МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРИКЛАДНОЙ МАТЕМАТИКИ
КАФЕДРА МАТЕМАТИЧЕСКОЙ КИБЕРНЕТИКИ

КУРСОВАЯ РАБОТА

НАХОЖДЕНИЕ МАКСИМАЛЬНОГО ПУТИ В НАГРУЖЕННОМ ГРАФЕ

Студент: Большаков М. В.
Группа: М8О-103Б

Преподаватель: доц. Смерчинская С. О.

Оценка:

Дата:

Москва
2021

Задание

Вариант 5

1. Определить для орграфа, заданного матрицей смежности:

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

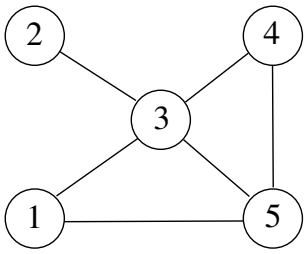
   а) матрицу односторонней связности;

   б) матрицу сильной связности;

   в) компоненты сильной связности;

   г) матрицу контуров.

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



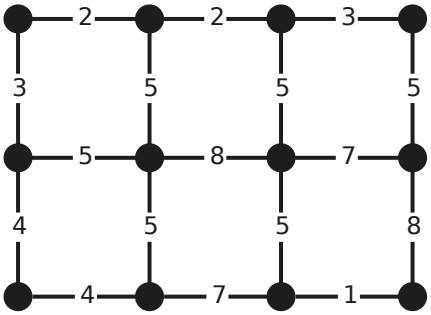
3. Используя алгоритм "фронта волны", найти все минимальные пути из первой вершины в последнюю орграфа, заданного матрицей смежности.

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

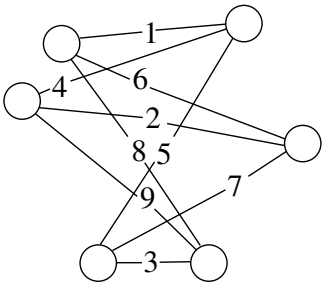
4. Используя алгоритм Форда, найти минимальные пути из первой вершины во все достижимые вершины в нагруженном графе, заданном матрицей длин дуг.

$$\begin{pmatrix} \infty & 4 & 6 & 3 & \infty & \infty & \infty \\ 10 & \infty & 2 & \infty & 3 & \infty & \infty \\ \infty & 2 & \infty & 2 & 1 & 4 & 7 \\ \infty & \infty & 2 & \infty & \infty & 7 & \infty \\ \infty & \infty & 1 & \infty & \infty & \infty & 4 \\ \infty & \infty & 4 & \infty & \infty & \infty & 2 \\ 4 & \infty & 3 & \infty & 5 & 7 & \infty \end{pmatrix}$$

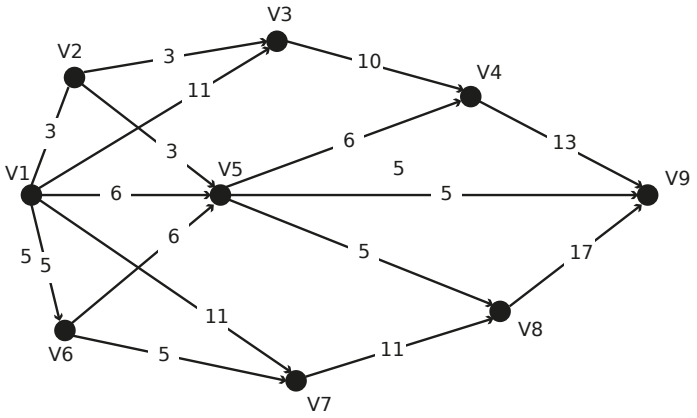
5. Найти остовное дерево с минимальной суммой длин входящих в него ребер.



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 $E1$ и $E2$, а остальные элементы являются сопротивлениями. Используя закон Ома, и, предполагая внутренние сопротивления источников тока равными нулю, получить систему уравнений для токов.



7. Построить максимальный поток по транспортной сети.



8. Нахождение максимального пути в нагруженном графе.

Задание №1.

а) Найдем матрицу односторонней связанности по формуле:
$T = E \vee A \vee A^2 \vee A^3$.

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

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

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

Найдем матрицу односторонней связанности итерационным методом:

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

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

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

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

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

б) Найдем матрицу сильной связанности с помощью формулы: $\bar{S} = T \& T^T$

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

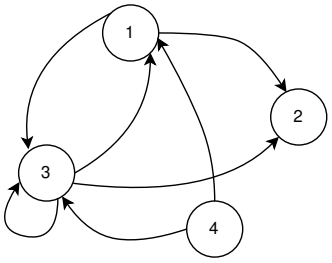
в) Найдем компоненты сильной связанности:

$$\begin{matrix} \downarrow & & \downarrow & \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ \{v_1, v_3\} \end{matrix} \sim \begin{matrix} & \downarrow & & \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ \{v_2\} \end{matrix} \sim \begin{matrix} & & & \downarrow \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ \{v_4\} \end{matrix}$$

