Содержание

7. Комбинаторика

Базовые понятия:

- **Алфавит** (Alphabet) Σ (или X, *Ex.* $X = \{a, b, c\}$) - множество символов в нашей системе

- **Диапазон** (Range) $[n] = \{1, \ldots, n\}$ - конечное множество последовательных натуральных чисел

- **Расстановка** (Ordered arrangement) - последовательность каких-либо элементов (тоже самое, что кортеж), *Ex.* $x = (a, b, c, d, b, b, c) \quad |x| = n$

  Расстановку можно представить как функцию $f : \underbrace{[n]}_{\text{domain}} \to \underbrace{\Sigma}_{\text{codomain}}$, которая по порядковому номеру выдает символ

  $ranf = \{c \in \Sigma \mid \exists i \in [n] \ : \ f(i) = c\}$

- **Перестановка** (Permutation) - $\pi : [n] \to \Sigma$, где $n = |\Sigma|$

  Расстановка π - биекция между $[n]$ и Σ

  Ex. $\pi = 2713546$

i	1	2	3	4	5	6	7
$\pi(i)$	2	7	1	3	5	4	6

  Одна из задач комбинаторики - посчитать количество различных расстановок или перестановок при заданных n и Σ

- **k-перестановка** (k-permutation) - расстановка из k различных элементов из Σ

  Ex. $\underbrace{|31475|}_{\text{5-perm из } \Sigma=[7]} = 5$

  k-перестановка - инъекция $\pi : [k] \to \Sigma$ $(k \leq n = |\Sigma|)$

- $P(n, k)$ - множество всех k-перестановок алфавита $\Sigma = [n]$ (если исходный алфавит не состоит из чисел, то мы можем сделать биекцию между ним и $[n]$)

  $P(n, k) = \{f \mid f : [k] \to [n]\}$

  Чаще интересует не само множество, а его размер, поэтому под обозначением $P(n, k)$ подразумевается $|P(n, k)|$

- $S_n = P_n = P(n, n)$ - множество всех перестановок. Также чаще всего нас будет интересовать не множество, а его размер

  $|S_n| = n!$ - всего существует $n!$ перестановок

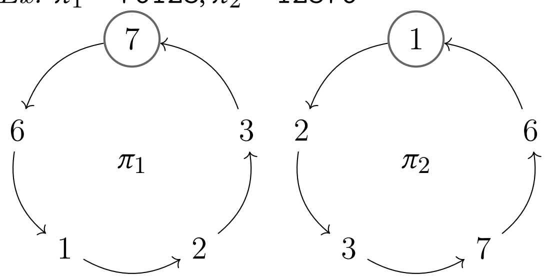
$|P(n,k)| = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-k+1) = \dfrac{n!}{(n-k)!}$

- **Циклические k-перестановки** (Circular k-permutations)

  $\pi_1, \pi_2 \in P(n,k)$ - циклически эквивалентны тогда и только тогда:

  $\exists s \mid \forall i \ \pi_1((i+s)\%k) = \pi_2(i)$

  Ex. $\pi_1 = 76123, \pi_2 = 12376$

  $P_C(n,k)$ - множество всех циклических k-перестановок в Σ

  $|P_C(n,k)| \cdot k = |P(n,k)|$

  $|P_C(n,k)| = \dfrac{|P(n,k)|}{k} = \dfrac{n!}{k(n-k)!}$

- **Неупорядоченная расстановка k элементов** (Unordered arrangement of k elements) - мультимножество Σ^* размера k

  Ex. $\Sigma^* = \{\triangle, \triangle, \square, \triangle, \circ, \square\}^* = \{3 \cdot \triangle, 2 \cdot \square, 1 \cdot \circ\} = (\Sigma, r)$

  Неупорядоченную расстановку можно представить как функцию:

  $r : \Sigma \longrightarrow \mathbb{N}, \quad r(x)$ - кол-во повторений объекта x

- **k-сочетание** (k-combination) - неупорядоченная перестановка из k различных элементов из Σ (еще называют k-подмножеством, k-subset)

  Соответственно $C(n,k)$ - множество всех таких k-сочетаний

  $|C(n,k)| = C_n^k = \dbinom{n}{k}$

  $C(n,k) = \dbinom{\Sigma}{k}$

  $\dbinom{n}{k} \cdot k! = |P(n,k)|$

  $|C(n,k)| = \dbinom{n}{k} = \dfrac{n!}{k!(n-k)!}$

Th. Биномиальная теорема (Binomial theorem):

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$$

$\binom{n}{k}$ - биномиальный коэффициент

Th. Мультиномиальная теорема (Multinomial theorem)

$$(x_1 + \dots + x_r)^n = \sum_{\substack{k_i \in 1..n, \\ k_1 + \dots + k_r = n}} \binom{n}{k_1, \dots, k_r} x_1^{k_1} \cdot \dots \cdot x_r^{k_r}$$

$\binom{n}{k_1, \dots, k_r} = \dfrac{n!}{k_1! \dots k_r!}$ - мультиномиальный коэффициент

Ex. мультиномиальной теоремы:
$(x+y+z)^4 = 1(x^4+y^4+z^4) + 4(xy^3+xz^3+x^3y+yz^3+y^3z+yz^3) + 6(x^2y^2+y^2z^2+x^2z^2) + 12(xyz^2 + xy^2z + x^2yz)$
Доказательство:

□

$(x_1 + \dots + x_r)^n = \sum\limits_{\substack{i_j \in [r] \\ j \in [n]}} x_{i_1}^1 \cdot \dots \cdot x_{i_n}^1 = \sum\limits_{\substack{i_j \in [r] \\ j \in [n]}} x_1^{k_1} \cdot \dots \cdot x_r^{k_r}$, где k_t - количество x с индексом t в одночлене $(k_t = |\{j \in [n] | i_j = t\}|)$

Получается мультиномиальный коэффиццnент $\binom{n}{k_1, \dots, k_r}$ будет равен количество способов поставить k_1 единиц в индексы в $x_{i_1}^1 \cdot \dots \cdot x_{i_n}^1$, k_2 двоек в индексы и так далее

У нас есть $\binom{n}{k_1}$ способов поставить единицу в индексы в одночлен, $\binom{n-k_1}{k_2}$ способов поставить двойку и т. д., получаем:

$$\binom{n}{k_1, \dots, k_r} = \binom{n}{k_1}\binom{n-k_1}{k_2} \dots \binom{n-k_1-\dots-k_{r-1}}{k_r} = [n-k_1-\dots-k_r = 0] =$$

$$\frac{n!}{k_1!(n-k_1)!} \frac{(n-k_1)!}{k_2!(n-k_1-k_2)!} \dots \frac{(n-k_1-\dots-k_{r-1})!}{k_r!0!} = \frac{n!}{k_1! \dots k_r!}$$

□

- **Перестановка мультимножества** Σ^* (Permutations of a multiset Σ^*)
  $\Sigma^* = \{\triangle^1, \triangle^2, \square, \star\} = (\Sigma, r) \quad r : \Sigma \to \mathbb{N}_0 \quad n = |\Sigma^*| = 4 \quad s = |\Sigma| = 3$

  Nota. $\begin{cases} \triangle^1, \triangle^2, \square, \star \\ \triangle^2, \triangle^1, \square, \star \end{cases}$ считаются равными перестановками

  $|P^*(\Sigma^*, n)| = \dfrac{n!}{r_1! \dots r_s!} = \binom{n}{r_1, \dots, r_s}$ - количество перестановок мультимножества, где r_i -

количество i-ого элемента в мультимножестве

- **k-комбинация бесконечного мультимножества** (k-combinations of infinite multiset) - такое субмультимножество размера k, содержащее элементы из исходного мультимножества. При этом соблюдается, что количество какого-либо элемента r_i в исходном мультимножестве не больше размера комбинации k
  $\Sigma^* = \{\infty \cdot \triangle, \infty \cdot \square, \infty \cdot \star, \infty \cdot \text{🐈}\}^* \quad n = |\Sigma^*| = \infty$
  $\Sigma = \{\triangle, \square, \star, \text{🐈}\} \quad s = |\Sigma| = 4$

  Ex. 5-комбинация: $\{\triangle, \star, \square, \star, \square\}$
  Разделяем на группы по Σ палочками:
  $\triangle\Big|\square\square\Big|\star\star\Big|$
  Заменяем элементы на точечки - нам уже не так важен тип элемента, потому что мы знаем из разделения:
  $\bullet\Big|\bullet\bullet\Big|\bullet\bullet\Big|$
  (другой *Ex.* $\bullet\bullet\bullet\bullet\Big|\Big|\Big|\bullet = \{4 \cdot \triangle, 1 \cdot \text{🐈}\}$)
  Получается всего k точечек и $s-1$ палочек, всего $k+s-1$ объектов. Получаем мультимножество $\{k \cdot \bullet, (s-1) \cdot \Big|\}$ (*Star and Bars method*)

