1,6) - 2; (1,8) - 5; (2,3) - 3; (3,4) - 1; (3,8) - 2; (4,5) - 7; (4,7) - 2; (5,6) - 2; (6,7) - 4; (7,8) - 1;$
13.  $(1,2) - 2; (1,3) - 2; (1,6) - 4; (2,3) - 5; (2,6) - 1; (3,4) - 2; (3,5) - 5; (4,5) - 7; (4,6) - 1; (5,6) - 3;$
14.  $(1,2) - 3; (1,3) - 1; (1,6) - 4; (2,3) - 7; (2,6) - 2; (3,4) - 5; (3,5) - 4; (3,6) - 1; (4,5) - 3; (4,6) - 2; (5,6) - 1;$
15.  $(1,2) - 3; (1,8) - 2; (2,3) - 3; (2,4) - 1; (2,8) - 5; (3,4) - 2; (4,5) - 4; (4,6) - 4; (5,6) - 1; (6,7) - 7; (6,8) - 1; (7,8) - 5;$
16.  $(1,2) - 2; (1,3) - 4; (1,4) - 1; (1,5) - 1; (2,3) - 3; (2,4) - 5; (2,5) - 2; (3,4) - 3; (3,5) - 4; (4,5) - 6;$
17.  $(1,2) - 2; (1,6) - 3; (1,8) - 4; (2,3) - 4; (3,4) - 5; (3,8) - 6; (4,5) - 1; (4,7) - 2; (5,6) - 1; (6,7) - 3; (7,8) - 1;$
18.  $(1,2) - 3; (1,3) - 6; (1,6) - 3; (2,3) - 2; (2,6) - 5; (3,4) - 1; (3,5) - 4; (4,5) - 1; (4,6) - 1; (5,6) - 2;$
19.  $(1,2) - 2; (1,3) - 2; (1,6) - 4; (2,3) - 5; (2,6) - 1; (3,4) - 2; (3,5) - 5; (3,6) - 3; (4,5) - 7; (4,6) - 1; (5,6) - 3;$
20.  $(1,2) - 2; (1,8) - 1; (2,3) - 7; (2,4) - 2; (2,8) - 2; (3,4) - 1; (4,5) - 4; (4,6) - 5; (5,6) - 3; (6,7) - 5; (6,8) - 3; (7,8) - 4;$
21.  $(1,2) - 7; (1,3) - 1; (1,4) - 3; (1,5) - 5; (2,3) - 3; (2,4) - 2; (2,5) - 2; (3,4) - 4; (3,5) - 4; (4,5) - 1;$
22.  $(1,2) - 7; (1,6) - 3; (1,8) - 4; (2,3) - 1; (3,4) - 2; (3,8) - 1; (4,5) - 3; (4,7) - 2; (5,6) - 5; (6,7) - 4; (7,8) - 2;$
23.  $(1,2) - 3; (1,3) - 1; (1,6) - 4; (2,3) - 7; (2,6) - 2; (3,4) - 5; (3,5) - 4; (4,5) - 3; (4,6) - 2; (5,6) - 1;$
24.  $(1,2) - 2; (1,3) - 7; (1,6) - 4; (2,3) - 1; (2,6) - 3; (3,4) - 5; (3,5) - 6; (3,6) - 5; (4,5) - 4; (4,6) - 7; (5,6) - 2;$

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

25.  $(1,2) - 2; (1,8) - 3; (2,3) - 4; (2,4) - 7; (2,8) - 5; (3,4) - 7; (4,5) - 4; (4,6) - 6; (5,6) - 2; (6,7) - 1; (6,8) - 5; (7,8) - 5;$
26.  $(1,2) - 1; (1,3) - 2; (1,4) - 4; (1,5) - 5; (2,3) - 2; (2,4) - 3; (2,5) - 1; (3,4) - 4; (3,5) - 6; (4,5) - 7;$
27.  $(1,2) - 1; (1,6) - 2; (1,8) - 6; (2,3) - 2; (3,4) - 3; (3,8) - 7; (4,5) - 4; (4,7) - 1; (5,6) - 5; (6,7) - 4; (7,8) - 7;$
28.  $(1,2) - 2; (1,3) - 7; (1,6) - 4; (2,3) - 1; (2,6) - 3; (3,4) - 5; (3,5) - 6; (4,5) - 4; (4,6) - 7; (5,6) - 2;$
29.  $(1,2) - 3; (1,3) - 1; (1,6) - 4; (2,3) - 7; (2,6) - 2; (3,4) - 5; (3,5) - 4; (3,6) - 5; (4,5) - 3; (4,6) - 2; (5,6) - 1;$
30.  $(1,2) - 3; (1,8) - 2; (2,3) - 3; (2,4) - 1; (2,8) - 5; (3,4) - 2; (4,5) - 4; (4,6) - 4; (5,6) - 1; (6,7) - 7; (6,8) - 5; (7,8) - 5.$

### 1.4. Сетевые алгоритмы. Выбор кратчайшего пути

Задача о нахождении кратчайшего пути имеет столько практических применений и интерпретаций, что важность её не нуждается в обсуждении. Ниже рассматриваются классические алгоритмы, которые должен знать каждый программист.

В дальнейшем мы используем понятие сети, оно нуждается в некотором уточнении, так как нет единого подхода в его понимании. Под *сетью* подразумевается просто связный ориентированный граф  $D(V, E)$ , в котором, возможно, выделены вход и выход. Более узкое толкование термина «сеть» предполагает существование одного источника и одного стока, об этом, по мере необходимости мы будем говорить отдельно. Нужно отметить, что существуют и другие толкования термина. Отметим также, что в ориентированном графе  $D(V, E)$  как и в неориентированном  $G(V, E)$  используется название «вершина», а не узел.

=====

**Нахождение кратчайшего пути в сети без контуров**

*Описание алгоритма*

Этот алгоритм используется для решения задачи о нахождении кратчайшего пути в сети без контуров. Пусть задана сеть из  $n + 1$  вершины, то есть ориентированный граф. В нем выделены две вершины – вход (нулевая вершина) и выход (вершина с номером  $n$ ). Для каждой дуги заданы числа, называемые *длинами дуг*. *Длиной пути (контур)* называется сумма длин, входящих в него дуг (если длины дуг не заданы, то длина пути (контур) определяется как число входящих в него дуг).

Задача заключается в поиске кратчайшего пути (пути минимальной длины) от входа до выхода сети. Будем предполагать, что в любую вершину сети можно попасть из входа, и из любой вершины можно попасть в выход (вершины, не удовлетворяющие этому требованию, можно удалить). Известно что, для существования кратчайшего пути необходимо и достаточно отсутствия в сети контуров отрицательной длины.

Предположим, что в сети нет контуров. Тогда всегда можно пронумеровать вершины таким образом, что для любой дуги  $(i, j)$  имеет место  $j > i$ . Такая нумерация называется *правильной*. В сети без контуров всегда существует *правильная нумерация*.

Если изначально в задаче дается *неправильная (произвольная) нумерация* вершин, то прежде чем использовать метод потенциалов нужно определить *правильную нумерацию*, используя алгоритм топологической сортировки и алгоритм Уоршалла.

Для того чтобы понять принцип алгоритма Уоршалла рассмотрим пример.

Пример 1.4.1

*Алгоритм Уоршалла*

Алгоритм заключается в вычислении транзитивного замыкания. Пусть задан граф с произвольной нумерацией вершин (Рисунок 1.4.1).

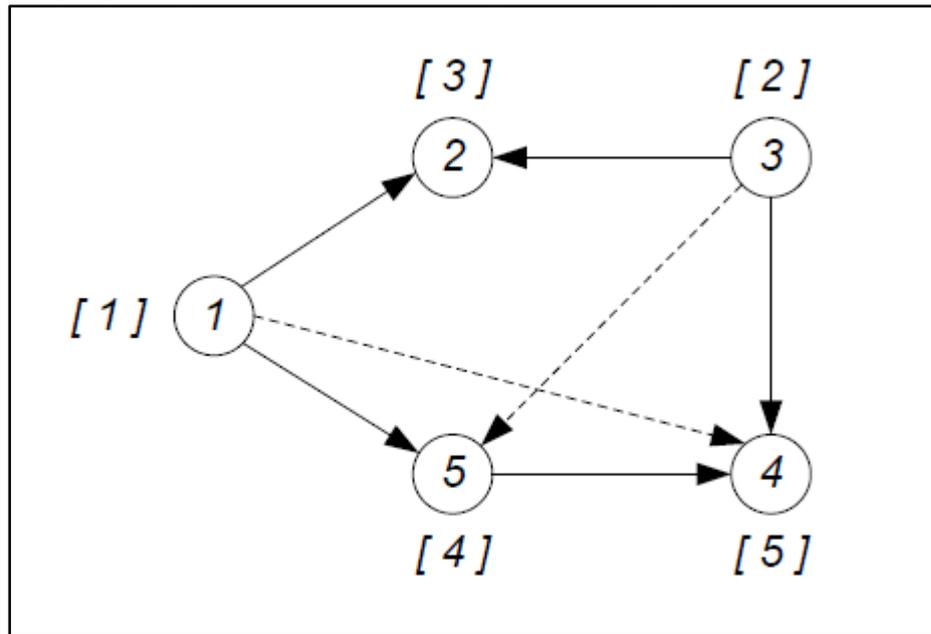


Рисунок 1.4.1 – Исходный граф

Составим для этого графа две матрицы смежности (Рисунки 1.4.2 и 1.4.3).

	1	2	3	4	5
1	0	1	0	0	1
2	0	0	0	0	0
3	0	1	0	1	0
4	0	0	0	0	0
5	0	0	0	1	0

Рисунок 1.4.2 - Матрица смежности для первоначального графа

	1	2	3	4	5
1	0	1	0	1	1
2	0	0	0	0	0
3	0	1	0	1	1
4	0	0	0	0	0
5	0	0	0	1	0

Рисунок 1.4.3 - Матрица смежности с транзитивным замыканием

Номер строки в матрице соответствует номеру исходящей вершины, а номер столбца – входящей. Единица на пересечении  $i$ -ой строки и  $j$ -го столбца ставится в том случае, если есть путь, выходящий из  $i$ -ой вершины и входящий в  $j$ -ую. Во все остальные ячейки матрицы заносятся нули.

Вторую матрицу с транзитивным замыканием можно построить с помощью исходной матрицы. Для этого сравниваем каждую строку с каждым столбцом, если в обоих есть хотя бы одна «1», то на их пересечении в ячейку заносится «1». При этом учитывается, что диагональ матрицы (из левого верхнего угла в правый нижний угол) всегда заполняется нулями, и что номер вершины, из которой идет дуга, должен быть меньше номера вершины, в которую она входит.

С помощью полученной матрицы смежности с транзитивным замыканием можно достроить на графе дополнительные дуги: добавленные в матрицу единицы обозначают новые дуги в графе (на рисунке 1.4.1 они обозначены пунктиром).

**Алгоритм топологической сортировки** – это алгоритм дополнения частичного порядка на конечном множестве. За основу берется матрица смежности, полученная алгоритмом Уоршалла. Рассматриваем в ней столбцы. Выбираем столбец, содержащий только нулевые

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

элементы, и условно его вычеркиваем вместе со строкой этого же номера. Номер этого столбца записываем, причем вариантов выкидывания может быть несколько из-за того, что может оказаться несколько таких столбцов. Остается матрица меньшая предыдущей на столбец и строку, с ней проделывается эта же операция, и так до тех пор, пока не останется один элемент, номер которого также записываем.

Каждый раз, записывая номер вычеркнутого столбца, получаем последовательность номеров вершин, которая идет не по порядку. Под ней выписываем номера по порядку.

$$\begin{array}{c} 1 - 3 - 2 - 5 - 4 \\ 1 - 2 - 3 - 4 - 5 \end{array}$$

Это означает, что номеру, произвольно поставленному у вершины, соответствует другой номер, полученный в результате топологической сортировки. Чтобы теперь получить правильную нумерацию, необходимо заменить первоначальные номера вершин на правильные (на рисунке 1.4.1 правильная нумерация дается в квадратных скобках).

Замечание: алгоритм Уоршела и топологическая сортировка выполнены верно, если после проставлении правильной нумерации на графе, номер вершины, из которой идет дуга, меньше номера вершины, в которую она входит.

После того, как будет выбрана правильная нумерация вершин, можно использовать метод потенциалов для нахождения кратчайшего пути на графе. Он заключается в следующих шагах:

Шаг 0: помечаем нулевую вершину (вход) индексом  $\lambda_0 = 0$ ; Шаг  $k$ : помечаем вершину  $k$  индексом  $\lambda_k = \min (\lambda_i + l_{ik})$ .

Здесь используются следующие обозначения:

- $l_{ij}$  – длина дуги  $(i, j)$ ;
- $\lambda_n$  – потенциал вершины (индекс выхода), равный длине кратчайшего пути.

Кратчайший путь здесь ищется по принципу оптимальности Беллмана. Он формулируется так: если ищется кратчайший путь между

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

двумя точками, то длина пути между любыми двумя точками кратчайшего пути также должна быть минимальна. Когда потенциалы установлены, кратчайший путь определяется методом обратного хода от выхода к входу, то есть кратчайшим является путь  $\mu = (0; i_1; i_2; \dots; i_{n-1}; n)$ , такой, что  $l_{i_{n-1};i_n} = \lambda_n - \lambda_{i_{n-1}}$  и т.д.

### Пример 1.4.2

#### ***Применение метода потенциалов для определения кратчайшего пути***

На рисунке 1.4.4 приведен пример применения метода потенциалов для определения кратчайшего пути.

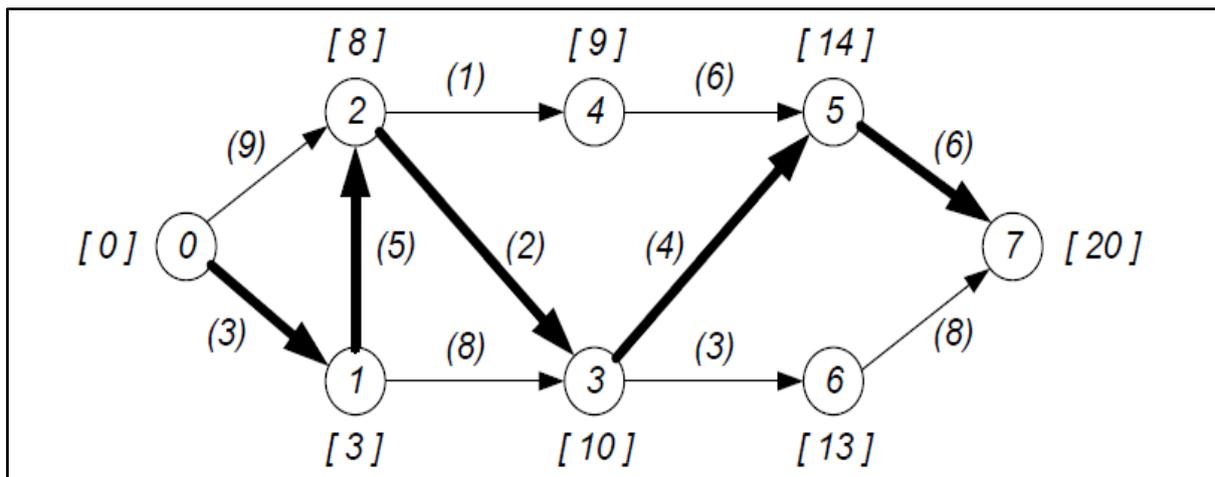


Рисунок 1.4.4 – Заданный граф примера 1.4.2

Задана сеть из 8 вершин, входом является вершина с номером 0, а выходом – вершина с номером 7. Числа у дуг равны длинам дуг. Нумерация вершин является правильной, т.к. для любой дуги  $(i, j)$  имеет место  $j > i$ .

Помечаем первую вершину (вход) индексом 0, т.е. потенциал этой вершины равен 0. Индексы вершин – потенциалы – на рисунке 35 помещены в квадратные скобки.

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

Далее определяем потенциал для первой вершины. В первую вершину можно попасть только из нулевой, поэтому складываем потенциал этой вершины с длиной дуги между ними (то есть  $[0]+3=[3]$ ). Потенциал первой вершины равен 3.

В отличие от первой во вторую вершину можно попасть либо из нулевой вершины, либо из первой. Определяем длины этих путей. В первом случае он будет равен 9 ( $[0]+9$ ), а во втором – 8 (потенциал первой вершины + длина дуги между первой и второй вершиной, т.е.  $[3]+5$ ). Из двух путей выбираем наименьший. Таким образом, потенциал второй вершины будет равен 8.

Определяя потенциал третьей вершины, рассматриваем пути только от ближайших вершин, то есть от первой и второй вершин. В первом случае путь равен 11, а во втором – 10. Соответственно потенциал 3-ей вершины равен 10.

Далее аналогичным образом расставляем потенциалы всех остальных вершин в графе. После того как все потенциалы расставлены, методом обратного хода устанавливаем кратчайший путь. На рисунке он выделен жирными линиями.

### *Описание алгоритма*

Предположим, что вершина 1 является узлом-входом и требуется найти длины кратчайших путей от вершины 1 до каждой другой вершины графа. Для этого алгоритма дуговые расстояния могут быть как положительными, так и отрицательными, но не должно быть циклов отрицательной длины. Предположим, что если в графе отсутствует та или иная дуга, то её вес равен бесконечно большому числу. Основная идея алгоритма Беллмана – Форда состоит в том, чтобы сначала найти длины кратчайших путей, при условии, что пути содержат не более двух дуг и т.д. Кратчайший путь при условии, что он содержит не более  $h$  дуг, будем называть кратчайшим путем в графе. Согласно идее Бел-

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

лмана-Форда, Р. Прим предложил алгоритм нахождения длин кратчайших путей, ведущих из произвольной вершины графа во все остальные его вершины (в которые есть пути из вершины-входа), состоящий в присвоении вершинам некоторых потенциалов. Признаком окончания алгоритма является остановка процесса изменения потенциалов.

Алгоритм, предложенный Р. Примом, состоит в выполнении следующих операций:

1. вершине-входу присваиваем потенциал  $\varphi_1 = 0$ ; остальным вершинам – потенциал  $\varphi_i = \infty$ ,  $i = 1 \dots n$ , где  $n$  - число вершин в рассматриваемом графе. для каждой вершины  $i$ , смежной с вершиной  $j$  из массива вершин  $L$ , находим  $\varphi_j + l(j, i)$  и проверяем неравенство:

$$\varphi_j + l(j, i) < \varphi_i$$

если неравенство выполняется, то

$$\varphi_i := \varphi_j + l(j, i)$$

Вершину  $i$  заносим в массив  $L_1$ .

3. проверяем массив  $L_1 \neq \emptyset$ . Если неравенство выполняется  $L := L_1$  и осуществляем переход к пункту 2.

4. конец алгоритма нахождения кратчайших путей. Рассмотрим работу алгоритма на примере.

### Пример 1.4.3

#### *Алгоритм Беллмана – Форда нахождения кратчайших путей*

Допустим, имеется граф (Рисунок 1.4.5), представленный списком дуг (Рисунок 1.4.6).

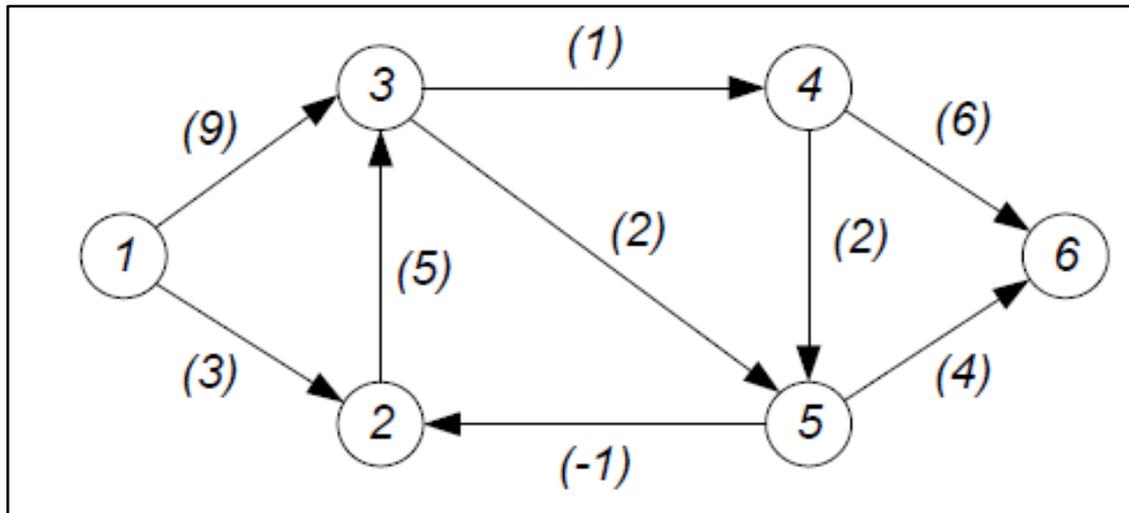


Рисунок 1.4.5 – Исходный граф

(1,2) – 3	(4,6) – 6
(1,3) – 9	(3,5) – 2
(2,3) – 5	(5,2) – -1
(3,4) – 1	(5,6) – 4
(4,5) – 2	(6,0) – 0

Рисунок 1.4.6 – Список дуг исходного графа

Решение:

Результаты работы алгоритма по шагам вычислений сведем в таблицу (Рисунок 1.4.7).

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

№ шага вычислений	Массив $L$	Потенциалы вершин						Массив $L_1$
		1	2	3	4	5	6	
0	$\emptyset$	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	1
1	1	0	3	9	$\infty$	$\infty$	$\infty$	2, 3
2	2, 3	0	3	8	10	11	$\infty$	3, 4, 5
3	3, 4, 5	0	3	8	9	10	15	4, 5, 6
4	4, 5, 6	0	3	8	9	10	14	6
5	6	0	3	8	9	10	14	$\emptyset$

Рисунок 1.4.7 - Результаты работы алгоритма по шагам вычислений

На нулевом шаге производится инициализация согласно пункту 1 алгоритма. Далее на первом шаге вычислений рассматриваются вершины, смежные с вершиной-входом, потенциалы рассматриваемых вершин становятся равными длине соединяющей их дуги, если таковое существует, в противном случае – потенциал вершины остается равным  $\infty$ .

На третьем шаге рассматриваем вершины, достижимые из вершины входа при помощи пути, состоящего из трех дуг. Таковыми являются:

- 1) 1 – 3 – 4 – 6: потенциал вершины 6 изменяется ( $\varphi_6 = 16$ );
- 2) 1 – 2 – 3 – 5: потенциал вершины 5 изменяется ( $\varphi_5 = 10$ );
- 3) 1 – 3 – 5 – 2: потенциал вершины 2 не изменится ( $\varphi_2 < 10$ );
- 4) 1 – 3 – 4 – 5: потенциал вершины 5 не изменится ( $\varphi_5 < 12$ );
- 5) 1 – 3 – 5 – 6: потенциал вершины 6 изменится ( $\varphi_6 = 15$ ).

На четвертом шаге вычислений рассматриваем вершины, достижимые из вершины-входа за путь, состоящий из четырех ребер. Таковыми являются пути:

- 1) – 2 – 3 – 5 – 6: потенциал вершины 6 изменится ( $\varphi_6 = 14$ );
- 2) – 2 – 3 – 4 – 5: потенциал вершины 5 не изменится

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

- ( $\varphi_5 < 11$ );
- 3) 1 – 2 – 3 – 5 – 2: потенциал вершины 2 не изменится ( $\varphi_2 < 9$ );
- 4) 1 – 3 – 5 – 2 – 3: потенциал вершины 3 не изменится ( $\varphi_3 < 15$ );
- 5) 1 – 3 – 4 – 5 – 2: потенциал вершины 2 не изменится ( $\varphi_2 < 11$ ).

На пятом шаге вычислений рассматриваются вершины, достижимые из вершины-входа за путь, состоящий из пяти дуг:

- 1) 1 – 3 – 5 – 2 – 3 – 5: потенциал вершины 5 не изменится ( $\varphi_5 < 17$ );
- 2) 1 – 3 – 5 – 2 – 3 – 4: потенциал вершины 4 не изменится ( $\varphi_4 < 16$ );
- 3) 1 – 3 – 4 – 5 – 2 – 3: потенциал вершины 3 не изменится ( $\varphi_3 < 14$ );
- 4) 1 – 2 – 3 – 4 – 5 – 2: потенциал вершины 2 не изменится ( $\varphi_2 < 10$ );
- 5) 1 – 2 – 3 – 4 – 5 – 6: потенциал вершины 6 не изменится ( $\varphi_6 = 14$ );
- 6) 1 – 2 – 3 – 5 – 2 – 3: потенциал вершины 3 не изменится ( $\varphi_3 < 14$ ).

На пятом шаге вычислений не произошло изменений ни в одном из потенциалов вершин графа, делаем вывод об окончании работы алгоритма.

Восстановим кратчайший путь из вершины 6 в вершину 1.

Для этого для каждой вершины  $j$  определим такую вершину  $i$ , что будет выполняться правило:

$$\varphi_i := \varphi_j + l(j, i)$$

Следуя правилу, последовательно находим кратчайший путь:

вершина 6:  $i = 5$ ;

вершина 5:  $i = 3$ ;

вершина 3:  $i = 2$ ;

вершина 2:  $i = 1$ .

Кратчайшим путем является путь  $1 - 2 - 3 - 5 - 6$  (Рисунок 1.4.8).

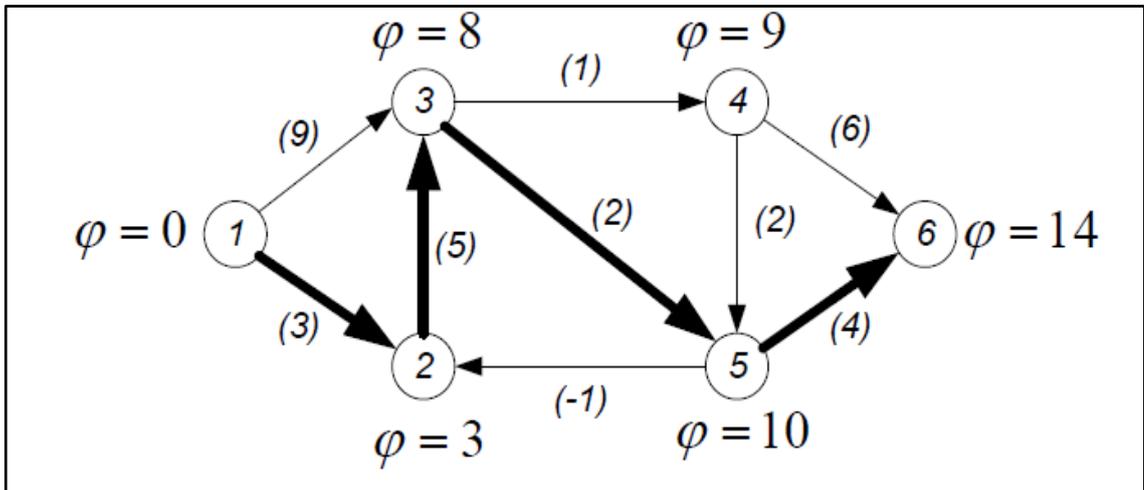


Рисунок 1.4.8 - Кратчайший путь

### Алгоритм Дейкстра

#### Описание алгоритма

Алгоритм Дейкстра требует, чтобы длины всех дуг были положительны. Объем вычислений в худшем случае для этого алгоритма значительно меньше, чем у алгоритма Беллмана – Форда. Основная его идея состоит в том, чтобы отыскивать кратчайшие пути в порядке возрастания длины пути.

Кратчайшим среди всех кратчайших путей от вершины-входа является путь, состоящий из одной дуги, соединяющий вершину-вход с ближайшим соседним узлом, так как любой путь, состоящий из нескольких дуг, будет всегда длиннее первой дуги вследствие предположения о положительности всех дуговых длин. Следующим кратчайшим среди кратчайших путей должен быть ли путь из одной дуги к следующему ближайшему соседу вершины-входа, либо кратчайший

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====  
путь из двух дуг, проходящий через вершину, выбранный на первом шаге и т.д.

Алгоритм Дейкстра состоит в выполнении следующих операций:

Шаг 0. Помечаем нулевую вершину индексом  $\lambda_0 = 0$ ;

Шаг  $k$ :

Пусть уже помечено некоторое количество вершин. Обозначим  $Q$  – множество непомеченных вершин, смежных с помеченными. Для каждой вершины  $k$  принадлежащей  $Q$  вычисляем величину

$$\xi_k = \min_k (\lambda_k + l_{ki}),$$

где минимум берется по всем помеченным вершинам  $i$ , смежным с вершиной  $k$ .

Помечаем вершину  $k$ , для которой величина  $\xi_k$  минимальна, индексом  $\lambda_k = \xi_k$ .

Подобную процедуру повторяем до тех пор, пока не будет помечена вершина  $n$ .

Длина кратчайшего пути равна  $\lambda_n$ , а сам кратчайший путь определяется так, как это было описано выше.

### Пример 1.4.4

#### *Алгоритм Дейкстра нахождения кратчайших путей*

Найдем кратчайший путь из вершины 1 в вершину 7 (Рисунок 1.4.9).

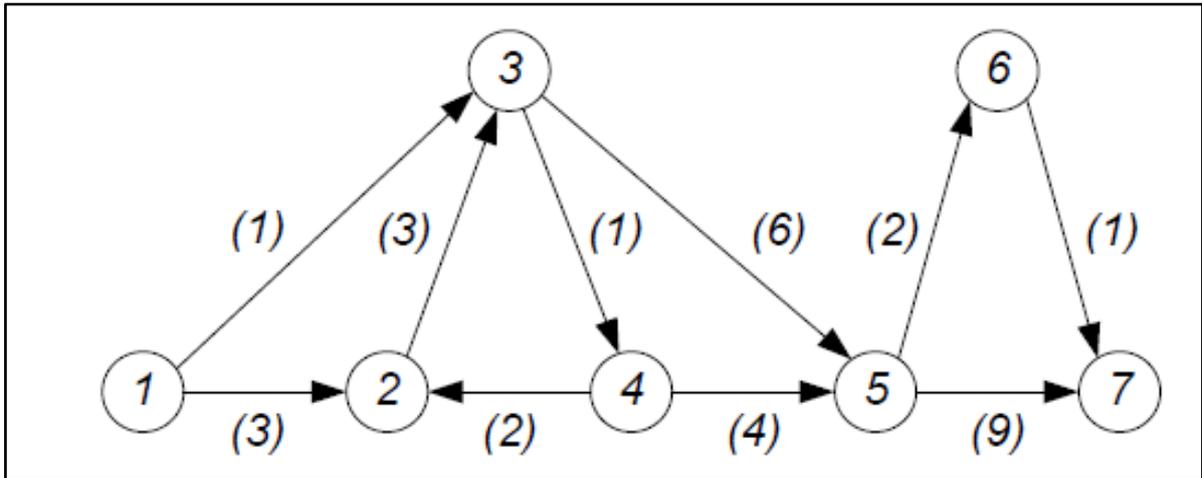


Рисунок 1.4.9 – Исходный граф

- 1) Пометим вершину 1 индексом  $\lambda_1 = 0$ .
- 2) Смежными с 1 являются вершины 2 и 3.  $\zeta_2 = 0 + 3 = 3$ .  $\zeta_3 = 0 + 1 = 1$ .
- 3) Величина  $\zeta_3$  минимальна. Помечаем вершину 3 -  $\lambda_3 = 1$ . Помечены вершины 1 и 3.
- 4) Смежными с помеченными вершинами (1 и 3) являются 2, 4 и 5.  $\zeta_2 = 0 + 3 = 3$ .  $\zeta_4 = 1 + 1 = 2$ .  $\zeta_5 = 1 + 6 = 7$ . Величина  $\zeta_4$  минимальна.
- 5) Помечаем вершину индексом  $\lambda_4 = 2$ . Помечены вершины 1, 3 и 4.
- 6) Смежными с помеченными вершинами (1, 3 и 4) являются 2 и 5.
- 7)  $\zeta_5 = 2 + 4 = 6$ .  $\zeta_5 = 1 + 6 = 7$ .  $\zeta_2 = 0 + 3 = 3$ .  $\zeta_2 = 2 + 2 = 4$ . Помечаем вершину  $\lambda_2 = 3$ . Помечены вершины 1, 2, 3, 4.
- 8) Смежной с помеченными вершинами (1, 2, 3, 4) является вершина 5.  $\zeta_5 = 2 + 4 = 6$ .  $\zeta_5 = 1 + 6 = 7$ . Помечаем 5 индексом  $\lambda_5 = 6$ . Помечены вершины 1, 2, 3, 4 и 5.
- 9) Смежными с помеченными вершинами (1, 2, 3, 4, 5) являются 6 и 7.  $\zeta_6 = 6 + 2 = 8$ .  $\zeta_7 = 6 + 9 = 15$ . Величина  $\zeta_6$  минимальна для вершины 6. Помечаем вершину  $\lambda_6 = 8$ .

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

- 10) Смежная с помеченными вершинами (1, 2, 3, 4, 5, 6) – вершина 7.  $\xi_7 = 6 + 9 = 15$ .  $\xi_7 = 8 + 1 = 9$ . Помечаем вершину индексом  $\lambda_7 = 9$ .
- 11) Для 6 вершины смежной является вершина 7.  $\xi_7 = 8 + 1 = 9$ . Величина  $\xi_7$  минимальна. Помечаем вершину 7 индексом  $\lambda_7 = 9$ .

С помощью алгоритма Дейкстра мы получили длину кратчайшего пути из вершины 1 в вершину 7, которая составит 9.

Кратчайшим путем является путь 1 – 3 – 4 – 5 – 6 – 7 (Рисунок 1.4.10).

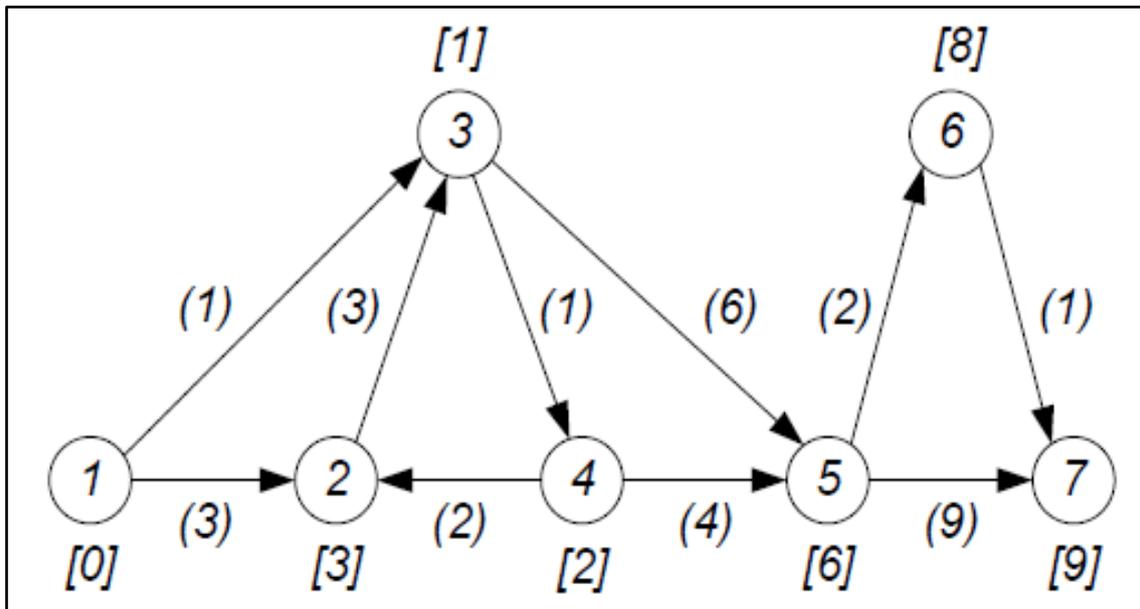


Рисунок 1.4.10 - Кратчайшим путем является путь  
1 – 3 – 4 – 5 – 6 – 7

=====

**Алгоритм Флойда – Уоршалла**

***Описание алгоритма***

Алгоритм Флойда находит кратчайшие пути между всеми парами вершин в (ор) графе. В отличие от алгоритма Дейкстра и алгоритма Беллмана – Форда находит кратчайшие пути сразу для всех пар вершин. Как и в алгоритме Беллмана – Форда, дуговые расстояния могут быть как положительными, так и отрицательными, но также не должно быть циклов отрицательной длины. Во всех трех алгоритмах решение находится методом итераций, но в каждом алгоритме итерируются разные величины. Если в алгоритме Беллмана – Форда итерируется число дуг в пути, в алгоритме Дейкстра – длина пути, то в алгоритме Флойда – Уоршалла итерируется множество вершин, которые допускается иметь в качестве промежуточных узлов на путях. Как и оба других алгоритма, алгоритм Флойда – Уоршалла начинает с расстояний одной дуги (то есть без промежуточных узлов), выбранных в качестве исходных оценок для длин кратчайших путей. Затем вычисляются кратчайшие пути с тем ограничением, что промежуточной вершиной может быть только вершина 1, затем с ограничением, что промежуточными вершинами могут быть только вершины 1 и 2 и т.д.

Этот алгоритм в его «чистом» варианте абсолютно не подходит для нахождения кратчайших путей вручную, так как он весьма трудоёмок (скажем, для графа  $4 \times 4$  мы имеем уже 64 итерации). Поэтому следует понять, на каких итерациях матрица не меняется и затем попросту исключить их из рассмотрения.

Пример 1.4.5

***Алгоритм Беллмана – Форда нахождения кратчайшего пути в орграфе***

Анализируется граф, представленный рисунком 1.4.11.

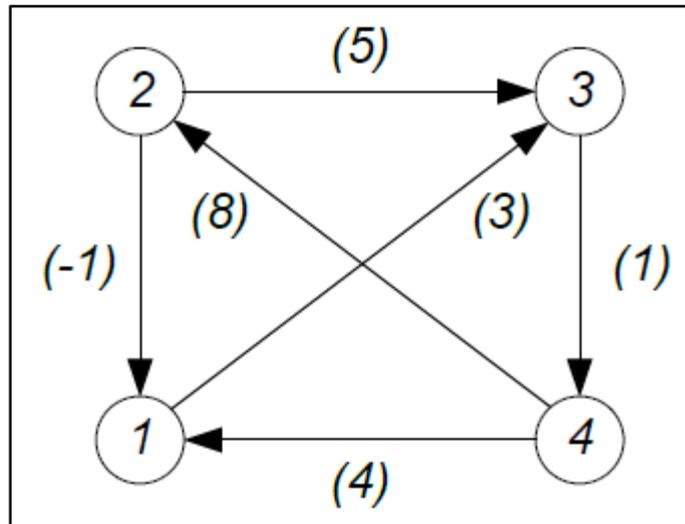


Рисунок 1.4.11 – Анализируемый граф

Условно процедуру нахождения кратчайшего пути в орграфе можно разбить на несколько шагов.

- 1) Инициализация. На данном этапе мы формируем матрицу  $C$   $[1..p, 1..p]$  и «предварительный вариант» матрицы  $H$ .

Элемент матрицы  $C_{ij}$  равен длине дуги из вершины  $i$  в вершину  $j$ , а если такой дуги нет, то он равен бесконечности. Соответствующие элементы матрицы  $H$  равны вершине, в которую проведена дуга, длина которой указана в матрице  $C$ :

$$T_{ij} = C_{ij} = \begin{pmatrix} 0 & \infty & 3 & \infty \\ -1 & 0 & 5 & \infty \\ \infty & \infty & 0 & 1 \\ 4 & 8 & \infty & 0 \end{pmatrix};$$

=====

$$H_{ij} = \begin{pmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \\ 1 & 2 & 0 & 0 \end{pmatrix}$$

- 2) «Вычёркиваем» первый столбец и первую строку матрицы  $T$ . Затем ищем в данной строке и данном столбце  $\infty$  и «вычёркиваем» соответственно столбец или строку, где  $\infty$  была найдена. Видим, что остаётся два элемента: 5 и  $\infty$  (Рисунок 1.4.12).

	0	$\infty$	3	$\infty$
$T_{ij}$	-1	0	5	$\infty$
	$\infty$	$\infty$	0	1
	4	8	$\infty$	0

Рисунок 1.4.12 - «Вычёркиваем» первый столбец и первую строку матрицы  $T$

- 3) Проверяем оставшиеся элементы  $t_{ij}$  на неравенство  $t_{ij} > t_{lj} + t_{il}$ . Если оно выполняется, то записываем на место проверяемого элемента сумму  $t_{lj} + t_{il}$  и параллельно пишем в соответствующую ячейку матрицы  $H$  номер вершины из её первого столбца, а если нет, то оставляем элемент без изменений (Рисунок 1.4.13).

$$T_{ij} = \begin{pmatrix} 0 & \infty & 3 & \infty \\ -1 & 0 & 2 & \infty \\ \infty & \infty & 0 & 1 \\ 4 & 8 & 7 & 0 \end{pmatrix}$$

$$H_{ij} = \begin{pmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 4 \\ 1 & 2 & 1 & 0 \end{pmatrix}$$

Рисунок 1.4.13 - Записываем на место проверяемого элемента сумму  $t_{ij} + t_{il}$  и параллельно пишем в соответствующую ячейку матрицы  $H$  номер вершины из её первого столбца

- 4) Повторяем шаги 1-3 для второго столбца и второй строки, третьего столбца и третьей строки, и т. д. до  $p$ -го столбца и  $p$ -ой строки (Таблица 1.4.1).

Таблица 1.4.1 - Повторяем шаги 1-3

$i$	Исходные данные на $i$ -м шаге	Результат на $i$ -м шаге
2	$T_{ij} = \begin{pmatrix} 0 & \infty & 3 & \infty \\ -1 & 0 & 2 & \infty \\ \infty & \infty & 0 & 1 \\ 4 & 8 & 7 & 0 \end{pmatrix}$	$H_{ij} = \begin{pmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 4 \\ 1 & 2 & 1 & 0 \end{pmatrix}$

3	$T_{ij} = \begin{pmatrix} 0 & \infty & 3 & \infty \\ -1 & 0 & 2 & \infty \\ \infty & \infty & 0 & 1 \\ 4 & 8 & 7 & 0 \end{pmatrix} \rightarrow$ $H_{ij} = \begin{pmatrix} 0 & 0 & \textcircled{3} & \boxed{0} \\ 1 & 0 & \textcircled{1} & \boxed{0} \\ 0 & 0 & 0 & 4 \\ 1 & 2 & 1 & 0 \end{pmatrix} \rightarrow$	$T_{ij} = \begin{pmatrix} 0 & \infty & 3 & \boxed{4} \\ -1 & 0 & 2 & \boxed{3} \\ \infty & \infty & 0 & 1 \\ 4 & 8 & 7 & 0 \end{pmatrix}$ $H_{ij} = \begin{pmatrix} 0 & 0 & \textcircled{3} & \boxed{3} \\ 1 & 0 & \textcircled{1} & \boxed{1} \\ 0 & 0 & 0 & 4 \\ 1 & 2 & 1 & 0 \end{pmatrix}$
4	$T_{ij} = \begin{pmatrix} 0 & \boxed{\infty} & 3 & 4 \\ -1 & 0 & 2 & 3 \\ \boxed{\infty} & \boxed{\infty} & 0 & 1 \\ 4 & 8 & 7 & 0 \end{pmatrix} \rightarrow$ $H_{ij} = \begin{pmatrix} 0 & \boxed{0} & 3 & \textcircled{3} \\ 1 & 0 & 1 & \textcircled{1} \\ \boxed{0} & \boxed{0} & 0 & 4 \\ 1 & 2 & 1 & 0 \end{pmatrix} \rightarrow$	$T_{ij} = \begin{pmatrix} 0 & \boxed{12} & 3 & 4 \\ -1 & 0 & 2 & 3 \\ \boxed{5} & \boxed{9} & 0 & 1 \\ 4 & 8 & 7 & 0 \end{pmatrix}$ $H_{ij} = \begin{pmatrix} 0 & \boxed{3} & 3 & \textcircled{3} \\ 1 & 0 & 1 & \textcircled{1} \\ \boxed{4} & \boxed{4} & 0 & 4 \\ 1 & 2 & 1 & 0 \end{pmatrix}$

- 5) Таким образом, на выходе мы получаем, как и указано в алгоритме, две матрицы:

$$T_{ij} = \begin{pmatrix} 0 & 12 & 3 & 4 \\ -1 & 0 & 2 & 3 \\ 5 & 9 & 0 & 1 \\ 4 & 8 & 7 & 0 \end{pmatrix}$$

$$H_{ij} = \begin{pmatrix} 0 & 3 & 3 & 3 \\ 1 & 0 & 1 & 1 \\ 4 & 4 & 0 & 4 \\ 1 & 2 & 1 & 0 \end{pmatrix}.$$

Каждый элемент  $t_{ij}$  матрицы  $T$  содержит длину кратчайшего пути из вершины  $i$  в вершину  $j$ , а каждый элемент  $h_{ij}$  матрицы  $H$  содержит номер первой вершины, проходимой нами при следовании по этому пути.

### Вопросы и задания

1. В чем особенность сетевых алгоритмов?
2. В чем заключается алгоритм нахождения кратчайшего пути в сети без контуров?
3. Что называется длиной пути (контур)?
4. В чем заключается алгоритм Уоршалла?
5. В чем особенность алгоритма топологической сортировки?
6. Какие шаги необходимо выполнить, применяя метод потенциалов для нахождения кратчайшего пути на графе?

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

7. По какому принципу ищется кратчайший путь при использовании метода потенциалов?
8. В чем состоит основная идея алгоритма Беллмана – Форда?
9. В выполнении каких операций состоит алгоритм, предложенный Р. Примом?
10. В чем состоит основная идея алгоритма Дейкстры?
11. В выполнении каких операций состоит алгоритм, предложенный Дейкстра?
12. Определить кратчайший путь в графе. Варианты заданий представлены в таблице 1.4.2.

Таблица 1.4.2 - Варианты заданий

Номер варианта	Рисунок
1	<pre> graph LR     0((0)) -- 5 --&gt; 3((3))     0 -- 6 --&gt; 4((4))     0 -- 1 --&gt; 2((2))     3 -- 4 --&gt; 2     4 -- 3 --&gt; 1((1))     2 -- 7 --&gt; 1             </pre>
2	<pre> graph LR     0((0)) -- 1 --&gt; 1((1))     0 -- 9 --&gt; 2((2))     1 -- 4 --&gt; 3((3))     1 -- 7 --&gt; 2     2 -- 5 --&gt; 4((4))     3 -- 2 --&gt; 4             </pre>

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.2

3	<pre> graph LR     1((1)) -- 4 --&gt; 0((0))     1((1)) -- 8 --&gt; 3((3))     3((3)) -- 6 --&gt; 0((0))     3((3)) -- 5 --&gt; 2((2))     0((0)) -- 3 --&gt; 4((4))     2((2)) -- 7 --&gt; 4((4))         </pre>
4	<pre> graph LR     0((0)) -- 8 --&gt; 2((2))     0((0)) -- 3 --&gt; 4((4))     2((2)) -- 2 --&gt; 1((1))     2((2)) -- 6 --&gt; 3((3))     3((3)) -- 9 --&gt; 1((1))     3((3)) -- 4 --&gt; 4((4))         </pre>
5	<pre> graph LR     2((2)) -- 1 --&gt; 1((1))     2((2)) -- 8 --&gt; 3((3))     2((2)) -- 6 --&gt; 0((0))     1((1)) -- 5 --&gt; 4((4))     3((3)) -- 4 --&gt; 0((0))     0((0)) -- 3 --&gt; 4((4))         </pre>
6	<pre> graph LR     0((0)) -- 7 --&gt; 1((1))     0((0)) -- 6 --&gt; 4((4))     1((1)) -- 9 --&gt; 3((3))     1((1)) -- 3 --&gt; 4((4))     2((2)) -- 2 --&gt; 4((4))     3((3)) -- 4 --&gt; 2((2))         </pre>

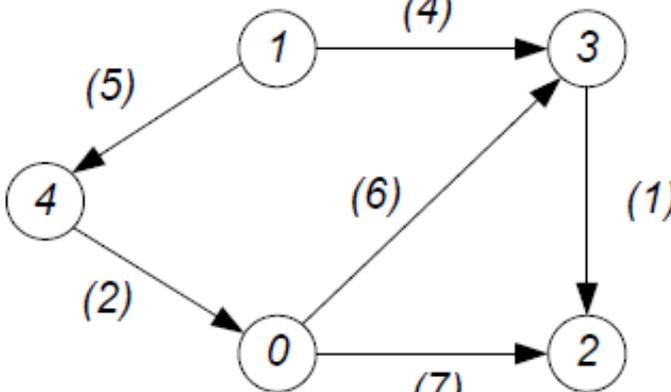
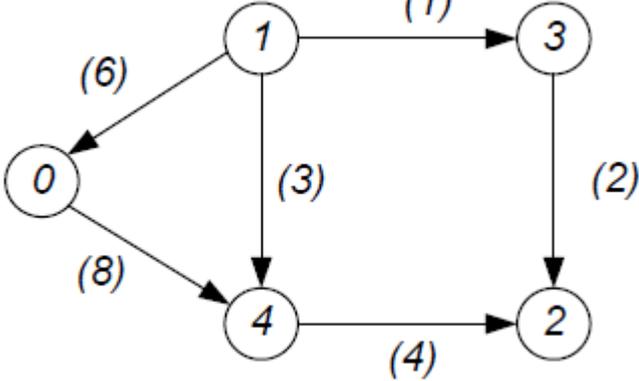
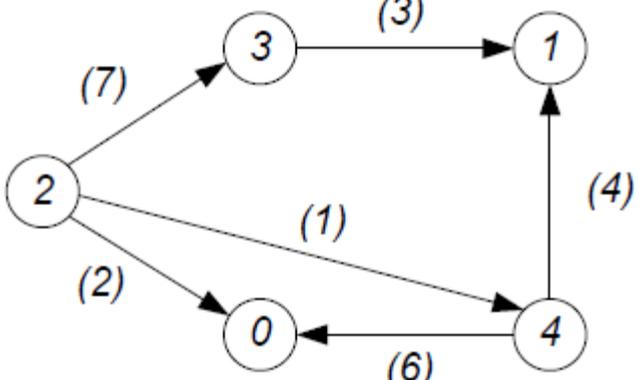
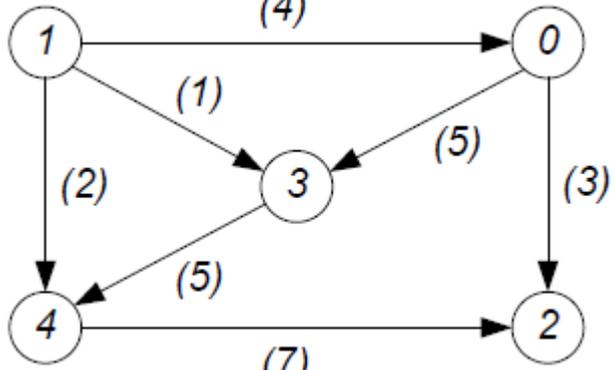
Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.2

	<pre> graph TD     1((1)) -- 4 --&gt; 2((2))     2 -- 2 --&gt; 4((4))     1 -- 8 --&gt; 3((3))     3 -- 3 --&gt; 4     3 -- 9 --&gt; 0((0))     4 -- 6 --&gt; 0             </pre>
8	<pre> graph TD     0((0)) -- 5 --&gt; 4((4))     2((2)) -- 9 --&gt; 4     2 -- 7 --&gt; 1((1))     1 -- 3 --&gt; 4     1 -- 9 --&gt; 3((3))     3 -- 6 --&gt; 4             </pre>
9	<pre> graph TD     1((1)) -- 3 --&gt; 4((4))     1 -- 7 --&gt; 3((3))     3 -- 4 --&gt; 4     3 -- 6 --&gt; 0((0))     4 -- 2 --&gt; 2((2))     0 -- 5 --&gt; 2             </pre>

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.2

10	
11	
12	
13	

4	
15	
16	
17	

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.2

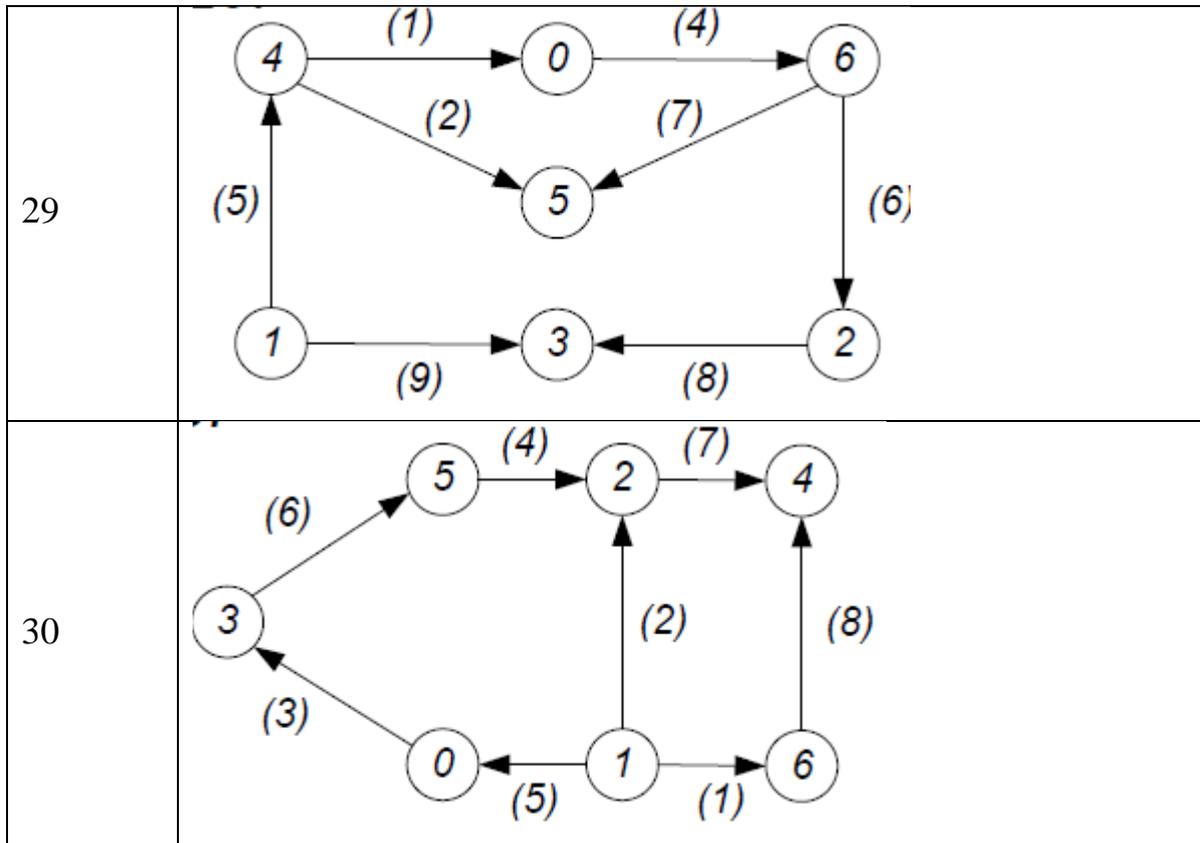
18	<p>Diagram 18: Directed graph with nodes 0, 1, 2, 3, 4, 5, 6. Edges and weights: (0,1) weight 1, (1,2) weight 5, (2,4) weight 6, (4,0) weight 9, (0,5) weight 7, (5,3) weight 3, (3,1) weight 4, (1,6) weight 2, (6,2) weight 2, (3,6) weight 8.</p>
19	<p>Diagram 19: Directed graph with nodes 0, 1, 2, 3, 4, 5, 6. Edges and weights: (0,1) weight 1, (1,4) weight 8, (4,6) weight 2, (6,0) weight 9, (0,5) weight 6, (5,2) weight 2, (2,3) weight 4, (3,1) weight 5.</p>
20	<p>Diagram 20: Directed graph with nodes 0, 1, 2, 3, 4, 5, 6. Edges and weights: (0,2) weight 6, (2,4) weight 6, (4,6) weight 2, (6,5) weight 7, (5,1) weight 3, (1,3) weight 5, (3,0) weight 4, (3,4) weight 1.</p>
21	<p>Diagram 21: Directed graph with nodes 0, 1, 2, 3, 4, 5, 6. Edges and weights: (0,1) weight 6, (0,3) weight 5, (3,4) weight 1, (4,2) weight 3, (2,6) weight 4, (6,0) weight 8, (0,5) weight 7, (1,5) weight 2.</p>

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.2

22	<p>Diagram 22: Directed graph with nodes 0, 1, 2, 3, 4, 5, 6. Edges: (0,6) labeled (3), (6,1) labeled (8), (1,2) labeled (2), (5,6) labeled (7), (5,2) labeled (1), (3,5) labeled (9), (3,4) labeled (5), (2,4) labeled (6), (4,0) labeled (4).</p>
23	<p>Diagram 23: Directed graph with nodes 0, 1, 2, 3, 4, 5, 6. Edges: (0,5) labeled (8), (5,1) labeled (3), (1,3) labeled (4), (0,6) labeled (6), (6,4) labeled (1), (4,2) labeled (5), (5,2) labeled (2), (2,3) labeled (7).</p>
24	<p>Diagram 24: Directed graph with nodes 0, 1, 2, 3, 4, 5, 6. Edges: (3,2) labeled (8), (2,0) labeled (1), (3,1) labeled (4), (1,4) labeled (2), (4,0) labeled (6), (6,3) labeled (3), (6,5) labeled (5), (0,5) labeled (7).</p>

25	
26	
27	
28	



13. Дан список рёбер. Составьте по нему рисунок ориентированного графа. Найдите для этого графа наименьший путь от вершины-входа до вершины с максимальным номером.

1.  $(0;1) - 3, (0;2) - 9, (1;2) - 5, (2;4) - 1, (1;3) - 8, (2;3) - 2, (3;5) - 4, (4;5) - 6.$
2.  $(0;1) - 4, (0;2) - 5, (1;2) - 8, (2;4) - 3, (1;3) - 11, (2;3) - 5, (3;5) - 3, (4;5) - 6.$
3.  $(0;1) - 3, (0;2) - 9, (1;2) - 12, (2;4) - 1, (1;3) - 2, (2;3) - 3, (3;5) - 10, (4;5) - 5.$
4.  $(0;1) - 6, (0;2) - 2, (2;1) - 3, (2;4) - 6, (1;3) - 1, (2;3) - 5, (3;5) - 8, (4;5) - 7.$
5.  $(0;1) - 6, (0;2) - 5, (1;2) - 1, (2;4) - 6, (1;3) - 7, (2;3) - 6, (3;5) - 8, (4;5) - 7.$

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

6.  $(0;1) - 3, (0;2) - 2, (2;1) - 1, (2;5) - 3, (1;5) - 4, (5;4) - 8, (5;3) - 5, (3;4) - 3, (4;6) - 2, (3;6) - 4.$
7.  $(0;1) - 10, (0;2) - 5, (2;1) - 4, (2;5) - 8, (1;5) - 3, (5;4) - 4, (5;3) - 2, (3;4) - 1, (4;6) - 5, (3;6) - 7.$
8.  $(0;1) - 3, (0;2) - 2, (2;1) - 2, (2;5) - 12, (1;5) - 8, (5;4) - 2, (5;3) - 6, (3;4) - 1, (4;6) - 8, (3;6) - 3. (0;1) - 2, (0;2) - 7, (2;1) - 1, (2;5) - 6, (1;5) - 12, (5;4) - 10, (5;3) - 5, (3;4) - 4, (4;6) - 2, (3;6) - 7.$
9.  $(0;1) - 4, (0;2) - 2, (2;1) - 1, (2;5) - 7, (1;5) - 5, (5;4) - 4, (5;3) - 1, (3;4) - 4, (4;6) - 3, (3;6) - 7.$
10.  $(0;2) - 2, (0;1) - 7, (2;1) - 4, (2;4) - 9, (1;3) - 3, (3;4) - 1, (4;6) - 2, (3;5) - 8, (6;5) - 4, (6;7) - 10, (5;7) - 5.$
11.  $(0;2) - 10, (0;1) - 5, (2;1) - 1, (2;4) - 4, (1;3) - 3, (3;4) - 5, (4;6) - 3, (3;5) - 10, (6;5) - 10, (6;7) - 5, (5;7) - 1.$
12.  $(0;2) - 4, (0;1) - 6, (2;1) - 4, (2;4) - 6, (1;3) - 3, (3;4) - 2, (4;6) - 4, (3;5) - 7, (6;5) - 3, (6;7) - 8, (5;7) - 5.$
13.  $(0;2) - 3, (0;1) - 1, (2;1) - 4, (2;4) - 2, (1;3) - 6, (3;4) - 4, (4;6) - 3, (3;5) - 6, (6;5) - 4, (6;7) - 12, (5;7) - 7.$
14.  $(0;2) - 8, (0;1) - 12, (2;1) - 3, (2;4) - 6, (1;3) - 5, (3;4) - 4, (4;6) - 10, (3;5) - 4, (6;5) - 6, (6;7) - 10, (5;7) - 6.$
15.  $(0;1) - 3, (0;2) - 3, (1;4) - 3, (2;5) - 3, (1;3) - 2, (0;3) - 4, (2;3) - 2, (3;4) - 2, (3;6) - 4, (3;5) - 2, (4;6) - 3, (5;6) - 3.$
16.  $(0;1) - 3, (0;2) - 2, (1;4) - 13, (2;5) - 13, (1;3) - 7, (0;3) - 11, (2;3) - 9, (3;4) - 5, (3;6) - 10, (3;5) - 3, (4;6) - 4, (5;6) - 7.$
17.  $(0;1) - 4, (0;2) - 5, (1;4) - 7, (2;5) - 7, (1;3) - 5, (0;3) - 8, (2;3) - 4, (3;4) - 2, (3;6) - 7, (3;5) - 1, (4;6) - 4, (5;6) - 6.$
18.  $(0;1) - 5, (0;2) - 5, (1;4) - 4, (2;5) - 5, (1;3) - 3, (0;3) - 10, (2;3) - 3, (3;4) - 3, (3;6) - 10, (3;5) - 3, (4;6) - 7, (5;6) - 5.$
19.  $(0;1) - 6, (0;2) - 7, (1;4) - 18, (2;5) - 19, (1;3) - 8, (0;3) - 14, (2;3) - 6, (3;4) - 8, (3;6) - 14, (3;5) - 5, (4;6) - 5, (5;6) - 9.$
20.  $(0;1) - 2, (1;2) - 3, (0;2) - 6, (2;3) - 4, (3;4) - 2, (2;4) - 7, (4;5) - 5, (5;6) - 3, (4;6) - 9.$
21.  $(0;1) - 3, (1;2) - 5, (0;2) - 7, (2;3) - 6, (3;4) - 5, (2;4) - 12, (4;5) - 3, (5;6) - 2, (4;6) - 4.$

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

22.  $(0;1) - 5, (1;2) - 4, (0;2) - 10, (2;3) - 4, (3;4) - 5, (2;4) - 8, (4;5) - 7, (5;6) - 6, (4;6) - 14.$
23.  $(0;1) - 2, (1;2) - 4, (0;2) - 5, (2;3) - 4, (3;4) - 5, (2;4) - 8, (4;5) - 3, (5;6) - 2, (4;6) - 4.$
24.  $(0;1) - 3, (1;2) - 2, (0;2) - 5, (2;3) - 1, (3;4) - 1, (2;4) - 3, (4;5) - 2, (5;6) - 2, (4;6) - 3.$
25.  $(0;1) - 10, (0;2) - 4, (1;4) - 8, (2;5) - 20, (3;1) - 5, (4;3) - 3, (2;3) - 4, (3;5) - 17, (4;6) - 7, (5;6) - 5.$
26.  $(0;1) - 6, (0;2) - 2, (1;4) - 4, (2;5) - 15, (3;1) - 2, (4;3) - 3, (2;3) - 13, (3;5) - 2, (4;6) - 7, (5;6) - 1.$
27.  $(0;1) - 2, (0;2) - 3, (1;4) - 1, (2;5) - 10, (3;1) - 1, (4;3) - 6, (2;3) - 4, (3;5) - 5, (4;6) - 20, (5;6) - 6.$
28.  $(0;1) - 2, (0;2) - 3, (1;4) - 7, (2;5) - 6, (3;1) - 1, (4;3) - 2, (2;3) - 2, (3;5) - 3, (4;6) - 8, (5;6) - 10.$
29.  $(0;1) - 3, (0;2) - 7, (1;4) - 8, (2;5) - 3, (3;1) - 1, (4;3) - 4, (2;3) - 2, (3;5) - 2, (4;6) - 7, (5;6) - 6.$

14. Для орграфа дана матрица весов. В позиции  $(i, j)$  записана длина дуги из вершины  $i$  в вершину  $j$  (ноль означает, что пути из  $i$  в  $j$  не существует). Если согласно матрице путь  $i \rightarrow j$  существует, то путь  $j \rightarrow i$  не существует. Задание: найти кратчайший путь из вершины 1 в вершину 5 (Таблица 1.4.3).

Таблица 1.4.3 – Варианты заданий

Номер варианта	Матрица весов орграфа
1	$\begin{pmatrix} 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

Продолжение таблицы 1.4.3

2	$\begin{pmatrix} 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$
3	$\begin{pmatrix} 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$
4	$\begin{pmatrix} 0 & 5 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$
5	$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.3

6	$\begin{pmatrix} 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$
7	$\begin{pmatrix} 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 2 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$
8	$\begin{pmatrix} 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$
9	$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.3

10	$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
11	$\begin{pmatrix} 0 & 0 & 0 & 3 & 9 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
12	$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 5 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \end{pmatrix}$
13	$\begin{pmatrix} 0 & 1 & 0 & 1 & 9 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.3

14	$\begin{pmatrix} 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
15	$\begin{pmatrix} 0 & 9 & 0 & 2 & 0 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
16	$\begin{pmatrix} 0 & 9 & 0 & 9 & 0 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
17	$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.3

18	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix}$
19	$\begin{pmatrix} 0 & 1 & 0 & 7 & 0 \\ 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix}$
20	$\begin{pmatrix} 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
21	$\begin{pmatrix} 0 & 4 & 2 & 0 & 0 \\ 0 & 0 & 0 & 4 & 7 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.3

22	$\begin{pmatrix} 0 & 1 & 2 & 0 & 3 \\ 0 & 0 & 0 & 4 & 3 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
23	$\begin{pmatrix} 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$
24	$\begin{pmatrix} 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$
25	$\begin{pmatrix} 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 & 5 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.3

26	$\begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 1 & 0 & 0 & 2 & 6 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
27	$\begin{pmatrix} 0 & 0 & 0 & 1 & 9 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
28	$\begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 3 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
29	$\begin{pmatrix} 0 & 0 & 0 & 1 & 9 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 5 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Окончание таблицы 1.4.3

30	$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
----	---

15. Для орграфа дана матрица весов. В позиции  $(i, j)$  записана длина дуги из вершины  $i$  в вершину  $j$  ( $\infty$  означает, что пути из  $i$  в  $j$  не существует). Задание: нарисовать изображение графа, составить матрицы кратчайших расстояний и маршрутов (Таблица 1.4.4).

