

# Algorithms and data structures

lecture #4. Master Theorem

Mentor: Il'yas Miftakhov

## lecture #4. Master Theorem

- Master Theorem
  - Описание
  - Общая форма
  - Применение

# Master Theorem (Основная теорема)

**Основная теорема** о рекуррентных соотношениях (Master theorem) используется в анализе алгоритмов для получения асимптотической оценки рекурсивных при анализе алгоритмов типа «разделяй и властвуй» (divide and conquer).

Теорема была введена и доказана Джоном Бентли, Доротеном Хакеном и Джеймсом Хакеном в 1980 году.

*Основная теорема о рекуррентных соотношениях — это формула, предназначенная для решения рекуррентных соотношений следующего вида:*

$$T(n) = aT(n/b) + f(n), \text{ где}$$

$n$  = объем входных данных

$a$  = количество подзадач в рекурсии

$n/b$  = размер каждой подзадачи. Предполагается, что все подзадачи имеют одинаковый размер.

$f(n)$  = оценка выполненной работы вне рекурсивных вызовов.

## Формулировка

Если  $a \geq 1$  и  $b > 1$  — константы, а  $f(n)$  — асимптотически положительная функция, то временная сложность рекуррентного соотношения задается выражением:

$$T(n) = aT(n/b) + f(n)$$

$T(n)$  имеет следующие асимптотические оценки:

1. Если  $f(n) = O(n \log b a^{-\epsilon})$ , то  $T(n) = \Theta(n \log b a)$
2. Если  $f(n) = \Theta(n \log b a)$ , то  $T(n) = \Theta(n \log b a \cdot \log n)$
3. Если  $f(n) = \Omega(n \log b a^{+\epsilon})$ , то  $T(n) = \Theta(f(n))$

$\epsilon > 0$  — константа.

## Пример использования

$$T(n) = 3T(n/2) + n^2$$

Что есть, что:

$$a = 3$$

$$n/b = n/2$$

$$f(n) = n^2$$

$$\log_b a = \log_2 3 \approx < 2$$

то есть  $f(n) < n \log_b a + \epsilon$ , где  $\epsilon$  — константа.

То есть, это третья оценка.

$$T(n) = f(n) = \Theta(n^2)$$

Двоичный поиск –

$$T(n) = T(n/2) + O(1) = O(\log n) \text{ – вторая оценка}$$

Сортировка слиянием –

$$T(n) = 2T(n/2) + O(n) = O(n \log n) \text{ – вторая оценка}$$

## Когда не работает

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

$a$  не является константой, для основной теоремы требуется постоянное количество подзадач;

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

между  $f(n)$  и  $n^{\log_b a}$  существует неполиномиальная зависимость;

- $T(n) = 0.5T\left(\frac{n}{2}\right) + n$

$a < 1$ , но основная теорема требует наличия хотя бы одной подзадачи;

- $T(n) = 64T\left(\frac{n}{8}\right) - n^2 \log n$

$f(n)$  является отрицательной величиной;