Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
(МАИ)

Институт № 8 «Компьютерные науки и прикладная математика»

КУРСОВАЯ РАБОТА

ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ

2-й семестр

Выполнил: студент группы М8О-108Б-22
Былькова К.А.

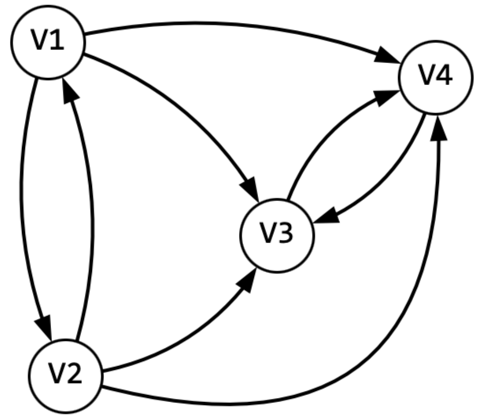
Проверил: Осипова В.А.

Москва 2023

Задание 1

Условие: Определить для орграфа, заданного матрицей смежности:

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



а) матрицу односторонней связности

б) матрицу сильной связности

в) компоненты сильной связности

г) матрицу контуров

Решение:

а) Матрица односторонней связности вычисляется по формуле

$T = E \vee A \vee A^2 \vee A^3$:

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A^3 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$T = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$ - матрица односторонней связности

б) Матрица сильной связности вычисляется по формуле $\overline{S} = T \& T^T$:

$$\overline{S} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \& \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$\overline{S} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$ - матрица сильной связности

в) Компоненты сильной связности:

$$\overline{S} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$\{\upsilon_1, \upsilon_2\}$ - первая компонента сильной связности

$$\overline{S_1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$\{\upsilon_3, \upsilon_4\}$ - вторая компонента сильной связности

$$\overline{S_2} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = 0$$

$\overline{S_2} = 0 \Rightarrow \overline{S_2}$ - нулевая матрица, значит компонент больше нет

г) Матрица контуров вычисляется по формуле K = $\overline{S}$&A:

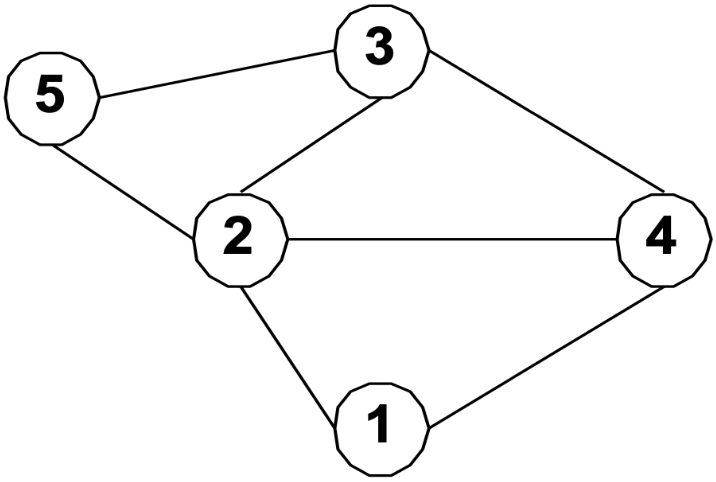
$$K = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \& \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Ответ: а) $T = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$; б) $\overline{S} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$;

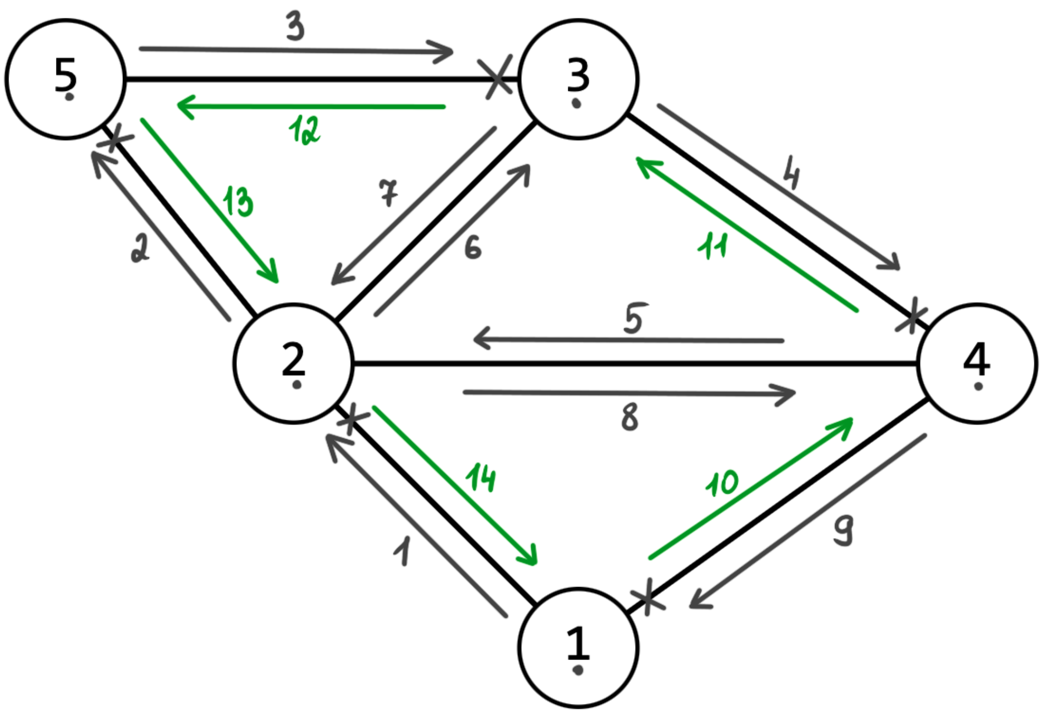
в) $\{\upsilon_1, \upsilon_2\}$, $\{\upsilon_3, \upsilon_4\}$; г) $K = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$.

Задание 2

Условие: Используя алгоритм Терри, определить замкнутый маршрут, проходящий ровно по два раза (по одному в каждом направлении) через каждое ребро графа :



Решение:



Ответ: 1 - 2 - 5 - 3 - 4 - 2 - 3 - 2 - 4 - 1 - 4 - 3 - 5 - 2 - 1.

Задание 3

Условие: Используя алгоритм “фронта волны”, найти все минимальные пути из первой вершины в последнюю орграфа, заданного матрицей смежности:

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Решение:

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	0	1	0	0	1	0	0
v_2	1	0	1	1	1	1	1	1
v_3	1	0	0	0	1	1	1	0
v_4	1	1	1	0	1	0	0	0
v_5	1	1	1	1	0	0	1	0
v_6	0	0	1	1	1	0	1	0
v_7	1	0	1	1	1	1	1	0
v_8	1	0	1	1	0	0	1	0

$W_0(v_1) = \{v_1\}$

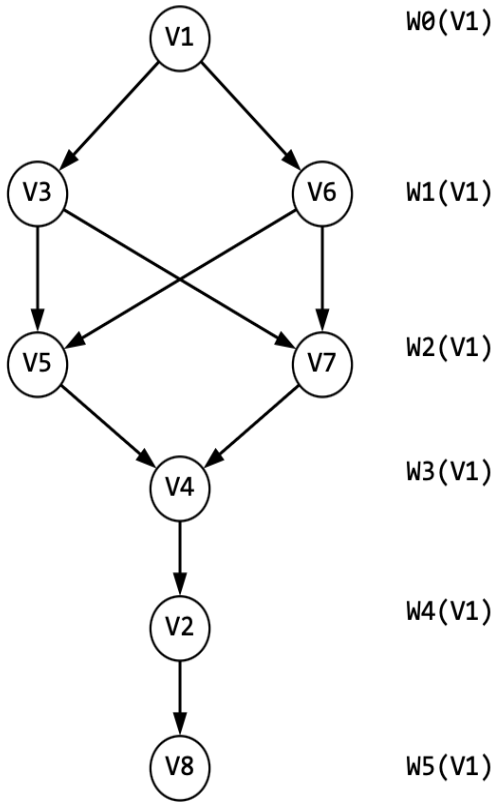
$W_1(v_1) = \{v_3, v_6\}$

$W_2(v_1) = \{v_5, v_7\}$

$W_3(v_1) = \{v_4\}$

$W_4(v_1) = \{v_2\}$

$W_5(v_1) = \{v_8\}$



Найдём кратчайшие пути:

