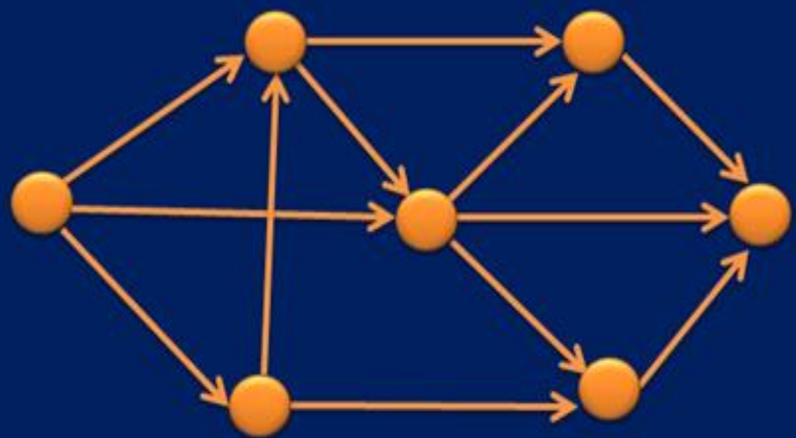
Смерчинская С.О.
Яшина Н.П.

Методические указания к выполнению курсовой работы по теории графов



МОСКВА

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Смерчинская С.О., Яшина Н.П.

Методические указания к выполнению курсовой работы

по теории графов

Москва
2018

УДК 519.17
ББК 22.17
Авторский код

Рецензенты

Смерчинская С.О., Яшина Н.П.

Методические указания к выполнению курсовой работы по теории графов

В учебном пособии рассматриваются основные алгоритмы теории графов, часто используемые для решения прикладных задач. Приводятся краткие теоретические сведения из разделов пути и контуры, деревья и циклы, транспортные сети. Курсовая работа содержит общую часть и индивидуальное задание, требующее не только самостоятельного изучения теоретического материала, но и творческого подхода к его выполнению. Подробно разобран пример выполнения курсовой работы по теории графов.

Пособие предназначено для студентов вузов, изучающих дисциплину «Дискретная математика», раздел «Теория графов и ее приложение».

ОБЩИЕ СВЕДЕНИЯ О КУРСОВОЙ РАБОТЕ

Курсовая работа предназначена для выполнения студентами факультета «Прикладная математика» по дисциплинам «Дискретная математика» и «Теория графов и». Работа содержит семь типовых заданий и одно индивидуальное задание. Выполнение восьмого, индивидуального задания требует самостоятельного изучения теории по литературе, применение изученного метода к решению поставленной задачи, программной реализации алгоритма с применением графических средств.

Правила оформления

Курсовая работа выполняется, а затем сдается преподавателю на листах формата А4. Первая страница – титульный лист оформляется согласно образцу.

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

КУРСОВАЯ РАБОТА

НАЗВАНИЕ ТЕМЫ ЗАДАНИЯ № 8

Студент: Иванов И.И.
Группа 80-101Б

Преподаватель: доц. Петров П.П.

Оценка:

Дата:

Вторая и третья страницы – индивидуальное задание на выполнение курсовой работы. Оформляется согласно приведенному примеру.

Задание

Вариант 30.

1. Определить для орграфа, заданного матрицей смежности:

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

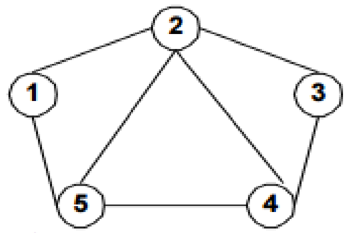
а) матрицу односторонней связности;

б) матрицу сильной связности;

в) компоненты сильной связности;

в) матрицу контуров.

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



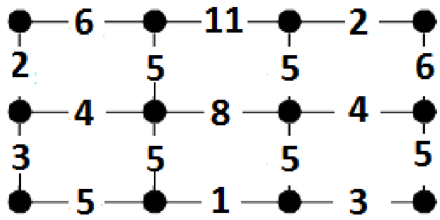
3. Используя алгоритм “фронта волны”, найти все минимальные пути из первой вершины в последнюю орграфа, заданного матрицей смежности.

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

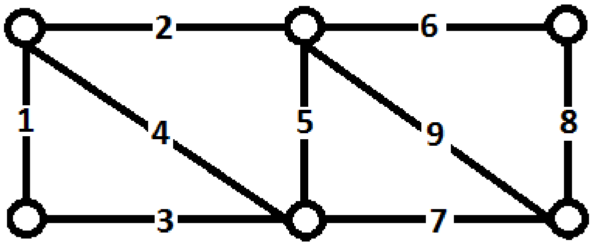
4. Используя алгоритм Форда, найти минимальные пути из первой вершины во все достижимые вершины в нагруженном графе, заданном матрицей длин дуг.

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

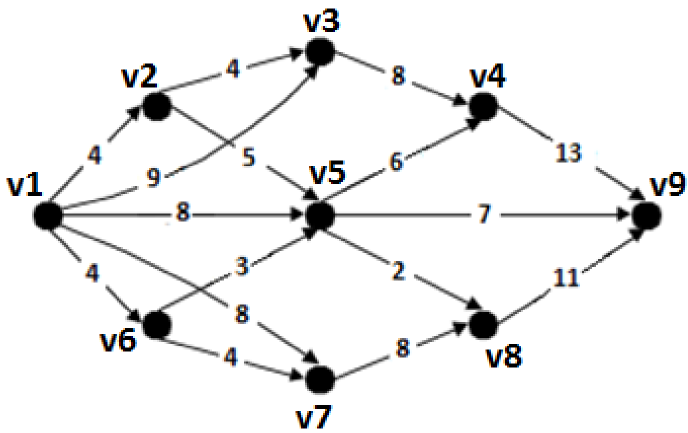
5. Найти остовное дерево с минимальной суммой длин входящих в него ребер.



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 E_1 и E_2, 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. Применение теории графов для ранжирования альтернатив.

Заданы ранжирования альтернатив экспертами.

1. Построить суммарное отношение.
2. Проверить граф на наличие контуров.
3. Разбить граф без контуров на уровни.

Выполнение курсовой работы

С четвертой страницы курсовой работы начинаются решения заданий. Первые семь заданий стандартные для всех студентов, по вариантам отличаются только числовые значения заданных параметров.

Восьмое задание индивидуально для каждого студента. Оформление этого задания должно содержать следующие пункты.

1. Основные понятия и определения по теме работы.
2. Описание алгоритма.
3. Логическая блок-схема.
4. Описание программы и инструкции по работе с ней.
5. Вычисление сложности алгоритма.
6. Тестовый пример с решением.
7. Скриншоты программы для данного примера.
8. Прикладная задача.
9. Текст программы алгоритма.

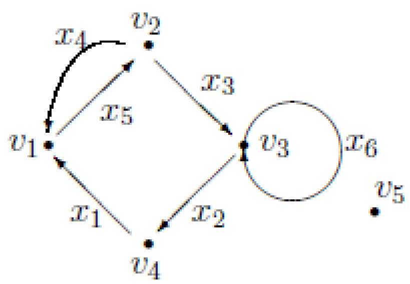
ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ ТЕОРИИ ГРАФОВ

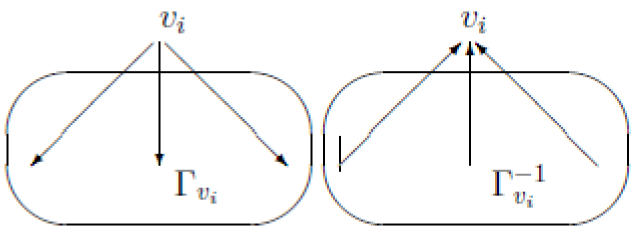
Определение 1. Ориентированным графом $G =< V, X >$ называется упорядоченная пара, где $V = \{v_1, \dots, v_n\}$ конечное непустое множество вершин орграфа, множество $X = \{x_1, \dots x_m\}$ множество дуг орграфа.

Каждая дуга x_k - упорядоченная пара вершин $x_k =< v_i, v_j >$; v_i - начало дуги, v_j - конец дуги $< v_i, v_j >$. Дуга x_k исходит из v_i, заходит в v_j. Вершины v_i, v_j смежны. Дуга x_k инцидентна вершинам v_i и v_j, а вершины инцидентны дуге. Вершина, которая не имеет инцидентных ей дуг - изолированная. Дуга $< v_i, v_i >$ - петля.

Определим множества $\Gamma v_i = \{v_j \mid \exists < v_i, v_j >\in X\}$ и $\Gamma^{-1} v_i = \{v_j \mid \exists < v_j, v_i >\in X\}$.

Пример 1.





$$\Gamma_{v_2} = \{v_1, v_3\}$$

$$\Gamma^{-1}_{v_1} = \{v_2, v_4\}$$

Определение 2. Последовательность дуг графа, такая что начало следующей дуги совпадает с концом предыдущей, называется путем.

Контур - путь, у которого начало первой дуги совпадает с концом последней (замкнутый путь).

Пример 1 (продолжение).

$v_1 \to v_4$ - путь. Описание через дуги: x_5, x_3, x_2; через вершины: v_1, v_2, v_3, v_4.

$v_1 \to v_1$ - контур. Описание через дуги: x_5, x_3, x_2, x_1через вершины: v_1, v_2, v_3, v_4, v_1.

Путь (контур) - простой, если все его дуги различны.

Путь (контур) - элементарный, если все его вершины различны (в контуре - кроме первой и последней).

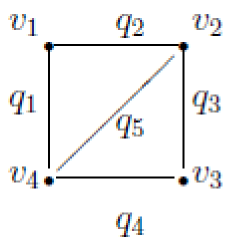
Определение 3. Неориентированный граф $G =< V, Q >$ - конечное непустое множество $V \neq \emptyset, V = \{v_1, \dots, v_n\}$ вершин графа и множество $Q = \{q_1, \dots q_m\}$ ребер графа, где каждое ребро $q_k = (v_i, v_j) = \{v_i, v_j\}$.

Определение 4. Цепь (маршрут) - последовательность ребер, которую заданием ориентации можно превратить в путь.

Цикл - замкнутая цепь (введением ориентации можно превратить в контур).

Цепи и циклы аналогично бывают простые и элементарные.

Пример 2.



Цепь: v_1, v_2, v_3, v_4 или q_2, q_3, q_4

Орграф	**Неориентированный граф**
дуга	ребро
путь	цепь (маршрут)
контур	цикл

Граф можно рассматривать как иллюстрацию бинарных отношений на конечных множествах. Если граф неориентированный, то это симметрическое отношение. Каждому графу соответствует отношение, а каждому отношению соответствует граф.

Матричное задание графов

Способы матричного задания графов:

1. матрица смежности;
2. матрица инцидентности.

Определение 5. ***Матрица смежности ориентированного графа*** – квадратная матрица порядка n (n - число вершин графа): $A = ||a_{ij}||$ с элементами

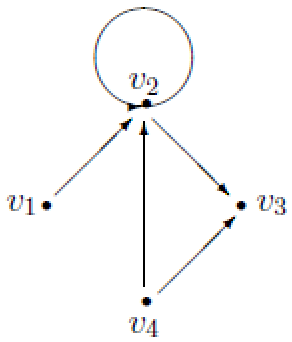
$$a_{ij} = \begin{cases} 1, \text{если } \exists < v_i, v_j > \in X \\ 0 \text{ в противном случае} \end{cases}$$

Бинарное отношение ρ можно задать матрицей смежности соответствующего орграфа.

