

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙ-
СКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»
ВОЛЖСКИЙ ФИЛИАЛ



Кафедра гуманитарных и естественнонаучных дисциплин

**Методические указания к лабораторным работам
по дисциплине
ДИСКРЕТНАЯ МАТЕМАТИКА**

Направление подготовки

09.03.01 Информатика и вычислительная техника

Направленность (профиль, специализация) образовательной программы

«Автоматизированные системы обработки информации и управления»

Квалификация

бакалавр

Чебоксары
2019

Содержание

ЛАБОРАТОРНАЯ РАБОТА № 1 «МНОЖЕСТВА. ДЕЙСТВИЯ НАД МНОЖЕСТВАМИ»	4
ЛАБОРАТОРНАЯ РАБОТА № 2 «ОТНОШЕНИЯ И ИХ СВОЙСТВА»	8
ЛАБОРАТОРНАЯ РАБОТА № 3 «КОМБИНАТОРНЫЕ КОНФИГУРАЦИИ И ЧИСЛА»	145
КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ ДЛЯ ЗАДАЧ ТЕОРИИ ГРАФОВ	21
ЛАБОРАТОРНАЯ РАБОТА № 4 «МАТРИЧНЫЕ СПОСОБЫ ПРЕДСТАВЛЕНИЯ ГРАФОВ	205
ЛАБОРАТОРНАЯ РАБОТА №5. «ПОИСК КРАТЧАЙШИХ ПУТЕЙ НА ГРАФАХ»	301
ЛАБОРАТОРНАЯ РАБОТА №6 «ПОСТРОЕНИЕ КРАТЧАЙШИХ ОСТОВЫХ ДЕРЕВЬЕВ ГРАФА»	390
ЛАБОРАТОРНАЯ РАБОТА № 7 «РАСКРАСКА ГРАФА»	456
ЛАБОРАТОРНАЯ РАБОТА № 8 «ПОТОКИ В СЕТЯХ. МЕТОД ФОРДА-ФАЛКЕРСОНА О МАКСИМАЛЬНОМ ПОТОКЕ»	501
ЛАБОРАТОРНАЯ РАБОТА № 9 «ЭЛЕМЕНТЫ ТЕОРИИ КОДИРОВАНИЯ»	54

ЛАБОРАТОРНАЯ РАБОТА №1

«МНОЖЕСТВА. ДЕЙСТВИЯ НАД МНОЖЕСТВАМИ»

Цель работы - научиться решать задачи операций над множествами.

Порядок выполнения работы

1. В соответствии с вариантом задания, определенным преподавателем, провести расчеты.
2. Оформить отчет о выполнении задания с приведением условия задачи, формул для расчета, результатов решения и заключения.

Теоретическая часть

Множеством называется совокупность некоторых элементов, объединенных каким-либо общим признаком. Элементами множества могут быть числа, фигуры, предметы, понятия и т.п.

Множества обозначаются прописными буквами, а элементы множества строчными буквами. Элементы множеств заключаются в фигурные скобки.

Если элемент x принадлежит множеству X , то записывают $x \in X$ ($\in$ - принадлежит).

Если множество A является частью множества B , то записывают $A \subset B$ ($\subset$ - содержится).

Множество может быть задано одним из двух способов: перечислением и с помощью определяющего свойства.

Например, перечислением заданы следующие множества:

- $A = \{1, 2, 3, 5, 7\}$ — множество чисел
- $X = \{x_1, x_2, \dots, x_n\}$ — множество некоторых элементов $x_1, x_2, \dots, x_n$
- $N = \{1, 2, \dots, n\}$ — множество натуральных чисел
- $Z = \{0, \pm 1, \pm 2, \dots, \pm n\}$ — множество целых чисел

Множество $(-\infty; +\infty)$ называется числовой прямой, а любое число — точкой этой прямой. Пусть a — произвольная точка числовой прямой и δ — положительное число. Интервал $(a - \delta; a + \delta)$ называется δ -окрестностью точки a .

Множество X ограничено сверху (снизу), если существует такое число c , что для любого $x \in X$ выполняется неравенство $x \leq c$ ($x \geq c$). Число c в этом случае называется верхней (нижней) гранью множества X . Множество, ограниченное и сверху и снизу, называется ограниченным. Наименьшая (наибольшая) из верхних (нижних) граней множества называется точной верхней (нижней) гранью этого множества.

Основные числовые множества

- N $\{1, 2, 3, \dots, n\}$ Множество всех натуральных чисел
- Z $\{0, \pm 1, \pm 2, \pm 3, \dots\}$ Множество целых чисел. Множество целых чисел включает в себя множество натуральных.
- Q Множество рациональных чисел.
- Кроме целых чисел имеются ещё и дроби. Дробь — это выражение вида p/q , где p —

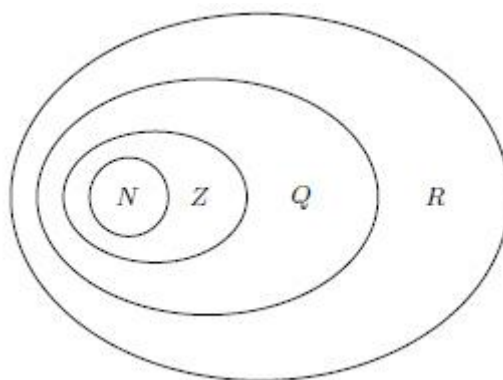
целое число, q — натуральное. Десятичные дроби также можно записать в виде p/q . Например: $0,25 = 25/100 = 1/4$. Целые числа также можно записать в виде p/q . Например, в виде дроби со знаменателем «один»: $2 = 2/1$. Таким образом, любое рациональное число можно записать десятичной дробью — конечно или бесконечной периодической.

Множество всех вещественных чисел.

Иррациональные числа — это бесконечные непериодические дроби. К ним относятся:

- R
- число π — отношение длины окружности к её диаметру;
 - число e — названное в честь Эйлера и др.;

Вместе два множества (рациональных и иррациональных чисел) — образуют множество действительных (или вещественных) чисел.



Если множество не содержит ни одного элемента, то оно называется пустым множеством и записывается $\emptyset$.

Операции над множествами

Два множества A и B равны ($A=B$), если они состоят из одних и тех же элементов.

Например, если $A=\{1,2,3,4\}$, $B=\{3,1,4,2\}$ то $A=B$.

Объединением (суммой) множеств A и B называется множество $A \cup B$, элементы которого принадлежат хотя бы одному из этих множеств.

Например, если $A=\{1,2,4\}$, $B=\{3,4,5,6\}$, то $A \cup B = \{1,2,3,4,5,6\}$

Пересечением (произведением) множеств A и B называется множество $A \cap B$, элементы которого принадлежат как множеству A , так и множеству B .

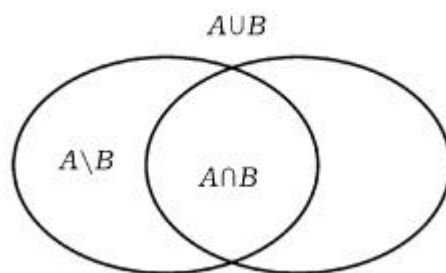
Например, если $A=\{1,2,4\}$, $B=\{3,4,5,2\}$, то $A \cap B = \{2,4\}$

Разностью множеств A и B называется множество AB , элементы которого принадлежат множеству A , но не принадлежат множеству B .

Например, если $A=\{1,2,3,4\}$, $B=\{3,4,5\}$, то $AB = \{1,2\}$

Симметричной разностью множеств A и B называется множество $A \Delta B$, являющееся объединением разностей множеств AB и BA , то есть $A \Delta B = (AB) \cup (BA)$.

Например, если $A=\{1,2,3,4\}$, $B=\{3,4,5,6\}$, то $A \Delta B = \{1,2\} \cup \{5,6\} = \{1,2,5,6\}$



Свойства операций над множествами

Свойства перестановочности

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

Сочетательное свойство

$$(A \cup B) \cup C = A \cup (B \cup C)$$

$$(A \cap B) \cap C = A \cap (B \cap C)$$

Варианты заданий

1. Для множеств A , B и C определить:

а) $A \cup C = ?$; б) $A \cap B = ?$; в) $A \cup B \setminus C = ?$; г) $\overline{(A \cup B \cup C)} = ?$.

Варианты:

1. $A = \{x \mid x \in [1; 5]\}$; $B = \{x \mid x \in (2; 5]\}$; $C = \{x \mid x \in [4; 7]\}$.

2. $A = \{x \mid x \in (-1; 5]\}$; $B = \{x \mid x \in [0; 4]\}$; $C = \{x \mid x \in (3; 7]\}$.

3. $A = \{x \mid x \in (2; 5]\}$; $B = \{x \mid x \in [3; 6]\}$; $C = \{x \mid x \in [5; 9]\}$.

4. $A = \{x \mid x \in (-5; 2]\}$; $B = \{x \mid x \in [0; 1]\}$; $C = \{x \mid x \in [1; 9]\}$.

5. $A = \{x \mid x \in [-9; 0]\}$; $B = \{x \mid x \in [-5; -1]\}$; $C = \{x \mid x \in [-1; 7]\}$.

6. $A = \{x \mid x \in (0; 4]\}$; $B = \{x \mid x \in [3; 5]\}$; $C = \{x \mid x \in [4; 8]\}$.

7. $A = \{x \mid x \in [1; 3]\}$; $B = \{x \mid x \in (2; 5]\}$; $C = \{x \mid x \in [3; 9]\}$.

8. $A = \{x \mid x \in (-2; 2]\}$; $B = \{x \mid x \in [-1; 1]\}$; $C = \{x \mid x \in [1; 5]\}$.

9. $A = \{x \mid x \in (-5; 3]\}$; $B = \{x \mid x \in [-4; 2]\}$; $C = \{x \mid x \in \mathbb{R} \setminus (-7; -4]\}$.

10. $A = \{x \mid x \in (-2; 5]\}$; $B = \{x \mid x \in [2; 6]\}$; $C = \{x \mid x \in [5; 8]\}$.

2. Какие выводы можно сделать относительно множеств A и B (соответствующую ситуацию, для наглядности, желательно изобразить с помощью кругов Эйлера), если для них верно приведенное равенство.

Варианты:

1. $A \cup B = A$.

2. $A \cap B = A$.

- 3 $A \setminus B = A$.
- 4 $A \setminus B = B \setminus A$.
- 5 $A \cap B = B; B \setminus A = \emptyset$.
- 6 $A \setminus B = B \setminus A = \emptyset$.
- 7 $A \cup B = A; A \setminus B = A$.
- 8 $A \setminus B = A; B \setminus A = B$.
- 9 $A \cap B = A; A \setminus B = \emptyset$.
- 10 $A \cup B = B; B \setminus A = B$.

3.Применив правила приоритета операций с множествами, расставьте скобки в заданном выражении.

Варианты:

1. $(A \setminus B \cup \bar{A} \cap D) \setminus B \cup \bar{D} = ?$.
2. $(\bar{B} \setminus A \cap D \cup \bar{C}) \setminus C \cap D = ?$.
3. $(A \setminus \bar{A} \cap D \cup \bar{C}) \setminus B \cup C = ?$.
4. $(\bar{B} \setminus A \cap D \cup C) \setminus C \cap \bar{D} = ?$.
5. $(A \cap \bar{B} \setminus B \cap D \cup \bar{C}) \setminus B = ?$.
6. $(A \cap B \setminus B \cup \bar{A} \cap D) \setminus \bar{D} = ?$.
7. $(A \cap \bar{B} \setminus \bar{A} \cap D \cup C) \setminus C = ?$.
8. $(\bar{B} \setminus B \cup \bar{A} \cap D) \setminus B \cup C = ?$.
9. $(A \setminus B \cup A \cap \bar{C}) \setminus C \cap \bar{D} = ?$.
10. $(A \cap \bar{B} \setminus A \cap D \cup \bar{C}) \setminus B = ?$.

4. Применяя законы алгебры множеств, упростите Ваше выражение.

Варианты:

1. $\overline{(\bar{A} \cup B \cup C)} \cap (A \cap (B \cup C)) \cap \bar{B} = ?$
2. $\overline{\bar{A} \cup C} \cup (B \cup B \cap C) \cap (\bar{B} \cap \overline{B \cup C}) = ?$
3. $(A \cap \bar{B} \cup C) \cap (A \cup B) \cap \bar{C} = ?$
4. $A \cap (((B \cap \bar{C} \cup \overline{C \cup B}) \cap \bar{C}) \cup \bar{A}) = ?$
5. $A \cap (B \cap C) \cup (\bar{A} \cap B \cap C \cup \bar{B}) = ?$
6. $(\bar{A} \cap B \cap C) \cap (A \cap (B \cup C)) \cap \bar{B} = ?$
7. $\bar{A} \cap C \cup (B \cup B \cap C) \cap (\bar{B} \cap \overline{B \cup C}) = ?$

$$8 \quad (\bar{A} \cup B \cap \bar{C}) \cap (A \cup B) \cap \bar{C} = ?$$

$$9 \quad \bar{A} \cup \left(\left((B \cap \bar{C} \cup \overline{C \cup B}) \cap \bar{C} \right) \cup \bar{A} \right) = ?$$

$$10 \quad (\bar{A} \cup \bar{B} \cup \bar{C}) \cup (\bar{A} \cap \bar{B} \cap C \cup \bar{B}) = ?$$

Контрольные вопросы

1. Опишите операции над множествами и их свойства.
2. Какие числовые множества вы знаете.
3. Охарактеризуйте полученные результаты.

ЛАБОРАТОРНАЯ РАБОТА № 2 «ОТНОШЕНИЯ И ИХ СВОЙСТВА»

Цель работы: изучить способы задания отношений, операции над отношениями и свойства отношений.

Порядок выполнения работы

1. В соответствии с вариантом задания, определенным преподавателем, провести расчеты.
2. Оформить отчет о выполнении задания с приведением условия задачи, формул для расчета, результатов решения и заключения.

Теоретическая часть

Декартово (прямое) произведение множеств. Отношения.

Определение 1. Декартовым (прямым) произведением множеств X и Y называется множество $X \times Y$, элементами которого являются все возможные упорядоченные пары (x, y) такие, что $x \in X, y \in Y$.

Для упорядоченных пар (x, y) используется также обозначение $\langle x, y \rangle$.

Определение 2. Прямым (декартовым) произведением множеств $X_1, X_2, \dots, X_n$ называется совокупность всех упорядоченных n -ок $(x_1, \dots, x_n)$ таких, что $x \in X_1, \dots, x_n \in X_n$. Если $X_1 = X_2 = \dots = X_n$, то пишут $X_1 \times X_2 \times \dots \times X_n = X^n$.

Пример.

1. Пусть $X = \{1, 2, 3\}$, $Y = \{0, 1\}$. Тогда

$$X \times Y = \{\langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 2, 0 \rangle, \langle 2, 1 \rangle, \langle 3, 0 \rangle, \langle 3, 1 \rangle\};$$

$$Y \times X = \{\langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 0, 3 \rangle, \langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 1, 3 \rangle\}.$$

Определение 2. Бинарным (или двуместным) отношением ρ на паре множеств A и B называется произвольное подмножество множества $X \times Y$.

Тот факт, что элементы a и b находятся в бинарном отношении R , записывается следующим образом: $(a, b) \in R$ или aRb .

Если ρ есть отношение и пара (x, y) принадлежит этому отношению, то наряду с записью $(x, y) \in \rho$ употребляется запись $x\rho y$. Элементы x и y называются компонентами (или координатами) отношения ρ .

Примеры бинарных отношений.

1. Отношения на множестве натуральных чисел N :

- отношение $\leq$ – выполняется, например, для пар $(7, 9)$ и $(7, 7)$, но не выполняется для пар $(9, 7)$;

- отношение “иметь общий делитель, отличный от единицы” – выполняется, например, для пар $(6, 9)$, $(4, 2)$, $(2, 4)$, $(4, 4)$, но не выполняется для пар $(7, 9)$ и $(9, 7)$;

2. Отношения на множестве точек плоскости:

- отношение “находиться на одинаковом расстоянии от начала координат” – выполняется для пар точек находящихся на одной и той же окружности с центром в начале координат;

- отношение “находиться на разном расстоянии от начала координат” выполняется для тех и только тех пар точек, для которых не выполняется предыдущее отношение/

Определение 3. N -арным отношением α на множествах $X_1, X_2, \dots, X_n$ называется произвольное подмножество множества $\alpha \subseteq X_1 \times X_2 \times \dots \times X_n$.

Говорят, что элементы $a_1, a_2, \dots, a_n$ находятся в отношении α , если $(a_1, a_2, \dots, a_n) \in \alpha$.

Определение 4. Областью определения бинарного отношения ρ называется множество $D_\rho = \{x \mid \text{существует такое } y, \text{ что } x\rho y\}$.

Определение 5. Областью значений бинарного отношения ρ называется множество $E_\rho = \{y \mid \text{существует такое } x, \text{ что } x\rho y\}$.

Пусть $\rho \subseteq X \times Y$ определено в соответствии с изображением на рис. 1. Область определения D_ρ и область значений E_ρ определяются соответственно:

$$D_\rho = \{x: (x, y) \in \rho\}, E_\rho = \{y: (x, y) \in \rho\}.$$



Рис.1. Изображение

Бинарное отношение можно задать любым из способов задания множеств. Помимо этого отношения, определенные на конечных множествах обычно задаются:

1. списком (перечислением) пар, для которых это отношение выполняется.
2. матрицей смежности – бинарному отношению соответствует квадратная матрица $c = (c_{ij})$ порядка n , в которой элемент c_{ij} , стоящий на пересечении i -той строки и j -го столбца, равен 1, если a_i и a_j находятся в отношении α , и – 0, в противном случае:

$$c_{i,j} = \begin{cases} 1, \text{ если } a_i \alpha b_j \\ 0, \text{ в противном случае} \end{cases}$$

Пример.

Пусть $M = \{1, 2, 3, 4, 5, 6\}$. Задать в явном виде (списком) и матрицей смежности отношение ρ , определенное на множестве $M \times M$, если ρ означает «быть строго меньше».

Отношение ρ , как множество, содержит все пары элементов a, b из M такие, что $a < b$. Тогда

$\rho = \{(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6), (5, 6)\}$.

Матрица смежности отношения ρ имеет вид:

$$\rho = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Свойства бинарных отношений. Специальные бинарные отношения

Определение 6. Бинарное отношение ρ на множестве X называется рефлексивным, если для любого элемента $x \in X$ выполняется $x \rho x$.

Определение 7. Бинарное отношение ρ на множестве X , заданное матрицей смежности, в которой все элементы главной диагонали – единицы, а на остальных местах стоят нули, называется диагональным.

Определение 8. Бинарное отношение ρ на множестве X называется симметричным, если для любых $x, y \in X$ из $x \rho y$ следует $y \rho x$.

Матрица смежности симметричного отношения симметрична относительно главной диагонали: $a_{i,j} = a_{j,i}$.

Определение 9. Отношение α называется антирефлексивным, если ни для одного элемента $a \in X$ не выполняется $a \alpha a$.

Главная диагональ его матрицы смежности содержит только нули.

Определение 10. Бинарное отношение ρ на множестве X называется транзитивным, если для любых $x, y, z \in X$ из $x \rho y$ и $y \rho z$ следует $x \rho z$.

Пример.

Отношения: “равенство”, $\leq$, “жить в одном городе” являются транзитивными, а отношение “быть сыном” не транзитивно.

Определение 11. Бинарное отношение, которое одновременно рефлексивно, симметрично и транзитивно отношение на множестве X называется отношением эквивалентности на множестве X .

