МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра «КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ»

Горюнов Ю.Ю., Леонова Т.Ю.

ДИСКРЕТНАЯ МАТЕМАТИКА

Конспект лекций

Пенза, 2017

ВВЕДЕНИЕ

Дискретная математика – раздел математики, изучающий дискретные (конечные) математические структуры.

В рамках учебных программ дискретная математика обычно рассматривается как совокупность разделов, связанных с приложениями к информатике и вычислительной технике: теория функциональных систем, теория графов, теория автоматов, теория кодирования, комбинаторика, целочисленное программирование.

Специфика задач дискретной математики в первую очередь предполагает отказ от основных понятий классической математики – предела и непрерывности.

Дискретная и непрерывная математика взаимно дополняют друг друга. Понятия и методы одной часто используются в другой.

1. Множества. Операции над множествами

Понятие *множества* принадлежит к числу фундаментальных неопределяемых понятий математики.

Под *множеством* понимают любое собрание попарно различных объектов, представляемое как единое целое. Эти объекты называются *элементами множества*.

Обычно множества обозначают прописными буквами латинского алфавита, а элементы множеств – строчными буквами.

Если объект x является элементом множества M, то говорят, что *x принадлежит M* и обозначают $x \in$ М. В противном случае говорят, что *x не принадлежит M* и обозначают $x \notin M$.

Универсумом (универсальным множеством) называется множество всех элементов, рассматриваемых в данной задаче. Обозначение: U.

Обозначения основных множеств, изучаемых в математике:

N - множество натуральных чисел, **Z** - целых чисел,**-Q** - рациональных чисел,

R - вещественных чисел, **C** - комплексных чисел.

Определение. Множество A называется *подмножеством* множества B, если

$$\forall x \in A \Rightarrow x \in B;$$

используется обозначение $A \subset B$.

Пример. $\{1, 2\} \subset \{4, 2, a, 1\}$

Определение. Множества A и B называются *равными* (состоящими из одинаковых элементов), если

$$A \subset B \text{ и } B \subset A$$

используется обозначение $A = B$.

Пример. $\{1, a\} = \{a, 1\}$.

Определение. Множество, не содержащее элементов, называется *пустым* и обозначается символом $\varnothing$; пустое множество является подмножеством любого множества.

Способы задания множеств

- *перечислением элементов* множества, например: $M = \{1, 2, a, \beta\}$;
- *описанием характеристических свойств*, которыми должны обладать элементы множества, например: $B = \{x : x \in \mathbf{Z} \text{ и } x > 7\}$;
- *порождающей процедурой*, которая описывает способ получения элементов множества, например, множество чисел Фибоначчи задается порождающей процедурой: $a_1 = 1$, $a_2 = 1$, $a_n = a_{n-1} + a_{n-2}$ для $n > 2$.

Диаграммы Эйлера-Венна

Диаграмма Эйлера-Венна – это представление множеств как единого целого посредством геометрических фигур. Диаграммы Эйлера-Венна используются для иллюстрации операций над множествами.

Построение диаграммы заключается в изображении большого прямоугольника, представляющего универсальное множество U, а внутри его – кругов (или каких-нибудь других замкнутых фигур), представляющих множества. Точки, лежащие внутри различных областей диаграммы, рассматриваются как элементы соответствующих множеств.

Парадокс Рассела

Задание множества характеристическим свойством может приводить к противоречию.

Например, если множество $Y = \{X : X \notin X\}$ существует, то нельзя ответить на вопрос $Y \in Y$?.

Действительно, если предположить, что $Y \in Y$, то из задания множества Y, следует, что $Y \notin Y$, если же предположить, что $Y \notin Y$, то получим $Y \in Y$.

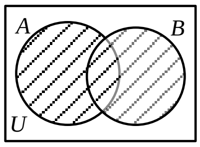
Полученное неразрешимое противоречие называется *парадоксом Рассела*.

Что бы таких парадоксов не возникало, при задании множества характеристическим свойством явно или нет указывают известное, заведомо существующее множество (универсум), которому элементы задаваемого множества, например, $B = \{x : x \in \mathbf{Z} \text{ и } x > 7\}$.

Операции над множествами

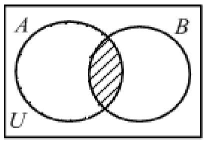
Определение. *Объединением* множеств A и B называется множество

$$A \cup B = \{x : x \in A \vee x \in B\}.$$



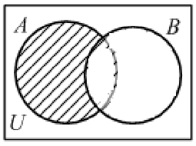
Определение. *Пересечением* множеств A и B называется множество

$$A \cap B = \{x : x \in A \wedge x \in B\}.$$



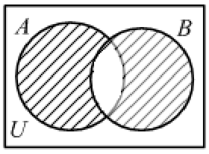
Определение. *Разностью* множеств A и B называется множество

$$A \setminus B = \{x : x \in A \wedge x \notin B\}.$$



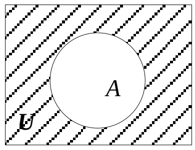
Определение. *Симметрической разностью* множеств A и B называется множество

$$A \Delta B = (A \setminus B) \cup (B \setminus A).$$



Определение. *Дополнением* множества A до универсального множества U называется множество

$$\overline{A} = U \setminus A.$$



Основные свойства операций над множествами

Коммутативность объединения и пересечения

$A \cup B = B \cup A$ $\qquad$ $A \cap B = B \cap A$

Ассоциативность объединения и пересечения

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

Дистрибутивность объединения относительно пересечения и пересечения относительно объединения

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

Свойства операций над пустым и универсальным множествами

$A \cup \varnothing = A$ $\qquad$ $A \cap U = A$

$A \cup \overline{A} = U$ $\qquad$ $A \cap \overline{A} = \varnothing$

$A \cup U = U$ $\qquad$ $A \cap \varnothing = \varnothing$

Свойства идемпотентности объединения и пересечения

$A \cup A = A$ $\qquad$ $A \cap A = A$

Свойства де Моргана

$\overline{A \cup B} = \overline{A} \cap \overline{B}$ $\qquad$ $\overline{A \cap B} = \overline{A} \cup \overline{B}$

Свойства поглощения

$A \cup (A \cap B) = A$ $\qquad$ $A \cap (A \cup B) = A$

Свойства склеивания

$(A \cap B) \cup (A \cap \overline{B}) = A$ $\qquad\qquad$ $(A \cup B) \cap (A \cup \overline{B}) = A$

Свойства Порецкого

$A \cup (\overline{A} \cap B) = A \cup B$ $\qquad\qquad$ $A \cap (\overline{A} \cup B) = A \cap B$

Свойство двойного дополнения

$\overline{\overline{A}} = A$

Для доказательства каждого тождества следует применить определение операций и воспользоваться законами алгебры высказываний.

Докажем свойство коммутативность объединения $A \cup B = B \cup A$.

По определению равенства двух множеств необходимо доказать два включения:

$$A \cup B \subset B \cup A \text{ и } B \cup A \subset A \cup B.$$

1) Доказательство $A \cup B \subset B \cup A$.

$$\forall x \in A \cup B \overset{опр.\cup}{\Rightarrow} \underbrace{x \in A}_{C} \vee \underbrace{x \in B}_{D} \overset{C \vee D = D \vee C}{\Rightarrow} x \in B \vee x \in A \overset{опр.\cup}{\Rightarrow} x \in B \cup A,$$

следовательно, по определению включения: $A \cup B \subset B \cup A$.

2) Доказательство $B \cup A \subset A \cup B$.

$$\forall x \in B \cup A \overset{опр.\cup}{\Rightarrow} \underbrace{x \in B}_{D} \vee \underbrace{x \in A}_{C} \overset{D \vee C = C \vee D}{\Rightarrow} x \in A \vee x \in B \overset{опр.\cup}{\Rightarrow} x \in A \cup B,$$

следовательно, по определению включения: $B \cup A \subset A \cup B$.

Доказанные два включения доказывают коммутативность объединения $A \cup B = B \cup A$.

Мощность конечных множеств

Определение. Мощностью $|A|$ конечного множества A называется количество его элементов.

Лемма (формула включений-исключений для множеств).

Пусть $A_1, A_2, \ldots, A_n$ – конечные множества, тогда

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i} |A_i| - \sum_{i<j} |A_i \cap A_j| + \sum_{i<j<k} |A_i \cap A_j \cap A_k| - \ldots + (-1)^{n-1} \left|\bigcap_{i=1}^{n} A_i\right|.$$

Доказательство.

n = 2. В сумме $|A_1| + |A_2|$ элементы пересечения $A_1 \cap A_2$ подсчитываются дважды, следовательно $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$.

n = 3.

$$|A_1 \cup A_2 \cup A_3| \overset{\substack{ассоциативность \\ объединения}}{=} |(A_1 \cup A_2) \cup A_3| \overset{n=2}{=} |A_1 \cup A_2| + |A_3| - |(A_1 \cup A_2) \cap A_3| =$$

$$\overset{n=2}{=} |A_1|+|A_2|-|A_1\cap A_2|+|A_3|+|(A_1\cup A_2)\cap A_3| =$$

$$\overset{\text{дистрибутивность}}{=} |A_1|+|A_2|-|A_1\cap A_2|+|A_3|-|(A_1\cap A_3)\cup(A_2\cap A_3)| =$$

$$\overset{n=2}{=} |A_1|+|A_2|-|A_1\cap A_2|+|A_3|-|A_1\cap A_3|-|A_2\cap A_3|+|(A_1\cap A_3)\cap(A_2\cap A_3)| =$$

$$\overset{\substack{\text{ассоциативность}\\ \text{пересечения;}\\ A_3\cap A_3=A_3}}{=} |A_1|+|A_2|+|A_3|-|A_1\cap A_2|-|A_1\cap A_3|-|A_2\cap A_3|+|A_1\cap A_2\cap A_3|.$$

В общем случае индукционный переход от n к $n + 1$ аналогичен доказательству для $n = 3$.

Булеан

Определение. Множество всех подмножеств множества A называется *булеаном* и обозначается 2^A.

Теорема. Если A – конечное множество, то $\left|2^A\right| = 2^{|A|}$.

Прямое (декартово) произведение множеств

Определение. *Упорядоченным набором* $(a_1, a_2,\ldots, a_n)$ из n элементов (*кортежем длины* n, *упорядоченной* n-кой), $n \geq 2$, называется множество, определяемое индуктивно:

$$(a_1, a_2) = \{\{a_1\}, \{a_1, a_2\}\},$$

$$(a_1, a_2, a_3) = ((a_1, a_2), a_3),$$

$$(a_1, a_2,\ldots, a_n) = ((a_1, a_2,\ldots, a_{n-1}), a_n),\ n \geq 4.$$

Примечание. Упорядоченный набор из двух элементов (a, b) принято называть *упорядоченной парой*.

Лемма (характеристическое свойство кортежей).

$$(a_1, a_2,\ldots, a_n) = (b_1, b_2,\ldots, b_n) \Leftrightarrow a_i = b_i,\ i = 1, 2,\ldots, n.$$

Доказательство леммы см. Приложение на стр. 46.

Определение. *Прямым произведением множеств* A_1, A_2, …, A_n, $n \geq 2$, называется множество кортежей длины n:

$$A_1 \times A_2 \times\ldots\times A_n = \{(a_1, a_2,\ldots, a_n) : a_i \in A_i,\ i = 1, 2,\ldots, n\}$$

Пример. $\{1, 2\} \times \{a, b\} = \{(1, a), (1, b), (2, a), (2, b)\}$.

Разбиения и покрытия множеств

Определение. Семейство подмножеств $A_1, A_2,\ldots, A_n$ множества A называется:

покрытием множества A, если $A = A_1 \cup A_2\cup\ldots\cup A_n$,

разбиением множества A, если $A = A_1 \cup A_2\cup\ldots\cup A_n$ и $A_i \cap A_j = \varnothing$, $i \neq j$.

Пример. $A = \{1, 2, 3, 4, 5\}$,

семейство $\{1, 2, 5\}, \{3, 4, 5\}, \{5\}, \{1, 3\}$ является покрытием множества A,

семейство $\varnothing, \{1, 2, 5\}, \{3, 4\}$ является разбиением множества A.

Представление множеств в ЭВМ

Представление множеств в ЭВМ подразумевает описание способа хранения информации об элементах множества и алгоритмов выполнения операций над множествами.

*Представление множеств массивами***.** Пусть мощность универсального множества U равна n не превосходит разрядности ЭВМ.

Пронумеруем элементы универсального множества $U = \{u_1, u_2,\ldots, u_n\}$. Тогда любое подмножество $A \subset U$ можно представить массивом C_A, в котором

$$C_A[i] = \begin{cases} 1, \text{ если } u_i \in A, \\ 0, \text{ если } u_i \notin A. \end{cases}$$

Если множество $B \subset U$ представляется массивом C_B, тогда:

объединение $A \cup B$ представляется массивом $C_{A\cup B}$, где $C_{A\cup B}[i] = max(C_A[i], C_B[i])$,

пересечение $A \cap B$ представляется массивом $C_{A\cap B}$, где $C_{A\cap B}[i] = min(C_A[i], C_B[i])$,

разность $A \backslash B$ представляется массивом $C_{A\backslash B}$, где $C_{A\backslash B}[i] = C_A[i]\cdot(C_A[i] - C_B[i])$.