  Получаем количество перестановок этого мультимножества: $\dfrac{(k+s-1)!}{k!(s-1)!} = \dbinom{k+s-1}{k, s-1} =$

  $\dbinom{k+s-1}{k} = \dbinom{k+s-1}{s-1}$
  что и является количеством возможных k-комбинаций бесконечного мультимножества

- **Слабая композиция** (Weak composition) неотрицательного целого числа n в k частей - это решение $(b_1, \dots, b_k)$ уравнение $b_1 + \dots + b_k = n$, где $b_i \geq 0$
  $|\{\text{слабая композиция } n \text{ в } k \text{ частей}\}| = \dbinom{n+k-1}{n, k-1}$
  Для решения воспользуемся аналогичным из доказательства мультиномиальной теоремы приемом:
  $n = 1 + 1 + 1 + \dots + 1$
  Поставим палочки:
  $n = 1 + 1\Big|1\Big|\dots + 1$
  Получаем задачу поиска количеств k-комбинаций в мультимножестве: $\{n \cdot 1, (k-1) \cdot \Big|\}$;
  получаем $\dbinom{n+k-1}{n, k-1}$

- **Композиция** (Composition) - решение для $b_1 + \dots + b_k = n$, где $b_i > 0$

$|\{\text{композиция } n \text{ в } k \text{ частей}\}| = \binom{n-k+k-1}{n-k, k-1}$
Мы знаем, что одну единичку получит каждая b_i, поэтому мы решаем это как слабую композицию для $n-k$ в k частей

- **Число композиций n в некоторой число частей** (Number of all compositions into some number of positive parts)
$\sum_{k=1}^{n} \binom{n-1}{k-1} = 2^{n-1}$
Пусть $t = k-1$, тогда $\sum_{t=0}^{n-1} \binom{n-1}{t} = 2^{n-1}$

- **Разбиения множества** (Set partitions) - множество размера k непересекающихся непустых подмножеств
Ex. $\{1,2,3,4\}, n=4, k=2 \longrightarrow$ [разбиение в 2 части] $\longrightarrow$ $\{\{1\},\{2,3,4\}\}$,
$\{\{1,2\},\{3,4\}\}$,
$\{\{1,2,3\},\{4\}\}$,
$\{\{1,3\},\{2,4\}\}$,
$\{\{1,4\},\{2,3\}\}$,
$\{\{2\},\{1,3,4\}\}$,
$\{\{3\},\{1,2,4\}\}$
$|\{\text{разбиение } n \text{ элементов в } k \text{ частей}\}| = \left\{ n \atop k \right\} = S_k^{II}(n) = S(n,k)$ - число Стирлинга второго рода
Для примера выше число Стирлинга $S(4,2) = \left\{ 4 \atop 2 \right\} = 7$
Согласно Википедии для формулы Стирлинга есть формула: $S(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k+j}\binom{k}{j} j^n$

- **Формула Паскаля** (Pascal's formula)
$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

- **Рекуррентное отношение для чисел Стирлинга** (Recurrence relation for Stirling$^{(2)}$ number):
$\left\{ n \atop k \right\} = \left\{ n-1 \atop k-1 \right\} + k \cdot \left\{ n-1 \atop k \right\}$
Возьмем какое-либо разбиение для $n-1$ элементов на k частей, тогда возможны два случая:
1) В k-ое множество нет ни одного элемента, тогда мы обязаны в него положить наш

n-ый элемент по определению, количество перестановок будет равно $\left\{ {n-1 \atop k-1} \right\} \cdot 1$

2) В k-ом множестве уже есть элементы, тогда все множества будут заполнены и у нас будет выбор из k множеств, куда положить k-ый элемент, то есть $k \cdot \left\{ {n-1 \atop k} \right\}$

Эти два случая независимы, поэтому получаем $\left\{ {n-1 \atop k-1} \right\} + k \cdot \left\{ {n-1 \atop k} \right\}$

- **Число Белла** (Bell number) - количество всех неупорядоченных разбиений множества размера n

  Число Белла вычисляется по формуле: $B_n = \sum\limits_{m=0}^{n} S(n,m)$

- **Целочисленное разбиение** (Integer partition) - решение для $a_1 + \cdots + a_k = n$, где $a_1 \geq a_2 \geq \cdots \geq a_k \geq 1$

  $p(n,k)$ - число целочисленных разбиений n в k частей

  $p(n) = \sum\limits_{k=1}^{n} p(n,k)$ - число всех разбиений для n

  Ex. $5 = 5 = 4+1 = 3+2 = 3+1+1 = 2+2+1 = 2+1+1+1 = 1+1+1+1+1$

- **Принцип включения/исключения** (Principle of Incusion/Exclusion (PIE))

  $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$

  Ex. есть $n = 11$ объектов, нужно распределить их между $k = 3$ группами A, B и C

  Эту задачу можно решить с помощью *Stars and bars method*, тогда мы получим $\binom{n+k-1}{n, k-1} = \binom{13}{2} = 78$

  Введем ограничение: пусть мощность каждого множества будет не больше 4.

  Посчитаем количество неподходящих вариантов:

  $|A| = |\{b_A \geq 5\}| = 1 \cdot \binom{11-5+3-1}{3-1} = \binom{8}{2} = 28$

  $|A \cap B| = |\{b_A \geq 5 \wedge b_B \geq 5\}| = \binom{3}{2} = 3$

  $|A \cap B \cap C| = |\{b_A \geq 5 \wedge b_B \geq 5 \wedge b_C \geq 5\}| = 0$

  Итого получаем $28 \cdot 3 + 3 \cdot 3 + 0 = 75$ вариантов.

  Далее исключаем эти варианты из количества всех вариантов, а значит подходящих вариантов всего $78 - 75 = 3$

- **Принцип включения/исключения** (Inclusion/Exclusion Principle (PIE))

  - X - начальное множество элементов

– $P_1, \ldots, P_m$ - свойства
– Пусть $X_i = \{x \in X \mid P_i$ - свойство для $x\}$
– Пусть $S \in [m]$ - множество свойств
– Пусть $N(S) = \bigcap\limits_{i \in S} X_i = \{x \in X \mid x$ имеет все свойства $P_1, \ldots, P_m\}$

$N(\varnothing) = X \quad |N(\varnothing)| = |X| = n$

- **Теорема ПВ/И** (Theorem PIE)
  $|X \setminus (X_1 \cup X_2 \cup \ldots \cup X_m)| = \sum\limits_{S \subseteq [m]} (-1)^{|S|} |N(S)|$ - количество элементов множества X, не имеющих никакое из свойств
  Доказательство:
  Пусть $x \in X$
  Если x не имеет свойств $P_1, \ldots, P_m$, то $x \in N(\varnothing)$ и $x \notin N(S)\ \forall S \neq \varnothing$
  Поэтому x дает в общую сумму 1
  Иначе, если x имеет $k \geq 1$ свойств $T \in \binom{[m]}{k}$,
  то $x \in N(S)$ тогда и только тогда, когда $S \subseteq T$.
  Поэтому x дает в сумму $\sum\limits_{S \subseteq T} (-1)^{|S|} = \sum\limits_{i=0}^{k} \binom{k}{i} (-1)^i = 0$

- **Следствие**
  $|\bigcup\limits_{i \in [m]} X_i| = |X| - \sum\limits_{S \subseteq [m]} (-1)^{|S|} |N(S)| = \sum\limits_{S \subseteq [m], S \neq \varnothing} (-1)^{|S|-1} |N(S)|$

- **Приложения**:
  * Определяете «плохие» свойства $P_1, \ldots, P_m$
  * Посчитываете $N(S)$
  * Применяете ПВ/И

- **Количество сюръекций (правототальных функций)**
  * $X = \{$функция $f : [k] \to [n]\}$
  * Плохое свойство $P_i\ :\ X_i = \{f : [k] \to [n] \mid \nexists j \in [k] : f(j) = i\}$
  * $|\{$сюръекции $f : [k] \to [n]\}| = |X \setminus (X_1 \cup \ldots \cup X_m)| \overset{\text{PIE}}{=} \sum\limits_{S \subseteq [m]} (-1)^{|S|} |N(S)| = \sum\limits_{S \subseteq [m]} (-1)^{|S|} (n - |S|)^k = \sum\limits_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n$

- **Количество биекций**
  $n! = \sum\limits_{i=0}^{n} (-1)^i \binom{n}{i} (n-i)^n$
- **Число Стирлинга** (опять)

Заметим, что сюръекция = разбиение, тогда:
$\sum_{i=0}^{k}(-1)^i \binom{k}{i} (k-i)^n = n! S_n^{II}(k)$