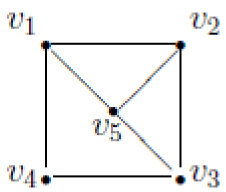
Определение 6. ***Матрица смежности неориентированного графа*** – квадратная (симметрическая) матрица порядка n: $A = ||a_{ij}||$ с элементами

$$a_{ij}=\begin{cases}1, \text{если } \exists(v_i,v_j)\in Q\\ 0 \text{ в противном случае}\end{cases}$$

Пример 3.



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



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

Матрица смежности неориентированного графа всегда симметрическая.

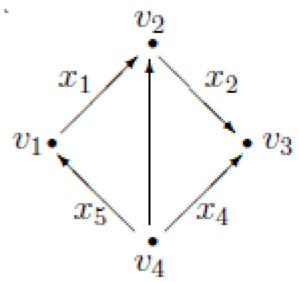
Определение 7. Матрица инцидентности ориентированного графа – матрица порядка $n \times \mathrm{m}$ (n – число вершин графа; m – число его $\frac{\text{ребер}}{\text{дуг}}$) : $B_{n\times m}=||b_{ij}||$ с элементами

$$b_{ij}=\begin{cases}-1, \text{если дуга } x_j \text{ исходит из } v_i\\ 1, \text{если дуга } x_j \text{ заходит в } v_i\\ 0, \text{если дуга } x_j \text{ не инсцидентна } v_i\end{cases}$$

Определение 8. Матрица инцидентности неориентированного графа - матрица порядка $n \times \mathrm{m}$: $B_{n\times m}=||b_{ij}||$ с элементами

$$b_{ij}=\begin{cases}1, \text{если ребро } q_j \text{ инсцидентно } v_i\\ 0, \text{если ребро } q_j \text{ не инсцидентно } v_i\end{cases}.$$

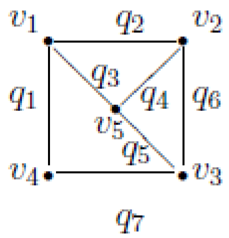
Пример 4.



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & -1 \end{pmatrix} \end{array}$$

Свойство: сумма строк матрицы инцидентности орграфа равна нулевой строке.

$$rg\, B = n - 1$$



$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

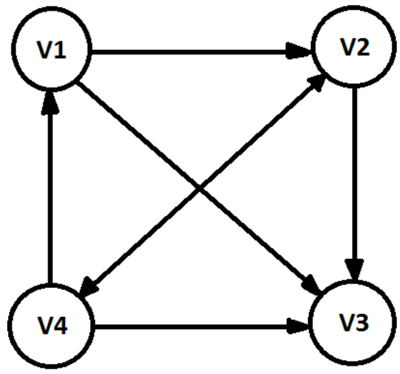
Выполнение курсовой работы

ЗАДАНИЕ № 1

Связность в орграфе

Определить для орграфа, заданного матрицей смежности:

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



а) матрицу односторонней связности;

б) матрицу сильной связности;

в) компоненты сильной связности;

г) матрицу, содержащую все дуги контуров.

Теоретические сведения

Определение 1.1. Ориентированный граф $G =< V, X >$ - односторонне связный, если для любой пары вершин $v_i, v_j (i \neq j)$ существует путь из v_i в v_j либо из v_j в v_i.

Опр. Ориентированный граф $G =< V, X >$ - сильно связный, если для любой пары вершин $v_i, v_j (i \neq j)$ существуют путь и из v_i в v_j и из v_j в v_i.

Аналогично определяются компоненты односторонней и сильной связной.

Определение 1.2. Матрица односторонней связности $T = ||t_{ij}||$ орграфа - квадратная матрица порядка n с элементами

$$t_{ij} = \begin{cases} 1, \text{если существует путь из } v_i \text{ в } v_j \\ \quad 0 \text{ в противном случае} \end{cases}$$

Определение 1.3. Матрица сильной связности $\bar{S} = \left||\bar{s}_{ij}|\right|$ орграфа - квадратная матрица порядка n с элементами

$$\bar{s}_{ij} = \begin{cases} 1, \text{если существует путь из } v_i \text{ в } v_j \text{ и из } v_j \text{ в } v_i \\ \quad 0 \text{ в противном случае} \end{cases}$$

Алгоритмы Уоршалла нахождения матрицы T односторонней связности орграфа по матрице смежности A

$$C \vee D = ||c_{ij} \vee d_{ij}|| \qquad C\&D = ||c_{ij}\&d_{ij}|| \qquad C * D = ||q_{ij}|| \quad q_{ij} = \bigvee_{k=1}^{n} (c_{ik}\&d_{kj})$$

Первый алгоритм Уоршалла.

$$T = E \vee A \vee A^2 \vee \dots \vee A^{n-1},$$

где $A^k = \underbrace{A * A * \dots * A}_{k}$

Второй алгоритм Уоршалла (итерационный). Незаменим при программной реализации: $O(n^3) -$ вычислительная сложность.

Находим $T^{(0)}, T^{(1)}, \dots, T^{(n)}$. Причем $T^{(n)} = T$.

0) $T^{(0)} = E \vee A$;

……………………….

$k)$ $T^{(k)} = ||t_{ij}^{(k)}||$, $t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)}\&t_{kj}^{(k-1)})$

……………………………..

$n)$ $T^{(n)} = T$.

$\bar{S} = \left|\left|\bar{s}_{ij}\right|\right|$ - матрица сильной связности вычисляется через матрицу односторонней связности T по формуле:

$$\bar{S} = T\&T^T.$$

Все дуги контуров графа определяются по формуле

$$K = \bar{S}\&A.$$

Алгоритм нахождения вершин компонент сильной связности орграфа по матрице сильной связности $\bar{S}$.

1. Соответствующие единицам первой строки номера вершины принадлежат первой компоненте сильной связности, число компонент $k = 1$. В матрице $\bar{S}$ обнуляем столбцы (можно строки), у которых в первой строке стоят единицы. Получаем матрицу $\bar{S}_1$.
2. Если $\bar{S}_1 \not\equiv (0)$, то $k = k + 1$. Находим не нулевую строку $\bar{S}_1$. Пусть ее номер i_1. Соответствующие единицам строки i_1 номера вершин принадлежат второй компоненте связности. В матрице $\bar{S}_1$ обнуляем столбцы (строки), у которых в строке i_1 стоят единицы. Получаем матрицу $\bar{S}_2$.

....

Процесс заканчивается, когда матрица $\bar{S}_t = (0)$. В этом случае все вершины графа будут принадлежать какой-либо компоненте сильной связности.

Для нахождения компонент сильной связности восстанавливаем дуги между вершинами внутри каждой компоненты.

Решение.

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

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

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

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

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

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

Найдем матрицу односторонней связности по итерационному алгоритму Уоршалла.

$$k = 0$$

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

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

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

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

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

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

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

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

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

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

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

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

в) Находим вершины компонент сильной связности и сами компоненты.

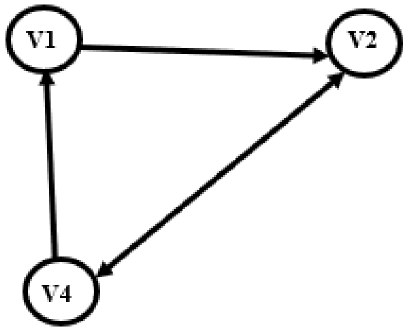
1. Выбираем первую строку, как ненулевую в матрице сильной связности

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

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

$$\{ v_1, v_2, v_4 \}.$$

Восстанавливаем дуги между вершинами v_1, v_2, v_4 и получаем первую компоненту сильной связности:



Обнуляем первый, второй и четвертый столбец матрицы $\overline{S}$. Получаем матрицу

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

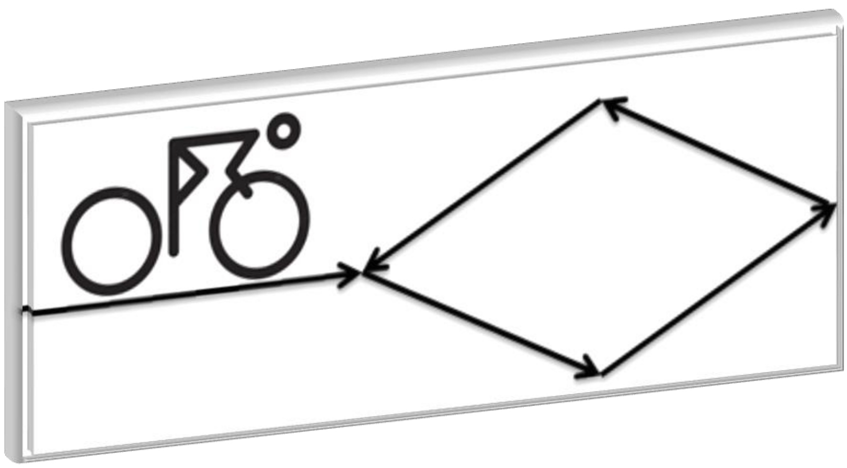
2. Ищем ненулевую строку матрицы $\overline{S}_1$: это третья строка. Единица одна – в третьем столбце. Следовательно, вторая компонента сильной связности: $\{v_3\}$.

Обнуляем третий столбец матрицы $\overline{S}_1$, получаем нулевую матрицу. Следовательно, других компонент сильной связности нет.

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

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

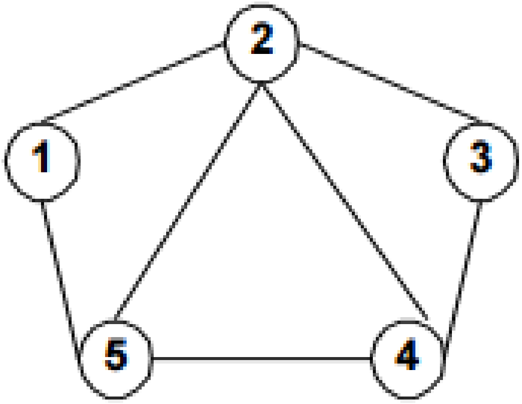
Следовательно, дуги $< v_1, v_2 >, < v_2, v_4 >, < v_4, v_1 >, < v_4, v_2 >$ принадлежат какому-либо контуру исходного графа.



ЗАДАНИЕ № 2

Алгоритм Терри

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



Теоретические сведения

Алгоритм Терри

Пусть задан неориентированный связный граф $G =< V, Q >$. Найдём маршрут из вершины v_i в вершину v_j.

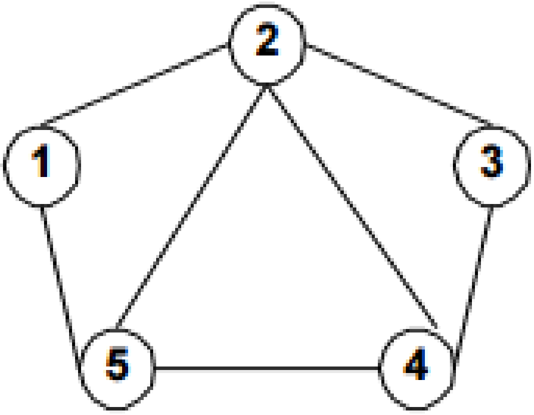
1. Проходя какое-либо ребро, помечаем направление, в котором мы его проходим. По каждому ребру можно идти не более двух раз (по разу в каждом направлении).
2. Если в вершину v_i первый раз мы зашли по ребру q_k, то это ребро помечаем - .
3. Идти можно только по ребру, которое ещё не было пройдено, или было пройдено в противоположном направлении.
4. Из вершины v_i по ребру q_k (помеченному в пункте 2) можно идти в обратном направлении только тогда, когда других возможностей нет.