Таблица 1.4.4 – Варианты заданий

Номер варианта	Матрица весов орграфа
1	$\begin{pmatrix} 0 & 9 & \infty & 3 & \infty \\ \infty & 0 & 3 & \infty & 1 \\ \infty & \infty & 0 & \infty & 2 \\ \infty & \infty & 4 & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.4

2	$\begin{pmatrix} 0 & \infty & \infty & 6 & 1 \\ \infty & 0 & \infty & \infty & 3 \\ \infty & \infty & 0 & \infty & \infty \\ \infty & 2 & 8 & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	
3	$\begin{pmatrix} 0 & 1 & \infty & 1 & \infty \\ \infty & 0 & 7 & \infty & \infty \\ \infty & \infty & 0 & \infty & 3 \\ \infty & \infty & \infty & 0 & 6 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	
4	$\begin{pmatrix} 0 & 9 & \infty & 8 & \infty \\ \infty & 0 & 1 & \infty & 4 \\ \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & 5 & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	
5	$\begin{pmatrix} 0 & \infty & \infty & 1 & 9 \\ \infty & 0 & \infty & \infty & 3 \\ \infty & \infty & 0 & \infty & \infty \\ \infty & 3 & 8 & 0 & 2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.4

6	$\begin{pmatrix} 0 & 5 & \infty & \infty & \infty \\ \infty & 0 & \infty & 5 & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & 5 & 0 & \infty \\ 3 & \infty & \infty & \infty & 0 \end{pmatrix}$
7	$\begin{pmatrix} 0 & 3 & \infty & \infty & \infty \\ \infty & 0 & 2 & 6 & \infty \\ \infty & \infty & 0 & 5 & \infty \\ \infty & \infty & \infty & 0 & 1 \\ \infty & 1 & \infty & \infty & 0 \end{pmatrix}$
8	$\begin{pmatrix} 0 & \infty & 1 & 2 & \infty \\ \infty & 0 & \infty & \infty & 8 \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 5 & 9 & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
9	$\begin{pmatrix} 0 & 5 & \infty & \infty & \infty \\ \infty & 0 & 1 & 4 & \infty \\ \infty & \infty & 0 & \infty & 2 \\ \infty & \infty & 2 & 0 & \infty \\ 3 & \infty & \infty & \infty & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.4

10	$\begin{pmatrix} 0 & 1 & \infty & \infty & 1 \\ \infty & 0 & 3 & 4 & \infty \\ \infty & \infty & 0 & 5 & \infty \\ \infty & \infty & \infty & 0 & 1 \\ \infty & 1 & \infty & \infty & 0 \end{pmatrix}$
11	$\begin{pmatrix} 0 & \infty & \infty & 1 & 9 \\ \infty & 0 & \infty & \infty & 4 \\ \infty & \infty & 0 & \infty & \infty \\ \infty & 5 & 5 & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
12	$\begin{pmatrix} 0 & 2 & \infty & 3 & \infty \\ \infty & 0 & \infty & 4 & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & 3 & 0 & \infty \\ 3 & \infty & \infty & \infty & 0 \end{pmatrix}$
13	$\begin{pmatrix} 0 & 3 & \infty & \infty & \infty \\ \infty & 0 & 4 & \infty & \infty \\ 2 & \infty & 0 & 2 & 4 \\ 3 & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.4

14	$\begin{pmatrix} 0 & 9 & \infty & \infty & \infty \\ \infty & 0 & \infty & 7 & 2 \\ \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & 2 & 0 & \infty \\ 8 & \infty & \infty & \infty & 0 \end{pmatrix}$
15	$\begin{pmatrix} 0 & \infty & 2 & 5 & \infty \\ \infty & 0 & \infty & 4 & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & 2 & 0 & \infty \\ 6 & \infty & \infty & \infty & 0 \end{pmatrix}$
16	$\begin{pmatrix} 0 & 1 & 3 & \infty & \infty \\ \infty & 0 & \infty & 4 & 1 \\ \infty & \infty & 0 & 2 & \infty \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
17	$\begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & 3 & 1 & \infty \\ \infty & \infty & 0 & \infty & 4 \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.4

18	$\begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & 9 & \infty & \infty \\ \infty & \infty & 0 & \infty & 4 \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
19	$\begin{pmatrix} 0 & \infty & \infty & 2 & \infty \\ \infty & 0 & \infty & \infty & 5 \\ \infty & \infty & 0 & \infty & 4 \\ \infty & 7 & \infty & 0 & \infty \\ 3 & \infty & \infty & \infty & 0 \end{pmatrix}$
20	$\begin{pmatrix} 0 & \infty & \infty & 2 & 9 \\ \infty & 0 & \infty & 6 & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & 5 & 0 & 2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
21	$\begin{pmatrix} 0 & \infty & 1 & 2 & \infty \\ \infty & 0 & \infty & \infty & 5 \\ \infty & \infty & 0 & \infty & 4 \\ \infty & 3 & \infty & 0 & \infty \\ 3 & \infty & \infty & \infty & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.4

22	$\begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & 1 & \infty & \infty \\ \infty & \infty & 0 & 5 & 2 \\ \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 3 & 0 \end{pmatrix}$
23	$\begin{pmatrix} 0 & 1 & \infty & 2 & \infty \\ \infty & 0 & 3 & \infty & 1 \\ \infty & \infty & 0 & \infty & 4 \\ \infty & \infty & 1 & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
24	$\begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & 3 & \infty & 3 \\ \infty & \infty & 0 & \infty & 4 \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
25	$\begin{pmatrix} 0 & \infty & \infty & 3 & \infty \\ \infty & 0 & \infty & 4 & \infty \\ \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & 2 & 0 & 2 \\ 3 & \infty & \infty & \infty & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.4.4

26	$\begin{pmatrix} 0 & 1 & \infty & 7 & \infty \\ \infty & 0 & 2 & 3 & \infty \\ \infty & \infty & 0 & 5 & \infty \\ \infty & \infty & \infty & 0 & 1 \\ \infty & 4 & \infty & \infty & 0 \end{pmatrix}$
27	$\begin{pmatrix} 0 & 1 & \infty & 1 & 9 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 3 \\ \infty & \infty & 2 & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
28	$\begin{pmatrix} 0 & 1 & 2 & \infty & 3 \\ \infty & 0 & \infty & 4 & 3 \\ \infty & \infty & 0 & 3 & \infty \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
29	$\begin{pmatrix} 0 & 1 & \infty & 2 & \infty \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 4 \\ \infty & 3 & \infty & 0 & 5 \\ 3 & \infty & \infty & \infty & 0 \end{pmatrix}$

30	$\begin{pmatrix} 0 & 4 & 2 & \infty & \infty \\ \infty & 0 & \infty & 4 & 7 \\ \infty & \infty & 0 & 3 & \infty \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$
----	---

16. Какие отличительные особенности алгоритма Флойда вы знаете?
17. Для решения каких задач не подходит алгоритм Флойда?

### 1.5. Алгоритм Форда – Фалкерсона

#### *Описание алгоритма*

В этом разделе под сетью мы понимаем связный орграф  $D(V, E)$ , у которого есть одна вершина  $s$  со степенью захода 0 (источник) и одна вершина  $t$  со степенью исхода 0 (сток).

Дуги сети нагружены неотрицательными вещественными числами, то есть, задана функция  $c: E \rightarrow R$ . Если  $e \in E$  - дуга, то  $c(e)$  - пропускная способность дуги.

Разрез сети - это разбиение множества вершин  $V$  на два непересекающихся подмножества  $S$  и  $T$  таких, что  $s \in S, t \in T$ .

Обозначим:

- через  $P^+$  множество всех дуг от  $S$  к  $T$  (то есть начало дуги в  $S$ , а конец дуги в  $T$ ),

- через  $P^-$  - множество всех дуг от  $T$  к  $S$ .

Под разрезом также понимают и  $P = P^+ \cup P^-$ ; под ориентированным разрезом (орразрезом) – множество дуг  $P^-$ .

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Сумма пропускных способностей всех дуг разреза (орразреза) называется пропускной способностью и обозначается соответственно  $C(P)$ ,  $C(P^+)$ .

Функция  $\varphi: E \rightarrow R$  является потоком в сети, если выполнены условия:

- 1) для любой дуги  $e \in E$   $0 \leq \varphi(e) \leq c(e)$ ;
- 2) для любой вершины  $u \in V \setminus \{s, t\}$  - сумма потоков по дугам входящим в  $u$  равна сумме потоков по дугам исходящим из  $u$  ( $\varphi(e)$  - поток по дуге  $e$ ).

Величиной потока назовем сумму потоков по дугам исходящим из  $S$ , обозначим её через  $W(\varphi)$ . Нетрудно показать, что сумма потоков по дугам, входящим в  $t$  равна  $W(\varphi)$ .

Если  $P = P^+ \cup P^-$  - разрез, то справедливо равенство:

$$W(\varphi) = \Phi(P^+) - \Phi(P^-),$$

где  $\Phi(P^+)$ ,  $\Phi(P^-)$  – суммы потоков по дугам, входящим в  $P^+$ ,  $P^-$  соответственно.

Поток максимальный, если его величина принимает наибольшее возможное значение, эта величина называется **пропускной способностью сети**. Орразрез называется минимальным, если минимальна его пропускная способность среди всех его орразрезов. Из равенства, приведенного выше, следует, что  $W(\varphi) \leq \Phi(P^+) \leq C(P^+)$ , то есть величина любого потока не превосходит пропускной способности любого орразреза. Следовательно, пропускная способность сети не превосходит пропускной способности минимального орразреза.

**Теорема Форда – Фалкерсона:** пропускная способность сети равна пропускной способности минимального орразреза.

Поскольку доказательство теоремы носит конструктивный характер и используется при построении алгоритма, то кратко изложим его суть.

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Соединим  $s$  с вершиной  $u$  при помощи цепи  $\langle s, u \rangle$ , не учитывая направления дуг. Цепь  $\langle s, u \rangle$  называется *аугментальной*, если дуги  $e^+$ , направленные против движения, имеют положительный поток; иными словами все величины  $c(e^+) - \varphi(e^+)$ ,  $\varphi(e^-)$  строго положительны. Пусть  $t$  достижима из  $s$  при помощи такой цепи, тогда величину потока можно увеличить. Обозначим через  $K$  минимальное среди всех значений  $c(e^+) - \varphi(e^+)$ ,  $\varphi(e^-)$ ;  $K > 0$ .

Перестроим поток следующим образом:  $\varphi(e^+)$  заменим на  $\varphi(e^+) + K$ , а  $\varphi(e^-)$  на  $\varphi(e^-) - K$ .

Условия (1) и (2) останутся выполненными, а величина потока при этом увеличится на  $K$ . Повторяем процедуру до тех пор, пока  $t$  перестанет быть достижимой. Пусть  $S$  - множество всех вершин, достижимых из  $s$ , а  $T = V \setminus S$  содержит  $t$ . Следовательно, это разрез, при этом дуги, входящие в  $P^+$  насыщены, а дуги из  $P^-$  имеют нулевой поток (иначе  $S$  можно было бы расширить).

Таким образом,  $W(\varphi) = \Phi(P^+) - \Phi(P^-) = C(P^+) - 0 = C(P^+)$ , то есть поток максимальный, а  $P^+$  - минимальный орразрез.

### Пример 1.5.1

*Найти максимальный поток  $\varphi$  для сети  $T$  (Рисунок 1.5.1)*

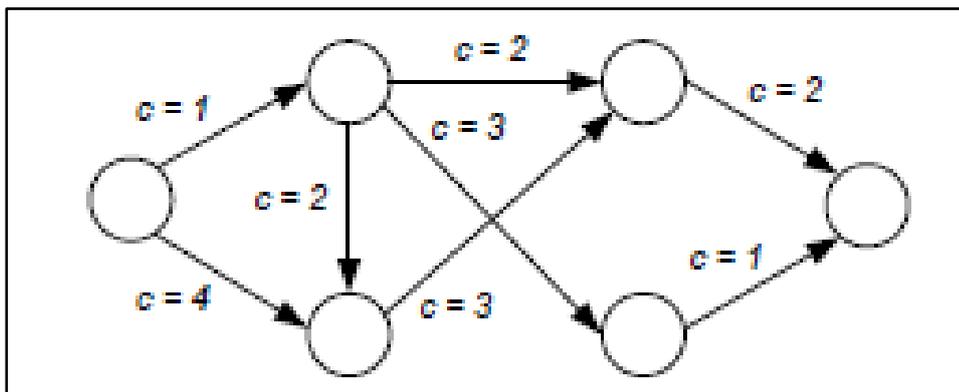


Рисунок 1.5.1 – Исходный граф

Решение:

- 1) Перенумеруем поток произвольным образом. Вершина-исток должна иметь номер 1, а вершина-сток – максимальное значение среди всех вершин сети. Зададим начальный поток в сети  $\varphi = 0$  и присвоим вершине истоку нулевую метку (Рисунок 1.5.2).

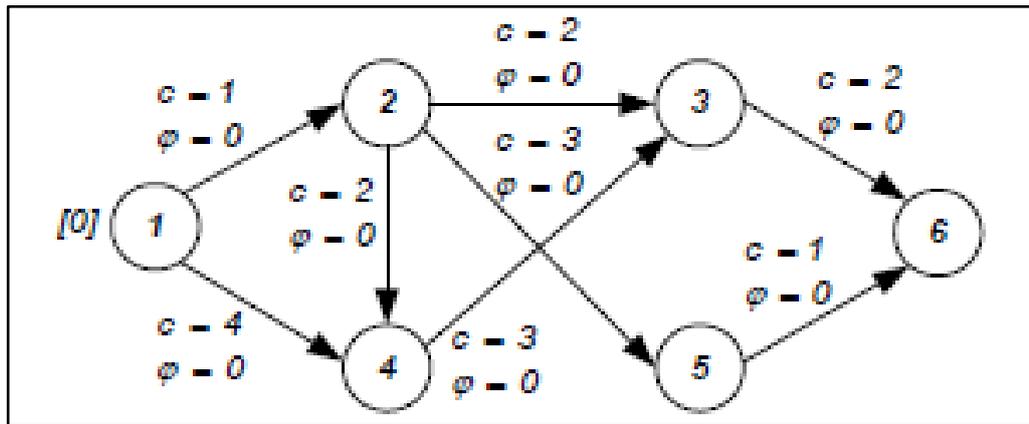


Рисунок 1.5.2 - Зададим начальный поток в сети  $\varphi = 0$  и присвоим вершине истоку нулевую метку

- 2) Пометим вершины и дуги сети (Рисунок 1.5.3), пользуясь следующим правилом: пусть  $v$  – помеченная вершина,  $w$  – непомеченная, тогда:

- если  $w = \Gamma^+(v)$ , то есть дуга направлена от вершины  $v$  к вершине  $w$ , и выполняется неравенство:

$$c(v, w) - \varphi(v, w) > 0,$$

вершине  $w$  присваивается метка, равная номеру вершины  $v$ , а дуге значение «+»;

- если  $w = \Gamma^-(v)$ , то есть дуга направлена от вершины  $w$  к вершине  $v$ , и выполняется неравенство:

$$\varphi(v, w) > 0,$$

вершине  $w$  присваивается потенциал, равный номеру вершины  $v$ , а дуге – значение «-» (Рисунок 1.5.3).

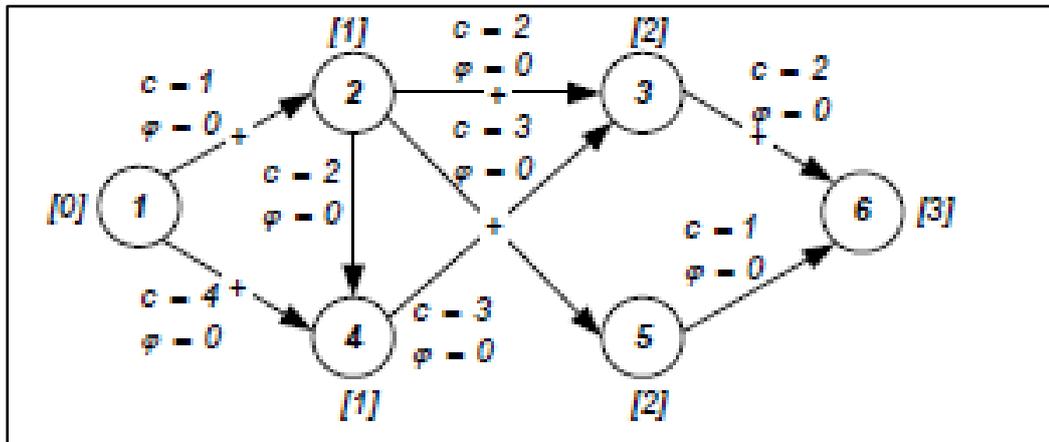


Рисунок 1.5.3 - Пометим вершины и дуги сети

Вершина-сток получила метку, поэтому продолжаем решение.

- 3) Рассмотрим подграф исходного графа (Рисунок 1.5.4), такой, что метка вершины  $w$  соответствует номеру вершины  $v$ , где  $v$  – вершина-начало дуги  $(v, w)$ .

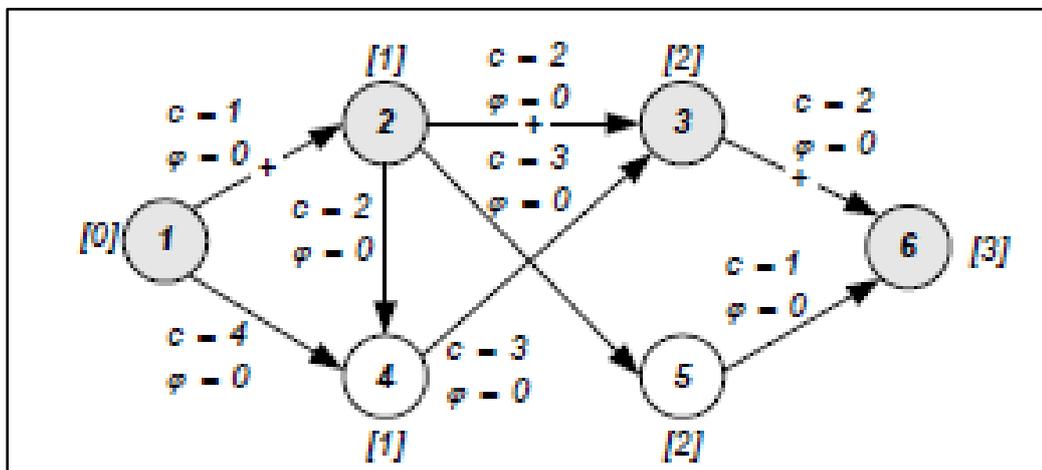


Рисунок 1.5.4 – Продолжение примера 1.5.1

- 4) Перестроим поток в сети по правилу:  
 - если некоторая дуга  $e$  не принадлежит рассматриваемому подграфу, то величина нового потока составит  $\varphi^*(e) = \varphi(e)$ ;

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

- если некоторая дуга  $e$  принадлежит рассматриваемому подграфу и имеет знак «+», то:  $\varphi^*(e) = \varphi(e) + K_1$ , где  $K_1 = \min [c(e) - \varphi(e)]$ ;
- если некоторая дуга  $e$  принадлежит рассматриваемому подграфу и имеет знак «-», то:  $\varphi^*(e) = \varphi(e) - K_2$ , где  $K_2 = \min [\varphi(e)]$ . В рассматриваемом случае (Рисунок 1.5.5)  $\varphi^* = 0 + \min [1, 2, 2] = 1$ .

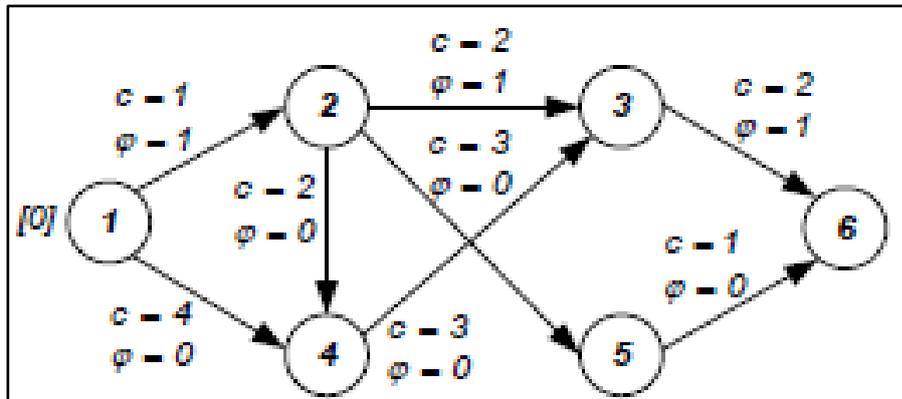


Рисунок 1.5.5 - Продолжение примера 5.1.1

Далее осуществляем переход к пункту 2, по результатам выполнения которого представляется граф с заново размеченными вершинами (Рисунок 1.5.6).

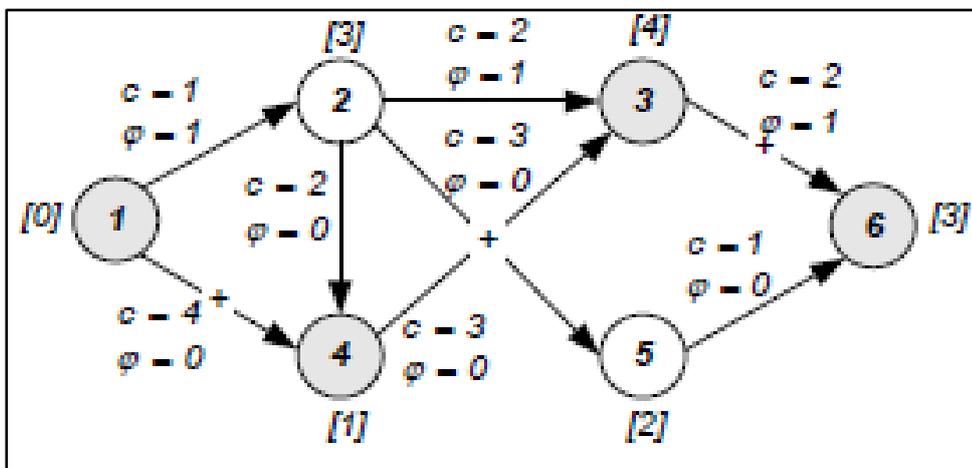


Рисунок 1.5.6 - Граф с заново размеченными вершинами

=====

Вершина-сток получила метку, поэтому продолжаем решение. Зададим новый поток при рассмотрении подграфа, отраженного на рисунке 1.5.6.  $K_1 = \min [4, 3, 1] = 1$ ;

$$\varphi^*(1,4) = 0 + K_1 = 1;$$

$$\varphi^*(4,3) = 0 + K_1 = 1;$$

$$\varphi^*(3,6) = 1 + K_1 = 2.$$

$$\varphi^* = \min [\varphi^*(1,4), \varphi^*(4,3), \varphi^*(3,6)] = 1$$

5) Увеличим поток в сети на величину  $\varphi^*$  (Рисунок 1.5.7)

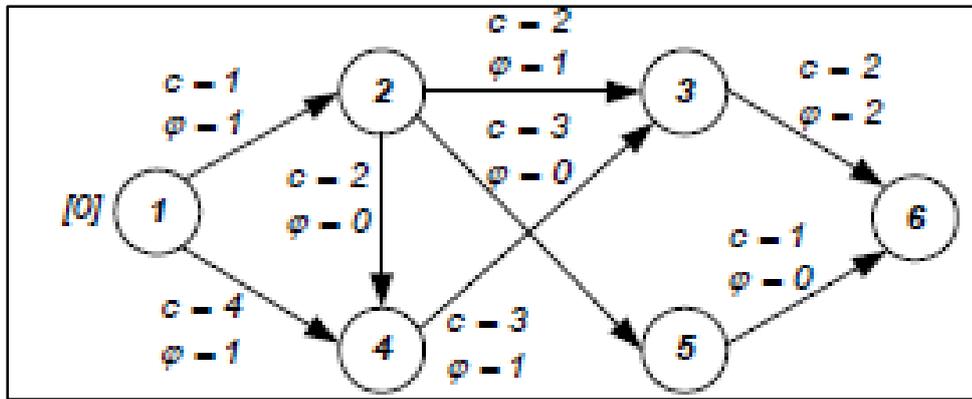


Рисунок 1.5.7 - Увеличим поток в сети

Осуществим переход к пункту 2, результат выполнения которого отражен на рисунке 1.5.8.

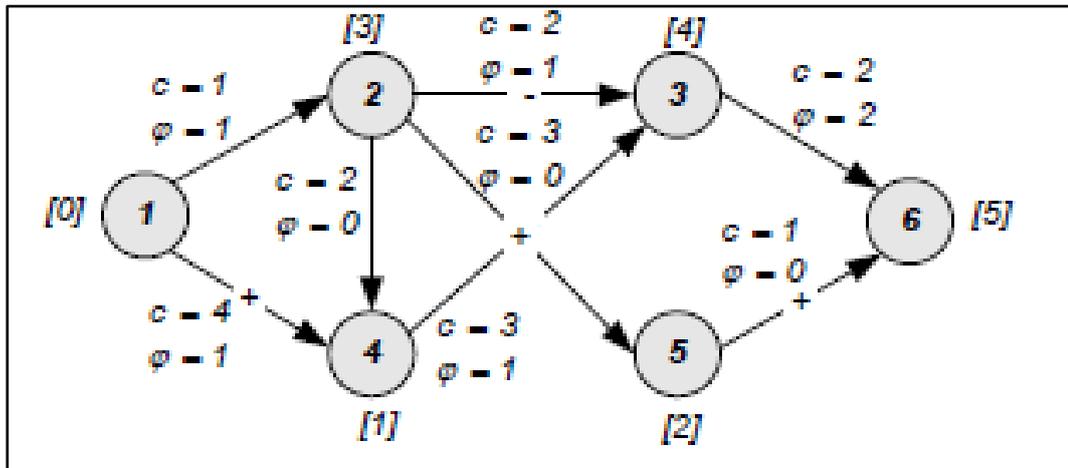


Рисунок 1.5.8 – Результат выполнения пункта 2

Вершине-стоку присвоен потенциал, поэтому продолжаем решение. Перестроим поток в сети:  $K_1 = \min [3, 2, 1, 3, 1] = 1$ ;

$$\varphi^*(1,4) = 1 + K_1 = 2;$$

$$\varphi^*(4,3) = 1 + K_1 = 2;$$

$$\varphi^*(3,2) = 1 + K_1 = 2;$$

$$\varphi^*(2,5) = 0 + K_1 = 1;$$

$$\varphi(5,6) = 0 + K_1 = 1;$$

$$\varphi^* = \min [\varphi^*(1,4), \varphi^*(4,3), \varphi^*(3,2), \varphi^*(2,5), \varphi^*(5,6)] = 1.$$

б) Увеличиваем поток в сети на величину  $\varphi^*$  (Рисунок 1.5.9).

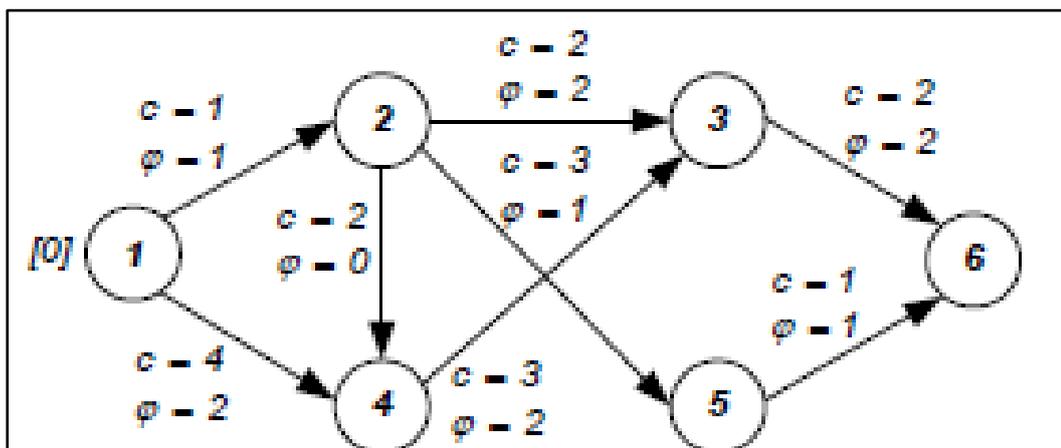


Рисунок 1.5.9 - Увеличиваем поток в сети  
Осуществляем переход к пункту 2 (Рисунок 1.5.10).

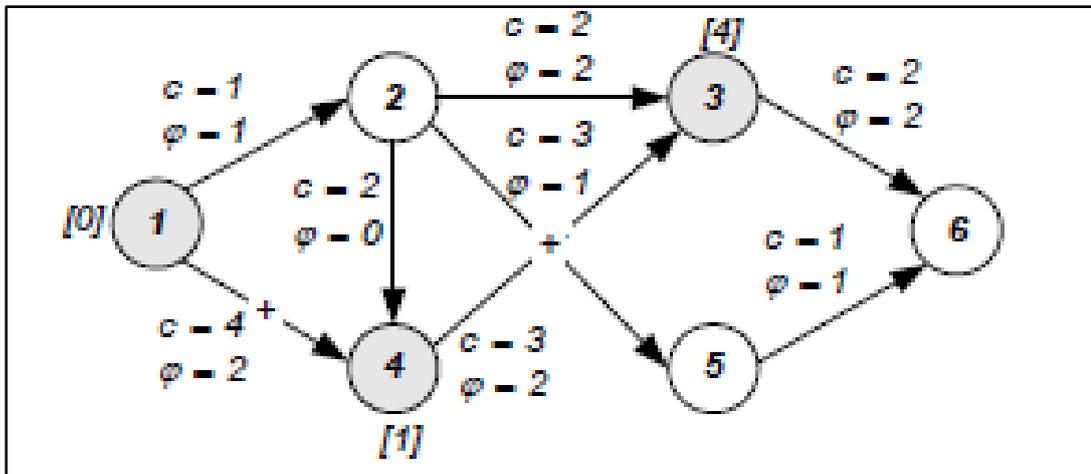


Рисунок 1.5.10 - Осуществляем переход к пункту 2

Вершина-сток не получила метку, что свидетельствует о том, что максимальный поток в рассматриваемой сети найден (Рисунок 1.5.11).

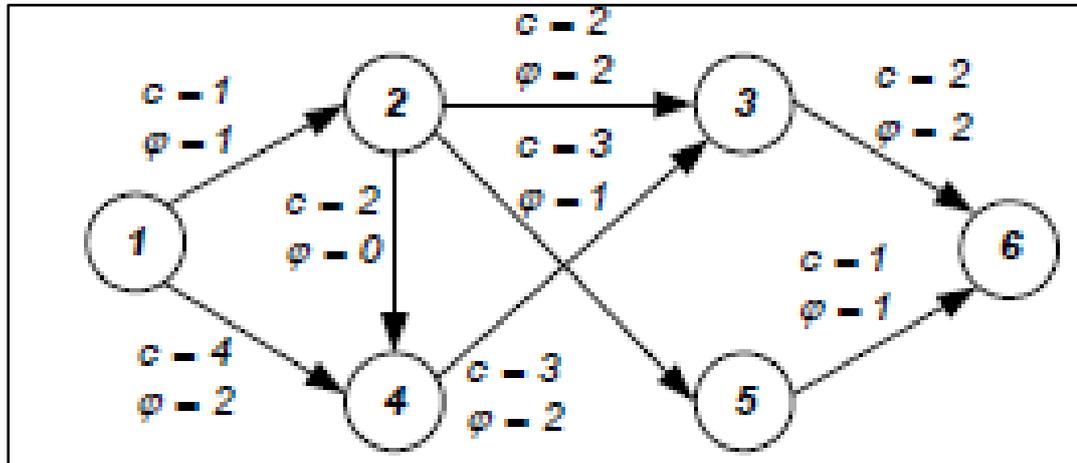


Рисунок 1.5.11 - Максимальный поток в рассматриваемой сети найден

Таким образом, по результатам выполнения алгоритма получили, что множество достижимых из вершины-истока вершин  $S = \{1, 3, 4\}$ ; множество недостижимых из вершины-источника вершин  $T = \{2, 5, 6\}$ . Минимальным орразрезом является  $P^+ : (3;6)$ .

Максимальный поток в сети равен 2.

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

### Вопросы и задания

1. Что понимается под понятиями «сеть» и «разрез сети» в алгоритме Форда – Фалкерсона?
2. Какие условия должны выполняться, чтобы функция  $f: E \rightarrow R$  являлась потоком в сети?
3. Какая величина называется пропускной способностью сети?
4. Какая теорема используется при построении алгоритма Форда – Фалкерсона?
5. Определить минимальный орразрез и найти максимальный поток в сети (Таблица 1.5.1).

Таблица 1.5.1 – Варианты заданий

Номер варианта	Матрица графа
1	$\begin{pmatrix} 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 4 & 5 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.5.1

2	$\begin{pmatrix} 0 & 2 & 5 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 5 & 2 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
3	$\begin{pmatrix} 0 & 2 & 1 & 4 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 5 & 6 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
4	$\begin{pmatrix} 0 & 2 & 1 & 5 & 1 \\ 0 & 0 & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
5	$\begin{pmatrix} 0 & 2 & 0 & 4 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.5.1

6	$\begin{pmatrix} 0 & 3 & 1 & 0 & 4 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
7	$\begin{pmatrix} 0 & 3 & 0 & 5 & 3 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
8	$\begin{pmatrix} 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 2 & 6 & 1 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
9	$\begin{pmatrix} 0 & 4 & 0 & 5 & 1 \\ 0 & 0 & 1 & 4 & 2 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.5.1

10	$\begin{pmatrix} 0 & 0 & 2 & 4 & 5 \\ 0 & 0 & 4 & 8 & 0 \\ 0 & 0 & 0 & 7 & 6 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
11	$\begin{pmatrix} 0 & 6 & 0 & 4 & 0 \\ 0 & 0 & 4 & 8 & 4 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
12	$\begin{pmatrix} 0 & 1 & 3 & 3 & 1 \\ 0 & 0 & 5 & 3 & 2 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
13	$\begin{pmatrix} 0 & 0 & 5 & 0 & 3 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 2 & 6 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.5.1

14	$\begin{pmatrix} 0 & 1 & 8 & 4 & 0 \\ 0 & 0 & 5 & 6 & 0 \\ 0 & 0 & 0 & 3 & 5 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
15	$\begin{pmatrix} 0 & 3 & 5 & 0 & 1 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 3 & 6 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
16	$\begin{pmatrix} 0 & 1 & 0 & 6 & 3 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
17	$\begin{pmatrix} 0 & 5 & 0 & 4 & 1 \\ 0 & 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.5.1

18	$\begin{pmatrix} 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
19	$\begin{pmatrix} 0 & 3 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$
20	$\begin{pmatrix} 0 & 5 & 0 & 3 & 0 \\ 0 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
21	$\begin{pmatrix} 0 & 5 & 0 & 3 & 0 \\ 0 & 0 & 0 & 5 & 3 \\ 0 & 0 & 0 & 3 & 2 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.5.1

22	$\begin{pmatrix} 0 & 1 & 1 & 0 & 4 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
23	$\begin{pmatrix} 0 & 5 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
24	$\begin{pmatrix} 0 & 3 & 4 & 0 & 3 \\ 0 & 0 & 3 & 0 & 6 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
25	$\begin{pmatrix} 0 & 1 & 8 & 4 & 0 \\ 0 & 0 & 5 & 6 & 0 \\ 0 & 0 & 0 & 3 & 5 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Продолжение таблицы 1.5.1

26	$\begin{pmatrix} 0 & 3 & 5 & 0 & 1 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 3 & 6 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
27	$\begin{pmatrix} 0 & 1 & 0 & 6 & 3 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
28	$\begin{pmatrix} 0 & 5 & 0 & 4 & 1 \\ 0 & 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
29	$\begin{pmatrix} 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

30	$\begin{pmatrix} 0 & 3 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$
----	---

### Список рекомендуемых источников

1. Алгоритмы. Построение и анализ: пер. с англ. / ред. Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – 3-е изд. – Москва ; Санкт-Петербург : Диалектика, 2019. – 1323 с.
2. Берж К. Теория графов и ее применения / К. Берж; пер. А. А. Зыков, И. А. Вайнштейн. – Москва : Изд-во иностранной литературы, 1962. – 319 с.
3. Бонцанини М. Анализ социальных медиа на Python: извлекайте и анализируйте данные из всех уголков социальной паутины на Python. / М. Бонцанини. – Москва : ДМК Пресс, 2018. – 287 с.
4. Гданский Н. И. Основы теории и алгоритмы на графах: учебное пособие для студентов высших учебных заведений, обучающихся по укрупненной группе направлений подготовки 09.03.00 «Информатика и вычислительная техника» (квалификация (степень) «бакалавр») : Высшее образование. Бакалавриат. / Н. И. Гданский. – Москва : ИНФРА-М, 2020. – 204 с.
5. Губанов Д. А. Социальные сети: модели информационного влияния, управления и противоборства. Социальные сети / Д. А. Губанов, Д. А. Новиков, А. Г. Чхартишвили. – Москва : Физматлит, 2010. – 225 с.
6. Дистель Р. Теория графов / Р. Дистель. – Новосибирск :

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

Изд-во Ин-та математики, 2002. – 333 с.

7. Додонов А. Г. Живучесть информационных систем / А. Г. Додонов, Д. В. Ландэ. – Киев : Наукова думка, 2011. – 255 с.

8. Евстигнеев В. А. Теория графов и программирование: Учеб. пособие. / В. А. Евстигнеев. – Новосибирск : НГУ, 1978. – 73 с.

9. Зыков А. А. Основы теории графов / А. А. Зыков. – Москва, 1987.

10. Зыков А. А. Теория конечных графов / А. А. Зыков. – Новосибирск : Наука. Сиб. отд-ние, 1969.

11. Информационная война. Информационное противоборство: теория и практика: монография. Информационная война. Информационное противоборство / ред. В. М. Щекотихин. – Москва : Акад. ФСО России, 2011. – 999 с.

12. Информационное управление в условиях активного противоборства: модели и методы. Информационное управление в условиях активного противоборства / ред. В. Л. Шульц. – Москва : Наука, 2011. – 186 с.

13. Камерон П. Теория графов, теория кодирования и блок-схемы / П. Камерон, Д. Ван Линт; пер. Б. С. Стечкин. – Москва : Наука, 1980. – 139 с.

14. Карпов Д. В. Теория графов. Теория графов / Д. В. Карпов. – Москва : Изд-во МЦНМО, 2022. – 554 с.

15. Касьянов В. Н. Графы в программировании: обработка, визуализация и применение. Графы в программировании / В. Н. Касьянов, В. А. Евстигнеев. – Науч. изд. – Санкт-Петербург : БХВ-Петербург, 2003. – 1104 с.

16. Колчин В. Ф. Случайные графы : Теория вероятностей. Математическая статистика : ТВМС / В. Ф. Колчин. – 2. изд. – М. : ФИЗМАТЛИТ, 2004. – 255 с.

17. Конечные графы и сети / Р. Басакер, Т. Л. Саати, В. Н. Буркова, А. И. Тейман. – Москва : Наука, 1974. – 366 с.

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

18. Кристофидес Н. Теория графов: Алгоритм. подход. Теория графов / Н. Кристофидес; пер. Э. В. Вершков, И. В. Коновальцев, Г. П. Гаврилов. – Москва : Мир, 1978. – 432 с.
19. Ландэ Д. В. Интернетика: навигация в сложных сетях: модели и алгоритмы. Интернетика / Д. В. Ландэ, А. А. Снарский, И. В. Безсуднов. – Москва : URSS, 2009. – 258 с.
20. Лекции по теории графов: [Для студентов по спец. «Математика» и «Прикл. механика»]. Лекции по теории графов / ред. В. А. Емеличев, О. И. Мельников, В. И. Сарванов, Р. И. Тышкевич. – Москва : Наука, 1990. – 382 с.
21. Ловас Л. Прикладные задачи теории графов: Теория паросочетаний в математике, физике, химии. Прикладные задачи теории графов / Л. Ловас, М. Д. Пламмер. – Москва : Мир, 1998. – 653 с.
22. Маркин А. В. Системы графовых баз данных NEO4J: учебное пособие для вузов: для студентов высших учебных заведений, обучающихся по ИТ-направлениям : Высшее образование. Системы графовых баз данных NEO4J / А. В. Маркин. – Москва : Юрайт, 2022. – 302 с.
23. Мельников О. И. Теория графов в алгоритмах и программах: книга для учителей, для школьников и не только! : Науку - всем! Шедевры научно-популярной литературы. Программирование. Теория графов в алгоритмах и программах / О. И. Мельников, А. А. Морозов. – Москва : URSS, 2019. – 197 с.
24. Нидхем М. Графовые алгоритмы: практическая реализация на платформах Apache Spark и Neo4j. Графовые алгоритмы / М. Нидхем, Э. Ходлер. – Москва : ДМК Пресс, 2020. – 257 с.
25. Одинец В. П. Избранные главы теории графов / В. П. Одинец, В. Шлензак. – Москва Ижевск : Институт компьютерных исследований ; Регулярная и хаотическая динамика, 2009. – 503 с.
26. Омельченко А. В. Теория графов / А. В. Омельченко. –

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

Москва : МЦНМО, 2021. – 416 с.

27. Оре О. Теория графов / О. Оре. – Изд. 2-е. – Москва : URSS ЛИБРОКОМ, 2008. – 352 с.

28. Райгородский А. М. Модели случайных графов / А. М. Райгородский. – Изд. 2-е, доп. – Москва : Изд-во МЦНМО, 2016. – 139 с.

29. Райгородский А. М. Экстремальные задачи теории графов и Интернет / А. М. Райгородский. – Долгопрудный : Интеллект, 2012. – 103 с.

30. Расторгуев С. П. Информационное противоборство в моделях и задачах: учебное пособие для студентов, обучающихся по направлениям подготовки и специальностям УГСНП 10.00.00 «Информационная безопасность»: Основы защиты информации. Информационное противоборство в моделях и задачах / С. П. Расторгуев. – М. : URSS ЛЕНАНД, 2017. – Вып. № 15. – 303 с.

31. Расторгуев С. П. Математические модели в информационном противоборстве: экзистенциальная математика. Математические модели в информационном противоборстве / С. П. Расторгуев. – Москва : АНО ЦСОиП, 2014. – 259 с.

32. Рафгарден Т. Совершенный алгоритм: графовые алгоритмы и структуры данных : Computer science. Совершенный алгоритм / Т. Рафгарден. – Санкт-Петербург [и др.] : Питер, 2020. – 255 с.

33. Робинсон Я. Графовые базы данных: новые возможности для работы со связанными данными. Графовые базы данных / Я. Робинсон. – 2-е издание. – Москва : ДМК Пресс, 2016. – 253 с.

34. Свами М. Н. Графы, сети и алгоритмы / М. Н. Свами, К. Тхуласираман; пер. В. А. Горбатов, М. В. Горбатова. – Москва : Мир, 1984. – 454 с.

35. Седжвик Р. Фундаментальные алгоритмы на С: Части 1-5. Анализ. Структуры данных. Сортировка. Поиск. Алгоритмы на графах: [Пер. с англ.]. Фундаментальные алгоритмы на С / Р. Седжвик. – М. [ др.] : DiaSoft, 2003. – 1127 с.

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

- =====  
36. Скиена С. С. Алгоритмы: руководство по разработке: [перевод с английского]. Алгоритмы / С. С. Скиена. – 2-е изд. – Санкт-Петербург : БХВ-Петербург, 2011. – 719 с.
37. Татт У. Т. Теория графов / У. Т. Татт; пер. Г. П. Гаврилов. – Москва : Мир, 1988. – 424 с.
38. Уилсон Р. Д. Введение в теорию графов / Р. Д. Уилсон. – Москва ; Санкт-Петербург : Диалектика, 2019. – 239 с.
39. Филлипс Д. Т. Методы анализа сетей / Д. Т. Филлипс, А. Гарсиа-Диас; пер. Б. Г. Сушков, Е. Г. Коваленко, М. Г. Фуругян. – Москва : Мир, 1984. – 496 с.
40. Харари Ф. Перечисление графов / Ф. Харари, Э. М. Палмер; пер. Г. П. Гаврилов. – Москва : Мир, 1977. – 324 с.
41. Харари Ф. Теория графов / Ф. Харари. – Изд. 4-е. – Москва : ЛИБРОКОМ, 2009. – 300 с.
42. Ahuja R. Network Flows: Theory, Algorithms, and Applications. Network Flows / R. Ahuja, T. Magnanti, J. Orlin. – 1st edition. – Englewood Cliffs, N.J : Pearson, 1993. – 864 с.
43. Barabási A.-L. Network Science / A.-L. Barabási, M. Pósfai. – 1st edition. – Cambridge, United Kingdom : Cambridge University Press, 2016. – 475 с.
44. Bollobas B. Modern Graph Theory / B. Bollobas. – Corrected edition. – New York : Springer, 1998. – 408 с.
45. Chartrand G. A First Course in Graph Theory / G. Chartrand, P. Zhang. – Illustrated edition. – Mineola, NY : Dover Publications, 2012. – 464 с.
46. Gross J. L. Graph Theory and Its Applications / J. L. Gross, J. Yellen. – 2nd edition. – Boca Raton : Chapman and Hall/CRC, 2005. – 800 с.
47. Hartsfield N. Pearls in Graph Theory: A Comprehensive Introduction. Pearls in Graph Theory / N. Hartsfield, G. Ringel. – Mineola, N.Y : Dover Publications, 2003. – 272 с.

## Глава 1. ПРИКЛАДНЫЕ АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

=====

48. Steen M. van. Graph Theory and Complex Networks: An Introduction. Graph Theory and Complex Networks / M. van Steen. – S.l. : Maarten van Steen, 2010. – 300 с.

49. Trudeau R. J. Introduction to Graph Theory / R. J. Trudeau. – 2nd edition. – New York : Dover Publications, 1994. – 224 с.

50. West D. Introduction to Graph Theory / D. West. – 2nd edition. – New York, NY : Pearson, 2017. – 608 с.

=====

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### 2.1. Использование графовых моделей в задаче оценки живучести распределенных информационных систем

Используются следующие термины с соответствующими определениями:

*Живучесть информационной системы* - свойство информационной системы, характеризующее способность выполнять установленный объем функций в условиях воздействий внешней среды и отказов компонентов системы в заданных пределах.

*Информационная система* - организационно упорядоченная совокупность документов (массивов документов) информационных технологий, в том числе с использованием вычислительной техники и связи, реализующие информационные процессы.

*Распределенная информационная система* – представляет собой объединение информационных систем, выполняющих собственные, не зависящие друг от друга функции, с целью коллективного использования информационных фондов и вычислительных ресурсов этих систем. Отдельные информационные системы, как правило, территориально удалены друг от друга.

*Корпоративная информационная система* - это масштабируемая система, объединяющая бизнес-стратегию и информационные технологии, предназначенная для комплексной автоматизации всех видов хозяйственной деятельности больших и средних предприятий, в том числе корпораций, состоящих из группы компаний, требующих единого управления.

*Граф* — это совокупность непустого множества вершин и множества пар вершин. Объекты представляются как вершины, или узлы графа, а связи — как дуги, или рёбра. Для разных областей применения виды графов могут различаться направленностью, ограничениями на

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

количество связей и дополнительными данными о вершинах или рёбрах.

*Распределенная информационная система* - представляет собой многоуровневую иерархическую структуру, включающую в себя множество узлов, связанных между собою определенным образом. Такой конструкции присуще свойство уязвимости, которая определяется тем, что за счет многочисленных узлов и связей между ними нередко проявляется «каскадный эффект», когда сбой в одном каком-либо месте провоцирует перегрузки и выход из строя многих других элементов распределенной информационной системы.

Проектирование новых распределенных информационных систем и развитие уже существующих связаны с проблематикой принятия решений по использованию имеющихся сетевых структур: управлению потоками, распределению ресурсов между узлами. Перечисленные проблемы тесно связаны с задачей определения связности и живучести, существующей или проектируемой распределенной информационной системы.

Проблеме оценки живучести распределенных информационных систем посвящено много работ, в которых разработаны аналитические модели, адекватно описывающие процесс расчета живучести распределенной информационной системы, тем не менее, в настоящее время актуальной является задача разработки аналитического описания, обобщающего полученные ранее результаты и позволяющего не только осуществить разработку новых методов проектирования и анализа распределенных информационных систем, но и ставить и решать задачи расчета живучести распределенной информационной системы большой размерности и сложной структуры.

Для систем со сложной архитектурой и множеством гетерогенных компонентов оценить надежность и живучесть представляется крайне сложной проблемой, даже если известны все необходимые метрики для всех входящих в состав данной корпоративной системы элементов.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
Цель раздела – показать каким образом можно использовать графовые модели в задаче оценки живучести распределенных информационных систем

Задачами являются:

- анализ понятия живучести применительно к информационным системам;
- рассмотрение моделей живучести распределенных информационных систем;
- разработка схемы сети для дальнейшего проведения расчетов и экспериментов;
- осуществление расчетов и экспериментов согласно рассмотренным моделям;
- анализ полученных результатов и выводы об адекватности рассмотренных моделей оценки живучести распределенных информационных систем.

Решение поставленных задач проведены с использованием методов теории графов.

### **Понятие живучести**

Под живучестью понимают свойство системы адаптироваться к новой ситуации и противостоять негативным влияниям, выполняя цель функционирования за счёт соответствующего изменения своей структуры и поведения.

Для определения понятия живучести были использованы формулировки из стандарта IEEE 1061-1992.

Распределенные (крупные) информационные системы (информационные сети) включают в себя много компонентов (узлов), которые должны предоставлять услуги конечному пользователю. Так как конечный пользователь, как правило, отправляет запрос на обслуживание, то определение живучести должно также учитывать время ожидания пользователя.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Определения самих сетевых систем также различаются, главным образом в том, является ли система ограниченной или неограниченной. Это различие играет главную роль в вычислении живучести.

Рекомендуется использовать ключевые компоненты IEEE 1061-1992 в качестве фундамента для формулировки стандартного определения живучести распределенных систем.

- *Система:* если определение живучести изменяется, то, по крайней мере, распределенная сеть, для которой оно было определено, должна быть упомянута. Различные виды важных услуг могут служить основанием для формулировки специальных определений живучести. Кроме того, должно быть указано, является система ограниченной или неограниченной.
- *Угроза:* угроза в системе может затруднять оказание услуг пользователю за установленное время, или может препятствовать оказанию услуг вообще. Угрозы системе могут быть классифицированы как случайные, преднамеренные и техногенные. Случайные угрозы включают ошибки программного обеспечения, ошибки аппаратных средств, и человеческие ошибки. Преднамеренные угрозы включают саботаж, вторжение, или террористические атаки. Техногенные угрозы, как правило, затрудняют обслуживание пользователей и включают в себя природные явления (грозы, ураганы, молнии, наводнение, землетрясение, и т.д.), военные действия и перебои в питании.
- *Адаптируемость:* в случае возникновения угрозы система должна быть способна приспособиться к угрозе и продолжить оказывать необходимые услуги пользователям.
- *Непрерывность обслуживания:* в соответствии с требованиями к системе, услуги должны быть доступны пользователю даже в случае угрозы. Производительность сети не должны падать.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- *Время*: услуги должны предоставляться пользователю за определенное время, установленное системой и ожидаемыми пользователями.

Из этих пяти ключевых элементов может быть сформулировано стандартное определение живучести для дальнейшего использования в науке и различных отраслях промышленности во избежание путаницы и неопределенности. Стандартное определение могло бы также помочь в вычислении живучести.

Согласно современной литературе модель живучести – комбинация, по крайней мере, двадцати признанных качественных моделей, их характеристик, факторов, а также других непризнанных качественных специфических моделей. Часто эти виды моделей пересекаются.

Современная литература использует предлагаемые подходы для описания моделей живучести, смешивает и подбирает другие качественные признаки, характеристики и факторы им свойственные. Определения, взятые из стандартов ISO/IEC 9126 и IEEE 1061-1992 о стандартизированных моделях, использовались для построения списка терминов, характеризующих модели живучести распределенных информационных систем.

Различают структурную и функциональную живучесть (Рисунок 2.1.1).

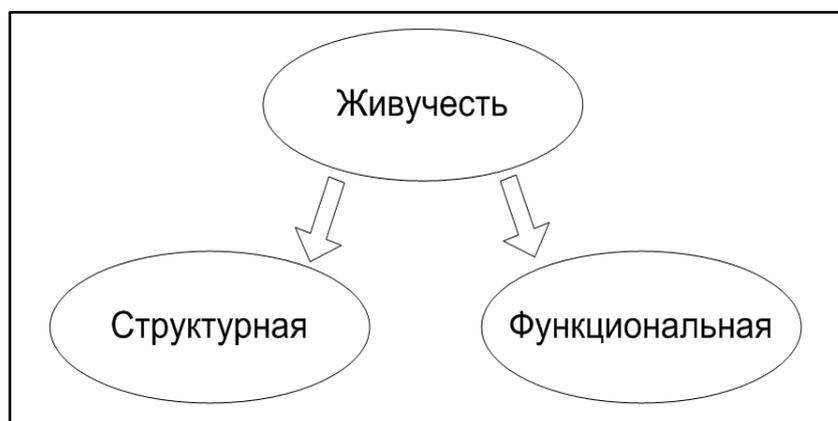


Рисунок 2.1.1 – Виды живучести

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

При рассмотрении структурной живучести учитывается топология сети связи и надёжностные характеристики компонентов. Задачи, связанные с анализом структурной живучести, сводятся к задачам оценки связности топологичной структуры в зависимости от понятия "разрушение".

При рассмотрении функциональной живучести рассматривается набор функций (функциональных задач), исполняемых системой и функциональная живучесть оценивается количеством задач, которые система способна выполнить при различных классах отказов.

Рассмотрим стандартную проектную ситуацию: имеется некий набор типовых проектных элементов, выступающих в качестве базовых «строительных блоков» для сложных технических систем. Для каждого из них известны все основные характеристики, включая показатели надёжности (наработка на отказ, время восстановления после сбоев и др.). Необходимо построить из этих блоков систему, которая в перспективе должна будет обладать заведомо более высокими техническими характеристиками, в том числе по надёжности, чем входящие в ее состав элементы.

Для построения высоконадежной системы из не обладающих необходимой степенью надёжности компонентов самым известным и наиболее типичным конструктивно-технологическим приемом, широко применяемым во многих технических дисциплинах, является избыточность.

Избыточность подразумевает резервирование критически важных блоков и устройств сложных технических систем за счет структурного и функционального дублирования, что позволяет поддерживать работоспособность системы при выходе из строя одного или нескольких компонентов. Действительно, параллельное функционирование в составе сложных технических систем, дублирующих работу друг друга одинаковых блоков и/или устройств, обеспечивает, как правило, увеличение степени надёжности такой системы по сравнению с надёжностью каждого из ее элементов.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

В сложных технических системах используется как «холодное», так и «горячее» резервирование. «Холодное» резервирование основано на принципе пассивной избыточности, когда резервные блоки и/или устройства включаются в работу только после выхода из строя основных компонентов. «Горячее» резервирование использует принцип активной избыточности, когда резервные элементы функционируют параллельно с основными на постоянной основе, что обеспечивает оперативную взаимозаменяемость, а значит и непрерывность работы критически важных устройств. Примером «горячего» резервирования служит так называемое «зеркалирование» дисковых накопителей, обычно применяемое в корпоративных информационных системах поддержки данных высокой степени готовности.

Основной недостаток подхода на основе избыточности – значительное увеличение стоимости проектных решений, не гарантирующее при этом пропорциональный рост надежности системы в целом. Тем не менее, для целого ряда особо ответственных сложных технических систем такой подход является жизненно необходимым, более того – единственно возможным.

Согласно ГОСТ 34.003-90, надежность (reliability) автоматизированных систем есть комплексное свойство автоматизированных систем сохранять во времени в установленных пределах значения всех параметров, характеризующих способность автоматизированных систем выполнять свои функции в заданных режимах и условиях эксплуатации. Функциональная надежность (dependability) рассматривается как способность автоматизированных систем выполнять заданный набор функций.

К другим важным показателям из этой области, т.е. теории и практики построения надежных сложных технических систем, относятся отказоустойчивость, готовность, восстанавливаемость и ремонтпригодность.

- *Отказоустойчивость* можно считать понятием более узким и ограниченным, чем надежность системы в целом, поскольку это качество подразумевает способность системы

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
противостоять определенным классам отказов (процессорные сбои, отказы оперативной и/или дисковой памяти, сбои в работе программного обеспечения и др.) путем целенаправленных мероприятий, ориентированных на борьбу с ними, включая специальные проектные решения.

- *Готовность* (availability) подразумевает гарантированное обеспечение доступности устройств и/или информации в строго определенные интервалы времени в процессе эксплуатации, т.е. предопределяет необходимое соотношение между периодами безотказной работы и временем восстановления системы.
- *Восстанавливаемость* и *ремонтпригодность* (maintainability) характеризуют способность автоматизированных систем восстанавливать свои функции после сбоев и отказов, в том числе путем временного изъятия из эксплуатации отказавших элементов (узлов и блоков) с целью их ремонта и замены резервными элементами. Особый интерес в этом плане представляют системы с возможностями самовосстановления (self-healing).

Так или иначе, для сложных систем с большим количеством компонентов и достаточно сложной конфигурацией бывает очень непросто построить соответствующую математическую модель, позволяющую произвести комплексную оценку надежности сложных технических систем, даже если известны все необходимые показатели и параметры для всех ее компонентов. Это особенно актуально для таких сверхсложных технических систем, как корпоративные информационные системы.

Следовательно, необходима выработка более перспективных подходов и методов оценки показателей качества функционирования сложных технических систем, основанных на комплексных критериях оценки надежно-стоимостных характеристик. Такие подходы должны позволять более эффективно и адекватно анализировать общее состояние системы с позиций надежности и отказоустойчивости на

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

всем протяжении жизненного цикла данного изделия. Представляется целесообразным в качестве такого наиболее содержательного интегрального показателя использовать «живучесть» сложных технических систем.

Под *живучестью* (survivability) автоматизированной системы, согласно определению, приведенному в ГОСТ 34.003-90, понимается свойство автоматизированной системы, характеризующее способность выполнять установленный объем функций в условиях воздействия внешней среды и отказов компонентов системы в заданных пределах.

С понятием живучести тесно связано также и такое свойство автоматизированной системы, как адаптивность, определяемое как способность системы изменяться для сохранения своих эксплуатационных показателей в заданных пределах при изменениях внешней среды.

*Функциональная живучесть* характеризует допустимые пределы снижения качества функционирования корпоративной распределенной информационной системы, т.е. эффективности выполнения системой заданного набора функций на протяжении всего жизненного цикла в условиях постоянной деградации ее ресурсов. Успешное выполнение системой в течение жизненного цикла всех предписанных функций и задач в конечном итоге и означает достижение заданных целей ее функционирования.

Исследование проблемы обеспечения живучести сложных технических систем требует, кроме всего прочего, проведения анализа уязвимостей и управления рисками.

Проблема обеспечения живучести на протяжении жизненного цикла связана с необходимостью поддерживать приемлемую (желательно - оптимальную) траекторию деградации функциональных системных возможностей по мере подготовки корпоративной распределенной информационной системы к выводу из эксплуатации и последующего демонтажа.

Живучесть обычно проявляется в условиях накопления в системе близкого к критическому множества отказов, когда возврат к прежним

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

режимам функционирования в силу внутренних и внешних обстоятельств представляется маловероятным. Например, может надолго выйти из строя важное устройство, либо на неопределенный период отказать каналы связи с некоторыми из корпоративных узлов обработки данных. Живучесть в этом случае может рассматриваться как функциональная надежность корпоративной распределенной информационной системы в долгосрочном плане, т.е. как динамическое функциональное соответствие системы изменяющимся условиям эксплуатации в течение длительного периода.

В результате закономерно возникают вопросы о том, какая система может считаться более живучей в функциональном плане? Как оценивать качественные и количественные показатели живучести, какие критерии применимы для такой оценки? Какие решения следует принимать службам эксплуатации для повышения степени живучести столь сложных систем, как современные корпоративной распределенной информационной системы?

В качестве ответа на первый вопрос можно предположить, что, по-видимому, более живучей следует считать систему, которая в наибольшей степени соответствует декларированным целям своего функционирования в условиях постепенного накопления отказов оборудования, моральной деградации программного обеспечения, ухудшения условий эксплуатации и воздействия других внешних неблагоприятных факторов на протяжении всего периода жизненного цикла.

Далее, из нескольких альтернативных вариантов более живучей может считаться система, способная выдержать на заданном временном отрезке большее число отказов или большее число тяжелых отказов по сравнению с другими аналогами, а также способная функционировать с качеством не ниже заданного в течение более длительного времени и в наиболее неблагоприятных условиях.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### **Критерии оценки живучести корпоративной распределенной информационной системы**

Учитывая высокую сложность и комплексный характер проблемы анализа надежности и живучести корпоративной распределенной информационной системы, логично предположить, что произвести оценку этих показателей только на основе одного какого-либо параметра крайне затруднительно. Поэтому в данном случае необходимо включать в рассмотрение некие интегральные многофакторные (многокритериальные) показатели, включающие как количественные, так и качественные характеристики, которые могут использоваться для формальной оценки ее живучести.

Функциональную живучесть корпоративной распределенной информационной системы целесообразно определять посредством оценки показателей качества ее функционирования в условиях возникновения отказов в процессе эксплуатации на протяжении всего жизненного цикла. В такой постановке качество выполнения системой своих функций можно оценивать, например, на основе следующих основных характеристик:

- соответствие корпоративной распределенной информационной системы целям и задачам корпоративного бизнеса;
- показатели производительности системы и отдельных ее узлов;
- функциональная готовность приложений и данных, связанная с показателями отклика (временем реакции системы);
- качество обслуживания пользователей и приложений QoS (качество обслуживания (англ. Quality of Service));
- рациональное использование информационно-коммуникационных – ресурсов, оцениваемое с позиций факторов TCO и ROI (TCO – совокупная стоимость владения активом (англ. Total Cost of Ownership), ROI – возврат на инвестиции в проект (англ. Return of Investment)).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

При проектировании корпоративной распределенной информационной системы, обладающей требуемой высокой степенью функциональной надежности и живучести, изначально следует определить соответствующие критерии для оценки различных системных качеств. Эти критерии целесообразно объединять по группам в соответствии с определенными принципами, характеризующими различные конструктивно-технологические аспекты, факторы и механизмы обеспечения живучести корпоративной распределенной информационной системы.

Группы могут формироваться, например, по следующим признакам:

1. критерии соответствия системы заданным показателям качества функционирования и/или оценки степени ее функциональной (физической и моральной) деградации;
2. критерии для оценки эффективности динамической реконфигурации и перераспределения информационно-коммуникационных ресурсов, а также динамики восстановления функциональных возможностей системы после сбоев;
3. критерии, характеризующие изменение производительности и реактивности системы при выполнении различных типов приложений в условиях деградации системных ресурсов;
4. критерии экономической эффективности использования информационно-коммуникационных активов.

**1.** К первой группе могут быть отнесены критерии, характеризующие:

- долю физически и морально устаревшего оборудования в составе инфраструктуры корпоративной распределенной информационной системы (серверных платформ, рабочих станций, сетевых устройств и проч.) по отношению к общему объему парка технических средств;

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

- долю морально устаревшего программного обеспечения в общем объеме прикладного и общесистемного программного обеспечения корпоративной распределенной информационной системы;
- долю гибкого, переносимого и хорошо масштабируемого сервис-ориентированного программного обеспечения по отношению к «монолитным» корпоративным приложениям класса ERP (Система планирования ресурсов предприятия (англ. Enterprise Resource Planning System));
- вероятность потери функциональности из-за невозможности выполнения некоторых ответственных приложений на оставшихся информационно-коммуникационных ресурсах в условиях выхода из строя части инфраструктуры;
- соответствие характеристик и параметров системы в течение жизненного цикла требуемым значениям показателей качества обслуживания типа QoS, заданным в соглашениях по SLA (соглашение об уровне сервиса (англ. Service Level Agreement));
- темпы развития/деградации инфраструктуры корпоративной распределенной информационной системы на протяжении определенного периода или всего жизненного цикла системы;
- нечеткие критерии, характеризующие качество функционирования корпоративной распределенной информационной системы и степень ее соответствия целям и задачам корпоративного бизнеса, построенные, как правило, на основе экспертных оценок.

2. Вторая группа объединяет критерии, характеризующие способность эксплуатируемой системы противостоять возможным откатам путем реконфигурации и адаптации к новым условиям функционирования:

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

- эффективность методов и средств виртуализации информационно-коммуникационных ресурсов, а также их объединения для решения критически важных и ответственных задач;
- эффективность перераспределения активов корпоративной распределенной информационной системы в целях сохранения максимально возможной степени функциональности системы;
- критерии для оценки адаптационных способностей корпоративной распределенной информационной системы к новым функциональным задачам и возможным архитектурным перестройкам;
- критерии для оценки скорости адаптации системы к условиям изменяющейся нагрузки;
- критерии оценки степени функциональной деградации вследствие роста требований к качеству функционирования, а также из-за изменения (ухудшения, ужесточения) других условий эксплуатации;
- критерии для оценки динамики (темпов) реконфигурации и адаптации системы, а также длительности восстановления функциональных возможностей.

