Алгоритмы. ДЗ на неделю 3.

Прохоров Юрий, 771

Задача 1 (ДЗ)

Пусть на доске написаны числа $a_1, a_2, \ldots, a_n$. Пусть

$$d = \gcd(a_1, \ldots, a_n).$$

Допустим, для определенности, на первом шаге рассматриваются числа a_1 и a_2:

$$a_1 < a_2, \qquad a_1 = k_1 d, \qquad a_2 = k_2 d.$$

Вместо a_2 будет записано число $a_2 - a_1 = (k_2 - k_1)d > 0$, которое делится на d. Докажем, что

$$\gcd(a_1, a_2) = \gcd(a_1, a_2 - a_1) :$$

Пусть $\gcd(a_1, a_2) = d \Longrightarrow a_1 = k_1 d, a_2 = k_2 d$, где k_1 и k_2 — взаимно простые числа. Покажем, что числа k_1 и $k_2 - k_1$ тоже взаимно простые. Допустим противное:

$$\gcd(k_1, k_2 - k_1) = t, \qquad k_1 = mt, \qquad k_2 - k_1 = nt \Longrightarrow k_2 = (m+n)t \Longrightarrow \gcd(k_1, k_2) = t,$$

что является противоречием. Используя следующее свойство,

$$\gcd(a_1, a_2, a_3) = \gcd(\gcd(a_1, a_2), a_3),$$

можно сделать вывод, что наибольший общий делитель всех чисел, записанных на доске после первого шага не изменится. Аналогично, этот наибольший общий делитель не изменится и после каждого следующего шага.

Все числа получаются положительные и целые, и каждое из них делится на d, поэтому меньше d на доске получить нельзя. Допустим, на доске все числа оказались равны $d' > d$. Проделав все разности в обратную сторону и восстановив исходные числа, получим, что каждое исходное число делилось на d', следовательно, d не являлось их наибольшим общим делителем — противоречие. Поэтому все числа на доске станут равны $d = \gcd(a_1, \ldots, a_n)$.

Задача 2 (ДЗ)

Известно, что алгоритм Евклида поиска наибольшего общего делителя, делая $O(n)$ рекурсивных вызовов, и в каждом из которых выполняя операцию деления с остатком, работающую за $O(n^2)$, в сумме требует $O(n^3)$ времени. Пусть $\operatorname{lcm}(a, b)$ — наименьшее общее кратное чисел a и b (least common multiple). Тогда

$$\operatorname{lcm}(a, b) = \frac{ab}{\gcd(a, b)}.$$

Для нахождения наименьшего общего кратного достаточно за $O(n^3)$ найти НОД, затем за $O(n^2)$ вычислить произведение $a \cdot b$ и за $O(n^2)$ найти конечное частное. Итоговая асимптотика — $O(n^3)$. Алгоритм настолько же эффективен с этой точки зрения, как и алгоритм Евклида поиска НОД.

Задача 3 (ДЗ)

Алгоритм

Построим онлайн-алгоритм, решающий данную задачу. Пусть число a_1 уже введено. Определим $Sum = 0, Ans = 0$. Тогда на каждом следующем k-м шаге ($k = 1, 2, \ldots$) будут пересчитываться следующие значения:

$$Ans = Ans + 2 \cdot Sum \cdot a_k,$$

$$Sum = Sum + a_k.$$

Кратко алгоритм можно записать следующим образом:

```
Ans = 0; Sum = 0;
for k = 1; k ≤ n; k += 1 do
  | Input: a[k];
  | Ans += 2 * Sum * a[k];
  | Sum += a[k];
end
Output: Ans;
```

Корректность

Докажем корректность по индукции. Докажем, что после k-го шага алгоритма выполняются равенства

$$Ans = \sum_{\substack{i \neq j \\ i,j \leq k}} a_i \cdot a_j, \qquad Sum = \sum_{i=1}^{k} a_i.$$

