

С.Ю. Белецкая, Ю.В.Литвиненко, Б.Н. Тишуков

***Модели и алгоритмы дискретной математики в
автоматизированных системах***

Учебное пособие

ВВЕДЕНИЕ

При решении практических задач в автоматизированных системах широкое распространение находят модели и алгоритмы дискретной математики. Аппарат дискретной математики используется на разных этапах жизненного цикла автоматизированных систем: при проектировании, разработке и эксплуатации. При этом особое значение имеет теория графов, которая является эффективным инструментом моделирования распределённых систем различной природы (вычислительных сетей, информационных систем, транспортных сетей, электрических сетей, организационных структур и т.д.). Кроме того, важную роль с практической точки зрения играет использование аппарата комбинаторики и теории множеств, в частности, нечётких множеств и отношений. Поэтому важным является формирование у студентов целостного представления о современных методах дискретной математики, а также развития у них практических навыков описания дискретных объектов в прикладных задачах.

Пособие состоит из пяти глав. В первой главе рассматриваются базовые понятия дискретной математики: множества и отношения. Вторая глава посвящена комбинаторике. Третья, четвертая и пятая главы излагают основы теории графов. В третьей главе приведены основные понятия и определения, в четвертой рассматриваются базовые алгоритмы на графах, пятая глава содержит примеры использования аппарата теории графов для решения практических инженерных задач в автоматизированных системах. В шестой главе представлены задания для проведения лабораторного практикума. Лабораторный практикум состоит из шести лабораторных работ и предполагает программную реализацию алгоритмов теории графов. Теоретический материал пособия иллюстрируется примерами.

Учебное пособие может быть использовано в лекционных курсах "Дискретная математика в программировании", "Методы оптимизации", "Теория алгоритмов", при проведении и лабораторных занятий по этим дисциплинам, а также в курсовом и дипломном проектировании.

1. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ И ОТНОШЕНИЙ

1.1. Основные понятия теории множеств

Под *множеством* понимают совокупность объединенных в единое целое объектов (элементов), различимых между собой. Запись $x \in X$ означает, что элемент x принадлежит множеству X .

Множества бывают конечными и бесконечными. Примером конечного множества является множество студентов студенческой группы. Примером бесконечного множества является множество всех действительных чисел. Для конечного множества M определено понятие мощности. Мощностью $|M|$ называется число элементов в данном конечном множестве. Если $|M|=0$, множество называется пустым и обозначается $\emptyset$. В теории множеств важным является также понятие универсального множества U . Универсальное множество U включает в себя все остальные множества.

Множества могут задаваться:

1. Перечислением элементов через запятую в фигурных скобках. Этот способ может быть применим только для конечных множеств. При этом порядок перечисления элементов не является существенным.

Примеры.

$A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

$\{1, 2, 3\} = \{3, 2, 1\} = \{1, 3, 3, 2\}$ - порядок перечисления элементов и их число не является существенным.

2. Описанием свойств, однозначно определяющих принадлежность элементов к данному множеству (для конечных и бесконечных множеств). Этому способу задания соответствует запись следующего вида:

$A = \{x \mid P(x)\}$.

Эта запись которая читается следующим образом: множество элементов x , обладающих свойством $P(x)$.

Пример.

$A = \{x \mid x \in \mathbb{Z}, 0 \leq x \leq 9\}$ - множество цифр от 0 до 9.

Множество A является *подмножеством* множества B , если каждый элемент множества A является элементом множества B . Это обозначается следующим образом: $A \subseteq B$.

Операции над множествами

1. *Объединением* множеств A и B называется множество $A \cup B$, все элементы которого являются элементами множества A или множества B .

Пример.

$$A=\{1, 2\} \quad B=\{2, 3\} \quad A \cup B=\{1, 2, 3\}.$$

2. *Пересечением* множеств A и B называется множество $A \cap B$, все элементы которого являются элементами обоих множеств.

Пример.

$$A=\{1, 2\} \quad B=\{2, 3\} \quad A \cap B=\{2\}.$$

Для операций объединения и пересечения выполняется свойство коммутативности: $A \cap B=B \cap A$ и $A \cup B=B \cup A$.

3. *Разностью* множеств A и B называется множество $A \setminus B$ всех тех элементов множества A , которые не принадлежат множеству B .

Пример.

$$A=\{1, 2, 3\} \quad B=\{2, 3, 5\} \quad A \setminus B=\{1\} \quad B \setminus A=\{5\}.$$

При этом $A \setminus B \neq B \setminus A$.

4. *Симметрической разностью* множеств A и B называется множество $A+B$, определяемое следующим образом: $A+B=(A \setminus B) \cup (B \setminus A)$.

Пример.

$$A=\{1, 2, 3\}, \quad B=\{2, 3, 5\}, \quad A+B=\{1, 5\}.$$

5. *Дополнением* множества A называется множество $\bar{A}$ всех элементов, которые не принадлежат множеству A . Можно записать: $\bar{A} = U \setminus A$.

Для наглядной иллюстрации операций над множествами могут быть использованы диаграммы Эйлера-Венна (рисунок 1.1):

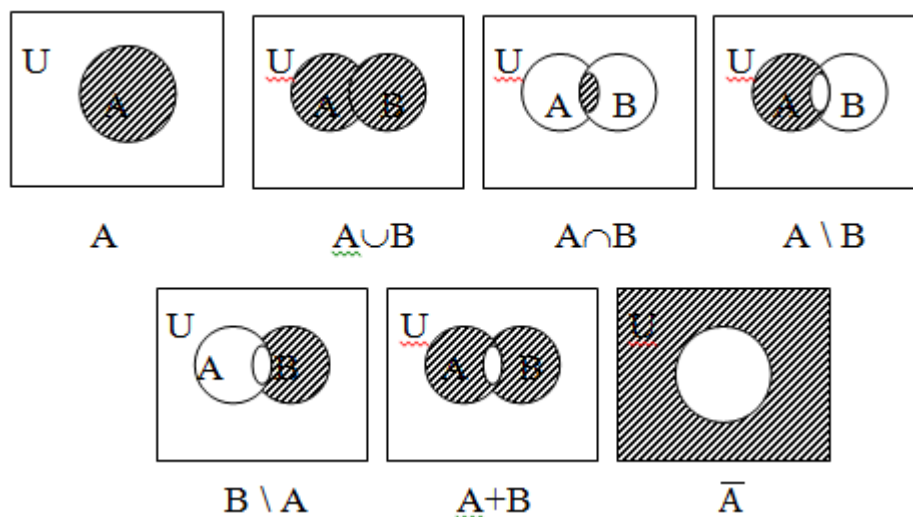


Рис. 1.1. Диаграммы Эйлера-Венна

Для преобразования выражений с множествами содержащих множества, используются основные тождества алгебры множеств. Приоритет операций (в порядке убывания): дополнение, пересечение, объединение, разность, симметрическая разность.

Основные тождества алгебры множеств

- | | |
|---|---|
| 1. $A \cup B = B \cup A$ | 1'. $A \cap B = B \cap A$
(свойство коммутативности) |
| 2. $A \cup (B \cap C) = (A \cup B) \cap C$ | 2'. $A \cap (B \cup C) = (A \cap B) \cup C$
(свойство ассоциативности) |
| 3. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ | 3'. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
(свойство дистрибутивности) |
| 4. $A \cup A = A$ | 4'. $A \cap A = A$
(свойство идемпотентности) |
| 5. $\overline{A \cup B} = \overline{A} \cap \overline{B}$ | 5'. $\overline{A \cap B} = \overline{A} \cup \overline{B}$
(законы де Моргана) |
| 6. $A \cup (A \cap B) = A$ | 6'. $A \cap (A \cup B) = A$
(законы поглощения) |
| 7. $A \cup \overline{A} = U$ | 7'. $A \cap \overline{A} = \emptyset$ |
| 8. $A \cup U = U$ | 8'. $A \cap U = A$ |
| 9. $A \cup \emptyset = A$ | 9'. $A \cap \emptyset = \emptyset$ |
| 10. $\overline{\overline{A}} = A$ (двойное отрицание) | |
| 11. $A \setminus B = A \cap \overline{B}$ | |

Покрывание и разбиение множеств

Совокупность множеств $\{A_1, A_2, \dots, A_k\}$ называется *покрыванием* множества A , если имеет место равенство $A = A_1 \cup A_2 \cup \dots \cup A_k$, где $A_1, A_2, \dots, A_k$ - блоки покрытия.

Разбиением множества A называется такое покрытие множества A , в котором все множества $A_1, A_2, \dots, A_k$ попарно не пересекаются.

Пример.

$$A = \{1, 2, 3, 4, 5\},$$

$$A_1 = \{1, 2\} \quad A_2 = \{2, 3\} \quad A_3 = \{3, 4, 5\} \text{ - блоки покрытия,}$$

$$A_1 = \{1, 2\} \quad A_2 = \{3, 4, 5\} \text{ - блоки разбиения.}$$

Для любого разбиения множества A справедливо правило суммы:

$$|A| = |A_1| + |A_2| + \dots + |A_k|.$$

Для покрытия множества A это правило запишется следующим образом:

$$|A| \leq |A_1| + |A_2| + \dots + |A_k|.$$

Декартово произведение множеств

Декартовым (прямым) произведением двух множеств A и B называется совокупность упорядоченных пар (a, b) таких, что $a \in A$ и $b \in B$. Это обозначает-

ся как $A \times B$. $A \times B = \{(a, b) \mid a \in A, b \in B\}$. При этом в упорядоченной паре порядок следования элементов является существенным.

Декартовым произведением n множеств $A_1, A_2, \dots, A_n$ называется совокупность упорядоченных наборов из n элементов $(a_1, a_2, \dots, a_n)$ таких, что $a_1 \in A_1, a_2 \in A_2, \dots, a_n \in A_n$.

Если $A_1 = A_2 = \dots = A_n = A$, то декартово произведение $A_1 \times A_2 \times \dots \times A_n = A^n$ называется n -декартовой степенью множества A , обозначается A^n .

Пример.

$$A = \{1, 2, 3\}, \quad B = \{4, 5\},$$

$$A \times B = \{(1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)\}.$$

$$B \times A = \{(4, 1), (4, 2), (4, 3), (5, 1), (5, 2), (5, 3)\}.$$

$$A^2 = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}.$$

$$A \times B \neq B \times A.$$

Пусть $A_1, A_2, \dots, A_n$ - конечные множества, тогда справедливо правило $|A_1 \times A_2 \times \dots \times A_n| = |A_1| \cdot |A_2| \cdot \dots \cdot |A_n|$, которое называется *правилом произведения*.

1.2. Нечеткие множества

Рассмотрим множество E и его подмножество A : $A \subseteq E$. Принадлежность элемента x из множества E подмножеству A обозначается $x \in A$, однако для задания множества можно использовать характеристическую функцию $\mu_A(x)$, значения которой указывает, является ли элемент x из множества E элементом подмножества A .

$$\mu_A(x) = \begin{cases} 1, & \text{если } x \in A \\ 0, & \text{если } x \notin A \end{cases}$$

Пусть характеристическая функция может принимать любое значение в интервале $[0, 1]$. *Нечетким подмножеством* A множества E называется множество упорядоченных пар $\{(x \mid \mu_A(x))\} \forall x \in E$. Здесь $\mu_A(x)$ - *функция принадлежности*, принимающая свои значения в множестве M и указывающая степень или уровень принадлежности элемента x подмножеству A . Наряду с термином "нечеткое подмножество" используется также термин "нечеткое множество".

Примеры.

1. Даны множество E и множество принадлежностей M :

$$E = \{a, b, c, d, e, f\}, \quad M = \{0, 0.5, 1\}.$$

Тогда одно из нечетких подмножеств A можно определить следующим образом:

$$A = \{(a \mid 0), (b \mid 1), (c \mid 0.5), (d \mid 0), (e \mid 0.5), (f \mid 1)\}.$$

2. Пусть N множество натуральных чисел, тогда нечеткое подмножество небольших натуральных чисел может быть определено в виде

$$A = \{(1 | 1), (2 | 0.8), (3 | 0.6), (4 | 0.4), (5 | 0.2), (6 | 0), (7 | 0), \dots\}.$$

Значения функции принадлежности задаются в каждом случае субъективно. Для построения функции принадлежности используются экспертные оценки. Например, построим функцию принадлежности нечеткого подмножества чисел x приблизительно равному данному числу y (рисунок 1.2).

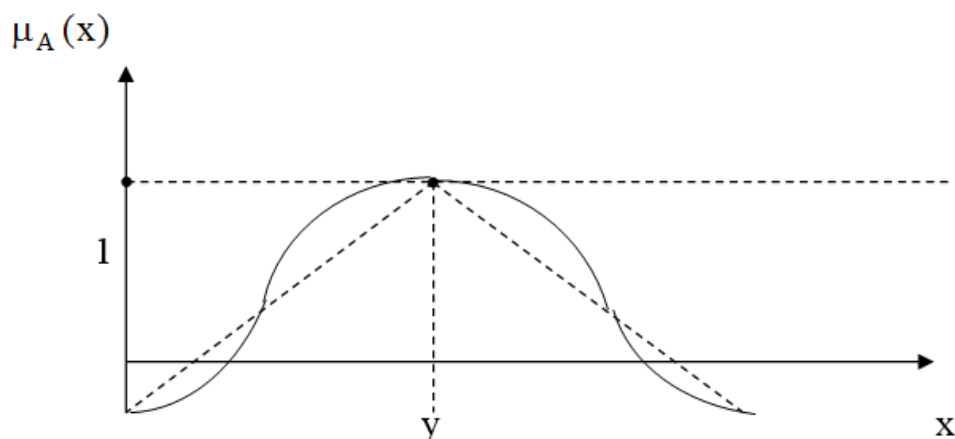


Рис 1.2. График функции принадлежности нечеткого подмножества « x приблизительно равно y »

Для конечных множеств E и M можно определить множество нечетких подмножеств $P(E)$. Если $|E|=n$, а $|M|=m$, то $|P(E)|=m^n$.

Пример.

$$E = \{x, y\}, M = \{0.2, 0.4\}.$$

$$P(E) = \{\{(x | 0.2), (y | 0.2)\}, \{(x | 0.2), (y | 0.4)\}, \{(x | 0.4), (y | 0.2)\}, \{(x | 0.4), (y | 0.4)\}\}.$$

Основные операции над нечеткими множествами

Рассмотрим множество E и два нечетких множества A и B .

1. *Включение.* Говорят, что множество A включено в B , если $\forall x \in E \mu_A(x) \leq \mu_B(x)$. При этом используется обозначение $A \subset B$.

Пример.

$$A = \{(x_1 | 0.4), (x_2 | 0.2), (x_3 | 0)\}.$$

$$B = \{(x_1 | 0.2), (x_2 | 0.1), (x_3 | 0)\}.$$

$$B \subset A.$$

2. *Равенство.* Нечеткие подмножества A и B равны (обозначение: $A=B$), если $\forall x \in E \mu_A(x) = \mu_B(x)$.

3. *Дополнение.*

Нечеткое подмножество $\bar{A}$ называется дополнением нечеткого подмножества A , если $\forall x \in E \mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

4. *Пересечением* нечетких подмножеств A и B называется нечеткое подмножество $A \cap B$ с функцией принадлежности:

$$\forall x \in E \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)).$$

5. *Объединением* двух нечетких подмножеств A и B называется нечеткое подмножество $A \cup B$ с функцией принадлежности:

$$\forall x \in E \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)).$$

6. *Разностью* нечетких подмножеств A и B называется нечеткое подмножество $A - B$, определяемое в виде: $A - B = A \cap \bar{B}$.

7. *Дизъюнктивной суммой* нечетких подмножеств A и B называется нечеткое подмножество $A \oplus B$, определяемое следующим образом: $A \oplus B = (A \cap \bar{B}) \cup (B \cap \bar{A})$.

8. *Алгебраическим произведением* нечетких подмножеств A и B называется нечеткое подмножество $A \bullet B$ с функцией принадлежности:

$$\forall x \in E \mu_{A \bullet B}(x) = \mu_A(x) \cdot \mu_B(x).$$

9. *Алгебраической суммой* двух нечетких подмножеств A и B называется нечеткое подмножество $A \hat{+} B$ с функцией принадлежности:

$$\forall x \in E \mu_{A \hat{+} B} = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x).$$

Пример. Рассмотрим два нечётких подмножества:

$$A = \{(x_1 | 0.2), (x_2 | 0.7), (x_3 | 1), (x_4 | 0), (x_5 | 0.5)\}$$

$$B = \{(x_1 | 0.5), (x_2 | 0.3), (x_3 | 1), (x_4 | 0.1), (x_5 | 0.5)\}.$$

Применим к этим подмножествам основные рассмотренные операции:

$$A \cap B = \{(x_1 | 0.2), (x_2 | 0.3), (x_3 | 1), (x_4 | 0), (x_5 | 0.5)\}.$$

$$A \cup B = \{(x_1 | 0.5), (x_2 | 0.7), (x_3 | 1), (x_4 | 0.1), (x_5 | 0.5)\}.$$

$$\bar{A} = \{(x_1 | 0.8), (x_2 | 0.3), (x_3 | 0), (x_4 | 1), (x_5 | 0.5)\}.$$

$$\bar{B} = \{(x_1 | 0.5), (x_2 | 0.7), (x_3 | 0), (x_4 | 0.9), (x_5 | 0.5)\}.$$

$$A - B = A \cap \bar{B} = \{(x_1 | 0.2), (x_2 | 0.7), (x_3 | 0), (x_4 | 0), (x_5 | 0.5)\}.$$

$$B - A = B \cap \bar{A} = \{(x_1 | 0.5), (x_2 | 0.3), (x_3 | 0), (x_4 | 0.1), (x_5 | 0.5)\}.$$

$$A \oplus B = (A \cap \bar{B}) \cup (B \cap \bar{A}) = \{(x_1 | 0.5), (x_2 | 0.7), (x_3 | 0), (x_4 | 0.1), (x_5 | 0.5)\}.$$

$$A \bullet B = \{(x_1 | 0.1), (x_2 | 0.21), (x_3 | 1), (x_4 | 0), (x_5 | 0.25)\}.$$

$$A \hat{+} B = \{(x_1 | 0.6), (x_2 | 0.79), (x_3 | 1), (x_4 | 0.1), (x_5 | 0.75)\}.$$

Пусть $\alpha \in [0, 1]$. *Подмножеством уровня α* нечеткого множества A называется обычное подмножество, определенное следующим образом:

$$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\}.$$

Если $\alpha = 0.5$, то множество $\underline{A} = A_{0.5}$ называется обычным множеством, ближайшим к данному нечеткому.

Если $\alpha_2 \geq \alpha_1$, то $A_{\alpha_1} \subset A_{\alpha_2}$.

Пример. Для нечеткого множества $A = \{(x_1 | 0.2), (x_2 | 0), (x_3 | 0.7), (x_4 | 0.8)\}$
 $A_{0.1} = \{x_1, x_3, x_4\}$, $A_{0.5} = \underline{A} = \{x_3, x_4\}$.

Теорема о декомпозиции. Любое нечеткое подмножество A можно разложить следующим образом: $A = \max_{\alpha_i} (\alpha_1 A_{\alpha_1}; \alpha_2 A_{\alpha_2} \dots \alpha_n A_{\alpha_n})$ $0 < \alpha_i < 1$.

Например, пусть $A = \{(x_1 | 0.2), (x_2 | 0), (x_3 | 0.5), (x_4 | 1), (x_5 | 0.7)\}$.

$A = \max(0.2 \{(x_1 | 1), (x_2 | 0), (x_3 | 1), (x_4 | 1), (x_5 | 1)\};$

$0.5 \{(x_1 | 0), (x_2 | 0), (x_3 | 1), (x_4 | 1), (x_5 | 1)\};$

$1 \{(x_1 | 0), (x_2 | 0), (x_3 | 0), (x_4 | 1), (x_5 | 0)\};$

$0.7 \{(x_1 | 0), (x_2 | 0), (x_3 | 0), (x_4 | 1), (x_5 | 1)\})$.

Расстояния между нечеткими множествами

1. $d(A, B) = \sum_{i=1}^n |\mu_A(x_i) - \mu_B(x_i)|$, $0 \leq d(A, B) \leq n$ - расстояние Хемминга или линейное расстояние.

2. $\sigma(A, B) = \frac{d(A, B)}{n}$ - относительное расстояние Хемминга.

3. $e(A, B) = \sqrt{\sum_{i=1}^n (\mu_A(x_i) - \mu_B(x_i))^2}$, $0 \leq e(A, B) \leq \sqrt{n}$ - Евклидово, или квадратичное расстояние;

4. $\varepsilon(A, B) = \frac{e(A, B)}{\sqrt{n}}$ - относительное Евклидово расстояние.

1.3. Основы теории отношений

N -местным (n -арным) отношением на множестве A называется подмножество R его n -й декартовой степени, т.е. $R \subseteq A^n$. При этом множество A называется носителем данного отношения. Говорят, что элементы $a_1, a_2, \dots, a_n$ находятся в отношении R , если упорядоченный набор $(a_1, a_2, \dots, a_n) \in R$. Одноместное отношение называется унарным и соответствует подмножеству множества A .

Двуместным или бинарным отношением на множестве A называется подмножество его квадрата $R \subseteq A^2$. Бинарным отношением между множествами A и B называется подмножество $R \subseteq A \times B$. Для обозначения принадлежности упорядоченной пара (a_1, a_2) отношению R используется запись:

$(a_1, a_2) \in R$ или $a_1 R a_2$.

Пример.

1. Пусть дано множество $A=\{1, 2, 3\}$. Определим на нем одно из возможных бинарных отношений.

$$A^2=\{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}.$$

$$R\subseteq A^2=\{(1, 2), (1, 3), (2, 3)\}.$$

2. Отношение равенства на множестве натуральных чисел - это бинарное отношение:

$$R=\{(a, b) \mid a\in\mathbb{N}, b\in\mathbb{N}, a=b\}.$$

Обратным отношением для отношения R называется отношение

$$R^{-1}=\{(b, a) \mid (a, b)\in R\}.$$

Таким образом, в обратном отношении элементы всех упорядоченных пар меняются местами.

Отношения можно задавать:

1. С использованием любого способа задания множеств.
2. С использованием матрицы.
3. С использованием графа.

Матрицей бинарного отношения R называется квадратная матрица R порядка n , элементы которой задаются следующим образом:

$$r_{ij} = \begin{cases} 1, & \text{если } (a_i, a_j) \in R, \\ 0, & \text{если } (a_i, a_j) \notin R. \end{cases}$$

Пример.

Для отношения $R=\{(1, 2), (2, 3), (3, 1), (3, 2)\}$ матрица будет иметь вид:

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

Каждому бинарному отношению можно поставить в соответствие граф $G=(X, U)$, представляющий собой совокупность множеств X и U , где X - множество вершин графа, а U - множество дуг, соединяющих все или некоторые из этих вершин. При этом вершины a_i, a_j соединяются дугой, если $(a_i, a_j)\in R$.

Например, граф рассматриваемого бинарного отношения имеет вид (рисунки 1.3):

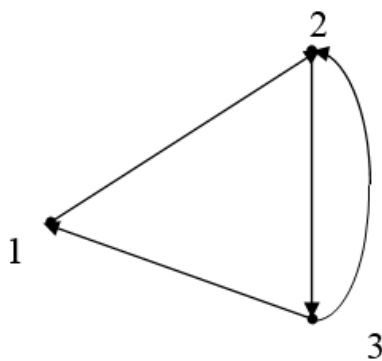


Рис. 1.3. Граф бинарного отношения

Операции над бинарными отношениями

Над бинарными отношениями такие же операции, как и над множествами: пересечение, объединение, разность, симметрическая разность, дополнение. При этом в качестве элементов используются упорядоченные пары.

Пример.

$$R_1 = \{(1, 2), (2, 3), (3, 1), (3, 2)\};$$

$$R_2 = \{(1, 1), (2, 1), (2, 3), (3, 3)\};$$

$$R_1 \cup R_2 = \{(1, 1), (1, 2), (2, 1), (2, 3), (3, 1), (3, 2), (3, 3)\};$$

$$R_1 \cap R_2 = \{(2, 3)\};$$

$$R_1 \setminus R_2 = \{(1, 2), (3, 1), (3, 2)\};$$

$$R_2 \setminus R_1 = \{(1, 1), (2, 1), (3, 3)\};$$

$$R_1 + R_2 = \{(1, 2), (3, 1), (3, 2), (1, 1), (2, 1), (3, 3)\}.$$

Композицией отношений R_1 и R_2 называется отношение

$$R_1 \bullet R_2 = \{(a, b) \mid \exists c: (a, c) \in R_1, (c, b) \in R_2\}.$$

Пример.

$$R_1 = \{(1, 2), (3, 4)\}.$$

$$R_2 = \{(2, 3), (4, 5)\}.$$

$$R_1 \bullet R_2 = \{(1, 3), (3, 5)\}.$$

Для любых бинарных отношений $(R^{-1})^{-1} = R$ и $R_1 \bullet R_2 = R_2^{-1} \bullet R_1^{-1}$.

Свойства бинарных отношений

Пусть на множестве A задано бинарное отношение R .

1. Бинарное отношение R называется *рефлексивным*, если

$$\forall a \in A \quad (a, a) \in R.$$

Матрица рефлексивного бинарного отношения содержит единицы на главной диагонали, а в графе рефлексивного бинарного отношения каждая вершина имеет петлю.

2. Бинарное отношение R называется *антирефлексивным*, если

$$\forall a \in A \quad (a, a) \notin R.$$

Матрица антирефлексивного бинарного отношения содержит нули на главной диагонали, а граф такого отношения не имеет петель.

3. Бинарное отношение называется *симметричным*, если

$$\forall a, b \in A \quad ((a, b) \in R \rightarrow (b, a) \in R).$$

Матрица симметричного бинарного отношения симметрична относительно главной диагонали. В графе симметричного бинарного отношения все вершины соединены парой противоположно направленных дуг.

4. Бинарное отношение называется *антисимметричным*, если

$$\forall a, b \in A, \quad ((a, b) \in R \text{ и } (b, a) \in R) \rightarrow a = b.$$

Матрица антисимметричного бинарного отношения не симметрична относительно главной диагонали, а в графе отсутствуют противоположно направленные дуги.

5. Бинарное отношение называется *транзитивным*, если

$$\forall a, b, c \in A \quad ((a, c) \in R \text{ и } (c, b) \in R) \rightarrow (a, b) \in R.$$

Примеры.

1. Отношение равенства на множестве целых чисел

$$R = \{(a, b) \mid a, b \in \mathbb{Z}, a = b\}$$

рефлексивно, симметрично и транзитивно.

2. Определим свойства отношения $R = \{(a, b) \mid a, b \in \mathbb{N}, a/b \in \mathbb{N}\}$:

- рефлексивно, т.к. $a/a \in \mathbb{N}$;
- антисимметрично: $a/b \in \mathbb{N}, b/a \notin \mathbb{N}$;
- транзитивно:
 $a/b = k_1 \in \mathbb{N}, b/c = k_2 \in \mathbb{N}, a/c = k_1 \cdot k_2 \in \mathbb{N}.$

Специальные бинарные отношения.

Отношением эквивалентности на множестве A называется бинарное отношение, обладающее свойствами рефлексивности, симметричности и транзитивности. Это обозначается $a \sim_R b$ (a эквивалентно b по отношению R). Иногда символ R в записи опускают.

Примеры. Отношениями эквивалентности являются:

- отношение равенства на любом множестве;
- отношение параллельности прямых;

Бинарное отношение на множестве A называется отношением *строгого порядка*, если оно обладает свойствами антирефлексивности, антисимметричности и транзитивности. Отношение строгого порядка обозначается следующим образом: $a \prec_R b$ (a меньше b по отношению R).

Бинарное отношение на множестве A называется отношением *нестрогого (частичного) порядка*, если оно обладает свойствами рефлексивности, антисимметричности и транзитивности. Обозначение: $a \leq_R b$ (a меньше или равно b по отношению R).

Отношение строгого и нестрогого порядка называют *отношениями упорядоченности* или *отношениями порядка*.

Примеры.

1. Отношение "меньше" ($<$) - это отношение строгого порядка.

2. Отношение "меньше или равно" ($\leq$)- отношение частичного порядка.

3. Пусть на множестве подмножеств некоторого универсального множества задано отношение "быть подмножеством". Тогда это отношение рефлексивно, так как $A \subseteq A$. Отношение антисимметричное, так как если $A \subseteq B$, то $B \not\subseteq A$, (условие "если $A \subseteq B$, то $B \subseteq A$ " выполняется только если $B=A$). Отношение транзитивное, так как если $A \subseteq B$ и $B \subseteq C$, то $A \subseteq C$. Следовательно, это отношение частичного порядка.

Если между элементами a и b установлено отношение порядка, они называются сравнимыми, иначе - несравнимыми. Отношение порядка на множестве A называется отношением *линейного порядка*, если любые два элемента множества A сравнимы.

Рассмотрим непустое конечное множество A , на котором задано отношение частичного порядка. Говорят, что элемент b покрывает элемент a , если $a \leq b$ и не существует такого элемента c , что $a \leq c \leq b$.

Любое частично упорядоченное множество можно представить в виде схемы, в которой каждый элемент изображается точкой на плоскости, и если b покрывает a , то точка a и точка b соединяются отрезком (или дугой), причем точка b располагается выше точки a . Такие схемы называются *диаграммами Хассе*.

Пусть задано множество $M=\{b, c, e\}$. Изобразим в виде диаграммы Хассе на множестве $P(M)$ отношение "быть подмножеством" (рисунок 1.4).

Для построения диаграммы Хассе предварительно запишем все подмножества множества M :

$$\{\emptyset, \{b\}, \{c\}, \{e\}, \{b, c\}, \{b, e\}, \{c, e\}, \{b, c, e\}\}.$$

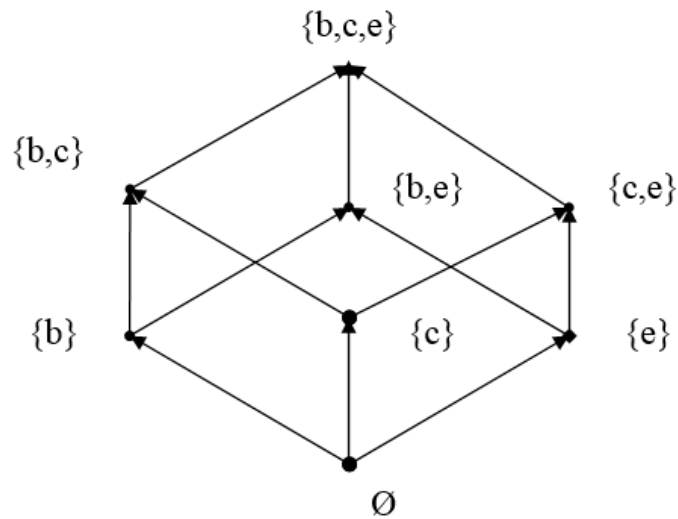


Рис. 1.4. Диаграмма Хассе для отношения «быть подмножеством»

1.4. Нечеткие отношения

Нечетким отношением R на множестве X называется нечеткое подмножество декартового произведения $X \times X$, характеризующееся функцией принадлежности $\mu_R \in [0, 1]$.

Если множество X конечное, то функция принадлежности μ_R представляет собой квадратную матрицу, элемент которой с индексом ij означает степень выполнения отношения для пары (x_i, x_j) .

Носителем нечеткого отношения R называется обычное отношение на множестве X , связывающее все пары (x_i, x_j) , для которых степень выполнения данного нечеткого отношения $\neq 0$:

$$S(R) = \{(x_i, x_j) \mid \mu_R(x_i, x_j) \neq 0\}.$$

Матрицу носителя нечеткого отношения можно получить, заменив в матрице исходного нечеткого отношения единицами все ненулевые элементы.

Отношением уровня α нечеткого отношения R на множестве X называется обычное отношение на множестве X , связывающее те пары (x_i, x_j) , для которых степень выполнения данного нечеткого отношения $\geq \alpha$:

$$R_\alpha = \{(x_i, x_j) \mid \mu_R(x_i, x_j) \geq \alpha\}.$$

Матрицу для отношения уровня α можно получить, заменив в матрице нечеткого отношения R единицами все элементы, значение которых $\geq \alpha$ и нулями все остальные элементы.

Обычное отношение, ближайшее к данному нечеткому, $\underline{R}$ определяется следующим образом:

$$\mu_{\underline{R}}(x_i, x_j) = \begin{cases} 1, \mu_R(x_i, x_j) \geq 0.5, \\ 0, \mu_R(x_i, x_j) < 0.5. \end{cases}$$

Пример. Рассмотрим множество $X = \{x_1, x_2, x_3\}$. Определим на этом множестве нечеткое отношение

$$R = \{((x_1, x_1) | 0), ((x_1, x_2) | 0), ((x_1, x_3) | 0.1), ((x_2, x_1) | 0), ((x_2, x_2) | 0.2), ((x_2, x_3) | 0), ((x_3, x_1) | 0.4), ((x_3, x_2) | 0.4), ((x_3, x_3) | 0.5)\}.$$

Матрица нечеткого отношения:
$$\begin{pmatrix} 0 & 0 & 0.1 \\ 0 & 0.8 & 0 \\ 0.4 & 0.4 & 0.5 \end{pmatrix}.$$

Носитель данного нечеткого отношения:

$$S(R) = \{(x_1, x_3), (x_2, x_2), (x_3, x_1), (x_3, x_2), (x_3, x_3)\}.$$

Матрица носителя:
$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

Отношение, ближайшее к данному нечеткому:

$$\underline{R} = \{(x_2, x_2) (x_3, x_3)\}.$$

Матрица отношения $\underline{R}$:
$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Операции над нечеткими отношениями

Над нечеткими отношениями определены те же операции, что и над нечеткими множествами.

1. *Включение.* Говорят, что отношение R включено в отношение Q ($R \subset Q$), если $\forall (x_i, x_j) \in X \times X \quad \mu_R(x_i, x_j) \leq \mu_Q(x_i, x_j)$.

2. *Равенство.* Нечеткие отношения R и Q равны между собой ($R=Q$), если их функции принадлежности равны: $\forall (x_i, x_j) \in X \times X \quad \mu_R(x_i, x_j) = \mu_Q(x_i, x_j)$.

3. *Дополнением* нечеткого отношения R называется нечеткое отношение $\overline{R}$ такое, что $\forall (x_i, x_j) \in X \times X, \mu_{\overline{R}}(x_i, x_j) = 1 - \mu_R(x_i, x_j)$.

4. *Пересечением* двух нечетких отношений R и Q называется нечеткое отношение $R \cap Q$, определяемое следующим образом:

$$\forall (x_i, x_j) \in X \times X \quad \mu_{R \cap Q}(x_i, x_j) = \min(\mu_R(x_i, x_j), \mu_Q(x_i, x_j)).$$

5. *Объединением* двух нечетких отношений R и Q называется нечеткое отношение $R \cup Q$, определяемое следующим образом:

$$\forall (x_i, x_j) \in X \times X \quad \mu_{R \cup Q}(x_i, x_j) = \max(\mu_R(x_i, x_j), \mu_Q(x_i, x_j)).$$

6. *Дизъюнктивной суммой* нечетких отношений R и Q называется нечеткое отношение $R \oplus Q$, определяемое следующим образом:

$$\forall (x_i, x_j) \in X \times X, R \oplus Q = (R \cap \bar{Q}) \cup (Q \cap \bar{R}).$$

7. *Разностью* двух нечетких отношений R и Q называется нечеткое отношение $R - Q$, определяемое следующим образом:

$$\forall (x_i, x_j) \in X \times X, R - Q = (R \cap \bar{Q}).$$

8. *Алгебраическим произведением* нечетких отношений R и Q называется нечеткое отношение $R \cdot Q$, определяемое следующим образом:

$$\forall (x_i, x_j) \in X \times X \quad \mu_{R \cdot Q}(x_i, x_j) = \mu_R(x_i, x_j) \cdot \mu_Q(x_i, x_j).$$

9. *Алгебраической суммой* нечетких отношений R и Q называется нечеткое отношение $R \hat{+} Q$, определяемое следующим образом:

$$\forall (x_i, x_j) \in X \times X, \mu_{R \hat{+} Q}(x_i, x_j) = \mu_R(x_i, x_j) + \mu_Q(x_i, x_j) - \mu_R(x_i, x_j) \cdot \mu_Q(x_i, x_j).$$

Пример.

$$R = \begin{pmatrix} 0 & 0.3 \\ 0.7 & 1 \end{pmatrix}, \quad Q = \begin{pmatrix} 0.2 & 0.5 \\ 0.4 & 0 \end{pmatrix},$$

$$R \cup Q = \begin{pmatrix} 0.2 & 0.5 \\ 0.7 & 1 \end{pmatrix}, \quad R \cap Q = \begin{pmatrix} 0 & 0.3 \\ 0.4 & 0 \end{pmatrix}, \quad \bar{R} = \begin{pmatrix} 1 & 0.7 \\ 0.3 & 0 \end{pmatrix}, \quad \bar{Q} = \begin{pmatrix} 0.8 & 0.5 \\ 0.6 & 1 \end{pmatrix},$$

$$R - Q = R \cap \bar{Q} = \begin{pmatrix} 0 & 0.3 \\ 0.6 & 1 \end{pmatrix}, \quad R \oplus Q = (R \cap \bar{Q}) \cup (Q \cap \bar{R}) = \begin{pmatrix} 0.2 & 0.5 \\ 0.6 & 1 \end{pmatrix},$$

$$R \cdot Q = (R \cap \bar{Q}) = \begin{pmatrix} 0.2 & 0.5 \\ 0.3 & 0 \end{pmatrix}, \quad R \bullet Q = \begin{pmatrix} 0 & 0.15 \\ 0.28 & 0 \end{pmatrix}, \quad R \hat{+} Q = \begin{pmatrix} 0.2 & 0.65 \\ 0.82 & 1 \end{pmatrix}.$$

Кроме того, над нечеткими отношениями можно определить несколько операций, которые отсутствуют у нечетких множеств:

1. *max-min композиция* или *максимная композиция*

$$\mu_{R \circ Q}(x, y) = \max_{z \in X} \min\{\mu_R(x, z), \mu_Q(z, y)\}.$$

2. *max-• или максумультипликативная композиция*

$$\mu_{R \bullet Q}(x, y) = \max_{z \in X} \{\mu_R(x, z) \cdot \mu_Q(z, y)\}.$$

3. *min-max композиция* или *минмаксная композиция*

$$\mu_{R*Q}(x, y) = \min_{z \in X} \max\{\mu_R(x, z), \mu_Q(z, y)\}.$$

Пример.

$$R = \begin{matrix} & \begin{matrix} z_1 & z_2 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \end{matrix} & \begin{pmatrix} 0 & 0.3 \\ 0.7 & 1 \end{pmatrix} \end{matrix}, \quad Q = \begin{matrix} & \begin{matrix} y_1 & y_2 \end{matrix} \\ \begin{matrix} z_1 \\ z_2 \end{matrix} & \begin{pmatrix} 0.2 & 0.5 \\ 0.4 & 0 \end{pmatrix} \end{matrix}.$$

$$1) R \circ Q = \begin{pmatrix} 0.3 & 0 \\ 0.4 & 0.5 \end{pmatrix} - \text{максиминная композиция.}$$

$$2) R \bullet Q = \begin{pmatrix} 0.12 & 0 \\ 0.4 & 0.35 \end{pmatrix} - \text{максмультипликативная композиция.}$$

$$3) R * Q = \begin{pmatrix} 0.2 & 0.3 \\ 0.7 & 0.7 \end{pmatrix} - \text{минмаксная композиция.}$$

Свойства нечетких отношений

Пусть R нечеткое бинарное отношение, заданное на множестве X .