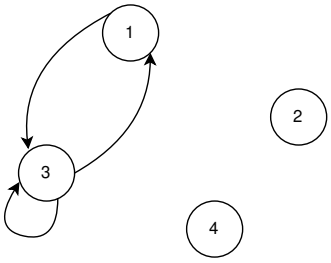
г) Найдем матрицу контуров: $K = \bar{S} \& A$

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

д)

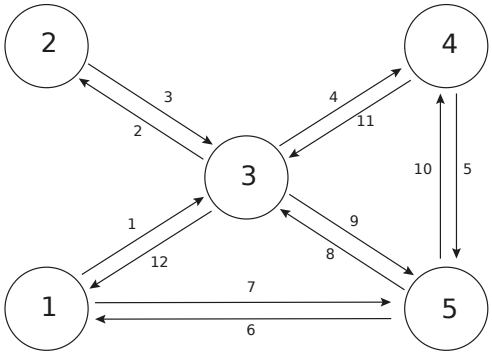


граф



компоненты сильной связанности

Задание №2.



Маршрут обхода:

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

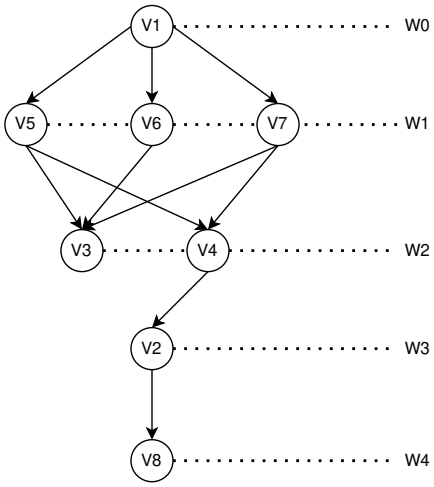
Задание №3.

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

Определим волны:
1.$W_0 = \{V_1\}$
2.$W_1 = \Gamma(W_0) = \Gamma(V_1) = \{V_5, V_6, V_7\}$
3.$W_2 = \Gamma(W_1) = \{V_3, V_4\}$
4.$W_3 = \Gamma(W_2) = \{V_2\}$
5.$W_4 = \Gamma(W_3) = \{V_8\}$

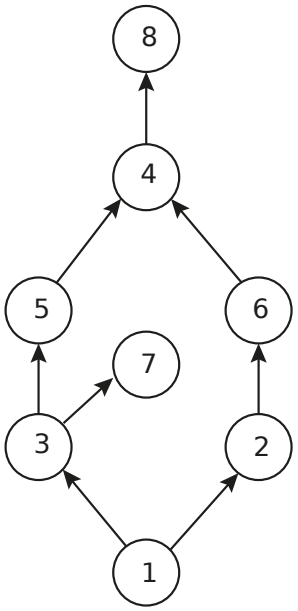
Восстановление пути:
1.$\omega_4 = \{V_8\}$
2.$\omega_3 = \Gamma(W_3) \cap \Gamma^{-1}(V_8) = \{V_2\}$
3.$\omega_2 = \Gamma(W_2) \cap \Gamma^{-1}(V_2) = \{V_4\}$
4.$\omega_1 = \Gamma(W_1) \cap \Gamma^{-1}(V_4) = \{V_5, V_7\}$
5.1.$\omega_0^1 = \Gamma(W_3) \cap \Gamma^{-1}(V_5) = V_1$
5.2.$\omega_0^2 = \Gamma(W_3) \cap \Gamma^{-1}(V_7) = V_1$

Кратчайших путей 2:
1.$V_1 \to V_5 \to V_4 \to V_2 \to V_8$
2.$V_1 \to V_7 \to V_4 \to V_2 \to V_8$



Задание №4.

$$\left(\begin{array}{cccccccc|cccccccc}
\infty & 3 & 5 & \infty & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & 1 & 9 & \infty & 5 & \infty & \infty & \infty & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
13 & 1 & \infty & \infty & 4 & \infty & 3 & \infty & \infty & 5 & 4 & 4 & 4 & 4 & 4 & 4 \\
\infty & \infty & \infty & \infty & 2 & \infty & \infty & 3 & \infty & \infty & 12 & 11 & 11 & 11 & 11 & 11 \\
\infty & \infty & \infty & 2 & \infty & \infty & \infty & 6 & \infty & \infty & 9 & 9 & 9 & 9 & 9 & 9 \\
\infty & \infty & \infty & 3 & \infty & \infty & 2 & \infty & \infty & \infty & 8 & 8 & 8 & 8 & 8 & 8 \\
\infty & \infty & \infty & \infty & 2 & 2 & \infty & \infty & \infty & \infty & 8 & 7 & 7 & 7 & 7 & 7 \\
2 & 3 & \infty & 5 & 4 & \infty & 8 & \infty & \infty & \infty & \infty & 15 & 13 & 13 & 13 & 13
\end{array}\right)$$



Восстановление пути:

$\lambda_8^{(4)} = \lambda_4^{(3)} + C_{48}$

$\lambda_4^{(3)} = \lambda_5^{(2)} + C_54 = \lambda_6^{(2)} + C_{64}$

$\lambda_5^{(2)} = \lambda_3^{(1)} + C_{35}$

$\lambda_3^{(1)} = \lambda_1^{(0)} + C_{13}$

$\lambda_6^{(2)} = \lambda_2^{(1)} + C_{26}$

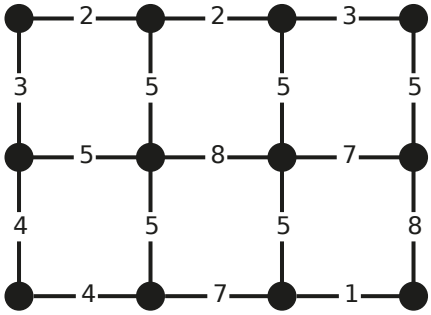
$\lambda_2^{(1)} = \lambda_1^{(0)} + C_{12}$

$\lambda_7^{(3)} = \lambda_3^{(1)} + C_{37}$

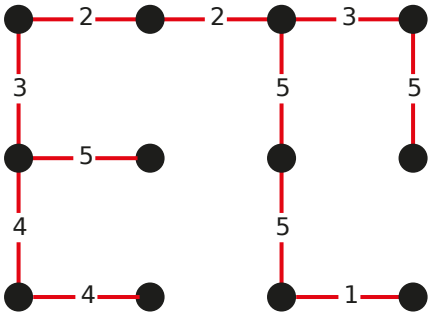
Тогда пути:

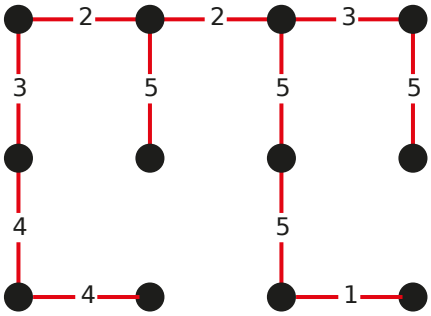
1. В вершину 2: $V_1 \to V_2$
2. В вершину 3: $V_1 \to V_3$
3. В вершину 4: $V_1 \to V_2 \to V_6 \to V_4$
4. В вершину 4: $V_1 \to V_3 \to V_5 \to V_4$
5. В вершину 5: $V_1 \to V_3 \to V_5$
6. В вершину 6: $V_1 \to V_2 \to V_6$
7. В вершину 7: $V_1 \to V_3 \to V_7$
8. В вершину 8: $V_1 \to V_2 \to V_6 \to V_4 \to V_8$
9. В вершину 8: $V_1 \to V_3 \to V_5 \to V_4 \to V_8$

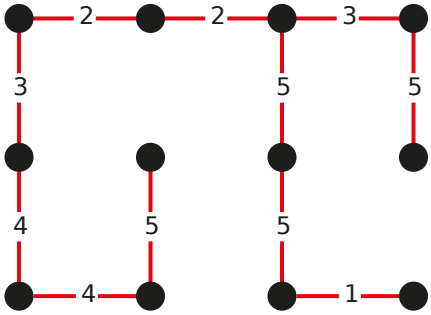
Задание №5.



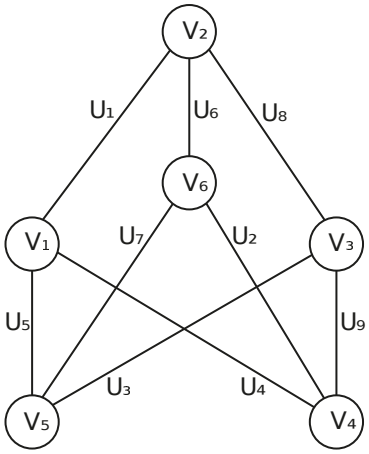
1. Выбрать все вершины графа
2. Добавить все дуги, имеющие минимальный вес — 1, циклов нет
3. Добавить все дуги, имеющие среди оставшихся минимальный вес — 2, циклов нет
4. Добавить все дуги, имеющие среди оставшихся минимальный вес — 3, циклов нет
5. Добавить все дуги, имеющие среди оставшихся минимальный вес — 4, циклов нет
6. Добавляем все дуги, имеющие минимальный вес – 5, так, чтобы не было циклов. Получаем три возможных вариантов остовных деревьев минимального веса. Минимальный вес остовного дерева $L(D)$=39.



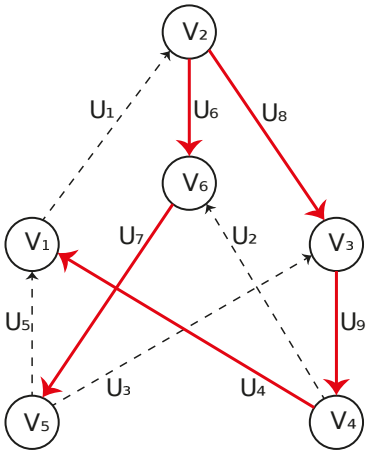




Задание №6.