Пример.

1. Отношение равенства на множестве целых чисел есть отношение эквивалентности.

2. Отношение подобия на множестве треугольников есть отношение эквивалентности.

3. Отношение «строго меньше» на множестве действительных чисел не рефлексивно, не симметрично и транзитивно на этом множестве.

4. Отношение перпендикулярности прямых не рефлексивно, симметрично, не транзитивно.

Пусть ρ - отношение эквивалентности на множестве X .

Определение 12. Классом эквивалентности, порожденным элементом x , называется подмножество множества X , состоящее из тех элементов $y \in X$, для которых $x\rho y$. Класс эквивалентности, порожденный элементом x , обозначается через $[x]$: $[x] = \{y \mid y \in X \text{ и } x\rho y\}$.

Определение 13. Бинарное отношение ρ на множестве X называется антисимметричным, если для любых $x, y \in X$ из $x\rho y$ и $y\rho x$ следует, что $x=y$.

Определение 14. Бинарное отношение, которое одновременно рефлексивно, антисимметрично и транзитивно называется отношением частичного порядка.

Пример.

1. Отношение « $x \leq y$ » на множестве действительных чисел – отношение частичного порядка.

2. Схема организации подчинения в учреждении – это отношение частичного порядка на множестве должностей.

Определение 15. Множество с заданным на нем отношением частичного порядка называется частично упорядоченным. Если при этом любые два элемента находятся в отношении, то такое множество называется линейно упорядоченным.

Элементы, находящиеся в отношении частичного порядка называются сравнимыми.

Определение 16. Бинарное отношение называется отношением строгого порядка, если оно одновременно антирефлексивно, антисимметрично и транзитивно.

Примеры.

1. Множество действительных чисел линейно упорядочено относительно отношений $\leq$, $\geq$.

2. Отношения $<$ и $>$ на множестве действительных чисел являются отношениями строгого порядка.

3. Отношение включения $\subseteq$ на системе подмножеств множества M задает частичный порядок, причем все подмножества множества M линейно упорядочены относительно отношения $\subseteq$.

4. Множества $\{1, 2\}$, $\{1, 2, 3\}$ сравнимы относительно отношения $\subset - \{1, 2\} \subset \{1, 2, 3\}$, а множества $\{1, 2\}$ и $\{1, 3, 4\}$ не сравнимы.

5. Отношения подчиненности на предприятии задает строгий частичный порядок. В нем несравнимыми будут сотрудники разных отделов.

6. Пусть порядок букв конечного алфавита A зафиксирован, т. е. всегда один и тот же. Тогда этот порядок определяет линейное упорядочение букв, которое назовем отношением предшествования и обозначим через $\prec$. Если a_i предшествует a_j в алфавите, то это записывается следующим образом: $a_i \prec a_j$. На основе отношения предшествования букв строится отношение предшествования слов. Это отношение задает линейное упорядочение множества всех конечных слов в алфавите A , которое называется лексикографическим упорядочением слов. Примером такого упорядочения является расположение слов в словаре.

8. Если рассматривать числа в позиционных системах счисления (например десятичной), как слова в алфавите цифр, то их лексикографическое упорядочение совпадает с обычным упорядочением чисел, при этом предполагается, что все сравниваемые числа имеют одинаковое число разрядов (такое дополнение разрядами можно провести искусственно):

$20 < 1073$;

$0020 \prec 1073$.

Варианты заданий.

1. Привести по два примера бинарных отношений следующей природы:

- отношения на множестве чисел;
- отношения на множестве геометрических объектов;
- отношения на множестве людей;
- отношения на множестве географических объектов;
- отношение на множестве предметов окружающего мира;

2. Описать матрицы смежности бинарных отношений на множестве из 7 элементов для двух отношений.

3. Привести примеры отношений (по два отношения для каждого примера) со следующими свойствами:

- рефлексивность;
- антирефлексивность;
- симметричность;
- антисимметричность;
- транзитивность;
- эквивалентность.

Дать обоснование каждому примеру.

4. Выяснить, какими свойствами обладают заданные бинарные отношения α и β ?

Результаты описать в виде таблицы:

	Рефлексивн.	Симметрич	Антисимметр.	Транзитив.	Эквивалент.	Частич. порядок	Функция
α							
β							

отметив соответствующий результат знаком «+» или «-», указать, если это необходимо, вид функции. Изобразить соответствующие отношения графически. Выводы обосновать.

Исходные данные:

матрицы смежности отношений

α

β

№		1	2	3	4				
1	1	0	1	1	0	0	0	1	0
	2	1	1	1	1	1	0	0	1
	3	0	1	0	0	0	0	0	0
	4	0	0	1	1	0	1	1	1
2		1	0	0	1	1	0	0	1
		1	0	0	1	1	0	0	0
		1	0	1	0	0	0	0	0
		0	1	0	0	0	0	0	1
3		0	1	1	1	1	1	0	1
		1	1	1	0	1	1	0	0
		1	1	0	1	0	1	0	0
		1	0	0	1	1	1	0	0
4		1	1	1	0	0	1	1	1
		1	1	1	1	1	0	0	0
		1	1	0	1	1	0	0	0
		0	0	0	1	0	1	1	1
5		1	1	1	1	0	0	0	1
		1	0	1	1	1	0	1	0
		0	1	0	1	0	1	0	1
		0	0	1	0	0	1	1	0
6		0	1	0	0	1	0	1	1
		0	1	0	1	0	0	0	0
		1	0	0	0	0	1	0	1
		1	1	0	1	1	0	0	1
7		0	0	0	0	1	1	0	0
		0	0	1	0	1	1	1	0

		<table><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	0	1	1	0	1	0	0	0	<table><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	1	0	0	0	0	1	1	1																
0	1	1	0																																
1	0	0	0																																
1	0	0	0																																
0	1	1	1																																
8		<table><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	<table><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0
1	0	0	0																																
0	0	1	1																																
0	1	0	0																																
1	0	1	0																																
1	0	0	1																																
1	1	1	1																																
1	1	1	1																																
1	0	1	0																																
9		<table><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>	1	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	<table><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	1	1	0	0	0	1	0	1	0	0	0	1	1	0	0	0
1	0	1	1																																
0	0	1	0																																
1	0	1	1																																
1	0	0	1																																
1	1	0	0																																
0	1	0	1																																
0	0	0	1																																
1	0	0	0																																
10		<table><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	<table><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	1	0	0	0	1	1	0	1	1	1	1	0	1	1	1	0
1	0	1	1																																
1	1	0	1																																
0	1	0	1																																
0	1	0	1																																
1	0	0	0																																
1	1	0	1																																
1	1	1	0																																
1	1	1	0																																

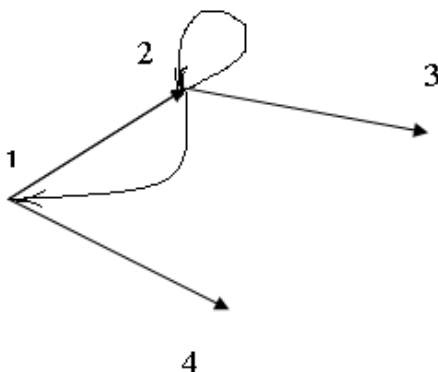
5. Подготовить отчет по следующему плану:

1. Краткие теоретические сведения.
2. Примеры бинарных отношений для задачи 1.
3. Матрицы смежности бинарных отношений для задачи 2.
3. Примеры бинарных отношений для задачи 3.
4. Анализ бинарных отношений для задачи 4.

Пример решения задачи 4 для отношения α :

0	1	0	1
1	1	1	0
0	0	0	0
0	0	0	0

Графическое изображение отношения α :



	Рефлексивн.	Симметрич	Антисимметр.	Транзитив.	Эквивалент.	Частич. порядок	Функция
α	—	—	—	—	—	—	—

Не рефлексивно, т.к. не каждая вершина имеет петлю.

Не симметрично, т.к., например, элементы 1 и 4 находятся в отношении α : $1 \alpha 4$, но не выполняется $4 \alpha 1$

Не антисимметрично, т. к. $1 \alpha 2$ и $2 \alpha 1$.

Не транзитивно, т. к. выполняется $1 \alpha 2$ и $2 \alpha 1$, но нет петли $1 \alpha 1$.

Не эквивалентность, т. к. не транзитивно.

Не частичный порядок, т.к. не антисимметрично.

Не функция, т. к. элемент 2 имеет два образа: 1 и 2.

Контрольные вопросы

1. Что называется декартовым произведением множеств.
2. Приведите примеры бинарных отношений.
3. Что называется областью бинарных отношений ρ .
4. Какие элементы называются сравнимыми.
5. Охарактеризуйте полученные результаты.

ЛАБОРАТОРНАЯ РАБОТА № 3 «КОМБИНАТОРНЫЕ КОНФИГУРАЦИИ И ЧИСЛА»

Цель работы: Изучить методы комбинаторного анализа.

Порядок выполнения работы

1. В соответствии с вариантом задания, определенным преподавателем, провести расчеты.
2. Оформить отчет о выполнении задания с приведением условия задачи, формул для расчета, результатов решения и заключения.

Теоретические сведения.

Комбинаторика – раздел математики, посвященный решению задач выбора и расположения элементов некоторого (обычно конечного) множества в соответствии с заданными правилами. Каждое такое правило определяет способ построения некоторой конструкции из элементов исходного множества, называемой комбинаторной конфигурацией. Простейшими примерами комбинаторных конфигураций являются перестановки, размещения, сочетания и разбиения.

Основные правила комбинаторики.

Определение. Размещениями из n элементов по m называют такие соединения, каждое из которых содержит m элементов, взятых из числа данных n элементов и которые отличаются друг от друга либо самими элементами (хотя бы одним), либо лишь порядком их расположения.

Число размещений из n элементов по m определяется следующими формулами:

$$A_n^m = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-m+1).$$

$$A_n^m = \frac{n!}{(n-m)!};$$

где $n! = 1 \times 2 \times 3 \times \dots \times n$, причем $0! = 1$.

Определение. Сочетаниями из n элементов по m называют такие соединения, каждое из которых содержит m элементов, взятых из числа данных n элементов и которые отличаются друг от друга, по крайней мере, одним элементом.

Число таких комбинаций можно вычислить по формуле:

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Справедливы следующие равенства:

$$C_n^n = 1; C_n^1 = n; C_n^0 = 1; C_n^m = C_n^{n-m}.$$

Определение. Перестановками из n элементов называются такие соединения, каждое из которых содержит все n элементов и, которые отличаются друг от друга только порядком расположения элементов.

Число перестановок определяется следующей формулой:

$$P_n = n!$$

Комбинации элементов с повторениями

Все приведенные формулы справедливы в том случае, когда n элементов множества A различны. Если же некоторые элементы повторяются, то в этом случае рассматриваются комбинации с повторениями, число которых вычисляется по другим формулам.

Определение. Размещениями с повторениями из элементов n классов по m называются такие выборки, которые, имея по m элементов, выбранных из числа элементов данных n классов генеральной совокупности с повторениями, отличаются одна от другой либо составом элементов, либо порядком их расположения.

Характерный пример таких размещений — совокупность n -значных номеров автомашин.

Число таких размещений обозначим символом $\overline{A_n^m}$.

$$\overline{A_n^m} = n^m$$

Определение. Сочетаниями с повторениями из элементов n классов по m называются такие размещения с повторениями из элементов n классов по m , которые одно от другого отличаются хотя бы одним элементом.

Число таких сочетаний обозначим $\overline{C_n^m}$.

$$\overline{C_n^m} = P_{m,n-1}.$$

Характерный пример таких сочетаний — всевозможные различные комплекты m предметов, составленные из предметов n различных видов.

Определение. Перестановками с повторениями по k элементов из p различных классов называются размещения с повторениями объема k , которые од-

но от другого отличаются только порядком расположения элементов, когда от i -го класса в каждой выборке участвует k_i элементов.

Число таких перестановок обозначим $P_{k_1; k_2; \dots; k_n}$.

Число перестановок с повторениями

$$P_{k_1; k_2; \dots; k_n} = \frac{k!}{k_1! \cdot k_2! \cdot \dots \cdot k_n!}, \text{ где } k = k_1 + k_2 + \dots + k_n.$$

Рассмотрим примеры применения выведенных формул.

Бином Ньютона

С числами C_n^k связано функциональное тождество, называемое формулой бинома Ньютона. Биномом Ньютона называют разложение вида:

$$\begin{aligned} (a+b)^n &= C_n^0 a^n b^0 + C_n^1 a^{n-1} b^1 + C_n^2 a^{n-2} b^2 + \dots + C_n^m a^{n-m} b^m + \dots + C_n^n a^0 b^n = \\ &= a^n + n a^{n-1} b + \frac{n(n-1)}{2!} a^{n-2} b^2 + \dots + \frac{n!}{(n-m)! m!} a^{n-m} b^m + \dots + b^n, \text{ где } m < n \end{aligned}$$

Варианты заданий

Задание 1.

1.1. Сколько четырехзначных чисел можно записать с помощью цифр 0, 1, 2, 3, 4, 5, если

- а) каждая цифра входит в число только один раз?
- б) цифры в числе могут повторяться?
- с) это число делится нацело на 4?
- д) сколько существует множеств, содержащих 4 любых цифры из указанных?
- е) числа не начинаются с 35?

1.2. Сколько пятизначных чисел можно записать с помощью цифр 0, 1, 2, 3, 4, 5, 6, если

- а) каждая цифра входит в число только один раз?
- б) цифры в числе могут повторяться?
- с) это число делится нацело на 5?
- д) в скольких из них каждая следующая цифра больше предыдущей?
- е) числа не начинаются с 35?

1.3. Сколько четырехзначных чисел можно записать с помощью цифр 0, 1, 2, 3, 4, 5, 6, если

- а) каждая цифра входит в число только один раз?
- б) цифры в числе могут повторяться?
- с) это число делится нацело на 4?
- д) сколько существует множеств, содержащих 4 любых цифры из указанных?
- е) в скольких из них каждая следующая цифра меньше предыдущей?

1.4. Сколькими способами можно составить трехцветный полосатый флаг, если имеется материал 5-ти различных цветов (красный, синий, желтый, зеленый, белый) так, чтобы

- а) все полосы имели различные цвета?
- б) одна из полос должна быть красной?
- в) ни одна из полос не должна быть красной?
- г) полосы могут иметь одинаковый цвет?

1.5. Существуют матрицы размерности $(3,4)$, состоящие из 0 и 1, они называются бинарными.

- а) сколько существует матриц размерности $(3,4)$, состоящих из 0 и 1?
- б) сколько среди таких матриц, у которых различны все строки?
- в) сколько среди таких матриц, у которых различны все столбцы?
- г) сколько среди таких матриц, у которых различны все строки и столбцы?

1.6. Сколько существует шестизначных телефонных номеров,

- а) состоящих из различных цифр?
- б) могут содержать одинаковые цифры?
- в) которые являются четными?
- г) которые начинаются с числа 44?

1.7. Имеется 10 электронных документов.

- а) сколькими способами три из них можно поместить в очередь на печать на лазерном принтере?
- б) электронные документы передаются по семи каналам связи из пункта А в пункт В. В пункте В документы редактируются и передаются обратно в пункт А. Сколькими способами можно передать документы из пункта А в пункт В и обратно?
- в) сколькими способами их можно передать по электронной почте, если каждый документ передается отдельно?
- г) сколькими способами их можно передать по электронной почте, если документы А и В должны быть переданы вместе?

1.8. Имеется 10 символов, которые можно использовать для составления пятибуквенного кода электронного документа?

- а) сколько существует таких кодов?
- б) сколько кодов не имеют повторяющихся символов?
- в) сколько кодов содержат два указанных символа α и β ?
- г) сколько кодов начинаются с символа α или β ?

1.9. Сколькими способами можно расположить в 12 лузах 3 белых шара и 9 черных

- а) часть луз может быть пустой и лузы считаются различными?
- б) ни одна из луз не может быть пустой?
- в) в первой лузе должен лежать белый шар и нет пустых луз?
- г) все шары считаются различными?

1.10. Двенадцать человек, включая Мари и Петера, являются кандидатами в комитет пяти.

- а) сколько разных комитетов можно набрать из 12 человек?
- б) сколько из них включают либо Мари, либо Петера, но не обоих?
- с) сколько из них включают Мари и Петера?
- д) сколько из них включают Мари или Петера?

Задание 2. Найти решение следующих выражений:

$$2.1. \text{ а) } A_{2n}^3 = 20A_n^2; \quad \text{б) } \frac{(n-1)!}{(n-3)!} < 72; \quad \text{в) } \begin{cases} A_x^y : A_x^{y-1} = 10, \\ C_x^y : C_x^{y-1} = \frac{5}{3}; \end{cases}$$

г) Найти коэффициент при x^3 в выражении $(1+x)^3 + (1+x)^4 + \dots + (1+x)^{15}$.

$$2.2. \text{ а) } A_x^3 + C_x^{x-2} = 14x; \quad \text{б) } C_{19}^{k-1} < C_{19}^k; \quad \text{в) } \begin{cases} A_{5x}^{n-3} : A_{5x}^{n-2} = 1:7, \\ C_{5x}^{n-3} : C_{5x}^{n-2} = 4:7; \end{cases}$$

г) В разложении $\left(x\sqrt{x} + \frac{1}{x^4}\right)^n$ биномиальный коэффициент третьего члена на 44 больше коэффициента второго члена. Найти член, не содержащий x .

$$2.3. \text{ а) } A_{k-4}^2 + A_{k-3}^2 + A_{k-2}^2 = 20; \quad \text{б) } \frac{(2n-1)!}{(2n-3)!} > 420; \quad \text{в) } \begin{cases} A_{2n}^{3x} : A_{2n}^{3x-1} = 8, \\ C_{2n}^{3x} : C_{2n}^{3x-1} = 8:9; \end{cases}$$

г) Определить коэффициент k в $ka^6b^6c^{12}$ следующем члене многочлена получаемого из алгебраического выражения $(a+b+c)^3(a^2+b^2+c^2)^6$:

$$2.4. \text{ а) } C_{x+1}^5 = \frac{3A_x^2}{8}; \quad \text{б) } C_{2n}^7 > C_{2n}^5; \quad \text{в) } \begin{cases} A_{k-2}^n : A_{k-2}^{n-1} = 8, \\ C_{k-2}^n : C_{k-2}^{n-1} = 1,6. \end{cases}$$

г) Определить коэффициент k в $ka^2b^6c^6$ следующем члене многочлена получаемого из алгебраического выражения $(a+b+c)^3(a^2+b^2+c^2)^6$:

$$2.5. \text{ а) } C_{x-1}^2 = 6; \quad \text{б) } \frac{(n+2)!}{(n+1)1(n+2)} < 1000; \quad \text{в) } \begin{cases} C_x^{y+1} = 2,5x \\ C_{x-1}^y = 10 \end{cases}$$

г) Найти x , если второй член разложения $(\sqrt[3]{x} + x^{-1})^5$ равен 20;

$$2.6. \text{ а) } A_x^4 \cdot P_{x-4} : P_{x-2} = 42; \quad \text{б) } \frac{(n+1)!}{(n-3)!} < 72; \quad \text{в) } \begin{cases} C_x^y = C_x^{y+2} \\ C_x^2 = 153 \end{cases}$$

г) Найти x , если четвертый член разложения $(\sqrt{x} + x^{-1})^6$ равен $\frac{5}{9}$.

$$2.7. \text{ а) } C_x^{x-2} = 28; \quad \text{б) } C_n^5 < C_n^4; \quad \text{в) } \begin{cases} A_y^x \div P_{x-1} + C_y^{y-x} = 126 \\ P_{x+1} = 720 \end{cases}$$

г) Найти x , если четвертый член разложения $(\sqrt{x} + x^{-1})^6$ равен $\frac{5}{9}$.

$$2.8. \text{ а) } \frac{P_{x+3}}{A_x^5 \cdot P_{x-5}} = 720; \quad \text{б) } C_n^5 > C_n^4 \quad \text{в) } \begin{cases} A_{2n}^{3x} \div A_{2n}^{3x-1} = 8 \\ C_{2n}^{3x} \div C_{2n}^{3x-1} = 8 \div 9 \end{cases}$$

г) Найти четвертый член разложения бинома $\left(\frac{a}{\sqrt{x}} + \frac{\sqrt{x}}{a}\right)^n$, если отношение коэффициента второго члена к коэффициенту первого члена равно $\frac{11}{2}$.

$$2.9. \text{ а) } C_x^7 = C_x^5; \quad \text{б) } C_{19}^{k-1} < C_{19}^k \quad \text{в) } \begin{cases} A_x^y \div A_x^{y-1} = 8 \\ C_x^y \div C_x^{y-1} = 1,6 \end{cases}$$

г) Найдите коэффициент при a^3b^5c после раскрытия скобок в выражении $(a + b + c + 5)^{10}$.

$$2.10. \text{ а) } C_{x+8}^{x+3} = 5A_{x+6}^3; \quad \text{б) } C_{15}^{k-2} < C_{15}^k \quad \text{в) } \begin{cases} C_x^{y+1} = 2,5x \\ C_{x-1}^y = 10 \end{cases}$$

г) Найдите коэффициент при a^3b^5c после раскрытия скобок в выражении $(a + b - c + 5)^{11}$.

