

Алгоритмы и модели вычислений.

Задание 1: Алгоритмы и оценка сложности

Сергей Володин, 272 гр.

задано 2014.02.13

(каноническое) Задача 1

$$f(n) \stackrel{\text{def}}{=} \sum_{i=1}^n \frac{1}{i}, g(n) = \log n. \text{ Доказать: } f = \Theta(g) \Leftrightarrow \begin{cases} f = O(g) \\ g = O(f) \end{cases} \Leftrightarrow \begin{cases} \exists C_1, n_1: \forall n \geq n_1 \hookrightarrow f(n) \leq C_1 g(n) & (1) \\ \exists C_2, n_2: \forall n \geq n_2 \hookrightarrow g(n) \leq C_2 f(n) & (2) \end{cases}$$

1. Докажем утверждение: пусть $f(n), g(n): \exists n_0, C_1 > 0: \forall n \geq n_0 \hookrightarrow \underbrace{f(n+1) - f(n)}_{\Delta_f(n)} \leq C_1 \underbrace{g(n+1) - g(n)}_{\Delta_g(n)}$. Тогда $f = O(g)$. Действительно, выберем $C_2 > 0$ таким образом, что $f(n_0) \leq C_2 g(n_0)$ (всегда можно сделать). Возьмем C для определения O как $C = \max(C_1, C_2)$. Докажем по индукции $\forall n \geq n_0 \hookrightarrow f(n) \leq C g(n)$:

(a) $f(n_0) \leq C_2 g(n_0) \leq C g(n_0)$ ■

(b) Пусть $f(n) \leq C g(n)$. Докажем для $n+1$: по условию $\Delta_f(n) = f(n+1) - f(n) \leq C_1 (g(n+1) - g(n)) \leq C (g(n+1) - g(n))$.
Перегруппируем, получим $f(n+1) - C g(n+1) \leq f(n) - C g(n) \stackrel{\text{предп.}}{\leq} 0$, т.е. $f(n+1) \leq C g(n+1)$ ■

2. Докажем (1).

(a) $\nexists \Delta_f(n) \stackrel{\text{def}}{=} f(n+1) - f(n) = \frac{1}{n+1} \leq \frac{1}{n}$.

(b) $\nexists \Delta_g(n) \stackrel{\text{def}}{=} g(n+1) - g(n) = \log(n+1) - \log n = \log \frac{n+1}{n} = \log(1 + \frac{1}{n}) = \frac{1}{n} + \bar{o}(\frac{1}{n}) = \boxed{*}$, $n \rightarrow \infty$. Но по определению $\bar{o} \exists n_1: \forall n \geq n_1 \hookrightarrow \boxed{*} \geq \frac{1}{n}(1 - \frac{1}{2}) = \frac{1}{2} \frac{1}{n}$. Тогда $\frac{1}{n} \leq 2 \boxed{*} = 2(g(n+1) - g(n))$

(c) Получаем $\Delta_f(n) = f(n+1) - f(n) \stackrel{2a}{\leq} \frac{1}{n} \stackrel{2b}{\leq} 2(g(n+1) - g(n)) = 2\Delta_g(n)$, и по 1 получаем $f = O(g)$.

3. Докажем (2).

(a) $\nexists \Delta_f(n) = \frac{1}{n+1}$. Докажем, что это больше, чем $\frac{1}{2} \frac{1}{n}$: $\frac{1}{n+1} - \frac{1}{2} \frac{1}{n} = \frac{2n-n-1}{2n(n+1)} = \frac{n-1}{2n(n+1)} \geq 0, n \geq 1$. Итак, $\Delta_f(n) \geq \frac{1}{2} \frac{1}{n}$

(b) $2b \Rightarrow \Delta_g(n) = \frac{1}{n} + \bar{o}(\frac{1}{n}) \leq \frac{1}{n}(1 + \frac{1}{2})$ при $n \geq n_2 > 1$. Значит, $\frac{3}{2} \frac{1}{n} \geq \Delta_g(n)$

(c) $\Delta_g(n) \stackrel{3b}{\leq} \frac{3}{2} \frac{1}{n} \stackrel{3a}{\leq} \frac{3}{2} \cdot 2 \cdot \Delta_f(n)$ при $n \geq n_2$, и по 1 получаем $g = O(f)$.

(каноническое) Задача 2

(каноническое) Задача 3

1. $T(n) = 9T(\frac{n}{3}) + f(n), f(n) = \Theta(n^2 \log n)$.

(a) Докажем, что теорема неприменима. $a = 9, b = 3 \Rightarrow \log_b a = \log_3 9 = 2$.