Зададим ореинтацию для графа и выделим остовное дерево:



$$\left.\begin{array}{l} n=6 \\ m=9 \\ p=1 \end{array}\right| \Rightarrow \nu(G) = m - n + p = 4$$

Вектор-циклы:

$\mathfrak{D} + U_1 : \mu(U_1) = V_1\ V_2\ V_3\ V_4\ V_1$
$C_1 = (1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1)$

$\mathfrak{D} + U_5 : \mu(U_5) = V_5\ V_1\ V_4\ V_3\ V_2\ V_6\ V_5$
$C_2 = (0\ 0\ 0\ \text{-}1\ 1\ 1\ 1\ \text{-}1\ \text{-}1)$

$\mathfrak{D} + U_2 : \mu(U_2) = V_4\ V_6\ V_2\ V_3\ V_4$
$C_3 = (0\ 1\ 0\ 0\ 0\ \text{-}1\ 0\ 1\ 1)$

$\mathfrak{D} + U_3 : \mu(U_3) = V_5\ V_3\ V_2\ V_6\ V_5$
$C_4 = (0\ 0\ 1\ 0\ 0\ 1\ 1\ \text{-}1\ 0)$

Цикломатическая матрица:

$$C = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & -1 & 1 & 1 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & -1 & 0 \end{pmatrix} \quad C \times U = 0 \quad \Rightarrow \begin{cases} \varepsilon_1 + U_4 + U_8 + U_9 = 0 \\ -U_4 + \varepsilon_2 + U_6 + U_7 - U_8 - U_9 = 0 \\ U_2 - U_6 + U_8 + U_9 = 0 \\ U_3 + U_6 + U_7 - U_8 = 0 \end{cases}$$

$$I \times R = 0 \quad \Rightarrow \begin{cases} -I_4 r_4 - I_8 r_8 - I_9 r_9 = \varepsilon_1 \\ I_4 r_4 + I_8 r_8 + I_9 r_9 - I_6 r_6 - I_7 r_7 = \varepsilon_2 \\ I_2 r_2 - I_6 r_6 + I_8 r_8 + I_9 r_9 = 0 \\ I_3 r_3 + I_6 r_6 + I_7 r_7 - I_8 r_8 = 0 \end{cases}$$

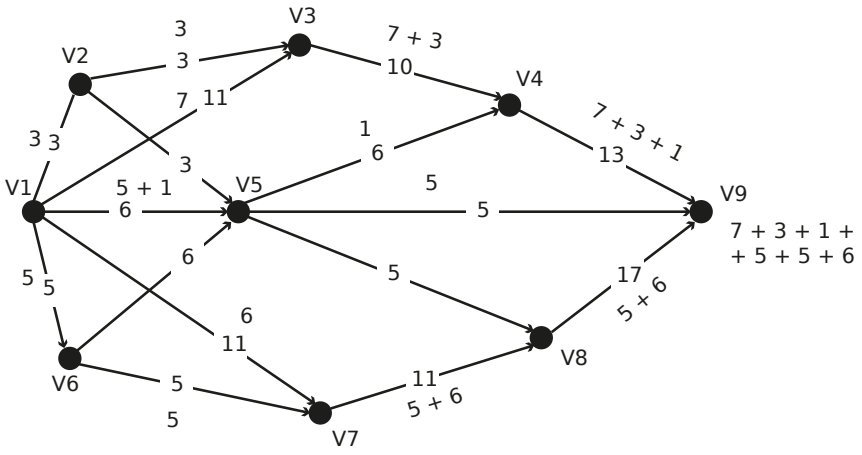
$$B = \begin{pmatrix} -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{pmatrix} \quad B \times I = 0 \quad \Rightarrow \begin{cases} -I_1 + I_4 + I_5 = 0 \\ I_1 - I_6 - I_8 = 0 \\ I_3 + I_8 - I_9 = 0 \\ I_9 - I_2 - I_4 = 0 \\ I_7 - I_3 - I_5 = 0 \\ I_2 + I_6 - I_7 = 0 \end{cases}$$

Общая система:

$$\begin{cases} -I_4 r_4 - I_8 r_8 - I_9 r_9 = \varepsilon_1 \\ I_4 r_4 + I_8 r_8 + I_9 r_9 - I_6 r_6 - I_7 r_7 = \varepsilon_2 \\ I_2 r_2 - I_6 r_6 + I_8 r_8 + I_9 r_9 = 0 \\ I_3 r_3 + I_6 r_6 + I_7 r_7 - I_8 r_8 = 0 \\ -I_1 + I_4 + I_5 = 0 \\ I_1 - I_6 - I_8 = 0 \\ I_3 + I_8 - I_9 = 0 \\ I_9 - I_2 - I_4 = 0 \\ I_7 - I_3 - I_5 = 0 \end{cases}$$

Задание №7.

1) Построение полного потока.



Ищем пути из источника в сток, не содержащие насыщенных дуг.

