ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Б1.Б.07 Дискретная математика и математическая логика

Направление подготовки (специальность) 09.03.01 Информатика и вычислительная техника

Профиль образовательной программы “Автоматизированные системы обработки информации и управления”

Форма обучения очная

СОДЕРЖАНИЕ

1. КОНСПЕКТ ЛЕКЦИЙ

1.1 Лекция № 1, 2, 3 (6 часов).

Тема: «Множества и бинарные отношения»

1.1.1 Вопросы лекции:

1. Множества и операции над ними. Диаграммы Венна-Эйлера.

2. Элементы алгебры множеств.

3. Бинарные отношения и их свойства.

1.1.2 Краткое содержание вопросов:

1. Множества и операции над ними. Диаграммы Венна-Эйлера.

1.Понятия множества, элемента множества, обозначения множества и его элементов, примеры.

2.Способы задания (описания) множеств перечислением элементов и с помощью предикатов.

3.Стандартные множества, их названия и обозначения

$\varnothing$ - пустое множество,

$N = \{1, 2, 3, ...\}$ - множество натуральных чисел(натуральный ряд);

$Z = \{0, \pm 1, \pm 2, \pm 3, ...\}$ - множество целых чисел;

$Q = \left\{ \frac{p}{q}, p, q \in Z, q \neq 0 \right\}$ - множество рациональных чисел;

R - множество всех вещественных чисел (всех десятичных дробей);

числовые промежутки $\langle a, b \rangle$.

4. Иллюстрация множеств диаграммами Венна-Эйлера.

5.Отношения и операции с множествами, их иллюстрация диаграммами Венна-Эйлера:

-равенство множеств $A = B$,

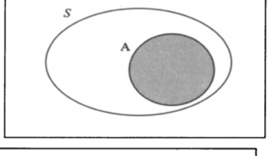
-включение $A \subset B, A \subseteq B$, понятие подмножества,

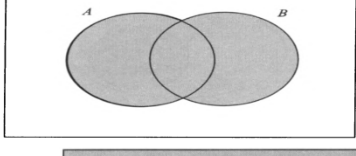
-объединение множеств $A \cup B$,

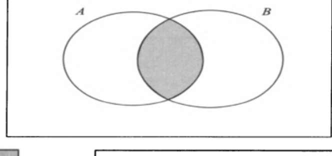
-пересечение множеств $A \cap B$,

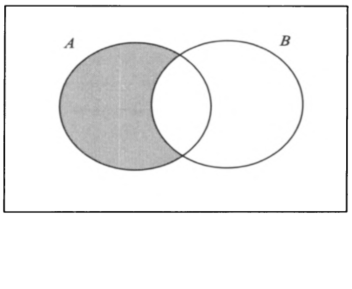
-разность множеств $A \setminus B$ и дополнение $C_A B$ множества B до множества A, универсальное множество U, дополнение множества A до универсального $\overline{A}$

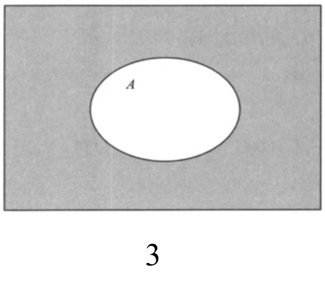
-симметрическая разность $A \Delta B$.

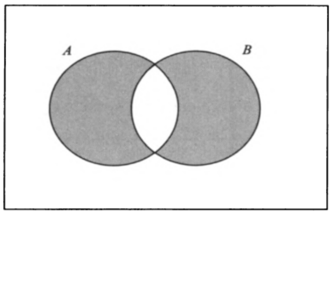












Дискретная математика представляет собой область математики, в которой изучаются свойства структур конечного характера, а также бесконечных структур, предполагающих скачкообразность происходящих в них процессов или отделимость составляющих их элементов. В отличие от дискретной математики *классическая* математика занимается изучением свойств структур непрерывного характера. Это деление достаточно условно, поскольку средства дискретной математики используются для изучения непрерывных моделей и наоборот.

Бурное развитие дискретной математики обусловлено прогрессом компьютерной техники, необходимостью создания средств обработки и передачи информации, а также представления различных моделей на компьютерах, которые по своей природе являются структурами конечным

Рассматриваются м*ножества и их спецификация, элементы и множества.* Множество, не содержащее элементов, называется пустым множеством и обозначается символом $\varnothing$. Если все рассматриваемые множества (в конкретной задаче) являются подмножествами более широкого множества U, то множество U называется универсальным множеством, или универсумом.

Мощность множества M обозначается как |M| и для конечного множества равняется числу элементов в нем.

Заметим, что $|\varnothing| = 0$, но $|\{\varnothing\}| = 1$.

2. Элементы алгебры множеств.

Теоретико-множественные соотношения (равенства множеств, включения) и методы их вывода и доказательства. Такие соотношения выражают законы алгебры множеств.

Свойства операций над множествами.

Законы ассоциативности

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

Законы коммутативности

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

Законы тождества

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

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

Законы идемпотентности

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

Законы дистрибутивности

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

Законы дополнения

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

$$\overline{U} = \varnothing \qquad \overline{\varnothing} = U$$

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

Законы де Моргана

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

Взаимосвязь законов алгебры множеств и алгебры логики, понятие о модели аксиоматической теории и интерпретации, понятие об алгебре Буля.

Булеан множества и его нахождение.

На основании этих свойств можно получить новые свойства и равенства.

Принцип двойственности.

Принцип двойственности состоит в том, что из любого равенства, относящегося к системе подмножеств фиксированного множества U, автоматически может быть получено другое, двойственное, равенство, путем замены всех рассматриваемых множеств их дополнениями, объединений множеств – пересечениями, пересечений множеств – объединениями.

3. Бинарные отношения и их свойства.

Прямое произведение множеств

Упорядоченная последовательность, содержащая n элементов некоторого множества, называется *n-кой*, или *набором из n элементов*. Обычно n-ка, образованная последовательностью $a_1, a_2, \ldots, a_n$ обозначается $(a_1, a_2, \ldots, a_n)$. При малых n говорят о двойках элементов, тройках и т.д.

Для множества чисел А = {1, 2, 3, 4} можно рассмотреть тройки: (1, 2, 2), (3, 4, 1), (2, 1, 2), причем первая и последняя тройки различны, несмотря на их одинаковый состав.

Прямым (или *декартовым*) *произведением* множеств $A_1, A_2, \ldots, A_n$ называется множество всех упорядоченных наборов $(x_1, x_2, \ldots x_n)$ таких, что $x_i \in A_i$ при $\forall i = 1, 2, \ldots, n$. Декартово произведение обозначается $A_1 \times A_2 \times \ldots \times A_n$. Если одним из сомножителей является пустое множество, то и произведение является пустым множеством.

Степенью множества А называется его прямое произведение само на себя n раз; обозначается A^n.

N-местным отношением R или *N-местным предикатом* R на множествах $A_1, \ldots, A_n$ называется любое подмножество прямого произведения $A_1 \times \ldots \times A_n$: $R \subseteq A_1 \times \ldots \times A_n$. Элементы $a_1, a_2, \ldots, a_n \mid a_i \in A_i$ при $\forall i = 1, 2, \ldots, n$ *связаны отношением* R тогда и только тогда, когда упорядоченный набор $(a_1, a_2, \ldots a_n) \in R$. При $N = 1$ отношение R является подмножеством множества A_1 и называется *унарным отношением* или *свойством*.

Наиболее часто встречается двухместное отношение ($N = 2$), которое называется *бинарным отношением* R из множества A в множество B, или *соответствием*: это подмножество произведения множеств A и B: $R \subseteq A \times B$. Если элементы a и b множеств A и B $(a,b) \in R$, то говорят, что они *находятся в отношении* R, для чего часто используется т.н. инфиксная форма записи: aRb. Если $R \subseteq A \times A$ (т.е. $A=B$), то R называется *бинарным отношением на множестве* A. Соответственно, отношение $R \subseteq A^n$ называется *N-местным предикатом на множестве* A.

Бинарное отношение можно задать *указанием всех пар*, для которых это отношение выполняется, или *графически*. Способы графического представления также могут быть различными.

1.2 Лекция № 4 (2 часа).

Тема: «Основные алгебраические структуры»

1.2.1 Вопросы лекции:

1. Понятие бинарного отношения. Способы задания отношений. Свойства отношений, классификация отношений.

2. Отношения эквивалентности и порядка. Отношения Парето. Принятие решений при многих критериях. Функции. Виды функций. Переключательные функции (ПФ).

1.2.2 Краткое содержание вопросов:

1. Понятие бинарного отношения. Способы задания отношений. Свойства отношений, классификация отношений.

Свойства отношений

Теорема: *Для любых бинарных отношений P, Q, R выполняются следующие свойства:*

1. $(P^{-1})^{-1}=P$; *2.* $(P\circ Q)^{-1}=Q^{-1}\circ P^{-1}$;

3. $(P\circ Q)\circ R=P\circ(Q\circ R)$ *(ассоциативность композиции).*

Бинарное отношение R на множестве A называется *рефлексивным*, если для любого его элемента a выполняется aRa: $\forall a \in A \;\; aRa$.

Бинарное отношение R на множестве A называется *антирефлексивным*, если для любых его элементов $a,b \;\; aRb \Rightarrow a \neq b$.

Бинарное отношение R на множестве A называется *симметричным*, если из его выполнения для a, b следует выполнение для b, a: $\forall\, a, b \in A \;\; aRb \Rightarrow bRa$.

Бинарное отношение R на множестве A называется *антисимметричным*, если из его выполнения для a, b и b, a следует, что a и b совпадают. $\forall\, a, b \in A \;\; aRb$ и $bRa \Rightarrow a = b$.

Бинарное отношение R на множестве A называется *транзитивным*, если из его выполнения для a,b и для b,c следует его выполнение для a,c: $\forall\, a, b, c \in A \;\; aRb$ и $bRc \Rightarrow aRc$.

Бинарное отношение R на множестве A называется *полным*, или *линейным*, если для любых двух различных элементов множества A оно выполняется или для a, b, или для b, a: $\forall\, a, b \in A \mid a \neq b \Rightarrow aRb$ или bRa

Рассмотрим отношение R на множестве натуральных чисел следующим образом: $R = \{(x,y) \mid x$ – делитель $y\}$. Это отношение является <u>рефлексивным</u>, т.к. $x/x = 1$ $\forall x \in \mathbf{N}$. Отношение R <u>антисимметрично</u>, т.к. если $x/y \in \mathbf{N}$ и $y/x \in \mathbf{N}$, то $x = y$. Проверим <u>транзитивность</u> R. $y/x \in \mathbf{N}$ и $z/y \in \mathbf{N} \Rightarrow z/x = z/y \cdot y/x \in \mathbf{N}$.

<u>Теорема (о проверке свойств отношения)</u>:

Отношение R на множестве A^2*:*

-R рефлексивно $\Leftrightarrow I \subset R$;

-R симметрично $\Leftrightarrow R = R^{-1}$;

-R транзитивно $\Leftrightarrow R\circ R \subset R$;

-R антисимметрично $\Leftrightarrow R \cap R^{-1} \subset I$;

-R полно $\Leftrightarrow R \cup I \cup R^{-1} = U$;

Представление отношений в ЭВМ

Удобным способом представления отношений в ЭВМ является *матричная форма*. Рассмотрим два конечных множества $A=\{a_1,a_2,\dots,a_m\}$, $B=\{b_1,b_2,\dots,b_n\}$ и бинарное отношение $P \subseteq A\times B$. Определим матрицу $[P] = (p_{ij})$ бинарного отношения P по следующему правилу: $p_{ij} = \begin{cases} 1, & если\ (a_i,b_j) \in P, \\ 0, & если\ (a_i,b_j) \notin P. \end{cases}$ Полученная матрица содержит полную информацию о связях между элементами и позволяет представлять эту информацию на компьютере.

Заметим, что любая матрица, состоящая из нулей и единиц, является матрицей некоторого бинарного отношения.

Основные свойства матриц бинарных отношений:

1. Если бинарные отношения $P,Q \subseteq A\times B$, $[P] =(p_{ij})$, $[Q] = (q_{ij})$, то $[P\cup Q] = (p_{ij}+q_{ij})$, $[P\cap Q] = (p_{ij}\cdot q_{ij})$, где умножение осуществляется обычным образом, а сложение – по логическим формулам (т.е. 0+0=0, во всех остальных случаях 1). Итак: $\mathbf{[P\cup Q] = [P]+[Q]}$, $\mathbf{[P\cap Q] = [P]*[Q]}$.

2. Если бинарные отношения $P \subseteq A\times B$, $Q \subseteq B\times C$, то $[P\circ Q] = [P]\cdot[Q]$, где умножение матриц [P] и [Q] осуществляется по обычному правилу, а произведение и сумма элементов из [P] и [Q] – по правилам пункта 1.

3. Матрица обратного отношения P^{-1} равна транспонированной матрице отношения P: $\mathbf{[P^{-1}]=[P]^T}$.

4. Если $P\subseteq Q$, $[P]=(p_{ij})$, $[Q]=(q_{ij})$, то $p_{ij} \le q_{ij}$. $\forall i,j$.

5. Матрица тождественного отношения единична: $[I_A] = (I_{ij})$: $I_{ij} =1 \Leftrightarrow i=j$.

6. Пусть R – бинарное отношение на A^2. Отношение R называется *рефлексивным*, если $\forall x\in A$ $(x,x)\in R$, т.е. $I_A\in R$ (на главной диагонали R стоят единицы). Отношение R называется *симметричным*, если $\forall x,y\in A$ $(x,y)\in R \Rightarrow (y,x)\in R$, т.е. $R^{-1}=R$, или $[R]=[R]^T$ (матрица симметрична относительно главной диагонали). Отношение R называется *антисимметричным*, если $R \cap R^{-1} \subseteq I_A$, т.е. в матрице $[R \cap R^{-1}]=[R]*[R]^T$ вне главной диагонали все элементы равны 0. Отношение R наз. *транзитивным*, если $(x,y)\in R$, $(y,z)\in R \Rightarrow (x,z)\in R$, т.е. $R\circ R\subseteq R$.

2. Отношения эквивалентности и порядка. Отношения Парето. Принятие решений при многих критериях. Функции. Виды функций. Переключательные функции (ПФ).

Отношение эквивалентности

Бинарное отношение R на множестве A называется *отношением эквивалентности*, если оно является рефлексивным, симметричным и транзитивным. Обычно отношение эквивалентности обозначают через $\equiv$ или $\sim$. Пусть R – отношение эквивалентности на множестве A. Определим *класс эквивалентности* $[x]$ для $x\in A$: $[x] = \{y \mid xRy\}$, т.е. это множество всех элементов A, которые R-эквивалентны x.

Пусть E – эквивалентность на множестве M. Тогда семейство классов эквивалентности множества M называется *фактор-множеством* множества M по отношению E и обозначается $M/E=\{E(x)|\ x\in M\}$. *Утверждение*: всякое отношение эквивалентности на множестве M определяет разбиение множества M, причем среди элементов разбиения нет пустых; и обратно, всякое разбиение множества M, не содержащее пустых элементов, определяет отношение эквивалентности на множестве M: $\equiv\subset M^2 \Leftrightarrow \exists\, \beta=\{B_i \mid B_i\subset M,\ B_i\neq\varnothing,$ $M=\cup B_i$ и $\forall\ i,j\ \ i\neq j \Rightarrow B_i\cap B_j=\varnothing$

Отношение порядка.

Бинарное отношение R на множестве A называется *отношением порядка*, если оно антисимметрично и транзитивно. Отношение порядка может быть рефлексивным, и тогда оно называется отношением *нестрогого порядка* (обычно обозначается $\le$). Если отношение порядка антирефлексивно, то оно называется отношением *строгого порядка* и обозначается обычно <. Отношение порядка может быть полным (линейным), и тогда оно называется отношением линейного порядка (если любые два элемента сравнимы между собой), а множество – вполне упорядоченным. Если отношение порядка не обладает свойством полноты, то оно называется отношением *частичного порядка*, а множество с заданным на нем отношением частичного порядка называется *частично упорядоченным множеством*. Обычно отношение порядка в общем случае обозначают <, и вместо aRb или $(a,b)\in R$ пишут $a<b$. Для отношения < обратным является >.

- Отношение < на множестве чисел является отношением строгого полного порядка, отношение $\le$ – нестрогого полного порядка. Следовательно, множество чисел является линейно упорядоченным. Отношение $\subset$ на булеане $P(M)$ является отношением нестрого частичного порядка.

Пусть дано ч.у.м. M с отношением порядка $\le$: Ũ= $\{M,\le\}$. М*аксимальный* и минимальный элементы, *наибольший* и *наименьший*. Наибольший (наименьший) элемент обычно называют *единицей*, а наименьший – *нулем* множества. Заметим, что всякий наибольший элемент (если он существует) является максимальным, а всякий наименьший – минимальным. Обратное утверждение неверно. Максимальных (минимальных) элементов может быть несколько; *верхняя грань множества*, *точная верхняя грань* **sup A**, *нижняя грань, точная нижняя грань* **inf A.**

Утверждение: Во всяком конечном непустом частично упорядоченном множестве существует минимальный элемент.

Замкнутость множества означает, что многократное повторение допустимых шагов не выводит за пределы этого множества.

Пусть R и R' – отношения на множестве M. Тогда отношение R' называется *замыканием отношения R относительно свойства C*, если:

-R' обладает свойством С: $C(R')$;

-R' является надмножеством R: $R\subset R'$;

-R' является наименьшим: $C(R'')$, $R\subset R'' \Rightarrow R'\subset R''$.

Пусть А – вполне упорядоченное множество с отношением порядка $\le$. Введем отношение $\le$ на множестве упорядоченных наборов из А следующим образом:

$$(a_1,\dots,a_m)\le(b_1,\dots,b_n) \Leftrightarrow \quad m\le n \ \text{ и } \forall i=1,..,m \quad a_i=b_i$$

или

$\exists\, k\le \min(n,m) \mid a_k\le b_k$ и $a_i=b_i\ \forall i<k$,

т.е. первые элементы совпадают, а k-й меньше.

Такое отношение называется *лексикографическим*, или *алфавитным* порядком.

1.3 Лекция № 5, 6 (4 часа).

Тема: «Элементы теории чисел»

1.3.1 Вопросы лекции:

1. Биекции. Эквивалентные множества.

2. Мощность множеств, счётные множества. Мощность континуума.

1.3.2 Краткое содержание вопросов:

1. Биекции. Эквивалентные множества.

Функции. Определение функции.

Бинарное отношение R между множествами A и B называется *однозначным*, если из его выполнения для a, b и a, c ($a \in A$, $b, c \in B$) следует, что b и c совпадают. $\forall a \in A$, $b, c \in B$ aRb и $aRc \Rightarrow b = c$ (одному элементу множества A не могут соответствовать разные элементы, находящиеся с ним в отношении R).

Однозначное отношение f между множествами A и B, заданное для каждого элемента множества A, называется *отображением* множества A в множество B, или *функцией из A в B*: $f: A \to B$.

Дадим формальное определение. Отношение f между элементами множеств A и B называется *функцией* из A в B и обозначается $f: A \to B$, если оно обладает следующими двумя свойствами:

а) $\forall x \in A \ \exists y \in B \mid (x,y) \in f$; б) если $(x, y) \in f$ и $(x, z) \in f \Rightarrow y = z$.

Для функции f обычно вместо записи $(x,y) \in f$ используется т.н. префиксная форма: $y = f(x)$. При этом x называется *аргументом*, а y – *значением функции* f.

Для $f: A \to B$ *область определения* $Dom(f) \equiv \{x \in A \mid \exists y \in B \mid y = f(x)\}$, *область значений* $Codom(f) \equiv \{y \in B \mid \exists x \in A \mid y = f(x)\}$.

Если $Dom(f) = A$, то функция называется *тотальной*, а если $Dom(f) \neq A$ – *частичной*. *Сужением* функции $f: A \to B$ на множество $M \subset A$ называется функция $f|_M$, определяемая следующим образом: $f|_M \equiv \{(x,y) \mid y = f(x), x \in M\}$.

Для тотальной функции ее сужение на множество $Dom(f)$ совпадает с самой функцией f.