Контрольные вопросы

1. Что называется размещением, сочетанием, повторением.
2. Укажите используемые формулы комбинаторики для элементов с повторениями.
3. Охарактеризуйте полученные результаты работы.

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ ДЛЯ ЗАДАЧ ТЕОРИИ ГРАФОВ.

Большинство рассматриваемых в рамках курса задач и алгоритмов уже давно имеют свою программную реализацию на различных языках программирования и входят в число различных программных комплексов, генераторов тестовых заданий и калькуляторов. Например, существует разработка преподавателя Саратовского университета Печенкина В.В., называемая **Grin**. Grin является простой, доступной, полезной для студентов и преподавателей университетов программой. Она может быть использована не только математиками, но и экономистами, социологами, всеми, кто так или иначе интересуется дискретными моделями. Одними из основных достоинств программы является простой и удобный интерфейс, легкость в освоении и работе (рис.2).

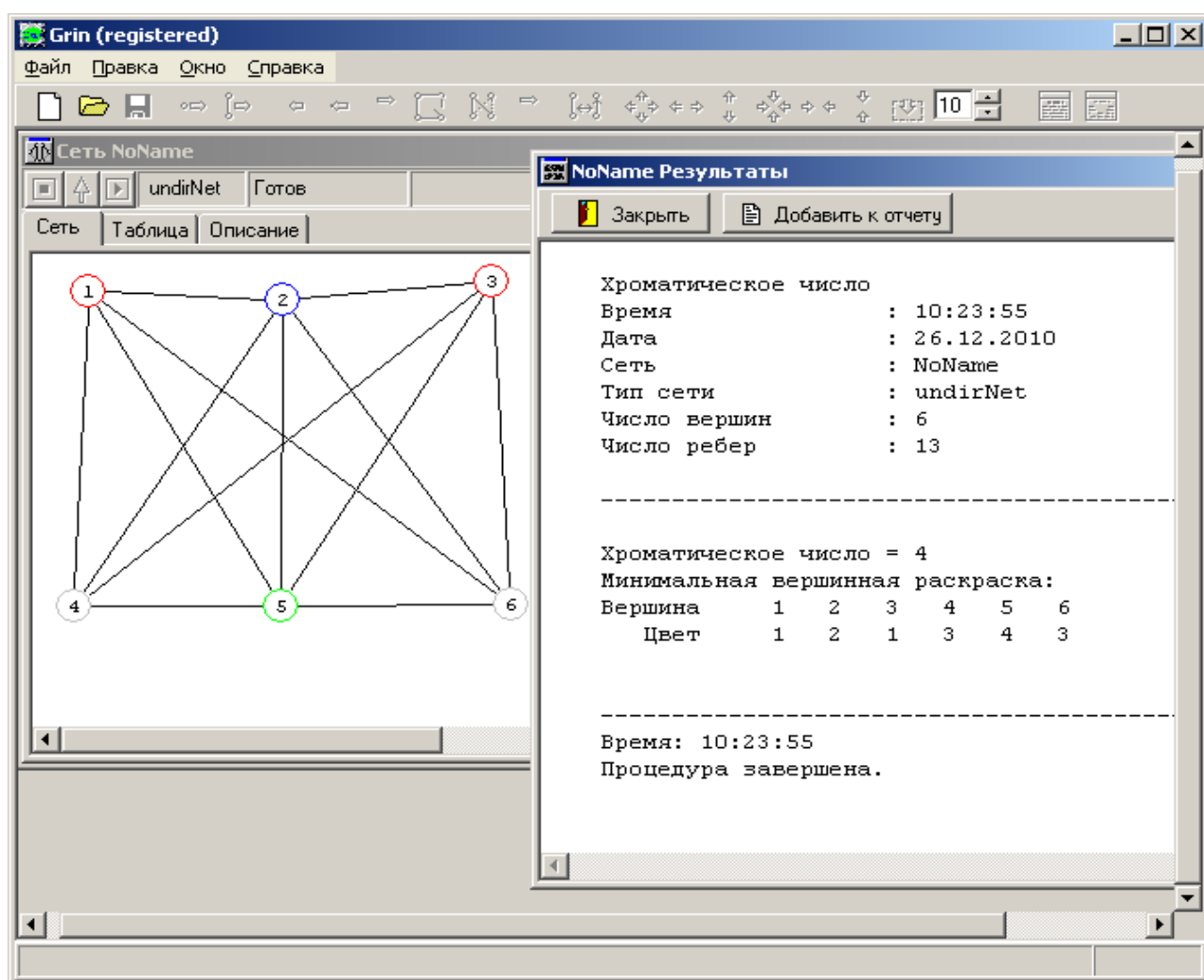


Рис.2. Программа Grin для решения задач на графах

С помощью Grin можно создавать, интерактивно редактировать и исследовать графы и сети (сеть в программе называют взвешенные графы, т.е. графы у которых все ребра имеют вес). Графы сохраняются на внешние носители и легко могут быть загружены. Справочная система содержит информацию не

только по самой программе, но и подробную справку по теории графов и оптимизационным задачам теории сетей.

Код программы не является свободным. Следовательно, используя Grin, нет возможности с ее помощью продемонстрировать решение задач алгоритмизации с использованием инструментария теории графов.

Еще одним средством для работы с графами является библиотека GOBLIN. Это библиотека классов, предоставляющая функции решения задач на графах. Данная программа является программным обеспечением с открытым исходным кодом и распространяется под лицензией GNU LGPL.

Библиотека GOBLIN реализует самые эффективные на сегодняшний момент полиномиальные алгоритмы для решения следующих задач:

- нахождение кратчайшего пути в графах и в двудольных графах, имеющих отрицательные веса ребер;

- нахождение минимальных остовных деревьев и 1-деревьев;

- нахождение решений задач о назначении всех типов;

- нахождение паросочетаний, b-сочетаний и f-факторов;

- нахождение решений многих других задач.

Библиотека также включает средства для решения некоторых NP-трудных задач, таких как симметрическая и асимметрическая задачи коммивояжера, задача о раскраске и пр.

Программа работает с помощью графического интерфейса, который позволяет пользователю рисовать различные графы с помощью инструментов «Вершина» и «Ребро», расположенных на панели инструментов. Предусмотрена возможность масштабирования изображения в окне, сохранения данных в файл и чтения данных из файла. Например, на рис. 3 приведено окно для рисования графа и задания весов его ребер.

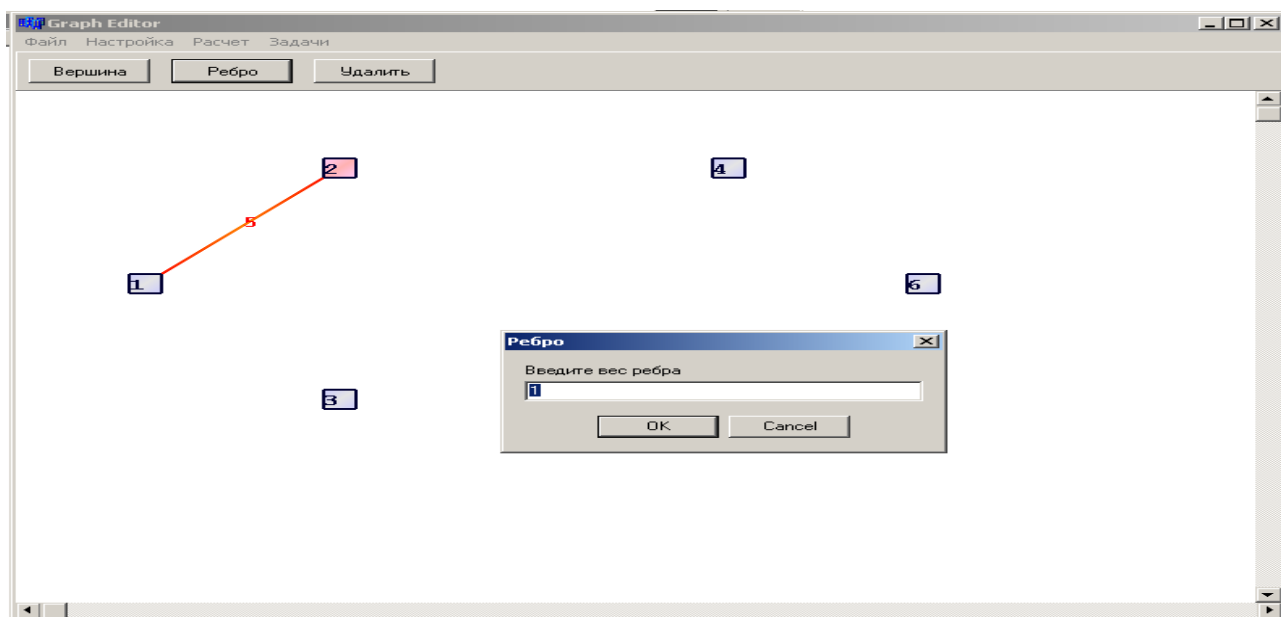


Рис.3. Окно для создания и редактирования графа.

К графической оболочке, позволяющей отображать графы, можно подключить любое количество модулей, решающих разнообразные задачи теории графов. Поэтому можно распределять работу над большим проектом между студентами.

В качестве пакета модулей можно рассчитать:

1. Поиск кратчайшего пути между двумя вершинами, выбранными пользователем (алгоритм Дейкстры). Пользователь выбирает концевые вершины пути с помощью щелчка мышью. Кратчайший путь в графе выделяется пунктиром (рис.4).
2. Поиск остовного дерева минимального веса (алгоритм Краскала). Как и в предыдущей задаче данный модуль предоставляет пользователю возможность выбора способа задания длины (веса) ребер.

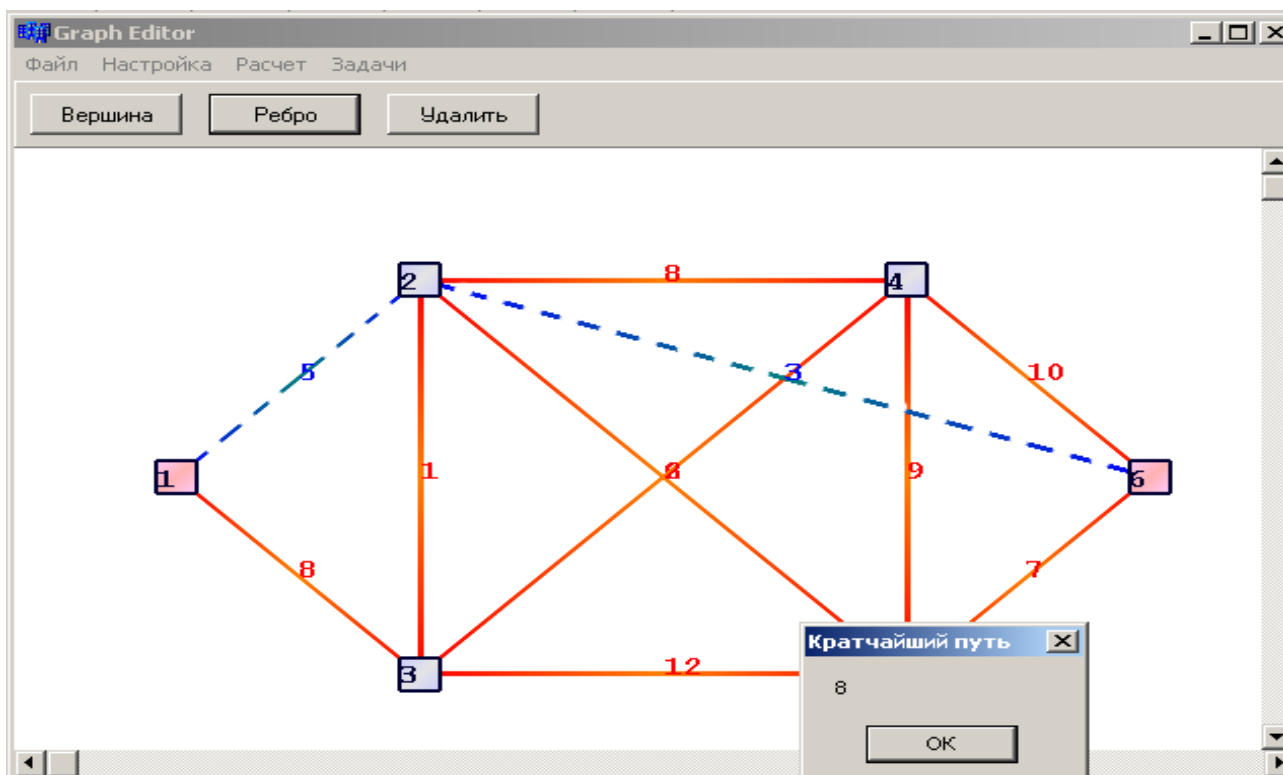


Рис.4. Решение задачи о кратчайшем пути

3. Подсчет числа компонент связности и сохранение матрицы инцидентности в блочном виде.
4. Поиск эйлеровых и гамильтоновых циклов и цепей. При запуске модуля проверяется возможность поиска решения. Если граф не является эйлеровым (гамильтоновым), пользователю выдается соответствующее сообщение.
5. Проверка заданного орграфа на наличие циклов. При наличии таковых вывести каждый цикл в виде последовательности вершин циклического пути.

Результаты работы модулей отображаются на рисунке, а также приводится ответ в текстовом формате: записывается в указанный пользователем файл и выводится в соответствующее окно для отображения ответа.

Дополнительные модули программы.

Кроме решения задач теории графов, программу можно ориентировать и на решение прикладных задач математики и экономики, например, следующие прикладные задачи.

Модуль для решения задачи сетевого планирования. Для графа, изображенного в области редактирования, в окне диалога задаются работы (для каждого созданного ребра пользователь может сопоставить с меткой ребра некоторое название работы) и время их выполнения. Пользователем выбираются источник (начальная вершина, $t=0$) и сток (конечная вершина, окончание проекта), а также задаются ориентации ребер. Модуль рассчитывает минимальное время выполнения проекта, находит всевозможные критические пути и резервы времени по дугам, не входящим в этот путь.

Модуль для решения задачи о назначениях. В качестве входных данных пользователю предлагается заполнить таблицу, соответствующую прибылям/убыткам, получаемыми организацией за занятость работника N на работе M. Число работников и работ совпадает. Пользователь самостоятельно выбирает, решается задача на минимум или на максимум. В качестве результата предоставляется список работников и назначенных им работ, а также общая сумма прибыли/убытков, полученных благодаря найденному распределению. Двудольный граф, полученный в процессе решения, отображается в окне редактирования (рис.5).

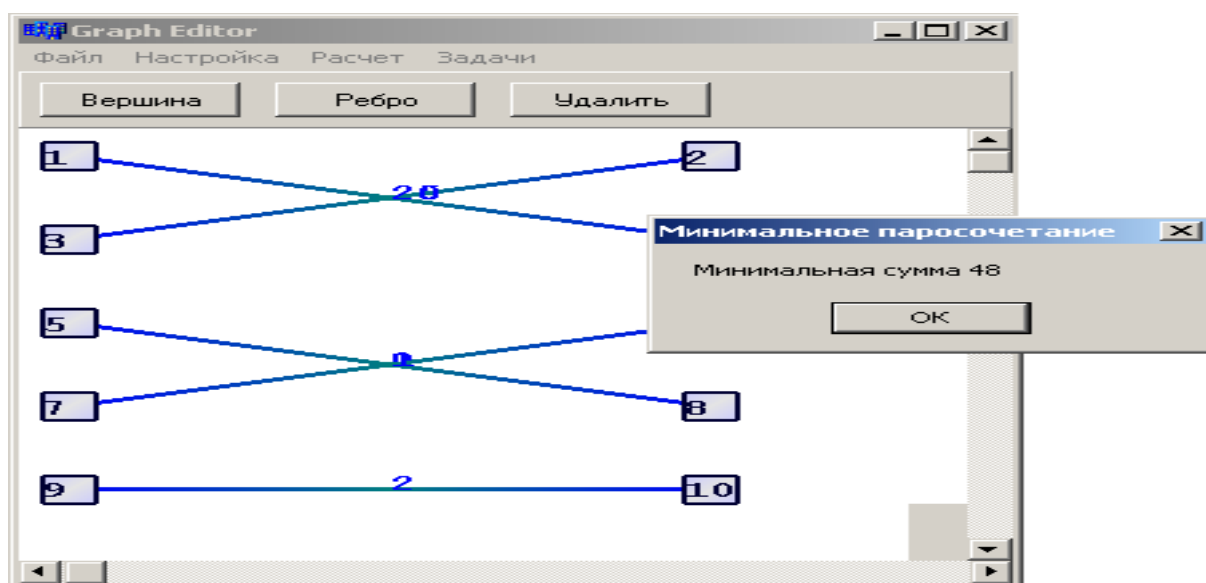


Рис.5. Решение задачи о назначениях

Модуль решения задачи о максимальном потоке. В диалоговом окне модуля для созданного графа задаются начальная и конечная вершины (источник и сток), а также пропускные способности всех ребер. Модуль определяет максимальную пропускную способность для графа с заданными параметрами.

«МАТРИЧНЫЕ СПОСОБЫ ПРЕДСТАВЛЕНИЯ ГРАФОВ»

Цель работы - изучение матричных способов представления графов.

Порядок выполнения работы

2.1. Получить задание у преподавателя в виде одного из двух способов матричного представления графа:

а) матрица смежности; б) матрица инцидентности

2.2. Составить алгоритм программы, реализующей перевод из заданного способа матричного представления графа в другой, учитывая при этом исходный тип графа (неориентированный, ориентированный, смешанный).

2.3. Создать программу, реализующую перевод из заданного способа матричного представления графа в другой. Предусмотреть консольный ввод исходных данных и вывод результатов работы программы на экран.