- i. Если $\exists \varepsilon > 0: f(n) = O(n^{2-\varepsilon})$, то $\exists C > 0 \exists n_0$, для $n \geq n_0$ получим $f(n)/n^{2-\varepsilon} \leq C > 0$, то есть $n^2 \log n / n^{2-\varepsilon} \equiv n^\varepsilon \log n \leq C$, что неверно (функция неограничена сверху).
- ii. Если $f = \Theta(n^2)$, то $\exists n_0 \exists C > 0: f \leq C n^2$ для $n \geq n_0$, и $\log n \leq C$, что неверно (функция неограничена сверху).
- iii. Если $\exists \varepsilon > 0: f = \Omega(n^{2+\varepsilon})$, то $\exists n_0: \forall n \geq n_0 \hookrightarrow f \geq C n^{2+\varepsilon}$, и $\log n \geq C n^\varepsilon$, откуда $\frac{\log n}{n^\varepsilon} \geq C > 0$, что неверно, так как $\forall \varepsilon > 0 \hookrightarrow \lim_{n \rightarrow \infty} \frac{\log n}{n^\varepsilon} = +0$

(b) Найдем ответ через дерево рекурсии. В корне ($i = 0$) выполняется $n^2 \log n$ операций, у каждой вершины 9 детей, на уровне $i + 1$ $n_{i+1} = n_i/3$. У листьев (по индукции по высоте дерева) $1 = n_h = \frac{n}{3^h}$, поэтому высота дерева (не считая корня) $h = \log_3 n$. Найдем суммарное время:

$$T(n) = \Theta(n^2 \log n + 9(\frac{n}{3})^2 \log \frac{n}{3} + 9^2(\frac{n}{3^2})^2 \log \frac{n}{3^2} + \dots + 9^{h-1}(\frac{n}{3^{h-1}})^2 \log \frac{n}{3^{h-1}}) + 9^h T(1)$$

$$\text{Найдем сумму в аргументе: } \Theta: \sum_{i=0}^{h-1} 9^i (\frac{n}{3^i})^2 \log \frac{n}{3^i} = n^2 \sum_{i=0}^{h-1} (\log n - i \log 3) = n^2 \log n (h-1) - n^2 \frac{h-1}{2} \log 3 = \\ = n^2 \log n (\log_3 n - 1) - n^2 \frac{\log_3 n - 1}{2} \log 3 = n^2 \log^2 n - n^2 \log n - n^2 \log n + C n^2 = \Theta(n^2 \log^2 n).$$

$$\text{Найдем } 9^h T(1) = C 9^{\log_3 n} = C n^2. \text{ Имеем } T(n) = \Theta(n^2 \log^2 n) + C n^2 = \boxed{\Theta(n^2 \log^2 n)}$$

2. $T(n) = 16T(\frac{n}{4}) + f(n), f(n) = \Theta(n^2)$. $a = 16, b = 4$. Применим второй пункт Теоремы: $\Theta(n^{\log_b a}) \equiv \Theta(n^2)$, поэтому $f(n) = \Theta(n^{\log_b a})$, и отсюда $T(n) = \boxed{\Theta(n^2 \log n)}$.

3. $T(n) = 4T(\frac{n}{2}) + \underbrace{\Theta(\frac{n^2\sqrt{n}}{\log^2 n})}_{g(n)}$. $a = 4, b = 2 \Rightarrow \log_b a = 2$. Возьмем $\varepsilon = \frac{1}{4}$ и применим третий пункт Теоремы: $f(n) \stackrel{?}{=} \Omega(n^{2+\varepsilon})$.

Рассмотрим $\frac{f(n)}{n^{2+\varepsilon}} = \frac{n^2\sqrt{n}}{n^{2+\varepsilon}\log^2 n} = \frac{n^{\frac{1}{2}-\varepsilon}}{\log^2 n} = \frac{n^{1/4}}{\log^2 n} = (\frac{n^{1/8}}{\log n})^2 \xrightarrow{n \rightarrow \infty} +\infty$, поэтому $\exists C > 0 \exists n_0 > 0: \forall n \geq n_0 \hookrightarrow f(n) \geq Cn^{2+\varepsilon}$.
Докажем, что $\exists 0 < C < 1 \exists n_1: af(n/b) \leq Cf(n)$. $f = \Theta(g) \Rightarrow \exists n_2: \forall n \geq n_2 \hookrightarrow C_1g(n) \leq f(n) \leq C_2g(n)$. Тогда $af(\frac{n}{b}) \leq 4C_2\frac{(\frac{n}{2})^{\frac{5}{2}}}{\log^2(\frac{n}{2})} = \frac{C_2}{\sqrt{2}C_1} \frac{\log^2 n}{\log^2(\frac{n}{2})} C_1 \frac{n^2\sqrt{n}}{\log^2 n} \leq \frac{C_2}{\sqrt{2}C_1} \frac{\log^2 n}{\log^2(\frac{n}{2})} f(n)$. Значит, оценка верна, и по теореме получаем $T(n) = \boxed{\Theta(\frac{n^{5/2}}{\log^2 n})}$

Сравним первую и вторую функции: $\frac{n^2\log^2 n}{n^2\log n} = \log n \xrightarrow{n \rightarrow \infty} +\infty$, поэтому первый алгоритм хуже. Сравним вторую и третью функции: $\frac{n^2\sqrt{n}}{\log^2 n} \frac{1}{n^2\log n} = \frac{n^{1/2}}{\log^3 n} = (\frac{n^{1/6}}{\log n})^3 \xrightarrow{n \rightarrow \infty} +\infty$, поэтому третий алгоритм хуже.