На первом шаге алгоритма $Ans = 0, Sum = 0$. После первого шага $Ans = 0 \cdot a_1 = 0, Sum = a_1$, что верно. Допустим, на первых $k-1$ шагах алгоритм работал верно. Покажем, что корректность сохранится и на k-ом шаге.

$$Ans = \sum_{\substack{i \neq j \\ i,j \leq k-1}} a_i a_j + 2 \cdot a_k \cdot \sum_{i=1}^{k-1} a_i = \sum_{\substack{i \neq j \\ i,j \leq k-1}} a_i a_j + \sum_{i=1}^{k-1} (a_i a_k + a_k a_i) = \sum_{\substack{i \neq j \\ i,j \leq k}} a_i a_j,$$

$$Sum = \sum_{i=1}^{i=k-1} a_i + a_k = \sum_{i=1}^{k} a_i.$$

Следовательно, алгоритм работает корректно.

Оценка по времени

Всего алгоритм делает ровно n циклических действий, каждое из которых состоит из нескольких элементарных арифметических операций. Будем считать, что числа, подаваемые на вход, ограничены (например, типом данных integer), тогда каждое циклическое действие требует $\Theta(1)$ времени.

Поэтому общее время работы алгоритма линейно — $\Theta(n)$.

Задача 4 (ДЗ)

Здесь и далее будет использоваться формулировка **основной теоремы о рекурсии**, приведенная в 3-м издании учебника Т. Кормена "Алгоритмы. Построение и анализ".

Основная теорема о рекурсии

Пусть $a \geq 1$, $b > 1$ — константы, $f(n)$ — функция, а $T(n)$ определена на множестве неотрицательных целых чисел с помощью рекуррентного соотношения

$$T(n) = aT\left(\left[\frac{n}{b}\right]\right) + f(n),$$

где $\left[\frac{n}{b}\right]$ интерпретируется либо как $\left\lfloor\frac{n}{b}\right\rfloor$, либо как $\left\lceil\frac{n}{b}\right\rceil$. Тогда $T(n)$ имеет следующие асимптотические границы.

1. Если $f(n) = O(n^{\log_b a - \varepsilon})$ для некоторой константы $\varepsilon > 0$, то $T(n) = \Theta(n^{\log_b a})$.
2. Если $f(n) = \Theta(n^{\log_b a})$, то $T(n) = \Theta(n^{\log_b a} \log n)$.
3. Если $f(n) = \Omega(n^{\log_b a + \varepsilon})$ для некоторой константы $\varepsilon > 0$ и если $af\left(\frac{n}{b}\right) \leq cf(n)$ для некоторой константы $c < 1$ и всех достаточно больших n, то $T(n) = \Theta(f(n))$.

Решение

(а)

$$T(n) = 36T\left(\frac{n}{6}\right) + n^2$$

Здесь $a = 36, b = 6, f(n) = n^2 = \Theta(n^2)$. Функция $n^{\log_b a} = n^2$ асимптотически сравнима с функцией $f(n)$. Эта задача подходит под второй случай основной теоремы, поэтому

$$T(n) = \Theta(n^2 \log n).$$

(б)

$$T(n) = 3T\left(\frac{n}{3}\right) + n^2$$

Здесь $a = 3, b = 3, f(n) = n^2 = \Theta(n^2) = \Omega(n^2)$. Функция $n^{\log_b a} = n$. Таким образом,

$$\exists \varepsilon = 1 > 0 : f(n) = n^2 = \Omega(n^{\log_b a + \varepsilon}) = \Omega(n^{1+1}) = \Omega(n^2)$$

Для применения третьего случая основной теоремы необходимо также проверить условие регулярности:

$$\exists c < 1, n_0 \in \mathbb{N} : \forall n \geq n_0 \longmapsto af\left(\frac{n}{b}\right) < cf(n),$$

то есть

$$3\left(\frac{n}{3}\right)^2 = \frac{n^2}{3} < cn^2,$$

что верно при любых натуральных n при, например, $c = \frac{2}{3}$. Согласно третьему пункту основной теоремы,