Теоретическая часть

В последнее время теория графов стала простым, доступным и мощным средством решения вопросов, относящихся к широкому кругу проблем. Это проблемы проектирования интегральных схем и схем управления, исследования автоматов, логических цепей, блок-схем программ, экономики и статистики, химии и биологии, теории расписаний и дискретной оптимизации.

Граф G задается множеством точек или вершин $x_1, x_2, \dots, x_n$ (которое обозначается через X) и множеством линий или ребер $a_1, a_2, \dots, a_n$ (которое обозначается символом A), соединяющих между собой все или часть этих точек. Таким образом, граф G полностью задается (и обозначается) парой (X, A) .

Если ребра из множества A ориентированы, что обычно показывается стрелкой, то они называются *дугами*, и граф с такими ребрами называется *ориентированным* графом (рисунок 6а). Если ребра не имеют ориентации, то граф называется *неориентированным* (рисунок 6б). В случае когда $G=(X, A)$ является ориентированным графом и мы хотим пренебречь направленностью дуг из множества A , то неориентированный граф, соответствующий G , будем обозначать как $G=(X, A)$.

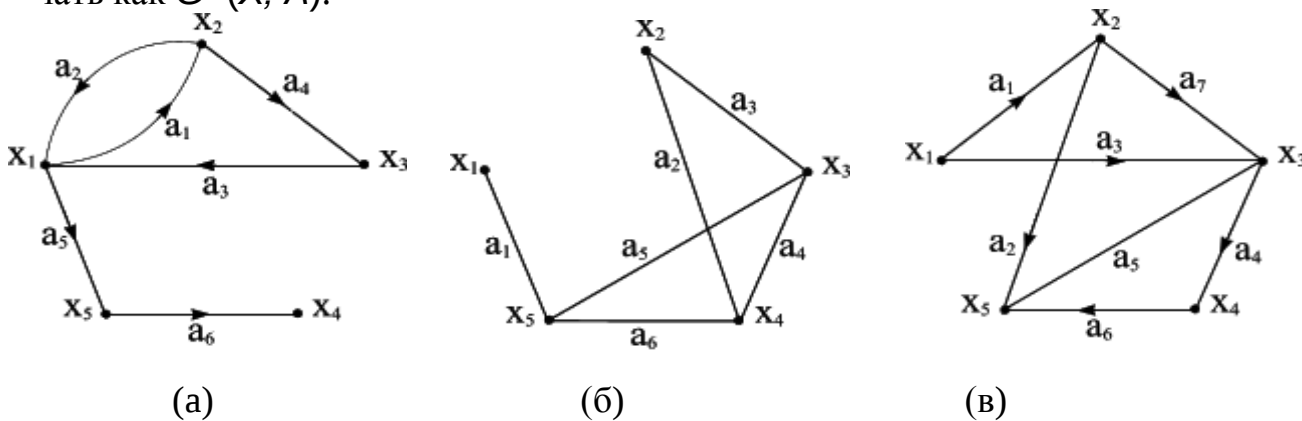


Рис. 6 (а) – ориентированный граф; (б) – неориентированный граф;
(в) – смешанный граф.

Если дуга обозначается упорядоченной парой, состоящей из *начальной* и *конечной* вершин (т. е. двумя *концевыми* вершинами дуги), ее направление предполагается заданным от первой вершины ко второй. Так, например, на рисунке обозначение (x_1, x_2) относится к дуге a_1 , а (x_2, x_1) - к дуге a_2 .

Другое, употребляемое чаще описание ориентированного графа G состоит в задании множества вершин X и *соответствия* Γ , которое показывает, как между собой связаны вершины. Соответствие Γ называется *отображением* множества X в X , а граф в этом случае обозначается парой $G=(X, \Gamma)$.

Для графа на рисунке имеем $\Gamma(x_1)=\{x_2, x_5\}$, т. е. вершины x_2 и x_5 являются конечными вершинами дуг, у которых начальной вершиной является x_1 .

$\Gamma(x_2)=\{x_1, x_3\}$, $\Gamma(x_3)=\{x_1\}$, $\Gamma(x_4)=\emptyset$ - пустое множество, $\Gamma(x_5)=\{x_4\}$.

В случае неориентированного графа или графа, содержащего и дуги, и неориентированные ребра (см., например, графы, изображенные на рисунках 6б и 6в, предполагается, что соответствие Γ задает такой эквивалентный ориентированный граф, который получается из исходного графа заменой каждого неориентированного ребра двумя противоположно направленными дугами, соединяющими те же самые вершины. Так, например, для графа, приведенного на рисунке, имеем $\Gamma(x_5)=\{x_1, x_3, x_4\}$, $\Gamma(x_1)=\{x_5\}$ и др.

Поскольку *прямое соответствие* или *образ* вершины $\Gamma(x_i)$ представляет собой множество таких вершин $x_j \in X$, для которых в графе G существует дуга (x_i, x_j) , то через $\Gamma^{-1}(x_i)$ естественно обозначить множество вершин x_k , для которых в G существует дуга (x_k, x_i) . Такое отношение принято называть *обратным соответствием* или *прообразом* вершины.

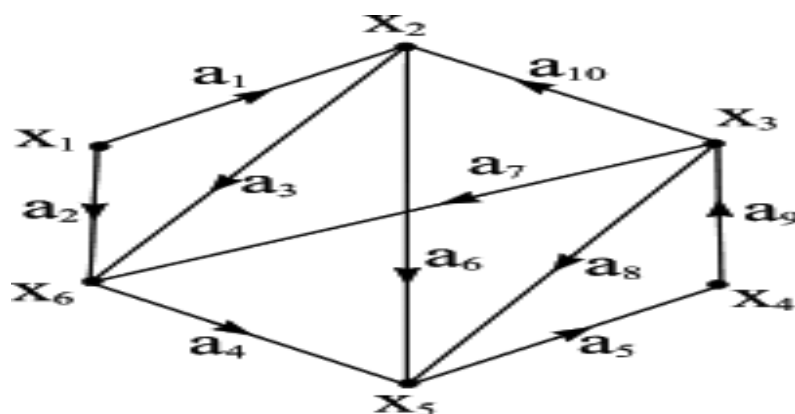


Рис. 7

Дуги $a=(x_i, x_j)$, $x_i \neq x_j$, имеющие общие концевые вершины, называются *смежными*. Две вершины x_i и x_j называются *смежными*, если какая-нибудь из двух дуг (x_i, x_j) и (x_j, x_i) или обе одновременно присутствуют в графе. Так, например, на рисунке 7 дуги a_1 , a_{10} , a_3 и a_6 как и вершины x_5 и x_3 , являются смежными, в то время как дуги a_1 и a_5 или вершины x_1 и x_4 не являются смежными.

Число дуг, которые имеют вершину x_i своей начальной вершиной, называется *полустепенью исхода* вершины x_i , и, аналогично, число дуг, которые имеют x_i своей конечной вершиной, называется *полустепенью захода* вершины x_i .

Таким образом, на рисунке 7 полустепень исхода вершины x_3 , обозначаемая через $\deg^+(x_3)$, равна $|\Gamma(x_3)|=3$, и полустепень захода вершины x_3 , обозначаемая через $\deg^-(x_3)$, равна $|\Gamma^{-1}(x_3)|=1$.

Очевидно, что сумма полустепеней захода всех вершин графа, а также сумма полустепеней исхода всех вершин равны общему числу дуг графа G , т. е.

$$\sum_{i=1}^n \deg^+(x_i) = \sum_{i=1}^n \deg^-(x_i) = m,$$

где n - число вершин и m - число дуг графа G .

Для неориентированного графа $G=(X,\Gamma)$ степень вершины x_i определяется аналогично - с помощью соотношения $\deg(x_i) \equiv |\Gamma(x_i)| = |\Gamma^{-1}(x_i)|$.

Петлей называется дуга, начальная и конечная вершины которой совпадают. На рисунке 8, например, дуги a_3 и a_{10} являются петлями.

Матричные представления

Матрица смежности

Пусть дан граф G , его матрица смежности обозначается через $A=[a_{ij}]$ и определяется следующим образом: $a_{ij}=1$, если в G существует дуга (x_i, x_j) , $a_{ij}=0$, если в G нет дуги (x_i, x_j) . Таким образом, матрица смежности графа A на рис. 8:

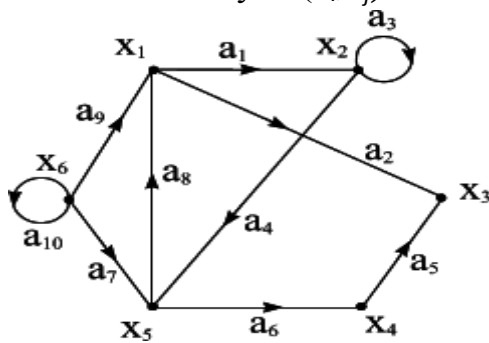


Рис. 8

$$A = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

Матрица смежности полностью определяет структуру графа. Например, сумма всех элементов строки x_i матрицы дает полустепень исхода вершины x_i , а сумма элементов столбца x_j - полустепень захода вершины x_j . Множество столбцов, имеющих 1 в строке x_i есть множество $\Gamma(x_i)$, а множество строк, которые имеют 1 в столбце x_j совпадает с множеством $\Gamma^{-1}(x_j)$.

Петли на графе представляют собой элементы, имеющие 1 на главной диагонали матрицы, например a_{22} , a_{66} для графа, изображенного на рисунке 8.

В случае неориентированного графа матрица смежности является симметричной относительно главной диагонали (рис. 9).

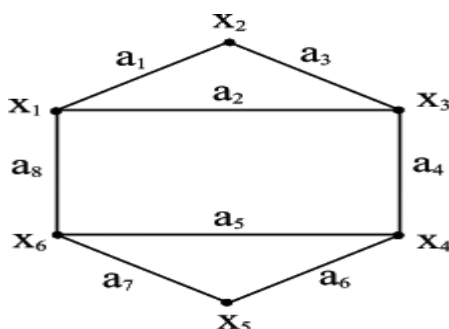


Рис. 9.

$$A = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

Матрица инцидентности

Пусть дан граф G с n вершинами и m дугами.

Матрица инцидентности графа G обозначается через $B=[b_{ij}]$ и является матрицей размерности $n \times m$, определяемой следующим образом:

$b_{ij}=1$, если x_i является начальной вершиной дуги a_j ;

$b_{ij}=-1$, если x_i является конечной вершиной дуги a_j ;

$b_{ij}=0$, если x_i не является концевой вершиной дуги a_j .

Для графа, приведенного на рисунке 8, матрица инцидентности имеет вид:

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
x_1	1	1	0	0	0	0	0	-1	-1	0
x_2	-1	0	± 1	1	0	0	0	0	0	0
x_3	0	-1	0	0	0	0	0	0	0	0
x_4	0	0	0	0	-1	-1	0	0	0	0
x_5	0	0	0	-1	1	1	-1	1	0	0
x_6	0	0	0	0	0	0	1	0	1	± 1

Поскольку каждая дуга инцидентна двум различным вершинам (за исключением случая, когда дуга образует петлю), то каждый столбец содержит один элемент, равный 1, и один - равный -1. Петля в матрице инцидентности не имеет адекватного математического представления (в программной реализации допустимо задание одного элемента $b_{ij}=1$).

Если G является неориентированным графом (рисунок 9), то его матрица инцидентности определяется следующим образом:

$b_{ij}=1$, если x_i является концевой вершиной дуги a_j ;

$b_{ij}=0$, если x_i не является концевой вершиной дуги a_j .

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
x_1	1	1	0	0	0	0	0	1
x_2	1	0	1	0	0	0	0	0
x_3	0	1	1	1	0	0	0	0
x_4	0	0	0	1	1	1	0	0
x_5	0	0	0	0	0	1	1	0
x_6	0	0	0	0	1	0	1	1

Матрица инцидентности, как способ задания графов, успешно применяется при описании мультиграфов (графов, в которых смежные вершины могут соединяться несколькими параллельными дугами).

Программа, описывающая обход графов в глубину.

```
Program graff;  
Var n, v, u: integer;  
fr: text;  
gr: array[1..30, 1..30] of integer;  
nov: array[1..15] of boolean;  
procedure dfs (v: integer);  
var u: integer;  
Begin  
  readln;  
  write (v, ' ');  
  nov [v]:=false;  
  for u:=1 to n do  
    if (gr[v, u]=1) and (nov[u]) then dfs (u);  
  end;  
begin  
  n:=3;  
  for v:=1 to n do  
    begin  
      nov [v]:=true;  
      writeln;  
      for u:=1 to n do begin  
        nov[u]:=true;  
        write (' gr[' , v, u, ']=');  
        read (gr [v, u] );  
      end;  
    end;  
    for v:=1 to n do begin  
      if nov[v] then dfs (v);  
    end;  
  readln;  
End.
```

Задания

1. Даны следующие матрицы. Установить, какие из них являются матрицами смежностей, какие - матрицами инцидентий и какие не являются ни теми, ни другими.

1)

0	0	1	0	1	0	1
0	0	0	1	0	1	1
1	0	0	1	1	0	0
0	1	1	0	0	1	1
1	0	1	0	0	0	1
0	1	0	1	0	0	1
1	1	0	1	1	1	0

2)

1	1	1	1	0	0
1	0	0	0	0	0
0	1	0	0	1	1
0	0	1	0	1	0
0	0	0	1	0	1

3)

0	1	0	1	0	1	0	1
1	0	1	0	1	0	0	1
1	0	0	1	1	1	0	0
0	0	1	0	1	0	1	0
0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	1
1	1	1	1	0	0	0	1
1	1	1	1	0	0	0	0

4)

1	1	1	1	1	0	0
0	0	0	1	0	1	0
0	1	0	0	1	1	1
0	0	1	0	0	0	1
1	0	0	0	0	0	0

5)

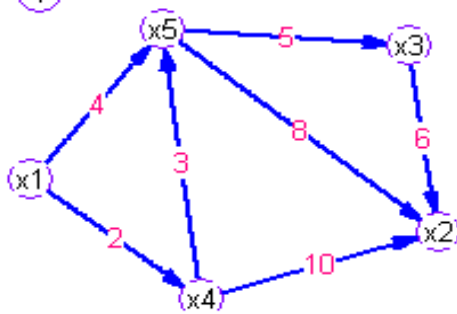
1	0	0	1	0	1	0
1	1	1	0	1	0	1
0	0	0	0	0	0	0
1	0	1	0	1	0	0

6)

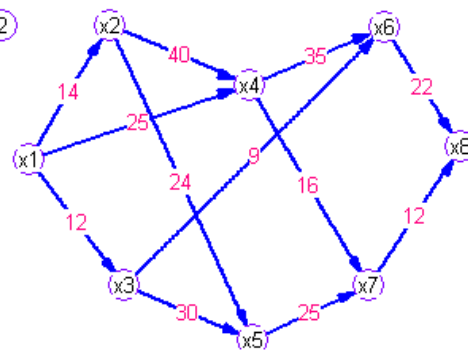
1	1	1	1	1	1
1	0	1	0	1	0
0	1	0	0	0	1
0	0	0	1	0	0
0	0	0	0	0	0

2. Для данных графов составить матрицы смежности и инцидентности.

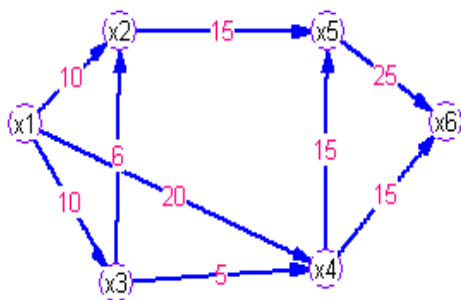
1)



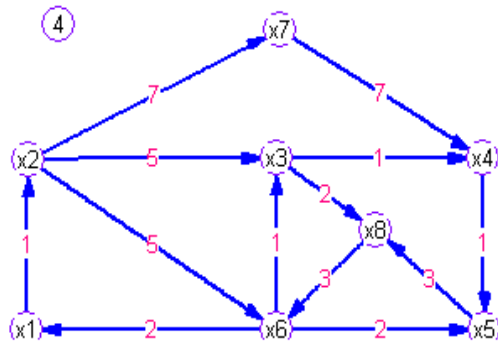
2)



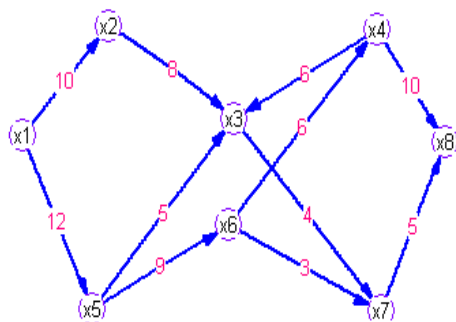
3)



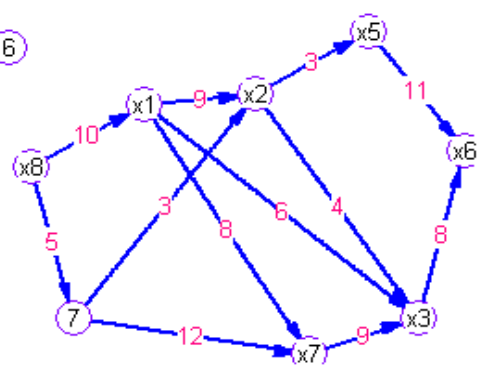
4)

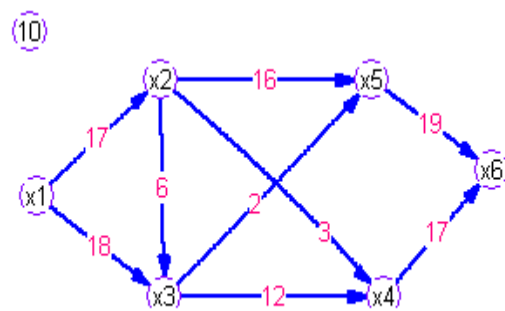
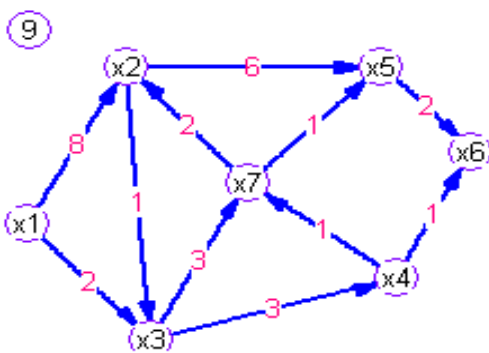
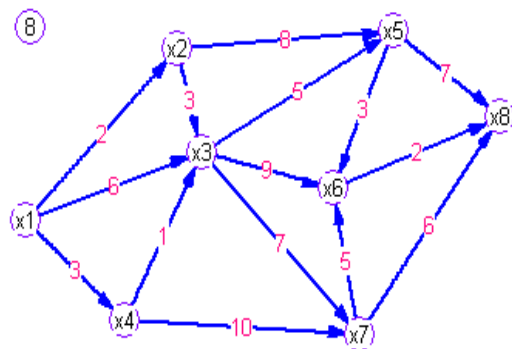
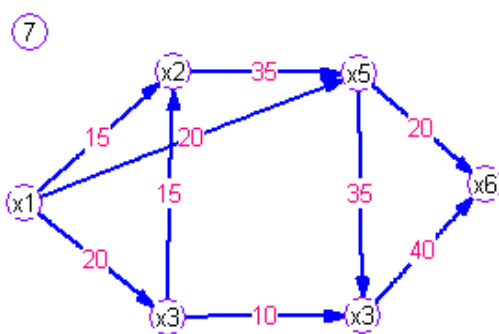


5)



6)





Контрольные вопросы

1. Перечислите основные способы представления графов.
2. Покажите на примере прямое и обратное соответствие для заданной вершины.
3. Чему равна сумма степеней всех вершин неориентированного графа?
4. В чем отличия матричного представления ориентированных и неориентированных графов?
5. В чем особенности представления графа матрицей смежности?
6. В чем особенности представления графа матрицей инцидентности?