*Представление однонаправленными списками***.** Множество представляется списком элементов типа запись (структура), информационные поля которых содержат элементы множества (их порядковые номера).

Специальные представления. В некоторых языках программирования предусмотрен специальный тип данных для представления множеств, например, в языках Паскаль и Python – это тип *set*.

Коды Грея и задачи перебора

Единственным способом решения многих задач дискретной математики является перебор всех или части подмножеств данного множества. Примером такой задачи является нахождение в данном множестве (массиве) пары элементов с максимальной суммой.

Возможный вариант перебора для n-элементного множества (массива) A:

- введем вспомогательный массив $C = \left\{ \underbrace{0, 0, \ldots 0}_{n} \right\}$,
- считая элементы C числами в 2СС, будем поочередно увеличивать их на 1,

- получив очередное «двоичное число», образуем подмножество
$$D = \{A[i]:\ C[i] = 1,\ i = 1,\ldots, n\} \subset A.$$

Так количество двоичных n-разрядных чисел равно 2^n, то получим все подмножества множества A.

Получение таким образом подмножеств имеет существенный недостаток – часто придется менять значения нескольких разрядов, например, при получении 1000 из 0001.

Для существенного сокращения количества изменяемых разрядов используется код Грея.

Определение. Бинарным кодом Грея порядка n называется последовательность всех 2^n n-разрядных двоичных чисел, в которой любые два соседних числа различаются только в одном разряде.

Пример бинарного кода Грея 3-го порядка:

000 001 011 010 110 111 101 100

С алгоритмом построения бинарного кода Грея Вы познакомитесь в лабораторной работе №2.

2. Отношения между множествами

Определение. n-арным отношением ρ между множествами $A_1,\ldots, A_n$ называется любое подмножество их прямого произведения, $\rho \subset A_1 \times \ldots \times A_n$.

Так как отношения являются множествами, то:

- они могут быть заданы перечнем элементов, характеристическим свойством, либо порождающей процедурой,
- над ними допустимы все теоретико-множественные операции.

Бинарные отношения

Определение. Если ρ есть бинарное отношение между множествами A и B ($\rho \subset A \times B$), то:

- $D_\rho = \{a \in A\colon \exists b \in B, (a, b) \in \rho\}$ называется *областью определения* ρ;
- $E_\rho = \{b \in B : \exists a \in A, (a, b) \in \rho\}$ называется *областью значений* ρ;
- наряду с записью $(a, b) \in \rho$ употребляется и запись $a\rho b$.

Пример.

$A = \{1, 2\}$, $B = \{x, y\}$, $A \times B = \{(1, x), (1, y), (2, x), (2, y)\}$,

$\rho = \varnothing$, $D_\rho = E_\rho = \varnothing$,

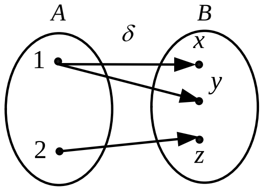
$\delta = \{(1, x), (2, x)\}$, $D_\delta = \{1, 2\}$, $E_\delta = \{x\}$.

Графическое представление бинарных отношений

Пусть $\rho \subset A \times B$. Представим множества A и B диаграммами Эйлера-Венна, их элементы точками, если $(a, b) \in \rho$, то соединим точки a и b стрелкой, получим *графическое представление бинарного отношения*.

Пример. $A = \{1, 2\}$, $B = \{x, y, z)\}$, $\delta = \{(1, x), (2, z), (1, y)\}$.

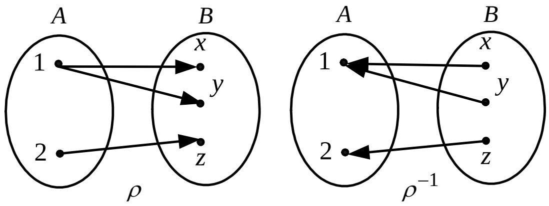
Графическое представление δ:



Операции над бинарными отношениями

Определение. Обратным к бинарному отношению $\rho \subset A \times B$ называется бинарное отношение $\rho^{-1} \subset B \times A$, где $\rho^{-1} = \{(b, a) : (a, b) \in A \times B\}$

Пример.



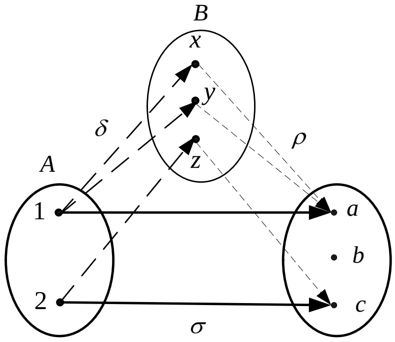
Определение. Композицией бинарных отношений $\delta \subset A \times B$ и $\rho \subset B \times C$ называется бинарное отношение $\sigma = \delta \circ \rho \subset A \times C$,

$\sigma = \{(a, c) : a \in A, c \in C$, для которых $\exists b \in B$, что $(a, b) \in \delta$ и $(b, c) \in \rho\}$.

Пример. $A = \{1, 2\}$, $B = \{x, y, z)\}$, $C = \{a, b, c)\}$,

$\delta = \{(1, x), (2, z), (1, y)\} \subset A \times B$, $\rho = \{(x, a), (y, a), (z, c)\} \subset B \times C$,

$\sigma = \{(1, a), (2, c)\} = \delta \circ \rho \subset A \times C$



Теорема. Для любых бинарных отношений δ, ρ, σ

$\left(\delta^{-1}\right)^{-1} = \delta$,

если $\exists\, \delta \circ \rho$, то $\left(\delta \circ \rho\right)^{-1} = \rho^{-1} \circ \delta^{-1}$, если $\exists(\delta\circ\rho)\circ\sigma$, то $(\delta\circ\rho)\circ\sigma=\delta\circ(\rho\circ\sigma)$.

Матрица бинарного отношения

Определение. Пусть $C = \{c_1,\ldots, c_m\}$, $D = \{d_1,\ldots, d_n\}$, $\rho \subset C \times D$.

Матрицей бинарного отношения ρ называется матрица M_ρ,

$$M_\rho=\left(a_{ij}\right)_{m\times n},\text{ где } a_{ij}=\begin{cases}1, \text{если } \left(c_i,\mathrm{d}_j\right)\in\rho,\\ 0, \text{если } \left(c_i,\mathrm{d}_j\right)\notin\rho.\end{cases}\text{t}$$

Пример. $C = \{1, 2, 3\}$, $D = \{d, e\}$, $\rho = \{(3, e), (1, d)\}$, тогда

$$M_\rho = \begin{array}{c c} & \begin{array}{cc} d & e \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix} \end{array}$$

Теорема. Свойства матриц бинарных отношений

Пусть $\delta \subset C \times D$, $\rho \subset C \times D$ и $\sigma \subset \mathrm{D} \times \mathrm{E}$ – бинарные отношения с матрицами

$$M_\delta=\left(a_{ij}\right)_{m\times n},\ M_\rho=\left(b_{ij}\right)_{m\times n}\text{ и } M_\sigma=\left(c_{ij}\right)_{n\times k},$$

тогда

$$M_{\delta\cup\rho}=\left(max\left(a_{ij},b_{ij}\right)\right)_{m\times n},\qquad M_{\delta\cap\rho}=\left(min\left(a_{ij},b_{ij}\right)\right)_{m\times n},\qquad M_{\delta^{-1}}=M_\delta^T,$$

$$M_{\delta\circ\sigma}=\begin{pmatrix} max\left[min\left(a_{11},c_{11}\right),\ldots,min\left(a_{1n},c_{n1}\right)\right] & \ldots & max\left[min\left(a_{11},c_{1k}\right),\ldots,min\left(a_{1n},c_{nk}\right)\right] \\ \ldots & \ldots & \ldots \\ max\left[min\left(a_{m1},c_{11}\right),\ldots,min\left(a_{mn},c_{n1}\right)\right] & \ldots & max\left[min\left(a_{m1},c_{1k}\right),\ldots,min\left(a_{mn},c_{nk}\right)\right] \end{pmatrix}.$$

Примечание. Элементы матрицы композиции двух бинарных отношений получаются как элементы произведения двух матриц, в котором вместо произведения используется функция *max*, а вместо суммы – функция *min*.

Виды бинарных отношений на множестве

Определение. Бинарное отношение $\delta \subset A \times A$ называется *бинарное отношение на множестве A*.

Определение. Бинарное отношение δ на множестве A называется:

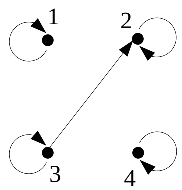
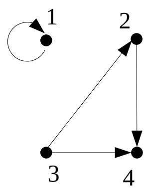
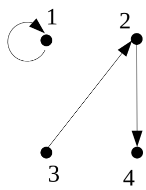
- *рефлексивным*, если $(a,\ a)\in\delta$ для $\forall a \in A$,
- *симметричным*, если $(a,\ b) \in \delta \Rightarrow (b,\ a) \in \delta$ для $\forall a, b \in A$,
- *антисимметричным*, если $(a,\ b) \in \delta$ и $(b,\ a) \in \delta \Rightarrow a = b$ для $\forall a, b \in A$,
- *транзитивным*, если $(a,\ b) \in \delta$ и $(b,\ c) \in \delta \Rightarrow (a,\ c) \in \delta$ для $\forall a, b, c \in A$.

Определение. Бинарное отношение на множестве A называется:

- *отношением эквивалентности*, если оно рефлексивное, симметричное и транзитивное на A,
- *отношением нестрогого порядка*, если оно рефлексивное, антисимметричное и транзитивное на A,
- *отношением строгого порядка*, если оно нерефлексивное, несимметричное и транзитивное множестве A

Примеры.

1. А = {1, 2, 3, 4}

		
рефлексивное, несимметричное, транзитивное (т.к. не нетранзититвное).	нерефлексивное, несимметричное, транзитивное.	нерефлексивное, несимметричное, нетранзититвное.

2. Отношение = на множестве $\mathbf{N}$ есть отношение эквивалентности.

3. Отношение ≤ на множестве $\mathbf{N}$ есть отношение нестрогого порядка.

4. Отношение < на множестве $\mathbf{N}$ есть отношение строгого порядка.

Фактор-множества

Определение. Пусть δ есть отношение эквивалентности на множестве A, $a \in A$.

Множество $\overline{a} = \{b \in A : (a,b) \in \delta\}$ называется *классом эквивалентности* элемента $a \in A$.

Обозначим через A/δ множество всех классов эквивалентности элементов множества A.

Определение. Множество A/δ называется *фактор-множеством множества* A относительно δ.

Пример. A = {1, 2, 3, 4, 5}, δ = {(1, 1), (2, 2), (3, 3), (1, 2), (2, 1)},

$\overline{1}$ = {1, 2}, $\overline{2}$ = {1, 2}, $\overline{3}$ = {3},

A/δ = { $\overline{1}$, $\overline{3}$ }.

Теорема. Пусть δ есть отношение эквивалентности на непустом множестве A, тогда фактор-множество A/δ образует разбиение множества A.

Доказательство. Так как $\forall a \in A \Rightarrow \overline{a} \in A/\delta$, то $A = \bigcup\limits_{\overline{a} \in A/\delta} \overline{a}$.

Осталось доказать, что $\overline{a} \cap \overline{b} = \varnothing$, если $\overline{a} \neq \overline{b}$.

Действительно, если $\overline{a} \cap \overline{b} \neq \varnothing$, то $\exists c \in \overline{a}$ и $c \in \overline{b}$.

Следовательно, $(c, a) \in \delta$ и $(c, b) \in \delta$, отсюда в силу симметричности и транзитивности δ, следует, что $(a, b) \in \delta$, то есть $\overline{a} = \overline{b}$, что противоречит условию $\overline{a} \neq \overline{b}$.

Теорема. Для любого разбиения $\{A_i\}$ непустого множества A существует отношение эквивалентности δ, что $A/\delta = \{A_i\}$.

Доказательство. Достаточно положить $\overline{a} = A_i$ для любого $a \in A_i$.

Пример. $A = \{1, 2, 3\}$, $\{A_i\} = \{\{1, 3\}, \{2\}\}$,

$\overline{1} = \{1, 3\}$, $\overline{2} = \{2\}$, $\delta = \{(1, 1), (3, 3), (1, 3), (3, 1), (2, 2)\}$.

