Дискретная математика. Теормин — IV

Содержание

itmo.arslee.me

1. Комбинаторика

1.1. Ordered arrangements

Будем обозначать $[n]$ как множество натуральных чисел от 1 до n.

Упорядоченная расстановка n элементов конечного множества X — отображение $s : [n] \to X$.

По сути это присвоение элементам из X номеров из $[n]$, благодаря чему и устанавливается порядок. Таким образом можно описать строку, последовательность, кортеж и т.п.

1.2. Pigeonhole principle

Он же обобщенный принцип Дирихле.

Пусть есть k попарно непересекающихся множеств $S_1, ..., S_k$, сумма мощностей которых равна n. Тогда:
- существует такое S_i, что $|S_i| \geq \lceil \frac{n}{k} \rceil$,
- существует такое S_j, что $|S_j| \leq \lfloor \frac{n}{k} \rfloor$,

Рассмотрим пример. Есть $k = 5$ гнезд, в которых живут суммарно $n = 17$ птиц. Значит:
- существует гнездо, в котором живет не меньше 4 птиц,
- существует гнездо, в котором живет не больше 3 птиц.

1.3. Permutations

Перестановка множества X — биекция $\pi : [n] \to X$, причем обычно $X = [n]$.

Пример перестановки чисел $i \in [5]$:

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

Множество всех перестановок $[n]$ обозначается как S_n. Его мощность вычисляется по формуле:

$$|S_n| = n!$$

1.4. Double counting

Двойной подсчет — подход к доказательству эквивалентности двух комбинаторных формул. Для этого мы формулируем комбинаторную задачу и пытаемся решить ее двумя различными способами, установив таким образом равенство.

Рассмотрим пример. Пусть у нас есть n человек. Хотим разбить их на k команд и выделить для каждой группы по 1 лидеру.
- С одной стороны, можно сначала посчитать количество способов разбить людей на k команд, а потом для каждой команды посчитать количество способов выбрать лидера.
- С другой стороны, можно сначала посчитать количество способов выбрать k лидеров, а затем для каждого лидера посчитать количество способов добавить к нему сокомандников.

Там получается две разные формулы, и поскольку мы решили одну и ту же задачу, эти формулы равны между собой.

1.5. k-permutations

k-перестановка — упорядоченная расстановка k *различных* элементов из x, то есть инъекция $\pi : [k] \to X$

Множество всех k-перестановок множества $[n]$ обозначается как $P(n, k)$. Его мощность вычисляется по формуле:

$$|P(n, k)| = n \cdot (n - 1) \cdot \ldots \cdot (n - k + 1) = \frac{n!}{(n - k)!}$$

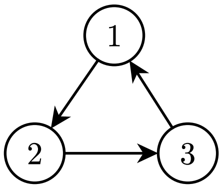
Эта формула также обозначается как $n^{\underline{k}}$ или $(n)_k$.

В частности, $S_n = P(n, n)$.

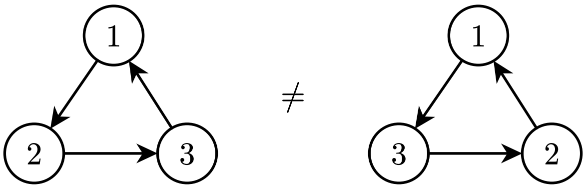
1.6. Cyclic permutations

Циклическая k-перестановка — расстановка k элементов из $[n]$ по кругу.

Например, пусть $n = k = 3$. Тогда расстановки $(1, 2, 3)$, $(2, 3, 1)$ и $(3, 1, 2)$ будут считаться идентичными:



Но $(1, 2, 3)$ и $(1, 3, 2)$ — различные.



Множество всех циклических k-перестановок обозначается как $P_c(n, k)$. Его мощность вычисляется по формуле:

$$|P_c(n, k)| = \frac{n!}{k \cdot (n - k)!}$$

1.7. Unordered arrangements

Неупорядоченная расстановка k элементов множества X — мультимножество $S = \langle X, r \rangle$ мощности k.

Более интересным является частный случай, когда кратность каждого элемента из S равна единице. Тогда просто $S \subseteq X$. И из этого также следует, что все элементы S различные. Все это в совокупности подводит к понятию k-сочетаний...

