

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

(b) $\Delta_g(n) \stackrel{\text{def}}{=} g(n+1) - g(n) = \log(n+1) - \log n = \log \frac{n+1}{n} = \log(1 + \frac{1}{n}) = \frac{1}{n} + \bar{o}(\frac{1}{n}) = \boxed{*}$, $n \rightarrow \infty$. 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) $\Delta_f(n) = \frac{1}{n+1}$. 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

$f(n) \stackrel{\text{def}}{=} C_{2n}^n \equiv \frac{(2n)!}{n!n!}$. Формула Стирлинга: $n! = \Theta(\sqrt{2\pi n}(\frac{n}{e})^n) = \Theta(\sqrt{n}(\frac{n}{e})^n)$, поэтому $f(n) \equiv \frac{(2n)!}{(n!)^2} = \Theta(\frac{\sqrt{2n}(\frac{2n}{e})^{2n}}{n(\frac{n}{e})^{2n}}) = \Theta(\frac{4^n}{\sqrt{n}})$

Ответ: $C_{2n}^n = \Theta(\frac{4^n}{\sqrt{n}})$

Попытка не через формулу Стирлинга (не дописано): Рассмотрим $f(n) = \frac{(2n)!}{n!n!} = \frac{(2n)(2n-1)(2n-2)!}{n^2(n-1)!(n-1)!} = (4 - \frac{2}{n})f(n-1)$. Таким образом, $\frac{f(n)}{f(n-1)} = 4 - \frac{2}{n}$. Определим $g(n) = \frac{4^n}{\sqrt{n}}$, докажем, что $f = \Theta(g)$. Рассмотрим $\frac{g(n)}{g(n-1)} = \frac{4\sqrt{n-1}}{\sqrt{n}} = 4\sqrt{1 - \frac{1}{n}} \equiv \boxed{=}$. По формуле Тейлора $\boxed{=} 4(1 - \frac{1}{2n} + \bar{o}(\frac{1}{n})) = 4 - \frac{2}{n} + \bar{o}(\frac{1}{n})$

Получаем, что $\frac{f(n)}{f(n-1)} - \frac{g(n)}{g(n-1)} = \bar{o}(\frac{1}{n})$

(каноническое) Задача 3

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

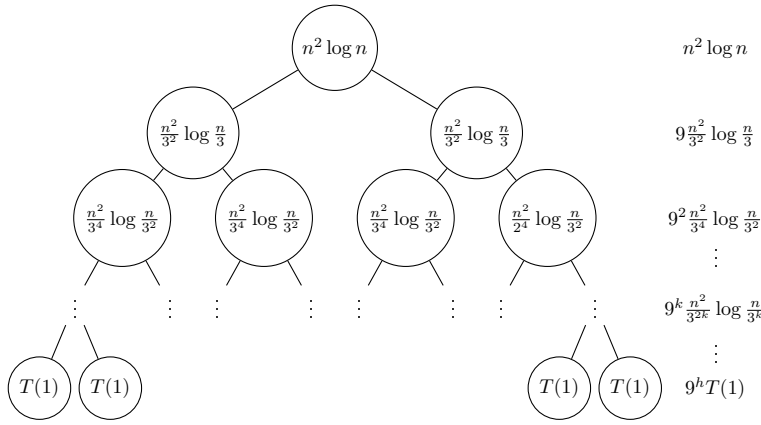
(a) Докажем, что теорема неприменима. $a = 9, b = 3 \Rightarrow \log_b a = \log_3 9 = 2$.