Нечеткое отношение R называется *рефлексивным*, если

$$\forall x \in X \quad \mu_R(x, x) = 1.$$

Нечеткое отношение называется *антирефлексивным*, если

$$\forall x \in X \quad \mu_R(x, x) \neq 1.$$

Нечеткое отношение называется *симметричным*, если

$$\forall (x_i, x_j) \in X \times X \quad \mu_R(x_i, x_j) = \mu_R(x_j, x_i).$$

Нечеткое отношение называется *антисимметричным*, если

$$\forall (x_i, x_j) \in X \times X \quad \mu_R(x_i, x_j) \neq \mu_R(x_j, x_i) \quad \text{или} \quad \mu_R(x_i, x_j) = \mu_R(x_j, x_i) = 0.$$

Совершенным антисимметричным отношением называется отношение, определяемое следующим образом:

$$\forall (x_i, x_j) \in X \times X \quad \mu_R(x_i, x_j) > 0, \text{ если } \mu_R(x_j, x_i) = 0.$$

Примеры.

$$R = \begin{pmatrix} 0.3 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.8 \end{pmatrix} - \text{антирефлексивное, симметричное отношение.}$$

$$R = \begin{pmatrix} 0.1 & 0.2 & 0 \\ 0 & 0.1 & 0.3 \\ 0.5 & 0 & 0.1 \end{pmatrix} - \text{антирефлексивное, совершенное антисимметричное.}$$

Пусть R нечеткое отношение на множестве X . Нечеткое отношение обладает свойством *max-min транзитивности*, если

$$\mu_R(x, y) \geq \max_z \min\{\mu_R(x, z), \mu_R(z, y)\}.$$

Это означает, что $R \circ R \subseteq R$

Нечеткое отношение обладает свойством *min-max транзитивности*, если

$$\mu_R(x, y) \geq \min_z \max\{\mu_R(x, z), \mu_R(z, y)\}, \text{ т.е., } R * R \subseteq R.$$

Пример. Проверим транзитивность отношения $R = \begin{pmatrix} 0.8 & 1 \\ 0 & 0.4 \end{pmatrix}$.

Проверим транзитивность:

$$R \circ R = \begin{pmatrix} 0.8 & 0.8 \\ 0 & 0.4 \end{pmatrix} \subseteq R, \text{ следовательно отношение обладает свойством}$$

max-min транзитивности.

$$R * R = \begin{pmatrix} 0.8 & 1 \\ 0.4 & 0.4 \end{pmatrix} \not\subseteq R, \text{ следовательно, отношение не обладает свойством}$$

min-max транзитивности.

Специальные типы нечетких отношений

Специальные типы нечетких отношений можно кратко представить следующим образом (таблица 1.1):

Таблица 1.1

Специальные типы нечетких отношений и их свойства

Отношения	Свойства					
	Рефлек-сивность	Антире-флек-сивность	Симмет-рич-ность	Анти-симмет-рич-ность	min-max транзи-тивность	Min-max транзи-тивность
Предпорядок	+					+
Строгий порядок		+		+		+
Нестрогий порядок	+			+		+
Подобие	+		+			+
Сходство	+		+			
Несходство		+	+			
Различие		+	+		+	

Упражнения

1. Доказать:

$$A \setminus (A \cap B) = A \setminus B;$$

$$A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C);$$

$$(A \setminus B) \setminus C = (A \setminus C) \setminus (B \setminus C);$$

$$A + (A \cap B) = A \setminus B;$$

$$(A + B) \cup (A \cap B) = A \cup B;$$

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

$$(A + B) \setminus A = B \setminus A;$$

$$(A + B) + C = A + (B + C).$$

2. Упростить выражения:

$$\frac{((A \setminus C) \setminus (B \setminus C)) \cap B}{(A \setminus B) \cap (\overline{A} \cup B)};$$

$$(A \setminus B) \cap (\overline{A} \cup B);$$

$$A + (B + (A \cap B));$$

$$A + (A + B);$$

$$(A \setminus (B + A)) + (A \setminus B);$$

$$(A \cap (B + A)) + (\overline{A} \setminus B);$$

$$(A \setminus B) + (B + \overline{A}).$$

3. Дано отношение $R = \{(x, y) \mid x, y \in \mathbb{Z}, (2x^2 - 2y^2) \text{ делится на } 5\}$. Определить его свойства и принадлежность к специальным типам бинарных отношений, если результат деления:

- 1) целое число, отличное от 0;
- 2) неотрицательное целое число;
- 3) натуральное число;
- 4) любое целое число.

4. На множестве целых чисел даны отношения:

- 1) “больше на n ”;
- 2) “меньше на n ”;
- 3) “больше в n раз”;
- 4) “меньше в n раз”.

Определить их свойства и принадлежность к специальным типам бинарных отношений.

2. КОМБИНАТОРИКА

2.1. Основные правила комбинаторики

В основе комбинаторики лежат два правила:

- правило суммы;
- правило произведения.

Пусть B – конечное множество, состоящее из n элементов. Тогда элемент b , который принадлежит множеству B , может быть выбран n способами. Основными правилами комбинаторики являются правило суммы и правило произведения.

1. *Правило суммы.* Если элемент x множества M может быть выбран n способами, а элемент y – m способами, то выбор "либо x , либо y " будет осуществлен $n+m$ способами.

Пример. В студенческой группе 5 “отличников” и 10 “хорошистов”. Поэтому для выбора “отличника” существует 5 способов, а для выбора “хорошиста” – 10 способов. Тогда выбор “отличника” или “хорошиста” по правилу суммы может быть осуществлён $5+10=15$ способами.

2. *Правило произведения.* Если элемент x может быть выбран n способами и после каждого из таких выборов элемент y может быть выбран m способами, то выбор упорядоченной пары (x, y) будет осуществлен $n \cdot m$ способами.

Пример. Пусть на студенческую конференцию нужно отправить одного “отличника” и одного “хорошиста”. Тогда количество способов выбора делегатов по правилу произведения равно $5 \cdot 10 = 150$ (каждому из 5 способов выбора “отличника” соответствует 10 способов выбора “хорошиста”).

2.2. Комбинаторные конфигурации

Набор элементов $(a_1, a_2, \dots, a_r)$, составленный из элементов множества $A = \{a_1 \dots a_n\}$, называется *выборкой* объема r из n элементов, или (n, r) – *выборкой*.

Упорядоченная (n, r) – выборка, в которой элементы могут повторяться, называется (n, r) – *размещением с повторениями*. Если элементы упорядоченной (n, r) – выборки попарно различны, то она называется (n, r) – *размещением без повторений*.

(n, n) – размещения без повторений называются n – *перестановками* или *перестановками из n элементов*.

Неупорядоченная (n, r) – выборка, в которой элементы могут повторяться, называется (n, r) – *сочетанием с повторениями*. Неупорядоченная (n, r) выборка, элементы которой не повторяются, называется (n, r) – *сочетанием без повторений*.

Введём следующие обозначения:

A_n^r - число (n, r) - размещений без повторений;

$\overline{A}_n^r$ - число (n, r) - размещений с повторениями;

C_n^r - число (n, r) - сочетаний без повторений;

$\overline{C}_n^r$ - число (n, r) - сочетаний с повторениями;

P_n - число перестановок из n элементов.

1. Число (n, r) - размещений с повторениями определяется по следующей формуле:

$$\overline{A}_n^r = n^r.$$

Действительно, каждое (n, r) - размещение с повторениями – это упорядоченная последовательность длины r . Каждый член последовательности может быть выбран n способами, следовательно, выбор r элементов может быть осуществлен n^r способами.

Пример. Сколькими способами из букв а, б, в, г, д можно составить слово из 3-х букв, если буквы могут повторяться?

Решение: Количество способов - это число размещений с повторениями из 5 по 3, и оно равно $\overline{A}_n^r = 5^3 = 125$.

2. Число (n, r) размещений без повторений определяется следующим образом:

$$A_n^r = \frac{n!}{(n-r)!}.$$

Действительно, каждое (n, r) - размещение без повторений – упорядоченная последовательность длины r . Первый элемент этой последовательности может быть выбран n способами, второй - $(n-1)$ способом, третий - $(n-2)$ способами и.т.д. Следовательно, число способов выбора r элементов определится в

виде: $A_n^r = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-r+1) = \frac{n!}{(n-r)!}.$

Пример. Сколькими способами из пяти цифр 1, 2, 3, 4, 5 можно составить трехзначное число, чтобы цифры не повторялись?

Решение. Число способов равно $A_5^3 = \frac{5!}{(5-3)!} = 20$.

3. Число перестановок из n элементов определяется следующим образом:
 $P_n = n!$

Действительно, $P_n = A_n^n = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 1 = n!$

Пример. Сколькими способами можно расставить на полке 4 книги?

Решение. Количество способов равно $4!=24$.

4. Число (n, r) - сочетаний без повторений определяется следующим образом:

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

Данная формула следует из того, что каждому (n, r) - сочетанию без повторений соответствует $r!$ размещений без повторений.

Примеры.

1. Сколькими способами из группы студентов, состоящей из 20 человек, можно выбрать 3 делегатов на конференцию?

Решение. В данном случае последовательность выбора роли не играет, поэтому искомое число способов равно количеству сочетаний без повторений из 20 по 3:

$$C_n^r = \frac{20!}{3!(17)!} = 1140.$$

2. В скольких случаях при угадывании 5 номеров из 36 будут правильно выбраны: а) три номера; б) четыре номера; в) пять номеров; г) не менее трех номеров; д) хотя бы один номер.

Решение:

а) три правильных номера из пяти могут быть выбраны C_5^3 способами, оставшиеся два неправильных номера из 31 могут быть выбраны C_{31}^2 способами. По правилу произведения результат равен $C_5^3 \cdot C_{31}^2 = 4650$.

б) аналогично $C_5^4 \cdot C_{31}^1 = 155$.

в) все пять правильных номеров могут быть выбраны 1 способом.

г) по правилу суммы результат будет равен $C_5^3 \cdot C_{31}^2 + C_5^4 \cdot C_{31}^1 + 1 = 4806$.

д) количество способов, при котором будет выбран хотя бы один правильный номер, можно определить, если из общего числа способов выбора 5 номеров из 36 вычесть число способов, при котором не будет выбран ни один правильный номер: $C_{36}^5 - C_{31}^5$.

Для сочетаний без повторений выполняется следующее свойство:

$$C_n^r = C_n^{n-r}.$$

$$\text{Действительно, } C_n^{n-r} = \frac{n!}{(n-r)!(n-n+r)!} = \frac{n!}{(n-r)!r!} = C_n^r.$$

5. Число (n, r) - сочетаний с повторениями определяется следующим образом:

$$\overline{C}_n^r = C_{n+r-1}^r.$$

Пример. Сколько костей домино можно составить из цифр 0,1,2,3,4,5,6?

Решение: Результат равен числу сочетаний с повторениями из 7 по 2: $\overline{C}_7^2$
 $= C_{7+2-1}^2 = \frac{8!}{2!6!} = 28.$

2.3. Урновые схемы

В теории вероятностей рассматриваются комбинаторные схемы, связанные с выбором k шаров из урны с n шарами. При этом возможны следующие варианты:

1. Неупорядоченный выбор с возвращением k шаров из урны с n различными шарами. Число способов выбора в данном случае равно числу неупорядоченных k -выборок с повторением:

$$N = C_{n+k-1}^k = \overline{C}_n^k.$$

2. Неупорядоченный выбор без возвращения k шаров из урны с n различными шарами. Число способов выбора равно количеству неупорядоченных k -выборок без повторения:

$$N = C_n^k.$$

3. Упорядоченный выбор с возвращением k шаров из урны с n различными шарами. Число способов выбора равно количеству упорядоченных k -выборок с повторением:

$$N = \overline{A}_n^k = n^k.$$

4. Упорядоченный выбор без возвращения k шаров из урны с n различными шарами. Число способов выбора определяется как количество упорядоченных k -выборок без повторения:

$$N = A_n^k.$$

Пример.

В урне содержатся 5 красных, 3 синих и 6 зеленых шаров. Из нее без возвращения выбирают 5 шаров, причем порядок выбора не существен. Сколькими способами можно выбрать:

- 1) 3 красных и 2 синих шара;
- 2) все зеленые шары;
- 3) ни одного красного шара;
- 4) не менее 2 синих шаров;

Решение:

- 1) $C_5^3 \cdot C_3^2 = 15$;
- 2) $C_6^5 = 6$;
- 3) $C_9^5 = 126$;
- 4) $C_3^2 \cdot C_{11}^3 + C_{11}^2 = 530$.

2.4. Бином Ньютона

Числа C_n^r называются *биномиальными коэффициентами*. Биномиальные коэффициенты обладают следующим важным свойством, который может использоваться для их определения:

$$C_{n+1}^r = C_n^r + C_n^{r-1}.$$

Для доказательства справедливости данного соотношения распишем его, используя формулы сочетаний без повторов:

$$\begin{aligned} C_n^r + C_n^{r-1} &= \frac{n!}{r!(n-r)!} + \frac{n!}{(r-1)!(n-r+1)!} = \frac{n!}{r \cdot (r-1)!(n-r)!} + \\ &+ \frac{n!}{(r-1)!(n-r)!(n-r+1)} = \frac{n!(n-r+1) + n!r}{r \cdot (r-1)!(n+1-r) \cdot (n-r)!} = \frac{n!(n+1-r+r)}{r!(n+1-r)!} = \\ &= \frac{(n+1)!}{r!(n+1-r)!} = C_{n+1}^r. \end{aligned}$$

Рассмотренное рекуррентное соотношение позволяет построить таблицу для последовательного определения биномиальных коэффициентов, которая называется *треугольником Паскаля*:

n	C_n^r									
0					1					
1				1		1				
2			1		2		1			
3		1		3		3		1		
4		1	4		6		4		1	
5	1		5		10		10		5	1
...	...	...	...	...	...	...	...	...	...	...

Рис.2.1. Треугольник Паскаля

Биномиальные коэффициенты составляют в треугольнике Паскаля строку с номером n . В строке с номером n последовательно слева направо записаны числа $C_n^0, C_n^1, \dots, C_n^n \dots$, причем крайние значения известны для любого n : $C_n^0 = C_n^n = 1$. Каждый биномиальный коэффициент получается путем сложения двух коэффициентов, стоящих над ним (слева и справа).

Для всех действительных чисел a и b и для всех натуральных чисел n справедлива следующая формула :

$$(a + b)^n = \sum_{r=0}^n C_n^r \cdot a^{n-r} \cdot b^r = C_n^0 \cdot a^n \cdot b^0 + C_n^1 \cdot a^{n-1} \cdot b^1 + \dots + C_n^n \cdot a^0 \cdot b^n.$$

Это выражение носит название *бином Ньютона*.

Если заменить b на $-b$, то получится следующее выражение:

$$(a - b)^n = \sum_{r=0}^n (-1)^r \cdot C_n^r \cdot a^{n-r} \cdot b^r.$$

Пример.

$$(a+b)^2 = C_2^0 \cdot a^2 \cdot b^0 + C_2^1 \cdot a^1 \cdot b^1 + C_2^2 \cdot a^0 \cdot b^2 = a^2 + 2ab + b^2.$$

$$(a-b)^3 = C_3^0 \cdot a^3 \cdot b^0 - C_3^1 \cdot a^2 \cdot b^1 + C_3^2 \cdot a^1 \cdot b^2 - C_3^3 \cdot a^0 \cdot b^3 = a^3 - 3a^2b + 3ab^2 - b^3.$$

Основные биномиальные тождества:

$$1. C_n^r = C_n^{n-r}.$$

$$2. C_{n+1}^r = C_n^r + C_n^{r-1}.$$

3. $C_n^0 + C_n^1 + \dots + C_n^n = 2^n$ (получается, если в формуле бинома Ньютона положить $a = b = 1$). Это свойство позволяет определить общее число подмножеств n -элементного множества.

4. $C_n^0 - C_n^1 + C_n^2 - \dots + (-1)^n C_n^n = 0$ (получается, если в формуле бинома Ньютона положить $a = 1, b = -1$).

Полиномиальная формула

Полиномиальная формула используется для вычисления выражений вида $(a_1 + a_2 + \dots + a_k)^n$ и записывается следующим образом:

$$(a_1 + a_2 + \dots + a_k)^n = \sum_{\substack{n_1, \dots, n_k \geq 0 \\ n_1 + \dots + n_k = n}} \frac{n!}{n_1! \dots n_k!} a_1^{n_1} \dots a_k^{n_k}.$$

Здесь суммирование осуществляется по всем решениям уравнения $n_1 + \dots + n_k = n$ в целых неотрицательных числах. При $k=2$ данное выражение превращается в формулу бинома Ньютона:

$$(a_1 + a_2)^n = \sum_{r=0}^n \frac{n!}{r!(n-r)!} \cdot a_1^{n-r} \cdot a_2^r = \sum_{r=0}^n C_n^r \cdot a_1^{n-r} \cdot a_2^r.$$

2.5. Разбиения

Пусть имеется множество A с числом элементов, равным n . Определим число разбиений данного множества на k подмножеств $A_1, A_2, \dots, A_k$ ($k > 1$), таких, что первое подмножество содержит n_1 , второе n_2 , k -е – n_k элементов.

$$\sum_{i=1}^k n_i = n; \quad |A| = n; \quad \bigcup_{i=1}^k A_i = A; \quad A_i \cap A_j = \emptyset \quad \forall i \neq j; \quad i, j = \overline{1, k}.$$

Число указанных разбиений при фиксированных n_i обозначим через

$C_n^{n_1 \dots n_k}$. Эти разбиения можно получить следующим образом: возьмем произвольное n_1 элементное подмножество A_1 множества A . Это можно сделать $C_n^{n_1}$ способами. Среди $n - n_1$ оставшихся элементов возьмем n_2 - элементное подмножество A_2 (это можно сделать $C_{n-n_1}^{n_2}$ способами) и т.д. Общее число способов выбора различных подмножеств $A_1, \dots, A_k$ по правилу произведения равно:

$$C_n^{n_1 \dots n_k} = C_n^{n_1} \cdot C_{n-n_1}^{n_2} \cdot \dots \cdot C_{n-n_1-\dots-n_{k-1}}^{n_k} =$$

$$\frac{n!}{n_1!(n-n_1)!} \cdot \frac{(n-n_1)!}{n_2!(n-n_1-n_2)!} \cdot \dots \cdot \frac{(n-n_1-\dots-n_{k-1})!}{n_k!(n-n_1-\dots-n_{k-1}-n_k)!} = \frac{n!}{n_1! \dots n_k!}$$

Таким образом получена формула определения числа разбиений n -элементного множества на k n_i -элементных подмножеств. Числа $C_n^{n_1 \dots n_k} = \frac{n!}{n_1! \dots n_k!}$ называются *полиномиальными коэффициентами*.

Пример. Сколькими способами можно группу из 20 человек разделить на 4 подгруппы, состоящие из 6, 6, 5, 3 человек?

Решение: $C_{20}^{6,6,5,3} = \frac{20!}{6! \cdot 6! \cdot 5! \cdot 3!}$

Полиномиальные коэффициенты имеют еще одну важную комбинаторную интерпретацию. Полиномиальный коэффициент $C_n^{n_1 \dots n_k}$ равен числу перестановок, которые можно составить из n элементов, среди которых имеются n_1 элемент первого типа, n_2 второго типа, n_k k -го типа:

$$P(n_1, \dots, n_k) = C_n^{n_1 \dots n_k} = \frac{n!}{n_1! \dots n_k!}.$$

Эта формула следует из того, что множество всех $n!$ перестановок распадается на $n_1! \cdot n_2! \cdot \dots \cdot n_k!$ одинаковых перестановок.

Пример. Сколькими способами можно переставить буквы слова "математика"?

Решение: Искомое количество способов равно числу перестановок множества из 10 элементов, среди которых имеются 2 элемента первого типа (две буквы "м"), 3 элемента второго типа (три буквы "а"), 2 элемента третьего типа (две буквы "т"), и по одному элементу остальных трех типов. Оно определяется следующим образом:

$$P(2,3,2,1,1,1) = \frac{10!}{2! \cdot 3! \cdot 2!}$$

Комбинаторика разбиений

Рассмотрим различные варианты комбинаторных задач на определение числа разбиений.

1. Число способов распределения n различных элементов по k различным ячейкам, такое, что в первую ячейку попадет n_1 элемент, в k -ю – n_k элементов. Порядок ячеек не существен.

$$N = C_n^{n_1 \dots n_k} = \frac{n!}{n_1! \dots n_k!}.$$

2. Та же задача при условии, что порядок ячеек не существен.

$$N = \frac{n!}{k!n_1!n_2!\dots n_k!}$$

(так как ячейки можно переставлять $k!$ способами).

3. Число способов распределения n различных элементов по k различным ячейкам при условии, что пустые ячейки допускаются.

$$N = k^n$$

(так как каждый элемент можно распределить k -способами, положив в одну из ячеек).

4. Число способов распределения n различных элементов по k различным ячейкам. Пустые ячейки не допускаются.

$$N = A_n^k \cdot k^{n-k}.$$

При этом сначала в каждую ячейку помещаем по одному элементу A_n^k способами, затем оставшиеся $n-k$ элементов распределяем по k ячейкам произвольным образом.

5. Число способов распределения n одинаковых элементов по k различным ячейкам. Пустые ячейки допускаются.

$$N = C_{n+k-1}^n = C_{n+k-1}^{k-1}.$$

Для доказательства данной формулы разделим множество из n элементов на k частей путём введения $(k-1)$ разделителя. Перемещая разделители, получим искомое число разбиений. Следовательно, искомое число разбиений равно количеству перестановок, которые можно составить из n элементов одного типа и $(k-1)$ элементов другого типа

$$P(n, k-1) = \frac{(n+k-1)!}{n!(k-1)!} = C_{n+k-1}^{k-1}.$$

6. Число способов распределения n одинаковых элементов по k различным ячейкам. Пустые ячейки не допускаются.

$$N = C_{n-1}^{k-1}$$

(сначала помещаем в каждую ячейку по 1 элементу, затем используем приведённую выше формулу): $N = C_{n+k-1-k}^{k-1} = C_{n-1}^{k-1}.$

Пример. 15 человек вошли в лифт на первом этаже девятиэтажного дома. Определить:

1) общее число способов распределения людей по этажам;

2) число способов распределения, при котором на каждом этаже выйдет хотя бы один человек;

3) число способов распределения, при котором на третьем этаже выйдут 5 человек, на пятом – 4, а на седьмом – 6.

Предполагается, что пассажиры могут выходить на восьми этажах.

Решение.

1) $N=8^{15}$ (распределение 15 различных элементов по 8 различным ячейкам, при котором пустые ячейки не допускаются);

2) $N=A_{15}^8 \cdot 8^7$ (распределение 15 различных элементов по 8 различным ячейкам, при котором пустые ячейки допускаются);

3) $N=C_{15}^{5, 6, 4}$.

Пример. Сколькими способами можно распределить 10 одинаковых пирожных между 3 людьми так, чтобы.

1) каждый мог получить любое количество;

2) каждый должен получить хотя бы одно пирожное.

Решение.

1) $C_{n+k-1}^{k-1} = C_{12}^2$.

2) $C_{n-1}^{k-1} = C_9^2$.

2.6. Формула включений и исключений

Пусть A_1 и A_2 – два конечных множества. Тогда если $A_1 \cap A_2 = \emptyset$, то $|A_1 \cup A_2| = |A_1| + |A_2|$.

Если $A_1 \cap A_2 \neq \emptyset$, то $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$. Таким образом, мы выразили количество элементов в объединении двух множеств через количество элементов в их пересечении.

Для произвольного числа $n \geq 2$ конечных множеств $A_1 \dots A_n$ эта формула примет следующий вид: :

$|A_1 \cup \dots \cup A_n| = (|A_1| + \dots + |A_n|) - (|A_1 \cap A_2| + |A_2 \cap A_3| + \dots + |A_{n-1} \cap A_n|) + (|A_1 \cap A_2 \cap A_3| + \dots + |A_{n-2} \cap A_{n-1} \cap A_n|) - \dots + (-1)^{n+1} |A_1 \cap \dots \cap A_n|$.

Эта формула называется *формулой включений и исключений*.

Пример. Пусть $A_1 = \{a, b, c\}$, $A_2 = \{b, e, f, g\}$, $A_3 = \{a, b, m, p\}$.

Тогда $|A_1 \cup A_2 \cup A_3| = 3+4+4-1-2-1+1=8$.

Следствие из формулы. Пусть A – конечное множество, $A_1 \dots A_n$ – его подмножества. Тогда

$|A \setminus (A_1 \cup \dots \cup A_n)| = |A| - (|A_1| + \dots + |A_n|) + (|A_1 \cap A_2|) - \dots + (|A_{n-1} \cap A_n|) - \dots + (-1)^{n+1} |A_1 \cap \dots \cap A_n|$.

Действительно,

$$(A \setminus (A_1 \cup \dots \cup A_n)) \cup (A_1 \cup \dots \cup A_n) = A;$$

$$(A \setminus (A_1 \cup \dots \cup A_n)) \cap (A_1 \cup \dots \cup A_n) = \emptyset.$$

Поэтому $|A \setminus (A_1 \cup \dots \cup A_n)| + |A_1 \cup \dots \cup A_n| = |A|$, откуда и следует данная формула.

Формула включений и исключений имеет еще одну интерпретацию. Пусть A – конечное множество из N элементов, $\alpha_1 \dots \alpha_n$ – некоторые свойства, которыми могут обладать или не обладать элементы множества A .

Введем обозначения:

$N(\alpha_i)$ – число элементов множества A , обладающих свойствами α_i ;

$N(\alpha_i, \alpha_j)$ – число элементов, обладающих свойствами α_i и α_j ;

α'_i – элемент, не обладающий свойством α_i ;

$N(\alpha'_1 \dots \alpha'_n) = N_0$ – число элементов, не обладающих ни одним свойством.

Тогда $N(\alpha'_1 \dots \alpha'_n) = N - N(\alpha_1) - \dots - N(\alpha_n) + N(\alpha_1, \alpha_2) + \dots + N(\alpha_{n-1}, \alpha_n) - N(\alpha_1, \alpha_2, \alpha_3) - \dots - N(\alpha_{n-2}, \alpha_{n-1}, \alpha_n) + \dots + (-1)^n N(\alpha_1, \alpha_2, \dots, \alpha_n)$.

Пример. На курсе учатся 67 студентов. 27 из них знают язык программирования Java, 15 студентов знают C#, 10 студентов знают Python, 7 – Java и C#, 5 – Java и Python, 3 – C# и Python, 2 – все три языка. Необходимо определить:

1) сколько студентов не знают ни одного из перечисленных языков программирования (N_0);

2) сколько студентов знают только один язык программирования:

- только Java (N_1);

- только C# (N_2);

- только Python (N_3);

3) сколько человек знают только два языка:

- Java и C# (N_{12});

- Java и Python (N_{13});

- C# и Python (N_{23});

Решение.

1) $N_0 = 67 - 27 - 15 - 10 + 5 + 7 + 3 - 2 = 28$.

2) $N_1 = 27 - 5 - 7 + 2 = 17$.

$N_2 = 15 - 7 - 3 + 2 = 7$.

$N_3 = 10 - 5 - 3 + 2 = 4$.

3) $N_{12} = 7 - 2 = 5$.

$N_{13} = 5 - 2 = 3$.

$N_{23} = 3 - 2 = 1$.

Очевидно выполнение равенства:

$$N_0 + N_1 + N_2 + N_3 + N_{12} + N_{13} + N_{23} + N_{123} = N,$$

где N – общее число студентов на курсе.

Сделаем проверку:

$$28 + 17 + 7 + 4 + 5 + 3 + 1 + 2 = 67$$

Упражнения

1. Цифры 1,2,3,4,5,6,7,8,9 записываются в случайном порядке. Определить
 - общее количество всех чисел, которые можно из них составить;
 - количество чисел, в которых цифры записываются в порядке возрастания;
 - количество чисел, которые начинаются с цифры 5, а цифры 1 и 2 следуют друг за другом в порядке возрастания;
 - количество чисел, в которых на четных местах в числе стоят четные цифры.
2. Из телефонной книги случайным образом выбирается номер, состоящий из 6 цифр. Считая комбинации цифр в телефонном номере равновероятными, найти:
 - общее число возможных способов выбора номера;
 - число способов выбора номера, начинающегося с цифры 5;
 - число способов выбора номера, не содержащего цифр 3 и 6;
 - число способов выбора номера, в котором три последние цифры одинаковы.
3. Из 10 букв а, б, в, г, д, е, ж, з, и, к составляется 5-буквенное слово, в котором буквы не повторяются. Определить:
 - общее число способов составления слов;
 - число способов, при котором все гласные буквы входят в слово;
 - число способов, при котором слово содержит 2 гласных и 3 согласных;
 - число способов, при котором слово начинается с буквы “и” и имеет слог “ва”;
 - число способов, при котором , буквы “г” и “д” входят в состав слова и не располагаются рядом;
 - число способов, при котором слово начинается с гласной буквы и заканчивается согласной;
 - число способов, при котором слово содержит хотя бы одну гласную букву.
4. Сколько можно составить 4-значных чисел из цифр 1 и 2?
5. Рассматривается число 1352142131.
 - сколькими способами можно переставить цифры в этом числе?
 - сколькими способами можно переставить цифры в этом числе, чтобы цифры “3” располагались рядом?
 - сколькими способами можно переставить цифры в числе, чтобы оно начиналось и заканчивалось цифрой “1”?
6. Сколькими способами можно разделить 30 разных открыток между 3 людьми так, чтобы:
 - каждый мог получить любое количество;

- каждый должен получить хотя бы по одной;
- один получит 10 открыток, другой 15, третий 5.

7. Сколькими способами можно переставить буквы слова “обороноспособность”, чтобы:

- слово начиналось с буквы “т” и имело слоги “ро” и “но”;
- в слове буквы “н” располагаются рядом;
- буквы “б” в слове не располагаются рядом.

8. В институтскую библиотеку пришли несколько студентов, каждый из которых взял хотя бы по одной книге. 10 из них взяли книги по курсу “Дискретная математика”, 8 – по курсу “Высшая математика”, 9 – по курсу “Физика”. 5 человек взяли книги по курсам “Дискретная математика” и “Высшая математика”, 3 – по курсам “Дискретная математика” и “Физика”, 7 – по курсам “Высшая математика” и “Физика”. 2 человека взяли книги по всем трем дисциплинам. Определить:

- сколько студентов было в библиотеке;
- сколько человек взяли книги только по курсу “Дискретная математика”.

9. Студенческая группа из 20 человек сдавала сессию из 3-х экзаменов и каждый студент сдал сессию. 10 человек получили “5”, 8 – “4”. 9 – “3”. 4 человека получили “5” и “4”, 5 – “4” и “3”, 2 – “5” и “3”. Определить:

- сколько человек получили все 3 оценки.
- сколько отличников в группе.
- сколько человек сдали сессию только на “3”.

10. В подъезде дома установлен замок с кодом. Дверь автоматически открывается, если в определенной последовательности набрать 4 цифры из имеющихся десяти. Определить:

- сколько различных комбинаций можно составить?
- сколько можно составить комбинаций, в которых первая и последняя цифры были бы одинаковы?
- сколько можно составить комбинаций, при условии что полученное число является нечетным и начинается с цифры “3”?

11. На кафедре работают 3 профессора и 10 доцентов и 2 ассистента. Сколькими способами можно составить комиссию из 5 человек для проведения госэкзамена при условиях:

- в комиссию должны быть включены все профессора.
- в комиссию должны быть включен хотя бы один профессор.
- в комиссию не должны включаться ассистенты.

12. В гостинице имеются 5 свободных номеров: один двухместный, один трёхместный, один четырёхместный и два одноместных. Сколькими способами можно расселить 11 человек по номерам?

3. ОСНОВЫ ТЕОРИИ ГРАФОВ

3.1. Ориентированные и неориентированные графы

Графом $G=(X,U)$ будем называть совокупность двух конечных множеств: множества вершин $X=\{x_1,...,x_n\}$ и множества ребер (дуг) $U=\{u_1,...,u_m\}$, состоящего из некоторых пар элементов (x_i, x_j) множества X . Геометрически граф может быть представлен в виде рисунка, в котором вершинам соответствуют точки, а ребрам – соединяющие их линии.

Если пары вершин (x_i, x_j) в множестве U являются неупорядоченными (т.е., порядок соединения вершин несущественен), то граф называется *неориентированным* (неорграфом), а пары (x_i, x_j) – *ребрами* этого графа (рис. 3.1, а).

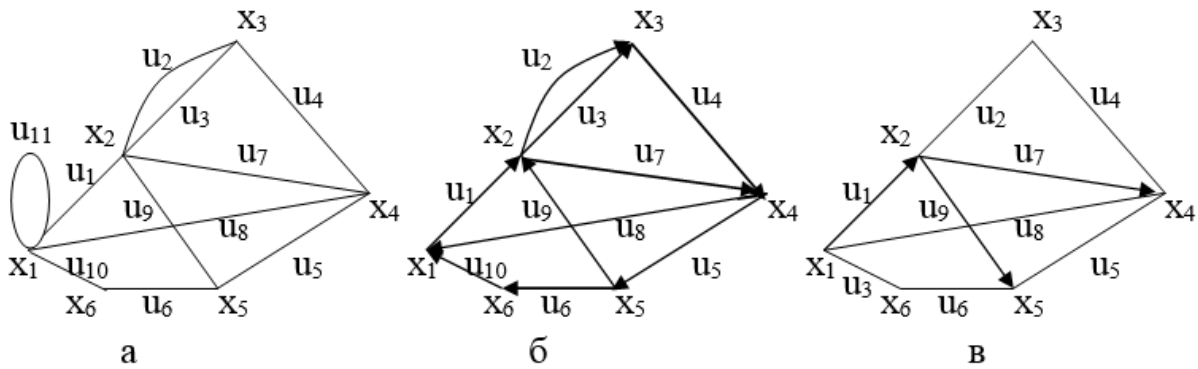


Рис. 3.1. Примеры графов

Если пары вершин (x_i, x_j) в множестве U являются упорядоченными (т.е., порядок соединения вершин существенен), то граф называется *ориентированным* (орграфом), а пары (x_i, x_j) – *дугами* (рис. 3.1, б). Дуги орграфа будем обозначать $\langle x_i, x_j \rangle$. При этом x_i – начальная вершина дуги. При изображении орграфа дуги обозначают стрелками. Графы, в которых имеются и неориентированные ребра, и дуги, называются *смешанными* (рис. 3.1, в). Заметим, что любой неориентированный граф может быть представлен в виде орграфа путем замены каждого его ребра двумя противоположно направленными дугами.

Пара вершин x_i, x_j может соединяться двумя или более ребрами (дугами), которые называют *кратными*. Количество одинаковых ребер (x_i, x_j) (дуг $\langle x_i, x_j \rangle$) называется *кратностью* соответствующего ребра (дуги). *Петлей* называется дуга (ребро), с совпадающими начальной и конечной вершинами.

Граф с петлями и кратными ребрами (дугами) называется *псевдографом*. Граф с кратными ребрами (дугами) и без петель называется *мультиграфом*. Граф, не содержащий петель и кратных ребер, называется *простым* графом.

Так, граф, представленный на рис. 3.1, а является псевдографом, так как содержит петлю (x_1, x_1) и кратное ребро (x_2, x_3) ; граф на рис. 3.1, б является мультиграфом, граф на рис. 3.1, в - простым графом.

Две вершины графа x_i и x_j называются *смежными*, если существует соединяющее их ребро (дуга). Два ребра (дуги) смежны, если они имеют общую вершину. Если ребро (дуга) u_k соединяет вершины x_i и x_j , то говорят, что ребро (дуга) *инцидентно* вершинам x_i и x_j , а вершины x_i и x_j инцидентны ребру (дуге) u_k . Так, на рис. 3.1 а) вершина x_5 инцидентна ребрам u_6, u_9, u_5 , ребра u_{10} и u_6 являются смежными; смежными являются вершины x_1 и x_6 .

Подграфом графа $G=(X,U)$ называется граф $G'=(X',U')$, для которого $X' \subseteq X$, $U' \subseteq U$. *Остовным подграфом* графа $G=(X,U)$ называется граф $G_0=(X, U_0)$, содержащий все вершины графа G . Так, на рис. 3.2, а представлен граф G , на рис. 3.2, б - один из остовных подграфов графа G .