1.8. k-combinations

k-сочетание множества X — неупорядоченная расстановка k *различных* элементов из X. Или же просто подмножество X мощности k.

Множество всех k-сочетаний обозначается так:

$$\binom{X}{k} = \{A \subseteq X \mid |A| = k\}$$

Например, множество ребер в простом неориентированном графе — это 2-сочетания его вершин, то есть $E \subseteq \binom{V}{2}$.

Если $|X| = n$, то имеет место обозначение:

$$\binom{n}{k} \coloneqq \left|\binom{X}{k}\right|$$

и вычисляется по формуле:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

1.9. Binomial theorem

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

где n — целое неотрицательное.

1.10. Multisets

Мультимножество — такое множество, которое допускает повторяющиеся элементы.
Обозначается как $M = \langle X, r \rangle$, где:

- X — базовое множество,
- $r : X \to \mathbb{N}_0$ — функция кратности.

Например, пусть дано:

$$X = \{a, b, c\}, \quad r(a) = 2, \quad r(b) = 1, \quad r(c) = 3.$$

Тогда $M = \langle X, r \rangle$ выглядит так:

$$M = \{a, a, b, c, c, c\}^* = \{2 \cdot a,\ 1 \cdot b,\ 3 \cdot c\}$$

P.S. Также встречается нотация кратности $r_a = 2,\ r_b = 1,\ r_c = 3$.

1.11. Permutations of multisets

Пусть M — конечное мультимножество над базовым множеством X. *k-перестановка M* — упорядоченная расстановка k элементов из M, при которой перестановка совпадающих элементов не важна.

Уточнение: $k = \sum k_i$, то есть это сумма кратностей элементов перестановки.

Более интересен частный случай, когда $n = k$. Количество всех перестановок M равно *мультиномиальному коэффициенту*, который вычисляется по формуле:

$$\binom{n}{k_1, \ldots, k_r} = \frac{n!}{k_1! \ \ldots \ k_r!}$$

1.12. Combinations of infinite multisets

Пусть дано мультимножество $S = \langle X, r \rangle$, у которого $|X| = s$. И пусть $r_x \geq k$ для выбранного k (либо просто $r_x = \infty$).

Количество k-сочетаний *достаточно большого* (или *бесконечного*) мультимножества вычисляется по формуле:

$$\binom{k+s-1}{k} \text{ или } \binom{k+s-1}{s-1}$$

Шары и перегородки…

1.13. Multinomial theorem

$$(x_1 + ... + x_r)^n = \sum_{\substack{0 \leq k_1, ..., k_r \leq n \\ k_1 + ... k_r = n}}^{n} \binom{n}{k_1, ..., k_r} \cdot x_1^{k_1} \cdot ... \cdot x_r^{k_r}$$

Биномиальная теорема — частный случай мультиномиальной при $r = 2$.

1.14. Number of bijections on an n-element set

$$n!$$

1.15. Number of subsets of an n-element set

$$2^n$$

1.16. Number of functions from an m-set to an n-set

$$n^m$$

1.17. Compositions

- *Слабая композиция* целого неотрицательного числа k из s частей — решение уравнения $b_1 + ... + b_s = k$, где $b_i \geq 0$.

  Например, пусть $k = 3$ и $s = 2$. Тогда возможные решения для $b_1 + b_2 = 3$ таковы:
  - $(b_1, b_2) = (0, 3)$
  - $(b_1, b_2) = (3, 0)$
  - $(b_1, b_2) = (1, 2)$
  - $(b_1, b_2) = (2, 1)$

  Общее число слабых композиций считается по формуле:

$$\binom{k+s-1}{k, s-1}$$

- *Композиция* целого положительного k из s частей —- решение того же уравнения, но $b_i > 0$. Общее число композиций вычисляется по формуле:

$$\binom{k-1}{s-1}$$

1.18. Stirling numbers of the second kind

Число Стирлинга второго типа — количество различных разбиений множества X мощности n на k блоков. Обозначается как $S(n,k)$ или $\left\{ n \atop k \right\}$ и вычисляется по рекурсивной формуле:

$$\left\{ n \atop k \right\} = \begin{cases} 1 & n = k = 0 \\ 0 & n = 0 \text{ или } k = 0 \\ \left\{ n-1 \atop k-1 \right\} + k \cdot \left\{ n-1 \atop k \right\} & \text{в остальных случаях} \end{cases}$$