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 $\exists n_0 \exists C > 0: f \leq Cn^2$ 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 $\exists n_0: \forall n \geq n_0 \hookrightarrow f \geq Cn^{2+\varepsilon}$, и $\log n \geq Cn^\varepsilon$, 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 (C_1, C_2 — из определения Θ для f , под записью « $\in [C_1, C_2]$ » подразумеваются два неравенства):

$$T(n) = \sum_{i=0}^{h-1} 9^i f\left(\frac{n}{3^i}\right) + 9^h T(1) \in [C_1, C_2] \left(n^2 \log n + 9 \left(\frac{n}{3}\right)^2 \log \frac{n}{3} + 9^2 \left(\frac{n}{3^2}\right)^2 \log \frac{n}{3^2} + \dots + 9^{h-1} \left(\frac{n}{3^{h-1}}\right)^2 \log \frac{n}{3^{h-1}} + 9^h T(1) \right)$$

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

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

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

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

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: $\frac{n^2 \sqrt{n}}{\log^2 n} \frac{1}{n^2 \log n} = \frac{n^{1/2}}{\log^3 n} = \left(\frac{n^{1/6}}{\log n}\right)^3 \xrightarrow{n \rightarrow \infty} +\infty$, поэтому третий алгоритм хуже.

Ответ: второй алгоритм имеет наименьшую асимптотическую стоимость.

(каноническое) Задача 4

1. $T(n) = 2T(\frac{n}{2}) + \underbrace{n}_{f(n)}$. Воспользуемся пунктом (2) Теоремы: $\log_b a = \log_2 2 = 1$, поэтому $f(n) \equiv n = \Theta(n^{\log_b a}) \equiv \Theta(n)$.

Ответ: $T(n) = \Theta(n \log n)$.

2. $T(n) = 3T(\frac{n}{3}) + \underbrace{n^2}_{f(n)}$. Воспользуемся пунктом (3) Теоремы: $\log_b a = 1, \lim_{n \rightarrow \infty} \frac{f(n)}{n^{\log_b a + \varepsilon}} = \lim_{n \rightarrow \infty} n^{1-\varepsilon} = +\infty$ например при $\varepsilon = \frac{1}{2}$. Поэтому из определения предела для $\varepsilon_{\lim} = 1 \exists n_0 > 0: \forall n \geq n_0 \hookrightarrow f(n) \geq \varepsilon_{\lim} n^{1+\varepsilon}$, значит, $f(n) = \Omega(n^{1+\varepsilon})$.
Докажем условие регулярности: $af(\frac{n}{b}) \equiv 2 \frac{n^2}{2^2} = \frac{1}{2} n^2 = \frac{1}{2} f(n) \leq \frac{1}{2} f(n)$, т.е. условие выполняется с $c = \frac{1}{2} < 1$.

Ответ: $T(n) = \Theta(n^2)$

3. $T(n) = 4T(\frac{n}{2}) + \underbrace{\frac{n}{\log n}}_{f(n)}$. Воспользуемся пунктом (1) Теоремы: $\log_b a = \log_2 4 = 2$.

Рассмотрим $\lim_{n \rightarrow \infty} \frac{f(n)}{n^{\log_b a - \varepsilon}} = \lim_{n \rightarrow \infty} \frac{1}{n^{1-\varepsilon} \log n} = 0$ например при $\varepsilon = \frac{1}{2}$. Из определения предела для

$$\varepsilon_{\lim} = 1 \exists n_0: \forall n \geq n_0 \hookrightarrow f(n) \leq \varepsilon_{\lim} n^{2-\varepsilon},$$

откуда следует $f(n) = O(n^{2-\varepsilon})$.

Ответ: $T(n) = \Theta(n^2)$

(каноническое) Задача 5

$$M(m) \stackrel{\text{def}}{=} Mult(m), A(m) \stackrel{\text{def}}{=} Add(m).$$

Элементарная битовая операция — конъюнкция, дизъюнкция, сложение, умножение двух бит.

Описание алгоритма. Пусть даны числа $p = a + bx$, $q = c + dx$. Пусть числа a, b, c, d — m -битные, $x = 2^m$. Требуется найти pq .

Но $pq = (a + bx)(c + dx) = ac + x(ad + bc) + bdx^2$. Рассмотрим $\begin{cases} t_1 = ac \\ t_2 = bd \\ t_3 = (a + b)(c + d) \end{cases}$. Тогда $pq = t_1 + (t_3 - t_1 - t_2)x + t_2x^2$.