**3.** К третьей группе целесообразно относить вероятностные критерии, позволяющие адекватно оценить:

- вероятность и параметры снижения производительности и реактивности системы при изменении конфигурации вследствие отказов, сбоев или вывода части информационно-коммуникационных ресурсов из эксплуатации (временного или постоянного);
- требования к активным системным информационно-коммуникационным ресурсам, необходимым для обеспечения

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
значений показателей по производительности и реактивности, адекватных входному трафику и характеру запросов на обслуживание;

- характеристики, связанные с вероятностными оценками удовлетворения системой требований по производительности и реактивности в условиях изменения нагрузки путем оптимизации конфигурации информационно-коммуникационных ресурсов;
- вероятностные оценки успешного выполнения ответственных и критически важных приложений в условиях прогнозируемых сбоев, отказов и деградации системных ресурсов.

4. Наконец, последняя группа содержит критерии, объединенные целью эффективного использования имеющихся в корпоративной распределенной информационной системе информационно-коммуникационных ресурсов, а именно:

- критерии для оценки рациональности вложения средств для модернизации/ утилизации информационно-коммуникационных ресурсов в связи с изменением внешней бизнес-среды или принятием принципиально иной корпоративной стратегии ведения бизнеса, что в свою очередь связано с неизбежным переопределением бизнес-целей;
- технико-экономические критерии типа ROI и TCO с отслеживанием динамики их изменений на протяжении всего жизненного цикла корпоративной распределенной информационной системы;
- показатели для оценки минимально необходимого уровня текущего финансирования для сохранения функциональных возможностей системы на требуемом уровне вплоть до ее вывода из эксплуатации и последующей утилизации.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Суммируя все вышесказанное, следует отметить, что именно живучесть может рассматриваться как наиболее объективный и адекватный показатель, позволяющий наилучшим образом оценить все аспекты структурно – функциональной надежности сложных технических систем, находящейся в постоянно изменяющейся внешней среде и подвергающейся перманентным модернизациям с целью улучшения показателей качества ее функционирования.

Действительно, при исследовании живучести акцент делается не на единичные сбои и отказы, вызывающие временную неработоспособность системы, а на ее способность выполнять свои функции в течение длительного периода времени, желательно – на протяжении всего жизненного цикла корпоративной распределенной информационной системы.

Конечной целью исследований в области изучения факторов живучести корпоративной распределенной информационной системы является построение высоконадежных систем, обладающих необходимыми адаптационными свойствами на случай неблагоприятного изменения внешней ситуации, например, резкого ухудшения условий эксплуатации или изменения целей корпоративного бизнеса.

Таким образом, понятие живучести есть смысл формулировать для каждой конкретной распределенной информационной системы отдельно, поскольку последние имеют различную сложность, особенности, задачи и требования к их выполнению. При этом в определении должны фигурировать такие ключевые термины как система, угроза, адаптируемость, непрерывность обслуживания и время.

Существует множество терминов, характеризующих состояние системы и ее реакцию на внешние воздействия и неблагоприятные факторы, оценивать по которым информационные системы не всегда удобно и возможно. Поэтому должен существовать единый показатель, который будет характеризовать распределенную информационную систему с этих позиций. Представляется целесообразным в качестве такого наиболее содержательного интегрального показателя использовать «живучесть».

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### Оценка живучести сетевых структур

В инфраструктуре современного информационного общества распределенные информационные системы занимают одно из ключевых мест. Это вызвано возрастающей ролью информации в наукоемком промышленном производстве. Информация в современных условиях выступает, как ресурс, позволяющий минимизировать расходы других ресурсов (сырьевых, материальных, энергетических, трудовых, финансовых).

Требования к качеству информации в современных условиях настолько возросли, что трудно представить нормальное функционирование общества без соответствующего информационного обеспечения. Например, в условиях рыночной экономики особое значение приобретают достоверность и полнота информации, без которых невозможна маркетинговая, финансовая и инвестиционная деятельность.

С другой стороны, в сферах телекоммуникаций, энергетики, в вопросах водоснабжения и водоотведения крупных промышленных предприятий или мегаполисов на первое место выходят вопросы анализа существующих (или вновь проектируемых) сетевых структур с точки зрения живучести при функционировании в режиме экстремальных нагрузок или быстрого роста абонентов – потребителей того или иного вида продукта, передаваемого по данным сетям.

Типичными задачами, возникающими перед аналитиками и инженерами-проектировщиками, являются расчет распределения потоков внутри информационной сети передачи данных при воздействии на сетевую структуру неблагоприятных внешних факторов (например, стихийных бедствий), выявление «узких мест» распределенной информационной системы (каналов, подверженных перегрузке, узлов, отказывающихся в обслуживании при увеличении нагрузки и т.д.).

Учитывая, что задачи анализа и синтеза сетевых структур средней и большой размерности являются NP-сложными, а для их решения часто приходится строить отдельную модель, объемы затрачиваемого

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

на расчеты времени, различных физических ресурсов могут быть велики.

Принципиальная схема функционирования сетевой структуры формализуется известной математической моделью, которая называется многопродуктовой потоковой сетью (МП-сетью) и задается с помощью графа.

Сетевая информационная система представляет собой распределенную структуру, размещенную на большой территории. Схема функционирования ее задается с помощью графа, который определяет физическую структуру распределенной информационной системы, его ребра соответствуют физическим компонентам распределенной информационной системы (таким, как каналы связи), проложенным от одной вершины графа (узла) к другому (Рисунок 2.1.2). Каждое ребро, входящее в вершину или исходящее из нее, называется инцидентным этой вершине.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

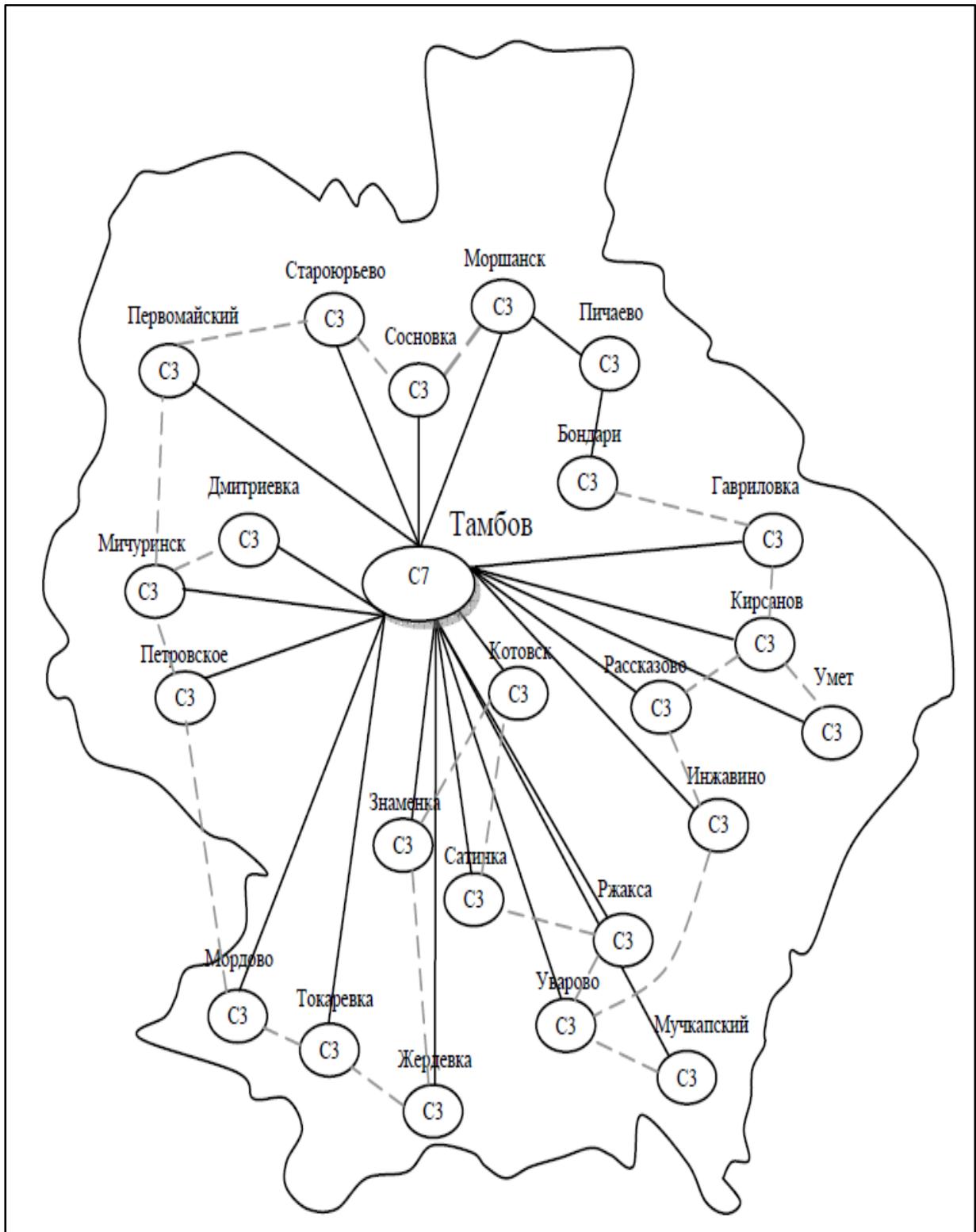


Рисунок 2.1.2 – Пример схемы распределенной информационной системы

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Узлы распределенной информационной системы соответствуют источникам/приемникам потоков либо осуществляют транзитные функции для существующих потоков.

Совокупность ребер, которые надо пройти потоку из вершины  $V_i$  до вершины  $V_j$ , называется путем  $(V_i, V_j)$ .

Если элементу графа (ребру, вершине) приписана какая-либо физическая величина (например, длина ребра, пропускная способность, задержка обработки информации), эта величина отмечается числом, называемым весом элемента (ребра, вершины).

Длина пути между вершинами определяется матрицей расстояний  $|V_i, V_j|$ : выбирается  $j$ -й столбец и суммируются по  $i$  все длины ребер, расположенные в столбце.

Вершина (или точка на ребре), расположенная на наименьшем расстоянии от всех остальных вершин, называется медианой графа, медианное расстояние  $R$  – радиусом графа.

Удаление даже одного ребра увеличивает радиус графа, так как в таком случае необходимо отыскать обходной и потому более длинный путь. Таким образом, удаление одного или даже нескольких ребер не всегда уничтожает связность графа. Такое свойство графа носит название живучести.

Построение графа осуществляется последовательным соединением смежных вершин, не превышая при этом их степени.

Введем следующие обозначения и допущения:

- $n$  – число вершин графа  $V_i$ ,  $V_i = 1, \dots, n$ ;
- $m$  – число ребер графа  $(V_i, V_j)$ ;
- $d(i)$  – степень вершины  $V_i$  графа;
- граф не имеет кратных ребер;
- все ребра графа имеют одинаковый вес, равный 1.

Условие связности графа:

$$m \geq n - 1.$$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Удаление всех ребер, инцидентных некоторой вершине, изолирует ее, прерывая все пути к другим вершинам – граф становится не связным, живучесть графа – равной нулю.

Для обеспечения наибольшей живучести надо строить граф с наибольшей степенью  $d(i)$  всех его вершин. Таким является полный граф, в котором каждая вершина связана ребром непосредственно с каждой другой вершиной. Число ребер

$$m = \frac{n \cdot (n - 1)}{2} \quad (2.1)$$

Каждая вершина имеет максимальную степень  $d = n - 1$ , но на практике не все вершины нуждаются в подобной «защите», такой усиленный граф нерентабелен. Необходимо создать такой граф из  $n$  вершин, чтобы каждая из них имела заданную ей степень  $K_{a_i}, i = 1, \dots, n$ , где  $1 \leq K_{a_i} \leq n - 1$ .

Задержки каналов, узлов, отказы технических элементов могут привести к блокаде значительного участка распределенной информационной системы, но не разрушить ее. Распределенная информационная система останется живучей, но временно бездействующей.

Имеется иной тип воздействий на распределенную информационную систему – внешний, и потому совершенно непредсказуемый (например, события, носящие стихийный характер).

Значительная часть распределенной информационной системы при таком типе воздействия может быть разрушена, но оставшиеся связанные между собою сегменты функционируют, пусть и в «усеченном» режиме. Исследованию процессов, протекающих при этом, посвящена теория связности и живучести распределенной информационной системы.

Рассмотрим несколько графовых характеристик, связанных с проблематикой живучести распределенной информационной системы.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

### *K-связность как мера живучести распределенной информационной системы*

Неориентированный граф  $G = (V, R)$  называется  $k$ -связным относительно пары вершин  $V_i, V_j \in V$ , если после удаления любых  $(k - 1)$  ребер обязательно останется путь, соединяющий вершины  $V_i$  и  $V_j$ .

Граф  $G$  называется  $k$ -связным, если он является  $k$ -связным относительно каждой пары своих вершин. В  $k$ -связном графе для любой пары вершин существует не менее  $k$  реберно-непересекающихся путей их соединения.

Основываясь на этих определениях, можно поставить задачу синтеза графа гарантированной высокой живучести.

### *Живучесть и диаметр графа распределенной информационной системы*

В реально функционирующих информационных сетях используется ограничение на число переключений одного сообщения. Соответственно, в модели такой распределенной информационной системы будут считаться связными только те пары узлов (вершин графа), между которыми существует путь, имеющий длину не более заданной.

При анализе подобных распределенных информационных систем используется понятие диаметра графа.

При анализе живучести распределенных информационных систем используют также верхние и нижние оценки диаметра графа. Пусть в графе  $G$  найдены  $L(V_i, V_j)$  - длины кратчайших путей между всеми парами вершин  $V_i, V_j \in V$ .

Тогда величину  $L = \max_{V_i, V_j \in V} L(V_i, V_j)$  называют диаметром графа.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

### *Условная связность*

Введением понятия диаметра графа на понятие связности было наложено определенное ограничение. Вершины графа СИС считаются связными, если длина соединяющего их пути не превосходит заданной величины. Например, граф  $G$  считается условно связным, если удаление некоторого минимального числа ребер (вершин) оставляет в образовавшихся компонентах присутствующие исходному графу свойства, например, заданную степень вершин.

### *Стойкость*

При синтезе распределенной информационной системы максимальной живучести возникает вопрос о минимальной величине затрат, обеспечивающих эту живучесть, т.е. проблема стойкости. Стойкость численно равна наименьшей средней стоимости создания новой компоненты связности. Было получено соотношение между стойкостью графа и его живучестью.

Если стойкость графа  $\sigma(G) \geq \sigma_0$ , то граф содержит не менее  $\sigma_0$  реберно-непересекающихся остовных деревьев. При этом было найдено важное приложение к определению живучести – вычисление плотности графа.

Пусть граф  $G^* = (V^*, R^*)$  - подграф графа  $G$ .

Плотностью  $\rho(G^*)$  подграфа  $G^*$  называется отношение мощности множества его ребер к мощности множества его вершин:

$$\rho(G^*) = |R^*/V^*|.$$

Плотные графы являются менее уязвимыми.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

### *Минимальный разрез как характеристика уязвимости распределенной информационной системы*

При построении процедурной модели живучести распределенной информационной системы под воздействием внешних неблагоприятных факторов для моделирования полного разрушения структуры пытаются разделить так называемые «тяготеющие пары», т.е. удалить множество таких ребер, что их удаление из графа распределенной информационной системы разрушает все пути соединения для всех тяготеющих пар (создают разрез).

Пропускная способность такого разреза равна сумме пропускных способностей всех входящих в него ребер. Минимальный разрез – разрез с минимальной пропускной способностью (т.е. включающий в себя наименьшее число ребер). При моделировании считается, что именно этот разрез будет подвергнут наибольшему воздействию неблагоприятных факторов и именно этот разрез укрепляют.

Метод отыскания минимального разреза в общем виде неэффективен. Лишь в некоторых специальных случаях его поиск сводится к простым комбинаторным задачам. При разрушении части распределенной информационной системы происходит перераспределение (перемаршрутизация) потоков.

Отмеченные выше показатели структурной живучести мало представительны: в основу их построения полагается лишь один из многих аргументов целевой функции живучести графа, как правило – связность. Отыскивается наиболее слабое звено графа, определяется минимальное сечение, которое, в той или иной степени, используется в выражении показателя живучести. Так называемое гарантированное значение живучести графа распределенной информационной системы задается наилучшим состоянием деформирования графа.

При удалении ребер одна из вершин графа оказывается в изоляции, но, если ее заранее объединить с какой-либо устойчивой вершиной (stable node), связность графа удастся сохранить. Продолжая далее

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

этот часто употребляемый прием удаления и контракции ребер, сводят исходный граф  $G$  к тривиальной петле.

Если всем удалениям ребер приписать одинаковую вероятность удаления  $p$ , то контракция их будет иметь вероятность  $q = 1 - p$ .

В результате совокупности всех таких действий создается многочлен из произведений  $p$  и  $q$  различной степени. Численное значение такого многочлена при заданном значении  $p$  принимают за критерий живучести  $R(G)$  графа  $G$ .

$$R(G) = \begin{cases} (1 - p(e))R(G/e); \\ R(G \setminus e); \\ p(e)R(G \setminus e) + (1 - p(e))R(G/e) \end{cases}$$

(2.2)

где «\» - знак удаления; «/» - контракции.

Применив  $R(G)$  к графам с различным содержанием ребер, получают график зависимости живучести от вероятности удаления ребра (Рисунок 2.1.3).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

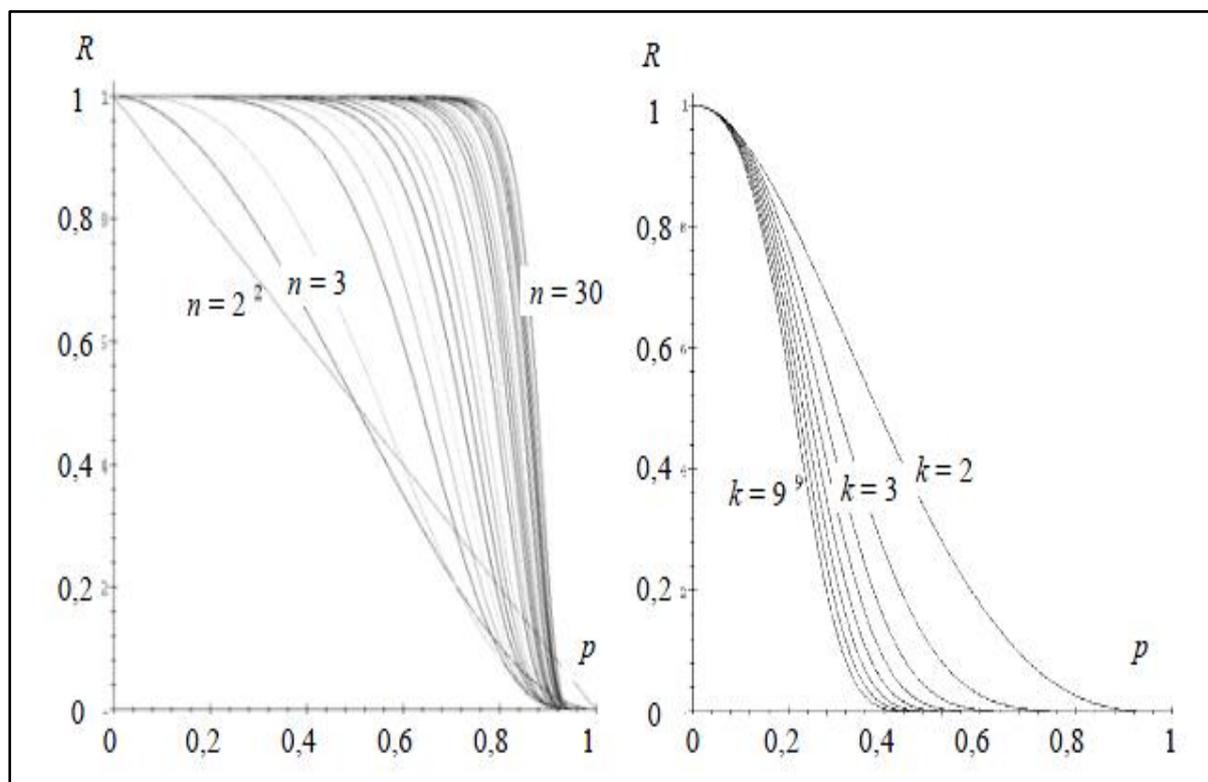


Рисунок 2.1.3 – График зависимости живучести от вероятности удаления ребра

Живучесть графа определяется только числом ребер и вероятностью их удаления при воздействии на физический граф распределенной информационной системы внешних неблагоприятных факторов.

Предлагаемая модель универсальна и приложима к любым графам. Численные значения критерия живучести не совпадают с тем, что предлагают другие исследователи, использующие вероятностные и теоретико-графовые подходы, но едва ли это отличие велико, к тому же все оценки живучести весьма условны.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### Модель оценки живучести распределенной информационной системы с радиальной топологией

Рассмотрим модель распределенной информационной системы на примере системы с радиальной топологией (как известно, частный случай древовидной структуры), изображенной на рисунке 2.1.4.

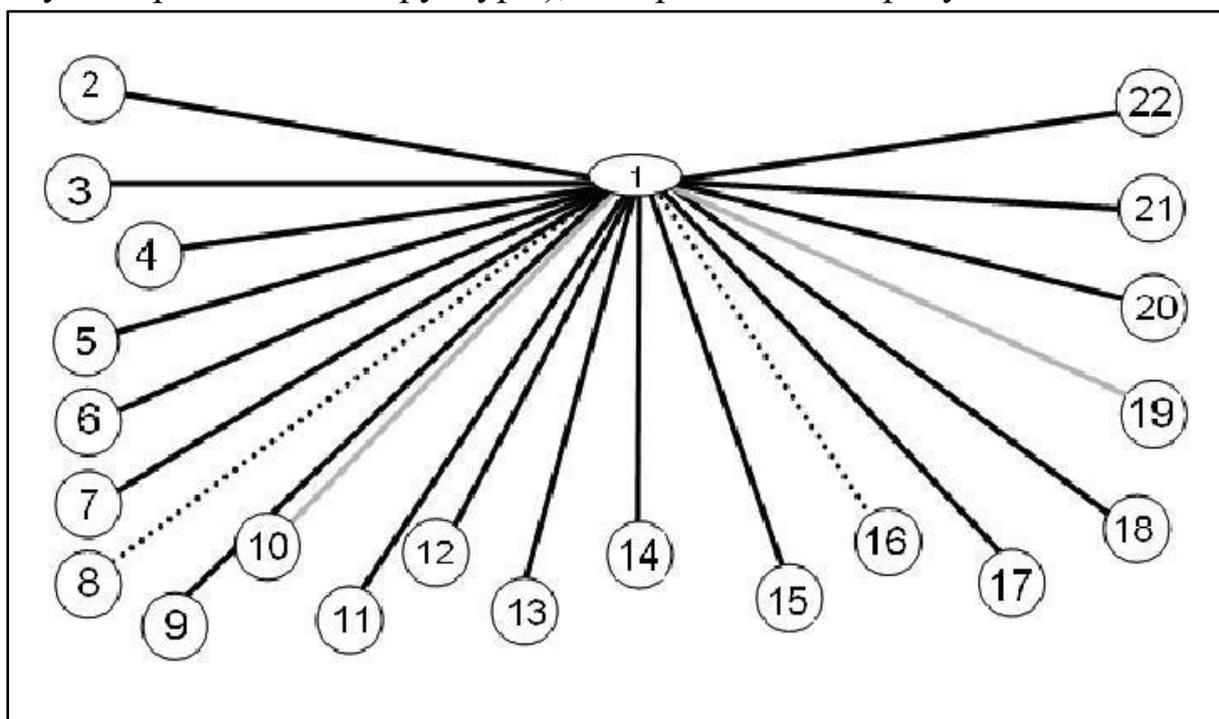


Рисунок 2.1.4 – Схема радиальной топологии распределенной информационной системы

Предполагается, что искомая распределенная информационная система имеет продуктивные способности, достаточные для своевременной доставки информации к транзитному узлу или структурному элементу от центрального управляющего узла за единицу времени.

Положим заданными следующие допущения:

- в распределенной информационной системе решается множество задач за установленные директивные сроки;

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

- распределенная информационная система имеет радиальную топологию с центральным управляющим узлом и транзитными узлами, взаимодействующими по коммуникационным каналам;
- каждый транзитный узел представляет собой узел одного варианта;
- каждая задача может быть реализована только на одном транзитном узле;
- производительность транзитного узла в каждом сегменте должна обеспечивать выполнение всех задач, поступивших от центрального управляющего узла;
- время выполнения любой задачи не должно превышать заданный директивный срок;
- живучесть распределенной информационной системы обеспечивается наличием некоторых значений минимальных значений безотказно действующих транзитных узлов и коммуникационных каналов.

Используя вышеприведенные допущения, построим модель живучести распределенной информационной системы радиальной топологии, выделив следующие этапы:

1. Задается множество  $H = \{1, \dots, m\}$  сложных задач, где при выполнении каждой из них необходимо отработать  $V_i$  объема информации за директивные сроки  $d_i$ .
2. Распределенная информационная система состоит из  $L$  транзитных узлов ( $L = \{1, \dots, l, \dots, L\}$ )  $r$ -го варианта построения,  $R = \{1, \dots, r, \dots, a_1\}$ , производительностью  $W_r$ , байт/с, и  $R^{\wedge}$  вариантов построения коммуникационных каналов,  $R^{\wedge} = \{1, \dots, r^{\wedge}, \dots, a_2\}$ .

Живучесть распределенной информационной системы обеспечивается наличием некоторых значений минимальных значений безотказно действующих транзитных узлов и коммуникационных каналов.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Безотказная работа транзитного узла  $r$ -го варианта в течение времени  $T$  определяется формулой

$$P_r(T) = \sum_{k=0}^{q_r} C_{n_r} P^k (1 - P)^{n_r - k},$$

где  $p = 1 - e^{-\lambda_r t}$  ;

$q_r$  – максимально допустимое число отказавших элементов транзитного узла  $r$ -го типа;

$n_r$  – число элементов в момент времени  $t = 0$ ;

$\lambda_r$  – интенсивность отказов компонентов транзитного узла  $r$ -го варианта;

$P_r(T)$  – показатель безотказной работы транзитного узла,

$d_i$  – директивный срок выполнения задачи,

$C_{n_r}$  – нормировочный коэффициент.

Безотказная работа коммуникационных каналов после воздействия неблагоприятных факторов (живучесть коммуникационных каналов) определяется с помощью выражения:

$$P_{ж}^* = \frac{\mu_y(\mu_y n k - 2\pi\lambda)}{\mu_y(\mu_y n n - 2\pi\lambda) + P_{nc} V_y \pi (2\mu_y n n - \pi\lambda)}$$

где  $n$  и  $k$  – соответственно минимально-допустимые числа исправно работающих центральных управляющих узлов и коммуникационных каналов;

$\mu_y = \frac{W(T)}{W^0}$  – относительная текущая интенсивность обслуживания заявок ( $W^0$  – номинальное значение интенсивности);

$\pi$  – средняя длина маршрута распределенной информационной системы;

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

$P_{nc}$  – коэффициент поражения распределенной информационной системы от неблагоприятных факторов;

$\lambda$  – суммарный объем исходящей за пределы распределенной информационной системы информации;

$V_y = \frac{V(T)}{V^0}$  – относительная интенсивность старения информации.

Для успешной реализации задачи в течение времени  $T$  требуется, чтобы живучесть распределенной информационной системы удовлетворяла соотношению:

$$\sum_{r \in R} P_{жr}(T) Y_{lr} \geq P_{ж}^*$$

здесь  $Y_{lr}$  – ценность транзитных узлов варианта  $r$ , расположенного в  $l$ -том сегменте.

### **Метод расчета приоритетов функциональных узлов распределенной информационной системы**

В вышеописанной модели используется такое понятие как ценность узла. Применяемые для их расчета методы экспертных оценок обладают большой ресурсоемкостью и не позволяют получить объективные количественные показатели. Предлагается метод расчета приоритетов функциональных элементов распределенной информационной системы, основанный на степени их значимости для обеспечения транспорта заданному (конечному) множеству прикладных информационных процессов.

Введем необходимые обозначения:

$S = \{s_1, \dots, s_n\}$  – множество всех функциональных элементов распределенной информационной системы (рабочие станции пользователей, сервера, телекоммуникационное оборудование),  $S_r$  – элемент данного множества;

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

$P = \{p_1, \dots, p_t\}$  - множество всех информационных процессов,  $p_i$  - элемент данного множества.

Введем величину  $R$  - показатель значимости (приоритет) элементов  $S$  для выполнения информационных процессов распределенной информационной системы.

Обозначим как  $R_r$  количественное значение приоритета элемента  $s_r$ .

Обозначим как  $R_r^i$  количественное значение приоритета элемента  $s_r$  для информационного процесса  $p_i$ .

Информационный процесс распределенной информационной системы будем трактовать как информационное взаимодействие пользователей структурных подразделений корпорации, с целью обработки информации в узлах распределенной информационной системы в контексте решаемой прикладной задачи. Реализация информационного процесса связана с использованием функциональных элементов  $S$ .

Представим множество  $P = \{p_1, \dots, p_t\}$  как связный неориентированный граф  $N(S, L)$ , где множество вершин графа - множество функциональных элементов распределенной информационной системы, множество ребер графа - каналы электросвязи распределенной информационной системы (Рисунок 2.1.5).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

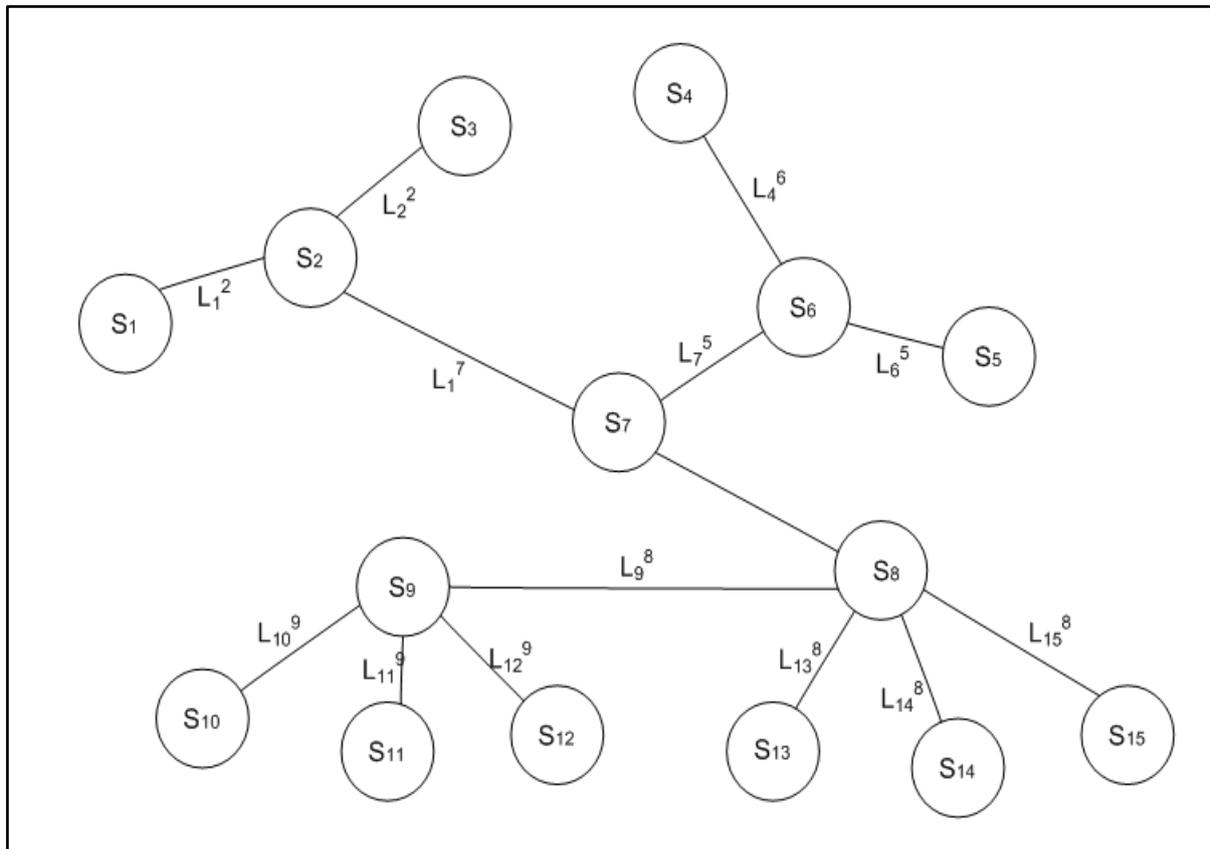


Рисунок 2.1.5 – Пример графа распределенной информационной системы

Как упорядоченное подмножество  $S$ , информационный процесс  $p_i$  будет представлять собой ориентированный подграф  $n_i$  искомого графа (Рисунок 2.1.6).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

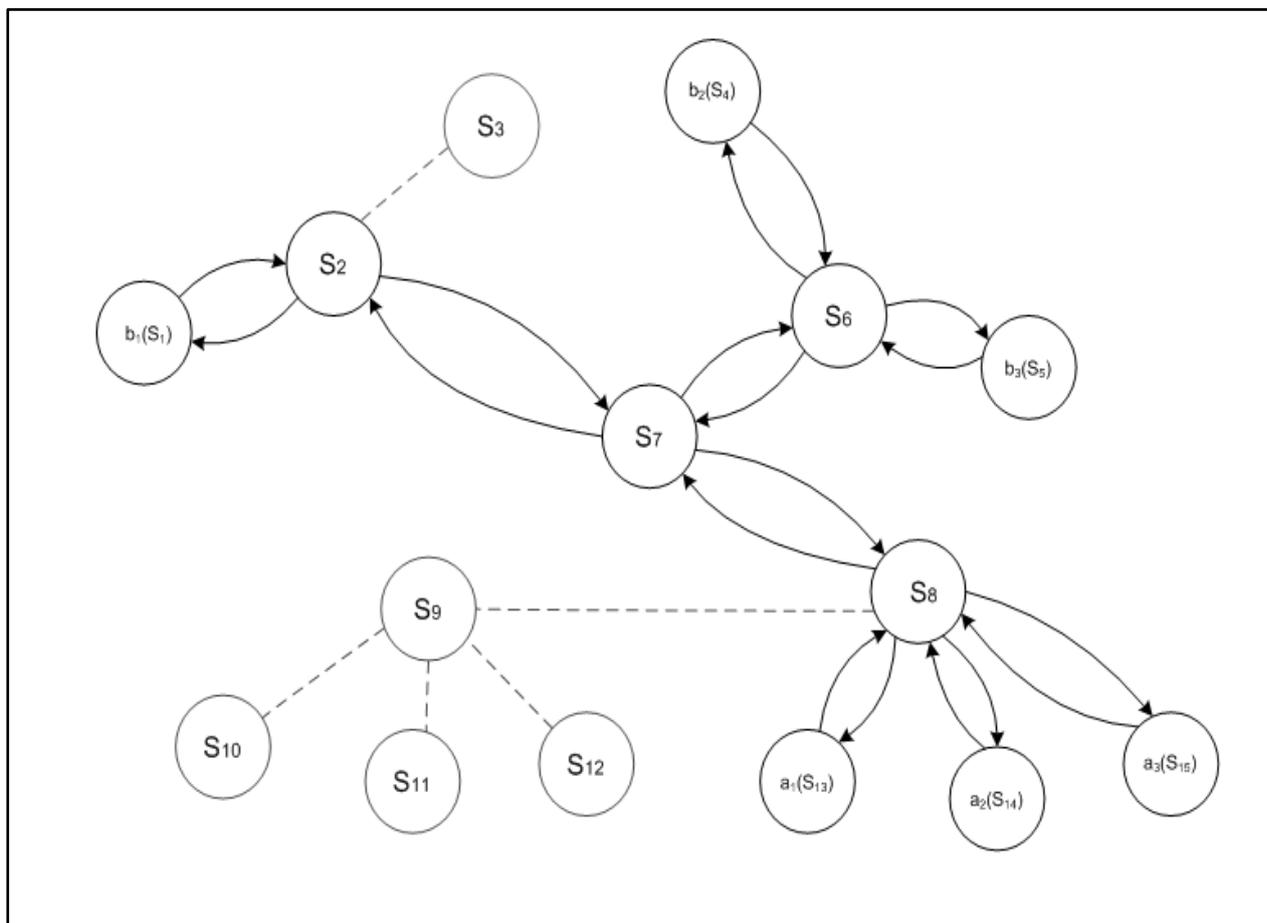


Рисунок 2.1.6 – Ориентированный подграф  $n_i$  информационного процесса  $p_i$

В общем случае  $p_i$  характеризуется четверкой  $\langle H_i, A_i, B_i, W_i \rangle$ , где

- $H_i$  - количественная оценка рангов информационного процесса  $p_i$  определяемая экспертной группой в соответствии со степенью важности и срочности;
- $A_i$  - множество потенциальных абонентов-отправителей процесса  $p_i$   $A_i = \{a_1^i, \dots, a_k^i\}$ ,
- $B_i$  - множество потенциальных абонентов-получателей процесса  $p_i$   $B_i = \{b_1^i, \dots, b_d^i\}$ ;

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

- $W_i$  - множество элементарных ориентированных путей по графу  $n_i$  (все потенциальные пути прохождения сетевого трафика между абонентами, легитимные с точки зрения сетей передачи данных - множество альтернативных способов исполнения  $p_i$ ),  $W_i = \{w_1^i, \dots, w_p^i\}$ , где за  $w_q^i$  обозначим элемент данного множества.

Например,  $w_q^i = \{a_1^i, \dots, s_r, \dots, b_d^i\}$  - путь информационного процесса  $p_i$  от узла абонента отправителя данных  $a_1^i$  к узлу абоненту получателю данных  $b_d^i$ . Запись  $(\dots s_r \dots)$  означает разные транзитные пути.

Процесс  $p_i$  является исполняемым в условиях хотя бы одного  $w_1^i$  для сочетания отправитель-получатель.

Степень участия элемента в реализации  $p_i$  есть частота его появления в общем количестве альтернативных элементарных путей ( $W_i$ ) и обозначается как  $\gamma$ .

Численное значение приоритета  $R_r^i$ , согласно предлагаемой модели, пропорционально  $\gamma_r^i$  элемента  $s_r$  в обеспечении транспорта при исполнении рассматриваемого процесса  $p_i$  и его (процесса) рангу  $H_i$ :

$$R_r^i = \gamma_r^i \times H_i.$$

Построим матрицу приоритетов  $\|R_r^i\|$  элементов  $S = \{s_1, \dots, s_n\}$  для каждого информационного процесса  $p_i$ .

Таблица 2.1.1 – Матрица приоритетов

	$p_1$	$p_2$	$p_3$	...	...	...	$p_t$
$s_1$	$R_1^1$	$R_1^2$	$R_1^3$	...	...	...	$R_1^t$
$s_2$	$R_2^1$	$R_2^2$	$R_2^3$				$R_2^t$
$s_3$	$R_3^1$	$R_3^2$	$R_3^3$				$R_3^t$
...	...	...	...	...	...	...	...
$s_n$	$R_n^1$	$R_n^2$	$R_n^3$	...	...	...	$R_n^t$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

В соответствии с таблицей 2.1.1 мы приходим к следующему. Каждому элементу  $s_r$  сопоставим  $t$ -вектор  $\{R_r^1, R_r^2, \dots, R_r^t\}$  приоритетов элемента в распределенной информационной системе для каждого из информационных процессов. С использованием Евклидовой метрики приоритет  $s_r$  (по всем информационным процессам) можно определить с применением весовой нормы:

$$R_r = \sqrt{(R_r^1)^2 + (R_r^2)^2 + \dots + (R_r^t)^2} = \sqrt{\sum_{h=1}^t (R_r^h)^2} .$$

где  $R_r$  — приоритет для элемента  $s_r$ ,  
 $h$  — номер информационного процесса.

Здесь  $R_r = 0$  соответствует элементу, не принадлежащему множеству  $P = \{p_1, \dots, p_t\}$ .

Множество всех альтернативных сочетаний между абонентом-отправителем и абонентом-получателем в рамках информационного процесса  $p_i$  назовем множеством пар и обозначим как  $M_i = \{m_1^i, \dots, m_z^i\}$ . За  $m_k^i$  обозначим элемент данного множества.

Обозначим как  $n_{M_i}$  количество элементов множества  $M_i$  (всех пар, характерных для  $p_i$ ). Обозначим как  $n_{A_i}$  количество абонентов-отправителей. Обозначим как  $n_{B_i}$  количество абонентов-получателей.

Рассчитаем  $n_{M_i}$  всех возможных пар отправитель-получатель для исследуемого процесса  $p_i$  по формуле:

$$n_{M_i} = n_{A_i} \times n_{B_i} .$$

Найдем  $W_i$  - множество элементарных ориентированных путей для  $n_{M_i}$  пар по графу  $p_i$ , используя математический аппарат теории графов.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

За  $W_i^{m_k^i}$  обозначим множество элементарных ориентированных путей по графу  $p_i$  пары  $m_k^i$ ,  $W_i^{m_k^i} \in W_i$ .

За  $n_{W_i^{m_k^i}}$  обозначим количество всех альтернативных путей по графу  $n_i$  информационного процесса  $p_i$  пары  $m_k^i$ .

За  $n_{W_i}$  обозначим общее количество всех альтернативных путей исполнения информационного процесса  $p_i$ :

$$n_{W_i} = \sum_{k=1}^{n_{M_i}} n_{W_i^k},$$

где  $k = 1, \dots, n_{M_i}$  - номер элемента  $M_i$  (пары).

За  $n_{W_i^{m_k^i}}^r$  обозначим количество всех альтернативных путей по графу  $n_i$  информационного процесса  $p_i$  пары  $m_k^i$  проходящих через элемент  $S_r$ .

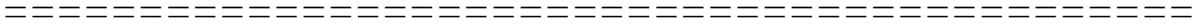
За  $n_{W_i}^r$  обозначим общее количество альтернативных путей исполнения  $p_i$  проходящих через элемент  $S_r$ :

$$n_{W_i}^r = \sum_{k=1}^{n_{M_i}} n_{W_i^{m_k^i}}^r.$$

Определим  $\gamma_r^i$ , как отношение общего количества альтернативных путей исполнения информационного процесса  $p_i$ , проходящих через элемент  $S_r$  ( $n_{W_i}^r$ ) к общему количеству всех альтернативных путей исполнения  $p_i$  ( $n_{W_i}$ ):

$$\gamma_r^i = \frac{n_{W_i}^r}{n_{W_i}}$$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ



Определим общую частоту использования  $S_r$  элемента -  $\gamma_r^i$  :

$$\gamma_r^i = \frac{\sum_{k=1}^{n_{M_i}} (n_{W_i^k})}{\sum_{k=1}^{n_{M_i}} (n_{W_i^k}^r)}$$

Определим  $R_r^i$  элемента  $S_r$  для исполнения информационного процесса  $p_i$ :

$$R_r^i = \frac{\sum_{k=1}^{n_{M_i}} (n_{W_i^k})}{\sum_{k=1}^{n_{M_i}} (n_{W_i^k}^r)} \times H_i.$$

Определим  $R_r$  элемента  $S_r$ :

$$R_r = \sqrt{\sum_{h=1}^t \left( \frac{\sum_{k=1}^{n_{M_i}} (n_{W_i^k})}{\sum_{k=1}^{n_{M_i}} (n_{W_i^k}^r)} \times H_i \right)^2}.$$

### Оценка живучести кольцевых структур

Наибольшей живучестью обладают полносвязные сети, представляющие собой кольца с различным числом дополнительных каналов связи.

Вероятность связности для кольцевых сетей связи, т.е. сетей, у которых степень для каждой вершины равна 2 (степенью вершины  $d$  называются число граней графа сети, инцидентных данной вершине),

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

$$p_c = p^n + np^{n-1}q.$$

На практике довольно редко встречаются полностью связанные сети. Обычно бывают сети с небольшими степенями вершин. Имеется большое семейство графов (так называемых равнопрочных), в которых степень вершины  $d$ , число вершин  $n$  и общее число граней  $m$  связаны следующим соотношением:  $d=2m/n$  (при  $n>2$ ).

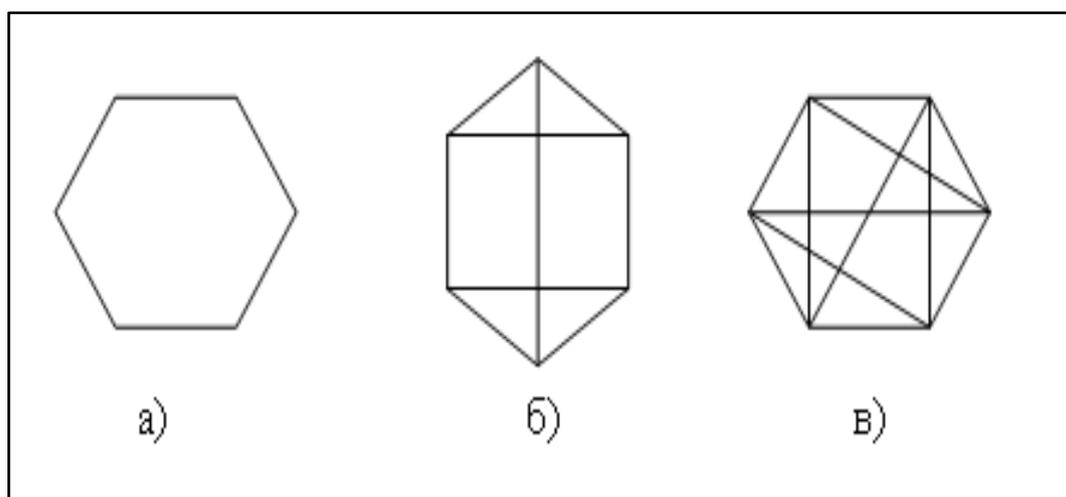


Рисунок 2.1.7 – Сеть «шестиугольник» с разной степенью вершин

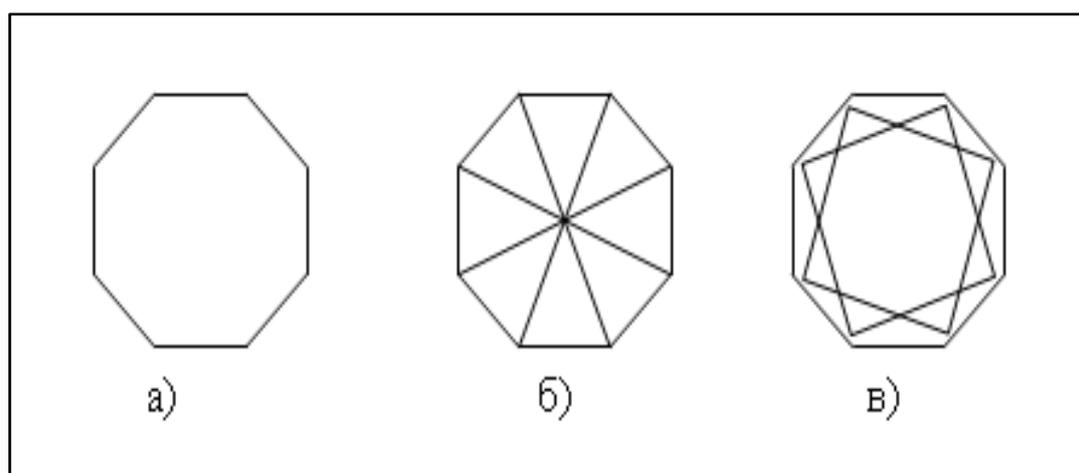


Рисунок 2.1.8 – Сеть «восьмиугольник» с разной степенью вершин

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Например, для шестиугольника ( $n=6$ ) без резервирования связей можно построить четыре различных графа с  $d=2, 3, 4, 5$ . Вероятности связности этих графов определяется следующими выражениями:

При  $d=2$  (рисунок 2.1.7, а)

$$p_c = p^6 + 6p^5q;$$

при  $d=3$  (рисунок 2.1.7, б)

$$p_c = p^9 + C_9^1 p^8 q + C_9^2 p^7 q^2 + [(C_9^3 - (n+1))] p^6 q^3 + [C_9^4 - (n-1)C_6^1] p^5 q^4;$$

при  $d=4$  (рисунок 2.1.7, в)

$$p_c = p^{12} + C_{12}^1 p^{11} q + C_{12}^2 p^{10} q^2 + C_{12}^3 p^9 q^3 + (C_{12}^4 - n) p^8 q^4 + (C_{12}^5 - nC_8^1) p^7 q^5 + [C_{12}^6 - nC_8^2 - (n+2)] p^6 q^6 + [C_{12}^7 - nC_8^3 - (n+2)C_6^1] p^5 q^7;$$

При  $n=8$  можно построить шесть различных графов с  $d=2, \dots, 7$ ; вероятность связности этих графов определится следующими выражениями:

$d=2$  (рисунок 2.1.8, а)

$$p_c = p^8 + 8p^7q;$$

$d=3$  (рисунок 2.1.8, б)

$$p_c = p^{12} + C_{12}^1 p^{11} q + C_{12}^2 p^{10} q^2 + (C_{12}^3 - n) p^9 q^3 + (C_{12}^4 - nC_9^1 - n) p^8 q^4 + (C_{12}^5 - nC_9^2 - nC_8^1 - n) p^7 q^5;$$

$d=4$  (рисунок 2.1.8, в)

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

$$\begin{aligned}
 p_c = & p^{16} + C_{16}^1 p^{15} q + C_{16}^2 p^{14} q^2 + C_{16}^3 p^{13} q^3 + (C_{16}^4 - n) p^{12} q^4 + (C_{16}^5 - n C_{12}^1) p^{11} q^5 + \\
 & + [C_{16}^6 - n C_{12}^2 - (2n + \frac{n}{2})] p^{10} q^6 + [C_{16}^7 - n C_{12}^3 - (2n + \frac{n}{2}) C_{10}^1] p^9 q^7 + \\
 & + [C_{16}^8 - n C_{12}^4 - (2n + \frac{n}{2}) C_{10}^2] p^8 q^8 + [C_{16}^9 - n C_{12}^5 - (2n + \frac{n}{2}) C_{10}^3] p^7 q^8.
 \end{aligned}$$

Расчетные формулы для  $p_c$  при  $d=5$  и  $6$  из-за громоздкости не приводятся.

### Примеры расчета живучести сетевых структур

#### *Вычисление живучести полносвязных сетей*

Определению живучести связи (вероятности связности) между двумя конкретными узлами сети  $i$  и  $j$  посвящен целый ряд работ. Однако расчет точного ее назначения сопряжен с большими вычислительными трудностями. Представляет интерес найти простой способ определения вероятности связности сети, который позволял бы оперативно и вручную проводить на стадии проектирования оценку различных вариантов их построения.

Рассмотрим сеть той же мостиковой структуры, что и в (Рисунок 2.1.9). Для простоты будем полагать вероятности исправного функционирования всех ребер сети одинаковыми и равными  $p$ , а неисправного функционирования - равными  $q=1-p$ .

Для оценки живучести воспользуемся методом прямого перебора состояний элементов сети связи. На основании биномиального закона вероятность пребывания сети связи в состоянии, когда  $i$  любых ребер сети отказали,  $C_N^i q^i p^{N-i}$ , где  $C_N^i = N! / i! * (N - i)!$  - биномиальный коэффициент;  $N$  - число ребер сети.

Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ  
В ПРАКТИЧЕСКИХ ЗАДАЧАХ

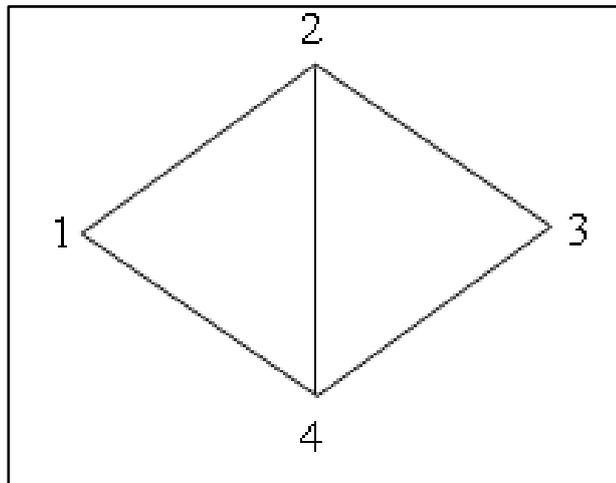


Рисунок 2.1.9 – Сеть мостиковой структуры

Например, для сети, изображенной на рис. 2.1.9, живучесть связи  $p_{13}$  зависит от следующей совокупности независимых событий:

- исправного состояния сети в целом – вероятность этого события равна  $p^3$ ;
- повреждения любого одного ребра сети – вероятность  $C_5^1 p^4 q = 5p^4 q$ ;
- одновременного повреждения любых двух ребер сети, за исключением двух случаев, когда оба ребра подходят к узлу 1 или к узлу 3 – вероятность  $(C_5^2 - 2)p^3 q^2 = 8p^3 q^2$ ;
- одновременного повреждения трех ребер сети, подходящих к узлу 2 или 4 – вероятность  $2p^2 q^3$ .

Суммируя все вероятности независимых событий, получаем искомое выражение:

$$p_{13} = 2p^5 - 5p^4 q + 2p^2 q^3,$$

Аналогично для всех остальных пар узлов сети рис. 2.1.9.

$$p_{12} = p_{14} = p_{23} = p_{34} = p^5 + 5p^4 q + 9p^3 q^2 + 5p^2 q^3 p q^4;$$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

$$p_{24} = p^5 + 5p^4q + 10p^3q^2 + 8p^2q^3 + pq^4.$$

Из анализа видно, что

$$p_{24} > p_{12} = p_{14} + p_{23} = p_{34} > p_{13}.$$

Связанной сетью являются сеть, в которой любой из узлов соединен с остальными узлами сети. Вероятность связанности сети на рисунке 2.1.9

$$p_c = p^5 + 5p^4q + 8p^3q^2.$$

Так как эта сеть допускает все одиночные повреждения ребер и восемь двойных повреждений ребер. Вероятность связности сети меньше или равна живучести связи между любой парой узлов сети, в данном случае  $p_c < p_{13}$ .

С точки зрения характеристики сети интерес представляют вероятность  $p_c$ , минимальная  $p_{мин}$  и максимальная  $p_{макс}$  живучести связи между любой парой узлов сети и соотношения между ними.

Также эти значения называют верхним пределом живучести распределенной информационной системы.

Прямой метод рекурсивного расчета полинома Тутте для полного графа был предложен Аннаном (Annan). Он заключается в следующем. Рассмотрим граф  $U_{m,r}$ , полученный из  $K_m$  добавлением новой вершины  $v$  и соединением ее с каждой вершиной  $K_m$  с помощью  $r$  кратных ребер.

$$T(U_{m,r}, x, y) = \sum_{i=1}^m \binom{m}{i} (y^{r-1} + y^{r-2} + \dots + 1)^i y^{\binom{i}{2}} T(U_{m-i,i}, x, y) + (x-1)T(U_{m-1,1}, x, y)$$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
 Применяя вышеупомянутую формулу, мы можем эффективно рассчитать функцию живучести полного графа с одинаковой вероятностью удаления ребер.

$$R(U_{m,r}, p) = \sum_{i=1}^m \binom{m}{i} (1-p^r)^i p^{r(m-i)} R(U_{m-i,i}, p),$$

где  $R(U_{0,r}, p) = 1$

Для данной сети:

$$p_c < p_{\min} = p_{13} < p_{12} = p_{14} = p_{23} = p_{34} < p_{24} = p_{\max}$$

Аналогично можно найти выражения для вероятности связности полносвязных сетей. Для сети с тремя вершинами ( $n=3$ ):

$$p_c = p^3 + 3p^2q;$$

для  $n=4$ :

$$p_c = p^6 + 6p^5q + 15p^4q^2 + (C_6^3 - 4)p^3q^3;$$

для  $n=5$ :

$$p_c = p^{10} + C_{10}^1 p^9 q + C_{10}^2 p^8 q^2 + C_{10}^3 p^7 q^3 + (C_{10}^4 - 5) p^6 q^4 + \\ + (C_{10}^5 - n C_6^1) p^5 q^5 + (C_{10}^6 n C_6^2 - n) p^4 q^6;$$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

для  $n=6$ :

$$\begin{aligned}
 p_c = & p^{15} + C_{15}^1 p^{14} q + C_{15}^2 p^{13} q^2 + C_{15}^3 p^{12} q^3 + C_{15}^4 p^{11} q^4 + (C_{15}^5 - n) p^{10} q^5 + \\
 & + (C_{15}^6 - n C_{10}^1) p^9 q^6 + (C_{15}^7 - n C_{10}^2) p^8 q^7 + (C_{15}^8 - n C_{10}^3 - n) p^7 q^8 + \\
 & + (C_{15}^9 - n C_{10}^4 - n C_7^1 - \frac{n}{2}) p^6 q^9 + (C_{15}^{10} - n C_{10}^5 - n C_7^2 - \frac{n}{2} C_6^1) p^5 q^{10}.
 \end{aligned}$$

Для  $p_c$  при  $n = 7 \dots 10$  расчетные формулы не приводятся из-за громоздкости.

Формулы можно упростить, заменив  $q$  на соответствующее значение (равное  $1-p$ ) и подставив численное значение сочетаний. На рисунке 2.1.10 приводятся примеры полиномов для полносвязных сетей с разным числом вершин.

Выполним расчет значений живучести при разных вероятностях удаления ребра полносвязных графов с различным числом вершин. Расчет выполнялся с помощью программы MathCAD 13 версии (Рисунок 2.1.11). Результаты расчетов приведены в таблице 2.1.2.

Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ  
В ПРАКТИЧЕСКИХ ЗАДАЧАХ

$n$	Полином живучести $R(K_n, p)$
2	$-p + 1$
3	$2p^3 - 3p^2 + 1$
4	$-6p^6 + 12p^5 - 3p^4 - 4p^3 + 1$
5	$24p^{10} - 60p^9 + 30p^8 + 20p^7 - 10p^6 - 5p^4 + 1$
6	$-120p^{15} + 360p^{14} - 270p^{13} - 90p^{12} + 120p^{11} + 20p^9 - 15p^8 - 6p^5 + 1$
7	$720p^{21} - 2520p^{20} + 2520p^{19} + 210p^{18} - 1260p^{17} + 210p^{16} - 70p^{15} +$ $+ 210p^{14} - 35p^{12} + 42p^{11} - 21p^{10} - 7p^6 + 1$
8	$-5040p^{28} + 20160p^{27} - 25200p^{26} + 3360p^{25} + 12810p^{24} - 5040p^{23} -$ $- 1960p^{21} + 420p^{20} + 560p^{19} - 336p^{18} + 336p^{17} - 35p^{16} - 56p^{15} +$ $+ 56p^{13} - 28p^{12} - 8p^7 + 1$
9	$40320p^{36} - 181440p^{35} + 272160p^{34} - 90720p^{33} - 128520p^{32} +$ $+ 90720p^{31} - 2520p^{30} + 15120p^{29} - 11340p^{28} - 7000p^{27} + 5544p^{26} -$ $- 4536p^{25} + 1386p^{24} + 1008p^{23} - 504p^{21} + 378p^{20} - 84p^{18} + 72p^{15} -$ $- 36p^{14} - 9p^8 + 1$
10	$-362880p^{45} + 1814400p^{44} - 3175200p^{43} + 1663200p^{42} +$ $+ 1247400p^{41} - 1489320p^{40} + 201600p^{39} - 75600p^{38} +$ $+ 189000p^{37} + 65100p^{36} - 105840p^{35} + 60480p^{34} - 27930p^{33} -$ $- 11970p^{32} + 5040p^{31} + 5040p^{30} - 5040p^{29} + 1260p^{28} + 1680p^{27} -$ $- 126p^{25} - 930p^{24} + 720p^{23} - 120p^{21} + 90p^{17} - 45p^{16} - 10p^9 + 1$

Рисунок 2.1.10 – Полиномы многосвязных сетей

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Таблица 2.1.2 – Результаты расчета значений живучести  
полносвязных графов

R/P	0.01	0.05	0.1
R2	0,99	0,95	0,9
R3	0,9997	0,9928	0,972
R4	1	0,9995	0,9958
R5	1	1	0,9995
R6	1	1	0,9999
R7	1	1	1
R8	1	1	1
R9	1	1	1
R10	1	1	1

R/P	0.3	0.5	0.7	0.9
R2	0,7	0,5	0,3	0,1
R3	0,784	0,5	0,216	0,028
R4	0,8925	0,5938	0,2186	0,0129
R5	0,9575	0,7109	0,2563	0,081
R6	0,985	0,8149	0,3169	0,0062
R7	0,9948	0,8899	0,3939	0,0055
R8	0,9982	0,9371	0,4799	0,0054
R9	0,9994	0,9647	0,5671	0,0057
R10	0,9998	0,9804	0,649	0,0064

По полученным значениям построен график (Рисунок 2.1.12).

Из таблицы 2.1.2 и рисунка 2.1.12 можно сделать вывод, что живучесть полносвязной сети очень сильно зависит от числа вершин. И чем больше у ней вершин, тем меньше влияет вероятность неисправного функционирования каналов связи (ребер графа) на живучесть.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

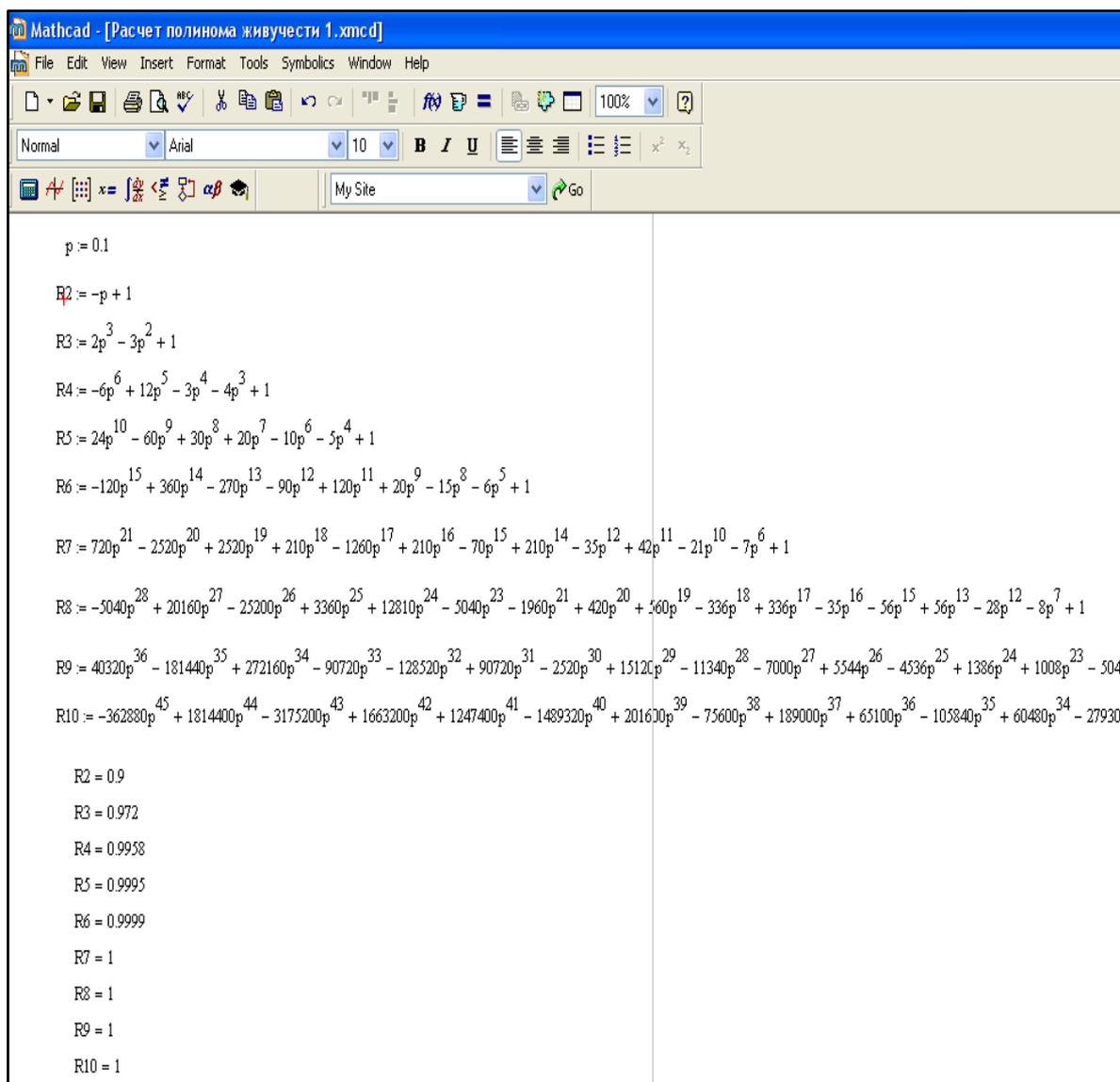


Рисунок 2.1.11 – Расчет значений живучести графов  
в программе MathCAD

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

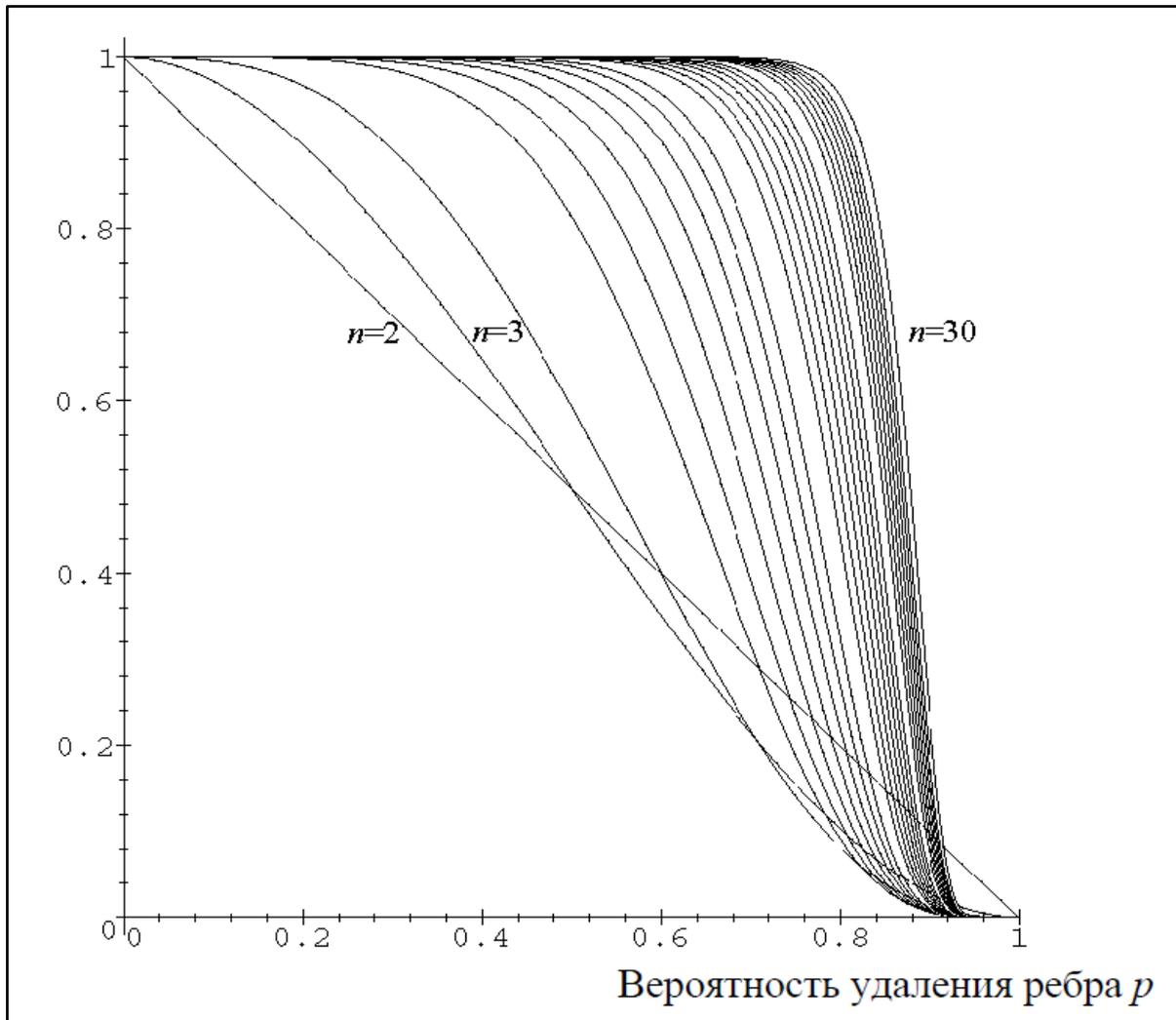


Рисунок 2.1.12 – График зависимости живучести от вероятности удаления ребра

### *Вычисление живучести кольцевых сетей*

Вероятность связности для кольцевых сетей связи, т.е. сетей, у которых степень для каждой вершины равна 2 (степенью вершины  $d$  называются число граней графа сети, инцидентных данной вершине):

$$p_c = p^n + np^{n-1}q.$$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Проведем расчет живучести для кольцевых сетей с разным количеством вершин при разных значениях  $p$  (Рисунок 2.1.13). Результаты расчетов представлены в таблице 2.1.3.

