Контролно по ДАА – 04.06.2013

Задача 1: Решете следните рекурентни отношения:

а) $T(n) = 2T\left(\frac{n}{\sqrt{2}}\right) + n^2 lgn$

От разширението на Master Th => $T(n) = \theta(n^2(lgn)^2)$

б) $T(n) = 3T\left(\frac{n}{3}\right) + 2n + 6$

Тъй като $n = \theta(2n + 6)$, то от 2ри случай на Master Th => $T(n) = \theta(nlgn)$

в) $T(n) = T(n - 2) + n^3$

От нехомогенната част получаваме 1 като четирикратен корен, а корените на характеристичното уравнение са 1 и -1, т.е. общо корените са {1, 1, 1, 1, 1, 1, -1} $\Rightarrow T(n) = \theta(n^4)$

г) $T(n) = 2T(n - 1) + 3T(n - 2) + n3^n$

От нехомогенната част получаваме 3 като двукратен корен, а корените на характеристичното уравнение са 3 и -1, т.е. общо корените са {3, 3, 3, -1} $\Rightarrow T(n) = \theta(n^2 3^n)$

Задача 2: Дадени са сортираните масиви $a[1, \dots, n]$ и $b[1, \dots, m]$. Да се предложи възможно най-бърз (в асимптотичен смисъл) алгоритъм за намирането на медианата на масива, който се получава при сливане на двата дадени масива.

Може да проверяваме дали елементът a[i] е медиана на новия масив с константна сложност – ако означим с $j = \frac{m+n}{2} - i + 1$, то ще искаме $b[j] \leq a[i] \leq b[j + 1]$. В противен случай може да разберем в коя половина е медианата. Ето защо, използвайки двоично търсене може да намерим медианата на масива със сложност $O(\lg(m + n))$.

Задача 3: Да се предложи алгоритъм за намиране на цикъл с нечетна дължина в неориентиран граф.

Задачата може да се реши с познатите алгоритми BFS и DFS, като при посещение в нов връх по време на алгоритъма да се използва оцветяване 0/1. По този начин, ако стигнем до връх, оцветен в различен цвят от този, който бихме ползвали в момента, то сме намерили нечетен цикъл.

Задача 4: Да се предложи алгоритъм за намиране на най-близкия общ предшественик (т.е. възможно най-отдалечен от корена) на два върха (u и v) в дърво.

Едно възможно бързо решение на задачата е да се направи списък с предшествениците на върховете – например за връх u – $(u', u'', \dots, r)$ и за v – $(v', v'', \dots, r)$ и след това да се намери първия общ елемент в двете редици (такъв винаги съществува – най-малко коренът на дървото).

Задача 5: Да се предложи алгоритъм за намиране на покриващо дърво T на неориентиран тегловен свързан граф G, такова че теглото на най-тежкото ребро в T е минимално.

Доказва се, че минималното покриващо дърво винаги изпълнява това условие (твърдението в обратната посока обаче не винаги е вярно – може да се направи контрапример), т.е. всеки от алгоритмите за намиране на минимално покриващо дърво е решение на задачата.