1. Для получения t_i необходимо 2 умножения чисел по m бит, одно умножение чисел по $m + 1$ бит, два сложения чисел по m бит: $2M(m) + M(m + 1) + A(m)$. Для вычисления pq таким образом требуется еще два сложения чисел длиной менее $m + 1$ и битовые сдвиги (их не считаем). Получаем $M(2m) = 2M(m) + M(m + 1) + A(m) + 2A(m + 1)$.

2. Докажем, что $A(m + 1) = A(m)$: пусть нужно сложить числа p и q по $m + 1$ бит. Представим их в виде $p = a_1 + t_1x$, $q = a_2 + t_2x$, где $x = 2^m$, и t_i — соответствующие старшие биты. Полусим $p + q = (a_1 + a_2) + (t_1 + t_2)x$. Сумму $a_1 + a_2$ вычислим за $A(m)$, сложение $t_1 + t_2$ — за константу (всего 4 возможных случая), далее вычислим $p + q$ за константу. Получаем $A(m + 1) = A(m) + O(1)$, откуда $A(m + 1) = O(A(m))$

3. Поскольку $M(m) \leq M(m + 1)$, получим $M(2m) \leq 3M(m + 1) + A(m) + 2A(m + 1)$, и по предыдущему пункту

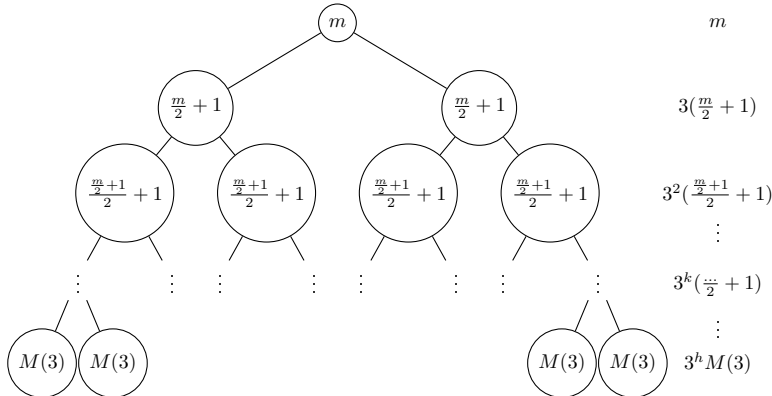
$$M(2m) \leq 3M(m + 1) + O(A(m))$$

4. Поскольку $A(m) = O(m)$, $M(2m) \leq 3M(m + 1) + O(m)$, т.е.

$$M(m) \leq 3M\left(\frac{m}{2} + 1\right) + f(m), \text{ где } f(m) = O(m).$$

5. Из определения $O(m)$ получаем $\exists m_0 \exists C > 0: \forall m \geq m_0 \hookrightarrow f(m) \leq Cm$

6. Дерево рекурсии (перестаем раскрывать, когда аргумент достигнет 3):



Найдем элементы последовательности аргументов f :

$$a_{i+1} = \frac{a_i}{2} + 1. \quad a_0 = m. \text{ По индукции докажем } a_i \stackrel{?}{=} a'_i = \frac{m}{2^i} + \sum_{k=0}^{i-1} \frac{1}{2^k}.$$

(а) База: $a_0 = n$, $a'_0 = m$ ■

(б) Переход. Пусть $a'_l = a_l$. Тогда $a'_{l+1} - a_{l+1} = \frac{m}{2^{l+1}} + \sum_{k=0}^l \frac{1}{2^k} - \frac{a_l}{2} - 1 \stackrel{?}{=} 0$. Но $a_l = a'_l$, поэтому

$$\stackrel{?}{=} \frac{m}{2^{l+1}} + \sum_{k=0}^l \frac{1}{2^k} - \frac{m}{2^{l+1}} - \sum_{k=0}^{l-1} \frac{1}{2^{k+1}} - 1. \text{ Сумма } \sum_{k=0}^{l-1} \frac{1}{2^{k+1}} = \sum_{k=1}^l \frac{1}{2^k}, \text{ поэтому } \stackrel{?}{=} \sum_{k=0}^l \frac{1}{2^k} - 1 - \sum_{k=1}^l \frac{1}{2^k} = 0 \blacksquare$$