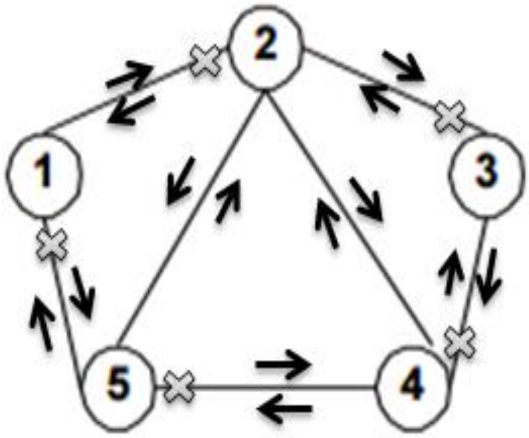
Алгоритм можно использовать для поиска выхода из лабиринта.

Следующая задача известна, как задача о поливальной машине. Необходимо полить все улицы, соответствующие заданному графу, проехав по два раза.

РГР № 8

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





Решение. Ищем путь из вершины v_i в вершину v_i, например, из v_1 в v_1.

Маршрут обхода:

$$1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 \Rightarrow 1 \Rightarrow 5 \Rightarrow 2 \Rightarrow 4 \Rightarrow 2 \Rightarrow 5 \Rightarrow 4 \Rightarrow 3 \Rightarrow 2 \Rightarrow 1$$

Поливальная машина проезжает по каждой улице ровно два раза: по разу в каждом направлении.

У каждой вершины только один крест!

ЗАДАНИЕ № 3

Алгоритм “фронта волны”

Используя алгоритм “фронта волны”, найти все кратчайшие пути из первой вершины в остальные вершины орграфа, заданного матрицей смежности:

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

Теоретические сведения

Определение 3.1. Путь из вершины v_i в v_j орграфа называется кратчайшим, если он содержит наименьшее количество дуг среди всех путей из v_i в v_j.

Алгоритм ’фронта волны’ поиска кратчайшего пути в орграфе.

0) Помечаем вершину v_1 индексом 0; $v_1 \in w_0(v_1)$ – фронт волны нулевого уровня.

1) Помечаем вершины из $\Gamma w_0(v_1) = \Gamma v_1$ единицей. $\Gamma v_1 = w_1(v_1)$

......

k) Помечаем не помеченные ранее вершины из $\Gamma w_{k-1}(v_1)$ индекс k, они принадлежат $w_k(v_1) \subseteq \Gamma\left(w_{k-1}(v_1)\right)$.

Если через k шагов мы дошли до вершины v_t (до конца), то длина кратчайшего пути равна k. Если через *n-1* шаг мы не дошли до v_t, то пути из v_i в v_t не существует.

Предположим, на k шаге мы дошли до вершины v_t. Найдем все вершины кратчайшего пути, начиная с последней v_t.

$$v_1 = v_{i_0}, v_{i_1}, v_{i_2}, \ldots\ldots, v_{i_k} = v_t$$

1. $v_t = v_{i_k}$
2. $v_{i_{k-1}} \in w_{k-1}(v_1) \cap \Gamma^{-1} v_{i_k}$
3. $v_{i_{k-2}} \in w_{k-2}(v_1) \cap \Gamma^{-1} v_{i_{k-1}}$

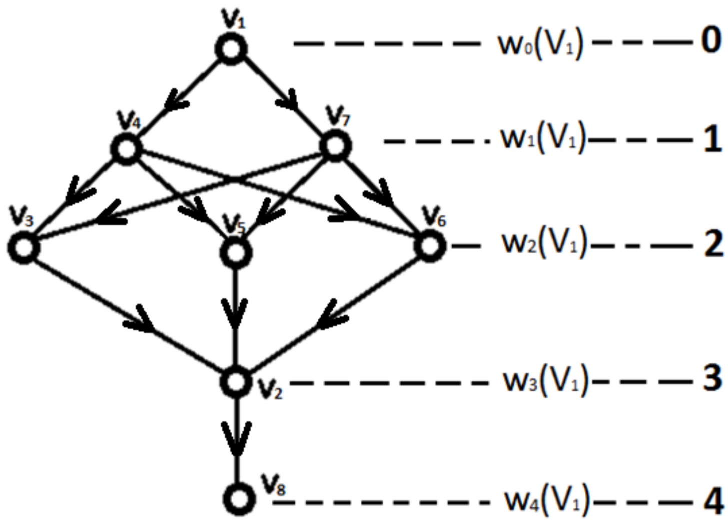
…………………

k. $v_1 = v_{i_0} \in w_0(v_1) \cap \Gamma^{-1} v_{i_1}$

Замечание. В качестве начальной можно выбрать любую вершину орграфа.

Решение.

1. Помечаем вершину v_1 индексом 0. Вершина v_1 принадлежит фронту волны нулевого уровня $W_0(v_1)$.
2. Вершины из множества $\Gamma v_i = \Gamma W_0(v_1) = \{v_4, v_7\}$ (соответствуют единицам в первой строке матрицы A) помечаем индексом 1, они принадлежат фронту волны первого уровня $W_1(v_1)$.
3. Непомеченные ранее вершины из множества $\Gamma W_1(v_1) = \Gamma\{v_4, v_7\} = \{v_3, v_5, v_6\}$ (соответствуют единицам в четвертой и седьмой строках матрицы A) помечаем индексом 2, они принадлежат фронту волны второго уровня $W_2(v_1)$.
4. Непомеченные ранее вершины из множества $\Gamma W_2(v_1) = \Gamma\{v_3, v_5, v_6\} = \{v_2\}$ (соответствуют единицам в третьей, пятой и шестой строках матрицы A) помечаем индексом 3, v_2 принадлежит фронту волны второго уровня $W_3(v_1)$.
5. Непомеченные ранее вершины из множества $\Gamma W_3(v_1) = \Gamma\{v_2\} = \{v_8\}$ (соответствуют единицам во второй строке матрицы A) помечаем индексом 4, v_2 принадлежит фронту волны второго уровня $W_2(v_1)$.
6. Вершина v_8 достигнута, помечена индексом 4, следовательно, длина кратчайшего пути из v_1 в v_8 равна четырем.



Промежуточные вершины кратчайших путей находятся согласно приведенным формулам (начинаем с последней вершины пути):

1) v_8
2) $w_3(v_1) \cap \Gamma^{-1} v_8 = \{v_2\} \cap \{v_2\} = \{v_2\}$.
3) $w_2(v_1) \cap \Gamma^{-1} v_2 = \{v_3, v_5, v_6\} \cap \{v_3, v_5, v_6, v_8\} = \{v_3, v_5, v_6\}$.

4.1) $w_1(v_1) \cap \Gamma^{-1}v_3 = \{v_4, v_7\} \cap \{v_2, v_4, v_5, v_6, v_7, v_8\} = \{v_4, v_7\}$.

4.2) $w_1(v_1) \cap \Gamma^{-1}v_5 = \{v_4, v_7\} \cap \{v_3, v_4, v_6, v_7\} = \{v_4, v_7\}$.

4.3) $w_1(v_1) \cap \Gamma^{-1}v_6 = \{v_4, v_7\} \cap \{v_2, v_3, v_4, v_5, v_7\} = \{v_4, v_7\}$.

5.1.1) $w_0(v_1) \cap \Gamma^{-1}v_4 = \{v_1\} \cap \{v_1, v_2, v_3, v_5\} = \{v_1\}$.

5.1.2) $w_0(v_1) \cap \Gamma^{-1}v_7 = \{v_1\} \cap \{v_1, v_4, v_5, v_7\} = \{v_1\}$.

5.2.1) $w_0(v_1) \cap \Gamma^{-1}v_4 = \{v_1\} \cap \{v_1, v_2, v_3, v_5\} = \{v_1\}$.

5.2.2) $w_0(v_1) \cap \Gamma^{-1}v_7 = \{v_1\} \cap \{v_1, v_4, v_5, v_7\} = \{v_1\}$.

5.3.1) $w_0(v_1) \cap \Gamma^{-1}v_4 = \{v_1\} \cap \{v_1, v_2, v_3, v_5\} = \{v_1\}$.

5.3.2) $w_0(v_1) \cap \Gamma^{-1}v_7 = \{v_1\} \cap \{v_1, v_4, v_5, v_7\} = \{v_1\}$.

Кратчайших путей шесть:

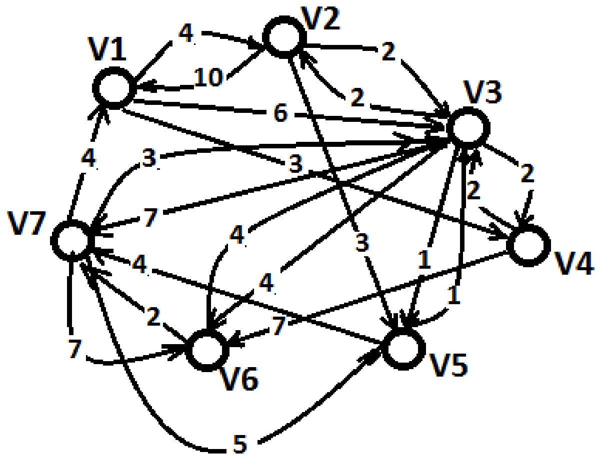
1) $v_1 - v_4 - v_3 - v_2 - v_8$
2) $v_1 - v_4 - v_5 - v_2 - v_8$
3) $v_1 - v_4 - v_6 - v_2 - v_8$
4) $v_1 - v_7 - v_3 - v_2 - v_8$
5) $v_1 - v_7 - v_5 - v_2 - v_8$
6) $v_1 - v_7 - v_6 - v_2 - v_8$

ЗАДАНИЕ № 4

Алгоритм Форда

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

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



Теоретические сведения

Определение 4.1. Нагруженным называется орграф, в котором каждой дуге $< v_i, v_j > \in X$ (каждому ребру для неориентированного графа) ставится в соответствие число $l_{ij} \geq 0$, называемое весом или длинной дуги (ребра).

Определение 4.2. Матрица весов нагруженного орграфа - квадратная матрица порядка n (n - число вершин) с элементами $C = ||c_{ij}||$.

$$c_{ij} = \begin{cases} l_{ij}, \text{если } \exists \text{ дуга } < v_i, v_j > \\ \infty \text{ в противном случае} \end{cases}$$

Определение 4.3. Длиной пути в нагруженном орграфе называется сумма длин его дуг:

$$L = \sum_{\substack{\text{по всем} \\ \text{дугам пути}}} l_{ij}$$

Определение 4.4. Путь из вершины v_i в v_j называется минимальным, если его длина наименьшая по сравнению со всеми путями из v_i в v_j.

Алгоритм нахождения минимального пути в нагруженном графе

$\lambda_i^{(k)}$ – длина минимального пути из вершины v_1 в v_i, содержащего не более k дуг.

$\lambda_i^0, \ldots, \lambda_i^{(n-1)}, \quad i = 1, \ldots, n$

0) Положим $\lambda_i^{(0)} = \infty, \quad i = 2, \ldots, n.$

   $\lambda_1^{(j)} = 0, \quad j = 0, \ldots, n-1$

   ……………….

k) $\lambda_i^{(k)} = \min\limits_{j=1,\dots,n} (\lambda_j^{(k-1)} + c_{ji})$

……………….

$\lambda_i^{(n-1)}$ - длина минимального пути из v_1 в v_i.

Найдем вершины минимального пути $v_1 = v_{i_0}, v_{i_1}, \dots, v_{i_k} = v_t$:

$$\lambda_{i_{k-1}}^{(k-1)} + c_{i_{k-1}i_k} = \lambda_{i_k}^{(k)}$$

$$\lambda_{i_{k-2}}^{(k-2)} + c_{i_{k-2}i_{k-1}} = \lambda_{i_{k-1}}^{(k-1)}$$

$$\dots..$$

$$\lambda_{j_0}^{(0)} + c_{i_0 i_1} = \lambda_{i_1}^{(1)}$$

Индексы по столбцам соответствуют номерам вершин минимального пути.

