Домашнее задание 3

Христолюбов Максим, 771

Задание 1

При вычитании делимость чисел на их НОД не меняется, значит, НОД всех чисел, записанных изначально на доске, и в конце одинаковый. Следовательно, так все числа будут равны, то их НОД будут равен им самим, т. е. на доске будет записан НОД всех этих чисел.

Задание 2

Найдем НОД по алгоритму Евклида. НОК находится из равенств НОК$\cdot$ НОД$= a \cdot b$. Сложность алгоритма $O(n^3)$, т к на умножение требуется $O(n^{\log_2^3})$ по алгоритму быстрого умножения, на деления $O(n^2)$, значит, сложность алгоритма нахождения НОК - $O(n^3)$

Задание 3

$\sum a_i \cdot a_j = \frac{1}{2} \cdot \left((\sum a_i)^2 - \sum a_i^2 \right)$

Для нахождение суммы элементов требуется $O(n)$ операций, возведение в квадрат одного числа, равного суме элементов - константа, для суммы квадратов - $O(n)$, т к возведение элемента в квадрат происходит за константу действий. Значит $\sum a_i \cdot a_j$ вычисляется за $O(n)$ операций.

Задание 4

а) $a = 36$, $b = 6$, $f(n) = n^2 = n^{\log_6 36} = \Theta(n^{log_b a})$

По мастер-теореме $T(n) = \Theta(n^2 \log n)$

б) $a = 3$, $b = 3$, $f(n) = n^2 = n^{\log_3 3+1} = \Omega(n^{\log_b a+\epsilon})$, где $\epsilon = \frac{1}{2}$

По мастер-теореме $T(n) = \Theta(f(n)) = \Theta(n^2)$

в) $a = 4$, $b = 2$, $f(n) = \frac{n}{\log n} = O(n^{\log_2 4 - \frac{1}{2}}) = O(n^{\log_b a - \epsilon})$, $\epsilon = \frac{1}{2}$

По мастер-теореме $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$

Задание 5

$T(n) = n \cdot T(\frac{n}{2}) + O(n) = n\left(\frac{n}{2} \cdot T\left(\frac{n}{4}\right) + O(n)\right) + O(n) = \frac{n^2}{2} \cdot T(\frac{n}{4}) + O(n^2) = \frac{n^3}{8} \cdot T(\frac{n}{8}) + O(n^3) = ... = \frac{n^k}{2^{\frac{k(k-1)}{2}}} \cdot T(\frac{n}{2^k}) + O(n^k)$, $k = \log_2 n$

$T(n) = \frac{n^{\log_2 n}}{\left(2^{\log_2 n}\right)^{\frac{\log_2 n - 1}{2}}} \cdot T(1) + O(n^{log_2 n}) = n^{\log_2 n - \frac{1}{2}\log_2 n + \frac{1}{2}} + O(n^{log_2 n}) =$
$= n^{\frac{1}{2}\log_2 n + \frac{1}{2}} + O(n^{log_2 n}) = O(n^{\log_2 n}\sqrt{n}) = \Omega(n^{\log_2 n})$

Задание 6

а) Пусть для определенности $\alpha < 1 - \alpha$

$T(n) \leq T(\alpha n) + T((1-\alpha)n) + \Theta(n) = T(\alpha^2 n) + 2T(\alpha(1-\alpha)n) + T((1-$
$-\alpha)^2 n) + 2\Theta(n) \leq ... \leq 2^{\log_\alpha n} T(1) + \log_\alpha n \cdot \Theta(n) = Cn + \Theta(n \log_\alpha n) = O(n \log n)$

$T(n) \geq 2^{\log 1 - \alpha n} T(1) + \log_{1-\alpha} n \Theta(n) = Cn + \Theta(n \log_{1-\alpha} n) = \Omega(n \log n)$

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

б) $T(n) = T(\frac{n}{2}) + 2T(\frac{n}{4}) + \Theta(n) = T(\frac{n}{4}) + 2T(\frac{n}{8}) + 4T(\frac{n}{16}) + 2\Theta(n) \leq ... \leq 2^{\log 2n} T(1) + \log_2 n \Theta(n) = O(n \log n)$

