# Домашнее задание 3 по алгоритмам LaTeX

Попов Николай

26 февраля 2018 г.

## №1

Равные числа будут равны НОД-у всех чисел, записанныз на доске, т.к. вычитанием из большего числа меньшее мы реализуем алгоритм евклида ( вычитание - его более медленный вариант, но в результате тоже остается отсаток от деления).

## №2

Запишем определения для НОК(n, m) = (n, m) и НОД(n, m) = [n, m], зная разложение этих чисел на простые множители (здесь показатели могут быть нулевыми, но основания степеней - простые числа - одни и те же)

$$n = p_1^{a_1} * p_2^{a_2} * ... * p_k^{a_k}$$

$$m = p_1^{b_1} * p_2^{b_2} * ... * p_k^{b_k}$$

Получим

$$(n, m) = \prod_i p_i^{min(a_i, b_i)}$$

$$[n, m] = \prod_i p_i^{max(a_i, b_i)}$$

Отсюда следует

$$(n, m) * [n, m] = nm$$

т.к. min(a,b)+max(a,b)=a+b

Значит, для нахождения НОК надо посчитать произведение чисел, найти их НОД по алгоритму Евклида и разделить первое на второе.

Корректность алгоритма следует из выше приведенных выкладок.

Пусть n - длина двоичной записи чисел. Асимптотика складывается из последовательно выполяемых алгоритмов умножения ($\Theta(n^2)$), алгоритма Евклида($\Theta(n^3)$) алгоритма деления($\Theta(n^2)$).В итоге, получаем $\Theta(n^3)$.

(Действителльно, деление и умножение можно сделать например по алгоритмам Divide и Multiply с указанной асимптотикой. В алгоритме Евклида через каждые два рекурентных вызова длина обоих чисел уменьшается вдвое (т.к. если $a \geqslant b$, то a mod b < a/2), а значит, вызовов будет не больше 2n. При этом на каждом шаге выполняется деление, т.е. поярдка $n^2$ операций. В итоге, Евклид дает $\Theta(n^3)$ асимптотику.)

## №3

Посмотрим, что нужно сделать чтобы посчитать искомую сумму попарно различных (по индексам) элементов массива длины n. Для простоты сделаем это для 6 чисел massiv[ ] ={a b c d e f}:

$$\sum = ab+ac+ad+ae+af+bc+bd+be+bf+cd+ce+cf+de+df+ef =$$

$$= a(b+c+d+e+f)+b(c+d+e+f)+c(d+e+f)+d(e+f)+ef$$

Т.е. надо для каждого числа посчитать сумму чисел, стоящих после него, тогда сумма произведений каждого из чисел массива на соответствующую ему частичную сумму и будет искомым результатом. Корректность очевидна.

| | | | | | | |
|---|---|---|---|---|---|---|
| sum[] | | b+c+d+e+f | c+d+e+f | d+e+f | e+f | f |
| a[] | a | b | c | d | e | f |

Псевдокод

res - результат, sum[i+1] - массив частичных сумм элемента a[i], i - вспомогательная переменная.

```
res = 0
sum[n] = a[n]
for i = n-1 to 2 do
        sum[i] = sum[i+1] + a[i]
for i = 1 to n-1 do
        res +=a[i]*sum[i+1]
return res
```

Не принимая во внимание время необходимое для совершения ариметических операций, получаем что алгоритм выполнит порядка n-3 шага, а значит является линейным.

