A2. Применение мастер-теоремы

Демченко Георгий Павлович , БПИ-235

Условие

$\forall\, T(n) : n = 1 \Rightarrow O(1)$

1. $T(n) = O(g(n))$

1.

- $T(n) = 7 \cdot T(\frac{n}{3}) + n^2 = 7 \cdot T(\frac{n}{3}) + O(n^2)$

$a = 7 \geq 1$

$b = 3 > 1$

$n^k = n^2, k = 1 > 0$

$f(n) = 1$ - константа

$log_b(a) = log_3(7) < 2 = k$

$\Rightarrow T(n) = O(n^2)$

2.

- $T(n) = 4 \cdot T(\frac{n}{2}) + log_2(n) = 4 \cdot T(\frac{n}{2}) + O(n)$

$a = 4 \geq 1$

$b = 2 > 1$

$n^k = n^1, k = 1 > 0$

$f(n) = 1$ - константа

$log_b(a) = log_2(4) = 2 > 1 = k$

$\Rightarrow T(n) = O(n^2)$

3.

- $T(n) = \frac{1}{2} \cdot T(\frac{n}{2}) + \frac{1}{n}$

$a = \frac{1}{2} < 1$

$\Rightarrow$ Применение мастер-теоремы невозможно

4.

- $T(n) = 3 \cdot T(\frac{n}{3}) + \frac{n}{2} = 3 \cdot T(\frac{n}{3}) + O(n)$

$a = 3 \geq 1$

$b = 3 > 1$

$n^k = n^1, k = 1 > 0$

$f(n) = 1$ - константа

$log_b(a) = log_3(3) = 1 = k$

$\Rightarrow T(n) = O(n \cdot log_2(n))$

5.

- $T(n) = T(n-1) + T(n-2) + n \cdot log_2(n) = T(n-1) + T(n-2) + O(n \cdot log_2(n))$

$\Rightarrow$ **Применение мастер-теоремы невозможно в виду неявной глубины рекурсии, т.е разных b**

2. $T(n) = O(f(n))$ для неразрешимых мастер-теоремой

- Под методом итерации понят метод дерева рекурсии.

1. $T(n) = \frac{1}{2} \cdot T(\frac{n}{2}) + \frac{1}{n}$

$m = log_2(n)$ - глубина рекурсии

$(\frac{1}{2})^i$ - количество подзадач на i-ом уровне рекурсии (если так вообще можно сказать при данном значении)

$\frac{1}{\frac{n}{2^i}} = \frac{2^i}{n}$ - затраты на подзадачи на i-ом уровне рекурсии

$\Rightarrow T(n) = \sum_{i=0}^{log_2(n)}(\frac{1}{2^i} \cdot \frac{2^i}{n}) = \sum_{i=0}^{log_2(n)}(\frac{1}{n}) = \frac{1}{n} \cdot \sum_{i=0}^{log_2(n)}(1) = \frac{1}{n} \cdot (log_2(n) + 1) = \frac{log_2(n)}{n} + \frac{1}{n} = O(\frac{log_2(n)}{n})$

$\Rightarrow T(n) = O(\frac{log_2(n)}{n})(\text{странно, но математически верно}) = O(log_2 n)$

2. $T(n) = T(n-1) + T(n-2) + n \cdot log_2(n)$

$\Rightarrow T(n) = T(n-1) + T(n-2) + n \cdot log_2(n) \leq 2 \cdot T(n-1) + n \cdot log_2(n)$

Так как в данном случае имеем максимальную глубину рекурсии и наибольшие затраты на подзадачи на уровне рекурсии

Предположим, что $T(n) = O(2^n \cdot n \cdot log_2(n))$

$\Rightarrow \exists\, d_1 \in \mathbb{R}^+ : \forall \mathbb{N} > N_0 : T(n) \leq d_1 \cdot 2^n \cdot n \cdot log_2(n)$

Пусть $N_0 = 3$

$T(n) = 2 \cdot T(n-1) + n \cdot log_2(n) \leq 2 \cdot d_1 \cdot 2^{n-1} \cdot (n-1) \cdot log_2(n-1) + n \cdot log_2(n) \leq d_1 \cdot 2^n \cdot n \cdot log_2(n)$

$d_1 \cdot (2^n \cdot n \cdot log_2(n) - 2^n \cdot (n-1) \cdot log_2(n-1)) \geq n \cdot log_2(n)$

$d_1 \cdot 2^n(n \cdot log_2(n) - n \cdot log_2(n-1) + log_2(n-1)) \geq n \cdot log_2(n)$

$\forall \mathbb{N} > N_0 (= 3)$

$d_1 \cdot 2^n(n \cdot log_2(n) - n \cdot log_2(n-1) + log_2(n-1)) \geq d_1 \cdot 2^n(n \cdot log_2(n) - n \cdot log_2(n-1)) \geq n \cdot log_2(n)$

$\Rightarrow d_1 \cdot 2^n \cdot n \cdot log_2(\frac{n}{n-1}) \geq n \cdot log_2(n)$

$d_1 \geq \frac{1}{2^n} \cdot \frac{log_2(n)}{log_2(\frac{n}{n-1})}$

$d_1 \geq 1 \in \mathbb{R}^+$

$\Rightarrow T(n) = O(2^n \cdot n \cdot log_2(n))$