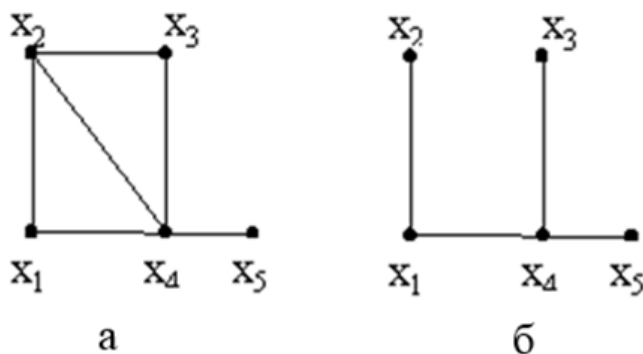


Рис. 3.2. Пример графа и одного из его остовных подграфов

Граф $\bar{G}=(X, \bar{U})$ называется *дополнением* простого графа $G=(X,U)$, если две вершины смежны в $\bar{G}$ тогда и только тогда, когда они не смежны в G . На рис. 3.3а представлен граф G , на рис. 3.3 б - его дополнение.

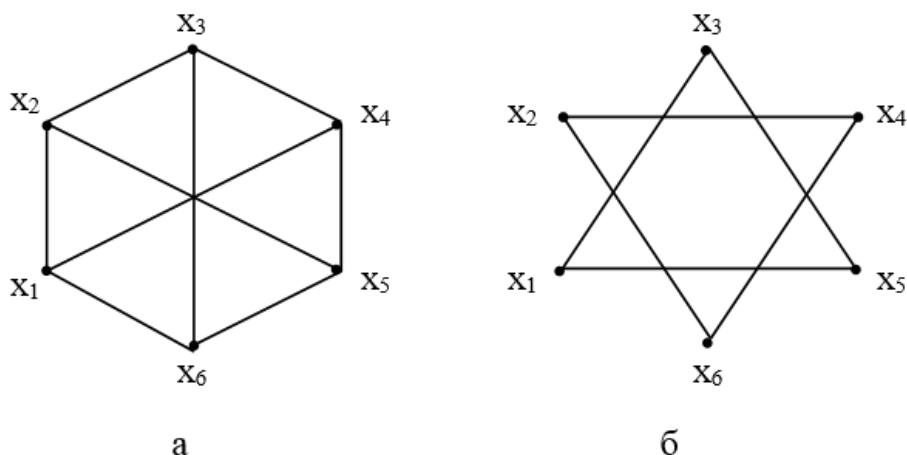


Рис. 3.3. Пример графа и его дополнения

Часто граф задают парой (X, Γ) . При этом для орграфа $\Gamma(x_i)$ - множество вершин $x_j \in X$, для которых в графе существует дуга $\langle x_i, x_j \rangle$; $\Gamma^{-1}(x_i)$ - множество вершин $x_k \in X$, для которых в графе существует дуга $\langle x_k, x_i \rangle$. Для неорграфа $\Gamma(x_i) = \Gamma^{-1}(x_i)$ и означает количество вершин, смежных с вершиной x_i .

Число дуг, исходящих из вершины x_i ориентированного графа, называется *полустепенью исхода* вершины x_i и обозначается $\rho^-(x_i)$. Число дуг, заходящих в вершину x_i , называется *полустепенью захода* вершины x_i и обозначается $\rho^+(x_i)$. Если в вершине x_i имеется петля, к $\rho^-(x_i)$ и $\rho^+(x_i)$ прибавляется 1. Так, для орграфа, представленного на рис. 3.1, б $\rho^-(x_1)=1$; $\rho^+(x_1)=2$; $\rho^-(x_2)=3$; $\rho^+(x_2)=2$. Для любого орграфа с m дугами выполняется равенство

$$\sum_{x_i \in X} \rho^-(x_i) = \sum_{x_i \in X} \rho^+(x_i) = m.$$

В неориентированном графе число ребер, инцидентных данной вершине x_i , называется *степенью* вершины x_i и обозначается $\rho(x_i)$. Вершина графа, имеющая степень 0, называется *изолированной*, а вершина, имеющая степень 1 - *висячей*. Если в вершине x_i имеется петля, к $\rho(x_i)$ прибавляется 2. Для неорграфа на рис. 3.1, а $\rho(x_4)=4$; $\rho(x_1)=5$. Для любого неорграфа с m рёбрами справедливо следующее равенство:

$$\sum_{x_i \in X} \rho(x_i) = 2m.$$

3.2. Матричный способ задания графов

Матрицей смежности неориентированного графа $G=(X, U)$ с n вершинами называется квадратная матрица A порядка n , элементы которой определяются следующим образом:

$$a_{ij} = \begin{cases} 1, & \text{если вершины } x_i \text{ и } x_j \text{ смежны;} \\ 0 & \text{в противном случае.} \end{cases}$$

Для ориентированного графа:

$$a_{ij} = \begin{cases} 1, & \text{если в графе существует дуга } \langle x_i, x_j \rangle; \\ 0 & \text{в противном случае.} \end{cases}$$

В псевдографе $a_{ij} = k$, где k - кратность ребра (x_i, x_j) (дуги $\langle x_i, x_j \rangle$). Необходимо заметить, что если в графе нет петель, диагональные элементы матрицы смежности равны 0.

Рассмотрим неориентированный граф:

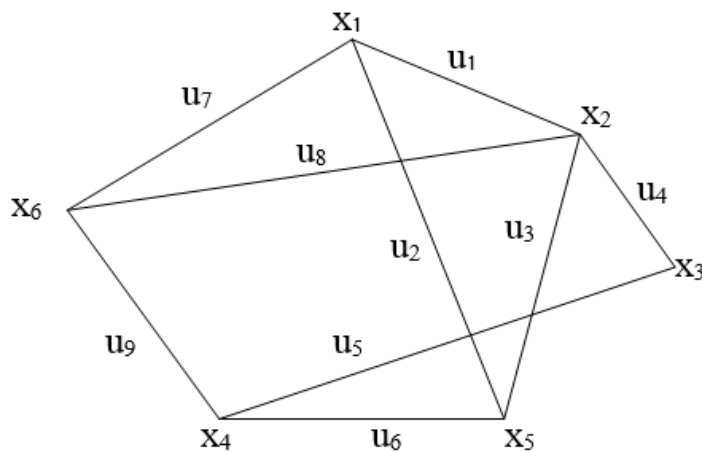


Рис. 3.4. Пример неориентированного графа

Для данного графа матрица смежности имеет вид:

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

Строки и столбцы матрицы соответствуют вершинам графа. Так как в графе нет петель, диагональные элементы матрицы смежности равны 0. На пересечении строки и столбца ставится 1, если соответствующие вершины смежны и 0 в противном случае. Так как граф является неориентированным, матрица смежности симметрична относительно главной диагонали. Кроме того, можно видеть, что сумма элементов матрицы A неориентированного графа по i -й строке (или по i -му столбцу) равна степени данной вершины $\rho(x_i)$.

При построении матрицы смежности ориентированного графа учитывается направление дуг. Элемент a_{ij} равен 1, если в графе существует дуга с

начальной вершиной x_i и конечной вершиной x_j . Если такой дуги нет, элемент a_{ij} равен 0. Таким образом, рассматривается прямое направление дуг, причём начальная вершина дуги берётся по строке, а конечная – по столбцу. Очевидно, что при этом матрица смежности не будет симметрична относительно главной диагонали.

Рассмотрим ориентированный граф:

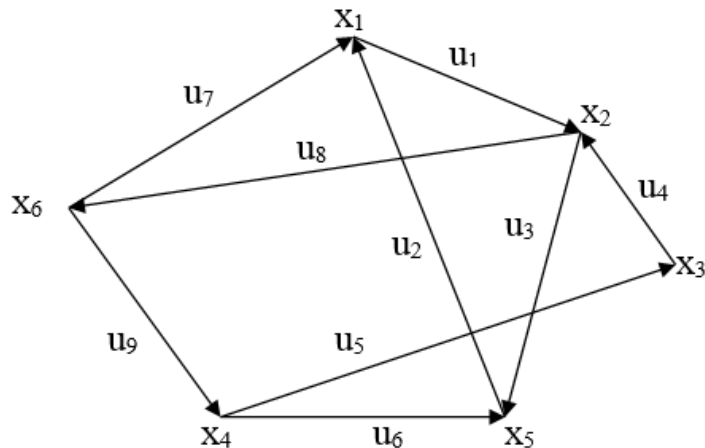


Рис. 3.5. Пример ориентированного графа

С учётом вышеизложенного построим матрицу смежности рассматриваемого ориентированного графа:

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

Например, в первой строке матрицы всего один элемент равен 1, так как существует всего лишь одна дуга, выходящая из вершины x_1 . Эта дуга заходит в вершину x_2 . Поэтому элемент a_{12} равен 1. Дуги u_2 и u_7 направлены в противоположную сторону. Для них вершина x_1 является конечной. Поэтому соответствующие элементы a_{16} и a_{15} равны 0. Аналогично заполняются остальные строки матрицы смежности. Можно заметить, что для орграфа сумма элементов матрицы смежности по i -й строке равна полустепени исхода соответствующей вершине.

ющей вершины $\rho^-(x_i)$, а сумма элементов по i -му столбцу равна полустепени захода данной вершины $\rho^+(x_i)$.

Матрицей инцидентности неориентированного графа $G=(X,U)$ с n вершинами и m ребрами называется матрица B размера $n \times m$, элементы которой определяются следующим образом:

$$a_{ij} = \begin{cases} 1, & \text{если вершина } x_i \text{ инцидентна ребру } u_j; \\ 0 & \text{в противном случае.} \end{cases}$$

Для ориентированного графа :

$$b_{ij} = \begin{cases} 1, & \text{если вершина } x_i \text{ - начальная вершина дуги } u_j; \\ -1, & \text{если вершина } x_i \text{ - конечная вершина дуги } u_j; \\ 0, & \text{если вершина } x_i \text{ не инцидентна дуге } u_j. \end{cases}$$

Построим матрицу инцидентности неориентированного графа, представленного на рис. 3.4. Строки матрицы соответствуют вершинам графа, а столбцы – рёбрам графа. Построение матрицы инцидентности удобнее производить по столбцам. Каждый столбец содержит две единицы, соответствующие двум вершинам, которые инцидентны данному ребру. Матрица инцидентности будет иметь вид:

$$B = \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$

Построение матрицы инцидентности ориентированного графа осуществляется аналогично с той разницей, что каждый столбец содержит 1 и -1. Знак “+” соответствует начальной вершине дуги, знак “-” соответствует конечной вершине дуги. Для графа, изображённого на рис. 3.5, матрица инцидентности имеет вид:

$$B = \begin{matrix} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \end{pmatrix} \end{matrix}$$

3.3. Основные операции над графами

1. *Объединение графов.* Объединением графов $G_1 = (X_1, U_1)$ и $G_2 = (X_2, U_2)$ называется граф $G_3 = (X_1 \cup X_2, U_1 \cup U_2)$.

2. *Пересечение графов.* Пересечением графов $G_1 = (X_1, U_1)$ и $G_2 = (X_2, U_2)$ называется граф $G_3 = (X_1 \cap X_2, U_1 \cap U_2)$.

На рис. 3.6, а, б представлены графы G_1 и G_2 , на рис. 3.6, в их объединение, а на рис. 3.6, г - пересечение.

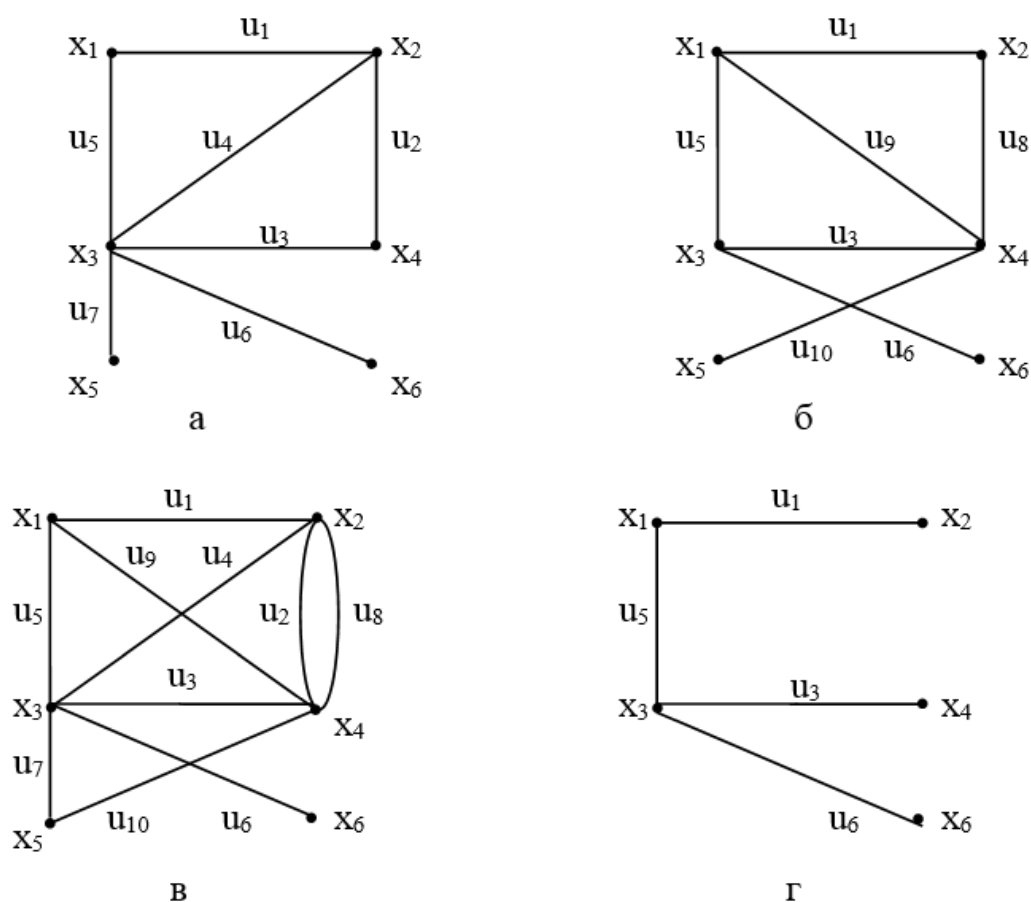


Рис. 3.6. Результат операций над графами

3. *Удаление вершины.* При удалении вершины из графа удаляются и все инцидентные ей ребра (дуги).

4. *Удаление ребра (дуги).* При удалении ребра (дуги) его концевые вершины не удаляются. Операцией, обратной к операции удаления ребра, является операция *добавления ребра*.

5. *Слияние или отождествление вершин.* Говорят, что вершины x_i и x_j в графе G отождествляются (сливаются), если они заменяются такой новой вершиной x_k , что все ребра (дуги) графа, инцидентные x_i и x_j , становятся инцидентными новой вершине x_k .

6. *Стягивание ребра.* Эта операция означает удаление ребра и отождествление его концевых вершин.

7. *Подразбиение ребра.* При выполнении этой операции из графа удаляется ребро (x_i, x_j) и добавляются два новых ребра (x_i, x_k) и (x_k, x_j) , где x_k - новая вершина графа.

На рис. 3.7, а представлен граф G , на рис. 3.7, б - граф G после удаления вершины x_3 , на рис. 3.7, в - граф G в результате удаления ребра u_2 , на рис. 3.7, г - граф G после слияния вершин x_3 и x_4 , на рис. 3.7, д - граф G в результате стягивания ребра u_1 , на рис. 3.7, е - граф G после подразбиения ребра u_2 .

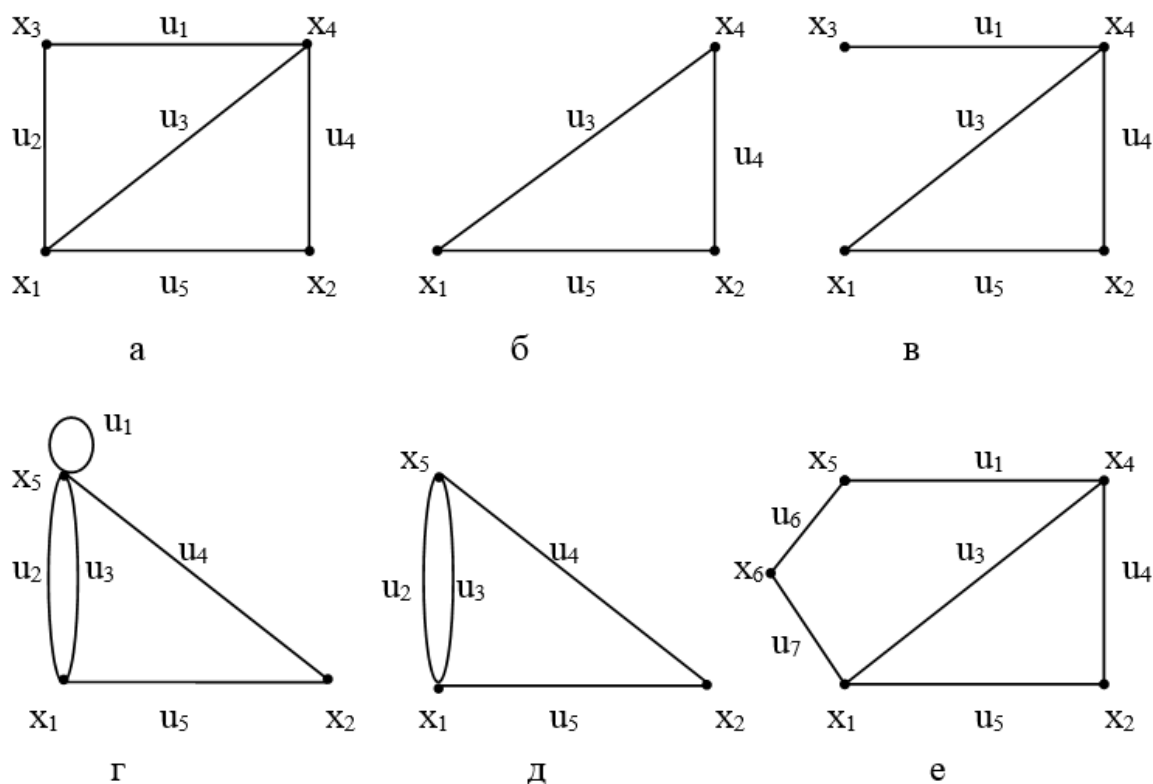


Рис. 3.7. Результат операций над графами

3.4. Изоморфизм и гомеоморфизм графов

Два графа G_1 и G_2 называются *изоморфными*, если можно установить взаимно-однозначное отображение между множествами их вершин, сохраняющее смежность. Примеры изоморфных графов приведены на рис. 3.8.

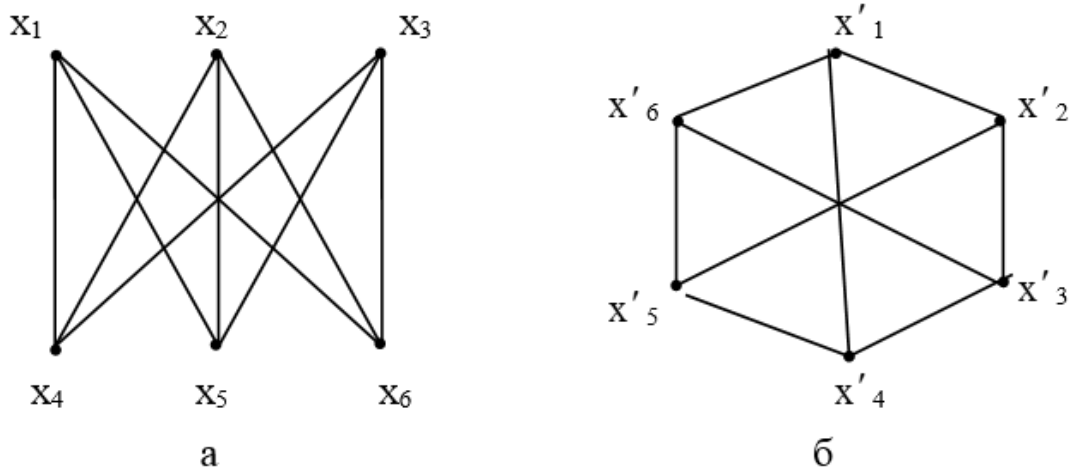


Рис. 3.8. Примеры изоморфных графов

Для графов на рис. 3.8., можно установить следующее соответствие:

$x'_1 - x_1$, $x'_4 - x_4$, $x'_6 - x_6$, $x'_3 - x_3$, $x'_5 - x_2$, $x'_2 - x_5$.

При решении практических задач информационного поиска часто рассматриваются задачи *изоморфного вложения*. Решение этих задач сводится к установлению изоморфизма между графом G и какой-либо частью другого графа G' . Например, на рис. 3.9 подграф, выделенный жирным шрифтом, изоморфен графам, представленным на рис. 3.8. Следовательно, эти графы изоморфно вложимы в граф G .

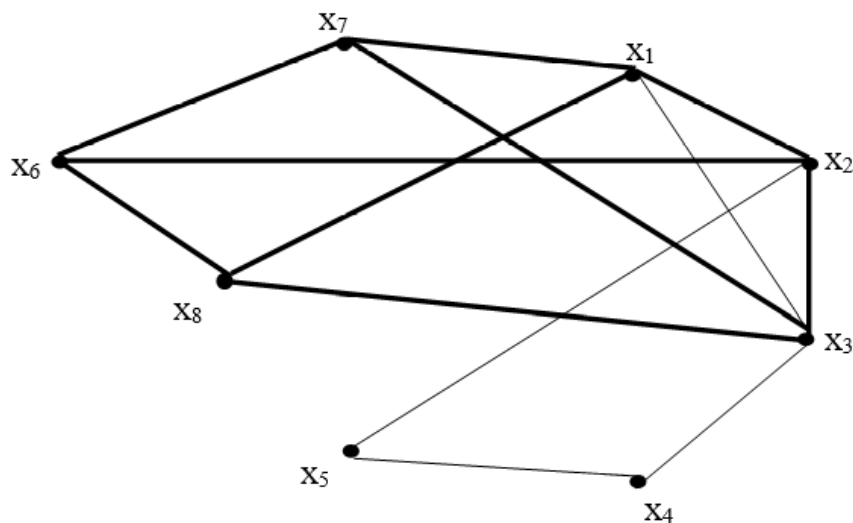


Рис. 3.9. Пример изоморфной вложимости

Два графа называются *гомоморфными*, если оба они могут быть получены из одного и того же графа подразбиением его ребер (рис. 3.10).

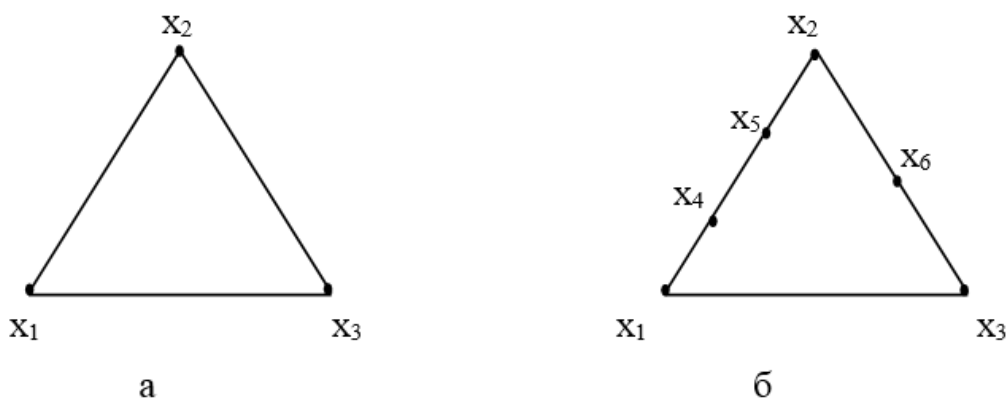


Рис. 3.10. Примеры гомоморфных графов

3.5. Маршруты, цепи, циклы в графах

Последовательность ребер неорграфа $(x_1, x_2), \dots, (x_{r-1}, x_r)$, в которой конец каждого предыдущего ребра совпадает с началом следующего, называется *маршрутом*, соединяющим вершины x_1 и x_r . Аналогом маршрута для орграфа является *ориентированный маршрут* из x_1 в x_r ,

Маршрут является *замкнутым*, если начальная вершина совпадает с конечной. Незамкнутый маршрут, в котором все ребра различны, называют *цепью*. Незамкнутый ориентированный маршрут, содержащий попарно различные дуги называется *путем*. Цепь (путь), в которой (в котором) все вершины попарно различны, называется *простой (простым)*. Замкнутая цепь называется *циклом*, замкнутая простая цепь - *простым циклом*. Замкнутый путь называется *контуром*, замкнутый простой путь - *простым контуром*. Граф, не содержащий циклов, называют *ациклическим*.

Так, для графа на рис. 3.1, а

$(x_2, x_3), (x_3, x_4), (x_4, x_5), (x_5, x_6), (x_6, x_5)$ - маршрут длины 5;

$(x_1, x_2), (x_2, x_3), (x_3, x_4), (x_4, x_2)$ - цепь;

$(x_1, x_2), (x_2, x_3), (x_3, x_4)$ - простая цепь;

$(x_1, x_2), (x_2, x_3), (x_3, x_4), (x_4, x_5), (x_5, x_2), (x_2, x_4), (x_4, x_1)$ - цикл;

$(x_1, x_2), (x_2, x_3), (x_3, x_4), (x_4, x_1)$ - простой цикл.

Для графа на рис. 3.1, б

$\langle x_1, x_2 \rangle, \langle x_2, x_3 \rangle, \langle x_3, x_4 \rangle, \langle x_4, x_5 \rangle, \langle x_5, x_2 \rangle$ - путь;

$\langle x_1, x_2 \rangle, \langle x_2, x_3 \rangle, \langle x_3, x_4 \rangle, \langle x_4, x_5 \rangle$ - простой путь;

$\langle x_1, x_2 \rangle, \langle x_2, x_3 \rangle, \langle x_3, x_4 \rangle, \langle x_4, x_1 \rangle$ - простой контур.

3.6. Основные типы графов

Граф, у которого все вершины имеют одинаковую степень k , называется *регулярным (однородным)* графом степени k . Число ребер m такого графа определяется следующим образом:

$$m = \frac{1}{2} n \cdot k.$$

Граф, не имеющий ребер, называется *пустым*, а его вершины - изолированными. Пустой граф, имеющий n вершин, обозначается O_n .

Граф называется *полным*, если все его вершины смежны. Полный граф порядка n обозначается K_n . Число ребер полного графа K_n определяется следующим образом:

$$m = \sum_{i=1}^n \rho(x_i)/2.$$

Граф $G=(X,U)$ называется *двудольным*, если существует такое разбиение множества его вершин X на два непересекающихся подмножества X_1 и X_2 , что каждое ребро имеет одну концевую вершину в подмножестве X_1 , а другую - в подмножестве X_2 . Подмножества X_1 и X_2 в данном случае называются *долями*. Если две вершины графа, входящие в разные доли, смежны, то граф называется *полным двудольным* и обозначается K_{n_1, n_2} , где $n_1 = |X_1|$, $n_2 = |X_2|$. Аналогично определяются k -дольный и полный k -дольный графы ($k > 2$).

На рис. 3.11 представлены регулярный граф степени 2 (рис. 3.11, а), графы K_5 (рис. 3.11, б), $K_{1,5}$ (рис. 3.11, в), $K_{3,3}$ (рис. 3.11, г)

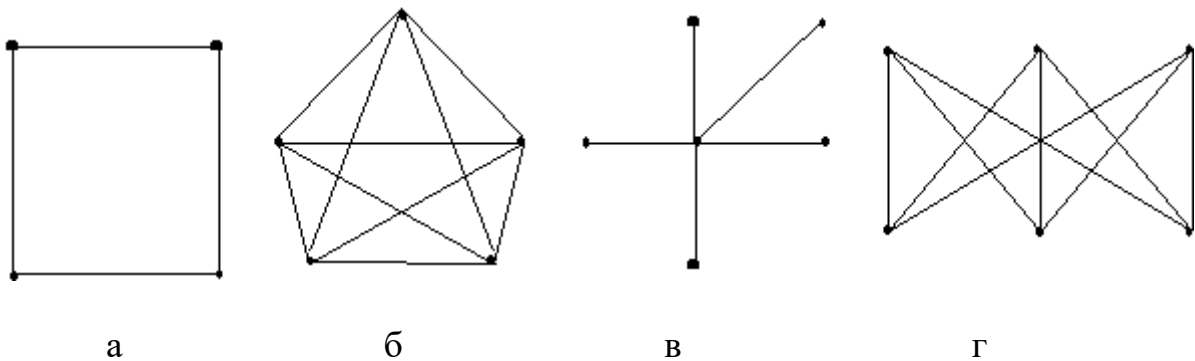


Рис. 3.11. Примеры графов

Реберным графом графа $G=(X,U)$ называется граф $L(G)=(U,V)$, вершинам которого соответствуют ребра графа G . Две вершины u_i и u_j графа $L(G)$ смежны, если соответствующие им ребра u_i и u_j графа G . На рис. 3.12, а представлен граф G , а на рис. 3.12, б - реберный граф $L(G)$.

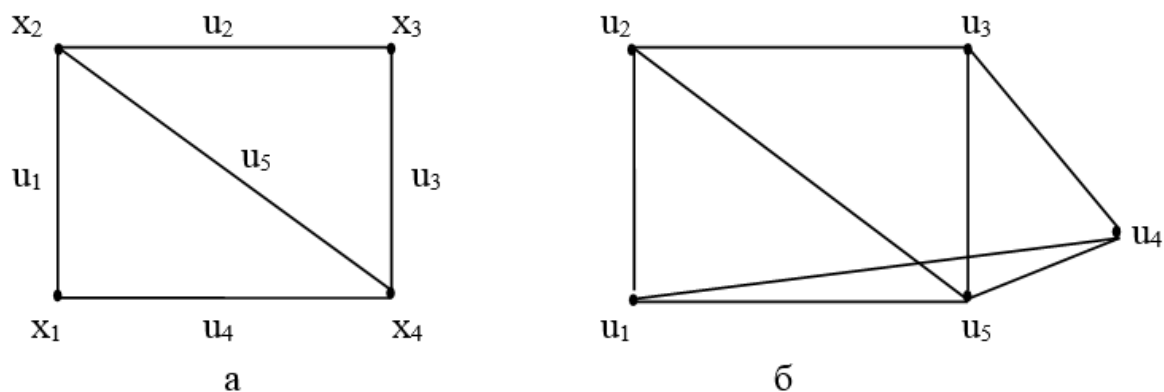


Рис. 3.12. Граф и его реберный граф

При конструкторском проектировании РЭА и ЭВА к топологическим чертежам часто предъявляются требование получения плоского изображения схем или их частей. В этой связи возникает задача определения *планарности* графа. Граф называется *планарным*, если его можно изобразить на плоскости таким образом, чтобы его ребра не пересекались. Граф, множество ребер которого расположено на плоскости таким образом, что ребра имеют точки пересечения лишь в вершинах, называется *плоским*. На рис. 3.13, а представлен планарный граф, а на рис. 3.13, б - изоморфный ему плоский граф.

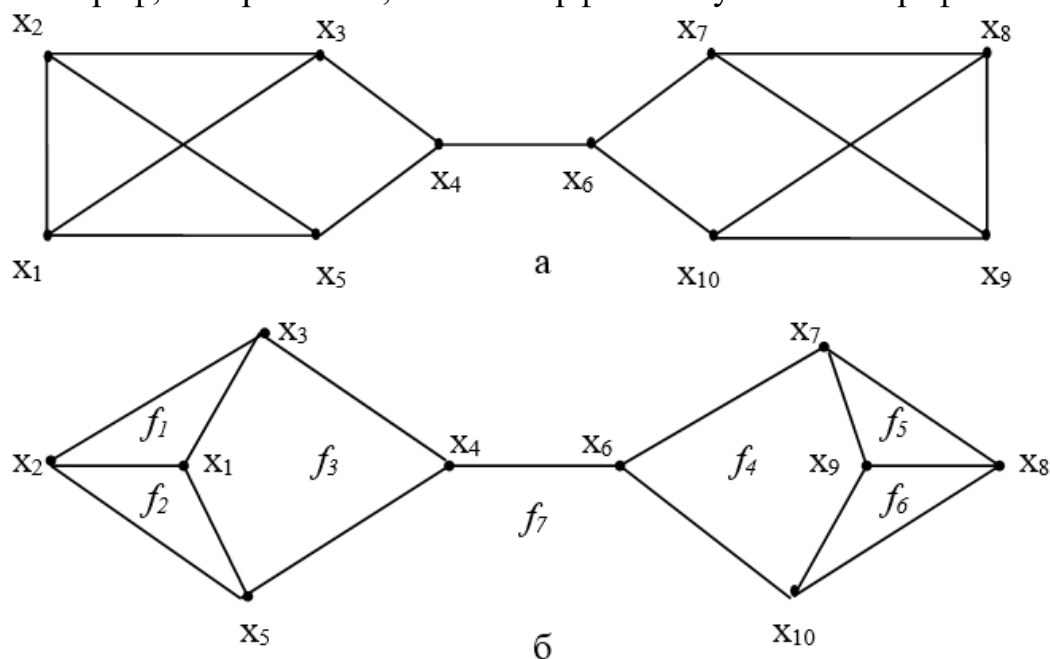


Рис. 3.13. Пример планарного графа и изоморфного ему плоского графа

Планарный граф, который при добавлении любого ребра становится непланарным, называют *максимальным планарным графом*. Наименьшее число ребер, которое необходимо удалить из непланарного графа, расположенного на плоскости, чтобы он стал планарным, называется *числом планарности* и обо-

значается $\theta(G)$. Для полного графа K_n , $n \geq 4$ $\theta(K_n) = (n-3)(n-4)/2$. Если $n=4$, то $\theta(K_n) = 0$. Граф, не имеющий ни одного цикла, всегда планарен.

3.7. Метрические характеристики графов

Рассмотрим граф $G=(X,U)$. Длина кратчайшего маршрута, соединяющего вершины x_i и x_j (пути из x_i и x_j), называется *расстоянием* между вершинами x_i и x_j и обозначается $d(x_i, x_j)$. Если вершины x_i и x_j не соединены маршрутом (путем), $d(x_i, x_j) = \infty$.

Матрицей расстояний графа с n вершинами называется квадратная матрица D порядка n , элементы которой определяются следующим образом:

$$d_{ij} = \begin{cases} 0, & \text{если } x_i = x_j; \\ d(x_i, x_j), & \text{если } x_i \neq x_j. \end{cases}$$

Рассмотрим вершину x_i графа. *Эксцентриситетом* (отклоненностью) вершины x_i называется величина:

$$e(x_i) = \max_{x_j \in X} d(x_i, x_j)$$

$$x_j \in X$$

Максимальный среди эксцентриситетов вершин называется *диаметром* графа G :

$$\text{diam}(G) = \max_{x_i \in X} e(x_i) = \max_{x_i \in X} \max_{x_j \in X} d(x_i, x_j),$$

$$x_i \in X \quad x_i \in X \quad x_j \in X$$

Минимальный из эксцентриситетов вершин связного графа называется его *радиусом*:

$$r(G) = \min_{x_i \in X} e(x_i) = \min_{x_i \in X} \max_{x_j \in X} d(x_i, x_j),$$

$$x_i \in X \quad x_i \in X \quad x_j \in X$$

Вершина с минимальным эксцентриситетом называется *центром* графа.

Передаточным числом для вершины x_i называется величина:

$$P(x_i) = \sum_{x_j \in X} d(x_i, x_j)$$

Вершина графа, которой соответствует минимальное передаточное число

$$\min P(x_i) = \min \sum d(x_i, x_j),$$

$$x_i \in X \quad x_i \in X \quad x_j \in X$$

называется *медианой* графа.

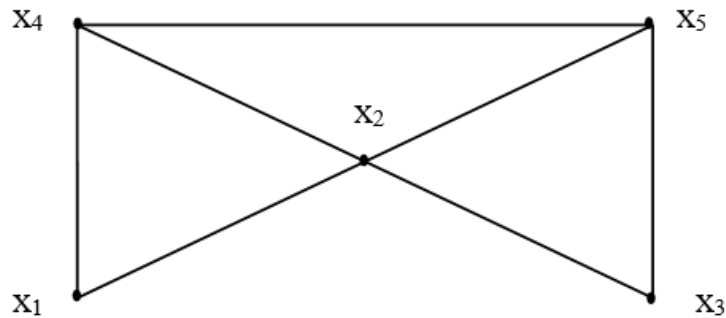


Рис. 3.14. Пример графа

Например, для графа, изображенного на рис. 3.14, матрица расстояний, эксцентриситеты вершин и передаточные числа определяются следующим образом:

$$D = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 0 & 1 & 2 & 1 & 2 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 2 & 1 \\ 1 & 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 1 & 0 \end{pmatrix} \end{matrix} \quad \begin{matrix} e(x_1) = 2, & P(x_1) = 6, \\ e(x_2) = 1, & P(x_2) = 4, \\ e(x_3) = 2, & P(x_3) = 6, \\ e(x_4) = 2, & P(x_4) = 5, \\ e(x_5) = 2, & P(x_5) = 5. \end{matrix}$$

Проанализировав полученные результаты, определим метрические характеристики графа: радиус графа равен 1; диаметр графа равен 2; центр графа - вершина x_2 ; медиана графа - вершина x_2 .

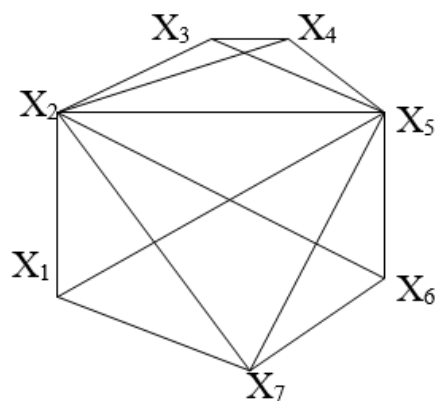
В графе может быть несколько центров и медиан.

Рассмотрим прикладные аспекты задачи определения метрических характеристик графа. С определением метрических характеристик графа часто приходится сталкиваться при решении задач размещения различных объектов (элементов на платах, теле- и радиостанций и т.д.). При этом если необходимо минимизировать расстояние от размещаемого элемента до наиболее удаленной вершины графа, то задачи такого рода называются *минимаксными* задачами размещения и сводятся к определению центров графа.

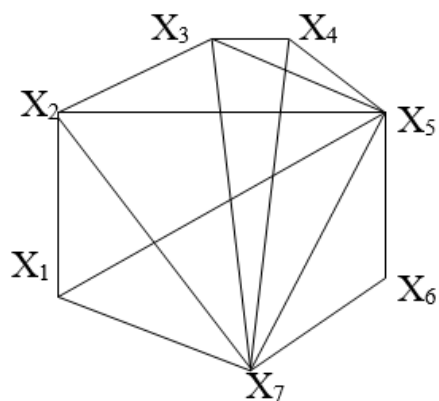
Если же необходимо минимизировать сумму расстояний от вершин графа до размещаемого объекта, то подобные задачи называются *минисуммными* задачами размещения и сводятся к определению медиан графа.