- **Беспорядки** (Derangements) - перестановка без фиксированных точек
  Если $f(i) = i$, то i - фиксированная точка
  * X = все $n!$ перестановок
  * Плохие свойства $P_1, \ldots, P_m : \pi \in X$ имеет свойство $P_i \iff \pi(i) = i$
  * Посчитаем $N(S)$: $\quad N(S) = (n - |S|)!$
  * Применяем ПВ/И: $X \setminus (X_1 \cup \ldots \cup X_n) = \sum_{S \subseteq [n]} (-1)^{|S|} N(S) = \sum_{S \subseteq [n]} (-1)^{|S|} (n - |S|)! = \sum_{i=0}^{n} (-1)^i \binom{n}{i} (n-i)!$

8. Рекуррентности и производящие функции

- **Производящие функции** (Generating Functions)

$\sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \dots$

Функция выше задает последовательность $a_0, a_1, a_2, \dots$

Ex. $3 + 8x^2 + x^3 + \frac{1}{7}x^5 + 100x^6 + \dots \to (3, 0, 8, 1, 0, \frac{1}{7}, 100, \dots)$

Ex. Последовательность $(1, 1, 1, \dots)$ задает функцию $1 + x + x^2 + \dots = \sum_{n=0}^{\infty} x^n$

Пусть $S = 1 + x + x^2 + \dots$, тогда $xS = x + x^2 + \dots$, $(1-x)S = 1 \Longrightarrow$

$\boxed{S = \frac{1}{1-x} \text{ задает последовательность } (1, 1, 1, \dots)}$

Ex.

$\frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots = \sum_{n=0}^{\infty} (-1)^n x^n$

$\frac{1}{1-3x} = 1 + 3x + 9x^2 + 27x^3 + \dots = \sum_{n=0}^{\infty} 3^n x^n$

$\frac{2}{1-x} = 2 + 2x + 2x^2 + 2x^3 + \dots = \sum_{n=0}^{\infty} 2x^n$

$(2, 4, 10, 28, 82, \dots) = (1, 1, 1, 1, 1, \dots) + (1, 3, 9, 27, 81, \dots)$

$\frac{1}{1-x} + \frac{1}{1-3x} = \frac{2-4x}{(1-x)(1-3x)}$

$\frac{1}{1-x^2} = 1 + x^2 + x^4 + x^6 + \dots = \sum_{n=0}^{\infty} x^{2n} \to (1, 0, 1, 0, \dots)$

$\frac{x}{1-x^2} = x + x^3 + x^5 + \dots = \sum_{n=0}^{\infty} x^{2n+1} \to (0, 1, 0, 1, \dots)$

Взятие производной:

$\frac{d}{dx}(\frac{1}{1-x}) = \frac{1}{(1-x)^2} = \frac{d}{dx}(1 + x + x^2 + \dots) = 1 + 2x + 3x^2 + 4x^3 + \dots \to (1, 2, 3, 4, \dots)$

Ex. Найти ПФ для $(1, 3, 5, 7, 9, \dots)$

$A(x) = 1 + 3x + 5x^2 + \dots$

$xA = 0 + x + 3x^2 + 5x^3 + \dots$

$(1-x)A = 1 + 2x + 2x^2 + 2x^3 + \dots$

$(1-x)A = 1 + \frac{2x}{1-x} \quad A = \frac{1 + \frac{2x}{1-x}}{1-x} = \frac{1+x}{(1-x)^2}$

Ex. Найти ПФ для $(1, 4, 9, 16, \dots)$

$A = 1 + 4x + 9x^2 + 16x^3 + \dots \quad (1-x)A =$

- **Подсчет, используя производящие функции**

Найти число решений для $x_1 + x_2 + x_3 = 6$, где $x_i \geq 0, x_1 \leq 4, x_2 \leq 3, x_3 \leq 5$

$A_1(x) = 1 + x + x^2 + x^3 + x^4$

$A_2(x) = 1 + x + x^2 + x^3$

$A_3(x) = 1 + x + x^2 + x^3 + x^4 + x^5$

$A(x) = A_1 \cdot A_2 \cdot A_3 = 1 + 3x + 6x^2 + 10x^3 + 14x^4 + 17x^5 + \underline{18x^6} + 17x^7 + \ldots$

Ответ - 18

- **Рекуррентные соотношения** (Recurrence relations)

  <u>Решить рекуррентное соотношение</u> - найти закрытую формулу

  Ex. Арифметическая прогрессия

  $$a_n = \begin{cases} a_0 = const & n = 0 \\ a_{n-1} + d, & n > 0 \end{cases}$$

  Решение: $a_n = a_0 + nd$ - анзац (Ansatz, догадка)

  Проверка: $a_n = a_0 + nd = a_{n-1} + d = a_0 + (n-1)d + d = a_0 + nd$ - 👍👍

- Метод характеристического уравнения

  Рекуррентное соотношение $\overset{a_n \to r^n}{\rightsquigarrow}$ Характеристическое уравнение $\overset{\text{решение}}{\rightsquigarrow}$ Корни $\overset{\text{магия}}{\rightsquigarrow}$ Решение $\rightsquigarrow$ Проверка

  Ex. $a_n = a_{n-1} + 6a_{n-2}$

  $r^n - r^{n-1} - 6r^{n-2} = 0$

  $r^{n-2}(r^2 - r - 6) = 0$

  $r_{1,2} = -2, 3$

  $$\boxed{\begin{array}{l} \text{Если } r_1 \neq r_2, \text{ то } a_n = ar_1^n + br_2^n \text{ - общее решение} \\ \text{Если } r_1 = r_2 = r, \text{ то } a_n = ar^n + bnr^n \end{array}}$$

  $a_n = a(-2)^n + b(3)^n$

  Пусть $\begin{cases} a_0 = 1 = a + b \\ a_1 = 8 = -2a + 3b \end{cases}$

  $$-5a = 5 \Longrightarrow \begin{cases} a = -1 \\ b = 2 \end{cases} \Longrightarrow a_n = -(-2)^n + 2 \cdot 3^n$$

- **Разделяй и властвуй** (Divide-and-Conquer)

  $$T(n) = \underbrace{2T\left(\frac{n}{2}\right)}_{\text{работа рекурсии}} + \underbrace{\theta(n)}_{\text{работа разделения/слияния}}$$

- **Основная теорема о рекуррентных соотношениях** (Master Theorem) *тык*

  $T(n) = aT\left(\frac{n}{b}\right) + f(n)$

  Из этого, $c_{crit} = \log_b a$

  <u>I случай</u>: слияние < рекурсия

  $f(n) \in O(n^c)$, где $c < c_{crit} \Longrightarrow T(n) \in \Theta(n^{c_{crit}})$

$f(n) \in O(n^c) \Longleftrightarrow f(n) \in o(n^{c_{crit}})$

II случай: слияние $\approx$ рекурсия

$f(n) \in \Theta(n^{c_{crit}} \log^k n) \Longrightarrow T(n) \in \Theta(n^{c_{crit}} \log^{k+1} n)$

Здесь $k \geq 0$. В общем случае см. википедию

III случай: слияние > рекурсия

$f(n) \in \Omega(n^c)$, где $c > c_{crit} \Longrightarrow T(n) \in \Theta(f(n))$

- **Метод Акра-Бацци** (Akra-Bazzi method) *тык*

  $T(n) = f(n) + \sum\limits_{i=1}^{k} a_i T(b_i n + h_i(n)) \Longrightarrow T(n) \in \Theta\left(n^p \cdot \left(1 + \int_1^n \frac{f(x)}{x^{p+1}} dx\right)\right)$, где p - решение для $\sum\limits_{i=1}^{k} a_i b_i^p = 1$

  $$\begin{cases} k = const \\ a_i > 0 \\ 0 < b_i < 1 \\ h_1(n) \in O(\frac{n}{\log^2 n}) \text{ - малые возмущения} \end{cases}$$

  Ex. $T(n) = T\left(\lfloor\frac{n}{2}\rfloor\right) + T\left(\lceil\frac{n}{2}\rceil\right) + n$ - асимптотика сортировки слиянием

  $T(n) = T\left(\frac{n}{2} + O(1)\right) + T\left(\frac{n}{2} - O(1)\right) + \theta(n)$

  Здесь $b_i = \frac{1}{2}, \quad h = \pm O(1) \in O\left(\frac{n}{\log^2 n}\right)$

  Ex. $T(n) = T\left(\frac{3n}{4}\right) + T\left(\frac{n}{4}\right) + n$

  $a_1 = 1, b_1 = \frac{3}{4}, a_2 = 1, b_2 = \frac{1}{4}, f(n) = n$

  $(\frac{3}{4})^p + \left(\frac{1}{4}\right)^p = 1$

  $p = 1$

  $\int_1^n \frac{x}{x^{1+1}} dx = \int_1^n \frac{dx}{x} = \ln x\Big|_1^n = \ln n$

  $T(n) \in \Theta(n \cdot (1 + \ln n))$

  $T(n) \in \Theta(n \ln n)$

- Решить рекуррентное соотношение $a_n = 3a_{n-1} - 2a_{n-1}$, где $a_0 = 1, a_1 = 3$