Для $f: A \to B$ и $x \in A$: если $y = f(x)$, то y называется *образом элемента* x, а x – *прообразом элемента* y. Для любого непустого подмножества $C \subset A$ его образом относительно f называется множество $f(C) = \{f(x) \mid x \in C\}$.

Функция $f: A_1 \times A_2 \times \ldots \times A_n \to B$ называется функцией n аргументов, или *n-местной* функцией.

Классификация функций

Отображение $f: A \to B$ называется (см. рисунок ниже):

инъективным (инъекцией), если любым различным значениям аргумента соответствуют различные значения функции: $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$;

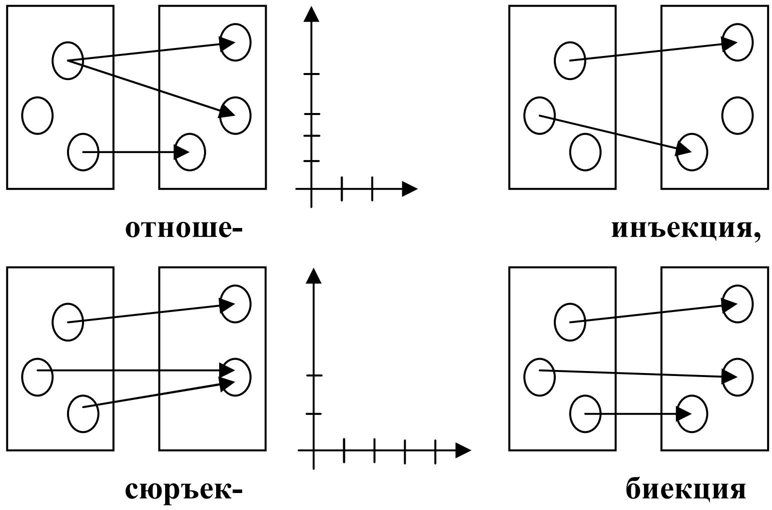
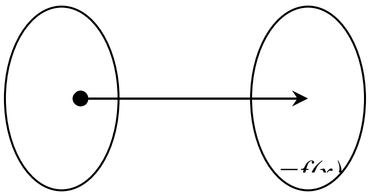


Рис. 1.4 Графическое пред-

сюръективным, сюръекцией, или *отображением* ***на***, если любому элементу y множества B соответствует элемент x множества A, такой, что $f(x) = y$: $\forall y \in B \;\; \exists\, x \in A \mid f(x) = y$;



биективным, биекцией, или *взаимно однозначным соответствием*, если оно является одновременно инъекцией и сюръекцией;

перестановкой множества A, если $A = B$ и функция $f : A \to A$ является взаимно однозначным соответствием.

Если функция $I : A \to A$ определена как $I(a)=a \;\; \forall a \in A$, то I называется *тождественной функцией* на множестве A.

Обратное отношение f^{-1}, которое определялось ранее, может не быть функцией, даже если f является функцией из A в B. Если обратное отношение f^{-1} является функцией, то ее называют *обращением функции*, или *обратной функцией*.

Теорема (об обратной функции):

Если функция $f : A \to B$ является биекцией, то обратное отношение f^{-1} также является функцией из B в A, причем биекцией. Обратно, если f^{-1} – функция из B в A, то f является биекцией.

Теорема: *Если функция $f : A \to B$ является биекцией, то:*

а) $\forall b \in B \; f(f^{-1}(b))=b$, *б)* $\forall a \in A \; f^{-1}(f(a))=a$.

Теорема: *Если функция $f : A \to A$ и I – тождественная функция на A, то $I \circ f = f \circ I = f$. Если для f существует обратная функция, то $f^{-1} \circ f = f \circ f^{-1} = I$.*

Ядро функции обозначается $\ker f = f \circ f^{-1}$.

Утверждение: ядро функции является отношением эквивалентности на области определения функции.

Теорема: *Пусть функции $g : A \to B$ и $f : B \to C$. Тогда: Если g и f – сюръекции, то их композиция – сюръекция; Если g и f – инъекции, то их композиция – инъекция; Если g и f –биекции, то их композиция – биекция;*

Некоторые специальные функции

1) *Перестановка* множества A была определена ранее.

2) *Тождественная* функция была определена ранее.

3) Пусть задано некоторое множество $M \subset U$. *Характеристической функцией* этого множества является функция χ, равная 1 на элементах множества M:
$$\chi(x) = \begin{cases} 1, & \text{если} \quad x \in M \\ 0, & \text{если} \quad x \notin M \end{cases}$$

4) *Бинарной операцией* на множестве A называется функция $b : A \times A \to A$. Образ пары (x,y) при отображении b записывается как $b(x,y)$ или как xby. Поскольку область значений бинарной операции на A по определению есть подмножество A, то множество A обладает свойством замкнутости относительно бинарной операции.

5)*Конечной последовательностью* называется функция из $N_0 = \{0, 1, 2, 3, \dots, n\}$ в некоторое множество A. $f : N_0 \to A$ *Бесконечной последовательностью* называется функция из $\{0, 1, 2, 3, \dots\}$ в некоторое множество A. Элементом последовательности является упорядоченная пара (n,a), в которой $a = f(n)$. Обычно эта пара обозначается через a_n, а последовательность $f : N_0 \to A$ – через $\{a_n\}$.

Иногда нумерацию членов последовательности начинают с 1, т.е. иногда последовательностью называют функцию, определенную на множестве **N**.

Широко известными видами последовательностей являются арифметическая и геометрическая прогрессии.

6)Еще одна известная специальная функция, которая далее потребуется при комбинаторных вычислениях – факториал. На примере этой функции уместно вспомнить о принципе математической индукции.

2. Мощность множеств, счётные множества. Мощность континуума.

Биекция и эквивалентные множества. Понятие мощности конечного множества, принцип Дирихле. Свойства эквивалентных множеств.

Мощность множеств, счётные множества. Мощность континуума. Мощность бесконечного множества, счётные множества, множества мощности континуум.

1.4 Лекция № 7, 8 (4 часа).

Тема: «Основы комбинаторики»

1.4.1 Вопросы лекции:

1. Правила комбинаторики.
2. Комбинаторные формулы

1.4.2 Краткое содержание вопросов:

1. Правила комбинаторики.

Комбинаторика – раздел математики, посвященный решению задач выбора и расположения элементов некоторого обычного множества в соответствии с заданными правилами. Каждое такое правило определяет способ построения некоторой конструкции из элементов исходного множества, называемой *комбинаторной конфигурацией*. Простейшими примерами комбинаторных конструкций являются перестановки, размещения, сочетания и разбиения, рассматриваемые ниже. Вычисления на дискретных математических структурах – комбинаторные вычисления – требуют комбинаторного анализа для установления свойств и оценки применимости алгоритмов.

Комбинаторные задачи и основные принципы*. **Во многих практических задачах возникает необходимость подсчитать количество возможных комбинаций объектов, удовлетворяющих определенным. условиям. Такие задачи называются комбинаторными. Среди всего многообразия таких задач есть ряд наиболее часто встречающихся, для которых известны способы подсчета. Для формулировки и решения комбинаторных задач используются различные модели комбинаторных конфигураций. Рассмотрим две наиболее популярные.***

Дано *n* предметов. Их нужно разместить по *m* ящикам так, чтобы выполнялись заданные ограничения. Сколькими способами это можно сделать?

1. Дано множество функций $F : X \to Y$, где $|X| = n$, $|Y| = m$, $X = \{1,2,\ldots,n\}$ (предметы – элементы множества X – перенумерованы, т.е. можно считать номер отличительным признаком предмета). Без ограничения общности можно считать, что элементы множества Y также перенумерованы: $Y = \{1,2,\ldots,m\}$, $F = [F(1),\ldots,F(n)]$, $1 \le F(i) \le m$. Сколько существует функций, удовлетворяющих заданным ограничениям?

Наиболее часто соответствие конфигураций 1-го и второго типа очевидно, поэтому анализ проблем и вывод формул можно проводить на любом языке.

Основные комбинаторные принципы

***Утверждение*:** Если множества A и B не пересекаются и содержат по *m* и *n* элементов соответственно, то множество $A \cup B$ содержит $m + n$ элементов: для множеств A и B | $A \cap B = \varnothing$: $|A \cup B| = |A| + |B|$.

Теорема(о мощности произведения конечных множеств): Для любых множеств A и B $|A \times B| = |A| \cdot |B|$.

Правило суммы (комбинаторный принцип сложения): Если объект $\alpha \in A$ можно выбрать *m* способами, а объект $\beta \in B$, отличный от α, *n* способами, причем α и β нельзя выбрать одновременно, то осуществить выбор «либо α, либо β» можно $m+n$ способами.

Пусть в киоске имеется 5 различных книг по математике и 7 – по физике. Если студент может купить только одну книгу, то у него есть 5 вариантов выбора первой книги и 7 вариантов – второй, т.е. 12 вариантов.

Правило произведения (комбинаторный принцип умножения) Если объект $\alpha \in A$ можно выбрать *m* способами, а после каждого такого выбора можно выбрать *n* способами объект $\beta \in B$, отличный от α, то выбор обоих объектов α и β в указанном порядке можно осуществить $m \cdot n$ способами.

Пусть в салоне связи имеется 50 различных моделей сотовых телефонов и по три вида чехлов для каждой модели. Сколькими способами можно выбрать телефон и чехол к нему? Очевидно: имеется 50 вариантов выбора телефона. Выбрав телефон, можно 3 способами выбрать чехол, т.е. всего $50 \times 3 = 150$ вариантов.

Сравнивая утверждение 2.1 и теорему 2.1 с правилами суммы и произведения, можно заметить, что в них речь идет об одних и тех же закономерностях, хотя и используются различные формулировки. Очевидным образом эти правила распространяются на случай большего количества множеств.

2. Комбинаторные формулы»

Комбинаторные конфигурации: перестановки и подстановки

Пусть дано множество $M = \{a_1, a_2, \ldots, a_n\}$. *Перестановкой* элементов множества M называется любой упорядоченный набор из n различных элементов множества M. Перестановки различаются только порядком входящих в них элементов. Перестановка элементов множества M может быть задана посредством *функции подстановки*. Будем определять подстановку как биекцию $\sigma : \mathrm{M} \to \mathrm{M}$ и задавать ее с помощью матрицы, состоящей из двух строк.

Пусть множество $\mathrm{M} = \{1,2,\ldots,n\}$, а $\sigma(k) = s_k$, $1 \le s_k \le n$, $k=1,\ldots,n$, $\{s_1, s_2, \ldots, s_n\} = \{1,2,\ldots,n\}$. Тогда матрица подстановки σ будет иметь вид: $[\sigma] \equiv \begin{pmatrix} 1 & 2 & \ldots & n \\ s_1 & s_2 & \ldots & s_n \end{pmatrix}$.

Число всех перестановок множества M ($|M| = n$) равно $n!$

Сколькими способами можно расставить на полке 6 томов книг? Это можно осуществить $P_6 = 6! = 720$ способами.

Понятие выборки. Пусть дано множество $\mathrm{M} = \{a_1, a_2, a_3, \ldots, a_n\}$, $m \le n$. Набор, состоящий из m элементов множества M, называется выборкой объема m из n элементов.

Выборки классифицируются следующим образом:

По критерию <u>повторяемости элементов</u>: С возвращением объема (с повторениями) и без возвращения объема (без повторений).

По критерию <u>упорядоченности</u>: Упорядоченные (размещения) и неупорядоченные (сочетания).

Иллюстрация: ящик с n нумерованными шариками.

Размещения из n элементов по m, *Сочетания без повторений из n элементов по m*, *Размещения с повторениями* (или *упорядоченными выборками с возвращениями*) *из n элементов по k* ,

В отличие от выборок без повторений, количество выбираемых объектов может быть больше, чем количество типов, т.е. может быть $k \ge n$. Если вернуться к примеру 2.12 (а), то можно рассматривать и 10-разрядные числа.

Теорема*(о мощности множества P(M)): Для конечного множества M* $|2^M| = 2^{|M|}$.

Следствие: *можно сгенерировать все подмножества конечного множества M, перечислив некоторым способом все наборы из нулей и единиц длины n.* Можно выполнять такую генерацию различными способами (например, все наборы с одной 1, все с двумя, …). Это можно сделать наиболее эффективно, используя т.н. *бинарный код Грея*. <u>Алгоритм построения бинарного кода Грея</u> позволяет генерировать последовательность всех подмножеств n-элементного множества таким образом, что каждое последующее подмножество получается из предыдущего добавлением или удалением единственного элемента.

Определим отношение эквивалентности на множестве размещений с повторениями из n элементов по k: $(a_1,a_2,\ldots,a_k) \sim (b_1,b_2,\ldots,b_k) \Leftrightarrow \forall c \in M$ число элементов $a_i = c$ совпадает с числом элементов $b_j = c$.

Тогда *сочетанием с повторениями* из n элементов по k или *неупорядоченной выборкой с возвращениями из n элементов по k* является множество, которое состоит из элементов, выбранных k раз из множества M, причем один и тот же элемент допускается выбирать повторно.

> В примере с множеством M={1,2,3,4,5} сочетания с повторениями из 5 элементов по 2 будут отличаться от размещений тем, что одинаковые по составу наборы будут независимо от порядка элементов в них считаться эквивалентными: (1,1), (1,2)~(2,1), (2,2), (5,2) и т.п.

При рассмотрении выборок с повторениями число n более наглядно трактуется как количество имеющихся в наличии типов объектов, а k – количество непосредственно выбираемых объектов. Раз объекты выбираются с повторениями, неважно, каково их реальное количество для каждого из типов. Можно считать их неисчерпаемыми.

Число всех сочетаний с повторениями обозначается $\overline{C}_n^k = \hat{C}(n,k)$ и вычисляется по формуле: $\hat{C}(n,k) = \overline{C}_n^k = C_{n+k-1}^k = \dfrac{(n+k-1)!}{k!(n-1)!}$ **(2.2)**

> Пусть в кондитерской продается 10 различных видов пирожных. (n=10 – число типов). Сколькими способами можно купить 12 пирожных? (k=12). $\hat{C}$(10,12)=C(10+12–1,12)=C(21,12)=21!/(12! (10–1)!)= 21!/(12! 9!).

1.5 Лекция № 9, 10 (4 часа).

Тема: «Основные понятия теории графов. Числовые характеристики графов»

1.5.1 Вопросы лекции:

1. Основные понятия теории графов. Виды графов.

2. Операции над графами

3. Способы задания графов. Матричное представление графов

1.5.2 Краткое содержание вопросов:

1. Основные понятия теории графов. Виды графов.

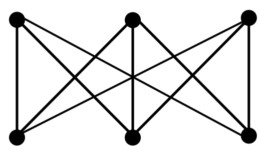
Определение графа. Часто бывает полезно и наглядно изобразить некоторую ситуацию в виде рисунка, состоящего из точек (вершин), представляющих основные элементы ситуации и линий (ребер), отражающих связи между элементами. Такие рисунки называются *графами*.

Между рассмотренным ранее понятием отношения и понятием графа существует тесная связь. Теория графов представляет собой удобный язык для описания программных и других моделей. Граф – это удобный способ изображения различных взаимосвязей (отношений). Граф может изображать сеть улиц в городе (вершины – перекрестки, улицы – ребра), блок-схемы программ, электрические цепи, географические карты и т.д

История теории графов. ***Теория графов возникла из решения различных прикладных задач. Первые задачи были связаны с решением математических развлекательных задач и головоломок. Рассмотрим эти задачи (пояснить подробнее)***

1. *Задача о Кенигсбергских мостах.* Необходимо обойти все 4 части суши, пройдя по каждому мосту один раз, и вернуться в исходную точку. Ее развитие привело к циклу задач об обходах графов (Леонард Эйлер, 1736 г.).

2. *Задача о трех домах и трех колодцах*. Есть три дома и три колодца. Жители домов поссорились. Требуется от каждого дома проложить тропинку к каждому колодцу так, чтобы эти тропинки не пересекались. (Куратовский, 1930)



3.Задача о четырех красках. Любую карту на плоскости раскрасить четырьмя красками так, чтобы никакие две соседние области не были закрашены одинаково. Эта задача была сформулирована в середине XIX века, и попытки ее решить привели к появлению некоторых исследований графов, имеющих теоретическое и прикладное значение.

Многие результаты середины XIX века были получены при решении практических проблем. (Например, Кирхгоф: система уравнений токов и напряжений в электротехнической схеме представлялась графом и решалась с помощью методов теории графов; химия; Задача о перевозках, решение которой привело к созданию эффективных методов решения транспортных задач …). В XX веке задачи, связанные с графами, получили распространение не только в физике, электротехнике, химии, биологии, экономике, но и внутри различных разделов математики (алгебра, теория чисел, теория вероятностей и др.).

В проблематике теории графов можно выделить направления комбинаторного и геометрического характера. К первому относятся задачи о построении графов с заданными свойствами, о подсчете и перечислении таких графов. Геометрический характер носят, например, задачи, связанные с обходами графов. Характерным специфическим направлением теории графов является цикл проблем, связанных с раскрасками, в которых изучаются разбиения множества вершин, обладающие определенными свойствами.

Основные понятия**.** ***Граф G* определяется как упорядоченная пара ⟨*V, E*⟩, где *V* – непустое множество *вершин*, отношение $E \subset V^2$ – множество *ребер* (набор неупорядоченных или упорядоченных пар вершин). Вершины и ребра графа называются его *элементами*.**

Граф, содержащий конечное число элементов, называется *конечным*. Число вершин конечного графа называется его *порядком* и обозначается $|V|$, число ребер обозначается как $|E|$: $G(V,E) = \langle V, E\rangle$, $V \neq \varnothing$, $E \subset V\times V$, $E = E^{-1}$.

Граф порядка *n*, имеющий *m* ребер, называется *(n,m)-графом*.

Обычно граф изображают *диаграммой*: вершины – точками или кружками, ребра – линиями (нарисовать). Такой способ задания графа является самым простым и наглядным, хотя и годится только для простейших случаев. Кроме того, затруднительно обрабатывать такой граф с помощью ЭВМ. Поэтому существуют специальные способы представления графа в ЭВМ, которые мы рассмотрим чуть позже.

Пусть v_1 и v_2 – вершины, e – соединяющее их ребро. Тогда ребро e и каждая из этих вершин называются *инцидентными* друг другу, вершины v_1 и v_2 называются *смежными*. Два ребра, имеющие одну общую вершину (инцидентные одной вершине), также называются *смежными*.

Множество вершин, смежных с вершиной v, называется *множеством смежности* (*окружением*) вершины v и обозначается $\Gamma^+(v) = \{u \in V \mid (u,v) \in E\}$, $\Gamma(v) = \Gamma^*(v) = \Gamma^+(v) \cup \{v\}$. Очевидно, что: $u \in \Gamma(v) \Leftrightarrow v \in \Gamma(u)$. Если не оговорено противное, то подразумевается Γ^+ и обозначается просто Г. Если A – множество вершин, то $\Gamma(A)$ – множество вершин, смежных с вершинами из A: $\Gamma(A) = \{u \in V \mid \exists v \in A \ \ u \in \Gamma(v)\} = \cup\Gamma(v) \ \forall v \in A$.

Другие определения графов и бинарные отношения.***Часто рассматриваются следующие разновидности графов.***

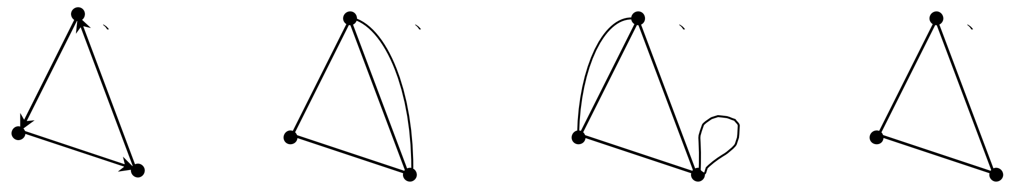
1. В некоторых задачах инцидентные ребру вершины рассматриваются в определенном порядке. Тогда элементами множества $E=\{(u,v) \mid u,v \in V\}$ являются упорядоченные пары, т.е. ребру приписывается направление от одной вершины к другой, и ребра называются *дугами* (говорят, что дуга *выходит* из вершины *u* и *заходит* в вершину *v*). Вершины в таком графе называются *узлами*, а сам граф, все ребра которого являются дугами, называется *ориентированным* графом, или *орграфом* (см. рис. а)). Иногда рассматриваются и *смешанные* графы, имеющие как дуги, так и неориентированные ребра.