1. $V_1 - V_2 - V_3 - V_4 - V_9$
   $min\{3, 3, 10, 13\} = 3$

2. $V_1 - V_6 - V_7 - V_8 - V_9$
   $min\{5, 5, 11, 17\} = 5$

3. $V_1 - V_5 - V_9$
   $min\{6, 5\} = 5$

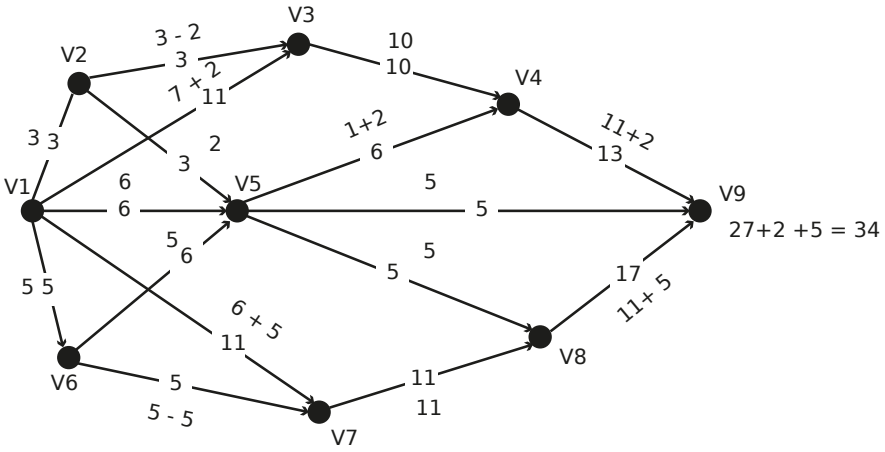
4. $V_1 - V_3 - V_4 - V_9$
   $min\{11, 7, 10\} = 7$

5. $V_1 - V_7 - V_8 - V_9$
   $min\{11, 6, 12\} = 6$

6. $V_1 - V_5 - V_4 - V_9$
   $min\{1, 4, 3\} = 1$

Величина полного потока $\Phi_{\text{пол.}} = 3 + 5 + 5 + 7 + 6 + 1 = 27$

2) Построение максимального потока.



Найдем увеличивающие цепи.

1. $V_1 - V_3 - V_2 - V_5 - V_4 - V_9$
   $\triangle_1 = min\{4, \underline{3}, 3, 2\} = 2$

2. $V_1 - V_7 - V_6 - V_5 - V_8 - V_9$
   $\triangle_2 = min\{5, \underline{5}, 6, 5, 6\} = 5$

Минус внизу помечает дуги, по которым поток можно уменьшить на 4.

Величина потока увеличилась на 6.

Величина максимального потока $\Phi_{\text{макс.}} = 27 + 2 + 5 = 34$

Задание №8.

Теоретические сведения. Описание алгоритма.

Задан ориентированный нециклический нагруженный граф.

Ориентированный граф (орграф) — это граф, в котором каждое ребро имеет направление. Орграф может служить, например, моделью системы дорог с односторонним движением.

Нагруженный граф — это граф, у которого каждому ребру сопоставлено некоторое число. В некоторых задачах это число может обозначать расстояние между вершинами, или время перехода от одной вершины к другой, или еще что-либо.

Ориентированный ациклический — орграф, в котором отсутствуют направленные циклы, но могут быть «параллельные» пути, выходящие из одного узла и разными путями приходящие в конечный узел.

Для ввода в программу данного графа используется матрица смежности данного графа.

Матрица смежности графа G с конечным числом вершин — это квадратная матрица A размера $n \times n$, в которой значение элемента a_{ij} равно числу рёбер из i-й вершины графа в j-ю вершину.

Сам алгоритм поиска максимального пути основан на обходе в глубину с дополнением, идет обход в глубину по тем дочерним элементам, у которых все родительские элементы были использованы. Для проверки этого используется маска. Также для каждой вершины запоминается предыдущая вершина для восстановления пути.

Поиск в глубину (DFS) — один из методов обхода графа. Стратегия поиска в глубину, как и следует из названия, состоит в том, чтобы идти «вглубь» графа, насколько это возможно. Алгоритм поиска описывается рекурсивно: перебираем все исходящие из рассматриваемой вершины рёбра. Если ребро ведёт в вершину, которая не была рассмотрена ранее, то запускаем алгоритм от этой нерассмотренной вершины, а после возвращаемся и продолжаем перебирать рёбра. Возврат происходит в том случае, если в рассматриваемой вершине не осталось рёбер, которые ведут в нерассмотренную вершину. Если после завершения алгоритма не все вершины были рассмотрены, то необходимо запустить алгоритм от одной из нерассмотренных вершин.

Битовая маска — определённые данные, которые используются для маскирования — выбора отдельных битов или полей из нескольких битов из двоичной строки или числа. Битовую маску возможно использовать для получения значения бита (в данном случае для проверки родителей).

Для вывода графа используется алгоритм обхода в ширину. При котором вершины графа разбиваются на уровни.