  Используем производящие функции:

  $A(x) = \frac{1}{1 - 3x + 2x^2} = \frac{1}{(1-x)(1-2x)} = \frac{-1}{1-x} + \frac{2}{1-2x} \to 2^{n+1} - 1$

- **Линейные рекуррентности** (Linear recurrences)

  $$\underbrace{k_1 a_n + k_2 a_{n-1} + k_3 a_{n-2} + \ldots}_{\text{линейная комб. рекуррентных членов}} = \underbrace{f(n)}_{\text{функция от } n}$$

Линейное рекуррентное соотношение - $\begin{cases} f = 0 \Longrightarrow \text{гомогенное (однородное)} \\ f \neq 0 \Longrightarrow \text{негомогенное (неоднородное)} \end{cases}$

Ex. Последовательность Фибоначчи:

$$F(n) = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ F(n-1) + F(n-2) \end{cases}$$

$F(n) - F(n-1) - F(n-2) = 0$ - однородное

- Операторы:

  Сумма: $(f + g)(n) = f(n) + g(n)$

  Умножение на число: $(\alpha \cdot f)(n) = \alpha f(n)$

  Сдвиг: $(Ef)(n) = f(n+1)$

  Ex. $E(f - 3(g - h)) = Ef + (-3)Eg + 3Eh$

  Составные операторы:

  $(E - 2)f = Ef + (-2)f = f(n+1) - 2f(n)$

  $E^2 f = E(Ef) = f(n+2)$

  Ex. $f(n) = 2^n$

  $2f = 2 \cdot 2^n$

  $Ef = 2^{n+1}$

  $(E^2 - 1)f(n) = E^2 f(n) - f(n) = 2^{n+2} - 2^n = 3 \cdot 2^n$

- **Аннигилятор** (Annihilator) - оператор, который трансформирует f в функцию, тождественную 0

  Ex. Оператор $(E - 2)$ аннигилирует функцию $f(n) = 2^n$

  Ex. $(E - c)$ аннигилирует c^n

  Ex. $(E - 3)(E - 2)$ аннигилирует $2^n + 3^n$

  Ex. $(E - c)^d$ аннигилирует любую функцию формы $p(n) \cdot C^n$, где $p(n)$ - многочлен степени не больше $d - 1$

  Nota. Любой составной оператор аннигилирует класс функций

  Nota. Любая функция, составленная из полинома и экспоненты, имеет свой единственный аннигилятор

  Если X аннигилирует f, то X также аннигилирует Ef

  Если X аннигилирует f и Y аннигилирует g, то XY аннигилирует $f \pm g$

- Аннигилирование рекуррентностей:
  1. Запишите рекуррентное соотношение в форме операторов
  2. Выделите аннигилятор для соотношения
  3. Разложите на множители (если понадобится)
  4. Выделите общее решение из аннигилятора
  5. Найдите коэффициенты используя базовые случаи (если даны)

  Ex. $r(n) = 5r(n-1), r(0) = 3$
  1. $r(n+1) - 5r(n) = 0 \quad (E-5)r(n) = 0$
  2. $(E-5)$ аннигилирует $r(n)$
  3. $(E-5)$ уже разложен
  4. $r(n) = \alpha \cdot 5^n$
  5. $r(0) = 3 \Longrightarrow \alpha = 3$

  Ex. $T(n) = 2T(n-1) + 1, \quad T(0) = 0$
  1. $(E-2)T(n) = 1$
  2. $(E-2)$ не аннигилирует $T(n)$, остается 1. Тогда добавим аннигилятор $(E-1)$, получим, что $(E-1)(E-2)$ аннигилирует $T(n)$
  3. Разложение не требуется
  4. $T(n) = \alpha \cdot 2^n + \beta$ - общее решение
  5. $T(0) = 0 = \alpha \cdot 2^0 + \beta$
  $T(1) = 1 = \alpha \cdot 2^1 + \beta$
  $\alpha = 1, \beta = -1$

- **Псевдонелинейные уравнения** (Pseudo-non-linear equations)

  Ex. $a_n = 3a_{n-1}^2, a_0 = 1$
  $\log_2 a_n = \log_2(3a_{n-1}^2)$
  Пусть $b_n = \log_2 a_n$
  $b_n = 2b_{n-1} + \log_2 3, b_0 = 0$
  $b_n = (2^n - 1)\log_2 3$
  $a_n = 2^{(2^n-1)\log_2 3} = 3^{2^n-1}$

X. Программа экзамена в 2023/2024

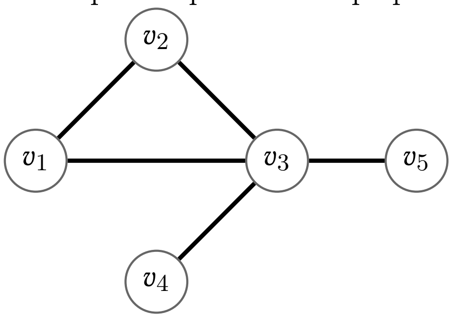
5. Теория графов.

1. **Ориентированные и неориентированные графы** (*Directed and undirected graphs*)

   Граф - множество вершин V и множество ребер E (в общем случае), соединяющие какие-либо две вершины: $G(V, E)$

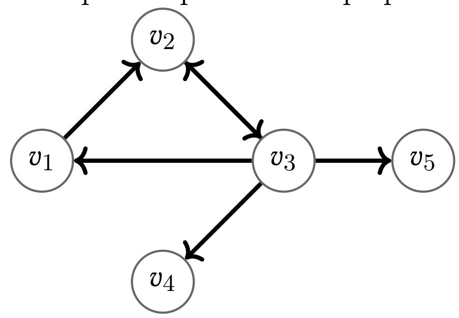
   По виду ребер различают:

   неориентированный граф

   - ребра не имеют направлений

   ориентированный граф

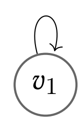
   

   - ребра имеют направления

2. **Простые графы и псевдографы** (*Simple graphs and pseudographs*)

   Простой граф $G(V, E)$ - граф, в котором

   - $V \neq \varnothing$ - граф не пустой
   - $E \subseteq V \times V$ - ребра представлены как множество пар вершин
   - $\forall v \in V \ \langle v, v \rangle \notin E$ - граф не содержит петлей - ребер, соединяющих одну вершину с собой же

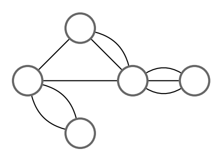
   Петля

   Псевдограф $G(V, E)$ - простой граф, в котором разрешены петли

3. **Мультиребра и мультиграфы** (*Multiedges and multigraphs*)

   Мультиребра - ребра, соединяющие одну и ту же пару вершин больше одного раза

   Мультиграфы - графы, содержащие мультиребра. В этом случае E - мультимножество

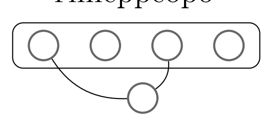
   

4. **Гиперграфы** (*Hypergraphs*)

   Гиперребро - ребро, соединяющее несколько вершин

   Гиперграф - граф, содержащий гиперребро

   гиперребро

5. **Нуль-граф, пустой граф и синглтон** (*Null, empty, singleton graphs*)

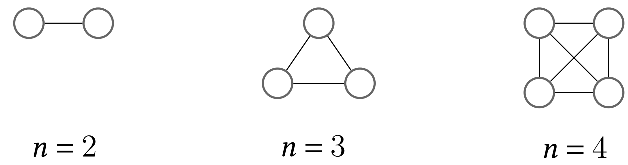
   Нуль-граф - граф, не содержащий вершин (и ребер)

   Пустой граф - граф, не содержащий ребер

   Синглтон - граф, содержащий из одной вершины

6. **Полный граф** (*Complete graph*)

   Полный граф K_n - простой граф из n вершин, в которой все вершин соединены друг с другом



7. **Взвешенный граф** (*Weighted graph*)

   Взвешенный граф - граф, в котором ребра (и/или вершины) имеют числовой вес. Иначе говоря, определена функция $w : E \to \mathbb{R}$

8. **Планарный граф** (*Planar graphs*)

   Планарный граф - граф, который можно изобразить на плоскости без пересечений рёбер

   По теореме Понтрягина-Куратовского граф планарен *тогда и только тогда, когда* он не содержит подграфов, гомеоморфных полному графу из пяти вершин K_5 или графу «домики и колодцы» $\mathrm{K}_{3,3}$

9. **Подграф** (*Subgraph*)

   Подграф графа $G(V, E)$ - граф $G'(V', E')$ такой, что $V' \subseteq V, E' \subseteq E$

10. **Остовный подграф** (*Spanning subgraph*)

    Остовный подграф графа $G(V, E)$ - такой подграф $G'(V, E')$, содержащий все вершины исходного