1.19. Bell numbers

Число Белла — количество всевозможных разбиений X ($|X| = n$) на произвольное количество блоков. Обозначается как B_n и вычисляется по формуле:

$$B_n = \sum_{k=0}^{n} \left\{ n \atop k \right\}$$

1.20. Integer partitions

Разбиение целого положительного числа n на k положительных слагаемых — решение уравнения $n = a_1 + ... + a_k$, где $a_1 \geq a_2 \geq ... \geq a_k \geq 1$. Порядок слагаемых не важен.

Количество различных k-разбиений обозначается как $p_k(n)$ вычисляется по рекурсивной формуле:

$$p_k(n) = \begin{cases} 1 & n = k = 0 \\ 0 & n \geq 1 \text{ и } k = 0 \\ 0 & k \geq n \\ p_k(n-k) + p_{k-1}(n-1) & 1 \leq k \leq n \end{cases}$$

Количество всевозможных разбиений (т.н. *функция разбиения*):

$$p(n) = \sum_{k=0}^{n} p_k(n)$$

1.21. Ferrers diagrams and Young tableaux

Графический способ изобразить разбиения целого числа.

Например, пусть $14 = 6 + 4 + 3 + 1$.

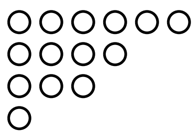


Диаграмма Ферререса

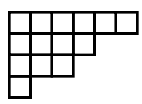


Таблица Юнга

1.22. Principle of Inclusion-Exclusion

Пусть дано:

- конечное множество X,
- свойства $P_1, ..., P_m$.

Определим следующие множества:

- $X_i = \{x \in X \mid x \text{ обладает свойством } P_i\}$,
- $S \subseteq [m]$,
- $N(S) = \{x \in X \mid \forall i \in S : x \text{ обладает свойством } P_i\}$, или же $N(S) = \bigcap_{i \in S} X_i$.

Количество элементов из X, которые не удовлетворяют никаким из свойств $P_1, ..., P_m$ вычисляется по формуле:

$$|x \setminus (X_1 \cup ... \cup X_m)| = \sum_{S \subseteq [m]} (-1)^{|S|} |N(S)|$$

Это и называется принципом включений-исключений.

1.23. Derangements

D_n — множество *беспорядков* множества $[n]$, то есть таких перестановок, что каждый позиция каждого элемента никогда не повторяется.

Мощность вычисляется по формуле:

$$|D_n| = \sum_{i=0}^{n} (-1)^i \binom{n}{i} (n-i)!$$

1.24. The Twelvefold Way

12 задач подсчета количества положить шары в коробки при различных условиях:

Отличимость		Количество шаров в коробке		
Шары	**Коробки**	≤ 1	≥ 1	**Произвольное**
Неотличимые	Различные	$\binom{k}{n}$	$\binom{n-1}{k-1}$	$\binom{k+n-1}{n}$
Различные	Неотличимые	$\begin{cases} 1 & n \leq k \\ 0 & n > k \end{cases}$	$\left\{ {n \atop k} \right\}$	$\sum_{m=1}^{k} \left\{ {n \atop m} \right\}$
Различные	Различные	$\frac{n!}{(n-k)!}$	$k! \left\{ {n \atop k} \right\}$	k^n
Неотличимые	Неотличимые	$\begin{cases} 1 & n \leq k \\ 0 & n > k \end{cases}$	$p_k(n)$	$\sum_{m=1}^{k} p_m(n)$

2. Рекуррентные соотношения

2.1. Recurrence relations

Рекуррентное соотношение — последовательность, очередой член которого выражается через его предыдущие.

Пример — арифметическая прогрессия:

$$a_0 = \text{const}, \quad a_n = a_{n-1} + d$$

Или числа Фибоначчи:

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

2.2. Solving recurrence relations using characteristic equations

Пусть нам дано *линейное однородное* рекуррентное соотношение степени k, то есть имеющее вид:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$$

Где c_i — константы, причем $c_k \neq 0$.