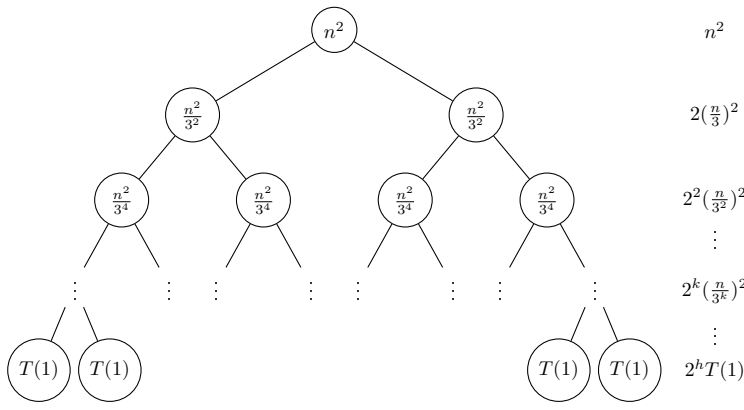
Ответ: второй алгоритм имеет наименьшую асимптотическую стоимость.

(каноническое) Задача 4

(каноническое) Задача 5

Задача 1

1. $T(n) = 2T(\frac{n}{3}) + f(n)$, $f(n) = \Theta(n^2)$. Дерево рекурсии:



Рассмотрим рекуррентность. Последовательно подставляя $T(n)$ в правую часть, получим некоторую сумму $T(n) = \sum_{i=0}^{h-1} C_i \cdot f(\frac{n}{3^i}) + C_h T(1)$. Она конечна, так как аргумент $T(\cdot)$ в правой части меньше, чем в левой, причем в 3 раза. Прекращаем подставлять, когда аргумент станет равен 1. C_i — некоторые коэффициенты, найти которые можно при помощи дерева слева. Корень соответствует $i = 0$ (база), та, каждый i -й уровень соответствует i -му слагаемому суммы. **ДОПИСАТЬ ФОРМАЛЬНО** При последней, h -й подстановке $\frac{n}{3^h} = 1$, откуда $h = \log_3 n$.

Таким образом, $T(n) = \sum_{k=0}^{h-1} \overbrace{2^k f(\frac{n^2}{3^{2k}})}^S + 2^h T(1)$.

(а) Обозначим $g(n) = n^2$, по условию $f(n) = \Theta(g(n))$. Из определения Θ получаем

$$\exists n_0 > 0, C_2 > C_1 > 0: \forall n \geq n_0 \hookrightarrow C_1 g(n) \leq f(n) \leq C_2 g(n)$$

. Рассмотрим первую сумму S при $n \geq n_0$:

$$n^2 C_1 \sum_{k=0}^{h-1} \frac{2^k}{3^{2k}} \leq \sum_{k=0}^{h-1} \overbrace{2^k f(\frac{n^2}{3^{2k}})}^S \leq n^2 C_2 \sum_{k=0}^{h-1} \frac{2^k}{3^{2k}} \quad (1)$$

Рассмотрим $S_1(n) \stackrel{\text{def}}{=} \sum_{k=0}^{h-1} \frac{2^k}{3^{2k}} \stackrel{\text{геом. прогр.}}{=} \frac{1 - \frac{2^{h-1}}{9^{h-1}}}{1 - \frac{2}{9}} \xrightarrow{n \rightarrow \infty} \frac{1}{1 - 2/9} = \frac{9}{7} \stackrel{\text{def}}{=} l$. Здесь использовалось $h = \log_3 n \xrightarrow{n \rightarrow \infty} +\infty$. Определение предела:

$$\forall \varepsilon > 0 \exists n_1(\varepsilon): \forall n \geq n_1 \hookrightarrow S_1(n) \in U_\varepsilon(l)$$

Фиксируем $\varepsilon = \varepsilon_0 = l/2$, определим $n \stackrel{\text{def}}{=} \max n_2(\varepsilon_0), n_0$. Тогда $\forall n \geq n_2 \hookrightarrow 0 < l - \varepsilon \leq S_1(n) \leq l + \varepsilon$.

