Tema 2

1: Demonstrati ca un arbore binar care nu este plin nu poate corespunde unui cod optim. Pentru definitia unui

cod optim, va rog sa consultati cursul despre coduri Huffman. Va reamintesc ca intr-un arbore binar plin,

orice nod cu exceptia frunzelor are exact 2 fii.

Sa presupunem prin absurd ca un arbore Huffman (sa-l numim A) incomplet ar corespunde unui cod optim.

Ca un arbore sa nu fie plin inseamna ca exista un nod care are un singur fiu (care totodata e si frunza. Sa ne referim la frunza drept f\_A.

Fie i multimea funzelor arborului fara f\_A.

Daca formam un subarbore a lui A eliminand nodul pe care se afla frunza f\_A, atunci acest subarbore va avea costul si adancimea mai mica. Sa numim acest subarbore A2.

cost(A) = 

cost(A2) = 

cost(A) < cost(A2)

2: Explicati cum se poate modi\_ca metoda de sortare quicksort pentru ca aceasta sa ruleze in cazul cel mai defavorabil (i.e., worst-case) in timp O(n log n), presupunand ca toate numerele ce trebuie sortate sunt distincte.

3: Fie T un arbore binar de cautare si x un nod din arbore care are doi copii. Demonstrati ca succesorul nodului x nu are fiu stang, iar predecesorul lui x nu are fiu drept.

4: Rezolvati recurenta T(n) = T(n/2) + T(n/3) + 1. Demonstrati.

De demonstrat: T(n) = O(n) <=> T(n) ≤ c\*n

Sa presupunem ca si ca

(relatia 1)

Pentru (relatia 2)

Din relatia 1 si 2 am demonstrat ca , unde