Домашнее Задание по алгоритмам №4

Павливский Сергей Алексеевич , 873

03.03.2019

Задание №1

Вычитая из одного элемента массива другой элемент массива нельзя получить число меньшее НОД всех чисел массива , т.к. раз оба числа кратны НОД , то и разность кратна НОД . Также процесс не остановится пока одно из чисел > НОД , т.к. из него можно вычесть меньшее (если меньших нет , а есть только большие , то мы из остальных можем вычесть это число ; если же все числа равны , то они все равны НОД всех чисел (НОД одинаковых чисел равно самим числам)) . Значит , т.к. в итоге останутся одинаковые числа , которые <= НОД , и которые также >= НОД , то они будут равны НОД .

Ответ : НОД всех элементов массива .

Задание №2

Используя алгоритм из дз номер 3 для нахождения НОК :

```
m := a; n := b; u := b; v := a;
инвариант: НОД (a,b) = НОД (m,n); m,n >= 0
while not ((m=0) or (n=0)) do begin
| if m >= n then begin
| | m := m - n; v := v + u;
| end else begin
| | n := n - m; u := u + v;
| end;
end;
if m = 0 then begin
| z:= v;
end else begin n=0
| z:= u;
```

end;

Мы считаем z за O(n^2) , деление пополам происходит за O(n) операций , т.е. НОК находится за O(n^2) операций .

Итого , количество операций равно O(n^2) операций .

Задание №3

Заведем переменную sum изначально равную 0. За первый проход по массиву прибавляем к sum каждый встреченный элемент массива , в итоге получим сумму всех элементов массива . Возводим sum в квадрат . За второй проход по массиву вычитаем из sum квадрат встреченного элемента массива . После завершения второго прохода делим sum на 2 . Получаем требуемое (верно из формулы квадрата суммы n слагаемых) .

Доказательство временной сложности :

За первый проход выполняется θ (n) операций (так как в данной задаче одна операция выполняется за θ (1)). Возведение в квадрат выполняется за θ (1) . За второй проход количество операций также θ (n) . Деление пополам выполняется за θ (1) операций . Значит итоговая временная сложность = θ (n) , т.е. время линейное .

Задание №4

а)

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

a = 36

b = 6

Т.к. n^2 = θ ($n^{log_b a}$),

то T(n) = $\theta(n^2 lgn)$

2)

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

a = 3

b = 3

Т.к. n^2 = Ω ($n^{log_b a+\varepsilon}$) , при ε = 0,5 , а также $3 * (\frac{n}{3})^2$ <= c * n^2 при c = $\frac{1}{3}$, то

то T(n) = $\theta(n^2)$

3)

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

a = 4

b = 2

Т.к. $\frac{n}{logn}$ = O($n^{log_b a - \varepsilon}$) , при ε = 1 , то

то T(n) = $\theta(n^2)$

Задание №5

T(n) = n * T($\frac{n}{2}$) + θ(n)

Рассмотрим дерево рекурсии :

0 - cn

1 - $\frac{cn^2}{2}$

i - $\frac{cn^{i+1}}{2^{\frac{i*(i+1)}{2}}}$

Высота $log_2 n$

n = 2^k

T(n) = $\sum_{i=1}^{k} \frac{c*2^{i+1}}{2^{\frac{i*(i+1)}{2}}}$ = $\theta(\frac{n^{logn+1}}{2})$

Задание №6

а)

T(n) = T(an) + T((1 - a)n)) + θ (n) (0 < a < 1)

Рассмотрим дерево рекурсии . На каждом уровне количество операций не больше cn . Общее количество операций <= Cnlog n , т.к. высота дерева log n . Также количество операций >= C1nlog n , т.к. min расстояние от корня до листьев также log n . Тогда T (n) = θ (n log n).

б)

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

Рассмотрим дерево рекурсии . Высота дерева $log_4 n$, min расстояние от корня до листьев $log_2 n$, операций на каждом уровне Cn . Тогда операций :

C1n $log_2 n$ <= T(n) <= C2n $log_4 n$

C1n $\frac{log_2 n}{log_2 2}$ <= T(n) <= C2n $\frac{log_2 n}{log_2 2}$

C1n $logn$ <= T(n) <= C2n $logn$

T(n) = θ (n log n)

в)

T(n) = 27T($\frac{n}{3}$) + $\frac{n^3}{log^2 n}$

Рассмотрим дерево рекурсии . На i-м уровне $\frac{n^3}{log^2 \frac{n}{3^i}}$. Высота дерева рекурсии равна $log_3 n$. Тогда операций $\sum_{i=1}^{log_3 n} \frac{n^3}{log^2 \frac{n}{3^i}} = n^3 * \sum_{i=1}^{log_3 n} \theta(\frac{1}{i^2}) = \theta(-n^3 \frac{1}{log_3 n} + n^3) = \theta(n^3/log_3 n)$