3. Функции

Определение. Бинарное отношение $f \subset X \times Y$ называется *функцией*, если

$$(x, y) \in f \text{ и } (x, z) \in f \Rightarrow y = z,$$

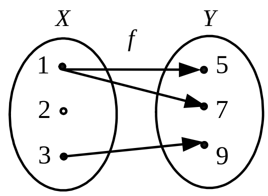
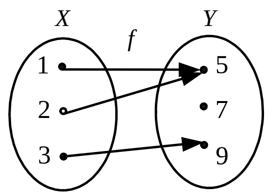
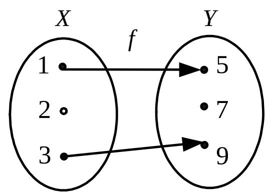
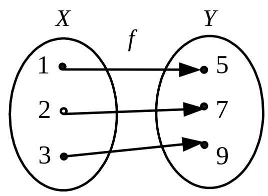
при этом

- $D_f = \{x \in X : \exists y \in Y, (x, y) \in f\}$ называется *областью определения* функции f,
- $R_f = \{y \in Y : \exists x \in X, (x, y) \in f\}$ называется *областью значений* функции f,
- используют обозначение $f: X \to Y$, $y = f(x)$.

Определение. Функция f называется:

- *инъективной*, если $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$,
- *сюръективной*, если для $\forall y \in Y\ \exists x \in X$, что $y = f(x)$,
- *биективной*, если f и сюръективная и инъективная.

Пример.

			
отношение, но не функция.	не инъективная и не сюръективная функция.	инъективная, но не сюръективная функция.	биективная функция.

Определение. *Суперпозицией функций* $f: X \to Y$ *и* $g: Y \to Z$ называется композиция бинарных отношений $f \circ g : X \to Z$.

4. Элементы комбинаторики

5. Основные понятия теории графов

Графические представления в широком смысле – любые наглядные отображения исследуемой системы, процесса, явления на плоскости. К ним могут быть отнесены рисунки, чертежи, графики зависимостей характеристик, планы-карты местностей, блок-схемы процессов, диаграммы и т.п. Такие изображения наглядно представляют различные взаимосвязи и взаимообусловленности: топологические, хронологические, логические, структурные, причинно-следственные и др.

В последние несколько лет теория графов стала важнейшим математическим инструментом, широко используемым во многих областях науки, начиная с исследования операций и лингвистики и кончая химией и генетикой. В то же самое время она выросла в самостоятельную математическую дисциплину. Для ее понимания требуется только знание элементарной теории множеств и теории матриц.

3.1. Основные определения

Определение. Пара (V(G), E(G)) называется **простым графом**, если V(G) – непустое конечное множество элементов, называемых **вершинами** (или узлами, или точками), а E(G) – конечное множество неупорядоченных пар различных элементов из V(G), называемых **ребрами** (или линиями). В простом графе данную пару вершин может соединять не более чем одно ребро.

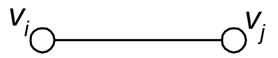
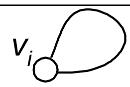
Между элементами множества вершин и множества ребер определено отношение инцидентности. Говорят, что ребро e инцидентно вершинам v_1, v_2, если оно соединяет эти вершины и наоборот, каждая из вершин v_1, v_2 инцидентна ребру e.

Рассмотрим графическое представление графов (табл. 66).

Таблица 66

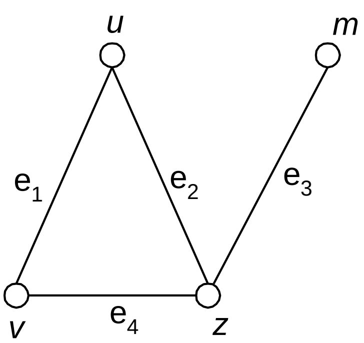
Графическое представление графов

	Элементы графов	Геометрические элементы
.	$x \in X$ – вершина	○ - точка в пространстве
.	$(v_i, v_j) \in V \wedge (v_i, v_j) \notin V$	$v_i \bigcirc\!\!\longrightarrow\!\!\bigcirc v_j$ - направленный отрезок, ориентированное

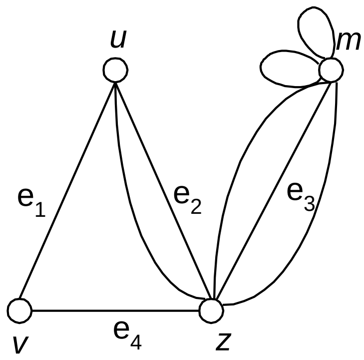
		ребро
.	$(v_i, v_j) \in V \wedge (v_i, v_j) \in V$	 - отрезок, неориентированное ребро
.	$(v_i, v_i) \in V$	 - замкнутый отрезок, петля

Многие результаты, полученные для простых графов, без труда модно перенести на более общие объекты, в которых две вершины могут быть соединены более чем одним ребром. Кроме того, часто бывает удобно снять ограничение, состоящее в том, что ребро должно соединять две различные вершины, и допустить существование петель. Получающийся при этом объект, в котором могут быть и кратные ребра и петли называется **графом (псевдографом)**. Псевдограф без петель называется **мультиграфом**.

Пример 68.



Простой граф



Граф (псевдограф)

Определение. Если ребро задаётся упорядоченной парой вершин, то оно является **ориентированным**. Если каждое ребро графа ориентированное, то граф называется **ориентированным** или **орграфом**.

Иногда удобно преобразовать неориентированный граф в ориентированный – заменой одного неориентированного ребра парой ребер с противоположной ориентацией.

Приведем ряд понятий и определений для ориентированных и неориентированных графов. Там, где это специально не оговорено, те же понятия и определения переносятся без изменений на ориентированные и неориентированные псевдографы.

3.1.1. Смежность, инцидентность, степени

Определение. Если e = {v, w} – ребро графа, то вершины v, w называются **концами ребра e**; в этом случае также говорят, что **ребро** e **соединяет вершины** v, w.

Определение. Если e = {v, w} – дуга орграфа, то вершина v называется **началом**, а вершина w – **концом дуги** e; в этом случае также говорят, что дуга e исходит из вершины v и заходит в вершину w.

Между элементами множества вершин и множества ребер определено отношение инцидентности. Говорят, что ребро e инцидентно вершинам v, w, если оно соединяет эти вершины и наоборот, каждая из вершин v, w инцидентна ребру e.

Определение. Две вершины называются **смежными**, если существует ребро, концами которого они являются. Два ребра называются смежными, если они имеют общую вершину.

Определение. **Степенью** вершины v графа G называется число $\delta(v)$ ребер графа, которым инцидентна эта вершина.

Определение. Вершина, локальная степень которой равна 0, называется **изолированной**; степень которой равна 1 – **висячей**.

Замечание. В случае неориентированного псевдографа обычно считается, что вклад каждой петли, инцидентной некоторой вершине v, равен 2 (тогда как вклад любого другого ребра, инцидентного вершине v, равен 1).

Определение. **Полустепенью исхода (захода)** вершины v орграфа D называется число $\delta^+(v)$ ($\delta^-(v)$) дуг орграфа D, исходящих из вершины v (заходящих в вершину v).

Замечание. В случае ориентированного псевдографа вклад каждой петли, инцидентной некоторой вершине v, равен 1, как в $\delta^+(v)$, так и в $\delta^-(v)$.

Количество вершин и ребер в графе G обозначим соответственно через n(G) и m(G), а количество вершин и дуг в орграфе D – через n(D) и m(D).

Утверждение. Для любого псевдографа G выполняется равенство $\sum_{v\in V}\delta(v) = 2m(G)$.

Утверждение. Для любого ориентированного псевдографа D выполняется равенство $\sum_{v\in V}\delta^+(v) = \sum_{v\in V}\delta^-(v) = m(G)$.

Пример 69.

Найти локальные степени графа (рис. 19) и орграфа (рис. 20).

Решение.

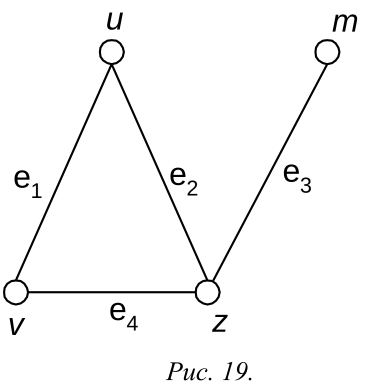


Рис. 19.

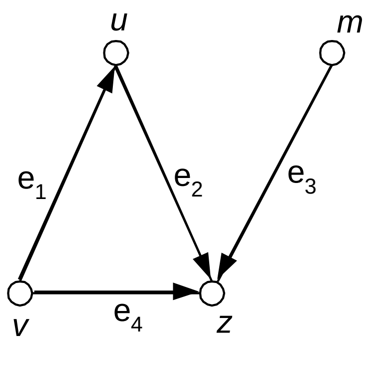


Рис. 20.

$\delta^+(u) = 1$; $\delta^-(u) = 1$;

$\delta(u) = 2$;

$\delta(v) = 2$;

$\delta(z) = 3$;

$\delta(m) = 1$.

$\delta^+(v) = 2$; $\delta^-(v) = 0$;

$\delta^+(z) = 0$; $\delta^-(z) = 3$;

$\delta^+(m) = 1$; $\delta^-(m) = 0$.

3.1.2. Изоморфизм, гомеоморфизм

Определение. Два графа G_1 и G_2 называются **изоморфными**, если существует взаимно однозначное соответствие между множествами их вершин, обладающее тем свойством, что число ребер, соединяющих любые две вершины в G_1, равно число ребер, соединяющих соответствующие вершины в G_2.

Из определения следует, что изоморфные графы можно одинаково изображать графически и отличаться они будут только метками вершин.

Изоморфизм графов есть отношение эквивалентности.

Пример 70.

На рисунке 21 изображены три изоморфных графа. На рисунке 22 изображены два неизоморфных графа.

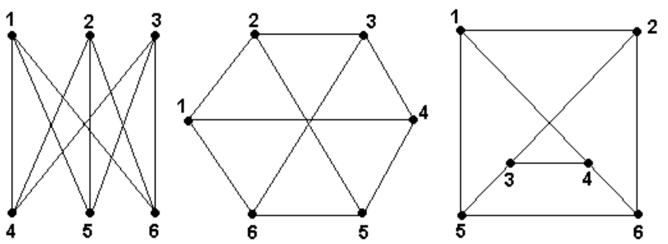


Рис. 21.

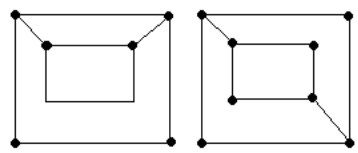


Рис. 22.

Графы рассматриваются с точностью до изоморфизма, то есть рассматриваются классы эквивалентности по отношению изоморфизма.

Пример 71.

С точностью до изоморфизма существует ровно четыре простых графа с тремя вершинами и одиннадцать с четырьмя вершинами. Постройте эти графы.

Решение.

1. Построим четыре простых графа с тремя вершинами с точностью до изоморфизма (рис. 23):

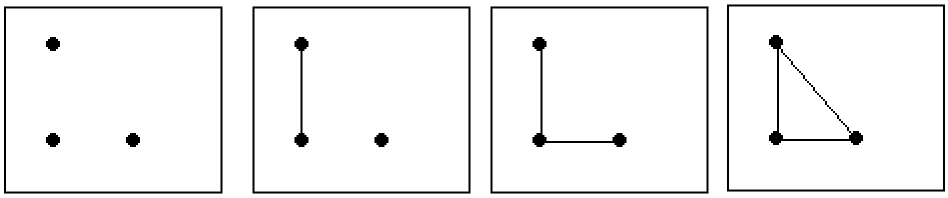


Рис. 23.

2. Построим одиннадцать простых графов с четырьмя вершинами с точностью до изоморфизма (рис. 24):

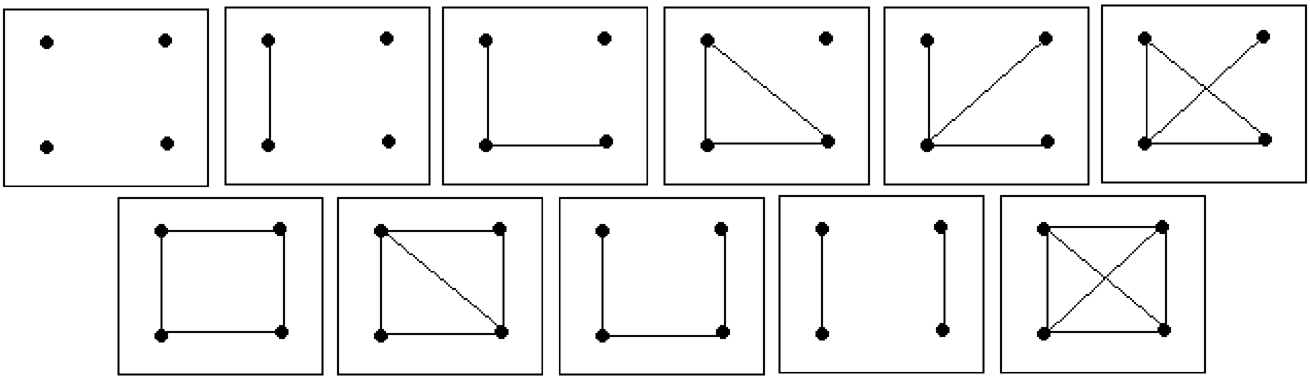


Рис. 24.

Определение. **Операция подразбиения (измельчения) дуги** (u, v) в орграфе D = (V, E) состоит в удалении из E дуги (u, v), добавлении к V новой вершины w и добавлении к E | {(u, v)} двух дуг (u, v), (w, v). Аналогично определяется операция подразбиения ребра в графах.

