

1.2 Алгебра подмножеств (Т)-теорема (Л)-лемма (С) -следствие м – множество

На 1 балл:

План: 1.2.1. Сравнение множеств: \subset — включение множеств; $=$ — равенство множеств; собственные и несобственные подмножества

1.2.2. Равномощные множества: \subseteq — вложение множеств; \sim — взаимно-однозначное соответствие; равномощность

1.2.3. Конечные и бесконечные множества: $|A|=\infty$

1.2.4. Счётные и несчётные множества: Т(n) — треугольные числа; алгоритм перечисления конечных подмножеств натурального ряда

1.2.5. Мощность конечного множества:

Принцип Дирихле

На 2 балла:

Основные понятия

\subset - М А содержится в м В (м В включает м А), если каждый элемент м А есть элемент м В. А — подмножество В, В — надмножество А. Два м =, если они являются подмножествами друг друга. (Т) Включение множеств обладает следующими

Если $A \subset B$ и $A \neq B, A \neq \emptyset$, то А называется подмножеством В, а В — собственным надмножеством А.

\subseteq - соответствие из м А в м В, если каждому элементу м А поставлен в соответствие 1 и только 1 элемент м В, причём различным элементам м А поставлены в соответствие различные элементы м В. Множество А вложено в В

\sim - Говорят, что между м А и В установлено взаимно-однозначное соответствие, если каждому элементу А поставлен в соответствие один и только один элемент В, и для каждого элемента В один и только один элемент А (множества изоморфны).

Если между А и В взаимно-однозначное \rightarrow они имеют одинаковую мощность (равномощны)

(Т) Равномощность множеств обладает следующими свойствами:

$$A \leftrightarrow B \Rightarrow |A| = |A^{\leftrightarrow}| \quad (\text{Л})$$

- 1) $\forall A (A \subset A)$; свойствами
 2) $\forall A, B (A \subset B \ \& \ B \subset A \Rightarrow A = B)$; собственным
 3) $\forall A, B, C (A \subset B \ \& \ B \subset C \Rightarrow A \subset C)$.

Множество А называется конечным, если у него нет равномощного собственного подмножества Для конечного множества А используется запись $|A| < \infty$. Все остальные множества бесконечные.

(Т) Множество, имеющее бесконечное подмножество, бесконечно. (С: Все подмножества конечного множества конечны)

Бесконечные множества, равномощные множеству натуральных чисел, называются счётными. (Если элементы можно пронумеровать)

Треугольные числа — число точек, которые могут быть расставлены в форме правильного треугольника. 1, 3, 6, 10...

(Т) Множество всех подмножеств натурального ряда несчётно

Алгоритм пересечения — алгоритм слияния (1 лаба)

(Т) Любое непустое конечное множество равномощно некоторому отрезку натурального ряда

(Л) Любое непустое подмножество множества натуральных чисел содержит наименьший элемент.

(Т) Любой отрезок натурального ряда конечен

Принцип Дирихле: Различные отрезки натурального ряда неравномощны (п кроликов не рассадить по ящикам)

Диаграммы Вена, подобие кругов Эйлера, иллюстрирующие операции над множествами.

Покрытие — семейство подмножеств множества. Семейство дизъюнктным — элементы попарно не пересекаются. Дизъюнктивное покрытие = разбиение. Элементы разбиения = блоки.

(Т) Если Е дизъюнктное семейство подмножеств М, то существует разбиение В множества М, что каждый элемент дизъюнктного семейства Е является подмножеством блока разбиения В.

(Т Кантора-Берништейна) Если А вложено в В, а В вложена в А, то они изоморфны.

1.2.6. Операции над множествами: \cup — объединение; \cap — пересечение; \setminus — разность; M' — симметрическая разность; дополнение, диаграмма Венна

1.2.7. Разбиения и покрытия: Дизъюнктное семейство, теорема Кантора-Берништейна

1.2.8. Булев : 2^M — множество подмножеств; алгебра подмножеств, парадокс Кантора

1.2.9. Свойства операций над множествами: Идемпотентность, коммутативность, ассоциативность, дистрибутивность, поглощение, инволютивность, законы де Моргана

(C: Если $A \subset B \subset C$ и $A \sim C$, то $B \sim C$)

Булеан – множество всех подмножеств M , обозначается 2^M

(T) Если множество M конечно, то $|2^M| = 2^{|M|}$

Семейство равномощных подмножеств – семейство подмножеств булеана имеющих одинаковую мощность.

(Т Кантора) Булеан множества мощнее множества (C: Не существует множества всех множеств <-
Парadox Кантора)

Свойства операции над множествами много!! Посмотрите сами, их названия в начале

1.3. Представление множеств в программах (Т) – Теорема (Л) – Лемма (С) - Следствие

На 1 балл:

1.3.1. Битовые шкалы: \wedge – конъюнкция; \vee – дизъюнкция; $+_2$ – сложение по модулю два; \neg – инвертирование; бит

1.3.2. Представление натуральных чисел: система счисления; основание; цифра

1.3.3. Перебор подмножеств множества: $B(i)$ – двоичный код числа i ; $Q(i)$ – число битов, в которых различаются $B(i-1)$ и $B(i)$

1.3.4. Алгоритм построения кода Грея: бинарный код Грея; левый зеркальный код Грея; рекуррентные вычисления

1.3.5. Представление множеств списками: Алгоритм слияния

1.3.6. Проверка включения слиянием 1.3.7. Вычисление объединения слиянием 1.3.8. Вычисление пересечения слиянием

1.3.9. Представление множеств итераторами: Первичные операции; алгоритм пересечения итераторов; алгоритм вычитания итераторов; алгоритм дизъюнктного объединения итераторов; оператор структурного перехода

На 2 балла:

Основные понятия

Булевский массив array [1..n] of 0..1 называют **битовой шкалой** (или **двоичным вектором**, или **машинным словом**, или **двоичным кодом**) длины n. Элементы битовой шкалы называют разрядами, или битами.

(Т) Существует 2^n различных битовых шкал длины n.

(Л) [1] $\forall k \in \mathbb{N} (\exists! m \geq 0 (2^m \leq k < 2^{m+1}))$

(Л) [2] $\sum_{i=0}^m 2^i = 2^{m+1} - 1$.

(Т) Для любого натурального числа k существует единственное множество $B(k) = \{m_1, \dots, m_s\}$ такое, что $k = \sum_{i=1}^s 2^{m_i}$

(С)[1] Любое натуральное число k такое, что $2^m \leq k < 2^{m+1}$, можно представить в виде линейной комбинации степеней числа 2

(С)[2] Существует взаимно-однозначное соответствие между битовыми шкалами длины n и целыми неотрицательными числами $0 \leq k \leq 2^n - 1$.

Способ интерпретации двоичных шкал называется **двоичным представлением**, или **двоичным кодом** числа

Алгоритм перебора выдает 2^n различных целых чисел, следовательно, 2^n различных кодов. Таким образом, все подмножества генерируются, причём ровно по одному разу. (К алгоритму перебора добавляется несколько следствий, все они про различные ситуации про цифры в разрядах и тд.)

Бинарный код Грея n-элементного множества — это последовательность всех подмножеств этого множества (то есть битовых шкал), в которой любые два соседних подмножества в последовательности, в том числе первое и последнее подмножество, различаются в точности одним элементом (разрядом).

Вход: число $n \geq 0$ – мощность множества.
Выход: последовательность кодов подмножеств B.

```
B : array [1..n] of 0..1 //битовая шкала
for i from 1 to n do B[i] := 0 end for //инициализация
yield B //пустое множество
for i from 1 to 2^n - 1 do
  p := n + 1 - Q(i) //определение номера элемента
  B[p] := 1 - B[p] //добавление или удаление элемента
  yield B //очередное подмножество
end for
Функция Q:
Вход: i – номер подмножества.
Выход: номер изменяемого разряда.
q := 1; j := i
while j mod 2 = 0 do j := j div 2; q := q + 1 end while
return q
```

<- Алгоритм (Т) G(i) - левый зеркальный код Грея $G(i) = B(i) +_2 (B(i) \text{ div } 2)$

Операции с битовыми шкалами можно выполнять не только параллельно, сразу для всех разрядов, но и последовательно, вычисляя биты один за другим и используя полученные значения при вычислении следующих битов. Такие вычисления называются **рекуррентными**. Если универсум очень велик (или бесконечен), а рассматриваемые подмножества универсума не очень велики, то представление с помощью битовых шкал не является эффективным с точки зрения экономии памяти. В этом

случае множества обычно представляются **списками элементов**. Элемент списка: информация и указатель на следующий элемент. **Алгоритмы слияния** (рассмотрены во 2ой лабораторной)

Набор операций, заранее определенный для внутренней структуры объекта называется **первичным**

Итератор – процедура, которая перебирает элементы множества и делает с ними то, что нужно.

1.4. Отношения (свойства бинарных отношений)

На 1 балл:

План:

Упорядоченные пары и наборы (a_1, \dots, a_n) — **кортеж**; $|(a_1, \dots, a_n)|$ — длина кортежа

Прямое произведение множеств: $A \times B$ — **декартово произведение** множеств; A^k — степень множества

Размеченное объединение множеств: $A \oplus B$ — прямая сумма множеств

Бинарные отношения: График отношения, **область отправления**, **область прибытия**; Dom — область определения, Im — область значений; **инфикасная форма записи**; R^{-1} — **обратное отношение**; R — **дополнение отношения**; U — универсальное отношение; тождественное отношение

Композиция отношений: \circ — операция композиции

Свойства отношений: рефлексивность, антирефлексивность, симметричность, антисимметричность, транзитивность, линейность

Степень отношения и циклы: Длина цикла, треугольник, петля, ациклическое отношение

Ядро отношения: ker — операция взятия ядра отношения

Представление отношений в программах: **Булевая матрица**, транспонирование, инвертирование, вычитание, умножение, дизъюнкция

На 2 балла:

Если a и b — объекты, то через (a, b) обозначим упорядоченную пару. Равенство упорядоченных пар определяется следующим образом: $(a, b) = (c, d) \text{ Def} = a=c \& b=d$.

Набор (a_1, \dots, a_n) можно определить рекурсивно, используя понятие упорядоченной пары:

$(a_1, \dots, a_n) \text{Def} = ((a_1, \dots, a_{n-1}), a_n)$. Количество элементов в наборе называется его **длиной** и обозначается следующим образом: $|(a_1, \dots, a_n)|$.

Два набора одной длины равны, если равны их соответствующие элементы: $\forall n((a_1, \dots, a_n) = (b_1, \dots, b_n) \Leftrightarrow \bigwedge_{i=1}^{n-1} a_i = b_i)$. Отсюда следует, что порядок «отщепления» элементов в рекурсивном определении кортежа не имеет значения.

Прямым (декартовым) произведением двух множеств A и B называется множество всех упорядоченных пар, в которых первый элемент принадлежит множеству A , а второй принадлежит множеству B :

$$A \times B = \text{Def} \{(a, b) | a \in A \& b \in B\}$$

Если множества A и B конечны, то $|A \times B| = |A||B|$; $|A^n| = |A|^n$

Пусть X_1, \dots, X_n — семейство множеств, возможно имеющих общие элементы. Размеченным объединением n множеств X_1, \dots, X_n называется множество всех упорядоченных пар, в которых первый элемент принадлежит множеству X_i , а второй элемент — натуральное число i : $X_1 \oplus X_2 \oplus \dots \oplus X_n = \text{Def} \{(x, i) | 1 \leq i \leq n \& x \in X_i\}$

Для конечных множеств $|X_1 \oplus X_2 \oplus \dots \oplus X_n| = \sum_{i=1}^n |X_i|$

Бинарным отношением между множествами A и B называется тройка $\langle A, B, R \rangle$, где R — подмножество прямого произведения A и B : $R \subset A \times B$. Множество R называется **графиком** отношения, A называется **областью отправления**, а B — **областью прибытия**.

Dom R $\text{Def} = \{a \in A | \exists b \in B ((a, b) \in R)\}$, **Dom R** $\subset A$

Im R $\text{Def} = \{b \in B | \exists a \in A ((a, b) \in R)\}$, **Im R** $\subset B$.

Обратное отношение: $R^{-1} \text{Def} = \{(b, a) | (a, b) \in R\} \subset B \times A$.

Дополнение отношения: $R \text{Def} = \{(a, b) | (a, b) \notin R\} \subset A \times B$.

Универсальное отношение: $U \text{Def} = \{(a, b) | a \in A \& b \in B\} = A \times B$.

Тождественное отношение: $I \text{Def} = \{(a, a) | a \in A\} \subset A^2$

Композицией двух отношений R_1 и R_2 называется отношение $R \subset A \times B$ из A в B , определяемое следующим образом: $R = R_1 \circ R_2 = \text{Def} \{(a, b) | a \in A \& b \in B \& \exists c \in C (aR_1c \& cR_2b)\}$

Пусть $R \subset A^2$. Тогда отношение R называется

рефлексивным, если $\forall a \in A (aR a)$;

антирефлексивным, если $\forall a \in A (\neg aR a)$;

симметричным, если $\forall a, b \in A (aR b \Rightarrow bR a)$;

антисимметричным, если $\forall a, b \in A (aR b \& bR a \Rightarrow a=b)$;

транзитивным, если $\forall a, b, c \in A (aR b \& bR c \Rightarrow aR c)$;

линейным, если $\forall a, b \in A (a=b \vee aR b \vee bR a)$.

Теорема. Пусть $R \subset A \times A$ — отношение на A . Тогда:

1. R рефлексивно $\Leftrightarrow I \subset R$;

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

3. R транзитивно $\Leftrightarrow R \circ R \subset R$;

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

5. R антирефлексивно $\Leftrightarrow R \cap I = \emptyset$;

6. R линейно $\Leftrightarrow R \cup R^{-1} = U$.

Пусть R — отношение на множестве A . Степенью отношения R на множестве A называется его n -кратная композиция с самим собой.

Если отношение R на множестве A антисимметрично и транзитивно, то оно **ациклическим**.

Если отношение R на множестве A ациклическим, то оно не симметрическим и не содержит симметрических пар

$R b \& b R a$.

Если $R \subset A \times B$ — отношение между множествами A и B , то композиция $R \circ R^{-1}$ называется **ядром** отношения R и обозначается $\text{ker } R$. Другими словами, $a_1 \text{ker } R a_2 \Leftrightarrow \exists b \in B (a_1 R b \& a_2 R b)$

1.5. Замыкание и сокращение отношений (алгоритм Уоршалла и диаграммы Хассе)

На 1 балл:

План

Транзитивное и рефлексивное замыкание: R^+ — объединение положительных степеней R ; R^* — объединение неотрицательных степеней R

Алгоритм Уоршалла

Транзитивное сокращение: Редукция, минимальное представление

Диаграммы Хассе

Симметричное замыкание и сокращение

На 2 балла:

Пусть R и R' — отношения на множестве M . Отношение R' называется **замыканием отношения R относительно свойства C** , если

1 обладает свойством C : $C(R')$;

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

3 R' является наименьшим: $C(R'') \& R \subset R'' \Rightarrow R' \subset R''$. Пусть R^+ — объединение положительных степеней R , а R^* — объединение неотрицательных степеней R : $R^+ = \text{Def } \bigcup_1^\infty R^i$; $R^* = \text{Def } \bigcup_1^\infty R^i$. Следующий алгоритм Уоршалла вычисляет транзитивное замыкание за $O(n^3)$ шагов.

Вход: матрица отношения R : $\text{array}[1..n, 1..n] \text{ of } 0..1$.

Выход: матрица замыкания T : $\text{array}[1..n, 1..n] \text{ of } 0..1$.

For i from 1 to n do for j from 1 to n do for k from 1 to n do

$T[j, k] := T[j, k] \vee T[j, i] \& T[i, k]$

Транзитивным сокращением отношения R на множестве A называется наименьшее (по количеству элементов) отношение, обозначаемое R^- , такое, что $R^- = R^+$. Транзитивное сокращение содержит минимально необходимые данные, по которым, используя

транзитивность, можно восстановить транзитивное замыкание отношения в полном объёме.

Пусть $S, T \subset A^2$ — конечные ациклические отношения на множестве A и $S^+ = T^+$. Тогда если $\exists (a, b) ((a, b) \in S \& (a, b) \notin T)$, то $(S \setminus \{(a, b)\})^+ = S^+$

Если отношение R ациклически и конечно, то семейство $P(R)$ замкнуто относительно пересечения, то есть $\forall S, T \in P(R) (S \cap T \in P(R))$.

В самом общем случае **диаграммой Хассе** для отношения R на множестве A (обозначение $H(R)$) называется отношение $H(R)$ Def= $\{(a, b) \in R \& a=b \mid \neg \exists c \in A (a \neq c \& c \neq b \& aRc \& cRb)\}$.

Диаграмма Хассе, подобно транзитивному сокращению, более экономно представляет исходное отношение. Однако диаграмма Хассе не тождественна транзитивному сокращению хотя бы потому, что диаграмма Хассе обязательно является подмножеством исходного отношения, в то время как транзитивное сокращение может даже не пересекаться с исходным отношением. Если конечное отношение R антисимметрично и транзитивно, то $H(R) = R^-$.

Симметричным сокращением отношения называется такое минимальное подмножество отношения, симметричное замыкание которого совпадает с симметричным замыканием исходного отношения.

1.6. Функции (суперпозиция функций)

На 1 балл:

Функциональные отношения: Функция, аргумент, значение, область отправления, область прибытия; Dom — область определения, Im — область значений; префиксная форма записи, многозначные отношения, тотальная и частичная функция, сужение и продолжение функции, преобразование, n-местная функция. **Инъекция, сюръекция, биекция:** Взаимно-однозначная функция, обратная функция.

Образы и прообразы: Переход к образам, переход к прообразам. **Суперпозиция функций:** \circ — операция суперпозиции. **Степень функции:** заключительно периодическая последовательность.

Представление функций в программах: Оператор возврата return. end

На 2 балла:

] f — отношение между множествами A и B такое, что $\forall a, b, c ((a, b) \in f \& (a, c) \in f \Rightarrow b = c)$. Такое свойство отношения называется **однозначностью**, или **функциональностью**, а **само отношение называется функцией** из A в B, пишется так: $f : A \rightarrow B$.

Если $f : A \rightarrow B$, то **префиксная** форма записи: $b = f(a)$ (def):= $(a, b) \in f$. Если $b = f(a)$, то a - **аргумент**, a b — **значение** функции. Поскольку функция является отношением, для функции $f : A \rightarrow B$ можно использовать понятия: множество A называется **областью отправления**, множество B — **областью прибытия** функции. Отношения, не обладающие свойством однозначности, называют многозначными. **область определения** функции: $\text{Dom } f \stackrel{\text{Def}}{=} \{a \in A \mid \exists b \in B (b = f(a))\}$;

область значений функции: $\text{Im } f \stackrel{\text{Def}}{=} \{b \in B \mid \exists a \in A (b = f(a))\}$.

$\text{Dom } f \subset A$, $\text{Im } f \subset B$. Если $\text{Dom } f = A$, то функция называется **тотальной**, а если $\text{Dom } f \neq A$ — **частичной**.

Сужением функции $f : A \rightarrow B$ на множество $M \subset A$ называется функция $f|_M$, определяемая следующим образом: $f|_M \stackrel{\text{Def}}{=} \{(a, b) \mid (a, b) \in f \& a \in M\}$

Функция f называется **продолжением** функции g, если g является сужением f.

Функция $f : A_1 \times \dots \times A_n \rightarrow B$ называется **функцией n аргументов**, или **n-местной функцией**.

Функциональность	Инъективность
$(a, b) \in f \& (a, c) \in f \Rightarrow b = c$	$(a, b) \in f \& (c, b) \in f \Rightarrow a = c$
Тотальность	Сюръективность
$\forall a \in A (\exists b \in B ((a, b) \in f))$	$\forall b \in B (\exists a \in A ((a, b) \in f))$

f — тотальная функция, называется **биективной**, или **биекцией**, если она инъективная и сюръективная.

Т. Если $f : A \rightarrow B$ — биекция, то отношение $f^{-1} \subset B \times A$ (**обратная функция**) является биекцией.

] элементы **связаны** отношением R. Число связей, в которых участвует элемент, называют его **степенью**. Свойства отношения можно определить исходя из таблицы:

	Элементы в множестве A	Элементы в множестве B
Степень не более 1	Однозначность	Инъективность
Степень не менее 1	Тотальность	Сюръективность

] $f : A \rightarrow B$, и пусть $A_1 \subset A$, $B_1 \subset B$. Тогда множество $F(A_1)$ (def):= $\{b \in B \mid \exists a \in A_1 (b = f(a))\}$ называется **образом** множества A_1 (при отображении f), а множество $F^{-1}(B_1)$ (def):= $\{a \in A \mid \exists b \in B_1 (b = f(a))\}$ называется **прообразом** множества B_1 (при отображении f).

Т. Если $f : A \rightarrow B$ — функция, то $F : 2^{\text{Dom } f} \rightarrow 2^{\text{Im } f}$ (**переход к образам**) и $F^{-1} : 2^{\text{Im } f} \rightarrow 2^{\text{Dom } f}$ (**переход к прообразам**) — тоже функции.

Композиция функций называется **суперпозицией**. $(f \circ g)(x)$ (def):= $f(g(x))$. \circ - знак суперпозиции.

Т. Суперпозиция функций является функцией.

Степенью n функции f (обозначение f^n) называется n-кратная суперпозиция функции f. $(f^0(x) = x, f^1(x) = f(x)$ и вообще $f^n = f \circ f^{n-1}$). **Л. Т. и С.** $ff^n = f^n f$, $f^n f^m = f^{n+m}$, $f^n f^m = f^{nm}$.

Бесконечная последовательность элементов $(a_i)_{i=1}^\infty$ называется **заключительно периодической**, если $\exists m, k \in \mathbb{N} (\forall i > m (a_i = a_{i+k}))$. Число k называется **периодом**. Если число m = 1, то последовательность называется просто **периодической**.

Т. Если функция на конечном множестве тотальна, то последовательность значений степеней этой функции для любого аргумента заключительно периодическая.

Представление функций в программах: массив либо процедура (называемая функцией).

1.7. Отношения эквивалентности (теорема о гомоморфизме)

На 1 балл:

Классы эквивалентности: Отношение эквивалентности, измельчение разбиения. **Ядро**

функционального отношения. **Фактормножество:** $\text{nat } R(x)$ — натуральное отображение, факторизация, индуцированная функция, разложение функции. **Множество уровня.** **Гомоморфизмы и изоморфизмы:** Теорема о гомоморфизме. **End**

На 2 балла:

Рефлексивное симметричное транзитивное отношение называется **отношением эквивалентности**.

Обычно отношение эквивалентности обозначают знаком \equiv .

\equiv — отношение эквивалентности на множестве M , и $x \in M$. Подмножество элементов множества M , эквивалентных x , называется **классом эквивалентности** для x : $[x] \equiv (\text{def}):= \{y \in M \mid y \equiv x\}$.

Леммы: $\forall a \in M ([a] \neq \emptyset); a \equiv b \Rightarrow [a] = [b]; a$ не эквивалентно $b \Rightarrow [a] \cap [b] = \emptyset$

Т. Если \equiv — отношение эквивалентности на множестве M , то классы эквивалентности по этому отношению образуют разбиение множества M , причём среди элементов разбиения нет пустых. И обратно, всякое разбиение $B = \{B_i\}$ множества M , не содержащее пустых элементов, определяет отношение эквивалентности на множестве M , классами эквивалентности которого являются элементы разбиения.

Всякая функция f , будучи отношением, имеет **ядро** $f \circ f^{-1}$, которое является отношением на области определения функции.

Т. Ядро функционального отношения является отношением эквивалентности на множестве определения.

Если R — отношение эквивалентности на множестве M , то множество классов эквивалентности называется **фактормножеством** множества M относительно эквивалентности R и обозначается M/R : $M/R (\text{def}):= \{[x]_R\}_{x \in M}$. Фактормножество является подмножеством булеана: $M/R \subset 2^M$.

Т. Если $f : A \rightarrow B$ функция, то $| \text{Dom } f / (f \circ f^{-1}) | = | \text{Im } f |$.

Функция $\text{nat } R : M \rightarrow M/R$ называется **отождествлением** (или **натуральным отображением**, или **канонической проекцией**) и определяется следующим образом: $\text{nat } R(x) (\text{def}) := [x]_R$.

Использование фактормножества вместо множества называется **факторизацией**.

Определим **индуцированную функцию** из фактормножества в множество B : $f/\ker f := \{(s, b) \mid s = [x]\ker f \& b = f(x)\}$. Тогда исходная функция может быть **факторизована**, или **разложена**, в суперпозицию $f = (\text{nat } f) \circ (f/\ker f)$.

$f : A \rightarrow B$ — функция и $f \circ f^{-1}$ — отношение эквивалентности на $\text{Dom } f$. Рассмотрим фактормножество $\text{Dom } f / (f \circ f^{-1})$. Класс эквивалентности (элемент фактормножества) — это подмножество элементов A , которые отображаются в один и тот же элемент $b \in B$. Такие множества называются **множествами уровня** функции f .

Пусть R — n -местное отношение на множестве X , причём $R \subset X^n$; а S — n -местное отношение на множестве Y , причём $S \subset Y^n$, $n > 1$. Тогда функция $f : X \rightarrow Y$ называется **гомоморфизмом**, если $\forall x_1, \dots, x_n \in X (R(x_1, \dots, x_n) = S(f(x_1), \dots, f(x_n)))$.