Таблица 2.1.3 – Результаты расчетов значений живучести кольцевых сетей

R/P	0,3	0,4	0,5
N5	0,031	0,087	0,188
N6	0,011	0,041	0,109
N7	$3.791 \cdot 10^{-3}$	0.019	0,063
N8	$1.29 \cdot 10^{-3}$	$8.52 \cdot 10^{-3}$	0,035
N9	$4.33 \cdot 10^{-4}$	$3.801 \cdot 10^{-3}$	0,02
N10	$1.437 \cdot 10^{-4}$	$1.678 \cdot 10^{-3}$	0,011

R/P	0,6	0,7	0,8	0,9
N5	0,337	0,528	0,737	0,919
N6	0,233	0,42	0,655	0,886
N7	0,159	0,329	0,577	0,85
N8	0,106	0,255	0,503	0,813
N9	0,071	0,196	0,436	0,775
N10	0,046	0,149	0,376	0,736

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

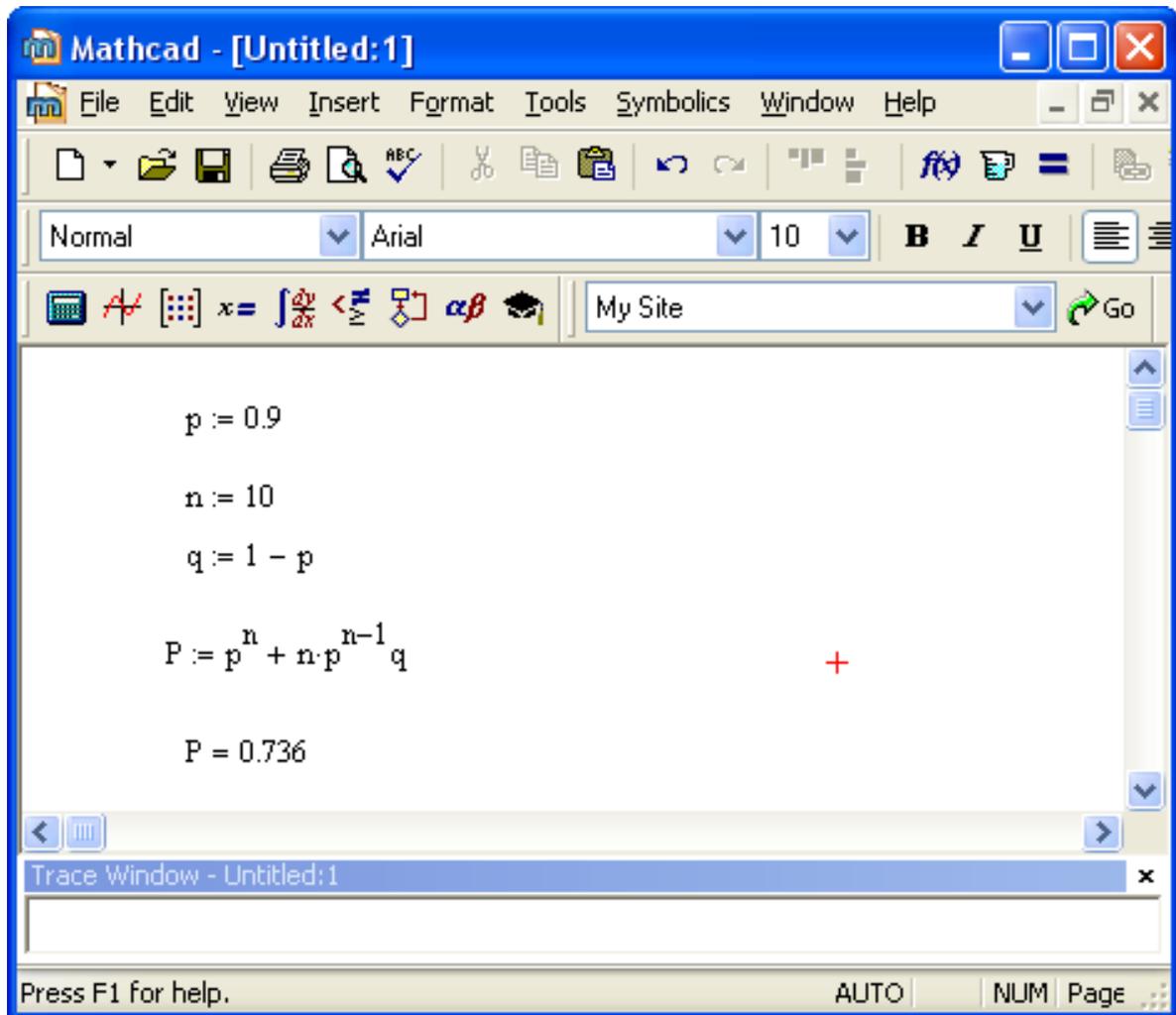


Рисунок 2.1.13 – Расчет значений живучести кольцевых сетей  
в программе MathCAD

На рисунке 2.1.14 определена зависимость  $p_c$  от  $p$  для кольцевых сетей при различных  $n$ . Из ее анализа видно, что вероятность связности кольцевых сетей падает с увеличением числа узлов сети при одних и тех же значениях  $p$ .

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

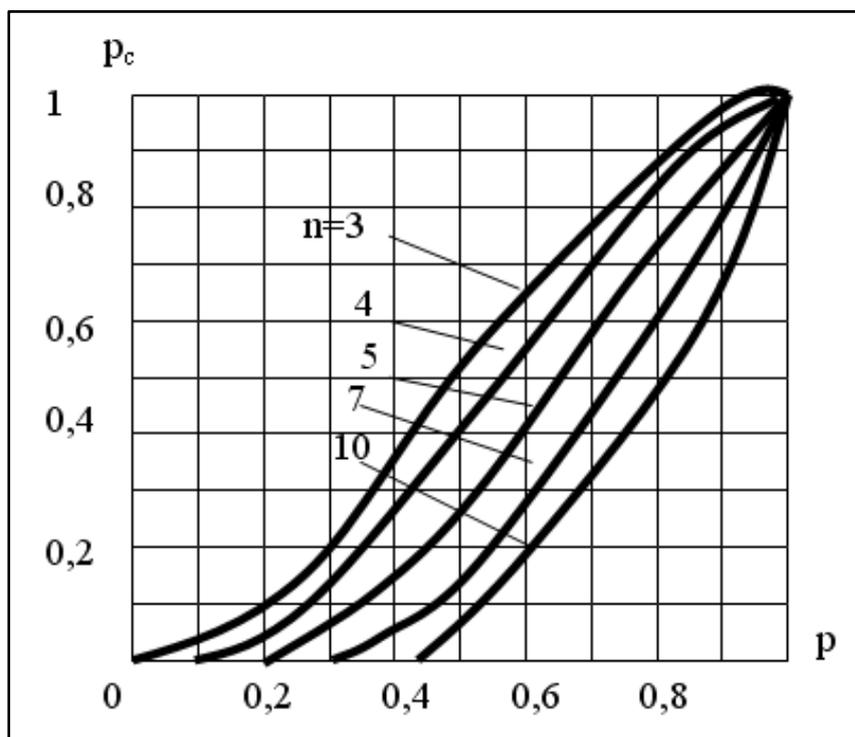


Рисунок 2.1.14 – Зависимость  $p_c$  от  $p$  для кольцевых сетей при различных  $n$

Из таблицы 2.1.3 и рисунка 2.1.14 можно сделать вывод, что в отличие от полносвязных сетей, живучесть кольцевых сетей в значительной степени зависит от вероятности исправного функционирования всех ребер сети.

### Вычисление живучести сетей с разной степенью вершин

Вычислим значения живучести для сетевых структур по приведенным формулам, с различными значениями вероятности исправного функционирования ребер сети при значениях  $d=2$ ,  $d=3$  и  $d=4$  соответственно.

Расчет проводился с помощью программы Mathcad (Рисунки 2.1.15 и 2.1.16). Результаты расчетов приведены в таблицах 2.1.4 и 2.1.5.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Таблица 2.1.4 – Результаты расчетов значений живучести сети с  $n=6$  при различных  $d$

R/P	0,1	0,2	0,3	0,4
R2	$5.5 \cdot 10^{-5}$	$1.6 \cdot 10^{-3}$	0,011	0,041
R3	$6.89 \cdot 10^{-4}$	0,015	0,08	0,221
R4	$2.396 \cdot 10^{-3}$	0,044	0,183	0,411

R/P	0,5	0,6	0,7	0,8	0,9
R2	0,109	0,223	0,42	0,655	0,886
R3	0,428	0,653	0,838	0,95	0,994
R4	0,656	0,844	0,949	0,99	0,999

Таблица 2.1.5 – Результаты расчетов значений живучести сети с  $n=8$  при различных  $d$

R/P	0,1	0,2	0,3	0,4
R2	$7.3 \cdot 10^{-7}$	$8.448 \cdot 10^{-5}$	$1.29 \cdot 10^{-3}$	$8.52 \cdot 10^{-3}$
R3	$2.839 \cdot 10^{-5}$	$2.307 \cdot 10^{-3}$	0,024	0,105
R4	$1.558 \cdot 10^{-4}$	0,01	0,086	0,288

R/P	0,5	0,6	0,7	0,8	0,9
R2	0,035	0,106	0,255	0,503	0,813
R3	0,288	0,522	0,764	0,926	0,991
R4	0,567	0,805	0,938	0,988	0,999



## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

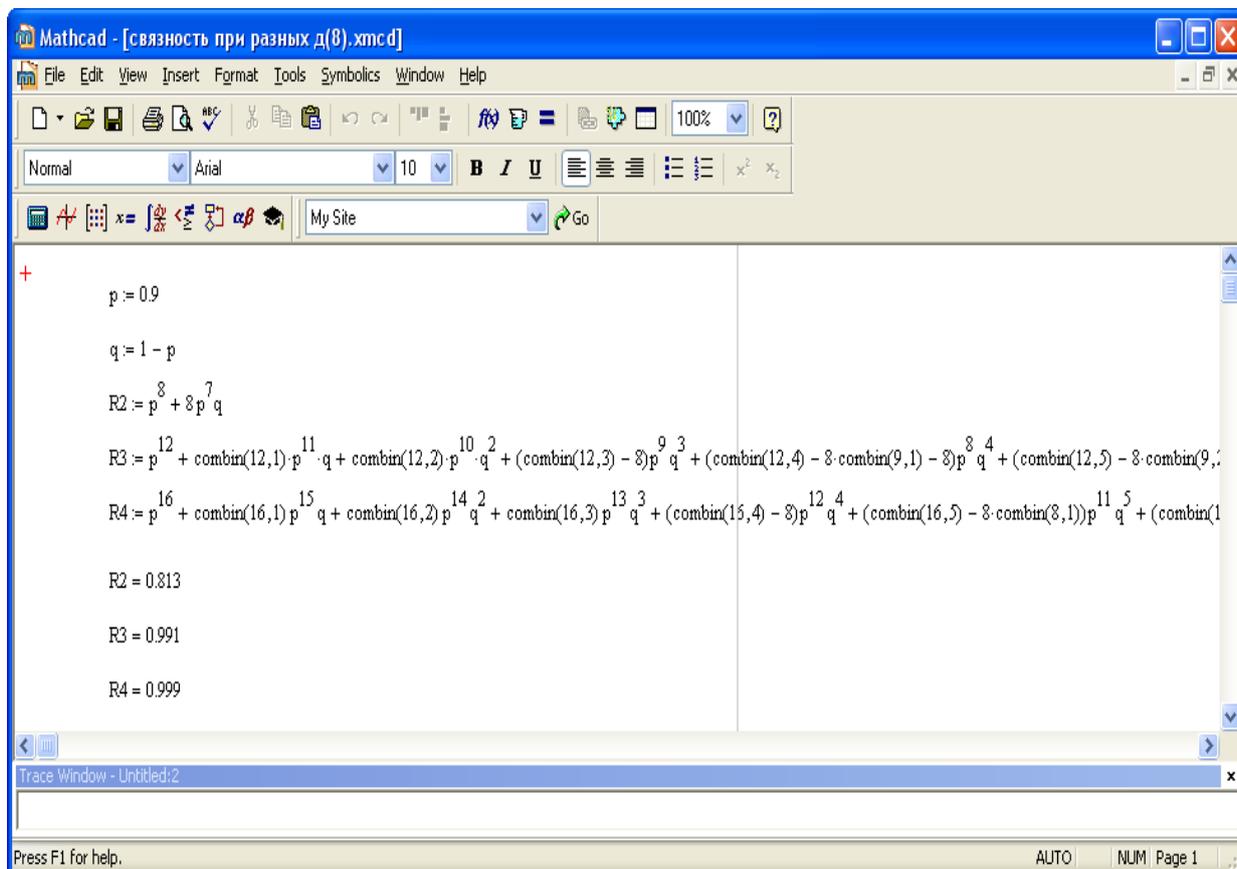


Рисунок 2.1.16 – Расчет значений живучести сети с  $n=8$   
при различных  $d$

На рисунках 2.1.17 и 2.1.18 представлены зависимости вероятности связности сети с  $n=6, 8$  соответственно при различных  $d$  (сплошные линии). Из рисунков видно, что увеличение вероятности связности сети с увеличением  $d$  при неизменном  $p$  объясняется тем, что с увеличением  $d$  возрастает разветвленность сети связи.

Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ  
В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

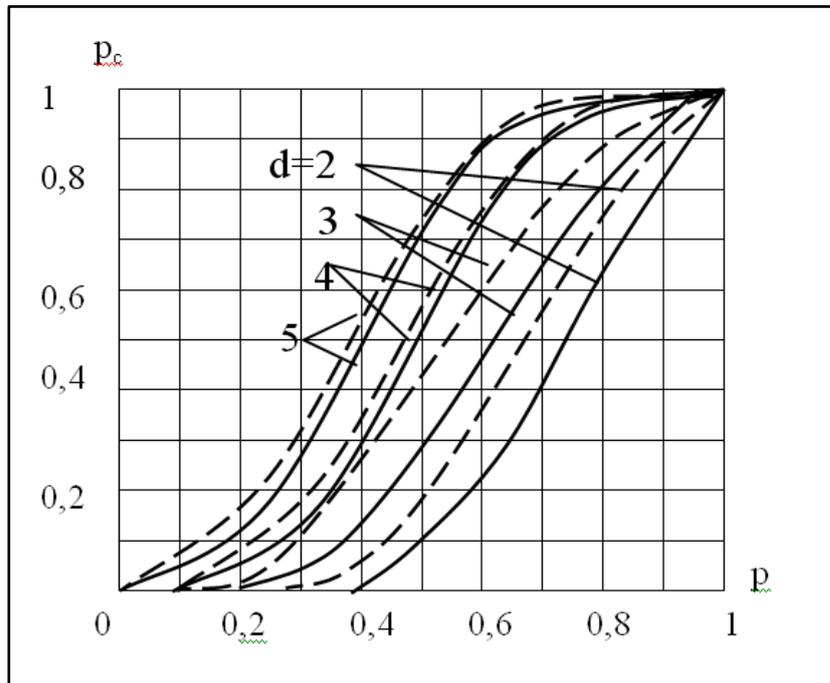


Рисунок 2.1.17 – Зависимости вероятности связности сети с  $n=6$  при различных  $d$

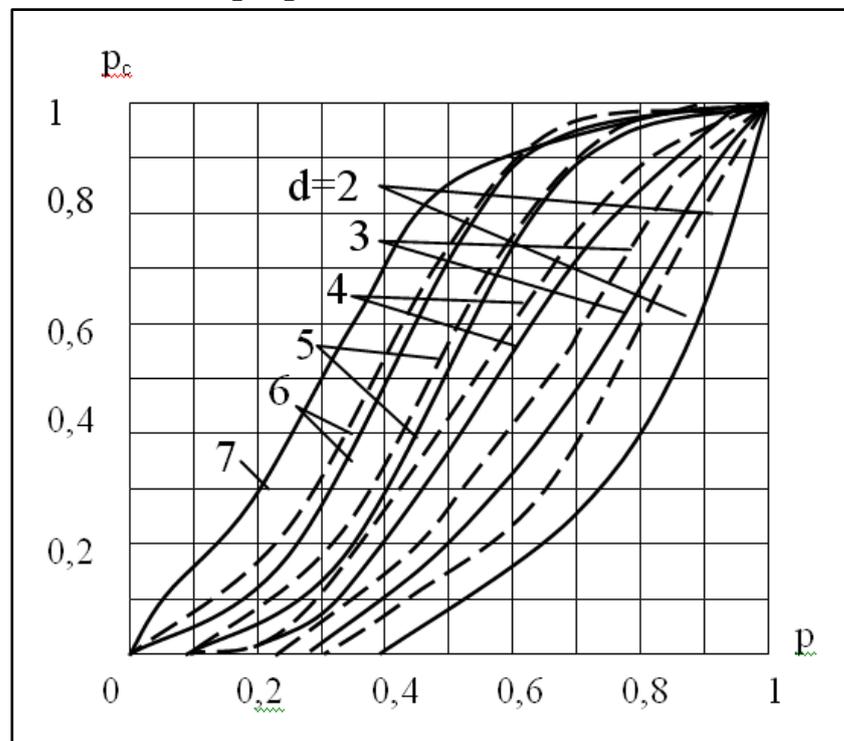


Рисунок 2.1.18 – Зависимости вероятности связности сети с  $n=8$  при различных  $d$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Вычисление живучести на основе определения вероятности связности позволило сделать вывод, что живучесть полносвязной сети очень сильно зависит от числа вершин. И чем больше у ней вершин, тем меньше влияет вероятность неисправного функционирования каналов связи (ребер графа) на живучесть. На практике довольно редко встречаются полносвязные сети. Обычно бывают сети с небольшими степенями вершин. Для таких сетей увеличение живучести с увеличением степени вершин при неизменном значении вероятности исправного функционирования каналов связи, объясняется тем, что с увеличением степени вершины возрастает разветвленность сети связи. На практике же такие системы являются очень дорогими и не оправдывают вложенных в них средств. Но для организаций, где циркулирует критически важная информация, подобные решения могут найти применение.

Сетевая информационная система представляет собой распределенную структуру, схема функционирования которой задается с помощью графа, определяющего ее физическую структуру. Основной характеристикой графа является его связность, то есть количество каналов связи, соединяющих узлы данной сетевой информационной системы. Наибольшей живучестью обладают полносвязные сети, представляющие собой кольца с различным числом дополнительных каналов связи. Рассмотренная модель живучести информационной системы с радиальной топологией достаточно подробно описывает процесс расчета, однако некоторые ее значения могут быть получены только экспериментальным путем и дополнительными расчетами. В частности, есть необходимость использовать методику расчета приоритетов узлов сети.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### **Практическая оценка живучести распределенной информационной системы**

#### *Цели:*

- разработать сеть корпоративной распределенной информационной системы;
- получить экспериментальным путем значения интенсивности отказов узлов;
- осуществить расчет приоритетов узлов;
- произвести окончательный расчет показателей живучести распределенной информационной системы;
- проанализировать результаты и сделать вывод об адекватности рассмотренных моделей живучести.

#### ***Разработка сети корпоративной распределенной информационной системы***

Для расчета была выбрана сеть, в которой центральное коммуникационное оборудование имеет коэффициент связности 2-3, то есть имеет дополнительные каналы связи (Рисунок 2.1.19).

Предлагаемая сеть состоит из 6 сегментов сети радиальной топологии и 1 сегмента с серверами. Структура сети представляет собой два кольца, объединенных посредством маршрутизаторов. Верхнее кольцо имеет коэффициент связности узлов равного 2, нижнее кольцо – 3.

Расчет живучести подобных сетей выполняется по частям, то есть для отдельных сегментов сети, которые впоследствии представляются как один узел. И так до тех пор, пока не будет получено конечное значение. При этом могут использоваться несколько моделей живучести.

В нашем случае для расчета нам понадобится две модели живучести:

- модель живучести информационной сети с радиальной топологией;

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

- модель расчета живучести кольцевых сетей с различными значениями степеней вершин.

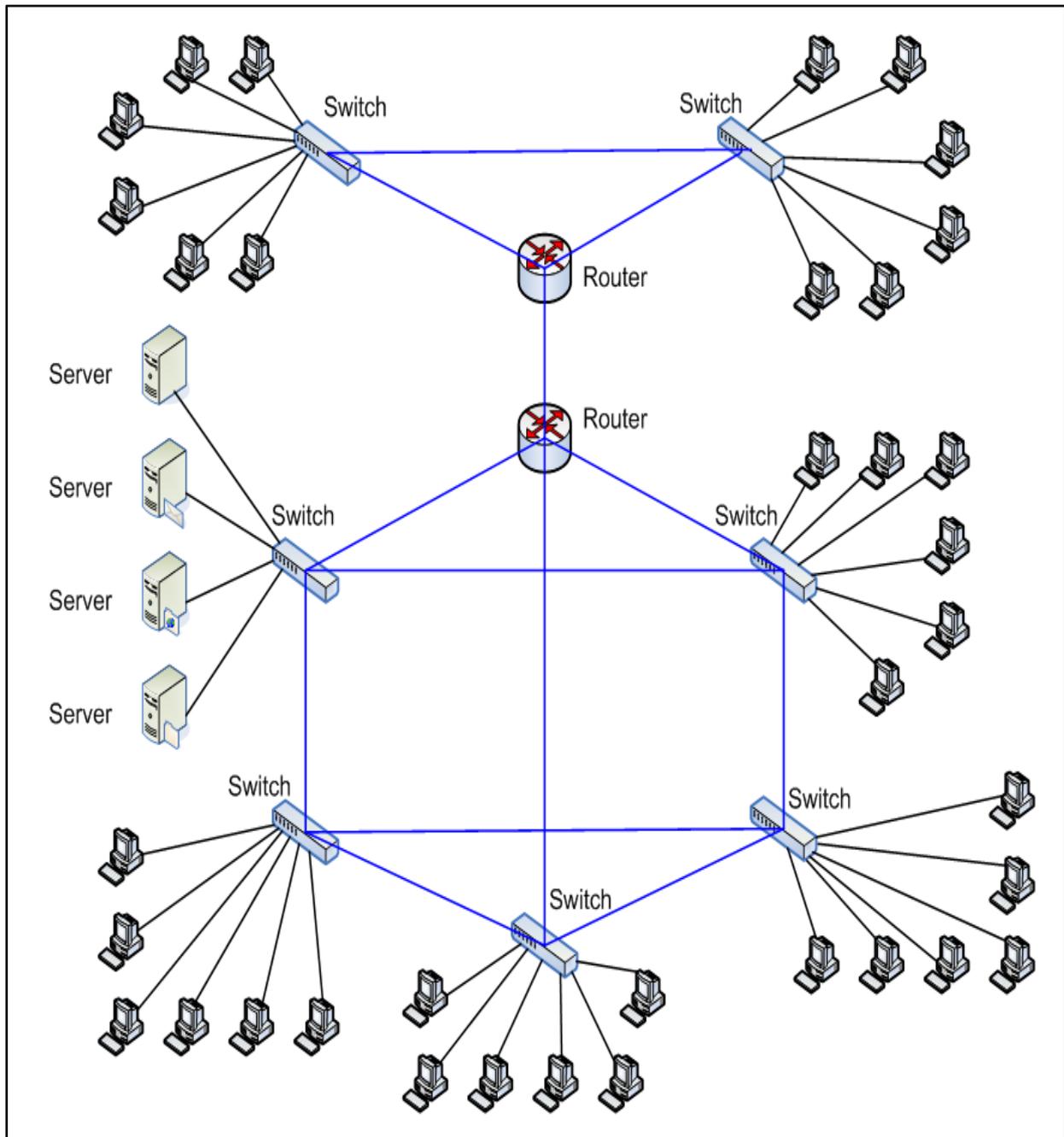


Рисунок 2.1.19– Схема сети

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Пусть в данной сети происходят наиболее типичные информационные процессы, характерные для любой корпоративной распределенной информационной системы. Перечислим их:

- скачивание файлов с FTP-сервера;
- скачивание почтовых сообщений с вложениями;
- просмотр Web-страниц;
- общение посредством мгновенных сообщений.

### *Описание экспериментальной установки*

Для создания разработанной сети были использованы средства виртуализации. Поскольку проведение натуральных экспериментов требует значительных финансовых и технических ресурсов, применение моделирования в данном случае является оптимальным решением.

Экспериментальная установка собрана на основе продукта GNS3: Graphical Network Simulator 3 – это графический эмулятор сети, который позволяет проектировать, эмулировать и создавать виртуальные сети. Данные сети могут иметь сложную топологию и управляемые Cisco устройства.

Вся эмуляция строится на основе специальных операционных систем устройств – Cisco IOS. Cisco IOS (от англ. Internetwork Operating System - Межсетевая Операционная система) – программное обеспечение, используемое в маршрутизаторах Cisco и некоторых сетевых коммутаторах. Cisco IOS – многозадачная операционная система, выполняющая функции сетевой организации, маршрутизации, коммутации и передачи данных.

Роль серверов и конечных узлов выполняют виртуальные машины на основе системы управления виртуальными машинами VirtualBox.

Конфигурирование GNS3: Graphical Network Simulator сводится к указанию двух путей к файлам: путь и выбор образа, ис-

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

пользуемого для сетевых устройств (это образ специальной операционной системы маршрутизаторов – Cisco IOS) и путь к программе – эмулятору Dynamips.

Следующим шагом будет расстановка в рабочей области программы элементов: маршрутизаторов, коммутаторов, «облаков» сетевых интерфейсов и каналов связи между ними. Все это осуществляется при помощи Drag&Drop. Каждое «облако» это своеобразный сетевой мост между интерфейсом сетевого устройства и реальным сетевым подключением ПК.

Маршрутизатор, который подключается к серверам, настраивается таким образом, что внешний интерфейс подключается к сетевой карте ПК, а внутренний к «виртуальному» подключению, созданному VirtualBox.

Таким образом GNS3 позволяет создавать сложные по топологии сети и объединять реальную сеть с виртуальным оборудованием, которое полностью как программно, так и аппаратно виртуализируется. Стоит также отметить, что необходимо использовать максимально эффективно имеющиеся вычислительные ресурсы. Поэтому было принято решение о установке минимальной операционной системы Ubuntu Server на ПК, которые виртуализируют конечные рабочие станции. Данная ОС не имеет графического интерфейса и управляется только из режима командной строки, а так как VirtualBox работает в графическом режиме при установке данного программного продукта требуется установить множества зависимых пакетов.

Поскольку сеть представляет собой два сегмента (кольца), было принято решение физически разделить верхний и нижний сегменты путем моделирования на двух разных ПК. Основной проблемой является моделирование нижнего кольца. В существующих корпоративных сетях передачи данных подобных связей стараются избегать, поскольку при неправильной настройке нарушается работа информационной системы из-за заикливания пакетов. Но теоретически живучесть подобных решений намного выше, чем у классических сетевых топологий.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Настройка сетевых устройств, входящих в кольцо сложная задача, требующая знания и умения настраивать протокол STP. Основной задачей STP является приведение сети Ethernet с множественными связями к древовидной топологии, исключающей циклы пакетов.

Для моделирования кольца следует разделить его на несколько возможных маршрутов и сравнить характеристики, полученные с кольцом и без него. Поэтому был сделан эксперимент, моделирующий кольцо из четырех коммутаторов. В данном эксперименте осуществлялось обращение к серверам (IM, Web, FTP, SMTP/POP3). Его целью было получение информации о времени обращения, пропускной способности, количеству отказов в сети. После проведения такого опыта было построено несколько маршрутов по кольцу и собраны те же самые данные. В результате сравнения были выявлены несущественные различия, а значит для проведения полноценных экспериментов можно воспользоваться маршрутами, являющимися частью моделируемой сети. Каждый информационный процесс может описываться своим маршрутом.

### *Расчет приоритетов узлов корпоративных сетей передачи данных*

Чтобы выполнить расчет приоритетов узлов сети, изображенной на рисунке 2.19, воспользуемся описанной методикой. Пронумеруем все функциональные элементы сети (Рисунок 2.1.20).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

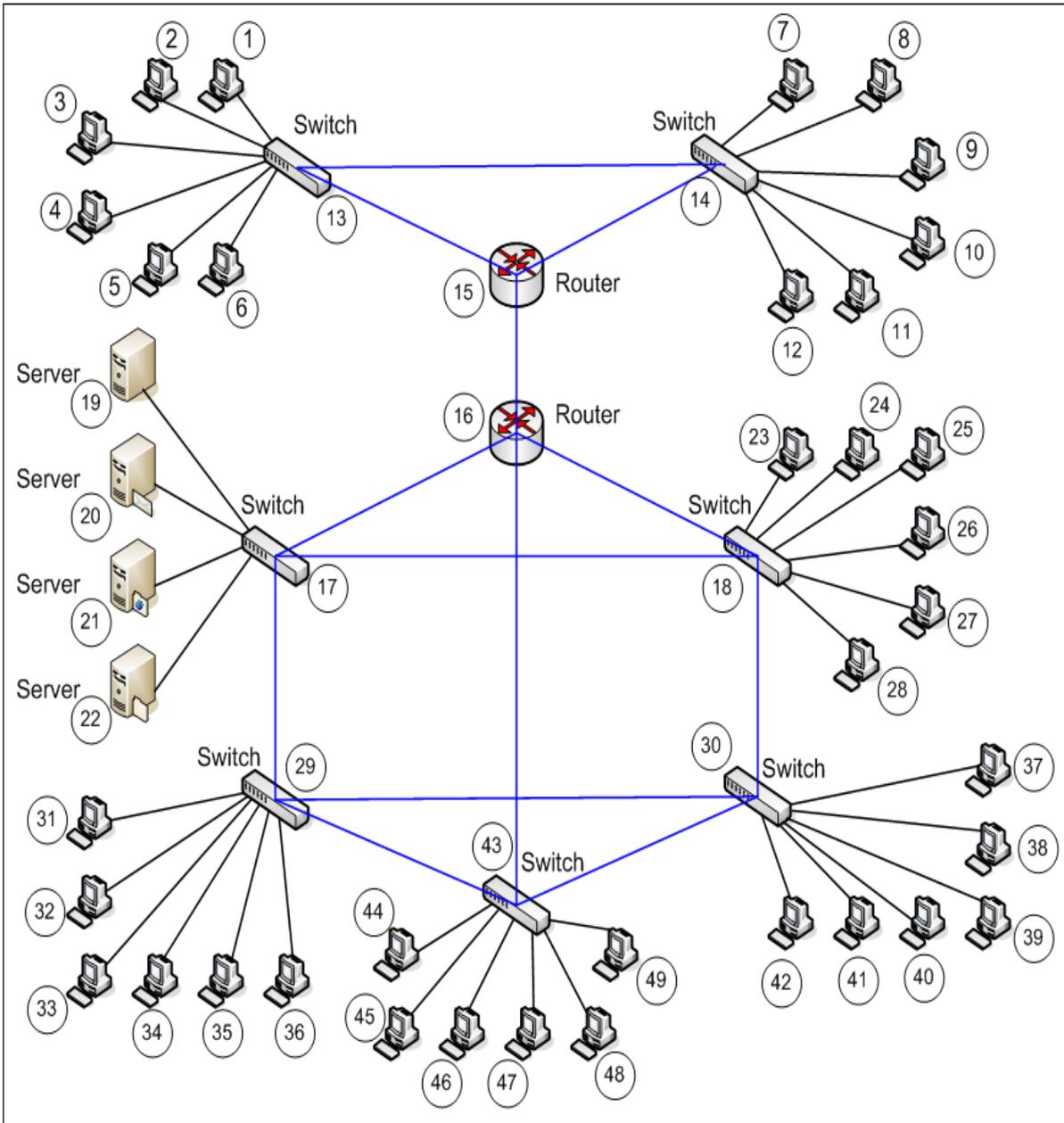


Рисунок 2.1.20 – Нумерация элементов сети

Опишем входные данные согласно предлагаемой методике.

$$H_1 = 17; H_2 = 9; H_3 = 10; H_4 = 12; H_5 = 15; H_6 = 15$$

$$A_1 = \{S_{39}, S_{40}, S_{41}, S_{42}, S_{48}, S_{49}\}$$

$$B_1 = \{S_{22}\}$$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

$$\begin{aligned} A_2 &= \{S_7, S_8, S_9, S_{10}\} & B_2 &= \{S_{20}\} \\ A_3 &= \{S_4, S_5, S_6, S_{23}, S_{24}\} & B_3 &= \{S_{21}\} \\ A_4 &= \{S_{44}, S_{45}, S_{46}, S_{47}\} & B_4 &= \{S_{21}\} \\ A_5 &= \{S_1, S_2, S_3, S_{11}, S_{12}\} & B_5 &= \{S_{19}\} \\ A_6 &= \{S_{19}\} & B_6 &= \{S_{31}, S_{32}, \\ & & & S_{33}, S_{34}, S_{35}, S_{36}\} \end{aligned}$$

Где индексы соответствуют следующим процессам, происходящим в предлагаемой информационной системе:

- Скачивание файлов с FTP-сервера;
- Просмотр почты;
- Просмотр web-страниц;
- Сохранение web-страниц;
- Отправка IM-сообщений;
- Получение IM-сообщений.

В качестве исходных данных рассматриваются множество всех информационных процессов для заданной сети с параметрами:  $H_1=17$ ,  $H_2=9$ ,  $H_3=10$ ;  $A_1=\{S_{39}, S_{40}, S_{41}, S_{42}, S_{48}, S_{49}\}$ ,  $A_2=\{S_7, S_8, S_9, S_{10}\}$ ,  $A_3=\{S_4, S_5, S_6, S_{23}, S_{24}\}$ ;  $B_1=\{S_{22}\}$ ,  $B_2=\{S_{20}\}$ ,  $B_3=\{S_{21}\}$ .

Три подграфа для каждого из информационных процессов ( $P_1, P_2, P_3$ ). Здесь  $n_1(A_1)=6$ ,  $n_1(B_1)=1$ ,  $n_2(A_2)=4$ ,  $n_2(B_2)=1$ ,  $n_3(A_3)=5$ ,  $n_3(B_3)=1$ .

Найдем для данных информационных процессов все пути взаимодействия  $W_i = \{w_1^i, \dots, w_p^i\}$  в виде последовательностей функциональных элементов – узлов графа  $P_i$ , включая соответствующие узлы-отправители и узлы-получатели. Всего таких путей  $l(W_i)$ . Для каждого из процессов ( $P_1, P_2, P_3$ ) количество путей соответственно –

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
 $l(W_1)=18$ ,  $l(W_2)=32$ ,  $l(W_3)=30$ . Количество путей, проходящих через каждый элемент, представлено в таблице 2.1.6.

Таблица 2.1.6 – Количество путей, проходящих через функциональные элементы сети

$P \setminus S_r$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$
$P_1$	0	0	0	0	0	0	0	0	0	0	0	0
$P_2$	0	0	0	0	0	0	8	8	8	8	0	0
$P_3$	0	0	0	8	8	8	0	0	0	0	0	0

$P \setminus S_r$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$	$S_{17}$	$S_{18}$	$S_{19}$	$S_{20}$	$S_{21}$	$S_{22}$	$S_{23}$	$S_{24}$	$S_{25}$
$P_1$	0	0	0	0	18	6	0	0	0	18	0	0	0
$P_2$	16	32	32	32	32	24	0	32	0	0	0	0	0
$P_3$	24	12	24	24	30	24	0	0	30	0	3	3	0

$P \setminus S_r$	$S_{26}$	$S_{27}$	$S_{28}$	$S_{29}$	$S_{30}$	$S_{31}$	$S_{32}$	$S_{33}$	$S_{34}$	$S_{35}$	$S_{36}$	$S_{37}$	$S_{38}$	$S_{39}$	$S_{40}$	$S_{41}$
$P_1$	0	0	0	12	16	0	0	0	0	0	0	0	0	3	0	3
$P_2$	0	0	0	16	16	0	0	0	0	0	0	0	0	0	0	0
$P_3$	0	0	0	16	16	0	0	0	0	0	0	0	0	0	0	0

$P \setminus S_r$	$S_{42}$	$S_{43}$	$S_{44}$	$S_{45}$	$S_{46}$	$S_{47}$	$S_{48}$	$S_{49}$
$P_1$	3	9	0	0	0	0	3	3
$P_2$	0	8	0	0	0	0	0	0
$P_3$	0	12	0	0	0	0	0	0

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Пусть пути  $W_i$  проходят через  $S_r$  функциональный элемент распределенной информационной системы  $k_r(W_i)$  раз. Следовательно, параметр  $\gamma_r^i(S_r)$  можно трактовать, как «относительное участие»  $S_r$  в реализации  $p_i$ .

Относительные частоты появления функциональных элементов сети на путях  $W_1$ ,  $W_2$  и  $W_3$  представлены в таблице 2.1.7.

Численное значение приоритета функционального элемента сети  $S_r$  для  $i$ -го информационного процесса определяется в соответствии с выражением:

$$R_r^i = \gamma_r^i \times H_i.$$

где  $\gamma_r^i$  - коэффициент «участия» функционального элемента сети  $S_r$  в реализации  $p_i$ , определяемый числом появлений  $S_r$  на множестве трасс  $W_i$ .

С использованием Евклидовой метрики суммарный приоритет  $S_r$  (по всем информационным процессам) можно определить с применением весовой нормы:

$$R_r = \sqrt{(R_r^1)^2 + (R_r^2)^2 + \dots + (R_r^t)^2} = \sqrt{\sum_{h=1}^t (R_r^h)^2}$$

где  $R_r$  — приоритет для элемента  $S_r$ ,

$h$  — номер информационного процесса.

Здесь  $R_r = 0$  соответствует элементу, не принадлежащему множеству  $P = \{p_1, \dots, p_t\}$ .

Рассчитанные приоритеты функциональных элементов сети по каждому информационному процессу и их суммарные значения представлены в таблице 2.1.8.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.7 – Относительные частоты появления  
функциональных элементов сети на путях  $W_1$ ,  $W_2$  и  $W_3$

$P \setminus S_r$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$
$P_1$	0	0	0	0	0	0	0	0	0	0	0	0
$P_2$	0	0	0	0	0	0	0,3	0,3	0,3	0,3	0	0
$P_3$	0	0	0	0,3	0,3	0,3	0	0	0	0	0	0

$P \setminus S_r$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$	$S_{17}$	$S_{18}$	$S_{19}$	$S_{20}$	$S_{21}$	$S_{22}$	$S_{23}$	$S_{24}$	$S_{25}$
$P_1$	0	0	0	0	1	0,3	0	0	0	1	0	0	0
$P_2$	0,5	1	1	1	1	0,8	0	1	0	0	0	0	0
$P_3$	0,8	0,4	0,8	0,8	1	0,8	0	0	1	0	0,1	0,1	0

$P \setminus S_r$	$S_{26}$	$S_{27}$	$S_{28}$	$S_{29}$	$S_{30}$	$S_{31}$	$S_{32}$	$S_{33}$	$S_{34}$	$S_{35}$	$S_{36}$	$S_{37}$
$P_1$	0	0	0	0,7	0,9	0	0	0	0	0	0	0
$P_2$	0	0	0	0,5	0,5	0	0	0	0	0	0	0
$P_3$	0	0	0	0,5	0,5	0	0	0	0	0	0	0

$P \setminus S_r$	$S_{38}$	$S_{39}$	$S_{40}$	$S_{41}$	$S_{42}$	$S_{43}$	$S_{44}$	$S_{45}$	$S_{46}$	$S_{47}$	$S_{48}$	$S_{49}$
$P_1$	0	0,2	0	0,2	0,2	0,5	0	0	0	0	0,2	0,2
$P_2$	0	0	0	0	0	0,3	0	0	0	0	0	0
$P_3$	0	0	0	0	0	0,4	0	0	0	0	0	0

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.8 – Рассчитанные приоритеты функциональных элементов сети по каждому информационному процессу и их суммарные значения

$P \setminus S_r$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$
$P_1$	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_2$	0	0	0	0	0	0	2,7	2,7	2,7	2,7	0	0	4,5
$P_3$	0	0	0	3	3	3	0	0	0	0	0	0	8
<b>P</b>	0	0	0	3	3	3	2,7	2,7	2,7	2,7	0	0	9,2

$P \setminus S_r$	$S_{14}$	$S_{15}$	$S_{16}$	$S_{17}$	$S_{18}$	$S_{19}$	$S_{20}$	$S_{21}$	$S_{22}$	$S_{23}$	$S_{24}$	$S_{25}$
$P_1$	0	0	0	17	5,1	0	0	0	17	0	0	0
$P_2$	9	9	9	9	7,2	0	9	0	0	0	0	0
$P_3$	4	8	8	10	8	0	0	10	0	1	1	0
<b>P</b>	9,8	12	12	21,7	11,9	0	9	10	17	1	1	0

$P \setminus S_r$	$S_{26}$	$S_{27}$	$S_{28}$	$S_{29}$	$S_{30}$	$S_{31}$	$S_{32}$	$S_{33}$	$S_{34}$	$S_{35}$	$S_{36}$	$S_{37}$
$P_1$	0	0	0	11,9	15,3	0	0	0	0	0	0	0
$P_2$	0	0	0	4,5	4,5	0	0	0	0	0	0	0
$P_3$	0	0	0	5	5	0	0	0	0	0	0	0
<b>P</b>	0	0	0	13,7	16,7	0	0	0	0	0	0	0

$P \setminus S_r$	$S_{38}$	$S_{39}$	$S_{40}$	$S_{41}$	$S_{42}$	$S_{43}$	$S_{44}$	$S_{45}$	$S_{46}$	$S_{47}$	$S_{48}$	$S_{49}$
$P_1$	0	3,4	0	3,4	3,4	8,5	0	0	0	0	3,4	3,4
$P_2$	0	0	0	0	0	2,7	0	0	0	0	0	0
$P_3$	0	0	0	0	0	4	0	0	0	0	0	0
<b>P</b>	0	3,4	0	3,4	3,4	9,8	0	0	0	0	3,4	3,4

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Таким образом, самыми критичными функциональные элементы сети являются устройства  $S_{17}$  и  $S_{30}$ .

Далее в качестве исходных данных рассматриваются следующие три информационные процессы для заданной сети с параметрами:  $H_4=12$ ,  $H_5=15$ ,  $H_6=15$ ;  $A_4=\{S_{44}, S_{45}, S_{46}, S_{47}\}$ ,  $A_5=\{S_1, S_2, S_3, S_{11}, S_{12}\}$ ,  $A_6=\{S_{19}\}$ ;  $B_4=\{S_{21}\}$ ,  $B_5=\{S_{19}\}$ ,  $B_6=\{S_{31}, S_{32}, S_{33}, S_{34}, S_{35}, S_{36}\}$ .

Три подграфа для каждого из процессов ( $P_4$ ,  $P_5$ ,  $P_6$ ). Здесь  $n_4(A_4)=4$ ,  $n_4(B_4)=1$ ,  $n_5(A_5)=5$ ,  $n_5(B_5)=1$ ,  $n_6(A_6)=1$ ,  $n_6(B_6)=6$ .

Найдем для данных ИП все пути взаимодействия  $W_i = (w_{i1}, w_{i2}, \dots, w_{il})$  в виде последовательностей функциональных элементов – узлов графа  $P_i$ , включая соответствующие узлы-отправители и узлы-получатели. Всего таких путей  $l(W_i)$ .

Для каждого из процессов ( $P_4$ ,  $P_5$ ,  $P_6$ ), количество таких путей соответственно –  $l(W_4)=12$ ,  $l(W_5)=40$ ,  $l(W_6)=18$ . Количество путей, проходящих через каждый элемент, представлено в таблице 2.1.9.

Относительные частоты появления функциональных сетевых элементов на путях  $W_1$ ,  $W_2$  и  $W_3$  представлены в таблице 2.1.10.

Рассчитанные приоритеты функциональных элементов сети по каждому информационному процессу и их суммарные значения представлены в таблице 2.1.11.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.9 - Количество путей, проходящих через функциональные элементы сети

$P \setminus S_r$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$
$P_4$	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_5$	8	8	8	0	0	0	0	0	0	0	8	8	32
$P_6$	0	0	0	0	0	0	0	0	0	0	0	0	0

$P \setminus S_r$	$S_{14}$	$S_{15}$	$S_{16}$	$S_{17}$	$S_{18}$	$S_{19}$	$S_{20}$	$S_{21}$	$S_{22}$	$S_{23}$	$S_{24}$	$S_{25}$
$P_4$	0	0	0	12	4	0	0	12	0	0	0	0
$P_5$	28	40	40	40	30	40	0	0	0	0	0	0
$P_6$	0	0	0	18	12	18	0	0	0	0	0	0

$P \setminus S_r$	$S_{26}$	$S_{27}$	$S_{28}$	$S_{29}$	$S_{30}$	$S_{31}$	$S_{32}$	$S_{33}$	$S_{34}$	$S_{35}$	$S_{36}$	$S_{37}$	$S_{38}$
$P_4$	0	0	0	8	8	0	0	0	0	0	0	0	0
$P_5$	0	0	0	20	20	0	0	0	0	0	0	0	0
$P_6$	0	0	0	18	12	3	3	3	3	3	3	0	0

$P \setminus S_r$	$S_{39}$	$S_{40}$	$S_{41}$	$S_{42}$	$S_{43}$	$S_{44}$	$S_{45}$	$S_{46}$	$S_{47}$	$S_{48}$	$S_{49}$
$P_4$	0	0	0	0	12	3	3	3	3	0	0
$P_5$	0	0	0	0	10	0	0	0	0	0	0
$P_6$	0	0	0	0	6	0	0	0	0	0	0

Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ  
В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.10 – Относительные частоты появления  
функциональных элементов сети на путях  $W_1$ ,  $W_2$  и  $W_3$

$P \setminus S_r$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$
$P_4$	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_5$	0,2	0,2	0,2	0	0	0	0	0	0	0	0,2	0,2	0,8
$P_6$	0	0	0	0	0	0	0	0	0	0	0	0	0

$P \setminus S_r$	$S_{14}$	$S_{15}$	$S_{16}$	$S_{17}$	$S_{18}$	$S_{19}$	$S_{20}$	$S_{21}$	$S_{22}$	$S_{23}$	$S_{24}$	$S_{25}$
$P_4$	0	0	0	1	0,3	0	0	1	0	0	0	0
$P_5$	0,7	1	1	1	0,8	1	0	0	0	0	0	0
$P_6$	0	0	0	1	0,7	1	0	0	0	0	0	0

$P \setminus S_r$	$S_{26}$	$S_{27}$	$S_{28}$	$S_{29}$	$S_{30}$	$S_{31}$	$S_{32}$	$S_{33}$	$S_{34}$	$S_{35}$	$S_{36}$	$S_{37}$
$P_4$	0	0	0	0,7	0,7	0	0	0	0	0	0	0
$P_5$	0	0	0	0,5	0,5	0	0	0	0	0	0	0
$P_6$	0	0	0	1	0,7	0,2	0,2	0,2	0,2	0,2	0,2	0

$P \setminus S_r$	$S_{38}$	$S_{39}$	$S_{40}$	$S_{41}$	$S_{42}$	$S_{43}$	$S_{44}$	$S_{45}$	$S_{46}$	$S_{47}$	$S_{48}$	$S_{49}$
$P_4$	0	0	0	0	0	1	0,3	0,3	0,3	0,3	0	0
$P_5$	0	0	0	0	0	0,3	0	0	0	0	0	0
$P_6$	0	0	0	0	0	0,3	0	0	0	0	0	0

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.11 – Рассчитанные приоритеты  
функциональных элементов сети по каждому информационному  
процессу и их суммарные значения

$P \backslash S_r$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$
$P_5$	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_5$	3	3	3	0	0	0	0	0	0	0	3	3	12
$P_6$	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>P</b>	3	3	3	0	0	0	0	0	0	0	3	3	12

$P \backslash S_r$	$S_{14}$	$S_{15}$	$S_{16}$	$S_{17}$	$S_{18}$	$S_{19}$	$S_{20}$	$S_{21}$	$S_{22}$	$S_{23}$	$S_{24}$	$S_{25}$
$P_5$	0	0	0	12	3,6	0	0	12	0	0	0	0
$P_5$	10,5	15	15	15	12	15	0	0	0	0	0	0
$P_6$	0	0	0	15	10,5	15	0	0	0	0	0	0
<b>P</b>	10,5	15	15	24,4	16,3	21,2	0	12	0	0	0	0

$P \backslash S_r$	$S_{26}$	$S_{27}$	$S_{28}$	$S_{29}$	$S_{30}$	$S_{31}$	$S_{32}$	$S_{33}$	$S_{34}$	$S_{35}$	$S_{36}$	$S_{37}$
$P_4$	0	0	0	8,4	8,4	0	0	0	0	0	0	0
$P_5$	0	0	0	7,5	7,5	0	0	0	0	0	0	0
$P_6$	0	0	0	15	10,5	3	3	3	3	3	3	0
<b>P</b>	0	0	0	18,8	15,4	3	3	3	3	3	3	0

$P \backslash S_r$	$S_{38}$	$S_{39}$	$S_{40}$	$S_{41}$	$S_{42}$	$S_{43}$	$S_{44}$	$S_{45}$	$S_{46}$	$S_{47}$	$S_{48}$	$S_{49}$
$P_4$	0	0	0	0	0	12	3,6	3,6	3,6	3,6	0	0
$P_5$	0	0	0	0	0	4,5	0	0	0	0	0	0
$P_6$	0	0	0	0	0	4,5	0	0	0	0	0	0
<b>P</b>	0	0	0	0	0	13,6	3,6	3,6	3,6	3,6	0	0

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таким образом, самыми критичными сетевыми функциональными элементами являются сетевые устройства  $S_{17}$  и  $S_{19}$ .

Используя методику расстановки приоритетов функциональных элементов в задачах администрирования распределенных информационных систем был произведен расчет приоритетов для элементов заданной сети. Результаты приведены в таблице 2.1.12.

Таблица 2.1.12 – Приоритеты элементов сети

<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>S<sub>3</sub></b>	<b>S<sub>4</sub></b>	<b>S<sub>5</sub></b>	<b>S<sub>6</sub></b>	<b>S<sub>7</sub></b>	<b>S<sub>8</sub></b>	<b>S<sub>9</sub></b>
3	3	3	3	3	3	2,7	2,7	2,7
<b>S<sub>18</sub></b>	<b>S<sub>19</sub></b>	<b>S<sub>20</sub></b>	<b>S<sub>21</sub></b>	<b>S<sub>22</sub></b>	<b>S<sub>23</sub></b>	<b>S<sub>24</sub></b>	<b>S<sub>25</sub></b>	<b>S<sub>26</sub></b>
28,2	21,2	9	22	17	1	1	0	0
<b>S<sub>35</sub></b>	<b>S<sub>36</sub></b>	<b>S<sub>37</sub></b>	<b>S<sub>38</sub></b>	<b>S<sub>39</sub></b>	<b>S<sub>40</sub></b>	<b>S<sub>41</sub></b>	<b>S<sub>42</sub></b>	<b>S<sub>43</sub></b>
3	3	0	0	3,4	0	3,4	3,4	28,4

<b>S<sub>10</sub></b>	<b>S<sub>11</sub></b>	<b>S<sub>12</sub></b>	<b>S<sub>13</sub></b>	<b>S<sub>14</sub></b>	<b>S<sub>15</sub></b>	<b>S<sub>16</sub></b>	<b>S<sub>17</sub></b>
2,7	3	3	21,2	20,3	27	24	46,1
<b>S<sub>27</sub></b>	<b>S<sub>28</sub></b>	<b>S<sub>29</sub></b>	<b>S<sub>30</sub></b>	<b>S<sub>31</sub></b>	<b>S<sub>32</sub></b>	<b>S<sub>33</sub></b>	<b>S<sub>34</sub></b>
0	0	32,5	32,1	3	3	3	3
<b>S<sub>44</sub></b>	<b>S<sub>45</sub></b>	<b>S<sub>46</sub></b>	<b>S<sub>47</sub></b>	<b>S<sub>48</sub></b>	<b>S<sub>49</sub></b>		
3,6	3,6	3,6	3,6	3,4	3,4		

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### *Экспериментальное получение значений интенсивности отказов узлов*

Целью экспериментов является получения экспериментальных значений переменных (интенсивности отказов  $\lambda$  и директивного времени выполнения задачи  $d_i$ ) в формулах расчета живучести узлов и коммуникационных каналов модели радиальной топологии.

Для получения этих значений был проведен ряд экспериментов на экспериментальной установке.

Каждый эксперимент соответствует одному из ранее описанных информационных процессов. Таким образом, были проведены следующие эксперименты:

- эксперимент по скачиванию файлов с FTP-сервера;
- эксперимент по скачиванию почтовых сообщений с почтового сервера;
- эксперимент по сохранению Web-страниц;
- эксперимент по обмену мгновенными сообщениями.

Рассмотрим подробнее каждый из этих экспериментов.

#### *Эксперимент по скачиванию файлов с FTP-сервера*

Целью данного эксперимента является определение переменных  $\lambda$  и  $d_i$  экспериментальным путем.

$\lambda$  - интенсивность отказов узла.

$d_i$  – директивный срок выполнения задачи (скачивания файлов).

Выполнение задачи в данном случае определяется по факту ее выполнения, то есть успел файл скачаться или нет. Поскольку размер файлов может сильно варьироваться, было решено выделить 3 типа файлов: большие, средние и малые. Размерность файла определялась исходя из средней пропускной способности моделируемой сети (1 Мбит = 128 Кб/сек)

Поскольку скорость в сети зависит от многих факторов, директивный срок выполнения задачи определялся как 80 % от средней пропускной способности сети (Таблица 2.1.13).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.13 – Ожидаемое значение  $d_i$  при различных размерах файлов в эксперименте с FTP-сервером

Тип файла	Размер, Мб	Время скачивания при 1Мбит, с	Ожидаемое время скачивания $\max d_i, c$
Большой	2	16	20
Средний	1	8	10
Малый	0,1	0,8	1

Если файл успевает скачаться за время  $d_i$ , то задача считается выполненной, если нет – то фиксируется отказ.

### *Методика проведения эксперимента.*

1. Установить и настроить на узле 22 согласно схеме на рисунке 2.19 настоящего документа FTP-сервер.
2. Создать 3 типа файлов, упомянутых выше и разрешить к ним доступ (скачивание).
3. Сделать настройку узлов 39-42, 48-49 для возможности скачивания с FTP-сервера файлов.
4. Осуществить несколько раз скачивание файлов разного размера для определения среднего времени скачивания.
5. Откорректировать при необходимости значения  $d_i$  для каждого типа файлов.
6. Провести эксперимент по скачиванию файлов в течении времени  $t$ , равному не менее 100 скачиванием файлов каждого типа; в ходе эксперимента осуществлять отключение случайных узлов каждые 1-2 мин., моделируя отказы узлов. Начать скачивание могут как одна машина, так и все (случайно).
7. Осуществить анализ файла логов экспериментальной установки для определения числа отказов и значения  $\lambda$ .

Схема сети, моделируемой в программе GNS3, приведена на рисунке 2.1.21.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

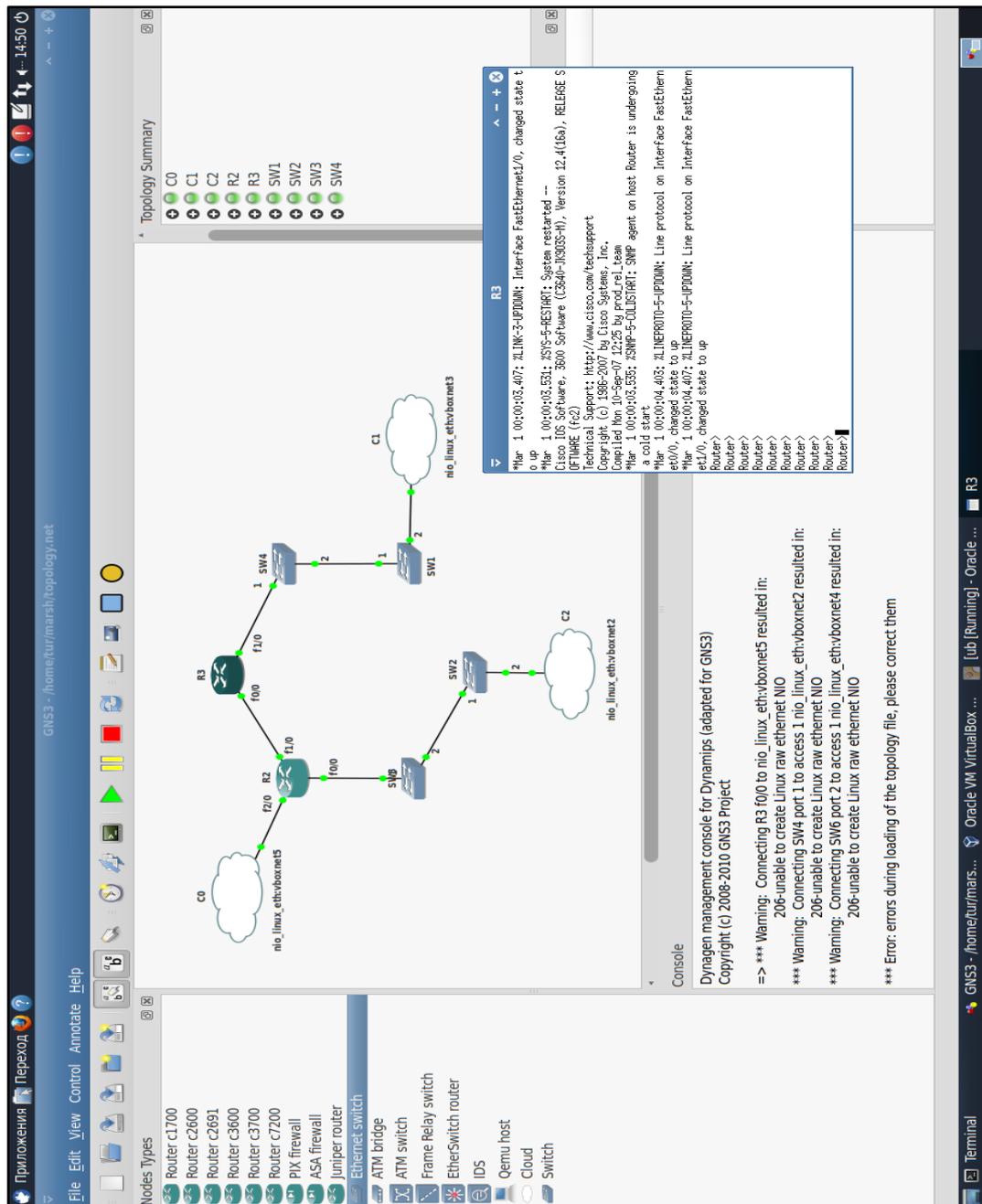


Рисунок 2.1.21– Схема сети в эксперименте с FTP-сервером

Для реализации вышеописанной методики был написан скрипт, автоматизирующий работу установки и имитирующий действия реальных пользователей.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Результаты проведения эксперимента представлены в виде таблицы отказов (Таблица 2.1.14).

Таблица 2.1.14 – Таблица отказов в эксперименте с FTP-сервером

№ виртуальной машины	Файл 100 Кб			Файл 1Мб		
	Попыток скачивания	Удачных	Отказов	Попыток скачивания	Удачных	Отказов
<b>1</b>	22	21	1	36	33	3
<b>2</b>	27	24	3	33	26	7
<b>3</b>	23	23	0	26	23	3
<b>4</b>	31	30	1	35	30	5
<b>5</b>	27	26	1	25	21	4
<b>6</b>	25	24	1	31	25	6
<b>Всего</b>	155	148	7	186	158	28

№ виртуальной машины	Файл 2 Мб		
	Попыток скачивания	Удачных	Отказов
<b>1</b>	35	24	11
<b>2</b>	37	24	13
<b>3</b>	34	26	8
<b>4</b>	32	25	7
<b>5</b>	31	24	7
<b>6</b>	35	30	5
<b>Всего</b>	204	153	51

Разделив общее количество отказов на общее количество попыток скачивания каждого из файлов, получим процент отказов:

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

100 кб.  $(7/155)*100 = 5\%$

1 Мб.  $(186/28)*100 = 18\%$

2 Мб.  $(51/204)*100 = 33\%$

Чтобы получить значение интенсивности отказов нужно разделить общее число отказов на время проведения эксперимента:

$$\lambda = 86/60 = 1,43 \text{ отказа в минуту}$$

Чтобы получить график отказов распишем их в интервалах времени равным 10 минутам (Таблица 2.1.15).

Таблица 2.1.15 – Таблица отказов в интервалах времени равным 10 минутам

Время, мин	Файл 100 Кб						Файл 1Мб					
	1	2	3	4	5	6	1	2	3	4	5	6
<b>0-10</b>	0	0	0	0	0	0	2	3	0	1	0	0
<b>10-20</b>	0	1	0	0	0	0	0	2	2	0	1	2
<b>20-30</b>	0	1	0	0	0	1	0	1	1	0	0	0
<b>30-40</b>	0	0	0	1	0	0	1	0	0	1	1	2
<b>40-50</b>	0	0	0	0	1	0	0	0	0	2	2	0
<b>50-60</b>	1	1	0	0	0	0	0	1	0	1	0	2

Время, мин	Файл 2 Мб					
	1	2	3	4	5	6
<b>0-10</b>	2	4	1	1	1	0
<b>10-20</b>	3	2	2	1	3	0
<b>20-30</b>	1	2	1	0	0	1
<b>30-40</b>	1	0	0	1	1	3
<b>40-50</b>	2	2	0	2	0	0
<b>50-60</b>	2	3	4	2	2	1

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

На рисунке 2.1.22 представлен график отказов в эксперименте с FTP-сервером

По графику видно, что наибольшее число отказов приходится на диапазон времени от 20 до 30 мин для файлов размером в 1 и 2 Мб, а также на последние 10 минут для файла весом 2Мб. Объясняется это тем, что виртуальные машины не сразу входят в «ритм» и постепенно нагружают установку. Конечный скачок, вероятно, является следствием большого количества случайно выпавших попыток скачивания файла размером 2Мб.