Определение. Орграф D_1 называется **подразбиением** орграфа D_2, если орграф D_1 можно получить из D_2 путем последовательного применения операции подразбиения дуг. Аналогично определяется подразбиение графа.

Определение. Орграфы D_1, D_2 называются **гомеоморфными**, если существуют их подразбиения, являющиеся изоморфными.

3.1.3. Матричное задание графов. Матрицы смежности, инцидентности

Пусть D = (V, X) – **орграф**, где $V=\{v_1, v_2, \ldots, v_n\}$, $X=\{x_1, x_2, \ldots, x_m\}$.

Определение. **Матрицей смежности** орграфа D называется квадратная матрица A(D)=[a_{ij}] порядка n, у которой

$$a_{ij} = \begin{cases} 1, \textit{если } (v_i, v_j) \in X; \\ 0, \textit{ если } (v_i, v_j) \notin X. \end{cases}$$

Определение. **Матрицей инцидентности** орграфа D называется (n×m) –матрица B(D)=[b_{ij}], у которой

$$b_{ij} = \begin{cases} 1, \textit{ если веошина } v_i \textit{ является концом дуги } x_j; \\ -1, \textit{ если вершина } v_i \textit{ являяется началом дуги } x_j; \\ 0, \textit{ если вершина } v_i \textit{ не инцидентна дуге } x_{j.} \end{cases}$$

Введем также матрицы смежности и инцидентности для неориентированных графов. Пусть G = (V, X) – граф, где V={v_1, v_2, …,v_n}, X={x_1, x_2, …, x_m}.

Определение. **Матрицей смежности** графа G называется квадратная матрица A(G)=[a_{ij}] порядка n, у которой

$$a_{ij} = \begin{cases} 1, \textit{если } (v_i, v_j) \in X; \\ 0, \textit{ если } (v_i, v_j) \notin X. \end{cases}$$

Определение. **Матрицей инцидентности** графа G называется (n×m) –матрица B(G)=[b_{ij}], у которой

$$b_{ij} = \begin{cases} 1, \textit{ если веошина } v_i \textit{ инцидентно ребру } x_j; \\ \\ 0, \textit{ если вершина } v_i \textit{ не инцидентна ребру } x_{j.} \end{cases}$$

С помощью введенных матриц удобно задавать графы для обработки на ЭВМ. Используя матрицу смежности легко определить локальные степени вершин графа: сумма элементов матрицы по строке равна локальной степени соответствующей вершины. Для орграфов по строке определяются полустепени исхода, по столбцам – полустепени захода.

Пример 72.

Построить матрицы смежности и инцидентности для графа G = (V, X) (рис. 25).

Решение.

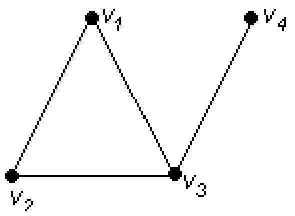


Рис. 25.

Матрица смежности имеет вид

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

Поскольку граф не имеет петель, то на главной диагонали стоят все нули. Для любого графа матрица смежности симметрична относительно главной диагонали.

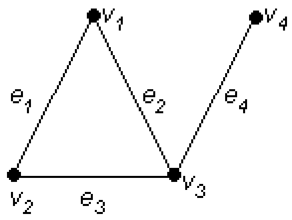


Рис. 26.

Для того, чтобы построить матрицу инцидентности необходимо пронумеровать ребра графа (рис. 26). Матрица инцидентности имеет вид:

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

Напомним, что в строках указываются вершины, в столбцах – ребра. Матрица инцидентности может быть как квадратной, так и прямоугольной.

Пример 73.

Построить матрицы смежности и инцидентности для орграфа D= (V, X) (рис. 27).

Решение.

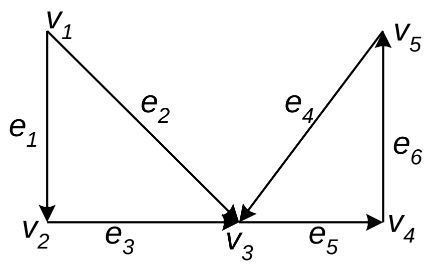


Рис. 27.

Матрица смежности имеет вид:

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

Матрица инцидентности имеет вид

$$\hat{A} = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}.$$

3.1.4. Примеры графов. Операции над графами

Рассмотрим некоторые важные типы графов.

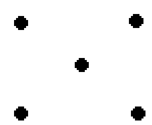


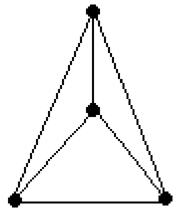
Рис.

Определение. Граф, у которого множество ребер пусто, называется **вполне несвязным** (или **пустым**) графом. Вполне несвязный граф обозначают N_n.

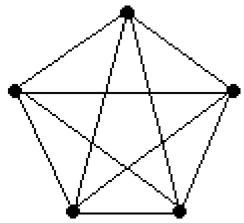
Заметим, что у вполне несвязного графа все вершины изолированы.

Определение. Простой граф, в котором любые две вершины смежны, называется **полным**. Полный граф обозначают K_n.

Пример 74.



K_4

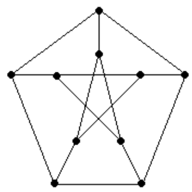


Заметим, что для полного графа выполняется равенство $m=\dfrac{n(n-1)}{2}$, где m – число ребер, n – число вершин графа.

Определение. Граф, у которого все вершины имеют одну и ту же локальную степень n, называется **регулярным** (или **однородным**) степени n.

Регулярные графы степени 3 называются кубическими (или трехвалентными).

Пример 75.

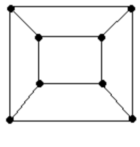


Ри

Известным примером кубического графа является граф Петерсона (рис. 29).

Среди регулярных графов особенно интересны так называемые платоновы графы – графы, образованные вершинами и ребрами пяти правильных многогранников – Платоновых тел: тетраэдра, куба, октаэдра, додекаэдра и икосаэдра.

Пример 76.



Ри

На рис. 30 приведен граф, соответствующий кубу.

Определение. Допустим, что множество вершин графа G можно разбить на два непересекающихся подмножества V_1 и V_2 так, что каждое ребро в G соединяет какую-нибудь вершину из V_1 с какой-нибудь вершиной из V_2, тогда данный граф называется **двудольным**.

Двудольный граф можно определить и по-другому – в терминах раскраски его вершин двумя цветами, скажем красным и синим. При этом граф называется двудольным, если каждую его вершину можно окрасить красным или синим цветом так, чтобы каждое ребро имело один конец красный, а другой – синий.

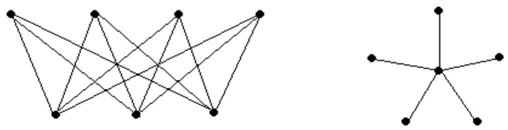
Определение. Если в двудольном графе каждая вершина из V_1 соединена с каждой вершиной из V_2, то граф называется **полным двудольным**.

Обозначение. $K_{m.\ n}$

Заметим, что граф $K_{m.\ n}$ имеет ровно m + n вершин и mn ребер.

Пример 77.

На рис. 31 изображены двудольные графы.



$K_{4,}$ *3*

Рис. 31.

Определение. **Объединением** графов $G_1 = (V_1, X_1),\ G_2 = (V_2,\ X_2)$ называется граф $G_1 \cup G_2 = (V_1 \cup V_2,\ X_1 \cup X_2)$.

Определение. **Пересечением** графов $G_1 = (V_1, X_1),\ G_2 = (V_2,\ X_2)$, где $V_1 \cap V_2 \neq \varnothing$, называется граф $G_1 \cap G_2 = (V_1 \cap V_2,\ X_1 \cap X_2)$.

Определение. **Соединением** графов G_1 и G_2 является новый граф, у которого $V = V_1 \cup V_2$, а множеством ребер являются все ребра первого и второго графа и ребра, соединяющие между собой каждую вершину первого графа с первой вершиной второго графа.

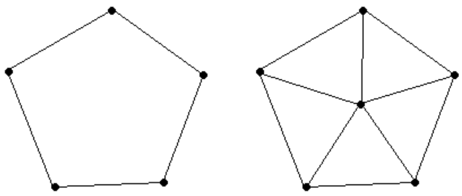
Определение. Граф называется **связным**, если его нельзя представить в виде объединения двух графов, и несвязным в противном случае.

Очевидно, что всякий несвязный граф можно представить в виде объединения конечного числа связных графов – каждый из таких связных графов называется компонентой связности графа.

Определение. Связный регулярный граф степени 2 называется **циклическим** графом. Обозначается C_n (рис. 32).

Определение. Соединение графов N_1 и C_{n-1} ($n \geq 3$) называется **колесом** с n вершинами. Обозначается W_n (рис. 32).

Пример 78.



C_5 W_6

Рис. 32.

Определение. **Дополнением** простого графа G называется простой граф с множеством вершин V(G), в котором две вершины смежны тогда и только тогда, когда они не смежны в исходном графе.

Обозначение. $\overline{G}$

Другими словами, дополнением графа является граф, содержащий все вершины исходного графа и только те ребра, которых не хватает исходному графу для того, чтобы он стал полным.

Определение. **Подграфом** графа G называется граф, все вершины и ребра которого содержатся среди вершин и ребер графа G. Подграф называется **собственным**, если он отличен от самого графа.

3.1.5. Маршруты, цепи, циклы

Введем понятие маршрута для графа G = (V, E) (и соответственно понятие пути для орграфа D = (V, E)).

Определение. **Маршрутом** в данном графе G называется конечная последовательность ребер вида $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{m-1}, v_m\}$ (обозначаемая также $v_0 \to v_1 \to v_2 \to \ldots \to v_m$).

Каждому маршруту соответствует последовательность вершин $v_0v_1v_2\ldots v_m$; v_0 – начальная вершина, а v_m – конечная вершина маршрута. Одна и та же вершина может одновременно оказаться начальной, конечной и внутренней. Таким образом, мы будем говорить о маршруте из v_0 в v_m.

Определение. **Длиной маршрута** называется число ребер в нем.

Определение. Маршрут называется **цепью**, если все его ребра различны, и **простой цепью**, если все вершины тоже различны (кроме, может быть, начальной и конечной вершин).

Определение. Цепь или простая цепь **замкнуты**, если начальная и конечная вершины совпадают.

Определение. Замкнутая простая цепь, содержащая, по крайней мере, одно ребро, называется **циклом**.

Определение. **Обхватом** графа называется длина его кратчайшего цикла.

3.1.6. Связность. Компоненты связности

Определение. **Вершина** w орграфа D (графа G) **достижима** из вершины v, если либо v=w, либо существует путь из v в w(маршрут, соединяющий v, w).

Дадим более удобное определение связных графов.

Определение. Граф называется **связным**, если для любых двух его вершин v, w существует простая цепь из v в w.

Определение. Граф (орграф) называется **связным (сильно связным)**, если для любых двух его вершин v, w существует маршрут (путь), соединяющий v, w (из v и w).

Определение. Орграф называется **односторонне связным**, если для любых его двух вершин, по крайней мере, одна достижима из другой.

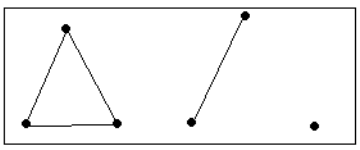
Определение. Если граф не является связным, то он называется **несвязным**.

Определение. **Компонентой связности** графа называется его связный подграф, не являющийся собственным подграфом никакого другого связного подграфа.

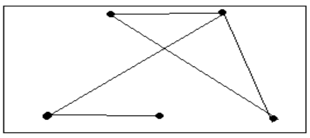
В дальнейшем количество компонент связности графа будем обозначать k.

Пример 79.

Данный граф не является связным: k = 3.



Данный граф является связным: k = 0.



Теорема. Пусть G – простой граф с n вершинами и k компонентами. Тогда число m его ребер удовлетворяет неравенствам

$$n-k \le m \le \frac{(n-k)(n-k+1)}{2}.$$

Следствие. Любой простой граф с n вершинами и более чем (т-1)(т-2)/2 ребрами связен.

При исследовании графов возникает вопрос: насколько сильно связен связный граф? Этот вопрос можно сформулировать и так: сколько ребер нужно удалить из графа, чтобы он перестал быть связным? Под операцией удаления вершин из графа будем понимать операцию, заключающуюся в удалении некоторой вершины вместе с инцидентными ей ребрами.

Определение. Вершина графа, удаление которой увеличивает число компонент связности, называется **разделяющейся**.

Определение. **Разделяющим множеством** связного графа G называется такое множество его ребер, удаление которого приводит к несвязному графу.

Определение. **Разрезом** называется такое разделяющее множество, никакое собственное подмножество которого не является разделяющим.

Определение. Разрез, состоящий из одного ребра, называется **мостом** (перешейком).

Пример 80.