Хотим найти *закрытую форму*, то есть нерекурсивную формулу для a_n. Для этого составим *характеристическое уравнение*:

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - \dots - c_k = 0,$$

где r — корни уравнения.

То есть мы сделали замену $a_i = r^i$ и перенесли все слагаемые в правую часть.

Теперь рассматриваем случаи (для наглядности, на уравнении $r^2 - c_1 r - c_2 = 0$):
- Если корни различные, то $a_n = \alpha_1 r_1^n + \alpha_2 r_2$, где α_i — константы.
- Если корни одинаковые, то $a_n = (\alpha_1 + \alpha_2 n) r^n$, где α_i — константы.
- Если корни комплексные, то $a_n = \rho^n (\alpha \cos(n\theta) + \beta \sin(n\theta))$, где:
  - α и β — константы,
  - ρ и θ — модуль и аргумент корней (в полярном представлении комплексных чисел).

Для нахождения констант, делаем систему уравнений, приравняв получившиеся формулы к известным значениям a_i.

А теперь пусть нам дано *линейное неоднородное* соотношение степени k, то есть имеющее вид:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + F(n)$$

Для нахождения ее закрытой формы делаем следующее:
1. Решаем однородную часть $a_n^{(h)}$, то есть отбросив $F(n)$.
2. Подбираем частное решение. Некоторые случаи:
   - Если $F(n) = kn + b$ (т.е. линейное), то $a_n^{(p)} = cn + d$. Коэффициенты c и d подбираем постановкой $a_n^{(p)}$ в исходное соотношение.
   - Если $F(n) = c \cdot r^n$ (т.е. экспоненциальное), то $a_n^{(p)} = \beta r^n$. Но если r является корнем характеристического уравнения, то берем $a_n^{(p)} = \beta n^m r^n$, где m — кратность корня r.
3. $a_n = a_n^{(h)} + a_n^{(p)}$
4. Подбираем неизвестные коэффициенты, составив систему уравнений с известными значениями.

2.3. Generating functions

Производящая функция — степенной ряд вида

$$G(x) = \sum_{n=0}^{\infty} a_n x^n,$$

где a_n — какая-то числовая последовательность.

Например, ряд $(1, 2, 3, 4, \ldots)$ можно «закодировать» в виде производящей функции:

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

Идея в том, что комбинируя между собой различные производящие функции, можно эффективно решать комбинаторные задачи.

2.4. Power series and operations on power series

Что такое степенной ряд, в общем-то, понятно из курса матанализа. Полагаю, что здесь более полезным будет описанть некоторые операции, которые встречаются в задачах с производящими функциями.

Пусть даны две производящие функции:
- $F(x) = \sum_{n=0}^{\infty} a_n x^n$,
- $G(x) = \sum_{n=0}^{\infty} b_n x^n$.

Операции:
- Дифференцирование каждого слагаемого:

$$F'(x) = \sum_{n=0}^{\infty} (n+1) a_{n+1} x^n$$

- Умножение слагаемых на константу:

$$\lambda F(x) = \sum_{n=0}^{\infty} \lambda a_n x^n$$

- Сложение $F(x)$ и $G(x)$ по слагаемым:

$$F(x) + G(x) = \sum_{n=0}^{\infty} (a_n + b_n) x^n$$

- Умножение $F(x)$ на $G(x)$ по слагаемым:

$$F(x) \cdot G(x) = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} a_k b_{n-k} \right) x^n$$

2.5. Solving linear recurrences using generating functions

Рассмотрим пример. Хотим найти закрытую форму для чисел Фибоначчи:

$$F_n = F_{n-1} + F_{n-2}, \quad F_0 = 0, \quad F_1 = 1$$

Запишем производящую функцию:

$$G(x) = F_0 + F_1 x + F_2 x^2 + \ldots = x + x^2 + 2x^3 + \ldots$$

Теперь умножим выражение на x и на x^2:

$$xG(x) = F_0 x + F_1 x^2 + F_2 x^3 + \ldots$$
$$x^2 G(x) = F_0 x^2 + F_1 x^3 + F_2 x^4 + \ldots$$

