

Алгоритмы и структуры данных

Сергей Григорян

11 сентября 2024 г.

Содержание

1	Инфа	3
2	Основные понятия	3
3	Асимптотика, время работы	4
4	Бинарный поиск	6

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.

```
1 int n;  
2 read(n);  
3 int a[n];  
4 read(a);  
5 int x = +inf;  
6 for i = 0..n-1:  
7     if (x < a[i]):  
8         x = a[i]  
9 print(x)
```

Листинг 1: Нахождение минимума

Асимптотика: $O(n)$

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

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

Пример.

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

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

Доказательство.

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

\Rightarrow) Надо обеспечить:

$$f(1) \leq Dg(1)$$

$$f(2) \leq Dg(2)$$

$$\vdots$$

$$f(N) \leq Dg(N).$$

$$\Rightarrow D = \max(C, \frac{f(1)}{g(1)}, \frac{f(2)}{g(2)}, \dots, \frac{f(N)}{g(N)})$$

□

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

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

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

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

Пример. 1. $n^a, n^b, a, b = \text{const}$

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

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

$$n^a = \Theta(n^b), a = b.$$

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

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

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 алгоритма на входных данных. Пусть:

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

$$\text{Тогда } T(n) = O(n \log n)$$

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

Докажем по индукции, что:

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

База индукции: $T(2), T(3), \dots, T(10)$ - Взяли C , чтоб было верно.

Переход: Пусть $T(k) \leq Ck \log_2 k, k \leq n - 1$

Докажем для $k = n$:

$$\begin{aligned}
T(n) &\leq 2 * T\left(\left\lceil \frac{n}{2} \right\rceil\right) + Dn \\
\left\lceil \frac{n}{2} \right\rceil &\leq \frac{n+1}{2} \Rightarrow \\
\Rightarrow T(n) &\leq 2 * T\left(\left\lceil \frac{n}{2} \right\rceil\right) + Dn \leq \\
&\leq 2 * C * \frac{n+1}{2} \log_2 \frac{n+1}{2} + Dn = C(n+1)(\log_2(n+1) - 1) + Dn == \\
&\quad n+1 \leq n\sqrt{2} \\
&\Rightarrow \log_2 n + 1 \leq \log_2 n\sqrt{2} = \log_2 n + \frac{1}{2} \\
== C(n+1)(\log_2 n - \frac{1}{2}) + Dn &= Cn \log_2 n - \frac{1}{2}Cn + C \log_2 n - \frac{1}{2}C + Dn \leq Cn \log_2 n.
\end{aligned}$$

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

Для этого дост. положить $C \geq 6D$:

$$C = 6D.$$

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

$$6 \log_2 n \leq 2n.$$

□

4 Бинарный поиск

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

Узнать, есть ли x в a .

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

Решение. Используем бинарный поиск:

```

1 int left = 0, right = n;
2 while (right - left > 1) {
3     mid = (left + right) / 2
4     if (a[mid] > x) right = mid
5     else left = mid
6 }
```

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

Листинг 2: Binary Search

Асимптотика: $O(\log_2 n)$