1) v_8

2) $D^{-1}(v_8) \cap W_4(v_1) = \{v_2\} \cap \{v_2\} = \{v_2\}$

3) $D^{-1}(v_2) \cap W_3(v_1) = \{v_4, v_8\} \cap \{v_4\} = \{v_4\}$

4) $D^{-1}\{v_4\} \cap W_2(v_1) = \{v_2, v_5, v_7\} \cap \{v_5, v_7\} = \{v_5, v_7\}$

5. 1) $D^{-1}\{v_5\} \cap W_1(v_1) = \{v_3, v_4, v_6\} \cap \{v_3, v_6\} = \{v_3, v_6\}$

5. 2) $D^{-1}\{v_7\} \cap W_1(v_1) = \{v_3, v_4, v_6\} \cap \{v_3, v_6\} = \{v_3, v_6\}$

6. 1) $D^{-1}\{v_3\} \cap W_0(v_1) = \{v_1, v_5, v_7\} \cap \{v_1\} = \{v_1\}$

6. 2). $D^{-1}\{v_6\} \cap W_0(v\ ) = \{v_1, v_5, v_7\} \cap \{v_1\} = \{v_1\}$

Ответ: кратчайших путей всего четыре:

$v1 \to v6 \to v7 \to v4 \to v2 \to v8$;

$v1 \to v6 \to v5 \to v4 \to v2 \to v8$;

$v1 \to v3 \to v5 \to v4 \to v2 \to v8$;

$v1 \to v3 \to v7 \to v4 \to v2 \to v8$.

Задание 4

Условие: Используя алгоритм Форда, найти минимальные пути из первой вершины во все достижимые вершины в нагруженном графе, заданном матрицей длин дуг:

$$\begin{pmatrix} \infty & 3 & 5 & 6 & \infty & \infty & \infty & \infty \\ 2 & \infty & 1 & 2 & \infty & \infty & \infty & \infty \\ \infty & 1 & \infty & \infty & 3 & \infty & \infty & \infty \\ 3 & \infty & \infty & \infty & 4 & 7 & \infty & 9 \\ 5 & \infty & \infty & 4 & \infty & \infty & 4 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & 2 \\ 7 & \infty & \infty & \infty & \infty & 1 & \infty & 2 \\ 8 & \infty & \infty & 13 & \infty & \infty & \infty & \infty \end{pmatrix}$$

Решение:

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$	$\lambda_i^{(6)}$	$\lambda_i^{(7)}$	$\lambda_i^{(8)}$
v_1	∞	3	5	6	∞	∞	∞	∞	0	0	0	0	0	0	0	0	0
v_2	2	∞	1	2	∞	∞	∞	∞	∞	3	3	3	3	3	3	3	3
v_3	∞	1	∞	∞	3	∞	∞	∞	∞	5	4	4	4	4	4	4	4
v_4	3	∞	∞	∞	4	7	∞	9	∞	6	5	5	5	5	5	5	5
v_5	5	∞	∞	4	∞	∞	4	∞	∞	∞	8	7	7	7	7	7	7
v_6	∞	∞	∞	∞	∞	∞	1	2	∞	∞	13	12	12	12	12	12	12
v_7	7	∞	∞	∞	∞	1	∞	2	∞	∞	∞	12	11	11	11	11	11
v_8	8	∞	∞	13	∞	∞	∞	∞	∞	∞	15	14	14	13	13	13	13

Минимальный путь из v_1 в v_2: $v_1 \rightarrow v_2$ длиной 3

$\lambda_1^{(0)} + C_{12} = 0 + 3 = 3 = \lambda_2^{(1)}$

Минимальный путь из v_1 в v_3: $v_1 \rightarrow v_2 \rightarrow v_3$ длиной 4

$\lambda_1^{(0)} + C_{12} = 0 + 3 = 3 = \lambda_2^{(1)}$

$\lambda_2^{(1)} + C_{23} = 3 + 1 = 4 = \lambda_3^{(2)}$

Минимальный путь из v_1 в v_4: $v_1 \rightarrow v_2 \rightarrow v_4$ длиной 5

$\lambda_1^{(0)} + C_{12} = 0 + 3 = 3 = \lambda_2^{(1)}$

$\lambda_2^{(1)} + C_{24} = 3 + 2 = 5 = \lambda_4^{(2)}$

Минимальный путь из v_1 в v_5: $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_5$ длиной 7

$\lambda_1^{(0)} + C_{12} = 0 + 3 = 3 = \lambda_2^{(1)}$

$\lambda_2^{(1)} + C_{23} = 3 + 1 = 4 = \lambda_3^{(2)}$

$\lambda_3^{(2)} + C_{35} = 4 + 3 = 7 = \lambda_5^{(3)}$

Минимальный путь из v_1 в v_6: $v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_6$ длиной 12

$\lambda_1^{(0)} + C_{12} = 0 + 3 = 3 = \lambda_2^{(1)}$

$\lambda_2^{(1)} + C_{24} = 3 + 2 = 5 = \lambda_4^{(2)}$

$\lambda_4^{(2)} + C_{46} = 5 + 7 = 12 = \lambda_6^{(4)}$

Минимальный путь из v_1 в v_7: $v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_7$ длиной 11

$\lambda_1^{(0)} + C_{12} = 0 + 3 = 3 = \lambda_2^{(1)}$

$\lambda_2^{(1)} + C_{23} = 3 + 1 = 4 = \lambda_3^{(2)}$

$\lambda_3^{(2)} + C_{35} = 4 + 3 = 7 = \lambda_5^{(3)}$

$\lambda_5^{(3)} + C_{35} = 7 + 4 = 11 = \lambda_7^{(5)}$

Минимальный путь из v_1 в v_8: $v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_7 \rightarrow v_8$ длиной 13

$\lambda_1^{(0)} + C_{12} = 0 + 3 = 3 = \lambda_2^{(1)}$

$\lambda_2^{(1)} + C_{23} = 3 + 1 = 4 = \lambda_3^{(2)}$

$\lambda_3^{(2)} + C_{35} = 4 + 3 = 7 = \lambda_5^{(3)}$

$\lambda_5^{(3)} + C_{35} = 7 + 4 = 11 = \lambda_7^{(5)}$

$\lambda_7^{(5)} + C_{35} = 11 + 2 = 13 = \lambda_8^{(7)}$

Ответ: минимальные пути:

$v_1 \rightarrow v_2 = 3$

$v_1 \rightarrow v_3 = 4$

$v_1 \rightarrow v_4 = 5$

$v_1 \rightarrow v_5 = 7$

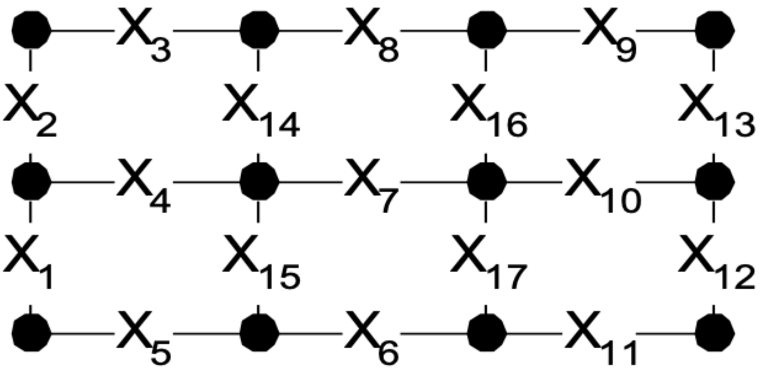
$v_1 \rightarrow v_6 = 12$

$v_1 \rightarrow v_7 = 11$

$v_1 \rightarrow v_8 = 13$

Задание 5

Условие: Найти остовное дерево с минимальной суммой длин входящих в него ребер.



Значения $x_1 - x_{13}$ приведены в задании, значения $x_{14} - x_{17}$ равны 5.

Решение:

$x_1 = 5,\ x_2 = 1,\ x_3 = 6,\ x_4 = 1,\ x_5 = 4,\ x_6 = 3,\ x_7 = 2,$
$x_8 = 5,\ x_9 = 6,\ x_{10} = 7,\ x_{11} = 2,\ x_{12} = 1,\ x_{13} = 4.$

1. Добавляем дуги с весом 1: x_2, x_4, x_{12}. Циклов нет.