Высота дерева $h \leq \log_2 m$, так как $a_h = 3 \Leftrightarrow \frac{m}{2^h} = 3 - \sum_{k=0}^{h-1} \frac{1}{2^k} \geq 1 \Leftrightarrow 2^h \leq m \Rightarrow h \leq \log_2 m$

Последовательность $a_l = \frac{m}{2^l} + \sum_{i=0}^{l-1} 2^{-i} \leq \frac{m}{2^l} + 2$. Получаем $M(m) \leq \sum_{k=0}^{h-1} 3^k f\left(\frac{m}{2^k} + 2\right) + 3^h M(2) \leq Cm \sum_{k=0}^{h-1} \left(\frac{3}{2}\right)^k + 2C \sum_{k=0}^{h-1} 3^k + 3^h M(2)$. Первая сумма $\sum_{k=0}^{h-1} \left(\frac{3}{2}\right)^k = \frac{1 - (3/2)^h}{1 - 3/2} \leq 2((3/2)^{\log_2 m} - 1) = 2m^{\log_2 3/2} - 2$

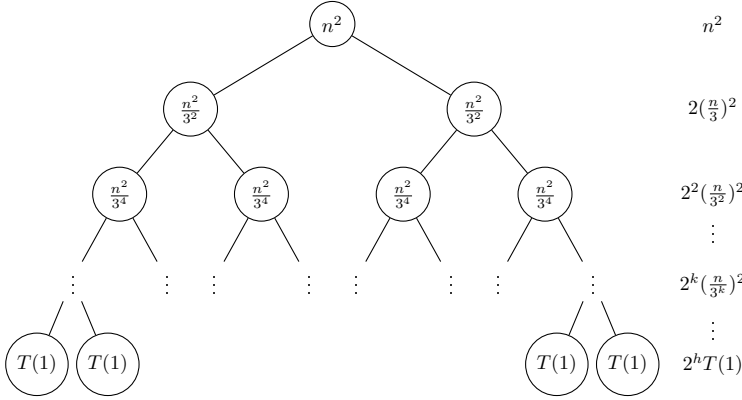
Вторая сумма $\sum_{k=0}^{h-1} 3^k = \frac{3^h - 1}{3 - 1} \leq \frac{1}{2}(m^{\log_2 3} - 1)$

Тогда $M(m) \leq Cm(2m^{\log_2 3/2} - 2) + 2C \frac{1}{2}(m^{\log_2 3} - 1) + m^{\log_2 3} T(2) = O(m^{\log_2 3})$, так как $m^{1 + \log_2(3/2)} = m^{\log_2 3}$

Ответ: $\boxed{Mult(m) = O(m^{\log_2 3})}$

Задача 1

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



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

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

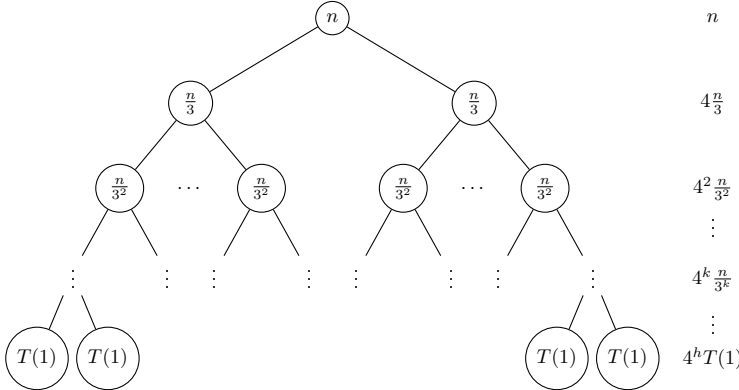
$$S(n) = \Theta(n^2)$$

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

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



Высота дерева $h = \log_3 n$, $T(n) = \sum_{k=0}^{h-1} 4^k f(\frac{n}{3^k}) + 4^h T(1)$. Из определения Ω $\exists n_0 \exists C > 0: \forall n \geq n_0 \hookrightarrow \sum_{k=0}^{h-1} 4^k f(\frac{n}{3^k}) \geq$