Поиск в ширину (BFS) — один из методов обхода графа. Пусть задан граф G = (V, E) и выделена исходная вершина s. Алгоритм поиска в ширину систематически обходит все ребра G для «открытия» всех вершин, достижимых из s вычисляя при этом расстояние (минимальное количество рёбер) от до каждой достижимой из s вершины. Алгоритм работает как для ориентированных, так и для неориентированных графов.

Далее для каждого уровня строится полуокружность, на котором находятся вершины. Если вершина входящая в этот уровень имеет дочерние уровни, то строится прямая на этом этой же высоте до крайней точки x этого уровня.

Алгоритм нахождения максимального пути.

1. Запускается цикл по вершинам графа, который работает, пока нельзя пройтись по всем вершинам.

2. В цикле из п.1 проверяется для каждой вершины проверяется есть ли отцы.
   Если отцов нет, то запускается для вершины основной алгоритм.
   После выполнения алгоритма цикл запускается заново.

3. В запущенном алгоритме из п.2 запускается цикл для каждой дочерней вершины.

4. Если у вершины из цикла (п.3) путь меньше, чем текущий, то путь запоминается и для дочерней вершины запоминается предыдущая вершина.
   Если у вершины все отцы были посещены, то для этой вершины запускается основной алгоритм из п.2.

5. Каждая вершина, в запущенном алгоритме из п.2. , помечается посещенной (с помощью специальной маски).

6. Если у вершины нет дочерних элементов и если путь в вершину больше текущего максимального пути, то запоминается новый максимальный путь и запоминается текущая вершина, как конечная.

Таким образом после выполнения алгоритма будет числовое значение максимального пути, а также вершина конца этого пути, с помощью которой можно восстановить путь, т.к. в п.3 запоминалась предыдущая вершина.

Алгоритм построения графа.

1. Запускается алгоритм обхода в ширину из начала максимального пути.

2. После этого часть вершин будет распределена на уровни.

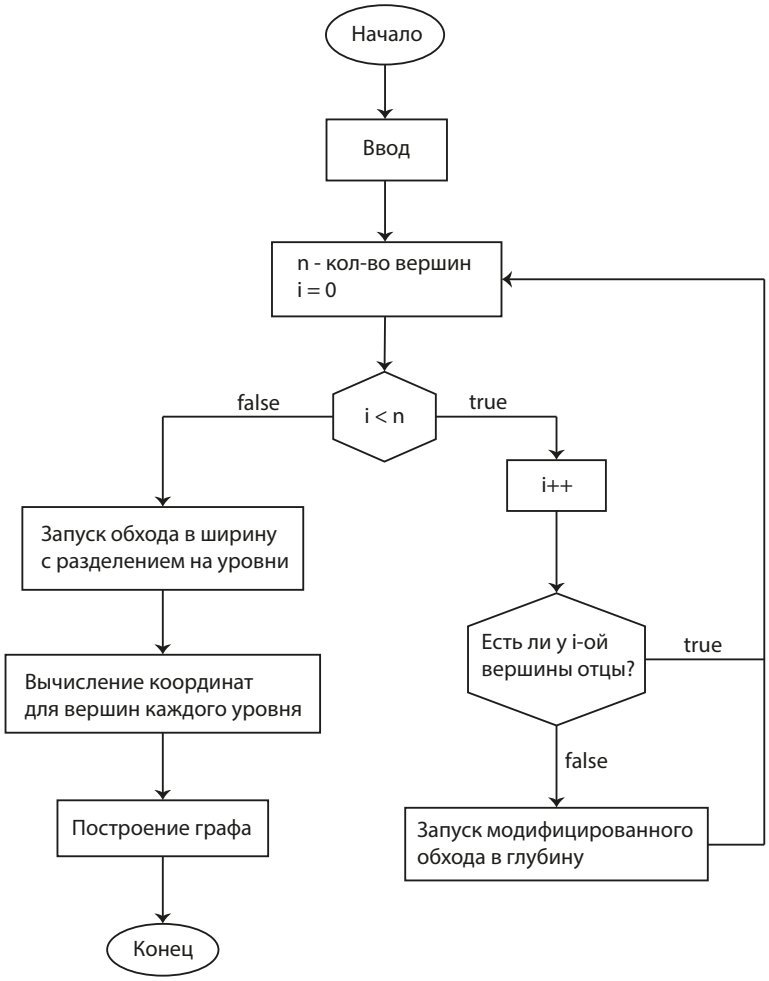
3. Для вершин, которые не входят не в один уровень и у которых нет отцов, тоже запускается алгоритм обхода в ширину.

4. В итоге все вершины распределены по уровням, после этого вычисляется полуокружности для каждого уровня, радиус которых зависит от количества элементов. Если у вершины в уровне есть дочерние элементы проводится прямая параллельная оси oX, до самой крайней точки этого уровня. Из построенной прямой будут проводится стрелки к следующим элементам. Если стрелок исходящих из этой прямой больше одной, то строится мини точка на конце прямой.