Т. Если $f : X \rightarrow Y$ — изоморфизм, то $f^{-1} : Y \rightarrow X$ — тоже изоморфизм.

Т. Изоморфность является отношением эквивалентности.

Т. [О гомоморфизме] Если $f : X \rightarrow Y$ — гомоморфизм, то $\text{Dom } f / \ker f \sim \text{Im } f$.

1.8. Отношения порядка (теорема о наименьшей неподвижной точке функции)

На 1 балл:

Отношение порядка позволяет сравнивать между собой различные элементы одного множества.

План: частичный и линейный порядок, минимальный и наименьший элементы, верхние и нижние границы, монотонные и непрерывные функции, наименьшая неподвижная точка, вполне упорядоченные множества, индукция

На 2 балла:

Частичный и линейный порядок

Антисимметричное транзитивное отношение называется отношением порядка.

Нестрогое порядка – рефлексивное отношение порядка.

Строгое порядка – антирефлексивное отношение порядка.

Линейного порядка – линейное отношение порядка.

Частичного порядка – не обладающее свойством линейности отношение порядка.

Обычно отношение строгого порядка (линейного или частичного) обозначается знаком \prec , а отношение нестрогого порядка – знаком \preceq .

Пример: отношение $<$ на множестве вещественных чисел является отношением строгое порядка.

Определение: множество, на котором задано отношение частичного порядка – *частично упорядоченное*.

Множество, на котором задано отношение линейного порядка – *линейно упорядоченное*. **Пример:** множество вещественных чисел упорядоченно линейно.

Теорема: если \prec – отношение частичного порядка, то обратное отношение \succ также является отношением частичного порядка.

Лемма: отношение R – антисимметрично и линейно тогда и только тогда, когда дополнение отношения \bar{R} совпадает с обратным отношением R^{-1} везде, за исключением диагонали.

Следствие: если R антисимметрично и линейно, то $\forall a, b (a \neq b \Rightarrow (aRb \Leftrightarrow \neg(aRb)))$.

Теорема: если отношение \prec является отношением линейного порядка, то его дополнение $\not\prec$ также является отношением линейного порядка.

Пример: отношение $<$ является отношением строго линейного порядка, а его дополнение \geq является нестрого линейного порядка на любом подмножестве числовой оси.

Определение: линейное упорядоченное множество $\{a_1, \dots, a_n, \dots\}$ можно рассмотреть, как последовательность $\langle a_1, \dots, a_n, \dots \rangle$, в которой $\forall i (a_i \prec a_{i+1} \& a_i \prec a_{i+1})$. Такие последовательности называются строго монотонно возрастающими.

Определение: длина – количество элементов в конечной последовательности.

Минимальные и наименьшие элементы

Определение: Элемент x множества M с отношением порядка \prec называется минимальным, если в множестве M не существует элементов, меньших, чем x . (аналогично с максимальным).

Теорема: во всяком конечном непустом частично упорядоченном множестве существует минимальный элемент.

Теорема: всякий частичный порядок на конечном множестве может быть дополнен до линейного.

Определение: пусть M – частично упорядоченное множество с отношением порядка \prec , а X – его подмножество: $X \subset M$. Элемент a называется наименьшим в множестве X ($a = \min X$), если $a \in X \& \forall x \in X (x \neq a \Rightarrow a \prec x)$.

Теорема: наименьший элемент, если он существует, является единственным.

Верхние и нижние границы

Пусть $X \subset M$ – подмножество упорядоченного множества M с отношением порядка \prec . Элемент $m \in M$ называется нижней границей для подмножества X , если $\forall x \in X (m \neq x \Rightarrow m \prec x)$. (аналогично для верхней границы). Верхние (Sup) и нижние (Inf) границы не обязательно существуют для любого множества, и если существуют, то не всегда единственны.

Теорема: пусть M – частично упорядоченное множество, а X – любое его подмножество. Тогда: если $\exists a = \min X$, то $a = \inf X$, если $\exists a = \inf X$ и $a \in X$, то $a = \min X$. (аналогично с верхней)

Определение: частично упорядоченное множество линейно полно, если любое его линейно упорядоченное подмножество имеет супремум.

Монотонные и непрерывные функции

Пусть A и B – упорядоченные множества, и $f: A \rightarrow B$. Тогда функция f называется

Монотонно возрастающей: $a_1 < a_2 \& a_1 \neq a_2 \Rightarrow f(a_1) < f(a_2) \vee f(a_1) = f(a_2)$.

Монотонно убывающей: $a_1 < a_2 \& a_1 \neq a_2 \Rightarrow f(a_2) < f(a_1) \vee f(a_1) = f(a_2)$.

Строго монотонно возрастающей: $a_1 < a_2 \& a_1 \neq a_2 \Rightarrow f(a_1) < f(a_2) \& f(a_1) \neq f(a_2)$.

Строго монотонно убывающей: $a_1 < a_2 \& a_1 \neq a_2 \Rightarrow f(a_2) < f(a_1) \& f(a_1) \neq f(a_2)$.

Теорема: суперпозиция одинаково монотонных функций монотонна в том же смысле.

Определение: Функция f называется непрерывной (по Скотту), если \forall последовательности $a_1 < \dots < a_n < \dots$, для которой существует супремум, выполнено равенство: $f(\sup\{a_i\}) = \sup f(a_i)$.

Теорема: непрерывная функция монотонно возрастает.

Теорема: суперпозиция непрерывных функций непрерывна.

Наименьшая неподвижная точка функции

Функция $f: M \rightarrow M$. Элемент $x \in M$ называется неподвижной точкой функции f , если $f(x) = x$.

Теорема: если множество X частично упорядоченно, линейно полно, имеет наименьший элемент $0 = \min X$, и функция $f: X \rightarrow X$ непрерывна и тотальна, то элемент $a = \sup\{f^n(0) | n \geq 0\}$ является наименьшей неподвижной точкой функции f .

Вполне упорядоченные множества

Частично упорядоченное множество X называется вполне упорядоченным, если любое его непустое подмножество имеет наименьший элемент.

Определение: два линейно упорядоченных множества A и B изоморфны, если между ними существует взаимно-однозначное соответствие, сохраняющее порядок:

$A \sim B \stackrel{\text{def}}{=} |A| = |B| \& (\forall a_1, a_2 \in A (a_1 < a_2 \& a_1 \mapsto b_1 \& a_2 \mapsto b_2 \Rightarrow b_1 < b_2))$.

Теорема: линейно упорядоченные множества A и B изоморфны тогда и только тогда, когда между ними существует строго монотонное взаимно-однозначное соответствие.

Замечание: понятие изоморфизма применимо к вполне упорядоченным множествам, поскольку они упорядочены линейно.

Индукция

Теорема (индукция по вполне упорядоченному множеству): пусть X – вполне упорядоченное множество, x_0 – его минимальный элемент, а P – некоторый предикат, зависящий от элементов X . Тогда, если $P(x_0) \& \forall x_1 \in X ((\forall x \in X (x < x_1 \Rightarrow P(x))) \Rightarrow P(x_1))$, то $\forall x \in X (P(x))$.

Замечание: обычная математическая индукция соответствует индукции по вполне упорядоченному множеству \mathbb{N} .

Теорема: любое множество может быть вполне упорядочено.

2.4 Элементарная теория чисел (китайская теорема об остатках)

На 1 балл:

План: делимость чисел, НОД и НОК, линейное представление НОД, простые числа, сравнения, системы вычислов, китайская теорема об остатках, вычисления в остаточных классах, функция Эйлера.

На 2 балла:

Делимость чисел

Число $a \in \mathbb{N}$ делится на число $b \in \mathbb{N}$ (обозначается $a|b$), если $\exists q \in \mathbb{N} (a = bq)$

Теорема: $\forall a, b \in \mathbb{N} (\exists q \in \mathbb{N}_0 (a = bq + r \& 0 \leq r < b))$. (q и r – единственны)

Определение: отношение делимости рефлексивно $\forall a \in \mathbb{N} (a|a)$, транзитивно $\forall a, b, c \in \mathbb{N} (a|b \& b|c \Rightarrow a|c)$ и антисимметрично $\forall a, b \in \mathbb{N} (a|b \& b|a \Rightarrow a = b)$. Таким образом, делимость чисел – это нестрогий частичный порядок на множестве \mathbb{N} .

Наибольший общий делитель и наименьшее общее кратное

Натуральное число c такое, что $c|a$ и $c|b$, называется общим делителем чисел a и b . Множество общих делителей обозначается $D(a, b)$. Наибольшее из таких чисел называется НОД ($\gcd(a, b)$).

Если $\gcd(a, b) = 1$, то a и b называются взаимно простыми ($a \perp b$).

Лемма: если $b|a$, то множество общих делителей чисел a, b совпадает с множеством делителей b .

Следствие: $b|a \Rightarrow \gcd(a, b) = b$.

Лемма: если $a = bq + r$, то $D(a, b) = D(q, r)$.

Число d называется общим кратным чисел a и b , если $a|d$ и $b|d$. Множество общих кратных $K(a, b) \stackrel{\text{def}}{=} \{d \in \mathbb{N} \mid a|d \& b|d\}$. Наименьшее из таких чисел называется НОК ($\text{lcm}(a, b)$) или $\text{lcm}(a, b)$.

Теорема: $\forall a, b, m \in \mathbb{N}$ справедливо: 1) $a|m \& b|m \Rightarrow \text{lcm}(a, b)|m$ 2) $\gcd(a, b) \text{lcm}(a, b) = ab$

Линейное представление наибольшего общего делителя

Теорема: $\forall a, b \in \mathbb{N} (\exists x, y \in \mathbb{Z} (\gcd(a, b) = ax + by))$. Такое представление НОД называется соотношением Безу, или линейным представлением, а числа x и y – коэффициентами Безу

Определение: линейные уравнение, в которых все числа целые, как известные коэффициенты, так и не известные величины, называются диофантовыми.

Теорема: диофантово уравнение $ax + by = c$ разрешимо тогда и только тогда, когда $\gcd(a, b)|c$.

Простые числа

Натуральное число $a > 1$ называется простым, если оно не имеет собственных делителей.

Теорема: $\forall a \in \mathbb{N} \& a > 1$ его наименьший делитель, отличный от 1, является простым числом.

Теорема: простых чисел бесконечно много

Для построения множества простых чисел, не превосходящих данного числа n , используют способ, называемый решетом Эратосфена

Лемма: $\forall a, b, c \in \mathbb{N} (b \perp c \Rightarrow \gcd(ac, b) = \gcd(a, b))$.

Следствие: $\forall a, b, c \in \mathbb{N} (b \perp c \& b|ac \Rightarrow b|a)$.

Лемма Евклида: $\forall a, b \in \mathbb{N}, p \in \mathbb{P} (p|ab \Rightarrow p|a \vee p|b)$.

Следствие: $\forall a_1, a_2, \dots, a_n \in \mathbb{N}, p \in \mathbb{P} (p|a_1a_2 \dots a_n \Rightarrow \exists i (p|a_i))$.

Теорема (основная теорема арифметики): любое натуральное число, большее 1, можно представить в виде произведения простых множителей, причем единственным образом.

Сравнения

Пусть числа $a, b, m \in \mathbb{N}$. Говорят, что число a сравнимо с числом b по модулю m , если оба числа имеют одинаковый остаток при делении на m , то есть $a = pm + r, b = qm + r$. ($a \equiv b \pmod{m}$)

Теорема: пусть $a, b \in \mathbb{N}, a > b$. Число a сравнимо с числом b по модулю m тогда и только тогда, когда m является делителем $a - b$: $a \equiv b \pmod{m} \Leftrightarrow m|(a - b)$.

Следствие: отношение $a \equiv b \pmod{m}$ является отношением эквивалентности на множестве \mathbb{N} .

Лемма: пусть $a \equiv b \pmod{m}$ и $c \equiv d \pmod{m}$. Тогда $(a + c) \equiv (b + d) \pmod{m}$.

Лемма: пусть $a \equiv b \pmod{m}$ и $c \equiv d \pmod{m}$. Тогда $ac \equiv bd \pmod{m}$.

Лемма: если $a \equiv b \pmod{m}, n \in \mathbb{N}$, то $a^n \equiv b^n \pmod{m}$.

Теорема: пусть $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$ – многочлен с натуральными коэффициентами. Тогда если $a \equiv b \pmod{m}$, то $f(a) \equiv f(b) \pmod{m}$.

Система вычетов

Поскольку деление с остатком определено для всех целых чисел, отношение сравнения по модулю также определяется на все множество целых чисел.

Классом вычетов по модулю m (обозначение $[a]_m$) называется класс эквивалентности по отношению сравнимости: $[a]_m \stackrel{\text{def}}{=} \{b \in \mathbb{Z} \mid b \equiv a \pmod{m}\}, a \in \mathbb{Z}, m \in \mathbb{N}$

Замечание: Слово «вычет» употребляется как синоним слова «остаток». Всего классов вычетов по модулю m ровно m и один класс вычетов составляет все числа, дающие один и тот же остаток при делении на m .

Определим над классом вычетов операции сложения и умножения

$$[x]_m +_m [y]_m \stackrel{\text{Def}}{=} [(x+y)]_m, \quad [x]_m *_m [y]_m \stackrel{\text{Def}}{=} [(xy)]_m$$

Множество классов вычетов с введенными операциями обозначается \mathbb{Z}_m .

Определение: полной системой вычетов по модулю m называют множество целых чисел, содержащее ровно по одному элементу из каждого класса вычетов по модулю m .

Теорема (о полной системе вычетов): пусть $\{a_1, \dots, a_n\}$ – полная система вычетов по модулю m , x, y – целые числа. $x \perp m$. Тогда $\{y + xa_1, \dots, y + xa_m\}$ – полная система вычетов по модулю m .

Определение: приведенной системой вычетов по модулю m называют множество целых чисел, взаимно простых с числом m , и содержащихся в различных классах вычетов по модулю m . Число элементов в приведенной системе вычетов равно количеству чисел, взаимно простых с m и меньших m .

Теорема: если $\{a_1, \dots, a_n\}$ – приведенная система вычетов по модулю m . $x \in \mathbb{Z}, x \perp m$, то $\{xa_1, \dots, xa_n\}$ – приведенная система вычетов по модулю m .

Китайская теорема об остатках

Была сформулирована в Китае, предположительно в 3 веке н.э., китайским математиком Сунь Цзы.

Теорема: для любых попарно взаимно простых натуральных чисел a_1, \dots, a_n и для любых целых чисел r_1, \dots, r_n таких, что $\forall i \in 1 \dots n (0 \leq r_i < a_i)$ существует такое число s , что деление его на каждое из чисел a_i дает в остатке число r_i .

Следствие: пусть $m := m_1 m_2 \dots m_n, \forall i, j \in 1 \dots n, i \neq j (m_i \perp m_j), x, a \in \mathbb{Z}$. Тогда $x \equiv a \pmod{m} \Leftrightarrow \forall i \in 1 \dots n (x \equiv a \pmod{m_i})$.

Вычисления в остаточных классах

Пусть $m := m_1 m_2 \dots m_n, \forall i, j \in 1 \dots n, i \neq j (m_i \perp m_j), x, a \in \mathbb{Z}$. По следствию к китайской теореме об остатках любое число $a \in 0 \dots (m-1)$ однозначно определяется остатками a_1, \dots, a_n от деления числа a на числа m_1, \dots, m_n . Такая непозиционная система счисления называется системой остаточных классов, или модулярной арифметикой. Набор чисел m_1, \dots, m_n называется основанием системы остаточных классов. Обозначение $a = (a_1, \dots, a_n)$.

Функция Эйлера

Функцией Эйлера от натурального аргументы n называется количеством чисел меньших n и взаимно простых с n . Стандартным обозначением для функции Эйлера является $\varphi(n)$. Из определения ясно, что $n > 1 \Rightarrow \varphi(n) < n$. По определению $\varphi(1) := 1$.

Числовая функция f называется аддитивной, если она уважает сложение, т.е. $f(x+y) = f(x) + f(y)$, и называется мультипликативной, если она уважает умножение, т.е. $f(xy) = f(x)f(y)$.

Теорема Эйлера: если $a \perp n$, то $a^{\varphi(n)} \equiv 1 \pmod{n}$.

Следствие (малая теорема Ферма): пусть p – простое число. Тогда $a^{p-1} \equiv 1 \pmod{p}$ для любого a , не делящегося на p .

2.6. Решётки (частичный порядок в решётке)

На 1 балл:

2.6.1 Дистрибутивные и ограниченные решётки - принцип двойственности, нижняя и верхняя грань решётки

Термины: решетка, свойства решетки, дистрибутивная решетка

Теорема о нижней(верхней) грани решетки

2.6.2 Решётки с дополнением

Теорема о свойствах дополнения

2.6.3 Частичный порядок в решетке - нижняя и верхняя границы, нижняя и верхняя грани

Теорема об отношении частичного порядка

Определение нижней и верхней грани элементов

2.6.4-5 Полная решетка (с супремумом и инфимумом), полурешетка (с одной операцией пересечение)

На 2 балла:

Определения:

Решётка — это множество M с двумя бинарными операциями, \cap и \cup , такими, что выполнены следующие условия (аксиомы решётки): идемпотентность, коммутативность, ассоциативность, поглощение

Принцип двойственности — если поменять пересеч. С объединением, все аксиомы будут работать

Дистрибутивная решетка — если выполняется $a \cup (b \cap c) = (a \cup b) \cap (a \cup c)$, И $a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$ (в других терминах это гомоморфизм)

Если в решетки есть 0 — то это **нижняя грань**, 1 — **верхняя**. С обеими — решетка **ограниченная**

Теорема. $a \cap b = b \iff a \cup b = a$.

Дополнение элемента a - если $(a \cap a' = 0)$ и $(a \cup a' = 1)$. Если так у каждого элемента — решетка с дополнением.

Теорема. [о свойствах дополнения] В ограниченной дистрибутивной решётке с дополнением выполняется следующее:

- 1) дополнение a' единственно;
- 2) дополнение инволютивно: $a'' = a$;
- 3) грани дополняют друг друга: $1' = 0, 0' = 1$;
- 4) законы де Моргана: $(a \cap b)' = a' \cup b', (a \cup b)' = a' \cap b'$.

В любой решётке можно естественным образом ввести частичный порядок, а именно:

$$a \prec b \stackrel{\text{Def}}{=} a \cap b = a.$$

Теорема. Отношение \prec является отношением частичного порядка (частич. порядок -Рефлексивность, Антисимметричность, Транзитивность)

Элемент x называется **нижней(верхней) грани элементов** a и b , если x — нижняя(верхняя) граница элементов a и b и для любой другой нижней(верхней) границы v элементов a и b выполняется $v \prec x$ ($x \prec v$). Это **инфимум** и **супремум** соответственно.

Теорема. Если в частично упорядоченном множестве для любых двух элементов существуют нижняя и верхняя грани, то это множество образует решётку относительно \inf и \sup .

Полная решетка — решетка, которая имеет инфимум и супремум. (пример — булеван)

Полурешетка — множество M с операцией \cap (пересечение), при условии если оно идемпотентно, ассоциативно и коммутативно.

3.1. Элементарные булевы функции (функции k-значной логики)

На 1 балл:

3.1.1. Функции алгебры логики - Булевы функции, истинностные значения, таблица истинности, установленный порядок

Теорема о мощности множества булевых функций от n переменных

3.1.2. Существенные и несущественные переменные

3.1.3. Булевы функции одной переменной

Константы, 0 — нуль, тождественная функция, $\neg x$ — отрицание, 1 — единица

3.1.4.(ВАЖНО!) Булевы функции двух переменных

$\&$ — конъюнкция, $+2$ — сложение по модулю, $|$ — штрих Шеффера и другие

3.1.5. Функции k-значной логики

На 2 балла:

Определения:

Функции $f: E_2^n \rightarrow E_2$, где $E_2 \stackrel{\text{def}}{=} \{0, 1\}$ называются функциями алгебры логики, или **булевыми функциями** от n переменных. Элементы множества E_2 называются **истинностными значениями**.

Множество булевых функций от n переменных обозначим P_n .

Булеву функцию от n переменных можно задать **таблицей истинности**.

Теорема. $|P_n| = 2^{2^n}$.

Булева функция f существенно зависит от переменной x_i , если существует набор $a_1..a_{i-1}, a_{i+1}..a_n$, что $f(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_n) \neq f(a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_n)$.

В таком случае x_i – **существенная переменная**. В противном случае – **несущественная**.

Булевы функции одной переменной:

		Переменная x	0	1	
Название	Обозначение				Несущественные
Нуль		0	0	0	x
Тождественная		x	0	1	
Отрицание	$\neg x, \bar{x}, x', \sim x$	1	0		
Единица		1	1	1	x

Булевы функции двух переменных:

		Переменная x	0	0	1	1	
		Переменная y	0	1	0	1	
Название	Обозначение					Несущие	
Нуль		0	0	0	0	x, y	
Конъюнкция	$\cdot, \&, \wedge$	0	0	0	1		
		0	0	1	0		
		0	0	1	1	y	
		0	1	0	0		
		0	1	0	1	x	
Сложение по модулю 2	$+, +_2, \not\equiv, \oplus, \Delta$	0	1	1	0		
Дизъюнкция	\vee	0	1	1	1		

		Переменная x	0	0	1	1	
		Переменная y	0	1	0	1	
Название	Обозначение					Нес.	
Стрелка Пирса	\downarrow		1	0	0	0	
Эквивалентность	\equiv		1	0	0	1	
			1	0	1	0	x
			1	0	1	1	
			1	1	0	0	y
Импликация	$\rightarrow, \Rightarrow, \supset$		1	1	0	1	
Штрих Шеффера	\mid		1	1	1	0	
Единица			1	1	1	1	x, y

Рассмотрим множество $E_k = \{0, 1, \dots, k-1\}$ и множество функций $P_{k,n}$, имеющих n аргументов типа E_k , и возвращающих значение типа E_k . Такие функции называются **функциями k-значной логики** и являются обобщением функций алгебры логики: $P_{2,n} = P_n$

Функцию k-значной логики можно представить **как систему функций обычной двузначной логики** следующим образом. Функция $f(x_1, \dots, x_n) \in P_{k,n}$, представляется системой функций $\{g_1(y_{11}, \dots, y_{1m}), \dots, g_m(y_{11}, \dots, y_{1m}, \dots, y_{n1}, \dots, y_{nm})\}$, где $m = [\log k]$. Здесь значения функции f и переменных x_i принадлежат множеству E_k , а значения функций g_j и переменных y_{ij} принадлежат множеству E_2 . Другими словами, функция g_j вычисляет j-ю цифру в двоичном представлении числа $f(x_1, \dots, x_n)$.

3.3 Двойственность и симметрия

На 1 балл:

- 3.3.1 Двойственные и самодвойственные функции
- 3.3.2 Реализация двойственной функции
- 3.3.3 Принцип двойственности
- 3.3.4 Симметрические функции

На 2 балла:

3.3.1 Двойственные и самодвойственные функции

Пусть $f(x_1, \dots, x_n) \in P_n$ - булева функция. Тогда функция $f^*(x_1, \dots, x_n) \stackrel{\text{Def}}{=} \overline{f(\bar{x}_1, \dots, \bar{x}_n)}$, называется **двойственной** к функции f . Из определения видно, что двойственность инволютивна: $f^{**} = f$, и по этой причине отношение “Быть двойственной к” на множестве булевых функций симметрично, то есть если $f^* = g$, то $g^* = f$. Если в таблице истинности булевой функции f инвертировать все значения, то получим таблицу истинности двойственной функции f^* . Если функции двойственны, то их отрицания также двойственны. Функция называется **самодвойственной**, если $f^* = f$.

Теорема(количество самодвойственных функций). Существует $2^{2^{n-1}}$ самодвойственных функций n переменных.

3.3.2 Реализация двойственной функции

Формула, реализующая двойственную функцию, определенным образом связана с формулой, реализующей исходную функцию.

Теорема(). Если функция $\varphi(x_1, \dots, x_n)$ реализована формулой $f(f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))$, то формула $f^*(f_1^*(x_1, \dots, x_n), \dots, f_n^*(x_1, \dots, x_n))$ реализует функцию $\varphi^*(x_1, \dots, x_n)$

3.3.3 Принцип двойственности

Принцип двойственности устанавливает связь между структурами формул, реализующих пару двойственных функций. Рассмотрим две системы булевых функций, $F = \{f_1, \dots, f_m\}$ и $F^* = \{f_1^*, \dots, f_m^*\}$, и введем обозначение $\mathcal{F}^*[F^*] \stackrel{\text{Def}}{=} \mathcal{F}[F][f_i^*/f_i]_{i=1}^m$.

Теорема(Принцип двойственности) Пусть $F = \{f_1, \dots, f_m\}$ - система булевых функций, а $F^* = \{f_1^*, \dots, f_m^*\}$ - система двойственных функций. Тогда если формула \mathcal{F} , над базисом F^* , полученная заменой функций f_i двойственными функциями f_i^* , реализует функцию f^* : $\text{func } \mathcal{F}[F] = f \implies \text{func } \mathcal{F}^*[F^*] = f^*$.

3.3.4 Симметрические функции

Булева функция n переменных называется **симметрической**, если её значение инвариантно относительно перестановок значений аргументов. Заметим, что при перестановке значений в наборе общее количество нулей и единиц сохраняется. Понятие симметрической функции применимо только к функциям двух и большего числа переменных.