Теперь заметим, что все коэффициенты в выражении $G(x) - xG(x) - x^2 G(x)$ уничтожатся, кроме $F_1 x = x$. Получаем уравнение:

$$G(x) - xG(x) - x^2 G(x) = x$$

Выражаем $G(x)$:

$$G(x) = \frac{x}{1 - x - x^2}$$

Раскладываем функцию на простые дроби:

$$G(x) = \frac{x}{1 - x - x^2} = \frac{1}{\sqrt{5}} \left(\frac{1}{1 - \varphi x} - \frac{1}{1 - \psi x} \right) = \frac{1}{\sqrt{5}} \sum_{n=0}^{\infty} (\varphi^n - \psi^n) x^n$$

$$\varphi = \frac{1 + \sqrt{5}}{2}, \qquad \psi = \frac{1 - \sqrt{5}}{2}$$

Ну вот как-то так рекурренты и решаются.

2.6. Solving combinatorial problems using generating functions

Рассмотрим пример. Хотим найти количество целочисленных решений для уравнения $y_1 + y_2 + y_3 = 12$, причем $0 \leq y_i \leq 6$.

Определим последовательность чисел a_n как количество целочисленных решений $y_1 = n$. Очевидно, что последовательность имеет вид $(1, 1, 1, 1, 1, 1, 1, 0, 0, ...)$.

Запишем данную последовательность в виде производящей функции:

$$G_1(x) = x^0 + x^1 + ... + x^6.$$

Сделаем то же самое для y_2 и y_3.

Теперь перемножим 3 функции:

$$G(x) = G_1(x) G_2(x) G_3(x) = \left(1 + x + x^2 + ... + x^6\right)^3$$

Коэффициент при x^n в $G(x)$ говорит о количестве решений $y_1 + y_2 + y_3 = n$. Поскольку нам интересно $n = 12$, то мы раскроем ряд и найдем коэффициент при x^{12}:

$$[x^{12}] G(x) = 28$$

Ответ: 28.

2.7. Operators and annihilators

Оператор берет функцию (или несколько функций) и возвращает новую функцию.

Интересуемые нами операторы:
- Сумма: $(f + g)(n) := f(n) + g(n)$
- Масштабирование: $(\alpha f)(n) = \alpha f(n)$
- Смещение аргумента: $(\mathbf{E} f)(n) := f(n + 1)$

Из данных операторов можно выделить свойства:
- Линейность: $\mathbf{E}(f + g) = \mathbf{E} f + \mathbf{E} g$ и $\mathbf{E}(3f) = 3 \mathbf{E} f$
- Композиция: $(\mathbf{E} - 2) f := \mathbf{E} f - 2f$
- Возведение в степень: $\mathbf{E}^k f(n) = f(n + k)$
- Произведение: $(\mathbf{E} - 1)(\mathbf{E} - 2)(f) = (\mathbf{E}^2 - 3\mathbf{E} + 2) f$

Аннигилятор функции f — это любой такой нетривиальный оператор X, что $Xf = 0$.

Например, пусть $f(n) = \alpha^n$. Пример ее аннигилятора: $(\mathbf{E} - 2)$. Действительно,

$$(\mathbf{E} - 2)(\alpha 2^n) = \alpha 2^{n+1} - 2\alpha 2^n = 0.$$

2.8. Solving linear recurrences using annihilators

Решение рекуррентных соотношений с помощью аннигиляторов сводится к следующему:

1. С помощью операторов привести выражение к виду $Xf(n) = ...$, где X — оператор(-ы).
2. Добавить операторов, чтобы исходное выражение аннигилировалось.
3. Разложить получившееся выражение на линейные множители вида $(\mathbf{E} - a)$.
4. Перевести множители по справочной таблице.
5. Решить систему уравнений, чтобы найти коэффициенты.

Для наглядности найдем закрытую форму для такой рекурренты:

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

1. Приведем выражение к более удобной форме:

$$T(n+1) = 2T(n) + 1$$

   Теперь применим оператор смещения аргумента для T:

$$\mathbf{E}T(n) = 2T(n) + 1$$

   Перенесем все члены с $T(n)$ в левую часть и применим свойство линейности:

$$(\mathbf{E} - 2)T(n) = 1$$

2. Как можно видеть, $(\mathbf{E} - 2)$ не является аннигилятором. Добьем остаток с помощью оператора $(\mathbf{E} - 1)$:

$$(\mathbf{E} - 1)(\mathbf{E} - 2)T(n) = 0$$

3. Выражение уже разложено на линейные множители.
4. Из справочной таблицы узнаем, что $T(n) = \alpha 2^n + \beta$.
5. Ищем α и β:

$$\begin{cases} T(0) = \alpha + \beta = 0 \\ T(1) = 2\alpha + \beta = 1 \end{cases} \implies \begin{cases} \alpha = 1 \\ \beta = -1 \end{cases}$$

Итак, $T(n) = 2^n - 1$.

