Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра математической кибернетики

Курсовая работа

Нахождение максимального пути в нагруженном графе

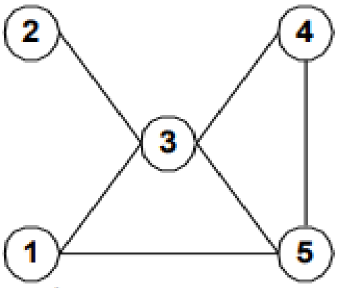
Студент: Бойцов И.А.
Группа: М8О–312Б–22
Преподаватель: Смерчинская С.О.
Оценка:___
Дата:________

Москва, 2024.

Вариант №5

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

2.

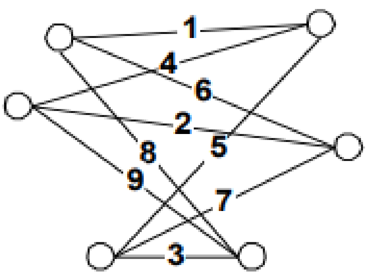


3. $$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

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

5. 4,3,2,5,4,7,8,2,3,7,1,8,5

6.



7. 3,5,5,10,3,11,5

8. Нахождение максимального пути в нагруженном графе.

Кофман А. Введение в прикладную комбинаторику

Задание №1

Определить для орграфа, заданного матрицей смежности:

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

а) матрицу односторонней связности (2-мя способами);
б) матрицу сильной связности;
в) компоненты сильной связности;
г) матрицу контуров
д) изображение графа и компонент сильной связности.

Решение:

а) Найдём матрицу связности по формуле: $T = E \vee A \vee A^2 \vee A^3$.

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

$$2)\ A^3 = A^2 \times A = \begin{pmatrix} 1&1&0&0 \\ 0&0&0&0 \\ 0&1&1&0 \\ 1&1&1&0 \end{pmatrix} \times \begin{pmatrix} 0&1&1&0 \\ 0&0&0&0 \\ 1&1&0&0 \\ 1&0&1&0 \end{pmatrix} = \begin{pmatrix} 0&1&1&0 \\ 0&0&0&0 \\ 1&1&0&0 \\ 1&1&1&0 \end{pmatrix}$$

$$3)\ T = E \vee A \vee A^2 \vee A^3 = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix} \times \begin{pmatrix} 0&1&1&0 \\ 0&0&0&0 \\ 1&1&0&0 \\ 1&0&1&0 \end{pmatrix} \times \begin{pmatrix} 1&1&0&0 \\ 0&0&0&0 \\ 0&1&1&0 \\ 1&1&1&0 \end{pmatrix} \times \begin{pmatrix} 0&1&1&0 \\ 0&0&0&0 \\ 1&1&0&0 \\ 1&1&1&0 \end{pmatrix} = \begin{pmatrix} 1&1&1&0 \\ 0&1&0&0 \\ 1&1&1&0 \\ 1&1&1&1 \end{pmatrix} = T -$$

– матрица односторонней связности.

Найдём T, используя итерационный алгоритм Уоршалла.

$$k = 0$$

$$T^{(0)} = E \vee A = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix} \vee \begin{pmatrix} 0&1&1&0 \\ 0&0&0&0 \\ 1&1&0&0 \\ 1&0&1&0 \end{pmatrix} = \begin{pmatrix} 1&1&1&0 \\ 0&1&0&0 \\ 1&1&1&0 \\ 1&0&1&1 \end{pmatrix}$$

$$k = 1,\ k-1 = 0$$

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

$$k = 2,\ k-1 = 1$$

$$T^{(2)} = \begin{pmatrix} 1&1&1&0 \\ 0&1&0&0 \\ 1&1&1&0 \\ 1&1&1&0 \end{pmatrix} \vee \begin{pmatrix} 0&1&0&0 \\ 0&1&0&0 \\ 0&1&0&0 \\ 0&1&0&0 \end{pmatrix} = \begin{pmatrix} 1&1&1&0 \\ 0&1&0&0 \\ 1&1&1&0 \\ 1&1&1&1 \end{pmatrix}$$

$$k=3,\quad k-1=2$$

$$T^{(3)}=\begin{pmatrix}1&1&1&0\\0&1&0&0\\1&1&1&0\\1&1&1&1\end{pmatrix}\vee\begin{pmatrix}1&1&1&0\\0&0&0&0\\1&1&1&0\\1&1&1&0\end{pmatrix}=\begin{pmatrix}1&1&1&0\\0&1&0&0\\1&1&1&0\\1&1&1&1\end{pmatrix}$$

$$k=4,\ k-1=3$$

$$T^{(4)}=\begin{pmatrix}1&1&1&0\\0&1&0&0\\1&1&1&0\\1&1&1&1\end{pmatrix}\vee\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\1&1&1&1\end{pmatrix}=\begin{pmatrix}1&1&1&0\\0&1&0&0\\1&1&1&0\\1&1&1&1\end{pmatrix}=T$$

д) Матрица сильной связности $\overline{S}=T\&T^T$

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

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

в.) Компоненты сильной связности

Выбираем первую строку, как ненулевую:

Номера вершин первой компоненты сильной связности соответствуют номерам столбцов матрицы $\overline{S}$, в которых в первой строке стоят единицы:

$\{v_1, v_3\}$

1. Обнуляем первый и третий столбец матрицы $\overline{S}$. Получаем матрицу

$$\overline{S}_1=\begin{pmatrix}0&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\end{pmatrix}$$

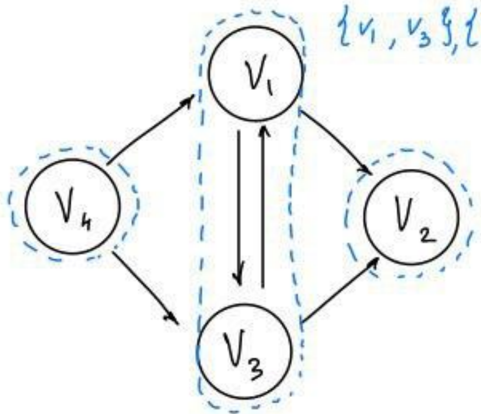
2. Ищем ненулевую строку матрицы $\overline{S}_1$: это вторая строка. Единица одна – во втором столбце. => Вторая компонента связности: $\{v_2\}$.
3. Обнулим второй столбец матрицы $\overline{S}_1$, остаётся еще одна единица в 4-ой строке и 4-ом столбце => Третья компонента связности $\{v_4\}$.
4. Обнуляем 4-ый столбец и получаем нулевую матрицу $\overline{S}_1$. => других компонент связности нет.

г.) Матрица контуров: $K = \overline{S} \& A$

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

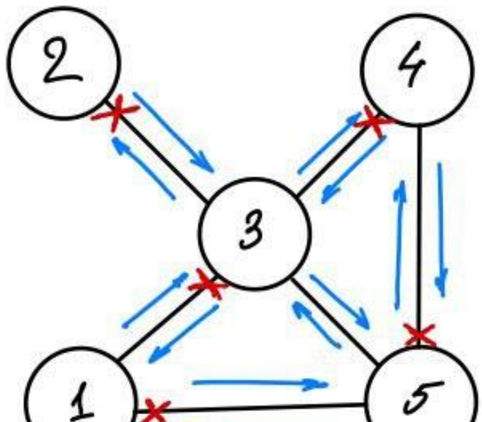
Следовательно, дуги $\langle v_1 v_3 \rangle$, $\langle v_3, v_1 \rangle$ – принадлежат какому-либо контуру исходного графа.

д.) Изображение графа и компонент связности:



Задание №2

Используя алгоритм Терри, определить замкнутый маршрут, проходящий ровно по 2 раза (по одному в кажд. напр.) через каждое ребро графа.



Решение:

Маршрут обхода:

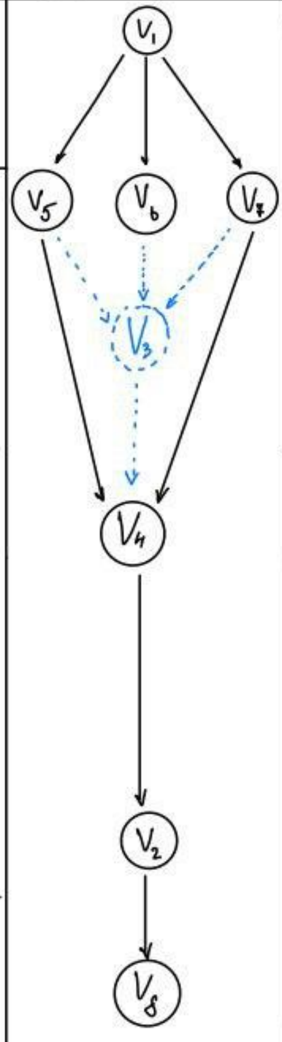
$1 \to 3 \to 2 \to 3 \to 4 \to 5 \to 1 \to 5 \to 3 \to 5 \to 4 \to 3 \to 1$

Задание 3.

Используя алгоритм „фронта волны", найти все минимальные пути из первой вершины в последнюю орграфа, заданного матрицей смежности.

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

Решение (прямой ход)

K	FW_k	Развертка графа	комментарий
0	$FW_0 = \{v_1\}$		В начальный момент времени волна сосредоточена в V_1
1	$FW_1 = \{$		Видно из первой строки матрицы смежности, однако забегая немного вперед, вершина V_6 не обеспечит нам минимальный путь.
2	$FW_2 = \{v_4\}$		Вершины V_5 и V_7 следует в V_4 (Они также, как и V_6 следуют в V_3, но эта вершина в свою очередь следует также в V_4, что удлиняет нам путь на единицу.
3	$FW_3 = \{v_2\}$		Из матрицы смежности из V_4 переходим в V_2
4	$FW_4 = \{v_8\}$		Достигнуто условие окончание прямого хода. Длина минимальных путей $\underline{4}$.

Ответ: минимальные пути из V_1 в V_8:

1.) $V_1 - V_5 - V_4 - V_2 - V_8$

2.) $V_1 - V_7 - V_4 - V_2 - V_8$

Всего 2 пути.

Задание 4.

