Алгоритмы и Задачи

Поиск в массиве

a   x → i : a[i] = x

i ↓ число

Сортировка

1 3 4 2

↓

1 2 3 4

Проверка на простоту

2037

↓

Yes/No

Пример : Сортировка

Размер вх. данных: n.

Алг 1 $20n^2 + 10n$ – время работы

Алг 2 $1000n + 5$

Кто лучше?

Асимптотика

$f, g: \mathbb{N} \to \mathbb{N}$

$\boxed{\leqslant}$

<u>Def</u> 1) $f = O(g)$, если

$\exists N \in \mathbb{N}$ и $\exists C > 0$

т.ч. $\forall n \geqslant N \quad f(n) \leqslant C g(n)$

Пример: $\boxed{1000n + 5 = O(20n^2 + 10n)}$

$1000n + 5 \overset{?}{\leqslant} C \cdot (20n^2 + 10n)$

$C - ? \quad \forall n \geqslant ?$

$\underset{\forall n \geqslant 1}{1000n + 5 \leqslant 1005n} \underset{\forall n \geqslant 1}{\leqslant 1005n^2} \leqslant$

$\underset{\forall n \geqslant 1}{\leqslant} 1005 (20n^2 + 10n)$

$C = 1005$

Def 1) $f = O(g)$, если

$\exists N \in \mathbb{N}$ и $\exists C > 0$

т.ч. $\forall n \geq N$ $f(n) \leq C g(n)$

Def 2: $f = \Omega(g)$, если

$\exists N \in \mathbb{N}$ и $\exists C > 0$

т.ч. $\forall n \geq N$ $f(n) \geq C g(n)$

Def 3 $f = \Theta(g)$, если

$\exists N \in \mathbb{N}$ и $\exists C_1, C_2 > 0$

т.ч. $\forall n \geq N$ $C_1 g(n) \leq f(n) \leq C_2 g(n)$

Св-ва: (везде $f, g, h \in \mathbb{N} \to \mathbb{N}$)

1) $f = O(g)$ $(\leq)$
$f = \Omega(g)$ $(\geq)$ $\Rightarrow$ $f = \Theta(g)$ $(=)$

$f = O(g) \Rightarrow \exists N_1, C_1 > 0 : \forall n \geq N_1 : f(n) \leq C_1 g(n)$

$f = \Omega(g) \Rightarrow \exists N_2, C_2 > 0 : \forall n \geq N_2 : f(n) \geq C_2 g(n)$

$$\forall n \geq \max(N_1, N_2): \ c_1 g(n) \leq f(n) \leq c_2 g(n)$$

2) $f = \Theta(g)^{\boxed{=}} \Rightarrow g = \Theta(f)$

3) $f = O(g)^{\boxed{\leq}}$

$g = O(f)^{\boxed{\leq}}$ $\Rightarrow f = \Theta(g)$

4) $f = O(g)$

$g = O(h)$ $\Rightarrow f = O(h)$

5) $f = O(g)^{\boxed{\leq}} \Leftrightarrow g = \Omega(f)$

6) $f = \Theta(g)$

$g = \Theta(h)$ $\Leftrightarrow f = \Theta(h)$

Def 2 $f = \bar{o}(g)$ если

$\forall c > 0 \quad \exists N \in \mathbb{N}$

т. з. $\quad \forall n \geq N \quad f(n) \leq C g(n)$

Def 3 $f = \omega(g)$, если

$\forall c > 0 \quad \exists N \in \mathbb{N}$

т. з. $\quad \forall n \geq N \quad f(n) \geq C g(n)$

Пример $100n = o(n^2)$

$\forall c > 0 : \quad \exists N(c)$

т.ч. $100n \leq c \cdot n^2$

$\forall n \geq N(c)$?

$100n \leq c \cdot n^2$

$100 \leq cn$

$\forall n \geq \underbrace{\boxed{100/c}}_{N(c)} \quad 100n \leq c \cdot n^2$

Св-ва

1) $f = o(g) \Leftrightarrow \lim\limits_{n\to\infty} f(n)/g(n) = 0$

2) $f = o(g) \Leftrightarrow g = \omega(f)$

3) $f \pm o(f) = \Theta(f)$

применение 3:

$100n+5 = \Theta(100n)$

$5 = o(100n)$ т.к. св-во 1: $\frac{5}{100n} \to 0$, $n \to \infty$

Д-во 3:

Пусть $g = o(f) \Rightarrow \exists N: \ g(n) \leq 0.1 f(n) \ \forall n \geq N$

Тогда д-ть: $f + g = \Theta(f)$

[$f = \Theta(g)$, если
$\exists N \in \mathbb{N}$ и $\exists c_1, c_2 > 0$
т.ч. $\forall n \geq N \quad c_1 g(n) \leq f(n) \leq c_2 g(n)$]

$$\underbrace{0.9}_{c_1} f \leq f + g \leq \underbrace{1.1}_{c_2} f \iff$$

$$\iff g(n) \leq 0.1 f(n) \iff$$

$$\iff \forall n \geq N$$

Rem: $f, g : \mathbb{N} \to \mathbb{N}$

$\mathbb{N} \to \mathbb{Z}$

Функции с огр. значениями

Def $f = O(g) \iff \exists N \in \mathbb{N}$ и $\exists C > 0$

$\forall n \geq N \quad |f(n)| \leq C|g(n)|$

Пример: $10n^2 + 5n + 1 = \Theta(n^2)$

$5n+1 = o(10n^2)$

$\lim \frac{5n+1}{10n^2} = 0$

$10n^2 + 5n + 1 = \Theta(10n^2)$

$10n^2 = \Theta(n^2)$: $\underset{c''}{10n^2} \leq 10n^2 \leq \underset{c'}{10n^2}$

$10n^2 + 5n + 1 = \Theta(n^2)$

Св-во: пусть $f = \sum_{i=0}^{k} a_i n^i$ $(a_k \neq 0)$

$$f = \Theta(n^k)$$

Д-во:

$f' := a_0 + \dots + a_{k-1} n^{k-1} = o(n^k)$

т.к. $\lim \frac{a_0 + \dots + a_{k-1} n^{k-1}}{n^k} = 0$

$$f = a_k n^k + f' = \Theta(a_k n^k) = \Theta(n^k)$$

Пример:

Алг1: $20n^2 + 10n = \Theta(n^2)$ $O(n^2)$

Алг2: $1000n + 5 = \Theta(n)$

Зоопарк Функций

1 — Константное

$\log_2 n$ — логарифм.

$\log_2^k n$ — полилог.

n — линейная

$n \log n$

n^k — полиномиальные

c^n — эксп. функции

$n!$, n^n

Задача 1

2^n

$n := 30 \Rightarrow 1$ sec

$n := 60 \Rightarrow 10^9$ sec

Задача 2.

n^3

1GHz

$n = 100$: 0.001 sec

$n = 1000$: 1 sec

$n = 10000$: 1000 sec

Св-во $\log_a n = \log_b n \cdot \log_a b$

$$f = O(\log_3 n) \qquad \log_3 n = \log_2 n \cdot \log_3 2$$

$$= O(\log_2 n)$$

$$f = O(\log n)$$

Сортировка.

1 7 3 2 6 2

$\downarrow$

1 2 2 3 6 7

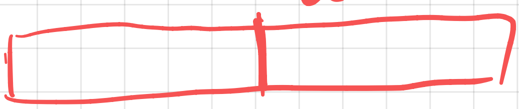
$$\underbrace{a_0 \dots \quad a_{n-1}}_{n}$$

merge-sort:

$n \le 1 \Rightarrow$ ничего не делать

$n \ge 2$

a^L $\quad$ a^R



Sort(a^L) $\quad$ Sort(a^R)

1 7 3 2 6 2

↓

[~~1~~ ~~3~~ ~~7~~] , [~~2~~ ~~2~~ ~~6~~]

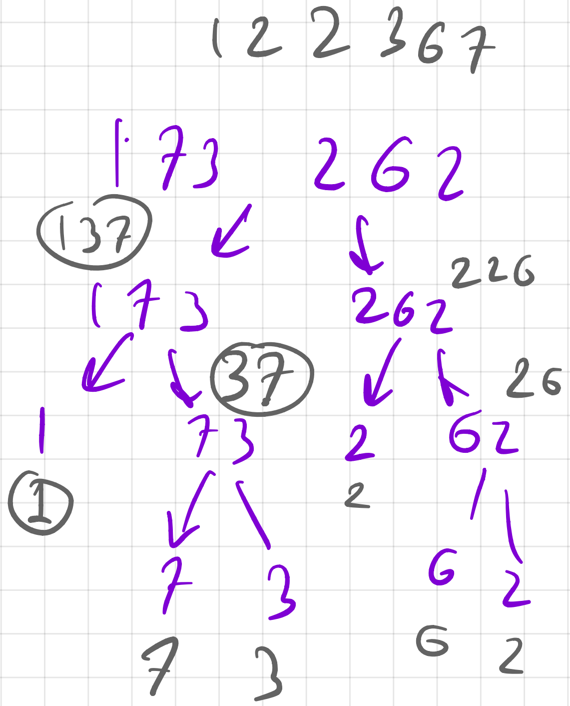
[1 2 2 3 6 7]