Снова рассмотрим (1): при $n \geq n_2$: $n^2 C_1 (l - \varepsilon) \leq n^2 C_1 \sum_{k=0}^{h-1} \frac{2^k}{3^{2k}} \leq \sum_{k=0}^{h-1} \underline{2^k f(\frac{n^2}{3^{2k}})} \leq n^2 C_2 \sum_{k=0}^{h-1} \frac{2^k}{3^{2k}} \leq n^2 C_2 (l + \varepsilon)$. Получаем

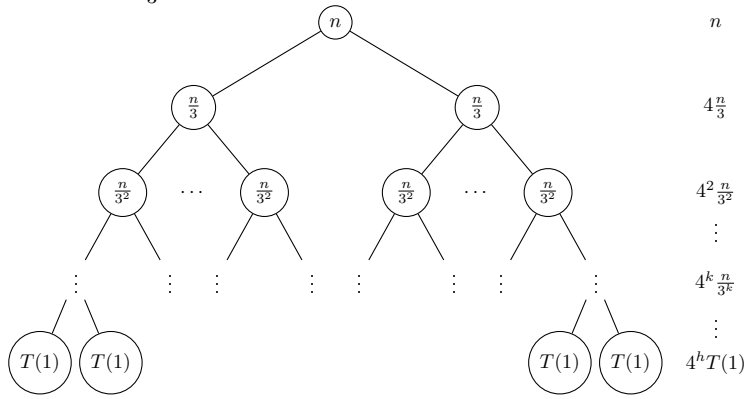
$$S(n) = \Theta(n^2)$$

(b) Рассмотрим $2^h T(1) = 2^{\log_3 n} T(1) = n^{\log_3 2} \underbrace{T(1)}_{\text{const}} = \Theta(n^{\log_3 2})$

(c) Получаем $T(n) = \Theta(n^2) + \Theta(n^{\log_3 2}) = \Theta(n^2)$. Доказательство последнего равенства в конце работы (1) ($2 > 1 > \log_3 2$, поэтому $n^{\log_3 2} = \bar{o}(n^2)$)

Ответ: $T(n) = \Theta(n^2)$

3. $T(n) = 4T(\frac{n}{3}) + f(n)$, $f(n) = \Omega(n)$. Дерево рекурсии (все ветвления не показаны):



Высота дерева $h = \log_4 n$, $T(n) = \sum_{k=0}^{h-1} 4^k f(\frac{n}{3^k}) + 4^h T(1)$. Из определения Ω $\exists n_0 \exists C > 0: \forall n \geq n_0 \Leftrightarrow \sum_{k=0}^{h-1} 4^k f(\frac{n}{3^k}) \geq Cn \sum_{k=0}^{h-1} \frac{4^k}{3^k} \stackrel{\text{геом. прогр.}}{=} Cn \frac{(4/3)^{h-1} - 1}{4/3 - 1}$

Задача 2

Задача 3

Вспомогательные утверждения

- Пусть $f_1 = \Theta(g_1)$, $f_2 = \Theta(g_2)$, $g_2 = \bar{o}(g_1)$, $g_2(n) > 0$. Тогда $f_1 + f_2 = \Theta(g_1)$. Доказательство:
Из определения Θ получаем $\exists n_0 \exists C_i^j > 0, (i, j) \in \overline{1, 2}^2: \forall n \geq n_0 \left\{ \begin{array}{l} C_1^1 g_1(n) \leq f_1(n) \leq C_2^1 g_1(n) \\ C_1^2 g_2(n) \leq f_2(n) \leq C_2^2 g_2(n) \end{array} \right.$ (n_0 — максимальное из двух определений). Тогда

$$C_1^1 \overset{n \rightarrow \infty}{\leftarrow} C_1^1 + C_1^2 \frac{g_2(n)}{g_1(n)} \overset{0}{\nearrow} = \frac{C_1^1 g_1(n) + C_1^2 g_2(n)}{g_1(n)} \leq \frac{f_1(n) + f_2(n)}{g_1(n)} \leq \frac{C_2^1 g_1(n) + C_2^2 g_2(n)}{g_1(n)} = C_2^1 + C_2^2 \frac{g_2(n)}{g_1(n)} \overset{0}{\nearrow} \overset{n \rightarrow \infty}{\rightarrow} C_2^1$$

Здесь использовалось определение \bar{o} . Из определения предела для $\varepsilon = \varepsilon_0 = \min(C_1^1, C_2^1)/2$ получаем при $n \geq n_0(\varepsilon)$ $(C_1^1 - \varepsilon)g_1(n) \leq f_1(n) + f_2(n) \leq (C_2^1 + \varepsilon)g_1(n)$, а из этого следует $f_1 + f_2 = \bar{o}(g_1)$ ■