Теорема(о количестве симметрических функций). Существует 2^{n+1} различных симметрических булевых функций n переменных.

3.4 Нормальные формы

На 1 балла:

- 3.4.1 Разложение булевых функций по переменным
- 3.4.2 Минимальные термы
- 3.4.3 Совершенные нормальные формы
- 3.4.4 Эквивалентные преобразования
- 3.4.5 Минимальные дизъюнктивные формы

На 2 балла:

3.4.1 Разложение булевых функций по переменным

Теорема. [О разложении булевой функции по переменным]

$$f(x_1, \dots, x_m, x_{m+1}, \dots, x_n) = \bigvee_{(\sigma_1, \dots, \sigma_m)} x_1^{\sigma_1} \wedge \dots \wedge x_m^{\sigma_m} \wedge f(\sigma_1, \dots, \sigma_m, x_{m+1}, \dots, x_n), \text{ где}$$

дизъюнкция берётся по всем возможным наборам $(\sigma_1, \dots, \sigma_m)$.

3.4.2 Минимальные термы

Выражение вида $x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n}$ имеет большое значение в теории булевых функций и носит специальное название: **минимальный терм**, или минтерм. Нетрудно видеть, что минтерм — это реализация булевой функции n переменных, которая имеет значение 1 ровно на одном наборе значений $(\sigma_1, \dots, \sigma_n)$ переменных, а именно на наборе 2^n различных минтермов. Любую булеву функцию, кроме 0, можно представить, как дизъюнкцию некоторых минтермов. Ясно, что при фиксированном n имеется ровно 2^n различных минтермов.

Говорят, что система булевых функций **ортогональна**, если их конъюнкция есть тождественный ноль.

Лемма.[1] Любое множество различных минтермов ортогонально.

Лемма.[2] Любое множество минтермов выполнимо.

3.4.3 Совершенные нормальные формы

Совершенной дизъюнктивной нормальной формой (СДНФ) называется реализация булевой функции $f(x_1, \dots, x_n)$ в виде формулы $\bigvee_{\{(\sigma_1, \dots, \sigma_n) | f(\sigma_1, \dots, \sigma_n) = 1\}} x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n}$

Теорема. Всякая булева функция имеет единственную СДНФ.

Теорема. Всякая булева функция имеет **единственную совершенную конъюнктивную нормальную**

$$f(x_1, \dots, x_n) = \bigwedge_{\{(\sigma_1, \dots, \sigma_n) | f^*(\sigma_1, \dots, \sigma_n) = 1\}} x_1^{\sigma_1} \vee \dots \vee x_n^{\sigma_n}.$$

форму (СКНФ):

3.4.4 Эквивалентные преобразования

Преобразование формулы в равносильную ей называется **эквивалентным преобразованием**.

Теорема. Для любых двух равносильных формул F_1 и F_2 существует последовательность эквивалентных преобразований из F_1 в F_2 , получаемая посредством равносильностей

3.4.5 Минимальные дизъюнктивные формы

Дизъюнктивной формой называется формула вида $\bigvee_{i=1}^k K_i$, где $K_i = \bigwedge_{p=1}^{m_i} x_{j_p}^{\sigma_p}$, K_i — элементарная конъюнкция. Количество переменных в конъюнкции называется её **рангом**

Теорема. Число различных дизъюнктивных форм n переменных равно 2^{3^n}

3.5 Представление булевых функций в программах

На 1 балл:

1 - Табличные представления

2 - Строковые представления

3 - Алгоритм вычисления значения булевой функции

4 - Представление булевых функций арифметическими полиномами

5 - Карты Карно и матрицы минтермов

6 - Булевы функции и карты Карно

7 - Минимизация формул картами Карно

8 - Деревья решений

На 2 балла:

1) Самое бесхитростное представление прямо воспроизводит таблицу истинности. Если условиться, что кортежи в таблице всегда идут в установленном порядке, то для представления булевой функции $f(x_1, \dots, x_n)$ достаточно хранить столбец значений: $F1 : array [0..(2n - 1)] of 0..1$.

Пример. Для булевой функции $g(x, y)$ имеем $F1(g) : array [0..7] of 0..1 := (1, 1, 1, 0, 0, 0, 1, 0)$, (g - мажоритарная функция). Либо пример проще, для конъюнкции $\&$ от двух аргументов имеем: $F1(\&) : array [0..3] of 0..1 := (0, 0, 0, 1)$.

Данное представление — не самое эффективное. В частности, если набор аргументов задан массивом $x : array [1..n] of 0..1$, то для вычисления значения функции f с помощью представления $F1$ необходимо сначала вычислить индекс d (то есть номер кортежа в установленном порядке), чтобы затем получить значение $F1[d]$. Индекс d нетрудно вычислить, например, с помощью следующего алгоритма вычисления номера кортежа

Вход: кортеж $x : array [1..n] of 0..1$ значений переменных.

Выход: номер d кортежа x при перечислении кортежей в установленном порядке.

d := 0 // начальное значение индекса

for i from 1 to n do

d := d * 2 + x[i] // сдвигаем влево и добавляем разряд

end for

Обоснование. Если рассматривать кортеж булевых значений как двоичную запись числа, то это число — номер кортежа в установленном порядке. Значение числа d , заданного в позиционной двоичной системе счисления цифрами $x_1 \dots x_n$, определяется следующим образом:

$$d = x_1 2^{n-1} + \dots + x_n 2^0 = \sum_{i=1}^n x_i 2^{n-i} = 2(2(\dots(2x_1 + x_2)\dots) + x_{n-1}) + x_n.$$

Цикл в алгоритме непосредственно вычисляет последнюю формулу, которая является частным случаем **схемы Горнера**.

Более эффективным представлением таблицы истинности является использование n -мерного массива: $F2 : array [0..1, \dots, 0..1] of 0..1$. В случае использования представления $F2$ значение функции $f(x_1, \dots, x_n)$ задаётся выражением $F2[x_1, \dots, x_n]$.

Пример. Для функции g имеем $F2 : array [0..1, 0..1, 0..1] of 0..1 = (((1, 1), (1, 0)), ((0, 0), (1, 0)))$. Для функции $\&$ имеем $F2 : array [0..1, 0..1] of 0..1 = ((0, 0), (0, 1))$.

2) Булеву функцию можно представить с помощью реализующей её формулы. Реализуем булеву функцию с помощью через формулу СДНФ (т.к. она единственна, если соблюдать лексиграфический порядок). СДНФ булевой функции может быть построена по заданной таблице истинности с помощью следующего алгоритма построения СДНФ.

Вход: вектор $x : array [1..n] of string$ идентификаторов переменных, вектор $F1 : array [0..(2n - 1)] of 0..1$ значений функции при установленном порядке кортежей.

Выход: последовательность символов, образующих запись формулы СДНФ для заданной функции.

```

 $f := \text{false}$  //признак присутствия левого операнда дизъюнкции
for  $i$  from 0 to  $2^n - 1$  do
    if  $F_1[i] = 1$  then
        if  $f$  then yield ' $\vee'$  else  $f := \text{true}$  end if //знак дизъюнкции
         $g := \text{false}$  //признак присутствия левого операнда конъюнкции
        for  $j$  from 1 to  $n$  do
            if  $g$  then yield ' $\wedge'$  else  $g := \text{true}$  end if //знак конъюнкции
             $v := (i \text{ div } 2^{j-1}) \text{ mod } 2$  // $j$ -й разряд  $i$ -го кортежа
            if  $v = 0$  then
                yield ' $\neg'$  //знак отрицания
            end if
            yield  $x[j]$  //добавление в формулу переменной
        end for
    end if
end for

```

Обоснование. Данный алгоритм буквально воспроизводит словесную запись следующего правила: для каждой строки таблицы истинности, для которой значение функции равно 1, построить дизъюнктивное слагаемое, включающее все переменные, причём те переменные, которые имеют значение 0 в этой строке, входят со знаком отрицания. Остальное в этом алгоритме — мелкие программистские «хитрости», которые полезно один раз посмотреть, но не стоит обсуждать.

Пример. Для функции g , используемой в примерах данного раздела, алгоритм построит строку $\neg x \wedge \neg y \wedge \neg z \vee \neg x \wedge \neg y \wedge z \vee \neg x \wedge y \wedge \neg z \vee x \wedge y \wedge \neg z$.

3) Рассмотрим алгоритм вычисления значения **булевой функции**, заданной в виде СДНФ, для заданных значений переменных x_1, \dots, x_n . В этом алгоритме используется следующее представление данных. СДНФ задана массивом $F_3 : \text{array}[1..k, 1..n]$ of 0..1, где строка $F_3[i, *]$ содержит набор значений $\sigma_1, \dots, \sigma_n$, для которого $f(\sigma_1, \dots, \sigma_n) = 1$, $i \in 1..k$.

Пример. Для функции g $F_3 : \text{array}[1..4, 1..3]$ of 0..1 = $((0, 0, 0), (0, 0, 1), (0, 1, 0), (1, 1, 0))$.

Вход: массив, представляющий СДНФ: $f : \text{array}[1..k, 1..n]$ of 0..1;
множество значений переменных $x : \text{array}[1..n]$ of 0..1.

Выход: 0..1 — значение булевой функции.

```

for  $i$  from 1 to  $k$  do
    for  $j$  from 1 to  $n$  do
        if  $f[i, j] \neq x[j]$  then
            next for  $i$  // $x_j \neq \sigma_j \implies x_j^{\sigma_j} = 0 \implies x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n} = 0$ 
        end if
    end for
    return 1 // $x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n} = 1 \implies \bigvee_{(\sigma_1, \dots, \sigma_n)} x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n} = 1$ 
end for
return 0 //все слагаемые в дизъюнкции равны нулю

```

Обоснование. Алгоритм основан на следующих правилах. Можно прекратить вычисление конъюнкции, как только получен конъюнктивный сомножитель, равный 0 (вся конъюнкция имеет значение 0). Можно прекратить вычисление дизъюнкции, как только получено дизъюнктивное слагаемое, равное 1 (вся дизъюнкция имеет значение 1).

Этот алгоритм в худшем случае выполняет $k \cdot n$ сравнений, а в среднем — гораздо меньше. Таким образом, он существенно эффективнее общего алгоритма интерпретации.

4) Представление полиномом синяя звезда, поэтому тут нет

5) **Карта Карно (или Диаграмма Вейча)** для представления функций n переменных является прямоугольной таблицей, которая содержит 2^n ячеек. Положим $k := n \text{ div } 2$. Если n чётное, то таблица содержит 2^k строк и 2^k столбцов, а если нечётное, то для удобства отображения обычно принимают, что таблица содержит 2^k строк и $2^{(k+1)}$ столбцов. Обозначим число строк в таблице $r(n) := 2k$, **число столбцов в таблице** $c(n) := 2k + (n \text{ mod } 2)$.

Поскольку всего существует 2^n минтермов для n переменных, можно считать, что каждая ячейка карты Карно символизирует один минтерм, а вся карта в целом символизирует перечисление минтермов.

Однако нам нужно построить матрицу минтермов, обладающую двумя полезными свойствами.

1) Прямоугольную таблицу можно рассматривать как развертку тора, то есть считать верхнюю сторону смежной с нижней, а правую — с левой. В этом случае у каждой ячейки имеются четыре соседа.

2) Можно расположить минтермы так, что любые два соседних отличаются ровно в одном разряде, как коды Грея и смежные вершины единичного гиперкуба.

Разобьём множество переменных на две группы, в одной $r(n)$ переменных, а в другой $c(n)$ переменных. Множество кортежей значений переменных каждой группы перечислим не в установленном порядке, а в порядке, задаваемом алгоритмом п. 1.3.4 (код Грея).

Эти коды отмечают строки и столбцы матрицы минтермов, а минтерм в ячейке получается конкатенацией кодов строки и столбца.

Вход: число переменных n .

Выход: матрица минтермов

```
M : array [0..(r(n) - 1), 0..(c(n) - 1)] of array [1..n] of 0..1
for i from 0 to r(n) - 1 do
    for j from 0 to c(n) - 1 do
        M[i, j] := G(i)G(j) //конкатенация кодов
    end for
end for
```

Пример. Для $n = 2, n = 3$.

$n = 2$		x_2	
		0	1
x_1	0	00	01
	1	10	11

$n = 3$		x_2, x_3			
		00	01	11	10
x_1	0	000	001	011	010
	1	100	101	111	110

Хранить в программе информацию, представленную в этих таблицах, совершенно не обязательно, поскольку при фиксированном способе заполнения матрицы легко вычислить минтерм, находящийся в клетке $M[i, j]$. Другими словами, функция $K(i, j)$, доставляющая код минтерма в ячейке $M[i, j]$, вычисляется так:

$$K(i, j) := G(i)G(j) = (B(i) +_2 (B(i) \text{ div } 2))(B(j) +_2 (B(j) \text{ div } 2)),$$

где функция $G(i)$ доставляет i -й зеркальный код Грея (см. теорему п. 1.3.4), функция $B(i)$ доставляет двоичный код числа i , а операция конкатенации битовых шкал никак не обозначается

6) Поскольку всякая булева функция представляется дизъюнкцией минтермов, указание некоторого подмножества ячеек матрицы минтермов является представлением булевой функции.

Рассмотрим булеву матрицу $K : \text{array}[0..(r(n)-1), 0..(c(n)-1)] \text{ of } 0..1$ того же размера, что и матрица минтермов. Каждой строке таблицы истинности булевой функции f соответствует некоторый минтерм. Если значение в этой строке равно 1, то записываем 1 в соответствующую ячейку матрицы, в противном случае записываем 0. Булеву матрицу K назовём **карточкой Карно** булевой функции f .

Пример. Функция $g(x, y, z)$, заданная в примере п. 3.5.1, имеет следующее представление картой Карно
 $F4(g) : \text{array}[0..1, 0..3] \text{ of } 0..1 := ((1, 1, 0, 1), (0, 1, 0, 0))$. Карту Карно для $\&$,

$F4(\&) : \text{array}[0..1, 0..1] := ((0, 0), (0, 1))$.

Карта Карно действительно является представлением булевой функции, поскольку с её помощью можно выполнять основную присущую функциям операцию: вычислять значение функции по заданным значениям аргументов. Вообще говоря, если булева функция представлена картой Карно и задан набор значений переменных $\sigma_1, \dots, \sigma_n$, то ответ получается за один шаг: значение функции на заданном наборе содержится в ячейке, соответствующей минтерму $(x_1^{\sigma_1} \& \dots \& x_n^{\sigma_n})$.

7) Карты Карно являются наглядным средством решения задачи минимизации дизъюнктивных форм, обсуждаемой в пп. 3.4.5–3.4.7.

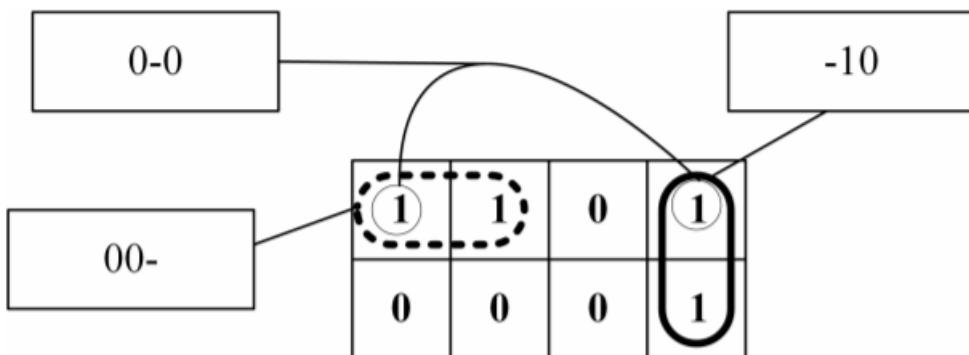
Пусть булева функция представлена картой Карно. Если при этом в соседних ячейках оказались 1, то по основному свойству карт Карно соответствующие мinterмы отличаются точно в одном разряде, и значит, к ним можно применить правило склеивания (п. 3.4.4). В таком случае дизъюнкцию данных мinterмов можно заменить элементарной конъюнкцией (п. 3.4.5) ранга $n - 1$. На карте Карно соответствующие ячейки можно объединить, показывая, что они образуют **компактную группу**.

Результат склеивания двух мinterмов обычно также обозначают мinterмом, причём в позиции разряда, по которому проведено склеивание, поставлен прочерк.

Пример. Результат склеивания мinterмов (0101) и (0111) обозначается (01-1).

Компактная группа, которая не входит ни в какую другую компактную группу, называется **максимальной**.

Пример. На рисунке показаны три компактные группы, которые имеются на карте Карно функции $g(x, y, z)$ из примера п. 3.5.1. Компактные группы (00-) и (-10) не пересекаются, а компактные группы (0-0) и (-10), а также (00-) и (0-0), напротив, пересекаются. Все три компактные группы являются максимальными. Хотя все четыре единицы на этом рисунке находятся в соседних ячейках, компактную группу они не образуют, поскольку форма не прямоугольная. Минимальное покрытие на карте Карно содержит два элемента: (00-) и (-10), что соответствует дизъюнктивной форме $\neg x \wedge \neg y \vee \neg y \wedge \neg z$



Компактную группу называют **избыточной** (в данном покрытии), если её можно удалить из покрытия без потери эквивалентности исходной СДНФ. Покрытие (и, соответственно, ДНФ) называется **тупиковым**, если оно не содержит избыточных компактных групп.

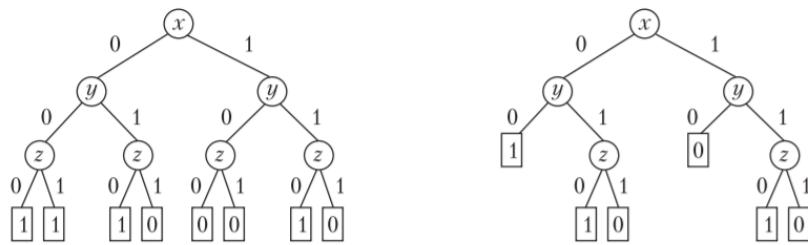
Лемма. [3] Минимальная ДНФ является тупиковой.

8) **Деревья решений.** Начнём с простого наблюдения. Таблицу истинности булевой функции n переменных можно представить в виде полного бинарного дерева высоты $n + 1$.

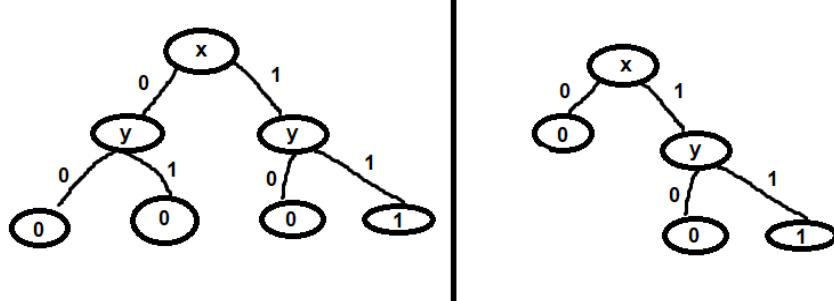
Опр: Ярусы дерева соответствуют переменным, дуги дерева соответствуют значениям переменных, скажем, левая дуга — 0, а правая — 1. Листья дерева на последнем ярусе хранят значение функции на кортеже, соответствующем пути из корня в этот лист. Такое дерево называется **деревом решений** (или **семантическим деревом**).

Дерево решений можно сократить, если заменить корень каждого поддерева, все листья которого имеют одно и то же значение, этим значением. Иногда такое сокращение значительно уменьшает объём дерева.

Пример. На рисунке слева представлено дерево решений для функции g , а справа представлено сокращённое дерево решений для функции g .



Или для функции & имеем



Алгоритм вычисления значения функции осуществляется проходом по дереву решений.

При этом тип узла дерева $N = \text{struct } \{i : 0..1; l, r : \uparrow N\}$.

Вход: указатель $T : \uparrow N$ на корень дерева решений, массив $x : \text{array [1..n] of 0..1}$ значений переменных.
Выход: $0..1$ — значение булевой функции.

$i := 0$ //номер переменной

while true **do**

if $T.l = \text{nil} \& T.r = \text{nil}$ **then return** $T.i$ **end if**

 //листовой узел — возвращаем значение

$i := i + 1$ //следующая переменная

if $x[i] \text{ then}$

$T := T.r$ //1 — переход вправо

else

$T := T.l$ //0 — переход влево

end if

end while

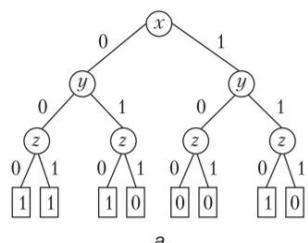
Дерево решений можно сделать ещё компактнее, если отказаться от древовидности связей, то есть допускать несколько дуг, входящих в узел. В таком случае мы получаем **бинарную диаграмму решений**. Бинарная диаграмма решений получается из бинарного дерева решений тремя последовательными преобразованиями:

- 1) отождествляются листовые узлы, содержащие 0 и содержащие 1;
- 2) в диаграмме выделяются изоморфные поддиаграммы и заменяются единственным их экземпляром;

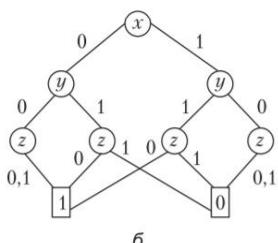
3) исключаются узлы, обе исходящие дуги которых ведут в один узел.

Интерпретация бинарной диаграммы решений (вычисление значения функций) производится в точности так же, как и для дерева решений, то есть по указанному выше алгоритму.

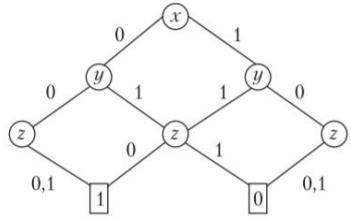
Показан результат преобразования для функции $g(x, y, z)$:



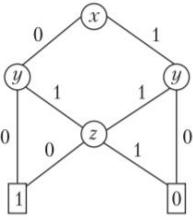
a



6



e



2

3.6 Полные системы булевых функций

На 1 балл:

План:

- 3.6.1 Замкнутые классы
- 3.6.2 Полные системы функций
- 3.6.3 Полнота двойственной системы
- 3.6.4 Теорема Поста

На 2 балла:

3.6.1 Замкнутые классы

Пусть $F = \{f_1, \dots, f_m\}, \forall i \in 1..m \ (f_i \in P_n)$. Замыканием F (обозначается $[F]$) называется множество всех булевых функций, реализуемых формулами над F:

$$[F] \stackrel{\text{Def}}{=} \{f \in P_n \mid f = \text{func } \mathcal{F} \ \& \ \mathcal{F} \in \mathfrak{F}[F]\}$$

1. $F \subset [F]$.
2. $[[F]] = [F]$.
3. $F_1 \subset F_2 \implies [F_1] \subset [F_2]$.

4. $([F_1] \cup [F_2]) \subset [F_1 \cup F_2]$.

Свойства замыкания:

Класс функций F называется **замкнутым**, если $[F] = F$.

Классы: **сохраняющих 0:** $T_0 \stackrel{\text{Def}}{=} \{f \mid f(0, \dots, 0) = 0\}$

Сохраняющих 1: $T_1 \stackrel{\text{Def}}{=} \{f \mid f(1, \dots, 1) = 1\}$

Самодвойственные: $T_* \stackrel{\text{Def}}{=} \{f \mid f = f^*\}$

Класс **монотонных** функций: $T_{\leq} \stackrel{\text{Def}}{=} \{f \mid \alpha \leq \beta \implies f(\alpha) \leq f(\beta)\}$, где $\alpha = (a_1, \dots, a_n)$, $\beta = (b_1, \dots, b_n)$, $a_i, b_i \in E_2$, $\alpha \leq \beta \stackrel{\text{Def}}{=} \forall i \ (a_i \leq b_i)$.

Класс **линейных** функций: $T_L \stackrel{\text{Def}}{=} \{f \mid f = c_0 + c_1x_1 + \dots + c_nx_n\}$, где + обозначает сложение по модулю 2, а знак & опущен.

Теорема. Классы замкнуты $T_0, T_1, T_*, T_{\leq}, T_L$

3.6.2 Полные системы функций

Класс функций F называется **полным**, если его замыкание совпадает с P_n : $[F] = P_n$. Другими словами, множество функций F образует полную систему, если любая функция реализуема в виде формулы над F.

Теорема. Пусть заданы две системы функций: $F = \{f_1, \dots, f_m\} \cup G = \{g_1, \dots, g_k\}$ Тогда, если система F полна и все функции из F реализуемы формулами над G, то система G также полна:

$$([F] = P_n \ \& \ \forall i \in 1..m \ (f_i = \text{func } \mathcal{G}_i[G])) \implies [G] = P_n.$$

3.6.3. Полнота двойственной системы

Теорема. Если система $F = \{f_1, \dots, f_k\}$ полна, то система двойственных функций $F^* = \{f_1^*, \dots, f_k^*\}$ также полна

3.6.4. Теорема Поста

Теорема Поста устанавливает необходимые и достаточные условия полноты системы булевых функций.