merge

```
def MergeSort(a: List[int]) -> List[int]:
    if len(a) <= 1:
        return a
    n = len(a)
    aL = a[0 .. n//2]
    aR = a[n//2 .. n]

    aL = Mergesort(aL)
    aR = mergesort(aR)

    return merge(aL, aR)
```



```
def merge(a: List[int],
          b: List[int]) -> List[int]

    c = []
    i = 0
    j = 0

    while i < len(a) and j < len(b):
        если a[i] < b[j]:
            c.append(a[i])
            i++
        else
            c.append(b[j])
            j++

    c.append(a_i ... a_len(a))
    c.append(b_j ... b_len(b))
```

a = [1 3 7 8 9]

b = [2 2 6]

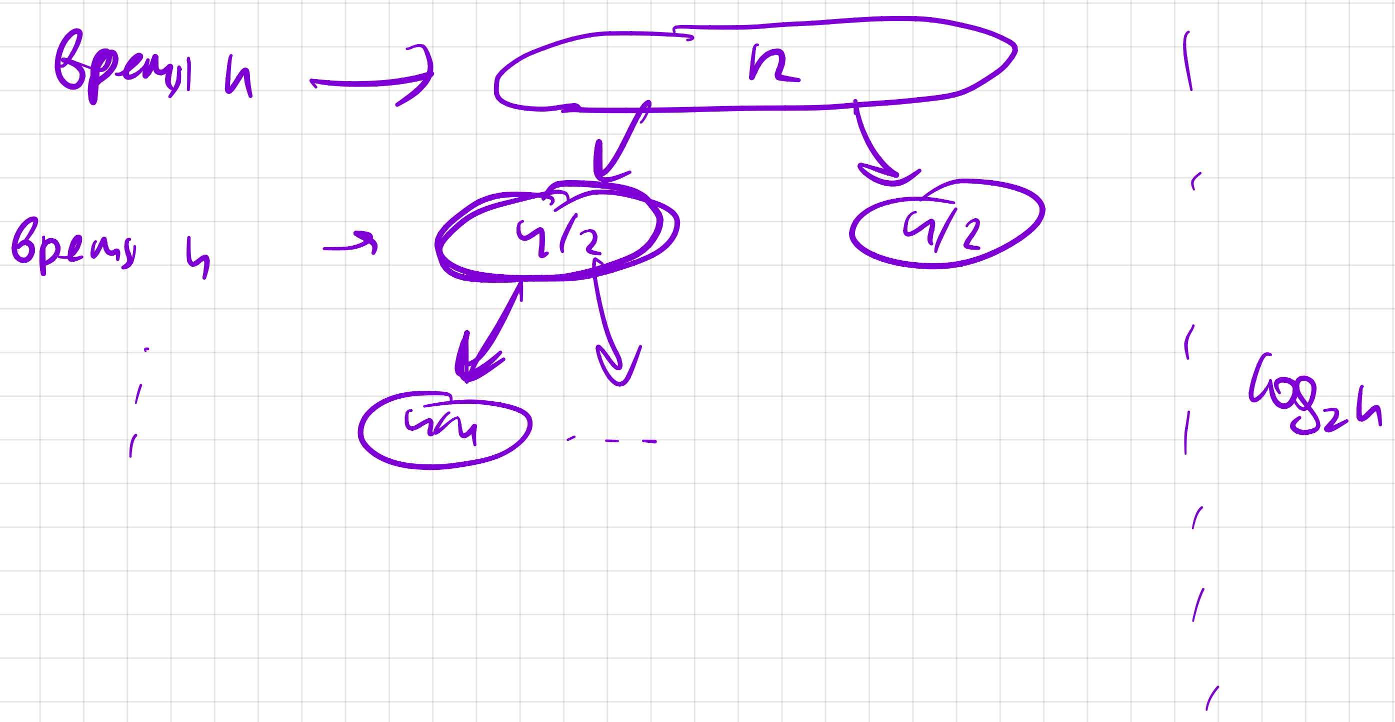
c = [1 2 2 6]

1 из частей пуста

Оценить время работы алгоритмы

$$merge(n, m) = O(n+m)$$

$$mergesort(n) = O(n \log n)$$



①

$$\boxed{mergesort(n) = 2\,mergesort(n/2) + n}$$

$$= O(n \log n)$$

$$mergesort(n) = \begin{cases} -\|- & \text{иначе} \\ C & \text{если } n \le 1 \end{cases}$$

Рекуррентные соотношения

[Мастер Теорема] или Основная теорема о рек. соотн.