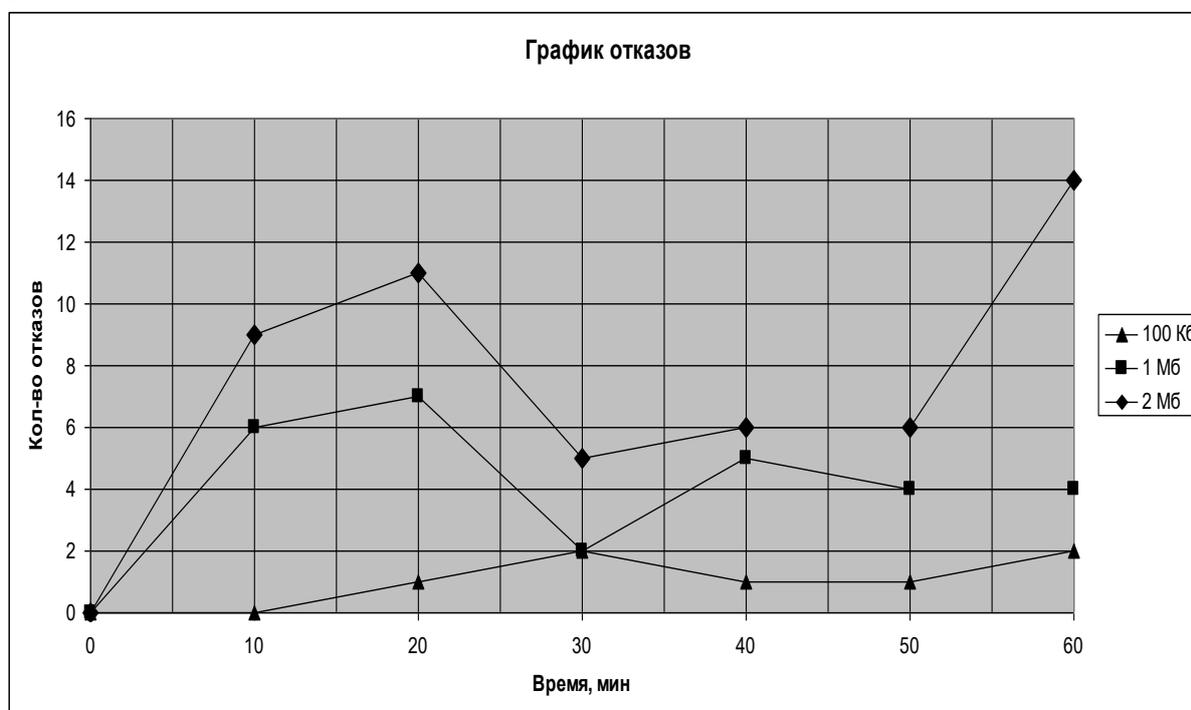


Рисунок 2.1.22– График отказов в эксперименте с FTP-сервером

Для более детального анализа построим график плотности отказов, отображающей количество отказов поминутно (Рисунок 2.1.23). Распределение отказов с интервалом в 1 минуту сведено в таблицу 2.1.16.

Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ  
В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.16 – Распределение отказов с интервалом в 1 минуту

	Время, мин	1	2	3	4	5	6	7	8	9	10	11	12
Кол- во от- казов	100 Кб	0	0	0	0	0	0	0	0	0	0	1	0
	1 Мб	0	0	0	1	1	0	1	0	3	0	0	3
	2 Мб	0	2	1	2	0	2	0	2	0	0	3	1

	Время, мин	13	14	15	16	17	18	19	20	21	22	23	24	25
Кол- во от- казов	100 Кб	0	0	0	0	0	0	0	0	0	1	1	0	0
	1 Мб	1	1	0	1	0	1	0	0	0	0	0	1	0
	2 Мб	3	0	2	1	0	1	0	0	2	0	0	1	1

	Время, мин	26	27	28	29	30	31	32	33	34	35	36
Кол- во от- казов	100 Кб	0	0	0	0	0	1	0	0	0	0	0
	1 Мб	0	1	0	0	0	2	1	0	0	1	0
	2 Мб	0	0	0	1	0	2	0	1	1	0	0

	Время, мин	37	38	39	40	41	42	43	44	45	46	47	48	49
Кол- во от- каза- зов	100 Кб	0	0	0	0	0	0	0	0	0	0	0	1	0
	1 Мб	1	0	0	0	1	1	0	1	1	0	0	0	0
	2 Мб	0	2	0	0	1	1	0	1	1	1	0	0	0

	Время, мин	50	51	52	53	54	55	56	57	58	59	60	Всего
Кол- во от- казов	100 Кб	0	0	1	0	1	0	0	0	0	0	0	7
	1 Мб	0	2	0	0	1	0	0	0	0	0	1	28
	2 Мб	1	3	0	2	2	0	0	1	0	1	5	51

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

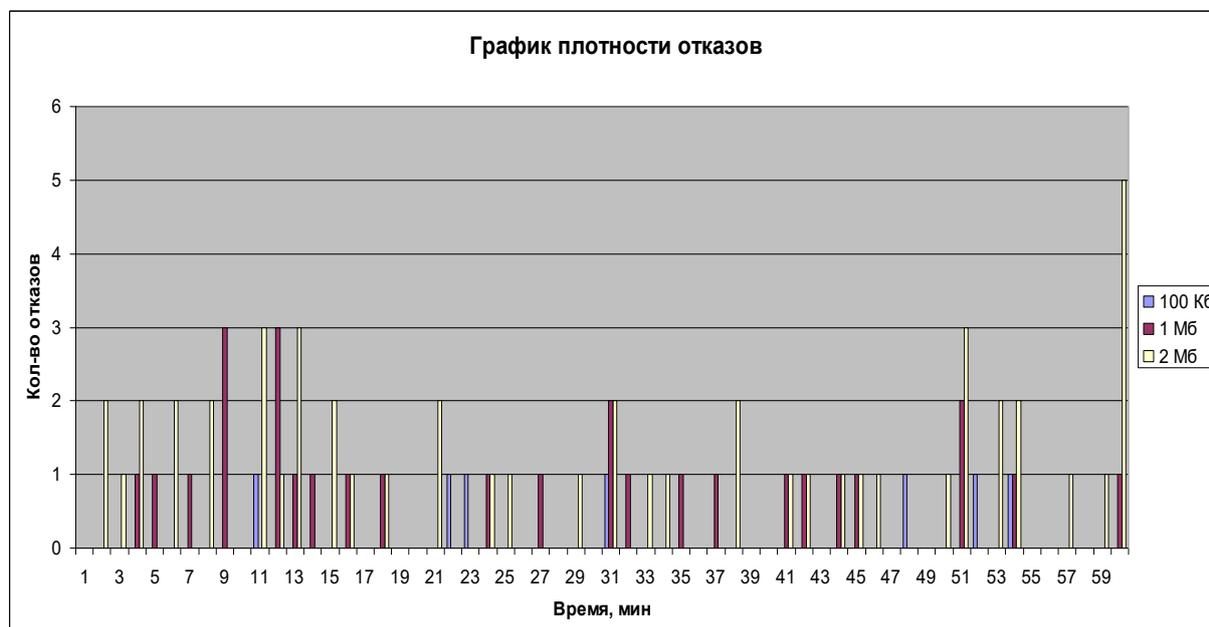


Рисунок 2.1.23 – График плотности отказов в эксперименте с  
FTP-сервером

Анализ графика показывает, что отказы распределились довольно равномерно. Однако есть их заметное превосходство в первые 20 минут проведения эксперимента, что можно объяснить медленным стартом установки и особенностью работы скрипта, имитирующего пользователей.

По графику плотности отказов можно построить функцию распределения интервалов отказов. Для этого посчитаем количество интервалов отказов в 1 минуту, 2 минуты и т.д. Далее определим число отказов, попавших в эти интервалы и разделив его на общее число отказов получим вероятность возникновения отказа  $P$  с данным интервалом. Данные занесем в таблицу (Таблица 2.1.17). По данным этой таблицы построим функцию распределения интервалов отказов (Рисунок 2.1.24).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.17 – Данные отказов с различными промежутками времени и вероятностью их возникновения в эксперименте с FTP-сервером

Промежуток времени между отказами	Число промежутков	Кол-во отказов	Вероятность, P
0	0	0	0
1	29	79	0,92
2	10	33	0,38
3	3	13	0,15
4	0	0	0
5	0	0	0



Рисунок 2.1.24 – Функция распределения интервалов отказов в эксперименте с FTP-сервером

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

### *Эксперимент по скачиванию почтовых сообщений с почтового сервера*

Целью данного эксперимента является определение переменных  $\lambda$  и  $d_i$  экспериментальным путем.

$\lambda$  - интенсивность отказов узла

$d_i$  – директивный срок выполнения задачи (скачивание почтового сообщения с вложениями)

Выполнение поставленной задачи фиксируется по факту ее выполнения. Считается, что проверяют почту накопленную за сутки. В течение дня сообщения приходят, но это слишком разряженные события, чтобы их можно было оценить в рамках данной работы. Моделируется получение 10 сообщений по 600 Кб (~ 6 Мб). Если задача не выполняется за директивный срок – фиксируется отказ.

#### *Методика проведения эксперимента*

1. Установить и настроить на узел 20 почтовый сервер.
2. Настроить и проверить работоспособность узлов 7-10 на возможность скачивания с SMTP-сервера почтовых сообщений и вложений.
3. Создать на сервере файл весом 6 Мб, имитирующий почтовые сообщения;
4. Осуществить скачивание данного файла для определения среднего времени скачивания.
5. Определить директивный срок выполнения задачи  $d_i$  (среднее время скачивания +20%).
6. Провести эксперимент по скачиванию почтовых сообщений в течении времени  $t$ , равному 1 часу; в ходе эксперимента осуществлять отключение случайных узлов каждые несколько минут, моделируя отказы узлов.
7. Осуществить анализ файла логов экспериментальной установки для определения числа отказов и значения  $\lambda$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Схема сети, моделируемой в программе GNS3, приведена на рисунках 2.1.25 и 2.1.26.

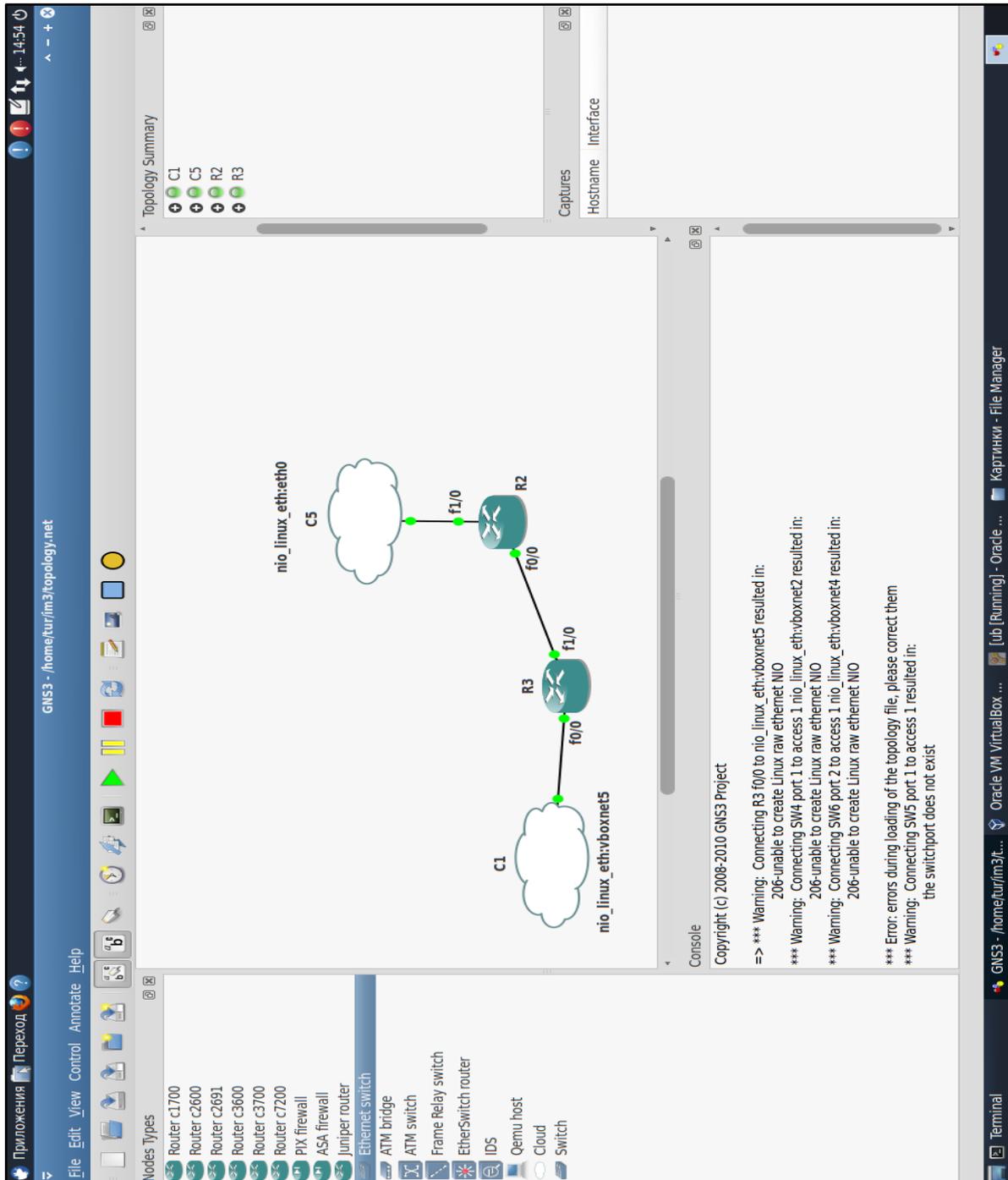


Рисунок 2.1.25 – Схема сети в эксперименте с почтовым сервером (нижний сегмент)

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

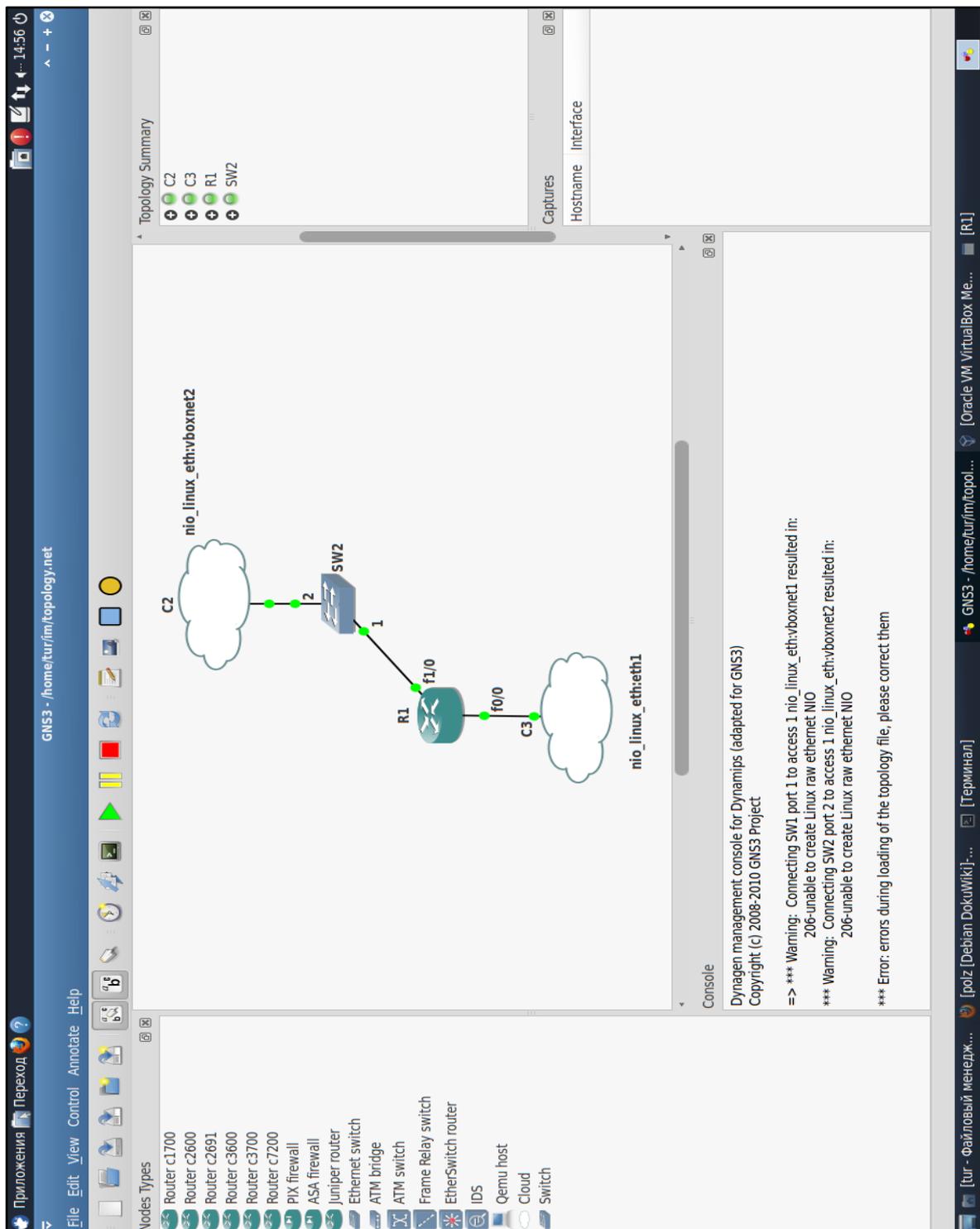


Рисунок 2.1.26 – Схема сети в эксперименте с почтовым сервером (верхний сегмент)

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Для реализации вышеописанной методики был написан скрипт, автоматизирующий работу установки и имитирующий действия реальных пользователей.

Результаты проведения эксперимента представлены в виде таблицы отказов (Таблица 2.1.18).

Таблица 2.1.18 – Таблица отказов в эксперименте с почтовым сервером

№ виртуальной машины	Попыток скачивания	Удачных	Отказов
1	59	45	14
2	61	48	13
3	59	43	16
4	58	41	17
<b>Всего</b>	237	177	60

Разделив общее количество отказов на общее количество попыток скачивания каждого из файлов, получим процент отказов.

$$(60/237)*100 = 25,3\%$$

Чтобы получить значение интенсивности отказов нужно разделить общее число отказов на время проведения эксперимента:

$$\lambda = 60/60 = 1 \text{ отказ в минуту}$$

Чтобы получить график распределения отказов (Рисунок 2.1.27) распишем их в интервалах времени равным 10 минутам (Таблица 2.1.19). Малое число отказов в начале объясняется медленным стартом экспериментальной установки и особенностью работы скрипта, имитирующего работу реальных пользователей. Впоследствии нагрузка на установку более равномерная и количество отказов лежит в пределах 10-12 за 10 минут.

Для более детального анализа построим график плотности отказов, отображающей количество отказов поминутно (Рисунок 2.1.28). С распределением отказов можно ознакомиться в Таблице 2.1.20.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.19 – Таблица отказов в интервалах времени равным 10 минутам в эксперименте с почтовым сервером

Время, мин	Почтовое сообщение с вложениями (6 Мб)			
	1	2	3	4
<b>0-10</b>	2	2	3	1
<b>10-20</b>	2	1	3	4
<b>20-30</b>	2	3	2	2
<b>30-40</b>	3	4	1	2
<b>40-50</b>	3	2	4	3
<b>50-60</b>	2	1	3	5
<b>Всего</b>	14	13	16	17

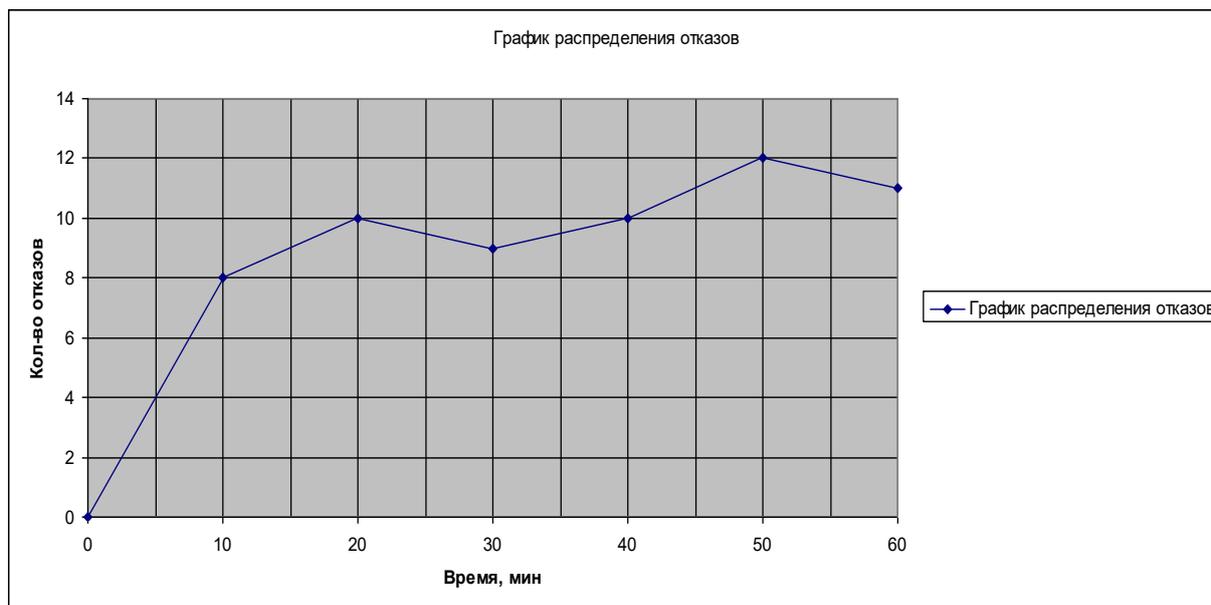


Рисунок 2.1.27 – График распределения отказов в эксперименте с почтовым сервером

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

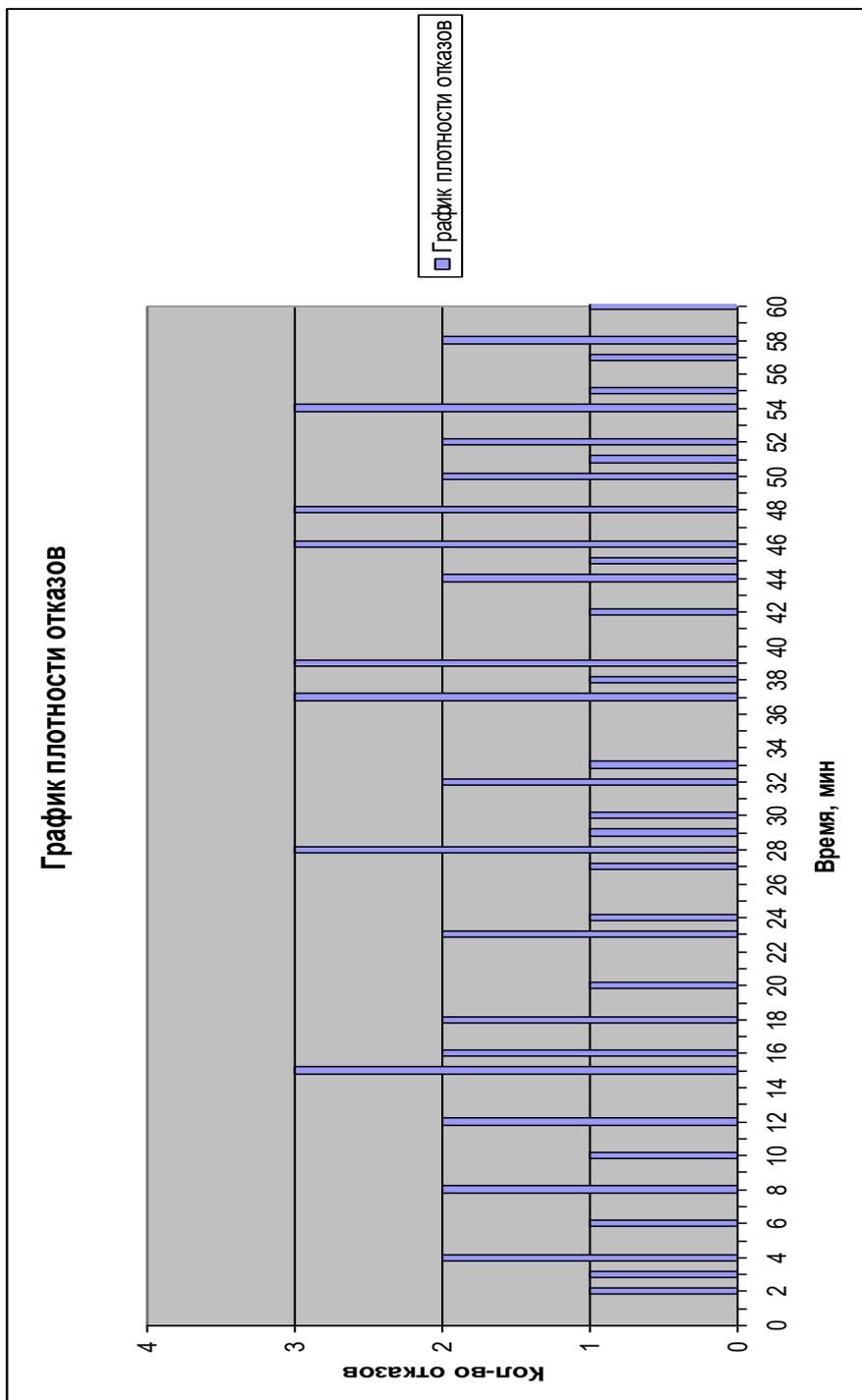


Рисунок 2.1.28 – График плотности отказов в эксперименте с почтовым сервером

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Анализ графика показывает, что отказы распределены равномерно и подтверждает случайность их возникновения посредством скрипта, имитирующего активность реальных пользователей.

По графику плотности отказов можно построить функцию распределения интервалов отказов. Для этого посчитаем количество интервалов отказов в 1 минуту, 2 минуты и т.д. Далее определим число отказов, попавших в эти интервалы и разделив его на общее число отказов получим вероятность возникновения отказа  $P$  с данным интервалом. Данные занесем в таблицу (Таблица 2.1.20). По данным этой таблицы построим функцию распределения интервалов отказов (Рисунок 2.1.29). Таблица 2.1.21 содержит данные отказов с различными промежутками времени и вероятностью их возникновения в эксперименте с почтовым сервером. График функции распределения интервалов отказов в эксперименте с почтовым сервером приведен на рисунке 2.1.29.

Таблица 2.1.20 – Распределение отказов с интервалом в 1 минуту в эксперименте с почтовым сервером

<b>Время, мин</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
<b>Отказов</b>	0	0	1	1	2	0	1	0	2	0	1	0	2

<b>Время, мин</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>
<b>Отказов</b>	0	0	3	2	0	2	0	1	0	0	2	1	0

<b>Время, мин</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>	<b>37</b>	<b>38</b>	<b>39</b>
<b>Отказов</b>	0	1	3	1	1	0	2	1	0	0	0	3	1	3

<b>Время, мин</b>	<b>40</b>	<b>41</b>	<b>42</b>	<b>43</b>	<b>44</b>	<b>45</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>
<b>Отказов</b>	0	0	1	0	2	1	3	0	3	0

<b>Время, мин</b>	<b>50</b>	<b>51</b>	<b>52</b>	<b>53</b>	<b>54</b>	<b>55</b>	<b>56</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>60</b>	<b>Всего</b>
<b>Отказов</b>	2	1	2	0	3	1	0	1	2	0	1	<b>60</b>

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.21 – Таблица отказов с различными промежутками времени и вероятностью их возникновения в эксперименте с почтовым сервером

Промежуток времени между отказами	Число промежутков	Кол-во отказов	Вероятность, P
0	0	0	0
1	17	54	0,9
2	13	46	0,76
3	4	14	0,23
4	1	4	0,07
5	0	0	0
6	0	0	0



Рисунок 2.1.29 – Функция распределения интервалов отказов в эксперименте с почтовым сервером

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### Эксперимент по сохранению Web-страниц

Целью данного эксперимента является определение переменных  $\lambda$  и  $d_i$  экспериментальным путем.

$\lambda$  - интенсивность отказов узла

$d_i$  – директивный срок выполнения задачи (сохранение Web-страниц)

Выполнение задачи определяется по факту подтверждения полной загрузки страницы. Если задача не выполняется за директивный срок – фиксируется отказ.

Поскольку размер файлов может сильно варьироваться, было решено создать 4 web-страницы разного размера (Таблица 2.1.22) Скорость в сети зависит от многих факторов, директивный срок выполнения задачи определялся как 80% от средней пропускной способности сети в 1 Мбит/сек.

Таблица 2.1.22 – Ожидаемое значение  $d_i$  при различных размерах файлов

№ страницы	Размер, Мб	Время скачивания при 1Мбит, с	Ожидаемое время скачивания $\max d_i, c$
1	2	16	20
2	1	8	10
3	0,6	4,7	5,6
4	0,2	1,6	1,9

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

### *Методика проведения эксперимента*

1. Установить и настроить Web-сервер на узле 21;
2. Установить и настроить консольный браузер на узлы 4-6, 23-23, 44-47 и проверить работоспособность;
3. Осуществить скачивание страниц с нескольких узлов с целью установить среднее время загрузки сайта;
4. Определить директивной срок выполнения задачи  $d_i$  (среднее время загрузки сайта +20%);
5. Провести эксперимент по скачиванию Web-страниц с 9 узлов в течение часа. Сымитировать отказ узлов путем их периодического отключения.
6. Осуществить анализ файла логов экспериментальной установки для определения числа отказов и значения  $\lambda$ .

Схема сети, моделируемой в программе GNS3, приведена на рисунках 2.1.30 – 2.1.31.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

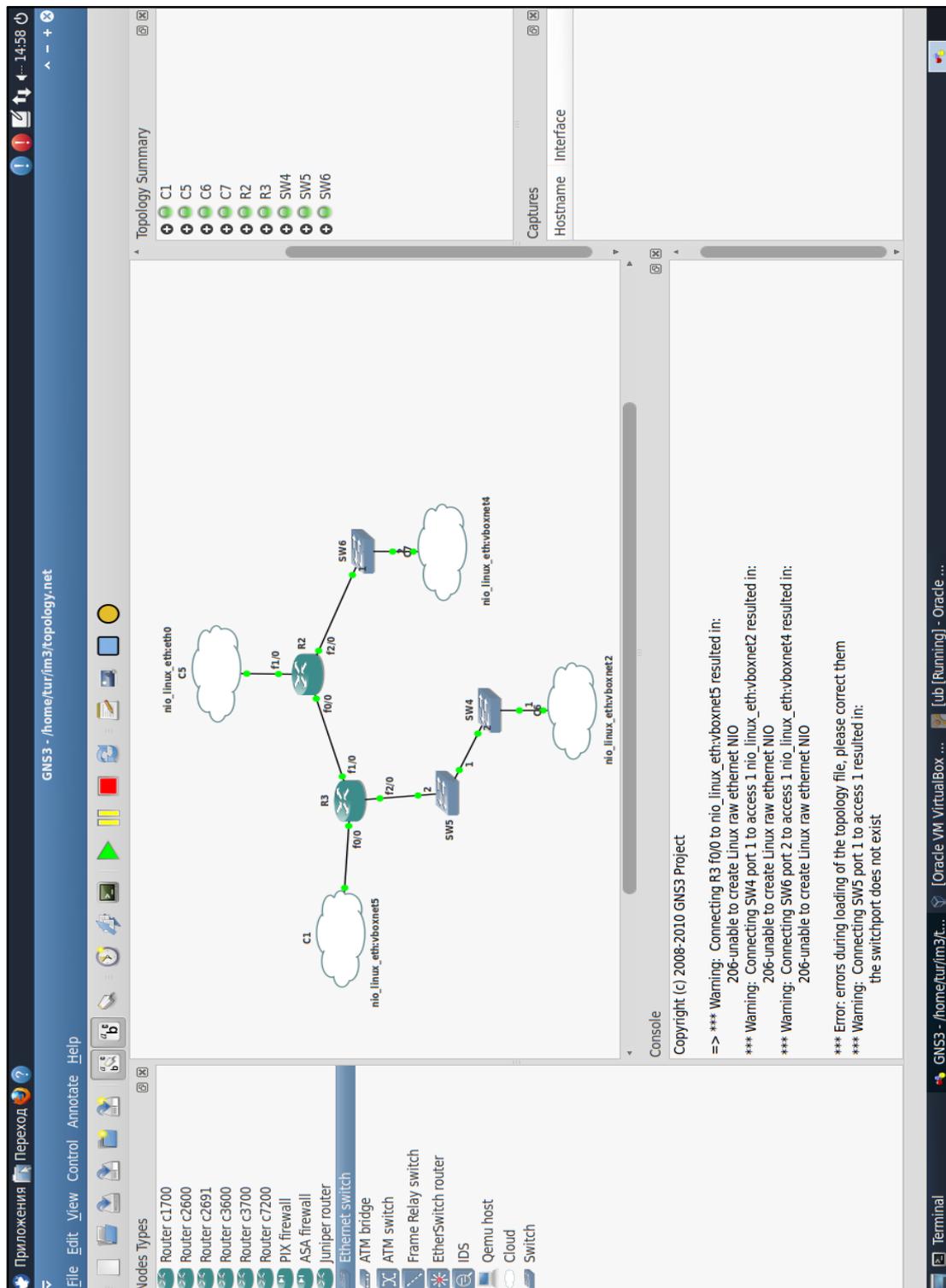


Рисунок 2.1.30 – Схема сети в эксперименте с Web-сервером  
(нижний сегмент)

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

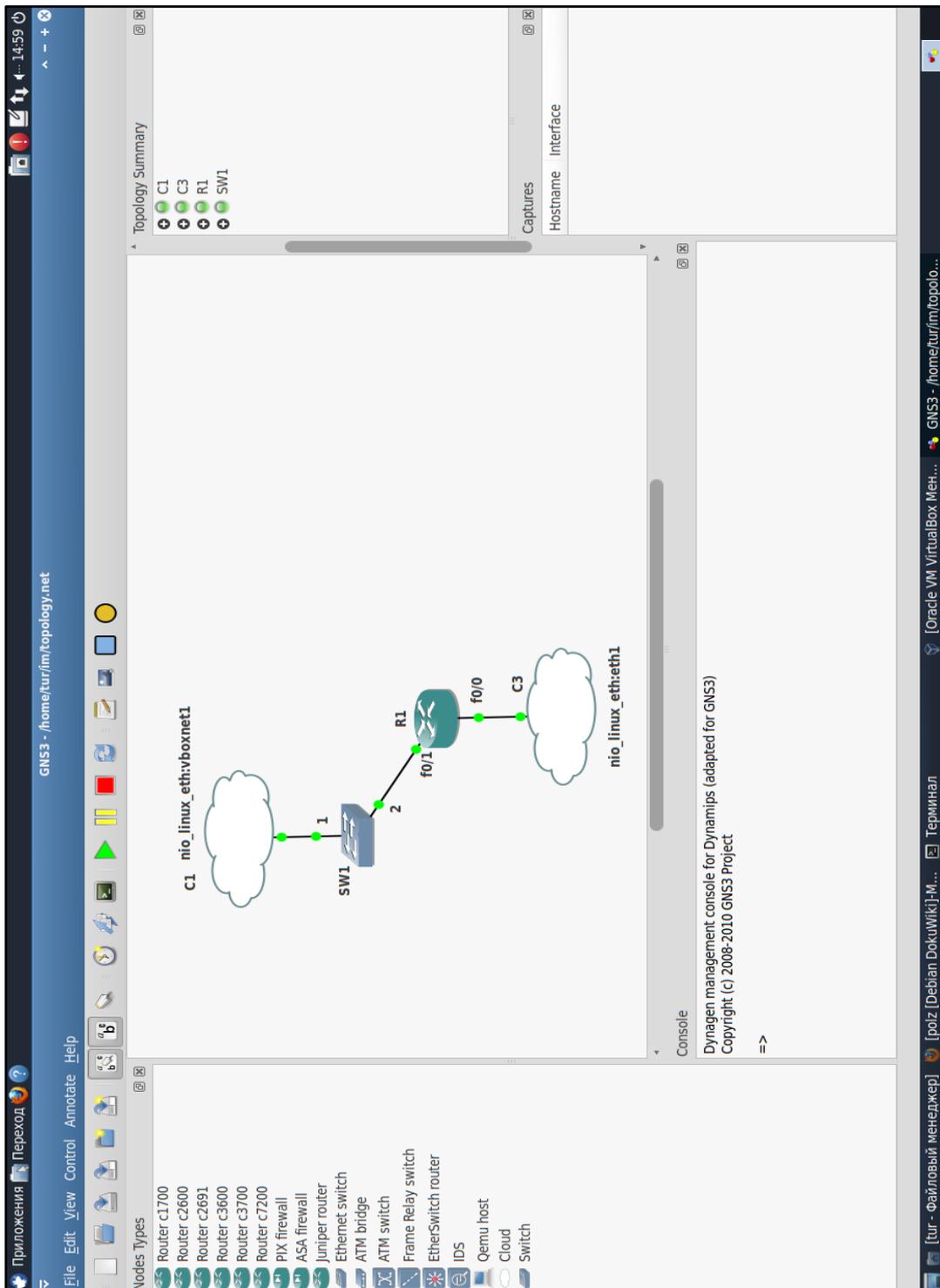


Рисунок 2.1.31 – Схема сети в эксперименте с Web-сервером (верхний сегмент)

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Для реализации вышеописанной методики был написан скрипт, автоматизирующий работу установки и имитирующий действия реальных пользователей.

Результаты проведения эксперимента представлены в виде таблицы 2.1.23.

Разделив общее количество отказов на общее количество попыток скачивания каждого из файлов, получим процент отказов.

$$(142/1092)*100 = 13\%$$

Чтобы получить значение интенсивности отказов нужно разделить общее число отказов на время проведения эксперимента:

$$\lambda = 142/60 = 2,36 \text{ отказа в минуту}$$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.23 – Таблица отказов в эксперименте с Web-сервером

№ виртуальной машины	200 Кб			600 Кб		
	Попыток скачивания	Удачных	Отказов	Попыток скачивания	Удачных	Отказов
1	30	29	1	32	31	1
2	31	29	2	30	27	3
3	29	29	0	31	28	3
4	32	29	3	29	26	3
5	28	27	1	30	28	2
6	29	28	1	26	25	1
7	33	30	3	32	28	4
8	30	28	2	31	28	3
9	30	28	2	28	26	2
<b>Всего</b>	<b>272</b>	<b>257</b>	<b>15</b>	<b>269</b>	<b>247</b>	<b>22</b>

№ виртуальной машины	1 Мб			2 Мб		
	Попыток скачивания	Удачных	Отказов	Попыток скачивания	Удачных	Отказов
1	29	26	3	27	22	5
2	33	26	7	32	25	7
3	30	25	5	30	24	6
4	32	28	4	34	23	11
5	31	26	5	29	23	6
6	27	24	3	32	24	8
7	34	28	6	30	24	6
8	29	26	3	30	21	9
9	30	26	4	32	25	7
<b>Всего</b>	<b>275</b>	<b>235</b>	<b>40</b>	<b>276</b>	<b>211</b>	<b>65</b>

Чтобы получить график отказов (Рисунок 2.1.32) распишем их в интервалах времени равным 10 минутам (Таблица 2.1.24).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.24 – Таблица отказов в интервалах времени равным  
10 минутам в эксперименте с Web-сервером

Т	Страница 200 Кб									Страница 600 Кб								
	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
<b>10</b>	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0
<b>20</b>	0	1	0	1	0	0	1	0	0	0	1	1	0	0	0	1	1	0
<b>30</b>	1	0	0	0	1	0	0	1	0	0	1	0	1	1	0	0	0	0
<b>40</b>	0	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	0	1
<b>50</b>	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	1	1	0
<b>60</b>	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1

Т	Страница 1 Мб									Страница 2 Мб								
	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
<b>10</b>	1	0	1	0	0	0	1	0	0	1	0	1	1	0	2	0	1	0
<b>20</b>	0	1	1	1	2	1	2	1	1	2	1	1	0	0	1	2	1	2
<b>30</b>	1	2	0	0	1	0	0	0	2	0	2	0	3	1	1	2	3	0
<b>40</b>	0	1	1	2	2	1	2	0	1	0	1	3	2	3	0	0	0	4
<b>50</b>	0	2	0	1	0	0	1	1	0	1	1	0	3	2	1	2	1	0
<b>60</b>	1	1	2	0	0	1	0	1	0	1	2	1	2	0	3	0	3	1

Малое число отказов в начале эксперимента объясняется медленным стартом экспериментальной установки и особенностью работы скрипта, имитирующего работу реальных пользователей. Далее по мере увеличения нагрузки на имитируемую сеть, количество отказов для каждого размера файла носит более равномерный характер.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

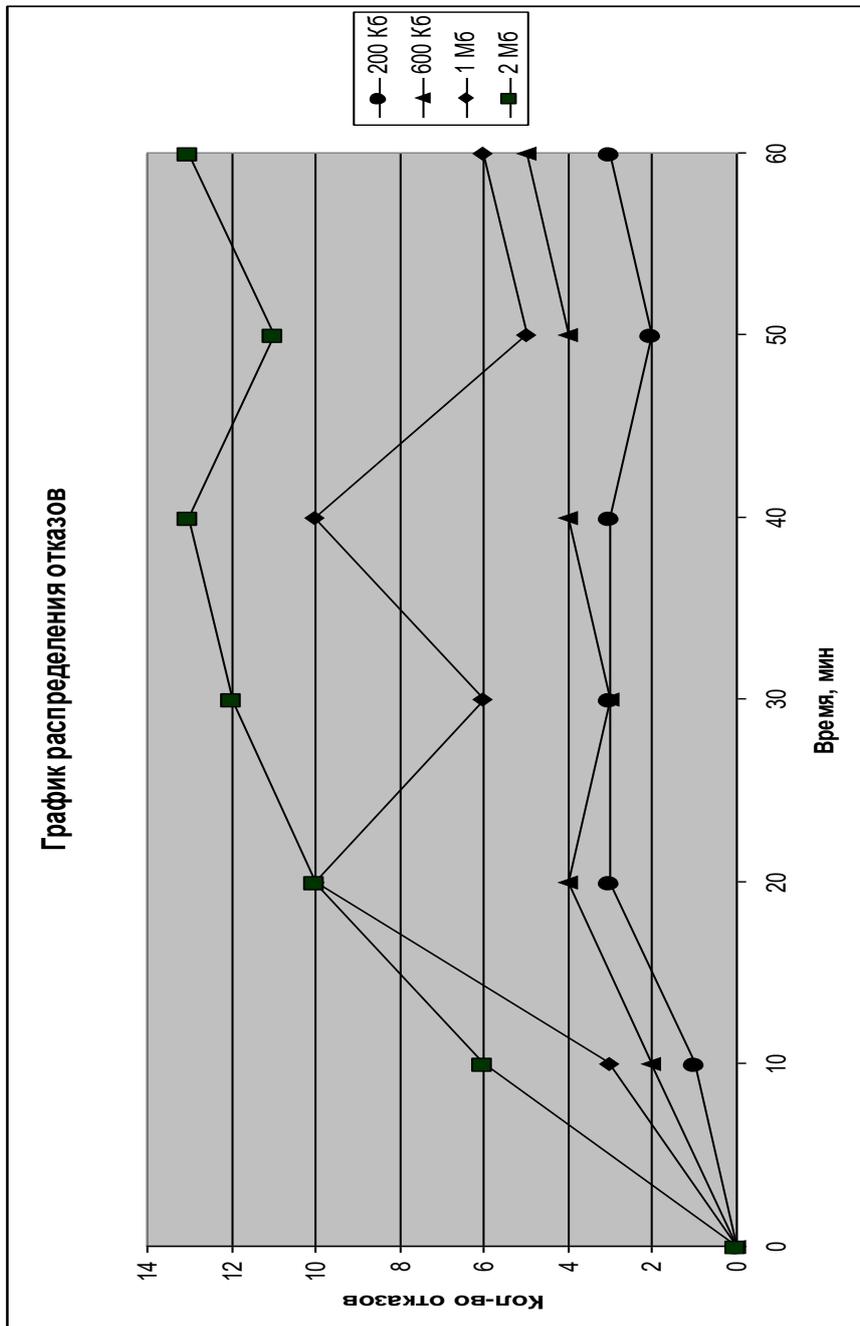


Рисунок 2.1.32 – График распределения отказов в эксперименте с Web-сервером

Для более детального анализа построим график плотности отказов, отображающей количество отказов поминутно (Рисунок 2.1.33). С распределением отказов можно ознакомиться в (Таблица 2.1.25).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.25 – Распределение отказов с интервалом в 1 минуту  
в эксперименте с Web-сервером

Время, мин	0	1	2	3	4	5	6	7	8	9	10	11	12
<b>200 Кб</b>	0	0	0	0	0	0	0	1	0	0	0	1	0
<b>600 Кб</b>	0	0	0	0	1	0	0	0	0	1	0	1	1
<b>1 Мб</b>	0	0	0	0	1	0	0	1	0	1	0	3	0
<b>2 Мб</b>	0	0	0	0	2	0	0	2	0	2	0	2	0

Время, мин	13	14	15	16	17	18	19	20	21	22	23	24
<b>200 Кб</b>	0	0	1	0	0	0	1	0	0	1	1	0
<b>600 Кб</b>	0	0	1	0	0	1	0	0	0	2	0	0
<b>1 Мб</b>	0	1	2	0	0	1	3	0	0	2	0	0
<b>2 Мб</b>	0	2	3	0	0	0	3	0	0	4	0	0

Время, мин	25	26	27	28	29	30	31	32	33	34	35	36
<b>200 Кб</b>	0	0	0	1	0	0	1	0	1	0	0	0
<b>600 Кб</b>	1	0	0	0	0	0	1	0	0	0	1	0
<b>1 Мб</b>	1	0	0	3	0	0	2	0	1	0	2	1
<b>2 Мб</b>	3	1	0	3	1	0	2	0	3	1	0	1

Время, мин	37	38	39	40	41	42	43	44	45	46	47	48	49
<b>200 Кб</b>	0	0	1	0	0	0	1	0	0	0	1	0	0
<b>600 Кб</b>	0	2	0	0	0	0	1	0	1	0	2	0	0
<b>1 Мб</b>	0	3	1	0	0	0	2	0	0	0	3	0	0
<b>2 Мб</b>	0	4	1	1	1	0	3	0	2	0	3	0	1

Время, мин	50	51	52	53	54	55	56	57	58	59	60	Всего
<b>200 Кб</b>	0	1	0	1	0	0	0	0	1	0	0	15
<b>600 Кб</b>	0	1	0	2	0	0	0	0	2	0	0	22
<b>1 Мб</b>	0	2	0	1	0	0	0	1	2	0	0	40
<b>2 Мб</b>	1	2	0	3	1	0	1	0	4	1	1	65

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

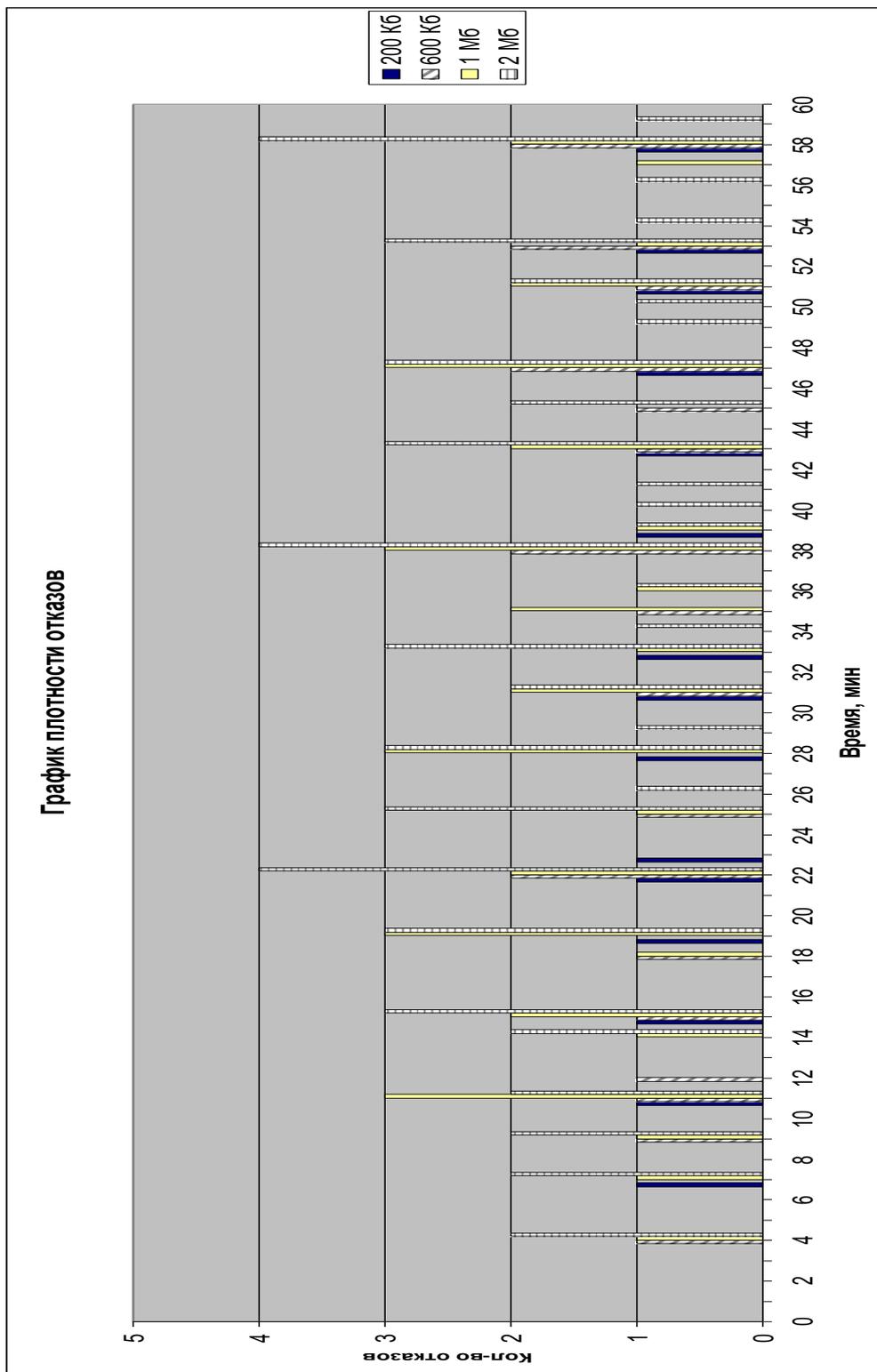


Рисунок 2.1.33 – График плотности отказов в эксперименте с Web-сервером

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Анализ графика показывает, что отказы в общем плане распределены равномерно, что подтверждает случайность их возникновения посредством скрипта, имитирующего активность реальных пользователей. Малое по сравнению с общим фоном число отказов в начале эксперимента, как уже было сказано, объясняется медленным стартом экспериментальной установки.

По графику плотности отказов можно построить функцию распределения интервалов отказов. Для этого посчитаем количество интервалов отказов в 1 минуту, 2 минуты и т.д. Далее определим число отказов, попавших в эти интервалы и разделив его на общее число отказов получим вероятность возникновения отказа  $P$  с данным интервалом. Данные занесем в таблицу (Таблица 2.1.26).

Таблица 2.1.26 – Таблица отказов с различными промежутками времени и вероятностью их возникновения в эксперименте с Web-сервером

Промежуток времени между отказами	Число промежутков	Кол-во отказов	Вероятность, $P$
0	0	0	0
1	18	117	0,82
2	14	121	0,85
3	3	33	0,23
4	0	0	0
5	0	0	0

График функции распределения интервалов отказов в эксперименте с Web-сервером приведен на рисунке 2.1.34.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

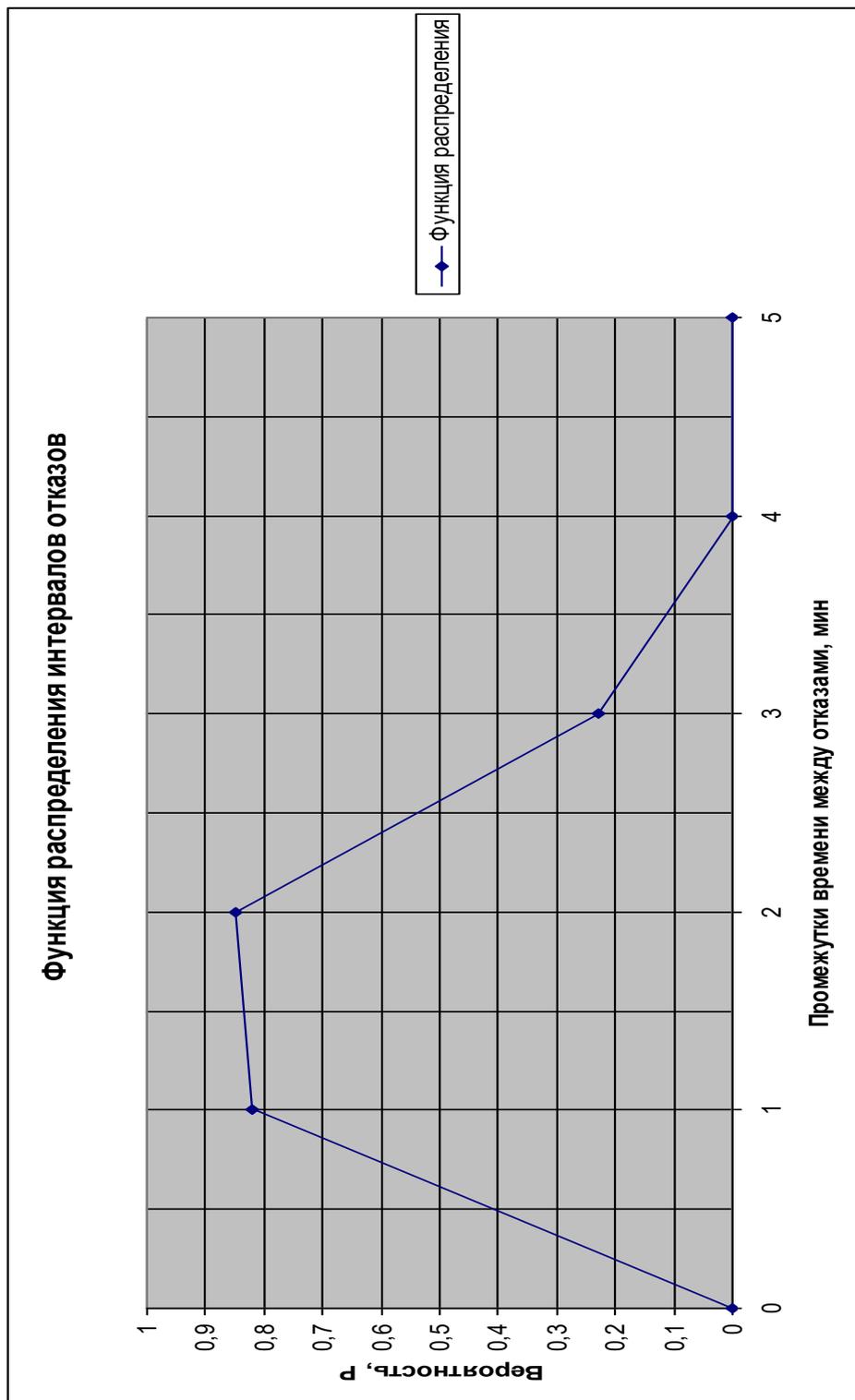


Рисунок 2.1.34 – График функции распределения интервалов отказов в эксперименте с Web-сервером

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### *Эксперимент по обмену мгновенными сообщениями*

Целью данного эксперимента является определение переменных  $\lambda$  и  $d_i$  экспериментальным путем.

$\lambda$  - интенсивность отказов узла

$d_i$  – директивный срок выполнения задачи (отправки/получения сообщений)

В данном случае задача считается выполненной в случае доставки сообщения средней длины за заданное время  $d_i$ . Если время доставки превышает заданное значение – фиксируется отказ.

Для определения среднего размера сообщения был проанализирован лог файла сообщений программы QIP2005 за 10 месяцев. Средняя длина сообщения составила 30-40 символов, что составляет по 30-40 байт, или если 2-байтовая система обмена мгновенными сообщениями - 60-80 байт. Среднее время доставки сообщения устанавливается экспериментальным путем.

#### *Методика проведения эксперимента*

1. Установить и настроить IM-сервер на узел №19;
2. Установить клиенты на узлы 1-3, 11-12, 31-36 для возможности обмена сообщениями;
3. Произвести обмен сообщениями средней длины между узлами для установки среднего значения времени доставки сообщений;
4. Принять среднее время доставки сообщения за  $d_i$ ;
5. Провести эксперимент по обмену IM-сообщения в течение времени  $t$  (1 час);
6. Осуществить анализ файла логов экспериментальной установки для определения числа отказов и значения  $\lambda$ .

Схема сети, моделируемой в программе GNS3, приведена на рисунках 2.1.35 и 2.1.36.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

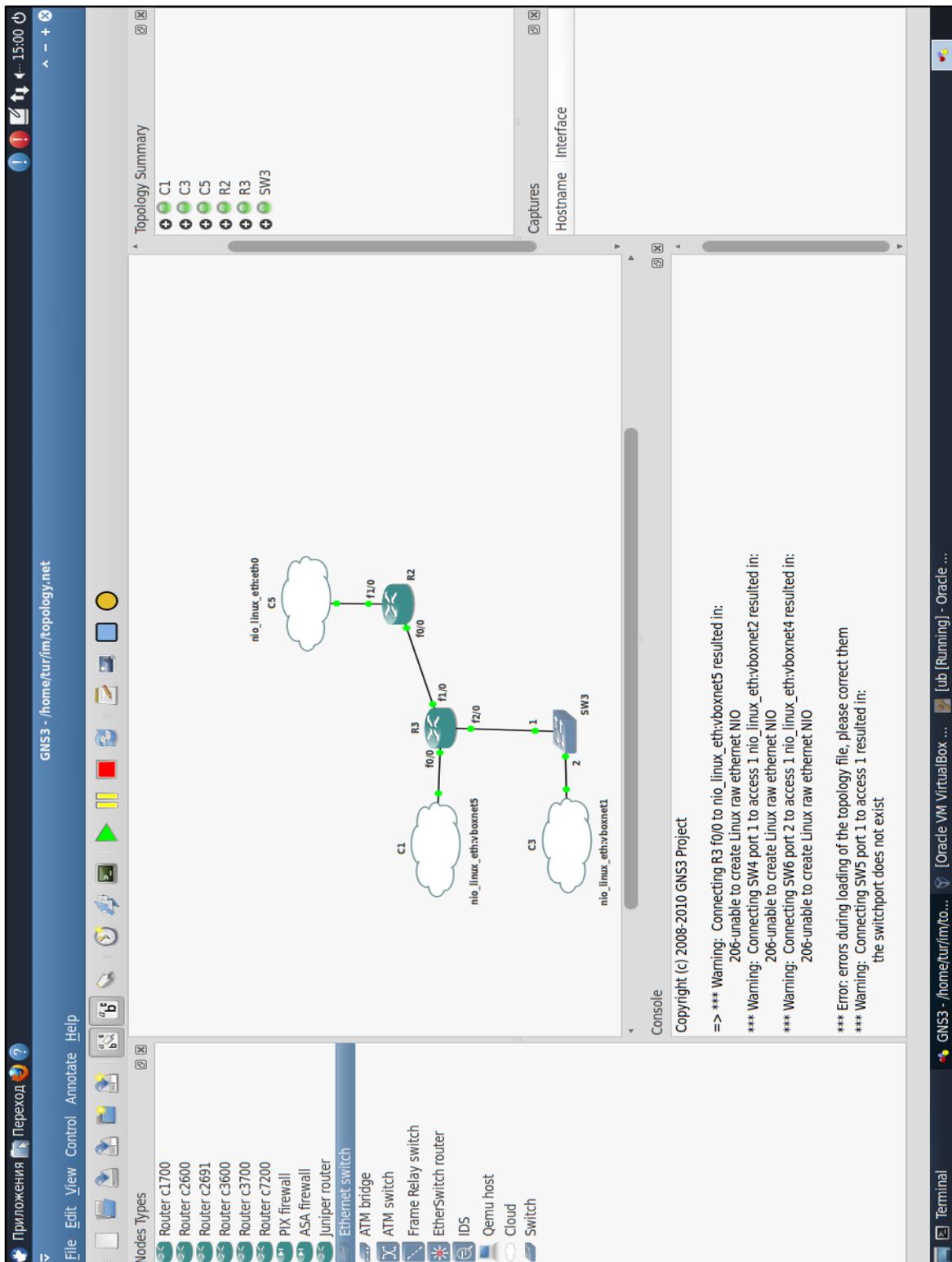


Рисунок 2.1.35 – Схема сети в эксперименте с ИМ-сервером  
(нижний сегмент)

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

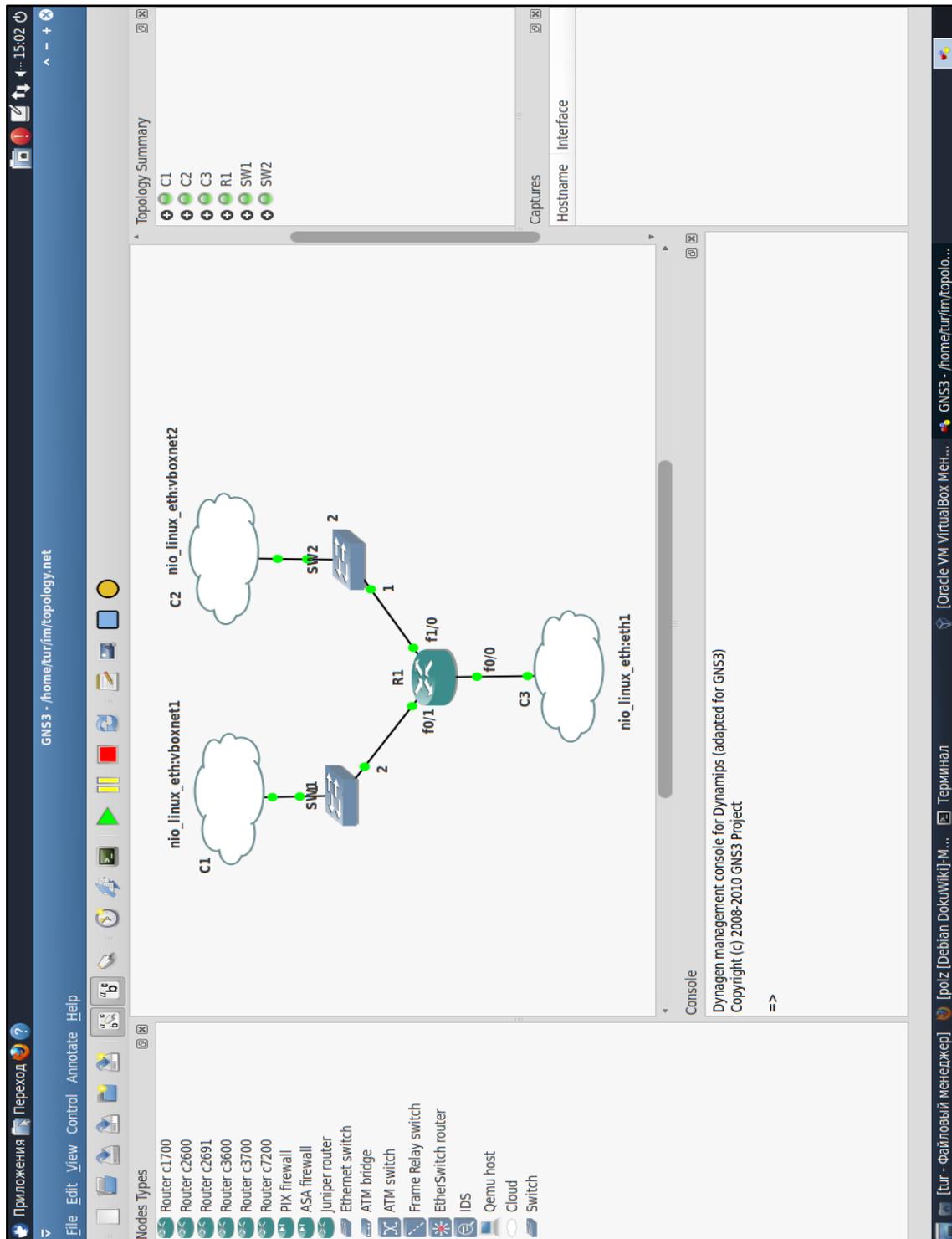


Рисунок 2.1.36 – Схема сети в эксперименте с ИМ-сервером  
(верхний сегмент)

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

В отличие от предыдущих экспериментов, в данном случае не пришлось имитировать работу реальных пользователей. Несколько человек общались посредством системы обмена мгновенными сообщениями Jabber.

Результаты проведения эксперимента представлены в виде таблицы отказов (Таблица 2.1.27).

Таблица 2.1.27 – Таблица отказов в эксперименте с IM-сервером

<b>Время, мин</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>	<b>30</b>	<b>35</b>	<b>40</b>	<b>45</b>	<b>50</b>	<b>55</b>	<b>60</b>
<b>Отказов</b>	12	9	5	0	0	0	0	0	0	0	0	0

Разделив общее количество отказов на общее количество попыток скачивания каждого из файлов, получим процент отказов.

$$(26/639)*100 = 4\%$$

Чтобы получить значение интенсивности отказов нужно разделить общее число отказов на время проведения эксперимента:

$$\lambda = 26/60 = 0,43 \text{ отказа в минуту}$$

Построим график отказов по данным, представленным в Таблице 2.1.28 (Рисунок 2.1.37).