Теорема. Система булевых функций F полна тогда и только тогда, когда она содержит хотя бы одну функцию, не сохраняющую нуль, хотя бы одну функцию, не сохраняющую единицу, хотя бы одну несамодвойственную функцию, хотя бы одну немонотонную функцию и хотя бы одну нелинейную функцию:

$$[F] = P_n \iff \neg(F \subset T_0 \vee F \subset T_1 \vee F \subset T_* \vee F \subset T_{\leq} \vee F \subset T_L).$$

5.1 Комбинаторные задачи

На 1 балл:

1. **Комбинаторные конфигурации** (правило суммы, правило произведения, принцип Дирихле)
2. **Размещения** ($U(m,n)$ – число размещений)
3. **Размещения без повторений** ($A(m,n)$ – число размещений без повторений, убывающий факториал)
4. **Перестановки** ($P(n)$ – число перестановок, матрёшка)
5. **Сочетания** ($C(n)$ – число сочетаний)
6. **Сочетания с повторениями** ($V(n)$ – число сочетаний с повторениями, возрастающий факториал)
7. **Дискретная вероятность** (Распределение вероятности, равномерное распределение, случайная величина, математическое ожидание)

На 2 балла:

Правило суммы - если возможности построения комбинаторной конфигурации взаимно исключают друг друга, то их количества следует складывать (неформальное)

Если все множества E_i конечны и множество индексов I конечно, а семейство $\xi = \{E_i\}_{i \in I}$ дизъюнктно, то

$$\left| \bigcup_{i \in I} E_i \right| = \sum_{i \in I} |E_i|$$

Правило произведения - если возможности построения комбинаторной конфигурации независимы, то их количества следует умножать (неформальное)

Следствие теоремы из 1.4.2: $|A_1 \times \dots \times A_n| = |A_1| \times \dots \times |A_n|$

Принцип Дирихле – если $n \neq m$, то невозможно разложить n мячей по m ящикам так, чтобы в каждом ящике был ровно один мяч (неформально)

Следствие теоремы из 1.2.5: Различные отрезки натурального ряда неравномощны, или $n \neq m \rightarrow |1 \dots n| \neq |1 \dots m|$

Число размещений – число всех функций $f: 1..n \rightarrow 1..m$ (при отсутствии ограничений) или число всех возможных способов разместить n мячей по m ящикам. Обозначается $U(m,n)$

Теорема: $U(m, n) = m^n$

Число размещений без повторений – число всех инъективных функций $f: 1..n \rightarrow 1..m$ или число способов разместить n мячей по m ящикам, не более чем по одному в ящик. Обозначается $A(m,n)$

Теорема: $A(m, n) = \frac{m!}{(m-n)!}$

$A(m,n) = 0$ ($n > m$) и $A(m, 0) = 1$ по определению

Убывающий факториал – функция $(x)_n = x(x-1) \dots (x-n+1) = \prod_{i=0}^{n-1} (x-(i))$

Если $|X| = |Y| = n$, то существуют взаимо – однозначные $f: X \rightarrow Y$.

Число перестановок – число взаимо-однозначных функций $f: 1..n \rightarrow 1..n$. Обозначается $P(n)$

Теорема: $P(n) = n!$

$P(n) = A(n, n) = n * (n-1) * (n-2) \dots * 1 = n!$

Цепочка в E – последовательность $\xi = (E_1 \dots E_m)$ непустых подмножеств множества E такая, что

$$\forall i \in 1 \dots (m-1) \quad (E_i \in E_{i+1} \text{ и } E_i \neq E_{i+1})$$

Полная цепочка (матрёшка) – такая цепочка ξ , что $|\xi| = |E|$. Количество полных цепочек – это количество перестановок элементов множества E , равное $m!$

Число сочетаний – число строго монотонных возрастающих функций $f: 1..n \rightarrow 1..m$ или число размещений n неразличимых мячей по m ящикам, не более чем по одному в ящик. Обозначается $C(m,n)$

$C(m, n) = 0$ ($n > m$) по определению

Теорема: $C(m,n) = \frac{m!}{n!(m-n)!}$

Число сочетаний с повторениями – число монотонных возрастающих функций $f: 1..n \rightarrow 1..m$ или число размещений n неразличимых мячей по m ящикам. Обозначается $V(m,n)$

Теорема: $V(m,n) = C(n+m-1, n)$

Теорема $V(m,n) = V(n+1, m-1)$

Возрастающий факториал – функция $x^{(n)} = x(x + 1) \dots (x + n - 1) = \prod_{i=0}^{n-1} (x + i)$

Пусть задано конечное множество X мощности n и тотальная функция $P: X \rightarrow [0;1]$, такая, что сумма всех P от x_i равна 1. Тогда число $p_i = P(x_i)$ называется **вероятностью** x_i , а набор чисел $(p_1 \dots p_n)$ – **распределением вероятностей**. Распределение, в котором все вероятности равны $1/n$ называется **равномерным**.

Рассмотрим множество $A \subseteq X$. Если на X задано распределение вероятности, то **вероятностью подмножества A** называется число, равное $\sum_{a \in A} P(a)$.

Случайная величина — это тотальная функция $\varphi: X \rightarrow R$, причём на множестве X заданы вероятности элементов.

Математическим ожиданием случайной величины при распределении вероятности $P_X = (p_1..p_n)$ называется число

$$E_\varphi = \sum_{k=1}^n \varphi(x_k)p_k = \varphi(x_1)p_1 + \dots + \varphi(x_n)p_n.$$

Теорема *Если $A, B \subset X$, то $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.*

Условная вероятность по Колмогорову – $P(A|B) = \frac{P(A \cap B)}{P(B)}$

Теорема (формула Байеса): $P(A|B)P(B) = P(B|A)P(A)$

Теорема. [формула полной вероятности] *Если семейство множеств B_1, \dots, B_n является разбиением множества X , то есть, если $B_1 \cup \dots \cup B_n = X$ и $i \neq j \Rightarrow B_i \cap B_j = \emptyset$, то*

$$P(A) = \sum_{i=1}^n P(A|B_i)P(B_i).$$

5.2. Перестановки

На 1 балл:

План:

1. Циклы в перестановках (Таблица подстановки; подвижная и неподвижная точка; петля; транспозиция; орбита)
2. Порядок перестановки
3. Инверсии ($I(f)$ — число инверсий; задача сортировки; сортировка методом пузырька)
4. Генерация перестановок (Лексикографический и антилексикографический порядок)
5. Двойные факториалы ($n!!$)
6. Быстрая сортировка (опорный элемент)
7. Трудоемкость в среднем (алгоритм бинарного поиска)
8. Гармонические числа
9. Оценка в среднем для быстрой сортировки

На 2 балла:

Таблица подстановки - таблица, в которой нижняя строка (значения функции) является перестановкой элементов верхней строки (значения аргумента), способ задания взаимо-однозначных функций на отрезках натурального ряда (перестановок)

Перестановка f (и соответствующая ей подстановка) элементов $1, \dots, n$ обозначается (a_1, \dots, a_n) , где $a_i = f(i)$ и a_1, \dots, a_n — все различные числа из диапазона $1..n$. Если $f(i) = i$, то i называется **неподвижной точкой** перестановки, если же $f(i) \neq i$, то i называется **подвижной точкой** перестановки.

Если задана перестановка f , то **циклом** называется последовательность элементов x_1, \dots, x_k , такая, что $f(x_i) = x_{i+1}$ при $1 \leq i < k$ и $f(x_k) = x_1$. Количество элементов в цикле называется **длиной цикла**. Цикл длины 1 называется **тривиальным циклом** или **петлёй**. Цикл длины 2 называется **транспозицией**.

Перестановка длины n , которая содержит ровно один нетривиальный цикл длины k , называется **циклической** (перестановкой длины k), $k \leq n$. В циклической перестановке все элементы цикла — подвижные точки, а остальные элементы — неподвижные точки

Множество элементов, входящих в цикл, называется **орбитой**. Орбита тривиального цикла состоит из одного элемента. Орбиты различных циклов в перестановке не пересекаются (лемма). Множество орбит образует разбиение перестановки, а отношение «принадлежать одной орбите» является эквивалентностью (следствие леммы)

Теорема: Всякая неединичная перестановка представляется как суперпозиция циклических перестановок с непересекающимися орбитами, причём представление единственно

Теорема: $(f^k)^{-1} = (f^{-1})^k$

Лемма: Для любой перестановки f существует такое натуральное число k , что $f^k = e$. Мин(k) такое, что $f^k = e$ — **порядок перестановки**.

Лемма. Порядок циклической перестановки равен длине цикла.

Теорема. Порядок перестановки равен наименьшему общему кратному длин всех циклов

Инверсия — пара (a_i, a_j) в перестановке такая, что $a_i > a_j$ при $i < j$. Число инверсий обозначается $I(f)$

Теорема: Произвольную перестановку f можно представить в виде суперпозиции $I(f)$ транспозиций соседних элементов. Метод сортировки, основанный на этой теореме, называется **методом пузырька**

Вход: массив $A : array [1..n]$ of B , где для значений типа B задано отношение $<$.

Выход: массив $A : array [1..n]$ of B , отсортированный по возрастанию.

```
for i from 2 to n do
    for j from n downto i do
        if A[j] < A[j - 1] then
            A[j] ↔ A[j - 1] //транспозиция соседних элементов
        end if
    end for
end for
```

А именно, говорят, что перестановка (a_1, \dots, a_n) **лексикографически** предшествует перестановке (b_1, \dots, b_n) , если $\exists k \leq n (a_k < b_k \& \forall i < k (a_i = b_i))$. Аналогично, говорят, что перестановка (a_1, \dots, a_n) **антилексикографически** предшествует перестановке (b_1, \dots, b_n) , если $\exists k \leq n (a_k > b_k \& \forall i > k (a_i = b_i))$.

Двойной факториал ($n!!$) — произведение числа n и всех меньших натуральных чисел той же чётности

Перестановки Стирлинга – такие перестановки набора чисел {1,1,2,2...k,k}, что между двумя равными числами могут стоять только большие по значению

Теорема. Число перестановок Стирлинга равно $(2k - 1)!!$

Неформальный алгоритм **быстрой сортировки**: в массиве выбирает **опорный элемент x**; Массив делится на два подмассива относительно опорного элемента, и далее элементы меняются местами таким образом, чтобы элементы левого подмассива были меньше либо равны x, элементы правого подмассива были больше x; Первые два шага рекурсивно применяются к левому и правому подмассивам, если они состоят более чем из одного элемента

Вход: числа l и r – индексы начала и конца части массива для сортировки соответственно.

но.

Выход: отсортированный по возрастанию массив $A[a..b]$.

```
if  $l < r$  then
     $i := \text{Partition}(l, r)$  //разбиение массива
    Quicksort( $l, i - 1$ ) //рекурсивный вызов для левой части
    Quicksort( $i + 1, r$ ) //рекурсивный вызов для правой части
end if
```

Функция Partition находит опорный элемент.

Вход: числа l и r – индексы начала и конца части массива для поиска соответственно.

Выход: индекс i опорного элемента.

```
 $x := A[r]$  //опорным является последний элемент массива
 $i := l - 1$  //инициализация за пределами массива
for  $j$  from  $l$  to  $r - 1$  do
    if  $A[j] \leq x$  then
         $i := i + 1$ 
         $A[i] \leftrightarrow A[j]$ 
    end if
end for
 $A[i] \leftrightarrow A[r]$  //опорный элемент помещается в нужное место
return  $i$ 
```

Трудоёмкость алгоритма в среднем — математическое ожидание количества вычислительного ресурса (памяти или времени), используемого алгоритмом при заданном распределении вероятности появления тех или иных входных данных P_x

Гармонические числа $H(n)$ определяются следующим образом: $H(0) = 0$; $H(n) = H(n-1) + \frac{1}{n}$

Нетрудно вывести, что $H(n) = \sum_{k=1}^n \frac{1}{k}$

По аналогии можно ввести понятие **обобщённых гармонических чисел порядка m**:

$$H(m, 0) \stackrel{\text{Def}}{=} 0, \quad H(m, n) \stackrel{\text{Def}}{=} H(m, n-1) + 1/n^m, \quad H(m, n) = \sum_{k=1}^n \frac{1}{k^m}.$$

Теорема: Время работы алгоритма «быстрой сортировки» в среднем $O(n \log n)$.

5.3 Биноминальные коэффициенты (алгоритмы вычисления числа сочетаний)

(слайд 881-915)

На 1 балл:

- План: 1. Свойства биноминальных коэффициентов 2. Бином Ньютона 3. Треугольник Паскаля 4. Применение рекуррентного соотношения 5. Генерация подмножеств 6. Расширенные биноминальные коэффициенты 7. Мульти множества и последовательности 8. Мультибиномиальные коэффициенты 9. Биномиальная система счисления

На 2 балла:

5.3.1. Основная формула для числа сочетаний $C(m, n) = C_m^n = \frac{m!}{n!(m-n)!}$

T1	1	$C(m, n) = C(m, m - n) = C_m^n = C_m^{m-n}$
	2	$C(m, n) = C(m - 1, n) + C(m - 1, n - 1) = C_m^n = C_{m-1}^n + C_{m-1}^{n-1}$
	3	$C(n, i) * C(i, m) = C(n, m) * C(n - m, i - m) = C_n^i * C_i^m = C_m^m * C_{n-m}^{i-m}$
	4	$C(m, n) = \frac{m}{n} C(m - 1, n - 1) = C_m^n = \frac{m}{n} C_{m-1}^{n-1}$
T2		$\sum_{n=0}^m n * C(m, n) = m * 2^{m-1} = \sum_{n=0}^m n * C_m^n$
T3		$C(m + n, k) = \sum_{i=0}^k C(m, i) * C(n, k - i) = \sum_{i=0}^k C_m^i C_n^{k-i}$ Способ выбрать k шаров из $m+n$ шаров. Известно как тождество Коши
T4		$\sum_{n=0}^m (C(m, n))^2 = C(2m, n) = \sum_{n=0}^m C_m^n = C_{2m}^n$
T5		$C(m, n) = \sum_{k=0}^n C(m - (k + 1), n - k) = \sum_{k=0}^n C_{m-(k+1)}^{n-k}$

5.3.2. Бином Ньютона

Число сочетаний $C(m, n) = C_m^n$ также называется **биноминальными коэффициентами**. Смысл названия устанавливается теоремой, известной как **бином Ньютона**.

Теорема: $(x + y)^m = \sum_{n=0}^m C(m, n) * x^n * y^{m-n} = \sum_{n=0}^m C_m^n * x^n * y^{m-n}$

Следствие: $\sum_{n=0}^m C(m, n) = 2^m \rightarrow$ Доказательство: достаточно взять $x = y = 1$

Следствие: $\sum_{n=0}^m (-1)^n C(m, n) = 0 \rightarrow$ Доказательство: достаточно взять $x = -1 \ y = 1$

5.3.3. Треугольник Паскаля

Из 5.3.1. T1.2 вытекает эффективный способ рекуррентного вычисления биноминальных коэффициентов, который можно представить в графической форме как **треугольник Паскаля**, равнобедренном треугольнике, числа нижнего ряда которого являются суммой двух верхних.

		1
	1	1
1	1	2
1	3	3
1	4	6
1	1	1
1	2	3
1	3	6
1	4	10

Число сочетаний $C(m, n)$ находится в $(m + 1)$ ряду на $(n + 1)$ месте.
Прямая рекурсивная реализация неэффективна из-за переполнения стека вызовов и пересчета некоторых биноминальных коэффициентов.
Для случая вычисления $C(m, n)$ имеем 2^n сложений. Решается это замечательным свойством треугольника, повернуть его на 45 градусов.
Получится матрица, где первая строка и столбец – единицы, остальные элементы – сумма левого и верхнего элемента.

5.3.4. Применение рекуррентного соотношения

Рекуррентное соотношение – один из основных приемов организации эффективных вычислений.

Замечание: однопараметрическое $a_i = f(a_{i-1})$ дает алгоритм без использования доп. памяти. Пусть $a_1 = m - n + 1; a_i = \frac{m-n+i}{i} * a_{i-1}$. Тогда $a_i = C(m - n + i, i) = C_{m-n+i}^i$

5.3.5. Генерация подмножеств

Элементы $\{1, \dots, m\}$ естественно упорядочены → каждое n -элементарное подмножество его рассматривается как упорядоченное. На множестве таких пос-тей определяется [цикографический порядок](#). Есть алгоритм, генерирующий все n -элементарные подмножества.

Пример: для $n=3$ (мощность подм-ва) и $m=4$ (мощность мн-ва): $(1,2,3);(1,2,4);(1,3,4);(2,3,4)$.

Используется для задач на поиск в множестве элемента с некоторыми св-вами. Порой исходное мн-во велико и элементы устроены сложно. Если заранее неизвестно, как искать, просто перебираем до нужного эл-та (**это переборные задачи**). При решении таковых нет смысла строить всё мн-во(**это пространство поиска**), достаточно иметь итератор, перебирающий в нужном порядке

5.3.6. Расширенные биноминальные коэффициенты

В ф-ле $C(m, n) = \frac{m!}{n!(m-n)!} = \frac{m(m-1)\dots(m-n+1)}{n!}$ сократили $(m-n)!$, из-за чего вычислимость расширилась с $m, n \in N$ до **расширенных биноминальных кф-ов** $m \in Z, n \in N_0$. (Замечание: также можно считать $m \in Q$ рациональные и вещественные $m \in R$

Теорема: $C(m, n) = (-1)^n C(n - m - 1, n)$

-4	-3	-2	-1	$\frac{m}{n}$	0	1	2	3	4
1	1	1	1	0	1	1	1	1	1
-4	-3	-2	-1	1	0	1	2	3	4
10	6	3	1	2	0	0	1	3	6
-20	-10	-4	-1	3	0	0	0	1	4
35	15	5	1	4	0	0	0	0	1

$$\sum_{k=0}^n (-1)^k C(m + n - 1, n) x^k$$

В 5.3.1. Т1.2 в $C(m, n) = C(m - 1, n) + C(m - 1, n - 1)$ не фигурирует $m > 0$, значит исходя из 2х тождеств выше мы можем составить **расширенный треугольник Паскаля**

Пример: разложение дробей в ряд $\frac{1}{(1+x)^m} = (1+x)^{-m} =$

5.3.7. Мульти множества и последовательности

Пусть задано мн-во $X = \{x_1, \dots, x_m\}$ и пос-ть $Y = \{y_1, \dots, y_n\} \forall i (y_i \in X)$, причем в пос-ти Y x_i встречается n_i раз. Тогда мульти множества $\hat{X} = \{x_1^{n_1}, \dots, x_m^{n_m}\}$ – **состав пос-ти Y**. Состав пос-ти определяется однозначно, у разных пос-тей может быть одинаковый. Число Эйлера 2 рода - это кол-во перестановок мульти множества $\{1,1,2,2, \dots, n, n\}$

Теорема:

$$C(n; n_1, \dots, n_m) = \frac{n!}{n_1! \dots n_m!}.$$

5.3.8.

Мультиномиальные коэффициенты

Числа

$C(n; n_1, \dots, n_m)$ иногда называют **мультиномиальными кф-ми**, потому что в частности справедлива формула, обобщающая бином Ньютона с их участием. Теорема: $(x_1 + \dots + x_m)^n =$

$$\sum_{n_1+\dots+n_m=n} C(n; n_1, \dots, n_m) * x_1^{n_1} \dots * x_m^{n_m}$$

$$\text{Следствие: } \sum_{n_1+\dots+n_m=n} \frac{n!}{n_1! \dots n_m!} =$$

$$m^n \quad \text{Пример: } C(32; 10, 10, 10, 2) = \frac{32!}{10!^3 2!}$$

5.3.9. Биномиальная система счисления

к-

биномиальной системой счисления называется непозиционная система счисления, где каждое натуральное n , где каждое натуральное n представлено как $n = \sum_{i=1}^k C(b_i, i) \quad (0 \leq b_1 < \dots < b_k)$

Пример: $k=2$, тогда $1=C(0,1)+C(2,2)=0+1$, $2=C(1,1)+C(2,2)=1+1$, $3=C(0,1)+C(3,2)=0+3$,

$4=C(1,1)+C(3,2)=1+3$, $5=C(2,1)+C(3,2)=2+3$, $6=C(0,1)+C(4,2)=0+6$

Теорема: при $k \in N \ \forall n \in N \ \exists! b_1, \dots, b_k \in N$ т.чт. $0 \leq b_1 < \dots < b_k$ и $n = \sum_{i=1}^k C(b_i, i)$

5.4 Разбиения

На 1 балл:

1. Числа Стирлинга	2. Числа Моргана
3. Вычисление чисел Стирлинга	4. Числа Белла
5. Треугольник Белла	

На 2 балла:

1.] $\beta = \{B_1, \dots, B_n\}$ – разбиение множества X из m элементов на n подмножеств:

$$B_i \subset X, \bigcup_{i=1}^n B_i = X, B_i \neq \emptyset, B_i \cap B_j \neq \emptyset \text{ при } i \neq j.$$

B_i – блоки разбиения.] E_1 и E_2 – два разбиения X ; каждый блок E_2 – объединение блоков $E_1 \Rightarrow E_1$ – измельчение E_2 . Число разбиений m -элементного множества на n блоков, называется **числом**

Стирлинга (число Стирлинга второго рода) и обозначается $S(m, n)$. $S(m, m) \stackrel{\text{def}}{=} 1$ при $m > 0$,

$S(m, 0) \stackrel{\text{def}}{=} 0$ при $m > 0$, $S(m, n) \stackrel{\text{def}}{=} 0$ при $n > m$

(Т) $S(m, n) = S(m - 1, n - 1) + nS(m - 1, n)$.

(Т) $S(m, n) = \sum_{i=n-1}^{m-1} C(m - 1, i)S(i, n - 1)$.

2. Число суръективных функций, т.е. число размещений m мячей по n ящикам (все заняты) – **число Моргана (число Стирлинга первого рода)**. Обозначение: $M(m, n)$ или $s(m, n)$.

(Т) $M(m, n) = n! S(m, n)$.

(Т) $M(m, n) = M(m - 1, n - 1) + (m - 1)M(m - 1, n)$.

(Т) $\sum_k^m M(m, k) = m!$.

3. В (1) есть 2 формулы (теоремы). Вторая требует вычисления бином. коэф-ов и много сумм/умножений. Первая требует хранить вычисленное.

$m \setminus n$	0	1	2	3	4	5	6	7
0	1							
1	0	1						
2	0	1	1					
3	0	1	3	1				
4	0	1	7	6	1			
5	0	15	25	10	1			
6	0	31	90	65	15	1		
7	0	63	301	350	140	21	1	

Значения, которые необходимы для вычисления числа $S(7, 3)$, обведены прямоугольниками.

Алгоритм. Вход: $m, n > 0$.

Выход: $S(m, n)$.

```

if m = n then return 1 end if //можем сразу вернуть известные значения
if n = 0 then return 0 end if
if n > m then return 0 end if
if n = 1 then return 1 end if
d := min(n, m - n + 1) //длина рабочего массива
D : array[1..d] of integer //рабочий массив
s := max(n, m - n + 1) //количество повторений
for i from 1 to d do D[i] := 1 end for //инициализация
for i from 2 to s do
    for j from 2 to d do
        D[j] := if d = n then D[j - 1] + j * D[j] else D[j] + i * D[j - 1] end if
    end for
end for
return D[d] //число Стирлинга

```

4. Число всех разбиений m -элементного множества называется **числом Белла**:

$$B(m) \stackrel{\text{def}}{=} \sum_{n=0}^m S(m, n), B(0) \stackrel{\text{def}}{=} 1.$$

(Т) $B(m + 1) = \sum_{i=0}^m C(m, i)B(i)$.

5. Для выч. числа Белла по рекуррентной ф-ле нужно $O(n^3)$. Есть $O(n^2)$ с хранением коэф-ов.

Треугольник Белла — это бесконечная треугольная матрица, где каждое число (кроме чисел на левой боковой стороне) является суммой двух чисел: стоящего слева от него в этой же строке и слева

над ним. Первое число в каждой строке (кроме первой) является последним числом из предыдущей строки. Число в первой строке — единица. Удобно хранить в нижнетреугольной матрице.

$$(Л) A_{n,k} = \sum_{i=n-k+1}^n C(k-1, i-n_k-1) A_{i,1}.$$

$$(Т) B(m) = A_{m+1,1} = A_{m,m}.$$

Алгоритм.

Вход: число m

Выход: число Белла $B(m)$

```
if  $m = 0$  then return 1 end if //по определению
A : array [1..m] of integer //рабочий массив,  $A[k]$  хранит  $A_{n,k}$ 
A[1] := 1 // $A_{1,1} = B(1) = 1$ 
for n from 2 to m do
    t := A[1] //переменная t хранит  $A_{n-1,k-1}$ 
    A[1] := A[n - 1] // $A_{n,1} = A_{n-1,n-1}$ 
    for k from 2 to n do
        s := A[k] //переменная s хранит  $A_{n-1,k}$ 
        A[k] := A[k - 1] + t // $A_{n,k} = A_{n,k-1} + A_{n-1,k-1}$ 
        t := s // $k := k + 1$ 
    end for
end for
return A[m] //число Белла
```

5.5. Включения и исключения (беспорядки и субфакториал)

На 1 балл:

1. Зачем нужен метод включений и исключений.
2. Формула включений и исключений.
3. Что такое беспорядки и субфакториал.
4. Формулы, связанные с субфакториалом.

На 2 балла:

- 1) Практические задачи не всегда прямо сводятся к известным комбинаторным конфигурациям. В этом случае используются различные методы сведения одних комбинаторных конфигураций к другим. Мы начинаем с самого простого и прямолинейного, но имеющего ограниченную область применения метода включений и исключений.
- 2) Следующая формула, известная как формула включений и исключений, позволяет вычислить мощность объединения нескольких множеств, если известны их мощности и мощности всех возможных пересечений.

Теорема.

