Филиппова Анастасия

Домашнее задание 4

Проверьте, пожалуйста, 5 задачу из предыдущего домашнего задания и задачу из контрольной. Заранее спасибо.

5) 3-ие домашнее задание:
$T(n) = nT(n/2) + O(n)$
На k - ом шаге:
$\frac{n^{k+1}}{2^{\frac{k(k+1)}{2}}} + \sum \frac{n^k}{2^{\frac{k(k-1)}{2}}}$, высота дерева рекурсии $k = \log_2 n$, тогда, так как на последнем уровне $T(1)$ - константное время, получаем $\frac{n^{\log_2 n+1}}{2^{\frac{\log_2 n(\log_2 n+1)}{2}}} = \frac{n^{\log_2 n}}{\sqrt{n^{\log_2 n} n}} = n^{\log_2 \sqrt{2n}}$
Теперь оценим сумму, которую дают O(n):
$\sum < \log_2 n(\frac{n^{log_2 n}}{2^{\frac{(log_2 n-1)log_2 n}{2}}}) = n^{(1/2\cdot\log_2 n+1)}\log_2 n \Rightarrow T(n) = O(n^{(1/2\cdot\log_2 n+1)}\log_2 n)$ и $T(n) = \Omega(n^{\log_2\sqrt{2n}})$
Но такая верхняя оценка очень грубая, так как на нижних уровнях дерева (при маленьких n) $O(n)$ растет медленно. Докажем по индукции, что $T(n) = O(n^{\log_2\sqrt{2n}})$:
$T(n) <= c\cdot n^{\log_2\sqrt{2n}}$, тогда: $T(n) <= c\cdot(n/2)^{\log_2\sqrt{n}} + cn < c\cdot n^{\log_2\sqrt{n}} \Rightarrow T(n) = \Theta(n^{\log_2\sqrt{2n}})$

Задача из контрольной работы: $T(n) = T(n-A) + T(B) + O(n)$
На k - ом шаге:
$T(n-kA) + kT(B) + \sum(cn - c(k-1)A)$, при $k = n/A, T(n-kA) = T(1) = const$. $T(B)$ можем пренебрь, потому что это просто какая- то константа. Оценим сумму сверху :$\sum < n/A(cn - n/A\cdot A) < n/A = dn^2$. Оценим снизу: $\sum > n/2(cn/2 - (n/2A)\cdot A) > d'n^2 \Rightarrow T(n) = \Theta(n^2)$.

(1) (a) $F(3,5) = 243 = 3^5$
  (b) Функцию возведения в степень.
  (c) Это алгоритм бинарного возведения в степень, только слева на право. $m = \overline{n_k n_{k-1}...n_0}_2, m = n_k 2^k ... n_0 2^0$
  $x^m = ((...((x^{n_k})^2)x^{m_{k-1}})^2)...)x^{m_0})$
  (d) Пусть длина входа числа m - n. В худшем случае $length(a) = 2n-1 \Rightarrow 2n-1$ сравнений и $2n-1$ умножений. Получается, данный алгоритм работает за линейное время. $(\Theta(n))$.

(2) Всего $2n$ точек (n левых концов, n правых концов). Расположим их в порядке возрастания координаты. Найдем $n/3$ порядковую статистику, далее найдем $n/3+1$ порядковую статистику.Применим процедуру Partition, взяв за опорный элемент $a_{(n/3)}$ - найдем точки, координаты которых больше $n/3$.Далее применим процедуру для $a_{(n/3+1)}$ - найдем все точки, координата которых меньше. Аналогично сделаем с другого конца прямой: найдем $5n/3$ статистику и $5n/3+1$ и также сделаем Partition. Полученные точки - точки, которые покрыты ровно 2n/3 отрезками.

(3) Алгоритм нахождения k - ой порядковой статистики при делении массива на $[n/7]$ групп аналогичен алгоритму при делении на $[n/5]$ групп.
$T(n) = T(n/7) + T(9n/14) + O(n), T(9n/14)$- так как это длина массива, на котором будет вызываться алгоритм в худшем случае (хотя бы $5n/14$ элементов будут отброшены).
По индукции докажем, что $T(n) = O(n)$
$T(n) <= cn, T(n) <= cn/7 + 9cn/14 + an = 11cn/14 + an \Rightarrow T(n) = O(n)$.То есть в худшем случае алгоритм работает за линейное время.

(4) Очевидно, что достаточно сделать процедуру Partition относительно любого элемента массива. Сортировка будет работать за линейное время.

(5) $M|(ax+b)$
$\exists\, k : ax + b = kM \Rightarrow ax + (-kM) = (-b)$.Чтобы данное уравнение имело решения $(a, M)|b$. С помощью расширенного алгоритма Евклида находим решения. Коректность алгоритма очевидна. Ассимптотика: пусть наибольшая длина входа из 3 это n. Тогда: $T(n) = T(n-1) + \Theta(n^2) \Rightarrow$ в худшем случае НОК находим за $O(n^3)$.Далее проверяем, есть ли решения.$(\Theta(n^2))$. Расширенный алгоритм Евклида: в худшем случае 2n вызовов, при каждом из которых $O(n^2)$ операций. Получаем: $\Theta(n^3) + O(n^3) = O(n^3)$.

(6) (a) На уже отсортированном массиве или на массиве, отсортированном по убыванию, глубина стека рекурсий для алгоритма быстрой сортировки будет глубтной n, потому что функция будет вынуждена вызвать сама себя n раз.
  (b) Чтобы уменьшить размера стека рекурсий необходимо по-другому выбирать опорный элемент при процедуре Partition. Если при каждом рекурсивном вызове мы будем находить медиану и делать Partition относительно нее, тоб очевидно, что глубина стека рекурсий будет $\log n$.