~1980

mergesort

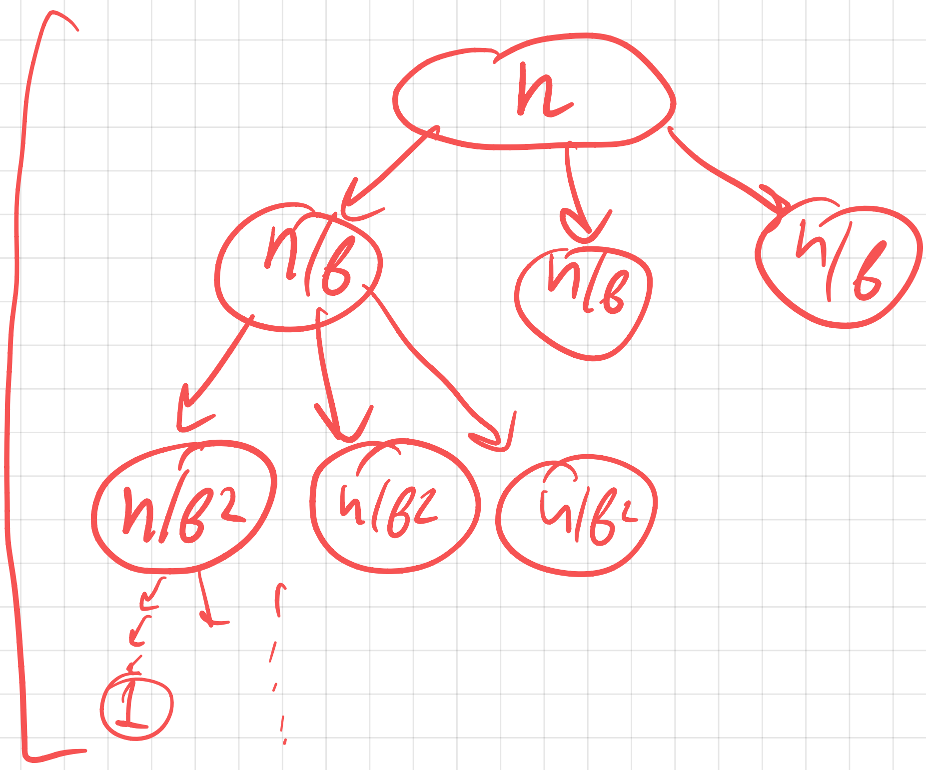
$a = 2$

$b = 2$

$c = 1$

Пусть $T(n) = aT(n/b) + n^c$

Тогда
$$T(n) = \begin{cases} \Theta(n^c \log n) & a = b^c \\ \Theta(n^c) & a < b^c \\ \Theta(n^{\log_b a}) & a > b^c \end{cases}$$



$\log_b n$ раз

1 штук. n^c

a штук $a \cdot \left(\frac{n}{b}\right)^c$

a^2 штук $a^2 \left(\frac{n}{b^2}\right)^c$

$a^{\log_b n} \cdot 1^c$

$$T(n) = \sum_{i=0}^{\log_b n} a^i \left(\frac{n}{b^i}\right)^c$$

$$= n^c \sum_{i=0}^{\log_b n} \frac{a^i}{b^{ic}} =$$

$$= n^c \sum_{i=0}^{\log_b n} \left(\frac{a}{b^c}\right)^i$$

пусть $a = b^c$:

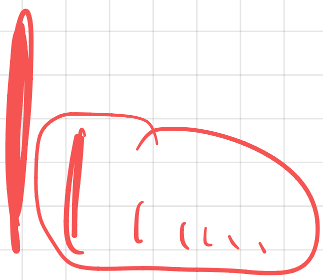
$$\boxed{T(n) = \Theta(n^c \cdot \log n)}$$

Пусть $a < b^c$

$$1 + q + \dots + q^k = \frac{1 - q^{k+1}}{1-q} = \frac{q^{k+1} - 1}{q - 1}$$

Если $a < b^c$, то $q := \frac{a}{b^c} < 1$

$$\lim_{k \to \infty} 1 + q + \dots + q^k = \frac{1}{1-q}$$



$$1 + q + \ldots + q^k = \Theta(1)$$

$$\boxed{T(n) = \Theta(n^c)}$$

Θ по k

это тоже самое что

Θ по n, т.к. $k = \log n$

Пусть $a > b^c$

$$1 + q + \ldots + q^k = \frac{q^{k+1} - 1}{q - 1} = \Theta(q^k)$$

последний уровень

$$T(n) = \Theta\left(a^{\log_b n}\right) =$$

$$= b^{\log_b a \cdot \log_b n} =$$

$$= \boxed{n^{\log_b a}}$$

$$\left(a = b^{\log_b a}\right)$$