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - \dots + (-1)^{n-1} |A_1 \cap \dots \cap A_n|.$$

3) Перестановка без неподвижных точек называется беспорядком. Количество всех беспорядков n предметов называется субфакториалом числа n и обозначается $!n$.

4)

Теорема. $!n = n! \sum_{k=0}^n (-1)^k / k!$.

Замечание. Сумма в правой части формулы $!n = n! \sum_{k=0}^n \frac{(-1)^k}{k!}$ является частичной суммой разложения $e^{-1} = 1 - 1/1! + 1/2! - \dots + (-1)^n/n! + \dots$. Поэтому субфакториал является ближайшим целым числом к числу $n!/e$.

Теорема. $!n = (n-1)(!(n-1)+!(n-2))$.

Следствие. $!n = !(n-1) \cdot n + (-1)^n$.

5.6. Формулы обращения (явные формулы для чисел Стирлинга)

На 1 балл:

1. Теорема обращения.
2. Применение теоремы обращения.
3. Лемма, которая используется для доказательства формул обращения.
4. Формулы обращения.
5. Вывод формулы для чисел Стирлинга.

На 2 балла:

- 1) Пусть $a_{n,k}$ и $b_{n,k}$ — некоторые (комбинаторные) числа, зависящие от параметров n и k , причём $0 \leq k \leq n$. Если известно выражение чисел $a_{n,k}$ через числа $b_{n,k}$, то в некоторых случаях можно найти и выражение чисел $b_{n,k}$ через числа $a_{n,k}$, то есть решить комбинаторное уравнение.

Теорема. Пусть $\forall n \left(\forall k \leq n \left(a_{n,k} = \sum_{i=0}^n \lambda_{n,k,i} b_{n,i} \right) \right)$ и пусть $\exists \mu_{n,k,i} \left(\forall k \leq n \left(\forall m \leq n \left(\sum_{i=0}^n \mu_{n,k,i} \lambda_{n,i,m} = \begin{cases} 1, & m = k, \\ 0, & m \neq k \end{cases} \right) \right) \right)$. Тогда $\forall k \leq n \left(b_{n,k} = \sum_{i=0}^n \mu_{n,k,i} a_{n,i} \right)$.

- 2) Применение теоремы обращения предполагает отыскание для заданных чисел $\lambda_{n,k,i}$ (коэффициентов комбинаторного уравнения) соответствующих чисел $\mu_{n,k,i}$, удовлетворяющих условию теоремы обращения. Особенно часто числами $\lambda_{n,k,i}$ являются биномиальные коэффициенты.

- 3) **Лемма.** $\sum_{i=0}^n (-1)^{i-m} C(n, i) C(i, m) = \begin{cases} 1, & m = n, \\ 0, & m < n. \end{cases}$

- 4) **Теорема.** Если $a_{n,k} = \sum_{i=0}^k C(k, i) b_{n,i}$, то $b_{n,k} = \sum_{i=0}^k (-1)^{k-i} C(k, i) a_{n,i}$.

Теорема. Если $a_{n,k} = \sum_{i=k}^n C(i, k) b_{n,i}$, то $b_{n,k} = \sum_{i=k}^n (-1)^{i-k} C(i, k) a_{n,i}$.

6.3. Формулы для чисел Стирлинга и Моргана

- 5) качестве примера использования формул обращения рассмотрим получение явных формул для чисел Стирлинга и Моргана. Рассмотрим множество функций $f: A \rightarrow B$, где $|A| = n$ и $|B| = k$. Число всех таких функций равно k^n . С другой стороны, число функций f , таких, что $|f(A)| = i$, равно $M(n, i)$, поскольку $M(n, i)$ — это число сюръективных функций $f: 1..n \rightarrow 1..i$. Но множество значений функции (при заданном i) можно выбрать $C(k, i)$ способами. Поэтому $k^n = \sum_{i=0}^k C(k, i) M(n, i)$.

Обозначив $a_{n,k} := k^n$ и $b_{n,i} := M(n, i)$, имеем по первой теореме [п. 5.6.2](#)

$$M(n, k) = \sum_{i=0}^k (-1)^{k-i} C(k, i) i^n.$$

Учитывая связь чисел Стирлинга и Моргана, окончательно имеем

$$S(n, k) = \frac{1}{n!} \sum_{i=0}^k (-1)^{k-i} C(k, i) i^n.$$

5.7 Производящие функции (числа Каталана)

На 1 балл:

1. Формальные степенные ряды
2. Метод неопределённых коэффициентов
3. Числа Фибоначчи и Каталана

На 2 балла:

Пусть есть последовательность комбинаторных чисел a_i и последовательность функций $\phi_i(x)$.
Рассмотрим формальный ряд

$$F(x) = \sum_i a_i * \phi_i(x)$$

Выражение $F(x)$ называется **производящей функцией** (для заданной последовательности комбинаторных чисел a_i относительно заданной последовательности функций $\phi_i(x)$).

пр: **Бином Ньютона** $(1 + x)^n = \sum_{i=0}^n C(n, i) * x^i$

Лемма: $V(m, n) = V(m-1, n) + V(m, n-1)$

Идея док-ва леммы: выразить $V(m, n)$ через $C(n+m-1, n)$

Степенные ряды рассматриваются здесь именно как формальные суммы, как конечные, так и бесконечные, причём вопросы сходимости рядов и аналитичности функций не ставятся.

Пр: $F(x) = 1 + x + x^2 + \dots = \sum_{i=0}^n x^i$

$F(x) = 1 + x(1 + x + x^2 + \dots) = 1 + x * F(x) \rightarrow F(x) = \frac{1}{1-x}$. Это не числовое равенство!!!

Числа Фибоначчи: $F(0) = 1, F(1) = 1, F(n+2) = F(n+1) + F(n)$

Производящая функция для Фибоначчи: $\phi(x) = \sum_{i=0}^n F(i)x^n = \dots = 1 + x + x^2\phi(x) + x(\phi(x) - 1)$

$\phi(x) = \frac{1}{1-x-x^2}$ Корни знаменателя: $\frac{1 \pm \sqrt{5}}{2}$;

Окончательное соотношение (**формула Бине**): $F(n) = \frac{1}{\sqrt{5}} * ((\frac{1+\sqrt{5}}{2})^{n+1} - (\frac{1-\sqrt{5}}{2})^{n+1})$

Числа Каталана: $C(0) = 1, C(n) = \sum_{k=0}^{n-1} C(k)C(n-k-1)$

$\phi(x) = \frac{1 \pm \sqrt{1-4x}}{2x}$

Теорема: $C(n) = C(2n, n)/(n+1)$

МНК: из матана известно, что если $F(x) = \sum_i a_i * \phi_i(x)$ и $F(x) = \sum_i b_i * \phi_i(x)$, то тогда
 $a_i = b_i$

С помощью МНК находятся выражения для чисел Фибоначии и Каталана после получения их производящих функций.

7.1 Определения графов

На 1 балл:

1. История (кенигсбергские мосты, 3 дома и 3 колодца)
2. Основное определение
3. Инцидентность, смежность, дополнение
4. Диаграмма
5. Орграфы, псевдографы, мультиграфы, гиперграфы
6. Изоформизм графов

На 2 балла:

Было много прикладных задач: **Задача о Кёнигсбергских мостах**, **Задача о трёх домах и трёх колодцах**, **Задача о четырёх красках**

Графом $G(V, E)$ называется совокупность двух множеств — непустого множества V (множества **вершин**) и множества E двухэлементных подмножеств множества V (E — множество **ребер**), $V \neq \emptyset$, $E \subset 2V$ & $\forall e \in E (|e|=2)$

Два графа **равны**, если равны множества вершин и рёбер.

Пусть v_1, v_2 — вершины, $e = (v_1, v_2)$ — соединяющее их ребро. Тогда вершина v_1 и ребро e **инцидентны**, ребро e и вершина v_2 также инцидентны. Два ребра, инцидентные одной вершине, называются **смежными**; две вершины, инцидентные одному ребру, также называются **смежными**.

Множество вершин, смежных с вершиной v , называется **множеством смежности** (или окрестностью) вершины v и обозначается $\Gamma(v)$. Множество рёбер, инцидентных вершине v , называется множеством инцидентности вершины v и обозначается $\Lambda(v)$

Рассмотрим граф $G(V, E)$ с некоторым множеством вершин V . **Дополнением** $G'(V, E)$ называется другой граф, с тем же множеством вершин V , в котором вершины смежны тогда и только тогда, когда они не смежны в исходном графе

Теорема: сумма ребер в графе+его дополнении = $p*(p-1)/2$

Идея док-ва: $C(p, 2)$

Обычно граф изображают **диаграммой**: вершины — точками (или кружками), рёбра — линиями

Часто рассматриваются следующие родственные графам объекты:

1. Если элементами множества E являются упорядоченные пары (то есть $E \subset V \times V$), то граф называется **ориентированным** (или **орграфом**). В этом случае элементы множества V называются **узлами**, а элементы множества E — **дугами**.

2. Если элементом множества E может быть пара одинаковых (не различных) элементов V , то такой элемент множества E называется петлёй, а граф называется **графом с петлями** (или **псевдографом**).

3. Если E является не множеством, а мульти множеством, содержащим некоторые элементы по несколько раз, то эти элементы называются **кратными рёбрами**, а граф называется **мультиграфом**.

Говорят, что два графа, $G1(V1, E1)$ и $G2(V2, E2)$, (вершинно) **изоморфны** (обозначается $G1 \sim G2$), если существует изоморфизм $h: V1 \rightarrow V2$, уважающий отношение смежности вершин.

7.2. Элементы графов

На 1 балл:

- 1- подграфы
- 2 – валентность
- 3 – маршрут, цепи, циклы
- 4 – связные/несвязные графы
- 5 – расстояние, ярус, эксцентрикитет, диаметр, радиус, центр
- 6 – степенные последовательности.

Надо знать: Валентность - $d(v)$ — степень; $\delta(G)$ — минимальная степень; $\Delta(G)$ — максимальная степень; $r(G)$ — степень регулярности; $d^+(v)$ — полустепень захода; $d^-(v)$ — полустепень исхода; изолированные и висячие вершины.

На 2 балла:

1) Опр: Граф $G'(V', E')$ называется **подграфом** графа $G(V, E)$ (обозначается $G' \subset G$), если $V' \subset V$ & $E' \subset E$. Если $V' = V$, то G' называется **остовным** подграфом G . Если $V' \subset V$ & $E' \subset E$ & $(V' \neq V \vee E' \neq E)$, то граф G' называется **собственным** подграфом графа G . Подграф $G'(V', E')$ называется **правильным** подграфом графа $G(V, E)$, если G' содержит все возможные рёбра G : $\forall u, v \in V' , ((u, v) \in E \Rightarrow (u, v) \in E')$. Правильный подграф $G'(V', E')$ графа $G(V, E)$ определяется подмножеством вершин V' .

Неправильные подграфы называют **изграфами**.

2) Количество рёбер, инцидентных вершине v , называется **степенью (или валентностью)** вершины v и обозначается $d(v)$: $\forall v \in V (0 \leq d(v) \leq p - 1)$, $d(v) = |\Gamma^+(v)|$.

Если степени всех вершин равны k , то граф называется **регулярным степени k**

Если $d(v) = 0$ – **изолированная** вершина, $d(v) = 1$ – **висячая**.

Для орграфа число дуг, исходящих из узла v , называется **полустепенью исхода**, а число входящих — **полустепенью захода**.

Теорема. [Лемма о рукопожатиях] Сумма степеней вершин графа (мультиграфа) равна удвоенному количеству рёбер.

3) Маршрут - чередующаяся последовательность вершин и рёбер, в которой любые два соседних элемента инцидентны. Если все ребра и вершины различны, то маршрут – **простая цепь (цепь – если ребра различны)**. Замкнутая цепь – **цикл**. Число ребер в цикле – **его длина**.

Лемма. Если есть какая-либо цепь, соединяющая вершины u и v , то есть и простая цепь.

4) Две вершины связаны, если существует цепь их соединяющая. **Граф связан**, если все вершины связаны. Если 2 вершины связаны, то по Лемме (из 3) существует простая цепь, соединяющая их.

Опр: Классы эквивалентности по отношению связности называются **компонентами связности графа**. $k(G) = 1$ – **граф связан**, $k(G) > 1$ – **несвязный**. Граф из изолированный вершин – **вполне несвязный**.

Теорема: Любой граф или его дополнение являются связным графом. Они могут быть связными одновременно, но не могут быть одновременно несвязными.

5) Расстояние между u и v – длина кратчайшей цепи (= геодезическая цепь). ($d(u,v)$ – расстояние).

Геодезический граф – если для любых 2х вершин $\exists!$ Геодезическая цепь.

Множество вершин, находящихся на заданном расстоянии n от вершины v (обозначение $D(v, n)$), называется **ярусом**.

Эксцентрикитетом $e(v)$ вершины v в связном графе $G(V, E)$ называется максимальное расстояние от вершины v до других вершин графа G .

Диаметром графа называется длиннейшая геодезическая цепь. обозначается $D(G)$.

Радиусом графа называется простая цепь, длина которой равна наименьшему из эксцентрикитетов вершин. обозначается $R(G)$.

Центральная вершина: $e(v) = R(G)$. Множество центральных вершин $C(G)$ называется **центром графа**.

Теорема. Если граф G связан, то $D(G)/2 \leq R(G) \leq D(G)$.

Последовательность степеней вершин некоторого графа называется **степенной последовательностью графа**.

Лемма. [1] Сумма степенной последовательности — чётное число

Лемма. [2] Если последовательность (d_1, \dots, d_p) степенная, то последовательность $(0, \dots, 0, d_1, \dots, d_p)$ также степенная, и наоборот.

Лемма. [3] В степенной последовательности (d_1, \dots, d_p) каждый элемент меньше длины последовательности: $\forall i \in 1..p (d_i < p)$.

7.3. Виды графов и операции над графиками

На 1 балл:

- 1 – полные и пустые графы
- 2 – двудольные графы
- 3 – звездные графы
- 4 – направленные орграфы и сети,
- 5 – операции над графиками
- 6 – свойства простых циклов.

На 2 балла:

1) Граф из одной вершины – **тривиальный**.

Граф, в котором любые две вершины смежны, называется **полным**. Граф, в котором нет смежных вершин, называется **пустым**.

Максимальный полный подграф (некоторого графа) называется **кликой** (этого графа). Максимальное число вершин среди клик данного графа называется **кликовым числом** графа.

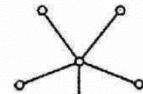
2) Граф $G(V, E)$ называется **двудольным**, если множество V может быть разбито на два непересекающихся множества V_1 и V_2 , причём всякое ребро из E инцидентно вершине из V_1 и вершине из V_2 . Множества V_1 и V_2 называются **долями** двудольного графа. Если двудольный граф содержит все рёбра, соединяющие множества V_1 и V_2 , то он называется **полным двудольным графом**.

Теорема. Граф является двудольным тогда и только тогда, когда в нём нет простых циклов нечётной длины.

3) Полный двудольный граф вида $K_{1,n}$, где $n > 1$ называется **звёздным** графом и обозначается S_n .

Единственная вершина, входящая в первую долю V_1 называется **центральной** вершиной звезды.

Теорема. Центр звёздного графа и его центральная вершина совпадают.



4) Если в графе ориентировать все рёбра, то получится орграф, который называется **направленным**, или **антисимметричным**. Направленный орграф, полученный из полного графа, называется **турниром**. Если $d^+(v) = 0$, то узел называется **источником**, если $d^-(v) = 0$ – **стоком**.

Направленный слабосвязный орграф с одним источником и одним стоком называется **сетью**.

5) **Объединение:** — $G_1(V_1, E_1) \cup G_2(V_2, E_2) = G(V, E), V_1 \cap V_2 = \emptyset$.

Пример. $K_{3,3} = C_3 \cup C_3$

Соединение: — $G_1(V_1, E_1) + G_2(V_2, E_2) = G(V, E), V_1 \cap V_2 = \emptyset$.

Пример. $K_{3,3} = \bar{C}_3 \cup \bar{C}_3$

Удаление вершины: $G_1(V_1, E_1) - v = G_2(V_2, E_2), v \in V_1$

Пример. $C_3 - v = K_2$.

Удаление ребра: $G_1(V_1, E_1) - e = G_2(V_2, E_2), e \in E_1$

Пример. $K_2 - e = \bar{K}_2$

Добавление вершины: $G_1(V_1, E_1) + v = G_2(V_2, E_2), v \notin V_1$

Пример. $K_2 + v = K_2 \cup K_1$.

Добавление ребра: $G_1(V_1, E_1) + e = G_2(V_2, E_2), e \notin E_1$

Пример. $P_3 + e = C_3$

Стягивание правильного подграфа: $G_1(V_1, E_1)/A = G_2(V_2, E_2), V_2 = (V_1 \setminus A) + v, A \subset V_1$.

Пример. $K_4/K_3 = K_2$.

Размножение вершины: $G_1(V_1, E_1) \uparrow v = G_2(V_2, E_2), V_2 = V_1 + v' \& E_2 = E_1 \cup \{(v, v')\} \cup \{e = (u, v') | u \in \Gamma^+(v)\}, v \in V_1, v' \notin V_1$.

Пример. $K_2 \uparrow v = C_3$.

Свойства операций:

- 1) $G_1 \cup G_2 = G_2 \cup G_1$
- 2) $G_1 = G_2 \iff \bar{G}_1 = \bar{G}_2$
- 3) $G_1 + G_2 = G_2 + G_1$
- 4) $\overline{G_1 \cup G_2} = \overline{G_1} + \overline{G_2}$

6) **Теорема.** [1] Регулярный граф степени 2 состоит из простых циклов.

Теорема. [2] В ациклическом графе рёбер меньше, чем вершин: $z(G) = 0 \implies q(G) < p(G)$.

7.4 Представление графов в программах (алгоритм поиска в ширину и в глубину)

На 1 балл:

Требования к представлению графов $O(n)$ — асимптотическая оценка

Матрица смежности M : array [1..p,1..p] of 0..1

Матрица инциденций H : array [1..p,1..q] of -1..1

Списки смежности Γ : array [1..p] of ↑ N, $N=struct\{v: 1..p;n:\uparrow N\}$

Массив дуг E :array [1..q] of struct {b,e: 1..p}

Обходы графов **Поиск в ширину и поиск в глубину**

На 2 балла:

$n(p,q)$ — **объём памяти** для каждого представления. Здесь p — число вершин, а q — число рёбер.

$M[i,j] := \exists e_j \in E (v_i \in e_j)$, то есть вершины v_i и v_j смежны. Для матрицы смежности $n(p,q) = O(p^2)$

$H[i,j] := \exists e_j \in E (v_i \in e_j)$, то есть вершина v_i инцидентна ребру e_j . Для орграфа тоже самое только если начало, то -1, а если конец 1. Для матрицы инциденций $n(p,q) = O(pq)$. Список смежности состоит из массива указателей Γ на списки смежных вершин, где элемент списка представлен структурой N . В случае представления неориентированных графов списками смежности $n(p,q) = O(p+2q)$, а в случае ориентированных графов $n(p,q) = O(p+q)$. Массив структур E :array[1..q]of struct {b,e: 1..p}, отражающий список пар смежных вершин (или, для орграфов, узлов), называется массивом рёбер(массивом дуг). Для массива рёбер (или дуг) $n(p,q) = O(2q)$.

Теорема. Если граф G связен и конечен, то поиск в ширину и поиск в глубину обходят все вершины по одному разу за время, пропорциональное не более чем суммарному числу вершин и рёбер.

Если структура данных T — это стек (LIFO — Last In First Out), то обход называется поиском в глубину. Если T — это очередь (FIFO — First In First Out), то обход называется поиском в ширину.

Вход:граф $G(V,E)$, представленный списками смежности Γ .
Выход:последовательность вершин обхода.
for $v \in V$ do $x[v] := 0$ end for //вначале все вершины не отмечены
select $v \in V$ //начало обхода — произвольная вершина
 $v \rightarrow T$ //помещаем v в структуру данных T ...
 $x[v] := 1$ //...и отмечаем вершину v
repeat
 $u \leftarrow T$ //извлекаем вершину из структуры данных T ...
 yield u //...и возвращаем её в качестве очередной пройденной
 for $w \in \Gamma(u)$ do
 if $x[w] = 0$ then
 $w \rightarrow T$ //помещаем w в структуру данных T ...
 $x[w] := 1$ //...и отмечаем вершину w
 end if
 end for
until $T = \emptyset$

7.5 Графы и отношения(достижимость и транзитивное замыкание)

Орграфы и бинарные отношения

Граф инциденций

*Гиперграфы и многоместные отношения Деревья И/ИЛИ

*Достижимость и частичное упорядочение Достижимый узел

*Достижимость и транзитивное замыкание

Любой орграф $G(V,E)$, возможно, с петлями, но без кратных дуг, задаёт бинарное отношение E на множестве V . Обратно, любое бинарное отношение E на множестве V определяет орграф $G(V,E)$: пара элементов принадлежит отношению $(a,b) \in E \subset V \times V$ тогда и только тогда, когда в графе G есть дуга (a,b) .

Орграф, соответствующий рефлексивному отношению, в каждой вершине имеет петлю, а значит является псевдографом. Если же отношение антирефлексивно, то орграф вовсегда имеет петель.

Обратный орграф — это орграф, соответствующий обратному отношению, то есть $(G(V,E))^{-1} = G(V, E^{\wedge -1})$. Степени бинарного отношения $E \subset V \times V$ соответствуют путям в орграфе. Степень $E^{\wedge 1}$ соответствует путям длины 1, то есть дугам, $E^{\wedge 2}$ соответствует путям длины 2 и т. д. Если отношение антисимметрично, то соответствующий ему орграф не имеет встречных дуг.

Антисимметричными являются, в частности, отношения порядка, которые рассматриваются в следующем параграфе с точки зрения теории графов. Если же отношение симметрично, то всякая дуга орграфа имеет встречную, и направление дуги не имеет значения. Таким образом, всякому симметричному отношению соответствует неориентированный граф, и обратно, граф (неориентированный) соответствует симметричному отношению. Полный граф соответствует универсальному отношению. Функция является частным случаем отношения и может быть представлена орграфом.

Отношение инцидентности между вершинами и рёбрами (узлами и дугами) также может быть представлено **графом (орграфом) инциденций**. Для графов отношение инцидентности симметрично, поэтому граф инциденций является двудольным графом (а не орграфом).

Для орграфов отношение инциденции задаётся орграфом. Граф инциденций гиперграфа — это двудольный граф без ограничений на степень вершин.

Игровые деревья, известные также как **деревья И/ИЛИ**, которые являются основной структурой данных в игровых программах в **символическом искусственном интеллекте**, фактически являются гиперграфами специального вида.

Узел v в орграфе $G(V,E)$ достижим из узла u , если существует путь из v в u . Путь из v в u обозначим. $\langle v \xrightarrow{} u \rangle$.

Теорема. Если отношение E есть строгое частичное упорядочение, то орграф $G(V,E)$ не имеет контуров.

Теорема. Если орграф $G(V,E)$ не имеет контуров, то отношение достижимости есть строгое частичное упорядочение.

Теорема. Если орграф не имеет контуров, то в нем есть узел, полустепень исхода которого равна 0.

Если E — бинарное отношение на V , то транзитивным замыканием $E^{\wedge +}$ на V будет отношение достижимости на орграфе $G(V,E)$. **Теорема.** Пусть M — матрица смежности орграфа $G(V,E)$. Тогда $M^k[i,j] = 1$ в том и только в том случае, если существует путь длиной k из узла v_i в узел v_j .

8.1. Компоненты связности (оценки числа рёбер)

На 1 балл:

План

1. Объединение графов и компоненты связности (*теорема о связности графа*)
 2. Точки сочленения, мосты и блоки (<- определения ; теоремы об эквивалентных утверждениях для вершин/ребер связного графа)
 3. Св-ва блоков (б свойств)
 4. Вершинная и реберная связность (определения)
 5. **Оценка числа ребер** (*Теорема об оценки ч. р. + 2 сл-я*)
-

На 2 балла:

Определения

1. Теорема: Граф связен т. и т.т., когда его нельзя представить в виде объединений 2х графов. ($G \neq G_1 \cup G_2$)

2. Точка сочленения (разделяющая вершина) – вершина графа.

Мост – ребро, удаление кот. увеличивает число компонент связности. **Блок** – связный граф без точек сочленения

Теоремы: (* $\langle u, v \rangle$ – **цепь** – марширует где все ребра различны>)

Теорема. Пусть $G(V, E)$ – связный граф и $v \in V$. Тогда следующие утверждения эквивалентны:

1. v – точка сочленения.
2. $\exists u, w \in V (u \neq w \wedge \forall \langle u, w \rangle_G (v \in \langle u, w \rangle_G))$.
3. $\exists U, W (U \cap W = \emptyset \wedge U \cup W = V - v \wedge \forall u \in U, w \in W (\forall \langle u, w \rangle_G (v \in \langle u, w \rangle_G)))$.

Теорема. Пусть $G(V, E)$ – связный граф и $x \in E$. Тогда следующие утверждения эквивалентны:

1. x – мост.
2. x не принадлежит ни одному простому циклу.
3. $\exists u, w \in V (\forall \langle u, w \rangle_G (x \in \langle u, w \rangle_G))$.
4. $\exists U, W (U \cap W = \emptyset \wedge U \cup W = V \wedge \forall u \in U (\forall w \in W (\forall \langle u, w \rangle_G (x \in \langle u, w \rangle_G))))$.