11. **Порожденный подграф** (*Induced subgraph*)

    Порожденный подграф $G[S]$ графа $G(V, E)$ - подграф $G'(S, E')$, который содержит все ребра, соединяющие вершины из S в исходном графе

12. **Отношение смежности** (*Adjacency relation*)

    Отношение смежности - отношение A между вершинами, соединенными ребром: $A = \{\langle u, v\rangle \mid \langle u, v\rangle \in E\}$

13. **Матрица смежности** (*Adjacency matrix*)

    Матрица смежности - матрица A_V, выражающее отношение смежности

14. **Отношение инцидентности** (*Incidence relation*)

    Отношение инцидентности - отношение B между вершиной и соединяющей ее ребром: $B = \{\langle u, e\rangle \mid u \in V \land e \in E \land \exists v \in V \mid (\langle u, v\rangle \in E \lor \langle v, u\rangle \in E)\}$

15. **Матрица инцидентности** (*Incidence matrix*)

    Матрица инцидентности - матрица A_V, выражающее отношение инцидентности

16. **Степень вершины** (*Vertex degree*)

    Степень $\deg(v)$ вершины v - количество и ребер из этой вершины (петли считаются дважды)

    Назовем $\delta(G)$ - минимальная степень вершины в графе, $\Delta(G)$ - максимальная степень вершины в графе

17. **Регулярный граф** (*Regular graph*)

    r-регулярный граф - граф, все степени вершин которого равны r - $\forall v \in V \ \deg(v) = r$

18. **Лемма о рукопожатиях** (*Handshaking lemma*)

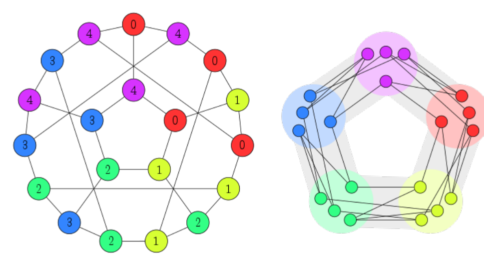
    $\sum\limits_{v \in V} \deg(v) = 2|E|$ - сумма степеней всех вершин равна удвоенному количеству ребер

19. **Изоморфизм графов** (*Graph isomorphism*)

    Графы $G(V, E)$ и $H(U, F)$ называются изоморфными, если существует биекция $f \mid V \to U$ такая, что если вершины v и u графа G смежны, то и вершины $f(v)$ и $f(u)$ графа H тоже смежны

20. **Гомоморфизм графов** (*Graph homomorphism*)

    Гомоморфизм графов - отображение вершин графа G в вершины графа H такое, что смежные вершины графа G отображаются в смежные вершины графа H



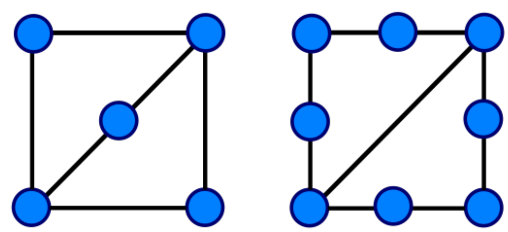
Гомоморфизм

21. **Гомеоморфизм графов** (*Graph homeomorphism*)

    Деление (Subdivision) ребра $\langle u, v\rangle$ - операция, добавляющее вершину w, ребра $\langle u, w\rangle$ и $\langle w, v\rangle$ и удаляющее ребро $\langle u, v\rangle$

    Исключение (Smoothing) вершины w (степени 2) - операция, обратная делению - исключение вершины w и ребер $\langle u, w\rangle$ и $\langle w, v\rangle$ и добавление ребра $\langle u, v\rangle$

    Графы G и H гомеоморфны, если граф H можно получить в результате деления или исключения графа G



Гомеоморфизм

22. **Пути и циклы** (*Walks, paths, trails, cycles*)

    Путь (Walk) - последовательность из вершин и ребер, соединяющих соседние вершины: $l = (v_0, e_0, v_1, e_1, \ldots, e_{n-1}, v_n)$

    Цепь (Trail) - путь (walk), все ребра которого различны

    Простая цепь (Path) - путь (walk), все вершины (и соответственно ребра) которого различны

    Замкнутый путь (Closed walk) - путь (walk), начальная вершина которого является конечной

    Контур (Circuit) - цепь (trail), являющаяся замкнутым

    Цикл (Cycle) - простая цепь (path), являющаяся замкнутым

    (*терминология из Википедии)

23. **Эйлеровы путь, цикл, граф** (*Eulerian path, cycle, graph*)

    Эйлеров путь - путь, содержащий все ребра графа

    Эйлеров цикл - замкнутый путь, содержащий все ребра графа

Граф называют эйлеровым, если в нем есть эйлеров цикл. Граф называют полуэйлеровым, если в ней есть эйлеров путь.

24. **Теорема Эйлера для графов** (*Euler's theorem for graphs*)

    Граф эйлеров, если все степени вершин четные, а ребра принадлежат одной компоненте связности

    Граф полуэйлеров, если ровно 2 вершины имеют нечетную степень, а ребра принадлежат одной компоненте связности

25. **Гамильтоновы путь, цикл, граф** (*Hamiltonian path, cycle, graph*)

    Гамильтонов путь - путь, содержащий все вершины графа

    Гамильтонов цикл - замкнутый путь, содержащий все вершины графа

    Граф называют гамильтоновым, если в нем есть гамильтонов цикл. Граф называют полугамильтоновым, если в ней есть гамильтонов путь.

26. **Теорема Оре** (*Ore's theorem*)

    Теорема Оре - достаточное условие существования гамильтонова цикла: если в графе $G(V,E)$ для любых $u,v \in V$ $\deg u + \deg v \geq |V|$, то граф G гамильтонов

27. **Теорема Дирака** (*Dirac's theorem*)

    Теорема Дирака - достаточное условие существования гамильтонова цикла: если в графе $G(V,E)$ для любой $u \in V$ $\deg u \geq \dfrac{|V|}{2}$, то граф G гамильтонов

28. **Эксцентриситет вершины** (*Eccentricity of a vertex*)

    Расстояние $\operatorname{dist}(u,v)$ - длина (количество ребер) кратчайшего пути между u и v

    Эксцентриситет $\varepsilon(v)$ - наибольшая длина кратчайшего пути от этой вершины до другой в этом графе: $\varepsilon(v) = \max\limits_{u \in V} \operatorname{dist}(v,u)$

29. **Радиус и диаметр графа** (*Radius and diameter of a graph*)

    Радиус графа $\operatorname{rad}(G)$ - наименьший эксцентриситет вершины из графа: $\operatorname{rad}(G) = \min\limits_{v \in V} \varepsilon(v)$

    Диаметр графа $\operatorname{diam}(G)$ - наибольший эксцентриситет вершины из графа: $\operatorname{diam}(G) = \max\limits_{v \in V} \varepsilon(v)$

30. **Центр графа** (*Center of a graph*)

    Центр графа - вершина (вершины), эксцентриситет которой равен радиусу графа: $\operatorname{center}(G) = \{v \in V \mid \varepsilon(v) = \operatorname{rad}(G)\}$

31. **Центроид дерева** (*Centroid of a tree*)

    Центроид дерева - вершина (или 2 вершины), удаление которой приведет к распаду на поддеревья, каждое из которое имеет не больше $\dfrac{|V|}{2}$ вершин

    Очевидно, что только деревья, состоящие из четного количества вершин, могут иметь 2 центроида

32. **Клика** (*Clique*)

    Клика графа - порожденный подграф, который является полный графом. 1-клика - вершина, 2-клика - 2 вершины и ребро, 3-клика - треугольник, n-клика - граф K_n

33. **Независимое (стабильное множество)** (*Independent set*)

    Независимое (стабильное) множество - множество вершин, каждая из которых не соединена ребром с другой вершиной из множества

34. **Паросочетание** (*Matching*)

    Паросочетание (независимое множество ребер) - множество ребер, каждые из которые не соединяют одну и ту же вершину

35. **Идеальное паросочетание** (*Perfect matching*)

    Идеальное паросочетание - паросочетание, ребра которого инцидентны ко всем вершинам графа (то есть паросочетание, являющееся реберным покрытием)

36. **Вершинное покрытие** (*Vertex cover*)

    Вершинное покрытие - множество вершин, к которым инцидентны все ребра графа

37. **Реберное покрытие** (*Edge cover*)

    Реберное покрытие - множество ребер, которые инцидентны ко всем вершинам

38. **Дерево** (*Tree*)

    Дерево - связный ацикличный граф

39. **Лес** (*Forest*)

    Лес - несвязный граф, каждая компонента которого не имеет циклов (граф, состоящий из деревьев)

40. **Минимальное остовное дерево** (*Minimum spanning tree*)

    Минимальное остовное дерево взвешенного графа $G(V, E, w)$ - дерево $T(V, E')$, сумма весов ребер которого имеет наименьшее значение

41. **Код Прюфера** (*Prüfer code*)

    Код Прюфера - алгоритм кодировки маркированного дерева размера n в последовательность чисел

    Кодировка:

    1. Делаем биекцию между названиями вершин и числа из диапазона $[1; n]$ (если необходимо)

    2. Берем лист с наименьшим значением, удаляем его, записываем в последовательность номер его родителя

    3. Повторяем 2. до тех пор, пока не останется 2 вершины - их кодировка тривиальна и не нуждается в хранении

    Декодировка:

    1. Создаем n вершин, и множество вершин W, которых нет в последовательности

    2. Читаем номер вершины из последовательности

    3. Соединяем эту вершину с вершиной из W с минимальным номером, удалив ее

    4. Добавляем вершину из последовательности в W

    5. Повторяем 2.-4.