Упражнения

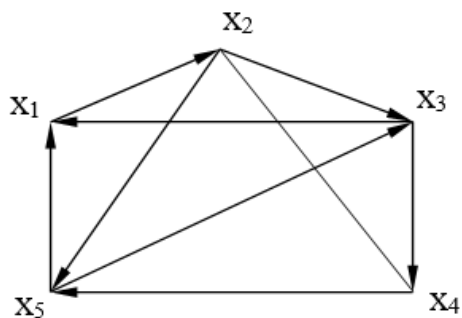
1. Для графов, изображенных на рис. 3.15, построить матрицы смежности, инцидентности и определить степени вершин:



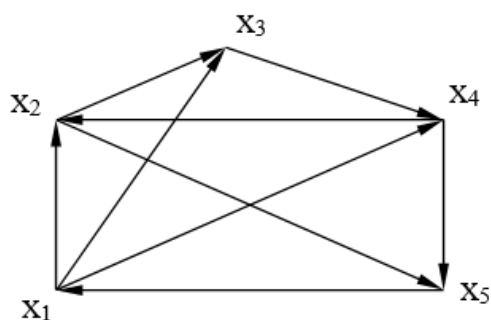
а



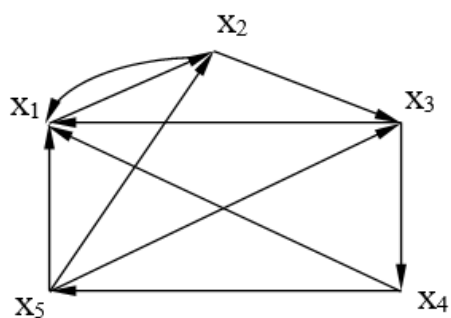
б



в



г



д

Рис. 3.15. Примеры графов для задания матричным способом

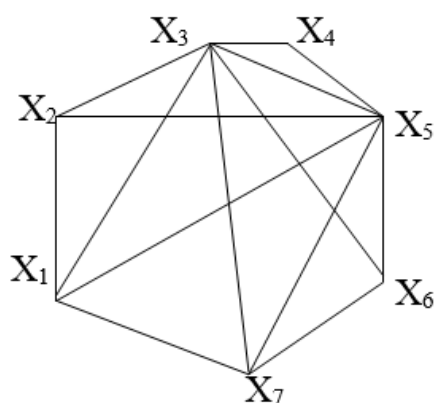
2. Представить геометрически граф, заданный матрицей смежности:

$$A = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{pmatrix}$$

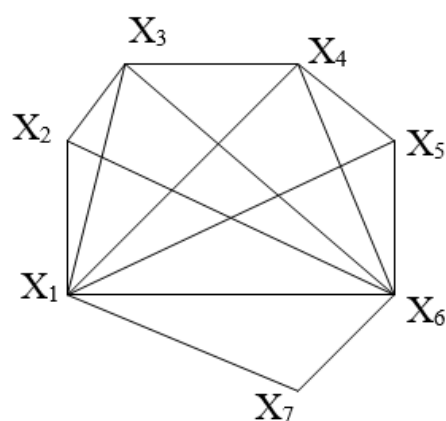
3. Для графа на рис. 3.16, а построить:

- порожденный подграф на множестве вершин $\{x_1, x_2, x_3\}$;
- остовный подграф, содержащий только те ребра (x_i, x_j) , в которых сумма $i+j$ является нечетным числом.

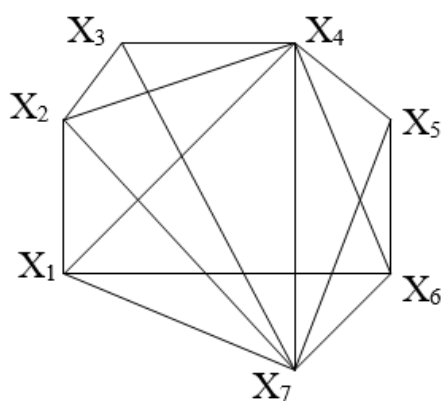
4. Определить метрические характеристики графов



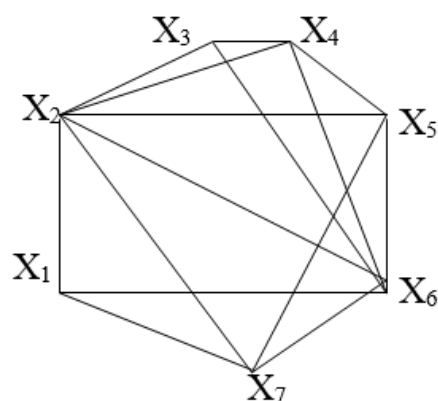
а



б



в



г

Рис. 3.16. Примеры графов для определения метрических характеристик

4. АЛГОРИТМЫ ТЕОРИИ ГРАФОВ

4.1. Достижимость и связность

Вершина x_j неорграфа (орграфа) *достижима* из вершины x_i , если существует маршрут, соединяющий x_i и x_j (путь из x_i в x_j). Если в орграфе существуют пути из x_i в x_j и обратно, то говорят, что вершины x_i и x_j *взаимно достижимы*.

Неорграф называется *связным*, если любые две его вершины соединены маршрутом. Орграф называется *сильно связным*, если любые две его вершины взаимно достижимы, *односторонне связным*, если для любых двух вершин по крайней мере одна достижима из другой и *слабо связным*, если связным является лежащий в его основе неорграф.

Так на рис. 4.1, а, б представлены несвязный и связный неорграфы, а на рис. 4.1, в – сильно связный орграф. Орграф на рис. 4.1, г односторонне связный, так как его вершины x_1 и x_5 не являются взаимно достижимыми. Орграф на рис. 4.1, д слабо связный и не является ни сильно, ни односторонне связным, так как вершина x_1 не достижима из вершины x_3 , а вершина x_3 из x_1 .

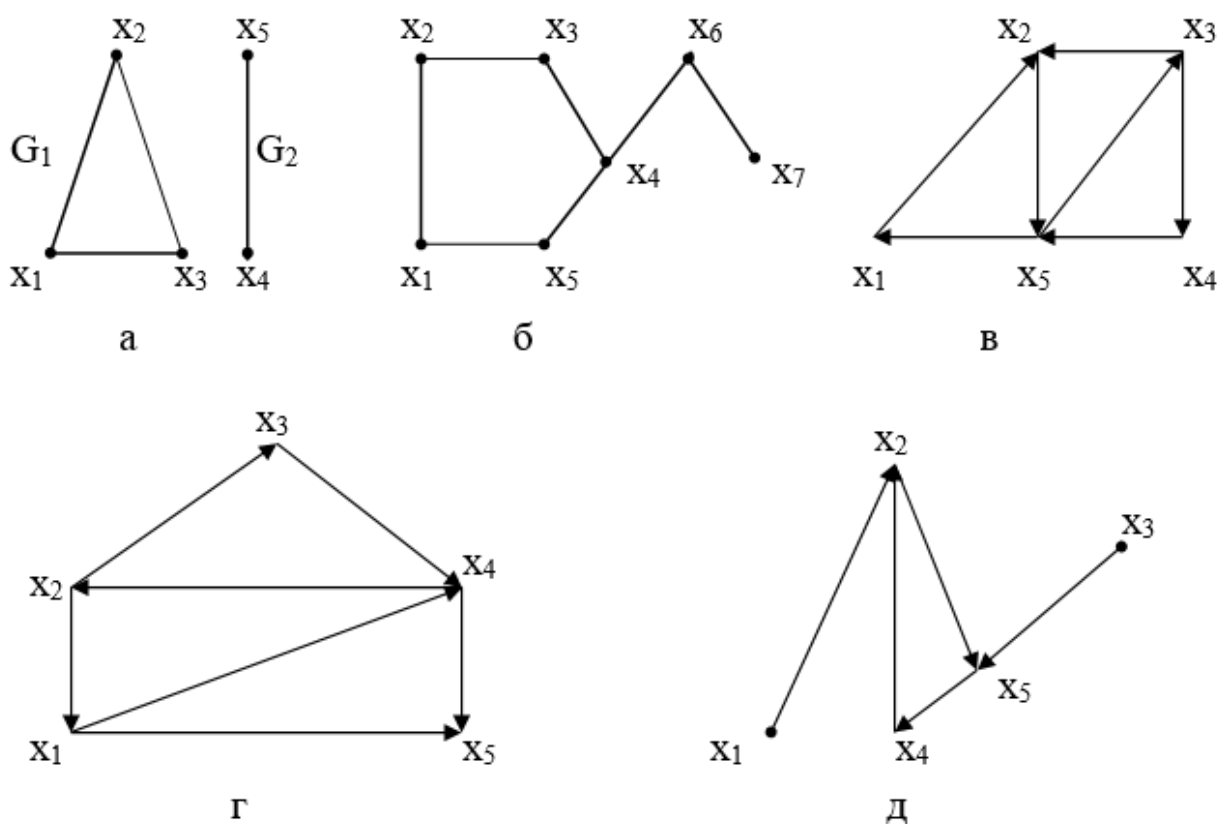


Рис. 4.1. Графы с разными типами связности

Компонентой связности неорграфа называется его максимальный связный подграф, то есть подграф, не содержащийся ни в каком другом связном подграфе этого графа. Аналогично *сильной компонентой* орграфа называется его максимальный сильно связный подграф. Так, неорграф на рис. 4.1, а состоит из двух компонент связности: G_1 и G_2 , оргграф на рис. 4.1, г содержит сильную компоненту, включающую вершины x_1, x_2, x_3, x_4 .

Если граф G моделирует схему РЭА или ЭВА, то выделение в нем компонент связности соответствует его *разбиению* на связные непересекающиеся подграфы (подсхемы), причем такое разбиение всегда однозначно.

Матрицей достижимости графа с n вершинами называется квадратная матрица R порядка n , элементы которой определяются следующим образом:

$$r_{ij} = \begin{cases} 1, & \text{если вершина } x_j \text{ достижима из } x_i; \\ 0, & \text{в противном случае.} \end{cases}$$

Матрицей контрдостижимости графа с n вершинами называется квадратная матрица Q порядка n , элементы которой определяются следующим образом:

$$q_{ij} = \begin{cases} 1, & \text{если вершина } x_j \text{ достижима из } x_i; \\ 0, & \text{в противном случае;} \end{cases}$$

$$r_{ii} = 1, q_{ii} = 1 \quad \forall i.$$

Матрицей сильной связности орграфа с n вершинами называется квадратная матрица S порядка n , элементы которой определяются следующим образом:

$$s_{ij} = \begin{cases} 1, & \text{если вершины } x_j \text{ и } x_i \text{ взаимно достижимы;} \\ 0, & \text{в противном случае.} \end{cases}$$

Матрицы R , Q и S связаны между собой следующим образом:

$$R = Q^T, S = R * Q,$$

где $*$ - операция поэлементного умножения матриц.

Если вершины x_i и x_j принадлежат одной сильной компоненте орграфа, то $s_{ij} = 1$. При этом строки (столбцы), соответствующие этим вершинам в матрице S , одинаковы. Для неорграфа матрицы достижимости и контрдостижимости совпадают и являются аналогами матрицы сильной связности.

При машинной реализации алгоритмов на графах матрицы расстояний, достижимости, контрдостижимости и сильной связности можно определять через матрицу смежности.

Пусть дан граф G , $A = (a_{ij})$ - его матрица смежности. Элементы матрицы расстояний D можно вычислить следующим образом: d_{ij} равно наименьшему из чисел n , для которых $a_{ij}^n > 0$, где a_{ij}^n - элемент матрицы A^n , и равно бесконечности, если таких чисел нет.

Пример. Дан граф G (рис. 4.2). Найти матрицу расстояний D .

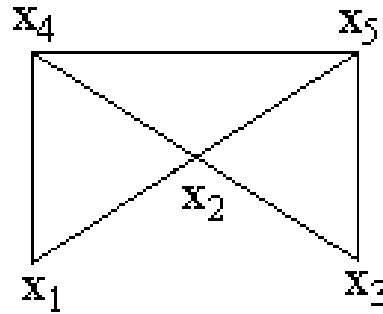


Рис.4.2. Исходный граф

Сначала найдем матрицу смежности A :

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

По матрице A начнём строить матрицу расстояний D . Диагональные элементы матрицы D равны нулю. Остальные элементы определяем по следующему правилу: если элемент a_{ij} матрицы A равен 1, то соответствующий элемент d_{ij} матрицы D тоже будет равен 1. Недоопределенная матрица D изображена ниже.

$$D = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 0 & 1 & & 1 & \\ 1 & 0 & 1 & 1 & 1 \\ & 1 & 0 & & 1 \\ 1 & 1 & & 0 & 1 \\ & 1 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

Умножим матрицу A саму на себя и найдем матрицу A^2 .

$$A^2 = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 2 & 1 & 1 & 1 & 2 \\ 1 & 4 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 3 & 1 \\ 2 & 2 & 1 & 1 & 3 \end{pmatrix} \end{matrix}$$

Так как в матрице A^2 нет нулевых элементов, то оставшиеся неопределёнными элементы матрицы D равны 2. Окончательно матрица расстояний D будет иметь вид:

$$D = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 0 & 1 & 2 & 1 & 2 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 2 & 1 \\ 1 & 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

Матрицу достижимости R можно найти с помощью матрицы смежности A следующим образом $R = T(E + A + A^2 + \dots + A^{n-1})$, где n – число вершин, E – единичная матрица, а оператор T определяется следующим образом:

$$T(a_{ij}) = \begin{cases} 1, & \text{если } a_{ij} > 0 \\ 0, & \text{если } a_{ij} = 0 \end{cases}$$

Матрица контрдостижимости Q находится транспонированием матрицы достижимости R : $Q = R^T$.

Пример. Дан граф G . Найти матрицу достижимости R и контрдостижимости Q .

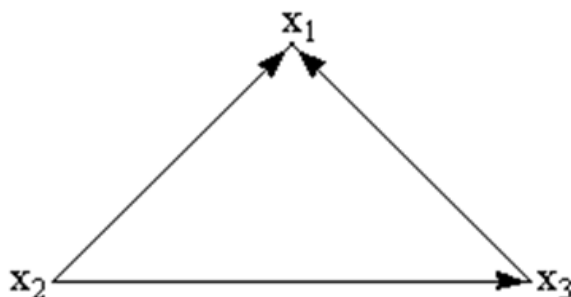


Рис. 4.3. Пример графа для построения матриц достижимости и контрдостижимости

A^3 : Запишем единичную матрицу, матрицу смежности A , ее квадрат A^2 и куб

$$E = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad A = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$A^2 = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

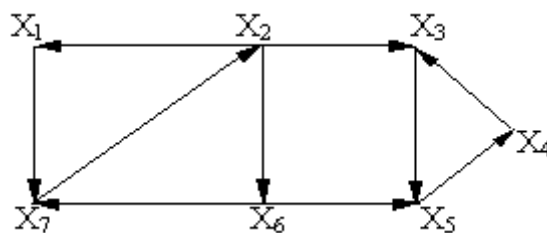
Складываем единичную матрицу E , матрицу A и матрицу A^2 и с помощью оператора T находим матрицу достижимости R и (транспонируя R) матрицу контрдостижимости Q :

$$R = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{matrix} \quad Q = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Алгоритм определения сильных компонент связности орграфа

1. Построение матрицы достижимости R .
2. Построение матрицы контрдостижимости $Q = R^T$.
3. Определение матрицы $S = R * Q$.
4. Выделение сильных компонент графа. Вершины, соответствующие одинаковым строкам и столбцам матрицы S , принадлежат одной сильной компоненте графа.

Пример 1. Определим сильные компоненты следующего графа:



1. Построим матрицу достижимости R . Диагональные элементы матрицы достижимости равны 1 (т.е. вершина достижима сама из себя). 2 Матрица строится следующим образом: если вершина x_j достижима из x_i (то есть существует путь из x_i в x_j), то соответствующий элемент матрицы достижимости r_{ij} равен 1, в противном случае он равен 0. При этом путь из x_i в x_j может проходить через любые промежуточные вершины. Например, вершина x_2 достижима из x_1 (через вершину x_7), поэтому элемент матрицы r_{12} равен 1. Из вершины x_1 достижимы все вершины графа, поэтому в первой строке матрицы все единицы. Вершина x_2 из x_3 недостижима (пути нет), поэтому элемент r_{32} равен 0.

Таким образом, при построении матрицы достижимости начальная вершина пути берётся по строке, а конечная по столбцу. Вершина, которая соответствует столбцу, должна быть достижима из вершины, которая соответствует строке.

Матрица достижимости имеет вид:

$$R = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$

2. Построим матрицу контрдостижимости Q :

$$Q = R^T = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Матрица контрдостижимости строится следующим образом: если вершина x_i достижима из x_j , то соответствующий элемент матрицы контрдостижимости q_{ij} равен 1, в противном случае он равен 0. В отличие от матрицы достижимости, при построении матрицы контрдостижимости начальная вершина пути берётся по столбцу, а конечная по строке.

Матрицы R и Q связаны между собой следующим образом:

$$Q = R^T$$

Таким образом, чтобы построить матрицу контрдостижимости, нужно транспонировать матрицу достижимости R.

$$Q = R^T = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

3. Построим матрицу сильной связности S:

$$S = R * Q = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

В матрице сильной связности элемент s_{ij} равен 1, если вершины x_i и x_j взаимно достижимы (то есть существует путь из x_i в x_j и обратно). Это выполняется в том случае, если соответствующие элементы r_{ij} и q_{ij} равны 1 (на одном и том же месте в матрицах находятся 1). Поэтому матрица S строится путём **поэлементного** умножения матриц R и Q. Матрицы R и Q умножаются

поэлементно, т.е. умножаются два элемента, стоящие в обеих матрицах на одинаковых местах. Например, $s_{12} = r_{12} \cdot q_{12} = 1 \cdot 1 = 1$, $s_{13} = r_{13} \cdot q_{13} = 1 \cdot 0 = 0$

4. По матрице S определим сильные компоненты графа.

Вершинам, принадлежащим одной сильной компоненте графа, соответствуют одинаковые строки (и одинаковые столбцы) матрицы S .

Для рассмотренного примера:

x_1, x_2, x_6, x_7 – первая сильная компонента (строки 1,2,6,7 одинаковы. Столбцы 1,2,6,7 тоже одинаковы).

x_3, x_4, x_5 – вторая сильная компонента (строки 3,4,5 одинаковы. Столбцы 3,4,5 тоже одинаковы).

Пример 2. Определим сильные компоненты графа, изображённого на рис. 4.1, д. Построим матрицу достижимости R , матрицу контрдостижимости Q и матрицу сильной связности S .

$$R = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

$$Q = R^t = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$

$$S = R * Q = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Таким образом, данный граф имеет одну сильную компоненту, состоящую из вершин x_2, x_4, x_5 .

Понятие связности играет важную роль при разработке схем РЭА и ЭВА, проектировании сетей ЭВМ и анализа их надежности. Анализ надёжности связан с определением показателей вершинной и реберной связности графовой модели сети. *Числом вершинной связности* $\delta(G)$ графа называется наименьшее число вершин, при удалении которых нарушается связность графа. Аналогично определяется *число реберной связности* $\lambda(G)$. $\delta(G) \leq \lambda(G) \leq \rho^*(G)$, где $\rho^*(G)$ - минимальная степень вершин графа. Вершина x_i называется *точкой сочленения*, если после ее удаления граф становится несвязным. Ребро графа называется *мостом*, если его удаление приводит к несвязному графу. Так, для графа на рис. 4.1 б вершина x_4 является точкой сочленения; а ребро (x_4, x_6) - мостом. Для этого графа $\delta(G)=1$, $\lambda(G)=1$. Таким образом, числа вершинной и реберной связности графа, соответствующего сети ЭВМ, отражают чувствительность сети к разрушению центров и каналов, а мостам и точкам сочленения соответствуют наиболее уязвимые места.

4.2. Деревья

Связный ациклический граф, имеющий не менее двух вершин, называется *деревом*. *Ориентированным деревом* называется орграф без циклов, в котором имеется вершина x_0 , из которой существует только один ориентированный путь в любую другую вершину. *Деревом графа* G называется любой его связный ациклический подграф. Дерево графа G , содержащее все вершины этого графа, называется *остовным деревом (остовом)* T графа G . При этом ребра остовного дерева T называются *ветвями*, а ребра графа G , не принадлежащие остову T - *хордами*. K - *деревом* называется ациклический граф, состоящий из k компонент связности. На рис 4.4, а представлен граф G , на рис. 4.4, б - одно из его деревьев, на рис. 4.4, в, г - два остова T_1 и T_2 графа G .

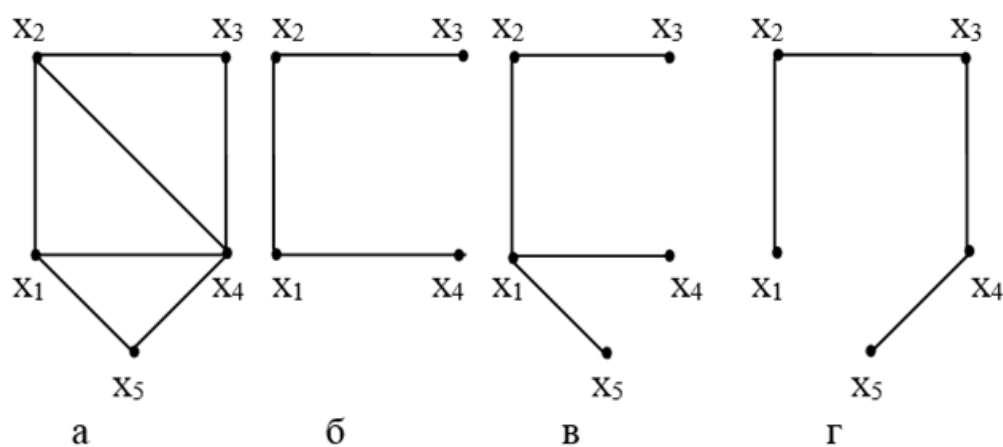


Рис. 4.4. Граф и его деревья

Пусть G - произвольный граф с n - вершинами, m - ребрами и k компонентами связности. Тогда наименьшее число ребер, которое необходимо удалить из графа, чтобы он стал ациклическим (деревом), равно $m-n+k$. Это число называется *цикломатическим числом* графа G и обозначается $v(G)$. Таким образом, для получения остова из графа G нужно удалить $v(G)$ ребер. Из определения $v(G)$ следует, что $v(G) \geq 0$. Граф не имеет циклов тогда и только тогда, когда $v(G)=0$ и имеет единственный цикл тогда и только тогда, когда $v(G)=1$. Остов произвольного графа G содержит $n-k$ ребер. Это число называется *коцикломатическим числом* графа G и обозначается $v^*(G)$. $v^*(G)=m-v(G)$. Таким образом, коцикломатическое число $v^*(G)$ равно количеству ветвей произвольного остова этого графа, а цикломатическое число $v(G)$ равно количеству его хорд. Остов связного графа ($k=1$) содержит $n-1$ ребро.

Пусть T - остовное дерево графа G . Цикл, получаемый добавлением k ветвям остовного дерева T какой-либо хорды (ребра графа G , не принадлежащего T), называется *фундаментальным (базисным) циклом* графа G относительно остова T . Так как в каждом фундаментальном цикле присутствует только одна хорда, то в любом графе можно выделить $v(G)$ фундаментальных циклов, причем сами эти циклы определяются неоднозначно и зависят от выбранного остова T . Фундаментальные циклы относительно остова T_1 (рис. 4.4, в): $x_1x_2x_3x_4x_1$, $x_1x_2x_4x_1$, $x_1x_4x_5x_1$. Фундаментальные циклы относительно остова T_2 (рис. 4.4, г): $x_1x_2x_3x_4x_1$, $x_2x_3x_4x_2$, $x_1x_2x_3x_4x_5x_1$.

Матрицей фундаментальных циклов графа G относительно остова T называется матрица F , строки которой соответствуют фундаментальным циклам, а столбцы - ребрам (дугам) графа G . Элементы матрицы F определяются следующим образом:

$$f_{ij} = \begin{cases} 1, & \text{если } j - \text{е ребро входит в } i - \text{й цикл,} \\ 0, & \text{в противном случае.} \end{cases}$$

Таким образом, матрица F состоит из $v(G)$ строк и m столбцов. Перед составлением матрицы F необходимо пронумеровать дуги графа. Матрица фундаментальных циклов графа G относительно остова T_1 (рис. 4.4, в) имеет вид:

$$F = \begin{matrix} & (x_1x_2) & (x_2x_3) & (x_3x_4) & (x_1x_4) & (x_2x_4) & (x_1x_5) & (x_4x_5) \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{matrix}.$$

Для орграфа элементы матрицы F определяются следующим образом:

$$f_{ij} = \begin{cases} 1, \text{ если } j\text{-я дуга входит в } i\text{-й цикл и ее ориентация соответствует} \\ \text{ориентации цикла,} \\ -1, \text{ если } j\text{-я дуга входит в } i\text{-й цикл и ее ориентация не соответствует} \\ \text{ориентации цикла,} \\ 0, \text{ если } j\text{-я дуга не входит в } j\text{-й цикл.} \end{cases}$$

Строки матрицы F называются *циклическими векторами* графа G .

Определение чисел $\nu(G)$, $\nu^*(G)$ и построение фундаментальных циклов графа имеет существенное значение при анализе электрических цепей. Так, цикломатическое число $\nu(G)$ равно наибольшему количеству независимых контуров в графе электрической цепи, то есть наибольшему числу независимых круговых токов, которые могут протекать в цепи. Эти круговые токи соответствуют фундаментальным циклам графа электрической цепи. Коцикломатическое число $\nu^*(G)$ равно наибольшему количеству независимых разностей потенциалов между узлами электрической цепи.

Алгоритмы построения остовных деревьев графа

Для построения произвольного остова связного графа могут быть использованы две стратегии: *поиск в глубину* и *поиск в ширину*.

Поиск в глубину начинается с произвольной вершины x_0 . Затем выбирается вершина x_1 , смежная с x_0 . После этого выбирается вершина x_2 , смежная с x_1 , и т.д. Если на k -м шаге для вершины x_k новых вершин не существует, осуществляется возврат к предыдущей вершине и поиск осуществляется от неё. Процесс продолжается до тех пор, пока не будут просмотрены все вершины графа. *Поиск в ширину* отличается от поиска в глубину тем, что выбор очередной вершины происходит путём просмотра не одной, а всех смежных вершин из списка новых.

Пример. Дан граф G (рис.4.5). Построить остов графа поиском в глубину и поиском в ширину.

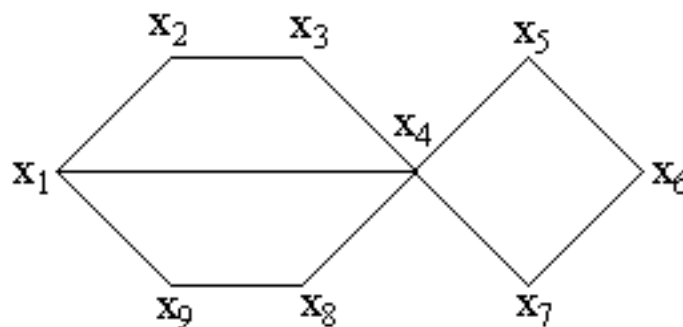


Рис.4.5. Пример графа для построения остовов

Поиск в глубину. Последовательность прохождения и выбора вершин:

$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_6, x_5, x_4, x_8, x_9$, построенный остов представлен на рис.4.6, а.

Поиск в ширину. Последовательность прохождения и выбора вершин:

$x_1, x_2, x_4, x_9, x_3, x_5, x_7, x_8, x_6$, построенный остов представлен на рис.4.6, б.

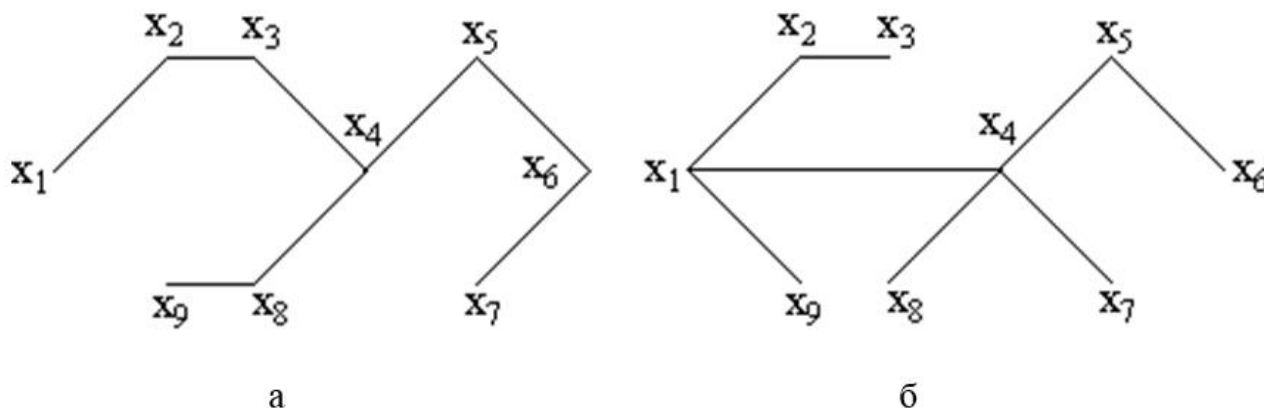


Рис.4.6. Остовы графа

Алгоритмы построения кратчайшего остова взвешенного графа

При конструкторском и топологическом проектировании РЭА и ЭВА, создании вычислительных сетей важное значение имеет задача определения кратчайшего остова (остова минимального веса) взвешенного графа. *Кратчайшим остовом* взвешенного графа G будем называть остов, у которого сумма весов всех ребер наименьшая.

Рассмотрим два наиболее распространенных алгоритма построения кратчайшего остова связного взвешенного графа G - *алгоритмы Краскала и Прима*. *Алгоритм Краскала* заключается в следующем. На начальном этапе из всех ребер графа G выбирается ребро u_1 минимального веса и включается в остов. На каждом последующем i -м шаге из еще не включенных в остов ребер выбирается ребро, имеющее минимальный вес и не составляющее циклов с уже выбранными ребрами. Процесс продолжается до тех пор, пока в остов не будет включено $n-1$ ребро. Таким образом, на каждом этапе работы алгоритма Краскала осуществляется построение ациклического подграфа графа G .

В качестве иллюстрации работы алгоритма Краскала рассмотрим взвешенный граф, представленный на рис. 4.7, а). Сначала выбираем ребро (x_3, x_4) минимального веса и полагаем $u_1 = (x_3, x_4)$. Далее из оставшихся ребер выбираем ребро (x_1, x_2) и полагаем $u_2 = (x_1, x_2)$. Аналогично $u_3 = (x_2, x_4)$. Среди оставшихся ребер минимальный вес имеет ребро (x_1, x_3) . Оно не пригодно для построения, так как составляет цикл с уже выбранными ребрами. Поэтому $u_4 = (x_2, x_5)$, $u_5 = (x_5, x_6)$. Кратчайший остов представлен на рис. 4.7, б).

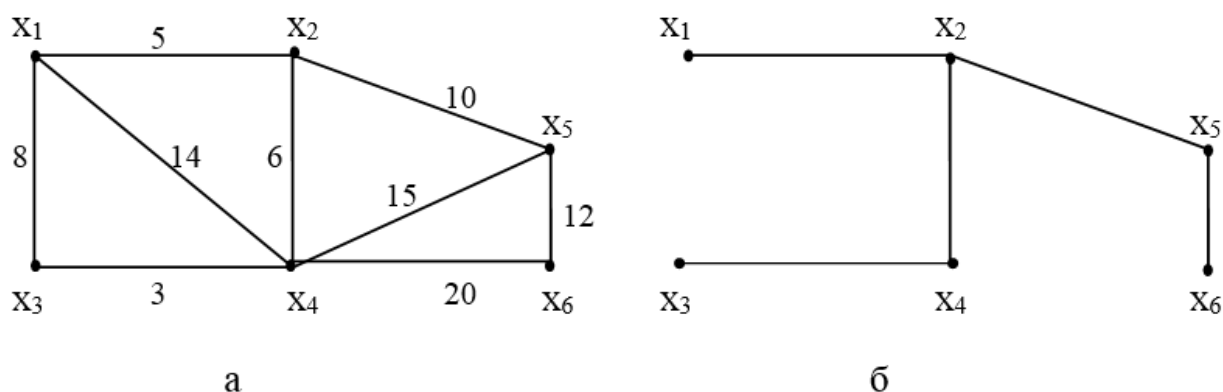


Рис. 4.7. Граф и его кратчайший остов

Алгоритм Прима отличается от алгоритма Краскала только тем, что в процессе построения связность построенного фрагмента остова не должна нарушаться. При этом если дерево T_i уже построено, то новое ребро выбирается не из всех оставшихся ребер графа G , а только из тех, которые соединяют вершины дерева T_i с вершинами, не включенными в T_i . Из этих ребер выбирается ребро минимального веса. При этом необходимо следить за тем, чтобы при добавлении нового ребра не появлялись циклы. Например, для графа на рис. 4.7, а $u_1 = (x_3, x_4)$. Следующее ребро минимального веса выбирается среди ребер $(x_1, x_3), (x_1, x_4), (x_2, x_4), (x_4, x_5), (x_4, x_6)$. Получаем $u_2 = (x_2, x_4)$. Аналогично $u_3 = (x_1, x_2), u_4 = (x_2, x_5), u_5 = (x_5, x_6)$.

Рассмотрим ещё один **пример**. С использованием алгоритмов Краскала и Прима построим кратчайший остов графа, представленного на рис. 4.8.

Решение. Построим кратчайший остов с использованием *алгоритма Краскала*.

На первом шаге выбираем ребро, имеющее минимальный вес. Это ребро (x_6, x_7) с весом 3. Затем из оставшихся ребер выбираем и добавляем к остову ребро с наименьшим весом – это ребро (x_1, x_2) с весом 5. Затем выбираем ребро (x_3, x_4) с весом 7, (x_1, x_3) с весом 8. Далее следуют два ребра с весом 9. Ребро (x_2, x_3) мы выбрать не можем, так как его добавление к построенным ребрам образует цикл. Поэтому рассматриваем следующее ребро (x_3, x_6) с весом 9. Оно включается в остов. Затем рассматривается следующее по весу ребро и т.д. Построение заканчивается, когда остов будет содержать $n - 1$ ребро.

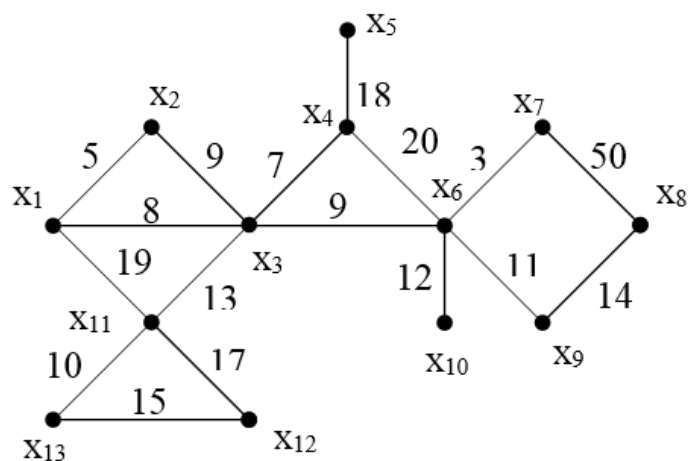
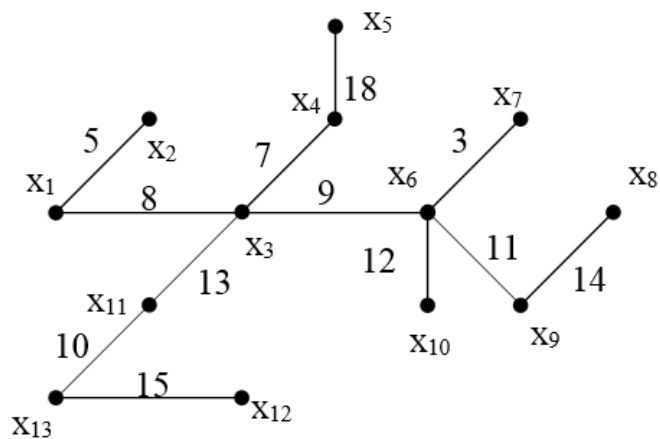


Рис. 4.8. Пример графа для построения кратчайшего остова

Последовательность выбора рёбер представлена в таблице.

Шаг	Рёбро	Вес
1	x_6, x_7	3
2	x_1, x_2	5
3	x_3, x_4	7
4	x_1, x_3	8
5	x_3, x_6	9
6	x_{11}, x_{13}	10
7	x_6, x_9	11
8	x_6, x_{10}	12
9	x_3, x_{11}	13
10	x_8, x_9	14
11	x_{12}, x_{13}	15
12	x_4, x_5	18
Суммарный вес		125

Кратчайший остов имеет следующий вид:



Теперь рассмотрим построение остова с использованием *алгоритма Прима*. На первом шаге выбираем ребро, имеющее минимальный вес. Это ребро (x_6, x_7) с весом 3. Следующее ребро мы можем выбрать только из тех рёбер, которые смежны с рёбром (x_6, x_7) . Ребро (x_1, x_2) с весом 5 мы выбрать не можем, так как оно не смежно с ребром (x_6, x_7) . Из всех рёбер, смежных с ребром (x_6, x_7) , наименьший вес имеет ребро (x_3, x_6) (вес равен 9). Включаем его в остов. Затем рассматриваем рёбра, смежные с рёбрами (x_6, x_7) и (x_3, x_6) , включёнными в остов. Это ребро (x_3, x_4) с весом 7. Следующее ребро (x_1, x_3) с весом 8. Затем (x_1, x_2) с весом 5. Ребро (x_2, x_3) мы выбрать не можем, так как его добавление к построенным рёбрам образует цикл. Последовательность добавления рёбер представлена в таблице.