$Cn \sum_{k=0}^{h-1} \frac{4^k}{3^k} \stackrel{\text{геом. прогр.}}{=} Cn \frac{(4/3)^{h-1} - 1}{4/3 - 1} = 3Cn(\frac{3}{4}(\frac{4}{3})^{\log_3 n} - 1) = 3Cn(\frac{4}{3} \frac{n^{\log_3 4}}{n} - 1) = 4Cn^{\log_3 4} - 3Cn$. Также $4^h = 4^{\log_3 n} = n^{\log_3 4}$,

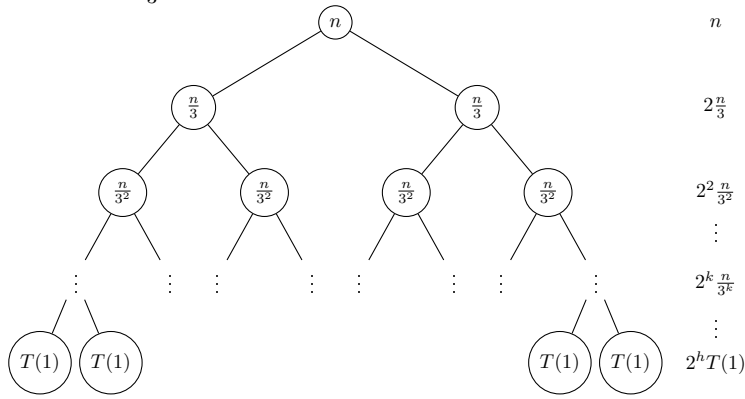
поэтому $T(n) \geq 4Cn^{\log_3 4} - 3Cn + n^{\log_3 4} T(1)$, откуда $T(n) = \Omega(n^{\log_3 4})$.

Асимптотическую оценку сверху получить не удастся, так как $T(n) \geq f(n)$, и нет верхней оценки для $f(n)$.

Ответ: $T(n) = \Omega(n^{\log_3 4})$

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 (это здесь не доказано). При последней, h -й подстановке $\frac{n}{3^h} = 1$, откуда $h = \log_3 n$.

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



Высота дерева $h = \log_3 n$. Получаем $T(n) = \sum_{k=0}^{h-1} 2^k f(\frac{n}{3^k}) + 2^h T(1)$. По определению $O \exists n_0 > 0 \exists C > 0: \forall n \geq n_0 \hookrightarrow$

$\sum_{k=0}^{h-1} 2^k f(\frac{n}{3^k}) \leq Cn \sum_{k=0}^{h-1} (\frac{2}{3})^k \leq Cn \frac{1}{1-2/3} = 3Cn = O(n)$. Оценим $2^h T(1) = 2^{\log_3 n} T(1) = n^{\log_3 2} T(1) = O(n^{\log_3 2})$. Получаем

$T(n) \leq O(n) + O(n^{\log_3 2})$. Но $\log_3 2 < 1$, поэтому $n^{\log_3 2} = \bar{o}(n)$, и по 2 получаем $T(n) = O(n)$.

С другой стороны, $T(n) \geq 2^h T(1) = \Omega(n^{\log_3 2})$.