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

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

Решение

1.) Составим таблицу итераций:

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	λ_i^0	λ_i^1	λ_i^2	λ_i^3	λ_i^4	λ_i^5	λ_i^6	λ_i^7
V_1	∞	3	5	∞	∞	∞	∞	∞	0	0	0	0	0	0	0	0
V_2	∞	∞	1	9	∞	5	∞	∞	∞	3	3	3	3	3	3	3
V_3	13	1	∞	∞	4	∞	3	∞	∞	5	4	4	4	4	4	4
V_4	∞	∞	∞	∞	2	∞	∞	3	∞	∞	12	11	10	10	10	10
V_5	∞	∞	∞	2	∞	∞	∞	6	∞	∞	9	8	8	8	8	8
V_6	∞	∞	∞	3	∞	∞	2	∞	∞	∞	8	8	8	8	8	8
V_7	∞	∞	∞	∞	2	2	∞	∞	∞	∞	8	7	7	7	7	7
V_8	2	3	∞	5	4	∞	8	∞	∞	∞	∞	15	14	13	13	13

2.) Длины минимальных путей из вершины V_1 во все остальные определены в последнем столбце таблицы

3.) Найдём вершины, входящие в минимальные пути из V_1 во все остальные вершины графа.

3.1.) Минимальный путь из V_1 в V_2: $V_1 - V_2$, его длина равна 3.

$$\lambda^{(0)} + c_{12} = 0 + 3 = \lambda_2^{(1)}$$

3.2.) Минимальный путь из V_1 в V_3: $V_1 - V_2 - V_3$ его длина равна 4.

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

3.3.) Минимальный путь из V_1 в V_4: $V_1 - V_2 - V_3 - V_5 - V_4$ его длина равна 10.

$$\lambda_3^{(2)} + c_{35} = 4 + 4 = \lambda_5^{(2)}$$
$$\lambda_5^{(2)} + c_{54} = 8 + 2 = \lambda_4^{(3)}$$

3.4.) Минимальный путь из V_1 в V_5: $V_1 - V_2 - V_3 - V_5$, его длина равна 8.

$$\lambda_3^{(2)} + c_{35} = 4 + 4 = \lambda_5^{(2)}$$

3.5.) Минимальный путь из V_1 в V_6: $V_1 - V_2 - V_6$, его длина равна 8.

$$\lambda_2^{(1)} + c_{26} = 3 + 5 = \lambda_6^{(2)}$$

3.6.) Минимальный путь из V_1 в V_7: $V_1 - V_2 - V_3 - V_7$, его длина равна 7.

$$\lambda_3^{(2)} + c_{37} = 4 + 3 = \lambda_7^{(3)}$$

3.7.) Минимальный путь из V_1 в V_8: $V_1 - V_2 - V_3 - V_5 - V_4 - V_8$, его длина равна 13.

$$\lambda_4^{(4)} + c_{48} = 10 + 3 = \lambda_8^{(5)}$$

Ответ: Минимальные пути из V_1 в V_2: $V_1\ V_2$

V_3: $V_1\ V_2\ V_3$

V_4: $V_1\ V_2\ V_3\ V_5\ V_4$

V_5: $V_1\ V_2\ V_3\ V_5$

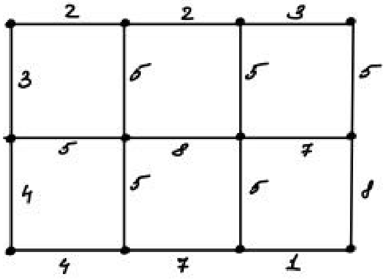
V_6: $V_1\ V_2\ V_6$

V_7: $V_1\ V_2\ V_3\ V_7$

V_8: $V_1\ V_2\ V_3\ V_5\ V_4\ V_8$

Задание 5.

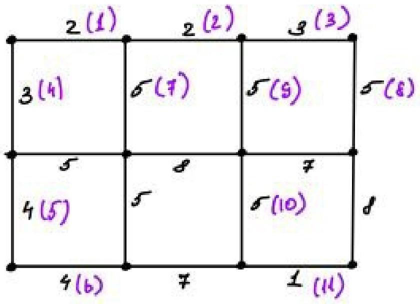
Найти остовное дерево с минимальной суммой длин входящих в него ребер.

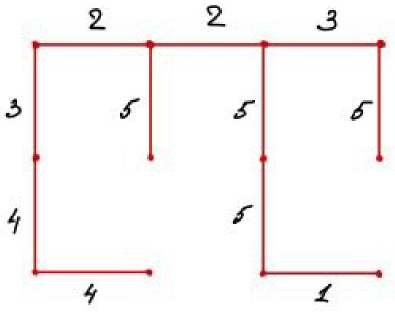


Решение:

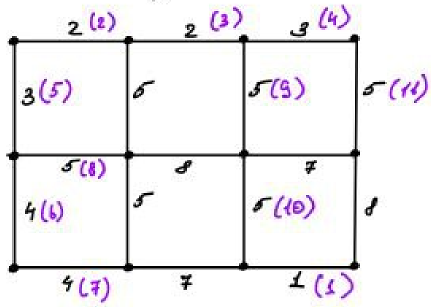
Решим задание 2-мя способами: алгоритмом Прима и Крускала.