Шаг	Ребро	Вес
1	x_6, x_7	3
2	x_3, x_6	9
3	x_3, x_4	7
4	x_1, x_3	8
5	x_1, x_2	5
6	x_6, x_9	11
7	x_6, x_{10}	12
8	x_3, x_{11}	13
9	x_{11}, x_{13}	10
10	x_8, x_9	14
11	x_{12}, x_{13}	15
12	x_4, x_5	18
Суммарный вес		125

Можно заметить, что остовные деревья, построенные по алгоритмам Краскала и Прима, одинаковы. Если в графе есть несколько рёбер с одинаковым весом, то деревья могут получиться разные, однако их суммарный вес будет равным.

4.3. Эйлеровы и гамильтоновы графы

Цикл (цепь) в графе G называется *эйлеровым* (эйлеровой), если он (она) проходит по одному разу через каждое ребро этого графа. Связный граф называется *эйлеровым*, если он содержит эйлеров цикл, и *полуэйлеровым*, если в нем существует незамкнутая эйлерова цепь. Согласно *теореме Эйлера*, связный граф является эйлеровым, тогда и только тогда, когда степени всех его вершин четны. Если число вершин нечетной степени в связном графе равно 2, то данный граф содержит эйлерову цепь. При этом вершины нечетной степени являются начальной и конечной вершинами цепи. На рис. 4.9, а, б изображены по-

луэйлеров и эйлеров графы. Порядок обхода этих графов показан стрелками с номерами соответствующих шагов (движение начинается из вершины x_1).

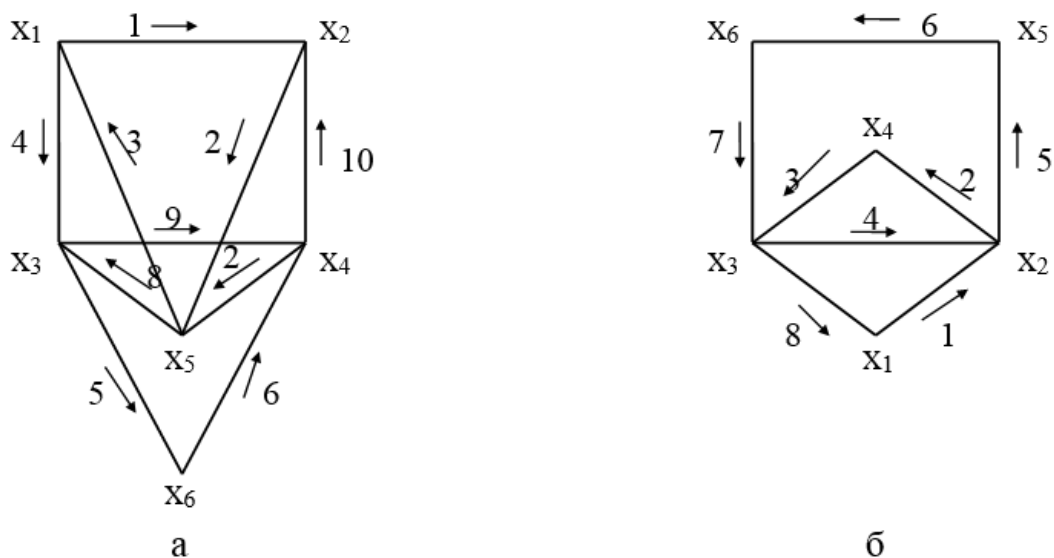


Рис. 4.9. Пример полуэйлерова и эйлерова графов

Примером прикладного значения нахождения эйлерова цикла в графе является задача выбора рационального маршрута исполнительного устройства технологических и чертежных автоматов (таких, как станки с ЧПУ, координатографы, графопостроители), решение которой позволяет находить пути движения инструмента без холостых ходов.

Для построения эйлерова цикла может быть использован *алгоритм Флери*. Необходимо начинать построение цикла с некоторой произвольной вершины графа и каждый раз вычеркивать пройденное ребро. По мосту можно проходить только в случае, когда нет других возможностей. Алгоритм может быть применим как к ориентированным, так и к неориентированным графам.

Цикл (цепь) в графе G называется *гамильтоновым* (гамильтоновой), если он (она) проходит через каждую вершину графа ровно один раз. Граф G называется гамильтоновым, если он содержит гамильтонов цикл (рис. 4.10).

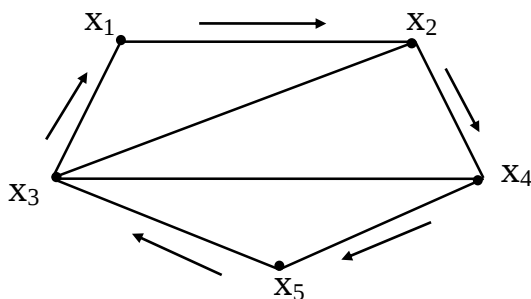


Рис. 4.10. Пример гамильтонова графа

В отличие от эйлеровых циклов, для гамильтоновых циклов неизвестен общий критерий существования. Очевидными условиями существования гамильтонова цикла в графе является связность этого графа и отсутствие в нем точек сочленения. Полный граф всегда имеет гамильтонов цикл.

Алгоритм построения гамильтоновых циклов в графе

К наиболее распространенным методам построения гамильтоновых циклов в графе можно отнести метод Робертса и Флореса.

Метод Робертса и Флореса построения гамильтонова цикла заключается в следующем.

1. Строится матрица M с элементами m_{ij} , число строк в которой равно максимальной степени вершин графа, число столбцов равно количеству вершин n . Элемент m_{ij} - i -я вершина, (например x_k) смежная с вершиной x_j . Вершины в столбцах матрицы M упорядочены.

2. $p = x_1$, где x_1 - начальная вершина. $S = \{x_1\}$, где S - множество вершин строящегося гамильтонова цикла.

3. Если в столбце матрицы M , соответствующем вершине p , существует возможная вершина (под возможной понимается вершина, ещё не принадлежащая S), то переход к шагу 4, если нет, то переход к шагу 7.

4. В столбце, соответствующем вершине p , выбирается первая возможная вершина x_k ; эта вершина присоединяется к множеству S и $p = x_k$.

5. Если мощность множества S равна $|S| = n$, то найдена гамильтонова цепь, переход к шагу 6, а если $|S| < n$, то переход к шагу 3.

6. Если существует дуга или ребро (p, x_1) , то найден гамильтонов цикл. Если надо найти все гамильтоновы циклы, то переход к шагу 7; иначе останов алгоритма.

7. Возвращение. Из множества S удаляется последняя включённая вершина x_r . Если при этом $S = \emptyset$ (множество S пустое), то следует остановка алгоритма, т.е. все цепи и циклы найдены (или нет).

Если $S \neq \emptyset$, то $p = x_{r-1}$, где $(r-1)$ - номер вершины, включенной в гамильтонов цикл.

8. Если в столбце p существуют возможные вершины, т.е. вершины, следующие за x_r , то переход к шагу 4. В противном случае $x_r = x_{r-1}$ и переход к шагу 7.

Пример. Для графа G , изображенного на рис.4.11, построить все гамильтоновы цепи и циклы.

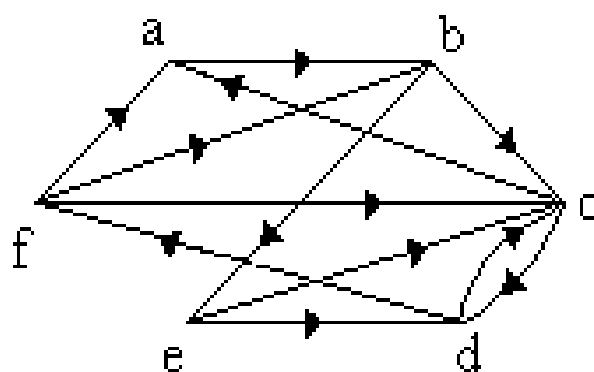


Рис.4.11. Граф для иллюстрации алгоритма Робертса и Флореса

Матрица М приводится ниже, вершины в каждом столбце расположены в алфавитном порядке:

$$M = \begin{matrix} & \begin{matrix} a & b & c & d & e & f \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} b & c & a & c & c & a \\ - & e & d & f & d & b \\ - & - & - & - & - & c \end{pmatrix} \end{matrix}$$

Поиск всех гамильтоновых циклов производится следующим образом (вершина а выбирается в качестве начальной вершины):

Шаг	Множество S	Комментарии
1	a	Добавляем первую возможную вершину в столбце a (то есть вершину b)
2	a, b	Добавляем первую возможную вершину в столбце b (то есть вершину c)
3	a, b, c	Первая вершина (a) в столбце c не является возможной ($a \in S$), добавляем следующую вершину в столбце (то есть вершину d)
4	a, b, c, d	Добавляем вершину f
5	a, b, c, d, f	В столбце f нет возможной вершины. Возвращение
6	a, b, c, d	В столбце d не существует возможной вершины, следующей за f. Возвращение
7	a, b, c	Аналогично предыдущему. Возвращение.
8	a, b	Добавляем вершину e
9	a, b, e	Добавляем вершину c

10	a, b, e, c	Добавляем вершину d
11	a, b, e, c, d	Добавляем вершину f
12	<u>a, b, e, c, d, f</u>	<u>Гамильтонов цикл</u> , замыкающийся дугой(f, a). Возвращение
13	a, b, e, c, d	Возвращение
14	a, b, e, c	Возвращение
15	a, b, e	Добавляем вершину d
16	a, b, e, d	Добавляем вершину f
17	a, b, e, d, f	Добавляем вершину c
18	<u>a, b, e, d, f, c</u>	<u>Гамильтонов цикл</u> , замыкающийся дугой (c, a). Возвращение
19	a, b, e, d, f	Возвращение
20	a, b, e, d	Возвращение
21	a, b, e	Возвращение
22	a, b	Возвращение
23	a	Возвращение
24	∅	Конец поиска

Задача определения гамильтонова цикла в графе имеет важное практическое значение. К ней сводятся, например, задачи отыскания монтажных цепей, определения оптимального маршрута для исполнительного устройства автоматов проводного монтажа. С понятием гамильтоновых циклов тесно связана так называемая задача *коммивояжера*: в взвешенном графе G определить гамильтонов цикл минимальной длины (иными словами, коммивояжер должен совершить поездку по городам и вернуться обратно, побывав в каждом городе ровно один раз и при этом стоимость такой поездки должна быть минимальной). К задаче коммивояжера сводится большое количество важных прикладных задач: задачи оптимальной маршрутизации, проектирования распределенных систем (вычислительных, электрических, транспортных сетей и так далее), составления расписания выполнения операций на машинах с минимумом переключений и т. д.

4.4. Раскраски графа

Вершинной r - раскраской графа называется присвоение его вершинам r различных цветов. Если при этом никакие две смежные вершины не окрашиваются одним цветом, раскраска называется *правильной*. Таким образом, правильная r - раскраска графа – это разбиение множества его вершин X на r непересекающихся подмножеств $X_1...X_r$, таких, что внутри каждого подмножества не содержится смежных вершин. Подмножества $X_1...X_r$ называются *цветными*

классами. Правильная раскраска графа с использованием минимального количества цветов называется *минимальной*. Число цветов, используемых при минимальной раскраске графа, называется *хроматическим числом графа* $\chi(G)$.

На рис. 4.11, а приведена правильная 3-раскраска графа, на рис. 4.12, б его минимальная раскраска. Хроматическое число $\chi(G)=2$.

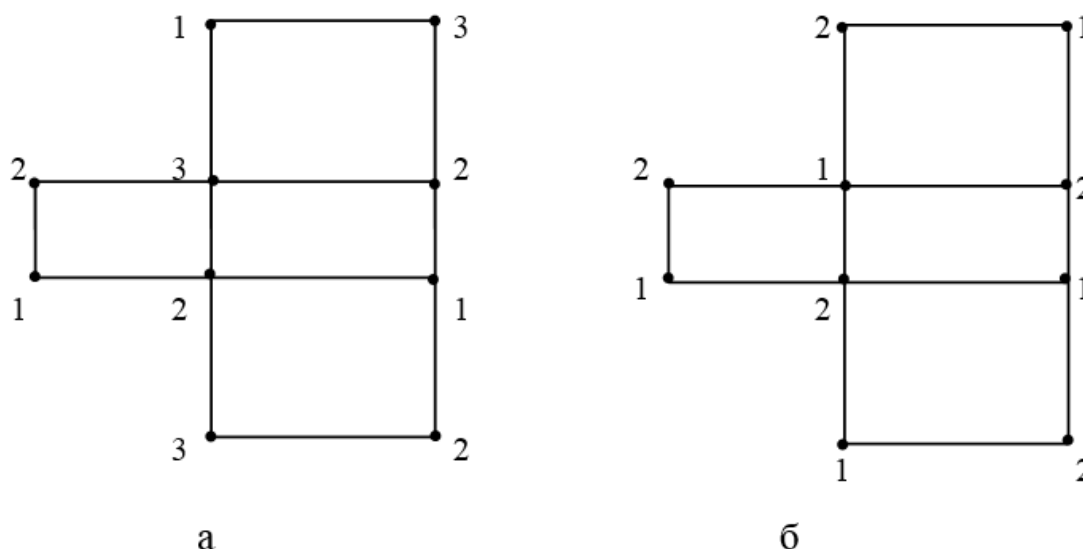


Рис. 4.12. Правильная 3-раскраска графа и его минимальная раскраска

На практике часто используются приближённые эвристические алгоритмы раскраски графа.

Эвристический алгоритм раскраски графа

Эвристический алгоритм раскраски графа G дает приближенную оценку хроматического числа. Сначала составляется список вершин графа, упорядоченный по убыванию их степеней. Первая вершина окрашивается в цвет 1 и удаляется из списка. Далее список просматривается, в цвет 1 окрашиваются и удаляются из него все вершины, не смежные с первой выбранной и между собой. Далее из оставшихся вершин списка выбирается первая, окрашивается в цвет 2 и удаляется. Процесс продолжается до тех пор, пока не будут окрашены все вершины. При этом приближенное значение хроматического числа будет равно количеству используемых при работе алгоритма цветов. Пример работы данного алгоритма рассмотрен в главе 5.

Аналогично вершинной g -раскраске определяется и *реберная* k -раскраска графа. Минимальное число k , при котором граф является реберно k -раскрашиваемым (то есть допускающим такую раскраску, при которой смежные ребра имеют разные цвета), называется *хроматическим индексом* графа G .

Задачи раскраски графа имеют широкий круг практических приложений. К задачам правильной раскраски сводятся задачи составления расписаний (с

ограничением на одновременное выполнение работ), задачи распределения оборудования (с ограничением на его одновременное использование на нескольких работах), составления графиков осмотра (проверки) оборудования с учетом ограничений на совместимость осмотров и т.д. Характерной особенностью задач раскраски является существование объектов, которые по каким-либо причинам не могут быть объединены в одну группу.

4.5. Определение кратчайших путей и маршрутов в графе

Одной из важнейших задач теории графов является задача определения кратчайшего пути во взвешенном графе. Граф называется *взвешенным*, если каждому его ребру (дуге) (x_i, x_j) ставится в соответствие числовая характеристика $w(x_i, x_j)$, называемая *весом*. Физический смысл этой характеристики зависит от особенностей решаемой задачи. Например, если граф является моделью транспортной сети, то веса могут характеризовать расстояния между населёнными пунктами или стоимость перевозки груза между ними. В информационно-телекоммуникационных сетях веса могут характеризовать скорость передачи данных. Задачи маршрутизации имеют очень важное значение в различных предметных областях.

Алгоритм Дейкстры определения кратчайшего пути между двумя фиксированными вершинами

Алгоритм Дейкстры — алгоритм поиска кратчайшего пути во взвешенном графе между двумя заданными вершинами s и t . При этом веса должны быть неотрицательны.

На каждой итерации алгоритма каждая вершина x_i графа имеет метку $l(x_i)$. Эта метка равна текущей длине кратчайшего пути от начальной вершины s до вершины x_i . Метка может быть постоянной и временной. Временные метки в процессе работы алгоритма могут уменьшаться, постоянные не изменяются.

Кроме $l(x_i)$, с вершинами графа связывается еще одна метка $Q(x_i)$. На каждой итерации $Q(x_i)$ является номером вершины, предшествующей x_i в кратчайшем (s, x_i) -пути.

После того, как последняя вершина t получила постоянную метку, с помощью меток $Q(x)$ легко указать последовательность вершин, составляющих кратчайший (S, t) -путь:

$$(s, \dots, Q^n(t), \dots, Q(Q(t)), Q(t), t), \\ Q^n(t) = Q(Q(\dots Q(t))) \quad (n \text{ раз}).$$

Основные шаги алгоритма:

1. Положить $l(s)=0$ и считать эту метку постоянной. Положить $l(x_i)=\infty$ для всех $x_i \neq s$, и считать эти метки временными. Положить $p = s$.

2. Обновление пометок. Для всех вершин x_i с временными метками, смежных с вершиной p , изменить пометки в соответствии с правилом:

$$l(x_i) = \min \{ l(x_i); l(p) + w(p, x_i) \}.$$

Если при этом метка $l(x_i)$ уменьшилась, присвоить метку $Q(x_i) = p$ (запоминаем вершину, которая в кратчайшем пути будет предшествовать вершине x_i).

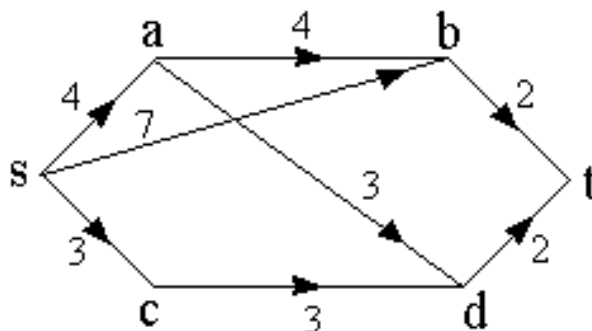
3. Если $l(x_i)=\infty$ для всех вершин x_i , пометки которых временные, то в исходном графе отсутствуют пути из вершины s в вершины с временными метками. Останов алгоритма. В противном случае переход к шагу 4.

4. Превращение пометок в постоянные. Среди всех вершин с временными метками найти такую вершину x_i^* , для которой $l(x_i^*) = \min l(x_i)$ (метка минимальная) и считать эту пометку постоянной. Положить $p=x_i^*$. Пометку $Q(x_i^*)$ также считать постоянной.

5. Если $p \neq t$, перейти к шагу 2, а если $p = t$, то $l(p)$ - длина кратчайшего пути из s в t .

После определения длины кратчайшего пути сам кратчайший путь восстанавливается по постоянным меткам $Q(x_i)$ так, как было указано выше.

Пример. Для взвешенного орграфа найти кратчайший путь из вершины s в вершину t .



1. Помечаем в соответствии с алгоритмом вершины графа:

$l(s) = 0$. Метки остальных вершин равны ∞ . Вершине s приписываем постоянную пометку, т.е. $p = s$.

В дальнейшем при решении постоянные метки будем подчёркивать.

2. Из вершины s помечаем вершины, смежные с ней:

$$l(a) = \min \{ \infty, 0 + 4 \} = 4,$$

$$l(b) = \min \{ \infty, 0 + 7 \} = 7,$$

$$\underline{l(c) = \min \{ \infty, 0 + 3 \} = 3},$$

$$l(d) = \infty ,$$

$$l(t) = \infty .$$

(Так как вершины d и t, не смежны с вершиной s, то их метки не изменяются. Переписываем их с предыдущего шага).

У вершин a, b, и c уменьшились пометки, следовательно запоминаем предшествующую вершину: $Q(a) = s$, $Q(b) = s$, $\underline{Q(c) = s}$.

Минимальную метку имеет вершина c. Следовательно, её метка $\underline{l(c)}$ становится постоянной. Пометка $\underline{Q(c)=s}$ также становится постоянной. (эти метки подчёркнуты).

Обозначим $p = c$. И на следующем шаге будем помечать вершины из вершины c.

Так как постоянные метки уже не изменятся до конца работы алгоритма, при переходе на следующий шаг можно их не переписывать.

3. Из вершины c помечаем остальные вершины:

$$\underline{l(a) = \min \{ 4, 3 + \infty \} = 4},$$

$$l(b) = \min \{ 7, 3 + \infty \} = 7,$$

$$l(d) = \min \{ \infty, 3 + 3 \} = 6,$$

$$l(t) = \infty .$$

У вершины d уменьшилась пометка, следовательно $Q(d) = c$.

Вершине a приписываем постоянную пометку, т.е. $p = a$. Пометка $\underline{Q(a)=s}$ становится постоянной.

4. Из вершины a помечаем остальные вершины:

$$l(b) = \min \{ 7, 4 + 4 \} = 7,$$

$$\underline{l(d) = \min \{ 6, 4 + 3 \} = 6},$$

$$l(t) = \infty .$$

Вершине d приписываем постоянную пометку, т.е. $p = d$. Пометка $\underline{Q(d)=c}$ становится постоянной.

5. Из вершины d помечаем остальные вершины:

$$\underline{l(b) = \min \{ 7, 6 + \infty \} = 7},$$

$$l(t) = \min \{ \infty, 6 + 2 \} = 8,$$

У вершины t уменьшилась пометка, следовательно $Q(t) = d$.

Вершине b приписываем постоянную пометку, т.е. $p=b$. Пометка $\underline{Q(b)=s}$ становится постоянной.

6. Из вершины b помечаем вершину t:

$$l(t) = \min \{8, 7 + 2\} = 8.$$

Метки $l(t)$ и $Q(t)=d$ становятся постоянными.

7. Восстанавливаем по постоянным меткам Q кратчайший путь из s в t:
Путь восстанавливается из конечной вершины t в начальную:

$$Q(t)=d; Q(d)=c; Q(c)=s.$$

Таким образом, получен путь scdt длиной 8.

Работу алгоритма можно проиллюстрировать таблицей:

Вершина	1	2	3	4	5	6
s	<u>0</u>	-	-	-	-	-
a	∞	4	<u>4</u>	-	-	-
b	∞	7	7	7	<u>7</u>	-
c	∞	<u>3</u>	-	-	-	-
d	∞	∞	6	<u>6</u>	-	-
t	∞	∞	∞	∞	8	<u>8</u>

Алгоритм Форда

Алгоритм Дейкстры имеет определённые ограничения в применении. Он не может быть использован, если веса некоторых дуг отрицательны (может быть получен неверный результат). В этом случае для поиска кратчайшего пути в графе с отрицательным весом применяется алгоритм Форда, являющийся модификацией алгоритма Дейкстры. Необходимые модификации состоят в следующем:

1. На шаге 2 алгоритма Дейкстры пересчет пометок осуществляется для всех вершин графа (как для вершин с временными метками, так и для вершин с постоянными метками).

2. Если для некоторой вершины x_i с постоянной меткой происходит уменьшение пометки $l(x_i)$, то эта пометка становится временной, а пометка $Q(x_i)$ уничтожается.

3. Алгоритм заканчивает работу тогда, когда все вершины получают постоянные метки и после выполнения шага 2 алгоритма Дейкстры ни одна из меток не меняется.

Пример. Для взвешенного графа, изображенного на рис.4.13, найти кратчайший путь между вершинами s и t.

Очевидно, что в данном случае алгоритм Дейкстры дал бы неверный результат. При решении алгоритмом Дейкстры был бы получен путь st длиной 1, хотя оптимальным является путь sdt длиной 0. Поэтому для решения будем использовать алгоритм Форда.

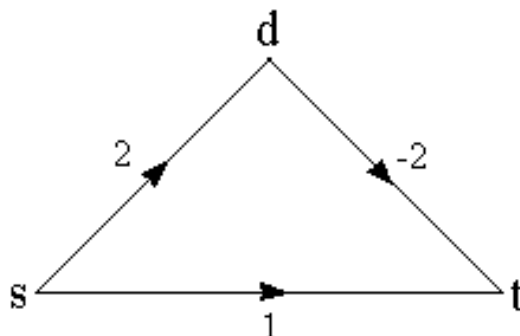


Рис.4.13. Граф для иллюстрации алгоритма Форда

Решение.

Вершины с постоянными метками будем подчеркивать.

1. Помечаем в соответствии с алгоритмом вершины графа:

$\underline{l(s)} = 0$, $l(d) = \infty$, $l(t) = \infty$. Вершине s приписываем постоянную пометку.

То есть $p=s$.

2. Из вершины s помечаем остальные вершины:

$\underline{l(s)} = 0$, $l(d) = 2$, $l(t) = 1$. У вершин d и t уменьшились пометки, следовательно, $Q(d) = s$, $Q(t) = s$. Вершине t приписываем постоянную пометку. То есть $p = t$.

3. Из вершины t помечаем остальные вершины:

$\underline{l(s)} = 0$, $l(d) = 2$, $\underline{l(t)} = 1$. Вершине d приписываем постоянную пометку. То есть $p = d$.

4. Из вершины d помечаем остальные вершины:

$\underline{l(s)} = 0$, $\underline{l(d)} = 2$, $l(t) = 0$. У вершины t уменьшилась пометка, следовательно, $Q(t)=d$. Пометка вершины t стала временной. Пометка $Q(t) = s$ уничтожается.

5. Так как имеется всего одна вершина с временной пометкой (вершина t), то ей приписывается постоянная пометка, т.е. $p = t$. Из вершины t помечаем остальные вершины:

$\underline{l(s)} = 0$, $\underline{l(d)} = 2$, $\underline{l(t)} = 0$. Пометки вершин s и d не изменились.

6. Выполняем еще раз шаг 2 алгоритма Дейкстры:

$\underline{l(s)} = 0$, $\underline{l(d)} = 2$, $\underline{l(t)} = 0$. Останов алгоритма, так как все вершины получили постоянные пометки и после выполнения очередного шага ни одна из пометок не изменилась.

Восстанавливаем по меткам Q кратчайший путь из s в t : путь sdt длиной 0.

Алгоритм Флойда определения кратчайших путей между всеми парами вершин графа

Пусть дан взвешенный оргграф с n вершинами и матрицей весов W , элементы которой определяются следующим образом:

$$w_{ij} = \begin{cases} \text{вес } w(x_i, x_j), & \text{если в графе существует дуга } \langle x_i, x_j \rangle \\ 0, & \text{если } i = j, \\ \infty, & \text{если в графе не существует дуги } \langle x_i, x_j \rangle. \end{cases}$$

Пронумеруем вершины графа от 1 до n . Обозначим W^k матрицу с элементами w_{ij}^k , каждый из которых равен длине кратчайшего пути из вершины i в вершину j , который может содержать в качестве промежуточных вершин только первые k вершин графа. По матрице W^0 вычисляется матрица W^1 и т.д. до тех пор, пока не будет определена матрица W^n , содержащая кратчайшие пути между всеми вершинами графа.

На каждой итерации вместе с матрицей W^k строится матрица P^k , каждый элемент которой p_{ij}^k равен номеру вершины, предшествующей вершине j в текущем ij пути.

Основные шаги алгоритма:

1. Пронумеровать вершины графа целыми числами. $k=0$. Определить матрицу W^0 . Определить матрицу P^0 , $p_{ij}^0 = i$, $\forall i \neq j$, $i, j = 1 \dots n$ и $p_{ii}^0 = 0$, $\forall i = j = 1 \dots n$.
2. Если $k = n$, работа алгоритма закончена (W^n - эта матрица весов кратчайших путей между всеми парами вершин графа, определяемых с помощью матрицы P^n). Если $k \neq n$, то $k = k+1$, переход к шагу 3.
3. Вычислить для всех $i, j = 1 \dots n$ элементы

$$w_{ij}^k = \min \{ w_{ik}^{k-1} + w_{kj}^{k-1}, w_{ij}^{k-1} \}.$$

Если $w_{ij}^{k-1} < w_{ik}^{k-1} + w_{kj}^{k-1}$, то $p_{ij}^k = p_{ij}^{k-1}$. Иначе $p_{ij}^k = p_{kj}^{k-1}$.

4. Если для некоторого $1 \leq q \leq n$ элемент с $w_{qq}^k < 0$, то в графе имеется контур отрицательной длины и работа алгоритма завершается. Иначе перейти к шагу 2.

Номера вершин, включаемых в кратчайший путь, определяются следующим образом:

$$\begin{aligned} (i, \dots, j_3, j_2, j_1, j), \\ j_1 = p_{ij}^n \\ j_2 = p_{ij_1}^n \text{ и т.д.} \end{aligned}$$

Пример. Дан взвешенный оргграф G, который содержит положительные и отрицательные веса (рисунок 4.14). Найти кратчайшие пути между всеми вершинами графа.

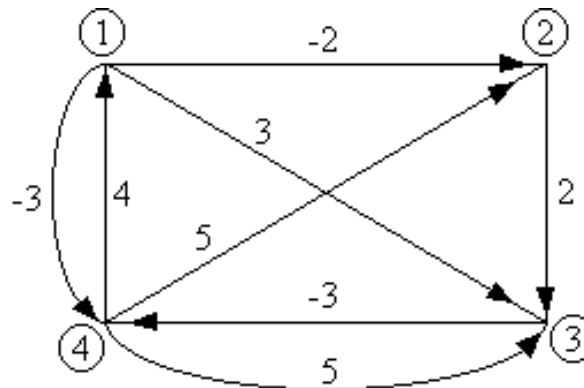


Рис. 4.14. Пример графа для поиска кратчайших путей

Решение.

Пронумеруем вершины графа целыми числами (1, 2, 3, и 4) и определим матрицы W^0 и P^0 :

$$W^0 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & -2 & 3 & -3 \\ \infty & 0 & 2 & \infty \\ \infty & \infty & 0 & -3 \\ 4 & 5 & 5 & 0 \end{pmatrix} \end{matrix} \quad P^0 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 2 & 0 & 2 & 2 \\ 3 & 3 & 0 & 3 \\ 4 & 4 & 4 & 0 \end{pmatrix} \end{matrix}$$

Согласно алгоритму определяем матрицы W^1 и P^1 .

$$W^1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & -2 & 3 & -3 \\ \infty & 0 & 2 & \infty \\ \infty & \infty & 0 & -3 \\ 4 & 2 & 5 & 0 \end{pmatrix} \end{matrix} \quad P^1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 2 & 0 & 2 & 2 \\ 3 & 3 & 0 & 3 \\ 4 & 1 & 4 & 0 \end{pmatrix} \end{matrix}$$

W^2 и P^2 :

$$W^2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & -2 & 0 & -3 \\ \infty & 0 & 2 & \infty \\ \infty & \infty & 0 & -3 \\ 4 & 2 & 4 & 0 \end{pmatrix} \end{matrix}$$

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

W^3 и P^3 :

$$W^3 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & -2 & 0 & -3 \\ \infty & 0 & 2 & -1 \\ \infty & \infty & 0 & -3 \\ 4 & 2 & 4 & 0 \end{pmatrix} \end{matrix}$$

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

W^4 и P^4 :

$$W^3 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & -2 & 0 & -3 \\ 3 & 0 & 2 & -1 \\ 1 & -1 & 0 & -3 \\ 4 & 2 & 4 & 0 \end{pmatrix} \end{matrix}$$

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

Таким образом, мы получили матрицу весов кратчайших путей W^4 между всеми вершинами графа. Определяем их с помощью матрицы P^4 .

Кратчайшие пути между вершинами:

- 1 и 2: $L(1, 2) = -2$, путь : 1-2;
- 1 и 3: $L(1, 3) = 0$, путь : 1-2-3;
- 1 и 4: $L(1, 4) = -3$, путь : 1-4;
- 2 и 1: $L(2, 1) = 3$, путь : 2-3-4-1;
- 2 и 3: $L(2, 3) = 2$, путь : 2-3;
- 2 и 4: $L(2, 4) = -1$, путь : 2-3-4;
- 3 и 1: $L(3, 1) = 1$, путь : 3-4-1;
- 3 и 2: $L(3, 2) = -1$, путь : 3-4-1-2
- 3 и 4: $L(3, 4) = -3$, путь : 3-4;
- 4 и 1: $L(4, 1) = 4$, путь : 4-1;
- 4 и 2: $L(4, 2) = 2$, путь : 4-1-2;
- 4 и 3: $L(4, 3) = 4$, путь : 4-1-2-3;

Заметим, что приведенные выше алгоритмы определения кратчайших путей могут быть применены и к неориентированным графам.

Алгоритмы определения кратчайших путей используются при решении задач маршрутизации в различных предметных областях (маршрутизация в транспортных сетях, маршрутизация в информационных сетях и т.д.)

4.6. Определение максимального потока в сети

Транспортной сетью называется ориентированный граф, удовлетворяющий следующим условиям:

1 В графе существует единственная вершина с нулевой полустепенью заходя (т.е. в эту вершину не заходит ни одна дуга). Эта вершина называется **источником** и обозначается s .

2 В графе существует единственная вершина с нулевой полустепенью исхода (т.е. из этой вершины не выходит ни одной дуги). Эта вершина называется **стоком** и обозначается t .

3 Каждой дуге графа $(x_i x_j)$ ставится в соответствие неотрицательное число c_{ij} , называемое **пропускной способностью дуги**.

Из источника s в сток t осуществляется перемещение единиц потока. При этом каждой дуге графа $(x_i x_j)$ ставится в соответствие величина потока f_{ij} . Для потоков в дугах выполняются следующие условия:

1 Величина потока в каждой дуге не превосходит пропускной способности дуги:

$$0 \leq f_{ij} \leq c_{ij}$$

2 Для всех промежуточных вершин транспортной сети $x_i \neq s \neq t$ сумма потоков по всем дугам, заходящим в вершину x_i , равна сумме потоков по всем дугам, исходящим из неё:

$$v = \sum_{x_j \in \Gamma(x_i)} f_{ij} = \sum_{x_k \in \Gamma^{-1}(x_i)} f_{ki}$$

3 Суммарный поток по дугам, исходящим из источника s , равен суммарному потоку по дугам, заходящим в сток t . Эта величина называется величиной потока в транспортной сети.

$$v = \sum_{x_j \in \Gamma(s)} f_{sj} = \sum_{x_k \in \Gamma^{-1}(t)} f_{kt}$$

Задача о максимальном потоке состоит в нахождении такого множества потоков по дугам, чтобы величина потока в сети была максимальной при условии отсутствия превышения пропускных способностей дуг. При этом также определяется распределение потоков по дугам транспортной сети.

Для решения данной задачи может быть использован алгоритм Форда-Фалкерсона.

Алгоритм Форда-Фалкерсона определения максимального потока в транспортной сети

Идея алгоритма состоит в следующем. Выбирается некоторый начальный поток из s в t и осуществляется поиск увеличивающего пути. Если увеличивающий путь найти не удаётся, выполнение алгоритма заканчивается, и текущий поток из s в t является максимальным. Если увеличивающий путь найден, то поток вдоль данного пути увеличивается до максимально возможного значения. Затем выполняется поиск нового увеличивающего пути, и т.д. (рисунок 4.15).

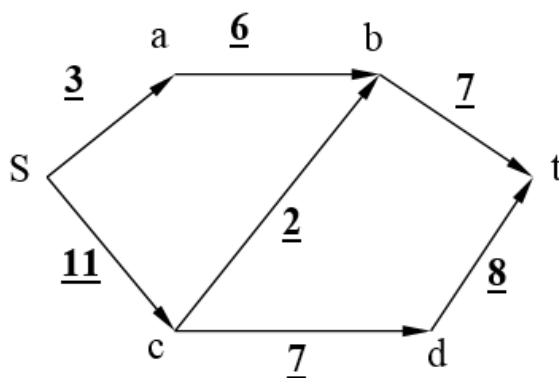


Рис. 4.15. Пример графа

Например, в представленном на рис. 4.15 графе имеются 3 пути:

Sabt

Scbt

Scdt

На первом шаге по пути $sabt$ мы можем переместить 3 единицы потока. Больше этот путь мы рассматривать не можем, так как в дуге (sa) величина потока равна пропускной способности дуги. Такая дуга называется **насыщенной**.

В дальнейшем на рисунках пропускная способность дуг будет выделяться жирным шрифтом, а рядом без выделения будет записана текущая величина потока в дуге. Нулевые значения записывать не будем.

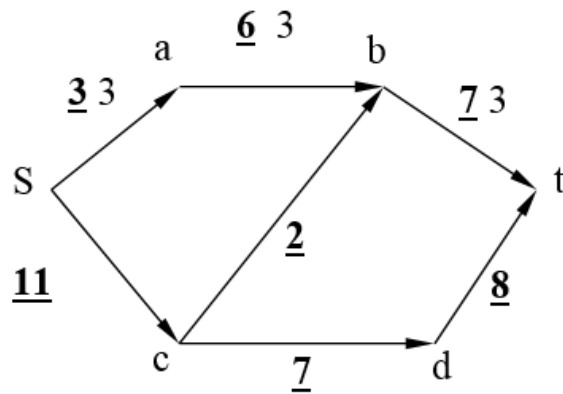


Рис. 4.16. Первый шаг определения максимального потока

По пути $scbt$ можно переместить 2 единицы потока (при этом дуга (cb) становится насыщенной):

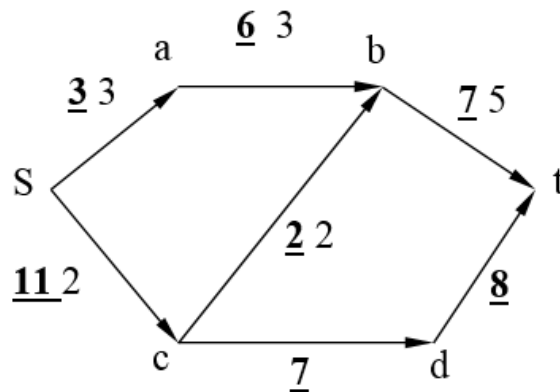


Рис. 4.17. Второй шаг определения максимального потока

Остаётся один увеличивающий путь $scdt$. По нему можно переместить 7 единиц потока:

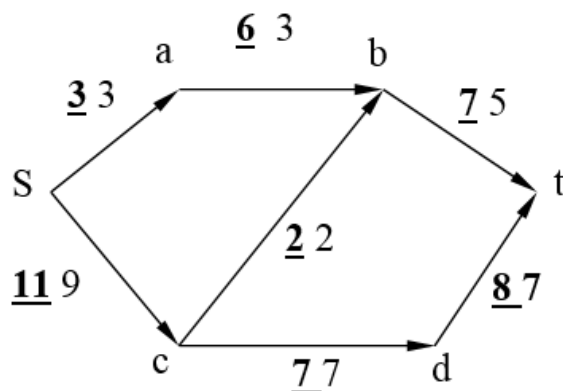


Рис. 4.18. Третий шаг определения максимального потока

В результате величина потока в сети равна 12.