5. Производится восстановление пути. После этого запоминаются все элементы лежащие на пути в графе.

6. Если поставлен в фильтрах выбрано выделение максимального пути, то элементы запомненные в п.5 выделяются красным цветом.

Логическая блок-схема.



Оценка сложности алгоритма.

Алгоритм нахождения макисмального пути в графе имеет сложность O(N + M), где N — количество вершин в графе, а M — количество ребер в графе, т.к. по алгоритму придется посетить каждую вершину и пройтись по каждому ребру один раз.

Инструкция к программе.

Программа написана на языке C++ с использованием библиотеки Qt, благодаря чему является кроссплатформенной, т.е. можно запустить на любом ПК.

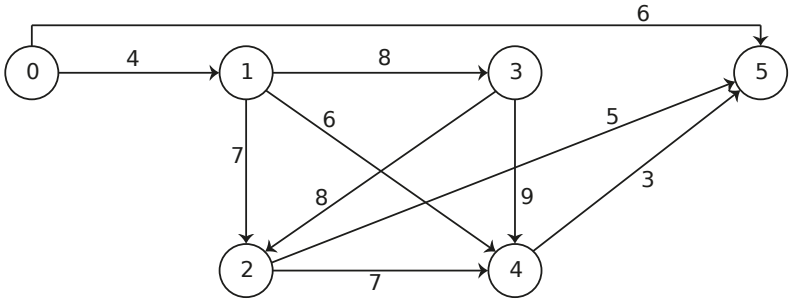
Программа имеет простой и понятный интерфейс. В левой части находится ввод матрицы смежности, фильтры для визуализации графа(выделение максимального пути, а также изменение масштаба вывода графа), кнопка для вычисления графа и его построения. А также есть вывод итогового результата. В левой части находится место для вывода графа. Программа имеет адаптивный интерфейс, т.е. при изменении размеров окна компоненты подстраиваются под новый размер. Также имеется ограничение на минимальный размер окна, для корректного отображения информации. Также в программе предусмотренно осуществление пользователем ошибок при вводе данных, и программа при некорректном вводе выводит предупреждения.

Тестовые примеры. Скриншоты программы.

Пример 1. Задан ациклический орграф с помощью матрицы смежности, необходимо найти максимальный путь.

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

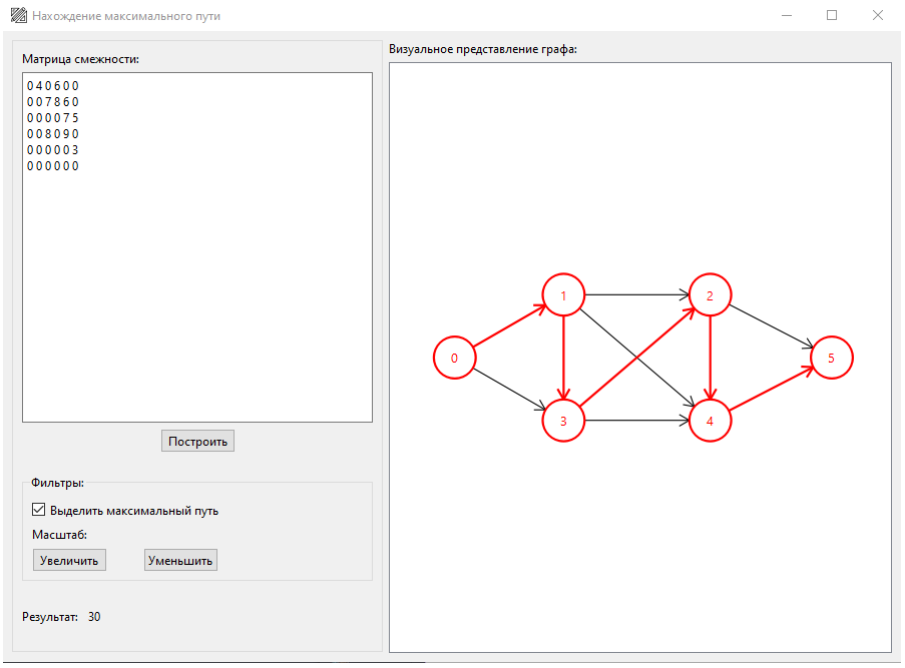
Построим граф для данной матрицы:



Начнем обход графа с вершины 0:

$0 \to 1$ (путь 4). У вершины 1 больше нет неиспользованных родителей, продолжаем алгоритм.
$1 \to 2$ (путь $4 + 7 = 11$). Спускаемся в вершину 1 (есть непосещенный родитель).
$1 \to 3$ (путь $4 + 8 = 12$). Продолжаем алгоритм.
$3 \to 2$ (путь 20 $(12 + 8 > 11)$). Продолжаем алгоритм.
$2 \to 4$ (путь $20 + 7$). Спускаемся в вершину 2.
$2 \to 5$ (путь $20 + 5$). Спускаемся в вершину 3.
$3 \to 4$ (путь 27 $(27 > 12 + 9)$). Спускаемся в вершину 1.
$1 \to 4$ (путь 27 $(27 > 4 + 6)$). Продолжаем алгоритм.
$4 \to 5$ (путь 30 $(27 + 3 > 25)$). Спускаемся в вершину 0.
$0 \to 5$ (путь 30 $(30 > 6)$). Конец.