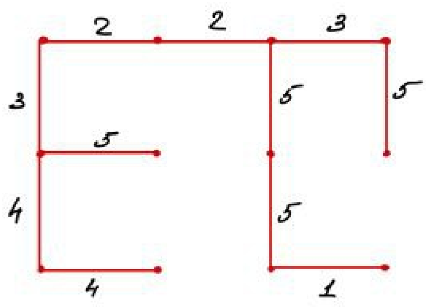
Прима





Крускала





$\sum = 2+2+3+3+4+4+5+5+5+5+5+1 = 39$

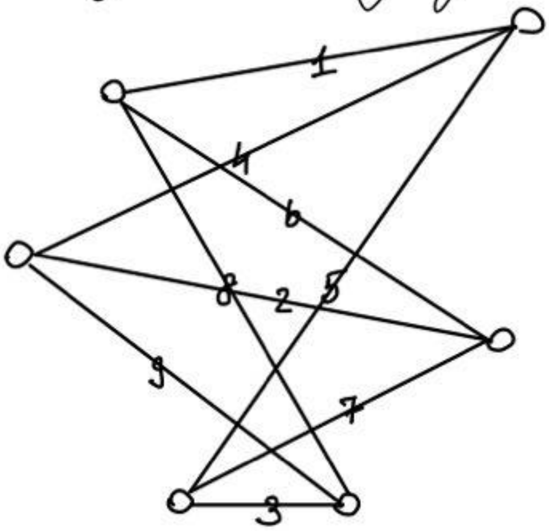
$\sum = 1+2+2+3+3+4+4+5+5+5+5 = 39$

Итак суммы ребер остовного дерева двух алгоритмов совпали
=> минимальная сумма длин ребер <u>39</u>.

Ответ: $L(D) = \underline{39}$.

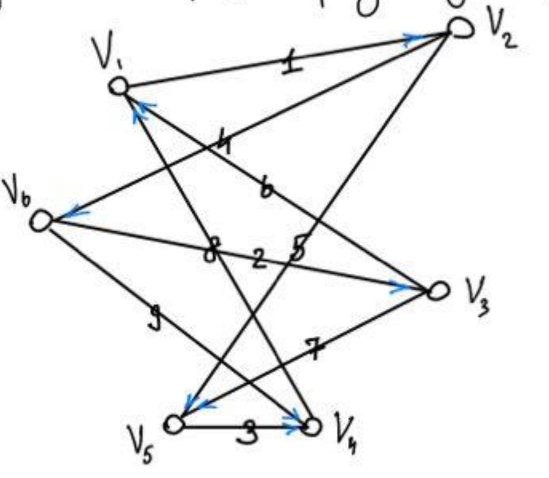
Задание 6.

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 $\mathcal{E}_1$ и $\mathcal{E}_2$, а остальные элементы являются сопротивлениями. Используя закон Ома, и, предполагая внутренние сопротивления источников тока равными нулю, получить систему уравнений для токов.

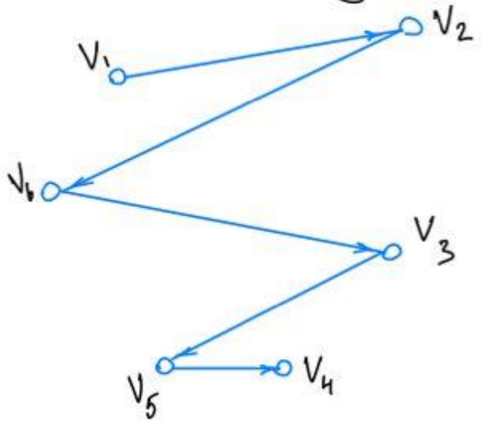


Решение:

1.) Зададим на граф произвольную ориентацию:



2.) Построим произвольное остовное дерево D заданного графа:



3.) Найдём базис циклов, добавляя по одному не вошедшему в него ребру. Затем найдём соответствующие вектор-циклы.

$(D + x_5): \mu_1: V_6 - V_2 - V_5 - V_3 - V_6 \Rightarrow C(\mu_1) = (0,-1,0,-1,1,0,-1,0,0)$

$(D + x_6): \mu_2: V_6 - V_3 - V_1 - V_2 - V_6 \Rightarrow C(\mu_2) = (1,1,0,1,0,1,0,0,0)$

$(D + x_8): \mu_3: V_5 - V_4 - V_1 - V_3 - V_5 \Rightarrow C(\mu_3) = (0,0,1,0,0,-1,1,1,0)$

$(D + x_9): \mu_4: V_3 - V_6 - V_4 - V_5 - V_3 \Rightarrow C(\mu_4) = (0,-1,-1,0,0,0,-1,0,1)$

4.) Цикломатическая матрица имеет вид:

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

5.) Выпишем закон Кирхгофа для напряжений:

$$\begin{pmatrix} 0 & -1 & 0 & -1 & 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \\ U_8 \\ U_9 \end{pmatrix} = 0$$

Напряжения соответствующие ребрам, не вошедшим в остовное дерево — базисные переменные системы.

$$\begin{cases} -U_2 - U_4 + U_5 - U_7 = 0 \\ U_1 + U_2 + U_4 + U_6 = 0 \\ U_3 - U_6 + U_7 + U_8 = 0 \\ -U_2 - U_3 - U_7 + U_9 = 0 \end{cases} \Rightarrow \begin{cases} U_5 = U_2 + U_4 + U_7 \\ U_1 = -U_2 - U_4 - U_6 \\ U_6 = U_3 + U_7 + U_8 \\ U_9 = U_2 + U_3 + U_7 \end{cases}$$