Может существовать несколько путей, тогда перебором находим все так же, как и в алгоритме «фронта волны».

Решение.

1. Составим таблицу итераций. Итерацию $\lambda_i^{(6)}$ выписывать не нужно, т.к. $\lambda_i^{(4)} = \lambda_i^{(5)}$.

	V1	V2	V3	V4	V5	V6	V7	$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$	$\lambda_i^{(6)}$
V1	∞	4	6	3	∞	∞	∞	0	0	0	0	0	0	0
V2	10	∞	2	∞	3	∞	∞	∞	4	4	4	4	4	4
V3	∞	2	∞	2	1	4	7	∞	6	5	5	5	5	5
V4	∞	∞	2	∞	∞	7	∞	∞	3	3	3	3	3	3
V5	∞	∞	1	∞	∞	∞	4	∞	∞	7	6	6	6	6
V6	∞	∞	4	∞	∞	∞	2	∞	∞	10	9	9	9	9
V7	4	∞	3	∞	5	7	∞	∞	∞	13	11	10	10	10

2. Длины минимальных путей из вершины v_1 во все остальные вершины определены в последнем столбце таблицы.
3. Найдем 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 4.

$\lambda_1^{(0)} + c_{12} = 0 + 4 = \lambda_2^{(1)}$

3.2. Минимальный путь из v_1 в v_3: $v_1 - v_4 - v_3$, его длина равна 5.

$\lambda_4^{(1)} + c_{43} = 3 + 2 = \lambda_3^{(2)}$

$\lambda_1^{(0)} + c_{14} = 0 + 3 = \lambda_4^{(1)}$

3.3. Минимальный путь из v_1 в v_4: $v_1 - v_4$, его длина равна 3.

$\lambda_1^{(0)} + c_{14} = 0 + 3 = \lambda_4^{(1)}$

3.4. Минимальный путь из v_1 в v_5: $v_1 - v_4 - v_3 - v_5$:, его длина равна 6.

$\lambda_3^{(2)} + c_{35} = 5 + 1 = \lambda_5^{(3)}$

$\lambda_4^{(1)} + c_{43} = 3 + 2 = \lambda_3^{(2)}$

$\lambda_1^{(0)} + c_{14} = 0 + 3 = \lambda_4^{(1)}$

3.5. Минимальный путь из v_1 в v_6: $v_1 - v_4 - v_3 - v_6$:, его длина равна 9.

$\lambda_3^{(2)} + c_{36} = 5 + 4 = \lambda_6^{(3)}$

$\lambda_4^{(1)} + c_{43} = 3 + 2 = \lambda_3^{(2)}$

$\lambda_1^{(0)} + c_{14} = 0 + 3 = \lambda_4^{(1)}$

3.6. Минимальный путь из v_1 в v_7: $v_1 - v_4 - v_3 - v_5 - v_7$:, его длина равна 10.

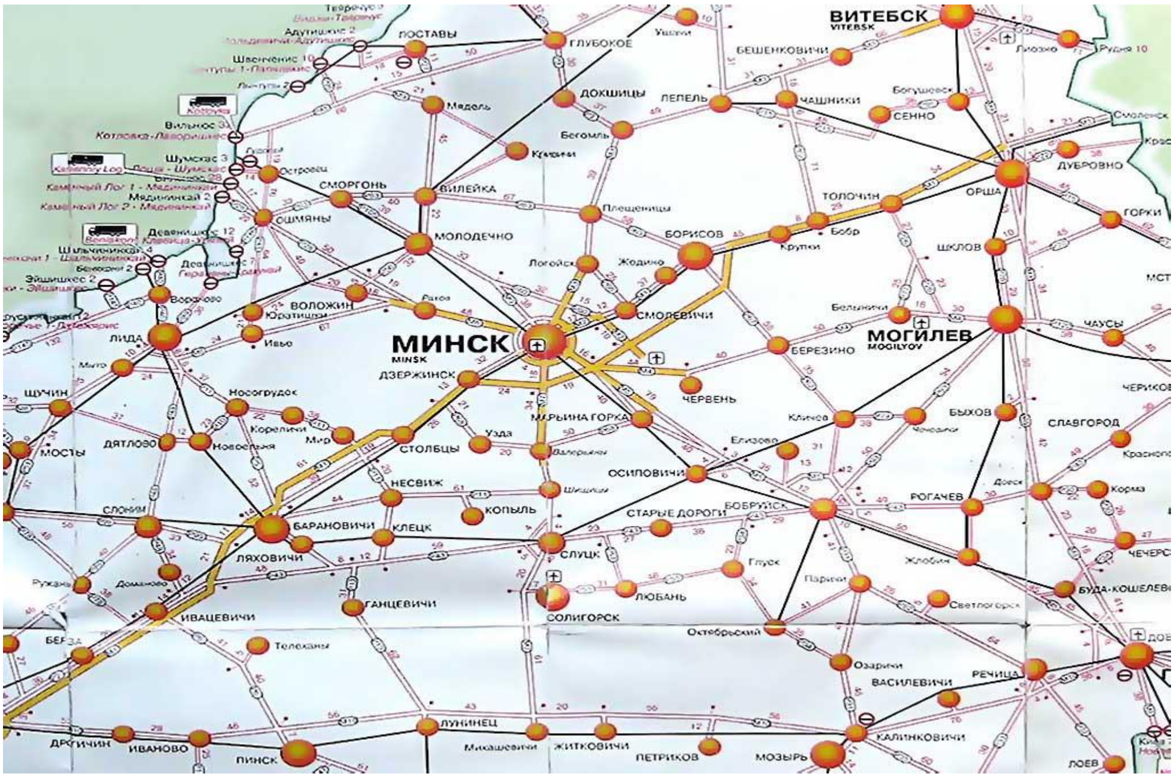
$\lambda_5^{(3)} + c_{57} = 6 + 4 = \lambda_7^{(4)}$

$\lambda_3^{(2)} + c_{35} = 5 + 1 = \lambda_5^{(3)}$

$\lambda_4^{(1)} + c_{43} = 3 + 2 = \lambda_3^{(2)}$

$\lambda_1^{(0)} + c_{14} = 0 + 3 = \lambda_4^{(1)}$

Хорошо выбирать путь минимальной длины там, где много хороших дорог.

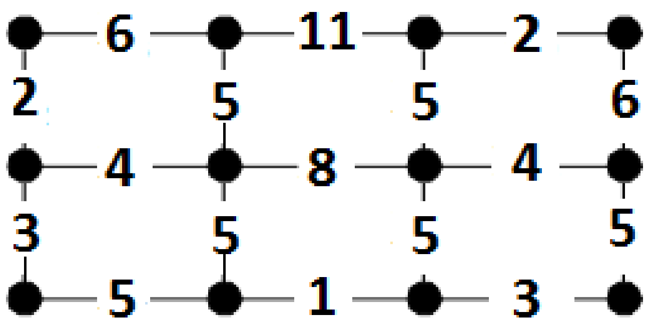


Карта дорог Белоруссии с километражем.

ЗАДАНИЕ № 5

Остовное дерево минимальной длины

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



Теоретические сведения

Определение 5.1. Дерево - связный граф без циклов (основное определение дерева).

Определение 5.2. Остовным деревом графа $G =< V, Q >$ называется подграф $D =< V, \overline{Q} >$, содержащий все вершины графа G и являющийся деревом.

Определение 5.3. Остовным деревом минимального веса называется остовное дерево с наименьшей суммой весов его ребер.

$$L(D) = \sum_{\text{по всем элементам}} l_{ij} \rightarrow min$$

Алгоритм Краскала построения дерева минимального веса.

1. Выбираем все вершины графа G.
2. Добавляем все дуги, имеющие минимальный вес, так, чтобы не было циклов.
3. Добавляем дуги минимального веса из оставшихся весов, так, чтобы не было циклов до получения *(n – 1)* дуги.

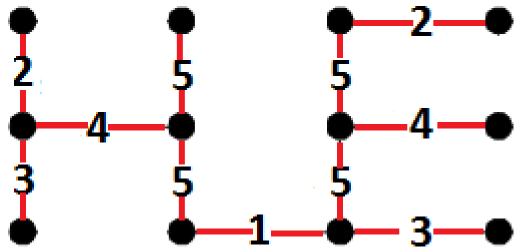
У графа может быть несколько остовных деревьев минимальной длины.

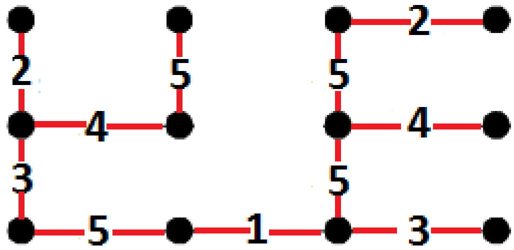
Решение

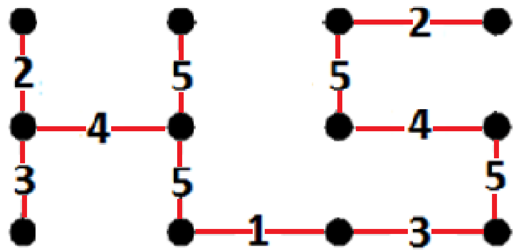
1. Выбираем все вершины графа.
2. Добавляем все дуги, имеющие минимальный вес – 1. Циклов нет.
3. Добавляем все дуги, имеющие минимальный вес среди оставшихся – 2 . Циклов нет.
4. Добавляем все дуги, имеющие минимальный вес среди оставшихся – 3. Циклов нет.
5. Добавляем все дуги, имеющие минимальный вес среди оставшихся – 4. Циклов нет.

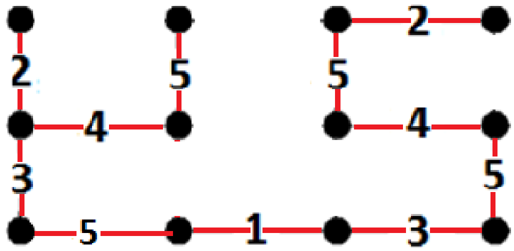
6. Добавляем все дуги, имеющие минимальный вес – 5, так, чтобы не было циклов. Получаем четыре возможных вариантов остовных деревьев минимального веса. Минимальный вес остовного дерева $L(D) = 39$.