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

(в)

$$T(n) = 4T\left(\frac{n}{2}\right) + \frac{n}{\log n}$$

Здесь $a = 4, b = 2, f(n) = \frac{n}{\log n}$. Функция $n^{\log_b a} = n^2$. Покажем, что при $\varepsilon = 1$ функция $f(n) = O(n^{2-\varepsilon}) = O(n)$, то есть, что

$$\exists c > 0, n_0 \in \mathbb{N} : \forall n \geq n_0 \longmapsto f(n) = \frac{n}{\log n} < cn.$$

Если взять n_0 за округленное вверх основание логарифма, а $c = 2$, то неравенство будет выполнено. Тогда этот случай удовлетворяет первому пункту основной теоремы, и

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

Задача 5 (ДЗ)

Время работы алгоритма можно представить рекуррентным соотношением

$$T(n) = nT\left(\frac{n}{2}\right) + f(n), \qquad \text{где} f(n) = O(n)$$

Будем считать, что $T(1) = O(1)$. Обозначим $k = \log_2 n$.

Верхняя оценка

$$T(n) \le nT\left(\frac{n}{2}\right) + cn \le n\left(\frac{n}{2}T\left(\frac{n}{4}\right) + c\frac{n}{2}\right) + cn = n\frac{n}{2}T\left(\frac{n}{4}\right) + c\left(n + n\frac{n}{2}\right) \le \ldots \le$$
$$\le n\frac{n}{2}\ldots\frac{n}{2^k}T(1) + c\left(n + n\frac{n}{2} + n\frac{n}{2}\frac{n}{4} + \ldots + n\frac{n}{2}\ldots\frac{n}{2^{k-1}}\right) \le c\left(n + n\frac{n}{2} + \ldots + n\frac{n}{2}\ldots\frac{n}{2^k}\right) \le$$
$$\le 2c\left(n + n\frac{n}{2} + \ldots + n\frac{n}{2}\ldots\frac{n}{2^{k-1}}\right) \le 2c(n + n^2 + \ldots + n^k) = 2c\frac{n(n^k-1)}{n-1} \le 2c\frac{n^{k+1}}{\frac{n}{2}} = 4cn^k = O(n^{\log_2 n}).$$

Однако эту функцию можно оценить и по-другому:

$$T(n) \le 2c\left(n + n\frac{n}{2} + \ldots + n\frac{n}{2}\ldots\frac{n}{2^{k-1}}\right) \le 4c\left(n + n\frac{n}{2} + \ldots + n\frac{n}{2}\ldots\frac{n}{2^{k-2}}\right) \le \ldots = O(n^{\log_2 n - \varepsilon}), \quad \varepsilon > 0.$$

Нижняя оценка

Рассмотрим самый быстрый случай, когда $f(n) = 0 = O(n)$. Тогда

$$T(n) = nT\left(\frac{n}{2}\right) = n\frac{n}{2}\frac{n}{4}\ldots\frac{n}{2^k} = \frac{n^{k+1}}{2^{0+1+\ldots+k}} = \frac{n^{k+1}}{2^{\frac{k(k+1)}{2}}} = n^k\frac{2^k}{2^{\frac{k(k+1)}{2}}} = \frac{n^k}{2^{\frac{k(k-1)}{2}}} = n^{k-\frac{k-1}{2}} = n^{\frac{\log_2 n+1}{2}}$$

Задача 6 (ДЗ)

(a)

$$T(n) = T(\lfloor \alpha n \rfloor) + T(\lfloor (1-\alpha)n \rfloor) + c_1 n$$

Докажем, что $T(n) = O(n \log n)$ по индукции. Предположим, что

$$\exists C > 0, N \in \mathbb{N} : \forall n > N \longmapsto T(n) < Cn \log n.$$