2. Добавляем дуги с весом 2: x_7, x_{11}. Циклов нет.

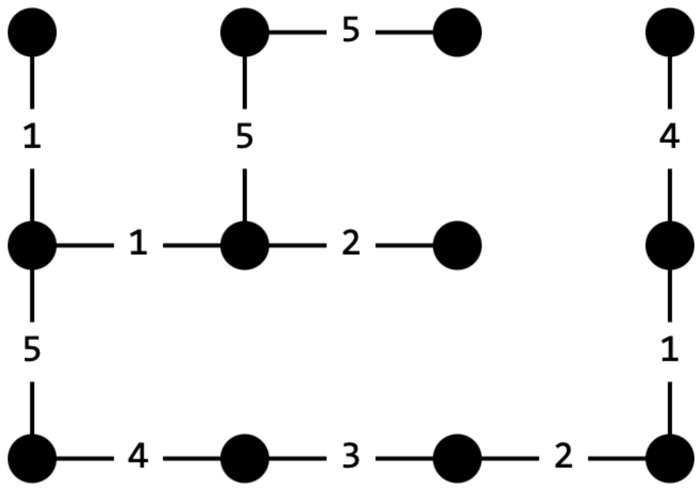
3. Добавляем дуги с весом 3: x_6. Циклов нет.

4. Добавляем дуги с весом 4: x_5, x_{13}. Циклов нет.

5. Добавляем дуги с весом 5: x_1, x_8, x_{14}, x_{15}, x_{16}, x_{17}. Если добавить еще x_{15}, x_{16} x_{17}, то будут циклы.

6. Добавляем дуги с весом 6: x_9. Если добавить эту дугу, то будет цикл.

7. Добавляем дуги с весом 7: x_{10}. Если добавить эту дугу, то будет цикл.

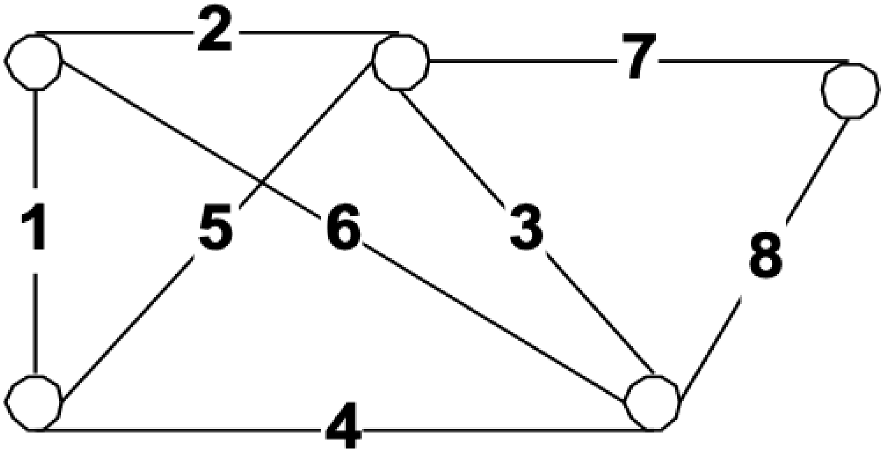


$L(D) = 1 * 3 + 2 * 2 + 3 * 1 + 4 * 2 + 5 * 3 = 33$

Ответ: 33 - вес минимального остовного дерева

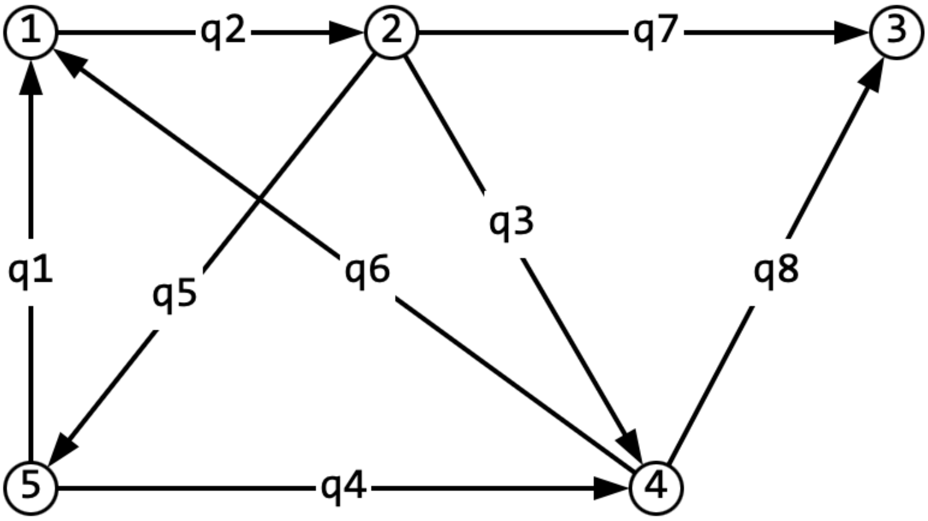
Задание 6

Условие: Пусть каждому ребру неориентированного графа соответствует некоторый элемент электрической цепи. Составить линейно независимые системы уравнений Кирхгофа для токов и напряжений. Пусть первому и пятому ребру соответствуют источники тока с ЭДС E1 и E2 (полярность выбирается произвольно), а остальные элементы являются сопротивлениями. Используя закон Ома, и, предполагая внутренние сопротивления источников тока равными нулю, получить систему уравнений для токов.

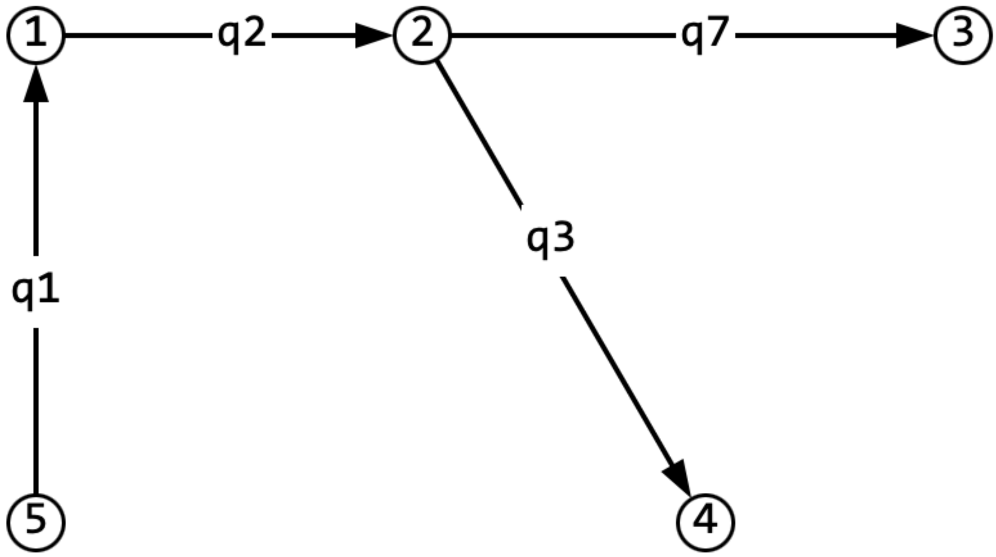


Решение:

1. Зададим произвольную ориентацию:



2. Построим произвольное остовное дерево D:



3. Найдем базис циклов и вектор-циклы:

3.1. $(D + q_4)$: μ_1: $U_5 - U_4 - U_1 - U_5 \Rightarrow C(\mu_1)$ = (-1 0 0 1 0 1 0 0)

3.2. $(D + q_5)$: μ_2: $U_2 - U_5 - U_4 - U_2 \Rightarrow C(\mu_2)$ = (0 0 -1 1 1 0 0 0)

3.3. $(D + q_6)$: μ_3: $U_1 - U_2 - U_4 - U_1 \Rightarrow C(\mu_3)$ = (0 1 1 0 0 1 0 0)

3.4. $(D + q_8)$: μ_4: $U_2 - U_4 - U_3 - U_2 \Rightarrow C(\mu_4)$ = (0 0 1 0 0 0 -1 1)