2. Различные ребра графа могут быть инцидентны одной и той же паре вершин, в этом случае они называются *кратными* ребрами, а сам граф – *мультиграфом* (см. рис. б)).

3. Если элементом множества E является пара одинаковых элементов V, то такое ребро соединяет вершину саму с собой. Тогда это ребро называется *петлей*, а граф – *псевдографом* (рис. в)). В псевдографе возможно также наличие кратных ребер.

4. В отличие от мультиграфа и псевдографа, граф без петель и кратных ребер называется *простым*.

5. Если задана функция $F : V \to M$ или $F : E \to M$, то множество M называется *множеством пометок*, а сам граф называется *размеченным* (т.е. всем его вершинам или всем ребрам присвоены некоторые метки, в качестве которых обычно используются буквы или целые числа – г)).



Далее, говоря «граф G(V,E)», будем иметь в виду неориентированный непомеченный граф без петель и кратных ребер.

Фактически, графы и бинарные отношения – это один и тот же класс объектов, описанный разными средствами. Отношения (в частности, функции) являются базовыми средствами для построения большинства математических моделей, используемых при решении практических задач. С другой стороны, графы допускают наглядное представление в виде диаграмм. Это объясняет широкое использование графов при кодировании и проектировании программ.

Любой граф с петлями, но без кратных ребер, задает бинарное отношение E на множестве V, и обратно. Пара элементов принадлежит отношению: $(a,b)\in E\subset V\times V \Leftrightarrow$ в графе есть G ребро (a,b). Неориентированный граф соответствует симметричному отношению. Изменение направления всех дуг соответствует обратному отношению. Мультиграф, все вершины которого имеют петли, задает рефлексивное отношение.

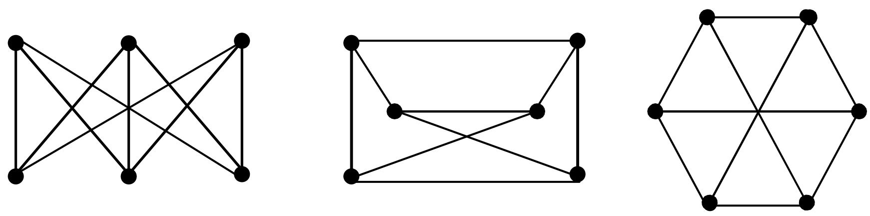
Изоморфизм графов: ***При изображении графа точки, обозначающие его вершины, берутся совершенно произвольно, поэтому рисунки одного и того же графа могут быть совершенно непохожими. Как же понять, одинаковы ли графы, изображенные разными чертежами? Решение проблемы стандартное – если можно взаимно однозначно отобразить множество вершин одного графа на множество вершин другого так, чтобы сохранилось отношение смежности, то это две копии графа.***

Говорят, что два графа $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$ *изоморфны*: $G_1\sim G_2$, если существует биекция (1-1 соответствие) $h: V_1\rightarrow V_2$, сохраняющая отношение инцидентности (при которой смежные вершины (ребра) графа G_1 переходят в смежные вершины (ребра) графа G_2):
$e_1=(u,v)\in E_1 \Rightarrow e_2=(h(u),h(v))\in E_2$; $\quad e_2=(u,v)\in E_2 \Rightarrow e_1=(h^{-1}(u),h^{-1}(v))\in E_1$;

Графы, отличающиеся только нумерацией вершин, являются *изоморфными*. Изоморфизм графов является отношением эквивалентности. Действительно, изоморфизм обладает всеми необходимыми свойствами: 1) рефлексивность – $G\sim G$, где требуемая биекция есть тождественная функция;

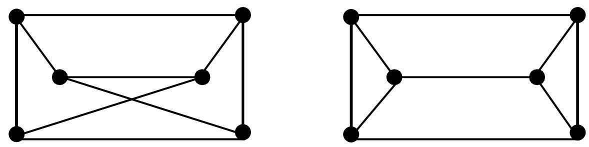
2) симметричность – если $G_1\sim G_2$ с биекцией h, то $G_2\sim G_1$ с биекцией h^{-1};
3) транзитивность – если $G_1\sim G_2$ с биекцией h, а $G_2\sim G_3$ с биекцией g; то $G_1\sim G_3$ с биекцией $g\circ h$.

Графы рассматриваются с точностью до изоморфизма, т.е. рассматриваются классы эквивалентности по отношению изоморфизма.



Три внешне различные диаграммы, приведенные на рисунке, являются диаграммами одного и того же графа.

Числовая характеристика, одинаковая для всех изоморфных графов, называется *инвариантом* графа. В частности, количество вершин и количество ребер – инварианты графа *G*.



Не известно никакого набора инвариантов, определяющих граф с точностью до изоморфизма.

2. Операции над графами

3. Способы задания графов. Матричное представление графов

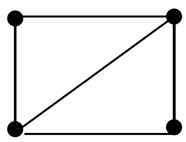
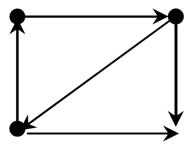
Представление графов в ЭВМ. Требования к представлению графов. ***Чтобы***

задать граф, нужно каким-либо способом описать множество его вершин, множество его ребер, а также указать, какие вершины и ребра инцидентны (или смежные), т.е. задать отношение инцидентности (смежности). Рассмотрим несколько способов представления графа в ЭВМ. Они различаются объемом занимаемой памяти и скоростью выполнения операций над графами. Представление выбирается по потребностям конкретной задачи.

Напомним: число вершин графа обозначаем через n, а число ребер – через m. Характеристика $M(n,m)$, приведенная для каждого представления, означает требуемый для него объем памяти.

Указанные представления пригодны для графов и орграфов, а после некоторой модификации – для псевдографов, мультиграфов и гиперграфов.

Все представления будем иллюстрировать на конкретных примерах графа G и орграфа D (см. рисунок.).

Способы представления графа

1) Матрица смежности.

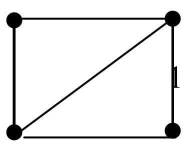
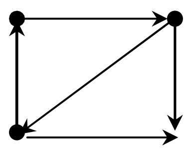
Матрица смежности A(G') графа (орграфа) – это квадратная матрица размера n×n, у которой для любых $i,j \in \{1,2,\dots,n\}$ элемент в i-й строке и j-м столбце равен 1, если i-я и j-я вершины соединены ребром (дугой с началом в вершине i), и равен 0 в противном случае.

$$a_{ij} = \begin{cases} 1, & \textit{если вершины } v_i \textit{ и } v_j - \textit{смежные } (\textit{для орграфа дуга идет из } v_i \textit{ в } v_j) \\ 0, & \textit{иначе} \end{cases}$$

Память $M(n,m)=O(n^2)$.

Фактически это уже знакомая нам матрица бинарного отношения. Очевидно, что матрица смежности неориентированного графа является симметричной, элементы главной диагонали равны нулю, а количество единиц в каждой строке равно степени вершины, которой соответствует эта строка. По матрице смежности легко построить диаграмму графа.

Матрица смежности орграфа, не являющегося мультиграфом, не может быть симметричной, т.к. при ее составлении вершины орграфа играют различные роли.

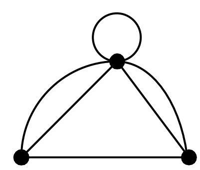



Матрицы смежности для заданных графа G и орграфа D

$$A(G)=\begin{pmatrix}0&1&0&1\\1&0&1&1\\0&1&0&1\\1&1&1&0\end{pmatrix}\quad A(D)=\begin{pmatrix}0&1&0&0\\0&0&1&1\\0&0&0&0\\1&0&1&0\end{pmatrix}$$

В матрице смежности мультиграфа или псевдографа число, находящееся на пересечении i-й строки и j-го столбца, совпадает с числом ребер, соединяющих вершины i и j, при этом каждая петля считается двумя ребрами.

Псевдограф, изображенный на рисунке, имеет матрицу смежности следующего вида: $A(P)=\begin{pmatrix}0&2&1\\2&2&2\\1&2&0\end{pmatrix}$



2) Матрица инцидентности.

Другой способ задать граф – определить *матрицу инцидентности* (или *инциденций*) $I(G)$, имеющую n строк и m столбцов, элементы которой задаются следующим образом:

$$i_{kl}=\begin{cases}1, & \textit{если вершина } v_k \textit{ инцидентна ребру } e_l\\ 0, & \textit{иначе}\end{cases}$$

Для ориентированного графа:

$$i_{kl}=\begin{cases}1, & \textit{если вершина } v_k \textit{ инцидентна ребру } e_l \textit{ и является его концом}\\ 0, & \textit{если вершина } v_k \textit{ и ребро } e_l \textit{ не инцидентны}\\ -1, & \textit{если вершина } v_k \textit{ инцидентна ребру } e_l \textit{ и является его началом}.\end{cases}$$

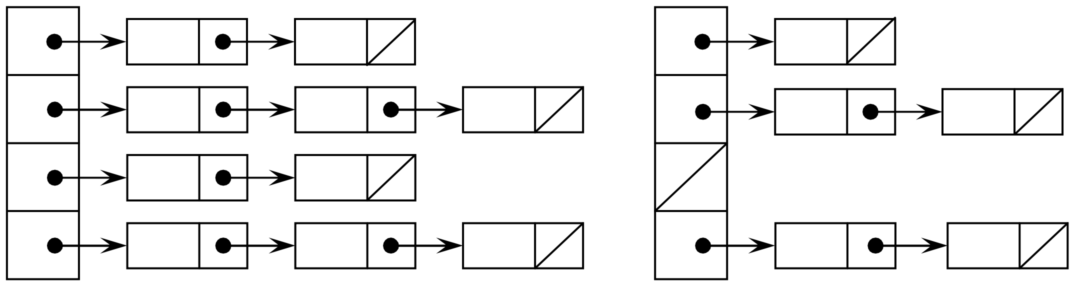
Матрицы инцидентности для заданных графа G и орграфа D $I(G)=\begin{pmatrix}1&0&0&1&0\\1&1&0&0&1\\0&1&1&0&0\\0&0&1&1&1\end{pmatrix}$

$$I(D)=\begin{pmatrix}-1&0&0&1&0\\1&-1&0&0&-1\\0&1&1&0&0\\0&0&-1&-1&1\end{pmatrix}$$

Очевидно, что в каждом столбце матрицы инцидентности только два элемента отличны от 0 (или один, если ребро является петлей), т.к. ребро может быть инцидентно не более чем двум вершинам (а столбец соответствует ребру). Поэтому матрица содержит много нулей и такой способ описания неэкономен. $M(n,m)=O(n\cdot m)$.

3) Списки смежности.

Граф представляется с помощью списочной структуры (списка смежности), отражающей смежность вершин и состоящей из массива указателей на списки смежных вершин. Элемент списка представлен структурой с двумя полями: номер вершины и указатель. Для неориентированных графов $M(n,m)=O(n+2m)$, для орграфов $M(n,m)=O(n+m)$.



Списки смежности для заданных графа G и орграфа D:

он	ач он
1	
1	
2	
2	
3	

4) Массив ребер (дуг).

Отношение инцидентности можно задать также списком ребер графа. Каждая строка этого списка соответствует ребру, в ней записаны номера вершин, инцидентных ему. $M=O(2m)$.

По списку ребер графа легко построить матрицу инцидентности, т.к. каждое ребро этого списка соответствует столбцу матрицы, а номера вершин в каждом элементе списка – это номера строк матрицы инцидентности, элементы в которых равны 1. Для орграфа координата начала – номер строки, где стоит –1, а координата конца – номер строки, где стоит 1.

1.6 Лекция № 11 (2 часа).

Тема: «Деревья. Планарные и хроматические графы»

1.6.1 Вопросы лекции:

1. Деревья.

2.Планарные графы. Хроматические графы

1.6.2 Краткое содержание вопросов:

1. Деревья

Деревья являются простейшим классом графов. Для них выполняются многие свойства, которые не всегда выполняются для обычных графов. Кроме того, деревья широко применяются в программировании при различного рода обработке данных, в частности, в алгоритмах сортировки, кодирования и т.п. Подробно алгоритмы работы с деревьями будут рассматриваться позднее в других курсах, а сейчас только краткое знакомство.

Дерево – это связный граф без циклов. Несколько деревьев (или несвязный граф без циклов) составляют *лес*. Таким образом, дерево является компонентой связности леса.

Пусть $G=(S,U)$ и $|S|=n$, $|U|=m$. Тогда справедлива эквивалентность следующих утверждений:

1). G - дерево;

2). G - связный граф и $m=n-1$;

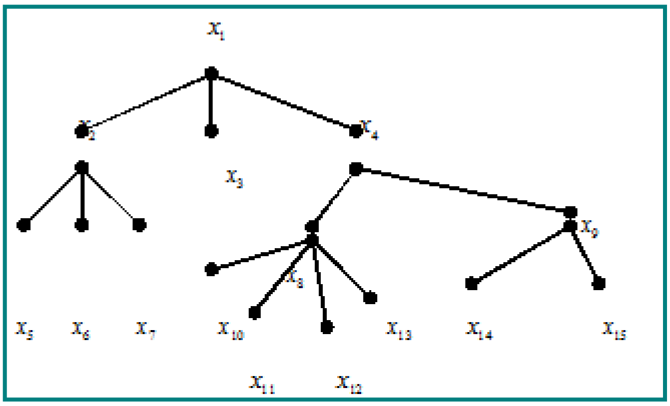
3). G - ациклический граф и $m=n-1$;

4). любые две несовпадающие вершины графа соединяет единственная простая цепь;

5). G - ациклический граф, обладающий тем свойством, что если какую- либо пару его несмежных вершин соединить ребром, то полученный граф будет содержать ровно один цикл.

Ориентированный граф называется ориентированным деревом (ордеревом), если: 1). существует ровно одна вершина $x_1 \in S$, называемая корнем, которая не имеет предшествующих вершин, то есть $P(x_1)=0$;

2). любой вершине $x_j \neq x_1$ в графе G непосредственно предшествует ровно одна вершина, то есть $P(x_j)=1$.



Неориентированное дерево можно превратить в ориентированное, выбрав в качестве корня произвольную вершину. Пусть $G=(S,U)$. Граф $G'=(S',U')$ называется подграфом графа G, если $S' \subset S$ и $U' \subset U$. Подграф G' графа G называется остовным подграфом, если $S'=S$. Подграф G' графа графа G называется остовным поддеревом (остовом, каркасом), если $S'=S$ и G - дерево.

Теорема Кэли[*]. *Число различных деревьев, которые можно построить на n различных вершинах, равно $t_n=n^{n-2}$.*

В этой формуле подсчитывается число всех деревьев с данными n вершинами. Многие из этих деревьев изоморфны, и возникает вопрос о числе не изоморфных деревьев среди них. Это более трудная задача, она решается для каждого конкретного случая по алгоритму теории Пойа.

Вернемся к произвольным графам. Матрицей Кирхгофа** графа G называется матрица $B_{n\times n}$, $n=|S|$, если $b_{ij}=\begin{cases}-1,\ x_i \text{ и } x_j \text{ смежны,}\\ 0,\ x_i \text{ и } x_j \text{ не смежны,}\\ P(x_i),\ i=j.\end{cases}$ Сумма элементов в каждой строке и каждом столбце этой матрицы равна нулю, то есть $\sum\limits_{i=1}^{n} b_{ij}=0,\ j=\overline{1,n}$, $\sum\limits_{j=1}^{n} b_{ij}=0,\ i=\overline{1,n}$.

Кроме того, из этого следует, что алгебраические дополнения всех элементов матрицы B равны между собой. Матрица Кирхгофа используется для подсчета числа остовов в графе.

* Артур Кэли(Кэйли) (1821-1895 г.г.) - английский математик.

** Густав Роберт Кирхгоф (1824-1887 г.г.) -немецкий физик

Теорема Кирхгофа. *Число остовных деревьев в связном графе* G *порядка* $n\geq 2$ *равно алгебраическому дополнению любого элемента матрицы Кирхгофа* $B(G)$.

1.7 Лекция № 12 (2 часа).

Тема: «Оптимизационные задачи на графах и сетях, алгоритмы их решения. Прикладные задачи и алгоритмы анализа графов. Потоки в сетях. Сетевое планирование»

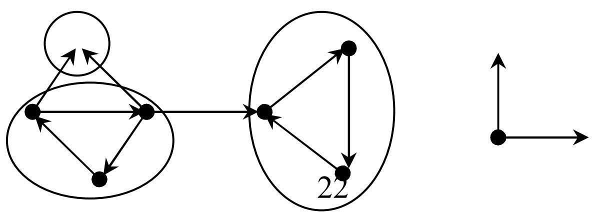
1.7.1 Вопросы лекции:

1. Оптимизационные задачи на графах и сетях, алгоритмы их решения.
2. Прикладные задачи и алгоритмы анализа графов**.**
3. Потоки в сетях. Задача о максимальном потоке.
4. Сетевое планирование.

1.7.2 Краткое содержание вопросов:

1. Оптимизационные задачи на графах и сетях, алгоритмы их решения.

1) Выделение компонент связности в орграфах. Компоненты сильной связности (КСС) орграфа G – это его максимальные сильно связные подграфы. Каждая вершина орграфа принадлежит только одной КСС. Если вершина не связана с другими, то считается, что она сама образует КСС. Орграф, который получается стягиванием в одну вершину каждой КСС, называется фактор-графом, или конденсацией орграфа G.



Слева на рисунке орграф, справа – его фактор-граф. Овальными линиями показаны КСС, стянутые в одну вершину фактор-графа.

Для выделения компонент сильной связности орграфа можно в качестве основы алгоритма использовать метод поиска в глубину.

Теорема. ***Любой граф представляется в виде объединения непересекающихся связных (сильных) компонент. Разложение графа на связные (сильные) компоненты определяется однозначно.***

2) Кратчайшие пути. Задача поиска кратчайшего пути (наиболее дешевого? короткого?) «от пункта А до пункта В» имеет массу практических приложений и различные алгоритмы решения. Математическая постановка задачи имеет следующий вид.

Рассматривается взвешенный граф (орграф) $G(V,E)$, ребрам (дугам) которого сопоставлены веса, обозначающие длину (или стоимость) пути из одного конца ребра в другой. Если из вершины v_i нет ребра (дуги) в вершину v_j, то вес ребра (v_i,v_j) считается равным ∞. Для ребер, являющихся петлями (диагональ матрицы смежности), их веса считаются равными 0. Все компоненты матрицы – веса ребер, соединяющих соответствующие вершины. Требуется определить кратчайший путь из одной вершины в другую.

Наиболее широко известны два алгоритма поиска кратчайших путей. Алгоритм Дейкстры находит кратчайшее расстояние от одной фиксированной вершины до другой и указывает сам путь, длина которого равна этому расстоянию. Алгоритм Флойда-Уоршалла позволяет найти кратчайшие расстояния между всеми парами вершин графа.

Алгоритм Дейкстры. Находит кратчайшее расстояние от вершины v_1 до вершины v_n

Рассмотрим еще несколько алгоритмов нахождения кратчайшего пути между двумя заданными вершинами в ориентированной сети. Пусть $G = \{S, U, \Omega\}$ - ориентированный граф со взвешенными дугами. Обозначим s - вершину - начало пути и t - вершину – конец пути.

Общий подход к решению задачи о кратчайшем пути был развит американским математиком Ричардом Беллманом[**], который предложил название динамическое программирование.

[*] Едсгер Дейкстра (1930-2002 г.г.) - нидерландский математик.

[**] Ричард Эрнест Беллман (1920-1984 г.г.) - американский математик.

Задача о кратчайшем пути частный случай следующей задачи: найти в заданном графе пути, соединяющие две заданные вершины и доставляющие минимум или максимум некоторой аддитивной функции, определенной на путях. Чаще всего эта функция трактуется как длина пути и задача называется задачей о кратчайших путях. Алгоритм Дейкстры одна из реализаций этой задачи. Его часто называют алгоритмом расстановки меток. В процессе работы этого алгоритма узлам сети $x_i \in S$ приписываются числа (метки) $d(x_i)$, которые служат оценкой длины (веса) кратчайшего пути от вершины s к вершине x_i. Если вершина x_i получила на некотором шаге метку $d(x_i)$, это означает, что в графе G существует путь из s в x_i, имеющий вес $d(x_i)$. Метки могут находиться в двух состояниях – быть временными или постоянными. Превращение метки в постоянную означает, что кратчайшее расстояние от вершины s до соответствующей вершины найдено.

