Алгоритмы и структуры данных. Лекция 1

Сергей Григорян

4 сентября 2024 г.

1 Инфа

Лектор: Степанов Илья Данилович.

`telegram: @irkstepanov`

2 Основные понятия

Фабула решения задачи

- Условие
- Алгоритм (реализация)
- Корректность
- Асимптотика/Время работы

Элементарные действия

- Сложение, умножения, сравнение чисел;
- Условные конструкции;
- Обращение по индексу (! Большое кол-во = иногда вредно);

Модель вычислений: RAM модель (Random Access Memory)

Замечание. *Бывают и др. модели:*

- *Параллельные вычисления*
- *Внешняя память*

3 Асимптотика, время работы

Пример. *Найти минимум в массиве.*

```
int n;
read(n);
int a[n];
read(a);
int x = +inf;
for i = 0..n-1:
    if (x < a[i]):
        x = a[i]
print(x)
```

Листинг 1: Нахождение минимума

Асимптотика: $O(n)$

Определение 3.1. Пусть $f, g : \mathbb{N} \to \mathbb{N}$. Тогда:

$$f = O(g) \iff \exists N, C\colon \forall n \geq N\colon f(n) \leq C * g(n)$$

Пример.

$$6n + 4 = O(n), 6n + 4 \leq 7n, \textit{ при } n \geq 4.$$

Утверждение 3.1. $f = O(g) \iff \exists D\colon \forall n\colon f(n) \leq D * g(n)$

Доказательство.

$\Leftarrow$) $N = 1\colon n \geq N, C = D, f(n) \leq c * g(n)$

$\Rightarrow$) Надо обеспечить:

$$f(1) \leq Dg(1)$$
$$f(2) \leq Dg(2)$$
$$\vdots$$
$$f(N) \leq Dg(N).$$
$$\Rightarrow D = max(C, \frac{f(1)}{g(1)}, \frac{f(2)}{g(2)}, \cdots, \frac{f(N)}{g(N)})$$

$\square$

Определение 3.2. Пусть $f, g : \mathbb{N} \to \mathbb{N}$. Тогда $f = \Omega(g)$, если $\exists C > 0, N \colon \forall n \geq N$:

$$f(n) \geq C * g(n)$$

Определение 3.3. $f, g : \mathbb{N} \to \mathbb{N}$. Тогда $f = \Theta(g) \iff \exists c_1, c_2 > 0, N \colon \forall n \geq N$:

$$c_1 * g(n) \leq f(n) \leq c_2 * g(n).$$

Пример.

1. $n^a, n^b,\ a, b = const$

$$n^a = O(n^b), a \leq b.$$
$$n^a = \Omega(n^b), a \geq b.$$
$$n^a = \Theta(n^b), a = b.$$

2. $\log_a n = \Theta(\log_b n); a, b = const$

3. $n^n = O(2^{2^n}), 2^{2^n} = \omega(n^n)$

Утверждение 3.2.

$$\log n^a < n^b < c^n, \forall a > 0, b > 0, c > 1.$$

Утверждение 3.3. *Пусть $T(n)$ - время работы влгоритма на входных данных. Пусть:*

$$T(n) = T(\left\lfloor \frac{n}{2} \right\rfloor) + T(\left\lceil \frac{n}{2} \right\rceil) + O(n).$$

Тогда $T(n) = O(n \log n)$

Доказательство. $T(n) \leq T(\left\lfloor \frac{n}{2} \right\rfloor) + T(\left\lceil \frac{n}{2} \right\rceil) + Dn$

Докажем по индукции, что:

$$T(n) \leq C * n \log n, \text{ при } n \geq 2$$

База индукции: $T(2), T(3), \cdots, T(10)$ - Взяли C, чтоб было верно.
Переход: Пусть $T(k) \leq Ck \log_2 k, k \leq n - 1$
Докажем для $k = n$:

$$T(n) \le 2 * T(\left\lceil \frac{n}{2} \right\rceil) + Dn$$

$$\left\lceil \frac{n}{2} \right\rceil \le \frac{n+1}{2} \Rightarrow$$

$$\Rightarrow T(n) \le 2 * T(\left\lceil \frac{n}{2} \right\rceil) + Dn \le$$

$$\le 2 * C * \frac{n+1}{2} \log_2 \frac{n+1}{2} + Dn = C(n+1)(\log_2 (n+1) - 1) + Dn ==$$

$$n + 1 \le n\sqrt{2}$$

$$\Rightarrow \log_2 n + 1 \le \log_2 n\sqrt{2} = \log_2 n + \frac{1}{2}$$

$$== C(n+1)(\log_2 n - \frac{1}{2}) + Dn = Cn \log_2 n - \frac{1}{2}Cn + C \log_2 n - \frac{1}{2}C + Dn \le Cn \log_2 n.$$

Достаточно д-ть, что $Dn + C \log_2 n \le \frac{1}{2}Cn$

Для этого дост. положить $C \ge 6D$:

$$C = 6D.$$

$$Dn + 6D \log_2 n \le 3Dn.$$

$$6 \log_2 n \le 2n.$$

□

4 Бинарный поиск

Задача 4.1. $a_0 \le a_1 \le a_2 \le \dots \le a_{n-1}$

Узнать, есть ли x в a.

Наивное решение: q запросов $\Rightarrow O(nq)$

Решение. *Используем бинпоиск:*

```
int left = 0, right = n;
while (right - left > 1) {
    mid = (left + right) / 2
    if (a[mid] > x) right = mid
    else left = mid
}
```

```
if (a[left] == x) print("Yes");
else print("No");
```

Листинг 2: Binary Search

Асимптотика: $O(\log_2 n)$