4. Цикломатическая матрица графа имеет вид:

$$C = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

5. Выпишем закон Кирхгофа для напряжений:

$$\begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} * \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \\ U_8 \end{pmatrix} = 0$$

$$\begin{cases} -U_1 + U_4 + U_6 = 0 \\ -U_3 + U_4 + U_5 = 0 \\ U_2 + U_3 + U_6 = 0 \\ U_3 - U_7 + U_8 = 0 \end{cases}$$

$$\begin{cases} U_1 = U_4 + U_6 \\ U_5 = U_3 - U_4 \\ U_2 + U_3 + U_6 = 0 \\ U_3 - U_7 + U_8 = 0 \end{cases}$$

6. Найдем матрицу инцидентности B орграфа:

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
U_1	1	-1	0	0	0	1	0	0
U_2	0	1	-1	0	-1	0	-1	0
U_3	0	0	0	0	0	0	1	1
U_4	0	0	1	1	0	-1	0	-1
U_5	-1	0	0	-1	1	0	0	0

$$\mathrm{B} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

7. Выпишем правила Кирхгофа для токов:

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \end{pmatrix} = 0$$

$$\begin{cases} I_1 - I_2 + I_6 = 0 \\ I_2 - I_3 - I_5 - I_7 = 0 \\ I_7 + I_8 = 0 \\ I_3 + I_4 - I_6 - I_8 = 0 \\ -I_1 - I_4 + I_5 = 0 \end{cases}$$

8. Подставим закон Ома:

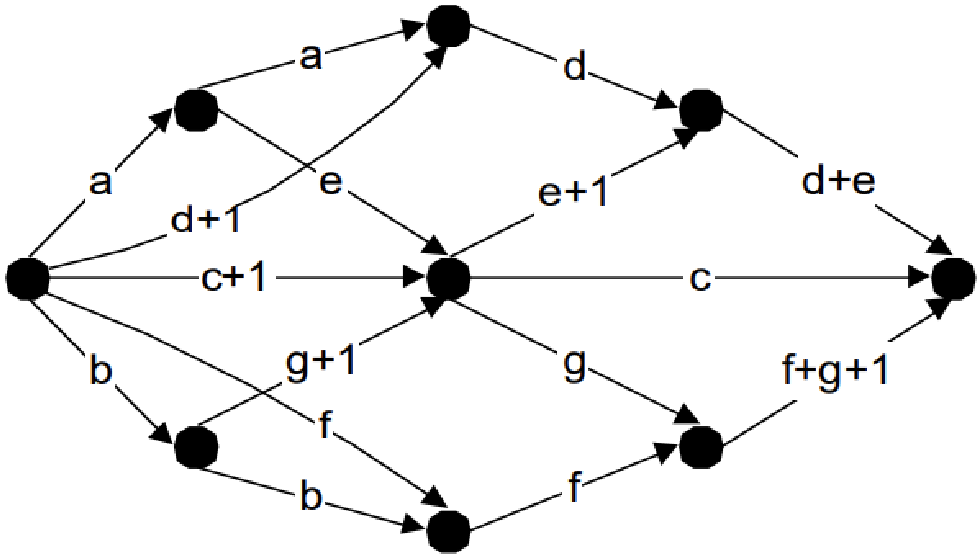
$$\begin{cases} I_1 - I_2 + I_6 = 0 \\ I_2 - I_3 - I_5 - I_7 = 0 \\ I_7 + I_8 = 0 \\ I_3 + I_4 - I_6 - I_8 = 0 \\ -I_1 - I_4 + I_5 = 0 \\ U_1 = U_4 + U_6 \\ U_5 = U_3 - U_4 \\ U_2 + U_3 + U_6 = 0 \\ U_3 - U_7 + U_8 = 0 \end{cases}$$

$$\begin{cases} I_1 - I_2 + I_6 = 0 \\ I_2 - I_3 - I_5 - I_7 = 0 \\ I_7 + I_8 = 0 \\ I_3 + I_4 - I_6 - I_8 = 0 \\ -I_1 - I_4 + I_5 = 0 \\ E_1 = I_4 R_4 + I_6 R_6 \\ E_2 = I_3 R_3 - I_4 R_4 \\ I_2 R_2 + I_3 R_3 + I_6 R_6 = 0 \\ I_3 R_3 - I_7 R_7 + I_8 R_8 = 0 \end{cases}$$

Ответ: Получена система уравнений для токов, состоящая из 9 уравнений и содержащая 8 неизвестных - токи I_1- I_8; ЭДС E_1, E_2 и сопротивления R_1, R_2 , R_3 , R_4 , R_5 , R_6 , R_7 , R_8 - известны.

Задание 7

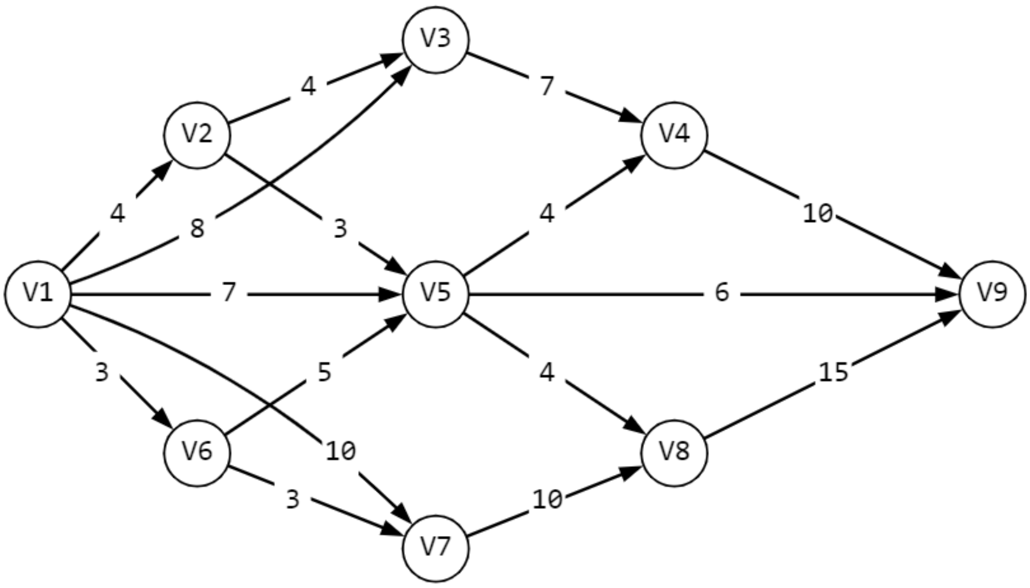
Условие: Построить максимальный поток по транспортной сети.



Значения величин a, b, c, d, e, f, g приведены в задании. Начинать с окаймляющих путей.

Решение:

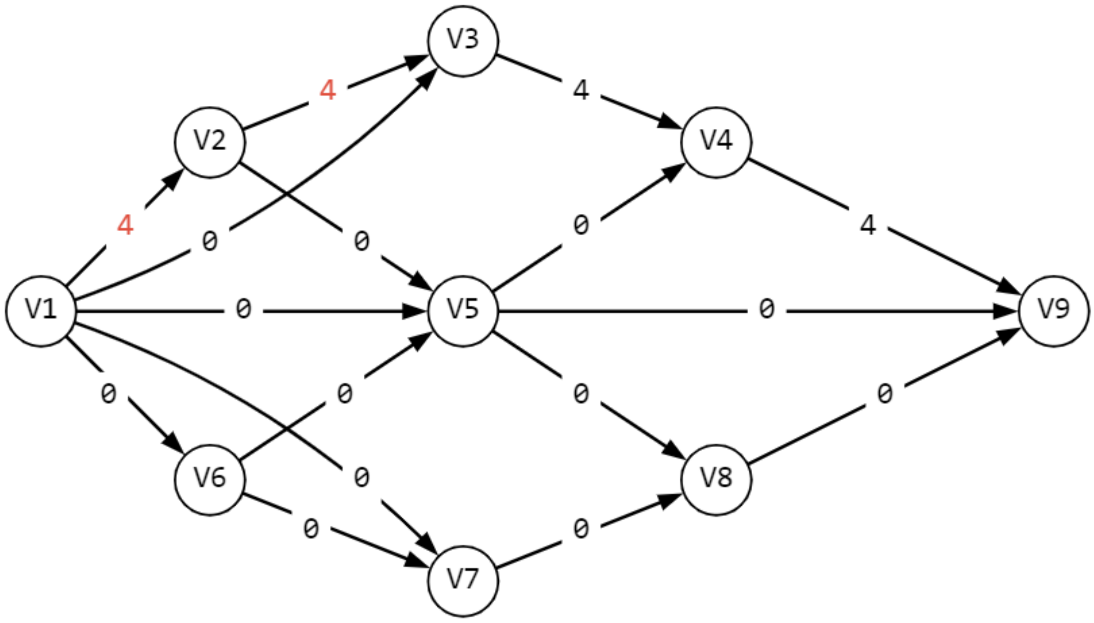
a = 4, b = 3, c = 6, d = 7, e = 3, f = 10, g = 4