3. Теорема: Если график $G(V, E)$ связен и $p(G) > 2$ (число вершин > 2), то

- 1) G – нетривиальный блок;
- 2) любые две вершины принадлежат одному простому циклу;
- 3) любые вершина и ребро принадлежат одному простому циклу;
- 4) любые два ребра принадлежат одному простому циклу;
- 5) для любых двух вершин и любого ребра существует простая цепь, соединяющая эти вершины и содержащая это ребро;
- 6) для любых трёх вершин существует простая цепь, соединяющая две вершины и содержащая третью.

1185/1605

4. Вершинная связность ($\kappa(G)$) – наим. число вершин, удаление кот. приводит к несвязному или тривиальному графу. **Реберная связность** ($\lambda(G)$) – наим. число ребер, удаление которых приводит к несвязному или тривиальному графу.

Теорема: $\kappa(G) \leq \lambda(G) \leq \delta(G)$. $\delta(G)$ – мин. степень вершины графа

5.* Теорема:] p – число вершин, q – ребер, k – компонент связности. Тогда $p - k \leq q \leq (p - k)(p - k + 1)/2$.

(*Доказательство по индукции; *компоненты связности – класс эквивалентности по отношению связности)

Следствия:

=>[1] Если $q > (p-1)(p-2)/2$ то график связан

=>[2] В связном графике $p - 1 \leq q \leq p(p-1)/2$

8.2. Теорема Менгера (с доказательством)

На 1 балл:

1. Непересекающиеся цепи и разделяющие множества
2. Теорема Менгера в “вершинное форме”
3. Варианты теоремы Менгера

На 2 балла:

Определения:

1. Пусть $G(V, E)$ – связный граф, u и v – несмежные вершины.

Две цепи $\langle u, v \rangle$ называются

- **вершинно-непересекающимися**, если у них нет общих вершин, отличных от u и v .
- **реберно-пересекающимися**, если у них нет общий ребер.

Мн-во S вершин или ребер графа **разделяет** 2 вершины, если они принадлежат разным компонентам связности графа $G-S$. Разделяющее мн-во ребер называется **разрезом**.

Теорема. [Менгера] Пусть u и v – несмежные вершины в графе G . Наименьшее число вершин в множестве, разделяющем u и v , равно наибольшему числу вершинно-непересекающихся простых $\langle u, v \rangle$ -цепей:

2. Теорема Менгера.

$$\max |P(u, v)| = \min |S(u, v)|.$$

Замечание. $|P| \leq |S|$ для связного графа. Утверждение теоремы состоит в том, что в любом графе существуют такие P и S , что достигается равенство $|P| = |S|$

Доказательство.

Пусть G – связный граф, вершины u и v несмежны. **Совместная индукция по p и q .**

База: граф, сост. из 3 вершин, u, w, v , и двух рёбер, (u, w) и (w, v) . В нём $P(u, v) = \{\langle u, w, v \rangle\}$ и $S(u, v) = \{w\}$. Таким образом, $|P(u, v)| = |S(u, v)| = 1$.

Переход. Пусть утверждение теоремы верно для всех графов с числом вершин меньше p и (или) числом рёбер меньше q . Рассмотрим граф G с p вершинами и q ребрами. Пусть $u, v \in V$, причём u, v – не смежны и S - некоторое наименьшее множество вершин, разделяющее u и v . Обозначим $n = |S|$.

Рассмотрим три случая.

1 случай. Пусть в S есть вершины, несмежные с u и несмежные с v . Тогда граф $G-S$ состоит из двух нетривиальных графов, G_1 и G_2 . Образуем 2графа, стягивая G_1 и G_2 к вершинам u, v : $Gu = G/G_1$, $Gv = G/G_2$. Тогда S все еще наим. разделяющ. мн-во в Gu и $Gv \Rightarrow$ по индук. предп-ю в Gu и Gv есть n вершинно-непересек. простых цепей \Rightarrow комбинируя их, получим n непересекающихся цепей в G

2 случай. Пусть все вершины S смежны с u и среди них есть вершина w , смежная и с u , и с v .

Рассмотрим $G' = G-w$. В нем $(S-w)$ – наим. разделяющее мн-во для u и v . По инд. предп-ю $|S-w| = n-1$ вершинно непересек. простых цепей. Добавим к ним цепь $\langle u, w, v \rangle$ и получим n штук.

3 случай. Пусть все вершины S смежны с u и среди них нет вершин, смежных одновр с u и v .

Рассмотрим кратчайшую цепь $\langle u, v \rangle$ – цепь $\langle u, w_1, w_2, \dots, v \rangle$, $w_1 \in S$, $w_2 \neq v$

Рассмотрим граф $G' = G/\{w_1, w_2\}$ (т.е. склеим вершины w_1 и w_2 в вершину w_2) Имеем $w_2 \in S$, иначе цепь была бы ещё более короткой. Следовательно, в графе G' множество S по-прежнему является наименьшим, разделяющим u и v , и граф G' имеет по крайней мере на одно ребро меньше. По индукционному предположению в G' существуют n простых непересекающихся цепей. Но цепи, которые не пересекаются в G' , не пересекаются и в G . Получаем n цепей в G . Чтд

3. Варианты т.Менгера

Теорема. Для любых двух несмежных вершин u и v графа G наибольшее число рёберно-не-пересекающихся $\langle u, v \rangle$ -цепей равно наименьшему числу рёбер в $\langle u, v \rangle$ -разрезе.

Теорема. Чтобы граф G был n -связным, необходимо и достаточно, чтобы любые две несмежные вершины были соединены не менее чем n вершинно-непересекающимися простыми цепями.

Другими словами, в любом графе G любые две несмежные вершины соединены не менее чем $\chi(G)$ вершинно-непересекающимися простыми цепями.

8.3 Паросочетания (теорема Холла о свадьбах)

На 1 балл:

1. Задача о свадьбах, трансверсалах и паросочетаниях.	2. Теорема Холла*
3. Латинские квадраты	4. Устойчивые паросочетания
5. Наибольшее паросочетание*	6. Алгоритм поиска наибольшего паросочетания*

На 2 балла:

Паросочетанием (или **независимым множеством рёбер**) - множество рёбер, в котором никакие два ребра не смежны. Паросочетание называется **максимальным**, если никакое его надмножество не является независимым. Паросочетание называется **наибольшим**, если оно включает наибольшее возможное число рёбер.

1. **Задача о свадьбе.**] имеется конечное множество юношей, каждый из которых знаком с некоторым подмножеством конечного множества девушек. В каком случае всех юношей можно женить так, чтобы каждый женился на знакомой девушке?

Задача о трансверсалах.] $S = \{S_1, \dots, S_m\}$ – семейство непустых подмножеств конечного множества E (могут пересекаться). **Системой различных представителей в S (трансверсалю)** называется \forall подмножество $C = \{c_1, \dots, c_m\}$ из m элементов $E : \forall k \in 1 \dots m (c_k \in S_k)$. В каком случае существует трансверсаль?

Задача о паросочетаниях. Пусть $G(V_1, V_2, E)$ – двудольный. **Совершенным паросочетанием** из V_1 в V_2 называется паросочетание, покрывающее вершины V_1 . *(Является максимальным и наибольшим). В каком случае существует совершенное паросочетание из V_1 в V_2 ?

Ответ. $\forall k \left(\frac{\text{юношей}}{\text{подмножеств}} \right)$ в совокупности $\left(\frac{\text{знакомы с}}{\text{содержат}} \right)$ не менее чем $k \left(\frac{\text{девушками}}{\text{элементов}} \right)$.

2. **ТЕОРЕМА ХОЛЛА.** Пусть $G(V_1, V_2, E)$ — двудольный граф. Совершенное паросочетание из V_1 в V_2 существует тогда и только тогда, когда $\forall A \subset V_1 (|A| \leq |\Gamma(A)|)$.

СЛЕДСТВИЕ. В любом не пустом регулярном двудольном графе $G(V_1, V_2)$ существует совершенное паросочетание.

3. Матрица $L : array [1..m, 1..n] of M$, где $M := \{a_1, \dots, a_n\}$ называется **латинским прямоугольником**, если элементы каждой строки этой матрицы образуют перестановку множества M , и в каждом столбце все элементы разные. По принципу Дирихле число строк латинского прямоугольника m не превосходит числа его столбцов n . $m = n \Rightarrow$ **латинский квадрат** порядка n .

Теорема. Любой латинский прямоугольник m на n , можно дополнить до латинского квадрата порядка n приписыванием новых $n - m$ строк.

Два латинских квадрата L_1 и L_2 называются **ортогональными**, если все упорядоченные пары $(L_1[i, j], L_2[i, j])$ различны.

4.] задано совершенное взаимно-однозначное паросочетание из множества мужчин в множество женщин и обратно (брачное отношение): Aa, Bb , т. д. Пусть теперь у каждого мужчины есть список женщин, упорядоченный по убыванию «привлекательности» для этого мужчины. Аналог для женщин. Паросочетание называется **неустойчивым**, если в нём найдутся мужчина X (брак с x) и женщина y (брак с Y), предпочитающие друг друга своим супругам, то есть:

- | | |
|---|---|
| 1) X считает y привлекательней, чем x ; | 2) y считает X привлекательней, чем Y . |
|---|---|

Паросочетание называется **устойчивым**, если таких пар в нём нет.

Поиск совершенного устойчивого паросочетания в полном двудольном графе реализуется **алгоритмом отложенного согласия**.

5. Простая цепь (ПЦ) называется **альтернирующей** относительно паросочетания M , если содержит ребра из M и из не M попарно. **Аугментальной** – если крайние ребра $\notin M$.

Вершины, инцидентные ребрам паросочетания M – **занятые**, а прочие – **свободные**.

Лемма. Э аугментальная цепь \Rightarrow паросочетание не наибольшее.

Теорема. Паросочетание наибольшее \Leftrightarrow \nexists аугментальной цепи.

Паросочетание **полное**, если не оставляет свободных вершин. Полное \Rightarrow наибольшее.

8.5 Связность в орграфах (алгоритм выделения сильной компоненты связности)

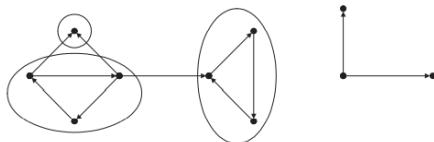
На 1 балл:

1. Сильная, односторонняя, слабая связность*	2. Компоненты сильной связности
3. Выделение компонент сильной связности	4. Базисы ориентированных графов

На 2 балла:

1. $G(V, E)$ – орграф, v_1, v_2 – узлы. **Сильная связность** \Leftrightarrow Эпуть (ор. цепь) из v_1 в v_2 и v_2 в v_1 .
Односторонняя $\Leftrightarrow \exists v_1 \rightarrow v_2$ или $v_2 \rightarrow v_1$. **Слабая** \Leftrightarrow связаны в G' , полученном из G забыванием ориентации дуг. Все вершины (сильно, односторонне, слабо) связаны \Leftrightarrow орграф (сильно, односторонне, слабо) связный.
Лемма. В \forall бесконтурном орграфе $G(V, E)$ узлы можно перенумеровать так, что у \forall дуги номера начала < номера конца: $\forall(v_i, v_j) \in E(i < j)$.
2. **Компонента сильной связности (КСС)** – максимальный сильный связный подграф. Каждый узел принадлежит только одной КСС. Если узел не связан сильно с другими \Rightarrow сам КСС. **Конденсацией** G^* орграфа G (**графом Герца, фактор-графом**) называется орграф, который получается стягиванием КСС в один узел.

Пример. На рисунке показаны диаграммы орграфа и его конденсации.



3. Алгоритм выделения компонент сильной связности.

Вход: $G(V, E)$ – список смежности $\Gamma(v)$.
Выход: список C компонент сильной связности.
 $C := \emptyset$
for $v \in V$ **do**
 $M[v] := \{v\}$ // $M[v]$ список узлов, \in КСС, что и v
 $e[v] := 0$ // все узлы не рассмотрены
end for
while $V \neq \emptyset$ **do**
 select $v \in V$; $v \rightarrow T$; // взять v , положить в стек
 $e[v] := 1$; КСС // отметить и вызвать КСС
end while

КСС:

```

if  $T = \emptyset$  then return end if // негде выделять
 $v := \text{top } T$  //  $v$  – верхний элемент стека
if  $\Gamma[v] = \emptyset$  then
   $C := C \cup M[v]$ ;  $V := V - v$ ;  $v \leftarrow T$  // это КСС, удалить узел, снять узел  $v$  со стека
  KCC // возврат из тупика
else
  for  $u \in \Gamma[v]$  do
    if  $e[u] = 0$  then
       $u \rightarrow T$ ;  $e[u] := 1$  // положить узел  $u$  в стек и отметить
    else
      repeat
         $w \leftarrow T$ ;  $V := V - w$ ;  $\Gamma[w] := \Gamma[u] \cup \Gamma[w]$ ;  $M[w] := M[u] \cup M[w]$ 
      until  $u = w$ 
       $w \rightarrow T$ ;  $V := V + w$  // чтобы не убрать узел, с которого начали
    end if
    KCC // поиск в глубину
  end for
end if

```

Обоснование. \forall контур принадлежит ровно 1 КСС; если в КСС несколько узлов, то они обязательно принадлежат некоторым контурам. При обходе в глубину мы попадаем в уже отмеченный узел u \Rightarrow обнаружен контур, причём предшествующие узлы найденного контура находятся на стеке, начиная от вершины стека и до рассматриваемого отмеченного узла u , который также присутствует в стеке. Все узлы найденного контура можно «склеить». При склеивании узла w его список смежности и список уже найденных узлов той КСС, которой принадлежит узел w , объединяются с соответствующими списками узла u . После этого можно продолжить поиск в глубину от узла u , который для этой цели следует оставить в стеке.

Следствие. Конденсация любого графа содержит источники.

Алгоритм Косарайо с использованием обратного орграфа.

Вход: орграф $G(V, E)$, заданный списками смежности $\Gamma : \text{array}[1..p] \text{ of set of } 1..p$.

Выход: массив разметки КСС $M : \text{array}[1..p] \text{ of } 1..p$, причём $M[u] = M[v]$ тогда и только тогда, когда узлы u и v принадлежат одной КСС.

```

 $X : \text{array}[1..p] \text{ of } 0..1$  //отметка посещения узла
for  $v$  from 1 to  $p$  do  $X[v] := 0$  end for //вначале все узлы не посещены
 $\Gamma^{-1} : \text{array}[1..p] \text{ of set of } 0..1$  //списки смежности обратного орграфа
for  $v$  from 1 to  $p$  do  $\Gamma^{-1}[v] := \emptyset$  end for //вначале списки пусты
 $T : \text{array}[1..p] \text{ of } 0..1$  //массив узлов в порядке, обратном порядку выхода из узла
for  $v$  from 1 to  $p$  do  $T[v] := 0$  end for //вначале последовательность выходов неизвестна
 $t := p$  //величина, обратная времени выхода
for  $v$  from 1 to  $p$  do  $F(v)$  end for //обход исходного орграфа в глубину
for  $v$  from 1 to  $p$  do  $B(T[v], T[v])$  end for //обход обратного орграфа в глубину

```

1259/1605

Рекурсивная процедура $F(u)$ обхода в глубину орграфа определяет порядок выхода и заодно строит списки смежности обратного орграфа.

```

if  $X[u] = 0$  then
     $X[u] := 1$  //отмечаем посещение узла
    for  $w \in \Gamma[u]$  do
         $F(w)$  //рекурсивный вызов в глубину
         $\Gamma^{-1}[w] := \Gamma^{-1}[w] + u$  //записываем  $u$  как смежный с  $w$  в обратном орграфе
    end for
     $T[t] := u$  //запись времени выхода в обратном порядке
     $t := t - 1$  //уменьшение счётчика времени выхода
end if

```

Рекурсивная процедура $B(u, v)$ обхода в глубину обратного орграфа относит узлы u и v к одной КСС.

```

if  $X[u] = 1$  then
     $X[u] := 0$  //отмечаем посещение узла
     $M[u] := v$  //у  $v$  принадлежит той же КСС, что и  $u$ 
    for  $w \in \Gamma^{-1}[u]$  do
         $B(w, v)$  //рекурсивный вызов в глубину
    end for
end if

```

Обоснование. Если начать поиск в глубину с любого узла u в любой КСС, то алгоритм посетит все узлы v в данной КСС, для которых $\exists u \rightarrow v, v \rightarrow u$. Граф не сильно связан \Rightarrow алгоритм мб посетит «лишние» узлы (\exists путь $u \rightarrow v, \nexists v \rightarrow u$). \exists путь $u \rightarrow v \& \nexists$ пути $v \rightarrow u \Rightarrow$ при любом обходе в глубину алгоритм выйдет из узла u позже, чем из узла v . Узел v пройден в результате $B(v, u)$ с началом в $u \Rightarrow 1)$ узел u позднее выходит, чем v (u левее v в массиве T); 2) \exists путь $u \rightarrow v$ в обратном графе ($\exists v \rightarrow u$ в обычном). При этом в исходном орграфе \exists путь $u \rightarrow v$, который вместе с обратным путем $v \rightarrow u$ даёт сильную связность узлов и заключение, что u и v принадлежат одной КСС. Вариант, $\exists v \rightarrow u$ в исходном графе $\& \nexists u \rightarrow v$ в обратном невозможен, т.к. противоречит тому, что узел v правее в массиве T .

4. Множество узлов B орграфа $G(V, E)$ – **порождающее** $\Leftrightarrow \forall$ узел орграфа достижим из некоторого узла \in этому множеству: $\forall v \in V \left(\exists u \in B \left(\exists \overrightarrow{(u, v)} \in G \right) \right)$.

Порождающее множество узлов – **базис** орграфа, если минимально по включению.

Лемма. Орграф не имеет контуров $\Rightarrow \exists!$ базис, сост. из всех источников.

Следствие. Конденсация орграфа имеет единственный базис, сост. из источников.

Базисными компонентами сильной связности называются такие компоненты сильной связности, которым соответствуют источники в конденсации этого орграфа.

Теорема. Подмножество узлов орграфа образует базис \Leftrightarrow оно является трансверсалю базисных компонент сильной связности.

Следствие. Все базисы любого орграфа равномощны.

8.6-Кратчайшие пути

На 1 балл:

1. Взвешенные орграфы
2. Кратчайшие пути в бесконтурном графе
3. Алгоритм Беллмана–Форда
4. Алгоритм Дейкстры
5. Область применения алгоритма Дейкстры
6. Анализ алгоритма Дейкстры
7. Алгоритм Флойда–Уоршалла
8. Алгоритмы с предобработкой

На 2 балла:

1) Взвешенные орграфы

Замечание. Если дуга $e = (u, v) \in E$ ведёт из узла u в узел v , то записи $w(e)$ и $w(u, v)$ обе считаются допустимыми и равноправными.

Если $M = \langle \vec{s}, \vec{t} \rangle$ – путь из узла s в узел t , то **длиной пути** называется величина

$$W(M) = W(\langle \vec{s}, \vec{t} \rangle) = W(s, t) = \sum_{e \in M} w(e),$$

а **длиной кратчайшего пути** называется величина

Орграф $G(V, E)$ называется **взвешенным**, если задана числовая функция $w: V \times V \rightarrow \mathbb{R}$, называемая **весом** или **длиной** дуги.

$$\Omega(s, t) = \min_{\langle \vec{s}, \vec{t} \rangle \subset E^*} W(s, t).$$

Теорема. Для взвешенных орграфов выполняется неравенство $\Omega(s, v) \leq \Omega(s, u) + \Omega(u, v)$ (неравенство треугольника)

2) Кратчайшие пути в бесконтурном графе

Существует эффективный алгоритм определения расстояний от источника, который позволяет найти в орграфе дерево кратчайших путей, растущее из заданного узла, даже если длины дуг могут быть отрицательными, но известно, что орграф не содержит контуров.

3) Алгоритм Беллмана–Форда

Алгоритм поиска кратчайшего пути во взвешенном графе. За время $O(|V|^{|E|})$ алгоритм находит кратчайшие пути от одной вершины графа до всех остальных. (Здесь V — множество вершин графа, E — множество его рёбер). Алгоритм Беллмана–Форда корректно работает на любых орграфах.

4) Алгоритм Дейкстры

Находит кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса.

5) Область применения алгоритма Дейкстры

Назовём **рекордом пути** $M = \langle \vec{u}, \vec{w} \rangle$ (обозначение $R(\langle \vec{u}, \vec{w} \rangle)$) максимальный вес начального отрезка этого пути, $R(\langle \vec{u}, \vec{w} \rangle) \stackrel{\text{Def}}{=} \max_{v \in M - u} W(u, v)$. Отдельно отметим, что $v \neq u$, то есть рекорд не может достигаться в начальном узле по определению. Из определения следует, что

$$\forall \langle \vec{u}, \vec{w} \rangle \quad \left(R(\langle \vec{u}, \vec{w} \rangle) \geq W(\langle \vec{u}, \vec{w} \rangle) \geq \Omega(u, w) \right).$$

Теорема. Алгоритм Дейкстры на взвешенном орграфе без контуров отрицательного веса строит дерево кратчайших путей с корнем в узле s тогда и только тогда, когда для каждого узла $t \in G - s$, достижимого из узла s , в сети $P(s, t)$ для каждой развилики $P(w, t)$ рекорд кратчайшего пути $\langle \vec{w}, \vec{t} \rangle$ строго меньше, чем рекорд любого не кратчайшего пути $\langle \vec{w}, \vec{t} \rangle$.

6) Анализ алгоритма Дейкстры

Алгоритм Дейкстры корректен на классе неотрицательно взвешенных орграфов.

7) Алгоритм Флойда–Уоршалла

В отличие от предыдущих алгоритмов, алгоритм Флойда–Уоршалла решает задачу о поиске не дерева кратчайших путей, а всех кратчайших путей в орграфе, и делает это за время $O(p^3)$.

8) Алгоритмы с предобработкой

Предобработка- предварительное преобразование данных задачи в более выгодный для решения вид. Алгоритм Джонсона: в этом алгоритме поиска кратчайших путей между всеми узлами орграфа используется предобработка типа «перевзвешивание» — введение функции потенциала.

Каждому узлу v присваивается потенциал $h(v)$. Веса дуг переопределяются. $O(p^2 * \log p + pq)$.

9.1. Свободные деревья (теорема об основных свойствах свободных деревьев)

На 1 балл:

1. Свободные деревья и их элементы;
2. Основные свойства деревьев (теорема об основных свойствах свободных деревьев);
3. Центр дерева (теорема о центре свободного дерева).

Основные понятия и теоремы:

Ациклический граф; Дерево (свободный граф); Древочисленный граф; Субциклический граф; Хорда; Теорема об основных свойствах свободных деревьев; Теорема о центре свободного дерева;

На 2 балла:

Граф без циклов называется **ациклическим**, или лесом. Связный ациклический граф называется **(свободным) деревом**. Таким образом, компонентами связности леса являются деревья.

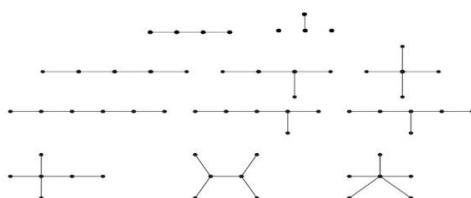
ЗАМЕЧАНИЕ: Слово «свободное» употребляется в том случае, когда нужно подчеркнуть отличие деревьев от других объектов, родственных деревьям: ориентированных деревьев, упорядоченных деревьев и т.д.

В связном графе G выполняется неравенство $q(G) \geq p(G) - 1$. Граф G (не обязательно связный!), в котором $q(G) = p(G) - 1$, называется **древочисленным**. В ациклическом графе G $z(G) = 0$. Пусть u, v — несмежные вершины графа $G, x = (u, v)$ знак (не принадлежит) E . Если граф $G+x$ имеет ровно один простой цикл, $z(G+x) = 1$, то граф G (не обязательно ациклический!) называется **субциклическим**.

ЗАМЕЧАНИЕ: Графы $K_3 \cup K_1$ и $K_3 \cup K_2$ являются древочисленными и субциклическими и в то же время не являются ни связными, ни ациклическими.

Ребро, соединяющее несмежные вершины свободного дерева, называется **хордой**. Хорда дерева не принадлежит дереву!

ПРИМЕР: На рисунках последовательно показаны диаграммы всех различных свободных деревьев с 4, 5 и 6 вершинами.



Основные свойства деревьев: ТЕОРЕМА: Пусть $G(V, E)$ — граф с p вершинами, q ребрами, k компонентами связности и z простыми циклами. Пусть далее x — ребро, соединяющее любую пару несмежных вершин в G . Тогда следующие утверждения эквивалентны.

1. G — дерево, то есть связный граф без циклов, $k(G) = 1$ & $z(G) = 0$.
2. Любые две вершины соединены в G единственной простой цепью, $\forall u, v / P(u, v) = 1$.
3. G — связный граф, и любое ребро есть мост, $k(G) = 1$ & $\forall e \in E / (k(G-e) > 1)$.
4. G — связный и древочисленный, $k(G) = 1$ & $q(G) = p(G) - 1$.
5. G — ациклический и древочисленный, $z(G) = 0$ & $q(G) = p(G) - 1$.
6. G — древочисленный и субциклический (за двумя исключениями), $q(G) = p(G) - 1$, $G \neq K_1 \cup K_3$, $G \neq K_2 \cup K_3$, $z(G+x) = 1$.
7. G — ациклический и субциклический, $z(G) = 0$ & $z(G+x) = 1$.
8. G — связный, субциклический и неполный, $k(G) = 1$, $G \neq K_p$, $p > 3$, $z(G+x) = 1$.