6.) Составим матрицу инцидентности:

$$B = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \end{array} \Rightarrow \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \end{pmatrix} = 0$$

$$\begin{cases} -I_1 + I_6 + I_8 = 0 \\ I_1 - I_4 - I_5 = 0 \\ I_2 - I_6 - I_7 = 0 \\ I_3 - I_8 + I_9 = 0 \\ -I_3 + I_4 + I_7 = 0 \\ \cancel{-I_2 + I_4 - I_9 = 0} \end{cases} \Rightarrow \begin{cases} -I_1 + I_6 + I_8 = 0 \\ I_1 - I_4 - I_5 = 0 \\ I_2 - I_6 - I_7 = 0 \\ I_3 - I_8 - I_9 = 0 \\ -I_3 + I_4 + I_7 = 0 \end{cases}$$

По свойствам матрицы инцидентности

8.) Подставим закон Ома:

$$\begin{cases} 0 = I_3 R_3 - I_6 R_6 + I_7 R_7 + I_8 R_8 \\ \mathcal{E}_1 = -I_2 R_2 + I_4 R_4 + I_7 R_7 \\ 0 = -I_2 R_2 - I_3 R_3 - I_7 R_7 + I_9 R_9 \\ \mathcal{E}_2 = I_2 R_2 + I_4 R_4 + I_7 R_7 \end{cases}$$

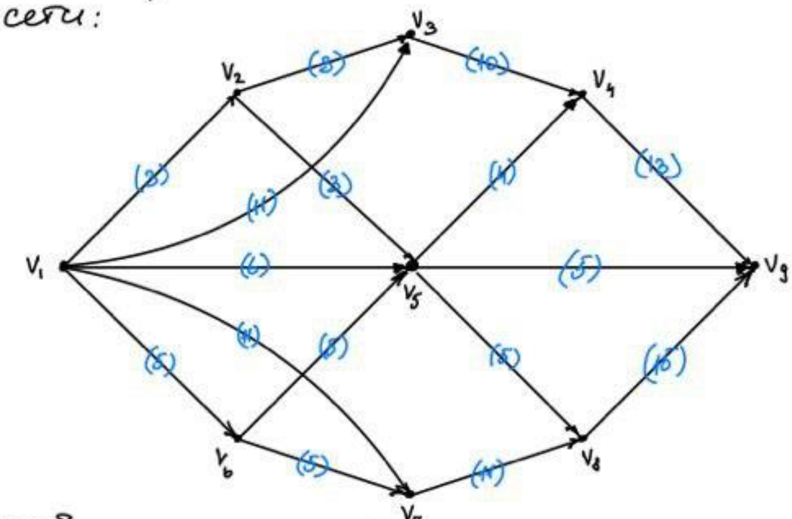
9.) Совместная система имеет вид:

$$\begin{cases} -I_1 + I_6 + I_8 = 0 \\ I_1 - I_4 - I_5 = 0 \\ I_2 - I_6 - I_7 = 0 \\ I_3 - I_8 - I_9 = 0 \\ -I_3 + I_4 + I_7 = 0 \\ 0 = I_3 R_3 - I_6 R_6 + I_7 R_7 + I_8 R_8 \\ \mathcal{E}_1 = -I_2 R_2 - I_4 R_4 - I_6 R_6 \\ 0 = -I_2 R_2 - I_3 R_3 - I_7 R_7 + I_9 R_9 \\ \mathcal{E}_2 = I_2 R_2 + I_4 R_4 + I_7 R_7 \end{cases}$$

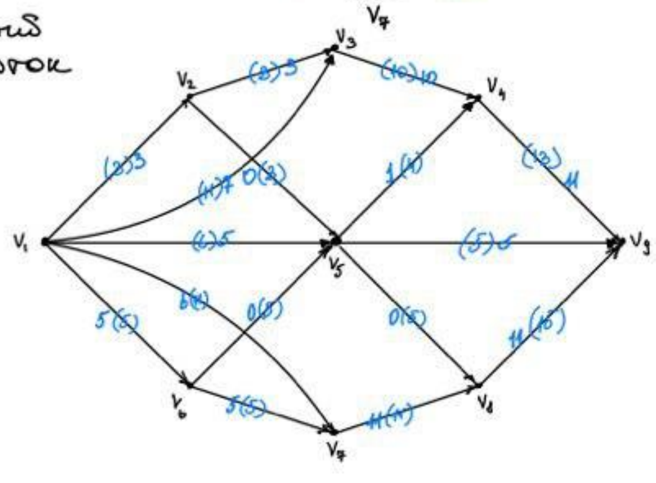
Ответ: Девять уравнений и девять неизвестных — токи $I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9$; Э.Д.С. $\mathcal{E}_1, \mathcal{E}_2$ и сопротивления $R_2, R_3, R_4, R_6, R_7, R_8, R_9$ известны.

Задание 7.

Построить максимальный поток по данной транспортной сети:



Полный поток



I) Построение полного потока.