Ответ: $T(n) = O(n)$, $T(n) = \Omega(n^{\log_3 2})$

Задача 2

Модифицируем Решето Эратосфена: для каждого вычеркнутого числа будем запоминать какую-либо пару (i, j) , «из-за которой» оно вычеркнуто. А именно:

Число $l < n$ — не простое $\overset{\text{корректность}}{\Leftrightarrow}$ число l вычеркнуто \Leftrightarrow выполнена строчка $\text{Prime}[\underbrace{i * i + i * j}_l] = \text{False} \Leftrightarrow$ существует (i, j)

из цикла, такая что i — простое, $i * (i + j) = l$.

Первая часть алгоритма (решето + запоминание пар):

```

for  $i := 1$  to  $n$  do
  Prime[i] := True  $\rightarrow c_1$ 
  I[i] := -1  $\rightarrow c_2$ 
  J[i] := -1  $\rightarrow c_3$ 
end
for  $i := 2$  to  $\lfloor \sqrt{n} \rfloor$  do
  if Prime[i] == True  $\rightarrow c_4$  then
    j := 0  $\rightarrow c_5$ 
    while  $i * i + i * j \leq n \rightarrow c_6$  do
      Prime[i*i+i*j] = False  $\rightarrow c_7$ 
      I[i*i+i*j] = i  $\rightarrow c_8$ 
      J[i*i+i*j] = j  $\rightarrow c_9$ 
      j = j + 1  $\rightarrow c_{10}$ 
    end
  end
end

```

Таким образом, для каждого числа $l \in \overline{2, n}$ известно, простое ли оно, и, если нет, один его простой делитель $I[l]$ и частное от деления $\frac{l}{I[l]} \equiv I[l] + J[l]$. Заметим, что для частного $I[l] + J[l]$ это свойство тоже выполняется (так как оно меньше, чем делимое). Поэтому будем повторять такое получение простых делителей:

```

i := n
while Prime[i] == False  $\rightarrow c_{11}$  do
  print I[i]  $\rightarrow c_{12}$ 
  i := I[i] + J[i]  $\rightarrow c_{13}$ 
end
print i

```

- Докажем конечность времени работы. Поскольку $I[i] + J[i]$ — частное от деления i на число, большее единицы (простой делитель i), то на каждой итерации i уменьшается.
- Докажем корректность. А именно, пусть $n = p_1^{k_1} \dots p_s^{k_s}$ — разложение на простые множители. Тогда алгоритм напечатает числа $\underbrace{p_1, \dots, p_1}_{k_1}, \dots, \underbrace{p_s, \dots, p_s}_{k_s}$ в некотором порядке, и, возможно, число 1.

Считаем, что до цикла цикл совершил $k = 0$ итераций. Утверждение:

$$P(k) = \begin{cases} \text{Напечатаны } k \text{ простых делителей } n: q_1, \dots, q_k & (1)_k \\ i - \text{частное от деления } n \text{ на } q_1 \cdot \dots \cdot q_k, \text{ т.е. } i = \frac{n}{\prod_{z=1}^k q_z} & (2)_k \end{cases}$$

- База. На нулевом шаге ($k = 0$) ничего не напечатано, поэтому $(1)_0, i = n = \frac{n}{1}$ (внизу пустое произведение), поэтому $(2)_0$
- Переход. Пусть выполнено t шагов, выполнено $P(t)$. Докажем $P(t + 1)$ (в случае, если цикл продолжает работу): цикл продолжает работу $\Rightarrow \text{Prime}[i] = \text{False}$. Значит, $i = I[i](I[i] + J[i])$, и $I[i]$ — простое. Поэтому будет напечатан простой делитель $q_{t+1} \stackrel{\text{def}}{=} I[i]$ числа i (он $t + 1$ -й по предположению индукции $(1)_t$). Но n делится на i по $(2)_t$, поэтому напечатан еще один простой делитель n , значит, $(1)_{t+1}$ ■

Из того же свойства $I[i] + J[i] = \frac{i}{I[i]} = \frac{i}{q_{t+1}} \stackrel{(2)_t}{=} \frac{n}{q_1 \dots q_t} \frac{1}{q_{t+1}}$, откуда $(2)_{t+1}$ ■

Итак, после последней, k -й итерации имеем $P(k)$. Цикл завершился, значит, i — простое (либо 1, см. заполнение массива Prime в самом начале). И из $P(k)$ следует, что $i = \frac{n}{q_1 \dots q_k}$, и q_1, \dots, q_k напечатаны. Последняя команда печатает последний простой делитель i (или, возможно, единицу). Корректность доказана

- Оценим время работы алгоритма.