Однако при большой размерности графа такой интуитивный способ решения задачи становится невозможным. Поэтому в процессе работы алгоритма вершинам присваиваются пометки. Пометка произвольной вершины x_i состоит из двух частей и имеет один из двух видов: $(+x_j, \delta)$ или $(-x_j, \delta)$. Часть $+x_j$ пометки первого типа означает, что поток допускает увеличение вдоль дуги (x_j, x_i) . Часть $-x_j$ пометки другого типа означает, что поток может быть уменьшен вдоль дуги (x_i, x_j) . В обоих случаях δ задаёт максимальную величину дополнительного потока, который может протекать от s к x_i .

Основные шаги алгоритма

А. Расстановка пометок

Шаг 1. Присвоить вершине s пометку $(+s, \delta(s) = \infty)$. Вершине s присвоена пометка, и она просмотрена, все остальные вершины без пометок.

Шаг 2. Взять некоторую непросмотренную вершину с пометкой; пусть её пометка будет $(\pm x_k, \delta(x_i))$.

(I) Каждой непомеченной вершине $x_j \in \Gamma(x_i)$, для которой $f_{ij} < c_{ij}$, присвоить пометку $(+x_i, \delta(x_j))$, где $\delta(x_j) = \min[\delta(x_i), c_{ij} - f_{ij}]$.

(II) Каждой непомеченной вершине $x_j \in \Gamma^{-1}(x_i)$, для которой $f_{ji} > 0$, присвоить пометку $(-x_i, \delta(x_j))$, где $\delta(x_j) = \min[\delta(x_i), f_{ji}]$. Теперь вершина x_i и помечена, и просмотрена, а вершины x_j , пометки которым присвоены в (I) и (II), являются непросмотренными.) Обозначить каким-либо способом, что вершина x_i просмотрена.

Шаг 3. Повторять шаг 2 до тех пор, пока либо вершина t будет помечена, и тогда перейти к шагу 4, либо t будет не помечена и никаких других пометок нельзя будет расставить; в этом случае алгоритм заканчивает работу с максимальной величиной потока.

Б. Увеличение потока

Шаг 4. Положить $x = t$ и перейти к шагу 5.

Шаг 5. (I) Если пометка в вершине x имеет вид $(+z, \delta(x))$, то изменить поток вдоль дуги (z, x) с $f(z, x)$ на $f(z, x) + \delta(x)$.

(II) Если пометка в вершине x имеет вид $(-z, \delta(x))$, то изменить поток вдоль дуги (x, z) с $f(x, z)$ на $f(x, z) - \delta(x)$.

Шаг 6. Если $z = s$, то стереть все пометки и вернуться к шагу 1, чтобы вновь начать расставлять пометки, но используя уже улучшенный поток, найденный на шаге 5. Если $z \neq s$, то положить $x = z$ и вернуться к шагу 5.

Пример решения задачи .

Найти максимальный поток от s к t для графа

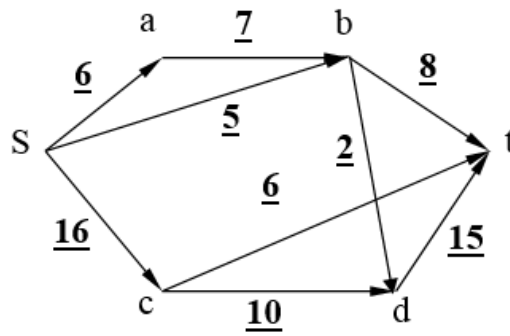


Рис. 4.19. Пример

Решение. В качестве начального возьмем поток с нулевыми значениями на всех дугах. Алгоритм работает следующим образом.

1 шаг

Расстановка пометок

Припишем вершине s пометку $(+s, \infty)$.

С вершиной s смежны три вершины: a, b, c. Можно выбрать любую. Выберем вершину a. Присвоим ей пометку $(+s, \min[\infty; 6-0]) = (+s, 6)$ (т.е. по дуге (s a) может пройти поток, равный 6).

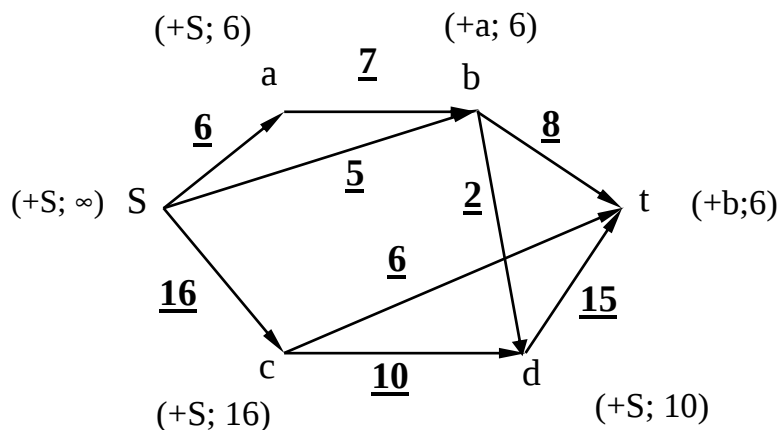
Вершину b можно пометить из вершины s или из вершины a (можно выбрать любой вариант). Пометим из вершины a. Тогда пометка вершины b будет равна $(+a, \min[6; 7-0]) = (+a, 6)$ (по дуге (ab) мог бы пройти поток, равный 7, но до вершины a дошло только 6 единиц потока).

Вершине c присваиваем пометку $(+s, 16)$.

Вершину d можно пометить из вершины b или из вершины c. Пометим из вершины c. Тогда метка вершины d будет равна $(+c, \min[16; 10-0]) = (+c, 10)$.

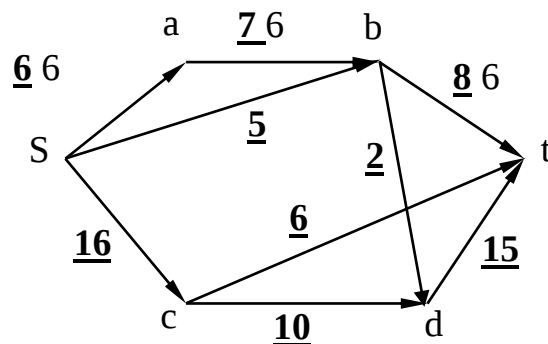
Вершину t можно пометить из вершины b или из вершины d. Выберем вершину b. Тогда метка вершины t будет равна $(+b, \min[6; 8-0]) = (+b, 6)$.

Граф с расставленными пометками будет выглядеть следующим образом:



Увеличение потока

Увеличиваем поток вдоль пути $sab t$ на 6 единиц:



2 шаг

Расстановка пометок

Припишем вершине s пометку $(+s, \infty)$.

Вершину a пометить не можем, так как дуга (sa) является насыщенной.

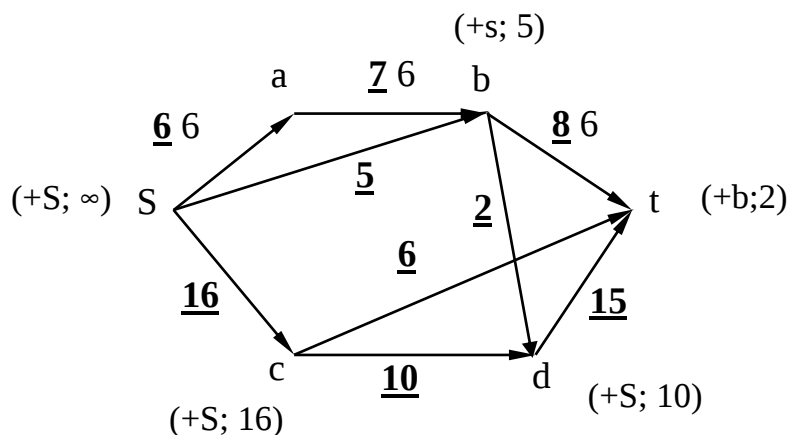
Вершину b можно пометить только из вершины s . Пометка вершины b будет равна $(+s, \min[\infty; 5-0]) = (+s, 5)$

Вершине c присваиваем пометку $(+s, 16)$.

Вершину d можно пометить из вершины b или из вершины c . Пометим из вершины c . Тогда метка вершины d будет равна $(+c, \min[16; 10-0]) = (+c, 10)$.

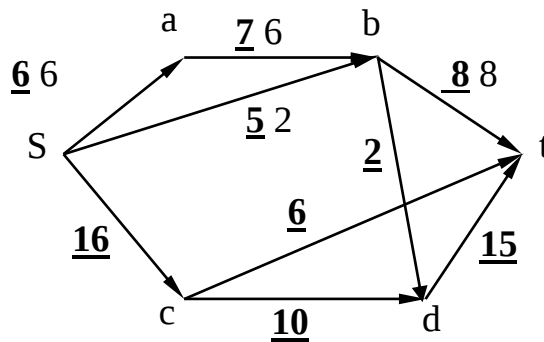
Вершину t можно пометить из вершины b или из вершины d . Выберем вершину b . Тогда метка вершины t будет равна $(+b, \min[5; 8-6]) = (+b, 2)$.

Граф с расставленными пометками будет выглядеть следующим образом:



Увеличение потока

Увеличиваем поток вдоль пути sbt на 2 единицы:



В результате дуга (bt) становится насыщенной.

3 шаг

Расстановка пометок

Припишем вершине s пометку $(+s, \infty)$.

Вершину a пометить не можем, так как дуга (sa) является насыщенной.

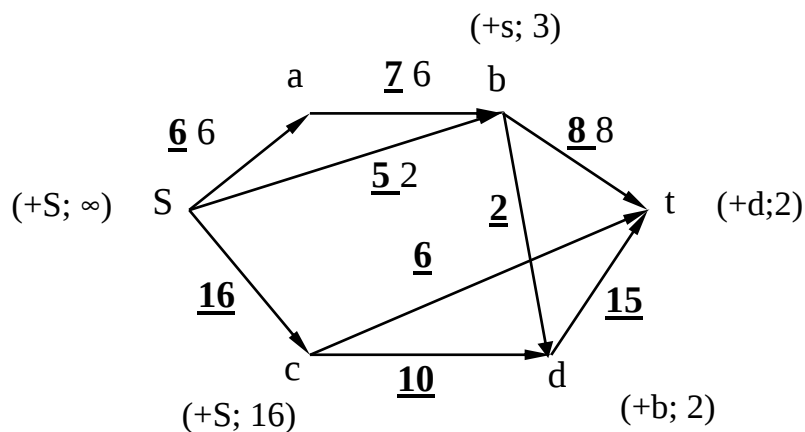
Вершину b можно пометить только из вершины s. Пометка вершины b будет равна $(+s, \min[\infty; 5-2]) = (+s, 3)$

Вершине c присваиваем пометку $(+s, 16)$.

Вершину d можно пометить из вершины b или из вершины c. Пометим из вершины b. Тогда метка вершины d будет равна $(+b, \min[3; 2-0]) = (+b, 2)$.

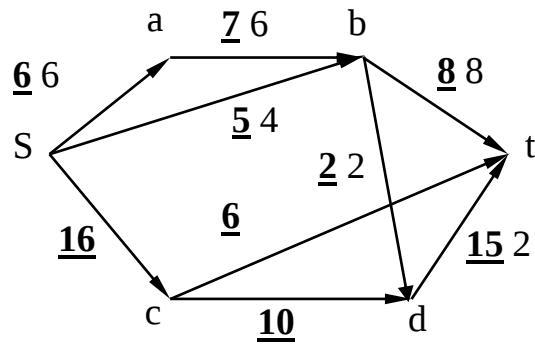
Вершину t можно пометить из вершины c или из вершины d. Выберем вершину d. Тогда метка вершины t будет равна $(+d, \min[2; 15-0]) = (+d, 2)$.

Граф с расставленными пометками будет выглядеть следующим образом:



Увеличение потока

Увеличиваем поток вдоль пути sbdt на 2 единицы:

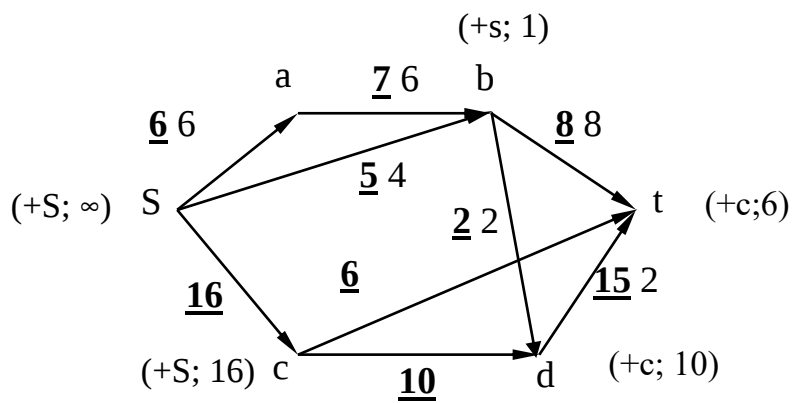


В результате дуга (bd) становится насыщенной.

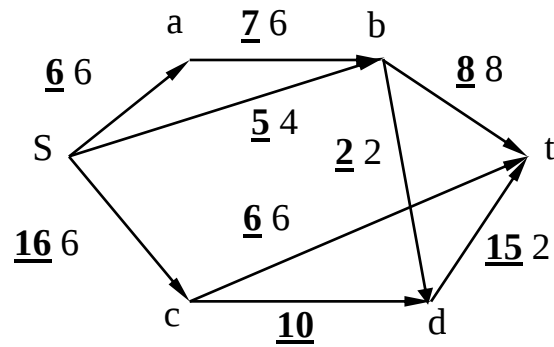
Дальнейшие два шага приводятся без пояснений.

4 шаг

Расстановка пометок

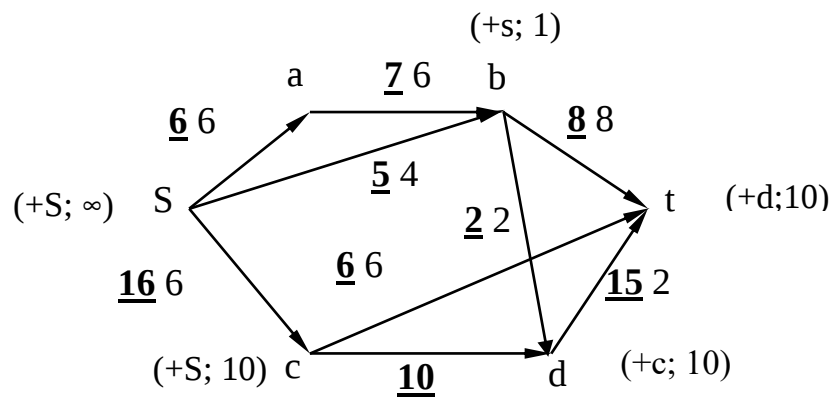


Увеличение потока

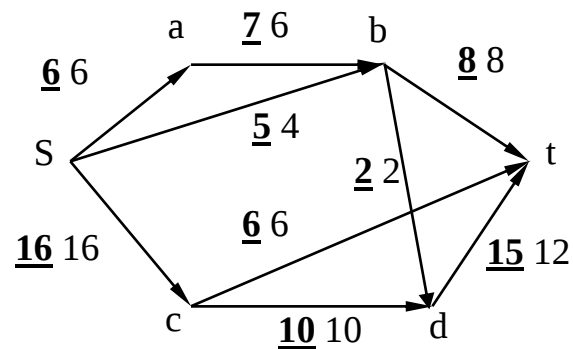


5 шаг

Расстановка пометок



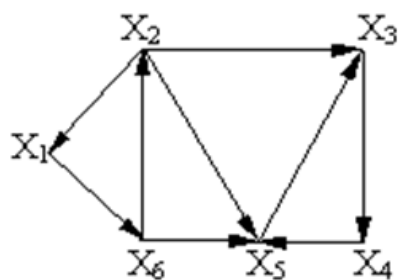
Увеличение потока



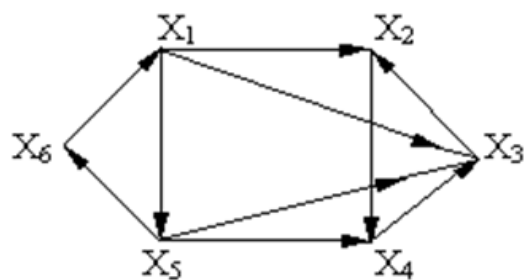
Дальнейшее увеличение потока невозможно. Задача решена. Величина потока равна 26 (складываем потоки по дугам, выходящим из источника или по дугам, заходящим в сток. Эти величины должны быть равны)

Упражнения

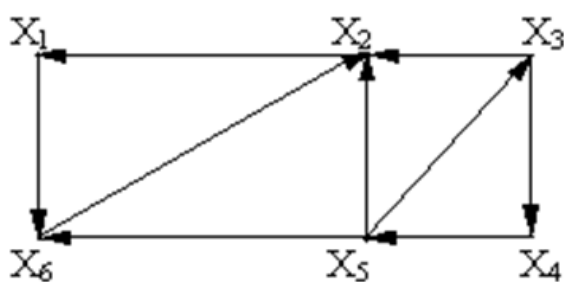
1. Определить сильные компоненты графов:



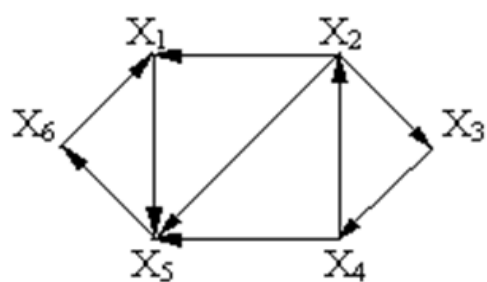
а



б



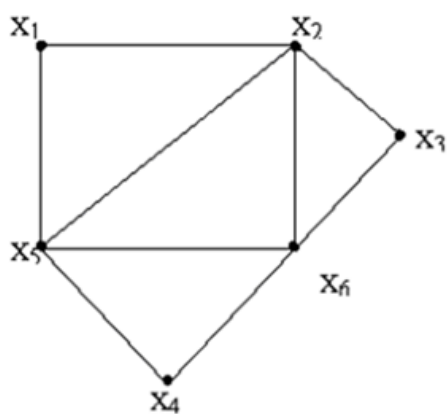
в



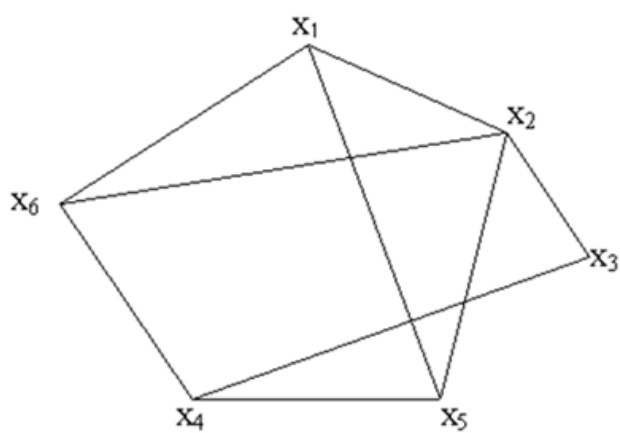
г

Рис. 4.20. Примеры для определения сильных компонент

2. Для графов, представленных на рис. 4.20, построить матрицы фундаментальных циклов.



а



б

Рис. 4.21. Примеры графов для построения матриц фундаментальных циклов

3. Построить остовные деревья графа с использованием алгоритмов поиска в глубину и в ширину.

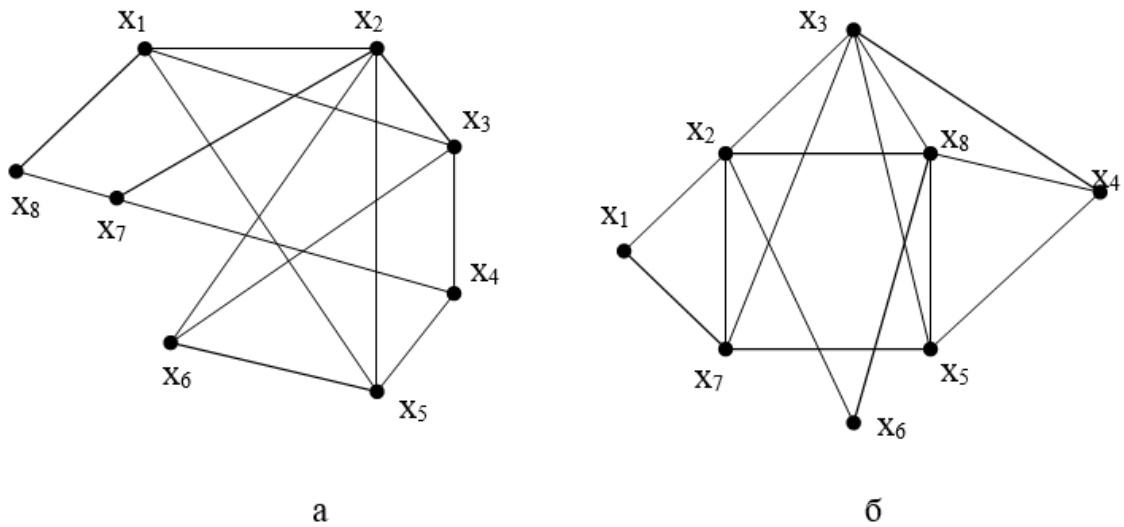


Рис. 4.22. Примеры графов для построения остовных деревьев

4. Построить кратчайшее остовное дерево графа с использованием алгоритмов Краскала и Прима.

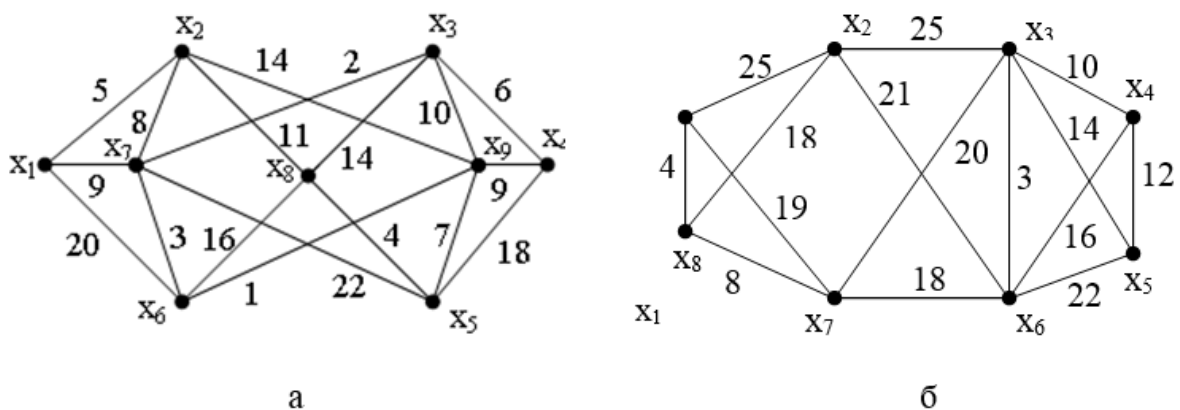


Рис. 4.23. Примеры графов для построения кратчайших остовов

5. Определить цикломатическое число графа на рис. 4.24.

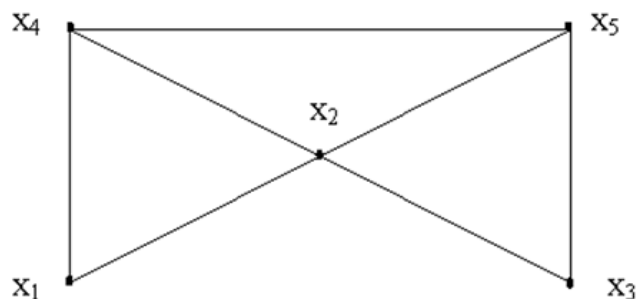


Рис. 4.24. Пример графа для определения цикломатического числа

6. Построить все гамильтоновы циклы и цепи графа с использованием алгоритма Роберта и Флореса.

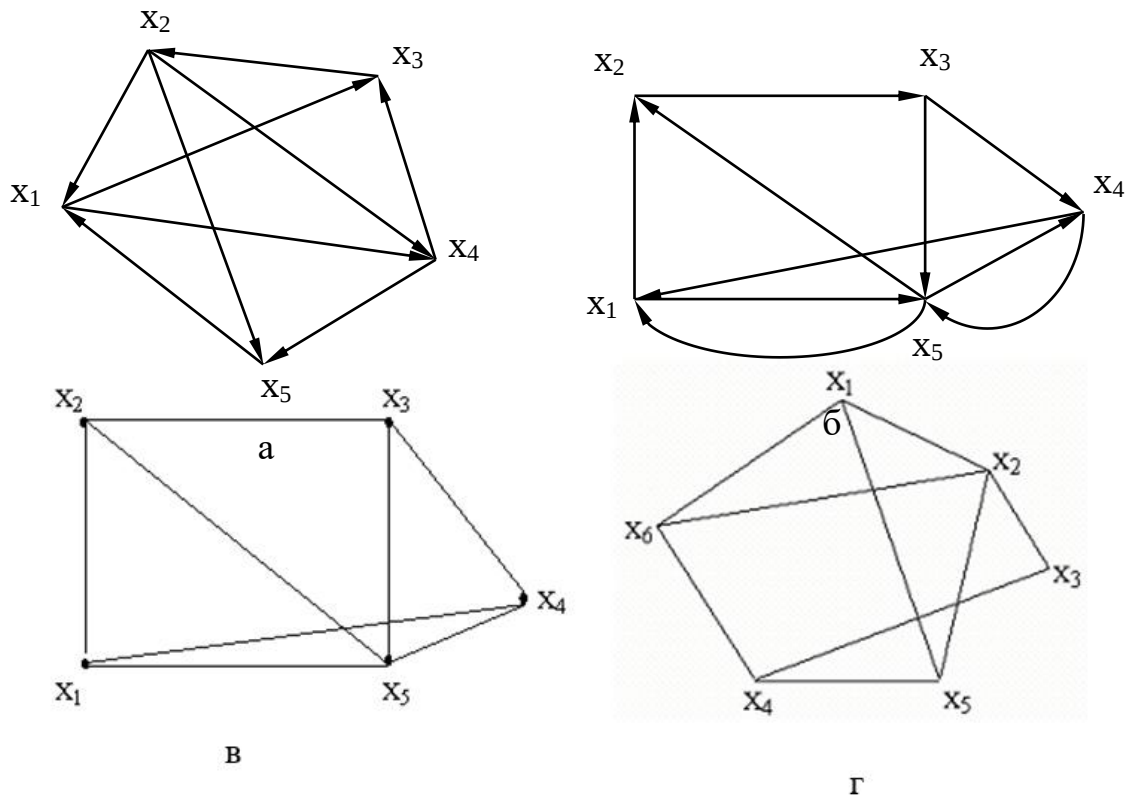


Рис. 4.25. Примеры графов для построения гамильтоновых циклов и цепей

7. Определить кратчайший путь от вершины s до вершины t с использованием алгоритма Дейкстры.

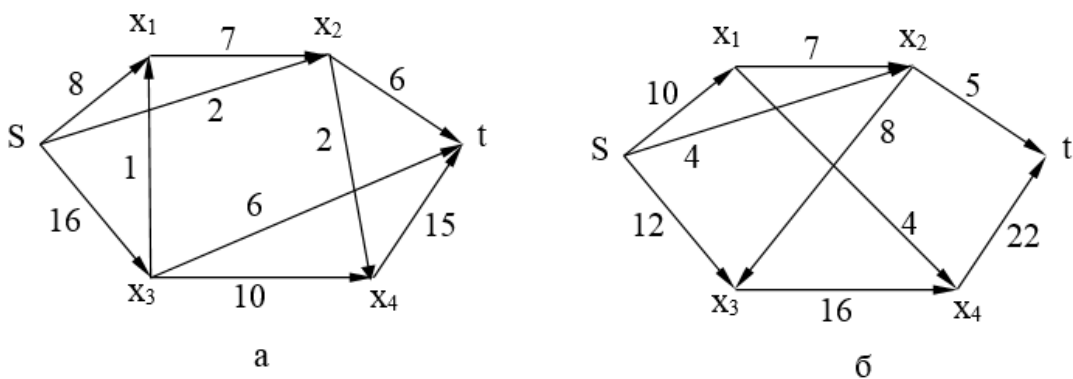


Рис. 4.26. Примеры графов для определения кратчайшего пути

8. Определить максимальный поток в сети с использованием алгоритма Форда-Фалкерсона.

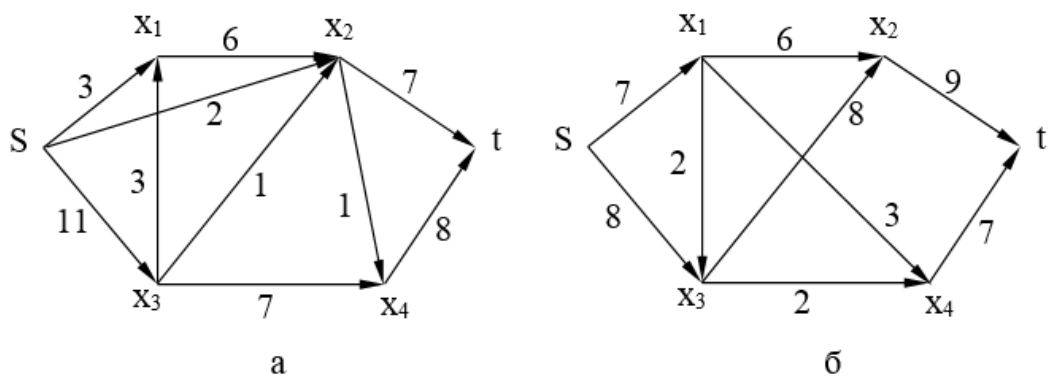


Рис. 4.27. Примеры графов для определения максимального потока

9. Определить кратчайшие пути между всеми парами вершин графов с использованием алгоритма Флойда

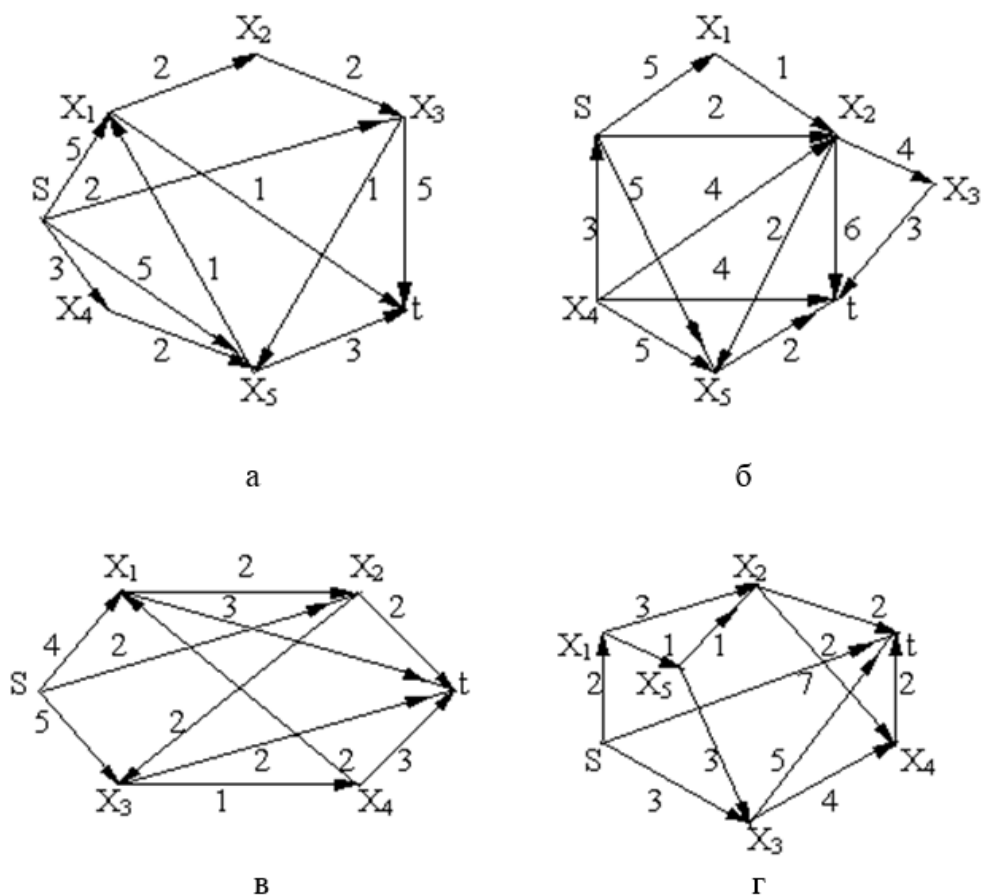


Рис. 4.28. Примеры графов для определения кратчайшего пути

10. Построить правильную раскраску графа. Определить хроматическое число графа.

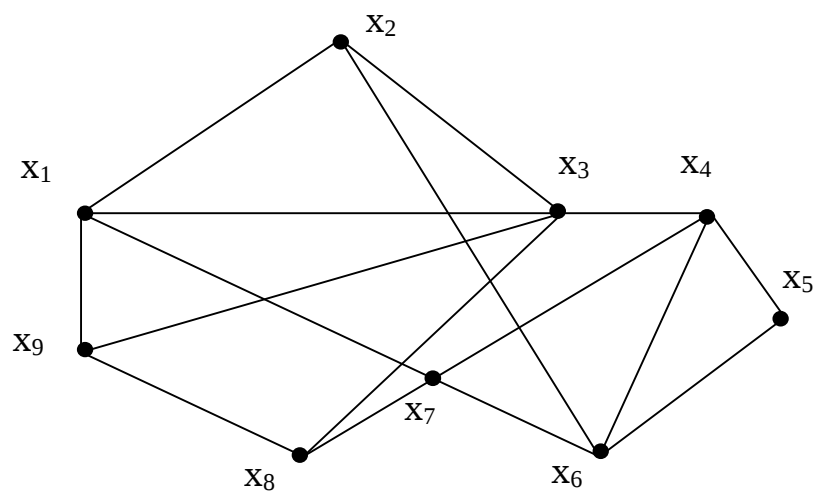


Рис. 4.29. Примеры графов для построения правильной раскраски

5. НЕКОТОРЫЕ ПРИЛОЖЕНИЯ ТЕОРИИ ГРАФОВ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ

Теория графов имеет широкий круг приложений в разных областях. Графы являются эффективным средством представления структурных моделей различных объектов и систем (электрических цепей, вычислительных сетей, транспортных сетей, программных комплексов и т. д.). В частности, графовые модели являются удобными моделями схем РЭА и ЭВА на различных этапах их проектирования.

На этапе схемотехнического проектирования РЭА и ЭВА графы используются для моделирования электрических цепей. Пусть имеется некоторая электрическая цепь S , представляющая собой набор элементов $a_1, \dots, a_m$ (например, сопротивлений, конденсаторов, индуктивностей, источников ЭДС и т. д.), соединенных проводниками. Для каждого элемента a_i обозначим через I_i величину тока, проходящего через этот элемент, а через U_i - напряжение между его полюсами. Направление токов I_i и напряжений U_i для каждого элемента цепи указывается на схеме стрелками. При этом если ток течет по направлению стрелки, его величина I_i положительна. Величина напряжения U_i положительна тогда, когда падение напряжения в элементе цепи a_i направлено вдоль стрелки. Данной электрической цепи можно поставить в соответствие ориентированный мультиграф, вершины которого соответствуют узлам электрической цепи, а дуги - элементам, расположенным между соответствующими узлами. Ориентация каждой дуги определяется в соответствии с направлением тока через данный элемент цепи. Поскольку направление тока заранее предсказать не всегда возможно, введенную ориентацию дуг можно рассматривать как условную. После определения I_i , U_i знаки этих величин будут показывать истинное направление тока и напряжения через каждый элемент a_i .

Пример. Составим графовую модель электрической цепи на рис. 5.1, а.

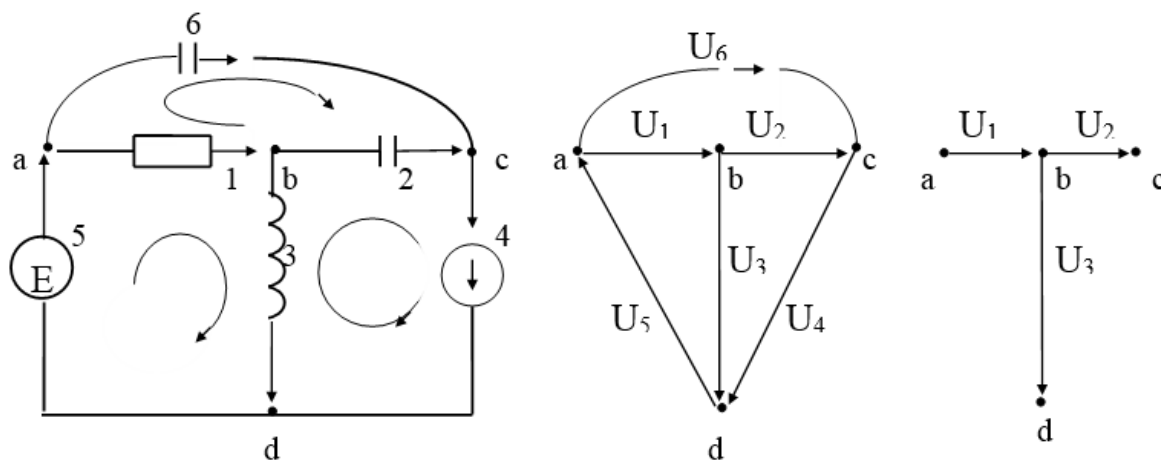


Рис. 5.1. Пример представления электрической цепи графом

Для построения графовой модели предварительно пронумеруем элементы цепи, а стрелками укажем направление токов. Графовая модель данной цепи приведена на рис. 5.1, б. При этом вершины графа a, b, c, d соответствуют узлам a, b, c, d электрической схемы, ребра $u_1 - u_6$ - элементам 1 - 6. Ориентация дуг графа определяется направлениями токов через элементы электрической цепи.