ЛАБОРАТОРНАЯ РАБОТА №5. «ПОИСК КРАТЧАЙШИХ ПУТЕЙ НА ГРАФАХ»

Цель работы - изучить алгоритмы поиска кратчайших путей на графах: алгоритм Дейкстры, алгоритм Форда – Беллмана, написать программу, используя изученные алгоритмы.

Порядок выполнения работы

1. В соответствии с вариантом задания, определенным преподавателем, провести расчеты.
2. Оформить отчет о выполнении задания с приведением условия задачи, формул для расчета, результатов решения и заключения.

Теоретические сведения.

Задача о нахождении кратчайшего пути состоит в нахождении кратчайшего пути от заданной начальной вершины до заданной конечной вершины, при условии, что такой путь существует.

Можно дать много практических интерпретаций задачи о кратчайших путях. Например, вершины могут соответствовать городам и каждая дуга - некоторому пути, длина которого представлена весом дуги, тогда необходимо найти кратчайшие пути между городами. Вес дуги может соответствовать стоимости (или времени) передачи информации между вершинами. В этом случае ищут самый дешевый (или самый скорый).

Пример. Головоломка о трех бидонах.

Восьмилитровый бидон заполнен некой жидкостью, а два бидона емкостью 5 и 3 литра пусты. Необходимо разделить 8 литров жидкости на две равные части, используя только три имеющихся бидона. Какое минимальное количество переливаний из бидона в бидон надо сделать, чтобы достичь желаемого результата?

В этой сетевой модели каждый узел будет соответствовать объемам жидкости в 8-, 5- и 3-литровом бидонах. Начальным узлом будет $(8, 0, 0)$, а конечным - $(4, 4, 0)$. Новый узел получается из текущего при однократном переливании жидкости из одного бидона в другой. На рис. 10 показаны различные маршруты, ведущие от начального узла $(8, 0, 0)$ к конечному $(4, 4, 0)$. Таким образом, наша головоломка сведена к задаче определения кратчайшего пути между узлами $(8, 0, 0)$ и $(4, 4, 0)$.

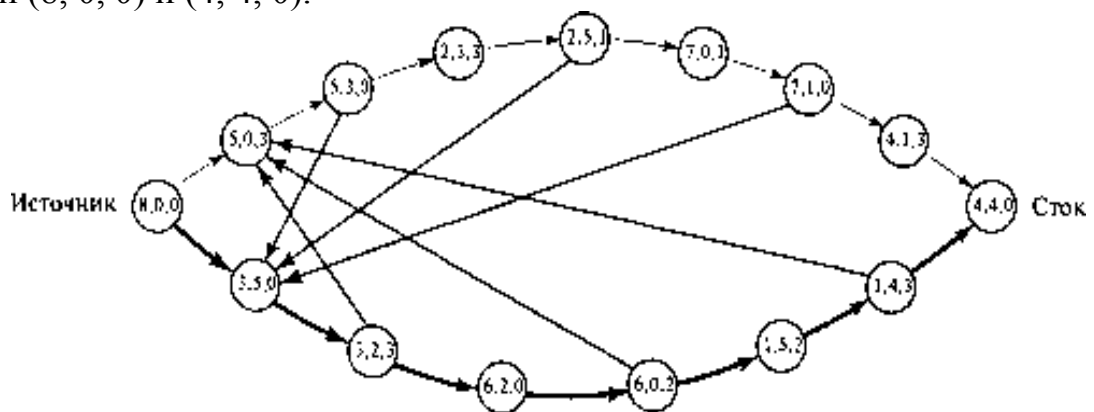


Рис. 10. задача вычисления кратчайшего пути.

Оптимальное решение, показанное в нижней части рис. 10, требует семи переливаний из бидона в бидон.

Описание алгоритмов нахождения кратчайшего пути

Алгоритм Дейкстры нахождения минимального пути

Для нахождения минимального пути между двумя произвольными вершинами для случая, когда все $c_{ij} > 0$ можно воспользоваться простым алгоритмом Дейкстры

Шаг 1. Установка начальных условий.

Ввести число вершин графа n и матрицу весов $C = (c_{ij})$. Ввести номер начальной вершины s и номер конечной вершины f . Для начальной вершины s положить $M(s) = 0$ и считать эту пометку постоянной. Положить $M(x_i)$ для всех остальных вершин $x_i \neq s$ равными достаточно большому числу (например, 99999999) и считать эти пометки временными. Положить номер текущей вершины $p = s$. Сформировать множество Π вершин с постоянной пометкой, вначале $\Pi = \{p\}$, и множество V вершин с временной пометкой, $V = \{X \setminus \Pi\}$.

Шаг 2. Обновление пометок. Заполнить множество вершин $G(p)$, являющихся образом вершины p . Для всех вершин $x_i \in G(p)$, имеющих временные пометки, т.е. для $x_i \in G(p) \cap V$, изменить пометки следующим образом:

$$M(x_i) - \min (M(x_i), M(p)+c(p,x_i)).$$

Шаг 3. Превращение временных пометок в постоянные. Из всех вершин, имеющих временные пометки, найти такую, для которой $M(x^*) = \min(M(x_i))$. Считать пометку вершины x^* постоянной. Положить $p = x^*$.

Шаг 4. Проверка, является ли найденный путь минимальным. Если $p = f$, то $M(p)$ является длиной минимального пути; перейти к шагу 6. Иначе перейти к шагу 5.

Шаг 5. Изменение множеств Π и V . Удалить номер вершины p из множества V и добавить номер вершины p в множество Π . Перейти к шагу 2.

Шаг 6. Восстановление минимального пути. Для любой вершины x_i , предшествующая ей вершина x_j определяется из соотношения:

$$M(x_j)+c(x_j,x_i)=M(x_i), \quad x_j \in G^{-1}(x_i) \cap \Pi, \text{ где } G^{-1}(x_i) - \text{прообраз вершины } x_i.$$

Последовательно применяя это соотношение, начиная от последней вершины f , найдем минимальный путь.

Пример нахождения кратчайшего пути с помощью алгоритма Дейкстры.

Определим минимальный путь из вершины x_1 а вершину x_6 в нагруженном графе, изображенном на рис. 11.

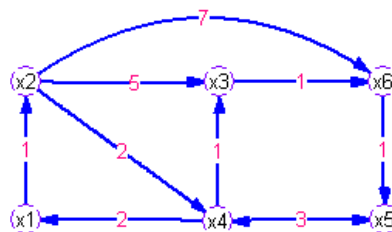


Рис. 11

Рассмотрим подробно работу алгоритма Дейкстры для этого примера.

Шаг 1. Установка начальных условий. Матрица весов имеет вид:

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

$$s=x_1; M(s)=0; p= x_1; \Pi=\{ x_1 \}; B=\{ x_2, x_3, x_4, x_5, x_6 \}.$$

1-ая итерация.

Шаг 2. $G(x_1)=\{x_2\}$; $G(x_1) \cap B=\{x_2\}$; $M(x_2)=\min\{\infty, 0+1\}=1$.

Шаг 3. $x^*=x_2$; $p=x_2$.

Шаг 4. $x_2 \neq x_6$

Шаг 5. $\Pi=\{x_1, x_2\}$; $B=\{x_3, x_4, x_5, x_6\}$.

2-ая итерация.

Шаг 2. $G(x_2)=\{x_3, x_4, x_6\}$; $G(x_2) \cap B=\{x_3, x_4, x_6\}$; $M(x_3)=\min\{\infty, 1+5\}=6$;

$M(x_4)=\min\{\infty, 1+2\}=3$; $M(x_6)=\min\{\infty, 1+7\}=8$.

Шаг 3. $x^*=x_4$; $p=x_4$.

Шаг 4. $x_2 \neq x_6$

Шаг 5. $\Pi=\{x_1, x_2, x_4\}$; $B=\{x_3, x_5, x_6\}$.

3-ая итерация.

Шаг 2. $G(x_4)=\{x_1, x_3, x_5\}$; $G(x_4) \cap B=\{x_3, x_5\}$;

$$M(x_3)=\min\{6, 3+1\}=4; M(x_5)=\min\{\infty, 3+4\}=7.$$

Шаг 3. $x^*=x_3$; $p=x_3$.

Шаг 4. $x_3 \neq x_6$

Шаг 5. $\Pi=\{x_1, x_2, x_4, x_3\}$; $B=\{x_5, x_6\}$. 4-ая итерация.

Шаг 2. $G(x_3)=\{x_6\}$; $G(x_3) \cap B=\{x_6\}$; $M(x_6)=\min\{8, 4+1\}=5$.

Шаг 3. $x^*=x_6$; $p=x_6$.

Шаг 4. $x_6 = x_6$.

Переходим к шагу 6.

Прежде, чем перейти к восстановлению минимального пути, представим полученные результаты в виде таблицы, которой удобно пользоваться для расчетов (табл.1.)

Таблица 1.

Восстановление минимального пути.

№ итерации	p	Π	M(Π)	B	M(B) старые	M(B) обновленные
0	1	1	0	2	∞	1
				3	∞	∞
				4	∞	∞
				5	∞	∞
				6	∞	∞
1	2	1	0	3	∞	6
		2	1	4	∞	3
				5	∞	∞
				6	∞	8

2	4	1	0	3	6	4
		2	1	5	∞	7
		4	3	6	8	8
3	3	1	9	5	7	7
		2	1	6	8	5
		4	3			
		3	4			
4	6					

Шаг 6. Определим вершину, предшествующую вершине x_6

$$G^{-1}(x_6) = \{x_2, x_3\};$$

$$G^{-1}(x_6) \cap \Pi = \{x_2, x_3\};$$

$$M(x_2) + c(x_2, x_6) = 1 + 7 \neq M(x_6),$$

$$M(x_3) + c(x_3, x_6) = 4 + 1 = 5 = M(x_6),$$

т.е. вершина x_3 предшествует вершине x_6 .

Определим вершину, предшествующую вершине x_3

$$G^{-1}(x_3) = \{x_2, x_4\};$$

$$G^{-1}(x_3) \cap \Pi = \{x_2, x_4\};$$

$$M(x_2) + c(x_2, x_3) = 1 + 5 \neq M(x_3),$$

$$M(x_4) + c(x_4, x_3) = 3 + 1 = 4 = M(x_3),$$

т.е. вершина x_4 предшествует вершине x_3 .

Определим вершину, предшествующую вершине x_6

$$G^{-1}(x_4) = \{x_2\};$$

$$G^{-1}(x_4) \cap \Pi = \{x_2\};$$

$$M(x_2) + c(x_2, x_4) = 1 + 2 = 3 = M(x_4),$$

т.е. вершина x_2 предшествует вершине x_4 .

Определим вершину, предшествующую вершине x_2

$$G^{-1}(x_2) = \{x_1\};$$

$$G^{-1}(x_2) \cap \Pi = \{x_1\};$$

$$M(x_1) + c(x_1, x_2) = 0 + 1 = 1 = M(x_2),$$

т.е. вершина x_1 предшествует вершине x_2 .

Итак, $x_1 \rightarrow x_2 \rightarrow x_4 \rightarrow x_3 \rightarrow x_6$ есть минимальный путь из вершины x_1 в вершину x_6 . Его длина равна 5.

Алгоритм нахождения минимального пути Форда-Беллмана

Предполагается, что ориентированный граф не содержит контуров отрицательной длины. Основными вычисляемыми величинами этого алгоритма являются величины $\lambda_i(k)$, где $i = 1, 2, \dots, n$ (n - число вершин графа); $k = 1, 2, \dots, n - 1$. Для фиксированных i и k величина $\lambda_i(k)$ равна длине минимального пути, ведущего из заданной начальной вершины x_1 в вершину x_i и содержащего не более k дуг.

Шаг 1. Установка начальных условий. Ввести число вершин графа n и матрицу весов $C = (c_{ij})$.

Шаг 2. Положить $k = 0$. Положить $\lambda_i(0) = \infty$ для всех вершин, кроме x_1 , положить $\lambda_1(0) = 0$.

Шаг 3. В цикле по $k, k = 1, \dots, n - 1$, каждой вершине x_i на k -ом шаге приписать индекс $\lambda_i(k)$ по следующему правилу:

$$\lambda_i(k) = \min_{1 \leq j \leq n} \{ \lambda_j(k-1) + c_{ij} \},$$

для всех вершин, кроме x_1 , положить $\lambda_1(k) = 0$.

В результате работы алгоритма формируется таблица индексов $\lambda_i(k)$, $i=1, 2, \dots, n$, $k = 0, 1, 2, \dots, n - 1$. При этом $\lambda_i(k)$ определяет длину минимального пути из первой вершины в i -ую, содержащего не более k дуг.

Шаг 4. Восстановление минимального пути. Для любой вершины x_s предшествующая ей вершина x_r определяется из соотношения:

$$\lambda_r(n-2) + c_{rs} = \lambda_s(n-1), \quad x_r \in G^{-1}(x_s), \text{ где } G^{-1}(x_s) - \text{прообраз вершины } x_s.$$

Для найденной вершины x_r предшествующая ей вершина x_q определяется из соотношения: $\lambda_q(n-3) + c_{qr} = \lambda_r(n-2)$, $x_q \in G^{-1}(x_r)$, где $G^{-1}(x_r)$ - прообраз вершины x_r и т. д.

Последовательно применяя это соотношение, начиная от последней вершины x_i , найдем минимальный путь.

Пример. С помощью алгоритма Форда - Беллмана найдем минимальный путь из вершины x_1 в вершину x_3 в графе, изображенном на рис. 12.

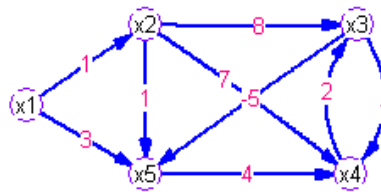


Рис. 12

Значения индексов $\lambda_i(k)$ будем заносить в таблицу индексов (табл. 2).

Шаг 1. Введем число вершин графа $n=5$. Матрица весов этого графа имеет вид:

$$C = \begin{pmatrix} \infty & 1 & \infty & \infty & 3 \\ \infty & \infty & 8 & 7 & 1 \\ \infty & \infty & \infty & 1 & -5 \\ \infty & \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & 4 & \infty \end{pmatrix}$$

Шаг 2. Положим $k=0$, $\lambda_1(0)=\lambda_2(0)=\lambda_3(0)=\lambda_4(0)=\lambda_5(0)=\infty$. Эти значения занесем в первый столбец табл. 2.

Шаг 3. $k=1$, $\lambda_1(1) = 0$.

Равенство (1) для $k=1$ имеет вид:

$$\lambda_i(1) = \min_{1 \leq j \leq 5} \{ \lambda_j(0) + c_{ji} \}$$

$$\begin{aligned} \lambda_2(1) &= \min \{ \lambda_1(0) + c_{12}; \lambda_2(0) + c_{22}; \lambda_3(0) + c_{32}; \lambda_4(0) + c_{42}; \lambda_5(0) + c_{52} \} = \\ &= \min \{ 0 + 1; \infty + \infty; \infty + \infty; \infty + \infty; \infty + \infty \} = 1 \end{aligned}$$

$$\begin{aligned} \lambda_3(1) &= \min \{ \lambda_1(0) + c_{13}; \lambda_2(0) + c_{23}; \lambda_3(0) + c_{33}; \lambda_4(0) + c_{43}; \lambda_5(0) + c_{53} \} = \\ &= \min \{ 0 + \infty; \infty + 8; \infty + \infty; \infty + 2; \infty + \infty \} = \infty \end{aligned}$$

$$\begin{aligned} \lambda_4(1) &= \min \{ \lambda_1(0) + c_{14}; \lambda_2(0) + c_{24}; \lambda_3(0) + c_{34}; \lambda_4(0) + c_{44}; \lambda_5(0) + c_{54} \} = \\ &= \min \{ 0 + \infty; \infty + 7; \infty + 1; \infty + \infty; \infty + 4 \} = \infty \end{aligned}$$

$$\lambda_5(1) = \min\{\lambda_1(0) + c_{15}; \lambda_2(0) + c_{25}; \lambda_3(0) + c_{35}; \lambda_4(0) + c_{45}; \lambda_5(0) + c_{55}\} = \\ = \min\{0 + 3; \infty + 1; \infty - 5; \infty + \infty; \infty + \infty\} = 3$$

Полученные значения $\lambda_i(1)$ занесем во второй столбец табл. 2. Убеждаемся, что второй столбец, начиная со второго элемента, совпадает с первой строкой матрицы весов, что легко объясняется смыслом величин $\lambda_i(1)$, которые равны длине минимального пути из первой вершины в i -ую, содержащего не более одной дуги,

$$k = 2.$$

$$\lambda_1(2) = 0.$$

Равенство (1) для $k = 2$ имеет вид:

$$\lambda_i(2) = \min_{1 \leq j \leq 5} \{\lambda_j(1) + c_{ji}\}$$

$$\lambda_2(2) = \min\{0 + 1; 1 + \infty; \infty + \infty; \infty + \infty; 3 + \infty\} = 1.$$

$$\lambda_3(2) = \min\{0 + \infty; 1 + 8; \infty + \infty; \infty + 2; 3 + \infty\} = 9.$$

$$\lambda_4(2) = \min\{0 + \infty; 1 + 7; \infty + 1; \infty + \infty; 3 + 4\} = 7.$$

$$\lambda_5(2) = \min\{0 + 3; 1 + 1; \infty - 5; \infty + \infty; 3 + \infty\} = 2.$$

Полученные значения $\lambda_i(2)$ занесем в третий столбец табл. 2. Величины $\lambda_i(2)$ равны длине минимального пути из первой вершины в i -ую, содержащего не более двух дуг.

$$k = 3.$$

$$\lambda_1(3) = 0.$$

Равенство (1) для $k = 3$ имеет вид:

$$\lambda_i(3) = \min_{1 \leq j \leq 5} \{\lambda_j(2) + c_{ji}\}$$

$$\lambda_2(3) = \min\{0 + 1; 1 + \infty; 9 + \infty; 7 + \infty; 2 + \infty\} = 1.$$

$$\lambda_3(3) = \min\{0 + \infty; 1 + 8; 9 + \infty; 7 + 2; 2 + \infty\} = 9.$$

$$\lambda_4(3) = \min\{0 + \infty; 1 + 7; 9 + 1; 7 + \infty; 2 + 4\} = 6.$$

$$\lambda_5(3) = \min\{0 + 3; 1 + 1; 9 - 5; 7 + \infty; 2 + \infty\} = 2.$$

Полученные значения $\lambda_i(3)$ занесем в четвертый столбец табл. 2. Величины $\lambda_i(3)$ равны длине минимального пути из первой вершины в i -ую, содержащего не более трех дуг.

$$k = 4.$$

$$\lambda_1(4) = 0.$$

Равенство (1) для $k = 4$ имеет вид:

$$\lambda_i(4) = \min_{1 \leq j \leq 5} \{\lambda_j(3) + c_{ji}\}$$

$$\lambda_2(4) = \min\{0 + 1; 1 + \infty; 9 + \infty; 6 + \infty; 2 + \infty\} = 1.$$

$$\lambda_3(4) = \min\{0 + \infty; 1 + 8; 9 + \infty; 6 + 2; 2 + \infty\} = 8.$$

$$\lambda_4(4) = \min\{0 + \infty; 1 + 7; 9 + 1; 6 + \infty; 2 + 4\} = 6.$$

$$\lambda_5(4) = \min\{0 + 3; 1 + 1; 9 - 5; 6 + \infty; 2 + \infty\} = 2.$$

Полученные значения $\lambda_i(4)$ занесем в пятый столбец табл. 3. Величины $\lambda_i(4)$ равны длине минимального пути из первой вершины в i -ую, содержащего не более четырех дуг.