1) $V_1 V_2 V_3 V_4 V_9$;

$\Delta_1 \min(3-0,\ 3-0,\ 10-0,\ 13-0) = 3$

2.) $V_1 V_6 V_7 V_8 V_9$

$\Delta_2 \min(5-0,\ 5-0,\ 11-0,\ 15-0) = 5$

3.) $V_1 V_5 V_9$

$\Delta_3 \min(6-0,\ 5-0) = 5$

4) $V_1 V_3 V_4 V_9$

$\Delta_4 \min(11-0,\ 10-3,\ 13-3) = 7$

5.) $V_1 V_7 V_8 V_9$

$\Delta_5 \min(11-0,\ 11-5,\ 15-5) = 6$

6.) $V_1 V_5 V_4 V_9$

$\Delta_6 \min(6-5,\ 4-0,\ 13-10) = 1$

Величина полного потока:

$\varphi_{полн} = 11 + 5 + 11 = 3 + 5 + 5 + 7 + 6 + 1 = \underline{\underline{27}}$

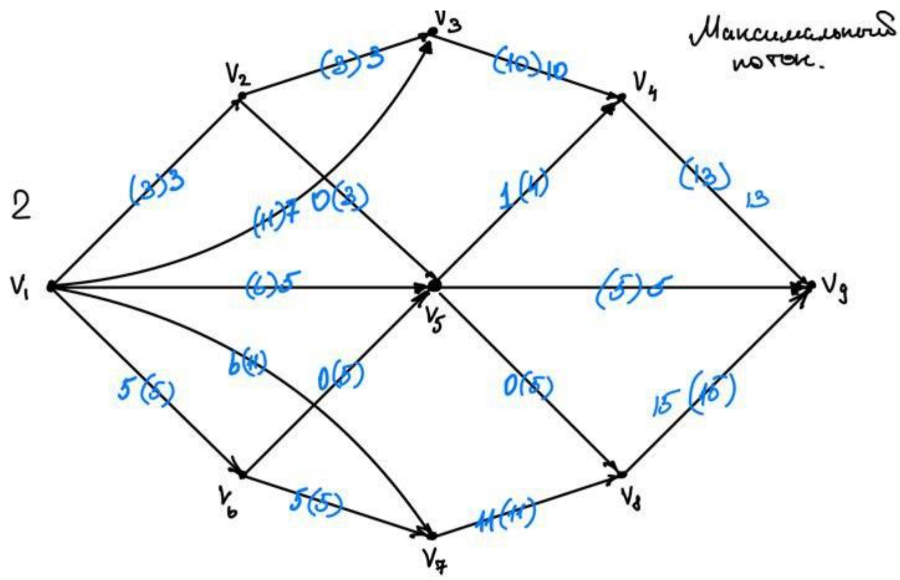
II.) Построение максимального потока:

7.) $V_1 V_3 V_2 V_5 V_4 V_9$

$\Delta_7 = \min(11-7,\ 3,\ 3-0,\ 4-1,\ 13-11) = 2$

8.) $V_1 V_7 V_6 V_5 V_8 V_9$

$\Delta_8 = \min(11-6,\ 5,\ 5-0,\ 5-0,\ 15-11) = 4$



Величина максимального потока

$\Phi_{макс} = 13+5+15 = \underline{\underline{33}}$

Ответ: $\Phi_{макс} = \underline{33}$

8 задание

1. Основные понятие и определения

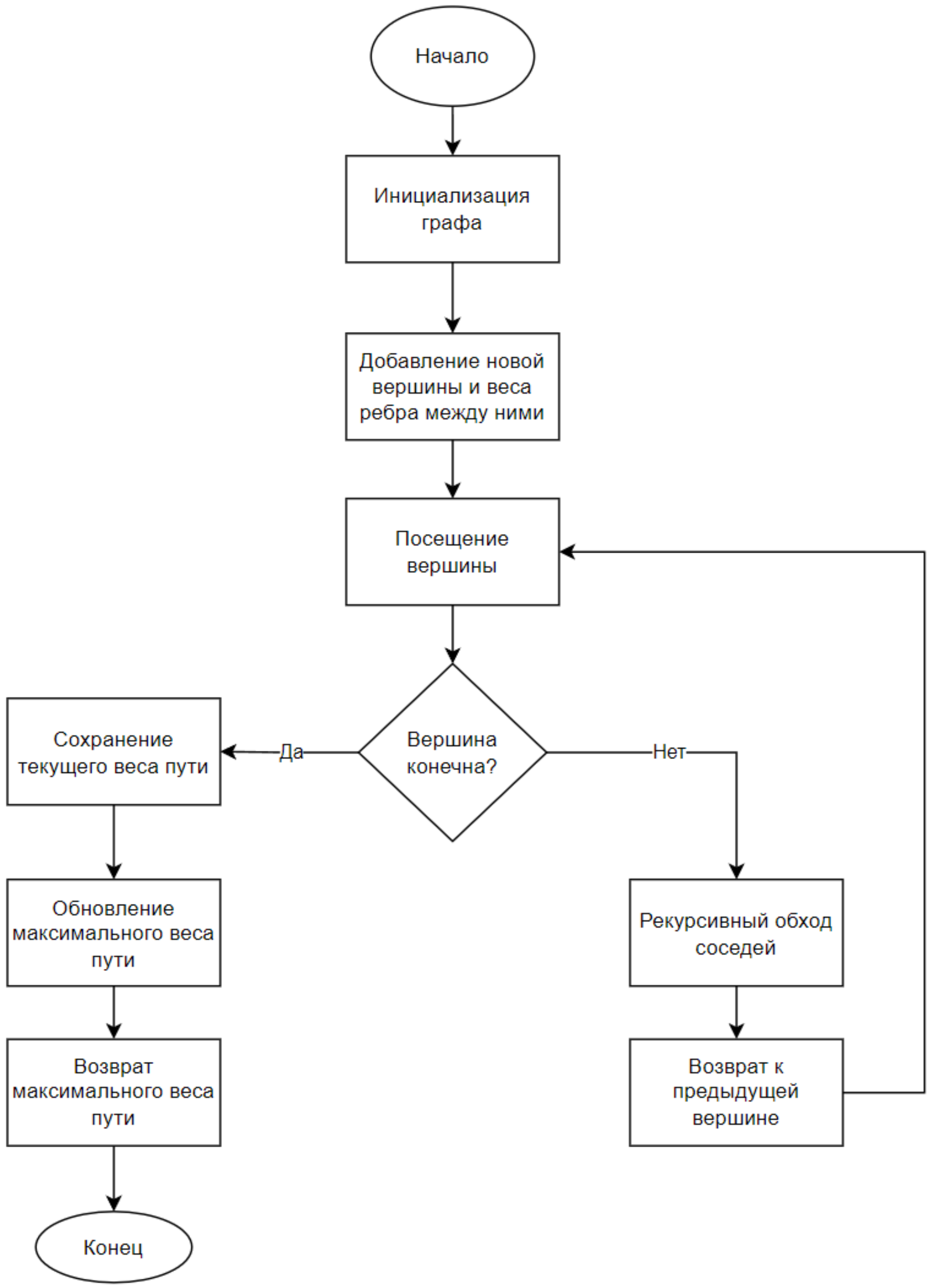
1. **Граф**:
   - Набор вершин (узлов) и рёбер, соединяющих эти вершины.
   - Бывает **ориентированным** (рёбра имеют направление) или **неориентированным** (рёбра не имеют направления).
2. **Нагруженный граф**:
   - Граф, в котором каждому ребру присвоен вес (например, расстояние или стоимость).
3. **Путь**:
   - Последовательность рёбер, соединяющая начальную и конечную вершину. Вес пути — сумма весов рёбер, входящих в путь.
4. **Максимальный путь**:
   - Путь с наибольшей суммой весов среди всех возможных путей между двумя заданными вершинами.
5. **Ацикличный граф (DAG)**:
   - Ориентированный граф, не содержащий циклов.
6. **DFS (алгоритм обхода в глубину в первую очередь)**:
   - Алгоритм обхода графа, который углубляется по каждому пути, прежде чем переходить к следующему.