СЛЕДСТВИЕ 1: В любом нетривиальном дереве имеются по крайней мере две висячие вершины.

ЗАМЕЧАНИЕ Легко видеть, в частности, что висячими вершинами в дереве являются концы любого диаметра.

СЛЕДСТВИЕ 2: Каждая не висячая вершина свободного дерева является точкой сочленения.

СЛЕДСТВИЕ 3: Если в связном графе нет висячих вершин, то в нём есть цикл.

Центр дерева Свободные деревья выделяются из других графов тем, что их центр всегда оправдывает своё название. **ТЕОРЕМА:** Центр свободного дерева состоит из одной вершины или из двух смежных вершин: ($z(G) = 0$ & $k(G) = 1$) $\Rightarrow (C(G) = K_1 \vee C(G) = K_2)$.

9.2 - Ориентированные, упорядоченные и бинарные деревья

На 1 балл:

Ориентированные, упорядоченные и бинарные деревья:

9.2.1. Ориентированные деревья.

Термины: ордерево, листы, крона, высота, уровень узла.

Теоремы: о св-вах ордерева

9.2.2. Эквивалентные определения ордерева.

9.2.3. Упорядоченные деревья.

Теоремы: о прямом обходе и нумерации предков и потомков:

9.2.4. Бинарные деревья.

Термины: бинарное дерево, т-ичное дерево

На 2 балла:

Ориентированным деревом (или **ордеревом**, или **корневым** деревом) называется орграф со следующими свойствами:

1. существует единственный узел r , полустепень захода которого равна 0, он называется **корнем** ордерева;
2. полустепень захода всех остальных узлов равна 1;
3. каждый узел достижим из корня.

Теорема о свойствах ордерева:

- 1) ордерево древочисленно, $q = p - 1$;
- 2) если в ордереве забыть ориентацию дуг, то получится свободное дерево;
- 3) в ордереве нет контуров;
- 4) для каждого узла существует единственный путь, ведущий в этот узел из корня;
- 5) подграф, определяемый множеством узлов, достижимых из узла v , является ордеревом с корнем v (**поддеревом** узла v);
- 6) если в свободном дереве любую вершину назначить корнем, то получится ордерево

Подграф, определяемый множеством узлов, достижимых из узла v , является ордеревом с корнем v (**поддеревом** узла v).

Сток ордерева называется **листом**. Множество листьев называется **кроной**. Путь из корня в лист называется **ветвью**. Длина наибольшей ветви ордерева называется его **высотой**.

Уровень узла ордерева — это расстояние от корня до узла. Сам корень имеет уровень 0.

Узлы одного уровня образуют **ярус** ордерева.

Узлы, достижимые из узла u , называются **потомками** узла u (потомки одного узла образуют дерево).

Если узел v является потомком узла u , то узел u называется **предком** узла v . Если в дереве существует дуга $(u; v)$, то узел u называется **отцом** (или **родителем**)

узла v , а узел v называется сыном узла u . Сыновья одного отца называются **братьями**.

Ордерево Т — это непустое конечное множество узлов, на котором определено разбиение, обладающее следующими свойствами:

1. Имеется один выделенный однэлементный блок frg , называемый корнем данного ордерева.
2. Остальные узлы (исключая корень) содержатся в k ($k > 0$) блоках T каждый из которых, в свою очередь, является ордеревом и называется поддеревом.

Теорема о прямом обходе и нумерации предков и потомков:

В упорядоченном ордереве с p узлами существует такая нумерация узлов числами из диапазона $1:p$, что номера потомков больше номеров предков и номера старших братьев больше номеров младших братьев

Если относительный порядок поддеревьев $T_1.., T_k$ в эквивалентном определении ордерева фиксирован, то ордерево называется **упорядоченным**.

Бинарное (или **двоичное**) дерево — это непустое конечное множество узлов, на котором определена структура, обладающая следующими свойствами:

1. имеется один выделенный узел r , называемый корнем данного бинарного дерева;
2. остальные узлы (исключая корень) содержатся в двух непересекающихся множествах (поддеревьях) — левом и правом, каждое из которых, в свою очередь, либо пусто, либо является бинарным деревом.

т-ичным деревом называется непустое конечное множество узлов, которое состоит из корня и t непересекающихся подмножеств, имеющих номера $1.., t$, каждое из которых, в свою очередь, либо пусто, либо является t -ичным деревом.

9.3 - Представление деревьев в программах

На 1 балл:

Представление деревьев в программах

9.3.1. Представление свободных деревьев

Алгоритмы упаковки и распаковки

9.3.2. Представление упорядоченных корневых деревьев

Алгоритм построения кода упорядоченного ордерева

9.3.3. Число упорядоченных ориентированных деревьев

9.3.4. Проверка правильности скобочной структуры

Алгоритм проверки правильности скобочной структуры.

9.3.5. Представление бинарных деревьев

Термины: Польская запись,

9.3.6. Обходы бинарных деревьев

Термины: прошитое дерево

На 2 балла:

Код Профера – один из способов представления свободных деревьев. Строится **алгоритмом упаковки**.

Алгоритм построения кода упорядоченного ордерева – алгоритм, который сгенерирует битовую шкалу (код) из $2q$ разря-дов, где q — число дуг в ордереве.

Восстановление упорядоченного ордерева по коду

алгоритм проверки правильности скобочной структуры

Способы представления:

1. Списочные структуры: каждый узел представляется записью, содержащей два поля (*l* и *r*) с указателями на левый и правый узлы и еще одно поле *i* для хранения указателя на информацию об узле некоторого типа *info*.
2. Упакованные массивы: узлы располагаются в массиве, так что все узлы поддерева данного узла располагаются вслед за этим узлом. Вместе с каждым узлом хранится индекс узла, который является первым узлом правого поддерева данного узла.
3. Размеченная степень: аналогично упакованным массивам, но вместо связей фиксируется «размеченная степень» каждого узла. Например, 0 означает, что это лист, 1 — есть правая связь, но нет левой, 2 — есть левая связь, но нет правой, 3 — есть обе связи. Для представления размеченной степени в случае бинарных деревьев достаточно битовой шкалы длины 2.

Польская запись – способ представления деревьев, не хранящий степень узла, если она известна из информации, хранящейся в самом узле.

Большинство алгоритмов работы с деревьями основаны на обходах. Возможны следующие основные обходы бинарных деревьев.

1. **Прямой (префиксный, левый)** обход: попасть в узел, обойти левое поддерево, обойти правое поддерево.
2. **Внутренний (инфиксный, симметричный)** обход: обойти левое поддерево, попасть в узел, обойти правое поддерево.
3. **Концевой (постфиксный, правый)** обход: обойти левое поддерево, обойти правое поддерево, попасть в узел.

Деревья, в которых пустые поля *l* и *r* в структуре *N* используются для хранения дополнительных связей, называются **прошитыми деревьями**.

9.6. Кратчайший остов. (алгоритм Краскала)

На 1 балл:

План:

1. Стягивающие деревья и остовы
2. Схема алгоритма построения кратчайшего остова
3. Алгоритм Краскала
4. Алгоритм Прима
5. (задача Штейнера) - дополнительно

Ключевые понятия:

Остовный подграф графа $G(V, E)$ – подграф, содержащий все вершины

Остов (или каркас) – остовный подграф, являющийся деревом.

Стягивающее дерево – подграф, являющийся и остовным подграфом и деревом одновременно

Кратчайший остов – (нужно: все ребра имеют длину) остов минимальной длины

На 2 балла:

1.

*остов и стягивающее дерево – одно и то же

*остов определяется множеством ребер

*несвязный граф не имеет остова (имеет остовный лес)

*связный граф может иметь много остовов

2. Рассмотрим следующую схему алгоритма построения кратчайшего остова. Пусть T – множество непересекающихся деревьев, являющихся подграфами графа G . Вначале T состоит из отдельных вершин графа G , в конце T содержит единственный элемент – кратчайший остов графа G .

Вход: граф $G(V, E)$, заданный матрицей длин рёбер C .

Выход: кратчайший остов T .

$T := V$

while в T больше одного элемента do

 взять любое поддерево из T

 найти к нему ближайшее

 соединить эти деревья в T

end while

3. Для обоснования алгоритма Краскала достаточно показать, что выдерживается схема алгоритма предыдущего параграфа. Действительно, поскольку в множестве рёбер T нет циклов по построению, это множество суть совокупность рёбер некоторого множества деревьев. Если множество T содержит более одного дерева, то существует ребро, при добавлении которого не возникает цикла – оно соединяет два дерева в T . Добавляемое ребро – кратчайшее возможное, значит, на нём реализуется расстояние между некоторыми деревьями в T . По построению в конце работы алгоритма множество рёбер T содержит $p - 1$ элемент, а значит, является искомым остовом.

4. В алгоритме Прима кратчайший остов порождается в процессе разрастания одного дерева, к которому присоединяются ближайшие одиночные вершины. Каждая одиночная вершина является деревом, и схема алгоритма п. 2 выполнена.

5. В этой задаче допускается использование дополнительных вершин, чтобы уменьшить расстояние

10.1. Фундаментальные циклы и разрезы (фундаментальная система циклов и циклический ранг) (жесть конечно тут..)

На 1 балл:

План:

1. Циклы и разрезы
2. Фундаментальная система циклов и циклический ранг
3. Фундаментальная система разрезов и коциклический ранг
4. Подпространства циклов и коциклов

На 2 балла:

Ключевые понятия:

Простой цикл – цикл, никакое собственное подмножество которого циклом не является.

(*разрез связного графа - множество рёбер, удаление которых делает граф несвязным.)

Правильный разрез – разрез связного графа $G(V, E)$, определяемый непустым подмножеством U множества вершин V

*Правильный разрез не содержит «лишних» рёбер, то есть таких рёбер, включение или исключение которых не меняет компонент связности, получаемых при удалении рёбер разреза. Ясно, что всякий разрез содержит некоторый правильный разрез.

Простой разрез – минимальный разрез, то есть такой разрез, никакое собственное подмножество которого разрезом не является

Пусть $T(V, E_T)$ — некоторый остов графа $G(V, E)$. **Кодеревом** T^* (V, E_T^*) остова T называется остовный подграф, такой, что $E_T^* = E \setminus E_T$. (Кодерево не является деревом!) Рёбра кодерева являются хордами остова. По теореме п 9.1.2 об основных свойствах деревьев каждая хорда $e \in T^*$ остова T порождает ровно один простой цикл, обозначим его Z_e . Таким образом, имеем систему простых циклов $Z \text{ Def} = \{Z_e\}$ $e \in T^*$, определяемых выбранным остовом T , которая называется **фундаментальной системой циклов**. Циклы фундаментальной системы называются **фундаментальными**, а количество циклов в (данной) фундаментальной системе называется **циклическим рангом** (или цикломатическим числом) графа G и обозначается $m(G)$.

Иногда множество объектов, определяемое всевозможными циклическими разностями фундаментальных циклов, называют **циклическими векторами**.

Пусть опять $T(V, E_T)$ – некоторый остов графа $G(V, E)$. Рассмотрим ребро остова $e \in E_T$ и определим разрез S_e следующим образом. Так как ребро e – разрез дерева T , множество вершин V разбивается на два непустых подмножества, V_1 и V_2 , так что $V_1 \subset V$, $V_1 \neq \emptyset$, $V_2 \subset V$, $V_2 \neq \emptyset$, $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$. Включим в разрез S_e ребро e и все хорды остова T , соединяющие вершины V_1 с вершинами V_2 :

$$S_e \stackrel{\text{Def}}{=} e + \{(v_1, v_2) \in T^* \mid v_1 \in V_1 \& v_2 \in V_2\}.$$

Тогда S_e – это простой разрез. Система разрезов $\mathcal{S} \stackrel{\text{Def}}{=} \{S_e\}_{e \in T}$ называется **фундаментальной системой разрезов**. Разрезы фундаментальной системы называются **фундаментальными**.

Множество циклов $\{Z_i\}_{i=1}^n$ называется **независимым**, если ни один из циклов Z_i не является линейной комбинацией остальных.

Множество разрезов $\{S_i\}_{i=1}^n$ называется **независимым**, если ни один из разрезов S_i не является линейной комбинацией остальных.

Максимальное независимое множество циклов (или минимальное множество циклов, от которых зависят все остальные, или независимое порождающее множество циклов) является **базисом** пространства циклов. Максимальное независимое множество разрезов (или минимальное множество разрезов, от которых зависят все остальные, или независимое порождающее множество разрезов) является **базисом** пространства разрезов.

10.2 Эйлеровы циклы

На 1 балл:

10.2.1 Эйлеровы графы

10.2.2 Построение эйлерова цикла в эйлеровом графе

10.2.3 Оценка числа эйлеровых графов

На 2 балла:

10.2.1

Если связный граф имеет цикл (не обязательно простой), содержащий все рёбра графа, то такой цикл называется **эйлеровым циклом**, а граф называется **эйлеровым графом**. Если связный граф имеет цепь (не обязательно простую), содержащую все рёбра, то такая цепь называется **эйлеровой цепью**, а граф называется **полуэйлеровым графом**. И эйлеров цикл и эйлерова цепь содержат не только все рёбра, но и все вершины графа (возможно, по нескольку раз).

Теорема. Если граф G связан и нетривиален, то следующие утверждения эквивалентны.

1. G — эйлеров граф.
2. Каждая вершина G имеет чётную степень.
3. Множество рёбер G можно разбить на простые циклы

10.2.2

Обоснование. Начав с произвольной вершины v , строим путь, удаляя рёбра и запоминая вершины в стеке, до тех пор, пока множество смежности очередной вершины не окажется пустым, что означает, что

Вход: эйлеров граф $G(V, E)$, заданный списками смежности ($\Gamma[v]$ — список вершин, смежных с вершиной v).

Выход: последовательность вершин эйлерова цикла.

```
S := ∅ //стек для хранения вершин
select v ∈ V //произвольная вершина
v → S //положить v в стек S
while S ≠ ∅ do
    v := top S //v — верхний элемент стека
    if Γ[v] = ∅ then
        v ← S; yield v //очередная вершина эйлерова цикла
    else
        select u ∈ Γ[v] //взять первую вершину из списка смежности
        u → S //положить u в стек
        Γ[v] := Γ[v] - u; Γ[u] := Γ[u] - v //удалить ребро (v, u)
    end if
end while
```

путь уединить нельзя. Заметим, что при этом мы обязательно придём в ту вершину, с которой начали. В противном случае вершина v имеет нечётную степень, что невозможно по условию. Таким образом, из графа были удалены рёбра цикла, а вершины цикла были сохранены в стеке S , при этом степени всех вершин остались чётными. Далее вершина v выводится в качестве первой вершины эйлерова цикла, а процесс продолжается с вершины, стоящей на вершине стека.

10.2.3

Пусть $\zeta(p)$ — множество всех графов с p вершинами, а $\varepsilon(p)$ — множество эйлеровых графов с p вершинами.

Теорема. Эйлеровых графов почти нет, то есть

$$\lim_{p \rightarrow \infty} \frac{|\mathcal{E}(p)|}{|\mathcal{G}(p)|} = 0.$$

10.3 Гамильтоновы циклы

На 1 балл:

10.3.1 Гамильтоновы циклы

10.3.2 Задача коммивояжёра

10.3.3 Теорема Поша

На 2 балла:

Если граф имеет простой цикл, содержащий все вершины графа (по одному разу), то такой цикл называется **гамильтоновым циклом**, а граф называется **гамильтоновым графом**.

Если граф имеет простую цепь, содержащую все вершины графа (по одному разу), то такая цепь называется **гамильтоновой цепью**, а граф называется **полугамильтоновым графом**.

Ясно, что гамильтоновым может быть только связный граф

Замечание. Любой граф G можно превратить в гамильтонов, добавив достаточное количество новых вершин и инцидентных им рёбер и не добавляя рёбер, инцидентных только старым вершинам.

Теорема. Если $p(G) \geq 3$ и $\delta(G) \geq \frac{p}{2}$, то граф G является гамильтоновым.

Следствие. При $p \geq 3$ полный граф K_p гамильтонов.

10.3.2

Имеется p городов, расстояния между которыми известны. Коммивояжёр должен посетить все p городов по одному разу, вернувшись в тот, с которого начал. Требуется найти маршрут движения, при котором суммарное пройденное расстояние будет минимальным. Ясно, что задача коммивояжёра — это задача отыскания кратчайшего гамильтонова цикла в нагруженном полном графе. Более того, известно, что задача коммивояжёра принадлежит к числу **NP-полных задач**.

Вкратце суть проблемы NP-полноты сводится к следующему. Есть задачи, принадлежащие к числу наиболее фундаментальных, для которых не удалось найти алгоритмов решения, имеющих полиномиальную сложность. Более того, если бы удалось отыскать эффективный алгоритм решения хотя бы одной из этих задач, то из этого немедленно следовало бы существование эффективных алгоритмов для всех остальных задач данного класса.

Можно показать, что почти все графы — гамильтоновы, то есть

$$\lim_{p \rightarrow \infty} \frac{|\mathcal{H}(p)|}{|\mathcal{G}(p)|} = 1,$$

где $\mathcal{H}(p)$ — множество гамильтоновых графов с p вершинами, а $\mathcal{G}(p)$ — множество всех графов с p вершинами.

В степенной последовательности гамильтонова графа не может быть много вершин с малыми степенями. Введём обозначение $N_=(G, n)$ числа вершин графа G степени n , $0 \leq n < p$:

$$N_=(G, n) \stackrel{\text{Def}}{=} |\{v \in V \mid d(v) = n\}|.$$

В обозначениях:

$$N_{\leq}(G, n) \stackrel{\text{Def}}{=} |\{v \in V \mid d(v) \leq n\}| = \sum_{i=0}^n N_=(G, i), \quad N_{>}(G, n) \stackrel{\text{Def}}{=} |\{v \in V \mid d(v) > n\}| = \sum_{i=n+1}^{p-1} N_=(G, i),$$

$$N_{\leq}(G, n) + N_{>}(G, n) = p.$$

Теорема. [Поша] Если $G(V, E)$ — связный граф с числом вершин $p \geq 3$,

и $\forall n, 1 \leq n < (p-1)/2$ ($N_{\leq}(G, n) < n$),

и для нечётных p выполняется неравенство $N_{\leq}(G, (p-1)/2) \leq (p-1)/2$,

то граф G гамильтонов.

Следствие. Если $p \geq 3$ и $d(u) + d(v) \geq p$ для любой пары u и v несмежных вершин графа G , то G — гамильтонов граф.

10.4 – Независимые и покрывающие множества (связь чисел независимости и покрытий)

На 1 балл:

1 – Независимые и покрывающие множества вершин и ребер (определения сказать)

2 – Связь чисел независимости и покрытий (теорема капсом)

3 – Оценка вершинной независимости (теорема 3 + средняя степень вершины)

4 – Теорема Кёнига

На 2 балла:

1) Говорят, что вершина **покрывает** инцидентные ей ребра, а ребро покрывает инцидентные ему вершины. **Вершинное покрытие** – множество вершин, вместе покрывающих все ребра. **Минимальное покрытие** – покрытие, никакое подмножество которого покрытием не является. **Наименьшее покрытие** – то, в котором наименьшее возможное число элементов (всякое наименьшее покрытие минимально, обратное неверно).

Число вершинного покрытия (α_0) – мощность наименьшего вершинного покрытия. Например, $\alpha_0(K_{m,n}) = \min(m, n)$ – надо выбрать все вершины наименьшей из долей полного двудольного графа. $\alpha_0(\overline{K_p}) = 0$ – для вполне несвязного графа (0 ребер, p вершин)

Число реберного покрытия (α_1) – мощность наименьшего реберного покрытия. Например, $\alpha_1(C_{2n}) = n$ – цикл четной длины, надо выбирать ребра через одно.

Независимое множество вершин/ребер – никакие два элемента не смежны. **Максимальное, наибольшее** независимое множество – по аналогии с покрытием. **Паросочетание** – то же, что и независимое множество ребер

Вершинное число независимости (β_0) – мощность наибольшего независимого множества вершин. Например, $\beta_1(K_p) = 1$ – в полном графе все вершины смежны.

Реберное число независимости (β_1) – мощность наибольшего независимого множества ребер. Например, $\beta_1(C_{2n}) = n$ – цикл четной длины, надо выбирать ребра через одно.

2) **ТЕОРЕМА О СВЯЗИ ЧИСЕЛ НЕЗАВИСИМОСТИ И ПОКРЫТИЙ:** $\beta_1 + \alpha_1 = p = \beta_0 + \alpha_0$

Доказательство – четыре неравенства, из которых следует 2 равенства. Основная идея – рассматривать 2 множества типа M_0 – наименьшее вершинное покрытие и V/M_0

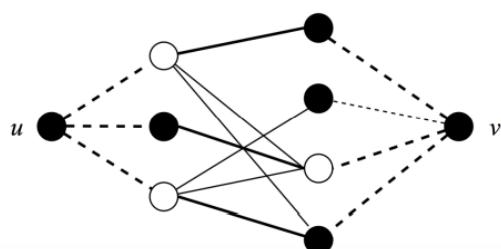
3) Теорема. Для любого графа справедливо $\beta_0 \geq \sum_{v \in V} 1/(1 + d(v))$, $d(v)$ – степень вершины

Средняя степень графа $\bar{d}(v) = \frac{1}{p} \sum_{v \in V} d(v)$

Для любого графа справедливо $\beta_0 \geq p/(1 + \bar{d})$ – доказывается через неравенство Коши-Буняковского и теорему 3

4) Теорема Кёнига. В двудольном графе $\beta_1 = \alpha_0$

Доказательство – добавить 2 вершины u, v и через теорему Менгера – минимальный разрез это и есть α_0 , а максимальное кол-во простых цепей - β_1



10.5. Переборные алгоритмы (алгоритм поиска с возвратами)

На 1 балл:

1. Постановка задачи отыскания наибольшего независимого множества вершин (эвристический алгоритм)
2. Поиск с возвратами (бэктрекинг)
3. Улучшенный перебор (хранение контекста)
4. Алгоритм построения независимых множеств

На 2 балла:

1. Методы рассматриваются на примере задачи отыскания наибольшего независимого множества вершин графа. Эта задача относится к числу трудоемких. Задачу можно поставить так. Пусть задан граф $G(V, E)$. Найти такое множество вершин

$$X, X \subset V, \text{ что } |X| = \max_{Y \in \mathcal{E}} |Y|, \text{ а } \mathcal{E} \stackrel{\text{Def}}{=} \{Y \subset V \mid \forall u, v \in Y ((u, v) \notin E)\}.$$

Можно предложить **эвристический алгоритм** - для поиска независимого множества, «блзкого» к наибольшему. Соответственно можно получить ответ, сколь угодно далекий от точного решения.

2. В силу конечности числа возможностей можно попробовать найти решение «полным перебором» всех возможных вариантов.

При решении переборных задач большое значение имеет способ организации перебора (в нашем случае — способ построения и последовательность перечисления множеств Y).

Поиск с возвратами (иначе «бэктрекинг») - способ организации перебора, основанный на идеи поиска в глубину.

Идея поиска с возвратами: Находясь в некоторой ситуации, пробуем изменить её допустимым образом в надежде найти решение. Если изменение не привело к успеху, то возвращаемся в исходную ситуацию и пробуем изменить её другим образом, и так до тех пор, пока не будут перебраны все возможности. Используется рекурсивная процедура.

Эффективный алгоритм – тот, трудоемкость (число шагов) которого ограничена полиномом от характерного размера задачи (для **неэффективных** – трудоемкость ограничена быстрорастущей функцией, экспонентой, например). Полный перебор неэффективен. Поиск с возвратами в худшем случае также неэффективен.

3. Улучшенный перебор: используя конкретную информацию о задаче, в некоторых случаях можно сократить трудоёмкость выполнения каждого шага перебора или уменьшить количество перебираемых возможностей в среднем. Рекурсия в поиске с возвратами удобна для понимания, но не эффективна. Рекурсия – для сохранения **контекста** (контекст - информация, характеризующая текущий вариант). Если использовать другие методы сохранения контекста, то можно сделать алгоритм без рекурсии => эффективнее.

4. Алгоритм построения независимых множеств (по сути, основан на улучшенном переборе) строит все максимальные независимые множества вершин заданного графа. (Пример – задача о восьми ферзях является задачей об отыскании максимальных независимых множеств).

10.7 - Раскраска графов (хроматические полиномы)

На 1 балл:

План:

10.7.1. Хроматическое число графа и его оценки

раскраска графа, $\chi(G)$ — хроматическое число, одноцветный класс

10.7.2. Хроматические числа графа и его дополнения

10.7.3. Точный алгоритм раскрашивания

10.7.4. Приближенные алгоритмы раскрашивания

алгоритм последовательного раскрашивания, улучшенный алгоритм последовательного раскрашивания

10.7.5. Хроматический полином

На 2 балла:

Ключевые понятия

10.7.1. Раскраской графа G называется такое приписывание цветов (натуральных чисел) его вершинам, что никакие две смежные вершины не получают одинаковый цвет. Если задана допустимая раскраска графа, использующая m цветов, то говорят, что граф **m-раскрашиваемый**. Наименьшее возможное количество цветов в раскраске называется **хроматическим числом** и обозначается $\chi(G)$. Очевидно, что существует m-раскраска графа G для любого m в диапазоне $\chi(G) \leq m \leq p$ (p — число вершин). Множество вершин, покрашенных в один цвет, называется **одноцветным классом**. Одноцветные классы образуют независимые множества вершин, то есть никакие две вершины в одноцветном классе не смежны. Способ явного выражения хроматического числа через другие стандартные инварианты графа неизвестен. Известны только некоторые оценки: **Теорема** $\chi(G) \leq 1 + \Delta(G)$ ($\Delta(G)$ — длина максимального простого цикла графа G) **Теорема** $p/\beta_0(G) \leq \chi(G) \leq p - \beta_0(G) + 1$ ($\beta_0(G)$ — вершинное число независимости (мощность наибольшего независимого множества вершин)).