    6. Соединяем 2 оставшиеся вершины из W

42. **Двудольный граф** (*Bipartite graph*)

    Двудольный граф $K_{n,m}$ - граф, вершины которого можно разбить на две части размеров n и m таким образом, что вершины из одной части не смежны друг с другом

43. **Теорема баланса регулярных двудольных графов** (*Theorem on the balance of regular bipartite graphs*)

    Если двудольный граф $K_{n,m}$ регулярный, то $n = m$

    □ Граф регулярный $\Longrightarrow \forall v \in V \ \deg v = r \in \mathbb{N} \Longrightarrow$ левая доля имеет nr исходящих ребер, а правая доля имеет mr входящих ребер, но так как вершины в долях не соединены ребрами, $nr = mr$ □

44. **Теорема существования идеального паросочетания регулярного двудольного графа** (*Theorem on the existence of a perfect matching in a regular bipartite graph*)

    Теорема: у любого r-регулярного двудольного графа ($r > 0$) существует идеальное паросочетание

    □

    Пусть $G(V, E)$ - граф, вершины разбиваются на две доли $X \oplus Y = V$

    Пусть $N(A) = \{y \in Y \mid \exists x \in X \ \langle x, y \rangle \in E\}$ - соседи (смежные вершины) вершин из множества $A \subseteq X$

    Докажем от противного: пусть идеального паросочетания не существует, тогда по теореме Холла $\exists S \subset X \mid |S| > |N(S)|$, но тогда кол-во ребер, выходящих из S, равно $r|S|$, но кол-во ребер, выходящих из $N(S)$, равно $r|N(S)|$

    Из этого $r|S| > r|N(S)|$, что невозможно, так как $N(S)$ - соседи S - противоречие

    □

45. **Теорема Холла** (*Hall's theorem (on the existence of an X-perfect matching in a bipartite graph)*)

    Пусть $G(V, E)$ - граф, вершины разбиваются на две доли $X \oplus Y = V$

    Тогда в графе $G(V, E)$ существует X-идеальное паросочетание (паросочетание, покрывающее все вершины X) тогда и только тогда, когда для любого $A \subset X$ $|A| \leq |N(A)|$

    □ Если существует такое A, что $|A| > |N(A)|$, то какой-либо вершине из A не найдется противоположная вершина из $N(A)$ и X-идеального паросочетания не выйдет □

46. **Связность в неориентированных графах** (*Connectivity in undirected graphs*)

    Компонента связность графа - максимальный подграф, в котором от каждой вершины до любой другой существует путь

    Граф считается связным, если он представляет собой одну компоненту связности

47. **Сильная и слабая связность в ориентированных графах** (*Strong and weak connectivity in directed graphs*)

    Компонента сильной связности - максимальный подграф, в котором для любых вершин u, v существует пути $u \rightsquigarrow v$ и $v \rightsquigarrow u$

    Компонента слабой связности - максимальный подграф, который является компонентой связности в неориентированном графе, полученном при удалении ориентации ребер у исходного

48. **Конденсация ориентированного графа** (*Condensation of a directed graph*)

    Конденсация графа - сжатие сильно связных компонент графа до вершин с целью получения упрощенного и ациклического графа

49. **Вершинная связность** (*Vertex connectivity*)

    Вершинная связность $\kappa(G)$ графа G - минимальное число вершин, которое нужно удалить в графе, чтобы он стал несвязным или синглтоном

50. **Реберная связность** (*Edge connectivity*)

    Реберная связность $\lambda(G)$ графа G - минимальное число ребер, которое нужно удалить в графе, чтобы он стал несвязным

51. **Теорема Уитни** (*Whitney's theorem*)

    Для любого графа $\kappa(G) \leq \lambda(G) \leq \delta(G)$

    □

    Допустим, что $\kappa(G) > \lambda(G)$, тогда после удаления $\lambda(G)$ ребер будет $k \leq \lambda(G)$ вершин со одной стороны и $m \leq \lambda(G)$ с другой. Но мы их тоже можем удалить, и граф распадется, значит $\lambda < \kappa(G) = \min(k, m) \leq \lambda(G)$ - противоречие

    Допустим, что $\lambda(G) > \delta(G)$, тогда мы можем найти в графе вершину с наименьшей степенью $\delta(G)$, при удалении $\delta(G)$ ребер граф распадется, значит $\lambda(G) = \delta(G)$ - противоречие

    □

52. ***k*-связный граф** (*k-connected graph*)

    k-вершинно-связный граф - граф, остающийся связным после удаления k вершин ($\kappa(G) \geq k$).

    НО: синглтон имеет $\kappa(G) = 0$, он не 1-вершинно-связный, при этом он связный; K_2 имеет $\kappa(G) = 1$, поэтому он не 2-вершинно-связный, но K_2 может быть блоком

    k-реберно-связный граф - граф, остающийся связным после удаления k ребер ($\lambda(G) \geq k$)

    НО: у синглтона $\lambda(G) = 0$, он не 1-реберно-связный, при этом синглтон - компонента реберной двусвязности

53. **Теорема Менгера** (*Menger's theorem*)

    Теорема (Менгера о реберной двойственности в ориентированном графе):

    Между вершинами u и v существует L реберно непересекающихся путей тогда и только тогда, когда после удаления любых $(L-1)$ ребер существует путь из u в v.

    Теорема (Менгера о вершинной двойственности в ориентированном графе):

    Между вершинами u и v существует L вершинно непересекающихся путей тогда и только тогда, когда после удаления любых $(L-1)$ вершин существует путь из u в v.

    Доказательства

54. **Двусвязность** (*Biconnectivity*)

    Двусвязность (вершинная) определяется как отношение эквивалентности 2 ребер, между концами которых существуют 2 вершинно-различных пути

Компонента (вершинной) двусвязности (также блок) - подграф, который включает все двусвязные ребра (класс эквивалентности двусвязности).

Реберная двусвязность определяется как отношение эквивалентности 2 вершины, между которыми существуют 2 реберно-различных пути

Компонента реберной двусвязности - подграф, который включает все двусвязные вершины (класс эквивалентности двусвязности).

55. **Точка сочленения** (*Articulation point*)

    Точка сочленения - вершина, принадлежащая нескольким компонентам (вершинной) двусвязности

56. **Мост** (*Bridge*)

    Мост - ребро, соединяющее две компоненты реберной двусвязности

57. **Блок** (*Blocks*)

    Блок - компонента вершинной двусвязности

58. **Дерево блоков и точек сочленений** (*Block-cut tree*)

    Дерево блоков и точек сочленений графа - дерево, в котором каждая вершина представляет собой либо точку сочленения, либо блок, при этом вершина точки сочленения соединена только с вершиной блока и наоборот

6. Теория автоматов.