Таблица 2.1.28 – Распределение отказов с интервалом  
в 1 минуту в эксперименте с IM-сервером

<b>Время, мин</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
<b>Отказов</b>	0	11	0	0	0	2	7	0	0	0	0	2	4	0	0

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

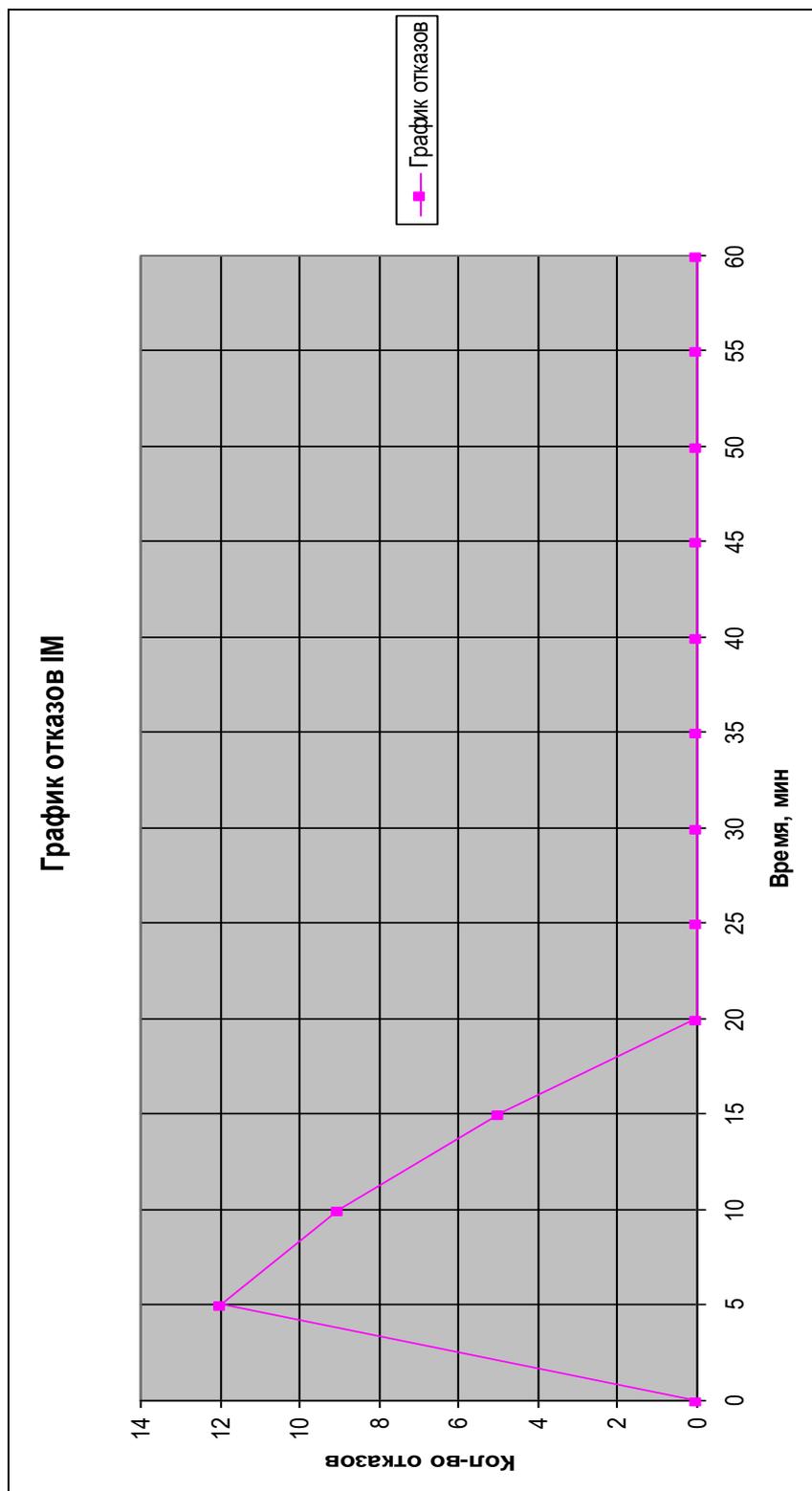


Рисунок 2.1.37– График отказов в эксперименте с IM-сервером

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
Как видно из графика, все отказы приходятся на первые 15 минут эксперимента. Это объясняется тем, что в условиях ограниченных вычислительных ресурсов обращение виртуальных машин к серверу довольно сильно загружает его, в результате чего возникают ощутимые задержки в доставке сообщений.

Для более детального анализа построим график плотности отказов, отображающей количество отказов поминутно (Таблица 2.1.29, Рисунок 2.1.38).

Таблица 2.1.29 – Таблица отказов с различными промежутками времени и вероятностью их возникновения в эксперименте с ИМ-сервером

<b>Промежуток времени между отказами</b>	<b>Число промежутков</b>	<b>Кол-во отказов</b>	<b>Вероятность, P</b>
<b>0</b>	0	0	0
<b>1</b>	2	15	0,58
<b>2</b>	0	0	0
<b>3</b>	0	0	0
<b>4</b>	1	13	0,5
<b>5</b>	1	9	0,35

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

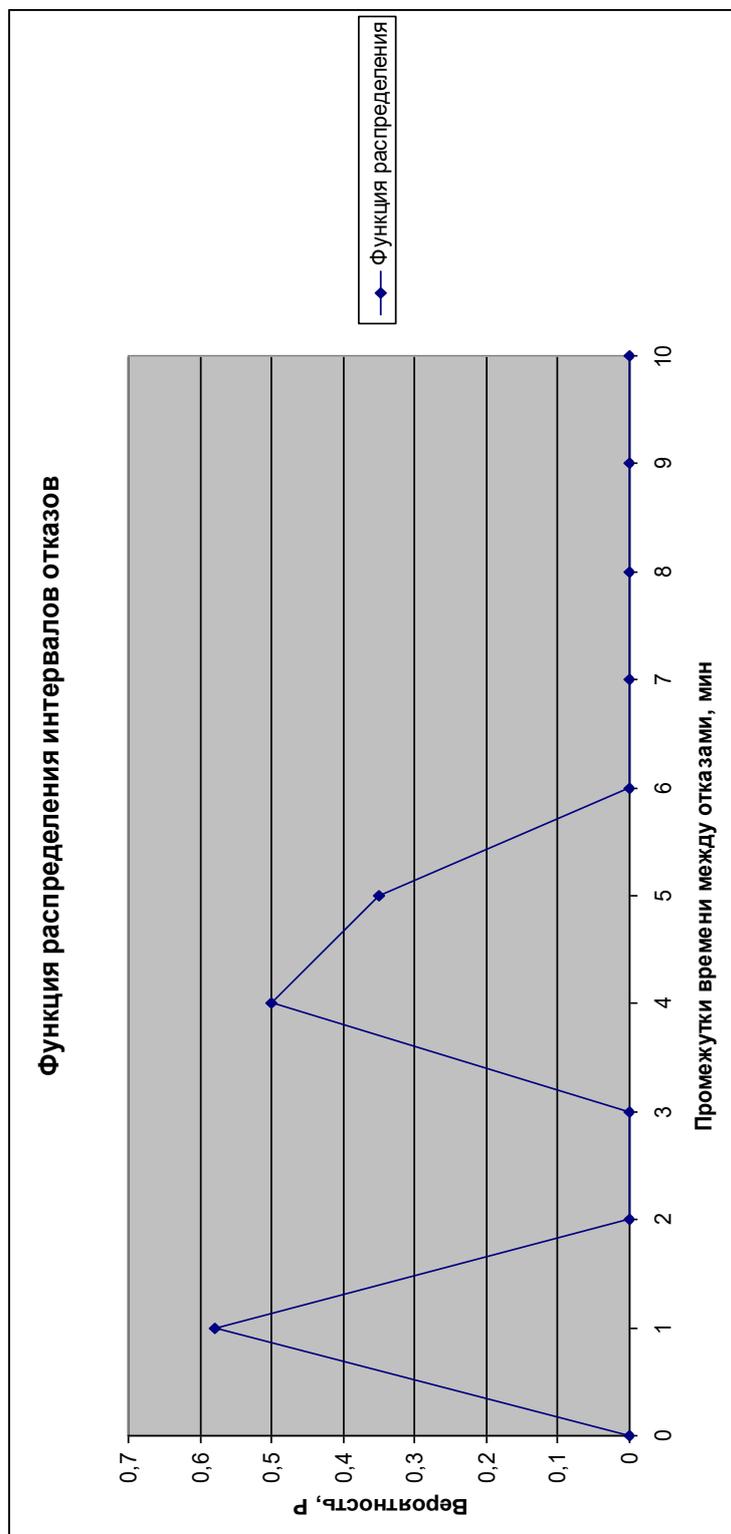


Рисунок 2.1.38 – Функция распределения интервалов отказов  
в эксперименте с ИМ-сервером

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
Попадающее число отказов приходится на 2 минуту проведения эксперимента, далее их число падает. Возможно, некоторые из клиентов не успели получить список контактов, поэтому постоянные запросы на сервер загружают его, вследствие чего могут возникать отказы.

По графику плотности отказов можно построить функцию распределения интервалов отказов. Для этого посчитаем количество интервалов отказов в 1 минуту, 2 минуты и т.д. Далее определим число отказов попавших в эти интервалы и разделив его на общее число отказов получим вероятность возникновения отказа  $P$  с данным интервалом.

График плотности отказов в эксперименте с ИМ-сервером представлен на рисунке 2.1.39.



Рисунок 2.1.39 – График плотности отказов в эксперименте с ИМ-сервером

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таким образом, в результате экспериментов были получены значения интенсивности отказов и директивные времена выполнения задач, представленные в таблице 2.1.30.

Таблица 2.1.30 – Результаты проведенных экспериментов

№	Название эксперимента	Значение $d_i$ , с	Значение $\lambda$ , 1/мин
1	Эксперимент по скачиванию файлов с FTP-сервера	35	1,43
2	Эксперимент по скачиванию почтовых сообщений с почтового сервера.	35	1
3	Эксперимент по сохранению Web-страниц	12	2,36
4	Эксперимент по обмену мгновенными сообщениями	1	0,43

Данные таблицы 2.1.30 необходимы для проведения дальнейших расчетов показателей живучести модели информационной системы радиальной топологии.

### *Расчет живучести сегментов сети радиальной топологии*

Проведем расчет показателей живучести сегментов предложенной сети, имеющих радиальную топологию. Для этого понадобятся значения, полученные в ходе экспериментальных исследований. На рисунке 2.1.40 выделены участки сети, для которых будет производиться расчет.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

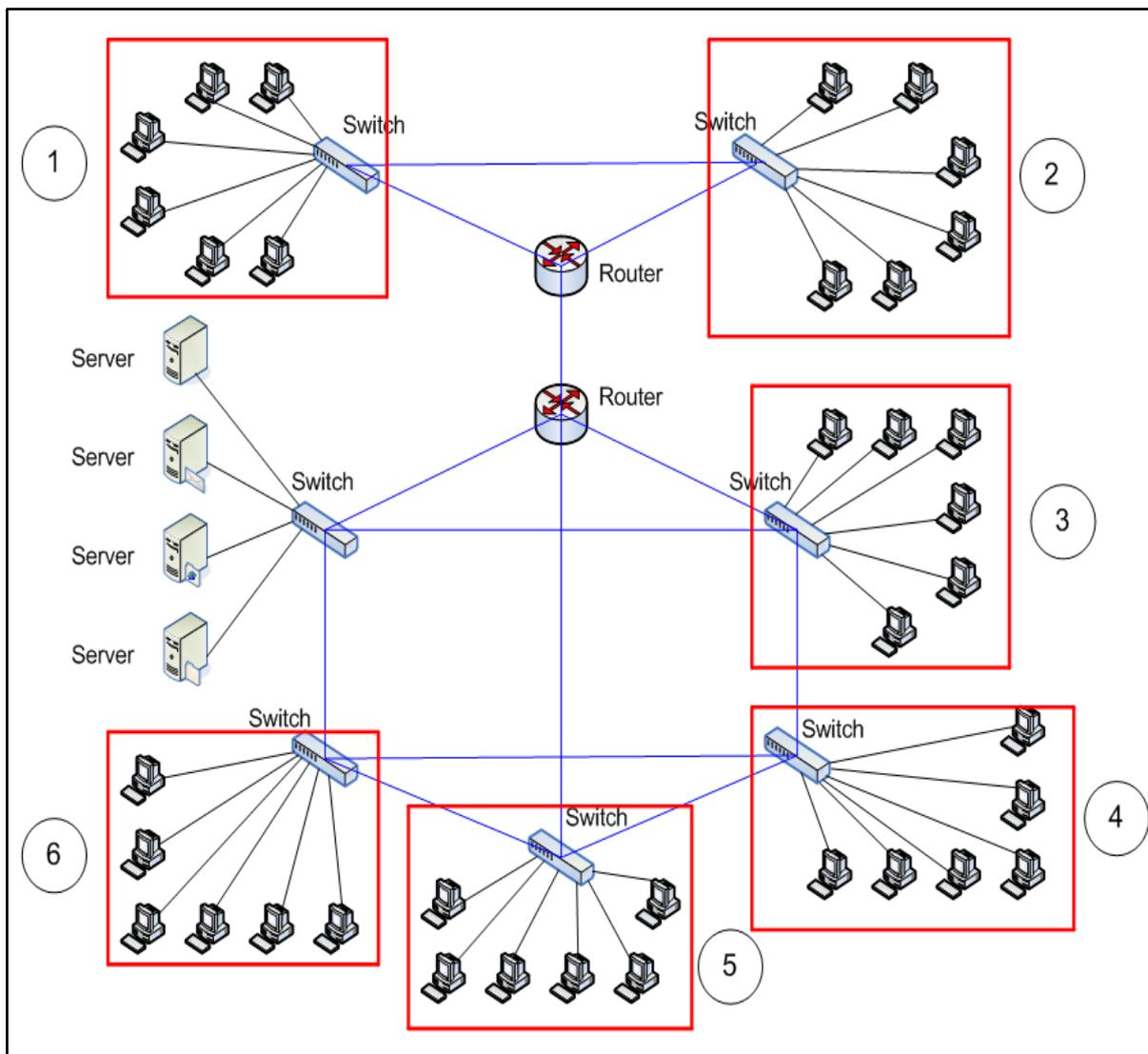


Рисунок 2.1.40 – Сегменты сети с радиальной топологией

Согласно модели оценки живучести информационной сети с радиальной топологией, расчет производится отдельно для функциональных элементов (транзитных узлов) и отдельно для коммуникационных каналов согласно формулам, представленным ниже.

Безотказная работа транзитного узла  $r$ -го варианта в течение времени  $T$  определяется формулой

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

$$P_r(T) = \sum_{k=0}^{q_r} C_{n_r} P^k (1 - P)^{n_r - k},$$

где  $p = 1 - e^{-\lambda_r t}$  ;

$q_r$  – максимально допустимое число отказавших элементов транзитного узла  $r$ -го типа;

$n_r$  – число элементов в момент времени  $t = 0$ ;

$\lambda_r$  – интенсивность отказов компонентов транзитного узла  $r$ -го варианта;

$P_r(T)$  – показатель безотказной работы транзитного узла,

$d_i$  – директивный срок выполнения задачи,

$C_{n_r}$  – нормировочный коэффициент.

Безотказная работа коммуникационных каналов после воздействия неблагоприятных факторов (живучесть коммуникационных каналов) определяется с помощью выражения:

$$P_{ж}^* = \frac{\mu_y(\mu_y n k - 2\pi\lambda)}{\mu_y(\mu_y n n - 2\pi\lambda) + P_{нс} V_y \pi (2\mu_y n n - \pi\lambda)}$$

где  $n$  и  $k$  – соответственно минимально-допустимые числа исправно работающих центральных управляющих узлов и коммуникационных каналов;

$\mu_y = \frac{W(T)}{W^0}$  – относительная текущая интенсивность обслуживания заявок ( $W^0$  – номинальное значение интенсивности);

$\pi$  – средняя длина маршрута распределенной информационной системы;

$P_{нс}$  – коэффициент поражения распределенной информационной системы от неблагоприятных факторов;

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

$\lambda$  – суммарный объем исходящей за пределы распределенной информационной системы информации;

$V_y = \frac{V(T)}{V^0}$  - относительная интенсивность старения информации.

Поскольку в сетевой (распределенной) информационной системе, предложенной выше, имеют место несколько информационных процессов, при расчете приоритетов узлов каждому из них были написаны задачи, которые должны выполняться. Информационные процессы были распределены следующим образом (Рисунок 2.1.41).

Синие надписи соответствуют следующим процессам и экспериментам:

- FTP – скачивание файлов с FTP-сервера;
- SMTP – сохранение почтовых сообщений с вложениями;
- Web – сохранение Web-страниц;
- IM - общение посредством программы обмена мгновенными сообщениями Jabber.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

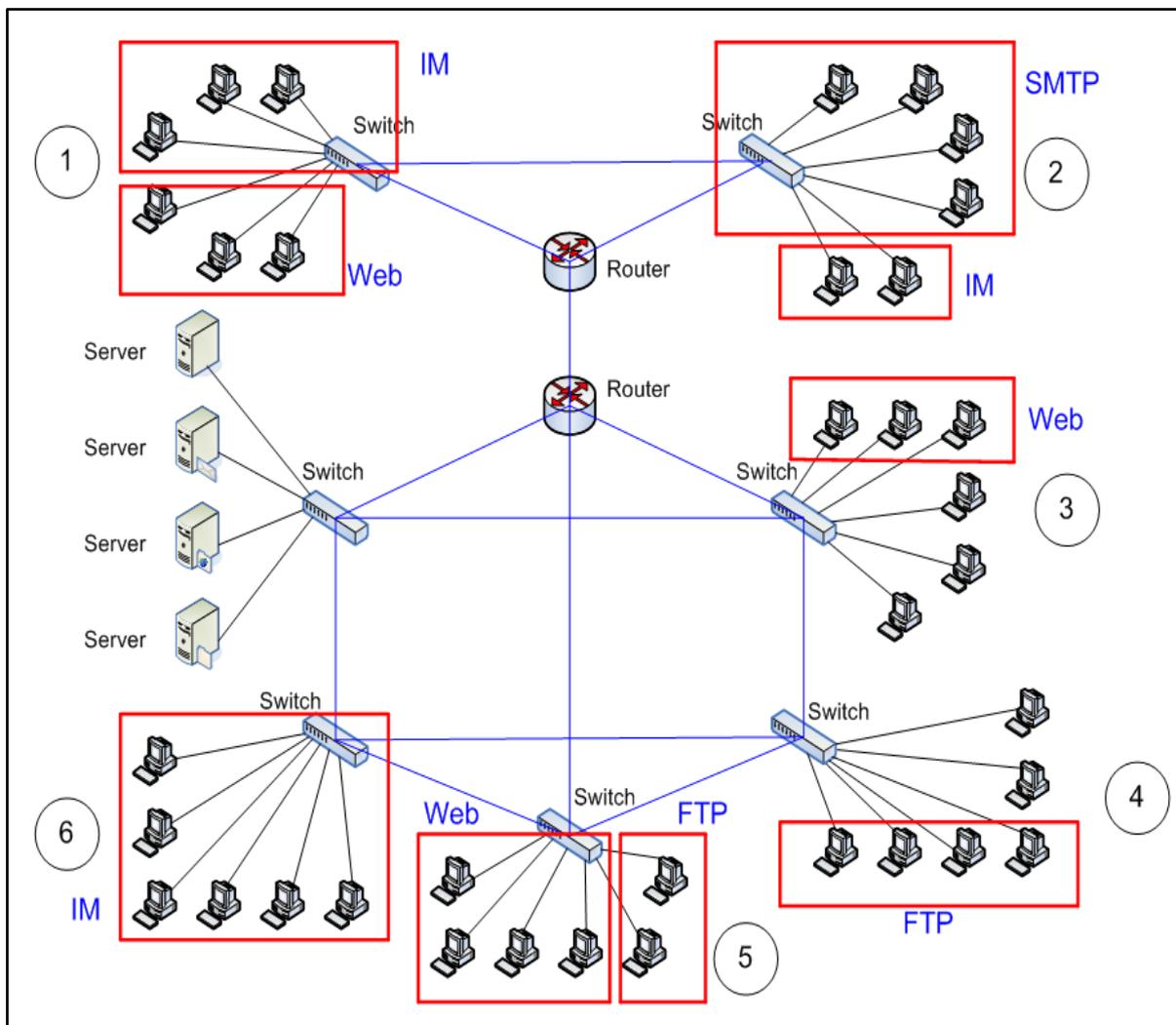


Рисунок 2.1.41 – Разделение сегментов сети по информационным процессам

*Расчет живучести TV.* Расчет производится по формуле для  $P_r(T)$ . Максимальный директивный срок выполнения задачи – это время, которое отводится узлу на выполнение поставленной задачи в нормальных условиях, то есть в нашем случае это скачивание файлов со скоростью, соответствующей средней пропускной способности сети и доставка сообщений за заданное время. Если задача не выполняется за директивный срок – фиксируется отказ.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Максимально допустимое число отказавших узлов  $q$  бралось исходя из того, чтобы не менее двух третей узлов сегмента сети находилось в рабочем состоянии. Значение  $k$  определялось аналогичным образом. Интенсивность отказов соответствует значениям, полученным в ходе проведения экспериментов на лабораторной установке. Число элементов  $n$  на момент времени  $t = 0$  сегмента сети определялось как количество узлов выполняющих одинаковую задачу (учувствуют в одном информационном процессе).

Расчет выполнялся в программе Excel пакета Microsoft Office (Рисунок 2.1.42).

	A	B	C	D	E	F	G	H	I
1									
2		<b>q =</b>	0		<b>p =</b>	0,564			
3		<b>n =</b>	2						
4		<b>λ =</b>	1,43						
5		<b>k =</b>	2						
6		<b>t =</b>	0,58						
7		<b>c =</b>	2						
8									
9		<b>P(T) =</b>	0,6355						
10									
11		<b>Pr(T) =</b>	0,0000						
12									
13									

Рисунок 2.1.42– Расчет показателя живучести узлов  
в программе Excel

Значения переменных, подставляемых при расчете, и результаты приведены в таблице 2.1.31.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Таблица 2.1.31 – Результаты расчетов живучести  
транзитных узлов

Процесс	№ сегмента	Переменные						$P_r(T)$
		q	n	$\lambda$	k	t	C	
<b>FTP</b>	<b>4</b>	1	4	1,43	3	0,58	4	0,24
	<b>5</b>	0	2	1,43	2	0,58	2	0,64
<b>SMTP</b>	<b>2</b>	1	4	1	3	0,58	4	0,19
<b>Web</b>	<b>1</b>	1	3	2,36	2	0,2	3	0,27
	<b>3</b>	0	2	2,36	2	0,2	2	0,28
	<b>5</b>	1	4	2,36	3	0,2	4	0,13
<b>IM</b>	<b>1</b>	1	3	0,43	2	0,016	3	0
	<b>2</b>	0	2	0,43	2	0,016	2	0
	<b>6</b>	2	6	0,43	4	0,016	6	0

### *Расчет живучести коммуникационных каналов*

Расчет производится по формуле для  $P_{ж}^*$ . Уровень воздействия на коммуникационные каналы неблагоприятных факторов определяется коэффициентом  $P_{nc}$ . Данное значение было взято из методики расчета показателя защищенности информационной системы курса «Экономика и защита информации», согласно которому среднее значение коэффициента поражения составляет 0,3.

В каждом сегменте сети есть только одно центральное устройство, поэтому значение минимально-допустимого числа работающих центральных устройств  $n$  равно 1. Минимально-допустимое число исправно работающих коммуникационных каналов  $k$  бралось исходя из того, чтобы не менее двух третей каналов сегмента сети находилось в рабочем состоянии.

Значение относительной интенсивности обслуживания заявок определялось по графику (Рисунок 2.1.43), отражающему плавное

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

уменьшение значения со временем и описываемому следующей аналитической формой:

$$y = a^{-x}, a > 1$$

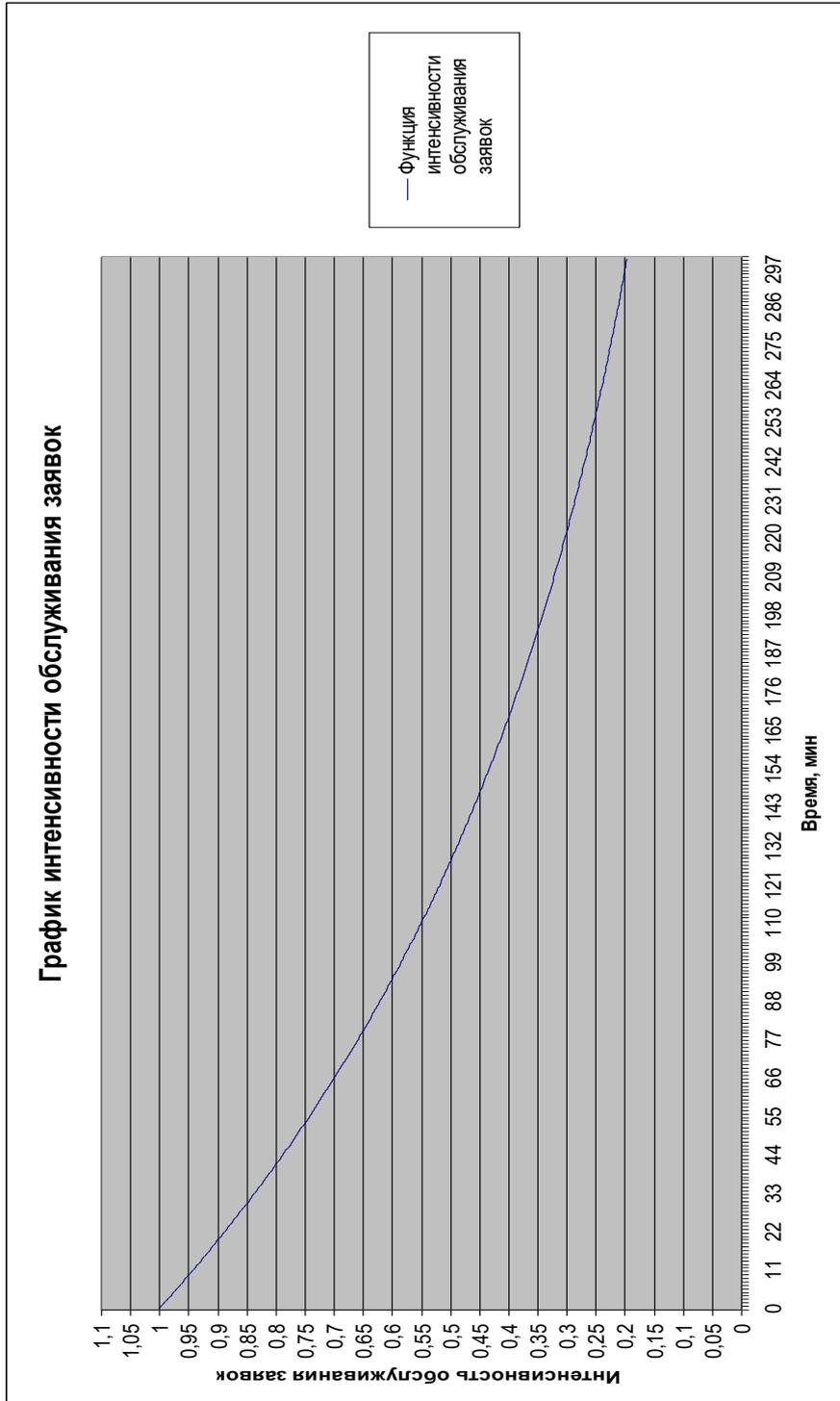


Рисунок 2.1.43– График интенсивности обслуживания заявок

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

В нашем случае, время проведения экспериментов 1 час соответствует интенсивности обслуживания заявок равной 0,72.

Средняя длина маршрута  $\pi$  соответствует среднему числу ребер графа информационной системы, которые учувствуют в соответствующих информационных процессах.

Поскольку в рассматриваемой распределенной информационной системе отсутствует выход за ее пределы, то суммарный объем циркулирующего трафика определялся на самом загруженном участке сети, а именно узле 17. Именно через него идут все запросы к серверам. Замеры проводились с помощью протокола NetFlow, разработанного компанией Cisco Systems для учёта сетевого трафика. За все время экспериментов трафик составил около 2900 Мб.

Параметр  $V$  в расчетах не учитывается, поскольку за такое короткое время проведения экспериментов информация не успевает устаревать ( $V = 1$ ).

Расчет выполнялся в программе Excel пакета Microsoft Office (Рисунок 2.1.44).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

	A	B
1		
2	<b>n =</b>	1
3	<b>k =</b>	2
4	<b>μ =</b>	0,72
5	<b>π =</b>	3
6	<b>pnc =</b>	0,3
7	<b>λ =</b>	2905
8	<b>v =</b>	1
9		
10		
11	<b>Pж(T) =</b>	0,6154
12		
13		

Рисунок 2.1.44 – Расчет показателя живучести коммуникационных каналов в программе Excel

Значения переменных, подставляемых при расчете, и результаты приведены в таблице 2.1.32.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Таблица 2.1.32– Результаты расчета живучести  
коммуникационных каналов

Процесс	№ сегмента	Переменные							Рж(Т)
		n	k	$\mu$	$\pi$	$\rho_{nc}$	$\lambda$	V	
<b>FTP</b>	<b>4</b>	1	3	0,72	3	0,3	2900	1	0,62
	<b>5</b>	1	2		3				0,62
<b>SMTP</b>	<b>2</b>	1	3		3				0,62
<b>Web</b>	<b>1</b>	1	2		3				0,62
	<b>3</b>	0	2		3				0,62
	<b>5</b>	1	4		3				0,62
<b>IM</b>	<b>1</b>	1	2		3				0,62
	<b>2</b>	0	2		3				0,62
	<b>6</b>	2	6		3				0,62

Как видно по результатам расчетов, значения живучести узлов получилось низкие. Это можно объяснить высокой интенсивностью отказов, полученной в результате экспериментов. В свою очередь, это можно объяснить особенностью лабораторной установки. Сеть эмулировалась на двух реальных компьютерах, которые в процессе проведения экспериментов были сильно загружены, что вызывало увеличение числа отказов и как следствие увеличение их интенсивности. Живучесть коммуникационных каналов сильно зависит от коэффициента поражения информационной системы после воздействия неблагоприятных факторов.

### *Расчет живучести кольцевых структур*

Модель живучести информационной сети с радиальной топологией учитывает как структурную, так и функциональную живучесть. Однако модель расчета показателей живучести кольцевых сетевых структур ориентирована на ее структурную составляющую. Применим эту модель к предложенной информационной системе.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

В нашем случае имеем информационную систему, содержащую в себе два кольца (Рисунок 2.1.45).

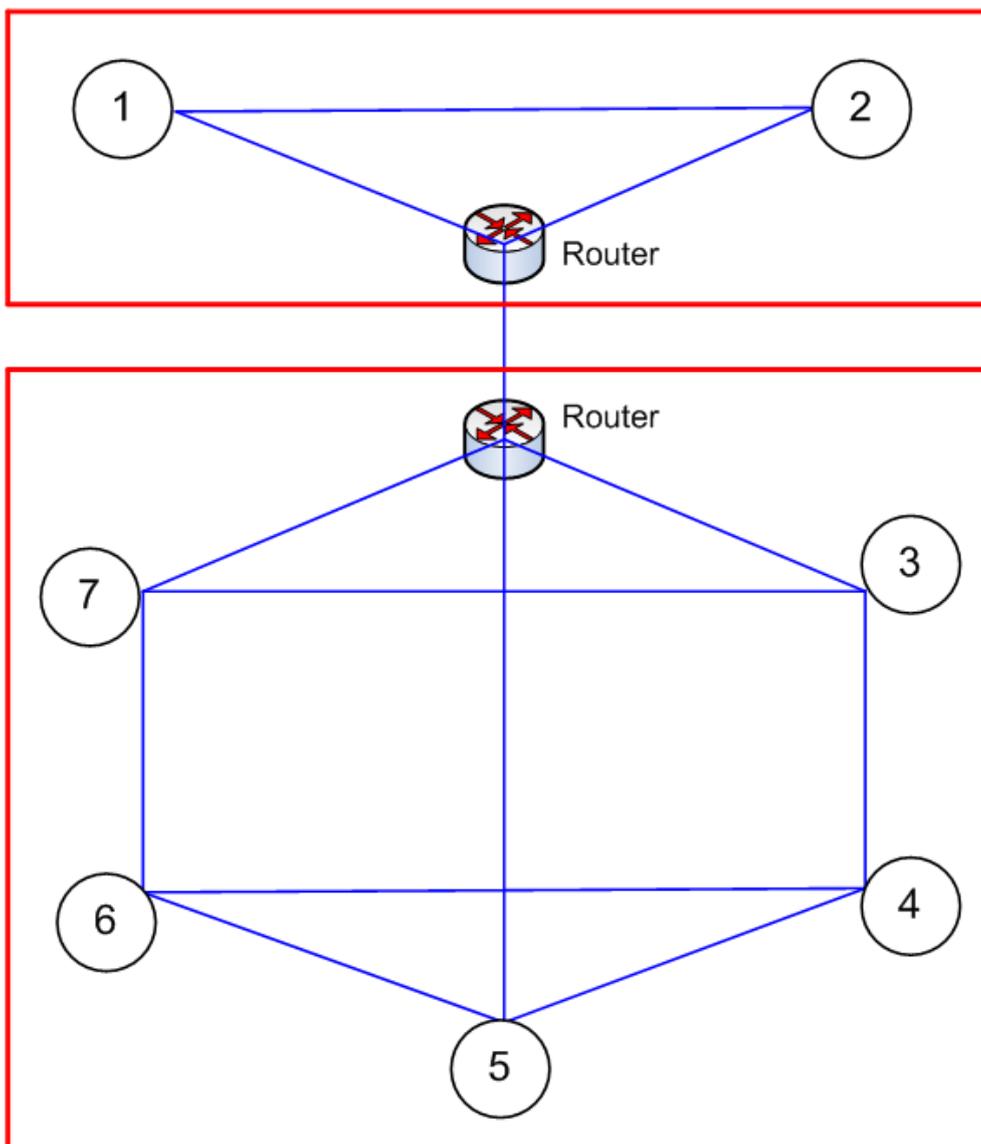


Рисунок 2.1.45 – Сегменты сети с кольцевой топологией

Верхнее кольцо имеет степень вершин  $d=2$ . Живучесть рассчитывается по формуле:

$$p_c = p^3 + 3p^2q,$$

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

где  $p$  - вероятность исправного функционирования всех ребер,  
 $q$  – вероятность неисправного функционирования ребер.

Нижнее кольцо типа «шестиугольник» имеет степень вершин  $d=3$ , то есть имеет дополнительные каналы связи.

Согласно модели формула для расчета живучести имеет следующий вид:

$$p_c = p^9 + C_9^1 p^8 q + C_9^2 p^7 q^2 + [(C_9^3 - (n+1))] p^6 q^3 + [C_9^4 - (n-1)C_6^1] p^5 q^4;$$

где  $n$  – число вершин,

$p$  - вероятность исправного функционирования всех ребер,

$q$  – вероятность неисправного функционирования ребер.

Поскольку кольцо имеет 6 вершин, формула примет вид:

$$p_c = p^9 + C_9^1 p^8 q + C_9^2 p^7 q^2 + [C_9^3 - 7] p^6 q^3 + [C_9^4 - 5C_6^1] p^5 q^4$$

Для расчета необходимо знать  $p$ . Как правило, в пределах одной корпоративной распределенной информационной системы кабельные системы одинаковые, поэтому было решено взять значение живучести коммуникационных каналов, то есть равным 0,62.

Тогда живучесть верхнего кольца равна:

$$p = 0,62^3 + 3*0,62^2*(1 - 0,62) = 0,676$$

Живучесть нижнего кольца рассчитываем с помощью программы MathCAD (Рисунок 2.1.46).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

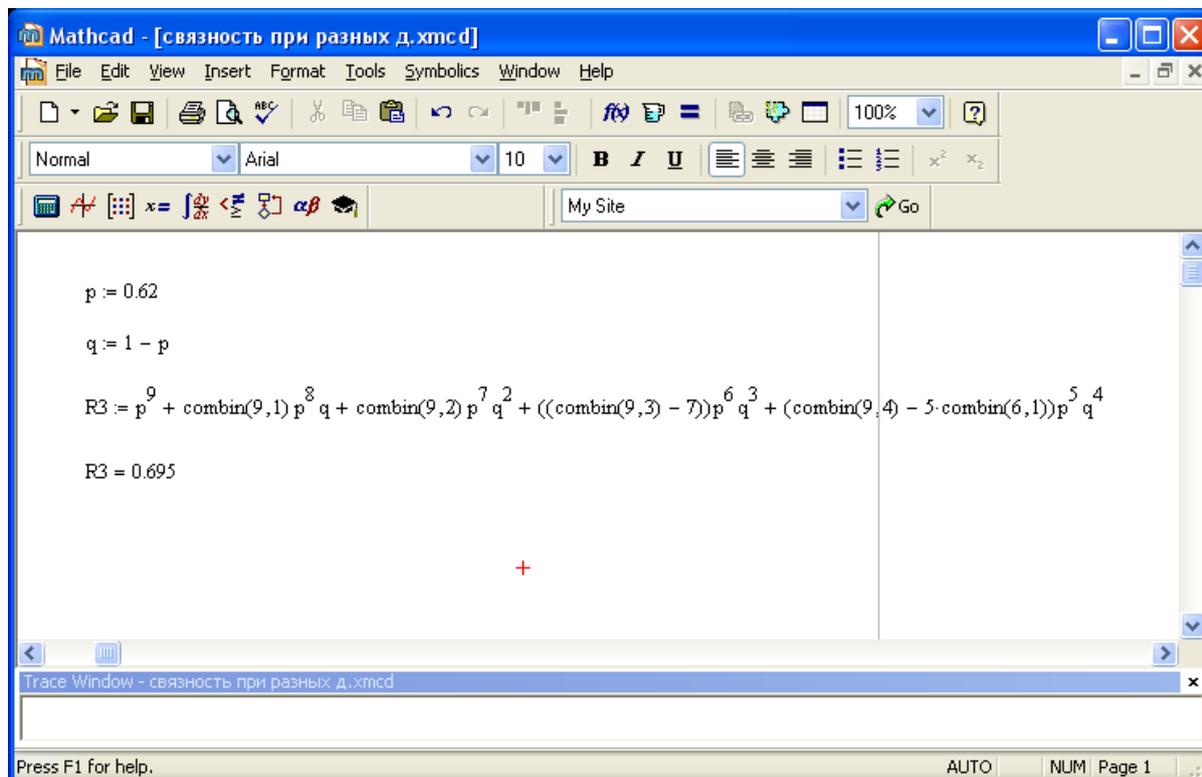


Рисунок 2.1.46 – Расчет показателя живучести кольца в программе MathCAD

По результатам вычисления в программе значение живучести нижнего кольца составила 0,7.

### *Расчет живучести конечной структуры*

После расчета показателей живучести кольцевых структур, последние представляются как один узел, и расчет производится для сети, изображенной на рисунке 2.1.47.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

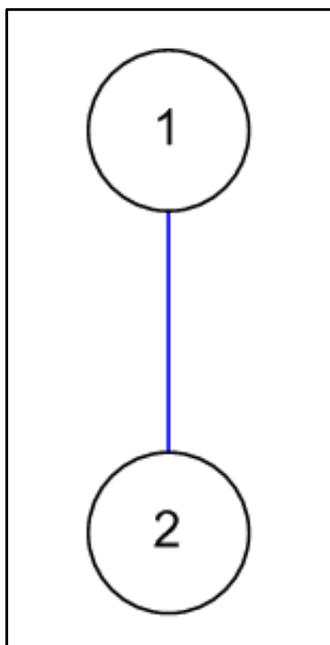


Рисунок 2.1.47 – Конечная структура

Данную топологию можно считать радиальной, если принять точку 2 за центральный узел, поскольку она представляет собой большую часть сети и в ней располагаются все сервера информационной системы. Подставим значения, полученные в ходе проведения экспериментов, в формулу для расчета коммуникационных каналов. Используем для этого программу Excel пакета Microsoft Office (Рисунок 2.1.48).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

	A	B
1		
2	<b>n =</b>	1
3	<b>k =</b>	1
4	<b>μ =</b>	0,72
5	<b>π =</b>	1
6	<b>рпс =</b>	0,3
7	<b>λ =</b>	2900
8	<b>v =</b>	1
9		
10		
11	<b>Pж(T) =</b>	0,8276
12		
13		

Рисунок 2.1.48– Расчет показателя живучести конечной структуры  
в программе Excel

Таким образом, показатель живучести данной топологии составил 0,83. Поскольку рассмотренные модели не предусматривают взаимной интерпретации, то для вычисления общего показателя структурной живучести предложенной распределенной информационной системы было использовано среднее арифметическое значений ранее вычисленных сетевых структур, а именно:

- Живучесть коммуникационных каналов радиальных структур – 0,62;
- Живучесть кольцевых топологий – 0,68 (верхнее кольцо) и 0,7 (нижнее кольцо);

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

- Живучесть коммуникационных каналов, соединяющего верхний и нижний сегменты сети – 0,82.

$$P_{\text{ж.общ.стр}} = (0,62 + 0,68 + 0,7 + 0,82) / 4 = 0,7$$

Отличие от значений, полученных в расчетах на основе модели информационных систем радиальной топологии, составляет:

$$100 - (0,62 / 0,7 * 100) = 11,5\%$$

### *Кратко резюмируем результаты*

В результате анализа предметной области были отмечены следующие моменты. Понятие живучести есть смысл формулировать для каждой конкретной информационной системы отдельно, поскольку последние имеют различную сложность, особенности, задачи и требования к их выполнению. При этом в определении должны фигурировать такие ключевые термины как система, угроза, адаптируемость, непрерывность обслуживания и время. Существует множество терминов, характеризующих состояние системы и ее реакцию на внешние воздействия и неблагоприятные факторы, оценивать по которым информационные системы не всегда удобно и возможно. Поэтому должен существовать единый показатель, который будет характеризовать ИС с этих позиций. Представляется целесообразным в качестве такого наиболее содержательного интегрального показателя использовать «живучесть».

Для осуществления практического расчета показателей живучести распределенной информационной системы была разработана сеть, включающая в себя несколько топологий и являющаяся теоретически живучей. Согласно ранее рассмотренным моделям оценки живучести, некоторые переменные, задействованные в формулах, могут быть получены только экспериментальным путем и с помощью дополнительных расчетов. К таким переменным относится интенсивность отказов узлов и директивное время выполнения задачи. Для их определения были проведены экспериментальные исследования на лабораторной

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
установке, виртуализирующей разработанную сеть. Также был выполнен расчет приоритетов узлов, показавший наиболее важные для функционирования системы элементы. По результатам экспериментов были произведены расчеты согласно рассмотренным моделям оценки живучести информационных систем.

Узлы сегментов сети радиальной топологии имеют низкую живучесть. Это можно объяснить высокой интенсивностью отказов, полученной в результате экспериментов. В свою очередь, это можно объяснить особенностью лабораторной установки. Сеть эмулировалась на двух реальных компьютерах, которые в процессе проведения экспериментов были сильно загружены, что вызывало увеличение числа отказов и как следствие увеличение их интенсивности. Живучесть коммуникационных каналов по всем сегментам сети радиальной топологии составила 0,62, что при довольно высоком значении коэффициента поражения ИС от неблагоприятных факторов является нормальным. Таким образом, данную модель можно применять при расчетах показателей живучести, однако нужны адекватные значения интенсивности отказов, желательно полученные в ходе экспериментов на реальной сети.

Модель расчета показателей живучести кольцевых сетевых структур ориентирована на ее структурную составляющую. Применив эту модель к предложенной информационной системе, получили значения показателей живучести верхнего и нижнего кольца, равные 0,68 и 0,7. Для расчета живучести коммуникационного канала, соединяющего верхний и нижний сегменты сети, снова использовалась модель живучести радиальной топологии (структурная составляющая). В результате было получено значение равное 0,82, что значительно больше, чем в первоначальных сегментах сети с аналогичной топологией. Это объясняется тем, что при расчетах средняя длина маршрута конечно структуры в 3 раза меньше чем в первоначальной.

Конечное значение общей живучести рассчитать не представляется возможным, поскольку модели не предусматривают взаимной интерпретации функциональных и структурных составляющих. Для грубой оценки общей структурной живучести было использовано среднее

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

арифметическое значений всех вычисленных сетевых структур. Показатель составил 0,7, что на фоне рассчитанных значений по моделям является усредненным значением и может считаться адекватным.

Для точной и качественной оценки общего показателя живучести распределенных информационных систем нужна модель, сочетающая в себе возможность расчета как структурных, так и функциональных составляющих живучести в различных сочетаниях сетевых топологий. На сегодняшний день подобных моделей не существует.

### Вопросы и задания

1. Что такое информационная система?
2. Какие критерии оценки живучести корпоративной распределенной информационной системы выделены в тексте?
3. Дайте определение понятию «живучесть информационной системы».
4. Охарактеризуйте модель оценки живучести распределенной информационной системы с радиальной топологией.
5. Приведите примеры расчета живучести сетевых структур.
6. Какова особенность экспериментальной установки, собранной на основе продукта GNS3?
7. Какие можно выделить виды живучести?
8. Какие процессы могут происходить в информационной системе?
9. Самостоятельно проведите эксперимент по скачиванию файлов с FTP-сервера.
10. В чем особенность эксперимента по скачиванию почтовых сообщений с почтового сервера?
11. Как называется совокупность непустого множества вершин и множества пар вершин?
12. Чем отличается распределенная информационная система от корпоративной?

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

### 1. Список рекомендуемых источников

2. Бекасов, В.Ю. Аспекты анализа структуры корпоративных мультисервисных сетей [Текст] / В.Ю.Бекасов; - СПб.: Питер, 2004. -208 с.
3. Бондаренко, А.Д. Методы и средства разработки интеллектуальных систем управления корпоративными компьютерными сетями [текст] : дис. ... канд. технич. наук: 05.13.13 : защищена 2007 г. / Бондаренко Алексей Дмитриевич. – Москва, 2007.
4. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации: Учебник для вузов. 2-е изд. - СПб.: Питер, 2006 - 703 с.
5. Вентцель, Е. С. Теория вероятностей и ее инженерные приложения / Е. С. Вентцель, Л. А. Овчаров. – М: Высш. шк., 2000. – 480 с.
6. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. - М.: Техносфера, 2003. - 512 с.
7. Воробиенко, П. П. Обобщенная информационная модель взаимодействия систем инфокоммуникаций [Текст] / П. П. Воробиенко, М. И. Стукало // Электросвязь. - 2004. - N 6. - С. 24-26.
8. Горшков В.В. Логико-вероятностный метод расчета живучести сложных систем. – АН УкрССР, Кибернетика, 1982, №1. – С.104 – 107.
9. Грибунин В.Г. Разработка и реализация политики безопасности предприятия [Электронный ресурс]. – Режим доступа: <http://bre.ru/security/22754.html>.
10. Громов Ю. Ю. Информационная система оценка живучести сетевых структур. / Ю. Ю. Громов, Д. Е. Винокуров, О. Г. Ива-

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- нова, А. В. Лагутин. – Информационные системы и процессы: сб. науч. тр. / под ред. В. М. Тютюнника. – Тамбов; М.; Баку; Вена: Изд-во «Нобелистика», 2006. Вып.4. – С. 34-40.
11. Громов, Ю. Ю. Анализ живучести информационных сетей / Ю.Ю. Громов, Д.Е. Винокуров, Т.Г. Самхарадзе, И.И. Пасечников // Информационные процессы и управление, №1 – 2006. – С. 138-155.
  12. Громов, Ю. Ю. Синтез и анализ живучести сетевых систем: монография / Ю.Ю. Громов, В.О. Драчев, К.А. Набатов, О.Г. Иванова. – М. : «Издательство Машиностроение-1», 2007. – 152 с.
  13. Груздева, Л.М. Модели повышения производительности корпоративных телекоммуникационных сетей в условиях воздействия угроз информационной безопасности [текст] : дис. ... канд. техн. наук: 05.12.13 : защищена 2011 г. / Груздева Людмила Михайловна. – Владимир, 2011.
  14. Додонов А. Г. Проблемы и тенденции создания живучих вычислительных систем: Метод. разработки / А. Г. Додонов, М. Г. Кузнецова. – К.: Наук. думка, 1981.
  15. Додонов, А. Г. Живучесть информационных систем. / А.Г. Додонов, Д.В. Ладнэ // К.: Наукова Думка, 2011. – 256 с.
  16. Додонов, А. Г. К вопросу живучести корпоративных информационных систем. /А.Г. Додонов, Д.В. Флейтман // Реєстрація, зберігання і обробка даних. Т. 6, № 2, 2004, - С. 33-41.
  17. Домарев В.В. Безопасность информационных технологий. Методология создания систем защиты . – К.:ООО «ТИД «ДС», 2001. – 688 с.
  18. Домнин, Л. Н. Элементы теории графов: учеб. пособие / Л. Н. Домнин. – Пенза: Изд-во Пенз. гос. ун-та, 2007. – 144с.
  19. Дэвис, Д., Барбер, Д., Прайс, У. Вычислительные сети и сетевые протоколы [Текст] / Д. Дэвис и др.; - Москва: Мир, 1982. – 214 с.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- =====
20. Елисеев А. И., Долгов А. А., Хорохорин М. А., Лаута О. С., Набатов К. А. Обеспечение живучести информационных систем (часть 3. Методы обеспечения и повышения живучести) // Вестник Воронежского института ФСИИ России, 2013, № 1. – С.91-94
  21. Елисеев А. И., Долгов А. А., Хорохорин М. А., Набатов К. А. Структура характеристик живучести информационных систем // Вестник Воронежского института ФСИИ России, 2013, № 1.- С.77-82.
  22. Коробейников А.Г., Тронилов И.Б., Жаринов И.О. Методы и модели оценки инфраструктуры системы защиты информации в корпоративных сетях промышленных предприятий: монография / Под ред. П.П. Парамонова. СПб: Изд-во ООО «Студия «НП-Принт», 2012. — 115 с.
  23. Крапивин В.Ф. О теории живучести сложных систем. // - М.: Сов. радио. 1978. - 235 с.
  24. Крупнов Л.С. Математические модели, комплексы программ и алгоритмы принятия решения при возникновении конфликтного взаимодействия компьютерных систем [текст] : дис. ... канд. технич. наук: 05. 13.18: защищена 2016 г. / Крупнов Леонид Сергеевич.- Рязань, 2016.
  25. Кульгин, М.В. Технологии корпоративных сетей [Текст]: энциклопедия / М.В. Кульгин; - СПб.: Питер, 1999. - 704с.
  26. Лысков, О.Э. Автоматизация поддержки процесса обеспечения работоспособности вычислительной сети предприятия [текст] : дис. ... канд. технич. наук: 05.13.06 : защищена 2008 г. / Лысков Олег Эдуардович. – Орел, 2008.
  27. Михайлов, А.В. Модели и алгоритмы повышения живучести распределенных информационно-вычислительных систем АСУП [текст] : дис. ... канд. технич. наук: 05.13.06 : защищена 2007 г. / Михайлов Андрей Витальевич. – Владимир, 2007.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- =====
28. Мишин, Д.В. Модели и алгоритмы административного управления корпоративной распределенной информационно-вычислительной средой АСУ [Текст]: диссертация кандидата технических наук: 05.13.06 / Д.В. Мишин [Место защиты: Владимир. гос. ун-т] Владимир, 2013. - 218 с.
  29. Монахов, М.Ю., Илларионов, Ю.А. Безопасное управление ресурсами в распределенных информационных и телекоммуникационных системах [Текст]: монография / М.Ю. Монахов и др.; Владимирский гос. ун-т. - Владимир, 2004. - 212 с.
  30. Монахов, Ю.М. Модели обнаружения аномального функционирования информационно-вычислительной среды интегрированных АСУ [текст]: дис. ... канд. технич. наук: 05.13.06: защищена 2009 г. / Монахов Юрий Михайлович. – Владимир, 2009.
  31. Монахов, Ю.М. Уязвимости протокола транспортного уровня ТСР [Текст] / Ю.М. Монахов; Алгоритмы, методы и системы обработки данных. Сборник научных статей. - М.: Горячая линия-Телеком, 2006. - с. 203-210.
  32. Монахов, Ю.М., Груздева, Л.М, Монахов, М.Ю. Вредоносные программы в компьютерных сетях [Текст]: учеб.пособие (с грифом УМО) / Ю.М. Монахов и др.; Владим. Гос.ун-т. – Владимир: Изд-во Владим. гос. ун-та, 2010. – 76 с.
  33. Монахова, М.М. Модели и алгоритмы контроля инцидентов информационной безопасности в корпоративной телекоммуникационной сети [текст] : дис. ... канд. технич. наук: 05.12.13 : защищена 2016 г. / Монахова Мария Михайловна. – Владимир, 2016.
  34. Мур М. и др. Телекоммуникации. / Авторы: Мур М., Притск Т., Риггс К., Сауфвик П. - СПб.: БХВ - Петербург, 2005. - 624 с.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

35. Орлов А.И. Организационно-экономическое моделирование: учебник в 3 ч. / А.И. Орлов. – М.: Изд-во МГТУ им. Н.Э. Баумана. – 2009. Ч. 2 : Экспертные оценки. – 2011. – 486 с.
36. Поликарпов А.К. Обзор существующих методов оценки рисков и управления информационной безопасностью – [Электронный ресурс]. – Режим доступа: <http://is.isa.ru/PolikOtc.html>
37. Пескова С.А., Кузин А.В., Волков А.Н. Сети и телекоммуникации. М.: Академия, 2008. - 576 с.
38. Погонин В.А., Схиртладзе А.Г. Интегрированные системы проектирования и управления. Корпоративные информационные системы: Учеб. пособие. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2006. - 144 с.
39. Рябинин И.А., Черкесов Г.Н. Логико-вероятностные методы исследования надежности структурно-сложных систем — М.: Радио и связь, 1984.— 238 с.
40. Стекольников, Ю. И. Живучесть систем. – С.-П.: Политехника, 2002. – 155 с.
41. Сурков, Е.В. Модель живучести как работоспособности корпоративной телекоммуникационной сети / Е.В. Сурков, М.Ю. Монахов // Динамика сложных систем — XXI век. – 2019. - №3. – С.21-26.
42. Сурков, Е.В. Экспериментальная проверка результатов по оценке живучести корпоративной телекоммуникационной сети / Е.В. Сурков, Ю.М. Монахов, А.П. Кузнецова // Проектирование и технология электронных средств. – 2019. - №3. - С.46-51.
43. Сурков, Е.В. Кибернетические аспекты управления рисками информационной безопасности в компьютерных сетях образовательных организаций / Е.В. Сурков, Е.Н. Надеждин // Современные наукоемкие технологии. - 2016. – С. 279-285.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

44. Сурков, Е.В. Игровая модель для анализа характеристик системы активной защиты сетевых ресурсов / Е.В. Сурков, Е.Н. Надеждин, Т.Л. Шершакова // *Фундаментальные исследования*. - 2017. – С. 303-311.
45. Сурков, Е.В. Модель ситуационного управления рисками сетевой безопасности / Е.В. Сурков, Е.Н. Надеждин // *Научный альманах*. – 2015. - С. 187-191.
46. Сурков, Е.В. Элементы теории обманных систем в задачах идентификации информационных угроз / Е.В. Сурков, Е.Н. Надеждин // *Сборник трудов Международной НПК «Инновационные направления в науке, технике, образовании»*. – Смоленск. – 2016. - С. 90 – 92.
47. Сурков, Е.В. Влияние вирусов-шифровальщиков на живучесть информационных систем образовательных учреждений / Е.В. Сурков, А.В. Ермошин // *Сборник трудов НПК «Информационная среда вуза»*. – Иваново. - Ивановский гос. политехн. ун-т. - 2017. - С. 34-38.
48. Сурков Е.В. «Усовершенствованная вероятностная модель Клементса-Хоффмана для оценки непоражаемости компонентов корпоративной телекоммуникационной сети» / *Сборник трудов IV Международной НПК «Проблемы информационной безопасности»*. – Гурзуф. - 2017. – С.92-95.
49. Сурков Е.В. Программа для расчета вероятности защищенности компонентов корпоративных телекоммуникационных сетей / Е.В. Сурков, Г.Е. Монахова // *Владимирский государственный университет // Свидетельство о регистрации программы для ЭВМ, № 2019619302 от 15.07.2019.*
50. Таненбаум Э. *Компьютерные сети*. СПб: "Питер", 2008. - 992 с.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

51. Черкесов Г.Н. Методы и модели оценки живучести сложных систем [Электронный ресурс]. – Режим доступа: <http://www.gcherkesov.com/articles/article02.pdf>.
52. Berk, V. H., Gray, R.S., Bakos, G. Using sensor networks and data fusion for early detection of active worms [Text] / V.H. Berk and others; SPIE-The International Society for Optical Engineering, 2003. - Volume 5071. - p. 92-104.
53. Blazek, R.B., Novel, A Approach to Detection of «Denial-of-Service» Attacks via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods [Text] / R.B. Blazek and others; - IEEE CS Press, 2001, p. 220–226.
54. Carl, G., Kesidis, G., Brooks, R. R., Rai, S. [Text] / G. Carl and others; Denial-of-Service Attack-Detection Techniques. IEEE Internet Computing, 2006. - vol. 10, № 1. - p. 82-89.
55. D.V. Mishin, M.M. Monakhova About the optimization of the administration corporate area networks of the data transmission under scarce administrative resources // Herald of the National Technical University "KhPI". Subject issue: Information Science and Modelling. - Kharkov: NTU "KhPI". - 2011. - №17. - P. 101 - 108.
56. Dorogovtsev S., Mendes J. Evolution of networks. Advances in Physics, 51: 2002. pp. 1079-1187.
57. Ellison R. J. Survivability: Protecting Your Critical Systems / R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. A. Longstaff, N. R. Mead. – <http://www.cert.org/archive/html/protect-critical-systems.html>
58. Enisa Annual Incident Report. - [Электронный ресурс]. – Режим доступа: <https://www.enisa.europa.eu/activities/Resilience-and-CIP/Incidents-reporting/annual-reports/annual-incident-reports-2013>.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- =====
59. Fluid Models and Solutions for Large-Scale IP Networks / Yong Liu, Francesco Lo Presti, Vishal Misra [и др.] // In Proc. of ACM SIGMETRICS. 2003. С. 91–101.
  60. Feinstein, L. Statistical Approaches to DDoS Attack Detection and Re-sponse [Text] / L. Feinstein; - IEEE CS Press, 2003. - vol. 1. - p. 303–314.
  61. Simulation of large scale networks using SSF / D.M. Nicol, J. Liu, M. Liljenstam [и др.] // Simulation Conference, 2003. Proceedings of the 2003 Winter. Vol.1. 2003. Dec. С. 650–657.
  62. IT-Baseline Protection Manual : [Электронный ресурс]. – Режим доступа: <http://www.iss.net>.
  63. Kaspersky Anti-Virus. [Электронный ресурс]. – Режим доступа: <http://www.kaspersky.ru/antivirus>.
  64. Liu Xin, Chien Andrew A. Realistic Large-Scale Online Network Simulation. 2004.
  65. Mao Yun, Saul Lawrence K. Modeling Distances in Large-scale Networks by Matrix Factorization // Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. IMC' 04. NY, USA: ACM, 2004. С. 278–287. URL: <http://doi.acm.org/10.1145/1028788.1028827>.
  66. Linger R. C. Requirements Definition for Survivable Network Systems / R. C. Linger, N. R. Mead, H. F. Lipson. – <http://www.cert.org/archive/pdf7iere.pdf>
  67. Mogul J.C. IP Network Performance. Internet System Handbook. Addison-Wesley, Reading, Mass. - pp. 575-675. NIST SP 800-61 Computer security incident handling guide.
  68. OpenVPN. [Электронный ресурс]. – Режим доступа: <https://openvpn.net/>.
  69. Ping Of Death. [Электронный ресурс]. – Режим доступа: <http://insecure.org/sploits/ping-o-death.html>

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

70. SecretNet. [Электронный ресурс]. – Режим доступа: [http://www.securitycode.ru/products/secret\\_net/](http://www.securitycode.ru/products/secret_net/).
71. Shade You VPN. [Электронный ресурс]. – Режим доступа: <https://shadeyouvpn.com/ru/>.
72. Open Web Application Security Project, OWASP Top Ten 2010. – Режим доступа: - <http://www.owasp.org/>.
73. USB-ключи eToken [Электронный ресурс]. – Режим доступа: <http://www.aladdin-rd.ru/catalog/etoken/models>.
74. Simulation of large scale networks using SSF / D.M. Nicol, J. Liu, M. Liljenstam [и др.] // Simulation Conference, 2003. Proceedings of the 2003 winter. Vol.1. 2003. Dec. С. 650–657

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### **2.2. Использование теории графов для моделирования распространения информации в крупномасштабных социальных сетях**

Информация носит нематериальный характер, если не нанесена на материальный носитель, но при этом способна играть важную роль. С развитием информационных технологий, появилась возможность неограниченно размножить информацию.

Дэвид Рид выделил в развитии информационных технологий три этапа, которые различаются принципами распространения информации, а именно широковещательный, транзакционный и групповой. Широковещательный принцип, по которому действуют средства массовой информации предполагает распространение от "одного ко многим". Транзакционный принцип "от одного к одному" начался с обычной почты, продолжился в телефонии и электронной почте. Возможность реализовывать групповой принцип появилась благодаря сети Интернет и последующему развитию средств, позволяющих производить обмен информацией внутри некоторых групп или объединений людей – социальных сетей.

Под социальной сетью понимается множество акторов (точек, вершин, агентов) которые могут вступать во взаимодействие друг с другом.

Исходя из того, что распространение некоторой информации может быть как желательно, так и нежелательно, возникает необходимость организации контроля над распространением информации в социальных сетях.

Поскольку данная научная область сформировалась относительно недавно, отсутствуют не только методы организации такого контроля, но и также фундаментальные знания о социальных сетях крупного масштаба.

Проведение исследования информационных процессов, протекающих в социальных сетях, требует наличия знаний об их статических характеристиках, как объекта исследования.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Однако сбор данных для анализа в объеме, сопоставимом с масштабом крупной социальной сети, такой как сеть города или страны, невозможен прямым изучением.

В отличие от классических социальных групп, социальное сообщество, действующее в Интернет через Интернет-сайты социальных сетей, допускает оперативное изучение, измерение и классификацию, а также, потенциально, и извлечение прибыли за счет организации эффективного распространения некоторой информации в ней.

Примером может служить предвыборная кампания на пост президента США Барака Обамы 2008 года. В ходе кампании Интернет-сайты социальных сетей были успешно использованы, чтобы собрать миллиард долларов взносов в поддержку кандидата с миллионов его сторонников.

Объектом в данной работе является крупномасштабная социальная сеть пользователей Интернет-сайта facebook, обладающая на данный момент наибольшим количеством активных пользователей в мире.

Цель: разработать метод моделирования распространения информации в крупномасштабных социальных сетях.

Для достижения поставленной цели в работе:

- получены статические характеристики доступной для исследования части социальной сети;
- на основании полученных данных сгенерирована недоступная часть социальной сети;
- смоделирован процесс распространения информации в полученной сети;
- разработаны алгоритмы исследования статических характеристик крупномасштабных социальных сетей, генерации недо-

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
ступных для исследования частей крупномасштабных социальных сетей, моделирования распространения информации в крупномасштабной социальной сети.

Разработанные алгоритмы объединены в метод, позволяющий проводить исследования как статических, так и динамических характеристик социальных сетей на основе неполных данных об исследуемой сети.

Применение разработанного метода в сочетании с более точными моделями распространения информации позволит предсказать время, в течение которого распространение информации примет вирусный характер, а также рассчитать скорость роста потенциальных затрат на противодействие распространению нежелательной информации.

Проведение достаточно большого количества экспериментов по распространению информации позволит выявить участки, наиболее подходящие для скорейшего распространения необходимой информации в этой социальной сети, а также получить зависимость скорости распространения от затрат на его организацию.

Далее рассматриваются модель данных, используемая при разработке метода, подготовительные этапы, составляющие основу предлагаемого метода, рассматривается подход к проведению распределенного моделирования распространения информации в крупномасштабных социальных сетях, анализируются результаты экспериментов, проведенных с применением предлагаемого метода.

### **Алгоритмы предварительного анализа топологии социальных сетей**

Предлагаемый метод моделирования распространения информации в социальных сетях представляется совокупностью последовательно применяемых к исследуемой социальной сети алгоритмов:

1. Получение доступной информации о топологии и характеристиках узлов социальной сети
2. Форматирование полученных данных

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

3. Выделение репрезентативного подмножества
4. Получение распределения степеней связности в доступной части сети
5. Получение распределения коэффициента кластеризации в доступной части сети
6. Экстраполяция распределения коэффициента кластеризации на недоступную часть сети
7. Генерация недоступной части сети
8. Форматирование полученных данных
9. Распределенное моделирование

Однако для рассмотрения подходов к алгоритмическому решению названных задач необходимо прежде описать используемую в алгоритмах модель данных.

### *Модель данных*

В основе модели данных лежит сущность «узел», хранящая уникальный идентификатор узла сети. Этот идентификатор обычно присутствует как характеристика члена социальной сети в используемом Интернет-ресурсе.

Однако множество данных идентификаторов следует предварительно отобразить на равное по мощности множество последовательно именованных идентификаторов. Данная операция необходима для упрощения последующего индексирования большого объема имеющихся данных и сокращения времени доступа к ним. Также данная операция позволяет анонимизировать участников социальной сети, в случае, если результаты исследований необходимо опубликовать, явно не раскрывая при этом отображения реальных социальных связей на топологию исследуемой сети.

Сущность узла также хранит флаг, определяющий, является ли данный узел открытым для исследования его положения в топологии исходной сети (далее открытые), или же, что сведения его о положении прямо получены быть не могут (далее закрытые).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

На рисунке 2.2.1 показано, что открытые узлы имеют связи в обоих направлениях, в то время как закрытые – лишь в одном направлении. Таким образом, иллюстрируются вводимые далее определения.