## №4

a)$T(n) = 36T(\frac{n}{6}) + n^2, a = 36, b = 6, f(n) = n^2$

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

Значит, по второму пункту мастер-теоремы, получаем

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

$$б) T(n) = 3T(\frac{n}{3}) + n^2, a = 3, b = 3, f(n) = n^2$$

$$n^{\log_b a} = n^{\log_3 3} = n$$

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

Проверим регулярность f(n):

$$af(n/b) = 3f(n/3) = 3\frac{n^2}{9} = \frac{1}{3}n^2 < 1 * n^2 = 1 * f(n)$$

По третьему пункту мастер-теоремы получаем:

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

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

$$n^{\log_b a} = n^{\log_2 4} = n^2$$

Т.к. $f(n) = O(n)$, то по 1 пункту мастер-теоремы получаем $T(n) = \Theta(n^2)$

## №5

$T(n) = n * T(\frac{n}{2}) + O(n)$
Расписав дерево рекурсии получаем формулу для числа операций на k-ом шаге:

$$\frac{n^k}{2^{\frac{1}{2}k(k-1)}} T\left(\frac{n}{2^k}\right) + O\left(\frac{n^k}{2^{\frac{1}{2}k(k-1)}}\right)$$

Тогда T(n)=C и при k, меняющемся от 1 до $\log_2 n$, получим:

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

## №6

а)

$$T(n) = T(\alpha n) + T((1-\alpha)n) + Cn = T(\alpha^2 n) + 2T(\alpha(1-\alpha)n) + T((1-\alpha)^2) + 2Cn = ...$$

$$... = kCn + \sum_{i=0}^{k} C_k^i T(\alpha^{k-i}(1-\alpha)^i n)$$

Оценим по уровням обрезованного и дополненного деревьев, получим:

$$Cn \log_{\frac{1}{\alpha}} n \leqslant T(n) \leqslant Cn \log_{\frac{1}{1-\alpha}} n \to T(n) = \Theta(n \log n)$$

б)

$$T(n) = T(\frac{n}{2}) + 2T(\frac{n}{4}) + Cn = T(\frac{n}{2^2}) + 4T(\frac{n}{2*4}) + 4T(\frac{n}{4^2}) + 2Cn = ...$$

$$... = \sum_{i=0}^{k} C_k^i 2^i T(\frac{n}{2^{k+i}}) + kCn$$

Оценка такая же по 2 деревьям:

$$Cn \log_4 n \leqslant T(n) \leqslant Cn \log_2 n \to T(n) = \Theta(n \log n)$$

в)

$$T(n) = 27T(\frac{n}{3}) + \frac{n^3}{\log^2 n} = 27^2 T(\frac{n}{3^2}) + \frac{n^3}{\log^2 \frac{n}{3}} = ... = 27^k T(\frac{n}{3^k}) + \sum_{i=1}^{k} \frac{n^3}{\log^2 \frac{n}{3^{i-1}}} =$$

$$= C_1 n^3 + n^3 \sum_{k=1}^{\log_3 n} \frac{1}{\log^2 \frac{n}{3^{k-1}}} = C_1 n^3 + n^3 \sum_{k=1}^{\log_3 n} \frac{1}{(\log_3 n - k + 1)^2} =$$

Т.к. сумма наша сумма обратных квадратов больше единицы (1 содержится в сумме) и меньше бесконечной суммы обратных квадратов, равной $\frac{\pi^2}{6}$, то $T(n) = \Theta(n^3)$

## №7

Для решения воспользуемся теоремой Вильсона:
Натуральное число p>1 - простое, iff (p-1)!+1 кратно p.
Тогда

$$-(p-1)! mod p = 1 mod p = i!\cdot(i!)^{-1} mod p \rightarrow [i!\cdot(i!)^{-1}+(p-1)!] \vdots p \rightarrow i![(i!)^{-1}+(i+1)(i+2)...(p-1)] \vdots p$$

т.к. i$\leqslant n < p$

$$\rightarrow (i!)^{-1} mod p = -(i+1)(i+2)...(p-1)$$

Значит, для нахождения искомого обратного остатка надо для каждого числа i от 1 до n в массиве найти произведение последующих чисел до (p-1)(это делается за линейное время наподобие номера 3, т.е. асимптотика $O(n)$). Остаток от деления этого числа, взятого с минусом, как показано выше и будет искомым результатом.