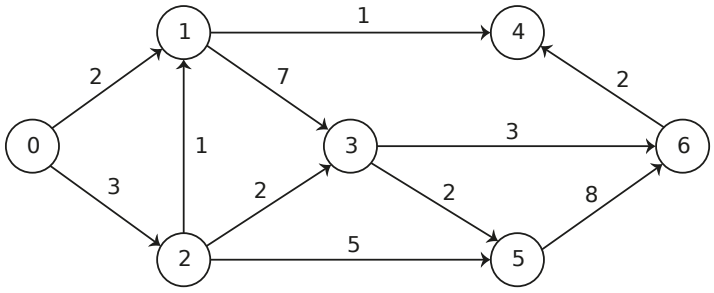
Ответ 30. Путь: $0 \to 1 \to 3 \to 2 \to 4 \to 5$



Пример 1. Задан ациклический орграф с помощью матрицы смежности, необходимо найти максимальный путь.

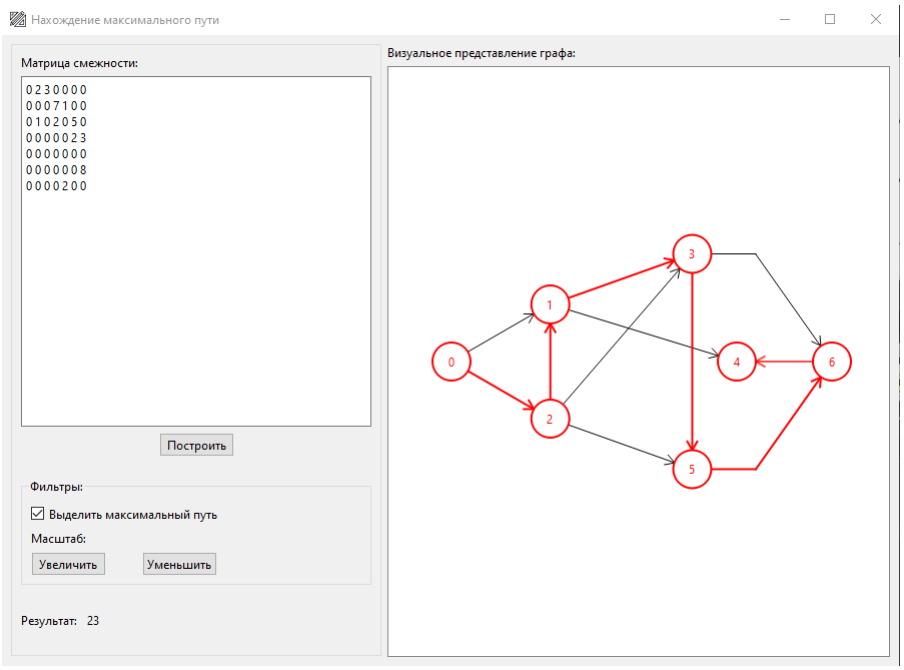
$$\begin{pmatrix} 0 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \end{pmatrix}$$

Построим граф для данной матрицы:



$0 \to 1$ (путь 2). Спускаемся в вершину 0.
$0 \to 2$ (путь 3). Продолжаем алгоритм.
$2 \to 1$ (путь 4 $(3 + 1 > 2)$). Продолжаем алгоритм.
$1 \to 3$ (путь $4 + 7$). Спускаемся в вершину 1.
$1 \to 4$ (путь $4 + 1$). Спускаемся в вершину 2.
$2 \to 3$ (путь 11 $(11 > 2 + 3)$). Продолжаем алгоритм.
$3 \to 5$ (путь $11 + 2$). Спускаемся в вершину 3.
$3 \to 6$ (путь $11 + 3$). Спускаемся в вершину 2.
$2 \to 5$ (путь 13 $(13 > 5 + 3)$). Продолжаем алгоритм.
$5 \to 6$ (путь 21 $(13 + 8 > 14)$). Продолжаем алгоритм.
$6 \to 4$ (путь 23 $(21 + 2 > 5)$). Конец.

Ответ 23. Путь: $0 \to 2 \to 3 \to 5 \to 6 \to 4$



Пример прикладной задачи.

Метод критического пути для планирования набора работ использует построение ориентированного ацикличного графа, в котором вершины представляют узловые события проекта, а дуги представляют работы, которые должны быть выполнены до узлового события и после него. Каждой дуге присваивается вес, равный оценочному времени выполнения работы. В таком графе самый длинный путь от первого узлового события до последнего является критическим путём, который определяет полное время завершения проекта.

Например:
Для создания новой аппаратуры для самолета необходимо выполнить N подзадач, между подзадачами даны веса, равные оценочному времени выполнению работы. Необходимо узнать полное время завершения проекта.