Алгоритм Дейкстры состоит из двух этапов. На первом этапе находится длина кратчайшего пути, на втором – строится сам путь от вершины s к вершине t.

Этап 1. Нахождения длины кратчайшего пути.

Шаг 1. Присвоение вершинам начальных меток.

Полагаем $d(s) = 0^*$ и считаем эту метку постоянной (постоянные метки помечаются сверху звездочкой). Для остальных вершин $x_i \in S,\ x_i \neq s$ полагаем $d(x_i) = \infty$ и считаем эти метки временными. Пусть $\tilde{x} = s,\ \tilde{x}$ - обозначение текущей вершины.

Шаг 2. Изменение меток.

Для каждой вершины x_i с временной меткой, непосредственно следующей за вершиной $\tilde{x}$, меняем ее метку в соответствии со следующим правилом:

$$d_{нов.}(x_i) = \min\{d_{стар.}(x_i), d(\tilde{x}) + \omega(\tilde{x}, x_i)\}. \qquad (4.7.1)$$

Шаг 3. Превращение метки из временной в постоянную.

Из всех вершин с временными метками выбираем вершину x_j^* с наименьшим значением метки

$$d(x_j^*) = \min\left\{ d(x_j) \big/ x_j \in S,\ d(x_j) - \text{временная} \right\}$$

(4.7.2)

Превращаем эту метку в постоянную и полагаем $\tilde{x} = x_j^*$.

Шаг 4. Проверка на завершение первого этапа.

Если $\tilde{x} = t$, то $d(\tilde{x})$ - длина кратчайшего пути от s до t. В противном случае происходит возвращение ко второму шагу.

Этап 2. Построение самого кратчайшего пути.

Шаг 5. Последовательный поиск дуг кратчайшего пути.

Среди вершин, непосредственно предшествующих вершине $\tilde{x}$ с постоянными метками, находим вершину x_i, удовлетворяющую соотношению

$$d(\tilde{x}) = d(x_i) + \omega(x_i, \tilde{x}). \qquad (4.7.3)$$

Включаем дугу $(x_i, \tilde{x})$ в искомый путь и полагаем $\tilde{x} = x_i$.

Шаг 6. Проверка на завершение второго этапа.

Если $\tilde{x} = s$, то кратчайший путь найден – его образует последовательность дуг, полученных на пятом шаге и выстроенных в обратном порядке. В противном случае возвращаемся к пятому шагу.

2. Прикладные задачи и алгоритмы анализа графов.

Задача об остове минимального веса. Пусть $G = (S, U)$ - связная сеть. В приложениях часто возникает задача о построении остова графа G, имеющего наименьший вес. Пусть, например, $G = (S, U, \Omega)$ служит моделью железнодорожной сети, соединяющей пункты $x_1, x_2, ..., x_n \in S$, а $\omega(x_i, x_j)$ - расстояние между пунктами x_i и x_j. Требуется проложить сеть телеграфных линий вдоль линий железнодорожной сети так, чтобы все пункты $x_1, x_2, ..., x_n$ были связаны между собой телеграфной сетью и общая протяженность линий телеграфной сети была наименьшей.

Известно несколько алгоритмов построения кратчайшего остовного дерева. Рассмотрим алгоритм Прима*, представляющий собой итерационную процедуру, состоящую из двух шагов и выполняющуюся $n-1$ раз на графе G с n вершинами.

Пусть $S^{/} \subset S$, $S^{//} \subset S$ и $S = S^{/} \cup S^{//}$, $S^{/} \cap S^{//} = \varnothing$, то есть $S^{/}$ и $S^{//}$ - разбиение множества узлов сети G на два непересекающихся подмножества. Определим пошаговое расстояние между множествами $S^{/}$ и $S^{//}$ следующим образом:

$$d\left(S^{/}, S^{//}\right) = \min\left\{ \omega\left(x_i, x_j\right) \Big/ x_i \in S^{/},\ x_j \in S^{//} \right\},$$

где $\left(x_i, x_j\right)$ - дуга, соединяющая вершины x_i и x_j.

В алгоритме Прима остовное дерево строится в результате последовательного расширения исходного поддерева. На каждой итерации число вершин и ребер поддерева увеличивается на единицу. Основные шаги алгоритма таковы.

Шаг 1. (Присвоение начальных значений).

Полагают $S^{/} = \{x_1\}$, где x_1 - произвольная вершина, $S^{//} = S / S^{/}$, $U^{/} = \varnothing$.

Шаг 2. (Обновление данных).

Находится ребро $\left(x_i, x_j\right)$ такое, что $x_i \in S^{/}$, $x_j \in S^{//}$ и $\omega\left(x_i, x_j\right) = \min\left\{ \omega\left(x_i, x_j\right) \Big/ x_i \in S^{/},\ x_j \in S^{//} \right\}$. Полагают $S^{/} = S^{/} \cup \{x_j\}$, $S^{//} = S / S^{/}$, $U^{/} = U^{/} \cup \{\left(x_i, x_j\right)\}$.

Шаг 3. (Проверка на завершение).

Если $S^{/} = S$, то $G^{/} = \left(S^{/}, U^{/}\right)$ - искомый остов. В противном случае переходят ко второму шагу.

Пример. Построить остов с наименьшим весом для сети, заданной матрицей смежности вершин. Построим по этой матрице сеть. Поскольку матрица симметрическая, то граф дан неориентированный. Исходный граф изображен внизу следующей страницы.

$$W = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{pmatrix} - & 5 & 10 & 14 & \infty & \infty \\ 5 & - & 5 & 6 & \infty & \infty \\ 10 & 5 & - & 7 & 8 & 9 \\ 14 & 6 & 7 & - & 4 & \infty \\ \infty & \infty & 8 & 4 & - & 12 \\ \infty & \infty & 9 & \infty & 12 & - \end{pmatrix} \end{array}.$$

Шаг1. $S^{/} = \{x_1\}$, $S^{//} = \{x_2, x_3, x_4, x_5, x_6\}$, $U^{/} = \varnothing$.

Первая итерация. Шаг 2.

$d\left(S^{/}, S^{//}\right) = \omega\left(x_1, x_2\right) = 5$, $S^{/} = \{x_1, x_2\}$, $S^{//} = \{x_3, x_4, x_5, x_6\}$,

$U^{/} = \{\left(x_1, x_2\right)\}$.

Шаг 3. $S^{/} \neq S$, переход на начало второго шага.

Вторая итерация. Шаг 2.

$d\left(S^{/}, S^{//}\right) = \omega\left(x_2, x_3\right) = 5$, $S^{/} = \{x_1, x_2, x_3\}$, $S^{//} = \{x_4, x_5, x_6\}$,

$U^{/} = \{\left(x_1, x_2\right), \left(x_2, x_3\right)\}$.

Шаг 3. $S^{/} \neq S$, переход на начало второго шага.

Третья итерация. Шаг 2.

$d\left(S^{/}, S^{//}\right) = \omega\left(x_2, x_4\right) = 6,\ S^{/} = \{x_1, x_2, x_3, x_4\},\ S^{//} = \{x_5, x_6\},$

$U^{/} = \{(x_1, x_2), (x_2, x_3), (x_2, x_4)\}.$

Шаг 3. $S^{/} \neq S$, переход на начало второго шага.

Четвертая итерация. Шаг 2. $d\left(S^{/}, S^{//}\right) = \omega\left(x_4, x_5\right) = 4,\ S^{/} = \{x_1, x_2, x_3, x_4, x_5\},\ S^{//} = \{x_6\},$

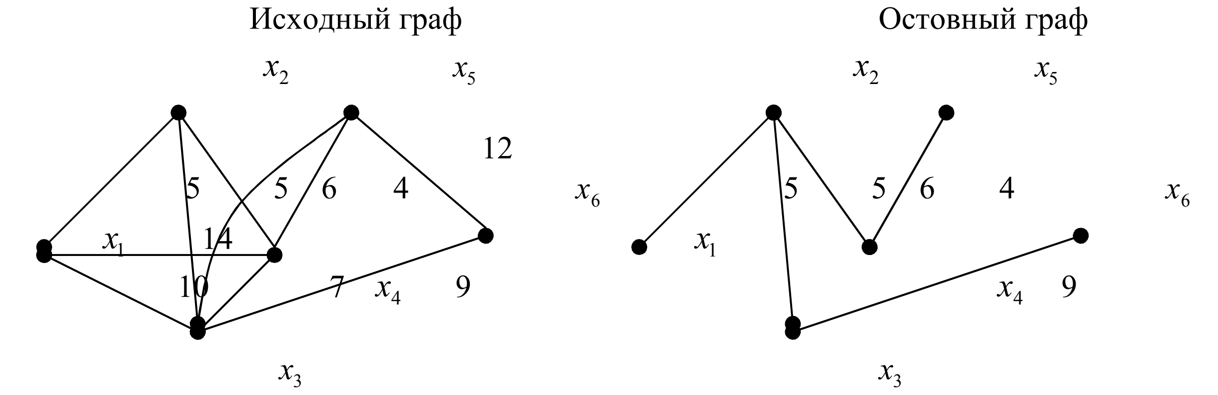
$U^{/} = \{(x_1, x_2), (x_2, x_3), (x_2, x_4), (x_4, x_5)\}.$

Шаг 3. $S^{/} \neq S$, переход на начало второго шага.

Пятая итерация. Шаг 2. $d\left(S^{/}, S^{//}\right) = \omega\left(x_3, x_6\right) = 9,\ S^{/} = \{x_1, x_2, x_3, x_4, x_5, x_6\},\ S^{//} = \varnothing,$

$U^{/} = \{(x_1, x_2), (x_2, x_3), (x_2, x_4), (x_4, x_5), (x_3, x_6)\}.$

Шаг 3. $S^{/} = S$. Итак, получен остовный граф. $G^{/} = \left(S^{/}, U^{/}\right)$ изображен на рисунке справа, его вес $\omega\left(G^{/}\right) = 5 + 5 + 6 + 4 + 9 = 29.$



3. Потоки в сетях. Задача о максимальном потоке.

Потоки в сетях. Функциональное назначение большинства физически реализованных сетей состоит в том, что они служат носителями систем потоков, то есть систем, в которых некоторые объекты текут, движутся или транспортируются по системе каналов (дуг сети) ограниченной пропускной способности. Примерами могут служить поток автомобильного транспорта по сети автодорог, поток грузов по участку железнодорожной сети, поток воды в городской сети водоснабжения, поток электрического тока в электросети, поток телефонных или телеграфных сообщений по каналам связи, поток программ в вычислительной сети. Ограниченная пропускная способность означает, что интенсивность перемещения соответствующих предметов по каналу ограничена сверху определенной величиной.

Наиболее часто в сети решается задача о максимальном потоке и минимальном разрезе. При этом граф $G = (S, U)$ должен удовлетворять следующим условиям:

1). G - связный граф без петель;

2). существует ровно одна вершина, не имеющая предшествующих; эта вершина называется источником и обозначается s;

3). существует ровно одна вершина, не имеющая последующих; эта вершина называется стоком и обозначается t;

4). каждой дуге $(x_i, x_j) \in U$ поставлено в соответствие неотрицательное число $c(x_i, x_j)$, называемое <u>пропускной способностью дуги.</u>

Функция $\varphi(x_i, x_j)$, определенная на множестве дуг сети $G = (S, U, \Omega)$, называется потоком, если $0 \le \varphi(x_i, x_j) \le c(x_i, x_j)\ \forall (x_i, x_j) \in U$ и $\sum\limits_{x_j \in S_{пр}(x_i)} \varphi(x_i, x_j) = \sum\limits_{x_j \in S_{сл}(x_i)} \varphi(x_i, x_j)$ для любой вершины $x_i \in S$ и $x_i \notin \{s, t\}$.

Последнее условие называется условием сохранения потока, в промежуточных вершинах потоки не создаются и не исчезают.

Величина $\Delta(x_i, x_j) = c(x_i, x_j) - \varphi(x_i, x_j)$ называется <u>остаточной пропускной способностью</u> дуги (x_i, x_j). Если $\varphi(x_i, x_j) = c(x_i, x_j)$, то дуга называется <u>насыщенной.</u>

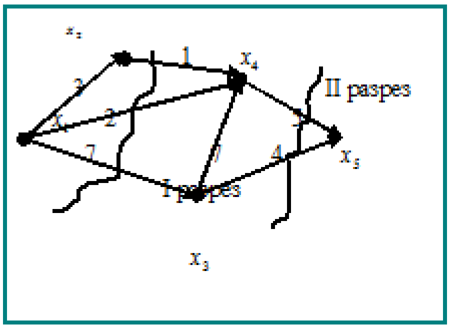
Максимальный поток определяется с помощью одного из основных понятий теории сетей – <u>разреза.</u> Разрез может быть определен как множество дуг, исключение которых из сети отделило бы некоторое множество узлов от остальной сети. Предположим, что множество вершин сети S разбито на два непустых непересекающихся подмножества $S = S' \cup S''$ и $S' \cap S'' = \varnothing$.

Множество дуг, начала которых лежат в S', а концы в S'', называется ориентированным разрезом и обозначается $(S' \to S'')$. Следовательно,

$$(S' \to S'') = \left\{ (x_i, x_j) \big/ x_i \in S',\ x_j \in S'' \right\}.$$

Пропускной способностью или величиной разреза $(S' \to S'')$ называется сумма пропускных способностей входящих в него дуг, то есть

$$c(S' \to S'') = \sum_{x_i \in S',\, x_j \in S''} c(x_i, x_j).$$



На рисунке слева изображена сеть, на которой около каждого ребра указана его пропускная способность. Произведены два разреза: I и II. При разрезе I вершины оказались разбиты на подмножества $S' = \{x_1, x_2\}$ и $S'' = \{x_3, x_4, x_5\}$, а ребрами, образующими разрез стали ребра $(x_1, x_3), (x_1, x_4), (x_2, x_4)$. При разрезе II $S' = \{x_1, x_2, x_3, x_4\}$, а $S'' = \{x_5\}$, разрез образуют ребра $(x_3, x_5), (x_4, x_5)$.

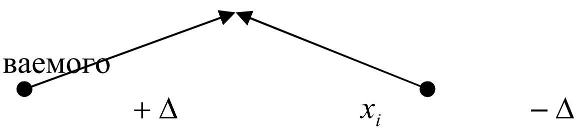
Теорема Форда* **– Фалкерсона**. *Для любой сети с одним источником и одним стоком величина максимального потока в сети от источника к стоку равна величине минимального разреза.*

Алгоритм Форда – Фалкерсона построения максимального потока и минимального разреза основан на следующих обстоятельствах.

1. Предположим, что в сети имеется некоторый поток и путь из s в t, состоящий из ненасыщенных дуг. Тогда очевидно, что поток в сети можно увеличить на величину Δ, равную минимальной из остаточных пропускных способностей дуг, входящих в этот путь. Перебирая все возможные пути из s в t и проводя такую процедуру увеличения потока, пока это возможно, получим в результате полный поток, то есть такой поток, для которого каждый путь из s в t содержит по крайней мере одну насыщенную дугу.

2. Рассмотрим произвольный маршрут (неориентированный путь) из s в t. Дуги, образующие этот маршрут, естественным образом делятся на два типа: прямые (ориентированные от s к t) и обратные (ориентированные от t к s). Пусть существует путь, в котором прямые дуги не насыщены, а потоки на обратных дугах положительны. Пусть Δ_1 - минимальная из остаточных пропускных способностей прямых дуг, а Δ_2 - минимальная из величин потоков обратных дуг. Тогда поток в сети можно увеличить на величину $\Delta = \min\{\Delta_1, \Delta_2\}$, прибавляя Δ к потокам на прямых дугах и вычитая Δ из потоков на обратных дугах. Очевидно, что при этом условие баланса (условие сохранения потока) $\sum\limits_{x_j \in S_{пр}(x_i)} \varphi(x_i, x_j) = \sum\limits_{x_j \in S_{сл}(x_i)} \varphi(x_i, x_j)$ для узлов, входящих в рассматриваемый маршрут, не нарушится.

Ясно, что если множество обратных дуг не пусто, то при такой процедуре увеличения потока в сети фактического перемещения объектов вдоль рассматриваемого маршрута не происходит, так как оно в принципе невозможно.



Однако эта процедура уменьшает потоки на некоторых дугах, которые, возможно, были перед этим насыщенными, образуя таким образом новые пути из ненасыщенных дуг, вдоль которых и происходит фактическое перемещение потока величины Δ.

Ясно также, что первая процедура является частным случаем второй.

4. Сетевое планирование.

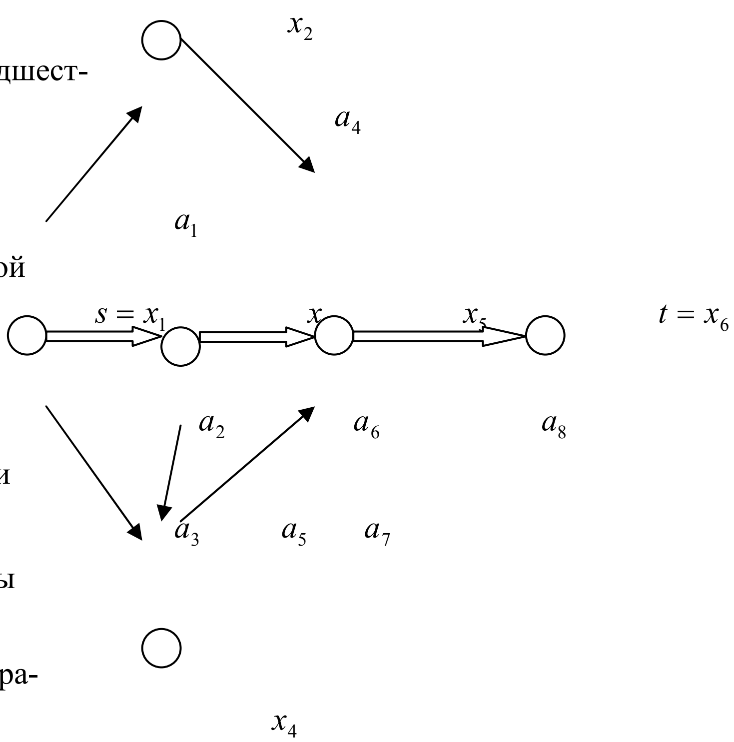
При планировании и управлении сложными комплексами работ используются их графические модели – сетевые графики. С математической точки зрения сетевой график – это связный орграф без петель и контуров. Основными понятиями сетевого планирования являются понятия работы и события.

Работа – это любые действия, сопровождающиеся затратами ресурсов и времени и приводящие к определенным результатам. Событие – это результат завершения одной или нескольких работ. Событие является предпосылкой для выполнения работ, следующих за ним. Любая работа на сети может быть определена двумя событиями, между которыми она находится. Событием может начинаться или заканчиваться несколько работ. Работы на сети изображают дугами, а события – вершинами сети.

Сетевой график обладает рядом особенностей, в частности он имеет только одно исходное событие (исток сети) и только одно завершающее событие – окончание всех работ. Рассмотрим пример построения сети по таблице последовательности работ.

Последовательность работ		
Исходная Работа	Опирается на работу	Продолжительность работ
a_1	-	3
a_2	-	6
a_3	-	4
a_4	a_1	5
a_5	a_2	1
a_6	a_2	9
a_7	a_3, a_5	6
a_8	a_4, a_6, a_7	8

Работы a_1, a_2 и a_3 не имеют предшествующих, поэтому реализация проекта начинается с этих работ, и изобразятся они дугами, выходящими из одной вершины – события x_1. Работе a_4 предшествует работа a_1, поэтому дуга a_4 на сети изображена вслед за дугой a_1. То же самое с дугами a_5 и a_6. Далее надо изобразить дуги a_7 и a_8. Работа a_7 опирается на работы a_3 и a_5. Итоговая работа a_8 опирается на a_4, a_6 и a_7. На рисунках



сети не рекомендуется во избежании путаницы изображать одновременно выполняемые работы параллельными дугами. Однако можно вводить дополнительные события и фиктивные работы (нулевой продолжительности), которые изображаются штриховыми линиями. Если бы, к примеру, работа a_5 опиралась бы еще на a_1, то между событиями x_2 и x_3 пришлось бы ввести штриховую дугу.