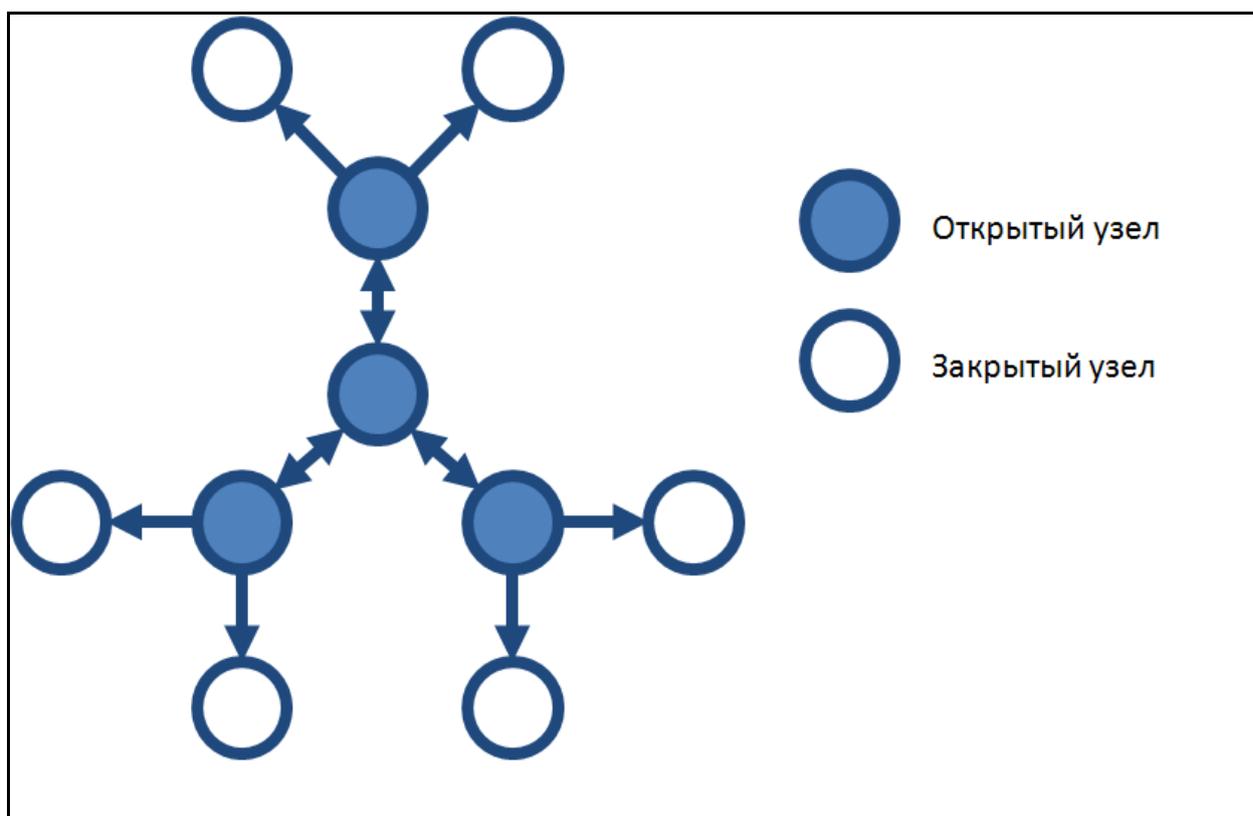


Рисунок 2.2.1 - Представление топологии согласно предлагаемой модели данных

Предположим, что в результате сбора информации о топологии исследуемой социальной сети независимо были получены следующие сведения:

- 1) узел А имеет связь с узлом Б;
- 2) узел Б имеет связь с узлом А.

Тогда, рассматривая узел А, будем называть связь узла А с узлом Б – прямой, а связь узла Б с узлом А – обратной.

Рассматривая узел Б, будем называть связь узла Б с узлом А – прямой, а связь узла А с узлом Б – обратной.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Из рисунка 2.2.1 видно, что закрытые узлы не имеют прямых связей ни с одним узлом, однако имеют обратные связи от некоторых открытых узлов. Открытые же узлы имеют лишь прямые связи с закрытыми узлами.

Для определения этого принципа в рамках предлагаемой модели данных введена сущность «Группа». Выберем некоторый открытый узел социальной сети и назовем его «Родительским». Выберем все узлы, с которыми у родительского есть прямая связь и назовем их «Дочерними». Сущность хранит ссылку на родительский узел, а также на все его дочерние узлы (Рисунок 2.2.2).

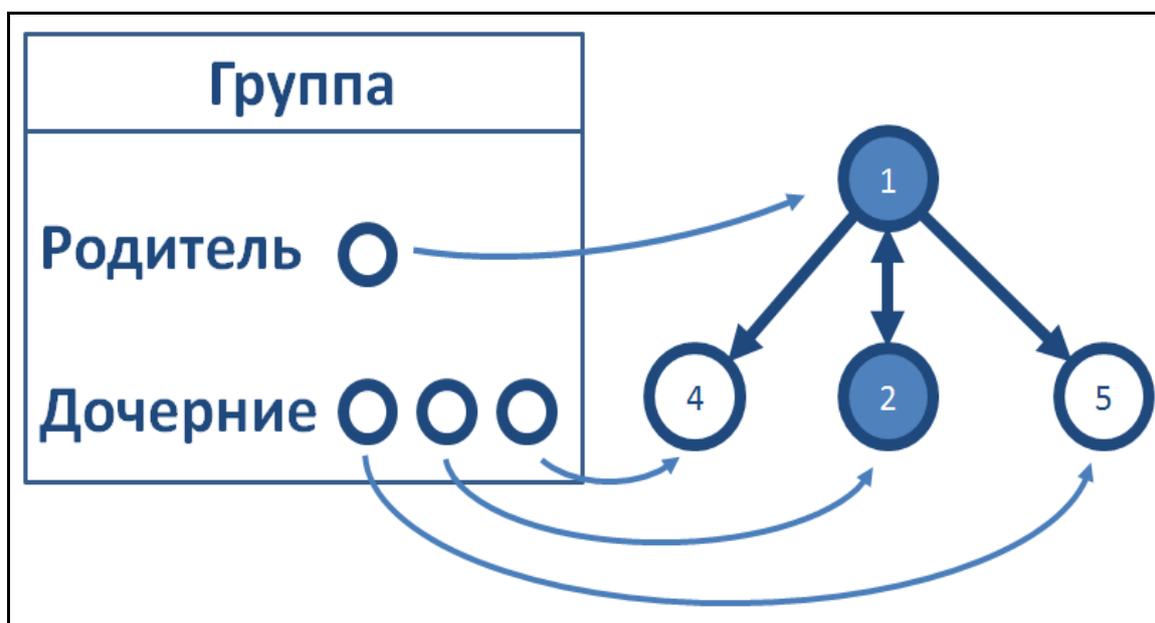


Рисунок 2.2.2 - Сущность «Группа»

Для отображения обратных связей выберем каждый узел, для которого существует прямая связь с родительским и добавим его к ним в список дочерних.

Таким образом, получаем множество групп, однозначно идентифицируемых по соответствующему им родительскому узлу и имеющих перекрестные связи с другими родительскими узлами через соот-

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

ветствующие им множества дочерних узлов. При этом каждому родительскому узлу соответствует ровно одна группа. Множество групп в данном случае представляет собой топологию сети.

Данное множество групп (топология сети) представляется сущностью «Карта». Сущность карта хранит хэши идентификаторов родительских узлов для каждой группы, таким образом, осуществляется индексирование карты, что позволяет быстро получать доступ к интересующей группе, то есть к дочерним узлам некоторого родительского.

Предлагаемая модель данных, описанная при помощи UML представлена на рисунке 2.2.3.

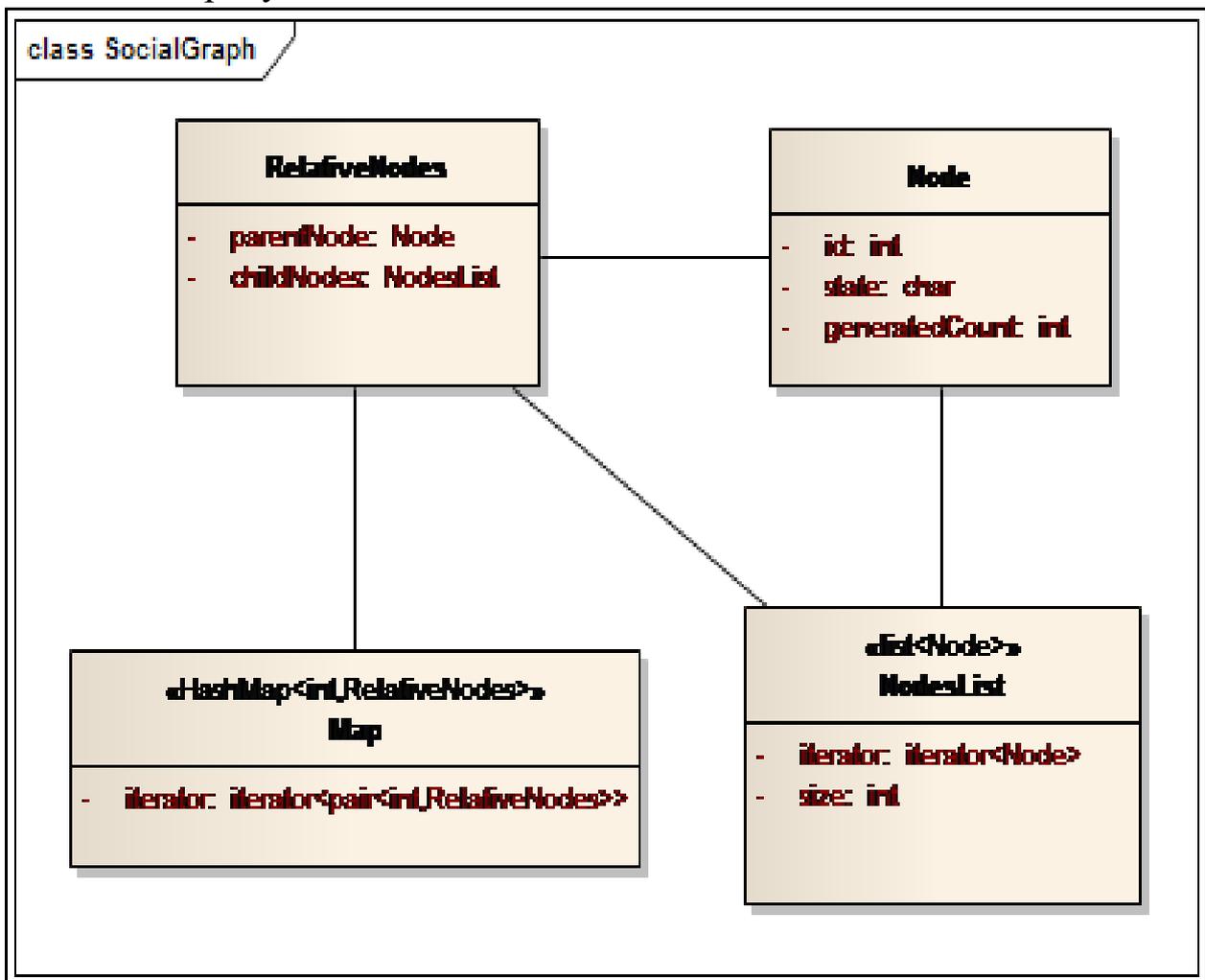


Рисунок 2.2.3 – Модель данных, описанная при помощи UML

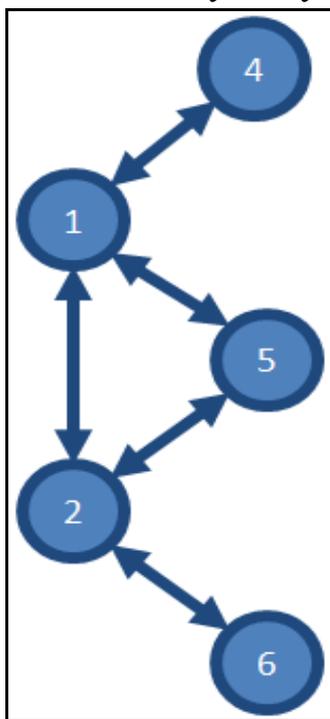
## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

В рамках предлагаемой модели возможно несколько способов представления исходной топологии социальной сети.

Неоднозначность представления является следствием неполной доступности исходной топологии сети, а также допущениями, возможными в рамках конкретных задач.

В данной работе рассматриваются 3 подхода к отображению топологии социальной сети на модель данных. Поскольку данные подходы отличаются степенью дублирования данных о топологии сети, будем называть их типами консистентности.

Под консистентностью понимается состояние модели, при котором для каждого узла сети существует лишь один его прообраз в рамках используемой модели. На рисунке 2.2.4 слева направо изображены введенные типы консистентности в порядке убывания ее величины, соответственно – полная, частичная и отсутствующая консистентность.

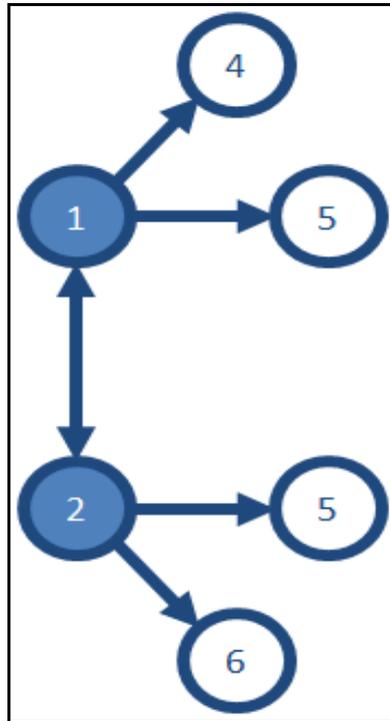


а)

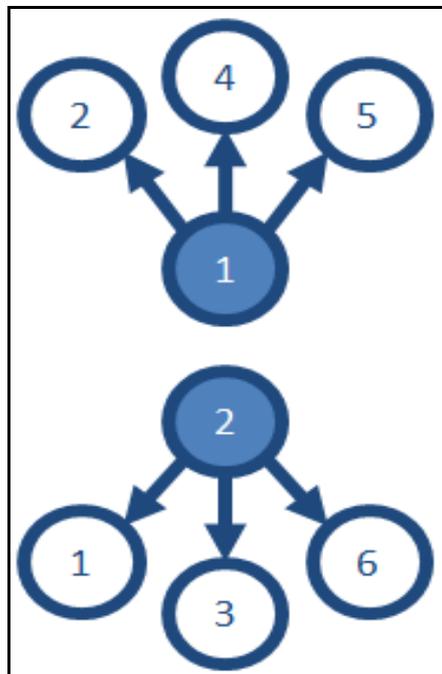
Рисунок 2.2.4 - Полная (а), частичная (б) и отсутствующая (в) консистентность (начало, окончание – на с. 265)

Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ  
В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---



б)



в)

Рисунок 2.2.4 – Окончание (начало см. на с. 264)

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Представление топологии при помощи полной консистентности видится наиболее желаемым, так как дает наибольшую свободу перемещения по карте вследствие наличия дополнительных связей, отсутствующих в других типах. Однако использование такого подхода приводит к заполнению карты преимущественно группами закрытых узлов, фактически не влияющими на результат вычисления некоторых характеристик сети, как следствие, - неоправданному увеличению времени поиска по карте. Данный тип наилучшим образом подходит для вычисления средней длины пути, как характеристики графа, представляющего социальную сеть.

Частичная консистентность обеспечивает консистентность лишь открытых узлов, что способствует снижению мощности множества групп, и вместе с этим позволяет проводить исследование открытой части, например при помощи предлагаемых алгоритмов вычисления и экстраполяции кластерных коэффициентов.

Отсутствующая консистентность может использоваться для последовательного (конвейерного) расчета средней степени связности всей сети как среднего по отдельным подсетям.

### *Получение доступной информации о топологии и характеристиках узлов социальной сети*

Получение данных для анализа характеристик социальных сетей сильно затруднено, поскольку не согласуется с интересами значительной части участников социальной сети, поскольку сбор такого рода информации сопряжен со значительными временными затратами.

Существует возможность косвенного исследования социальных сетей на основании информации, доступной через специализированные Интернет-сайты социальных сетей. Каждому человеку, зарегистрированному на Интернет-сайте социальных сетей, соответствует страница, где он может опубликовать список ссылок, ведущих на страницы людей, с которыми у него имеются социальные связи, а также другую личную информацию.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
Таким образом, множество людей–узлов социальной сети отображается на множество страниц Интернет-сайта социальных сетей, а множество связей между этими людьми – на множество ссылок, опубликованных на каждой странице. Из данных множеств можно восстановить топологию социальной сети, частично повторяющую исходную.

Топология социальной сети представляется графом  $G=\{V,E\}$ , где  $V$  (множество вершин графа) - множество людей, входящий в социальную сеть, а  $E$  (множество ребер графа) – множество связей между людьми.

В данном приближении необходимо ввести условие, что этот граф является неориентированным, то есть, что все связи в отображаемой сети – двунаправленные. Данное условие следует из условия публикации ссылок на страницы Интернет-сайтов социальных сетей – ссылка публикуется, только если обе стороны признают существование связи между ними. Также из предметной области следует, что любые две вершины графа могут быть связаны не более чем одним ребром.

Для упрощения первичных исследований граф считается не взвешенным, т.е. сила социальных связей, как величина, не отображается на вес соответствующих ребер.

Примером описанного выше исследования является *Walking in Facebook*. Целью данного исследования являлось изучение различных алгоритмов сбора информации о топологии социальных сетей, а также экспериментальное измерение эффективности разработанного коллективом авторов алгоритма.

В результате проведения данного исследования были также получены образцы социальной сети facebook, на которой и проводился эксперимент.

В ходе эксперимента были предприняты две независимые попытки собрать сведения о всех доступных узлах сети. Оба результата были предоставлены для апробации разработанного в данной работе

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

метода моделирования распространения информации в социальных сетях и позволили внести предложения по определению формата представления социального графа. Полученные независимые результаты исследования были агрегированы и использовались совместно для получения потенциально более точных сведений об исследуемом классе сетей.

### *Форматирование полученных данных*

В качестве формата представления исходных данных был выбран текстовый файл, в котором каждой строке соответствует группа узлов, как сущность модели данных, описанной в данной работе. Каждому узлу соответствует численный идентификатор в файле. Идентификаторы разделены символами пробела. Первый узел в строке соответствует родительскому узлу группы, последующие - дочерним.

Данный формат предполагает 2 вида представления топологии: с избыточностью и без избыточности (Рисунок 2.2.5).

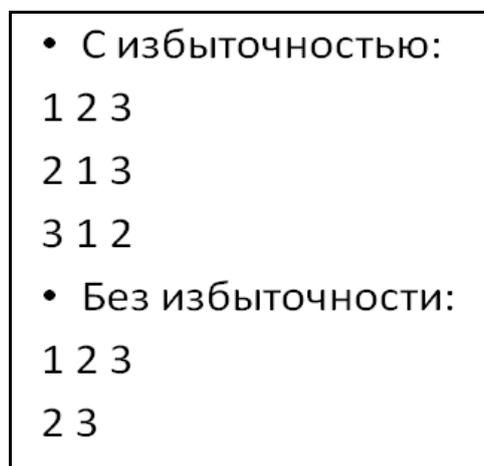


Рисунок 2.2.5 - Представление топологии в используемом формате

В полученных данных множество родительских узлов состоит из 1,653,129 выбранных открытых узлов социальной сети пользователей Интернет-сайта facebook. Эти родительские узлы имеют в сумме

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

94,553,550 связей. Таким образом, в полученном множестве узлов, мощность подмножества закрытых узлов значительно больше мощности подмножества открытых узлов, приблизительно в 57 раз, данный показатель назовем разреженностью выборки. Этот показатель, как указывалось ранее, создает основное препятствие для изучения данного класса сетей. Следовательно, необходимо разработать алгоритм снижения разреженности выборки и учитывать данную особенность при дальнейшем исследовании.

Еще раз рассмотрим модель данных и рисунок 1.

Наличие некоторых обратных связей у закрытых узлов не гарантирует полного восстановления всех его связей. Тем не менее, наличие более одной такой связи дает ненулевую вероятность того, что восстановление этих прямых связей приведет к образованию нового пути в графе, тем самым повышая репрезентативность всей сети (Рисунок 2.2.6).

Из этих рассуждений следует, что с целью снижения разреженности, являющейся препятствием для определения характеристик сети, необходимо выделить прямые связи закрытых узлов, явно не представленные в этом формате.

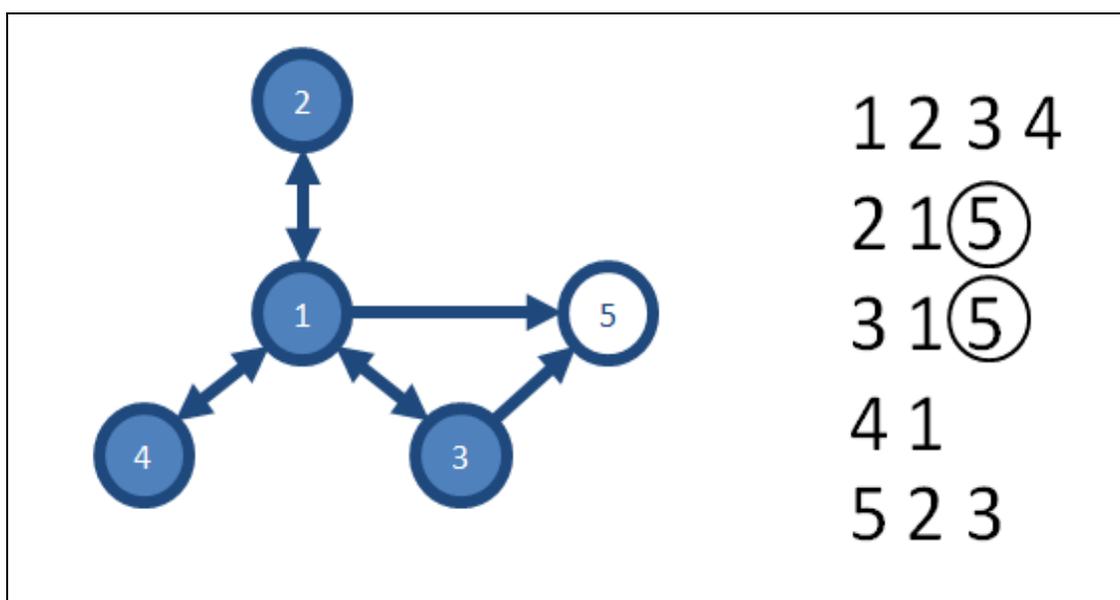


Рисунок 2.2.6 - Выделение прямых связей закрытых узлов

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Однако данная операция потребует полного чтения всего файла каждый раз когда потребуются восстановить связи очередного закрытого узла. В связи с этим было принято решение произвести реформатирование исходных данных в формат с избыточностью, а также произвести индексирование данных для уменьшения среднего времени случайного доступа.

Алгоритм индексирования заключается в разбиении всего множества узлов на непересекающиеся подмножества таким образом, чтобы можно было однозначно сказать, содержится ли искомый узел в некотором подмножестве, не производя при этом поиск нем.

Поскольку мощность этих множеств будет обратно пропорциональна количеству разбиений, поиск в каждом из них будет соответственно быстрее.

Критерием разбиения в данном случае является кратность значения идентификатора узла некоторой константе. Эмпирическим путем было выбрано значение 10,000 узлов на подмножество. Подмножеством в данном случае является файл, который содержит 10,000 узлов с идентификаторами от  $10,001 \cdot n$  до  $10,000 \cdot (n+1)$ . В качестве значения индекса в данном случае выступает имя файла. Часть имени указывает на интервал идентификаторов, который содержится в данном файле. Таким образом, при поиске некоторого узла будет произведено обращение к файлу, в названии которого содержится интервал, соответствующий идентификатору искомого узла.

Чтобы покрыть все множество узлов и учесть все представленные прямые связи закрытых узлов необходимо произвести поиск по всему множеству имеющихся данных для каждого узла. Несмотря на высокую ресурсоемкость, осуществить данную операция необходимо только один раз. Это позволит не только ускорить дальнейшие исследования, но и снизить разреженность выборки, тем самым повысив ее репрезентативность.

Псевдокод алгоритма реформатирования представлен на рисунке 2.2.7.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

```
строка = получитьСтрокуИзФайла ( ) ;
Пока (строка.длина > 0) {
  группа = заполнитьГруппуИзСтроки (строка) ;
  родУзел = группа.родительский ;
  Если (целевой (родУзел) ) {
    карта.добавитьГруппу (родУзел.ид, группа) ;
  } Иначе {
    ДляКаждого (дочУзел в группа.дочерние) {
      Если (целевой (дочУзел) ) {
        карта.получитьГруппу (дочУзел.ид) . доба-
вить (родУзел) ;
      }
    }
  }
  строка = получитьСтрокуИзФайла ( ) ;
}
записатьКартуВФайл ( ) ;
```

Рисунок 2.2.7 - Псевдокод алгоритма переформатирования

В результате выполнения данной операции было получено множество, состоящее из 9,621 файла по 10 тыс. групп узлов в каждом, упорядоченных по идентификаторам родительских узлов и отражающих связи закрытых узлов так же, как и открытых.

### *Выделение репрезентативного подмножества*

В виду того, что данные, доступные для исследования не обладают необходимой полнотой, а именно, являются случайной выборкой с точки зрения топологии исходной сети, встает задача выбора из множества исходных данных репрезентативного подмножества, отражающего в большей мере топологию исходной сети.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Сложность данной задачи состоит в невозможности выбрать подсеть, полностью состоящую из открытых узлов, и при этом достаточно большого диаметра, чтобы её можно было считать репрезентативной.

Причина проблемы состоит в низкой вероятности нахождения такой подсети даже в выборке несравнимо большего размера, поскольку отношение количества открытых узлов к закрытым на 2 порядка меньше единицы по результатам анализа исходных данных.

Следующий этап исследования заключается в выделении частично представительного подмножества обходом переформатированного исходного графа в ширину (Рисунок 2.2.8).

Переформатированный граф, как упоминалось, индексирован, поэтому время случайного доступа снижено, относительно исходного формата.

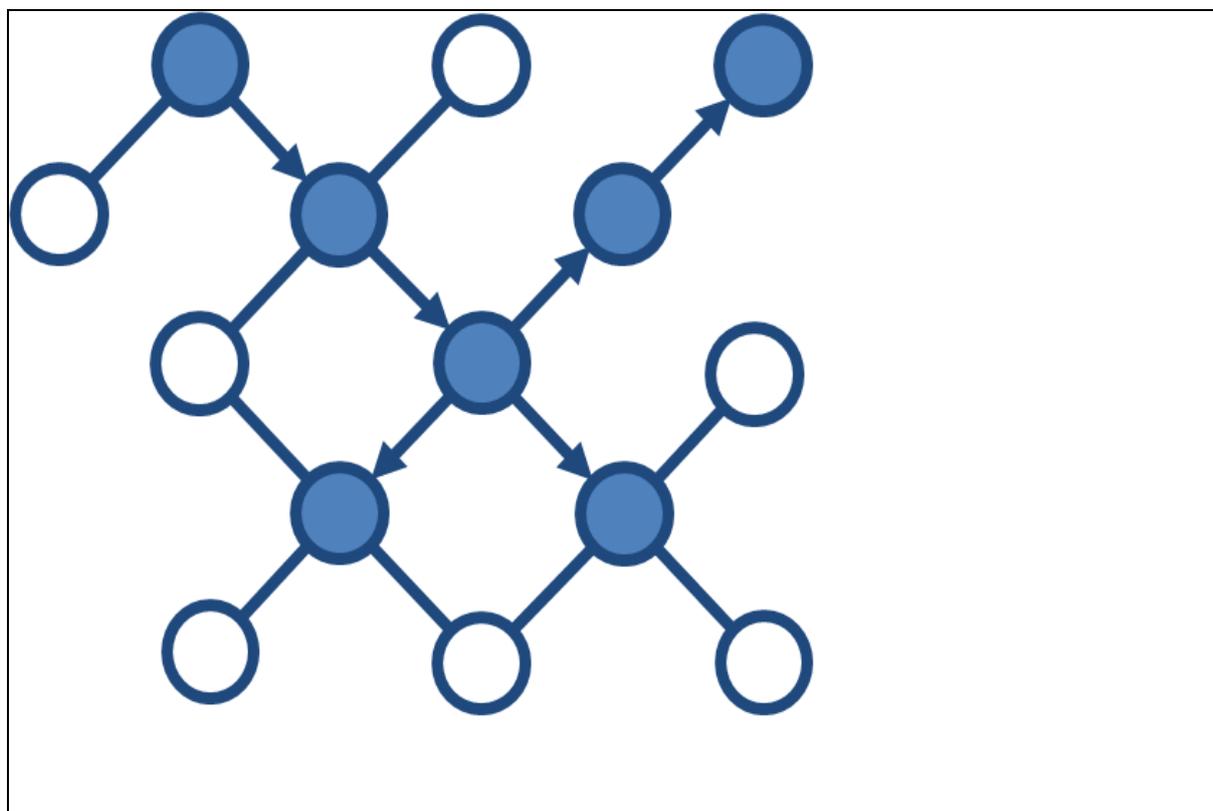


Рисунок 2.2.8 - Обход в ширину

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
На каждом шаге алгоритма (уровне глубины обхода) считается, что дочерние узлы, еще не присутствующие на карте, являются закрытыми.

На рисунке 2.2.9 представлен алгоритм обхода сетевого графа в ширину, скорость выполнения которого, как видно, сильно зависит от упомянутого времени случайного доступа.

```
Целое глубинаОбхода = 0 ;
Список<Целое> текущийУровень = Список<Целое> ( ) ;
Список<Целое> следующийУровень = Список<Целое> ( ) ;
текущийУровень . добавить ( номерИсходногоУзла ) ;
Пока ( глубинаОбхода < требуемаяГлубинаОбхода ) {

    ДляКаждого ( Целое номер : текущийУровень ) {
        Группа группа = получитьСтрокуИзФайла ( файл , номер ) ;
        Если ( группа . родитель . открытый И НЕ ( карта . содержитГруппу ( номер ) ) ) {
            карта . добавить ( группа ) ;
            следующийУровень . добавитьНомераУзлов ( группа . дочерние ) ;
        }
    }
    текущийУровень . очистить ( ) ;
    текущийУровень = следующийУровень ;
    глубинаОбхода += 1 ;
}
```

Рисунок 2.2.9 - Алгоритм обхода сетевого графа в ширину

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### *Получение распределения степеней связности в доступной части сети*

Предлагаемый алгоритм вычисления средней степени связности позволяет полностью отказаться от поддержания консистентности. Это обусловлено тем, что результирующая степень связности вычисляется исходя из независимых характеристик всех учитываемых узлов, а также тем, что необходимо учитывать только те узлы, характеристики которых известны, т.е. открытые.

Алгоритм вычисления средней степени связности представлен на рисунке 2.2.10.

```
Пока (не конецфайла (файл)) {
    карта.добавитьГруппу (новая          Группа (получить-
СледСтрокуИзфайла (файл)) );
}
Целое дочерние = 0;
Целое родительские = 0;
ДляКаждого (Группа группа : карта) {
    Если (группа.родитель.открытый) {
        дочерние += группа.дочерние.количество;
        родительские += 1;
    }
}
средСтепень = дочерние / родительские;
```

Рисунок 2.2.10 - Алгоритм вычисления средней степени связности

Полученные в результате применения данного алгоритма характеристики являются основой для применения алгоритма, описанного далее.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### *Получение распределения коэффициентов кластеризации в доступной части сети*

Основная проблема точного вычисления коэффициента кластеризации, как было сказано ранее, прямо связана с проблемой выделения достаточно большого репрезентативного подмножества.

В рамках научной работы был разработан алгоритм, позволяющий проводить вычисления с частично представительным подмножеством (Рисунок 2.2.11).

Для нахождения клики в произвольном множестве узлов необходимо чтобы минимум два узла являлись открытыми смежными и имели общий смежный узел. Для некоторого открытого узла можно посчитать количество образуемых при его участии клик, как мощность множества паросочетаний смежных ему узлов, включающего только пары также смежных узлов.

```
ДляКаждого (Группа группа : карта) {
    количествоКлик = 0;
    ДляКаждого (Узел дочернийУзел : группа.дочерние)
    {
        Если (карта.содержит (дочернийУзел.номер) ) {
            группаДочернего = карта.группа (дочер-
нийУзел.номер) ;
            количествоКлик += мощностьПересече-
ния (группа.дочерние, группаДочернего.дочерние) ;
        }
    }
    вывести (группа.родитель.номер, количествоКлик) ;
}
```

Рисунок 2.2.11 - Алгоритм, позволяющий проводить вычисления с частично представительным подмножеством

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Следует учитывать, что при использовании частичной консистентности пересечение множеств следует определять по равенству идентификаторов узлов, нежели ссылок на объекты. Кроме того, для корректного и простого подсчета количества клик необходимо прибавлять к общей сумме 1, если клика образована парой открытых узлов и одним закрытым, и 0.5, если клика образована тремя открытыми узлами. Поскольку подсчет ведется для некоторого открытого узла, то перебор всех пар смежных ему узлов дважды учтет те пары, в которых оба узла открыты (Рисунок 2.2.12).

```
Целое мощностьПересечения (  
Список<Узел> узлы1, Список<Узел> узлы2) {  
    Целое мощность = 0;  
    дляКаждого (Узел узел1 : узлы1) {  
        дляКаждого (Узел узел2 : узлы2) {  
            Если (узел1.номер == узел2.номер) {  
                если (узел1.открытый И узел2.открытый) {  
                    мощность += 0.5;  
                } иначе {  
                    мощность += 1;  
                }  
            }  
        }  
    }  
    вернуть мощность;  
}
```

Рисунок 2.2.12 – Уточнение алгоритма, представленного  
на рисунке 2.2.11

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### *Экстраполяция распределения коэффициентов кластеризации на недоступную часть сети*

Описанный алгоритм учитывает только клики, образуемые одним или двумя открытыми соседями некоторого узла.

Предлагаемый алгоритм экстраполяции основан на предположении о стохастичности распределения образуемых клик между открытыми и закрытыми соседями случайно выбранного узла, следовательно, о том, что способность образовывать клики и состояние (открыт/закрыт) узла – независимы.

Эта способность характеризуется вероятностью, с которой рассматриваемый узел может образовать клику с парой смежных узлов, иными словами, кластерным коэффициентом.

Всего узел может образовать  $P(O)+P(C)+O*C$  клик, где  $P(O)$  - количество всех возможных паросочетаний его открытых соседей,  $P(C)$  – закрытых,  $O$  и  $C$  – количество открытых и закрытых соседей соответственно.

Поскольку приведенный выше алгоритм подсчитывает количество клик образованных парами открытых узлов, а также открытыми и закрытыми узлами, и не учитывает только клики, проходящие через пары закрытых соседей, вероятность образования клики можно вычислить как отношение количества найденных клик к количеству всех возможных паросочетаний, за вычетом количества пар закрытых узлов:

$$K = \frac{C_k}{P_a - P_c},$$

где  $C_k$  - количество найденных клик,

$P_a$  – количество всех возможных паросочетаний,

$P_c$  – количество паросочетаний закрытых узлов.

Тогда усредненное количество всех клик, образуемых некоторым узлом, можно вычислить как произведение вероятности образования клики на количество всех возможных паросочетаний узлов, смежных рассматриваемому:

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

$$K_e = K * P_a .$$

Псевдокод функции, осуществляющей экстраполяцию, согласно описанному выше правилу приведен на рисунке 2.2.13.

```
Вещественное экстраКК (количествоКлик, группа) {
  Целое дочерних = группа.дочерние.количество;
  Целое дочернихЗакрытых = группа.дочерние.количествоЗакрытых;
  Целое всегоПар = дочерних * (дочерних - 1) / 2;
  Целое закрытыхПар =
  дочерЗакрытых * (дочернихЗакрытых - 1) / 2;
  экстраКК = количествоКлик / (всегоПар -
  закрытыхПар);
  вернуть экстраКК;
}
```

Рисунок 2.2.13 - Псевдокод функции, осуществляющей экстраполяцию

Таким образом, можно вычислить средний кластерный коэффициент всей сети на основании частичных данных о ее топологии, как усредненный коэффициент каждого узла, экстраполированный на закрытые участки через количество возможных паросочетаний.

Для этого модифицируем рассмотренный ранее алгоритм подсчета количества клик каждого узла сети (Рисунок 2.2.14).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

```
Список<Вещественное> списокЭКК = новый Спи-
сок<Вещественное>();
ДляКаждого (Группа группа : карта) {
    количествоКлик = 0;
    ДляКаждого (Узел дочернийУзел : группа.дочерние)
    {
        Если (карта.содержит (дочернийУзел.номер)) {
            группаДочернего = карта.группа (дочер-
нийУзел.номер);
            количествоКлик += мощностьПересече-
ния (группа.дочерние, группаДочернего.дочерние);
        }
    }
    Вещественное экстраКК =НОК
    экстраКК (группа.родитель.номер, количествоКлик);
    списокЭКК.добавить (экстраКК);
    вывести (экстраКК);
}
Целое среднийКК = 0;
ДляКаждого (Вещественное кк: списокКК) {
    среднийКК += кк;
}
среднийКК = среднийКК / списокКК.количество;
вывести (среднийКК);
```

Рисунок 2.2.14 - Алгоритм подсчета количества клик каждого узла  
сети

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

### *Генерация недоступной части сети*

Поскольку алгоритм построен по архитектуре со многими вычислительными процессами (дочерними) и одним процессом, управляющим вычислениями (главным), далее будут присутствовать пояснения, уточняющие, на каком типе процесса выполняется данная часть алгоритма.

Если в названии функций присутствует префикс «глав\_» это так же значит, что функция выполняется на главном процессе, префикс «доч\_» соответствует функциям дочерних процессов.

Выполнение алгоритма начинается всеми процессами одновременно, но первая его часть относится только к главному процессу, а остальные ожидают выполнения этой части (Рисунок 2.2.15).

```
глав_получитьДанные () {
    Карта общаяКарта = парсингСПолнойКонсистентно-
стью (файл) ;
    ДляКаждого (Группа группа : общаяКарта) {
        Если (группа.родитель.открытый) {
            картаОткрытых.добавить (группа) ;
        } Иначе {
            картаЗакрытых.добавить (группа) ;
        }
    }
    общаяКарта.очистить () ;
    вывести (картаОткрытых) ;
    картаОткрытых.очистить () ;
}
```

Рисунок 2.2.15 – Алгоритм генерации недоступной части сети

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

Необходимо назначить степень связности закрытым узлам, для которых мы будем генерировать связи. Эта степень будет назначаться узлам функцией, генерирующей псевдослучайную последовательность (Рисунок 2.2.16).

```
глав_назначитьСтепень () {  
  ДляКаждого (Группа группа : картаЗакрытых) {  
    группа.родитель.генСтепСвяз =  
    сгенерироватьСлучайнуюСтепеньСвязности ();  
  }  
}
```

Рисунок 2.2.16 – Алгоритм назначения степени связности закрытым узлам

Далее необходимо распределить полученные данные по остальным процессам для их обработки. На каждый процесс необходимо отправить список узлов, для которых нужно сгенерировать назначенное количество связей. На данном этапе отправляются только сами закрытые узлы, но не их уже возможно найденные соседи (Рисунок 2.2.17).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

```
глав_распределитьДанные () {
    КартаРаспределения картаРаспределения =
    новая Карта<Целое, Пара<Целое,Целое>> ();
    Для (идПроц = 1; идПроц < количествоПроцессов;
идПроц++) {
        СписокУзлов порция =
        получитьПорциюДляОтправки (картаЗакрытых,
идПроц) ;
        отослать (порция, идПроц) ;
        картаРаспределения.добавить (идПроц,
порция.минИдПроц (), порция.максИдПроц ()) ;
        порция.очистить () ;
    }
}

доч_распределитьДанные () {
    СписокУзлов порция = полу-
чить (идГлавПроц) ;
    общаяКарта.добавить (порция) ;
}
```

Рисунок 2.2.17 – Псевдокод распределения полученных данных по остальным процессам для их обработки

Далее необходимо добавить соседей закрытых узлов в соответствующие контейнеры, чтобы учесть уже имеющиеся связи при дальнейшей генерации (Рисунок 2.2.18).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

```
=====  
глав_добавитьДочерниеВКонтейнер (картаЗакрытых ,  
картаРаспределения) {  
  ДляКаждого (Группа группа : картаЗакрытых) {  
    Целое    идПроц    =    определитьКонтей-  
нер (группа .родитель) ;  
    отправить (группа , идПроц) ;  
  }  
  окончитьСеанс () ;  
}  
доч_добавитьДочерниеВКонтейнер () {  
  Пока (Не сеансОкончен ()) {  
    Группа порция = получить (идГлавПроц) ;  
    Группа группа =  
      общаяКарта .получить (группа .родитель .ид) ;  
    группа .дочерние .добавить (порция .дочерние) ;  
  }  
}
```

Рисунок 2.2.18 – Алгоритм добавления соседей закрытых узлов  
в соответствующие контейнеры

Перед генерацией нужно получить список закрытых узлов, отсортированный по количеству недостающих связей в порядке убывания. Таким образом, в первую очередь будут сгенерированы связи для наиболее связных узлов.

Данное условие необходимо учитывать по той причине, что, как описывалось ранее, между двумя узлами, в рамках предметной области, может существовать лишь одна ненаправленная связь. То есть, для каждого узла, для которого предсказано наличие  $N$  связей, нужно найти  $N$  уникальных узлов, таких, что для каждого из них при добавлении новой связи, не будет превышено предсказанное для него значение степени связности.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

В связи с тем, что накопление связей узлами с низким предсказанным значением степени связности происходит быстрее, эти узлы раньше удаляются из множества возможных сочетаний, тем самым уменьшая его мощность и, следовательно, вероятность в очередной момент времени найти в нем  $N$  уникальных сочетаний.

Данная задача должна быть выполнена на главном процессе, распределяющем задачи генерации между остальными. Здесь под задачей подразумевается генерация связей для некоторого множества узлов.

На этом подготовительная часть алгоритма завершается и начинается выполняться основная задача – распределенная генерация связей до тех пор, пока характеристики графа не будут соответствовать ожидаемым (Рисунок 2.2.19).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

```
глав_распределеннаяГенерация (списокЗакрытых ,
картаРаспределения) {
Пока (Не списокЗакрытых.пуст ()) {
СписокУзлов порция = глав_получитьПорцию (спи-
сокЗакрытых) ;
СписокУзлов готовые = выбратьГотовые (пор-
ция) ;
глав_получитьДочИзКонт (готовые) ;
вывести (готовые) ;
глав_удалитьИзКонтейнеров (готовые) ;
глав_генерацияДляПорции (порция) ;
Карта результат = глав_собратьРезуль-
тат (порция) ;
вывести (результат) ;
глав_удалитьИзКонтейнеров (результат) ;
}
окончитьСеанс () ;
}

доч_распределеннаяГенерация () {
Пока (Не сеансОкончен ()) {
доч_получитьДочИзКонт () ;
доч_удалитьИзКонтейнеров () ;
Карта результат = доч_генерацияДляПорции () ;
доч_собратьРезультат (результат) ;
доч_удалитьИзКонтейнеров () ;
}
}
```

Рисунок 2.2.19 – Алгоритм распределенной генерации связей между узлами

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

### **Моделирование**

Одним из способов решения трудоёмких вычислительных задач являются распределённые вычисления. Они производятся при помощи нескольких компьютеров, чаще всего объединённых в параллельную вычислительную систему. Особенностью распределённых вычислительных систем является возможность неограниченного наращивания производительности за счет масштабирования.

Параллельные вычисления использовались много лет в основном в высокопроизводительных вычислениях, но в последнее время к ним возрос интерес вследствие существования физических ограничений на рост тактовой частоты процессоров.

В данной работе предлагается алгоритм распределенного моделирования распространения информации в крупномасштабных социальных сетях, основанный на распределении множества задач распространения информации в отдельных участках сети по вычислительным ресурсам с дальнейшей агрегацией независимо полученных результатов.

Данный алгоритм является последним этапом исследования в предлагаемом методе. Он оперирует данными, полученными в результате последовательного выполнения изложенных выше алгоритмов.

### *Модель распространения*

С точки зрения распространения информации в крупномасштабной социальной сети целью процесса моделирования является перевод некоторых узлов сети из начального состояния, при котором считается, что узлу неизвестна рассматриваемая информация, в конечное состояние, при котором считается, что рассматриваемая информация узлу известна.

Перевод из начального состояния в конечное может осуществляться через множество промежуточных этапов.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Это множество определяется применяемой при моделировании моделью распространения информации в сети.

В данной работе моделирование осуществляется с использованием SIR-модели. В данной модели узлу могут соответствовать 3 условных состояния:

- S – состояние восприимчивости к воздействию распространяемой информации. Это начальное состояние, в котором пребывает большинство узлов рассматриваемой сети. То есть в начале моделирования узлам, находящимся в данном состоянии неизвестна рассматриваемая информация;
- I – состояние узла, при котором он находится под воздействием распространенной на него информации, будем называть такой узел зараженным. Узел, находящийся в данном состоянии стремится передать рассматриваемую информацию, другим узлам, с которыми у него есть связь, то есть всем смежным с ним узлам. Переход в данное состояние возможен только из состояния S. Успешность перехода в состояние I зависит от соответствующего коэффициента, определяющего вероятность перехода, который является одним из параметров модели, а также количества смежных узлов распространяющих информацию, т.е. от количества источников воздействия на рассматриваемый узел. Также считается, что до начала моделирования некоторая, обычно достаточно малая, часть узлов находится в этом состоянии. В рамках моделирования процесса распространения эти узлы являются его инициаторами;
- R – состояние невосприимчивости к воздействию распространяемой информации. Это конечное состояние, при котором узлу известна рассматриваемая информация, однако сам он больше не стремится распространить ее и, следовательно, не может перейти обратно в состояние I.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

### *Распределенное решение задачи*

На основании сформулированных ограничений построим граф состояний для некоторого узла сети (Рисунок 2.2.20) Символами  $i$  и  $r$  обозначены вероятности перехода в очередное состояние.

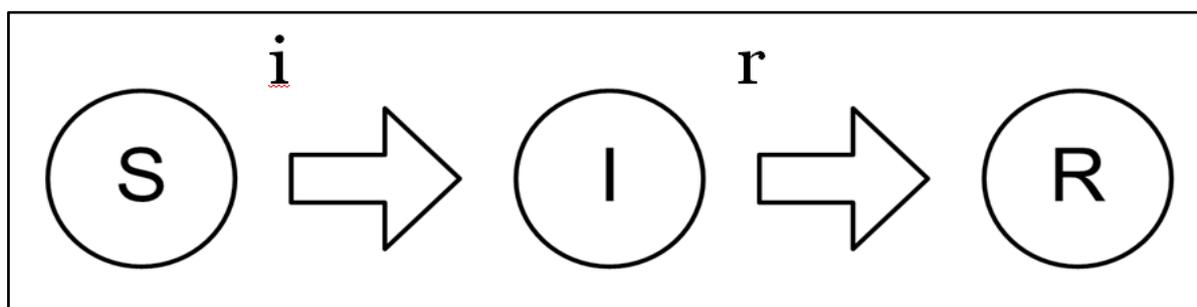


Рисунок 2.2.20 - Граф состояний узла в рамках SIR-модели

Поскольку данный граф представляется собой некоторый конечный вероятностный автомат Мура и его выходные сигналы зависят лишь от состояния самого автомата, или, иначе говоря, от момента времени, в который происходит снятие сигнала, то следует ввести в модель тактование.

Таким образом, в модели определяется некоторый квант времени, соотношение продолжительности которого с реальным временем необходимо определять отдельно для каждой сети. Данное значение может сильно различаться в отдельных участках сети, в связи с чем, в модель необходимо вносить соответствующие корректировки для более точного прогнозирования процесса распространения. В данной работе рассматривается случай одинакового значения продолжительности кванта по всей сети. Таким образом процесс распространения можно рассматривать с точки зрения дискретного времени (Рисунок 2.2.21).

На основании изложенных выше рассуждений можно выделить две элементарные задачи, решаемые в один квант времени для некоторого зараженного узла сети:

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- вероятностное осуществление перевода всех смежных узлов, находящихся в состоянии  $S$ , некоторого зараженного узла в состояние  $I$ ;
- вероятностное осуществление перевода некоторого зараженного узла в состояние  $R$ .

Узлы, находящиеся в состоянии  $S$ , но не смежные ни с одним зараженным могут не рассматриваться в данном кванте.

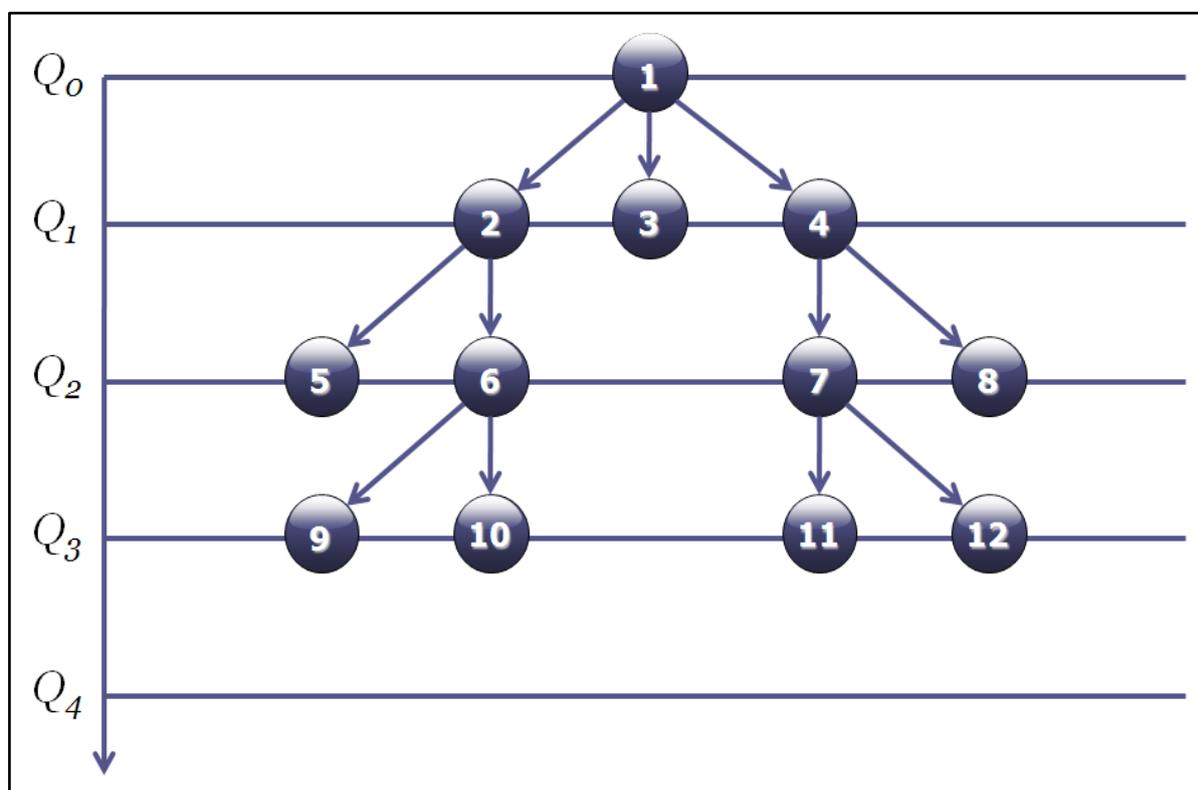


Рисунок 2.2.21 - Процесс распространения в дискретном времени

### *Алгоритм моделирования*

Обычно параллельные вычисления требуют координации действий. В предлагаемом алгоритме также присутствует координирующий процесс, осуществляющий агрегацию результатов решения указанных задач на других вычислительных процессах.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Координирующий блок осуществляет следующие функции:

- сохраняет состояние процесса в течение всего моделирования;
- распределяет задачи по вычислительным ресурсам;
- осуществляет тактовую синхронизацию вычислений;
- агрегирует результаты решения отдельных задач в текущий квант.

Вычислительные процесс осуществляют следующие функции:

- принимает задачу на выполнение. Задача представляется множеством зараженных в предшествующие кванты узлов;
- извлекает из локальной копии карты сети множество узлов, смежных с зараженными;
- осуществляет попытку заразить извлеченные смежные узлы;
- осуществляет попытку иммунизировать полученные зараженные узлы;
- отправляет результат координирующему блоку.

Более детально алгоритм моделирования выглядит следующим образом (Рисунок 2.2.22).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

```
=====  
  
глав_моделирование () {  
    СписокУзлов инфУзлы = списокИнициаторов ;  
    СписокУзлов  иммУзлы  =  новый  Списо-  
кУзлов () ;  
    синхронизировать (инфУзлы. количество) ;  
    Если (инфУзлы. количество > 0)  
    {  
        глав_распредПоПроцессам (ин-  
фУзлы) ;  
        СписокУзлов  квантИнфУзлы  =  
глав_получитьИнфУзлы () ;  
        агрегировать (квантИнфУзлы,  ин-  
фУзлы) ;  
        СписокУзлов  квантИммУзлы  =  
глав_получитьИммУзлы () ;  
        агрегировать (квантИммУзлы,  им-  
мУзлы) ;  
        удалитьИммИзИнф (инфУзлы,  им-  
мУзлы) ;  
        вывести (инфУзлы,  иммУзлы) ;  
        синхронизировать (инфУзлы. коли-  
чество) ;  
    }  
}
```

Рисунок 2.2.22 - Алгоритм моделирования

Как видно из псевдокода, на начальном этапе алгоритм оповещает вычислительные процессы о том, какие узлы в данный квант инфицированы.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Множество этих узлов тривиально разбивается на непересекающиеся подмножества сравнимой мощности, данный этап подробно не описывается.

Далее, после значительного простоя, координирующий процесс получает результаты моделирования от вычислительных процессов, так как основная часть задач решается ими.

Функция агрегации производит слияние двух списков узлов, исключая при этом дублирование.

Далее из агрегированного списка исключаются иммунизированные узлы, о состоянии которых неизвестно вычислительным процессам, вследствие чего им присваивается состояние инфицирован.

После этого агрегация результатов считается завершенной и агрегированный результат за данный квант выводится во внешний файл.

Производится синхронизация количества инфицированных на данный момент узлов с целью оповещения вычислительных процессов о конце моделирования в случае если оно равно нулю.

Теперь рассмотрим детально алгоритм, выполняющийся вычислительными узлами (Рисунок 2.2.23).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

```
=====  
  
доч_моделирование () {  
    СписокУзлов инфУзлы = новый СписокУзлов ();  
    СписокУзлов иммУзлы = новый СписокУзлов ();  
    синхронизировать (инфУзлы. количество) ;  
    Если (инфУзлы. количество > 0)  
    {  
        инфУзлы = доч_получитьЗадачу () ;  
        СписокУзлов соседи = загрузитьСоседей (инфУзлы) ;  
        СписокУзлов квантИнфУзлы = записать (соседи) ;  
        СписокУзлов квантИммУзлы = вылечить (инфУзлы) ;  
        доч_передатьИнфУзлы (квантИнфУзлы) ;  
        доч_передатьИммУзлы (квантИммУзлы) ;  
        синхронизировать (инфУзлы. количество) ;  
    }  
}
```

Рисунок 2.2.23 - Алгоритм, выполняющийся вычислительными узлами

Вычислительный процесс получает от координирующего задачу, которая, как было описано ранее, представлена множеством инфицированных в текущий квант узлов.

Процесс загружает все узлы, смежные с инфицированными, в общий список из локальной копии карты, индексированной на раннем этапе исследования.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Процесс предпринимает попытку распространить влияние информации на эти узлы (инфицировать их). Затем предпринимается попытка иммунизировать полученные узлы, чтобы к концу текущего кванта некоторые из них стали невосприимчивы к информации.

Полученные результаты передаются координирующему процессу и производится синхронизация.

В результате выполнения алгоритма в выходном файле оказываются данные о динамике распространения инфекции, представленные списками зараженных и иммунизированных узлов в каждый квант времени.

### **Анализ результатов экспериментов**

Для апробации предлагаемого алгоритма был проведен ряд экспериментов, включающих сбор данных о таких характеристиках, как степень связности и коэффициент кластеризации сети.

Полученные данные были проанализированы. На основе проведенного анализа были выдвинуты гипотезы о существовании в исследуемом классе сетей определенных закономерностей, что будет подробно рассмотрено далее.

### ***Распределение степеней связности***

Применением алгоритма выделения репрезентативного подмножества было получено несколько выборок подсетей различного диаметра.

Для полученных выборок было построено распределение относительной встречаемости узлов по интервалам значений соответствующих им степеней связности.

Анализ по всем выборкам показал схожие результаты, далее будет приведен анализ по одной из выборок.

На рисунке 2.2.24 изображено построенное распределение и аппроксимирующая его кривая.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

---

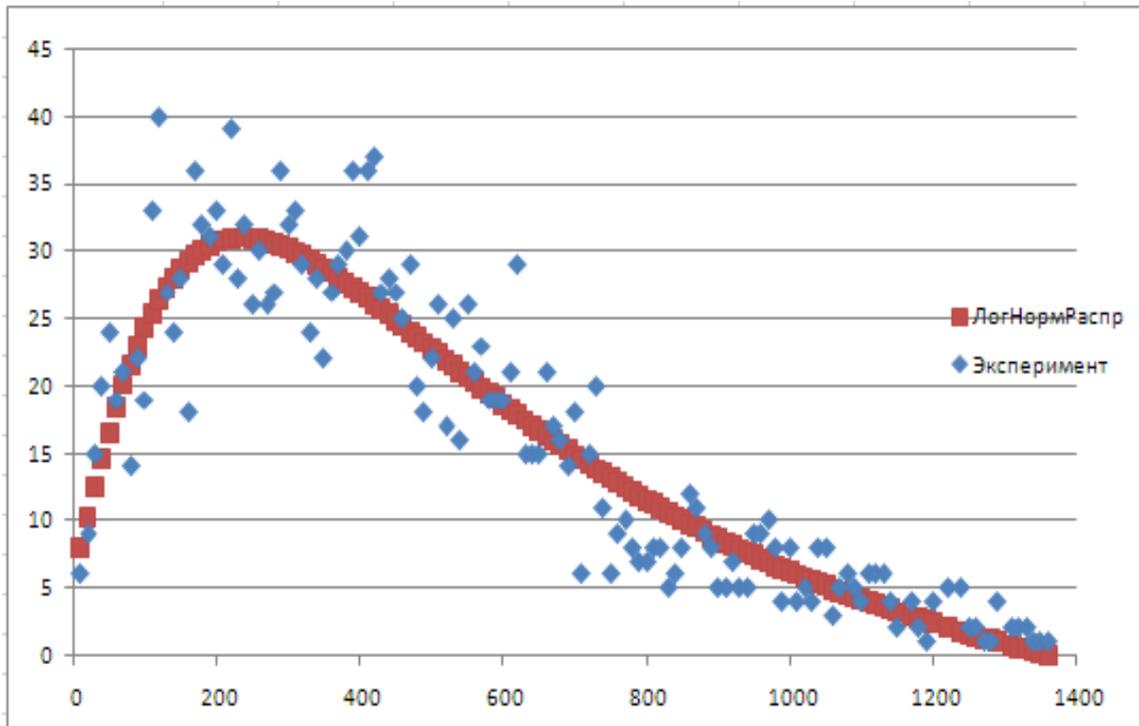


Рисунок 2.2.24 - Распределение степеней связности в выбранной подсети

Аппроксимирующая кривая является функцией лог-нормального распределения. Коэффициенты данной функции были получены из экспериментальных данных. Среднее значение логарифма составляет 5,817521, стандартное отклонение логарифма составляет 0,852231.

Для подтверждения достоверности предлагаемой модели был рассчитан коэффициент детерминации.

Максимальное значение данного коэффициента было достигнуто при смещении функции на -100 единиц по оси значений степени связности и на -9 единиц по оси абсолютной частоты попаданий. Значение коэффициента детерминации составило 0,871945, что является показателем достаточной достоверности для использования предлагаемой модели.

Данные результаты были применены при генерации недоступной части сети.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
Для сгенерированной топологии было построено аналогичное распределение (Рисунок 2.2.25).

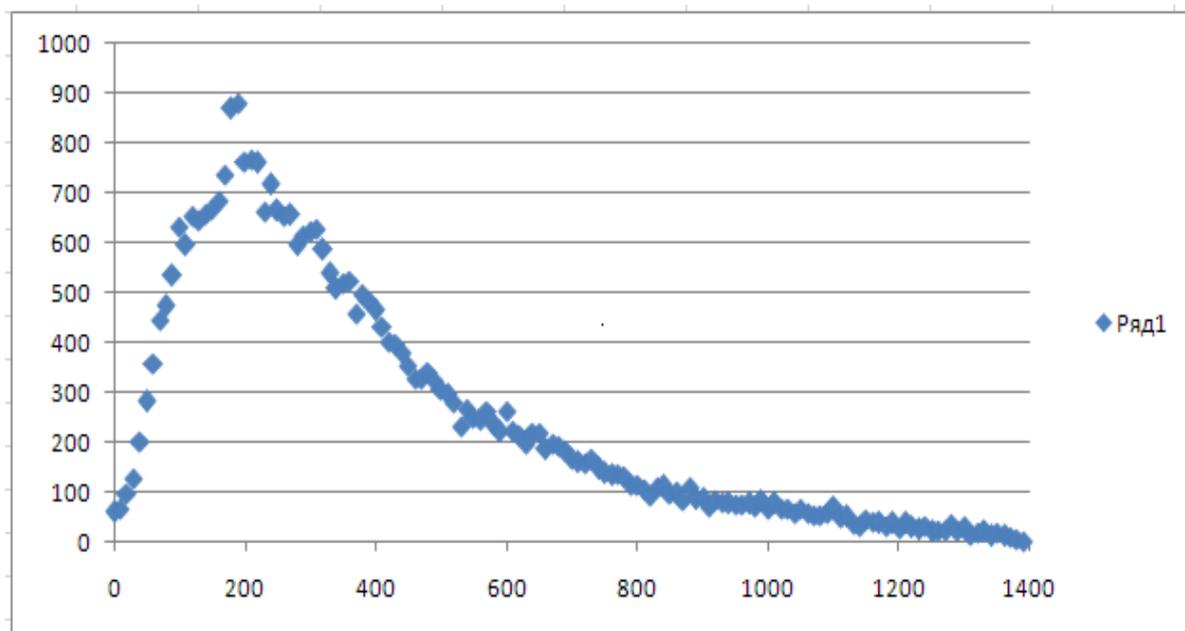


Рисунок 2.2.25 - Распределение степеней связности  
в сгенерированной сети

Для данного распределения также были рассчитаны коэффициенты функции лог-нормального распределения. Их значения соответствовали ожидаемым с относительной погрешностью 0.02.

### *Распределение коэффициентов кластеризации*

Для того же участка сети было построено распределение коэффициента кластеризации (Рисунок 2.2.26).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

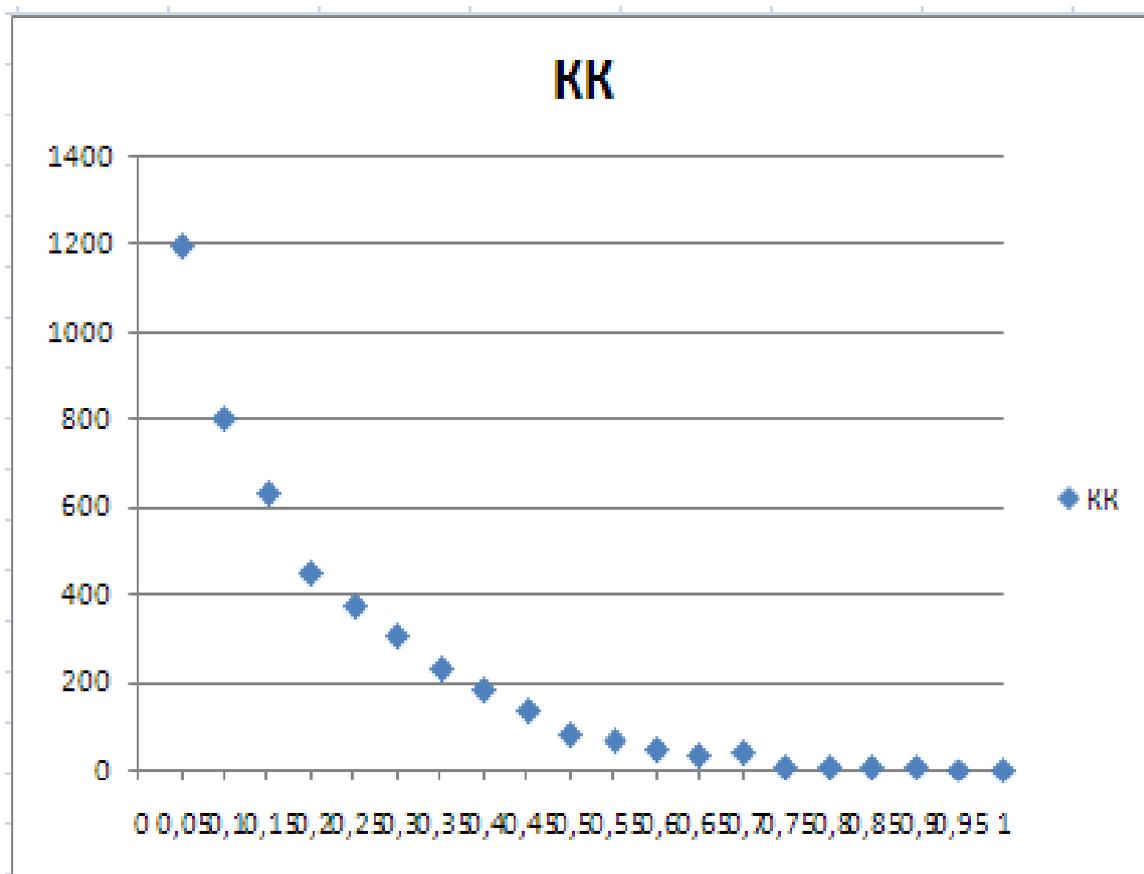


Рисунок 2.2.26 - Распределение коэффициента кластеризации  
в сгенерированной сети

Для аппроксимации данного распределения был предложен степенной закон. Однако при рассмотрении интервала значений от 0 до 0,05 распределение демонстрирует несвойственное для данного закона поведение (Рисунок 2.2.27).