2. Описание алгоритма

Модифицированный DFS для поиска максимального пути

1. **Инициализация**:
   - Граф представлен в виде списка смежности: каждая вершина содержит список своих соседей с весами рёбер.
   - Указываются начальная и конечная вершины.
2. **Рекурсивный обход графа**:
   - Из текущей вершины рекурсивно исследуются все пути к конечной вершине.
   - Суммарный вес пути сравнивается с текущим максимальным.
3. **Backtracking**:
   - После обработки всех соседей возвращаемся на предыдущую вершину, чтобы попробовать другие маршруты.
4. **Условия завершения**:
   - Если достигли конечной вершины, фиксируем текущий вес пути как потенциально максимальный.

3. Логическая блок-схема



4. Описание программы

Основные модули:

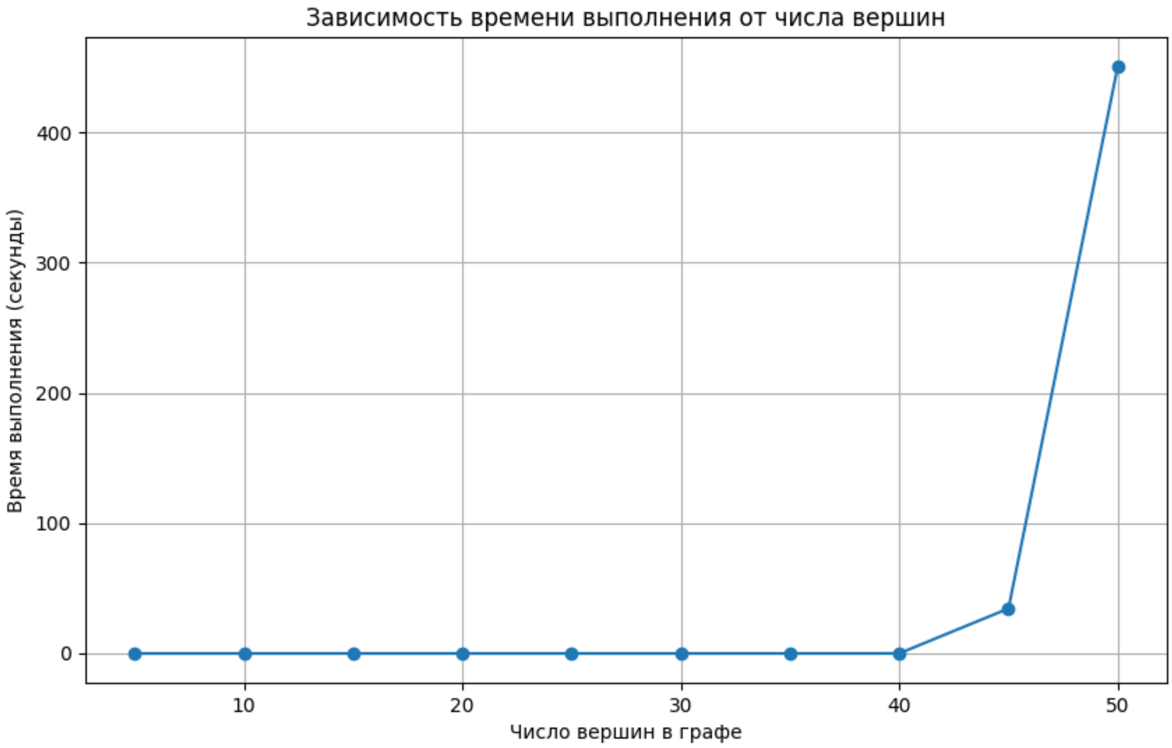
- **Инициализация графа**: задаётся список смежности.
- **Алгоритм DFS**: реализован с учётом рекурсии.
- **Вывод результата**: отображается максимальный путь и его вес.