1. Построение полного потока:

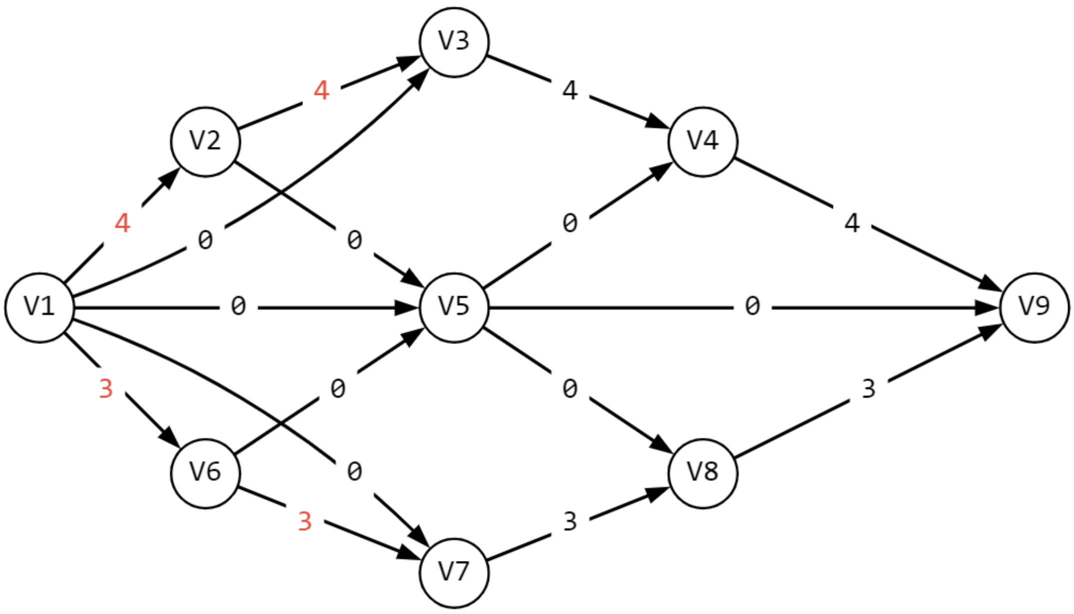
1) $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_9$

$\min\{4, 4, 7, 10\} = 4$



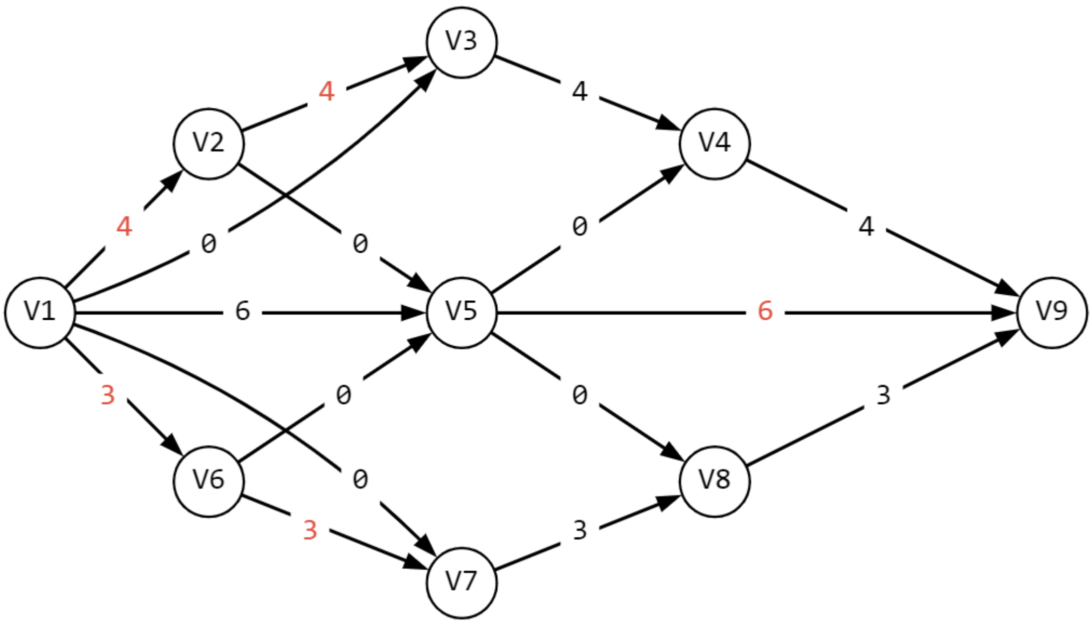
2) $v_1 \rightarrow v_6 \rightarrow v_7 \rightarrow v_8 \rightarrow v_9$

$\min\{3, 3, 10, 15\} = 3$



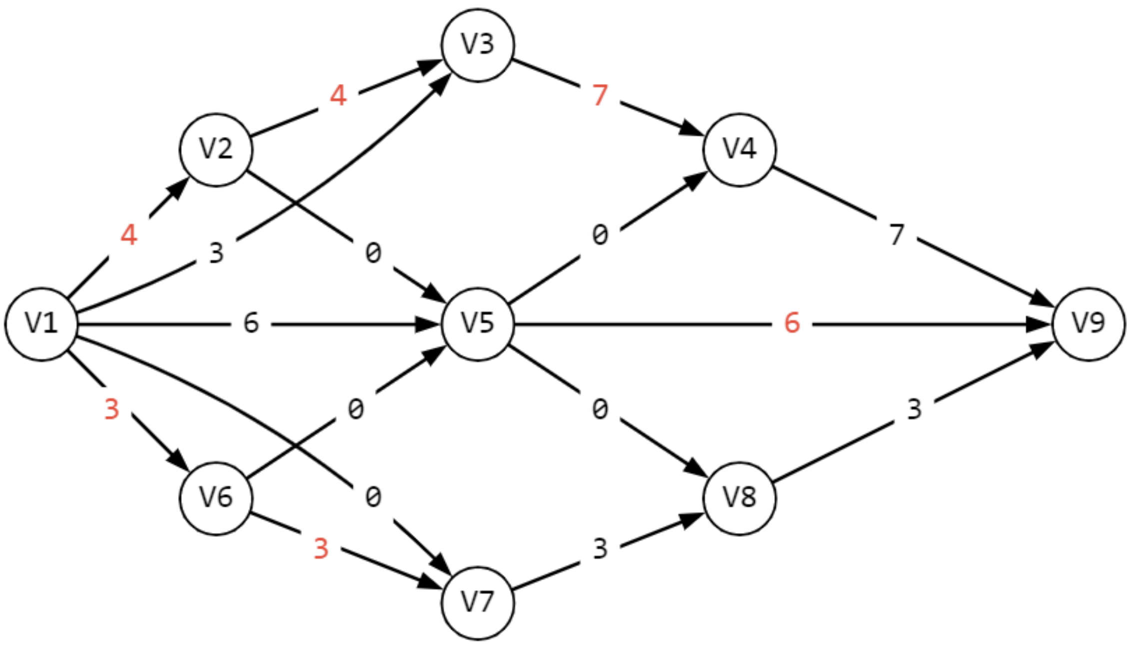
3) $v_1 \rightarrow v_5 \rightarrow v_9$