2.9. Catalan numbers

Числа Каталана можно определить по-разному:

- количество правильных скобочных последовательностей длины $2n$,
- количество неизоморфных полных бинарных деревьев с $n+1$ листьями (т.е. таких, у которых либо 0, либо 2 листа в каждом родителе),
- количество разбиений выпуклого $(n+2)$-угольника на треугольники
- и т.д.

n-ое число Каталана вычисляется по формуле:

$$C_n = \binom{2n}{n} \frac{1}{n+1}$$

Выводится как-то там через производящие функции.

2.10. Generalized binomial theorem

$$(1+x)^n = \sum_{k=0}^{\infty} \binom{n}{k} x^k$$

- $n, x \in \mathbb{R}$
- $\binom{n}{k} = \dfrac{p(n,k)}{k!}$
- $p(n,k) = n \cdot (n-1) \cdot \ldots \cdot (n-k+1)$ (см. Раздел 1.5).

2.11. Gamma function

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} \,\mathrm{d}t,$$

где $z \in \mathbb{C}$ и $\operatorname{Re}(z) > 0$.

Обладает всякими свойствами:
- $\Gamma(x+1) = x \cdot \Gamma(x)$
- $\Gamma(n) = (n-1)!$
- $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$
- $\Gamma(x)\Gamma(1-x) = \pi / \sin(\pi x)$
- и многим-многим другим…

Есть также определение в предельной форме:

$$\Gamma(x) = \lim_{n \to \infty} \frac{n! \cdot n^x}{\prod_{k=0}^{n} (x+k)}$$

Или вот такое *произведение Вейерштрасса*:

$$\frac{1}{\Gamma(x)} = x \cdot e^{\gamma x} \cdot \prod_{n=1}^{\infty} \left(\left(1 + \frac{x}{n}\right) e^{-x/n} \right),$$

где $\gamma = \lim_{n \to \infty} \left(\sum_{k=1}^{n} 1/k - \ln n \right)$ — *постоянная Эйлера-Максерони.*

По идее это интерполяция факториала на $\mathbb{R}$.

2.12. Asymptotic notation

- $f(n) \in O(g(n))$ — f *асимптотически ограничена сверху* функцией g, умноженной на константу:

$$f(n) \in O(g(n)) \qquad \Longleftrightarrow \qquad \exists c > 0. \quad \exists n_0. \quad \forall n > n_0. \quad |f(n)| \leq c \cdot g(n)$$

- $f(n) \in \Omega(g(n))$ — f *асимптотически ограничена снизу* функцией g, умноженной на константу:

$$f(n) \in \Omega(g(n)) \qquad \Longleftrightarrow \qquad \exists c > 0. \quad \exists n_0. \quad \forall n > n_0. \quad |f(n)| \geq c \cdot g(n)$$

- $f(n) \in o(g(n))$ — f *строго доминитруется* функцией g, то есть никакая константа не поможет «не отставать» от g:

$$f(n) \in o(g(n)) \qquad \Longleftrightarrow \qquad \forall c > 0. \quad \exists n_0. \quad \forall n > n_0. \quad |f(n)| \leq c \cdot g(n)$$

- $f(n) \in \omega(g(n))$ — f *строго доминирует* над функцией g, то есть при любой константе f будет быстрее g:

$$f(n) \in \omega(g(n)) \qquad \Longleftrightarrow \qquad \forall c > 0. \quad \exists n_0. \quad \forall n > n_0. \quad |f(n)| \geq c \cdot g(n)$$

- $f(n) \in \Theta(g(n))$ — f асимтотически ограничена *сверху и снизу* функцией g, то есть f и g растут с одинаковой скоростью

$$f(n) \in \Theta(g(n)) \qquad \Longleftrightarrow \qquad \begin{cases} f \in O(g) \\ f \in \Omega(g) \end{cases}$$