Имея сеть работ некоторого проекта можно посчитать время выполнения всего проекта и различных его частей, состоящих из разного набора работ. Для этого введем еще несколько определений. Определим сначала минимальное время, за которое можно выполнить все работы комплекса. Для этого найдем продолжительность $t(\mu_i)$ всех полных путей μ_i. В нашем случае таких путей четыре: $\mu_1: 1-2-5-6$; $\mu_2: 1-3-5-6$; $\mu_3: 1-4-5-6$; $\mu_4: 1-3-4-5-6$. Их продолжительности $t(\mu_1)=16$, $t(\mu_2)=23$, $t(\mu_3)=18$, $t(\mu_4)=21$. Наиболее продолжителен второй путь. Такой путь называют критическим. Этот путь определяет минимальное время выполнения всех работ комплекса. Минимальное время называют критическим сроком и обозначают $t_{кр.}$. Итак, в рассматриваемом примере $t_{кр.}=23$.

Все работы и события, лежащие на критическом пути, называют критическими, все остальные работы и события – некритическими. Задержка любой критической работы вызывает задержку выполнения всего комплекса. Следовательно, чтобы уменьшить время выполнения комплекса работ, надо сократить сроки критических работ. Некритические работы допускают некоторое запаздывание их выполнения без нарушения критического срока. Это запаздывание измеряется резервом времени событий и работ.

Свершением события называется момент, к которому заканчиваются все входящие в него работы и может быть начата любая выходящая работа. Некоторые события можно совершать в разные моменты, то есть варьировать свершение этих событий. Например, событие x_2 может свершиться через три дня (по окончании работы a_1), но может наступить и позже на срок до семи дней, поскольку на пути μ_1, где лежит это событие, есть резерв времени $t_{кр.}-t(\mu_1)=23-16=7$ дней. Поэтому для событий различают ранний и поздний сроки свершения.

Ранним сроком $t_p(x_j)$ свершения события x_j называется самый ранний момент времени, к которому завершатся все работы, предшествующие этому событию. Ранние сроки для всех событий могут быть рассчитаны по формуле

$$t_p(x_j)=\max_{(x_i,x_j)\in U_j^+}\left(t_p(x_i)+t(x_i,x_j)\right),$$

где U_j^+ - множество работ, входящих в x_j событие, $t_p(x_i)$ - ранний срок свершения начального события работы (x_i,x_j), $t(x_i,x_j)$ - продолжительность работы (x_i,x_j).

Поздним сроком $t_n(x_i)$ свершения события x_i называется самый поздний момент времени, после которого остается ровно столько времени, сколько необходимо для завершения всех работ, следующих за этим событием.

В нашем случае $t_n(x_6)=23$. Чтобы не нарушался критический срок, событие x_5 должно произойти в крайнем случае на восемь дней раньше, поэтому $t_n(x_5)=23-8=15$. Аналогично, $t_n(x_2)=15-5=10$. Таким образом, поздние сроки событий рассчитываются по формуле

$$t_n(x_i)=\min_{(x_i,x_j)\in U_j^-}\left(t_n(x_j)-t(x_i,x_j)\right),$$

где U_i^- - множество работ, выходящих из x_i события, $t_n(x_j)$ - поздний срок свершения конечного события работы (x_i, x_j).

Разности между поздним и ранним сроками свершения события x_i составляет

<u>резерв времени</u> $R(x_i)$ **этого события** $R(x_i) = t_n(x_i) - t_p(x_i)$.

Резерв показывает, на какой предельно допустимый срок может задержаться свершение события x_i без изменения срока наступления итогового события t. У критических событий ранние и поздние сроки совершения совпадают, ибо резерв времени у них равен нулю.

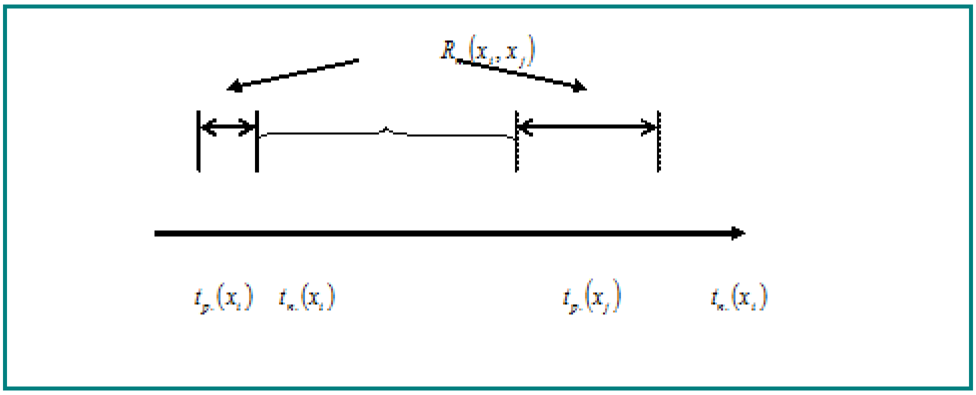
Зная сроки свершения событий, можно найти ранние и поздние сроки начала и окончания работы (x_i, x_j). Очевидно, что

$$\begin{cases} t_{р.н.}(x_i, x_j) = t_{p.}(x_i), \; t_{p.o.}(x_i, x_j) = t_{p.}(x_i) + t(x_i, x_j), \\ t_{п.о.}(x_i, x_j) = t_{п.}(x_j), \; t_{п.н.}(x_i, x_j) = t_{п.}(x_j) - t(x_i, x_j). \end{cases}$$

Для работ определяются два резерва времени. <u>Полный резерв времени работы</u> – это максимальное количество времени, на которое можно задержать начало работы или увеличить ее продолжительность, не нарушая критический срок

$$R_{п.}(x_i, x_j) = t_{п.}(x_j) - t_{p.}(x_i) - t(x_i, x_j).$$

Формулу (5.1.5) можно проиллюстрировать следующим рисунком.



Отдельные работы, помимо полного резерва, имеют <u>свободный резерв времени</u>, составляющий часть полного резерва, остающуюся после исключения резерва времени $R(x_j)$ конечного события x_j данной работы

$$R_{c.}(x_i, x_j) = t_{p.}(x_j) - t_{p.}(x_i) - t(x_i, x_j).$$

Свободный резерв времени – это запас времени, на который можно отсрочить начало работы или увеличить ее продолжительность при условии, что она начнется в свой ранний срок и при этом ранние сроки начала последующих работ не изменятся. Понятно, что все резервы критических работ равны нулю.

1.8 Лекция № 13, 14, 15 (6 часов).

Тема: «Основы теории булевых функций»

1.8.1 Вопросы лекции:

1. Булевы функции. Элементарные булевы функции.

2. Представление булевых функций формулами.

3. Минимизация булевых функций в классе ДНФ

1.8.2 Краткое содержание вопросов:

1. Булевы функции. Элементарные булевы функции.

Булевы функции, булевы константы. Булевыми функциями (или функциями алгебры логики или истинностными функциями) называются функции, значения которых равны 0 или 1 и аргументы которых принимают только два значения 0 и 1.

Булевы функции могут быть заданы специальными таблицами истинности или аналитически в виде специальных высказывательных форм, называемых иногда булевыми формами.

Выражения, содержащие одну или несколько переменных (аргументов), соединенных знаками логических операций, называются *логическим формами*. Высказывания, не содержащие ни одной переменной, называются константами. В логике, в отличие от арифметики, только две константы 0 - false и 1- true.

Напомним, что *форма называется числовой,* если при допустимом значении своих аргументов, она обозначает число (является числом). Булева форма является частным случаем числовой формы. Т.о. при помощи суперпозиции, исходя из логических операций над логическими переменными, можно строить сложные составные высказывания и затем вычислять их. Такого рода составные высказывания являются частным случаем так называемых булевых функций, которые являются предметом изучения математической логики. Обобщая все сказанное, можно дать определение булевых функций:

Булевыми функциями, называются предикаты, все аргументы которых определены на множестве {0, 1}, интерпретируемые как {ложь, истина}.

Можно сказать, что понятие булевой функции является частным случаем понятия предиката. Отличие состоит лишь в том, что у булевой функции четко фиксирована как область определения {0, 1}, так и область значений функции {0, 1}, в то время как у предиката четко фиксирована только одна область значений {0, 1}, в то время как область определения задана произвольным множеством.

В свою очередь понятие предиката является частным случаем понятия функции, отличие состоит в том, что у предиката четко фиксирована область значений {0, 1}, а у функции это может быть вся числовая ось.

2. Представление булевых функций формулами.

Булевы функции и формулы. **ФАЛ называются также *булевыми* функциями, *двоичными* функциями или *переключательными* функциями. Аргументы булевой функции являются булевыми переменными. Булеву функцию можно задать таблицей истинности.**

Утверждение Для булевой функции от n аргументов существует 2^n различных наборов аргументов.

Булева функция $f(x_1, x_2, \ldots, x_n)$ называется *полностью определенной*, если ее значения определены на всех 2^n наборах переменных. В противном случае функция *частично определенная*.

Функция $\boldsymbol{f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n)}$ *существенно зависит от переменной* x_i, (или переменная x_i – *существенная*), если $\exists$ такой набор значений $x_1, x_2, \ldots, x_n$ $(\boldsymbol{\sigma_1,\ldots,\sigma_{i-1},\sigma_i,\sigma_{i+1},\ldots,\sigma_n})$, что

$\boldsymbol{f(\sigma_1,\ldots,\sigma_{i-1},0,\sigma_{i+1},\ldots,\sigma_n) \neq f(\sigma_1,\ldots,\sigma_{i-1},1,\sigma_{i+1},\ldots,\sigma_n)}$. В противном случае переменная x_i – *несущественная (фиктивная)*.

1	2	1	2

Пусть две булевы функции заданы таблицей истинности. Для них переменная x_1 существенная, а x_2 – несущественна.

По определению булевы функции равны, если одна из другой получается введением или удалением несущественных переменных.

Одна и та же функция может иметь множество реализаций формулами над данным базисом (т.е. множеством логических операций). Формулы, реализующие одну и ту же функцию, называются *равносильными* (т.е. на всех наборах переменных их значение истинности совпадает). Отношение равносильности формул является отношением эквивалентности.

Формулы алгебры логики, при образовании которых используются только операции отрицания, конъюнкции и дизъюнкции, называются *булевыми формулами*.

Для любой формулы алгебры логики существует равносильная ей булева формула.

Способы представления булевых функций. Нормальные формы

Табличный способ определения истинности сложного выражения имеет ограниченное применение, т.к. с увеличением числа логических переменных число вариантов становится слишком большим. Тогда может быть использован способ приведения формул к *нормальной форме*.

Аналитическое выражение функции (или формула) находится в *нормальной форме*, если в ней отсутствуют знаки эквивалентности, импликации, двойного отрицания, а знаки отрицания находятся только при переменных.

Элементарной дизъюнкцией (*произведением*) называется дизъюнкция (произведение) переменных или их отрицаний, в котором каждая переменная встречается только один раз.

ДНФ – это дизъюнкция элементарных произведений. *КНФ* – это произведение элементарных дизъюнкций. Как ДНФ, так и КНФ функции не единственна. Обычно предполагают, что входящие в ДНФ (КНФ) элементарные конъюнкции (дизъюнкции) попарно различны.

ДНФ (КНФ) называется *совершенной*, если каждая переменная формулы входит в каждую элементарную конъюнкцию (дизъюнкцию) ровно один раз.

СДНФ (СКНФ) функции единственна.

Элементарные дизъюнкции: $x \vee \bar{y}$, z. Элемент. конъюнкции: $x \cdot \bar{y} \cdot z$, x. $f(x,y,z) = xyz \vee \bar{x}y$ – ДНФ ; $f(x,y,z) = (x \vee \bar{y}) \cdot z$ – КНФ.

Введем обозначения: $x^{\alpha} = \begin{cases} x, & \alpha = 1 \\ \bar{x}, & \alpha = 0 \end{cases}$

О разложении булевой функции по k переменным (знак $\cup \equiv \vee$**).**

$$f(x_1, \dots, x_n) = \bigvee_{\alpha_1, \dots, \alpha_k} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_k^{\alpha_k} \cdot f(\alpha_1, \alpha_2, \dots, \alpha_k, x_{k+1}, \dots, x_n)$$

n=3, k=2.

<u>Доказательство:</u>

Выберем какой-либо набор значений для переменных $x_1, \dots, x_n$. Пусть это будет $\sigma_1, \dots, \sigma_n$.. Заметим, что $\sigma_i^{\alpha_i} = \begin{cases} 1, & \sigma_i = \alpha_i \\ 0, & \sigma_i \neq \alpha_i \end{cases}$ ($1^1=1$, $0^0=1$, $1^0=\neg 1=0$, $0^1=0$)

Подставим в правую часть формулировки теоремы вместо $x_1, \dots, x_n$ набор $\sigma_1, \dots, \sigma_n$. Получим $\bigvee_{(\alpha_1, \dots, \alpha_k)} \sigma_1^{\alpha_1} \sigma_2^{\alpha_2} \dots \sigma_k^{\alpha_k} \cdot f(\alpha_1, \alpha_2, \dots, \alpha_k, \sigma_{k+1}, \dots, \sigma_n)$. Поскольку коэффициент перед функцией равен 1 только при равных значениях σ_i и α_i, в разложении останется только один член: $\sigma_1^{\alpha_1} \dots \sigma_k^{\alpha_k} \cdot f(\alpha_1, \dots, \alpha_k, \sigma_{k+1}, \dots, \sigma_n)$, и $\sigma_i = \alpha_i$, т.е. $f(\sigma_1, \dots, \sigma_k, \sigma_{k+1}, \dots, \sigma_n)$. Получена левая часть формулы теоремы 4.6. Поскольку набор был выбран произвольно, получаем, что утверждение верно $\forall$ набора $x_1, \dots, x_n$. ■

<u>Следствие 1</u>: Разложение Шеннона

$$f(x_1, x_2, \dots, x_n) = \bar{x}_1 \cdot f(0, x_2, \dots, x_n) \vee x_1 \cdot f(1, x_2, \dots, x_n)$$

<u>Следствие 2</u>: При k=n получаем: $f(x_1, \dots, x_n) = \bigvee_{f=1} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$, т.е. выбираем те слагаемые, на которых функция равна 1. Полученная формула представляет собой *СДНФ*.

Построение совершенных нормальных форм

<u>Построение СДНФ</u>

<u>1</u>. <u>Построение по ТИ</u>.

Найти строки в ТИ, где $f = 1$.

1) $\forall$ найденному набору $\sigma_1, \ldots, \sigma_n$. поставить в соответствие произведение $\tilde{x}_1 \cdot \tilde{x}_2 \cdot \ldots \cdot \tilde{x}_n$, где $\tilde{x}_i = \begin{cases} x_i, & \text{если } \sigma_i = 1 \\ \overline{x_i}, & \text{если } \sigma_i = 0 \end{cases}$

2) Составить дизъюнкцию из произведений п.2.

2. Получение из ДНФ.

Если некоторое произведение ДНФ не содержит какой-либо переменной, то необходимо домножить это произведение на дизъюнкцию этой переменной и ее отрицания и применить дистрибутивный закон.

Построение СКНФ

1. Построение по ТИ.

Найти строки в ТИ, где $f = 0$.

1) $\forall$ найденному набору $\sigma_1, \ldots, \sigma_n$. поставить в соответствие дизъюнкцию $\tilde{x}_1 \vee \tilde{x}_2 \vee \ldots \vee \tilde{x}_n$, где $\tilde{x}_i = \begin{cases} x_i, & \text{если } \sigma_i = 0 \\ \overline{x_i}, & \text{если } \sigma_i = 1 \end{cases}$

2) Составить произведение дизъюнкций из п.2.

2. Получение из КНФ.

Если некоторая элементарная дизъюнкция КНФ не содержит какой-либо переменной, то необходимо дизъюнктивно добавить в нее произведение этой переменной и ее отрицания и применить дистрибутивный закон.

3. Минимизация булевых функций в классе ДНФ

Минимизация булевых функций

Минимальная ДНФ – это такая ДНФ функции, которая содержит наименьшее количество вхождений переменных по сравнению с остальными.

Элементарная конъюнкция называется *импликантой функции* $f(x_1, \ldots, x_n)$, если она равна 0 на тех наборах, на которых f обращается в 0. *Простой импликантой* называется импликанта, в которой отбрасывание любой буквы ведет к получению элементарной конъюнкции, которая не является импликантой (т.е. никакая часть простой импликанты сама импликантой не является). Каждая импликанта соответствует покрытию на карте Карно, а простая импликанта – покрытию наибольшей размерности.

1) $f(x_1, x_2, x_3) = x_1 x_2 \vee \overline{x_1 x_2 x_3} \vee x_1 x_2 x_3$. $\overline{x_1 x_2 x_3}$ – импликанта, причем простая; $x_1 x_2 x_3$ – импликанта, но не простая, т.к. удаление x_3 снова дает импликанту $x_1 x_2$ (которая является простой).

2) Найдем импликанты и простые импликанты для функции $f(x_1, x_2) = x_1 \to x_2$. Всего имеется 8 элементарных конъюнкций с переменными x_1, x_2. Приведем их таблицы истинности.

1	2	x_1 $\to x_2$	.	.	.	$_1x_2$			1	2

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

Из таблицы истинности заключаем, что $\bar{x}_1\bar{x}_2$, $\bar{x}_1 x_2$, x_1x_2, $\bar{x}_1$, x_2 являются импликантами функции *f*. Из них простыми являются $\bar{x}_1$ и x_2.

Дизъюнкция всех простых импликант функции называется *сокращенной ДНФ.* Сокращенная ДНФ функции единственна.

Сокращенная ДНФ может содержать лишние импликанты, удаление которых не меняет значения функции.

Если из сокращенной ДНФ удалить все лишние дизъюнктивные члены, и удаление любого из оставшихся приведет к изменению значения функции, то такая форма называется *тупиковой* ДНФ. Та из всех тупиковых ДНФ, которая имеет наименьшее число вхождений переменных, является *минимальной* ДНФ.

Процесс нахождения минимальной ДНФ из СДНФ можно разбить на следующие этапы:

1) нахождение сокращенной ДНФ (она единственна);
2) нахождение всех тупиковых ДНФ (их м.б. несколько);
3) выбор из всех тупиковых минимальной ДНФ (их тоже м.б. несколько).

Известны аналитические и графические способы построения минимальной ДНФ. Графический способ использует представление на картах Карно.

1.9 Лекция № 16 (2 часа).

Тема: «Элементы теории алгоритмов»

1.9.1 Вопросы лекции:

1. Основные подходы к формализации понятия алгоритма (Машина Тьюринга. Рекурсивный алгоритм, нормальные алгоритмы Маркова).

2. Понятие эффективности и сложности алгоритмов

1.9.2 Краткое содержание вопросов:

1. Основные подходы к формализации понятия алгоритма (Машина Тьюринга. Рекурсивный алгоритм, нормальные алгоритмы Маркова).

Формализация понятия алгоритма. Универсальные модели алгоритмов.

Интуитивное понятие алгоритма обладает целым рядом недостатков. Очевидно, что такие понятия, использованные при описании общих свойств алгоритмов, как элементарность шагов, сами нуждаются в уточнении. Очевидно, что их словесные определения будут содержать новые понятия, которые снова потребуют уточнения и т.д. Начиная с 30-х годов, было предложено несколько уточнений понятия алгоритма. Считается, что все они достаточно полно отражают основные черты интуитивного понятия алгоритма. Действительно, все формальные определения алгоритма в некотором смысле эквивалентны друг другу. Поэтому в теории алгоритмов применяется другой подход: выбирается конечный набор исходных объектов, которые объявляются элементарными и конечный набор способов построения их них новых объектов. Этот метод был уже использован в теории множеств и получил название *конструктивного подхода*.

Алгоритмические модели, которые претендуют на право считаться формализацией понятия «алгоритм», должны быть универсальными, т.е. допускать описание любых алгоритмов.

