Алгоритмы 3

Максим Пасько

25 февраля 2018 г.

1

Пусть k - НОД всех исходных чисел. Тогда i-тый элемент представляется как kx_i, где все x_i попарно взаимно простые. Тогда каждый раз при вычитании из одного числа другое, мы будем получать число, кратное k. То есть, при окончании процесса получается число, кратное k. Пусть получились все числа, равные $m = nk$. Тогда, шагая назад к исхоному массиву, складывая элементы, кратные m между собой, мы получим, что все исходные элементы кратны m, но $m = nk$, а k - НОД. То есть, это может случится, только если $n = 1$. Тогда получаем, что в конце может получится лишь НОД всех чисел исходного массива.

2

Пусть даны числа a и b. Очевидно, что

$$\text{НОК}(a, b) = \frac{a * b}{\text{НОД}(a, b)}.$$

Найдём НОД алгоритмом Евклида(через разности), время работы которого $O(n^2)$. Умножение также $O(n^2)$. Деление также $O(n^2)$. То есть, общее время работы алгоритма $O(n^2)$.

3

$for(i = 1; i \leq n; + + i)$
$\{$
$sum_1 + = a_i;$
$sum_2 + = (a_i)^2;$
$\}$
$out = (sum_1)^2 - sum_2;$
$print(out);$

4

a)

$n^2 = \Theta(n^2)$. Тогда по основной теореме рекурсии $T(n) = \Theta(n^2 \log n)$.

б)

$n^2 = \Omega(n^{1+\varepsilon}), \varepsilon = 1.$
$3(\frac{n}{3})^2 = \frac{n^2}{3} \leq cn^2, c = \frac{1}{2}$
Тогда по основной теореме рекурсии $T(n) = \Theta(n^2)$.

в)

$\frac{n}{\log n} = O(n^{2-\varepsilon}), \varepsilon = \frac{1}{2}$. Тогда по основной теореме рекурсии $T(n) = \Theta(n^2)$.

5

Исходная задача разбивается на n задач размера $\frac{n}{2}$, каждая из которых разбивается на $\frac{n}{2}$ задач размера $\frac{n}{4}$ и т.д. То есть, на первом уровне $O(n)$, на втором $n * O(\frac{n}{2})$, на третьем $\frac{n^2}{2}$ и т.д. Получается геометрическая прогрессия, сумма которой

$$O(n) + 2\sum_{i=1}^{\log n} \frac{n^i}{2^i} * O(\frac{n}{2^i}) = O(n^{\log n})$$

как сумма убывающей геометрической прогрессии.

6

а)

$$\sum_{i=0}^{\log_{\frac{1}{\alpha}} n} 2^i * \Theta(n) = \Theta(n \log n)$$

Аналогично,

$$\sum_{i=0}^{\log_{\frac{1}{1-\alpha}} n} 2^i * \Theta(n) = \Theta(n \log n)$$

Эти две функции ограничивают $T(n)$ сверху и снизу соответственно(смотря какое α), поэтому $T(n) = \Theta(n \log n)$.

б)

$$T(n) \leq \sum_{i=0}^{\log n - 1} (3^i * \Theta(n)) + 3^{\log n} = \Theta(n \log n)$$

С другой стороны,

$$T(n) \geq \sum_{i=0}^{\log n - 1} ((\frac{3}{2})^i * \Theta(n)) + (\frac{3}{2})^{\log n} = \Theta(n \log n)$$

Тогда $T(n) = \Theta(n \log n)$;