Положим $\beta = max(\alpha, (1-\alpha)) < 1$. Тогда

$$T(n) = T(\lfloor \alpha n \rfloor) + T(\lfloor (1-\alpha)n \rfloor) + c_1 n < C\lfloor \alpha n \rfloor \log\lfloor \alpha n \rfloor + C\lfloor (1-\alpha)n \rfloor \log\lfloor (1-\alpha)n \rfloor + c_1 n \le$$
$$\le C\alpha n \log \alpha n + C(1-\alpha)n \log(1-\alpha)n + c_1 n \le C\beta n \log \beta n + C(1-\beta)n \log \beta n + c_1 n =$$
$$= Cn \log \beta n + c_1 n = n\left(C \log \beta b + c_1\right) = n\left(C \log \beta + C \log n + c_1\right) < Cn \log n.$$

Отсюда

$$C \log \beta + c_1 + C \log n < C \log n,$$
$$\beta < 1 \Longrightarrow \log \beta < 0 \Longrightarrow C > \frac{c_1}{\log \frac{1}{\beta}}.$$

Таким образом, при любом C, удовлетворяющем последнему неравенству, и любом n верно, что

$$T(n) < Cn \log n.$$

Возьмем C, если необходимо, настолько большим, что неравенство верно, для начальных значений n. Теперь покажем, что $T(n) = \Omega(n \log n)$. Пусть $\alpha = \frac{1}{2}, n = 2^k$. Тогда

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n).$$

Это соотношение удовлетворяет второму пункту мастер-теоремы, поэтому в данном случае

$$T(n) = \Theta(n \log n).$$

В общем случае отсюда следует, что

$$T(n) = \Omega(n \log n) \Longrightarrow T(n) = \Theta(n \log n).$$

(б)

$$T(n) = T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + 2T\left(\left\lfloor\frac{n}{4}\right\rfloor\right) + c_1 n$$

Докажем верхнюю оценку $T(n) = O(n\log n)$. Допустим, для некоторого $C > 0$ при достаточно больших n $T(n) < Cn\log n$. Тогда

$$T(n) = T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + 2T\left(\left\lfloor\frac{n}{4}\right\rfloor\right) + c_1 n < C\left\lfloor\frac{n}{2}\right\rfloor\log\left\lfloor\frac{n}{2}\right\rfloor + 2C\left\lfloor\frac{n}{4}\right\rfloor\log\left\lfloor\frac{n}{4}\right\rfloor + c_1 n \le$$
$$\le C\frac{n}{2}\log\frac{n}{2} + C\frac{n}{2}\log\frac{n}{4} + c_1 n = \frac{Cn}{2}\log\frac{n^2}{8} + c_1 n = n\left(\frac{C}{2}(2\log n - \log 8) + c_1\right) =$$
$$= n\left(C\log n - \frac{C\log 8}{2} + c_1\right) < Cn\log n.$$

Отсюда

$$C\log n - \frac{C\log 8}{2} + c_1 < C\log n,$$
$$C > \frac{2c_1}{\log 8}.$$

Начальные случаи оговариваются аналогично пункту (а). При данных C неравенство верно, поэтому верхняя оценка $T(n) = O(n\log n)$. Чтобы доказать нижнюю оценку, рассмотрим частный случай, когда $n = 4^p = 2^k$. Тогда

$$T(n) = T\left(\frac{n}{2}\right) + 2T\left(\frac{n}{4}\right) + c_1 n,$$
$$T(n) = f(n)\cdot n,$$
$$f(n)n = \frac{n}{2}f\left(\frac{n}{2}\right) + 2\frac{n}{4}f\left(\frac{n}{4}\right) + c_1 n = \frac{n}{2}f\left(\frac{n}{2}\right) + \frac{n}{2}f\left(\frac{n}{4}\right) + c_1 n,$$
$$f(n) = \frac{1}{2}f\left(\frac{n}{2}\right) + \frac{1}{2}f\left(\frac{n}{4}\right) + c_1,$$
$$g(k) = f(n) = f(2^k),$$
$$g(k) = \frac{1}{2}g(k-1) + \frac{1}{2}g(k-2) + c_1,$$
$$g(k) = 2h(k) + 2k\left(1 + \frac{1}{3}c_1\right),$$
$$2h(k) + 2k + \frac{2kc_1}{3} = h(k-1) + (k-1)\left(1 + \frac{1}{3}c_1\right) + h(k-2) + (k-2)\left(1 + \frac{1}{3}c_1\right) + c_1,$$
$$h(k) = \frac{1}{2}h(k-1) + \frac{1}{2}h(k-2) - \frac{3}{2},$$
$$h(k) = t(k) - k,$$
$$t(k) = \frac{1}{2}(t(k-1) + t(k-2)).$$