- $f(n) \sim g(n)$ — *асимтотически эквивалентные*:

$$f(n) \sim g(n) \qquad \Longleftrightarrow \qquad \lim_{n \to \infty} \frac{f(n)}{g(n)} = 1$$

2.13. Divide-and-Conquer algorithms analysis using recursion trees

Пусть дано рекуррентное соотношение такого вида:

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n).$$

Алгоритм с такой оценкой времени работает примерно так: задача разбивается на a подзадач размера $\frac{n}{b}$, и их сплит/мердж работает за $f(n)$.

Суть в том, чтобы оценить асимптотику глубины дерева рекурсий и перемножить на асимптотику сплита/мерджа листьев.

На примере алгоритма сортировки слиянием:

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$

Здесь мы делим массив пополам, и рекурсивно запускаем с этих двух половинок алгоритм. После чего за $\Theta(n)$ мерджим два массива.

Получается, что глубина рекурсии равна $\Theta(\log n)$, ну и мердж за $\Theta(n)$. Итого алгоритм работает за $\Theta(n) \cdot \Theta(\log n) = \Theta(n \log n)$.

2.14. Master theorem

Пусть дана рекуррента вида разделяй-и-властвуй:

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n),$$

где $a \geq 1$ и $b > 1$.

Посчитаем *критическую экспоненту*: $c_{\text{crit}} = \log_b a$.

Оценим ситуацию:

1. «Мердж быстрее рекурсии»:

$$\begin{cases} f(n) \in O(n^c) \\ c < c_{\text{crit}} \end{cases} \implies T(n) \in \Theta(n^{c_{\text{crit}}})$$

2. «Мердж примерно равен по времени рекурсии»:

$$\begin{cases} f(n) \in O(n^{c_{\text{crit}}} \log^k n) \\ k \geq 0 \end{cases} \implies T(n) \in \Theta\big(n^{c_{\text{crit}}} \log^{k+1} n\big)$$

3. «Мердж медленнее рекурсии»:

$$\begin{cases} f(n) \in \Omega(n^c) \\ c > c_{\text{crit}} \end{cases} \implies T(n) \in \Theta(f(n))$$

   Для данного случая также требуется *условие регулярности*: для некоторого $k < 1$ и достаточно больших n должно выполняться $af\left(\frac{n}{b}\right) \leq kf(n)$.

Теорема работает не для всех $f(n)$. Например, если $f(n) = \frac{n}{\log n}$, то, взяв соотношение $f(n)/n^{c_{\text{crit}}}$, можно увидеть, что $1/\log n < n^{\varepsilon}$ для любого $\varepsilon > 0$. То есть разница между $f(n)$ и $n^{c_{\text{crit}}}$ не является полиномом, отчего мастер теорема в данном случае неприменима.

2.15. Akra–Bazzi method

Обобщение мастер теоремы.

Пусть дана рекуррента такого вида:

$$T(n) = f(n) + \sum_{i=1}^{k} a_i T(b_i n + h_i(n)),$$

где:
- k — константа,
- $a_i > 0$,
- $0 < b_i < 1$,
- $h_i(n) \in O\left(\frac{n}{\log^2 n}\right)$ — *небольшое возмущение*.

Тогда ее асимптотика:

$$T(n) \in \Theta\left(n^p \cdot \left(1 + \int_1^n \frac{f(x)}{x^{p+1}} \,\mathrm{d}x\right)\right),$$

где p — решение уравнения $\sum_{i=1}^{k} a_i b_i^p = 1$.

2.16. Stirling's approximation

Для приближенного вычисления $n!$.

При $n \to \infty$:

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

Если быть более точным:

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\tau_n},$$

где $1/(12n+1) < \tau_n < 1/(12n)$.

2.17. Möbius Function

Для решетки делимости $\mathcal{P} = (\mathbb{Z}^+, |)$:

$$\mu(d) = \begin{cases} 1 & d = 1 \\ (-1)^k & d = p_1 ... p_k \text{ (все простые делители различные)} \\ 0 & d \text{ делится на квадрат простого числа} \end{cases}$$