- Докажем, что асимптотика первой части не поменялась, т.е. равна асимптотике Решета. Действительно, добавились только константы c_2, c_3, c_8, c_9 к другим константам. Таким образом, время работы первой части $O(n \log \log n)$.
- Оценим время работы второй части алгоритма: цикл совершает одну итерацию на каждый простой делитель по доказанному $(1)_k$. Найдем худший случай. Пусть $n = p_1^{k_1} \dots p_s^{k_s}$. Количество напечатанных чисел — $k_1 + \dots + k_s$. Фиксируя набор делителей и сумму $k_1 + \dots + k_s$, получаем, что минимальное число n при них — p_1^k . Теперь, меняя набор делителей при фиксированном k найдем, что минимальное число n будет при $p_1 = 2$ (минимальное простое). То есть, худший случай — степени двойки (чтобы при ограниченном сверху n получить максимальное число $k_1 + \dots + k_s$ нужно взять ближайшую степень двойки снизу). Для них $k = \log_2 n$, т.е. последняя часть алгоритма

совершит $\log n$ шагов. На каждом шаге выполняется константное число действий, поэтому время работы второй части $O(\log n)$

Итоговое время $T(n) = O(n \log \log n) + O(\log n) = O(n \log \log n)$, т.е. совпадает с временем работы Решета.

Задача 3

Задачу рассказывал Пименов на курсе Алгоритмы: построение и анализ

Фиксируем n . Рассмотрим внешний цикл. Если число i не простое, совершается C_1 операций (эта и следующая константы не зависят от n), иначе выполняется внутренний цикл, который совершает не более, чем $C_2 \lceil \frac{n}{i} \rceil$ операций (из условия $i * i + i * j \leq n$ получаем $j \leq \frac{n - i * i}{i} \leq \frac{n}{i}$). Пусть P_n — множество простых чисел, не превосходящих n . Тогда $T(n) \leq C_2 \sum_{i \in P_n} \lceil \frac{n}{i} \rceil + C_1 \sum_{x \in \overline{2, n} \setminus P_n} 1$.

Мощность $|\overline{2, n} \setminus P_n| \leq n$, поэтому второе слагаемое — $O(n)$.

Оценим первую сумму. $\sum_{i \in P_n} \lceil \frac{n}{i} \rceil \leq C_3 n \sum_{i \in P_n} \frac{1}{i} \ll$.

Используем факт из Википедии [1]: $\sum_{p \in P_n} \frac{1}{p} \leq C_4 \log \log x$, подставим: $\ll C_3 C_4 n \log \log n = O(n \log \log n)$.

Получаем $T(n) \leq O(n) + O(n \log \log n) = O(n \log \log n)$ ■

Вспомогательные утверждения

1. Пусть $f_1 = \Theta(g_1)$, $f_2 = \Theta(g_2)$, $g_2 = \bar{\bar{o}}(g_1)$, $g_2(n) > 0$. 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

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

Здесь использовалось определение $\bar{\bar{o}}$. 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 $f_1 + f_2 = \bar{\bar{o}}(g_1)$ ■

2. Пусть $f_1 = O(g_1)$, $f_2 = O(g_2)$, $g_2 = \bar{\bar{o}}(g_1)$, $g_2(n) > 0$. Тогда $f_1 + f_2 = O(g_1)$. Доказательство выше (нужно взять правую часть большого неравенства).

Список литературы

[1] Википедия: Простое число. Раздел «некоторые свойства»