Таблица 2.

Определение длины минимального пути.

i (номер вершины)	$\lambda_i(0)$	$\lambda_i(1)$	$\lambda_i(2)$	$\lambda_i(3)$	$\lambda_i(4)$
1	0	0	0	0	0
2	∞	1	1	1	1
3	∞	∞	9	9	8
4	∞	∞	7	6	6
5	∞	3	2	2	2

Шаг 4. Восстановление минимального пути.

Для последней вершины x_3 предшествующая ей вершина x_r определяется из соотношения (2) полученного при $s=3$: $\lambda_r + c_{r3} = \lambda_3(4)$, $x_r \in G^{-1}(x_3)$,

где $G^{-1}(x_3)$ – прообраз вершины x_3

$$G^{-1}(x_3) = \{x_2, x_4\}.$$

Подставим последовательно $r = 2$ и $r = 4$, чтобы определить, для какого r это равенство выполняется:

$$\lambda_2(3) + c_{23} = 1 + 8 \neq \lambda_3(4) = 8,$$

$$\lambda_4(3) + c_{43} = 6 + 2 = \lambda_3(4) = 8$$

Таким образом, вершиной, предшествующей вершине x_3 , является вершина x_4 .

Для вершины x_4 предшествующая ей вершина x_r определяется из соотношения (2) полученного при $s=4$: $\lambda_r(2) + c_{r4} = \lambda_4(3)$, $x_r \in G^{-1}(x_4)$,

где $G^{-1}(x_4)$ - прообраз вершины x_4 .

$$G^{-1}(x_4) = \{x_2, x_3, x_5\}.$$

Подставим в полученное соотношение последовательно $r = 2$, $r = 3$ и $r = 5$, чтобы определить, для какого r это равенство выполняется:

$$\lambda_2(2) + c_{24} = 1 + 7 \neq \lambda_4(3) = 6;$$

$$\lambda_3(2) + c_{34} = 1 + 1 \neq \lambda_4(3) = 6;$$

$$\lambda_5(2) + c_{54} = 2 + 4 \neq \lambda_4(3) = 6.$$

Таким образом, вершиной, предшествующей вершине x_4 , является вершина x_5 .

Для вершины x_5 предшествующая ей вершина x_r определяется из соотношения полученного при $s=5$: $\lambda_r(1) + c_{r5} = \lambda_5(2)$, $x_r \in G^{-1}(x_5)$,

где $G^{-1}(x_5)$ - прообраз вершины x_5 .

$$G^{-1}(x_5) = \{x_1, x_2, x_3\}.$$

Подставим последовательно $r=1$ и $r = 2$, чтобы определить, для какого r это равенство выполняется:

$$\lambda_1(1) + c_{15} = 0 + 3 \neq \lambda_5(2) = 2;$$

$$\lambda_2(1) + c_{25} = 1 + 1 \neq \lambda_5(2) = 2.$$

Таким образом, вершиной, предшествующей вершине x_5 , является вершина x_2 .

Для вершины x_2 предшествующая ей вершина x_r определяется из соотношения (2) полученного при $s=2$: $\lambda_r(0) + c_{r2} = \lambda_2(1)$, $x_r \in G^{-1}(x_2)$,

где $G^{-1}(x_2)$ - прообраз вершины x_2 .

$$G^{-1}(x_2) = \{x_1\}.$$

Подставим в (6) $r = 1$, чтобы определить, выполняется ли это равенство:

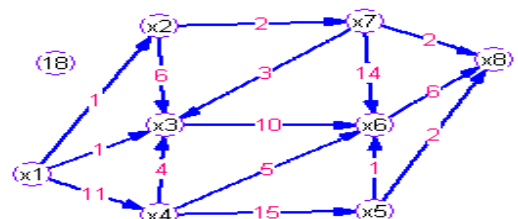
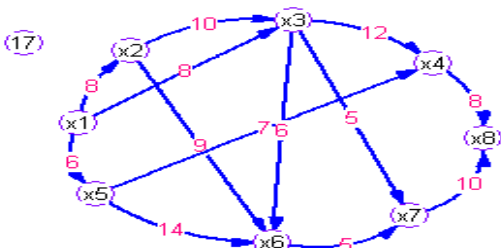
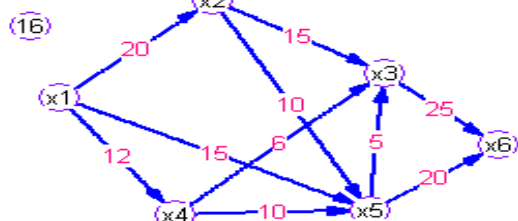
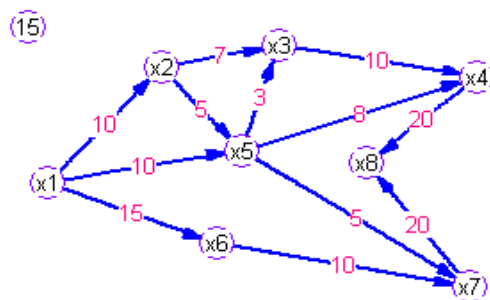
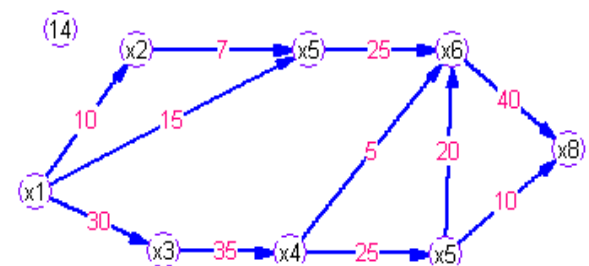
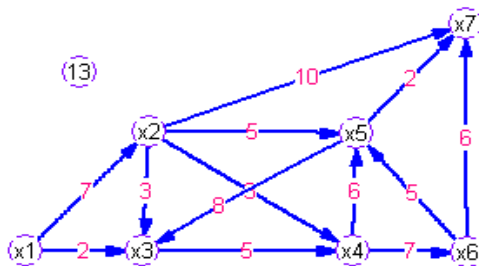
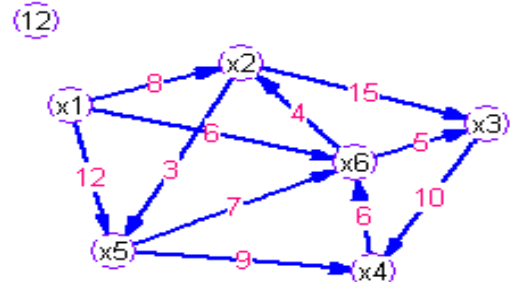
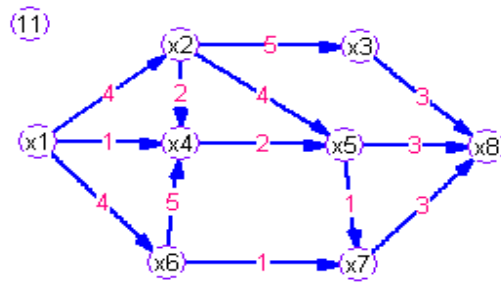
$$\lambda_1(0) + c_{12} = 0 + 1 = \lambda_2(1) = 1.$$

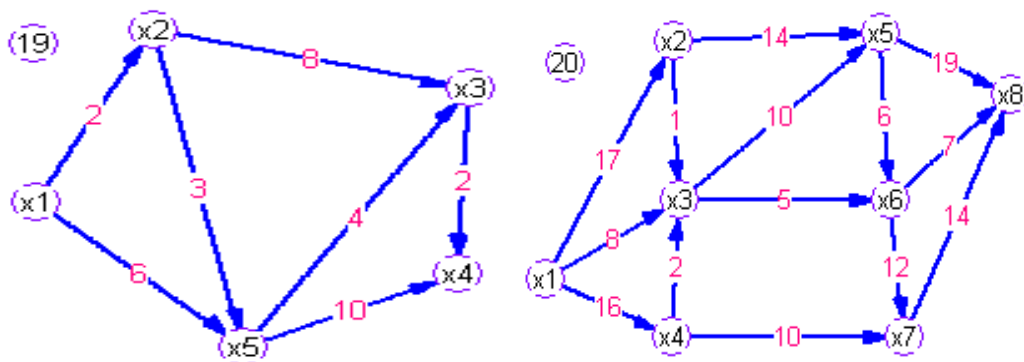
Таким образом, вершиной, предшествующей вершине x_2 , является вершина x_1 .

Итак, найден минимальный путь – x_1, x_2, x_5, x_4, x_3 , его длина равна 8.

Варианты заданий

1. Написать программу, реализующую алгоритм Дейкстры.
2. Написать программу, реализующую алгоритм Форда-Беллмана.
3. Используя алгоритм Дейкстры и алгоритм Форда-Беллмана, найти кратчайшие пути в графах, изображенных на рисунках:





Контрольные вопросы

1. Дайте определение пути, маршрута, цепи, контура.
2. Какой граф называется взвешенным?
3. Как определяется длина пути графа?
4. Задача нахождения кратчайшего пути на графе.
5. Суть алгоритма Дейкстры?
6. Суть алгоритма Форда – Беллмана?

ЛАБОРАТОРНАЯ РАБОТА №6 «ПОСТРОЕНИЕ КРАТЧАЙШИХ ОСТОВЫХ ДЕРЕВЬЕВ ГРАФА»

Цель работы - изучение метода построения кратчайших остовых деревьев графа на примере алгоритма Прима-Краскала, написать программу, используя изученный алгоритм

Порядок выполнения работы

1. В соответствии с вариантом задания, определенным преподавателем, провести расчеты.
2. Оформить отчет о выполнении задания с приведением условия задачи, формул для расчета, результатов решения и заключения.

Теоретическая часть

Существует простой и важный тип графов, которому разные авторы дали одинаковое название, - это деревья. Часто при решении какой-либо задачи о графах её сначала исследуют на деревьях.

Неориентированное дерево (или просто дерево) - это связный ациклический граф (неорграф называется связанным или каждая пара различных вершин может быть соединена по крайней мере одной цепью).

Граф, не содержащий циклов, называется *лесом* (ясно, что компонентами леса являются деревья). Если зафиксирована некоторая вершина A_6 , то она на-

зывается корнем дерева, а само дерево называется деревом с корнем. Известны и другие определения понятия "дерево", которые неявно содержатся в следующем теореме.

Для графа G , имеющего p вершин и q ребер, следующие утверждения эквивалентны:

1. Граф G - дерево
2. Любые две вершины в графе G соединены единственной цепью;
3. Граф G - связный граф и $p=q+1$;
4. Граф G - ациклический граф и $p=q+1$;
5. Граф G - ациклический граф и, если любую пару несмежных вершин соединить ребром X , то в графе $G+X$ будет точно один цикл.

Очевидны 2 следствия данной теории:

1. в любом дереве имеется, по крайней мере, две висячие вершины;
2. каждой дереву с n - вершинами имеет ровно $n-1$ ребром.

Неорграф, получающийся в результате "снятия ориентации с дуг" орграфа G , называется *лежащим в основе неорграфа G* . Орграф G называется ориентированным (корневым) деревом, если он является деревом и имеет корень.

Поддеревом корневого дерева T называется его часть T , которая является корневым деревом, удовлетворяющим следующему условию: вместе с корнем V дерево T содержит все вершины и дуги достижимые в T из V .

Вершины графа G с нулевой полустепенью исхода (иными словами, не имеющий приемников) называется листьями. Корневое дерево, как следует из определения, удовлетворяет следующим условиям:

- а) имеется ровно одна вершина, называемая корнем, в которую не входит ни одна дуга.
- б) в каждую вершину, отличную от корня, входит ровно одна дуга
- в) приемники всякой вершины упорядочены.

Поэтому имеет смысл следующее определение.

Ориентированный конечный граф (V, E) есть прадревос с корнем x_1 принадлежит V (корневое дерево с корнем x_1 принадлежит V) если:

1. в каждую вершину, отличную от вершины x_1 , заходит единственная дуга.
2. в вершину x_1 не заходит ни одна дуга.
3. граф (V, E) не содержит контуров.

Т.е. *прадревос* - это дерево, каждое ребро которого однозначно ориентировано, и для любой вершины X , которого существует путь от x_1 к X).

С точки зрения представления в памяти важно различать 2 типа деревьев: бинарные и сильно ветвящиеся. *Бинарные (двоичное) дерево* - это корневое дерево, в котором каждая вершина имеет не более двух приемников, называемых левыми и правыми приемниками (сыновьями). В сильно ветвящемся дереве количество приемников может быть произвольно. Бинарные деревья классифицируются по нескольким признакам. Степенью узла в дереве называется количество дуг, которая из него выходит. Степенью дерева равно максимальной степени узла, входящего в дерево.

Ясно, что степень узла бинарного дерева не превышает числа 2. При этом

листьями в дереве являются вершины, имеющие степень нуль.

Другим важным признаком структурной классификации бинарных деревьев является строгость бинарного дерева.

Строго бинарное дерево состоит из узлов, имеющих степень 2 или степень 0. Нестрого бинарное дерево содержит узлы со степенью равной 1.

Остовным деревом связного графа G называется любой его подграф, содержащий все вершины графа G и являющийся деревом.

Пусть граф G имеет n вершин и m ребёр. Так как всякое дерево с n вершинами по определению имеет $n-1$ ребёр, то любое остовное дерево графа G получается из этого графа в результате удаления $m-(n-1)=m-n+1$ ребёр. Число $\gamma = m-n+1$ называется цикломатическим числом графа. Найдём остовное дерево связного псевдографа (т.е. графа, в котором могут быть кратные рёбра) с помощью следующего алгоритма.

Граф называется *нагруженным*, если ребра этого графа поставлены в соответствии веса, так что ребру (x_i, x_j) сопоставлено некоторое число $c(x_i, x_j) = c_{ij}$, называемое длиной (или весом, стоимостью ребра).

Длиной (или весом или стоимостью) пути S , состоящего из некоторой последовательности дуг (x_i, x_j) , называется число $l(s)$, равное сумме длин дуг входящих в этот путь, т.е. $l(s) = \sum c_{ij}$ причем суммирование ведется по всем дугам $(x_i, x_j) \in S$. Матрица $C = (C_{ij})$ называется *матрицей весов*.

Пример. Рассмотрим граф.

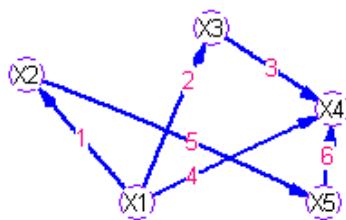


Рис.13. Пример нагруженного дерева.

Длина пути (x_1, x_2, x_5, x_4) равна $1+5+6=12$. Пусть G – связный нагруженный граф. Задача построения минимального основного дерева заключается в том, чтобы из множества основных деревьев найти дерево у которого сумма длин ребер минимальна.

Приведем типичные случаи, когда возникает необходимость построения минимального основного дерева графа:

1. Нужно соединить n городов железнодорожными линиями (автомобильными дорогами, линиями электропередач, сетью трубопроводов и т.д.) так, чтобы суммарная длина линий или стоимость была бы минимальной
2. Требуется построить схему электрической сети, в которой элементы должны быть соединены с помощью проводов наименьшей общей длины.

Описание алгоритма построения минимального основного дерева
Алгоритм Краскала

Шаг 1. Установка начальных значений. Вводится матрица длин ребер C графа G

Шаг 2. Выбрать в графе G ребро минимальной длины. Построить граф G_2 , состоящий из данного ребра и инцидентных ему вершин. Положить $i = 2$

Шаг 3. Если $i=n$, где n – число ребер графа, то значить работу (задача решена) в противном случае перейти к шагу 4.

Шаг 4. Построить граф G_{i+1} , добавляя к графу G_i новое ребро минимальной длины, выбранное среди всех ребер графа G , каждое из которых инцидентно какой-нибудь вершине графа G_i и одновременно инцидентно какой-нибудь вершине графа G , не содержащейся в G_i . Вместе с этим ребром включаем в G_{i+1} и инцидентную ему вершину, не содержащуюся в G_i . Присваиваем $i := i+1$ и переходим к шагу 3.

Пример: Найти минимальное остовное дерево для графа G (рис. 14).

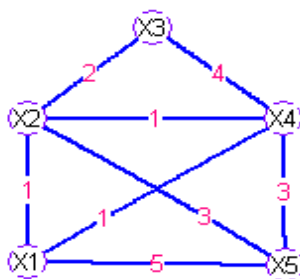


Рис.14 Задача построения минимального остовного дерева

Шаг 1. Установка начальных значений. Введем матрицу длин ребер G

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

Шаг 2. Выберем ребро минимальной длины. Минимальная длина ребра равна единице. Таких ребер три (x_1, x_2) , (x_2, x_4) , (x_1, x_4) В этом случае можно взять любое.

Возьмем (x_1, x_2) . Построим граф G_2 , состоящий из данного ребра и инцидентных ему вершин x_1 и x_2 . Положим $i=2$ к шагу 4.

Шаг 4. Строим граф G_3 , добавляя к графу G_2 новое ребро минимальной длины, выбранное среди всех ребер графа G , каждое из которых инцидентно одной из вершин x_1, x_2 и одновременно инцидентно какой-нибудь вершине графа G , не содержащейся в G_2 т.е. одной из вершин x_3, x_4, x_5 . Таким образом нужно выбрать ребро минимальной длины из ребер (x_1, x_4) (x_1, x_5) (x_2, x_3) (x_2, x_4) (x_2, x_5) . Таких ребер длины единица два (x_1, x_4) и (x_2, x_4) . Можно выбрать любое. Возьмем (x_1, x_4) . Вместе с этим ребром включаем в G_3 вершину x_4 , не содержащуюся в G_2 . Полагаем $i=3$ и переходим к шагу 3.

Шаг 3. Так как $i \neq n$, то переходим к шагу 4.

Шаг 4. Строим граф G_4 , добавляя к графу G_3 новое ребро минимальной длины из ребер (x_1, x_5) , (x_1, x_5) , (x_2, x_3) , (x_2, x_5) , (x_4, x_5) . Такое ребро длины два одно (x_2, x_3) . Вместе с этим ребром включаем в G_4 вершину x_3 , не содержащуюся в G_3 . помогаем $i=4$ и переходим к шагу 3.

Шаг 3. Так как $i \neq n$, переходим к шагу 4.

Шаг 4. Строим граф G_5 , добавляя к графу G_3 новое ребро минимальной длины из ребер (x_1, x_5) , (x_2, x_5) , (x_4, x_5) . Таких ребер длины 3 два (x_2, x_5) и (x_4, x_5) возьмем (x_2, x_5) .

Вместе с этим ребром включаем в G_5 вершину x_5 , не содержащуюся в G_4 . Полагаем $i=5$ и переходим к шагу 3.

Шаг 3. Так как $i=n$, то граф G_5 – искомое минимальное остовное дерево. Суммарная длина ребра равна $1+1+2+3=7$.