Каждый элемент последовательности $t(k)$ — среднее арифметическое последних двух ее элементов, поэтому $t(k)$ не может превосходить $\max(t(1), t(2))$, то есть $t(k)$ — ограниченная последовательность (функция). Тогда $t(k) = \Theta(1)$, можно считать, что $t(k) = c_0$.

$$h(k) = c_0 - k,$$
$$g(k) = 2c_0 - 2k + 2k + \frac{2kc_1}{3} = 2c_0 + \frac{2kc_1}{3},$$
$$f(n) = 2c_0 + \frac{2c_1\log_2 n}{3},$$
$$T(n) = c'n + c''n\log n = \Theta(n\log n)$$

. Поэтому в общем случае $T(n) = \Omega(n\log n)$, откуда и из предыдущей верхней оценки следует, что

$$T(n) = \Theta(n\log n).$$

(в)

$$T(n) = 27T\left(\frac{n}{3}\right) + \frac{n^3}{\log^2 n}, \qquad n = 3^k$$

Данное соотношение не укладывается ни в один из случаев мастер-теоремы, так как добавочная функция $\frac{n^3}{\log^2 n}$ не отличается от n^3 на некую полиномиальную сложность. Преобразуем данное соотношение:

$$\begin{aligned} T(n) = 27T\left(\frac{n}{3}\right) + \frac{n^3}{\log^2 n} = 3^3 T\left(3^{k-1}\right) + \frac{3^{3k}}{\log^2 3^k} = 3^3 T\left(3^{k-1}\right) + \frac{3^{3k}}{k^2 \log^2 3} = \\ = 3^3\left(3^3 T\left(3^{k-2}\right) + \frac{3^{3(k-1)}}{(k-1)^2\log^2 3}\right) + \frac{3^{3k}}{k^2\log^2 3} = 3^6 T\left(3^{k-2}\right) + \frac{3^{3k}}{(k-1)^2\log^2 3} + \frac{3^{3k}}{k^2\log^2 3} = \\ = \ldots = 3^{3k}T(3^{k-k}) + \frac{3^{3k}}{(k-(k-1))^2\log^2 3} + \ldots + \frac{3^{3k}}{(k-1)^2\log^2 3} + \frac{3^{3k}}{k^2\log^2 3} = \\ = 3^{3k}T(1) + \frac{3^{3k}}{1^2\log^2 3} + \frac{3^{3k}}{2^2\log^2 3} + \ldots + \frac{3^{3k}}{k^2\log^2 3} = 3^{3k}c + \frac{3^{3k}}{\log^2 3}\left(\frac{1}{1^2} + \frac{1}{2^2} + \ldots + \frac{1}{k^2}\right). \end{aligned}$$

Оценим функцию сверху. Известно, что ряд

$$\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi}{6}.$$

Тогда

$$T(n) \le 3^{3k}\left(c + \frac{1}{\log^2 3}\sum_{i=1}^{\infty}\frac{1}{i^2}\right) = 3^{3k}\left(c + \frac{\pi}{6\log^2 3}\right) = C3^{3k} = Cn^3 = O(n^3).$$

В качестве нижней оценки рассмотрим

$$T(n) \ge c3^{3k} = cn^3 = \Omega(n^3).$$

Отсюда следует, что

$$T(n) = \Theta(n^3).$$