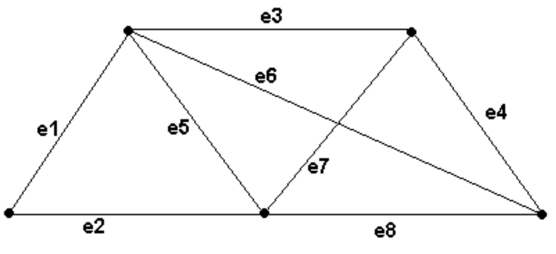


Рис. 33.

Для графа, изображенного на рис.33, каждое из множеств $\{e_1, e_2, e_5\}$ и $\{e_3, e_6, e_7, e_8\}$ является разделяющим.

Разрезом является множество ребер $\{e_1, e_2\}$.

В графе возможно выделить несколько разделяющих множеств и разрезов.

3.2. Задачи поиска маршрутов (путей) в графе (орграфе)

3.2.1. Поиск путей (маршрутов) с минимальным числом дуг (ребер)

При решении некоторых прикладных задач нередко возникает необходимость найти маршрут, соединяющий заданные вершины в графе G. Данная задача решается при использовании алгоритма Тэрри.

Рассмотрим более сложную задачу: нахождение пути (маршрута) с минимальным числом дуг (ребер).

Определение. Путь в орграфе D из вершины v в вершину w, где $v \neq w$, называется **кратчайшим**, если он имеет минимальную длину среди всех путей орграфа D из v и w. Аналогично определяется и кратчайший маршрут в графе G.

Утверждение. Любой кратчайший путь (маршрут) является простой цепью.

Рассмотрим задачу поиска минимального пути (маршрута). Введем некоторые обозначения. Пусть D=(V, X) – орграф, $v \in V$, $V_1 \subset V$. Обозначим $D(v)=\{w \in V | (v, w) \in X\}$ – **образ вершины** v; $D^{-1}(v)=\{w \in V| (w,v) \in X\}$ – **прообраз вершины** v; $D(V_1) = \bigcup_{v \in V_1} D(v)$ - образ множества вершин V_1; $D^{-1}(V_1) = \bigcup_{v \in V_1} D^{-1}(v)$ - прообраз множества вершин V_1. Пусть G=(V, X) – граф, $v \in V$, $V_1 \subset V$. Обозначим $G(v)=\{w \in V|\{v, w\} \in X\}$ – образ вершины v; $G(V_1) = \bigcup_{v \in V_1} G(v)$ - образ множества вершин V_1.

Пусть D=(V, X) – орграф с $n \geq 2$ вершинами и v, w – заданные вершины из V, где $v \neq w$. Опишем алгоритм поиска кратчайшего пути из v в w в орграфе D.

Алгоритм фронта волны.

Шаг 1. Помечаем вершину v индексом 0. Затем помечаем вершины, принадлежащие образу вершины v, индексом 1. Множество вершин с индексом 1 обозначаем $FW_1(v)$. Полагаем k=1.

Шаг 2. Если $FW_k(v)=\varnothing$ или выполняется k=n-1, $w \notin FW_k(v)$, то вершина w не достижима из v, и работа алгоритма на этом заканчивается. В противном случае переходим к шагу 3.

Шаг 3. Если $w \notin FW_k(v)$, то переходим к шагу 4. В противном случае существует путь из v в w длины k, и этот путь является кратчайшим. Последовательность вершин

$vw_1w_2\ldots w_{k-1}w$, где

$$w_{k-1} \in FW_{k-1}(v) \cap D^{-1}(w);$$
$$w_{k-2} \in FW_{k-2}(v) \cap D^{-1}(w_{k-1});$$
....
$$w_1 \in FW_1(v) \cap D^{-1}(w_2),$$

и есть искомый кратчайший путь из v и w. На этом работа алгоритма заканчивается.

Шаг 4. Помечаем индексом k+1 все непомеченные вершины, которые принадлежат образу множества вершин с индексом k. Множество вершин с индексом k+1 обозначаем $FW_{k+1}(v)$. Присваиваем k:=k+1 и переходим к шагу 2.

Замечание. Множество $FW_k(v)$ обычно называют фронтом волны k-го уровня.

Замечание. Вершины $w_1w_2…w_{k-1}$, вообще говоря, могут быть выделены неоднозначно. Эта неоднозначность соответствует случаям, когда существует несколько различных кратчайших путей из v и w.

Пример 81.

Определить кратчайший путь из v_1 и v_6 в орграфе D, заданном матрицей смежности (табл. 67).

Таблица 67

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

Решение.

Согласно алгоритму Фронта волны, последовательно определяем

$FW_1(v_1) = \{v_4, v_5\}$;

$FW_2(v_1) = D(FW_1(v_1))\backslash\{v_1, v_4, v_5\} = \{v_2, v_3\}$;

$FW_3(v_1) = D(FW_2(v_1))\setminus\{v_1, v_2, v_3, v_4, v_5\} = \{v_6\}$.

Таким образом, $v_6 \in FW_3(v_1)$, а значит, существует путь из v_1 в v_6 длины 3, и этот путь является кратчайшим. Найдем теперь кратчайший путь из v_1 в v_6. Определим множество

$$FW_2(v_1) \cap D^{-1}(v_6) = \{v_2, v_3\} \cap \{v_2.v_3\} = \{v_2, v_3\}.$$

Выберем любую вершину из найденного множества, например вершину v_3. Определим далее множество

$$FW_1(v_1) \cap D^{-1}(v_3) = \{v_4, v_5\} \cap \{v_4.v_5, v_6\} = \{v_4, v_5\}.$$

Выберем любую вершину из найденного множества, например вершину v_5. Тогда $v_1v_5v_3v_6$ – искомый кратчайший путь из v_1 в v_6 (длины 3) в орграфе D.

3.2.2. Расстояния в графе. Диаметр, центр, радиус графа

Утверждение. Если для двух вершин существует маршрут, связывающий их, то обязательно найдется минимальный маршрут, соединяющий эти вершины. Обозначим длину этого маршрута через d(v, w).

Определение. Величину d(v, w) (конечную или бесконечную) будем называть **расстоянием между вершинами v, w**. Это расстояние удовлетворяет аксиомам метрики:

1. $d(v, w) \geq 0$, причем $d(v, w) = 0$ тогда и только тогда, когда $v=w$;
2. $d(v, w) = d(w, v)$;
3. $d(v, w) \leq d(v, u) + d(u, w)$.

Определение. **Диаметром** связного графа называется максимально возможное расстояние между двумя его вершинами.

Определение. **Центром** графа называется такая вершина, что максимальное расстояние между ней и любой другой вершиной является наименьшим из всех возможных; это расстояние называется **радиусом** графа.

Пример 82.

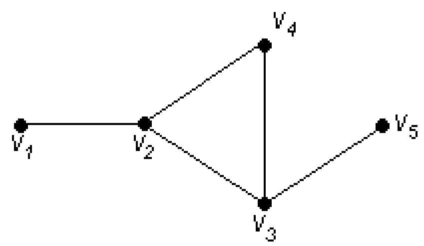


Рис. 34.

Для графа G, изображенного на рисунке 34, найти радиус, диаметр и центры.

Решение.

Чтобы определить центры, радиус, диаметр графа G, найдем матрицу D(G) расстояний между вершинами графа, элементами d_{ij} которой будут расстояния между вершинами

v_i и v_j. Для этого воспользуемся графическим представлением графа. Заметим, что матрица D(G) симметрична относительно главной диагонали. $D(G) = \begin{bmatrix} 0 & 1 & 2 & 2 & 3 \\ 1 & 0 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 3 & 2 & 1 & 2 & 0 \end{bmatrix}$.

С помощью полученной матрицы для каждой вершины графа G определим наибольшее удаление из выражения: $r(v_i) = \max\limits_j d(v_i, v_j)$ для i, j = 1, 2, …, 5. В результате получаем: $r(v_1) = 3$, $r(v_2) = 2$, $r(v_3) = 2$, $r(v_4) = 2$, $r(v_5) = 3$. Минимальное из полученных чисел является радиусом графа G, максимальное – диаметром графа G. Значит, R(G) = 2 и D(G) = 3, центрами являются вершины v_2, v_3, v_4.

3.2.3. Нахождение минимального пути в нагруженном графе

Определение. Назовём орграф $D = (V, X)$ **нагруженным**, если на множестве дуг X определена некоторая функция $l : X \to R$, которую часто называют **весовой** функцией.

Тем самым и нагруженном орграфе D каждой дуге $x \in X$ поставлено в соответствие некоторое действительное число $l(x)$. Значение $l(x)$ будем называть длиной дуги x.

Для любого пути π нагруженного орграфа D обозначим через $l(\pi)$ сумму длин входящих в π дуг, при этом каждая дуга учитывается столько раз, сколько она входит в путь. Величину $l(\pi)$ будем называть длиной пути π в нагруженном орграфе D. Ранее так называлось количество дуг в пути π. В связи с этим заметим, что если длины дуг выбраны равными 1, то $l(\pi)$ выражает введенную ранее длину пути π в ненагруженном орграфе. Следовательно, любой ненагруженный орграф можно считать нагруженным с длинами дуг, равными 1. Аналогично определяется и нагруженный граф, а также длина маршрута в нем.

Определение. Путь в нагруженном орграфе D из вершины υ в вершину ϖ, где $\upsilon \neq \varpi$, называется **минимальным**, если он имеет минимальную длину среди всех путей орграфа D из υ в ϖ. Аналогично определяется и минимальный маршрут в нагруженном графе G.

Рассмотрим теперь задачу поиска минимальных путей (маршрутов) в нагруженном орграфе (графе). При этом для определенности рассуждения будем проводить для орграфа (для графа они аналогичны).

Пусть $D = (V, X)$ - нагруженный орграф, $V = \{\upsilon_1, ..., \upsilon_n\}$, $n \ge 2$. Введем величины $\lambda_i^{(k)}$, где $i = 1, ..., n$, $k = 1, 2, ...$ Для каждых фиксированных i и k величина $\lambda_i^{(k)}$ равна длине минимального пути среди путем из υ_1 в υ_i, содержащих не более k дуг; если же таких путей нет, то $\lambda_i^{(k)} = \infty$. Кроме того, если произвольную вершину $\upsilon \in V$ считать путем из υ в υ нулевой длины, то величины $\lambda_i^{(k)}$ можно ввести также и для $k = 0$, при этом

$$\lambda_1^{(0)} = 0, \lambda_i^{(0)} = \infty, \ i = 2, ..., n.$$

(1)

Введем также в рассмотрение квадратную матрицу $C(D) = \lfloor c_{ij} \rfloor$ порядка n с элементами

$$c_{ij} = \begin{cases} l(\upsilon_i, \upsilon_j), \text{ если } (\upsilon_i, \upsilon_j) \in X; \\ \infty, \text{ если } (\upsilon_i, \upsilon_j) \overline{\in} X, \end{cases}$$

которую будем называть матрицей длин дуг нагруженного орграфа D.

Следующее утверждение дает простые формулы для вычисления величин $\lambda_i^{(k)}$.

Утверждение. *П* $i = 2, p., n, k \ge 0$ *и в ы я р п а*

$\lambda_i^{(k+1)} = \min\limits_{1 \le j \le n} \{\lambda_j^{(k)} + c_{ji}\}$ (2),

а при $i = 1, k \ge 0$ *справедливо равенство* $\lambda_1^{(k+1)} = \min\limits_{1 \le j \le n} \{\lambda_j^{(k)} + c_{j1}\}$ (3) .

Используя данное утверждение, нетрудно описать алгоритм нахождения таблицы значений величин $\lambda_i^{(k)}$ (будем записывать её в виде матрицы, где i - номер строки, $k+1$ - номер столбца). Действительно, используя рекуррентные соотношения (2), (3) и исходя из (1), последовательно определяем набор величин $\lambda_1^{(k)}, ..., \lambda_n^{(h)}$ (($k+1$)-й столбец матрицы), начиная с $k+1$, а затем шаг за шагом увеличиваем значение k до любой необходимой величины.

Алгоритм Форда – Беллмана нахождения минимального пути в нагруженном орграфе D из υ_1 в υ_{i_1} $(i_1 \ne 1)$

Шаг 1. Пусть мы уже составили таблицу величин $\lambda_i^{(k)}, i = 1, 2, ..., n, k = 0, 1, ..., n-1$. Если $i \in \{2, ..., n\}$ не достижима из υ_1 (предполагаем, что все величины $l(x)$, $x \in X$, конечны). В этом случае работа алгоритма заканчивается.

Шаг 2. Пусть $\lambda_{i_1}^{(n-1)} < \infty$. Тогда число $\lambda_{i_1}^{(n-1)}$ выражает длинны любого минимального пути из υ_1 в υ_{i_1} в нагруженном орграфе D. Определим минимальное число $k_1 \geq 1$, при котором выполняется равенство $\lambda_{i_1}^{(k_1)} = \lambda_{i_1}^{(n-1)}$. По определению чисел $\lambda_i^{(k)}$ получим, что k_1 - минимальное число дуг в пути среди всех минимальных путей из υ_1 в υ_{i_1} в нагруженном орграфе D.