При решении задач конструкторско-топологического проектирования РЭА и ЭВА графы используются для представления коммутационных схем. Напомним, что коммутационной схемой называется набор конструктивных модулей (микросхем, БИС, схемных сочетаний микросхем, различных блоков и т. д.), заданным образом соединенных между собой цепями. Каждый элемент коммутационной схемы X_i имеет некоторое множество соединительных выводов, называемых контактами. Кроме контактов элементов в схеме имеются внешние контакты, которые осуществляют связь рассматриваемой схемы с другими схемами. Будем обозначать контакты элементов C_{ij} , а внешние контакты C_{0j} , где i -номер элемента, j -номер контакта. Два контакта считаются связанными, если они объединяются одной электрической цепью e_k . Под электрической цепью будем понимать множество контактов, принадлежащих одному эквипотенциалу.

Графовое представление коммутационных схем используется при решении основных задач конструкторско-топологического проектирования: задач покрытия, типизации, компоновки, размещения, трассировки. При этом вершины графа соответствуют элементам (модулям) схемы, а дуги (ребра) - соединяющим их цепям.

Пример. На рис. 5.2 представлен условный фрагмент коммутационной схемы и ее граф. При этом вершины графа $X_1 - X_4$ соответствуют элементам $X_1 - X_4$ коммутационной схемы. Две вершины соединяются дугой, если соответствующие элементы схемы соединены цепью.

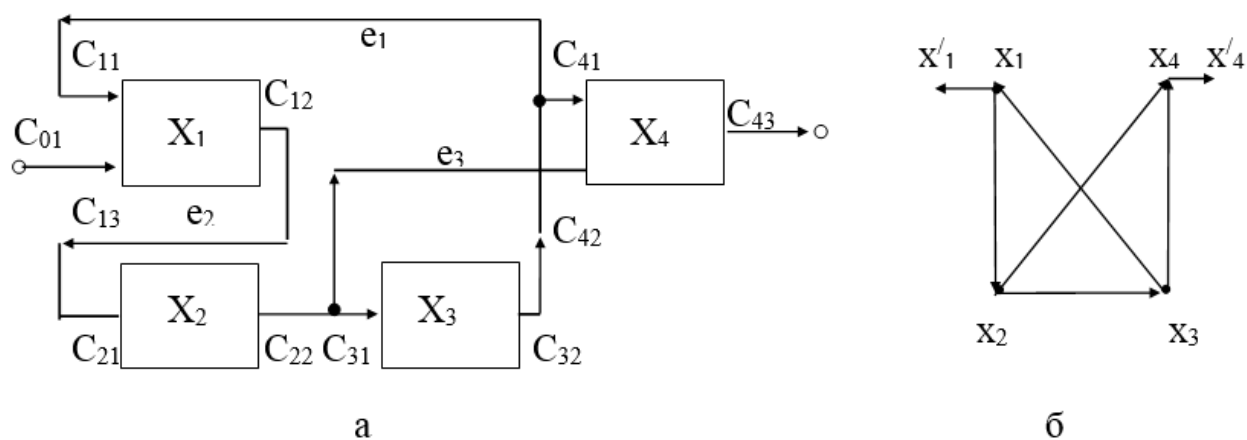


Рис. 5.2. Пример представления коммутационной схемы графом

Широкое применение элементы теории графов находят при анализе электрических цепей. Для заданной электрической цепи S одна из задач анализа заключается в определении для всех элементов напряжений и токов, которые удовлетворяют следующим законам Кирхгофа:

1. *Закон Кирхгофа для токов.* Алгебраическая сумма токов, вытекающих из узла, равна нулю.

2. *Закон Кирхгофа для напряжений.* Алгебраическая сумма напряжений в любом замкнутом контуре равна нулю.

Рассмотрим графовый подход к составлению уравнений Кирхгофа для токов и напряжений. Пусть S - электрическая цепь с m элементами. Обозначим через $I = (I_1, \dots, I_m)$, $U = (U_1, \dots, U_m)$ векторы токов и напряжений в цепи S . Представив электрическую цепь в виде ориентированного графа G , систему уравнений Кирхгофа для токов можно записать следующим образом:

$$BI = 0,$$

где B - матрица инцидентности графа G .

Система уравнений Кирхгофа для напряжений имеет вид

$$FU = 0,$$

где F - матрица фундаментальных циклов графа G . Эта система называется *базисной системой* уравнений Кирхгофа для напряжений.

Пример. Составим системы уравнений Кирхгофа для токов и напряжений в электрической цепи, представленной на рис. 5.1, а. Матрица инцидентности графа данной электрической цепи (рис. 5.1, б) запишется следующим образом:

$$B = \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 1 \\ -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & -1 & 1 & 0 \end{pmatrix} \end{matrix}$$

Следовательно, уравнения Кирхгофа для токов будут иметь вид:

$$I_1 - I_5 + I_6 = 0,$$

$$-I_1 + I_2 + I_3 = 0,$$

$$-I_2 + I_4 - I_6 = 0,$$

$$-I_3 - I_4 + I_5 = 0.$$

Остов графа рассматриваемой электрической цепи представлен на рис. 5.1, в. В данном графе можно выделить следующие три фундаментальных цик-

ла, относительно этого остова: $u_1u_3u_5$, $u_2u_4u_3$, $u_6u_2u_1$. Матрица фундаментальных циклов имеет вид:

$$F = \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

Тогда базисную систему уравнений Кирхгофа для напряжений можно записать следующим образом:

$$\begin{aligned} U_1 + U_3 + U_5 &= 0, \\ U_2 - U_3 + U_4 &= 0, \\ -U_1 - U_2 + U_6 &= 0. \end{aligned}$$

Во многих практических приложениях встречается задача построения кратчайшего остова взвешенного графа. Рассмотрим несколько примеров.

Пример. Необходимо проложить сеть проводов, связывающих n терминалов вычислительной аппаратуры, причем так, чтобы из одного терминала можно было связаться с любым другим. Если требуется, чтобы количество затраченного провода было минимально, то граф, вершины которого соответствуют терминалам, а ребра - соединяющим их проводам, должен быть деревом. Рассмотрим полный взвешенный граф K_n , вершины которого соответствуют терминалам, а вес каждого ребра равен количеству провода, необходимого для соединения i -го и j -го терминала. Тогда задача состоит в построении кратчайшего остова данного графа.

Пусть $n=5$, а расстояние между терминалами задается матрицей

$$\begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} 0 & 5 & 9 & 8 & 14 \\ 5 & 0 & 10 & 6 & 4 \\ 9 & 10 & 0 & 3 & 15 \\ 8 & 6 & 3 & 0 & 19 \\ 4 & 4 & 15 & 19 & 0 \end{pmatrix} \end{matrix}.$$

Тогда соответствующий граф представлен на рис. 5.3, а. На рис. 5.3, б приведен кратчайший остов данного графа, построенный с использованием ал-

алгоритма Краскала. Последовательность выбора ребер:
 $(x_3, x_4), (x_2, x_5), (x_1, x_2), (x_2, x_4)$.

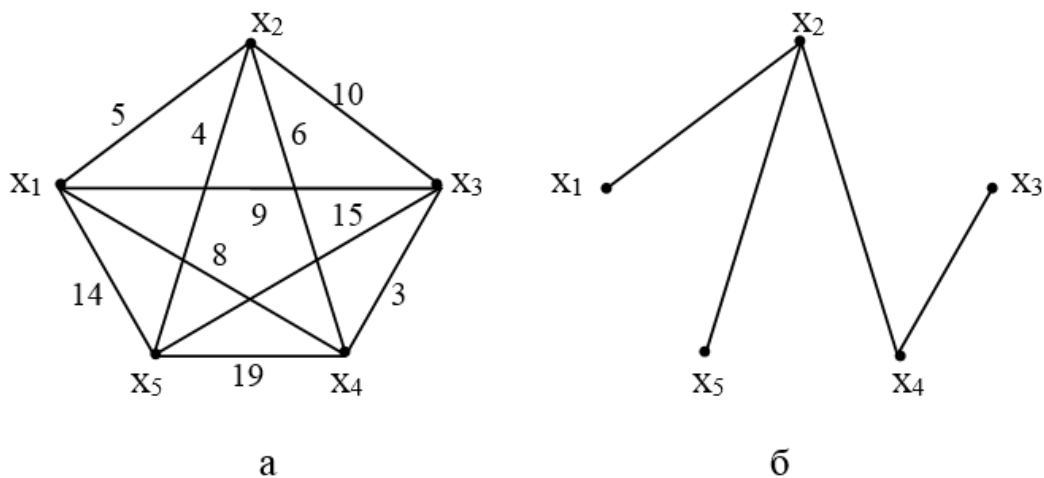


Рис. 5.3. Пример представления вычислительной сети графом и ее минимальная конфигурация

Пример. Рассмотрим задачу трассировки проводных соединений при конструкторско-топологическом проектировании РЭУ. Пусть имеется n точек плоскости, соответствующих выводам (контактам) произвольной цепи с заданными расстояниями между ними. Необходимо соединить все контакты цепью таким образом, чтобы суммарная длина соединений была минимальна. Рассмотрим полный граф K_n , вершины которого соответствуют контактам, а ребра - соединениям между парами выводов. Каждому ребру u_i приписывается вес W_i , который может представлять линейную комбинацию нескольких характеристик соединения $W_i = \mu_1 h_{i1} + \dots + \mu_k h_{ik}$, где μ_j - весовые коэффициенты, h_{ij} - некоторая j -я характеристика соединения u_i (длина, стоимость и т.д.). Тогда задача сводится к построению в данном графе кратчайшего остова. Например, необходимо соединить цепью эквипотенциальные контакты, обозначенные цифрами 1, 2, 3, 4 на рис. 5.4, а. Расстояния между ними задаются матрицей

$$\begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 9 & 2 & 10 \\ 9 & 0 & 15 & 5 \\ 2 & 15 & 0 & 7 \\ 10 & 5 & 7 & 0 \end{pmatrix} \end{matrix}$$

Граф K_4 представлен на рис. 5.4, б. Для построения кратчайшего остова этого графа используем алгоритм Прима. При этом последовательность выбора

ребер будет иметь вид: (1, 3), (3, 4), (2, 4). Кратчайший остов изображен на рис. 5.4, в. На рис 5.4, г представлены результаты трассировки.

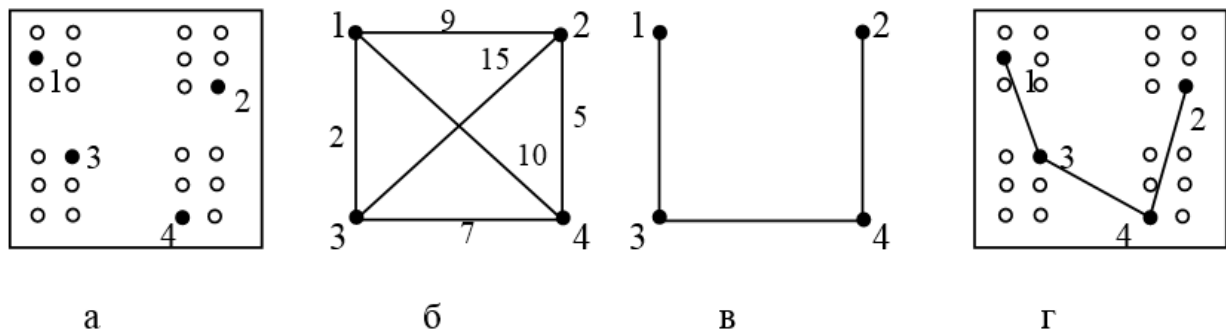


Рис. 5.4. Результаты трассировки проводных соединений

Многие практические задачи сводятся к раскраске графа. Это задачи распределения ресурсов, разработки графиков осмотра оборудования, составления расписаний и т. д. Рассмотрим одну из задач, возникающих при трассировке многослойных печатных плат.

Пример. Пусть имеется коммутационная схема, элементы которой определенным образом размещены на плате. Проведена трассировка печатных соединений, в результате которой некоторые цепи пересекаются друг с другом. Необходимо таким образом распределить цепи по параллельным слоям, чтобы в каждом из слоев они бы не пересекались, а число слоев было бы минимальным. Распределение связей (цепей) без пересечений с минимизацией числа слоев эквивалентно минимизированной раскраске графа пересечений схемы $G' = (X, U)$, где X - множество вершин графа, соответствующее множеству цепей схемы. Ребро $u \in U$ существует и соединяет две вершины $x_1, x_2 \in X$, если соответствующие цепи пересекаются. После раскраски цепи, сопоставленные одноцветным вершинам, помещаются в один слой.

На рис. 5.5, а приведен фрагмент схемы. На рис. 5.5, б показан граф пересечений цепей этого фрагмента. Осуществим раскраску вершин данного графа с использованием эвристического алгоритма. Строим список:

x_i	x_3	x_1	x_4	x_6	x_7	x_9	x_5	x_8	x_2	x_{10}
$\rho(x_i)$	5	4	4	4	4	3	3	3	2	2

Вершины x_3, x_7, x_5 окрашиваются в цвет 1 и удаляются из списка. Получаем новый список:

x_i	x_1	x_4	x_6	x_9	x_8	x_2	x_{10}
$\rho(x_i)$	4	4	4	3	3	2	2

Вершины x_1, x_4, x_9 окрашиваются в цвет 2 и удаляются из списка. Оставшиеся вершины не смежны между собой. Поэтому они окрашиваются в

цвет 3. В итоге получаем следующее распределение цепей по слоям: слой 1 - цепи 1, 4, 9; слой 2 - цепи 3, 5, 7; слой 3 - цепи 2, 6, 8, 10. Цепи трех слоев изображены на рис. 5.5, в.

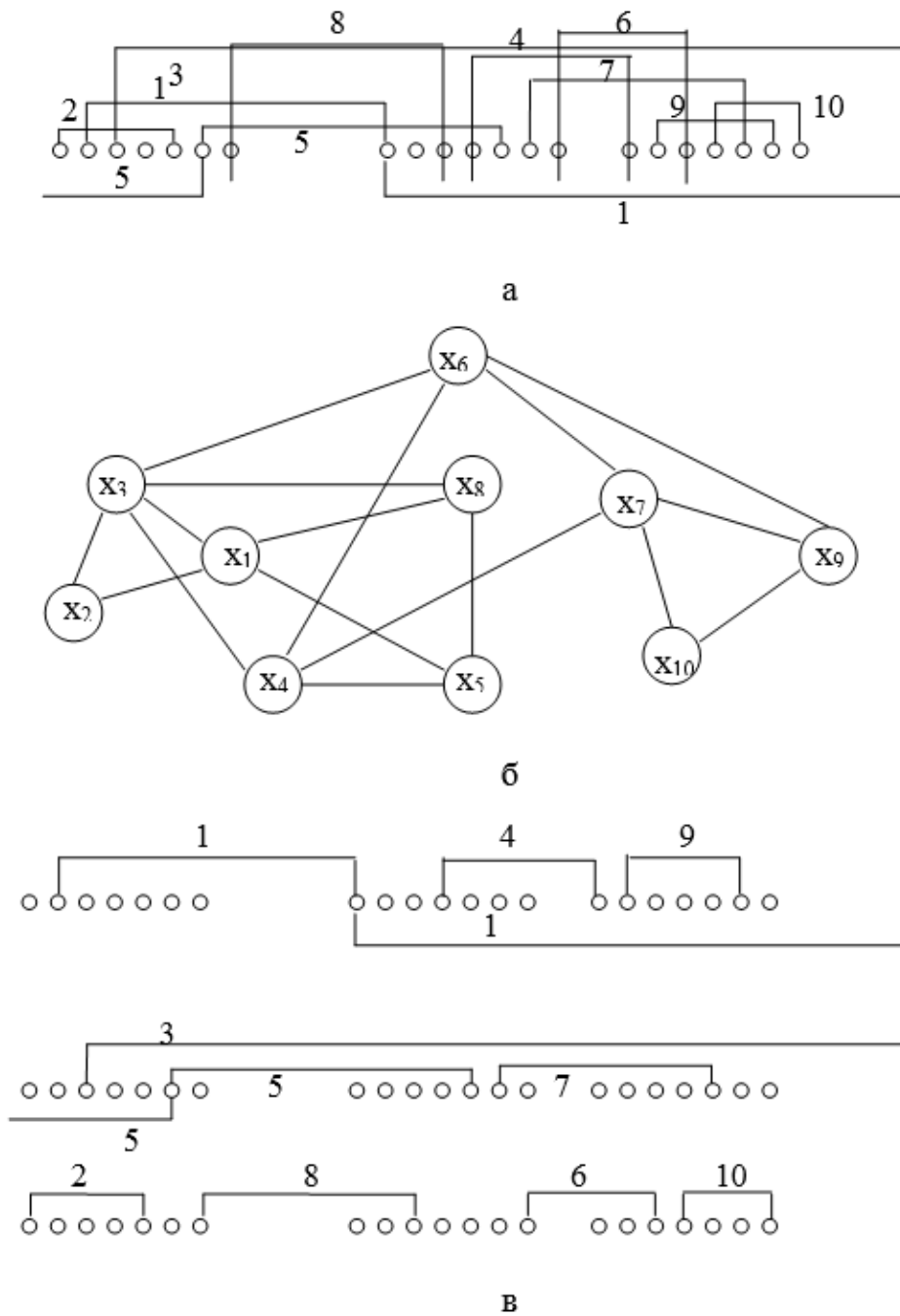


Рис. 5.5. Результаты решения задачи распределения связей без пересечений с минимизацией числа слоев на плате

В итоге получаем следующее распределение вершин по цветам:

Цвет 1: X_1 X_4 X_9

Цвет 2: X_3 X_5 X_7

Цвет 3: X_2 X_6 X_8 X_{10}

Хроматическое число графа равно 3.

Важное значение имеют также задачи построения эйлеровых и гамильтоновых циклов в графах. Такие задачи в частности возникают при проверке различных распределенных систем (вычислительных, электрических, транспортных сетей, телефонных или железнодорожных линий и т. д.).

Пример. Рассмотрим вычислительную сеть, графовая модель которой представлена на рис. 5.6.

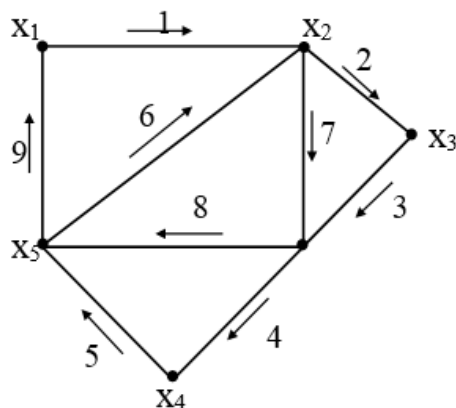


Рис. 5.6. Пример вычислительной сети, представленной графом

Необходимо определить такую последовательность проверки всех кабелей связи, чтобы каждый кабель был проверен только один раз. Эта задача сводится к построению в данном графе эйлерова цикла. Эйлеров граф, определенный алгоритмом Флери, показан на рис. 5.6 стрелками. В данном графе цикл не является единственным.

Широкое применение в автоматизированных системах находят алгоритмы поиска кратчайших путей. Алгоритмы данного класса используются при решении логистических задач и топологических задач в транспортных сетях (определение кратчайшего пути в транспортной сети), при решении задач маршрутизации в информационных сетях и т.д.

Пример. На рис. 5.7. представлена графовая модель транспортной сети. Необходимо определить кратчайший путь от вершины s до вершины t .

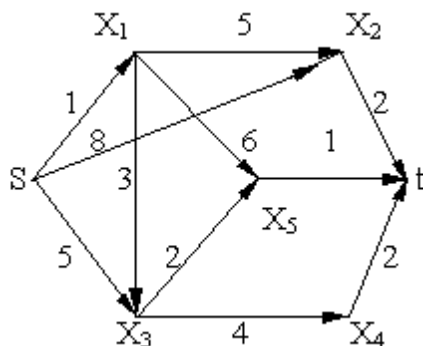


Рис. 5.7. Пример представления транспортной сети графом для определения кратчайшего пути между вершинами

Для нахождения кратчайшего пути используем алгоритм Дейкстры. Работу алгоритма можно проиллюстрировать таблицей:

Вершина	1	2	3	4	5	6
s	<u>0</u>	-	-	-	-	-
x ₁	∞	<u>1</u>	-	-	-	-
x ₂	∞	8	6	<u>6</u>	-	-
x ₃	∞	5	<u>4</u>	-	-	-
x ₄	∞	∞	∞	8	8	8
x ₅	∞	∞	7	6	<u>6</u>	-
t	∞	∞	∞	∞	8	<u>7</u>

Таким образом, на 6-й итерации найден кратчайший путь $s x_1 x_3 x_5 t$ длиной 7.

6. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

6.1. Лабораторная работа № 1.

Определение метрических характеристик графовой модели.

Цель работы: Изучение технологии определения метрических характеристик графа, приобретение практических навыков разработки программного обеспечения для определения метрических характеристик графовой модели.

Лабораторные задания.

1. По главе 3 настоящего пособия ознакомиться с основными типами графов, способами задания графов, метрическими характеристиками графа и технологией их определения.

2. Разработать программу для определения метрических характеристик графовой модели. Требования к программе:

а) граф должен быть задан **матрицей смежности** (матрица расстояний определяется программно на основе матрицы смежности);

б) программа должна определять и выводить следующие характеристики графа:

- матрицу расстояний;
- степени вершин;
- радиус графа;
- диаметр графа;
- центр графа (если центров несколько, выводятся все центры);
- медиану графа (если медиан несколько, выводятся все медианы).

3. Выбрать граф в соответствии с вариантом и решить задачу определения метрических характеристик графа с использованием разработанной программы.

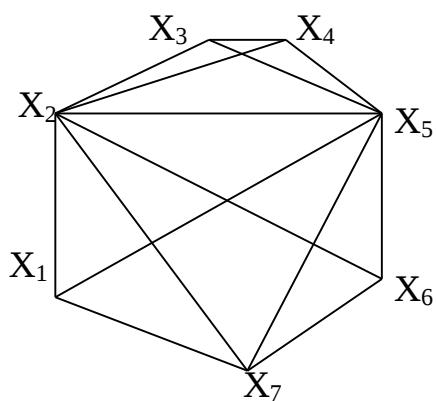
4. Решить выбранную задачу вручную, сравнить полученные результаты.

5. Рассмотреть прикладные аспекты задачи определения метрических характеристик графа (примеры практических задач данного класса в автоматизированных системах).

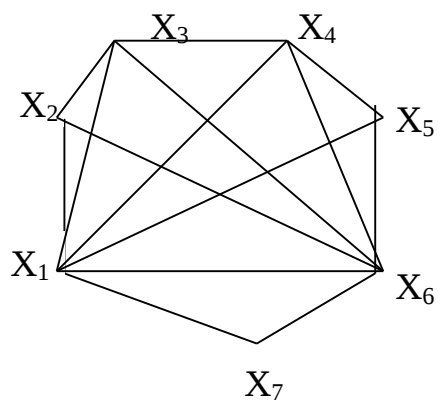
Отчёт должен содержать:

1. Название лабораторной работы и ее цель.
2. Постановку задачи и описание технологии её решения.
3. Описание и блок-схему реализованного алгоритма.
4. Листинг разработанной программы.
5. Контрольный пример работы программы: исходные данные (граф в соответствии с вариантом) и полученные результаты.
6. Решение задачи вручную.
7. Выводы (результаты лабораторной работы и прикладные аспекты рассмотренных алгоритмов).

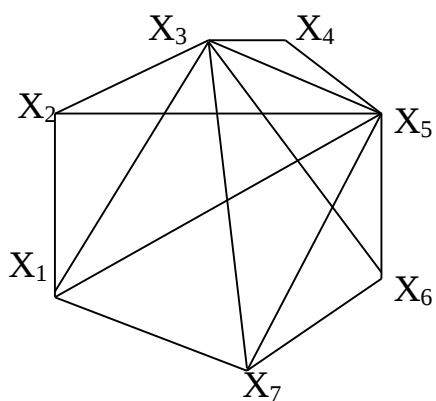
Варианты заданий



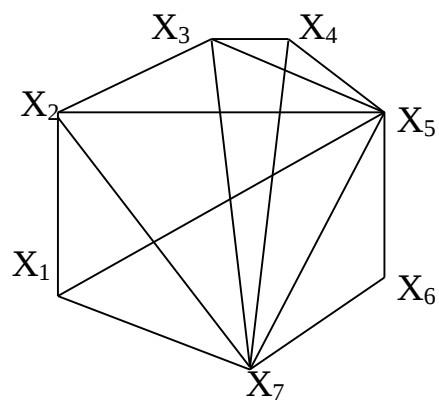
Вариант 1



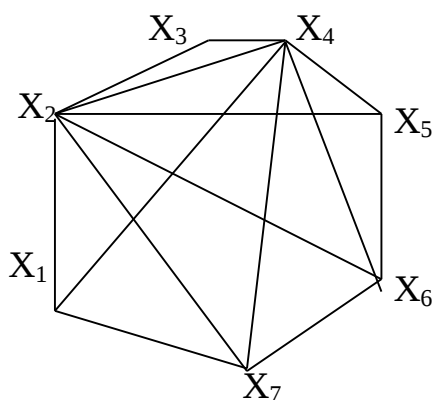
Вариант 2



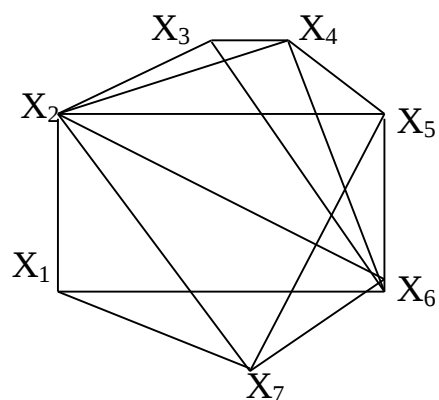
Вариант 3



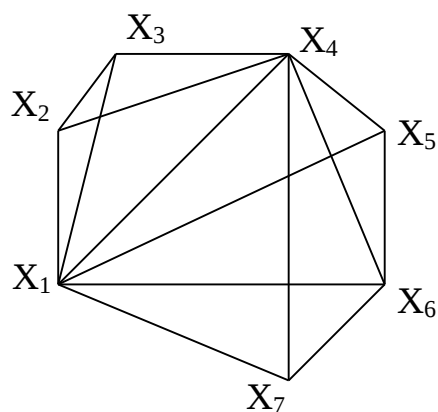
Вариант 4



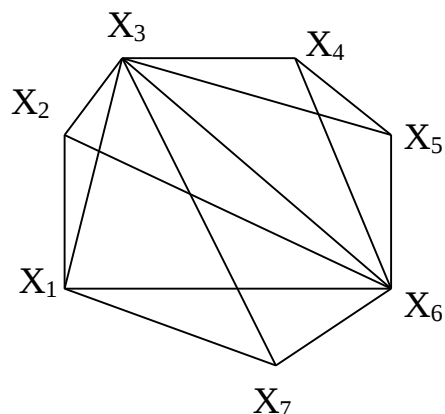
Вариант 5



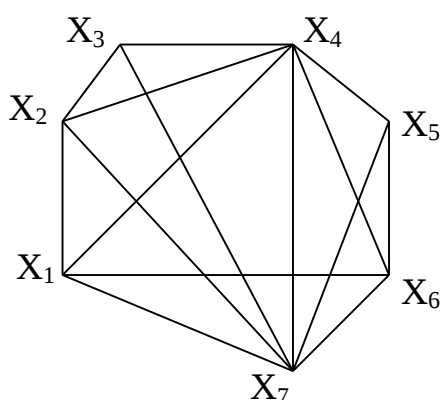
Вариант 6



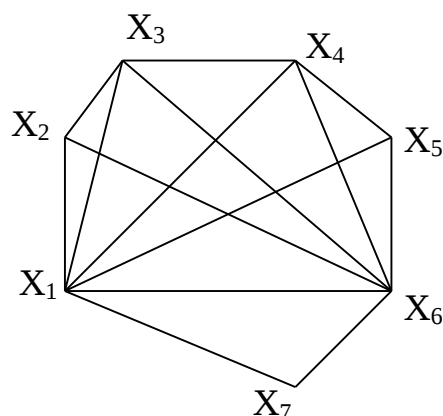
Вариант 7



Вариант 8



Вариант 9



Вариант 10

Контрольные вопросы

1. Каким образом осуществляется построение матрицы смежности неориентированного и ориентированного графа?
2. Дайте определение маршрута.
3. Как определяется расстояние между вершинами графа?
4. Чему равны элементы матрицы расстояний графа?
5. Каким образом можно построить матрицу расстояний по матрице смежности графа?
6. Дайте определение эксцентриситета вершины? Каким образом определяется эксцентриситет?
7. Дайте определение передаточного числа вершины? Каким образом определяется передаточное число?
8. Как определяются метрические характеристики графа - радиус, диаметр, центр, медиана?
9. Какие классы практических задач сводятся к определению метрических характеристик графа?

6.2. Лабораторная работа № 2.

Анализ связности графовой модели.

Определение сильных компонент графа.

Цель работы: Изучение технологии анализа связности графовой модели, приобретение практических навыков разработки программного обеспечения для определения сильных компонент графа.

Лабораторные задания.

1. По п. 4.1 настоящего пособия ознакомиться с понятиями достижимости вершин, связности и сильной связности графа. Рассмотреть основные этапы определения сильных компонент графа.

2. Разработать программу для определения сильных компонент графа. Требования к программе:

а) граф должен быть задан **матрицей смежности** (матрицы достижимости, контрдостижимости и сильной связности определяются программно на основе матрицы смежности);

б) программа должна определять и выводить:

- матрицы достижимости, контрдостижимости и сильной связности графа;

- сильные компоненты графа.

3. Выбрать граф в соответствии с вариантом и решить задачу определения сильных компонент графа с использованием разработанного программного обеспечения.

4. Решить выбранную задачу вручную, сравнить полученные результаты с результатами работы программы.

5. Рассмотреть прикладные аспекты задачи определения сильных компонент графа (примеры практических задач данного класса в автоматизированных системах).

Отчёт должен содержать:

1. Название лабораторной работы и ее цель.

2. Постановку задачи, основные понятия и определения, описание технологии анализа связности графа и поиска сильных компонент.

3. Описание и блок-схему реализованного алгоритма определения сильных компонент графа.

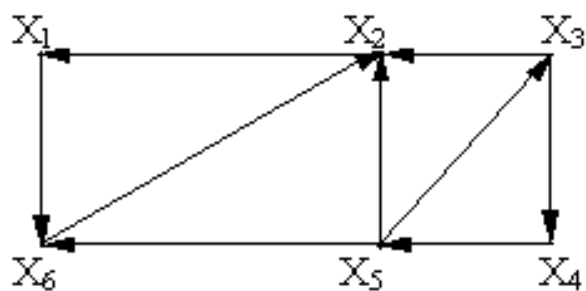
4. Листинг разработанной программы.

5. Контрольный пример работы программы: исходные данные (граф в соответствии с вариантом) и полученные результаты.

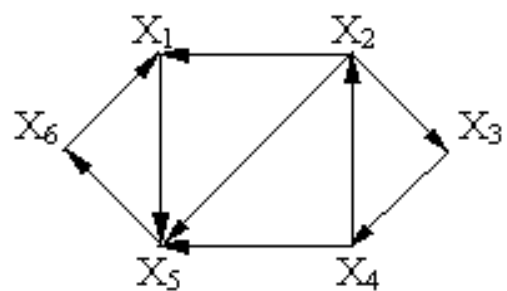
6. Решение задачи вручную.

7. Выводы (результаты лабораторной работы и прикладные аспекты рассмотренных алгоритмов).

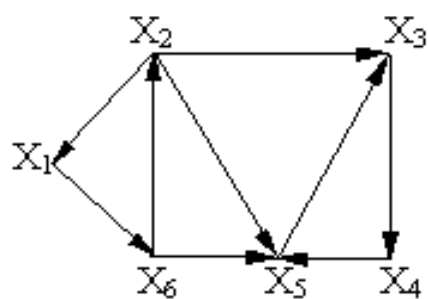
Варианты заданий



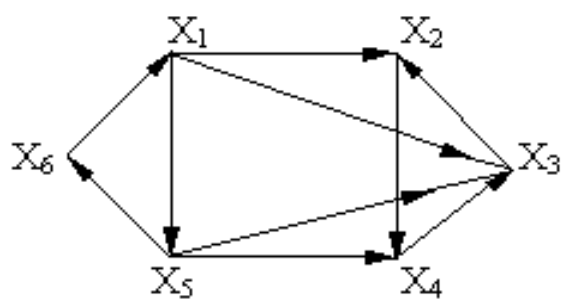
Вариант 1



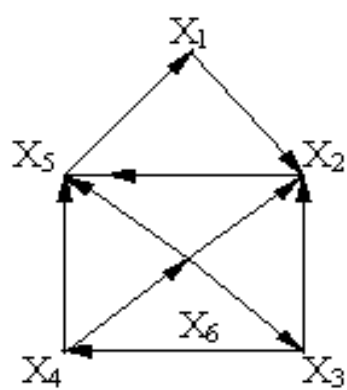
Вариант 2



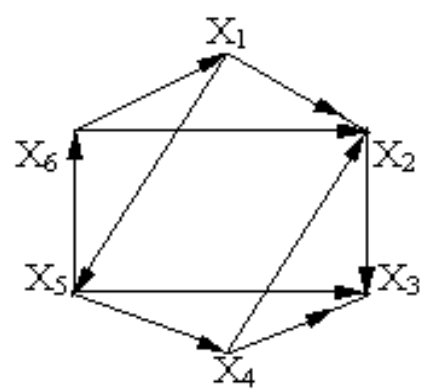
Вариант 3



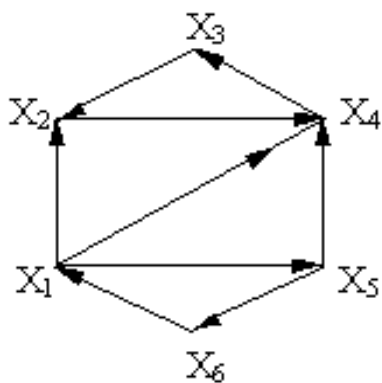
Вариант 4



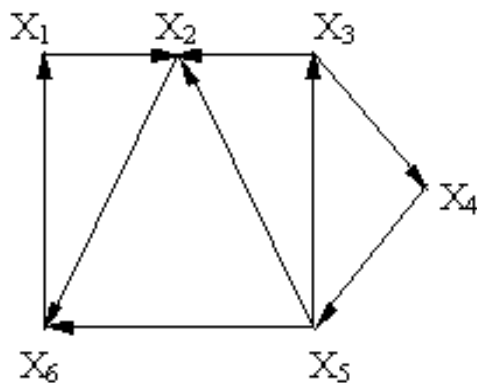
Вариант 5



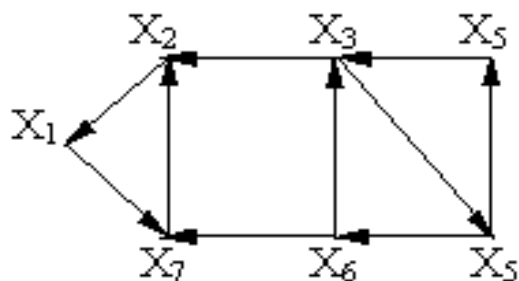
Вариант 6



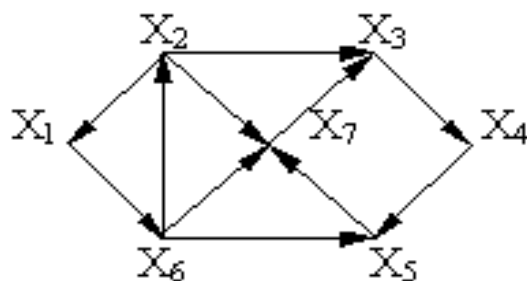
Вариант 7



Вариант 8



Вариант 9



Вариант 10

Контрольные вопросы:

1. Дайте определения достижимости и взаимной достижимости вершин в графе.
2. Какие типы графов называют связным, сильно связным, односторонне связным?
3. Дайте определение сильной компоненты ориентированного графа.
4. Как строится матрица достижимости графа? Чему равны её диагональные элементы?
5. Каким образом строится матрица контрдостижимости графа? Как она связана с матрицей достижимости?
6. Каким образом строится матрица сильной связности? Как её можно построить, зная матрицы достижимости и контрдостижимости?
7. Каким образом можно построить матрицу достижимости по матрице смежности графа?
8. Перечислите основные шаги алгоритма определения компонент связности неориентированного графа и сильных компонент ориентированного графа.
9. Какие классы практических задач сводятся к определению компонент связности графа?

6.3. Лабораторная работа № 3.

Построение остовных деревьев графа

в задачах топологического проектирования вычислительных сетей

Цель работы: Изучение алгоритмов построения остовных деревьев графа и приобретение практических навыков их программной реализации.

Лабораторные задания.

1. По п. 4.2 настоящего пособия ознакомиться с понятием остовного дерева графа. Изучить алгоритмы построения остовных деревьев графа.

2. Программно реализовать алгоритмы Краскала и Прима построения остовных деревьев графа. Программа должны выводить следующие результаты:

- последовательность включения рёбер в остовное дерево по алгоритму Краскала;
- последовательность включения рёбер в остовное дерево по алгоритму Прима;
- суммарный вес рёбер построенного остова.

3. Выбрать граф в соответствии с вариантом и решить задачу построения остовного дерева с использованием разработанного программного обеспечения. Граф, представленный в задании, представляет собой возможные варианты соединений компьютеров при проектировании вычислительной сети. Определение оптимальной топологии сети сводится к построению кратчайшего остовного дерева данного графа.

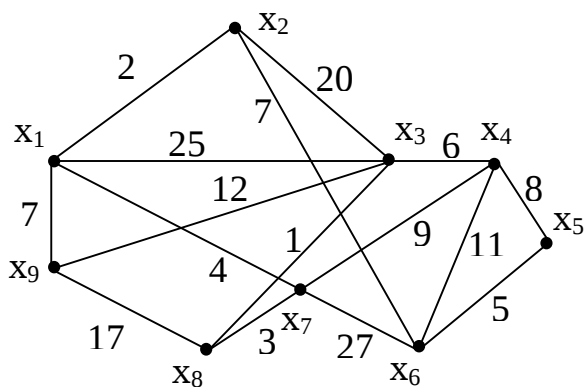
4. Решить выбранную задачу вручную, сравнить полученные результаты с результатами работы программы.

