

SPb HSE, ПАДИИ, 1 курс, осень 2024/25

Практика по алгоритмам #1

Асимптотики

5 сентября

Собрано 6 сентября 2024 г. в 17:55

---

## Содержание

1. Асимптотики	1
2. Разбор задач практики	2
3. Домашнее задание	3
3.1. Дополнительная часть . . . . .	3

# АСИМПТОТИКИ

## 1. Докажите, что

- Если  $f = \Theta(g)$ ,  $g = \mathcal{O}(h)$ , то  $f = \mathcal{O}(h)$  и  $f \neq \Theta(h)$ .
- Если  $g = \mathcal{O}(f)$ ,  $h = \Theta(f)$ , то  $t = o(f)$ , то  $g + h + t = \Theta(f)$ .

## 2. Дан код, сколько работает?

```

1 for (int i = 0; i < n; i++)
2     for (int j = 0; j < i; j++)
3         for (int k = 0; k < j; k++)
4             ;

```

## 3. Дан код, сколько работает?

```

1 for (int i = 1; i < n; i++)
2     for (int j = 0; j < n; j += i)
3         ;

```

## 4. Рекурренты

За сколько работает программа, если:

- $T(n) = 3T(\frac{n}{3}) + n$
- $T(n) = T(\frac{n}{2}) + n^2$
- $T(n) = 4T(\frac{n}{2}) + n^2$
- $T(n) = 5T(\frac{n}{2}) + n \log n$

## 5. Задча про квадраты

Найдите  $a, b$ :  $a^2 + b^2 = n$ .

- Решите за  $\Theta(n^{1/2})$ ,
- избавьтесь от корней.

## 6. MergeSort без рекурсии

Напишите версия сортировки слиянием без рекурсии.

## 7. (\*) Merge-Sort-Tree

Пусть по ходу рекурсии merge-sort мы сохранили все промежуточные массивы, сколько потрачено памяти?

## 8. (\*) InsertionSort rocks

Пусть мы взяли отсортированный массив и 3 раза сделали swap случайной пары элементов, за сколько работает insertion sort?

## 9. (\*) Merge-Sort-Tree Rocks

Рассмотрим дерево merge-sort-tree, научитесь обрабатывать запрос  $\langle x, L, R \rangle$ .

Выведите все числа  $\leq x$  на отрезке  $[L, R]$  за  $\mathcal{O}(k + \log n)$ , где  $k$  – размер ответа.

## Разбор задач практики

### 1. Докажите, что

$$1. f = \Theta(g) \Rightarrow \exists N_1, C_0, C_1: \forall n \geq N_1 \ C_0 g(n) \leq f(n) \leq C_1 g(n)$$

$$g = \mathcal{O}(h) \Rightarrow \exists N_2, C_2: \forall n \geq N_2 \ g(n) \leq C_2 h(n)$$

$$\Rightarrow \forall n \geq \max(N_1, N_2) \ f(n) \leq C_1 C_2 h(n) \Rightarrow f = \mathcal{O}(h).$$

$$2. g = \mathcal{O}(f) \Rightarrow \exists C > 0: g \leq C \cdot f$$

$$h = \Theta(f) \Rightarrow \exists C_1, C_2 > 0: C_1 \cdot f \leq h \leq C_2 \cdot f$$

$$t = o(f) \Rightarrow \forall C > 0: t \leq C \cdot f, \text{ возъмём } C = 1 \Rightarrow$$

$$(0 + C_1 + 0) \cdot f \leq g + h + t \leq (C + C_2 + 1) \cdot f$$

### 2. Дан код, сколько работает?

За  $\Theta(n^3)$ : точно не больше  $n^3$ , точно не меньше  $(\frac{n}{3})^3$   
 (учитываем только  $0 \leq k < \frac{n}{3} \leq j < \frac{2n}{3} \leq i < n$ ).

### 3. Дан код, сколько работает?

$$T(n) = \sum_i \frac{n}{i} = n \cdot \sum_i \frac{1}{i} = n \cdot \Theta(\log n) = \Theta(n \log n).$$

Почему  $\log n$ ? См. конспект ПМИ.

## Домашнее задание

### 1. (2+2) Докажите

Рассмотрим код.

```
1   for (int i = 1; i < n; i *= 2)
2       for (int j = 1; j < i; j++)
3           f(n, i, j);
```

Здесь  $f$  – какая-то функция, работающая за время  $\Theta(n^k)$ .

- Оцените время работы, зависимость от  $n$ .
- Докажите время работы формально через определения.

### 3.1. Дополнительная часть

#### 1. (2) Рекуррентные соотношения

Оцените асимптотику:  $T(n) = nT(\frac{n}{2}) + n$ ,  $T(1) = 1$ .