Возможные остовные деревья с минимальной суммой длин ребер - 39:

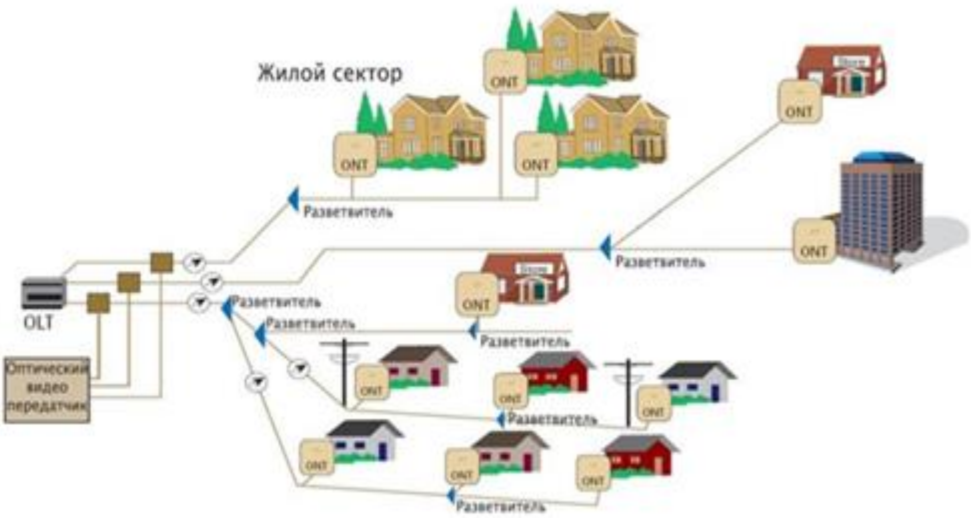








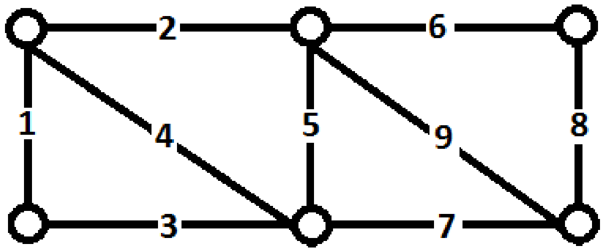
Алгоритм применяется, в частности, для подведения газа, воды и электричества к жилым домам. При этом используется минимальная общая длина труб или провода.



ЗАДАНИЕ № 6

Деревья и циклы. Законы Кирхгофа.

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 E_1 и E_2 (полярность выбирается произвольно), а остальные элементы являются сопротивлениями. Используя закон Ома, и, предполагая внутренние сопротивления источников тока равными нулю, получить систему уравнений для токов.



Теоретические сведения

Цикломатическое число графа. Базис циклов.

Определение 6.1. Цикломатическое число графа $G =< V, Q >$ равно $\gamma(G) = m - n + p$, где p - число компонент связности, $m -$ число ребер, $n -$число вершин.

Для связного графа p = 1 и цикломатическое число равно $m - n + 1$

Цикломатическое число для дерева равно нулю (циклов нет).

Рассмотрим неориентированный граф и зададим на нем произвольную ориентацию. Каждому циклу μ поставим в соответствие вектор – цикл $c(\mu)$.

Определение 6.2. Вектор – циклом $c(\mu)$ назовем вектор с m компонентами $c(\mu) = (c_1(\mu), \dots, c_m(\mu))$, где $c_i(\mu)$ равно разности числа проходов ребра q_i по и против направления ориентации.

Определение 6.3. Циклы $\mu_1 \dots \mu_k$ называются независимыми, если соответствующие вектор – циклы $c(\mu_1) \dots c(\mu_n)$ ЛНЗ.

Определение 6.4. . Цикл μ называют линейной комбинацией циклов $\mu_1 \dots \mu_k$, если вектор-цикл $c(\mu)$ - линейная комбинация $c(\mu_1) \dots c(\mu_k)$.

Определение 6.5. Циклы $\mu_1 \dots \mu_n$ образуют базис циклов, если:

1) Они независимые.
2) Любой цикл можно представить как линейную комбинацию $\mu_1 \dots \mu_n$.

Утверждение 6.1. Число циклов в базисе равно цикломатическому числу графа.

Алгоритм нахождения базиса циклов связного графа $G =< V, Q >$.

1) Строим остовное дерево графа с $n-1$ ребром (вершин - n)

$$q_{i_1}, \ldots, q_{i_{n-1}}$$

2) Добавляем по одному из оставшихся ребер $q_{i_n}, \ldots, q_{i_m}$ к остовному дереву и получаем ровно один простой цикл, который и берем в базис циклов.

$$D + q_{i_n} = \mu_1, \ldots \text{и т.д.}$$

При нахождении остовного дерева графа используется следующий алгоритм.

Алгоритм нахождения остовного дерева связного графа

1) $D_1 = (v_{i_1}, \emptyset)$

2) $D_2 = (\{v_{i_1}, v_{i_2}\}, \{v_{i_1}, v_{i_2}\})$ – добавляем к D_1 вершину v_{i_2}, смежную с v_{i_1}, и соединяем их
ребром.

……………….

k) $D_k = D_{k-1} + \{v_{i_k}\} + \{v_{i_j}, v_{i_k}\}, j = 1, \ldots, k-1$ – добавляем новую вершину v_{i_k}, смежную хотя бы с одной из ранее выбранных и соответствующую дугу.

Продолжаем процесс, пока ни выберем все n вершин графа.

Определение 6.6. Цикломатической матрицей C неориентированного графа называется матрица размерности $\gamma \times m$, по строкам которой стоят вектор – циклы базиса циклов.

Законы Кирхгофа для токов и напряжений

Закон Кирхгофа для напряжений.

Пусть $\mathrm{U} = \begin{pmatrix} U_1 \\ \vdots \\ U_m \end{pmatrix}$ - вектор напряжений. Тогда для замкнутого контура выполняется $(c(\mu_i), U) = 0$. Откуда получаем:

$$\boldsymbol{C} * \boldsymbol{U} = \boldsymbol{0}$$

При решении данной системы напряжения, соответствующие добавленным ребрам при построении базиса циклов, являются базисные переменные.

Закон Кирхгофа для токов

Пусть $I = \begin{pmatrix} I_1 \\ \vdots \\ I_m \end{pmatrix}$ - вектор токов.

$$B * I = 0$$

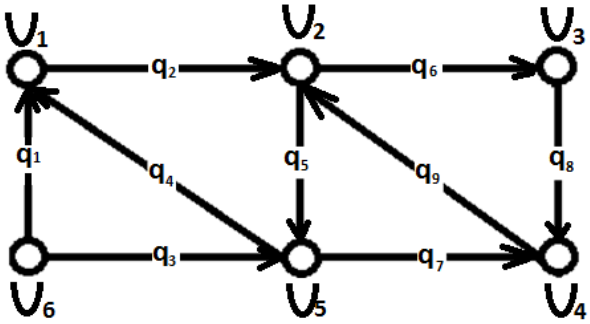
B- матрица инцидентности графа. По свойству матрицы инцидентности: $rgB = n - 1$.

Закон Ома

$$U = I * R$$

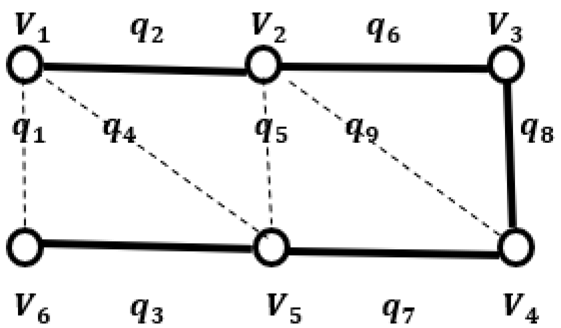
Решение.

1. Зададим на графе произвольную ориентацию:



2. Построим произвольное остовное дерево D заданного графа:

   Ребра q_1 и q_5, соответствующие ЭДС, в остовное дерево не взяли.



3. Найдем базис циклов, добавляя к остовному дереву по одному не вошедшему в него ребру (изображены пунктиром). Затем найдем соответствующие вектор-циклы.

   (D+q_1): μ_1: $v_1 - v_2 - v_3 - v_4 - v_5 - v_6 - v_1$ ⟹C(μ_1) = (1, 1, -1, 0, 0, 1, -1, 1, 0);

   (D+q_4): μ_2: $v_1 - v_2 - v_3 - v_4 - v_5 - v_1$ ⟹ C(μ_2) = (0, 1, 0, 1, 0, 1, -1, 1, 0);

   (D+q_5): μ_3: $v_2 - v_3 - v_4 - v_5 - v_2$ ⟹ C(μ_3) = (0, 0, 0, 0, -1, 1, -1, 1, 0);

   (D+q_9): μ_4: $v_2 - v_3 - v_4 - v_2$ ⟹ C(μ_4) = (0, 0, 0, 0, 0, 1, 0, 1, 1);

4. Цикломатическая матрица графа имеет вид:

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

5. Выпишем закон Кирхгофа для напряжений:

$$\begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 1 & -1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} * \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = 0$$

Напряжения, соответствующие ребрам, не вошедшим в островное дерево u_1, u_4, u_5, u_9– базисные переменные системы. Остальные – свободные. Выразим переменные базисные через свободные.

$$\begin{cases} u_1 + u_2 - u_3 + u_6 - u_7 + u_8 = 0 \\ u_2 + u_4 + u_6 - u_7 + u_8 = 0 \\ -u_5 + u_6 - u_7 + u_8 = 0 \\ u_6 + u_8 + u_9 = 0 \end{cases} \qquad \begin{cases} u_1 = -u_2 + u_3 - u_6 + u_7 - u_8 \\ u_4 = -u_2 - u_6 + u_7 - u_8 \\ u_5 = u_6 - u_7 + u_8 \\ u_9 = -u_6 - u_8 \end{cases}$$

6. Выпишем уравнения Кирхгофа для токов.

Найдем матрицу инцидентности B орграфа:

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9
v_1	1	-1	0	1	0	0	0	0	0
v_2	0	1	0	0	-1	-1	0	0	1
v_3	0	0	0	0	0	1	0	-1	0
v_4	0	0	0	0	0	0	1	1	-1
v_5	0	0	1	-1	1	0	-1	0	0
v_6	-1	0	-1	0	0	0	0	0	0

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

$$\begin{pmatrix} 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 & 0 & -1 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \end{pmatrix} = 0$$

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

7. Подставим формулы закона Ома:

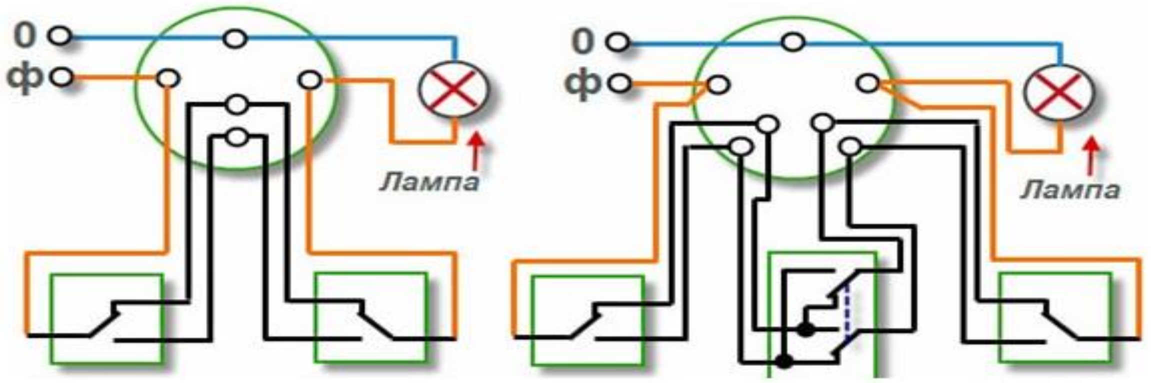
$$\begin{cases} -I_2R_2 + I_3R_3 - I_6R_6 + I_7R_7 - I_8R_8 = E_1 \\ I_2R_2 + I_4R_4 + I_6R_6 - I_7R_7 + I_8R_8 = 0 \\ I_6R_6 - I_7R_7 + I_8R_8 = E_2 \\ I_6R_6 + I_8R_8 + I_9R_9 = 0 \end{cases}$$

8. Совместная система имеет вид:

$$\begin{cases} I_1 - I_2 + I_4 = 0 \\ I_2 - I_5 - I_6 + I_9 = 0 \\ I_6 - I_8 = 0 \\ I_7 + I_8 - I_9 = 0 \\ -I_1 - I_3 = 0 \\ -I_2R_2 + I_3R_3 - I_6R_6 + I_7R_7 - I_8R_8 = E_1 \\ I_2R_2 + I_4R_4 + I_6R_6 - I_7R_7 + I_8R_8 = 0 \\ I_6R_6 - I_7R_7 + I_8R_8 = E_2 \\ I_6R_6 + I_8R_8 + I_9R_9 = 0 \end{cases}$$

Девять уравнений и девять неизвестных - токи $I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9$; ЭДС - E_1, E_2 и сопротивления $R_2, R_3, R_4, R_6, R_7, R_8, R_9$ известны.