$\min\{7, 6\} = 6$



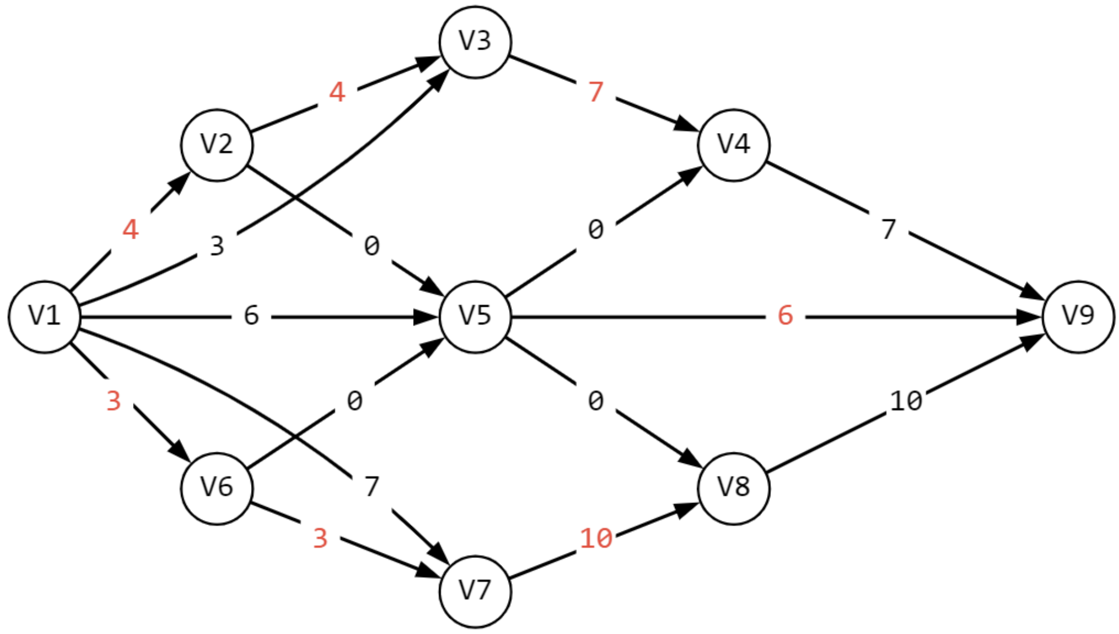
4) $v_1 \rightarrow v_3 \rightarrow v_4 \rightarrow v_9$

$\min\{8, 7 - 4, 10 - 4\} = 3$



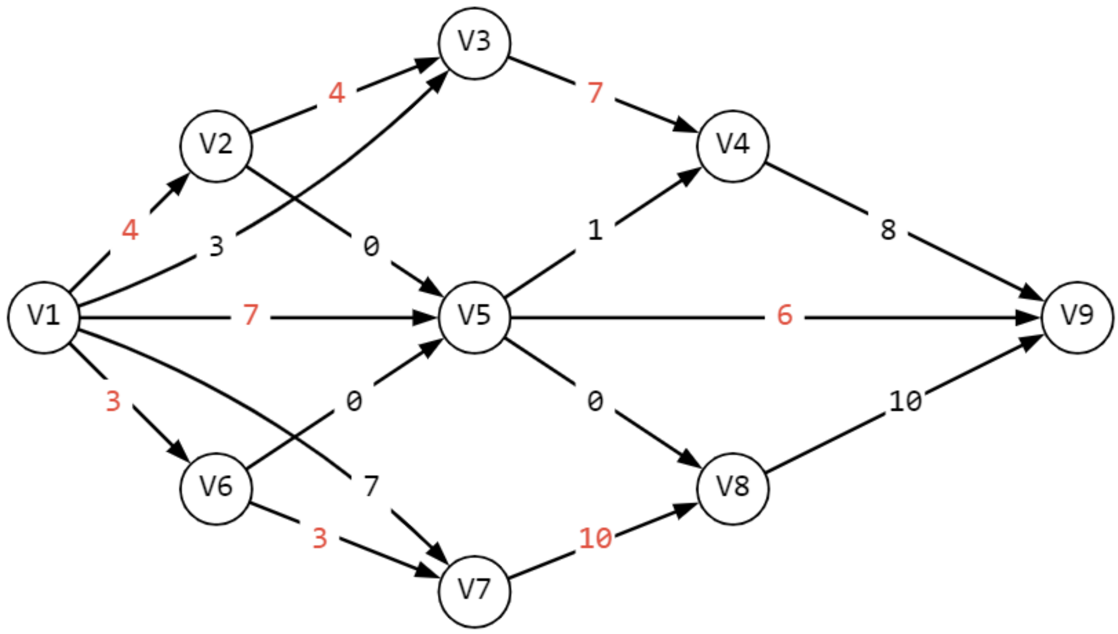
5) $v_1 \rightarrow v_7 \rightarrow v_8 \rightarrow v_9$

min{10, 10 - 3, 15 - 3} = 7



6) $v_1 \rightarrow v_5 \rightarrow v_4 \rightarrow v_9$

min{7 - 6, 4, 10 - 7} = 1

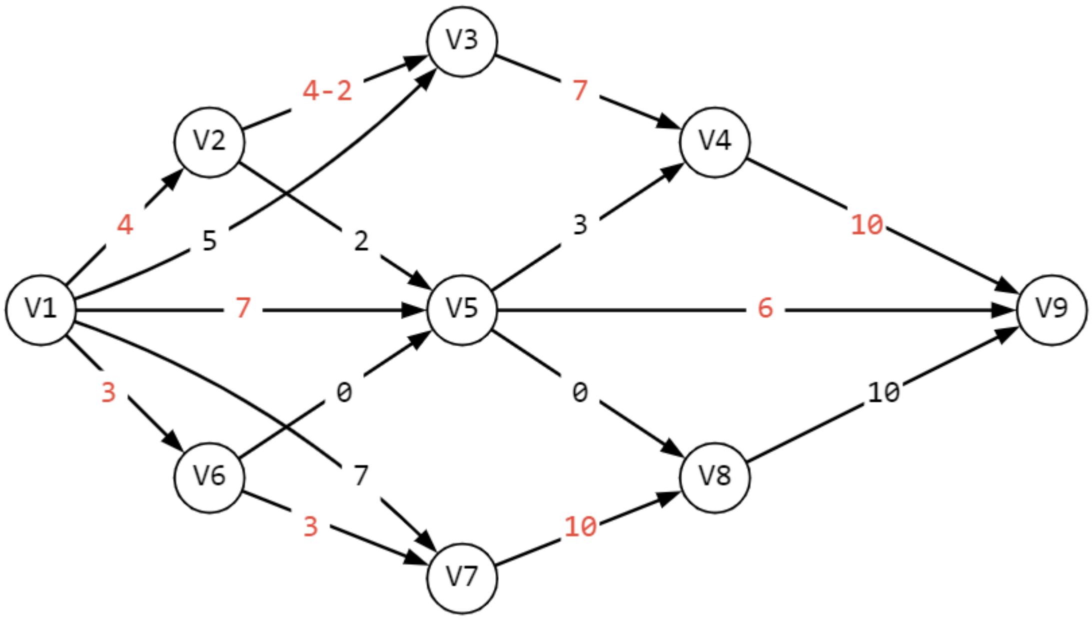


$\Phi_{\text{полн.}} = 1 + 7 + 3 + 6 + 3 + 4 = 24$

2. Построение максимального потока:

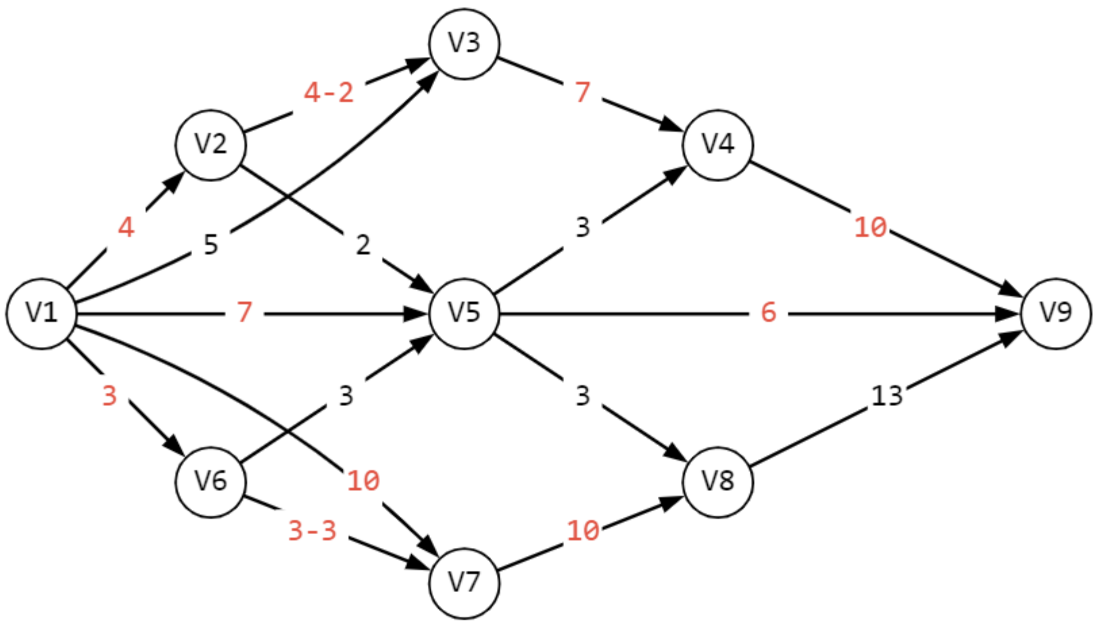
1) $v_1 \to v_3 \to v_2 \to v_5 \to v_4 \to v_9$

$\Delta_1 = \min\{8 - 3, 4 - 0, 3 - 0, 4 - 1, 10 - 8\} = 2$



2) $v_1 \to v_7 \to v_6 \to v_5 \to v_8 \to v_9$

$\Delta_2 = \min\{10 - 7, 3 - 0, 5 - 0, 4 - 0, 15 - 10\} = 3$



$\Phi_{макс} = 24 + 2 + 3 = 29$

Задание 8

8.1 Тема

Гамильтоновы пути (циклы). Поиск оптимального гамильтонового цикла в графе. Задача о коммивояжере.

8.2 Теоретические сведения

Пусть G = (V, Г) — граф, где V = $\{v_1, ..., v_n\}$. Для задания графа будем использовать матрицу смежности.

Матрицей смежности неориентированного графа G называется квадратная матрица A(G) = $[a_{ij}]$, $i, j = 1, 2, ..., n$, у которой

$$a_{ij} = \begin{cases} 1, \text{ если существует дуга, исходящая из } v_i \text{ и заходящая в } v_j \\ 0, \text{ в противном случае} \end{cases}$$

Взвешенным графом называется граф, в котором каждому ребру присваивается число (вес). Тогда в матрице смежности вместо всех единиц записываются веса ребер.

Гамильтоновым путем называется простой путь, проходящий через каждую вершину графа G ровно один раз.

Гамильтоновым циклом называют простой цикл, который проходит через каждую вершину графа G ровно один раз.

Граф называется **гамильтоновым**, если он содержит гамильтонов цикл.

Задача о коммивояжере (англ. Travelling salesman problem, TSP) — задача, в которой коммивояжер должен посетить N городов, побывав в каждом из них ровно по одному разу и завершив путешествие в том городе, с которого он начал. В какой последовательности ему нужно обходить города, чтобы общая длина его пути была наименьшей?

Время работы алгоритма, решающего задачу коммивояжера, существенно зависит от размера входных данных, то есть от количества городов.

8.3 Описание алгоритма

1) Задаем граф матрицей смежности, в которой 0 означает отсутствие ребра, а какое-то число - вес ребра;
2) Создаем вектор `path` для хранения пути и переменную `minPathLength` для хранения его длины;
3) Вызываем функцию `tsp`, которая принимает на вход вектор, хранящий матрицу смежности графа, число вершин графа и стартовую вершину. А возвращает длину минимального гамильтонова цикла графа;
4) В вектор `vertex` добавляем индексы всех вершин без стартовой;
5) В цикле высчитываем путь каждой перестановки из `vertex` и если путь минимален, то добавляем его в `path.` Также проверяем, существует ли гамильтонов путь в данном графе;
6) Вызываем функцию `generateDotFile`, которая генерирует файл `path.dot`;
7) Конвертируем полученный файл `path.dot` в изображение `path.png` с изображением полученного пути.

Таблица используемых переменных:

Название	Описание
`vertNum`	количество вершин графа
`graph`	матрица смежности графа
`start`	вершина, с которой начинается путь
`path`	найденный путь
`minPathLength`	длина минимального пути

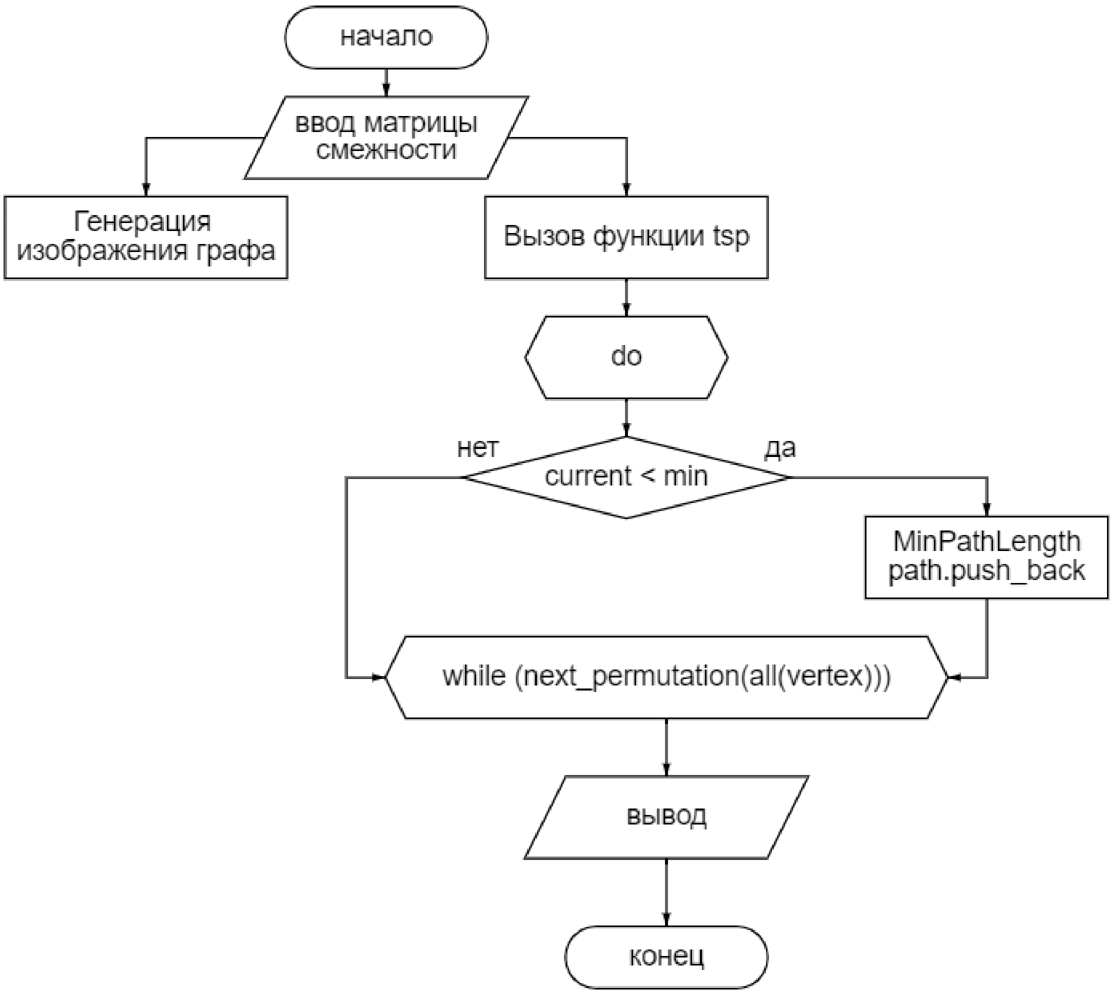
`vertex`	индексы всех вершин без стартовой
`currentPathLength`	длина пути текущей перестановки
`k`	переменная, необходимая для нахождения текущего пути

8.4 Оценка сложности алгоритма

Всего вариантов перестановок у нас $n!$, в каждой из которых $n - 1$ элементов.