Инструкции по использованию:

1. Установите Python 3.6+.
2. Запустите программу, указав вершины графа и начальную/конечную точки.
3. Ознакомьтесь с результатами в консоли.

5. Вычисление сложности алгоритма

- **Временная сложность**: (O(V + E)), где (V) — количество вершин, (E) — количество рёбер.
- **Пространственная сложность**: (O(V)) для хранения списка смежности и стека рекурсии.



6. Тестовый пример с решением

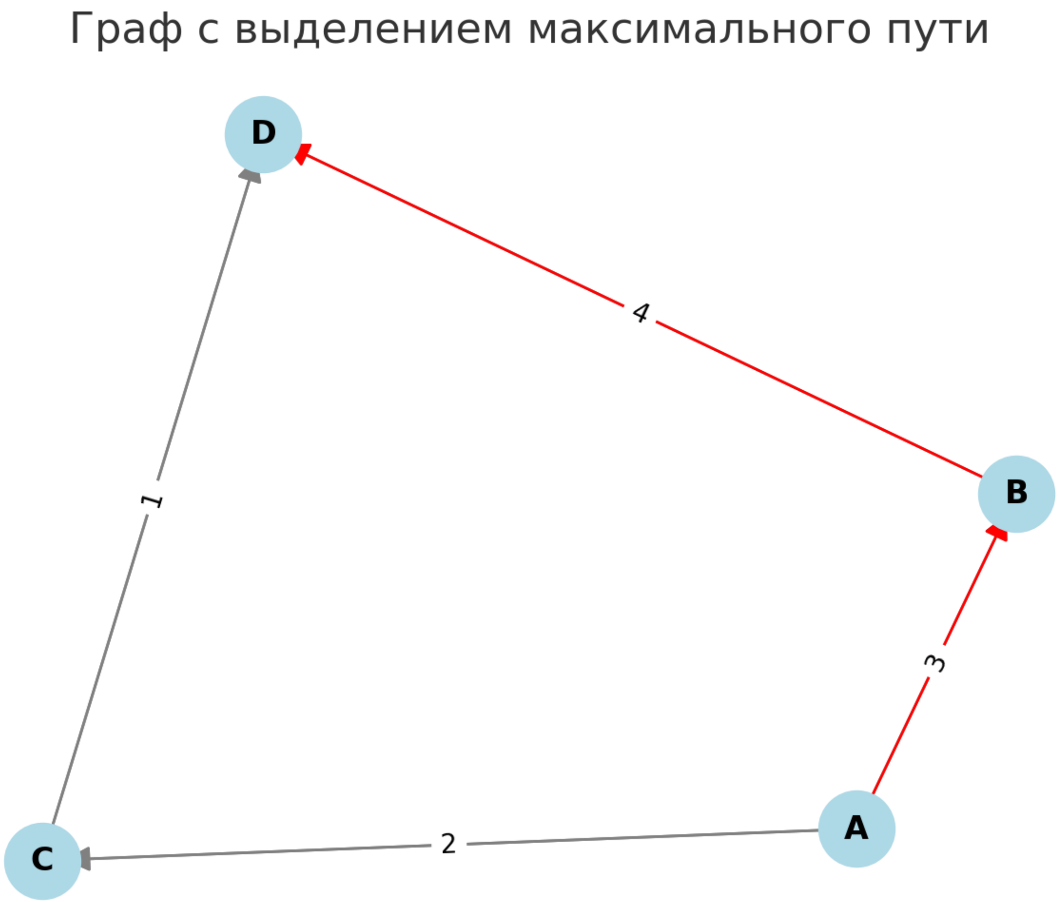
Пример:

Граф:

- Вершины: A, B, C, D.
- Рёбра:
  - A → B (вес 3),
  - A → C (вес 2),
  - B → D (вес 4),
  - C → D (вес 1).

Начальная вершина: A.
Конечная вершина: D.

Результат: Максимальный путь: A → B → D.
Вес пути: 7.



7. Скриншоты программы

```
# Пример работы
g = Graph()
g.add_edge("A", "B", 3)
g.add_edge("A", "C", 2)
g.add_edge("B", "D", 4)
g.add_edge("C", "D", 1)

start, end = "A", "D"
result = g.find_max_path(start, end)
print(f"Максимальный путь из {start} в {end}: {result}")
```

```
[Running] python -u "c:\Users\User\Documents\Work\Algorithms-Practise\general\DFS\DFS.py"
Максимальный путь из A в D: 7
```

8. Прикладная задача

Задача: Оптимизация маршрутов грузоперевозок

- **Сценарий**: Компания ищет маршрут между пунктами с максимальной пропускной способностью.
- **Описание**: Вершины графа представляют пункты доставки, рёбра — маршруты, веса рёбер — грузоподъёмность.
- **Решение**: Алгоритм находит маршрут с максимальным весом между двумя пунктами.