$T(n) \geq 2^{\log 4n} T(1) + \log_4 n \Theta(n) = \Omega(n \log n)$

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

в) $T(n) = 27T(\frac{n}{3}) + \frac{n^3}{\log^2 n} = 3^{3 \cdot 2} T(\frac{n}{3^2}) + 3^3 \cdot \frac{\left(\frac{n}{3}\right)^3}{\log^2 \frac{n}{3}} + \frac{n^3}{\log^2 n} = 3^{3 \cdot 2} T(\frac{n}{3^2}) + \frac{n^3}{\log^2 \frac{n}{3}} + \frac{n^3}{\log^2 n} = 3^{3 \cdot 3} T\left(\frac{n}{3^3}\right) + \frac{n^3}{\log^2 \frac{n}{3^2}} + \frac{n^3}{\log^2 \frac{n}{3}} + \frac{n^3}{\log^2 n} = 3^{3 \cdot \log_3 n} T(1) + n^3 \sum\limits_{k=0}^{\log_3 n - 1} \frac{1}{\log^2 \frac{n}{3^k}} =$
$n^3 T(1) + n^3 \sum\limits_{k=0}^{\log_3 n - 1} \frac{1}{\log^2 \frac{n}{3^k}} = \Omega(n^3)$

Т.к. бесконечный ряд обратных квадратов натуральных чисел сходится, т.е. имеет конечную сумму, а $\log^2 \frac{n}{3^k}$ - некоторые натуральные числа, причем не равные друг другу и упорядоченные, то $\sum\limits_{k=0}^{\log_3 n - 1} \frac{1}{\log^2 \frac{n}{3^k}}$ меньше чем конечная сумма обратных квадратов натуральных чисел, значит, меньше бесконечной суммы меньше некоторой константы.

$T(n) = O(n^3 + n^3 \cdot C) = O(n^3) = \Theta(n^3)$

Задание 7

$(i!)^{-1}(\mod p) = x$

$x \cdot i! \equiv 1(\mod p)$

$x \cdot i! + p \cdot k = 1$

Это уравнение решается в целых числах с помощью алгоритма Евклида за константное кол-во C_k арифметических действий для конкретного x_k. Решим это уравнение для $i \in 1..p-1$ и получим x_1,x_2,..,x_i. Это можно сделать за $\Theta(n)$ арифметических действий, т.к. сумма действий ограничена сверху $C_k n$, где C_k-самый большой коэффициент, а снизу - $C_j \frac{n}{2}$, где C_j- медиана массивов коэффициентов.

$(i!)^{-1}(\mod p) = (1 \cdot 2 \cdot ... \cdot p \cdot ... \cdot i)^{-1}(\mod p) = 1^{-1} \cdot 2^{-1} \cdot ... \cdot (i)^{-1}(\mod p) = x_1 x_2 ... x_i(\mod p)$

x_i перемножаются за $\Theta(n)$ действий, а делится по модулю за константу. Значит можно найти $x_1 = 1^{-1}$ по алгоритму Евклида, записать в массив, умножить его на 2^{-1} и получить x_2? записать в массив и тп - на это потребуется n шагов, на каждом - константное кол-во действий - всего $\Theta(n)$.

Следовательно весь алгоритм работает за $\Theta(n)$ и вычисляет $(i!)^{-1}(\mod p)$.

Задание 7 с семинара

N_k - длина числа в k-ой записи.

$n = N_2 \leq N_{10} \leq N_{16} = 4 \cdot N_2 = O(n)$, значит, $N_{10} = \Theta(n)$, где $n = N_2 = \Theta(n)$

Умножение числа N длины n на число M. На первом шаге:

$N = M + N_1$,где $N_1 < M$

$N \cdot M = M^2 + M * N_1$

Найдем $M \cdot N_1$ тем же алгоритмом или же если одно из них достаточно маленькое просто умножим "столбиком"за константное время, и будем продолжать, пока N_i или M_i не станет равно 0.

Корректность. На каждом шаге или N_i, или M_i уменьшается хотябы на 1, значит настанет момент когда один из них станет равен 0.

Если одно из чисел достаточно маленькое, то его можно считать константой, а умножение числа длины n на константу происходит за $\Theta(n)$. Если считать что второе число достаточно большое, то это значит, что числа отличаются на небольшое число шагов, и кол-во действий которое надо произвести - константа, так как на каждом шаге $\Theta(n)$ операций, то всего потребуется $\Theta(n)$.