5. Рассмотреть прикладные аспекты задачи построения остовных деревьев графа (примеры практических задач данного класса в автоматизированных системах).

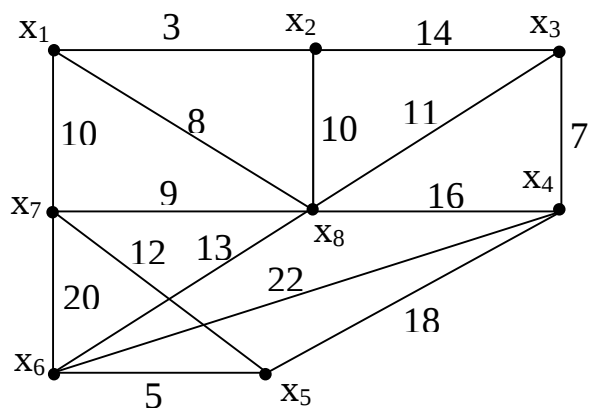
Отчёт должен содержать:

1. Название лабораторной работы и ее цель.
2. Постановку задачи, основные понятия и определения.
3. Описание и блок-схемы алгоритмов Краскала и Прима построения кратчайшего остовного дерева графа.
4. Листинг разработанной программы.
5. Контрольный пример работы программы: исходные данные (граф в соответствии с вариантом) и полученные результаты.
6. Решение задачи вручную.
7. Выводы (результаты лабораторной работы и прикладные аспекты рассмотренных алгоритмов).

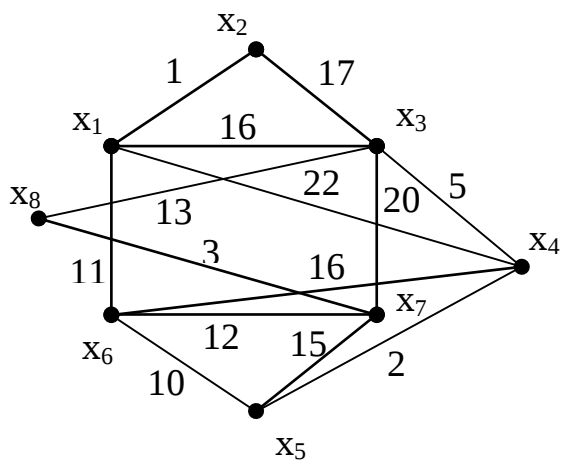
Варианты заданий



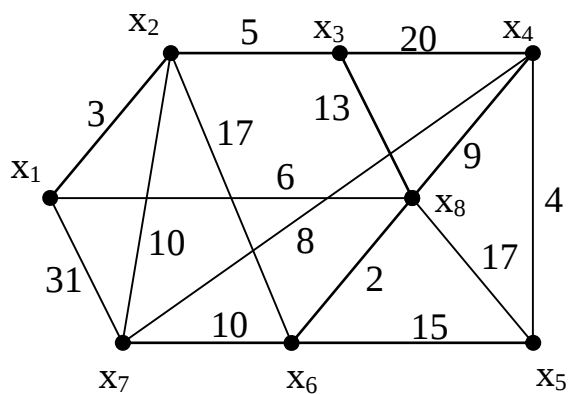
Вариант 1



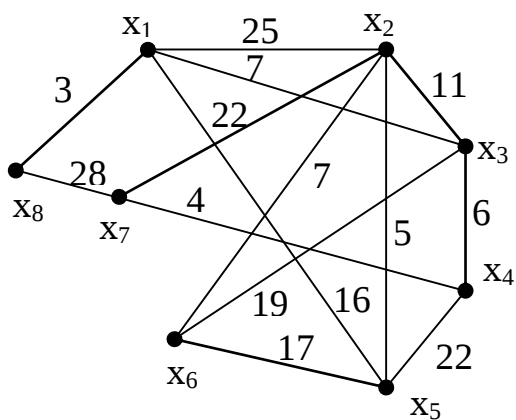
Вариант 2



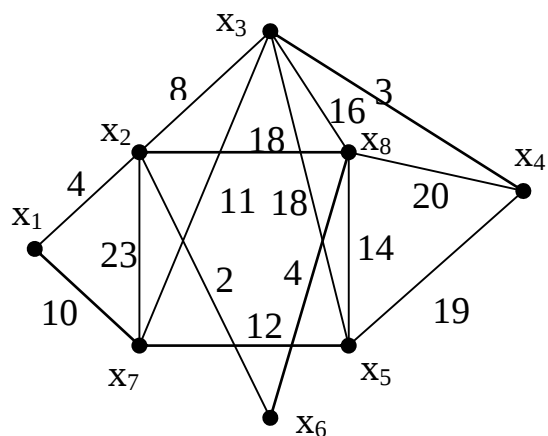
Вариант 3



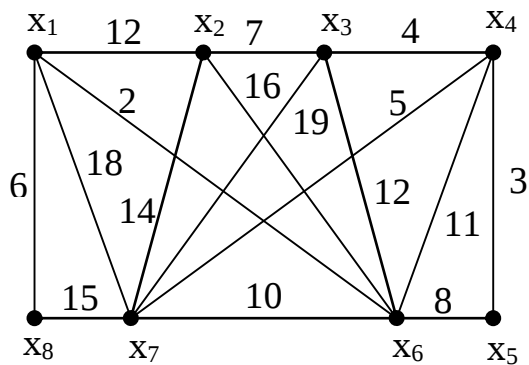
Вариант 4



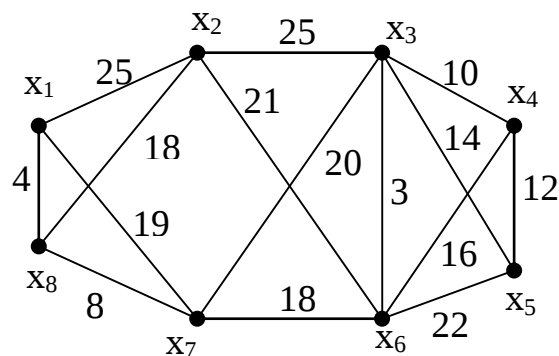
Вариант 5



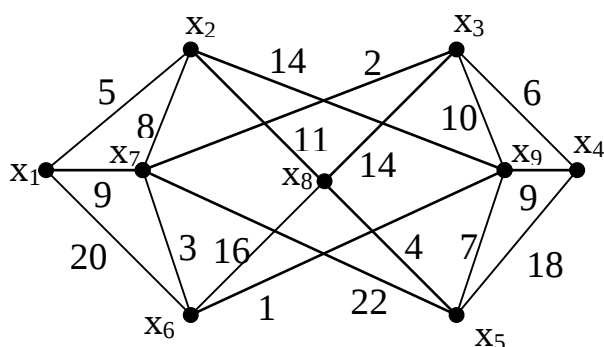
Вариант 6



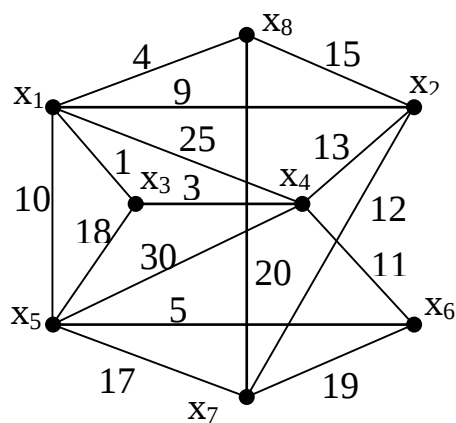
Вариант 7



Вариант 8



Вариант 9



Вариант 10

Контрольные вопросы:

1. Какой граф называется деревом?
2. Дайте определение остовного дерева графа.
3. В графе n вершин. Сколько рёбер содержит остовное дерево?
4. Как определить цикломатическое число графа? Что оно характеризует?
5. Каким образом осуществляется построение остовного дерева с использованием стратегии поиска в глубину?
6. Каким образом осуществляется построение остовного дерева с использованием стратегии поиска в ширину?
6. Какие основные шаги включает алгоритм Краскала?
7. Какие основные шаги включает алгоритм Прима? В чём его отличие от алгоритма Краскала?
8. Какие классы практических задач сводятся к построению остовных деревьев графа?

6.4. Лабораторная работа № 4.

Алгоритмы обхода графа

Цель работы: Изучение алгоритмов построения эйлеровых и гамильтоновых циклов в графе и приобретение практических навыков их программной реализации.

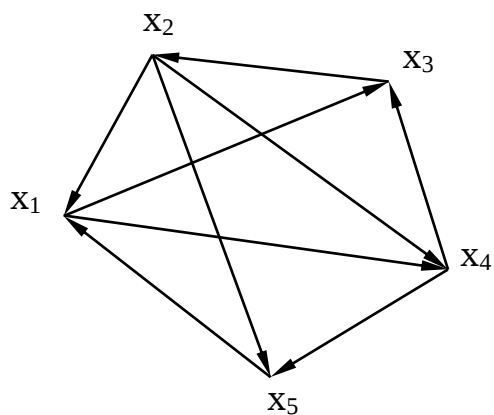
Лабораторные задания.

1. По п.4.3 настоящего пособия ознакомиться с алгоритмами построения эйлеровых и гамильтоновых циклов в графе.
2. Программно реализовать алгоритм Робертса и Флореса построения гамильтоновых циклов и цепей в графе. Требования к программе:
 - граф задаётся матрицей смежности;
 - программа должна выводить **все** гамильтоновы цепи и циклы графа (последовательность обхода вершин для каждого цикла и цепи).
3. Выбрать граф в соответствии с вариантом и решить задачу построения гамильтоновых циклов и цепей с использованием разработанного программного обеспечения.
4. Решить выбранную задачу вручную, сравнить полученные результаты.
5. Рассмотреть области применения задачи построения гамильтоновых и эйлеровых циклов в автоматизированных системах.

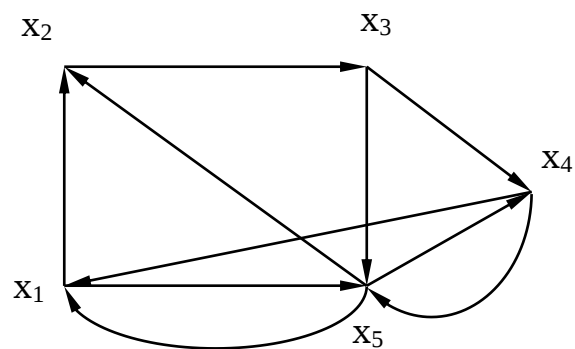
Отчёт должен содержать:

1. Название лабораторной работы и ее цель.
2. Определения эйлерова и гамильтонова цикла в графе, особенности этих циклов.
3. Теоретические сведения по алгоритмам поиска эйлеровых и гамильтоновых циклов в графе (краткий обзор алгоритмов).
4. Описание и блок-схему алгоритма Робертса и Флореса построения гамильтоновых циклов и цепей графа.
5. Листинг разработанной программы.
6. Контрольный пример работы программы: исходные данные (граф в соответствии с вариантом) и полученные результаты.
7. Решение задачи вручную. Решение оформляется в соответствии с примером, представленным в п. 4.3 пособия. При этом выводятся все промежуточные результаты.
8. Выводы (результаты лабораторной работы и прикладные аспекты рассмотренных алгоритмов).

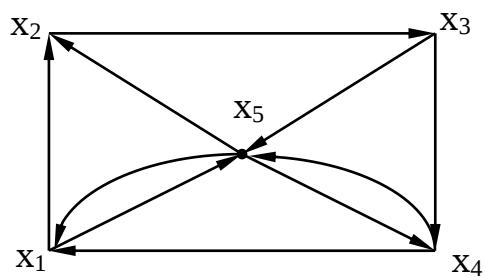
Варианты заданий:



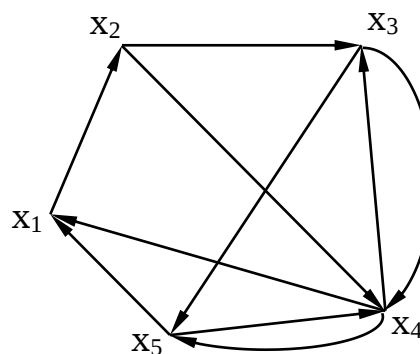
Вариант 1



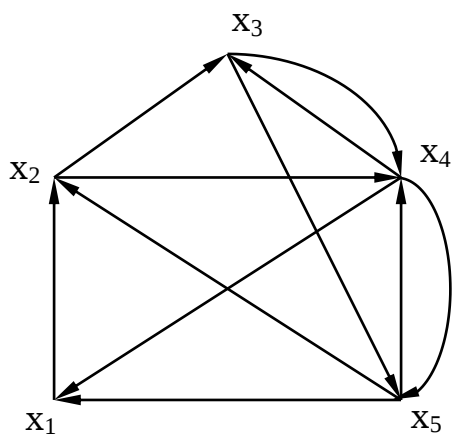
Вариант 2



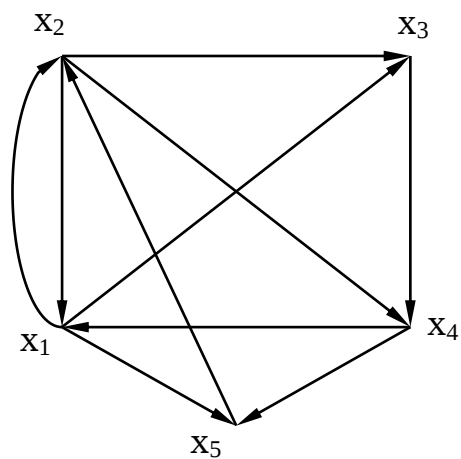
Вариант 3



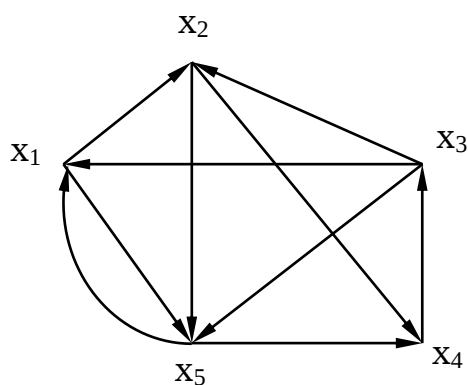
Вариант 4



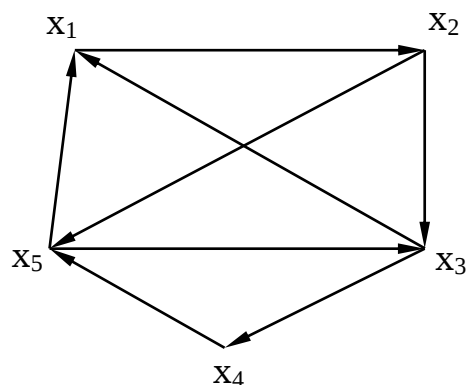
Вариант 5



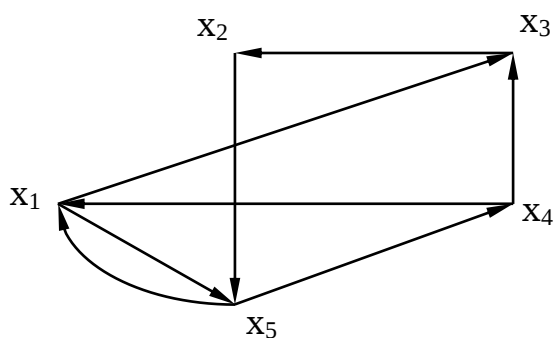
Вариант 6



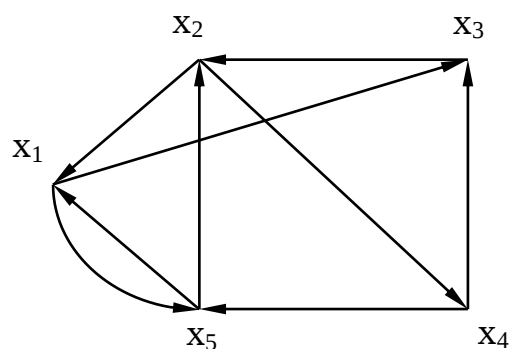
Вариант 7



Вариант 8



Вариант 9



Вариант 10

Контрольные вопросы:

1. Дайте определение эйлера цикла графа.
2. Сформулируйте теорему Эйлера. Каким условиям должен удовлетворять граф для существования эйлера цикла?
3. Как определяется степень вершины графа?
4. Чем цикл в графе отличается от цепи?
5. Приведите пример построения эйлера цикла в графе с использованием алгоритма Флери.
6. Какое ребро в графе называется мостом?
7. Дайте определение гамильтонова цикла графа.
8. Какие основные шаги включает алгоритм Робертса и Флореса для построения гамильтоновых циклов и цепей графа?
9. Какие классы практических задач сводятся к построению эйлеровых и гамильтоновых циклов графа?

6.5. Лабораторная работа № 5.

Решение задач маршрутизации в графах

Цель работы: Изучение алгоритмов поиска кратчайших путей в графах и приобретение практических навыков их программной реализации

Лабораторные задания.

1. По п.4.5 пособия ознакомиться с алгоритмами поиска кратчайшего пути в графе.

2. Разработать программу, реализующую алгоритм Дейкстры поиска кратчайшего пути в графе.

Требования к программе:

- граф задаётся матрицей весов W . Элементы матрицы W определяются следующим образом:

$$w_{ij} = \begin{cases} \text{вес } w(x_i, x_j), & \text{если в графе существует дуга } \langle x_i, x_j \rangle \\ 0, & \text{если } i = j, \\ \infty, & \text{если в графе не существует дуги } \langle x_i, x_j \rangle. \end{cases}$$

- программа должна выводить последовательность вершин, входящих в кратчайший путь, а также длину кратчайшего пути.

3. Выбрать граф в соответствии с вариантом и решить задачу поиска кратчайших путей с использованием разработанного программного обеспечения. При этом s – начальная вершина пути, t – конечная вершина пути.

4. Решить выбранную задачу вручную, сравнить полученные результаты.

5. Рассмотреть области применения задачи о кратчайшем пути в автоматизированных системах.

Отчёт должен содержать:

1. Название лабораторной работы и ее цель.

2. Постановку задачи о кратчайшем пути и краткий обзор методов её решения.

3. Описание и блок-схему реализованного алгоритма поиска.

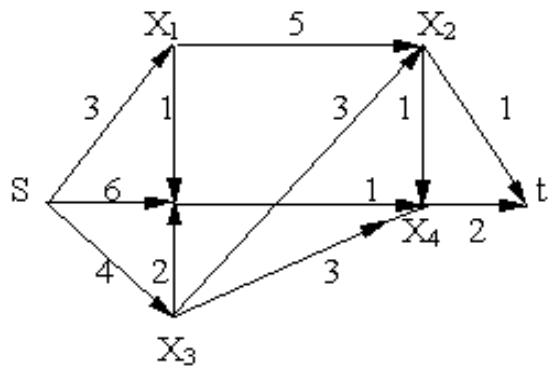
4. Листинг разработанной программы.

5. Контрольный пример работы программы: исходные данные (граф в соответствии с вариантом) и полученные результаты.

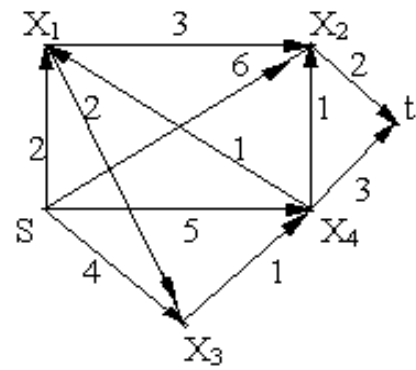
6. Решение задачи вручную. Решение оформляется в соответствии с примером, представленным в п. 4.5. пособия. Выводятся все промежуточные результаты.

7. Выводы (результаты лабораторной работы и прикладные аспекты рассмотренных алгоритмов).

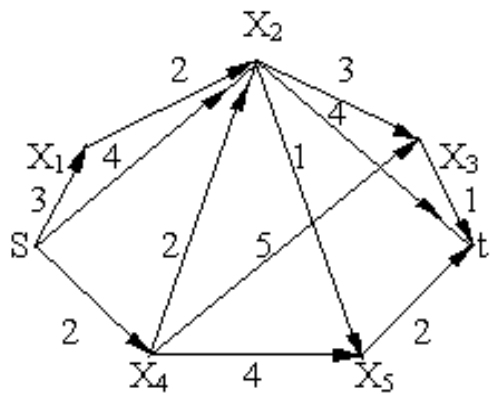
Варианты заданий



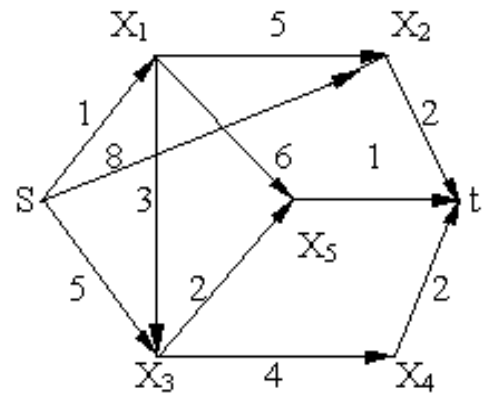
Вариант 1



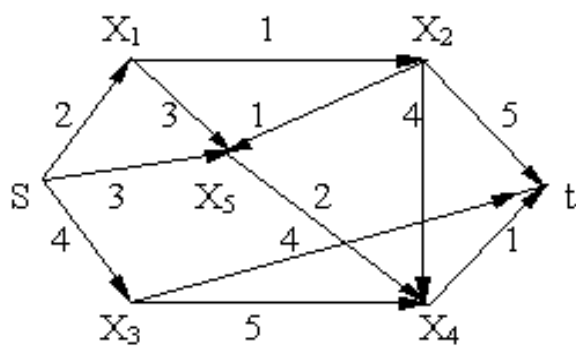
Вариант 2



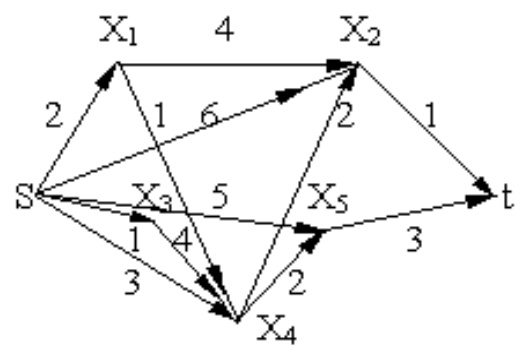
Вариант 3



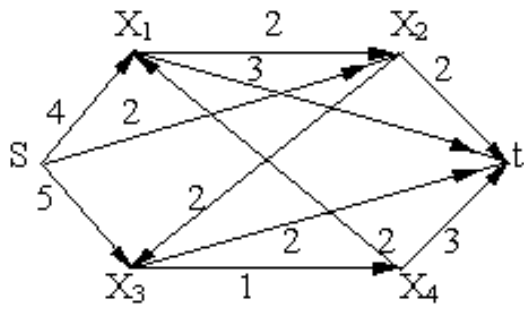
Вариант 4



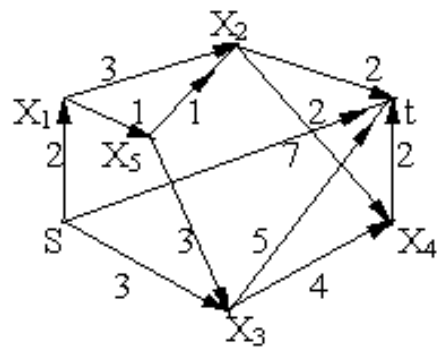
Вариант 5



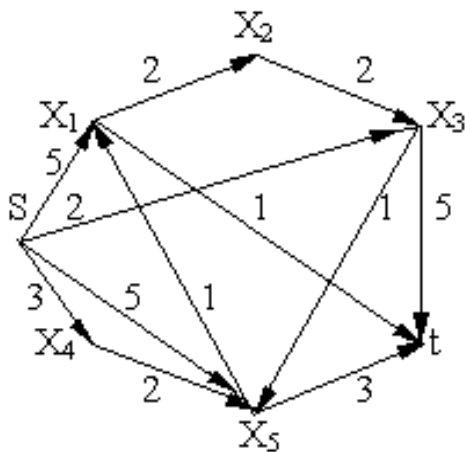
Вариант 6



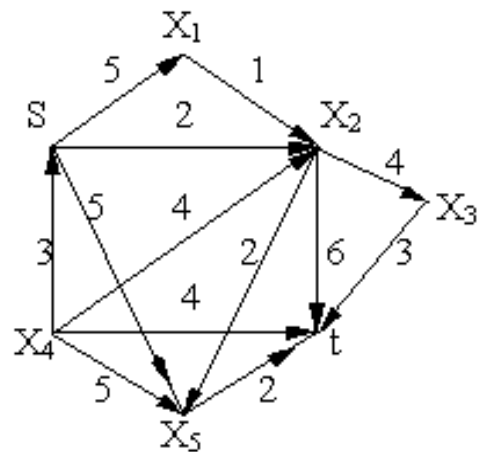
Вариант 7



Вариант 8



Вариант 9



Вариант 10

Контрольные вопросы:

1. Дайте определение кратчайшего пути в графе.
2. Какие основные шаги включает алгоритм Дейкстры? Каким образом вершинам присваиваются метки, как они пересчитываются? Приведите пример.
3. Какие ограничения в использовании имеет алгоритм Дейкстры? В каких случаях может быть получен неверный результат?
4. Чем алгоритм Форда отличается от алгоритма Дейкстры? Приведите пример.
5. Какие основные шаги включает алгоритм Флойда? Приведите пример решения задачи о кратчайших путях с использованием алгоритма Флойда.
6. Какие классы практических задач сводятся к определению кратчайших путей и маршрутов в графе?

6.6. Лабораторная работа № 6

Определение максимального потока в сети

Цель работы: Изучение и программная реализация алгоритма Форда-Фалкерсона определения максимального потока в сети

Лабораторные задания.

1. По п.4.6 настоящего пособия ознакомиться постановкой задачи о максимальном потоке и методами её решения.

2. Разработать программу для определения максимального потока в сети с использованием алгоритма Форда-Фалкерсона.

Программа в качестве результатов должна выводить величину максимального потока, а также распределение потоков по дугам сети.

3. Выбрать граф в соответствии с вариантом и решить задачу определения максимального потока с использованием разработанного программного обеспечения. Числовые характеристики на дугах графа соответствуют пропускным способностям дуг.

4. Решить выбранную задачу вручную, сравнить полученные результаты.

5. Рассмотреть области применения задачи о максимальном потоке в автоматизированных системах.

Отчёт должен содержать:

1. Название лабораторной работы и ее цель.

2. Постановку задачи о максимальном потоке.

3. Описание алгоритма Форда-Фалкерсона (суть алгоритма и основные шаги).

4. Блок-схему алгоритма Форда-Фалкерсона.

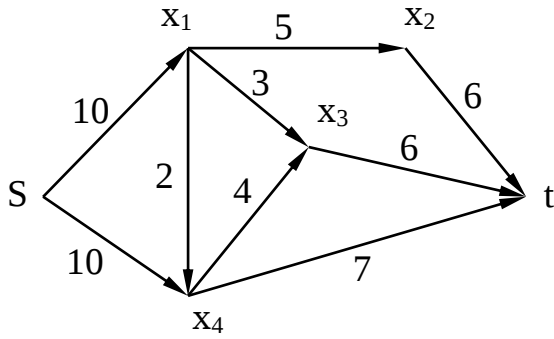
5. Листинг разработанной программы.

6. Контрольный пример работы программы: исходные данные (граф в соответствии с вариантом) и полученные результаты.

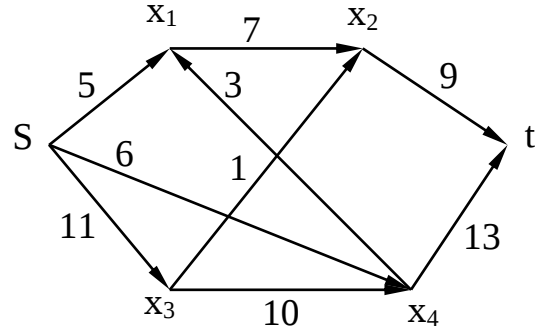
7. Решение задачи вручную. Решение оформляется в соответствии с примером, представленным в п. 4.6. пособия. Выводятся все промежуточные результаты.

7. Выводы (результаты лабораторной работы и прикладные аспекты рассмотренного алгоритма).

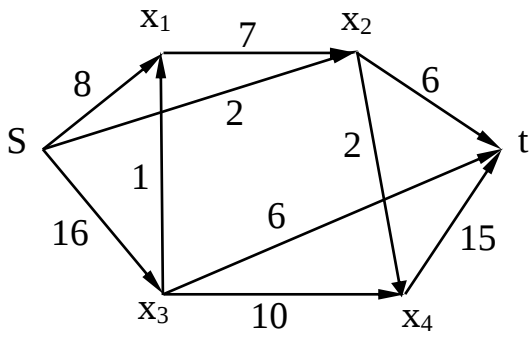
Варианты заданий



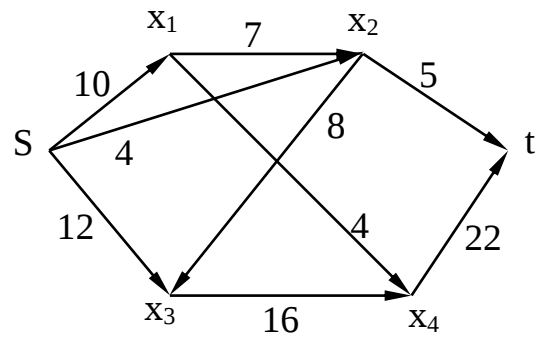
Вариант 1



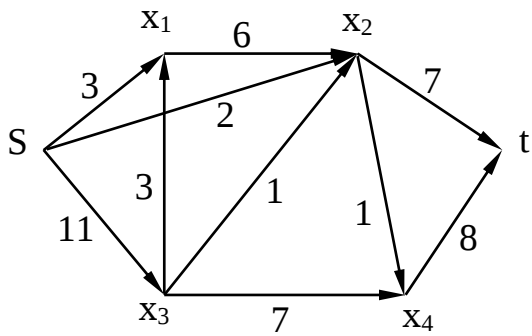
Вариант 2



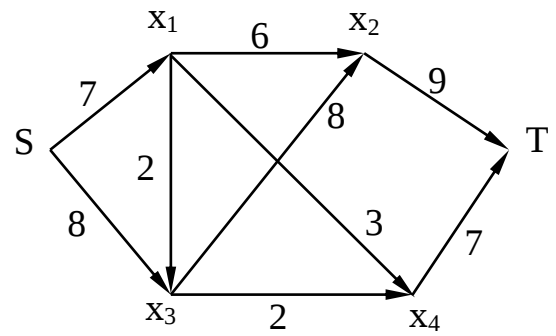
Вариант 3



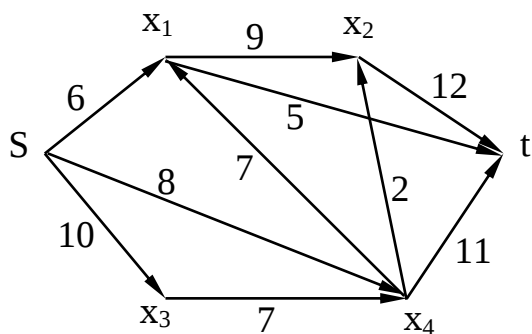
Вариант 4



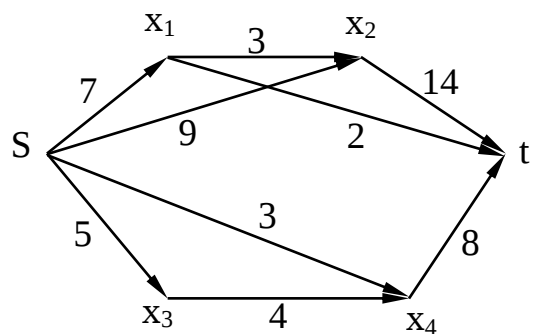
Вариант 5



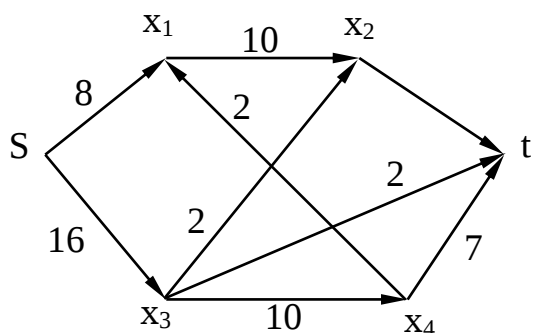
Вариант 6



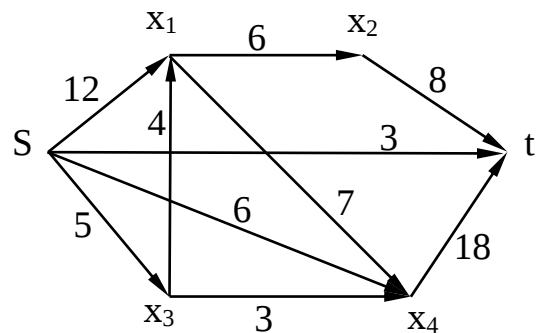
Вариант 7



Вариант 8



Вариант 9



Вариант 10

Контрольные вопросы:

1. Граф какого типа называется транспортной сетью?
2. В чём заключается задача о максимальном потоке?
3. Каким условиям должны удовлетворять потоки в дугах транспортной сети?
4. Какие вершины транспортной сети называются источником и стоком?
5. В чём суть алгоритма Форда-Фалкерсона?
6. Каким образом пересчитываются метки вершин в алгоритме Форда-Фалкерсона?
7. Какие основные шаги включает алгоритм Форда-Фалкерсона? Приведите пример.
8. Какие классы практических задач сводятся к определению максимального потока в сети?

ЗАКЛЮЧЕНИЕ

Учебное пособие ориентировано на изучение студентами базовых разделов дискретной математики и формирование у них практических навыков формализации и алгоритмизации задач, связанных с дискретными объектами. В пособии рассматриваются модели и алгоритмы дискретной математики, используемые в автоматизированных системах. Особое внимание уделяется теории графов, имеющей важнейшее значение при моделировании, анализе и оптимизации распределённых систем.

В пособии представлены различные алгоритмические схемы, которые могут быть доведены до программной реализации. Дается изложение наиболее распространенных в инженерной практике графовых алгоритмов, связанных с определением сильных компонент и метрических характеристик графа, построением остовных деревьев и гамильтоновых циклов, поиском кратчайших путей и маршрутов в графах, определением максимального потока в сетях, проверкой планарности и раскраской графов. Рассматриваемые алгоритмы могут быть программно реализованы. Важное место в учебном пособии отводится рассмотрению практических задач, сводящихся к использованию графовых моделей и алгоритмов. Кроме теории графов, в пособии излагаются основы теории множеств и комбинаторики.

Лабораторный практикум, представленный в 6 главе пособия, ориентирован на приобретение студентами практических навыков программной реализации графовых моделей и алгоритмов. В этой связи материал учебного пособия может быть полезен как с теоретической, так и с практической точки зрения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кузнецов О.П. Дискретная математика для инженера: Учебник.– СПб: Лань, 2009. – 400 с.
2. Судоплатов С.В. Дискретная математика: Учебник. – М: ИНФРАМ, 2005. – 256 с.
3. Белецкая С.Ю. Комбинаторика. Графы. Алгоритмы: Учеб. пособие. – Воронеж: ВГТУ, 2003. – 102 с.
4. Новиков Ф.А. Дискретная математика для программистов Учебник. – СПб: Питера, 2005. – 364 с.
5. Шевелёв Ю.П. Дискретная математика: Учебник.– СПб: Лань, 2009. – 380 с.
6. Белоусов А.И. Дискретная математика: учеб. Пособие. – М.: МГТУ им. Баумана, 2002. – 743 с.
7. Иванов Б.Н. Дискретная математика: алгоритмы и программы: Учеб. пособие: М: Лаборатория базовых знаний, 2002. – 288 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ И ОТНОШЕНИЙ.....	5
1.1. Основные понятия теории множеств.....	5
1.2. Нечеткие множества.....	8
1.3. Основы теории отношений.....	11
1.4. Нечеткие отношения.....	16
2. КОМБИНАТОРИКА.....	22
2.1. Основные правила комбинаторики.....	22
2.2. Комбинаторные конфигурации.....	22
2.3. Урновые схемы.....	25
2.4. Бином Ньютона.....	26
2.5. Разбиения.....	28
2.6. Формула включений и исключений.....	31
3. ОСНОВЫ ТЕОРИИ ГРАФОВ.....	35
3.1. Ориентированные и неориентированные графы.....	35
3.2. Матричный способ задания графов.....	37
3.3. Основные операции над графами.....	41
3.4. Изоморфизм и гомеоморфизм графов.....	43
3.5. Маршруты, цепи, циклы в графах.....	44
3.6. Основные типы графов.....	45
3.7. Метрические характеристики графов.....	47
4. АЛГОРИТМЫ ТЕОРИИ ГРАФОВ	51
4.1. Достижимость и связность.....	51
4.2. Деревья.....	59
4.3. Эйлеровы и гамильтоновы графы.....	65
4.4. Раскраски графа.....	69
4.5. Определение кратчайших путей и маршрутов в графе.....	71
4.6. Определение максимального потока в сети.....	79
5. НЕКОТОРЫЕ ПРИЛОЖЕНИЯ ТЕОРИИ ГРАФОВ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ.....	93
6. ЛАБОРАТОРНЫЙ ПРАКТИКУМ	102
6.1. Лабораторная работа №1. Определение метрических характеристик графовой модели.....	102
6.2. Лабораторная работа №2. Проверка связности графовой модели. Определение сильных компонент графа.....	105
6.3. Лабораторная работа №3. Построение остовных деревьев графа в задачах топологического проектирования вычислительных сетей.....	108
6.4. Лабораторная работа №4. Алгоритмы обхода графа.....	111
6.5. Лабораторная работа №5. Решение задач маршрутизации в графах.....	114

6.6. Лабораторная работа №6. Определение максимального потока в сети.....	117
ЗАКЛЮЧЕНИЕ.....	120
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	121