Можно выделить три основных типа универсальных алгоритмических моделей, различающихся исходными эвристическими соображениями относительно того, что такое алгоритм. Первый тип связывает понятие алгоритма с наиболее традиционными понятиями математики – вычислениями и числовыми функциями. Наиболее развитая и изученная модель этого типа – рекурсивные функции – является исторически первой формализацией понятия алгоритма.

Второй тип модели связан с развитием вычислительной техники и основан на представлении об алгоритме как о некотором детерминированном устройстве, способном выполнять в каждый отдельный дискретный момент времени весьма примитивные операции. Такое представление не оставляет сомнений в однозначности алгоритма и элементарности его шагов. Кроме того, эвристика этой модели близка к ЭВМ и, следовательно, к инженерной интуиции. Основной теоретической моделью этого типа является созданная в 30-х годах концепция машины Тьюринга. Именно машина Тьюринга явилась моделью современной ЭВМ и способствовала развитию современной вычислительной техники.

Наконец, третий тип алгоритмических моделей – это преобразование слов в произвольных алфавитах, в которых элементарными операциями являются подстановки, т.е. замены части слова (подслова) другим словом. Преимущества этого типа моделей заключаются в максимальной абстрактности и возможности применить понятие алгоритма к объектам произвольной, не обязательно числовой природы. Примерами моделей этого типа являются канонические системы Поста и нормальные алгоритмы Маркова. При этом общность формализации в конкретной модели не теряется и доказывается сводимость одних моделей к другим, т.е. показывается, что всякий алгоритм, описанный средствами одной модели, может быть описан средствами другой.

Благодаря взаимной сводимости моделей в общей теории алгоритмов удалось выработать инвариантную по отношению к моделям систему понятий, позволяющую говорить о свойствах алгоритмов независимо от того, какая формализация алгоритма выбрана. Эта система понятий основана на понятии вычислимой функции, т.е. функции, для вычисления которой существует алгоритм.

МАШИНА ТЬЮРИНГА.

В 1935 г. возникло такое положение: свойства, обнаруженные у некоторого точно определенного класса вычислимых теоретико-числовых функций, изучавшихся Чёрчем и Клини в 1932—1935 гг. и названных "λ-определимыми функциями", упорно подсказывали мысль, что этот класс, может быть, охватывает все функции, которые в соответствии с нашим интуитивным представлением можно рассматривать как вычислимые. При этих обстоятельствах Чёрч выдвинул тезис (опубликован в 1936 г.), что все функции, которые интуитивно мы можем рассматривать как вычислимые, или, говоря его словами, как «эффективно вычислимые», являются λ-определимыми, или, эквивалентным образом, общерекурсивными.

Несколько позже, но независимо появилась статья Тьюринга (1936), в которой был введен еще один точно определенный класс интуитивно вычислимых функций, которые мы будем называть «функциями, вычислимыми по Тьюрингу», и относительно этого класса было высказано такое же утверждение; это утверждение мы называем *тезисом Тьюринга.* Вскоре Тьюрингом [1937] было показано, что его вычислимые функции — это то же самое, что λ-определимые функции, и, следовательно, то же самое, что и общерекурсивные функции. Поэтому тезисы Тьюринга и Чёрча эквивалентны. Мы будем обычно ссылаться на оба эти тезиса как на *тезис Чёрча,* а в связи с тем его вариантом, в котором идет речь о «машинах Тьюринга»,— как на *тезис Чёрча — Тьюринга.* В 1936 г. Пост независимо от Тьюринга опубликовал в довольно сжатом изложении формулировку, в основе ту же, что у Тьюринга. В 1943 г., основываясь на своей неопубликованной работе 1920—1922 гг., он опубликовал третий эквивалент аналогичного тезиса. Еще одну эквивалентную формулировку дает теория алгоритмов Маркова [1951г].

Область использования машины Тьюринга

Понятие *машины Тьюринга* возникает в результате прямой попытки разложить интуитивно известные нам вычислительные процедуры на элементарные операции: Тьюринг привел ряд доводов в пользу того, что повторения его элементарных операций было бы достаточно для проведения любого возможного вычисления. Поэтому машина Тьюринга (МТ) используется:

1) если требуется доказать *возможность* алгоритмической реализации вычислительной функции;

2) если требуется *оценить вычислительную сложность* или *трудоемкость* решения задачи по данному алгоритму, т.е. время выполнения алгоритма.

Для этого мы моделируем работу произвольного алгоритма в терминах рассматриваемой задачи. Затем *определяется* класс машин-вычислителей, которые могут решить данную задачу – формально описываются правила работы машины, исходные данные, ограничения и т.д. (поскольку в определении задачи ничего не говорится о программах так таковых в привычном для нас понимании, то алгоритмическая *разрешимость* или *неразрешимость*, сводится к проблеме остановки произвольного алгоритма решения задачи). В качестве машины-вычислителя выберем машину Тьюринга, поскольку ранее было показано, что всякая вычислимая функция реализуема на МТ и сведем решение данной задачи к существующим группам задач, для которых известно, что они решаются на МТ.

Принцип работы машины Тьюринга.

Какая именно команда программы будет выполняться в данный момент, определяется двумя параметрами: читаемым головкой символом и состоянием машины.

Результатами выполнения команды являются: новый символ, записанный на ленту в ту ячейку, напротив которой находится в данный момент головка; перемещение головки на одну позицию (ячейку) вправо или влево вдоль ленты; переход машины в новое состояние. В частных случаях новый символ может быть равен старому, перемещение может отсутствовать, состояние может остаться прежним.

Формат команды имеет следующий вид:

a q b r D,

где *a* — читаемый символ; *q* — текущее состояние; *b* — символ записываемый в обозреваемую ячейку ленты вместо символа *a; r* — новое состояние; *D* — направление движения головки машины относительно ленты.

Символы выбираются из конечного алфавита $A = \{a_1, \ldots, a_l\}$.

В дальнейшем будем использовать трехсимвольный алфавит {*e* 0, 1}, причем *e* будет означать «пустой (empty)» символ — отсутствие информации в ячейке, а с помощью нуля и единицы будут кодироваться все данные. Иногда используют двухсимвольный алфавит $A = \{e, 1\}$. В этом случае числа кодируются только единицами: нуль кодируется одной единицей, число один кодируется двумя единицами, а число x кодируется $x + 1$ единицами Это — единичная система счисления. Однако она плоха с точки зрения сложности задач (см. гл. 5).

Множество состояний обозначим $Q= \{ q_1, \ . \ . \ . \ , q_k\}$. Направление движения D выбирается из множества $\{L, R, S\}$ где L — движение влево, R — движение вправо, S — отсутствие движения.

Таким образом, команда $1\ q_3\ 0\ q_6\ L$ означает: если, находясь в состоянии q_3, машина Тьюринга обозревает ячейку ленты в которой записана 1, то машина должна записать в эту ячейку 0, произвести сдвиг головки относительно ленты влево на одну ячейку и перейти в состояние q_6.

Это описание действия, соответствующего команде говорит о том, что команда может рассматриваться как отображение пар (a, q) в тройки (b, r, D), т. е. отображение

$$AxQ => AxQx\ \{L, R, S\}.$$

Данное отображение является частичным, так как не для любой пары-аргумента существует тройка-результат. Но для произвольной пары существует не более одной тройки, т. е. отображение не является многозначным.

Все действия производятся в дискретном времени. Иначе говоря, можно рассматривать целочисленные моменты времени t = = 0, 1, 2, 3, . . . Любое изменение происходит мгновенно в момент $t = i$ *и* ничего не меняется между двумя соседними моментами времени.

Работает машина Тьюринга следующим образом. Стартовая конфигурация: на ленте находятся исходные данные — строка символов в алфавите A, состояние внутренней памяти соответствует некоторому оговоренному (всегда одному и тому же) начальному состоянию, например, q_1. При этом головка машины обозревает некоторую ячейку ленты с записанным там символом a. Нормальным считается начальное положение головки напротив самого левого непустого символа, т. е. не совпадающего с e.

Момент старта рассматривается как нулевой момент времени. В момент старта выполняется первая команда, это единственная команда, начинающаяся с пары (a, q_1). В результате выполнения команды машина перейдет в новое состояние, и головка машины прочтет новый символ с ленты. Эта пара (новый символ, новое состояние) станет начальной частью следующей команды и т. д. Машина будет продолжать работать в дискретном времени, шаг за шагом переходя из состояния в состояние, и постепенно изменяя содержимое ленты. Наконец, для некоторой пары (a, q) не окажется команды в программе. Такая ситуация считается завершающей. Машина прекращает функционирование. Оставшаяся запись на ленте считается записью результата.

Таким образом, машина Тьюринга реализует вычисление некоторой функции — отображения исходной строки символов в результирующую строку.

Существует несколько способов представления программы машины Тьюринга (множества команд). Два наиболее употребительных:

1) двумерная таблица (рис. 1.2);

2) диаграмма (нагруженный псевдограф).

В двумерной таблице строки помечаются различными символами алфавита, а столбцы — именами различных состояний машины, т. е. таблица имеет размер *Ik*. Каждой команде программы

Со-	q	..	q	..	q
a_1					
			b		
a1					

Рис. 1.2. Табличная форма программы машины Тьюринга

соответствует единственная клетка в таблице. Она определяется для команды *a q b r D* следующим образом: в клетку, находящуюся на пересечении строки, помеченной символом я, и столбца помеченного состоянием *q.* вписывается тройка b *r D.*

Для некоторых пар *(a, q)* в программе нет команд, следовательно, соответствующие клетки таблицы остаются пустыми. При достижении в процессе работы пустой клетки машина Тьюринга останавливается.

В качестве простого примера приведем программу вычисления функции S(x) = x + 1, т. е. увеличение аргумента на единицу (рис. 1.3). Используем алфавит A = *{e,* 0, 1}, причем x будем кодировать последовательностью нулей и единиц так. как это принято при двоичном кодировании целых неотрицательных чисел.

предположим также, что в момент старта головка машины Тьюринга находится напротив крайней левой ячейки с символом 1**.**

	q	q	q_3	q_4
	0	1	0	
	1	0	1	
	e	1	e	

Р и с. 1.3. Программа машины Тьюринга для вычисления функции S(x)=x+1

Первая выполняемая команда —$1q_11q_1R$ оставляет 1 в ячейке ленты, оставляет неизменным состояние q_1 и производит сдвиг головки вправо по ленте. В новой читаемой ячейке может оказаться любой из трех символов алфавита. Если это 0 или 1, то производится дальнейшее движение вправо до окончания кода числа *x.* Если же встретится символ *e,* то это будет означать, что код числа закончился и головка находится справа от младшей цифры кода числа. После этого, собственно, и начинается процесс прибавления единицы. Если младшая цифра — 0 то достаточно заменить се на 1 (команда 0 q_2 1 q_3 *L)* и начать обратное движение к исходной позиции. Если младшая цифра — 1 то результатом в данной ячейке будет 0 (сложение по mod 2), и единица перейдет в следующий по старшинству разряд (влево). Процесс распространения переноса может закончиться где-то внутри кода числа и тогда необходимо осуществить «прокрутку» ленты так, чтобы машина остановилась на крайней левой единице кода результата (x+1)

Второй пример - программа вычисления функции *Z(x)* = 0(x) = 0, превращающей запись любого аргумента. x в запись нуля (рис. 1.4). Эта программа стирает с ленты код x, т. е. запол

	q_1	q_2
0	eq_1R	
1	eq_1R	
e	$0q_2R$	

Р 11 с. 1.4. Программа машины Тьюринга для вычисления функции 0(x) = 0

няет клетки символом *e* и перед остановкой записывает в текущую клетку 0.

Более длинная программа получается для вычисления функции I^n_m ($x_1,x_2, \dots , x_n$) выбирающей m-й аргумент из последовательности *n* аргументов, $1 \le m \le n$, I^n_m (x_1, x_2, ... ,x_n)=x_m (рис. 1.5).

Представление последовательности *n* аргументов зададим на ленте в виде записанных один за другим через разделитель — пустую клетку *e* — двоичных кодов x_i. Программа, написанная для конкретного значения *m,* действует следующим образом. Сначала стирается (заменяется на *e ...e)* первый аргумент, затем стирается второй, ..., стирается *(m* - 1)-й; затем подтверждается *m-й* аргумент; затем стираются оставшиеся аргументы.

Рис. 1.5. Программа машины Тьюринга для вычисления функции

$I^n_m(x_1, x_2, \dots , x_n)$

Другой способ представления программы машины Тьюринга — диаграмма (рис. 1.6).

Диаграмма представляет собой геометрический объект, состоящий из вершин (обозначаемых точками или окружностями) и дуг (рисуемых в виде направленных отрезков прямой со стрелкой на одном из концов или в виде отрезков несамопересекающихся кривых). Каждой вершине приписывается состояние машины Тьюринга: таким образом вершин в диаграмме ровно столько, сколько имеется состояний. Дуге, соединяющей две вершины q_i и q_j, приписывается некоторый символ *a* алфавита *A* и двойка *b D* так, что запись $a\ q_i\ b\ q_j\ D$ образует команду программы машины Тьюринга.

Дуга (стрелка) символизирует переход из состояния q_i в состояние q_j при условии, что головка читает символ *a.* Одновременно с этим символ *a* заменяется на символ *b* и совершается движение *D*.

Программа вычисления *Z(x)* = O может быть изображена диаграммой, изображенной на рис. 1.7.

Алан Тьюринг сформулировал тезис, связывающий понятие алгоритма и машины: *«Для всякого (неформального) алгоритма может быть построен Тьюрингов алгоритм (программа машины Тьюринга), дающий при одинаковых исходных данных тот же результат».*

Это недоказуемое математическими методами утверждение играет важную роль при проектировании программного обеспечения, особенно на начальных этапах проектирования. Первоначальная постановка задачи зачастую является словесной, неформальной. Если ее решение удается описать в виде конечной последовательности шагов, каждый из которых достаточно прост, то в соответствии с Тезисом Тьюринга это означает, что может быть написана программа на каком-либо алгоритмическом языке, решающая поставленную задачу.

Рекурсивные функции.

Всякий алгоритм однозначно ставит в соответствие исходным данным (в случае если определен на них) определенный результат. Поэтому с каждым алгоритмом однозначно связана функция, которую он вычисляет. Исследование этих вопросов привело к созданию в 30-х годах прошлого века теории рекурсивных функций. В этой теории, как и вообще в теории алгоритмов принят конструктивный, финитный подход, основной чертой которого является то, что все множество исследуемых объектов (в данном случае функций) строится из конечного числа исходных объектов – базиса – с помощью простых операций, эффективная вычислимость которых достаточна очевидна. Операции над функциями будем называть *операторами*.

Будем рассматривать только числовые функции, т.е. функции, аргументы и значения которых принадлежат множеству натуральных чисел N (в теории рекурсивных функций полагают N=0, 1, 2, …). Иначе говоря, числовой n-местной функцией $f(x_1, x_2, \ldots, x_n)$ называется функция, определенная на некотором подмножестве $N \subseteq N^n$ с натуральными значениями. Если область определения совпадает с множеством N^n, т.е. $f\colon N^n \to N$, то говорят, что функция f всюду определенная, в противном случае – частично определенная.

Например: $f(x, y) = x + y$ – всюду определенная двуместная функция.

$f(x, y) = x - y$ – частично определенная функция (она определена при $x \geq y$).

Рекурсивным определением функции принято называть такое определение, при котором значения функции для данных аргументов определяются значениями функции для более простых аргументов (уже вычисленных) или значениями более простых функций.

Простейшим примером рекурсивного определения являются числа Фибоначчи, представляющие собой последовательность чисел $f(n)$, удовлетворяющих условиям

$f(0) = 1, \quad f(1) = 1, \quad f(n+2) = f(n) + f(n+1),$

1) $0(x)=0$ – нуль-функция.
2) $S(x)=x+1$ – функция следования.
3) $I_m(x_1, x_2, \ldots, x_n) = x_m$, где $m = 1, \ldots, n$ – проектирующая функция.

Оператор суперпозиции. Суперпозиция является мощным средством получения новых функций из уже имеющихся. Напомним, что суперпозицией называется любая подстановка функций в функции. Оператором суперпозиции P_m^n называется подстановка в функцию от m переменных m функций от n одних и тех же переменных. Например, для функций $h(x_1, x_2, \ldots, x_m), g_1(x_1, x_2, \ldots, x_n), \ldots, g_m(x_1, x_2, \ldots, x_n)$

Оператор примитивной рекурсии. Оператор примитивной рекурсии R_n определяет $(n+1)$ -местную функцию f через n-местную функцию g и $(n+2)$ -местную функцию h следующим образом:

1. Функции $0(x)$, $S(x)$ и $I_m^n(x)$ для всех натуральных n, m, где $m \leq n$, являются примитивно рекурсивными.

2. Если $g_1(x_1, x_2, \ldots, x_n), \ldots, g_m(x_1, x_2, \ldots, x_n), h(x_1, x_2, \ldots, x_n)$ примитивно рекурсивные, то $P_m^n(h, g_1, \ldots, g_m)$ – примитивно-рекурсивные функции для любых натуральных n, m.

3. Если $g_1(x_1, \ldots, x_n)$ и $h(x_1, \ldots, x_n, y, z)$ – примитивно рекурсивные функции, то $R_n(g, h)$ – примитивно-рекурсивная функция.

4. Других примитивно-рекурсивных функций нет.

1.10 Лекция № 17 (2 часа).

Тема: «Конечные автоматы»

1.10.1 Вопросы лекции:

1. Понятие конечного автомата. Историческая справка. Способы задания конечного автомата. Примеры конечных автоматов. Виды автоматов. Общие задачи теории автоматов

1.10.2 Краткое содержание вопросов:

1. Понятие конечного автомата. Историческая справка. Способы задания конечного автомата. Примеры конечных автоматов. Виды автоматов. Общие задачи теории автоматов

2. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

2.1 Практическое занятие № 1, 2, 3 (6 часов).

Тема: «Множества и бинарные отношения».

2.1.1 Задание для работы:

1. Множества и операции над ними. Диаграммы Венна-Эйлера.

2. Элементы алгебры множеств.

3. Бинарные отношения и их свойства.

2.1.2 Краткое описание проводимого занятия:

1. Множества и операции над ними. Диаграммы Венна-Эйлера.

Задание 1.

1.Перечислите элементы следующих множеств:

Аудиторные *Для самостоятельного выполнения*

а) $A=\{x: x\in Z, 10\le x\le 18\}$ ……………… $A=\left\{x: x\in Z, \dfrac{1}{x^2}\ge\dfrac{1}{16}\right\}$;

б) $B=\{x: x\in Z, 6x^2+x-1=0\}$ ………….. $B=\{x: x\in R, 6x^2+x-1=0\}$.

2.Описать множества с помощью предикатов:

а) $C=\{2,5,8,11,...\}$ …………………….. $C=\left\{1,\dfrac{1}{3},\dfrac{1}{7},\dfrac{1}{15},...\right\}$.

3. $U=\{1,2,3,4,5,6,7,8\}$ -универсальное множество,

$A=\{1,2,3,4\}, B=\{3,5,7\}, C=\{1,4,5,6\}$. Найти элементы множеств:

а) $B\cap C, (A\cup B)\cap(A\cap C)$ ………………… $A\cup C$, $A\cap B\cap C$,

б) $B\setminus C, \overline{A\cup B}$ …………………………... $B\Delta C, \overline{C}$.

4. $A=\{3n: n\in Z, n\ge 4\}, B=\{2n: n\in Z\}, C=\{n: n\in Z, n^2\le 100\}$. С помощью операций на множествах выразить через A,B,C следующие множества:

а) $\{\pm1,\pm3,\pm5,...\}$ ……………………………$\{6n: n\in Z, n\ge 2\}$

б) $\{-9,-7,-5,-3,-1,0,1,3,5,7,9\}$ ……………… $\{-10,-8,-6,-4,-2,0,2,4,6,8,10\}$.

2. Элементы алгебры множеств.

5.Проиллюстрируйте диаграммами Венна тождества:

а) закон дистрибутивности $A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$,

б) $A\cap(B\Delta C)=(A\cap B)\Delta(A\cap C)$.

6.Докажите тождества:

а) $(\overline{A\cap\overline{B}})\cup B=\overline{A}\cup B$ …………………….. $(A\setminus B)\setminus C=A\setminus(B\cup C)$.

7.Найти булеан $P(A)$ множества A:

а) $A=\{a,b,c\}$ .... $A=\{2,5,8,9\}$.

