Второ контролно по ДАА, Специалност КН, група 1, 06.06.2013 г.

(15 точки) 1 Задача. Решете следните рекурентни отношения чрез метода с характеристичното уравнение.

а) $T(n) = 2T(n-1) - T(n-2) + n$

б) $T(n) = \frac{1}{3}T(n-1) + \frac{4}{3}T(n-2) + \left(\frac{4}{3}\right)^n + \left(\frac{4}{6}\right)^n n$

в) $T(n) = 3T(n-1) + 3^n(\sqrt{3} + 5n) + 3^n(1+n)$

Решение:

(за всяка подточка по 5 точки)

а) Корените на характеристичното уравнение са $\{1,1\}_M$, от нехомогенната част още $\{1,1\}_M$, общо $\{1,1,1,1\}_M$, оттук $T(n) = \Theta(n^3)$.

б) Корените на характеристичното уравнение са $\{\frac{4}{3}, -1\}_M$, от нехомогенната част още $\{\frac{4}{3}, \frac{2}{3}, \frac{2}{3}\}_M$, общо $\{\frac{4}{3}, -1, \frac{4}{3}, \frac{2}{3}, \frac{2}{3}\}_M$, оттук $T(n) = \Theta(n(\frac{4}{3})^n)$.

в) Корените на характеристичното уравнение са $\{3\}_M$, от нехомогенната част още $\{3,3\}_M$, общо $\{3,3,3\}_M$, оттук $T(n) = \Theta(n^2 3^n)$.

(12 точки) 2 Задача. Иванчо отива да си купи парче торта от лавката и там магазинера трябва да му върне ресто C с възможно най-малко монети. Той разполага с много монети по v1,v2,v3,...,vn стотинки, като 1 = v1 < v2 < v3 < ... < vn. Предложете алгоритъм, по който магазинера да знае с колко най-малко монети от наличните може да върне това ресто.

Решение: Задачата се решава с ДП, ще си пазим за всяка стойност от 0 стотинки до C стотинки с колко най-малко монети може да се плати в масив $M[0,1,2,...,C]$. Така после с едно обхождане по тези суми от 1 до C ще получаваме с колко най-малко монети може да се плати. И това обхождане ще ползва $M[i] = \min\limits_{v_k\ ,\ k=1,2,...,n}(M[i-v_k]+1)$

(13 точки) 3 Задача. Предложете алгоритъм с колкото е възможно по-малка сложност по време, който при вход неориентиран граф G(V,E) връща ДА, ако графът има цикъл с дължина 3, и НЕ, в противен случай.

Решение:

Алгоритъм със сложност $O(n^3)$, където n е броя на върховете е явен на пръв поглед. За всеки три върха от графа проверяваме дали образуват цикъл с дължина 3, дали има ребра измежду всеки два върха от тях. Ако за всяко ребро проверяваме за всеки връх, дали има цикъл с дължина 3 съставен от това ребро и този връх получаваме алгоритъм със сложност $O(n * m)$.