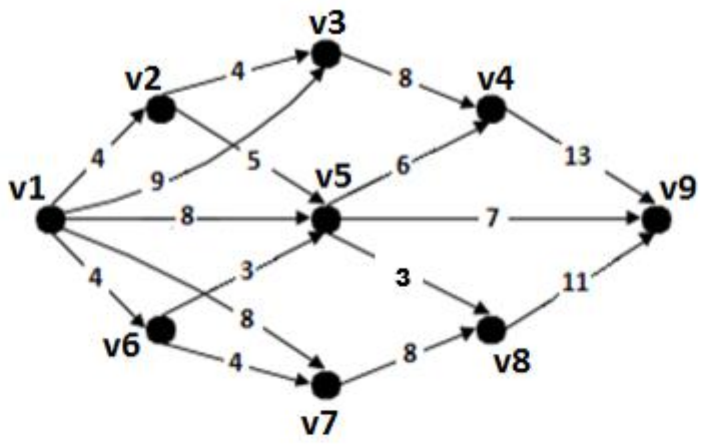
Замечание. Ребра графа, соответствующие ЭДС E_1 и E_2 для простоты и красоты уравнений желательно не брать в остовное дерево.



ЗАДАНИЕ № 7

Транспортные сети.

Построить максимальный поток по данной транспортной сети



Теоретические сведения

Транспортные сети

Определение 7.1. Транспортной сетью называется орграф $G =< V, X >$, $V = \{v_1, \dots, v_n\}$, для которого выполняется:

1) $\exists$ единственная вершина v_1 (источник): $\Gamma^{-1}v_1 = \emptyset$;

2) $\exists$ единственная вершина v_n (сток): $\Gamma v_n = \emptyset$;

3) для каждой дуги $< v_i, v_j > \epsilon X$ задана пропускная способность $c(v_i, v_j) = c_{ij} \geq 0$.

Определение 7.2. Функцией потока (потоком в транспортной сети G) называется функция

$\varphi: X \to R^+ \cup \{0\}$, удовлетворяющая следующим условиям:

1) для каждой дуги $< v_i, v_j > \epsilon X$ выполняется $0 \leq \varphi(v_i, v_j) \leq c_{ij}$ $\forall i, j = 1, .., n$;

2) для любой промежуточной вершины u:

$$\sum_{v \epsilon \Gamma^{-1}\mathrm{u}} \varphi(v, \mathrm{u}) = \sum_{v \epsilon \Gamma \mathrm{u}} \varphi(\mathrm{u}, v)$$

Определение 7.3. Величина потока равна сумме потоков по всем дугам, входящим в сток.

$$\Phi = \sum_{v \epsilon \Gamma^{-1} v_n} \varphi(v, v_n)$$

Эта величина также равна сумме потоков по всем дугам, исходящим из источника

$$\Phi = \sum_{v\epsilon\Gamma v_1} \varphi(v_1, v)$$

Определение 7.4. Дуга называется насыщенная, если значение функции потока по ней равно пропускной способности.

Определение 7.5. Поток называется полным, если любой путь из источника в сток содержит хотя бы одну насыщенную дугу.

Обычно полный поток ищут как приближение к максимальному, в частном случае они могут совпасть, но в общем случае полный используется как начальный для построения максимального.

Определение 7.6. Поток называется максимальным, если значение величины потока наибольшее по сравнению со всеми потоками в данной транспортной сети.

Алгоритм построения полного потока

0) Выбираем нулевой поток в качестве начального $\varphi_{ij} = 0 \quad \forall i, j.$
1) Проверяем, является ли построенный поток полным, т.е. существует ли путь из $v_1 \to v_n$, не содержащий насыщенных дуг. Если такого пути нет ⇒ полный поток построен, если есть, то п.2.
2) Вдоль пути, не содержащего насыщенных дуг, увеличиваем поток на одну и ту же величину до тех пор, пока хотя бы одна дуга не станет насыщенной. Переходим к п.1.

Для построения максимального потока введем понятие увеличивающей цепи.

Определение 7.7. Увеличивающей цепью называется последовательность вершин транспортной сети из $v_1 \to v_n$:

$$v_1 = \mathrm{u}_1, \mathrm{u}_2, \dots, \mathrm{u}_k = v_n \ , \qquad (*)$$

где $< \mathrm{u}_j, \mathrm{u}_{(j+1)} > \epsilon X$, либо $< \mathrm{u}_{(j+1)}, \mathrm{u}_j > \epsilon X$, причем для каждой пары вершин цепи $< \mathrm{u}_j, \mathrm{u}_{(j+1)} >$ определена положительная величина

$$\triangle_{j(j+1)} = \begin{cases} c_{j(j+1)} - \varphi_{j(j+1)}, & \text{если } < \mathrm{u}_j, \mathrm{u}_{(j+1)} > \epsilon X \\ \varphi_{(j+1)j}. & \text{Если } < \mathrm{u}_{(j+1)}, \mathrm{u}_j > \epsilon X \end{cases}$$

Для увеличивающей цепи найдем величину

$$\triangle = \min_{j=1,\dots,k-1} \triangle_{j(j+1)}$$

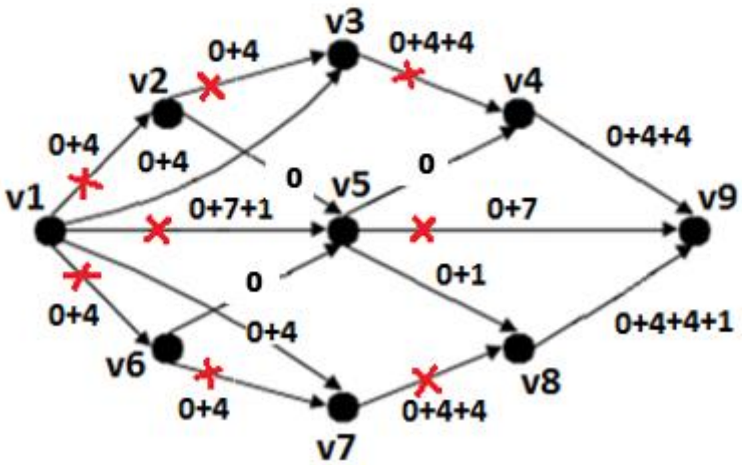
Алгоритм поиска максимального потока в транспортной сети

0) Выбираем полный поток в качестве начального (можно любой).
1) Проверяем, является ли построенный поток максимальным, т.е. существует ли увеличивающая цепь из $v_1 \to v_n$. Если такой цепи нет $\Rightarrow$ максимальный поток построен, если есть, то п.2.
2) Вдоль увеличивающей цепи изменяем поток на величину $\triangle$, причем если идем по дуге, то увеличиваем на $\triangle$, если против направления дуги уменьшаем на $\triangle$. Переходим к п.1.

Решение

1) Построение полного потока.

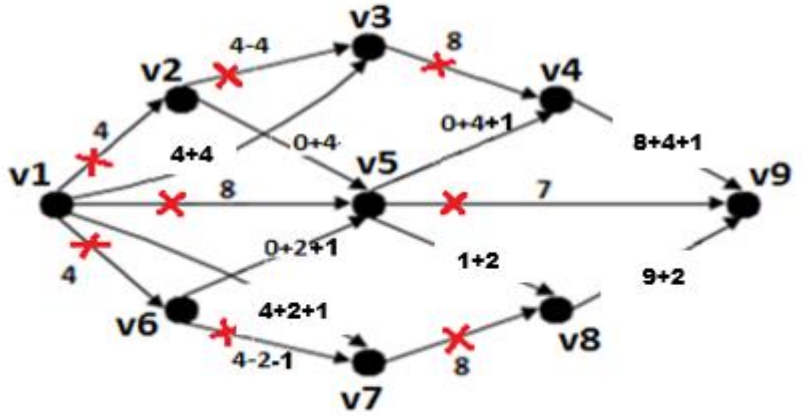
Ищем пути из источника в сток, не содержащие насыщенных дуг.



1. $v_1 - v_2 - v_3 - v_4 - v_9$
   $\min\{4, 4, 8, 13\} = 4$
2. $v_1 - v_6 - v_7 - v_8 - v_9$
   $\min\{4, 4, 8, 11\} = 4$
3. $v_1 - v_5 - v_9$
   $\min\{\ 8,\ 7\} = 7$
4. $v_1 - v_3 - v_4 - v_9$
   $\min\{9, 8-4, 13-4\} = 4$
5. $v_1 - v_7 - v_8 - v_9$
   $\min\{8, 8-4,\ 11-4\} = 4$
6. $v_1 - v_5 - v_8 - v_9$
   $\min\{\ 8-7,\ 3, 11-8\} = 1.$

Величина полного потока $\mathbf{\Phi}_{\text{пол.}} = \mathbf{8 + 7 + 9 = 24.}$

2) Построение максимального потока.



Найдем увеличивающие цепи.

1. $$v_1 - v_3 - v_2 - v_5 - v_4 - v_9$$

$$\Delta_1 = \min\{9-4,\ \underline{4},\ 5,\ 6,\ 13-8\} = 4$$

2. $$v_1 - v_7 - v_6 - v_5 - v_8 - v_9$$

$$\Delta_2 = \min\{8-4,\ \underline{4},\ 3,\ 3-1,\ 11-9\} = 2$$

3. $$v_1 - v_7 - v_6 - v_5 - v_4 - v_9$$

$$\Delta_3 = \min\{8-6,\ \underline{2}, 3-2, 6-4, 13-12\} = 1$$

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

Величина потока увеличилась на 6 (4+2): величина максимального потока

$$\mathbf{\Phi_{\text{макс.}} = 13 + 7 + 11 = 31.}$$

Только в 1955 году, Лестер Форд и Дельберт Фалькерсон впервые разработали алгоритм построения максимального потока в транспортной сети. В дальнейшем решение этой задачи много раз улучшалось. Приведенный здесь алгоритм описан в работе: Липский В. Комбинаторика для программистов.

А в древние века классическая задача о портовых перевозках решалась не оптимально.

ЗАДАНИЕ № 8 (Индивидуальное)

Применение теории графов для ранжирования альтернатив при принятии решений.

Пример выполнения и оформления индивидуального задания

Теоретические сведения. Описание алгоритма.

Заданы множество альтернатив $A = \{a_1,\ a_2, \dots, a_n\}$ и множество экспертов $E = \{E_1, E_2, \dots, E_m\}$. Профиль индивидуальных предпочтений экспертов на множестве A задан бинарными отношениями предпочтения - ранжированиями $\rho_1,\ \rho_2, \dots, \rho_m$. Требуется построить агрегированное ранжирование $\hat{\rho}$.

Для заданных экспертных ранжирований строятся матрицы предпочтений.

Бинарному отношению ρ поставим в соответствие квадратную матрицу $R = \|r_{ij}\|$ порядка n (n – число альтернатив) с элементами

$$r_{ij} = \begin{cases} 1, \text{если } a_i \text{ менее } a_j \text{предпочтительна;} \\ \frac{1}{2}, \text{если } a_i \text{ и } a_j \text{ равноценны;} \\ 0, \text{если } a_j \text{ менее предпочтительна } a_i \text{ или } a_i, a_j \text{ не сравнимы} \end{cases} .$$

при $i \neq j$. Элемент $r_{ij} = 1$ (i=1, …, n).