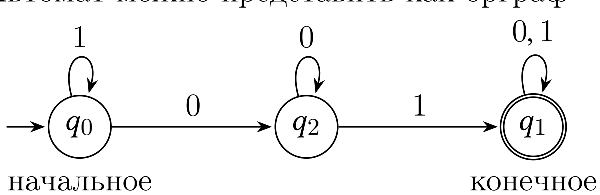
1. **Детерминированный конечный автомат** (*Deterministic Finite Automaton (DFA)*)

   Детерминированный конечный автомат $A = (Q, \Sigma, \delta, q_0, F)$ - объект, представляющий собой множество состояний Q, множество входных символов Σ, функция переходов $\delta : Q \times \Sigma \to Q$, начальное состояние q_0 и множество конечных состояний F

   Автомат принимает какую-то цепочку символов из Σ^* и решает, принадлежит ли она соответствующему автомату регулярному языку L

   Для простоты обычно выбирают $\Sigma = \{0, 1\}$

   Автомат можно представить как орграф



   Или как таблицу функции переходов

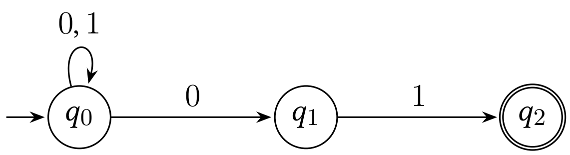
	0	1
$\to q_0$	q_2	q_0
$*q_1$	q_1	q_1
q_2	q_2	q_1

2. **Недетерминированный конечный автомат (НКА)** (*Non-deterministic Finite Automaton (NFA)*)

   Недетерминированный конечный автомат $A = (Q, \Sigma, \delta, q_0, F)$ - объект, представляющий собой множество состояний Q, множество входных символов Σ, функция переходов $\delta : P(Q) \times \Sigma \to \mathcal{P}(Q)$, начальное состояние q_0 и множество конечных состояний F

   Главное отличие НКА от ДКА: от одного состояния в НКА можно перейти сразу к нескольким другим или к ни одному

   Пример:



	0	1
$\to q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\varnothing$	$\{q_1\}$
$*q_2$	$\varnothing$	$\varnothing$

3. **Формальные языки** (*Formal languages*)

   Формальный язык L - множество конечных слов над конечным алфавитом символов Σ

4. **Операции над формальными языками (конкатенация, объединение, замыкание Клини)** (*Operations on formal languages (concat, union, Kleene closure)*)

   Конкатенация LM языков L и M - множество слов, состоящих из записанных подряд слова из L и слова из M: $LM = \{uw \mid u \in L \land w \in M\}$

   Объединения $L \cup M$ языков L и M - множество слов, которые содержатся в L или/и в M: $L \cup M = \{w \mid w \in L \lor w \in M\}$

   Замыкание Клини L^* языка L - множество слов, которые могут быть получены в результате конкатенации слов из L: $L^* = \{w_1 w_2 \dots w_n \forall n \geq 0 \mid w_i \in L\}$ (включая пустое слово ε)

5. **Регулярные языки** (*Regular languages*)

   Регулярный язык - формальный язык, который задается некоторым автоматом

   Также регулярный язык задается индуктивно:

   1. Пустое множество $\varnothing$ и множество из пустой строки $\{\varepsilon\}$ являются регулярными языками

   2. Множество из однобуквенного слова $\{a\}$, где $a \in \Sigma$ является регулярным языком

   3. Для регулярных языков α и β объединение $\alpha \cup \beta$, конкатенация $\alpha\beta$ и замыкание Клини α^* - тоже регулярные языки

   4. Других регулярных языков нет

6. **Регулярное выражение** (*Regular expression*)

   Регулярное выражение - способ описания регулярного языка

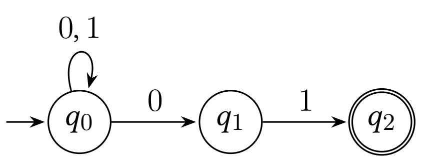
Регулярное выражение	Язык, который оно описывает
	$\varnothing$
ε	$\{\varepsilon\}$
a (какое-либо РВ)	α
b (какое-либо РВ)	β
(a)	α
ab	$\alpha\beta$
$a+b$	$\alpha\cup\beta$
$a*$	α^*

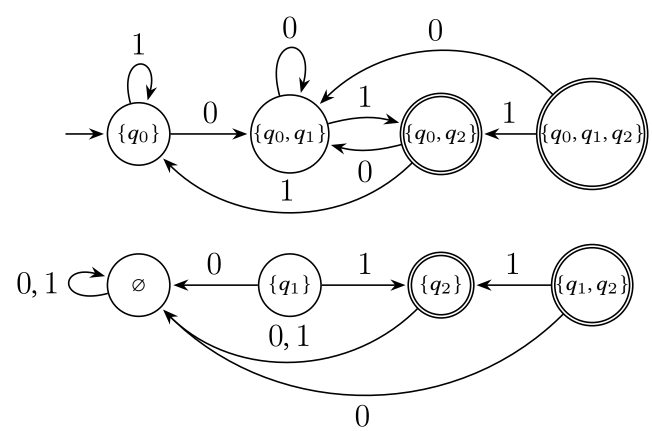
7. **Теорема Клини** (*Kleene's theorem*)

   Для любого регулярного выражения существует конечный автомат, и они описывают равные регулярные языки

8. **Конструкция подмножеств (ДКА из НКА)** (*Powerset construction (DFA from NFA)*)

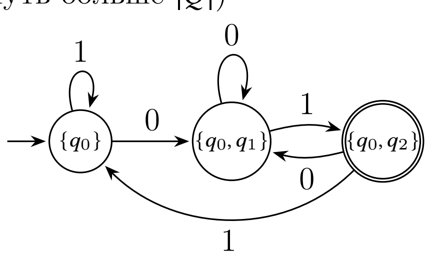
   Из состояний Q НКА построим ДКА с состояниями, каждое из которых представляет собой подмножество Q. Далее при помощи магии умным образом строим переходы





	0	1
$\varnothing$	$\varnothing$	$\varnothing$
$\to\{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\{q_1\}$	$\varnothing$	$\{q_2\}$
$\{q_0,q_1\}$	$\{q_0,q_1\}$	$\{q_0,q_2\}$
$*\{q_2\}$	$\varnothing$	$\varnothing$
$*\{q_2,q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$*\{q_2,q_1\}$	$\varnothing$	$\{q_2\}$
$*\{q_2,q_1,q_0\}$	$\{q_0,q_1\}$	$\{q_0,q_2\}$

   Как можем видеть, 5 состояний являются недостижимыми, поэтому их мы можем удалить. В итоге в ДКА остается 3 состояния (зачастую количество состояний не $2^{|Q|}$, а чуть больше $|Q|$)

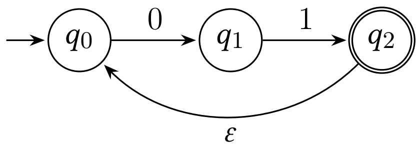


9. **ε-НКА** (ε-*NFA*)

ε-НКА $A = (Q, \Sigma, \delta, q_0, F)$ - НКА, допускающий ε переходы (переходы по пустым строчкам)

Тогда $\delta\ :\ Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

Пример - автомат, допускающий цепочки 01(01)∗:



10. **Конструкция НКА из ε-НКА** (*NFA construction from ε-NFA*)

    Алгоритм:

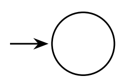
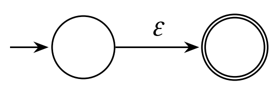
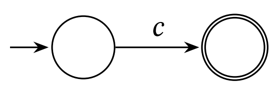
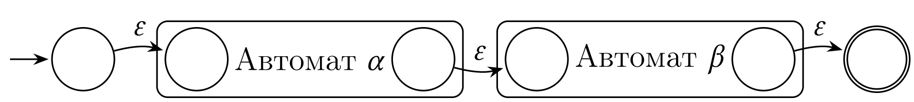
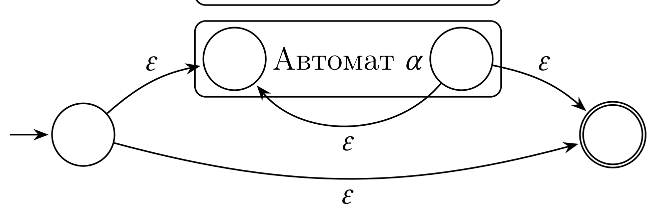
    1. Транзитивное замыкание: если из состояния u мы можем сделать больше одного ε-перехода в состояние w, то мы можем сделать сразу ε-переход из u в w

    2. Добавление допускающих состояний: если есть ε-переход из u в w, причем w - допускающее состояние, то u можно сделать тоже допускающем

    3. Добавление ребер: если есть переходы $\delta(u, \varepsilon) = v$ и $\delta(v, c) = w$, то сделаем равное ребро $\delta(u, c) = w$

    4. Удаление ε-переходов

11. **Конструкция Томпсона (ε-НКА из регулярного выражения)** (*Thompson's construction (ε-NFA from regular expression)*)

Регулярное выражение	Язык, который оно описывает	Автомат
	$\varnothing$	
ε	$\{\varepsilon\}$	
c (символ)	$\{c\}$	
ab	$\alpha\beta$	
$a + b$	$\alpha \cup \beta$	
$a*$	α^*	

Пользуясь этими преобразованиям, можно построить ε-НКА

12. **Алгоритм Клини** (*Kleene's algorithm*)

Алгоритм Клини - алгоритм для превращения ДКА в регулярное выражение

Пусть ДКА $(Q, \Sigma, \delta, q_0, F)$, а $Q = \{q_0, \ldots, q_n\}$, $F = \{q_i \mid i \in \mathbb{N}_F \subset \mathbb{N}\}$

Определим $R_{ij}^{-1} = a_1 + \cdots + a_m$, где $q_j \in \delta(q_i, a_k)$ для k - другими словами все символы, по которым можно перейти из q_i в q_j. Для $i = j$ $R_{ii}^{-1} = a_1 + \cdots + a_m + \varepsilon$