Шаг 3. Последовательно определяем номера $i_2, ..., i_{k_1+1}$ такие что

$$\lambda_{i_2}^{(k_1-1)} + c_{i_2,i_1} = \lambda_{i_1}^{(k_1)};$$

$$\lambda_{i_3}^{(k_1-2)} + c_{i_3,i_2} = \lambda_{i_2}^{(k_1-1)};$$

$$. \; . \; . \; . \; . \; . \qquad (4)$$

$$\lambda_{i_{k_1+1}}^{(0)} + c_{i_{k_1+1},i_{k_1}} = \lambda_{i_{k_1}}^{(1)}$$

Из (4) с учётом того, что $\lambda_{i_1}^{(k_1)} = \lambda_{i_1}^{(n-1)} < \infty$, имеем $c_{i_2,i_1} < \infty, ..., c_{i_{k_1+1},i_{k_1}} < \infty, \quad \lambda_{i_{k_1+1}}^{(0)} < \infty$, откуда, используя (1), получаем

$$\upsilon_{i_2}, \upsilon_{i_1}, ..., (\upsilon_{i_{k_1+1}}, \upsilon_{i_{k_1}}) \in X, \quad l(\upsilon_{i_2}, \upsilon_{i_1}) = c_{i_2,i_1}, ..., l(\upsilon_{i_{k_1+1}}, \upsilon_{i_{k_1}}) = c_{i_{k_1+1},i_{k_1}}; \lambda_{i_{k_1+1}}^{(0)} = 0, i_{k_1+1} = 1, \upsilon_{i_{k_1+1}} = \upsilon_1. \qquad (5)$$

Складывая равенства (4) и учитывая (5), имеем

$$l(\upsilon_1 \upsilon_{k_1} ... \upsilon_{i_2} \upsilon_{i_1}) = \lambda_{i_1}^{(k_1)},$$

т.е. $\upsilon_1 \upsilon_{k_1} ... \upsilon_{i_2} \upsilon_{i_1}$ - искомый минимальный путь из υ_1 в υ_{i_1} в нагруженном орграфе D. Заметим, что в этом пути ровно k_1 дуг Следовательно, мы определили путь с минимальным числом дуг среди всех минимальных путей из υ_1 в υ_{i_1} в нагруженном орграфе D.

Замечание. Номера $i_2, i_3, ..., i_{k_1}$, удовлетворяющие (4) вообще говоря, могут быть выделены неоднозначно. Эта неоднозначность соответствует случаям, когда существует несколько различных путей из υ_1 в υ_{i_1} с минимальным числом дуг среди минимальных, путей из υ_1 в υ_{i_1} в нагруженном орграфе D.

Замечание. Алгоритм можно модифицировать, с тем чтобы определить минимальный путь из υ_1 в заданную вершину υ_{i_1} среди путей из υ_1 в υ_{i_1}, содержащих не более k_0 дуг, где k_0 - заданное число, $k_0 \geq 1$. Для этого в алгоритме вместо $\lambda_{i_1}^{(n-1)}$ следует воспользоваться $\lambda_{i_1}^{(k_0)}$.

Пример 83.

Определить минимальный путь из v_1 в v_6 в нагруженном орграфе D, изображенном на рисунке 35.

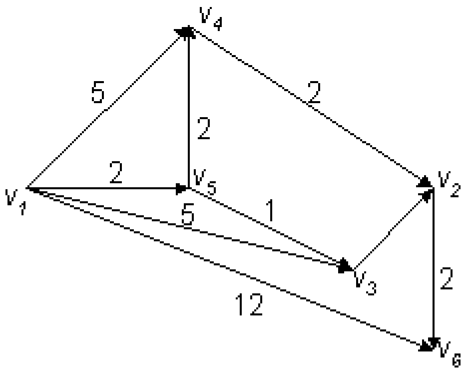


Рис. 35.

Решение.

Составим матрицу C(D) длин дуг нагруженного орграфа D (табл. 68). Справа от матрицы C(D) припишем шесть столбцов, которые будем определять, используя рекуррентное соотношение (2) и исходя из (1).

Величина $\lambda_6^{(5)} = 7$ выражает длину минимального пути из v_1 в v_6 в нагруженном орграфе D. Найдем минимальное число $k_1 \geq 1$, при котором выполняется равенство $\lambda_6^{(k_1)} = \lambda_6^{(5)}$. Получаем, что $k_1 = 4$. Таким образом, минимальное число дуг в пути среди всех минимальных путей из v_1 в v_6 в нагруженном графе D равняется 4. Определим теперь последовательность номеров i_1, i_2, i_3, i_4, i_5, где $i_1 = 6$, удовлетворяющих (4) (для этого используем формулу (2)).

Таблица 68

	1	2	3	4	5	6			$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
1						2								
2														
3														
4														
5														
6									2	2				

Получаем, что в качестве такой последовательности надо взять номера 6, 2, 3, 5, 1, так как

$\lambda_2^{(3)} + \tilde{n}_{2,6} = 5 + 2 = 7 = \lambda_6^{(4)}$;

$\lambda_3^{(2)} + c_{3,2} = 3 + 2 = 5 = \lambda_2^{(3)}$;

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

$\lambda_1^{(0)} + c_{1,5} = 0 + 2 = 2 = \lambda_5^{(1)}$.

Тогда $v_1v_5v_3v_2v_6$ – искомый минимальный путь из v_1 в v_6 в нагруженном орграфе D, причем он содержит минимальное число дуг среди всех возможных минимальных путей из v_1 в v_6 .

3.2.4. Эйлеровы цепи и циклы

Классической в теории графов является следующая задача. В городе Кенигсберге имеется два острова, соединенных семью мостами с берегами реки Преголь и друг с другом так, как показано на рисунке 36. Задача состоит в следующем: осуществить прогулку по городу таким образом, чтобы, пройдя по одному разу по каждому мосту, вернуться обратно. Решение этой задачи сводится к нахождению некоторого специального маршрута в графе.

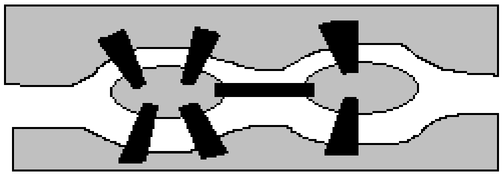


Рис. 36.

Пусть G – псевдограф.

Определение. Цепь (цикл) в G называется **эйлеровой (эйлеровым)**, если она (он) проходит по одному разу через каждое ребро псевдографа G.

Поставим в соответствие схеме мультиграф G, изображенный на рисунке 37, в котором каждой части суши соответствует вершина, а каждому мосту – ребро, соединяющее соответствующие вершины. На языке теории графов задача звучит следующим образом: найти эйлеров цикл в мультиграфе G.

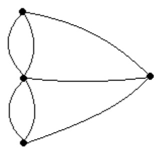


Рис.

Определение. Граф является **эйлеровы**м, если он содержит эйлеров цикл.

Рассмотрим вопрос о наличии эйлеровой цепи и цикла в псевдографе.

Теорема. Связный граф является эйлеровым тогда и только тогда, когда каждая вершина имеет четную локальную степень.

Теорема. Связный граф содержит эйлерову цепь тогда и только тогда, когда ровно две вершины имеют нечетную локальную степень.

Завершая рассмотрение эйлеровых графов, рассмотрим алгоритм построения эйлеровой цепи в данном эйлеровом графе. Этот метод известен под названием **алгоритма Флёри**.

Теорема. Пусть G – эйлеров граф; тогда следующая процедура всегда возможна и приводит к эйлеровой цепи графа G. Выходя из произвольной вершины и, идем по ребрам графа произвольным образом, соблюдая лишь следующие правила:

- стираем ребра по мере их прохождения и стираем также изолированные вершины, которые при этом образуются;
- на каждом этапе идем по мосту только тогда, когда нет других возможностей.

Пример 84.

Любой простой полный граф с нечетным количеством вершин является эйлеровым. Любой циклический граф является эйлеровым. Граф, являющийся колесом, не является эйлеровым.

3.2.5. Гамильтоновы цепи и циклы

Пусть G – псевдограф.

Определение. Цепь (цикл) в G называется **гамильтоновой (гамильтоновыми)**, если она (он) проходит по одному разу через каждую вершину псевдографа G.

Определение. Граф является **гамильтоновым**, если он содержит гамильтонов цикл.

С понятием гамильтоновых циклов тесно связана так называемая задача коммивояжера: в нагруженном графе G определить гамильтонов цикл минимальной длины (иными словами, коммерсант должен совершить поездку по городам и вернуться обратно, побывав в каждом городе ровно один раз, и при этом стоимость такой поездки должна быть минимальной).

Приведем **теорему Дирака**, которая отвечает на вопрос: существует ли в графе гамильтонов цикл.

Теорема. Если в простом графе с n ($\geq$ 3) вершинами локальная степень каждой вершины не менее n/2, то граф является гамильтоновым.

Пример 85.

Любой простой полный граф является гамильтоновым. Любой циклический граф является гамильтоновым. Граф, являющийся колесом, является гамильтоновым.

3.3. Деревья и циклы

3.3.1. Определение и свойства деревьев

Определение. Граф G называется **деревом**, если он является связным и не имеет циклов. Граф G, все компоненты связности которого являются деревьями, называется **лесом**.

Пример 86.

Граф G_1 является деревом (рис.38). Граф G_2 является лесом (рис. 38), он содержит три связные компоненты, каждая из которых является деревом.

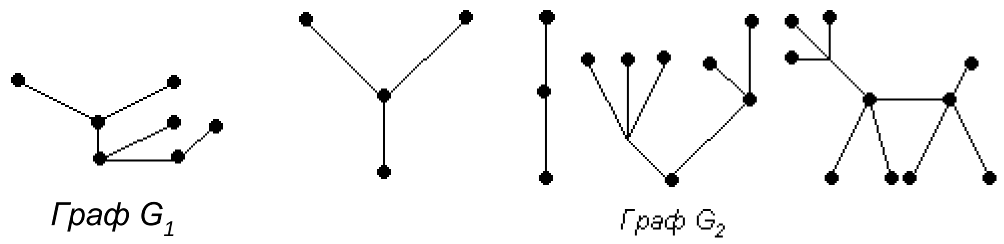


Рис. 38.

Следующие утверждения эквивалентны:

- граф G есть дерево;
- граф G является связным и не имеет простых циклов;
- граф G является связным и число его ребер ровно на единицу меньше числа вершин;
- любые две различные вершины графа G можно соединить единственной (и притом простой) цепью;
- граф G не содержит циклов, но, добавляя к нему любое новое ребро, получаем ровно один (с точностью до направления обхода и начальной вершины обхода) и притом простой цикл (проходящий через добавляемое ребро).

Утверждение. Если у дерева G есть, по крайней мере, одно ребро, то у него обязательно найдется висячая вершина.

Утверждение. Пусть G – дерево. Тогда любая цепь в G будет простой.

3.3.2. Остовное дерево связного графа

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

Пусть G связный граф. Тогда остовное дерево графа G (если оно существует) должно содержать n(G) – 1 ребер. Таким образом, любое остовное дерево графа G есть результат удаления из G ровно m(G)-(n(G)-1)=m(G)-n(G)+1 ребер.

Определение. Число m(G)-n(G)+1 называется **цикломатическим числом** связного графа G и обозначается через v(G).

Замечание. Понятие остовного дерева и цикломатического числа аналогичным образом определяется и для произвольного связного псевдографа G.

Покажем существование остовного дерева для произвольного связного псевдографа G=(V,X), описав алгоритм его выделения.

Шаг 1. Выбираем в G произвольную вершину u_1, которая образует подграф G_1 псевдографа G, являющийся деревом. Полагаем i=1.

Шаг 2. Если i=n, где n=n(G), то задача решена, и G_i – искомое остовное дерево псевдографа G. В противном случае переходим к шагу 3.

Шаг 3. Пусть уже построено дерево G_i, являющееся подграфом псевдографа G и содержащий некоторые вершины $u_1, \ldots, u_i$, где $1 \le i \le n-1$. Строим граф G_{i+1}, добавляя к графу G_i новую вершину $u_{i+1} \in V$, смежную в G с некоторой вершиной u_j графа G_i, и новое ребро (u_{i+1}, u_j) (в силу связности G и того обстоятельства, что $i < n$, указанная вершина u_{i+1} обязательно найдется). Согласно утверждению граф G_{i+1} также является деревом. Присваиваем i:=i+1 и переходим к шагу 2.

Пример 87.

Используя алгоритм, выделим остовное дерево графа G, изображенного на рисунке 39.

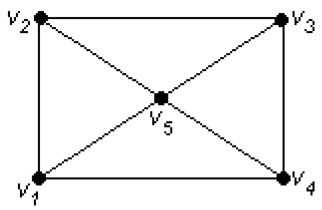


Рис. 39.

Решение.

На рисунке 40 приведена последовательность графов G_i, i=1, 2, 3, 4, 5, получаемых в результате выполнения алгоритма.

При этом в силу того, что n(G)=5, G_5 - остовное дерево графа G.