10.7.2. Хроматические числа графа G и его дополнения \bar{G} связаны: если в G сравнительно мало рёбер, то и $\chi(G)$ будет невелико, но тогда в \bar{G} много рёбер и $\chi(\bar{G})$ будет близко к p .

Теорема Пусть $\chi := \chi(G)$, $\bar{\chi} := \chi(\bar{G})$. Тогда $2\sqrt{p} \leq \chi + \bar{\chi} \leq p + 1$, $p \leq \chi\bar{\chi} \leq \left(\frac{p+1}{2}\right)^2$.

10.7.3. Точный алгоритм раскрашивания: рассмотрим следующую схему рекурсивной процедуры P:

1. Выбрать в графе G некоторое максимальное независимое множество вершин S; 2. Покрасить вершины множества S в очередной цвет; 3. Применить процедуру P к графу G-S.

Теорема Если график G - k-раскрашиваемый, то существует такая последовательность выборов множества S на шаге 1 процедуры P, что применение процедуры P к графу G построит не более чем k-раскраску графа G.

10.7.4. Задача построения минимальной раскраски имеет переборный характер и является NP-полной, поэтому на практике целесообразно рассматривать приближённые алгоритмы, которые не всегда находят точное решение задачи, но зато достаточно эффективны. **Алгоритм последовательного раскрашивания.** Идея состоит в том, чтобы красить вершины последовательно, выбирая среди допустимых цвет с наименьшим номером. **Улучшенный алгоритм последовательного раскрашивания** строит допустимую раскраску, применяя такую эвристику: начинать раскрашивать следует с вершин наибольшей степени, поскольку если их раскрашивать в конце процесса, то более вероятно, что для них не найдётся свободного цвета и придётся использовать ещё один (основной цикл идет не по вершинам, а по цветам).

10.7.5. Раскраски (допустимые) нетривиальных графов не являются единственными. Пусть $P(G,n)$ - число раскрасок графа G в n цветов. **Лемма 1** $\forall G (\max \{n \in 0..p \mid P(G,n)\} = P(G,p) = p!)$ **Лемма 2**

$P(G_1 \cup G_2, n) = P(G_1, n)P(G_2, n)$ Очевидно, что $\chi(G) = \min \{n \in 0..p \mid P(G,n) > 0\}$, поэтому, зная числа $P(G,n)$, нетрудно найти хроматическое число $\chi(G)$, обратное же неверно. Как найти компактное описание множества $\{n \in 0..p \mid P(G,n)\}$? Пусть дан (p, q) — график G. Полином $P_G(x)$ такой, что $\forall n \in 0..p (P_G(n) = P(G,n))$ называется **хроматическим полиномом** графа G. В общем случае метод построения хроматического полинома для произвольного графа неизвестен.

Теорема (редукции) Пусть $e = (u, v)$ — любое ребро в графике G. Тогда $P_G(x) = P_{G-e}(x) - P_{G/e}(x)$

Следствие Если ребро $e = (u, v)$ не входит в график G, то $P_G(x) = P_{G+e}(x) + P_{G/e}(x)$.

10.8 Плановость

На 1 балл:

План:

1. Укладка графов;
2. Эйлерова характеристика (теорема Эйлера);
3. Теорема о пяти красках.

Ключевые термины и обозначения:

1. Планарный граф, плоский граф, $f(G)$ —число граней графа;
2. Операция подразбиения ребра, гомеоморфизм графов, теорема Куратовского.
- 3.

На 2 балла:

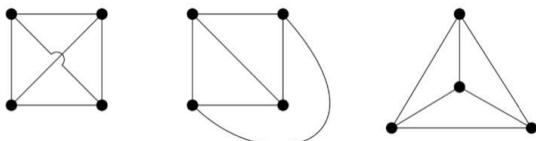
Укладка графов

Граф **укладывается** на некоторой поверхности, если его можно нарисовать на этой поверхности так, чтобы рёбра графа при этом не пересекались. Граф называется **планарным**, если его можно уложить на плоскости. **Плоский график** — это график, уже уложенный на плоскости.

Область, ограниченная ребрами в плоском графике, называется **гранью**. Грань не содержит других граней. **Число граней** плоского графа G обозначается $f(G)$.

ЗАМЕЧАНИЕ: Внешняя часть плоскости также образует грань.

ПРИМЕР: На рисунке показаны диаграммы планарного графа K_4 и две его укладки на плоскости. Этот график имеет 4 грани.



Эйлерова характеристика

Для графов, уложенных на некоторой поверхности, справедливо определённое соотношение между числом вершин, рёбер и граней графов, которые укладываются на этой поверхности.

ТЕОРЕМА (формула Эйлера): Для связного планарного графа справедливо следующее соотношение: $p - q + f = 2$ (p -вершины, q -ребра, f -грани).

ЗАМЕЧАНИЕ: Число в правой части этого соотношения называется **эйлеровой характеристикой** поверхности.

СЛЕДСВИЕ 1: Если G — связный планарный график ($p > 3$), то $q \leq 3p - 6$.

СЛЕДСВИЕ 2: Графы K_5 и $K_{3,3}$ не планарны.

ЗАМЕЧАНИЕ: Операция **подразбиения** ребра $x = (u, v)$ состоит в замене его двумя рёбрами, (u, w) и (w, v) , где w — новая вершина. Два графа называются **гомеоморфными**, если они могут быть получены из одного графа подразбиением рёбер.

Теорема Куратовского: Граф планарен тогда и только тогда, когда он не содержит подграфов, гомеоморфных K_5 или $K_{3,3}$.

СЛЕДСВИЕ 3: В любом планарном графике существует вершина, степень которой не больше 5.

Теорема о пяти красках

ТЕОРЕМА: Всякий планарный график можно раскрасить пятью красками.

ДОКАЗАТЕЛЬСТВО: индукцией по числу p вершин графа.

17. ** Нормальные формы 3.4.1.-3.4.7.

План:

1. разложение булевых функций по переменным: Мажоритарная функция
2. Минимальные термы: Минтерм, ортогональная и выполнимая система булевых функций, макстерм.
3. Совершенные нормальные формы: Разрешимость, СДНФ, СКНФ
4. Эквивалентные преобразования: Правило склеивание\расщепление, элиминация операций, протаскивание отрицаний, дизъюнктивная форма, удаление нулей, расщепление переменных, системы подстановок термов, системы правил переписывания
5. Минимальные дизъюнктивные формы: Ранг конъюнкций
6. Геометрическая интерпретация
7. Сокращенные дизъюнктивные формы: Допустимая конъюнкция, максимальная конъюнкция.

(Т) О разложении булевой функции по

$$f(x_1, \dots, x_m, x_{m+1}, \dots, x_n) = \bigvee_{(\sigma_1, \dots, \sigma_m)} x_1^{\sigma_1} \wedge \dots \wedge x_m^{\sigma_m} \wedge f(\sigma_1, \dots, \sigma_m, x_{m+1}, \dots, x_n).$$

переменным.

, где дизъюнкция берется по всем возможным набором $(\sigma_1, \dots, \sigma_m)$

Док-во: Рассмотрим значение формулы в правой части на наборе значений a_1, \dots, a_n .

$$\text{Имеем } \left(\bigvee_{(\sigma_1, \dots, \sigma_m)} x_1^{\sigma_1} \wedge \dots \wedge x_m^{\sigma_m} \wedge f(\sigma_1, \dots, \sigma_m, x_{m+1}, \dots, x_n) \right) (a_1, \dots, a_n) = \boxed{\bigvee_{(\sigma_1, \dots, \sigma_m)} a_1^{\sigma_1} \wedge \dots \wedge a_m^{\sigma_m} \wedge f(\sigma_1, \dots, \sigma_m, a_{m+1}, \dots, a_n)}.$$

Все конъюнкции, в которых $\exists i (a_i \neq \sigma_i)$, равны 0, и их можно опустить, поэтому остается только одно слагаемое, для которого $\forall i \in 1..m (a_i = \sigma_i)$, значит

$$a_1^{a_1} \wedge \dots \wedge a_m^{a_m} \wedge f(a_1, \dots, a_m, a_{m+1}, \dots, a_n) = f(a_1, \dots, a_n).$$

(С) Разложение булевой функции по одной переменной

$$f(x_1, \dots, x_{n-1}, x_n) = (x_n \wedge f(x_1, \dots, x_{n-1}, 1)) \vee (\bar{x}_n \wedge f(x_1, \dots, x_{n-1}, 0))$$

(С) Разложение булевой функции по всем переменным

$$f(x_1, \dots, x_n) = \bigvee_{\{(\sigma_1, \dots, \sigma_n) | f(\sigma_1, \dots, \sigma_n) = 1\}} x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n}$$

- Выражение вида $x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n}$ носит специальное название: **минимальный терм**, или **совершенный одночлен**, или **конституента единицы**, или, наиболее коротко, **минтерм**. **Минтерм** — это реализация булевой функции n переменных, которая имеет значение 1 ровно на одном наборе значений переменных, а именно на наборе $(\sigma_1, \dots, \sigma_n)$.

(С) Любую булеву функцию, кроме 0, можно представить как дизъюнкцию некоторых минтермов

- Говорят, что система булевых функций **ортогональна**, если их конъюнкция есть тождественный нуль. Пример. Система $\{x, \bar{x}\}$ ортогональна.

(Л) Любое множество различных мinterмов ортогонально.

Док-во. Не существует такого набора значений, на котором два различных мinterма принимают значение 1. Говорят, что система булевых функций **выполнима**, если их дизъюнкция не есть тождественный нуль. Пример. Система $\{\underline{x}, \bar{x}\}$ выполнима.

(Л) Любое множество мinterмов выполнимо.

Док-во. Дизъюнкция выполняется на всех наборах, на которых какой-либо из мinterмов имеет значение 1.

- Двойственным к мinterму является **макстерьм** — булева функция, которая принимает значение 0 на единственном наборе значений переменных
- Совершенной дизъюнктивной нормальной формой (СДНФ) называется реализация булевой функции $f(x_1, \dots, x_n)$ в виде формулы $\bigvee_{\{(\sigma_1, \dots, \sigma_n) | f(\sigma_1, \dots, \sigma_n) = 1\}} x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n}$

(Т) Всякая булева функция имеет единственную СДНФ

Док-во: По следствию к теореме о разложении булевой функции по переменным имеем $f(x_1, \dots, x_n) = \bigvee_{\{(\sigma_1, \dots, \sigma_n) | f(\sigma_1, \dots, \sigma_n) = 1\}} x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n}$. Если при записи СДНФ используется установленный порядок, то СДНФ однозначно определяет множество наборов значений переменных, на которых функция, реализуемая СДНФ, принимает значение 1. Тем самым СДНФ однозначно определяет таблицу истинности реализуемой функции, откуда любые две различные СДНФ над одним набором переменных неравносильны. Пример: . СДНФ конъюнкции: $x \wedge y = x^1 \wedge y^1$

(С) Всякая булева функция может быть выражена через дизъюнкцию, конъюнкцию и отрицание $\forall f \in P_n (\exists \mathcal{F} \in \mathfrak{F}[\vee, \wedge, \neg] (f = \text{func } \mathcal{F}))$

(Т) Всякая булева функция имеет единственную совершенную конъюнктивную нормальную

форму (СКНФ): $f(x_1, \dots, x_n) = \bigwedge_{\{(\sigma_1, \dots, \sigma_n) | f^*(\sigma_1, \dots, \sigma_n) = 1\}} x_1^{\sigma_1} \vee \dots \vee x_n^{\sigma_n}$

Док-во. По принципу двойственности из предыдущей теоремы.

- **Эквивалентное преобразование** - преобразование формулы в равносильную ей

(Т) Для любых двух равносильных формул F1 и F2 существует последовательность эквивалентных преобразований из F1 в F2, получаемая посредством равносильностей

■ ■ ■

13. ** Решетки и булевы алгебры 2.6.1.-2.6.6. (Кирилл Желудев)

На 1 балл (план):

- 2.6.1. Дистрибутивные и ограниченные решетки: принцип двойственности, нижняя и верхняя грани решетки
- 2.6.2. Решетки с дополнением
- 2.6.3. Частичный порядок в решётке: нижняя и верхняя границы, нижняя и верхняя грани
- 2.6.4. Полные решетки
- 2.6.5. Полурешетки
- 2.6.6. Булева алгебра

Основные понятия, теоремы и доказательства:

(О) Решетка

Решётка – это множество M с двумя бинарными операциями, \cap и \cup , такими, что выполнены следующие условия (аксиомы решётки):

- 1) идемпотентность: $a \cap a = a, a \cup a = a;$
- 2) коммутативность: $a \cap b = b \cap a, a \cup b = b \cup a;$
- 3) ассоциативность: $(a \cap b) \cap c = a \cap (b \cap c), (a \cup b) \cup c = a \cup (b \cup c);$
- 4) поглощение: $(a \cap b) \cup a = a, (a \cup b) \cap a = a.$

(О) Принцип двойственности

Замечание. В каждой паре аксиом решётки одна формула пары получается из другой подстановкой $\cup \leftrightarrow \cap$, то есть аксиомы в каждой паре **двойственны** друг другу. Отсюда следует, что любое утверждение, истинное для произвольных элементов решётки, остаётся истинным при подстановке $\cup \leftrightarrow \cap$. Этот полезный факт называется **принципом двойственности** для решёток.

(О) Дистрибутивная решетка

Решётка называется **дистрибутивной**, если

$$a \cup (b \cap c) = (a \cup b) \cap (a \cup c),$$

$$a \cap (b \cup c) = (a \cap b) \cup (a \cap c).$$

(О) Нижняя и верхняя грани решетки, ограниченная решетка

Если в решётке $\exists 0 \in M$ ($\forall a (0 \cap a = 0)$), то 0 называется **нулем** (или **нижней границей**) решётки. Если в решётке $\exists 1 \in M$ ($\forall a (1 \cup a = 1)$), то 1 называется **единицей** (или **верхней границей**) решётки. Решётка с верхней и нижней границами называется **ограниченной**.

(Т) Единственность грани решетки

Теорема. Если нижняя (верхняя) грань существует, то она единственна.

Доказательство. Пусть $0'$ – ещё один нуль решётки. Тогда $0' = 0 \cap 0' = 0' \cap 0 = 0$. □

(Т) О связи операций решетки

Теорема. $a \cap b = b \iff a \cup b = a$.

Доказательство.

$$[\implies] a \cup b = a \cup (a \cap b) = (a \cap b) \cup a = a.$$

$$[\Leftarrow] a \cap b = (a \cup b) \cap b = (b \cup a) \cap b = b.$$

□

(C) О связи операций с гранями

Следствие. $0 \cup a = 1 \cap a = a$.

(O) Дополнение элемента; решетка с дополнением

В ограниченной решётке элемент a' называется **дополнением** элемента a , если $a \cap a' = 0$ и $a \cup a' = 1$. Если дополнение существует у любого элемента, то ограниченная решётка называется **решёткой с дополнением**. Вообще говоря, дополнение не обязано существовать и не обязано быть единственным.

(T) О свойствах дополнения

Теорема. [о свойствах дополнения] В ограниченной дистрибутивной решётке с дополнением выполняется следующее:

- 1) дополнение a' единствено;
- 2) дополнение инволютивно: $a'' = a$;
- 3) грани дополняют друг друга: $1' = 0$, $0' = 1$;
- 4) законы де Моргана: $(a \cap b)' = a' \cup b'$, $(a \cup b)' = a' \cap b'$.

Доказательство.

$$\begin{aligned} [1] \text{ } & \text{Пусть } x, y - \text{ дополнения } a. \text{ Тогда } x = x \cap 1 = x \cap (a \cup y) = \\ & = (x \cap a) \cup (x \cap y) = 0 \cup (x \cap y) = x \cap y = y \cap x = 0 \cup (y \cap x) = \\ & = (y \cap a) \cup (y \cap x) = y \cap (a \cup x) = y \cap 1 = y; \end{aligned}$$

$$[2] \text{ } (a \cup a' = 1 \implies a' \cup a = 1, a \cap a' = 0 \implies a' \cap a = 0) \implies a = a'';$$

$$[3] \text{ } (1 \cap 0 = 0, 0' \cap 0 = 0) \implies 1 = 0', (1 \cup 0 = 1, 1 \cup 1' = 1) \implies 0 = 1';$$

$$[4] \text{ } (a \cap b) \cap (a' \cup b') = (a \cap b \cap a') \cup (a \cap b \cap b') =$$

$$= (0 \cap b) \cup (a \cap 0) = 0 \cup 0 = 0,$$

$$(a \cap b) \cup (a' \cup b') = (a \cup a' \cup b') \cap (b \cup a' \cup b') =$$

$$= (1 \cup b') \cap (1 \cup a') = 1 \cap 1 = 1,$$

$$(a \cup b) \cap (a' \cap b') = (a \cap a' \cap b') \cup (b \cap a' \cap b') =$$

$$= (0 \cap b') \cup (a' \cap 0) = 0 \cup 0 = 0,$$

$$(a \cup b) \cup (a' \cap b') = (a \cup b \cup a') \cap (a \cup b \cup b') =$$

$$= (1 \cup b) \cap (1 \cup a) = 1 \cap 1 = 1.$$

□

(O) Частичный порядок в решётке

В любой решётке можно естественным образом ввести частичный порядок, а именно: $a \prec b \stackrel{\text{Def}}{=} a \cap b = a$.

(T) Отношение “птичка” - отношение частичного порядка

Теорема. Отношение \prec является отношением частичного порядка.

Доказательство.

[Рефлексивность] $a \cap a = a \implies a \prec a$.

[Антисимметричность] $a \prec b \& b \prec a \implies a = a \cap b = b \cap a = b$.

[Транзитивность] $a \prec b \& b \prec c \implies a \cap c = (a \cap b) \cap c = a \cap (b \cap c) = a \cap b = a \implies a \prec c$.

□

(О) Нижняя и верхняя границы

Пусть M – частично упорядоченное множество с частичным порядком \prec . Напомним, (см. [п. 1.8.5](#)), что элемент x называется **нижней границей** для a и b , если $x \prec a \& x \prec b$. Аналогично, y называется **верхней границей** для a и b , если $a \prec y \& b \prec y$.

Здесь $a, b, x, y \in M$.

(О) Нижняя и верхняя грани

Элемент x называется **нижней гранью** элементов a и b , если x – нижняя граница элементов a и b и для любой другой нижней границы v элементов a и b выполняется $v \prec x$. Обозначение: $x = \inf(a, b)$.

Аналогично, y называется **верхней гранью** элементов a и b , если y – верхняя граница элементов a и b и для любой другой верхней границы u элементов a и b выполняется $y \prec u$. Обозначение: $y = \sup(a, b)$.

(Л) Единственность грани

Лемма. Если нижняя (верхняя) грань любых двух элементов существует, то она единственна.

Доказательство. $x = \inf(a, b) \& y = \inf(a, b) \implies y \prec x \& x \prec y \implies x = y$.

□

(Т) Наличие нижней и верхней грани для любых двух элементов ЧУМ \rightarrow решетка относительно \sup и \inf

Теорема. Если в частично упорядоченном множестве для любых двух элементов существуют нижняя и верхняя грани, то это множество образует решётку относительно \inf и \sup .

Доказательство. Пусть $x \cap y := \inf(x, y)$, $x \cup y := \sup(x, y)$.

[1] $\inf(x, x) = x \implies x \cap x = x$, $\sup(x, x) = x \implies x \cup x = x$.

[2] $\inf(x, y) = \inf(y, x) \implies x \cap y = y \cap x$, $\sup(x, y) = \sup(y, x) \implies x \cup y = y \cup x$.

[3] $\inf(x, \inf(y, z)) = \inf(\inf(x, y), z) \implies x \cap (y \cap z) = (x \cap y) \cap z$,

$\sup(x, \sup(y, z)) = \sup(\sup(x, y), z) \implies x \cup (y \cup z) = (x \cup y) \cup z$.

[4] $\sup(\inf(a, b), a) = a \implies (a \cap b) \cup a = a$, $\inf(\sup(a, b), a) = a \implies (a \cup b) \cap a = a$.

□

(О) Полная решетка

Частичный порядок можно естественным образом ввести в любой решётке, но при этом не всякое подмножество носителя решётки обязано иметь верхнюю и нижнюю грани. Решётка называется **полной**, если любое подмножество имеет супремум и инфимум. Полная решётка L является ограниченной, если положить: $0 := \inf L$, $1 := \sup L$. Кроме того, ясно, что полная решётка является линейно полным частично упорядоченным множеством.

(Т) Полная решётка + монотонная функция -> наличие у f наименьшей неподвижной точки

Теорема. Если L – полная решётка, а функция $f: L \rightarrow L$ монотонна, то функция f имеет наименьшую неподвижную точку.

Доказательство. Рассмотрим множество $A := \{x \in L \mid f(x) \prec x\}$ и элемент $a = \inf A$. Имеем $\forall x \in A$ ($a \prec x$), и значит, $\forall x \in A$ ($f(a) \prec f(x) \prec x$), следовательно, $f(a)$ – нижняя граница для множества A и $f(a) \prec a$. Далее, $f(f(a)) \prec f(a)$, и значит, $f(a) \in A$, откуда по определению элемента a имеем $a \prec f(a)$ и по антисимметричности $f(a) = a$. Таким образом, a – неподвижная точка функции f . Пусть теперь b – любая неподвижная точка функции f , то есть $f(b) = b$. По рефлексивности естественного частичного порядка $f(b) \prec b$. Тем самым $b \in A$ и значит $a \prec b$. \square

(О) Полурешетка

Множество M с операцией $\cap: M \times M \rightarrow M$ называется **полурешёткой**, если выполнены следующие условия (аксиомы полурешётки):

- 1) идемпотентность: $a \cap a = a$;
- 2) коммутативность: $a \cap b = b \cap a$;
- 3) ассоциативность: $a \cap (b \cap c) = (a \cap b) \cap c$.

(О) Дистрибутивная функция на полурешётке

Функция $f: M \rightarrow M$, определённая на полурешётке M , называется **дистрибутивной**, если $f(x \cap y) = f(x) \cap f(y)$.

(О) Ограниченнная полурешётка

Полурешётка называется **ограниченной**, если у неё существуют верхняя и нижняя грань.

Поскольку в полурешётке определён естественный частичный порядок, определены и все другие понятия:

- ▶ верхние и нижние границы,
- ▶ верхние и нижние грани (**супремум и инфимум**),
- ▶ максимальные и минимальные элементы,
- ▶ наибольшие и наименьшие элементы,
- ▶ линейная полнота,
- ▶ конечная высота.

(Т) Полурешётка с некоторыми свойствами, монотонная функция -> наличие у функции наименьшей неподвижной функции

Теорема. Если L – ограниченная полурешётка конечной высоты, а $f: L \rightarrow L$ – монотонная функция, то функция f имеет наименьшую неподвижную точку, которая может быть получена методом итераций из нижней грани.

Доказательство. В ограниченной полурешётке нижняя грань 0 является наименьшим элементом. Поскольку полурешётка имеет конечную высоту, всякое линейно упорядоченное подмножество можно рассматривать как монотонную последовательность, которая заканчивается наибольшим элементом (супремумом).

В частности, последовательность $\langle f^0(0), f^1(0), f^2(0), \dots \rangle$ монотонна в силу монотонности функции f и имеет супремум $a = \sup \{f^n(0) \mid n \geq 0\}$, который является наименьшей неподвижной точкой (см. доказательство теоремы о наименьшей неподвижной точке в [п. 1.8.7](#)). \square

(О) Булева алгебра и его свойства

Дистрибутивная ограниченная решётка, в которой для каждого элемента существует дополнение, называется **булевой алгеброй**. Свойства булевой алгебры:

- 1) $a \cup a = a, a \cap a = a$ – по определению решётки;
- 2) $a \cup b = b \cup a, a \cap b = b \cap a$ – по определению решётки;
- 3) $a \cup (b \cup c) = (a \cup b) \cup c, a \cap (b \cap c) = (a \cap b) \cap c$ – по определению решётки;
- 4) $(a \cap b) \cup a = a, (a \cup b) \cap a = a$ – по определению решётки;
- 5) $a \cup (b \cap c) = (a \cup b) \cap (a \cup c), a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$ – по дистрибутивности;
- 6) $a \cup 1 = 1, a \cap 0 = 0$ – по ограниченности;
- 7) $a \cup 0 = a, a \cap 1 = a$ – по следствию из теоремы 2 [п. 2.6.1](#);
- 8) $a'' = a$ – по теореме [п. 2.6.2](#);
- 9) $(a \cap b)' = a' \cup b', (a \cup b)' = a' \cap b'$ по теореме [п. 2.6.2](#);
- 10) $a \cup a' = 1, a \cap a' = 0$ – по определению.