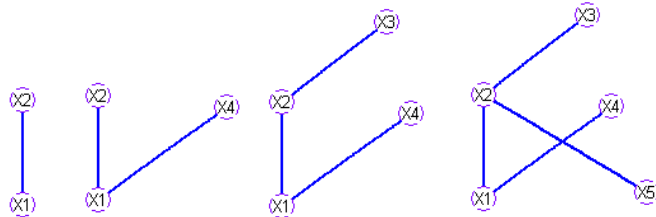


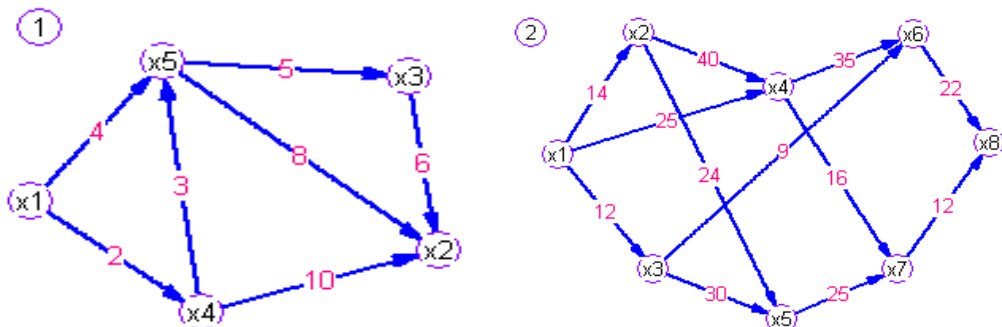
Рис.15. Ход построения минимального остовного дерева.

Ориентация неориентированного дерева осуществляется следующим образом. В дереве G отмечается корень x_0 и все ребра такого дерева с корнем ориентируются от этой вершины. Вершину x_i ребра (x_i, x_{i+1}) можно соединить единственной цепью L с корнем x_0 . Если эта цепь не содержит ребра (x_i, x_{i+1}) в это ребро вводится ориентация от x_i к x_{i+1} , в противном случае от x_{i+1} к x_i . Данная ориентация дерева с корнем x_0 единственная ориентированное таким образом дерево с корнем называют ориентированным деревом. В нем все ребра имеют направление от корня. При выборе другой вершины-корня получаем другой орграф-дерево.

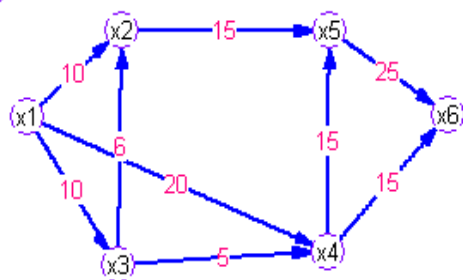
Множество всех вершин, связанных с корнем цепями, проходящими через вершин X порождает подграф $G(X)$ называемый ветвью вершины X в дереве с корнем x_0 .

Варианты заданий

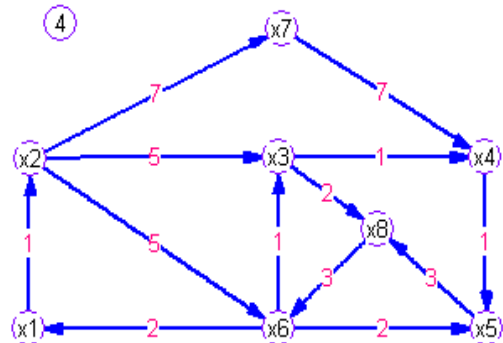
1. Написать программу, реализующую алгоритм Краскала.
2. Используя алгоритм Краскала, построить минимальное остовное дерево:



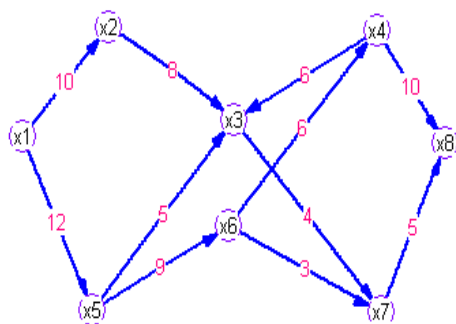
3



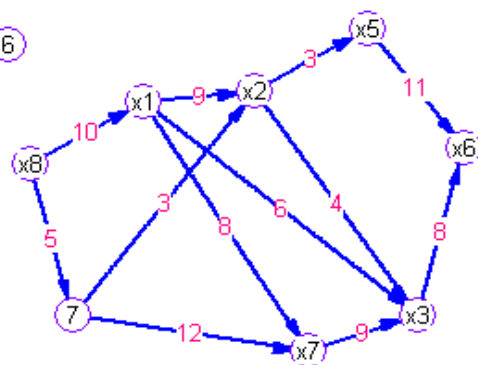
4



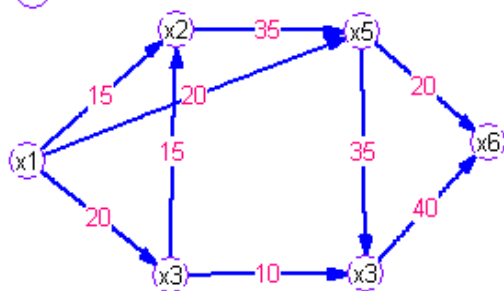
5



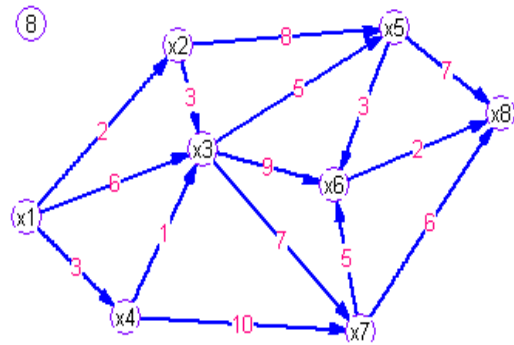
6



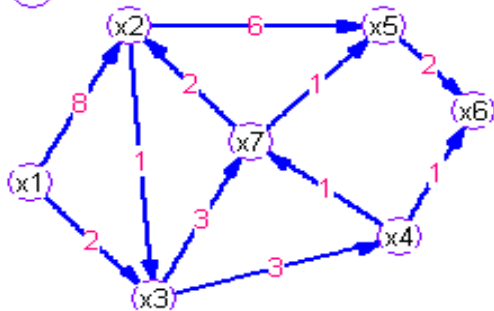
7



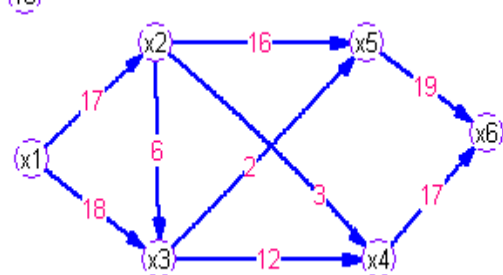
8



9



10



Контрольные вопросы.

1. Дайте определение дерева; ориентированного дерева.
2. Какое дерево называется остовым?
3. Свойства остовых деревьев. Теорема Кэли.
4. Что называется корнем дерева?
5. Как преобразовать неориентированное дерево в ориентированное?
6. Сколько ребер содержит остовое дерево графа?
7. Задача нахождения кратчайшего остова графа.
8. Приведите практические примеры нахождения кратчайшего остова графа.
9. Реализация алгоритма Прима-Краскала для нахождения кратчайшего остова графа.

ЛАБОРАТОРНАЯ РАБОТА № 7 «РАСКРАСКА ГРАФА»

Цель работы - изучение способа правильной раскраски графа на основе эвристического алгоритма.

Порядок выполнения работы

1. Получить задание у преподавателя в виде исходного неориентированного графа.
2. Составить блок-схему программы, определяющей раскраску графа с помощью эвристического алгоритма.
3. Создать программу, реализующую эвристический алгоритм раскраски графа. Исходный граф задается в виде матрицы смежности, вводимой построчно с помощью консоли. Программа должна вывести список полученных цветов для всех вершин графа.

Теоретическая часть

Разнообразные задачи, возникающие при планировании производства, составлении графиков осмотра, хранения и транспортировке товаров и т.д., могут быть представлены часто как задачи теории графов, тесно связанные с так называемой "задачей раскраски". Графы, рассматриваемые в данной лабораторной работе, являются неориентированными и не имеют петель.

Граф G называют l -хроматическим, если его вершины *могут быть раскрашены* с использованием l цветов (красок) так, что не найдется двух смежных вершин одного цвета. Наименьшее число l , такое, что граф G является l -хроматическим, называется *хроматическим числом* графа G и обозначается

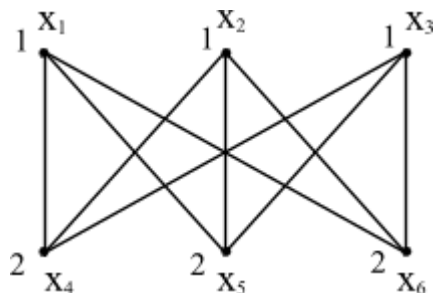
$\gamma(G)$. Задача нахождения хроматического числа графа называется *задачей о раскраске* (или *задачей раскраски*) графа. Соответствующая этому числу раскраска вершин разбивает множество вершин графа на r подмножеств, каждое из которых содержит вершины одного цвета. Эти множества являются независимыми, поскольку в пределах одного множества нет двух смежных вершин.

Задача нахождения хроматического числа произвольного графа явилась предметом многих исследований в конце XIX и в XX столетии. По этому вопросу получено много интересных результатов.

Хроматическое число графа нельзя найти, зная только числа вершин и ребер графа. Недостаточно также знать степень каждой вершины, чтобы вычислить хроматическое число графа. При известных величинах n (число вершин), m (число ребер) и $\deg(x_1), \dots, \deg(x_n)$ (степени вершин графа) можно получить только верхнюю и нижнюю оценки для хроматического числа графа.

Пример раскраски графа приведен на рисунке 16. Этот граф является одной из запрещенных фигур, используемых для определения планарности. Цифрами “1” и “2” обозначены цвета вершин.

Максимальное число независимых вершин графа $\alpha(G)$, равное мощности наибольшего множества попарно несмежных вершин, совпадает также с мощностью наибольшего множества вершин в G , которые могут быть окрашены в один цвет, следовательно:



$$\gamma(G) \geq \left\lceil \frac{n}{\alpha(G)} \right\rceil,$$

где n - число вершин графа G , а $\lceil x \rceil$ обозначает наибольшее целое число, не превосходящее x .

Рисунок 16 – Двудольный бихроматический граф Кенига.

Еще одна нижняя оценка для $\gamma(G)$ может быть получена следующим образом:

$$\gamma(G) \geq \frac{n^2}{n^2 - 2m}$$

Верхняя оценка хроматического числа может быть вычислена по формуле: $\gamma(G) \leq 1 + \max_{x_i \in X} [d(x_i) + 1]$.

Применение оценок для хроматического числа значительно сужает границы решения. Для определения оценки хроматического числа также могут использоваться другие топологические характеристики графа, например, свойство планарности.

Граф, который можно изобразить на плоскости так, что никакие два его ребра не пересекаются между собой, называется *планарным*.

Теорема о пяти красках. Каждый планарный граф можно раскрасить с помощью пяти цветов так, что любые две смежные вершины будут окрашены в разные цвета, т. е. если граф G - планарный, то $\gamma(G) \leq 5$.

Гипотеза о четырех красках (недоказанная). Каждый планарный граф можно раскрасить с помощью четырех цветов так, что любые две смежные вершины будут окрашены в разные цвета, т. е. $\gamma(G) \leq 4$, если граф G - планарный.

В 1852 г. о гипотезе четырех красок говорилось в переписке Огюста де Моргана с сэром Вильямом Гамильтоном. С тех пор эта "теорема" стала, наряду с теоремой Ферма, одной из самых знаменитых нерешенных задач в математике.

Полный граф G_n всегда раскрашивается в n цветов, равных количеству его вершин.

Эвристические алгоритмы раскрашивания.

Точные методы раскраски графа сложны для программной реализации. Однако существует много эвристических процедур раскрашивания, позволяющих находить хорошие приближения для определения хроматического числа графа. Такие процедуры также могут с успехом использоваться при раскраске графов с большим числом вершин, где применение точных методов не оправдано ввиду высокой трудоемкости вычислений.

Из эвристических процедур раскраски следует отметить последовательные методы, основанные на упорядочивании множества вершин.

В одном из простейших методов вершины вначале располагаются в порядке убывания их степеней. Первая вершина окрашивается в цвет 1; затем список вершин просматривается по убыванию степеней и в цвет 1 окрашивается каждая вершина, которая не является смежной с вершинами, окрашенными в тот же цвет. Потом возвращаемся к первой в списке неокрашенной вершине, окрашиваем ее в цвет 2 и снова просматриваем список вершин сверху вниз, окрашивая в цвет 2 любую неокрашенную вершину, которая не соединена ребром с другой, уже окрашенной в цвет 2 вершиной. Аналогично действуем с цветами 3, 4 и т. д., пока не будут окрашены все вершины. Число использованных цветов будет тогда приближенным значением хроматического числа графа.

Эвристический алгоритм раскраски вершин графа:

Шаг 1. Сортировать вершины графа по степеням убывания:

$\deg(x_i) \geq \deg(x_j), \forall x_i, x_j \in G$; Установить текущий цвет $p=1; i=1$.

Шаг 2. Выбрать очередную не раскрашенную вершину из списка и назначить ей новый цвет: $\text{col}(x_i)=p; X_p=\{x_i\}$.

Шаг 3. $i=i+1$. Выбрать очередную не раскрашенную вершину x_i и проверить условие смежности: $x_i \cap \Gamma(X_p)=\emptyset$, где X_p – множество вершин, уже раскрашенных в цвет p . Если вершина x_i не является смежной с данными вершинами, то также присвоить ей цвет p : $\text{col}(x_i)=p$.

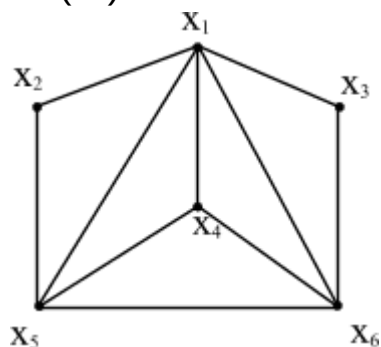
Шаг 4. Повторить п.3, пока не будет достигнут конец списка ($i=n$).

Шаг 5. Если все вершины графа раскрашены, то – конец алгоритма;

Иначе: $p=p+1$; $i=1$. Повторить п.2.

Пример. Раскрасим граф G , изображенный на рисунке 15. Промежуточные данные для решения задачи будем записывать в таблицу. Отсортируем вершины графа по убыванию их степеней. В результате получается вектор отсортированных вершин $X^*=\{x_1, x_5, x_6, x_4, x_2, x_3\}$.

Соответствующие данным вершинам степени образуют второй вектор: $D=\{5, 4, 4, 3, 2, 2\}$. В первой строке таблицы запишем вектор X^* ; во второй – вектор D . Последующие строки отражают содержание вектора раскраски $col(X^*)$



Номера вершин X^*	x_1	x_5	x_6	x_4	x_2	x_3
Степени вершин D	5	4	4	3	2	2
$p=1$	1	-	-	-	-	-
$p=2$	1	2	-	-	-	2
$p=3$	1	2	3	-	3	2
$p=4$	1	2	3	4	3	2

Рисунок 17.

Таким образом, данный граф можно раскрасить не менее чем в четыре цвета, т.е. $\gamma(G)=4$.

Псевдокод разработанного алгоритма

Sk - независимое множество вершин, покрашенных в цвет k .

begin

1. $k := 1$;

2. **for** $i=0$ to n

3. **if** $i \notin V$ **then** continue /*vertex i is coloured*/

4. $Sk := Sk \cup \{i\}$;

5. $V := V \setminus \{i\}$;

6. **while** i has not coloured non-neighbours /* row i contains 0*/

7. **for** $j=i+1$ to $n-1$ /* find the first non- neighbor*/

8. **if** $a_{ij} = 0$ AND $j \in V$ **then** break

9. **end for**

10. **if** $j = n$ **then** break

11. $Sk := Sk \cup \{j\}$;

12. $V := V \setminus \{j\}$;

13. $A_i := A_i \vee A_j$;

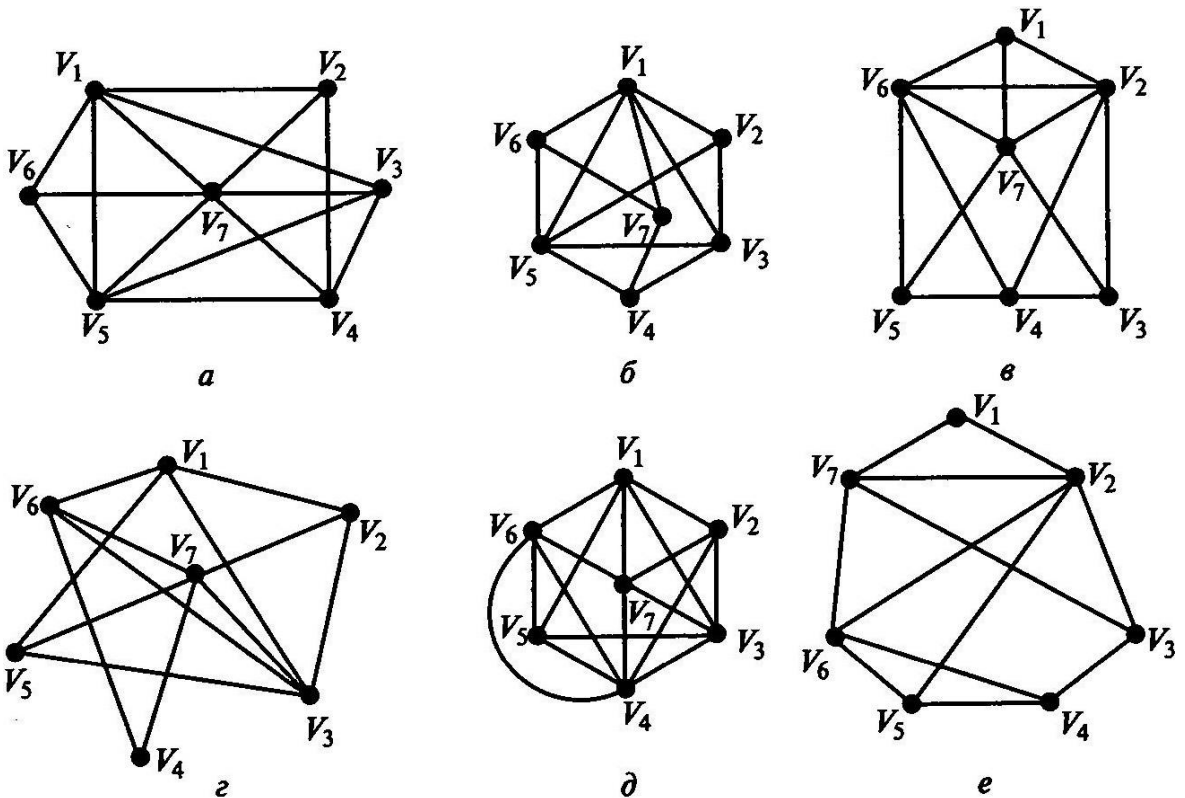
14. **end while**

15. $k := k+1$;

16. **end for**

end

Варианты заданий



Контрольные вопросы

1. Сформулируйте задачу раскраски графа.
2. Какой граф называется Γ -хроматическим?
3. Что называется хроматическим числом графа?
4. Как определяются нижняя и верхняя оценки хроматического числа?
5. Какой граф называется планарным? Во сколько цветов можно его раскрасить?
6. Во сколько цветов можно раскрасить полный граф?
7. Эвристический алгоритм раскраски графа.
8. Реализация эвристического алгоритма для нахождения хроматического числа графа.