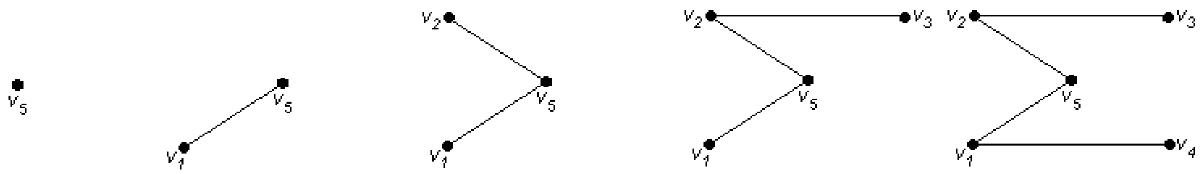


Рис. 40.

Замечание. Остовное дерево связного графа может быть выделено, вообще говоря, не единственным способом.

Общее число остовных деревьев связного графа может оказаться весьма большим. Например, для полного графа с n вершинами оно равно n^{n-2}.

В общем случае обозначим через G произвольный граф с n вершинами, m ребрами и k компонентами. Как уже было сказано, применяя выше описанный алгоритм, получим в результате граф, являющийся остовным лесом.

Определение. Число ребер, удаленных при построении остовного леса произвольного графа G, называется **цикломатическим числом** (или цикломатическим рангом) графа G и обозначается через γ (G) = m – n + k.

Пример 88.

Циклический ранг дерева равен нулю, а циклический ранг циклического графа равен единице.

Определение. **Коциклическим рангом** графа G называется число ребер в его остовном дереве.

С понятием остовного леса T графа G тесно связано понятие фундаментальной системы циклов, ассоциированной с T.

Определение. Если добавить к T любое не содержащиеся в нем ребро графа G, то получим единственный цикл; множество всех циклов, получаемых таким способом (т.е. путем добавления по отдельности каждого ребра из G, не содержащегося в T), называется **фундаментальной системой циклов, ассоциированной с T**.

3.3.3. Минимальные остовные деревья нагруженных графов

Пусть теперь каждому ребру x $\in$ X связного графа G=(V,X) с непустым множеством ребер X поставлена в соответствие величина l(x) – длина ребра x, т.е. граф G является нагруженным. Приведем алгоритм, позволяющий найти остовное дерево графа G с минимальной суммой длин содержащихся в нем ребер (по сравнению со всеми другими остовными деревьями графа G).

Определение. Остовное дерево связного нагруженного граф G с минимальной суммой длин содержащихся в нем ребер будем называть **минимальным остовным деревом** (МОД) графа G.

Алгоритм выделения МОД нагруженного связного графа G:

Шаг 1. Выберем в графе G ребро минимальной длины. Вместе с инцидентными ему вершинами оно образует подграф G_2 графа G. Положим i=2.

Шаг 2. Если i=n, где n=n(G), то задача решена, и G_i – искомое МОД графа G. В противном случае переходим к шагу 3.

Шаг 3. Строим граф G_{i+1}, добавляя к графу G_i новое ребро минимальной длины, выбранное среди всех ребер графа G, каждое из которых инцидентно к какой-нибудь вершине графа G_i и одновременно инцидентно какой-нибудь вершине графа G, не содержащейся в G_i. Вместе с этим ребром включаем в G_{i+1} и инцидентные ему вершины, не содержащуюся в G_i . Присваиваем i:=i+1 и переходим к шагу 2.

Пример 89.

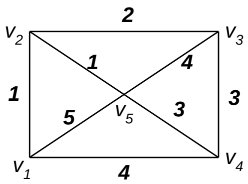


Рис. 41

Определить МОД нагруженного графа G, изображенного на рисунке 41, используя алгоритм.

Решение.

На рисунке 42 приведена последовательность графов G_i, i=2,3,4,5, получаемых в результате выполнения алгоритма. При этом в силу того, что

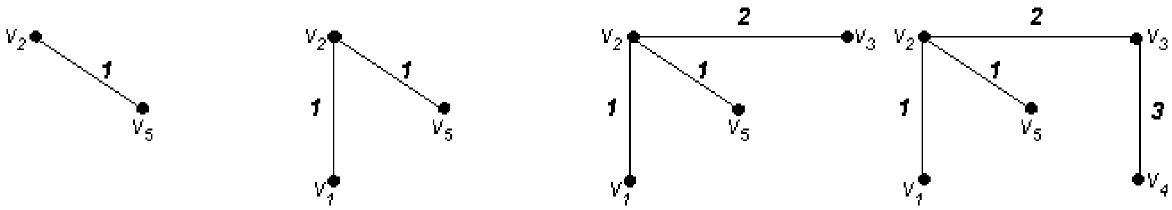


Рис.42.

n(G)=5, G_5 - МОД графа G. Причем, МОД(G) = 7.

Замечание. Возможно выделить в графе несколько остовных деревьев, каждое из которых будет являться минимальным, при этом величина МОД для всех этих деревьев будет равной.

Замечание. Для выделения МОД нагруженного псевдографа G следует предварительно удалить из G петли, из кратных ребер оставить лишь ребра минимальной длины.

Сети

Определение. **Транспортной сетью** называется орграф D = (V, X) с множеством вершин V, для которого выполняются условия:

1) существует одна и только одна вершина v_1, называемая **источником**, такая, что $D^{-1}(v_1) = 0$ (т.е. ни одна дуга не заходит в v_1);

2) существует одна и только одна вершина v_n, называемая **стоком**, такая, что $D(v_n) = 0$ (т.е. из v_n не исходит ни одной дуги);

3) каждой дуге $x \in X$ поставлено в соответствие целое число $c(x) \geq 0$, называемое **пропускной способностью дуги**.

Определение. Вершины в транспортной сети, отличные от источника и стока, называются **промежуточными**.

3.4.1. Поток в транспортной сети

Определение. Функция $\varphi(x)$, определенная на множестве X дуг транспортной сети D, и принимающая целочисленные значения, называется **допустимым потоком** (или просто потоком) в транспортной сети D, если:

1) для любой дуги $x \in X$ величина $\varphi(x)$, называемая **потоком по дуге** x, удовлетворяет условию $0 \leq \varphi(x) \leq c(x)$;

2) для любой промежуточной вершины v выполняется равенство

$$\sum_{\varpi \in D^{-1}(v)} \varphi(\omega, v) - \sum_{\varpi \in D(v)} \varphi(v, \omega) = 0$$

т.е. сумма потоков по дугам, заходящими в v, равна сумме потоков по дугам, исходящими из v.

Определение. **Величиной потока φ** в транспортной сети D называется величина φ, равная сумме потоков по всем дугам, заходящим в v_n, или, что то же самое, величина, равная сумме потоков по всем дугам, исходящим из v_1, т.е.

$$\overline{\varphi} = \sum_{v \in D^{-1}(v_n)} \varphi(v, v_n) = \sum_{v \in D(v_1)} \varphi(v_1, v)$$

Пусть φ - допустимый поток в транспортной сети D.

Определение. Дуга $x \in X$ называется **насыщенной**, если поток по ней равен ее пропускной способности, т.е. если $\varphi(x) = c(x)$. Поток φ называется **полным**, если любой путь в D из v_1 в v_n содержит, по крайней мере, одну насыщенную дугу.

Определение. Поток φ называется **максимальным**, если его величина принимает максимальное значение по сравнению с другими допустимыми потоками в транспортной сети D.

Очевидно, что максимальный поток φ обязательно является полным. Обратное неверно. Существуют полные потоки, не являющиеся максимальными. Тем не менее, полный поток можно рассматривать как некоторое приближение к максимальному потоку.

Алгоритм построения полного потока в транспортной сети D:

Шаг 1. Полагаем $\forall x \in X$ $\varphi(x) = 0$ (т.е. начинаем с нулевого потока). Кроме того, полагаем D\` = D.

Шаг 2. Удаляем из орграфа D\` все дуги, являющиеся насыщенными при потоке $\varphi$ в транспортной сети D. Полученный орграф снова обозначаем через D\`.

Шаг 3. Ищем в D\` простую цепь η из v_1 в v_n. Если такой цепи нет, то φ - искомый полный поток в транспортной сети D. В противном случае переходим к шагу 4.

Шаг 4. Увеличиваем поток $\varphi(x)$ по каждой дуге x из η на одинаковую величину **a** > 0 такую, что, по крайней мере, одна дуга из η оказывается насыщенной, а потоки по остальным дугам из η не превышают их пропускных способностей. При этом величина потока также увеличивается на **a**, а сам поток φ в транспортной сети D остается допустимым (поскольку в каждую промежуточную вершину, содержащуюся в η, дополнительно вошло **a** единиц потока и из нее вышло также **a** единиц потока). После этого переходим к шагу 2.

Пример 90.

Построить полный поток в транспортной сети, изображенной на рисунке 43.

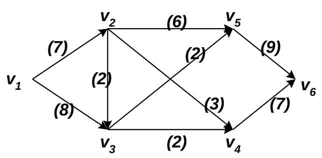


Рис. 43.

Решение.

Начинаем с нулевого потока (рис. 44). Пошагово выделяем простые цепи и увеличиваем поток по ним таким образом, чтобы хотя бы одна дуга в каждой из них стала насыщенной. Полученную насыщенную дугу помечаем пунктиром, помня, что по насыщенной дуге больше идти нельзя.

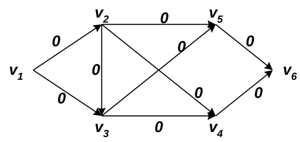


Рис. 44.

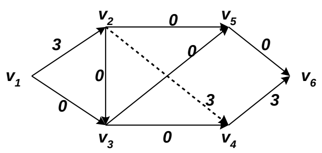


Рис. 45.

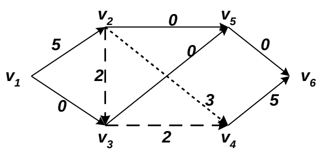


Рис. 46.

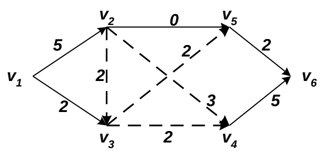


Рис. 47.

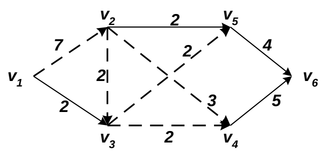


Рис. 48.

1. $\eta_1 = v_1v_2v_4v_6$, $a_1 = \min\{7, 3, 7\} = 3$ (рис. 45).

2. $\eta_2 = v_1v_2v_3v_4v_6$, $a_2 = \min\{7-3, 2, 2, 7-3\} = 2$ (рис. 46).

3. $\eta_3 = v_1v_3v_5v_6$, $a_3 = \min\{8, 2, 9\} = 2$ (рис. 47).

4. $\eta_4 = v_1v_2v_5v_6$, $a_4 = \min\{7-5, 6, 9-2\} = 2$ (рис. 48).

Заметим, что в полученной транспортной сети не существует пути из источника в сток, по которому возможно пройти. Следовательно, построенный поток в транспортной сети является полным и φ = 3+2+2+2=9.

3.4.2. Орграф приращений

Выделим для заданной транспортной сети D и допустимого потока φ в этой сети орграф приращений I(D, φ), имеющий те же вершины, что и сеть D. Каждой дуге x = (v, ω)∈X транспортной сети D в орграфе приращений I(D, φ) соответствуют две дуги: x и x` = (ω,v) - дуга, противоположная по направлению дуге x. Припишем дугам x = (v, ω)∈X, x` = (ω, v) орграфа приращений I(D, φ) длину l:

$$l(x) = \begin{cases} 0, \text{ если } \varphi(x) < c(x); \\ \infty, \text{ если } \varphi(x) = c(x). \end{cases}$$

$$l'(x) = \begin{cases} 0, \text{ если } \varphi(x) > 0; \\ \infty, \text{ если } \varphi(x) = 0. \end{cases}$$

т.е. орграф I(D, φ) является нагруженным. При этом, очевидно, что длинна любого пути из v_1 в v_n орграфе I(D, φ) равна либо 0, либо ∞.

Пример 91.

Построить орграф приращений для данной транспортной сети и построенного полного потока в этой сети.

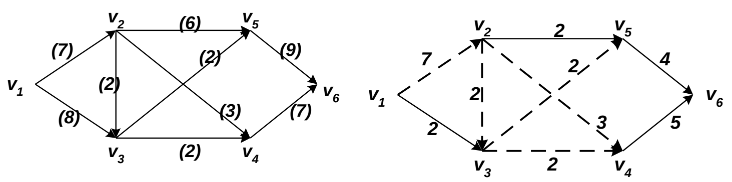


Рис. 49.

Решение.

Напомним, что орграф приращений имеет в два раза больше дуг, чем исходная транспортная сеть.

Для дуги прямой направленность вес равен 0, если дуга не является насыщенной, ∞ - в противном случае. Для дуги обратной направленности вес равен 0, если поток по ней не равен нулю, ∞ - в противном случае. На рисунке 50 изображен орграф приращений, соответствующий данному потоку в исходной транспортной сети.