Алгоритмы агрегирования экспертных предпочтений используют процедуры на графах, в частности процедуру построения нагруженного мажоритарного графа. Кроме того, для выбора наилучших вариантов альтернатив или упорядочения их по предпочтительности применяются такие классические алгоритмы на графах, как нахождение внешне и внутренне устойчивых подмножеств графа, ядра графа, разбиение графа на уровни. В связи с этим введем в рассмотрение графы отношений. Поставим бинарному отношению ρ в соответствие граф G, причем $G = (A,\ \rho)$ – ориентированный граф с множеством вершин-альтернатив $A = \{a_1,\ a_2, \dots, a_n\}$ и множеством дуг ρ. Множество дуг – это множество упорядоченных пар $< a_i, a_j >$ $(a_i, a_j \in A)$, входящих в отношение ρ. Под матрицей смежности произвольного бинарного отношения ρ будем понимать матрицу смежности соответствующего графа.

Определение 1. Строгим нагруженным мажоритарным графом назовем ориентированный нагруженный граф $G =< A,\ \rho_\Sigma >$ с множеством вершин-альтернатив $A = \{a_1,\ a_2, \ldots, a_n\}$ и дугами $\rho_\Sigma = \{< a_i, a_j >\big|\ a_i, a_j \in A$ и $l_{ij} > 0\}$, где $l_{ij} = \sum_{k=1}^{m}(r_{ij}^k - r_{ji}^k)$. Причем каждой дуге $< a_i, a_j >\in \rho_\Sigma$ поставим в соответствие вес l_{ij}.

Под знаком суммы стоят элементы матриц предпочтений отношений ρ_t, t=1, …, m, ($R^t = \|r_{ij}^t\|$). Матрицу смежности графа $G =< A,\ \rho_\Sigma >$ обозначим R_Σ.

Введем матрицу весов $C = \|c_{ij}\|$ – квадратная матрица порядка n, где n – число альтернатив, причем

$$c_{ij} = \begin{cases} l_{ij}, \text{если } \exists \text{ дуга } < a_i, a_j > \in \rho_\Sigma\ , \\ \infty, \text{иначе.} \end{cases}$$

Введем матрицу суммарных предпочтений $P = \|p_{ij}\|$ – квадратную матрицу порядка n (число альтернатив), где $p_{ij} = \sum_{k=1}^{m} r_{ij}^k$. С ее помощью удобно находить матрицу весов C:

$$c_{ij} = \begin{cases} p_{ij} - p_{ji}, \text{если } \exists \text{ дуга } < a_i, a_j > \in \rho_\Sigma\ , \\ \infty, \text{иначе.} \end{cases}$$

Граф проверяется на наличие контуров (см. задание 1). Если граф не содержит контуров, ранжируем альтернативы с помощью алгоритма Демукрона – разбиения графа на уровни. Если есть контуры, ранжирование проводится на основе индексов Демукрона.

Определение. Уровнями $N_0,\ N_1, \ldots, N_k$ графа $G = (A,\ \rho)$ без контуров называются следующие непустые множества вершин графа

$N_0 = \{a_i\ |a_i \in A, \Gamma a_i = \emptyset$;

$N_1 = \{a_i \mid a_i \in A\backslash N_0, \Gamma a_i \subseteq N_0\}$;

……………………..…………………………………………....

$$N_k = \{a_i \mid a_i \in A\backslash \bigcup_{i=0}^{k-1} N_j\, , \Gamma a_i \subseteq \bigcup_{i=0}^{k-1} N_j\}.$$

$\Gamma^{-1}N_k = \emptyset$.

Уровни $N_0, N_1, \ldots, N_k$ образуют разбиение вершин графа: $V = \bigcup_{j=0}^{k} N_j$

$$N_i \cap N_j = \emptyset,\ i \neq j;\ i, j = 0, \ldots, k$$

Теорема. Если граф не содержит контуров, то его можно разбить на уровни, и обратно, если граф можно разбить на уровни, то он не содержит контуров.

Алгоритм Демукрона разбиения графа без контуров на уровни

0) Ищем вектор $L^{(0)} = \begin{pmatrix} l_1^{(0)} \\ l_2^{(0)} \\ \dots \\ l_n^{(0)} \end{pmatrix}$, где $l_i^{(0)}$ - сумма единиц в i-той строке матрицы смежности. Если l_i^0=0, то вершина $a_i \in N_0$. В этом случае обнуляем i-й столбец матрицы смежности A.

1) Ищем вектор $L^{(1)} = \begin{pmatrix} l_1^{(1)} \\ l_2^{(1)} \\ \dots \\ l_n^{(1)} \end{pmatrix}$, где $l_i^{(1)}$ равно сумме единиц в i-той строке матрицы A или $*$, если $l_i^{(0)} = 0$. Если $l_i^{(1)} = 0$, то вершина $a_i \in N_1$. В этом случае обнуляем i-й столбец матрицы А.

……………………………………………….

k) Ищем вектор $L^{(k)} = \begin{pmatrix} l_1^{(k)} \\ l_2^{(k)} \\ \dots \\ l_n^{(k)} \end{pmatrix}$, где $l_i^{(k)}$ равно сумме единиц в i-той строке матрицы A или $*$, если $l_i^{(k-1)} = 0$ или $l_i^{(k-1)} =*$.

Если $l_i^{(k)} = 0$, то вершина $a_i \in N_{\mathrm{k}}$. В этом случае обнуляем i-й столбец матрицы А.

Алгоритм заканчивает работу, когда в матрице A все элементы равны нулю.

Индекс Коупленда для вершины-альтернативы a_i равен сумме весов по всем дугам, заходящим в вершину мажоритарного графа a_i, минус сумма весов дуг по всем дугам, исходящим из a_i.

Логическая блок-схема алгоритма

Основные этапы работы алгоритма представлены на рис. 8.1 логической блок схемы.

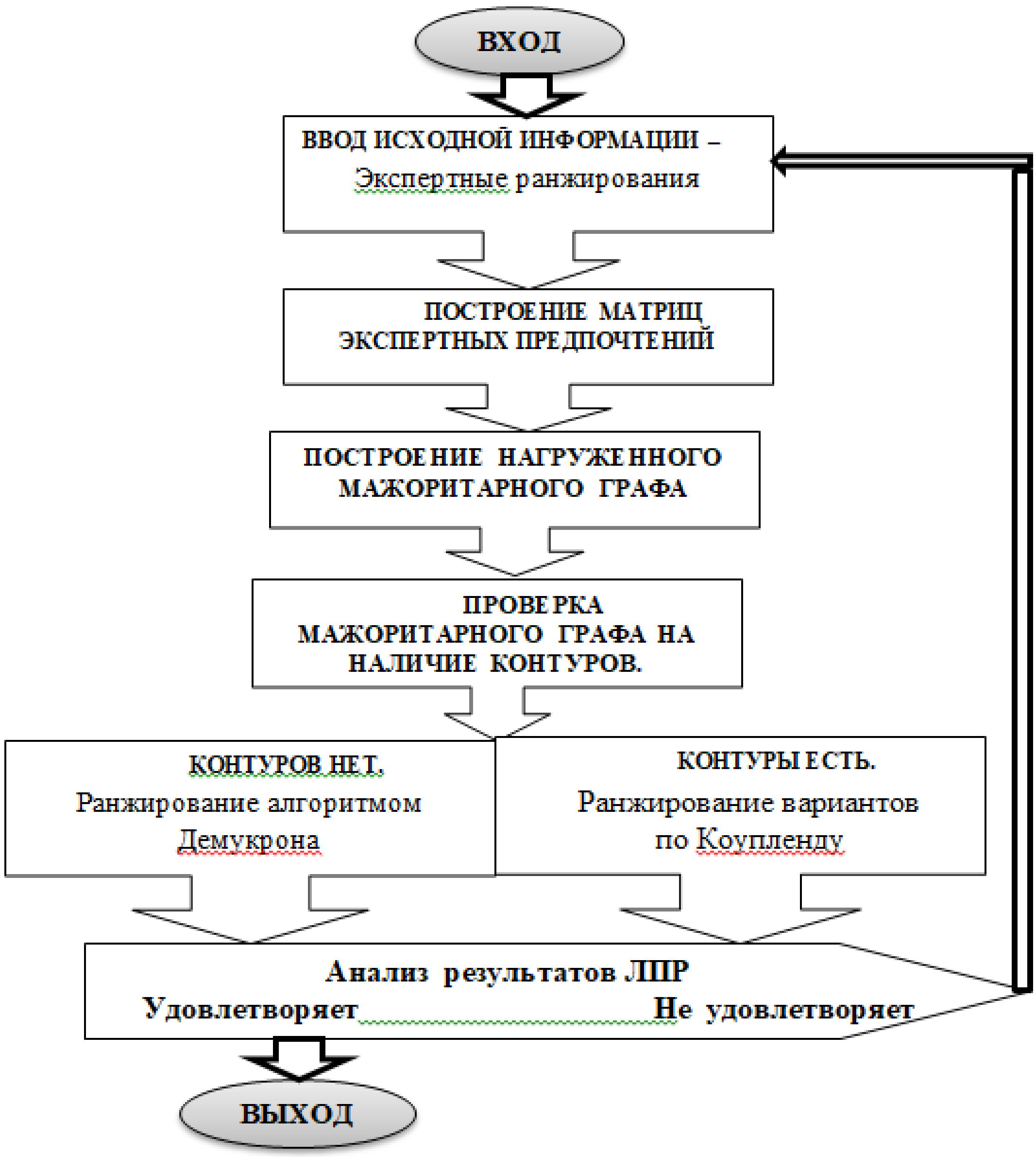


Рис. 8.1.

Программа написана на языке JAVA, в результате чего она имеет ряд интересных технических особенностей. Программы транслируются в байт-код и в дальнейшем выполняются виртуальной Java-машиной. Это позволяет выполнять приложения на любом устройстве, поддерживающем виртуальную машину (планшет или смартфон).

Оценка сложности алгоритма

Наибольшую сложность имеет подпрограмма нахождения контуров графа, а именно алгоритм Уоршалла нахождение матрицы односторонней связности графа - $O(n^3)$, что соответствует максимальному числу вложенных циклов в программе – три.

Тестовые примеры. Скриншоты программы.

Пример 1. Четыре эксперта ранжировали пять альтернатив (наилучшие в верхней строке таблицы):

ρ_1	ρ_2	ρ_3	ρ_4
a_1	a_3	a_5	a_1
a_2	a_4	a_3	a_3
a_3	a_5	a_1	a_4
a_4	a_1	a_2	a_5
a_5	a_2	a_4	a_2

Матрицы предпочтений этих отношений имеют вид:

$$R^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}, R^2 = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix},$$

$$R^3 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, R^4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Матрица суммарных предпочтений P=$R^1 + R^2 + R^3 + R^4$ и равна

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

Тогда матрица весов и матрица смежности мажоритарного графа соответственно равны:

$$C = \begin{pmatrix} \infty & \infty & \infty & \infty & \infty \\ 4 & \infty & 4 & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & 4 & \infty & \infty \\ \infty & \infty & 2 & 2 & \infty \end{pmatrix}, \qquad R_\Sigma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Проведем проверку мажоритарного графа на наличие контуров (см. задание 1 курсовой работы). Получим матрицу контуров

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

Следовательно, мажоритарный граф не содержит контуров и его можно разбить на уровни, используя алгоритм Демукрона.

$$R_\Sigma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \Rightarrow L^{(0)} = \begin{pmatrix} 0 \\ 3 \\ 0 \\ 2 \\ 2 \end{pmatrix} \Rightarrow a_1, a_3 \in N_0.$$

Первый и третий столбцы матрицы обнуляются.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \Rightarrow L^{(1)} = \begin{pmatrix} * \\ 1 \\ * \\ 0 \\ 1 \end{pmatrix} \Rightarrow a_4 \in N_1.$$

Четвертый столбец матрицы обнуляется.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow L^{(2)} = \begin{pmatrix} * \\ 1 \\ * \\ * \\ 0 \end{pmatrix} \Rightarrow a_5 \in N_2.$$

Пятый столбец матрицы обнуляется.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow L^{(3)} = \begin{pmatrix} * \\ 0 \\ * \\ * \\ * \end{pmatrix} \Rightarrow a_2 \in N_3.$$

Получаем ранжирование альтернатив: $\{a_1, a_3\} - a_4 - a_5 - a_2$. Наилучшие альтернативы a_1 и a_3.