Далее для каждого k от 0 до n итеративно определяем

$R_{ij}^{k} = R_{ik}^{k-1}(R_{kk}^{k-1}) * R_{kj}^{k-1} | R_{ij}^{k-1}$

Таким образом, ответом будет регулярное выражение $\bigcup\limits_{i \in \mathbb{N}_F} R_{0i}^{n}$

13. **Лемма о накачке для регулярных языков** (*Pumping lemma for regular languages*)

Если L - регулярный язык, то существует константа $p \geq 1$, зависящая от L, такая, что любая строка $w \in L(|w| \geq p)$ может быть записана $w = xyz$ так, что удовлетворены условия:

1. $|y| \geq 1$
2. $|xy| \leq p$
3. Для любого $n \geq 0$ $xy^n z \in L$

14. **Свойства замыкания регулярных языков** (*Closure properties of regular languages*)

Для регулярных языков L и M:

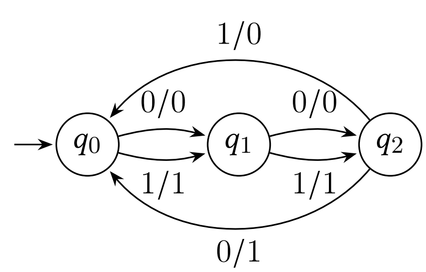
1. L^* (замыкание Клини) - регулярный язык
2. $L \cup M$ (объединение) - регулярный язык
3. LM (конкатенация) - регулярный язык
4. $L \cap M$ (пересечение) - регулярный язык
5. $\overline{L}$ (дополнение - $\Sigma^* \setminus L = \overline{L}$) - регулярный язык
6. L^R (инверсия - $abac \to caba$) - регулярный язык
7. $L \setminus M$ (разность) - регулярный язык
8. $h(L)$ (гомоморфизм $h \mid \Sigma \to \Sigma^*$, например $h(0) = ab, h(1) = ba$) - регулярный язык
9. $h^{-1}(L)$ (обратный гомоморфизм $h^{-1} \mid \Sigma^* \to \Sigma$, например $h^{-1}(01) = a, h^{-1}(10) = b$) - регулярный язык

15. **Автомат Мили** (*Mealy machine*)

Автомат Мили $M_{\text{Mealy}} = (Q, \Sigma, \Omega, q_0, \delta, \lambda)$ - автомат, выводящий последовательность, которая зависит от входной последовательности

Здесь Ω - алфавит выходящей последовательности, а $\lambda : Q \times \Sigma \to \Omega$ - функция выходов, зависящая от состояния и входного символа

Значение функции λ на ребре графа обозначают после переходного символа



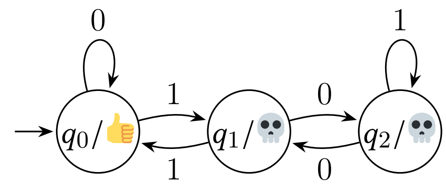
Этот автомат Мили преобразует каждый 3-ий символ с 0 на 1 и наоборот: `100101` $\to$ `101100`

16. **Автомат Мура** (*Moore machine*)

   Автомат Мура $M_{\text{Moore}} = (Q, \Sigma, \Omega, q_0, \delta, \lambda)$ - автомат, выводящий последовательность, зависящую от входной последовательности

   Как и в автомате Мили, в автомате Мура Ω - алфавит выходящей последовательности, но $\lambda : Q \to \Omega$ - функция выходов, зависящая от текущего состояния

   Значение функции λ на графе обозначают в вершине состояния



Этот автомат Мура выдает 👍, если двоичное число делится на 3, иначе 💀

17. **Пустота языка конечного автомата** (*Emptiness of finite automaton language*)

   Язык автомата L считается пустым в том случае, если язык не содержит никаких цепочек (в том числе пустых) - $L = \varnothing$

   По конечному автомату можно понять, является ли язык пустым: если какое-либо допускающее состояние можно достигнуть из начального, то язык автомата не является пустым (это можно определить при помощи обхода графа)

18. **Конечность языка конечного автомата** (*Finiteness of finite automaton language*)

   Язык автомата L считается конечным, если он содержит конечное множество цепочек

   Конечность языка можно определить так: если есть такое состояние v, что к нему можно прийти из начального состояния, от него можно прийти к какому-либо допускающему состоянию, а из v можно каким-либо образом прийти в v, то язык бесконечный - мы можем сколь угодно раз зацикливаться по v и получать бесконечное количество цепочек

19. **Эквивалентность конечных автоматов** (*Equivalence of finite automata*)

   Автоматы эквивалентны, если они допускают одно и то же множество цепочек.

   Пусть автомат $A = (Q, \Sigma, \delta, q_0, F)$. Введем функцию $\lambda : Q \to \{0, 1\}$, возвращающую `1`, если состояние допускающее, иначе `0`

   Введем такое отношение эквивалентности $R_0 \subset Q \times Q$ между состояниями. Определим, что $q\, R_0\, p$ в том случае, если $\lambda(q) = \lambda(p)$

   Теперь определим R_1 как отношение состояний q и p, для которых $\lambda(q) = \lambda(p)$ и $\lambda(\delta(q, c)) = \lambda(\delta(p, c))$ для любого символа $c \in \Sigma$

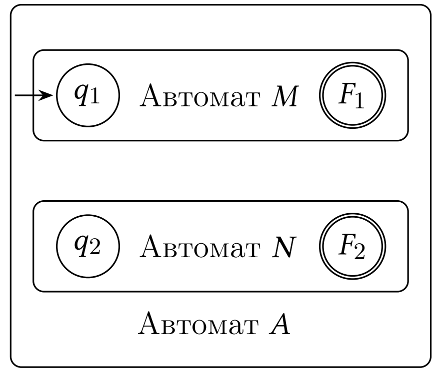
   Теперь определим R_2 как отношение состояний q и p, для которых $\lambda(\hat{\delta}(q, w)) = \lambda(\hat{\delta}(p, w))$ для любой цепочки $w \in \Sigma^*$ длины не больше 2

   Окончательно определим R как отношение состояний q и p, для которых $\lambda(\hat{\delta}(q, w)) = \lambda(\hat{\delta}(p, w))$ для любой цепочки $w \in \Sigma^*$

Пусть даны автоматы $M = (Q_1, \Sigma, \delta_1, q_1, F_1)$ и $N = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

Теперь построим такой автомат $A = (Q, \Sigma, \delta, q_1, F)$, выбрав какое-либо начальное состояние, объединив множества состояний и множества допускающих состояний и расширив функцию переходов

Автоматы M и N эквивалентны, если состояния q_1 и q_2 принадлежат одному классу эквивалентности, то есть $q_1 \, R \, q_2$



20. **Теорема Майхилла-Нероуда** (*Myhill-Nerode theorem*)

    На языке L определим различимое расширение как строку z, которой можно расширить строки x и y до строк xz и yz так, что только одна из этих строк принадлежит языку L

    Определим отношение эквивалентности $\sim_L$ на языке L как отношение между такими строками x и y, что не существует никакого различимого расширения z (то есть либо строки xz, yz принадлежат языку, либо не принадлежат). Отношение $\sim_L$ разделяет цепочки на классы эквивалентности

    Теорема Майхилла-Нероуда гласит:

    1) Язык L регулярен тогда и только тогда, когда количество классов эквивалентности конечно

    2) Минимальный ДКА, допускающий язык L, имеет столько состояний, сколько классов эквивалентности

    3) Любой минимальный ДКА, допускающий L, изоморфен следующему: пусть каждый класс эквивалентности $[x]$ для строки x будет соотнесен к состоянию, причем существуют переходы $[x] \to [xa]$ для $a \in \Sigma$, начальным состоянием будет состояние класса $[\varepsilon]$, а допускающими состояниями будут состояния классов $[s]$ для $s \in L$

7. Комбинаторика.

1. (*Ordered arrangements*)
2. (*Permutations*)
3. (*k-permutations*)
4. (*Cyclic permutations*)
5. (*Unordered arrangements*)
6. (*k-combinations*)
7. (*Multisets*)
8. (*Permutations of multisets*)
9. (*Combinations of infinite multisets*)
10. (*Compositions*)

11. (*Set partitions*)
12. (*Stirling numbers of the second kind*)
13. (*Integer partitions*)
14. (*Principle of Inclusion-Exclusion*)

8. Рекуррентности и производящие функции.

1. (*Recurrence relations*)
2. (*Solving recurrence relations using characteristic equations*)
3. (*Generating functions*)
4. (*Power series*)
5. (*Solving linear recurrences using generating functions*)
6. (*Solving combinatorial problems using generating functions*)
7. (*Operators and annihilators*)
8. (*Solving linear recurrences using annihilators*)
9. (*Catalan numbers*)
10. (*Divide-and-Conquer algorithms analysis using recursion trees*)
11. (*Master theorem*)
12. (*Akra-Bazzi method*)