Далее нам нужно их перебрать и выбрать минимальное значение $\Rightarrow$ $\Rightarrow$ получаем сложность $O(n! \ * \ (n - 1))$.

8.5 Блок-схема алгоритма



8.6 Код программы

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <fstream>

#define all(x) (x).begin(), (x).end()

using namespace std;

const int INF = 1e9;

void generateDotFile(const vector<int>& path, int vertNum)
{
    ofstream file("path.dot");
    if (!file.is_open()) {
        cout << "Failed to open path.dot file." << endl;
        return;
    }
    file << "digraph {" << endl;
    file << "node [shape = circle,\nstyle = filled,]" <<
endl;
    // Запись вершин графа
    file << path[0] + 1 << " [fillcolor=brown1];" << endl;
    for (int i = 1; i < vertNum; i++) {
        file << "  " << path[i] + 1 << ";" << endl;
    }
    // Запись ребер графа
    for (int i = 0; i < vertNum; i++) {
        file << "  " << path[i] + 1 << " -> " << path[i +
1] + 1 << ";" << endl;
    }
    file << "}" << endl;
    file.close();
}
// Travelling salesman problem function
int tsp(vector<vector<int>>& graph, int vertNum, int
start, vector<int>& path) {
```

```
    vector<int> vertex; // Вектор индексов всех вершин без
стартовой
    for (int i = 0; i < vertNum; ++i) {
        if (i != start)
            vertex.push_back(i);
    }
    int minPathLength = INF;
    bool flag = true;
    do {
        int currentPathLength = 0;
        int k = start;
        // Вычисляем длину пути текущей перестановки
        for (int i = 0; i < vertex.size(); ++i) {
            if (graph[k][vertex[i]] == 0) {
                flag = false;
                break;
            }
            currentPathLength += graph[k][vertex[i]];
            k = vertex[i];
        }
        if (!flag) break;
        currentPathLength += graph[k][start];
        if (graph[k][start] == 0) {
                flag = false;
                break;
            }
        // Проверяем минимален ли путь и, если да, то
записываем его
        if (currentPathLength < minPathLength) {
            minPathLength = currentPathLength;
            path.clear();
            path.push_back(start);
            for (int i = 0; i < vertex.size(); ++i)
                path.push_back(vertex[i]);
            path.push_back(start);
        }
    } while (next_permutation(all(vertex)));
    if (flag) {
        return minPathLength;
    }
```

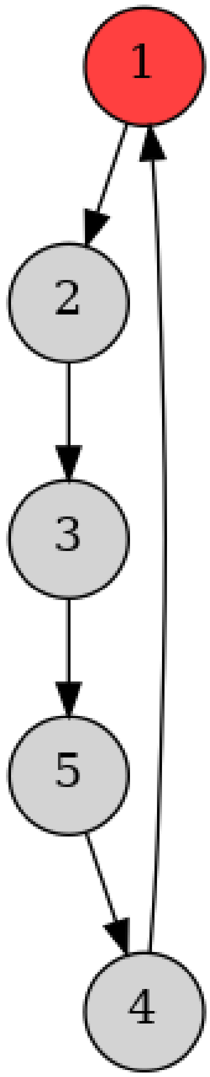
```
    else {
        return INF;
    }
}

int main() {
    int vertNum;
    cout << "Enter the number of vertices in the graph: ";
    cin >> vertNum;
    vector<vector<int>> graph(vertNum,
vector<int>(vertNum));
    cout << "Enter the adjacency matrix:\n";
    for (int i = 0; i < vertNum; ++i) {
        for (int j = 0; j < vertNum; ++j) {
            cin >> graph[i][j];
        }
    }
    int start;
    cout << "Enter the starting vertex: ";
    cin >> start;
    --start;
    vector<int> path;
    int minPathLength = tsp(graph, vertNum, start, path);
    if (minPathLength == INF) {
        cout << "There is no Hamiltonian path" << endl;
    }
    else {
        cout << "Minimum path length: " << minPathLength
<< endl;
        cout << "Path: ";
        for (int i = 0; i < path.size(); ++i)
            cout << path[i] + 1 << " ";
        cout << endl;
        generateDotFile(path, vertNum);
        system("dot -Tpng path.dot -o path.png");
    }
    return 0;
}
```

8.7 Тестовые примеры

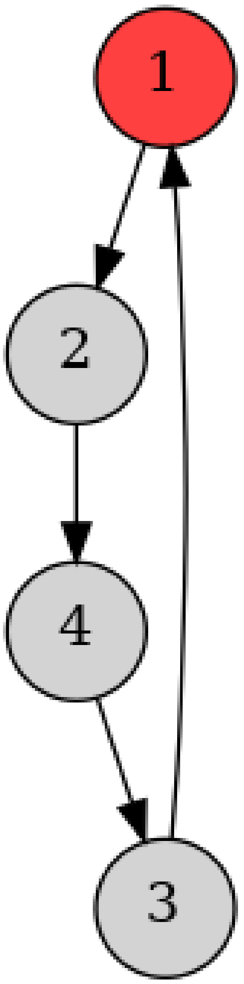
Тест 1

```
kristinab@LAPTOP-SFU9B1F4:/mnt/c/Users/Admin/Projects$ g++
DM_cw8.cpp && ./a.out
Enter the number of vertices in the graph: 5
Enter the adjacency matrix:
0 10 15 20 25
10 0 35 40 45
15 35 0 40 20
20 40 40 0 10
25 45 20 10 0
Enter the starting vertex: 1
Minimum path length: 95
Path: 1 2 3 5 4 1
```



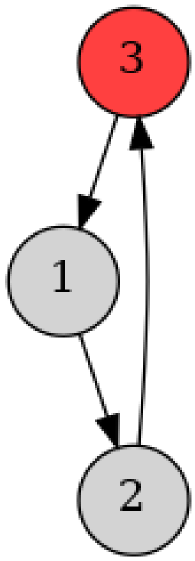
Тест 2

```
kristinab@LAPTOP-SFU9B1F4:/mnt/c/Users/Admin/Projects$ g++
DM_cw8.cpp && ./a.out
Enter the number of vertices in the graph: 4
Enter the adjacency matrix:
0 10 15 20
10 0 35 25
15 35 0 30
20 25 30 0
Enter the starting vertex: 1
Minimum path length: 80
Path: 1 2 4 3 1
```



Тест 3

```
kristinab@LAPTOP-SFU9B1F4:/mnt/c/Users/Admin/Projects$ g++
DM_cw8.cpp && ./a.out
Enter the number of vertices in the graph: 3
Enter the adjacency matrix:
0 5 10
100 0 100
10 10 0
Enter the starting vertex: 3
Minimum path length: 115
Path: 3 1 2 3
```



Тест 4

```
kristinab@LAPTOP-SFU9B1F4:/mnt/c/Users/Admin/Projects$ g++
DM_cw8.cpp && ./a.out
Enter the number of vertices in the graph: 3
Enter the adjacency matrix:
0 4 10
0 0 2
0 1 0
Enter the starting vertex: 1
There is no Hamiltonian path
```

8.8 Примеры прикладных задач

Задачу коммивояжера о поиске оптимального гамильтонова цикла, как следует из названия, можно использовать для составления маршрута человека, который должен посетить ряд пунктов и, в конце концов, вернуться в исходный пункт.

В глобальных масштабах данная задача помогает оптимизировать работу транспортных отделов и отделов логистики, упрощает работу почтовых и курьерских служб, повышает эффективность мониторинга объектов, например, станций сотовых операторов и нефтяных вышек.

8.9 Вывод

В ходе данной курсовой работы был изучен поиск оптимального гамильтонового цикла в графе. Была составлена программа для решения задачи о коммивояжере, благодаря которой можно найти минимальный гамильтонов цикл графа и его длину. Были приобретены навыки, которые будут полезны для выполнении других лабораторных и курсовых работ. Также было рассмотрено практическое применение задачи о коммивояжере.

8.10 Список литературы

1) В.Н. Нефедов, В.А, Осипова “Курс дискретной математики”
2) А. Кофман “Введение в прикладную комбинаторику”
3) Дж. Андерсон “Дискретная математика и комбинаторика”
4) В. Липский “Комбинаторика для программистов”