Разбиение графа на уровни представлена на рис. 8.2.

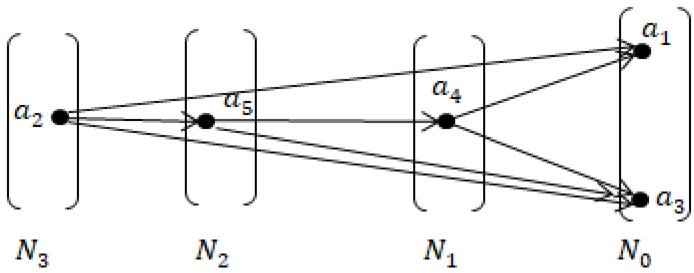


Рис. 8.2.

Пример 2. Дополнительно был приглашен пятый эксперт. Таблица ранжирований приняла вид:

ρ_1	ρ_2	ρ_3	ρ_4	ρ_5
a_1	a_3	a_5	a_1	a_4
a_2	a_4	a_3	a_3	a_2
a_3	a_5	a_1	a_4	a_3
a_4	a_1	a_2	a_5	a_5
a_5	a_2	a_4	a_2	a_1

Матрицы предпочтений этих отношений имеют вид:

$$R^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}, R^2 = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix},$$

$$R^3 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, R^4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}, R^5 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Матрица суммарных предпочтений P=$R^1 + R^2 + R^3 + R^4 + R^5$ и равна

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

Тогда матрица весов и матрица смежности мажоритарного графа соответственно равны:

$$C = \begin{pmatrix} \infty & \infty & 1 & \infty & 1 \\ 3 & \infty & 1 & 1 & 1 \\ \infty & \infty & \infty & \infty & \infty \\ 1 & \infty & 3 & \infty & \infty \\ \infty & \infty & 3 & 3 & \infty \end{pmatrix}, \qquad R_{\Sigma} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Проведем проверку мажоритарного графа на наличие контуров (см. задание 1 курсовой работы). Получим матрицу контуров

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

Следовательно, мажоритарный граф содержит контуры (рис. 8.3).

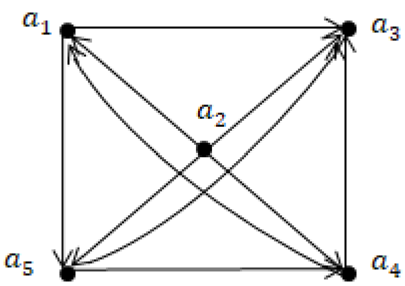


Рис. 8.3.

Вычисляем индексы Коупленда для всех вершин графа:

$a_1(2),\ a_2(-6),\ a_3(5),\ a_4(0), a_5(-1)$;

что соответствует ранжированию $a_3 - a_1 - a_4 - a_5 - a_2$.

Скриншоты программы.

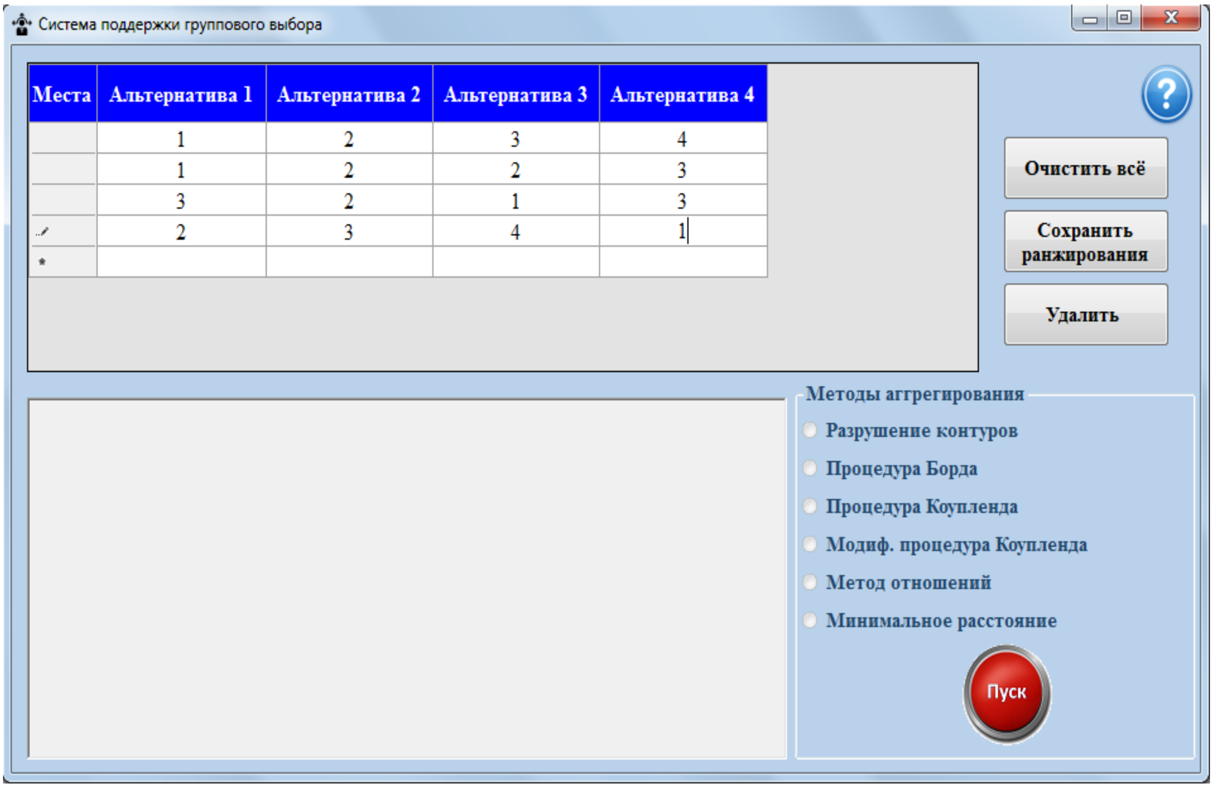


Рис. Ввод экспертных предпочтений

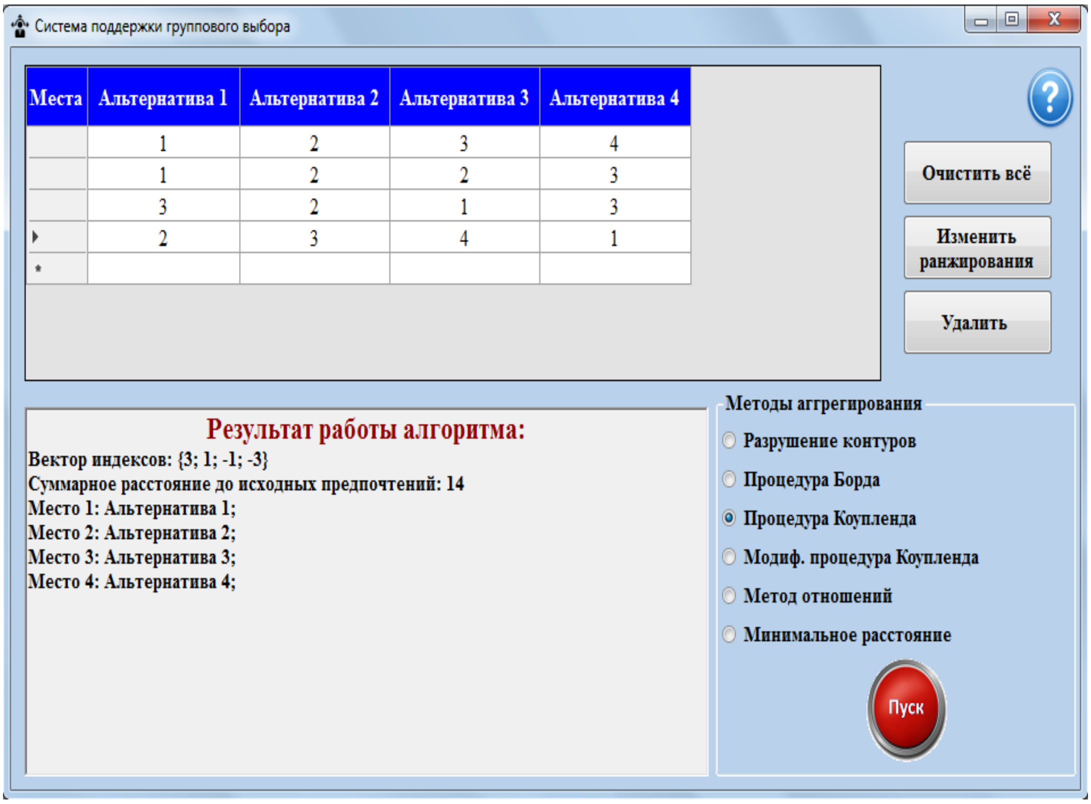


Рис. Вывод результата агрегирования

Пример прикладной задачи

Авиационно-транспортная компания решила закупить k самолетов, которые необходимо было выбрать из n предложенных. Компания пригласила m экспертов и предложила ранжировать самолеты по предпочтительности. На основе полученных от экспертов ранжирований было построено агрегированное упорядочение. Затем выбраны для закупки компанией лучшие согласно полученному упорядочению самолеты.

СПИСОК ЛИТЕРАТУРЫ

1. Берж К. Теория графов и её применение. М.: Изд-во Иностранной литературы, 1962. - 320 с.
2. Гроссман И., Магнус В. Группы и их графы. Пер. с англ. Г.М. Цукерман Под ред. В.Е. Тараканова М. Мир, 1971. - 231 с.
3. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. М.: Либроком, 2009.
4. Кофман А. Введение в прикладную комбинаторику. М.: Наука, 1975.
5. Н. Кристофидес: Теория графов. Алгоритмический подход. - М.: МИР, 1978.
6. Липский В. Комбинаторика для программистов. М.: Мир, 1988.
7. Нефедов В.Н., Осипова В.А., Смерчинская С.О., Яшина Н.П. Непротиворечивое агрегирование отношений строгого порядка // Известия высших учебных заведений. Математика. 2018. №5, с. 71-85.
8. Нефедов В.Н., Осипова В.А. Курс дискретной математики. М.; Издательство МАИ. 1992.
9. Оре О. Теория графов. М. : Наука, 1980. 336 с.
10. Оре О. Графы и их применение. М.: Мир, 1965, 174 с.
11. Осипова В.А. Курс дискретной математики. М.; Форум. 2013.
12. Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. М.: Мир, 1980, с. 476.
13. Харари Ф. Теория графов. М.: Мир, 1973.
14. Bondy J.A., Murty U.S.R. Graph theory. Springer, 2008.

ОГЛАВЛЕНИЕ

Смерчинская С.О., Яшина Н.П.

Методические указания к выполнению курсовой работы по теории графов