Задание 2.

1.Пусть A, B, C - произвольные конечные множества. Доказать:

а) $A\times(B\cap C)=(A\times B)\cap(A\times C)$,

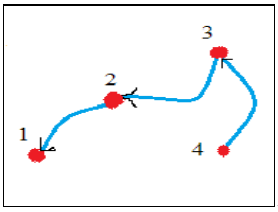
б) $(A\cup B)\times C=(A\times C)\cup(B\times C)$.

2. $U=\{1,2,3,4,5\}$, $A=\{1,3,5\}$, $B=\{3,4\}$. Найти характеристические векторы подмножеств A, B, по ним найти характеристические векторы множеств а) $A\cup B$, б) $A\cap B$, в) $\overline{B}$ и перечислить элементы этих множеств.

3. Бинарные отношения и их свойства.

Задание 3.

1. На множестве $A=\{1,2,3,4\}$ отношение R, данное перечислением пар $R=\{(2,1),(3,2),(4,3)\}$, изобразить графом и задать матрицей.

Решение. 

.

Матрица отношения равна $R=\begin{pmatrix}0&0&0&0\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{pmatrix}$.

2. Отношение на паре $A=\{1,2,3\}, B=\{a,b,c,d\}$ множеств задано матрицей $\begin{pmatrix}1&0&1&0\\1&0&1&0\\0&1&1&0\end{pmatrix}$. Описать отношение перечислением пар и изобразить орграфом.

3. Отношения на множестве натуральных чисел N заданы предикатами:

а) $R=\{(x,y):2x+y=9\}$; б) $S=\{(x,y):x+y<7\}$; в) $T=\{(x,y):y=x^2\}$.

Задать эти отношения перечислением пар.

4. На множестве $A=\{1,2,3,4\}$ отношение определено предикатом

$R = \{(x, y): x + 2y = 2n - 1, n \in A\}$. Представить R каждым из способов:

а) в виде множества упорядоченных пар;

б) графом;

в) матрицей.

5. Указать, какие из следующих отношений на Z являются рефлексивными, симметричными, транзитивными?

а) $x + y$ – нечётное число; б) $x + y$ – чётное число; в) $x \cdot y$ – нечётное число;

г) $x + x \cdot y$ – чётное число.

6. На множестве Z заданы отношения:

а) $xRy \Leftrightarrow x - y$ – чётное; б) $xTy \Leftrightarrow x, y$ – при делении на модуль $m = 5$ имеют одинаковые остатки.

Выяснить, являются ли R и T отношениями эквивалентности и если являются, то найти разбиения на классы эквивалентных элементов, фактор-множества, индексы разбиения.

2.1.3 Результаты и выводы: в результате проведенного занятия студенты:
- освоили понятия об основных операциях с множествами и алгебре множеств, бинарных отношениях;
- приобрели умения и навыки выполнения операций с множествами, построения бинарных отношений;

2.2 Практическое занятие № 4 (2 часа).

Тема: «Основные алгебраические структуры»

2.2.1 Задание для работы:

1. Понятие бинарного отношения. Способы задания отношений. Свойства отношений, классификация отношений.

2. Отношения эквивалентности и порядка. Отношения Парето. Принятие решений при многих критериях.

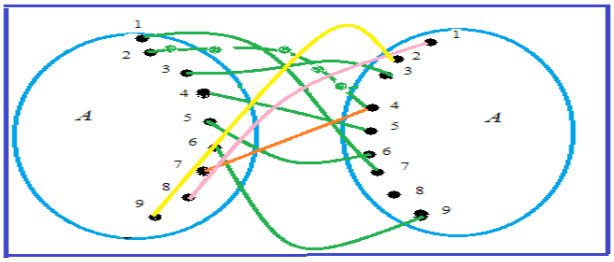
2.2.2 Краткое описание проводимого занятия:

1. Понятие бинарного отношения. Способы задания отношений. Свойства отношений, классификация отношений.

1. Даны пары $\langle x,y\rangle \in \rho$, причем $x \in \{1,...,9\}$, $y \in \{1,...,9\}$

Является ли отношение ρ функцией? Инъективной функцией? Сюръективной функцией? Биективной функцией? Обосновать.

Решение. В первой задаче удобно изобразить графически данное отношение на A^2, $A = \{1,2,3,4,5,6,7,8,9\}$:



Отношение ***является функцией*** на множестве А , т.к. каждый элемент множества А находится в отношении только с одним элементом (из каждой точки левого круга выходит только одна стрелка). Отношение ***не является сюръекцией***, т.к. элемент 8 не имеет прообраза и ***не является инъекцией***, т.к. два разных элемента (2, 7) имеют один и тот же образ 4; такая функция ***не является биекцией.***

2. Отношения эквивалентности и порядка. Отношения Парето. Принятие решений при многих критериях.

2. Является ли следующее отношение рефлексивным? Симметричным (антисимметричным)? Транзитивным? Отношением эквивалентности или частичного порядка? Линейного порядка? Обосновать.

$$R=\left\{(x,y): x=mq_1+r,\ y=mq_2+r,\ q_i\in Z,\ m\geq 0,\ m\in Z\right\},\ m=3$$

Решение. Целые числа x, y находятся в данном отношении тогда и только тогда, когда они имеют одинаковые остатки при делении на модуль m . Отношение

-рефлексивно, т.к. два одинаковых числа имеют одинаковые остатки,

-симметрично,

-транзитивно, т.к. если x, y имеют одинаковые остатки, y, z имеют одинаковые остатки, то у чисел x, z остатки одинаковые.

Поэтому данное отношение является отношением эквивалентности. Оно разбивает Z на классы эквивалентных элементов, называемых классами вычетов целых чисел по модулю $m=3$. Множество таких классов (здесь 3) называется фактор-множеством А по данному отношению:

$[0]=\{\ldots,-6,-3,0,3,6,\ldots\}$, $[1]=\{\ldots,-5,-2,1,4,7,\ldots\}$, $[2]=\{\ldots,-4,-1\,2,5,8,\ldots\}$.

2.2.3 Результаты и выводы:

В результате проведенного занятия студенты:

- освоили понятия бинарного отношения, свойства и классификацию отношений;
- приобрели умения и навыки классификации отношений, выявления свойств отношений.

2.3 Практическое занятие № 5, 6 (4 часа).

Тема: «Элементы теории чисел»

2.3.1 Задание для работы:

1. Биекции. Эквивалентные множества.

2. Мощность множеств, счётные множества. Мощность континуума.

2.3.2 Краткое описание проводимого занятия:

1. Биекции. Эквивалентные множества.

1. Даны пары $<x,y>\in\rho$, причем $x\in\{1,...,9\}$, $y\in\{1,...,9\}$

Является ли отношение ρ функцией? Инъективной функцией? Сюръективной функцией? Биективной функцией? Обосновать. Будут ли множества

Решение.

1. В первой задаче удобно изобразить графически данное отношение на

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

Отношение ***является функцией*** на множестве А , т.к. каждый элемент множества А находится в отношении только с одним элементом (из каждой точки левого круга выходит только одна стрелка). Отношение ***не является сюръекцией***, т.к. элемент 8 не имеет прообраза и ***не является инъекцией***, т.к. два разных элемента (2, 7) имеют один и тот же образ 4; такая функция ***не является биекцией.***

2. Мощность множеств, счётные множества. Мощность континуума.

2. Что можно утверждать относительно приведенных множеств ($|A|=|B|$, $|A|<|B|$ или $|A|>|B|$)? Обосновать.

$$A=\{2,4,6,8,...\}\quad,\quad B=\left\{b\,\middle|\,b=\frac{2}{n},\ n\in N\right\}$$

Решение. **5.**Каждое из множеств А и В эквивалентно множеству натуральных чисел N. Поэтому $|A|=|B|=|N|=a$, т.е. множества A,B счётные и справедливо утверждение $|A|=|B|$.

2.3.3 Результаты и выводы:

В результате проведенного занятия студенты:

- освоили понятия биекции и эквивалентных множеств, мощности множества, счётного множества, множества мощности континуум;

- приобрели умения и навыки устанавливать эквивалентность числовых множеств, идентифицировать счётные множества и множества мощности с.

2.4 Практическое занятие № 7, 8 (4 часа).

Тема: «Основы комбинаторики»

2.4.1 Задание для работы:

1. Правила комбинаторики.

2. Комбинаторные формулы.

2.4.2 Краткое описание проводимого занятия:

1. Правила комбинаторики.

1. Оценить применимость алгоритма.

Агенство недвижимости, база данных. Запись – пара (предложение, спрос). Найти варианты обмена (т.е. такие пары, где первая компонента одной совпадает со второй компонентой другой). Оценить простейший вариант поиска – «лобовой».

Решение. Трудоемкость $n\times(n-1)/2$. Если на одну проверку нужна 1 миллисекунда, то при $n = 100$ потребуется около 5 секунд, при n=100 000 – 5×10^6 сек, т.е. около 1389 часов. Алгоритм непригодный.

2. Пусть в киоске имеется 5 различных книг по математике и 7 по физике. Если студент может купить только одну книгу, то сколько у него есть вариантов?

Решение. 5 вариантов выбора первой книги и 7 вариантов – второй, т.е. 12 вариантов.

3. Пусть в салоне связи имеется 50 различных моделей сотовых телефонов и по три вида чехлов для каждой модели. Сколькими способами можно выбрать телефон и чехол к нему?

Решение. Очевидно: имеется 50 вариантов выбора телефона. Выбрав телефон, можно 3 способами выбрать чехол, т.е. всего 50×3=150 вариантов.

2. Комбинаторные формулы.

4.На тренировках занимаются 8 баскетболистов. Сколько разных пятерок может быть образовано тренером?

Решение. Т.к. при образовании пятерки важен только ее состав, то достаточно определить $C_8^5 = \dfrac{8!}{5!(8-5)!} = \dfrac{8!}{5!3!} = \dfrac{6\cdot7\cdot8}{1\cdot2\cdot3} = 56$ пятерок.

5. Сколько различных трехзначных чисел можно составить из цифр 1, 2, 3, 4, 5? при условии, что ни одна цифра не повторяется?

Решение. Составить разные числа можно: $\overline{A}_5^3 = 5^3 = 125$ способами (размещения с повторениями). Если ни одна цифра не должна повторяться, то таких способов будет $A_5^3 = \dfrac{5!}{(5-3)!} = \dfrac{5!}{2!} = 5\cdot4\cdot3 = 60$ (размещения без повторений).

2.4.3 Результаты и выводы:

В результате проведенного занятия студенты:
- освоили основные комбинаторные принципы и формулы;
- приобрели умения и навыки решать простейшие комбинаторные задачи.

2.5 Практическое занятие № 9, 10 (4 часа).

Тема: «Основные понятия теории графов. Числовые характеристики графов»

2.5.1 Задание для работы:

1. Основные понятия теории графов. Виды графов.

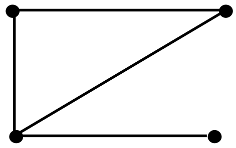
2. Операции над графами. Способы задания графов. Матричное представление графов.

2.5.2 Краткое описание проводимого занятия:

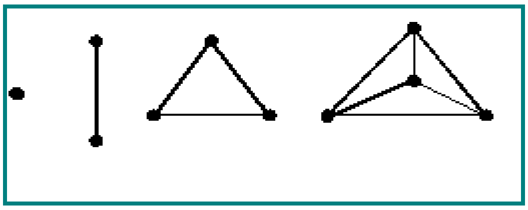
1. Основные понятия теории графов. Виды графов.

1. Указать смежные вершины, рёбра, инцидентные вершины и рёбра.

Решение. Вершины v_1 и v_2 являются смежными, вершина v_1 инцидентна ребрам $e_2 = (v_1,v_2)$ и $e_1 = (v_1,v_3)$. $V = \{v_1, v_2, v_3, v_4\}$, $= \{e_1, e_2, e_3, e_4\}$. Ребра e_1, e_2, e_3 являются попарно смежными, а ребра e_2, e_4 – несмежными, так же как и вершины v_1, v_4 и v_2, v_4.



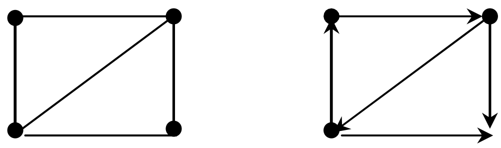
2. Являются ли графы полными? Указать их порядки.



Решение. На рисунке изображены полные графы порядка 1, 2, 3 и 4. Они обозначаются K_n.

2. Операции над графами. Способы задания графов. Матричное представление графов.

3. Составить матрицы смежности графов.

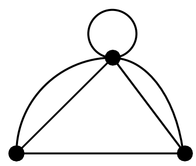


Решение. Матрицы смежности для заданных графа G и орграфа D

$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad A(D) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

4. Найти матрицу смежности псевдографа.

$$A(P) = \begin{pmatrix} 0 & 2 & 1 \\ 2 & 2 & 2 \\ 1 & 2 & 0 \end{pmatrix}$$

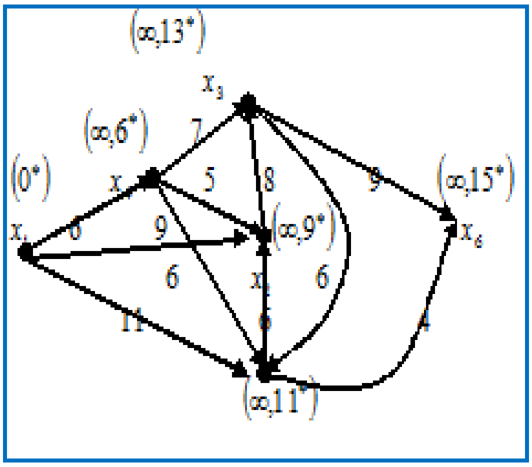


5. Найти матрицы инцидентности для заданных графа G и орграфа D

$$I(G)=\begin{pmatrix}1&0&0&1&0\\1&1&0&0&1\\0&1&1&0&0\\0&0&1&1&1\end{pmatrix}\qquad I(D)=\begin{pmatrix}-1&0&0&1&0\\1&-1&0&0&-1\\0&1&1&0&0\\0&0&-1&-1&1\end{pmatrix}$$

6. Задана весовая матрица сети P. Построить по этой матрице сеть,
Изобразим теперь сам граф по данной матрице весов.

$$P=\begin{array}{c} \\ x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\end{array}\begin{array}{c}\begin{matrix}x_1&x_2&x_3&x_4&x_5&x_6\end{matrix}\\ \begin{pmatrix}-&9&\infty&6&11&\infty\\ \infty&-&8&\infty&\infty&\infty\\ \infty&\infty&-&\infty&6&9\\ \infty&5&7&-&6&\infty\\ \infty&6&\infty&\infty&-&4\\ \infty&\infty&\infty&\infty&\infty&-\end{pmatrix}\end{array}$$



2.5.3 Результаты и выводы:

В результате проведенного занятия студенты:

- освоили основные понятия теории графов, виды графов, операции над графами, способы задания графов, матричное представление графов;

- приобрели умения и навыки применять основные понятия теории графов, операции над графами, способы задания графов, матричное представление графов при решении задач.

2.6 Практическое занятие № 11 (2 часа).

Тема: «Деревья. Планарные и хроматические графы»

2.6.1 Задание для работы:

1. Деревья.

2.Планарные графы. Хроматические графы

2.6.2 Краткое описание проводимого занятия:

1. Деревья.

Для графа, заданного матрицей весов,
а) построить по этой матрице сеть (исходный граф),
б) построить остов наименьшего веса,
в) найти его вес.

$$W=\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 5 & 10 & 14 & \infty & \infty \\ x_2 & 5 & - & 5 & 6 & \infty & \infty \\ x_3 & 10 & 5 & - & 7 & 8 & 9 \\ x_4 & 14 & 6 & 7 & - & 4 & \infty \\ x_5 & \infty & \infty & 8 & 4 & - & 12 \\ x_6 & \infty & \infty & 9 & \infty & 12 & - \end{array}.$$

Шаг1. $S^{/}=\{x_1\},\ S^{//}=\{x_2,x_3,x_4,x_5,x_6\},\ U^{/}=\varnothing.$

<u>Первая итерация.</u> Шаг 2.

$d\left(S^{/},S^{//}\right)=\omega\left(x_1,x_2\right)=5,\ S^{/}=\{x_1,x_2\},\ S^{//}=\{x_3,x_4,x_5,x_6\},$

$U^{/}=\{(x_1,x_2)\}.$

Шаг 3. $S^{/}\neq S$, переход на начало второго шага.

<u>Вторая итерация.</u> Шаг 2.

$d\left(S^{/},S^{//}\right)=\omega\left(x_2,x_3\right)=5,\ S^{/}=\{x_1,x_2,x_3\},\ S^{//}=\{x_4,x_5,x_6\},$

$U^{/}=\{(x_1,x_2),(x_2,x_3)\}.$

Шаг 3. $S^{/}\neq S$, переход на начало второго шага.

<u>Третья итерация.</u> Шаг 2.

$d\left(S^{/},S^{//}\right)=\omega\left(x_2,x_4\right)=6,\ S^{/}=\{x_1,x_2,x_3,x_4\},\ S^{//}=\{x_5,x_6\},$

$U^{/}=\{(x_1,x_2),(x_2,x_3),(x_2,x_4)\}.$

Шаг 3. $S^{/}\neq S$, переход на начало второго шага.

<u>Четвертая итерация.</u> Шаг 2. $d\left(S^{/},S^{//}\right)=\omega\left(x_4,x_5\right)=4,\ S^{/}=\{x_1,x_2,x_3,x_4,x_5\},\ S^{//}=\{x_6\},$

$U^{/}=\{(x_1,x_2),(x_2,x_3),(x_2,x_4),(x_4,x_5)\}.$

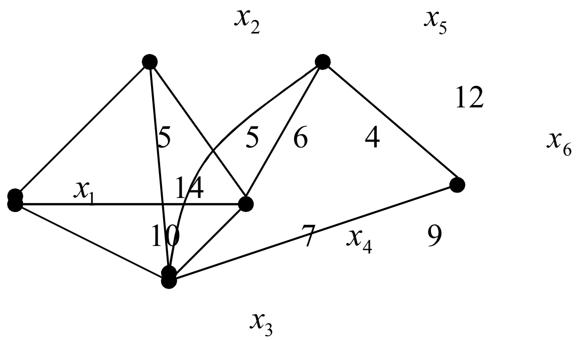
Шаг 3. $S^{/}\neq S$, переход на начало второго шага.

<u>Пятая итерация.</u> Шаг 2. $d\left(S^{/},S^{//}\right)=\omega\left(x_3,x_6\right)=9,\ S^{/}=\{x_1,x_2,x_3,x_4,x_5,x_6\},\ S^{//}=\varnothing,$

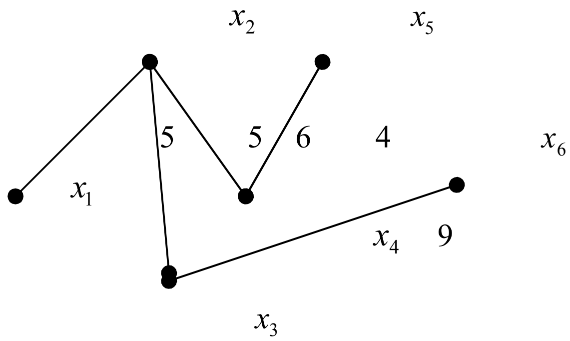
$U^{/}=\{(x_1,x_2),(x_2,x_3),(x_2,x_4),(x_4,x_5),(x_3,x_6)\}.$

Шаг 3. $S^{/}=S$. Итак, получен остовный граф. $G^{/}=\left(S^{/},U^{/}\right)$ изображен на рисунке справа, его вес $\omega\left(G^{/}\right)=5+5+6+4+9=29.$

Исходный граф



Остовный граф



2.Планарные графы. Хроматические графы

2.6.3 Результаты и выводы:

В результате проведенного занятия студенты:

- освоили понятия дерева, планарных хроматических графов;
- приобрели умения и навыки применять понятия дерева, планарных хроматических графов при решении задач.

2.7 Практическое занятие № 12 (2 часа).