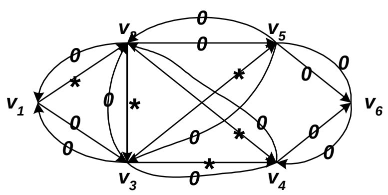


Рис. 50.

3.4.3. Алгоритм построения максимального потока в транспортной сети

Алгоритм построения максимального потока в транспортной сети D:

Шаг 1. Полагаем i = 0. Пусть φ_0 - любой допустимый поток в транспортной сети D. (например, полный; можно начинать с нулевого потока: $\varphi_0(x)$, $x \in X$).

Шаг 2. По сети D и потоку φ_i строим орграф приращений I(D, φ_i).

Шаг 3. Находим простую цепь η_i, являющуюся минимальным путем из v_1 в v_n в нагруженном орграфе I(D, φ_i) (например, используя алгоритм Форда - Беллмана). Если длина этой цепи равна ∞, то поток φ_i максимален, и работа алгоритма закончена. В противном случае увеличиваем поток вдоль цепи η_i на максимально допустимую величину $a_i > 0$, где $a_i \in Z$ (прибавляя ее для каждой дуги $x \in X$, через которую проходит цепь η_i, к уже имеющейся величине потока по дуге x, если направления x и η_i совпадают, и, вычитая, если направления x и η_i противоположны).

Пример 92.

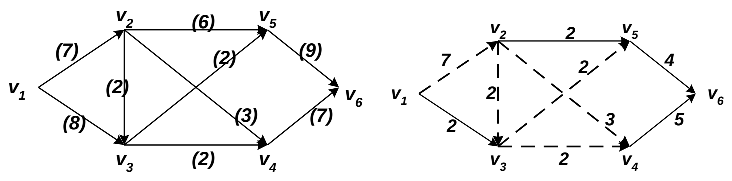


Рис. 51.

Выяснить является ли полный поток максимальным (рис. 51), если нет, то дополнить его до максимального.

Решение.

Для решения используем алгоритм Форда-Беллмана нахождения минимального пути в нагруженном орграфе.

Построим матрицу длин дуг C(D) и λ-матрицу (табл. 69).

Таблица 69

	1	2	3	4	5	6			$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
1														
2														
3														
4														

5													
6													

Поскольку $\lambda_6^{(5)} = 0$, то существует нулевой путь из источника v_1 в сток v_6. Значит, полный поток не является максимальным. Дополним его до максимального. Для этого найдем путь нулевой длины: $\lambda_6^{(5)} = 0 = \lambda_6^{(4)}$

Получаем, что $k_1 = 4$. Таким образом, минимальное число дуг в пути среди всех нулевых путей из v_1 в v_6 в орграфе приращений равняется 4. Определим теперь последовательность номеров i_1, i_2, i_3, i_4, i_5,где $i_1 = 6$.

Получаем, что в качестве такой последовательности надо взять номера 1, 3, 2, 5, 6, так как

$$\lambda_5^{(3)} + \tilde{n}_{5,6} = 0 + 0 = 0 = \lambda_6^{(4)};$$
$$\lambda_2^{(2)} + c_{2,5} = 0 + 0 = 0 = \lambda_5^{(3)};$$
$$\lambda_3^{(1)} + c_{3,2} = 0 + 0 = 0 = \lambda_2^{(2)};$$
$$\lambda_3^{(0)} + c_{1,3} = 0 + 0 = 0 = \lambda_3^{(1)}.$$

Тогда $v_1v_3v_2v_5v_6$ – искомый нулевой путь из v_1 в v_6 . Дуги, совпадающие по направлению с дугами исходной транспортной сети помечаем знаком «+», не совпадающие – знаком «-».

Получаем, $v_1 \underset{+}{\to} v_3 \underset{-}{\to} v_2 \underset{+}{\to} v_5 \underset{+}{\to} v_6$. Теперь необходимо найти величину, которую будем перемещать по полученному контуру. Для этого, каждому ребру в контуре поставим в соответствие число $\alpha(i, j)$, которое находим по следующему правилу: если направление ребра (i, j) в контуре совпадает с направлением ребра x в транспортной сети, то $\alpha(i, j)=c(x)-\varphi(x)$; если направление ребра в контуре не совпадает с направлением ребра в транспортной сети, то $\alpha(i, j)=\varphi(x)$. Итак, из чисел $\alpha(i, j)$ найдем минимальное:

min{8-2=6, 2, 6-2=4, 9-4=5}=2.

Перемещаем по контуру 2.

В результате получаем поток, изображенный на рисунке 52.

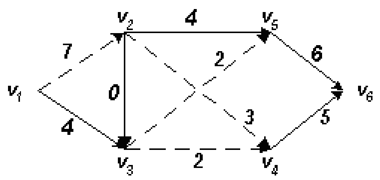


Рис. 52.

Заметим, что в полученной транспортной сети не существует пути из источника в сток, по которому возможно пройти. Следовательно, построенный поток в транспортной сети является полным и $\varphi = 11$. Проверим, является ли он максимальным.

Построим матрицу длин дуг C(D) и λ-матрицу (табл. 70).

Таблица 70

	1	2	3	4	5	6			$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
1														
2														
3														
4														
5														
6														

Так как $\lambda_6^{(5)} = \infty$, то нулевого пути из v_1 в v_6 не существует. Значит, поток $\overline{\varphi} = 11$ является максимальным.

6. Элементы булевой алгебры

Приложение. Лемма. $(a_1, a_2,\ldots, a_n) = (b_1, b_2,\ldots, b_n) \Leftrightarrow a_i = b_i,\ i = 1, 2,\ldots, n$

Доказательство. Так как $(a_1, a_2,\ldots, a_n) = ((a_1, a_2,\ldots, a_{n-1}), a_n)$, то лемму достаточно доказать для $n = 2$ и применить индукцию по n.

Необходимость: $(a, b) = (c, d) \Rightarrow a = c$ и $b = d$.

a) $a = b$;

так как $(a, a) = \{\{a\}, \{a, a\}\} = \{\{a\}\}$, то из $(a, a) = \{\{a\}\} = (b, c) = \{\{b\}, \{b, c\}\}$ следует

$a = b = c$;

аналогично, из $c = d$ следует $a = b$;

b) $a \neq b$ и $c \neq d$;

по определению упорядоченной пары:

$(a, b) = (c, d) \Rightarrow \{\{a\}, \{a, b\}\} = \{\{c\}, \{c, d\}\}$;

по определению равенства и включения множеств:

$$\{\{a\}, \{a, b\}\} = \{\{c\}, \{c, d\}\} \Rightarrow \{a\} \in \{\{c\}, \{c, d\}\} \text{ и } \{a, b\} \in \{\{c\}, \{c, d\}\};$$

по определению принадлежности элемента множеству:

$$\{a\} \in \{\{c\}, \{c, d\}\} \Rightarrow \{a\} = \{c\} \text{ или } \{a\} = \{c, d\};$$

так как второе равенство возможно лишь при $a = c = d$, что невозможно, так как во множестве все элементы попарно различны, то $a = c$;

если $a = c$ и $\{a, b\} \in \{\{c\}, \{c, d\}\}$, то $\{a, b\} \in \{\{a\}, \{a, d\}\}$;

так как $a \neq b$, то из последней принадлежности следует, что $\{a, b\} = \{a, d\}$, следовательно, $b = d$.

Достаточность: для $n = 2$ следует из определения равенства множеств.

Литература

а) основная литература:

1. Асанов М.О. Дискретная математика: графы, матроиды, алгоритмы [Электронный ресурс] : учебное пособие / М.О. Асанов, В.А. Баранский, В.В. Расин. — СПб.: Лань, 2010. — 363 с. ЭБС Лань http://e.lanbook.com/books/element.php?pl1_id=536
2. Бондаренко Л. Н. Дискретная математика. Методические указания для выполнения лабораторных работ. – Пенза: Изд-во Пензенского государственного университета, 2007. – 98 с.
3. Гаврилов Г. П., Сапоженко А.А. Задачи и упражнения по дискретной математике: учеб. пособие. – М.: Физматлит, 2005. – 416 с.
4. Копылов В. И. Курс дискретной математики: учебное пособие. — СПб.: Лань, 2011. — 207 с. ЭБС Лань http://e.lanbook.com/books/element.php?pl1_id=1798
5. Кузнецов О.П. Дискретная математика для инженера. — СПб.: Лань, 2009. — 396 с. ЭБС Лань http://e.lanbook.com/books/element.php?pl1_id=220
6. Мальцев И.А. Дискретная математика. — СПб.: Лань, 2011. — 304 с. ЭБС Лань http://e.lanbook.com/books/element.php?pl1_id=638
7. Новиков Ф. А. Дискретная математика для программистов: учебник для вузов. – 3-е изд. – СПб.: Питер, 2009. — 384 с.
8. Соболева Т.С., Чечкин А.В. Дискретная математика. — М.: Академия, 2014. — 256 с. ЭБС Academia, 2014. — 256 с. http://www.academia-moscow.ru/catalogue/4831/106190/
9. Шевелев Ю.П. Дискретная математика. — СПб.: Лань, 2008. — 592 с. ЭБС Лань http://e.lanbook.com/books/element.php?pl1_id=437
10. Яблонский С. В. Введение в дискретную математику: учебное пособие. – - 5-е изд., стереотип. – М.: Высш. шк., 2008. – 384 с.

б) дополнительная литература:

1. Белоусов, А. И. Дискретная математика: учебник / А. И. Белоусов, С. Б. Ткачев. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2006. – 744 с.
2. Лекции по дискретной математике / Ю. В. Капитонова и др. – СПб: БХВ – Петербург, 2004. – 642 с.
3. Новиков, Ф. А. Дискретная математика для программистов / Ф. А. Новиков. – СПб.: Питер, 2002. – 304 с.
4. Харари, Ф. Теория графов. – М.: КомКнига, 2006. – 296 с.
5. Берж, К. Теория графов и ее применения / К. Берж. – М.: Изд. иностр. литер., 1962. – 320 с.

6. Грэхем, Р. Конкретная математика. Основание информатики / Р. Грэхем, Д. Кнут, О. Паташник. – М.: Мир, 1998. – 703 с.
7. Кирсанов, М. Н. Графы в Maple / М. Н. Кирсанов. – М.: Физматлит, 2007, 168 с.
8. Лекции по теории графов / В. А. Емеличев, О. И. Мельников, В. И. Сарванов, Р. И. Тышкевич – М.: Наука, 1990. – 384 с.
9. Ловас, Л. Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии / Л. Ловас, М. Пламмер. – М.: Мир, 1998. – 653 с.
10. Оре, О. Теория графов / О. Оре. – М.: Наука, 1968. – 352 с.
11. Стенли, Р. Перечислительная комбинаторика / Р. Стенли. – М.: Мир, 1990. – 440 с.
12. Стенли, Р. Перечислительная комбинаторика / Р. Стенли. Т. 2. – М.: Мир, 2009. – 768 с.
13. Татт, У. Теория графов / У. Татт. – М.: Мир, 1988. – 424 с.
14. Хаггарти, Р. Дискретная математика для программистов / Р. Хаггарти. – М. Техносфера, 2012. – 400 с.
15. Ху, Т. Ч. Комбинаторные алгоритмы / Т. Ч. Ху, М. Т. Шинг. – Нижний Новгород: Издво Нижегородского ун-та им. Н. И. Лобачевского, 2004. – 330 с.

в) программное обеспечение и Интернет-ресурсы:

В Интернете имеется огромное число ресурсов по дисциплине «Дискретная математика». Выделим электронный справочник

http://oeis.org/ – Sloane N. J. A. The on-line encyclopedia of integer sequences, содержащий более четверти миллиона статей по последовательностям, используемым в комбинаторике и теории графов, а также форум по Maple www.exponenta.ru.

Оглавление

1. Множества, мультимножества. Элементы комбинаторики

Мультимножества..

Элементы комбинаторики. Числа Фибоначчи и числа Каталана. Их свойства

2. Отношения. Функции.

Отношения, их представления в ЭВМ и свойства. Композиция отношений

Отношения эквивалентности и порядка, их свойства. Представление отношений матрицами

Функции и их представления в ЭВМ

3. Основные понятия теории графов.

Основные понятия и терминология теории графов. Простейшие виды графов

Матрицы смежности и инцидентности, их основные свойства

Подграфы, изоморфизм графов и подграфов. Операции над графами

4. Маршруты, цепи, циклы. Эйлеровы и гамильтоновы графы. Деревья и их свойства.

Маршруты, цепи, циклы. Компоненты, связность. Цикломатическое число и его свойства

Эйлеровы и гамильтоновы графы. Деревья. Матрица Кирхгофа и ее свойства

Перечисление деревьев

5. Алгоритмы на графах. Алгоритмы Дейкстры, Краскала и Прима

Прикладные задачи на графах. Алгоритмы на графах

Кратчайшие пути на графах. Алгоритм Дейкстры

Жадные алгоритмы. Задача о кратчайшем остове. Алгоритмы Краскала и Прима

6. Прикладные задачи на графах. Планарность. Раскраски графа

Планарность. Формула Эйлера. Критерии планарности

Раскраски графа. Хроматическое число, индекс и их свойства