ЛАБОРАТОРНАЯ РАБОТА № 8 «ПОТОКИ В СЕТЯХ. МЕТОД ФОРДА-ФАЛКЕРСОНА О МАКСИМАЛЬНОМ ПОТОКЕ»

Цели работы: изучить задачу о максимальном потоке; изучить алгоритм нахождения максимального потока; написать программу, используя изученный алгоритм.

Порядок выполнения работы

1. В соответствии с вариантом задания, определенным преподавателем, провести расчеты.
2. Оформить отчет о выполнении задания с приведением условия задачи, формул для расчета, результатов решения и заключения.

Теоретическая часть

Задача о максимальном потоке является классической и имеет множество применений.

Постановка проблемы: Дан взвешенный ориентированный граф с неотрицательными весами (пропускными способностями). Выделены две вершины: исток S и сток T такие, что любая другая вершина лежит на пути из S в T . Поток назовем функцию $F: V \times V$ с такими свойствами

Ограничение пропускной способности. Поток по ребру не может быть больше его (ребра) пропускной способности.

Антисимметричность. Для каждого ребра (u, v) : $F(u, v) = -F(v, u)$.

Сохранение потока. Для каждой вершины (кроме S и T), количество входящего потока (отрицательного) равен количеству исходящего потока (положительного). То есть, алгебраическая сумма потоков для каждой вершины (кроме S и T) равна нулю.

Алгоритм нахождения максимального потока

Для нахождения максимального потока используются, например, известный алгоритм Форда-Фалкерсона и его модификации.

Шаг 1. Построить для данного потока φ увеличивающую цепь из x_1 в x_n . Если такой цепи нет, то поток φ - максимальный, иначе перейти к шагу 2.

Шаг 2. Для увеличивающей цепи $x_1 = v_1, u_2, v_2, u_2, \dots, v_k, u_k, v_{k+1} = x_n$ найти значение величины:

$$\Delta = \min_{1 \leq i \leq k} (\Delta u_i), \text{ где}$$
$$\Delta u_i = \begin{cases} c(v_i, v_{i+1}) - \varphi(v_i, v_{i+1}), & \text{если } v_{i+1} \in G(v_i) \\ \varphi(v_i, v_{i+1}), & \text{если } v_{i+1} \in G^{-1}(v_i) \end{cases}$$

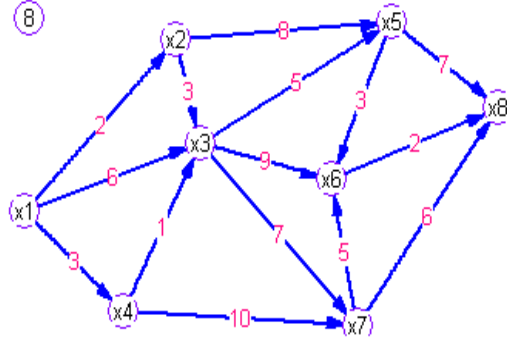
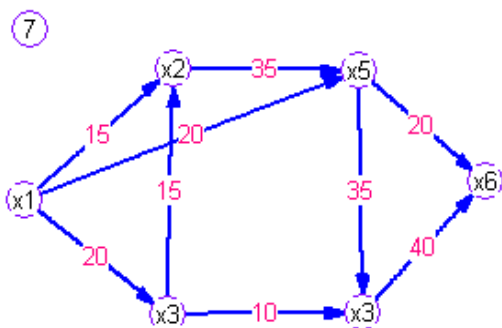
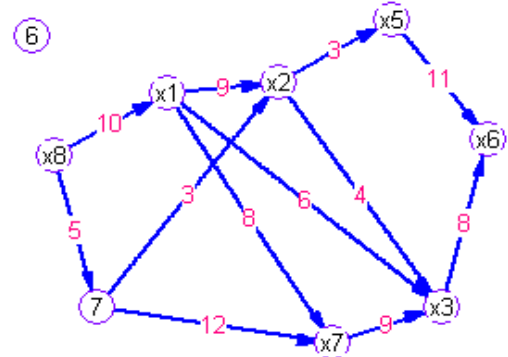
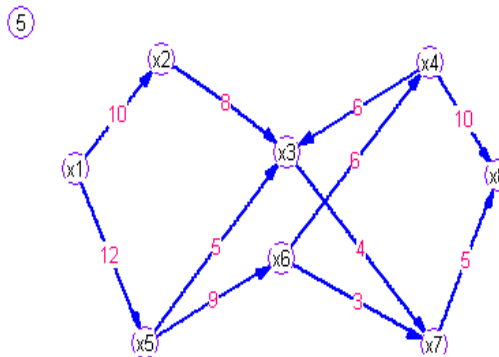
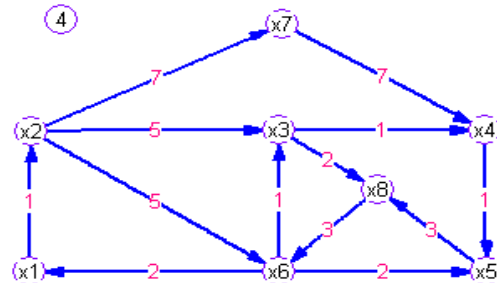
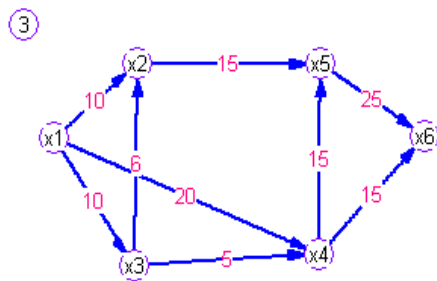
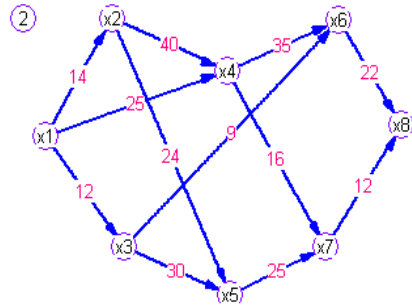
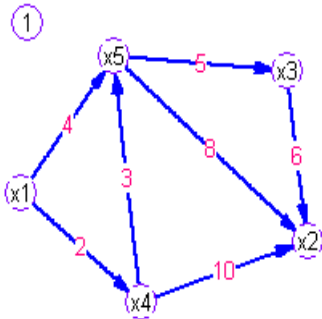
Шаг 3. Построить новый поток по следующему правилу: на дугах увеличивающей цепи значение потока по дуге u_i увеличить на Δ если v_{i+1} е

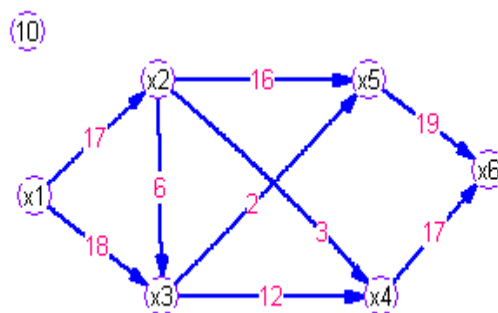
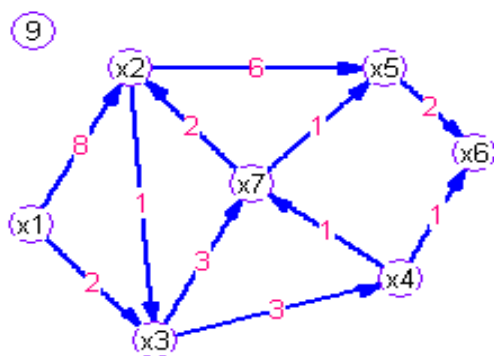
Δ

$G(v_i)$ и уменьшить на θ , если $v_i \in G=(v_j)$. По дугам, не входящим в эту цепь, значение потока не менять и перейти к шагу 2.

Варианты заданий

Пользуясь алгоритмом, найти полный поток в транспортной сети.





Контрольные вопросы

1. Сформулируйте задачу о максимальном потоке.
2. Что называется потоком?
3. Приведите алгоритм нахождения максимального потока.
4. Укажите формулу асимметрии ребер.

ЛАБОРАТОРНАЯ РАБОТА № 9 «ЭЛЕМЕНТЫ ТЕОРИИ КОДИРОВАНИЯ»

Цель работы состоит в изучении основных понятий теории кодирования; овладении алгоритмами кодирования, широко используемыми в информатике.

Порядок выполнения работы

1. В соответствии с вариантом задания, определенным преподавателем, провести расчеты.
2. Оформить отчет о выполнении задания с приведением условия задачи, формул для расчета, результатов решения и заключения.

Теоретический материал

Зададим конечное множество $A = \{a_1, a_2, \dots, a_n\}$, которое назовем алфавитом. Элементы $a_i \in A$ будем называть буквами.

Последовательность букв, т. е. буквы алфавита, записанные друг за другом, назовем словом, его будем обозначать α . Количество букв в слове α назовем его длиной $|\alpha|$.

Говорят, что задано алфавитное кодирование, если каждой букве a_i алфавита A сопоставлено определенное слово b_i алфавита B , причем такие слова называют элементарными кодами. Таблица, в которой записаны все буквы a_i алфавита A и соответствующие им элементарные коды b_i , называется схемой алфавитного кодирования. Если задано алфавитное кодирование, то любое слово алфавита A можно представить последовательностью элементарных кодов алфавита B . Последовательность слов α алфавита A на-

зывают текстом на языке A . Перевод всех слов данного текста в коды алфавита B называют кодированием.

Если задан текст на языке A , то можно подсчитать для каждой буквы a_i частоту ее появления в этом тексте. Обозначим ее m_i . Исключим все буквы, не встречающиеся в тексте. Если задать схему кодирования, то можно найти цену схемы кодирования.

Алгоритм кодирования Фано состоит в следующем.

Расположим встречающиеся в заданном тексте буквы так, чтобы $m_1 > m_2 > \dots > m_n > 0$, т. е. по убыванию частоты их появления в тексте. Каждая буква a_i имеет такой же номер, что и m_i – число ее появления в этом тексте, написанном, между прочим, на русском языке. Если буквы имеют одинаковую частоту, то располагаем их по алфавиту русского языка. Для каждой буквы a_i требуется найти элементарный код b_i . В алгоритме Фано алфавит B состоит из 0 и 1. Разобьем отсортированный список всех букв на две группы так, чтобы суммы частот появлений букв каждой из групп были как можно более близки друг к другу. Кодам, соответствующим буквам из первой группы, приписываем 0. Зато кодам, сопоставленным буквам из второй группы, приписываем 1. По тому же принципу каждая из полученных групп снова разбивается на подгруппы, и это разбиение определяет символ, 0 или 1, приписываемый справа к коду i . Процедура разбиения продолжается до тех пор, пока все множество букв не будет разбито на подгруппы, состоящие только из одной буквы. В результате каждой букве a_i будет соответствовать элементарный код b_i , образованный из нулей и единиц. Заметим, что буквами можно считать пробелы, запятые, точки и другие знаки препинания, встречающиеся в тексте русского языка.

Пример. Закодируем текст «шла Саша по шоссе и сосала сушку» с помощью алгоритма Фано.

Подсчитаем, что буква «ш» встречается 4 раза, «л» – 2 раза, «а» – 5, «с» – 6, «п» – 1, «о» – 3, «е» – 1, «и» – 1, «у» – 2, «к» – 1 раз, пробелов « » – 6 раз. Расположим частоты появления букв по убыванию. Заметим, что номер буквы в списке букв соответствует номеру частоты в упорядоченном по убыванию списке частот, причем буквы с одинаковой частотой располагаются по алфавиту русского языка. Таким образом, имеем

Номер	Буква	Частота в тексте буквы
1	Пробел« »	6
2	с	6
3	а	5
4	ш	4
5	о	3
6	л	2
7	у	2
8	е	1
9	и	1
10	к	1
11	п	1

Сумма всех частот равна 32. Суммируем частоты до тех пор, пока разность не станет минимальной среди всех разбиений на две группы. Находим, что сумма первых трех частот $6+6+5=17$ – такова, что разность $(6+6+5)-(4+3+2+2+1+1+1+1)$ минимальна. Поэтому первые три буквы кодируем пока нулями, а остальные – единицами.

Номер	Буква	Частота появления буквы в заданном тексте	Код
1	Пробел « »	6	0
2	с	6	0
3	а	5	0
4	ш	4	1
5	о	3	1
6	л	2	1
7	у	2	1
8	е	1	1
9	и	1	1
10	к	1	1
11	п	1	1

На втором шаге делим обе полученные группы в свою очередь еще раз на две подгруппы. Заметим, что разность, равная пяти, меньше разности, равной семи. В силу этого верхняя группа поделится таким образом: первая буква в одной подгруппе, а вторая с третьей буквой – во второй подгруппе. Нижняя группа поделится следующим образом. Так как разность $(4+3)-(2+2+1+1+1+1)$ является минимальной среди всех разбиений, то четвертая и пятая буквы попадут в одну подгруппу, а остальные в другую. Таким образом, коду первой буквы дописываем справа ноль, коду второй и третьей буквы – единицу. Для нижней «половины»: коду четвертой и пятой буквы дописываем ноль, а остальным – единицу.

Номер	Буква	Частота в тексте	Код
1	Пробел « »	6	00
2	с	6	01
3	а	5	01
4	ш	4	10
5	о	3	10
6	л	2	11
7	у	2	11
8	е	1	11
9	и	1	11
10	к	1	11
11	п	1	11

Первая «четвертинка» уже не может поделиться, так как содержит всего одну букву. Вторая «четвертинка», очевидно, поделится так, что к коду второй буквы потребуется дописать справа 0, а к коду третьей буквы добавим 1. Аналогично, поделив пополам третью «четвертинку», мы получим, что код четвертой буквы удлинится на 0, а код пятой буквы – на 1. Заметим, что раз-

ность $(2+2)-(1+1+1+1)$ минимальна среди всех разбиений последней «четвертинки» на подгруппы, следовательно, к кодам шестой и седьмой буквы допишем ноль, а к остальным – единицу. Имеем:

Номер	Буква	Частота	Код
1	Пробел « »	6	00
2	с	6	010
3	а	5	011
4	ш	4	100
5	о	3	101
6	л	2	110
7	у	2	110
8	е	1	111
9	и	1	111
10	к	1	111
11	п	1	111

На четвертом шаге кроме последних двух «восьмушек» никакие другие не могут поделиться, поэтому разобьем их пополам и получим

Номер	Буква	Частота	Код
1	Пробел « »	6	00
2	с	6	010
3	а	5	011
4	ш	4	100
5	о	3	101
6	л	2	1100
7	у	2	1101
8	е	1	1110
9	и	1	1110
10	к	1	1111
11	п	1	1111

Произведем последний раз разбиение на подгруппы.

Номер	Буква	Частота	Элементарный код
1	Пробел « »	6	00
2	с	6	010
3	а	5	011
4	ш	4	100
5	о	3	101
6	л	2	1100
7	у	2	1101
8	е	1	11100
9	и	1	11101
10	к	1	11110
11	п	1	11111

Мы получили схему кодирования. Цена этой схемы кодирования $\square\square\square\square$ в 3,2 единицы означает, что для каждой буквы текста потребуется

примерно три позиции в закодированном сообщении. Так как букв в заданном тексте 32, то всего закодированный текст займет примерно 102 позиции ($32 \cdot 3,2 = 102,4$). Сам закодированный текст будет выглядеть следующим образом:

«100110001100010011100011001111110100100101010010111000011101000101010100111100011000101101100111101101».

Варианты заданий

Вариант1.

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода.

$a_1 - 0,33$; $a_2 - 0,24$; $a_3 - 0,18$; $a_4 - 0,14$; $a_5 - 0,09$; $a_6 - 0,02$

II. Закодировать данное слово кодом Хэмминга.

01011100001011000101101.

III. Пользуясь кодом Хэмминга найти ошибку в сообщении.

11111001100010101100101101

Вариант2

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода

$a_1 - 0,38$; $a_2 - 0,21$; $a_3 - 0,18$; $a_4 - 0,15$; $a_5 - 0,08$

II. Закодировать данное слово кодом Хэмминга.

01000111001011100011010

III. Пользуясь кодом Хэмминга найти ошибку в сообщении.

1101011001011100100110001

Вариант3

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода

$a_1 - 0,4$; $a_2 - 0,16$; $a_3 - 0,15$; $a_4 - 0,14$; $a_5 - 0,1$; $a_6 - 0,05$

II. Закодировать данное слово кодом Хэмминга.

10010111010111000011011

III. Пользуясь кодом Хэмминга найти ошибку в сообщении.

1101011001011100100110001

Вариант4

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода

$a_1 - 0,4$; $a_2 - 0,3$; $a_3 - 0,2$; $a_4 - 0,08$; $a_5 - 0,02$

II. Закодировать данное слово кодом Хэмминга.

11100100110001010110101

III. Пользуясь кодом Хэмминга найти ошибку в сообщении.

1101011001011100100110001

Вариант5

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода

$a_1 - 0,31; a_2 - 0,23; a_3 - 0,22; a_4 - 0,14; a_5 - 0,1$

II. Закодировать данное слово кодом Хэмминга.

00010111010011100001010

III. Пользуясь кодом Хэмминга найти ошибку в сообщении.

01000111000011001010111001

Вариант6

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода

$a_1 - 0,4; a_2 - 0,3; a_3 - 0,15; a_4 - 0,1; a_5 - 0,05$

II. Закодировать данное слово кодом Хэмминга.

11100011100011100011100

III. Пользуясь кодом Хэмминга найти ошибку в сообщении.

01110010011000101010111011

Вариант7

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода

$a_1 - 0,4; a_2 - 0,2; a_3 - 0,18; a_4 - 0,12; a_5 - 0,08; a_6 - 0,02$

II. Закодировать данное слово кодом Хэмминга.

10011000111010100110101

III. Пользуясь кодом Хэмминга найти ошибку в сообщении.

0011011000111100011001011

Вариант8

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода

$a_1 - 0,5; a_2 - 0,2; a_3 - 0,15; a_4 - 0,1; a_5 - 0,05$

II. 01001101011100100100101

Пользуясь кодом Хэмминга найти ошибку в сообщении.

III. 111110010010110001011100110

Вариант9

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода

$a_1 - 0,37; a_2 - 0,25; a_3 - 0,19; a_4 - 0,12; a_5 - 0,06; a_6 - 0,01$

II. Закодировать данное слово кодом Хэмминга.

11001000101110010110101

III. Пользуясь кодом Хэмминга найти ошибку в сообщении.

1011010101110100100011001

Вариант10

I. Построить оптимальный код, пользуясь алгоритмами Хаффмана и Фано, в каждом случае найти стоимость кода

$a_1 - 0,4$; $a_2 - 0,2$; $a_3 - 0,15$; $a_4 - 0,13$; $a_5 - 0,12$

II. Закодировать данное слово кодом Хэмминга.

10100111010100011001010

III. Пользуясь кодом Хэмминга найти ошибку в сообщении.

1011110011001100001010111

Контрольные вопросы

1. Что включает в себя алфавитное кодирование.
2. Охарактеризуйте цену кодирования.
3. Дайте схематичное описание алгоритма кодирования Фано.
4. Как найти оптимальный код, используя алгоритм Хаффмана?
5. В чем отличие алгоритмов Фано и Хаффмана.