Тема: «Оптимизационные задачи на графах и сетях, алгоритмы их решения. Прикладные задачи и алгоритмы анализа графов. Потоки в сетях. Сетевое планирование»

2.7.1 Задание для работы:

1. Оптимизационные задачи на графах и сетях, алгоритмы их решения. Прикладные задачи и алгоритмы анализа графов.

2. Потоки в сетях. Задача о максимальном потоке. Сетевое планирование. Критический путь и критическое время сетевого графа

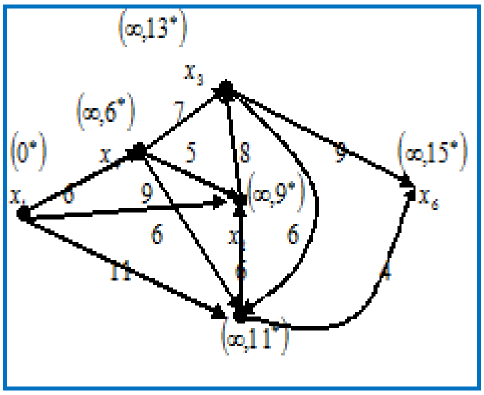
2.7.2 Краткое описание проводимого занятия:

1. Оптимизационные задачи на графах и сетях, алгоритмы их решения. Прикладные задачи и алгоритмы анализа графов.

Задание. Задана весовая матрица сети G. Найти минимальный путь из вершины x_1 в вершину x_6 по алгоритму Дейкстры.

Решение. Изобразим теперь сам граф по данной матрице весов.

$$P = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{pmatrix} - & 9 & \infty & 6 & 11 & \infty \\ \infty & - & 8 & \infty & \infty & \infty \\ \infty & \infty & - & \infty & 6 & 9 \\ \infty & 5 & 7 & - & 6 & \infty \\ \infty & 6 & \infty & \infty & - & 4 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix} \end{array} .$$



Этап1. Шаг1. Полагаем $d(x_1) = 0^*, \widetilde{x} = x_1, x_5$

$d(x_2) = d(x_3) = d(x_4) = d(x_5) = d(x_6) = \infty.$

1-я итерация. Шаг 2. Множество вершин, непосредственно следующих за $\widetilde{x} = x_1$ с временными метками $\widetilde{S} = \{x_2, x_4, x_5\}$. Пересчитываем временные метки этих вершин

$$d(x_2) = \min\{\infty, 0^* + 9\} = 9,\ d(x_4) = \min\{\infty, 0^* + 6\} = 6,\ d(x_5) = \min\{\infty, 0^* + 11\} = 11.$$

Шаг 3. Одна из временных меток превращается в постоянную $\min\{9, \infty, 6, 11, \infty\} = 6^* = d(x_4),\ \tilde{x} = x_4$.

Шаг 4. $\tilde{x} = x_4 \neq t = x_6$, происходит возвращение на второй шаг.

<u>2-я итерация.</u> Шаг 2. $\tilde{S} = \{x_2, x_3, x_5\},\ d(x_2) = \min\{9, 6^* + 5\} = 9$, $d(x_3) = \min\{\infty, 6^* + 7\} = 13$, $d(x_5) = \min\{11, 6^* + 6\} = 11$.

Шаг 3. $\min\{d(x_2), d(x_3), d(x_5), d(x_6)\} = \min\{9, 13, 11, \infty\} = 9^* = d(x_2),\ \tilde{x} = x_2$.

Шаг 4. $x_2 \neq x_6$, возвращение на второй шаг.

<u>3-я итерация.</u> Шаг 2. $\tilde{S} = \{x_3\},\ d(x_3) = \min\{13, 9^* + 8\} = 13$.

Шаг 3. . $\min\{d(x_3), d(x_5), d(x_6)\} = \min\{13, 11, \infty\} = 11^* = d(x_5),\ \tilde{x} = x_5$.

Шаг 4. $x_5 \neq x_6$, возвращение на второй шаг.

<u>4-я итерация.</u> Шаг 2. $\tilde{S} = \{x_6\},\ d(x_6) = \min\{\infty, 11^* + 4\} = 15$.

Шаг 3. . $\min\{d(x_3), d(x_6)\} = \min\{13, 15\} = 13^* = d(x_3),\ \tilde{x} = x_3$.

Шаг 4. $x_3 \neq x_6$, возвращение на второй шаг.

<u>5-я итерация.</u> Шаг 2. $\tilde{S} = \{x_6\},\ d(x_6) = \min\{15, 13^* + 9\} = 15$.

Шаг 3. $\min\{d(x_6)\} = \min\{15\} = 15^*,\ \tilde{x} = x_6$.

Шаг 4. $x_6 = t = x_6$, конец первого этапа.

<u>Этап 2.</u> Шаг5. Составим множество вершин, непосредственно предшествующих $\tilde{x} = x_6$ с постоянными метками $\tilde{S} = \{x_3, x_5\}$. Проверим для этих двух вершин выполнение равенства (4.7.3).

$d(\tilde{x}) = 15 = 11^* + 4 = d(x_5) + \omega(x_5, x_6),\ d(\tilde{x}) = 15 \neq 13^* + 9 = d(x_3) + \omega(x_3, x_6)$. Включаем дугу (x_5, x_6) в кратчайший путь. $\tilde{x} = x_5$.

Шаг 6. $\tilde{x} \neq s = x_1$, возвращение на пятый шаг.

<u>2-я итерация.</u> Шаг 5. $\tilde{S} = \{x_1, x_4\}$.

$d(\tilde{x}) = 11 = 0^* + 11 = d(x_1) + \omega(x_1, x_5),\ d(\tilde{x}) = 11 \neq 6^* + 6 = d(x_4) + \omega(x_4, x_5)$. Включаем дугу (x_1, x_5) в кратчайший путь. $\tilde{x} = x_1$.

Шаг 6. $\tilde{x} = s = x_1$, завершение второго этапа.

Итак, кратчайший путь от вершины x_1 до вершины x_6 построен. Его длина (вес) равна 15, сам путь образует следующая последовательность дуг $\mu = (x_1, x_5) - (x_5, x_6)$.

2. Потоки в сетях. Задача о максимальном потоке. Сетевое планирование. Критический путь и критическое время сетевого графа

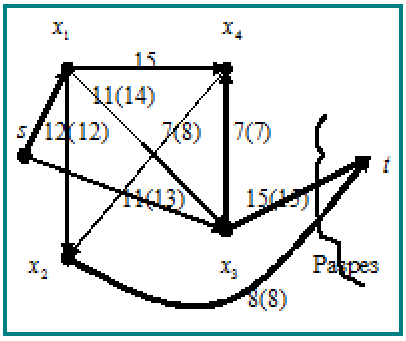
Задание. Пропускные способности дуг заданы следующей матрицей. Построить сеть, найти максимальный поток от s к t и указать минимальный разрез, отделяющий s от t.

Решение.

$$W=\begin{array}{c} \\ s \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ t \end{array}\begin{array}{c} \begin{array}{cccccc} s & x_1 & x_2 & x_3 & x_4 & t \end{array} \\ \begin{pmatrix} - & 12 & - & 13 & - & - \\ - & - & 11 & 14 & 15 & - \\ - & - & - & - & - & 8 \\ - & - & - & - & 7 & 15 \\ - & - & 8 & - & - & - \\ - & - & - & - & - & - \end{pmatrix} \end{array}.$$

<u>Этап 1.</u> Путь $s\xrightarrow{12}x_1\xrightarrow{14}x_3\xrightarrow{15}t.$

$\delta=\min(12,14,15)=12.$ Увеличим по этому пути поток до 12 единиц, ребро (s,x_1) становится насыщенным. Поставим величину потока на дугах (x_1,x_3) и (x_3,t).



$\delta=\min(13,15-12)=3.$ Поток можно увеличить на три единицы. Дуга (x_3,t) станет насыщенной. Путь $s\xrightarrow{3(13)}x_3\xrightarrow{7}x_4\xrightarrow{8}x_2\xrightarrow{8}t.$ Можно увеличить поток на семь единиц;

Дуга (x_3,x_4) станет насыщенной, потоки примут вид

$$s\xrightarrow{10(13)}x_3\xrightarrow{7(7)}x_4\xrightarrow{7(8)}x_2\xrightarrow{7(8)}t.$$

Больше путей нет. Конец первого этапа.

<u>Этап 2.</u> Рассмотрим теперь маршруты, содержащие противоположные дуги.

Маршрут $s\xrightarrow{10(13)}x_3\xleftarrow{12(14)}x_1\xrightarrow{11}x_2\xrightarrow{7(8)}t.$ Поток можно увеличить на единицу на дуге (x_2,t). Тогда потоки по дугам этого маршрута станут такими $s\xrightarrow{11(13)}x_3\xleftarrow{11(14)}x_1\xrightarrow{1(11)}x_2\xrightarrow{8(8)}t.$ Дуга (x_2,t) стала насыщенной.

Больше маршрутов нет. Поток максимален. Делаем разрез вокруг t по насыщенным дугам и получаем его величину 15+8=23 единицы.

2.7.3 Результаты и выводы:

В результате проведенного занятия студенты:

- освоили понятия об оптимизационных задачах на графах и сетях, алгоритмах их решения, прикладных задачах и алгоритмах анализа графов; освоили понятия потока в сетях, понятие задачи о максимальном потоке; освоили элементы сетевого планирования;

- приобрели умения и навыки решать оптимизационные задачи на графах и сетях.

2.8 Практическое занятие № 13, 14 (4 часа).

Тема: «Основы теории булевых функций»

2.8.1 Задание для работы:

1. Булевы функции. Элементарные булевы функции
2. Представление булевых функций формулами.
3. Минимизация булевых функций в классе ДНФ

2.8.2 Краткое описание проводимого занятия:

1. Булевы функции. Элементарные булевы функции.

1.Составим таблицу истинности для формулы $\overline{B} \to \overline{A}$:

				$\overline{B} \to$
				1
				0
				1
				1

2. Проверим эквивалентность формул $A \vee B$ и $\overline{A \wedge \overline{B}}$, составив для них таблицы истинности.

Формулы не эквивалентны, так как 3-й и 6-й столбцы таблицы не совпадают.

2. Представление булевых функций формулами.

1. Задание. Записать ДНФ и КНФ формулы.

Решение. Элементарные дизъюнкции: $x \vee \overline{y}$, z. Элемент. конъюнкции: $x \cdot \overline{y} \cdot z$, x.

$f(x,y,z) = xyz \vee \overline{x}y$ – ДНФ ; $f(x,y,z) = (x \vee \overline{y}) \cdot z$ – КНФ.

2. Для упрощения формулы используем правило исключения импликации: $A_1 \to A_2 = \overline{A}_1 \vee A_2$.

$$\neg(A_1 \to A_2) \vee (A_2 \to \overline{A}_1) = \overline{(\overline{A}_1 \vee A_2)} \vee \overline{A}_2 \vee \overline{A}_1 = (\overline{\overline{A}}1 \wedge \overline{A}_2) \vee \overline{A}_2 \vee \overline{A}_1 =$$

$$= (A_1 \wedge \overline{A}_2) \vee \overline{A}_2 \vee \overline{A}_1 = \overline{A}_2 \wedge (A_1 \vee 1) \vee \overline{A}_1 = \overline{A}_2 \vee \overline{A}_1.$$

3. Используя законы логики приведем формулу $\overline{(A \wedge B) \vee C}$ к виду, содержащему только дизъюнкции элементарных конъюнкций. Полученная формула и будет искомой ДНФ: $\overline{(A \wedge B) \vee C} = \overline{(A \wedge B)} \wedge \overline{C} = (\overline{A} \vee \overline{B}) \wedge \overline{C} = (\overline{A} \wedge \overline{C}) \vee (\overline{B} \wedge \overline{C})$

Для построения СДНФ составим таблицу истинности для данной формулы:

			$A \wedge B$	$(A \wedge B) \vee C$	$\overline{(A \wedge B}$
		0	0	0	1
		1	0	1	0
		0	0	0	1
		1	0	1	0
		0	0	0	1
		1	0	1	0
		0	1	1	0
		1	1	1	0

Помечаем те строки таблицы, в которых формула (последний столбец) принимает значение “1”. Для каждой такой строки выпишем формулу, истинную на наборе переменных *A,B,C* данной строки: строка 1 – $\overline{A} \wedge \overline{B} \wedge \overline{C}$; строка 3 – $\overline{A} \wedge B \wedge \overline{C}$; строка 5 – $A \wedge \overline{B} \wedge \overline{C}$. Дизъюнкция этих трех формул будет принимать значение “1” только на наборах переменных в строках 1, 3, 5, а следовательно и будет искомой совершенной дизъюнктивной нормальной формой (СДНФ): $(\overline{A} \wedge \overline{B} \wedge \overline{C}) \vee (\overline{A} \wedge B \wedge \overline{C}) \vee (A \wedge \overline{B} \wedge \overline{C})$

4. Задание. Найти СДНФ и СКНФ двумя способами.

Решение.

1) Получим СДНФ и СКНФ по ТИ:

$$f(x,y,z)=\bar{x}\bar{y}\bar{z}\vee\bar{x}\bar{y}z\vee x\bar{y}z\vee xy\bar{z}\vee xyz-\textit{СДНФ},$$

$$f(x,y,z)=(x\vee y\vee\bar{z})(x\vee\bar{y}\vee z)(\bar{x}\vee\bar{y}\vee z)-\textit{СКНФ}$$

2) Получим СДНФ и СКНФ из ДНФ и КНФ:

$$g(x,y,z)=xy\vee\bar{x}z=xy(z\vee\bar{z})\vee\bar{x}z(y\vee\bar{y})=xyz\vee xy\bar{z}\vee\bar{x}yz\vee\bar{x}\bar{y}z$$

$$g(x,y,z)=(\bar{x}\vee y)(x\vee z)(y\vee z)=(\bar{x}\vee y\vee z\bar{z})(x\vee y\bar{y}\vee z)(x\bar{x}\vee y\vee z)=$$

$$(\bar{x}\vee y\vee z)(\bar{x}\vee y\vee\bar{z})(x\vee y\vee z)(x\vee\bar{y}\vee z)(x\vee y\vee z)(\bar{x}\vee y\vee z)=$$

$$(\bar{x}\vee y\vee z)(\bar{x}\vee y\vee\bar{z})(x\vee y\vee z)(x\vee\bar{y}\vee z)$$

3. Минимизация булевых функций в классе ДНФ

1) $f(x_1,x_2,x_3)=x_1x_2\vee\bar{x}_1\bar{x}_2\bar{x}_3\vee x_1x_2x_3$. $\bar{x}_1\bar{x}_2\bar{x}_3$ – импликанта, причем простая; $x_1x_2x_3$ – импликанта, но не простая, т.к. удаление x_3 снова дает импликанту x_1x_2 (которая является простой).

2) Найдем импликанты и простые импликанты для функции $f(x_1,x_2)=x_1\to x_2$. Всего имеется 8 элементарных конъюнкций с переменными x_1, x_2. Приведем их таблицы истинности.

1	2	$x_1 \to x_2$				$_1x_2$			1	2
		1	1	0	0					
		1	0	1	0					
		0	0	0	1					
		1	0	0	0					

Из таблицы истинности заключаем, что $\bar{x}_1\bar{x}_2$, $\bar{x}_1x_2$, x_1x_2, $\bar{x}_1$, x_2 являются импликантами функции *f*. Из них простыми являются $\bar{x}_1$ и x_2.

2.8.3 Результаты и выводы:

В результате проведенного занятия студенты:

- освоили понятия о булевых функциях, представлении булевых функций формулами, минимизации булевых функций;

- приобрели умения и навыки решать задачи, связанные с булевыми функциями, представлением булевых функций формулами, минимизацией булевых функций.

2.9 Практическое занятие № 15, 16 (4 часа).

Тема: «Элементы теории алгоритмов»

2.9.1 Задание для работы:

1. Основные подходы к формализации понятия алгоритма (машина Тьюринга; рекурсивный алгоритм, нормальные алгоритмы Маркова).

2. Понятие эффективности и сложности алгоритмов.

2.9.2 Краткое описание проводимого занятия:

1. Основные подходы к формализации понятия алгоритма (машина Тьюринга; рекурсивный алгоритм, нормальные алгоритмы Маркова).

1.

	a_0	1
q_1	$1Hq_0$	$1\Pi q_1$

Из любой начальной конфигурации(УУ обозревает не пустой символ) эта машина Тьюринга переводит слово 11 в слово-...(Отв.: 111)

2. В команде $a_3q_2 \to a_0Лq_0$ следующее состояние машины Тьюринга

+а) q_0

б) q_2

в) q_1

г) a_0

д) q_2

3. Одной из моделей (формализаций) алгоритма является

+а) машина Тьюринга

б) задача линейного программирования

в) эйлеровы графы

г) алгебра множеств

д) алгебра логики

4. Пусть заданы число a и функция $\psi(x,y)$. Функцию $f(y)$, определённую системой равенств $\begin{cases} f(0)=a \\ f(y+1)=\psi\left(y,f(y)\right) \end{cases}$, называют полученной по схеме …

(Отв. примитивной рекурсии)

5. 10. Пусть заданы число $a=2$ и функция $\psi(x,y)=y+3$. Функция $f(y)$ получена по схеме примитивной рекурсии: $\begin{cases} f(0)=a \\ f(y+1)=\psi\left(y,f(y)\right) \end{cases}$. Тогда $f(1)$ равно-…

ОТВЕТ:5

2. Понятие эффективности и сложности алгоритмов.

1. Оценить применимость алгоритма.

Агенство недвижимости, база данных. Запись – пара (предложение, спрос). Найти варианты обмена (т.е. такие пары, где первая компонента одной совпадает со второй компонентой другой). Оценить простейший вариант поиска – «лобовой».

Решение. Трудоемкость $n\times(n–1)/2$. Если на одну проверку нужна 1 миллисекунда, то при n = 100 потребуется около 5 секунд, при n=100 000 – 5×10^6 сек, т.е. около 1389 часов. Алгоритм непригодный.

2.9.3 Результаты и выводы: в результате проведенного занятия студенты:

- освоили понятия об основных подходах к формализации понятия алгоритма (машина Тьюринга; рекурсивный алгоритм, нормальные алгоритмы Маркова); понятие эффективности и сложности алгоритмов;
- приобрели умения и навыки алгоритмизации простейших задач.

2.10 Практическое занятие № 17 (4 часа).

Тема: «Конечные автоматы**»**

2.10.1 Задание для работы:

1. Понятие конечного автомата. Историческая справка. Способы задания конечного автомата. Примеры конечных автоматов. Виды автоматов. Общие задачи теории автоматов.

2.10.2 Краткое описание проводимого занятия:

1. Понятие конечного автомата. Историческая справка. Способы задания конечного автомата. Примеры конечных автоматов. Виды автоматов. Общие задачи теории автоматов.

Задание 1. Задан конечный автомат $A=\{X;Q;Y;\lambda(x,q);\delta(x,q)\}$ - элемент задержки (элемент памяти): $X=\{0,1\}$, $Q=\{0,1\}, Y=\{0,1\}$, функция переходов $\lambda(0,0)=0, \lambda(0,1)=0, \lambda(1,0)=1$, $\lambda(1,1)=1$, функция выходов $\delta(0,0)=0$, $\delta(0,1)=1$, $\delta(1,0)=0$, $\delta(1,1)=1$. При входном сигнале $x_2=1$ из состояния $q_1=0$ автомат переходит в состояние-…

(Отв.: 1)

2. Задан конечный автомат $A=\{X;Q;Y;\lambda(x,q);\delta(x,q)\}$ - элемент задержки (элемент памяти): $X=\{0,1\}$, $Q=\{0,1\}$, $Y=\{0,1\}$, функция переходов $\lambda(0,0)=0$, $\lambda(0,1)=0$, $\lambda(1,0)=1$, $\lambda(1,1)=1$, функция выходов $\delta(0,0)=0$, $\delta(0,1)=1$, $\delta(1,0)=0$, $\delta(1,1)=1$. При входном сигнале $x_1=0$ в состоянии $q_2=1$ автомат выдаёт выходной сигнал-…

(Отв.:1)

2.10.3 Результаты и выводы:

В результате проведенного занятия студенты:

- освоили понятия конечного автомата, способы задания конечного автомата, примеры конечных автоматов, виды автоматов;

- приобрели умения и навыки решения простейших задач по теме «Конечные автоматы».