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

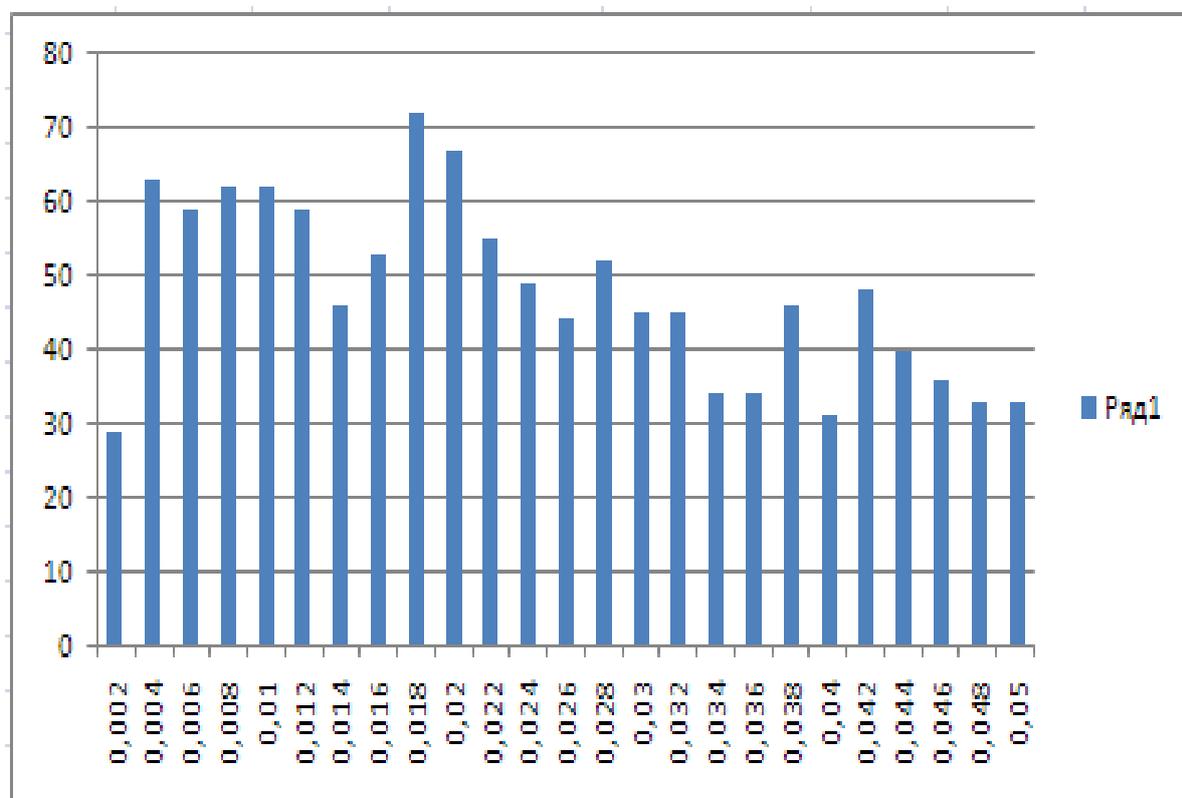


Рисунок 2.2.27 - Распределение коэффициента кластеризации  
в интервале от 0 до 0,05

Также по мнению некоторых экспертов, наличие степенного распределение коэффициента кластеризации в данном классе маловероятно.

Была выдвинута гипотеза о наличии в крупномасштабных социальных сетях лог-нормального распределения коэффициента кластеризации. Однако для экспериментального подтверждения данной гипотезы необходимо исследовать более объемную репрезентативную выборку, поскольку, как видно из графиков, относительная частота попадания характеристики узла в указанный интервал значительно меньше, чем в остальные. Таким образом, в интервале от 0 до 0,05 ошибка модели будет крайне велика и вычислить характеристики предполагаемого закона в данный момент не представляется возможным.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====  
Резюмируя выполненную работу, отметим следующее:

- разработан метод моделирования распространения информации в крупномасштабных социальных сетях;

- получены статические характеристики доступной для исследования части социальной сети, на основании полученных данных сгенерирована недоступная часть социальной сети, смоделирован процесс распространения информации в полученной сети;

- результаты теоретически обоснованы с помощью аппарата теории графов, теории вероятностей, теории множеств;

- разработанные алгоритмы, позволяют проводить исследования крупномасштабных социальных сетей с использованием распределенных вычислительных ресурсов. Применение разработанного метода в сочетании с более точными моделями распространения информации позволит рассчитать скорость роста потенциальных затрат на противодействие распространению нежелательной информации, а также получить зависимость скорости распространения от затрат на его организацию.

Для дальнейшего исследования информационных процессов в классе крупномасштабных социальных сетей необходимо разработать специальные модели распространения информации, а также провести более масштабный эксперимент, задействовав распределенные вычислительные ресурсы для уточнения характеристик существующих моделей.

### **Вопросы и задания**

1. Что лежит в основе модели данных?
2. Почему в качестве формата представления исходных данных был выбран именно текстовый файл?
3. Охарактеризуйте алгоритм обхода сетевого графа в ширину.
4. Что такое степень связности графа?
5. Чем отличается алгоритм «обход в глубину» от алгоритма «обход в ширину»?

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

6. Дайте определение понятию «консистентность».
7. Какие 3 условных состояния есть в SIR-модели?
8. Перечислите функции, которые осуществляет координирующий блок.
9. Что такое аппроксимирующая прямая?
10. Каким образом происходит распределение коэффициента кластеризации в сгенерированной сети?
11. Что лежит в основе алгоритма экстраполяции?
12. Назовите два вида представления топологии социальной сети.

### Список рекомендуемых источников

1. Абрамов, К.Г. Влияние перколяционного кластера на распространение нежелательной информации в социальных медиа [Текст] / К.Г. Абрамов; Проблеми інформатики і моделювання. Тезиси одинадцятої міжнародної науково-технічної конференції. – Харків-Ялта, 2011. - С. 4-5.
2. Абрамов, К.Г., Монахов, Ю.М., Медведникова, М.А., Трусова, А.И., Бодров, И.Ю. Статистические параметры топологии социальных сетей [Текст] / К.Г. Абрамов [и др.]; Математика и математическое моделирование. Труды научно-практической конференции, Мордовский государственный педагогический институт имени М.Е. Евсевьева. – 2011.
3. Абрамов, К.Г., Монахов, Ю.М., Медведникова, М.А., Трусова, А.И., Бодров, И.Ю. К вопросу о моделировании процесса пропаганды в социальных сетях [Текст] / К.Г. Абрамов [и др.]; Математика и математическое моделирование. Труды научно-практической конференции,

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

- Мордовский государственный педагогический институт имени М. Е. Евсевьева. – 2011.
4. Абрамов, К.Г., Малышев, Р.В., Монахов, Ю.М. К вопросу о топологических характеристиках социальной сети «В КОНТАКТЕ» [Текст] / К.Г. Абрамов, Р.В. Малышев, Ю.М. Монахов; Перспективные технологии в средствах передачи информации: Материалы 10-й международной научно-технической конференции, Владим. гос. ун-т. - 2013. - т. 2. – С. 115-118.
  5. Абрамов, К.Г., Монахов, Ю.М. Модель распространения спама в социальных сетях [Текст] / К.Г. Абрамов, Ю.М. Монахов; Современные информационные технологии в образовательном процессе и научных исследованиях: Материалы III Международной научно-практической конференции - Шуя-Иваново-Владимир: Изд. ГОУ ВПО "ШГПУ", 2010. - 136 с. - ISBN 978-5-86229-219-0.
  6. Абрамов, К.Г., Монахов, Ю.М. Алгоритмическая модель экстраполяции топологических характеристик социальных сетей [Текст] / К.Г. Абрамов, Ю.М. Монахов; Всероссийский научно-технический журнал «Проектирование и технология электронных средств», №4. - 2012. - С. 35-39.
  7. Абрамов, К.Г., Монахов, Ю.М. Моделирование распространения нежелательной информации в социальных медиа [Текст] / К.Г. Абрамов, Ю.М. Монахов; Труды XXX Всероссийской научно-технической конференции. Проблемы эффективности и безопасности функционирования сложных технических и информационных систем / Серпуховский ВИ РВ. - 2011. – ч. IV. - С. 178-182. - ISBN 978-5-91954-029-8.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- =====
8. Абрамов, К.Г., Монахов, Ю.М. Программа для моделирования распространения нежелательной информации в социальных сетях ModelGraph [Электронный ресурс] / К.Г. Абрамов, Ю.М. Монахов; Материалы выставки основных результатов научных исследований, разработок, технического творчества студентов, аспирантов и молодых специалистов - Дни науки студентов ВлГУ - Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых (6-8 апреля 2011 г.). – Режим доступа: <http://izi.vlsu.ru/НТС/5.pptx>
  9. Абрамов, К.Г., Монахов, Ю.М. Топологические характеристики социальной сети «ВКОНТАКТЕ» [Текст] / К.Г. Абрамов, Ю.М. Монахов; Труды XXXII Всероссийской научно-технической конференции. Проблемы эффективности и безопасности функционирования сложных технических и информационных систем / Серпуховский ВИ РВ. - 2013. - ч.IV. - С. 136-140. - ISBN 978-5-91954-074
  10. Абрамов, К.Г., Монахов, Ю.М. Стохастические модели распространения нежелательной информации в социальных сетях [Текст] / К.Г. Абрамов, Ю.М. Монахов; Сборник научных трудов SWorld. Материалы международной научно-практической конференции «Современные проблемы и пути их решения в науке, транспорте, производстве и образовании '2011», №4. – Одесса: Черноморье, 2011. – 411-0886 – С. 42-46
  11. Абрамов, К.Г., Монахов, Ю.М., Бодров, И.Ю. К вопросу о моделировании топологии социальных сетей [Текст] / К.Г. Абрамов [и др.]; Труды пятой всероссийской научно-практической конференции по имитационному

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

- моделированию и его применению в науке и промышленности "Имитационное моделирование. Теория и практика" ИММОД-2011. – Санкт-Петербург: ОАО "Центр технологии и судостроения", 2011. - 448 с.; - С.373-378. - ISBN 978-5-905526-02-2.
12. Абрамов, К.Г., Монахов, Ю.М. Модели распространения вредоносных программ в топологически гетерогенных социальных сетях [Электронный ресурс] / К.Г. Абрамов, Ю.М. Монахов; Труды НТС. Комитет по информатизации, связи и телекоммуникациям Администрации Владимирской области. – 2010. – Режим доступа: <http://ksi.avo.ru/>
  13. Абрамов, К.Г., Монахов, Ю.М. Некоторые аспекты безопасности Интернета в условиях инфраструктуры web 2.0 [Текст] / К.Г. Абрамов, Ю.М. Монахов; Труды X Российской научно-технической конференции "Новые информационные технологии в системах связи и управления". (Калуга, 1-2 июня 2011г.) - Калуга: Изд. "Ноосфера", 2011. - 610 с.; - С. 593-595. - ISBN 978-5-89552-322-3.
  14. Абрамов, К.Г., Монахов, Ю.М., Никиташенко, А.В. К вопросу об уточнении моделей распространения нежелательной информации в социальных сетях Интернета [Электронный ресурс] / К.Г. Абрамов, Ю.М. Монахов, А.В. Никиташенко; Информационные системы и технологии ИСТ-2011: материалы XVII международной научно-технической конференции (Н.Новгород, 22 апреля 2011 года) - Н. Новгород: Электронное издание, 2011. – 149 с.; - ISBN 978-5-9902087-2-8.
  15. Абрамов, К.Г., Монахов, Ю.М., Распространение нежелательной информации в социальных сетях

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

Интернета [Текст] / К.Г. Абрамов, Ю.М. Монахов;  
Перспективные технологии в средствах передачи информации: Материалы 9-ой международной научно - технической конференции; редкол.: А.Г. Самойлов [и др]. - Владимир: издат. ВлГУ, 2011. - Т. 1. - 272 с.; - ISBN 978-5-905527-02-9.

16. Безруков, Н.Н. Компьютерная вирусология [Текст] / Н.Н. Безруков; - Киев: Укр. сов. энцикл., 1991. - 416 с.
17. Биячуев, Т.А. Безопасность корпоративных сетей [Текст]: учеб. пособие / Т.А. Биячуев; под ред. Осовецкого Л.Г. - СПб.: СПбГУ ИТМО, 2004. - 161 с.
18. Бреер, В.В. Стохастические модели социальных сетей [Текст] / В.В. Бреер; Управление большими системами, № 27. – 2009. - С. 169-204.
19. Груздева, Л.М., Монахов, Ю.М., Монахов, М.Ю. Оценка сетевых характеристики компьютерных сетей в условиях информационного вредоносного воздействия [Текст]: учеб. пособие (с грифом УМО) / Л.М. Груздева, Ю.М. Монахов, М.Ю. Монахов; Владим. Гос.ун-т. – Владимир: Изд-во Владим. Гос. ун-та, 2010. – 86 с.
20. Груздева, Л.М., Абрамов, К.Г., Монахов, Ю.М. Экспериментальное исследование корпоративной сети передачи данных с адаптивной системой защиты информации [Текст] / Л.М. Груздева, К.Г. Абрамов, Ю.М. Монахов; Приборостроение. – М., 2012. - Т. 55, № 8. - С. 57-59
21. Губанов, Д.А., Новиков, Д.А., Чхартишвили, А.Г. Социальные сети: модели информационного влияния, управления и противоборства [Текст] / Д.А. Губанов, Д.А. Новиков, А.Г. Чхартишвили; под ред. чл.-

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- корр. РАН Д.А. Новикова — М.: Издат. физико-математической литературы, 2010. - 228 с.; - ISBN 9785-94052-194-5.
22. Гусева, А.И. Технология межсетевых взаимодействий [Текст] / А.И. Гусева; - М.: Бинум, 1997. - 238 с.
  23. Жаринов, И.В., Крылов, В.В. Конструирование графов с минимальной средней длиной пути [Текст] / И.В. Жаринов, В.В. Крылов; Вестник ИжГТУ, №4. – 2008. - С. 164-169. - ISSN 1813-7903.
  24. Качалин, А.И. Моделирование процесса распространения сетевых червей для оптимизации защиты корпоративной сети [Текст] / А.И. Качалин; Искусственный интеллект, № 2. – 2006. - С. 84-88.
  25. Монахов, Ю.М., Абрамов, К.Г. Моделирование распространения нежелательной информации в социальных медиа [Текст] / Ю.М. Монахов, К.Г. Абрамов; Вестник КГУ им. Н.А. Некрасова. - 2011. – Т.17, №3. - С. 15-18
  26. Чипига, А.Ф., Пелешенко, В. С. Математическая модель процессов связи узлов в сети при обнаружении и предотвращении несанкционированного доступа к информации [Электронный ресурс] / А.Ф. Чипига, В.С. Пелешенко. – Режим доступа: [http://science.ncstu.ru/articles/ns/002/elen/29.pdf/file\\_download](http://science.ncstu.ru/articles/ns/002/elen/29.pdf/file_download).
  27. Чипига, А.Ф., Пелешенко В.С. Формализация процедур обнаружения и предотвращения сетевых атак [Электронный ресурс] / А.Ф. Чипига, В.С. Пелешенко; Режим доступа: <http://www.contrterror.tsure.ru/site/magazine8/05-17-Chipiga.htm>

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- =====
28. Чубин, И. ARP-spoofing [Электронный ресурс] / И. Чубин; Режим доступа: <http://xgu.ru/wiki/ARP-spoofing>
  29. Ahn, Y., Han, S., Kwak, H., Moon, S., Jeong, H., Analysis of topological characteristics of huge online social networking services [Text] / Y. Ahn, S. Han, H. Kwak, S. Moon, H. Jeong; 16<sup>th</sup> International Conference on the World Wide Web. – 2007. - P. 835-844.
  30. Albert, R., Barabasi, A., Statistical mechanics of complex networks [Text] / R. Albert, A. Barabasi; Reviews of Modern Physics. – 2002. - Vol. 74, no. 1. - P. 47-97.
  31. Albert, R., Jeong, H., Barabasi, A., Diameter World Wide Web [Text] / R. Albert, H. Jeong, A. Barabasi; Nature. - 1999. - Vol. 401, no. 6749. –130 p.
  32. Andersson, H., Britton, T. Stochastic Epidemic Models and Their Statistical Analysis [Text] / H. Andersson, T. Britton; Lecture Notes in Statistics. - Springer-Verlag, 2000.
  33. Bace R., Mell P. Special Publication on Intrusion Detection Systems. [Text] / R. Bace, P. Mell; Tech. Report SP 800-31; National Institute of Standards and Technology. – 2001.
  34. Barabasi, R. Albert Emergence of scaling in random networks [Text] / Albert R. Barabasi; Science. - 1999. – P. 509-512.
  35. Barabasi, R. Albert, H. Jeong Physica [Text] / Barabasi, R. Albert, H. Jeong; A 272. - 1999. – P.173.
  36. Barillère, R., Baggiolini, V., Beharell, M., Chmielewski, D., Gras, P., Mil-cent, H., Kostro, K., Khomoutnikov, V. Results of the OPC evaluation done within JCOP for the control of the LHC experiments [Text] / R. Barillère [et al.]; International Conference on Accelerator and Large Experimental Physics Control Systems. - Trieste, Italy, 1999. - P. 511-513.
  37. Blazek, R.B. A Novel Approach to Detection of «Denial-of-

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- =====
- Service» Attacks via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods [Text] / R.V. Blazek; Proc. IEEE Workshop Information Assurance and Security. - IEEE CS Press, 2001. - P. 220–226.
38. Bollobás, B. Random Graphs [Text] / B. Bollobás; Cambridge University Press. - 2001. – 520 p. - ISBN 0521809207.
39. Chwe, M.S. Communication and Coordination in Social Network [Text] / M.S. Chwe; Review of Economic Studies, № 67. - 2000. - P.1-16.
40. Cohen, F. Simulating Cyber Attacks, Defenses, and Consequences [Text] / F. Cohen; IEEE Symposium on Security and Privacy. - Berkeley, 1999.
41. Cohen, R., Havlin, S. Scale-free networks are ultrasmall [Text] / R. Cohen, S. Havlin; Phys. Rev. Lett., 90. - 2003.
42. Deszo, Z., Barabasi, A.L. Halting viruses in scale free networks [Electronic resource] / Z. Deszo, A.L. Barabasi. Access mode: [http://www.arxiv.org/PS\\_cache/cond-mat/pdf/0107/0107420.pdf](http://www.arxiv.org/PS_cache/cond-mat/pdf/0107/0107420.pdf).
43. Dorogovtsev, S.N., Mendes, J.F.F. Scaling properties of scale-free evolving networks: continuous approach [Text] / S.N. Dorogovtsev, J.F.F. Mendes; Phys. Rev., E 63. - 2001.
44. Dorogovtsev, S.N., Mendes, J.F.F. Evolution of Networks: From Biological Networks to the Internet and WWW [Text] / S.N. Dorogovtsev, J.F.F. Mendes; - Oxford, USA: Oxford University Press, 2003. — 280 p. - SBN 978-0198515906.
45. Dorogovtsev, S.N., Mendes, J.F.F., Samukhin, A. N. Giant strongly connected component of directed networks [Text] / S.N. Dorogovtsev, J.F.F. Mendes, A. N. Samukhin; Phys. Rev., E 64. - 2001.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

46. Dorogovtsev, S.N., Mendes, J.F.F., Samukhin, A.N. Structure of Growing Networks: Exact Solution of the Barabasi Albert's Model [Text] / S.N. Dorogovtse, J.F.F. Mendes, A. N. Samukhin; Phys. Rev. Lett. 85. - 2000.
47. Easley, D., Kleinberg, J., Networks, Crowds, and Markets Reasoning About a Highly Connected World [Text] / D. Easley, J. Kleinberg; - 2010.
48. Erdős, P., Rényi, A. On the evolution of random graphs [Text] / P. Erdős, A. Rényi; Publications of the Mathematical Institute of the Hungarian Academy of Sciences, 5. - 1960. – P. 17-61.
49. Erdős, P., Rényi, A. On random graphs [Text] / P. Erdős, A. Rényi; Publicationes Mathematicae. - 1959. -Vol. 6, no. 26. - P. 290-297.
50. Espinoza, Vicente, Social Networks Among the Urban Poor, in Networks in the Global Village [Text] / Espinoza, Vicente; - 1999.
51. Ferrara, E., Fiumara, G., Topological features of Online Social Networks. Communications on Applied and Industrial Mathematics [Text] / E. Ferrara, G. Fiumara; - 2011.
52. Garetto, M., Gong, W., Towsley, D. Modeling Malware Spreading Dynamics [Text] / M. Garetto, W. Gong, D. Towsley; Proc. of 22nd Annual Joint Conference of the IEEE Computer, Communications societies (INFOCOM) (March-April 2003) – 2003.
53. Gjoka, M., Kurant, M., Butts, C. T., Markopoulou, A. A Walk in Facebook: Uniform Sampling of Users in Online Social Networks [Text] / M. Gjoka [et al.]; IEEE INFOCOM '10. IEEE Journal on Selected Areas in Communications - 2010.
54. Gjoka, M., Kurant, M., Butts, C.T., Markopoulou, A. Multigraph Sampling of Online Social Networks [Text] / M.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

- Gjoka [et al.]; IEEE J. Sel. Areas Commun. on Measurement of Internet Topologies - 2011.
55. Gjoka, M., Kurant, M., Butts, C.T., Markopoulou, A. Walking on a Graph with a Magnifying Glass: Stratified Sampling via Weighted Random Walks [Text] / M. Gjoka [et al.]; - Sigmetrics, 2011.
  56. Gjoka, M., Kurant, M., Butts, C. T., Markopoulou, A. Walking in Facebook: A Case Study of Unbiased Sampling of OSNs [Text] / M. Gjoka, M. Kurant, C.T. Butts, A. Markopoulou; IEEE INFOCOM (San Diego, CA, 2010) – 2010.
  57. Gjoka, M., Kurant, M., Wang, Y., Almquist, Z.W., Butts, C.T., Markopoulou, A. Coarse-Grained Topology Estimation via Graph Sampling [Electronic resource]: Arxiv preprint / M. Gjoka [et al.]; - 2011. -arXiv:1105.5488
  58. Gjoka, M., Sirivianos, M., Markopoulou, A., Yang, X. Poking facebook: characterization of osn applications [Text] / M. Gjoka [et al.]; Proc. of WOSN - 2008.
  59. Golbeck, J., Hendler, J. Inferring binary trust relationships in web-based social networks [Text] / J. Golbeck, J. Hendler; Transactions on Internet Technology - 2006. - Vol. 6, no. 4. - P. 497-529.
  60. Goldenberg, J., Libai, B., Muller, E. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth [Text] / J. Goldenberg, B. Libai, E. Muller; Marketing Letters - 2001. - № 2. - P. 11-34.
  61. Granovetter, M. The strength of weak ties [Text] / M. Granovetter; American Journal of Sociology - 1973. – Vol. 78. – P. 1360–1380.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- =====  
62. Granovetter, M. Threshold Models of Collective Behavior [Text] / M. Granovetter; American Journal of Sociology - 1978. – Vol. 83, no. 6. – P. 1420–1443.
63. Grimaldi, R. P. Discrete and Combinatorial Mathematics [Text] / R.P. Grimaldi; an applied introduction. - 4th edition. - New York, 1998.
64. Heberlein, L.T., Dias, G.V., Levitt, K.N., Mukherjee, B., Wood, J., Wolber, D.A. Network security monitor [Text] / L.T. Heberlein [et al.]; Proc. of IEEE Symposium on Re-search in Security and Privacy. – Los Alamitos, CA, USA: IEEE Computer Soci-ety, 1990. - P. 296–304.
65. Hethcote, H.W. The Mathematics of Infectious Diseases [Text] / H.W. Hethcote; - 2000. - P. 599-653,
66. Hofmeyr, S.A., Forrest, S., Somayaji, A. Intrusion detection using se-quences of system calls [Text] / S.A. Hofmeyr, S. Forrest, A. Somayaji; Journal of Computer Security. - Amsterdam: IOS Press, 1998. – Vol. 6, no 3. - P. 151-180.
67. Kenah, E., Robins, J. M. Network-based analysis of stochastic SIR epidemic models with random and proportionate mixing [Text] / E. Kenah, J. M. Robins; Departments of Epidemiology and Biostatistics Harvard School of Public Health. - 2007.
68. Kephart, J.O., White, S.R. Directed-Graph Epidemiological Models of Computer Viruses [Text] / J.O. Kephart, S.R. White; Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy. -1991. P. 343 - 359.
69. Kolotov, A. Мониторинг сети с помощью tcpdump [Electronic resource] / A. Kolotov; - Access mode: <http://www.linuxshare.ru/docs/net/tcpdump.html>

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- =====  
70. Kulkarni, R.V., Almaas, E., Stroud, D. Evolutionary dynamics in the Bak-Sneppen model on small-world networks [Text] / R.V. Kulkarni, E. Almaas, D. Stroud; - 2008.
71. Kumar, R., Novak, J., Tomkins, A. Structure and evolution of online social networks [Text] / R. Kumar, J. Novak, A. Tomkins; Link Mining: Models, Algorithms, and Applications. - 2010. - P. 337-357.
72. Kurant, M., Markopoulou, A., Thiran, P. On the bias of BFS (Breadth First Search) [Text] / M. Kurant, A. Markopoulou, P. Thiran; in Proc. 22nd Int. Teletraffic Congr.; - 2010. - arXiv:1004.1729.
73. Leskovec, J., Faloutsos, C., Sampling from large graphs [Text] / J. Leskovec, C. Faloutsos; 12<sup>th</sup> International Conference on Knowledge Discovery and Data Mining. - 2006. - P. 631-636.
74. Leskovec, J., Adamic, L.A., Huberman, B.A. The Dynamics of Viral Marketing [Text] / J. Leskovec, L.A. Adamic, B.A. Huberman; HP Labs Palo Alto, CA 94304. - 2008.
75. Leskovec, J., Kleinberg, J., Faloutsos, C. Graphs over time: densification laws, shrinking diameters and possible explanations [Text] / J. Leskovec, J. Kleinberg, C. Faloutsos; in Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. - ACM, 2005. - P. 177-187.
76. Leveille, J. Epidemic Spreading in Technological Networks [Text] / J. Leveille; Information Infrastructure Laboratory HP Laboratories Bristol. - 2002. – P. 65-76.
77. Liben-Nowell D., Kleinberg J. The link-prediction problem for social networks [Text] / D. Liben-Nowell, J. Kleinberg; J. American Society for Information Science and Technology. - 2007. - Vol. 58, no. 7. - P. 1019-1031.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- =====  
78. Mislove, A., Marcon, M., Gummadi, K., Druschel, P., Bhattacharjee, B. Measurement and analysis of online social networks [Text] / A. Mislove [et al.]; 7th ACM conference on Internet measurement. – 2007. – P. 29-42.
79. Newman, M.E.J., Strogatz, S.H., Watts, D.J. Random graphs with arbitrary degree distributions, their applications [Text] / M.E.J. Newman, S.H. Strogatz, D.J. Watts; Phys. Rev. - 2001. – Vol. E 64.
80. Newman, M.E.J., Jensen, I., Ziff, R.M. Percolation, epidemics in a two-dimensional small world [Text] / M.E.J. Newman, I. Jensen, R.M. Ziff; Phys.Rev. E 65. - 2002.
81. Newman, M.E.J. The Structure and Function of Complex Networks [Text] / M.E.J. Newman; SIAM REVIEW. - 2003. - Vol. 45, No. 2. - P. 167-256.
82. Newman, M.E.J., Ziff, R.M. Efficient Monte Carlo algorithm high-precision results for percolation [Text] / M.E.J. Newman, R.M. Ziff; Phys. Rev. Lett. 85. – 2000.
83. Newman, Mark, Barabasi, Albert-Laszlo, Duncan, Watts, J. The Structure and Dynamics of Networks: (Princeton Studies in Complexity) [Text] / Mark Newman, Albert-Laszlo Barabasi, Duncan, J. Watts; — Princeton, USA: Princeton University Press, 2006. — 624 p. — ISBN 978-0691113579.
84. Parallel Boost Graph Library [Electronic resource]. – Access mode:[http://www.boost.org/doc/libs/1\\_55\\_0/libs/graph\\_parallel/doc/html/index.html](http://www.boost.org/doc/libs/1_55_0/libs/graph_parallel/doc/html/index.html).
85. Pastor-Satorras, R., Vespignani, A. Epidemic Spreading in Scale-Free Networks [Text] / R. Pastor-Satorras, A. Vespignani; Phys. Rev. Lett., 86. – 2001.
86. Pastor-Satorras, R., Vespignani, A., Absence of epidemic threshold in scale-free networks with connectivity correlations

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

- [Text] / R. Pastor-Satorras, A. Vespignani; Phys. Rev. Lett. – Pub.: American Physical Society, 2002. – Vol. 90, Iss. 2. – P. 1-4.
87. Pastor-Satorras, R., Vespignani, A. Critical load, congestion instabilities in scale-free networks [Text] / R. Pastor-Satorras, A. Vespignani; Europhys. Lett. - 2002. - Vol. 62. - P. 292.
88. Pastor-Satorras, R., Vespignani, A. Dynamical patterns of epidemic outbreaks in complex heterogeneous networks [Text] / R. Pastor-Satorras, A. Vespignani; Journal of Theoretical Biology. - 2005. – P. 275-288.
89. Pastor-Satorras, R., Vespignani, A. Emergence of clustering correlations communities in a social network model [Text] / R. Pastor-Satorras, A. Vespignani; - 2003.
90. Pastor-Satorras, R., Vespignani, A. Epidemic dynamics, endemic states in complex networks [Text] / R. Pastor-Satorras, A. Vespignani; Phys. Rev. E. - 2001.
91. Pastor-Satorras, R., Vespignani, A. Epidemic spreading in complex networks with degree correlations [Text] / R. Pastor-Satorras, A. Vespignani; Contribution to the Proceedings of the XVIII Sitges Conference "Statistical Mechanics of Complex Networks". - Berlin, 2003.
92. Pastor-Satorras, R., Vespignani A. Epidemic spreading in correlated complex networks [Text] / R. Pastor-Satorras, A. Vespignani; Phys. Rev. E Stat. Nonlin. Soft. Matter. Phys. - 2002.
93. Pastor-Satorras, R., Vespignani, A. Immunization of complex networks [Text] / R. Pastor-Satorras, A. Vespignani; Phys. Rev. E. - 2002.
94. Pastor-Satorras, R., Vespignani, A. Large-scale topological, dynamical properties of the Internet [Text] / R. Pastor-Satorras, A. Vespignani; Phys. Rev. E. – 2002. - Vol. 65.

## Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

95. Pastor-Satorras, R., Vespignani, A. Reaction-diffusion processes, meta-population models in heterogeneous networks [Text] / R. Pastor-Satorras, A. Vespignani; Nature Physics 3. - 2007. – P. 276-282.
96. Pastor-Satorras, R., Vespignani, A. Velocity, hierarchical spread of epidemic outbreaks in scale-free networks [Text] / R. Pastor-Satorras, A. Vespignani; Phys. Rev. Lett. – 2004. - Vol. 92. - P. 178-701.
97. Pastor-Satorras, R., Vespignani, A. Dynamical, Correlation Properties of the Internet [Text] / R. Pastor-Satorras, A. Vespignani; Phys. Rev. Lett. – 2001. - Vol. 87, No. 258701.
98. Pastor-Satorras, R., Vespignani, A. Topology, Hierarchy, Correlations in Internet Graphs [Text] / R. Pastor-Satorras, A. Vespignani; Lecture Notes in Physics. - Berlin – Heidelberg: Springer, 2004. – P. 425-440.
99. Sala, A., Zheng, H., Zhao, Gaito, S., Rossi Brief announcement: revisiting the power-law degree distribution for social graph analysis [Text] / A. Sala, H. Zheng, Zhao, S. Gaito, Rossi; PODC. – 2010. – P. 400-401.
100. Tarnow, E. Like Water and Vapor, Conformity and Independence in the Large Group [Electronic resource] / E. Tarnow; Access mode: URL: <http://cogprints.org/4274/1/LargeGroupOrderTarnow.pdf>.
101. Tictrac [Electronic resource] / Access mode: <https://www.tictrac.com>
102. Ugander, J., Karrer, B., Backstrom, L., Marlow, K. The Anatomy of the Facebook Social Graph [Text] / J. Ugander, B. Karrer, L. Backstrom, K. Marlow; CoRR. - 2011.
103. Volz, E. SIR dynamics in random networks with heterogeneous connectivity [Text] / E. Volz; Journal of Mathematical Biology manuscript. - 2007.

Глава 2. ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ  
В ПРАКТИЧЕСКИХ ЗАДАЧАХ

=====

104. Wang, H., Guo, Y. Consensus on scale-free network [Text] / H. Wang, Y. Guo; American Control Conference. - 2008. - P.748 – 752.
105. Watts, D., Strogatz, S. Collective dynamics of small-world networks [Text] / D. Watts, S. Strogatz; Nature. – 1998. - Vol. 393, No. 6684. - P. 440-442.
106. Williamson, M.M., Léveillé, J. An epidemiological model of virus spread and cleanup [Text] / M.M. Williamson, J. Léveillé; Information Infrastructure Laboratory HP Laboratories Bristol HPL. - 2003.
107. Zhang D., Gatica-Perez D., Bengio S., Roy D. Learning Influence among Interacting Markov Chains [Text] / D. Zhang, D. Gatica-Perez, S. Bengio, D. Roy; Neural Information Processing Systems (NIPS). -2005. - P. 132-141

=====

## **ЗАКЛЮЧЕНИЕ**

Пособие дает практические навыки программирования по реализации алгоритмов на графах и кодовых деревьях, необходимых для специалистов по информационным технологиям и защите компьютерной информации и имеющих широкое применение в прикладных задачах программирования (синтаксический разбор в вычислении математических выражений, построение сбалансированных деревьев для оптимизации, сортировки и поиска, построение кодовых деревьев для создания экономных кодов сжатия без потерь).

Читателей, желающих глубже изучить материалы по теории графов и ее применение в практических задачах, автор отсылает к приведенным спискам рекомендуемых источников.

=====

## ПРИЛОЖЕНИЕ

### Ответы к заданиям

#### 1. Обходы графов в ширину и глубину

	1	2	3	4	5	6
1	2, 3		4, 5			
2	2, 3	4, 5	6, 7			
3	2, 3	4	5, 6			
4	2, 3	4	5, 6			7
5	2, 3	4, 5				
6	2, 3	4, 5			6, 7	
7	2		1, 4, 5			
8	2	4, 5	1, 6, 7			
9	2	4	1, 5, 6			
10	2	4	1, 5, 6			7
11	2	4, 5	1			
12	2	4, 5	1		6, 7	
13	3	1	4, 5			
14	3	1, 4, 5	6, 7			
15	3	1, 4	5, 6			7
16	3	1, 4, 5			6, 7	
17	3	1, 4	5, 6			
18	3	1, 4, 5				
19	2		1, 5	3		
20	3	1, 5	6, 7	2		
21	3	1	5, 6	2		
22	3	1	5, 6	2		7
23	3	1, 5		2		
24	3	1, 5		2	6, 7	
25	2, 3	4, 5			6, 7, 8	
26	2, 3	4, 5, 6				
27	2, 3	4, 5, 6	7			
28	2, 3, 7	4, 5, 6				
29	2, 7, 3	4, 5, 6	8			
30	2, 3	4, 5, 6	7, 8			

## ПРИЛОЖЕНИЕ

---

### 2. Алгоритм Тэрри

1. 3 – 2 – 4 – 1 – 5
2. 1 – 2 – 3 – 5 – 4
3. 5 – 2 – 4 – 1 – 3
4. 1 – 2 – 5 – 4 – 3
5. 2 – 5 – 4 – 3 – 1
6. 2 – 1 – 3 – 4 – 5
7. 4 – 3 – 2 – 5 – 1
8. 1 – 2 – 3 – 5 – 4
9. 3 – 5 – 4 – 2 – 1
10. 5 – 3 – 4 – 2 – 1
11. 1 – 3 – 2 – 4 – 5 – 6
12. 4 – 3 – 5 – 6 – 1 – 2
13. 5 – 4 – 3 – 6 – 1 – 2
14. 2 – 3 – 4 – 6 – 5 – 1
15. 6 – 4 – 5 – 3 – 2 – 1
16. 2 – 1 – 3 – 4 – 5 – 6
17. 6 – 5 – 2 – 1 – 3 – 4
18. 3 – 2 – 5 – 6 – 4 – 1
19. 5 – 6 – 2 – 4 – 3 – 1
20. 1 – 2 – 5 – 6 – 4 – 3
21. 1 – 2 – 3 – 4 – 5 – 6
22. 3 – 1 – 2 – 4 – 5 – 6
23. 4 – 3 – 5 – 1 – 2 – 6
24. 6 – 5 – 4 – 3 – 2 – 1
25. 2 – 3 – 5 – 6 – 4 – 1
26. 2 – 1 – 3 – 4 – 5
27. 5 – 3 – 4 – 1 – 2
28. 1 – 3 – 5 – 4 – 2
29. 3 – 5 – 4 – 2 – 1
30. 4 – 5 – 3 – 1 – 2

## ПРИЛОЖЕНИЕ

### 3. Матроиды, жадные алгоритмы, алгоритм Краскала

1. (2,5) (1,4) (3,4) (4,5)
2. (3,4) (4,5) (7,8) (4,7) (5,6) (2,3) (1,6)
3. (2,6) (4,5) (4,6) (3,4) (1,3)
4. (1,2) (3,6) (2,6) (4,5) (3,4)
5. (1,2) (6,8) (7,8) (1,8) (2,3) (5,6) (3,4)
6. (2,3) (1,4) (2,4) (1,5)
7. (1,6) (3,4) (4,5) (4,7) (5,6) (1,2) (7,8)
8. (1,2) (2,6) (4,5) (3,4) (5,6)
9. (2,6) (4,5) (4,6) (3,4) (1,3)
10. (1,8) (2,3) (3,4) (2,8) (6,8) (7,8) (4,5)
11. (2,4) (1,5) (2,3) (2,5)
12. (3,4) (7,8) (1,6) (3,8) (5,6) (2,3) (6,7)
13. (2,6) (4,6) (1,2) (1,3) (5,6)
14. (1,3) (3,6) (5,6) (2,6) (4,6)
15. (2,4) (5,6) (6,8) (1,8) (3,4) (1,2) (7,8)
16. (1,4) (1,5) (1,2) (2,3)
17. (4,5) (5,6) (7,8) (1,2) (4,7) (1,6) (2,3)
18. (3,4) (4,5) (4,6) (2,3) (1,2)
19. (2,6) (4,6) (1,2) (1,3) (5,6)
20. (1,8) (3,4) (1,2) (2,4) (5,6) (6,8) (7,8)
21. (1,3) (4,5) (2,4) (1,4)
22. (2,3) (3,8) (3,4) (4,7) (1,6) (4,5) (1,8)
23. (1,3) (5,6) (2,6) (4,6) (1,2)
24. (2,3) (1,2) (5,6) (2,6) (4,5)
25. (6,7) (1,2) (5,6) (1,8) (2,3) (4,5) (6,8)
26. (1,2) (2,5) (1,3) (2,4)
27. (1,2) (4,7) (1,6) (2,3) (3,4) (4,5) (1,8)
28. (2,3) (1,2) (5,6) (2,6) (4,5)
29. (1,3) (5,6) (2,6) (4,6) (1,2)
30. (2,4) (5,6) (1,8) (3,4) (1,2) (4,5) (7,8)

## ПРИЛОЖЕНИЕ

### **4. Нахождение кратчайшего пути в сети без контуров**

1. Правильная нумерация: 0-0, 2-1, 3-2, 4-3, 1-4.  
Потенциалы: 0-0, 1-1, 2-5, 3-6, 4-8. Кратчайший путь: 0 1 4.
2. Правильная нумерация: 0-0, 1-1, 2-2, 3-3, 4-4.  
Потенциалы: 0-0, 1-1, 2-9, 3-5, 4-11. Кратчайший путь: 0 1 3 4.
3. Правильная нумерация: 1-0, 2-1, 3-2, 1-3, 4-4.  
Потенциалы: 0-0, 1-8, 2-4, 3-13, 4-7. Кратчайший путь: 0 2 4.
4. Правильная нумерация: 0-0, 2-1, 3-2, 1-3, 4-4.  
Потенциалы: 0-0, 1-8, 2-14, 3-10, 4-3. Кратчайший путь: 0 4.
5. Правильная нумерация: 2-0, 0-1, 1-2, 3-3, 4-4.  
Потенциалы: 0-0, 1-4, 2-1, 3-8, 4-6. Кратчайший путь: 0 2 4.
6. Правильная нумерация: 3-0, 0-1, 1-2, 2-3, 4-4.  
Потенциалы: 0-0, 1-6, 2-9, 3-4, 4-6. Кратчайший путь: 0 3 4.
7. Правильная нумерация: 2-0, 1-1, 3-2, 4-3, 0-4.  
Потенциалы: 0-0, 1-4, 2-12, 3-2, 4-8. Кратчайший путь: 0 3 4.
8. Правильная нумерация: 2-0, 0-1, 1-2, 3-3, 4-4.  
Потенциалы: 0-0, 1-9, 2-7, 3-16, 4-10. Кратчайший путь: 0 2 4.
9. Правильная нумерация: 1-0, 3-1, 0-2, 4-3, 2-4.  
Потенциалы: 0-0, 1-7, 2-13, 3-3, 4-5. Кратчайший путь: 0 3 4.
10. Правильная нумерация: 1-0, 0-1, 3-2, 2-3, 4-4.  
Потенциалы: 0-0, 1-2, 2-4, 3-5, 4-5. Кратчайший путь: 0 4.
11. Правильная нумерация: 1-0, 0-1, 3-2, 4-3, 2-4.  
Потенциалы: 0-0, 1-6, 2-1, 3-3, 4-3. Кратчайший путь: 0 2 4.
12. Правильная нумерация: 2-0, 3-1, 4-2, 0-3, 1-4 .  
Потенциалы: 0-0, 1-7, 2-1, 3-2, 4-5. Кратчайший путь: 0 2 4.
13. Правильная нумерация: 1-0, 0-1, 3-2, 4-3, 2-4.  
Потенциалы: 0-0, 1-4, 2-1, 3-2, 4-7. Кратчайший путь: 0 1 4.
14. Правильная нумерация: 1-0, 3-1, 0-2, 2-3, 4-4.  
Потенциалы: 0-0, 1-4, 2-2, 3-5, 4-8. Кратчайший путь: 0 2 4.
15. Правильная нумерация: 4-0, 0-1, 1-2, 2-3, 3-4.  
Потенциалы: 0-0, 1-3, 2-10, 3-7, 4-9. Кратчайший путь: 0 1 3 4.
16. Правильная нумерация: 2-0, 0-1, 5-2, 6-3, 1-4, 3-5, 4-6.  
Потенциалы: 0-0, 1-6, 2-9, 3-7, 4-8, 5-15, 6-5. Кратчайший путь: 0 6.

## ПРИЛОЖЕНИЕ

- =====  
**17.** Правильная нумерация: 4-0, 1-1, 2-2, 3-3, 5-4, 0-5, 6-6.  
Потенциалы: 0-0, 1-4, 2-6, 3-8, 4-11, 5-8, 6-13. Кратчайший путь:  
0 1 6.
- 18.** Правильная нумерация: 4-0, 0-1, 2-2, 5-3, 3-4, 1-5, 6-6.  
Потенциалы: 0-0, 1-9, 2-2, 3-16, 4-19, 5-7, 6-27. Кратчайший путь:  
0 1 3 4 6.
- 19.** Правильная нумерация: 0-0, 1-1, 2-2, 3-3, 4-4, 5-5, 6-6.  
Потенциалы: 0-0, 1-1, 2-2, 3-6, 4-9, 5-6, 6-9. Кратчайший путь:  
0 6.
- 20.** Правильная нумерация: 3-0, 0-1, 1-2, 2-3, 4-4, 6-5, 5-6.  
Потенциалы: 0-0, 1-4, 2-5, 3-5, 4-1, 5-3, 6-8. Кратчайший путь:  
0 2 4 6.
- 21.** Правильная нумерация: 1-0, 0-1, 2-2, 3-3, 4-4, 5-5, 6-6.  
Потенциалы: 0-0, 1-6, 2-4, 3-11, 4-7, 5-2, 6-14. Кратчайший путь:  
0 1 6.
- 22.** Правильная нумерация: 3-0, 0-1, 1-2, 5-3, 2-4, 4-5, 6-6.  
Потенциалы: 0-0, 1-4, 2-8, 3-9, 4-10, 5-5, 6-7. Кратчайший путь: 0  
1 6.
- 23.** Правильная нумерация: 0-0, 1-1, 3-2, 4-3, 5-4, 2-5, 6-6.  
Потенциалы: 0-0, 1-3, 2-7, 3-1, 4-8, 5-6, 6-6. Кратчайший путь:  
0 6.
- 24.** Правильная нумерация: 3-0, 0-1, 1-2, 2-3, 4-4, 6-5, 5-6.  
Потенциалы: 0-0, 1-6, 2-4, 3-7, 4-6, 5-3, 6-8. Кратчайший путь:  
0 5 6.
- 25.** Правильная нумерация: 6-0, 0-1, 3-2, 1-3, 4-4, 2-5, 5-6.  
Потенциалы: 0-0, 1-1, 2-9, 3-4, 4-5, 5-6, 6-8. Кратчайший путь:  
0 1 6.
- 26.** Правильная нумерация: 4-0, 0-1, 1-2, 2-3, 3-4, 6-5, 5-6.  
Потенциалы: 0-0, 1-2, 2-8, 3-4, 4-5, 5-9, 6-12. Кратчайший путь:  
0 1 5 6.
- 27.** Правильная нумерация: 5-0, 0-1, 1-2, 2-3, 3-4, 4-5, 6-6.  
Потенциалы: 0-0, 1-2, 2-4, 3-1, 4-7, 5-5, 6-12. Кратчайший путь:  
0 2 6.

## ПРИЛОЖЕНИЕ

- =====
- 28.** Правильная нумерация: 0-0, 1-1, 2-2, 3-3, 4-4, 5-5, 6-6.  
Потенциалы: 0-0, 1-1, 2-4, 3-5, 4-6, 5-11, 6-4. Кратчайший путь:  
0 1 6.
- 29.** Правильная нумерация: 1-0, 0-1, 2-2, 3-3, 4-4, 5-5, 6-6.  
Потенциалы: 0-0, 1-1, 2-6, 3-9, 4-5, 5-7, 6-5. Кратчайший путь:  
0 1 6.
- 30.** Правильная нумерация: 1-0, 0-1, 2-2, 3-3, 5-4, 6-5, 4-6.  
Потенциалы: 0-0, 1-5, 2-2, 3-8, 4-14, 5-1, 6-9. Кратчайший путь:  
0 5 6.

### 5. Алгоритм Беллмана-Форда

1. 0 – 1 – 2 – 3 – 5
2. 0 – 2 – 3 – 5
3. 0 – 1 – 3 – 2 – 4 – 5
4. 0 – 2 – 1 – 3 – 5
5. 0 – 2 – 4 – 5
6. 0 – 2 – 5 – 3 – 6
7. 0 – 2 – 1 – 5 – 3 – 4 – 6
8. 0 – 1 – 5 – 3 – 6
9. 0 – 2 – 5 – 3 – 4 – 6
10. 0 – 2 – 1 – 5 – 4 – 6
11. 0 – 2 – 1 – 3 – 4 – 6 – 5 – 7
12. 0 – 1 – 3 – 4 – 6 – 7
13. 0 – 1 – 3 – 5 – 7
14. 0 – 2 – 4 – 6 – 5 – 7
15. 0 – 2 – 1 – 3 – 5 – 7
16. 0 – 3 – 6
17. 0 – 1 – 3 – 4 – 6
18. 0 – 3 – 4 – 6
19. 0 – 2 – 5 – 6
20. 0 – 2 – 3 – 4 – 6
21. 0 – 1 – 2 – 3 – 4 – 5 – 6
22. 0 – 2 – 3 – 4 – 6
23. 0 – 1 – 2 – 4 – 5 – 6

## ПРИЛОЖЕНИЕ

---

**24.** 0 – 2 – 4 – 6

**25.** 0 – 1 – 2 – 3 – 4 – 6

**26.** 0 – 2 – 3 – 1 – 4 – 6

**27.** 0 – 1 – 4 – 3 – 5 – 6

**28.** 0 – 2 – 3 – 5 – 6

**29.** 0 – 1 – 4 – 6

**30.** 0 – 2 – 5 – 6

### **6. Алгоритм Дейкстра**

**1.** 1 → 4 → 5

Длина пути: 7

**2.** 1 → 2 → 4 → 3 → 5

Длина пути: 12

**3.** 1 → 2 → 3 → 5

Длина пути: 7

**4.** 1 → 4 → 3 → 5

Длина пути: 6

**5.** 1 → 3 → 5

Длина пути: 5

**6.** 1 → 2 → 5

Длина пути: 7

**7.** 1 → 4 → 5

Длина пути: 5

**8.** 1 → 3 → 5

Длина пути: 3

**9.** 1 → 2 → 5

Длина пути: 4

**10.** 1 → 2 → 4 → 5

Длина пути: 7

**11.** 1 → 4 → 5

Длина пути: 5

**12.** 1 → 2 → 3 → 5

Длина пути: 4

**13.** 1 → 4 → 3 → 5

Длина пути: 6

## ПРИЛОЖЕНИЕ

---

**14.**  $1 \rightarrow 2 \rightarrow 5$

Длина пути: 2

**15.**  $1 \rightarrow 4 \rightarrow 3 \rightarrow 5$

Длина пути: 7

**16.**  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$

Длина пути: 11

**17.**  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$

Длина пути: 5

**18.**  $1 \rightarrow 5$

Длина пути: 1

**19.**  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$

Длина пути: 5

**20.**  $1 \rightarrow 2 \rightarrow 5$

Длина пути: 2

**21.**  $1 \rightarrow 3 \rightarrow 4 \rightarrow 5$

Длина пути: 10

**22.**  $1 \rightarrow 5$

Длина пути: 3

**23.**  $1 \rightarrow 3 \rightarrow 5$

Длина пути: 5

**24.**  $1 \rightarrow 4 \rightarrow 5$

Длина пути: 4

**25.**  $1 \rightarrow 2 \rightarrow 5$

Длина пути: 6

**26.**  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

Длина пути: 11

**27.**  $1 \rightarrow 4 \rightarrow 2 \rightarrow 5$

Длина пути: 8

**28.**  $1 \rightarrow 3 \rightarrow 5$

Длина пути: 2

**29.**  $1 \rightarrow 4 \rightarrow 5$

Длина пути: 3

**30.**  $1 \rightarrow 5$

Длина пути: 1

ПРИЛОЖЕНИЕ

7. Алгоритм Флойда – Уоршалла

Таблица А1

Номер варианта	Матрица кратчайших расстояний $T_{ij}$	Матрица маршрутов $H_{ij}$
1	$\begin{pmatrix} 0 & 9 & 7 & 3 & 9 \\ \infty & 0 & 3 & \infty & 1 \\ \infty & \infty & 0 & \infty & 2 \\ \infty & \infty & 4 & 0 & 6 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 4 & 4 & 4 \\ 0 & 0 & 3 & 0 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
2	$\begin{pmatrix} 0 & 8 & 14 & 6 & 1 \\ \infty & 0 & \infty & \infty & 3 \\ \infty & \infty & 0 & \infty & \infty \\ \infty & 2 & 8 & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 4 & 4 & 4 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
3	$\begin{pmatrix} 0 & 1 & 8 & 1 & 7 \\ \infty & 0 & 7 & \infty & 10 \\ \infty & \infty & 0 & \infty & 3 \\ \infty & \infty & \infty & 0 & 6 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 4 & 4 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
4	$\begin{pmatrix} 0 & 9 & 10 & 8 & 11 \\ \infty & 0 & 1 & \infty & 2 \\ \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & 5 & 0 & 6 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 4 & 3 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

ПРИЛОЖЕНИЕ

Продолжение таблицы А1

5	$\begin{pmatrix} 0 & 4 & 6 & 1 & 3 \\ \infty & 0 & \infty & \infty & 3 \\ \infty & \infty & 0 & \infty & \infty \\ \infty & 3 & 5 & 0 & 2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 4 & 4 & 4 & 4 \\ 0 & 0 & 3 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
6	$\begin{pmatrix} 0 & 5 & 15 & 10 & 16 \\ 14 & 0 & 10 & 5 & 11 \\ 4 & 9 & 0 & 14 & 1 \\ 9 & 14 & 5 & 0 & 6 \\ 3 & 8 & 18 & 13 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 4 & 2 & 4 \\ 5 & 0 & 4 & 4 & 4 \\ 5 & 5 & 0 & 5 & 5 \\ 5 & 5 & 3 & 0 & 3 \\ 1 & 1 & 4 & 2 & 0 \end{pmatrix}$
7	$\begin{pmatrix} 0 & 3 & 5 & 9 & 10 \\ \infty & 0 & 2 & 6 & 7 \\ \infty & 7 & 0 & 5 & 6 \\ \infty & 2 & 4 & 0 & 1 \\ \infty & 1 & 3 & 7 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 2 & 4 \\ 0 & 0 & 3 & 4 & 4 \\ 0 & 5 & 0 & 4 & 4 \\ 0 & 5 & 5 & 0 & 5 \\ 0 & 2 & 2 & 2 & 0 \end{pmatrix}$
8	$\begin{pmatrix} 0 & 7 & 1 & 2 & 2 \\ \infty & 0 & \infty & \infty & 8 \\ \infty & \infty & 0 & \infty & 1 \\ \infty & 5 & 9 & 0 & 10 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 4 & 3 & 4 & 3 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 2 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

## ПРИЛОЖЕНИЕ

Продолжение таблицы А1

9	$\begin{pmatrix} 0 & 5 & 6 & 9 & 8 \\ 6 & 0 & 1 & 4 & 3 \\ 5 & 10 & 0 & 14 & 2 \\ 7 & 12 & 2 & 0 & 4 \\ 3 & 8 & 9 & 12 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 2 & 3 \\ 5 & 0 & 3 & 4 & 3 \\ 5 & 5 & 0 & 5 & 5 \\ 5 & 5 & 3 & 0 & 3 \\ 1 & 1 & 2 & 2 & 0 \end{pmatrix}$
10	$\begin{pmatrix} 0 & 1 & 4 & 5 & 1 \\ \infty & 0 & 3 & 4 & 5 \\ \infty & 7 & 0 & 5 & 6 \\ \infty & 2 & 5 & 0 & 1 \\ \infty & 1 & 4 & 5 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 2 & 5 \\ 0 & 0 & 3 & 4 & 4 \\ 0 & 5 & 0 & 4 & 4 \\ 0 & 5 & 5 & 0 & 5 \\ 0 & 2 & 2 & 2 & 0 \end{pmatrix}$
11	$\begin{pmatrix} 0 & 6 & 6 & 1 & 9 \\ \infty & 0 & \infty & \infty & 4 \\ \infty & \infty & 0 & \infty & \infty \\ \infty & 5 & 5 & 0 & 9 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 4 & 4 & 4 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
12	$\begin{pmatrix} 0 & 2 & 6 & 3 & 7 \\ 11 & 0 & 7 & 4 & 8 \\ 4 & 6 & 0 & 7 & 1 \\ 7 & 9 & 3 & 0 & 4 \\ 3 & 5 & 9 & 6 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 4 & 4 & 4 \\ 5 & 0 & 4 & 4 & 4 \\ 5 & 5 & 0 & 5 & 5 \\ 5 & 5 & 3 & 0 & 3 \\ 1 & 1 & 4 & 1 & 0 \end{pmatrix}$

## ПРИЛОЖЕНИЕ

Продолжение таблицы А1

13	$\begin{pmatrix} 0 & 3 & 7 & 9 & 11 \\ 6 & 0 & 4 & 6 & 8 \\ 2 & 5 & 0 & 2 & 4 \\ 3 & 6 & 10 & 0 & 14 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 3 & 3 \\ 3 & 0 & 3 & 3 & 3 \\ 1 & 1 & 0 & 4 & 5 \\ 1 & 1 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
14	$\begin{pmatrix} 0 & 9 & 18 & 16 & 11 \\ 10 & 0 & 9 & 7 & 2 \\ 9 & 18 & 0 & 25 & 1 \\ 11 & 20 & 2 & 0 & 3 \\ 8 & 17 & 26 & 4 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 4 & 2 & 2 \\ 5 & 0 & 4 & 4 & 5 \\ 5 & 5 & 0 & 5 & 5 \\ 5 & 5 & 3 & 0 & 3 \\ 1 & 1 & 4 & 2 & 0 \end{pmatrix}$
15	$\begin{pmatrix} 0 & \infty & 2 & 5 & 3 \\ 13 & 0 & 6 & 4 & 7 \\ 7 & \infty & 0 & 12 & 1 \\ 9 & \infty & 2 & 0 & 3 \\ 6 & \infty & 8 & 11 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 3 & 4 & 3 \\ 5 & 0 & 4 & 4 & 4 \\ 5 & 0 & 0 & 5 & 5 \\ 5 & 0 & 3 & 0 & 3 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$
16	$\begin{pmatrix} 0 & 1 & 3 & 5 & 2 \\ \infty & 0 & \infty & 4 & 1 \\ \infty & \infty & 0 & 2 & 7 \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 3 & 2 & 2 \\ 0 & 0 & 0 & 4 & 5 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

## ПРИЛОЖЕНИЕ

Продолжение таблицы А1

17	$\begin{pmatrix} 0 & 1 & 4 & 2 & 7 \\ \infty & 0 & 3 & 1 & 6 \\ \infty & \infty & 0 & \infty & 4 \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 2 & 4 \\ 0 & 0 & 3 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
18	$\begin{pmatrix} 0 & 1 & 10 & \infty & 14 \\ \infty & 0 & 9 & \infty & 13 \\ \infty & \infty & 0 & \infty & 4 \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 0 & 3 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
19	$\begin{pmatrix} 0 & 9 & \infty & 2 & 14 \\ 8 & 0 & \infty & 10 & 5 \\ 7 & 16 & 0 & 9 & 4 \\ 15 & 7 & \infty & 0 & 12 \\ 3 & 12 & \infty & 5 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 4 & 3 & 4 & 4 \\ 5 & 0 & 3 & 5 & 5 \\ 5 & 5 & 0 & 5 & 5 \\ 5 & 2 & 0 & 0 & 2 \\ 1 & 4 & 0 & 2 & 0 \end{pmatrix}$
20	$\begin{pmatrix} 0 & \infty & 7 & 2 & 4 \\ \infty & 0 & 11 & 6 & 8 \\ \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & 5 & 0 & 2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 4 & 4 & 4 \\ 0 & 0 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 3 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

## ПРИЛОЖЕНИЕ

Продолжение таблицы А1

21	$\begin{pmatrix} 0 & 5 & 1 & 2 & 5 \\ 8 & 0 & 9 & 10 & 5 \\ 7 & 12 & 0 & 9 & 4 \\ 11 & 3 & 12 & 0 & 8 \\ 3 & 8 & 4 & 5 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 4 & 3 & 4 & 3 \\ 5 & 0 & 5 & 5 & 5 \\ 5 & 5 & 0 & 5 & 5 \\ 5 & 2 & 5 & 0 & 2 \\ 1 & 4 & 1 & 1 & 0 \end{pmatrix}$
22	$\begin{pmatrix} 0 & 1 & 2 & 7 & 4 \\ \infty & 0 & 1 & 6 & 3 \\ \infty & \infty & 0 & 5 & 2 \\ \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 3 & 3 \\ 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \end{pmatrix}$
23	$\begin{pmatrix} 0 & 1 & 3 & 2 & 2 \\ \infty & 0 & 3 & \infty & 1 \\ \infty & \infty & 0 & \infty & 4 \\ \infty & \infty & 1 & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 4 & 4 & 2 \\ 0 & 0 & 3 & 0 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
24	$\begin{pmatrix} 0 & 1 & 4 & \infty & 4 \\ \infty & 0 & 3 & \infty & 3 \\ \infty & \infty & 0 & \infty & 4 \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 0 & 2 \\ 0 & 0 & 3 & 0 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

## ПРИЛОЖЕНИЕ

Продолжение таблицы А1

25	$\begin{pmatrix} 0 & \infty & 5 & 3 & 5 \\ 9 & 0 & 6 & 4 & 6 \\ 4 & \infty & 0 & 7 & 1 \\ 5 & \infty & 2 & 0 & 2 \\ 3 & \infty & 8 & 6 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 4 & 4 & 4 \\ 5 & 0 & 4 & 4 & 4 \\ 5 & 0 & 0 & 5 & 5 \\ 5 & 0 & 3 & 0 & 5 \\ 1 & 0 & 4 & 1 & 0 \end{pmatrix}$
26	$\begin{pmatrix} 0 & 1 & 3 & 4 & 5 \\ \infty & 0 & 2 & 3 & 4 \\ \infty & 10 & 0 & 5 & 6 \\ \infty & 5 & 7 & 0 & 1 \\ \infty & 4 & 6 & 7 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 2 & 2 & 4 \\ 0 & 0 & 3 & 4 & 4 \\ 0 & 5 & 0 & 4 & 4 \\ 0 & 5 & 5 & 0 & 5 \\ 0 & 2 & 2 & 2 & 0 \end{pmatrix}$
27	$\begin{pmatrix} 0 & 1 & 3 & 1 & 6 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 3 \\ \infty & \infty & 2 & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
28	$\begin{pmatrix} 0 & 1 & 2 & 5 & 3 \\ \infty & 0 & \infty & 4 & 3 \\ \infty & \infty & 0 & 3 & 8 \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 3 & 2 & 5 \\ 0 & 0 & 0 & 4 & 5 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

## ПРИЛОЖЕНИЕ

Окончание таблицы А1

29	$\begin{pmatrix} 0 & 1 & \infty & 2 & 7 \\ \infty & 0 & \infty & \infty & \infty \\ 7 & 8 & 0 & 9 & 4 \\ 8 & 3 & \infty & 0 & 5 \\ 3 & 4 & \infty & 5 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 0 & 5 & 5 \\ 5 & 2 & 0 & 0 & 5 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$
30	$\begin{pmatrix} 0 & 4 & 2 & 5 & 10 \\ \infty & 0 & \infty & 4 & 7 \\ \infty & \infty & 0 & 3 & 8 \\ \infty & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 & 3 & 3 & 4 \\ 0 & 0 & 0 & 4 & 5 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

### 8. Алгоритм Форда – Фалкерсона

1.  $S = \{1\}; T = \{2, 3, 4, 5\};$   
 $P^+: (1, 2); (1, 4)$   
 Максимальный поток: 5
2.  $S = \{1, 3\}; T = \{2, 3, 4, 5\};$   
 $P^+: (3, 4); (3, 5)$   
 Максимальный поток: 7
3.  $S = \{1\}; T = \{2, 3, 4, 5\};$   
 $P^+: (1, 2); (1, 3); (1, 4)$   
 Максимальный поток: 7
4.  $S = \{1, 4\}; T = \{2, 3, 5\};$   
 $P^+: (1, 2); (1, 3); (1, 4); (4, 5)$   
 Максимальный поток: 7
5.  $S = \{1\}; T = \{2, 3, 4, 5\};$

## ПРИЛОЖЕНИЕ

---

$P^+$ : (1, 2); (1, 4)

Максимальный поток: 6

6.  $S = \{1, 2, 3\}; T = \{4, 5\};$

$P^+$ : (1, 5); (3, 4)

Максимальный поток: 6

7.  $S = \{1, 2, 3, 4\}; T = \{5\};$

$P^+$ : (1, 5); (2, 5); (4, 5)

Максимальный поток: 7

8.  $S = \{1\}; T = \{2, 3, 4, 5\};$

$P^+$ : (1, 3); (1, 5)

Максимальный поток: 6

9.  $S = \{1, 2, 3, 4\}; T = \{5\};$

$P^+$ : (1, 5); (2, 5); (4, 5)

Максимальный поток: 6

10.  $S = \{1, 4\}; T = \{2, 3, 5\};$

$P^+$ : (1, 3); (1, 5); (4, 5)

Максимальный поток: 10

11.  $S = \{1\}; T = \{2, 3, 4, 5\};$

$P^+$ : (1, 2); (1, 4)

Максимальный поток: 10

12.  $S = \{1, 3\}; T = \{2, 3, 4, 5\};$

$P^+$ : (1, 2); (1, 4); (1, 5); (3, 4)

Максимальный поток: 7

13.  $S = \{1\}; T = \{2, 3, 4, 5\};$

$P^+$ : (1, 3); (1, 5)

Максимальный поток: 8

14.  $S = \{1, 2, 3, 4\}; T = \{5\};$

$P^+$ : (3, 5); (4, 5)

Максимальный поток: 9

15.  $S = \{1\}; T = \{2, 3, 4, 5\};$

$P^+$ : (1, 2); (1, 3); (1, 5)

Максимальный поток: 9

16.  $S = \{1, 4\}; T = \{2, 3, 5\};$

## ПРИЛОЖЕНИЕ

---

$P^+$ : (1, 2); (1, 5); (4, 5)

Максимальный поток: 8

17.  $S = \{1\}$ ;  $T = \{2, 3, 4, 5\}$ ;

$P^+$ : (1, 2); (1, 4); (1, 5)

Максимальный поток: 10

18.  $S = \{1\}$ ;  $T = \{2, 3, 4, 5\}$ ;

$P^+$ : (1, 3); (1, 4); (1, 5)

Максимальный поток: 6

19.  $S = \{1\}$ ;  $T = \{2, 3, 4, 5\}$ ;

$P^+$ : (1, 2); (1, 3); (1, 4)

Максимальный поток: 8

20.  $S = \{1\}$ ;  $T = \{2, 3, 4, 5\}$ ;

$P^+$ : (1, 2); (1, 4)

Максимальный поток: 8

21.  $S = \{1, 2, 4\}$ ;  $T = \{3, 5\}$ ;

$P^+$ : (2, 5); (4, 5)

Максимальный поток: 6

22.  $S = \{1, 2\}$ ;  $T = \{3, 4, 5\}$ ;

$P^+$ : (1, 5); (2, 4); (2, 5)

Максимальный поток: 7

23.  $S = \{1, 2\}$ ;  $T = \{3, 4, 5\}$ ;

$P^+$ : (1, 5); (2, 4); (2, 5)

Максимальный поток: 7

24.  $S = \{1, 3\}$ ;  $T = \{2, 4, 5\}$ ;

$P^+$ : (1, 2); (1, 5); (3, 4)

Максимальный поток: 9

*Учебное электронное издание*

МОНАХОВ Юрий Михайлович

ПРИКЛАДНЫЕ АЛГОРИТМЫ

Алгоритмы на графах и сетях

Учебное пособие

*Издается в авторской редакции*

**Системные требования:** Intel от 1,3 ГГц; Windows XP/7/8/10; Adobe Reader;  
дисковод CD-ROM.

**Тираж 25 экз.**

Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых  
Изд-во ВлГУ  
gio.vlgu@yandex.ru

Институт информационных технологий и радиоэлектроники  
кафедра информатики и защиты информации  
mmonakhov@vlsu.ru