

Examen - Algoritmi si Structuri de Date

Seria 15

3 Septembrie 2017

Alexandru Popa (curs), Andrei Patrascu (laborator)

In primul rand, va rog sa va scrieti NUMELE si GRUPA pe foaia de examen! Timpul de lucru este de 2 ore. Nu aveti voie sa aveti asupra dumneavoastra decat instrumentul de scris si foile pe care vi le vor oferi instructorii. Daca vom gasi asupra dumneavoastra telefoane mobile, laptopuri, tablete, fituici sau alte materiale ce contin informatii ajutatoare, veti fi scosi din sala de examinare. Daca aveti intrebari, ridicati mana si unul dintre instructori va veni la dumneavoastra in cel mai scurt timp. Scrieti raspunsurile in spatiul indicat. Ultima foaie capsata poate fi folosita ca ciorna. Daca mai aveti nevoie de hartie, adresati-va unui instructor.

1 Exerciitii de nota 4,5 - (4,5 puncte)

1.1 1,5 (0,5 puncte pe exercitiu)

Exprimati functiile urmatoare in notatia Θ :

(a) $n^3 - \frac{n^3}{2}$

(b) $\ln^2 n + \sqrt{n}$

(c) $\ln \ln n + \ln n$

1.2 1 punct (0,5 puncte pe intrebare)

Este adevarat ca $2^{n+1} = O(2^n)$? Este adevarat ca $2^{2n} = O(2^n)$?

1.3 1 punct (0,5 puncte pe exercitiu)

Sa se construiasca heap-uri(ansamble) prin insertia pe rand a urmatoarelor chei. Apoi, sa se extraga radacina din heap-uri-le rezultate.

(a) min-heap: 10, 22, 2, 18, 5, 33, 30

(b) max-heap: 1, 4, 28, 5, 45, 2, 88

1.4 1 punct

Care este codul (arborele) Huffman optim pentru urmatoarele frecvente: $a:10$ $b:4$ $c:8$ $d:3$ $e:5$ $f:15$ $g:13$ $h:21$?

Daca exista mai multi arbori optimi, oricare din ei va primi punctajul maxim.

2 Exerciitii cu demonstratii - (4,5 puncte)

2.1 1,5 puncte

Demonstrati ca orice algoritm de sortare *bazat pe comparatii intre chei* are timp de rulare $\Omega(n \log n)$.

2.2 1,5 puncte

Demonstrati ca orice algoritm care construiesc un arbore binar de cautare cu n numere ruleaza in timp $\Omega(n \log n)$.

2.3 1,5 puncte

Demonstrati ca daca $f(n) = \Theta(g(n))$ si $g(n) = \Theta(h(n))$ atunci $f(n) = \Theta(h(n))$.

3 Exerciitiu cu algoritmi - (2 puncte)

Fie $X[1 :: n]$ si $Y[1 :: n]$ doi vectori, fiecare continand n numere *sortate*. Prezentați un algoritm care sa gaseasca mediana celor $2n$ elemente. Mediana unei multimi de n elemente este elementul de pe pozitia $\lceil n/2 \rceil$ in sirul sortat. De exemplu, mediana multimii 3, 1, 7, 6, 4, 9 este 4.

In functie de timpul de rulare al algoritmului veti primi urmatoarele punctaje: $O(n \log n)$ - (0,25 puncte); $O(n)$ - (0,5 puncte); $O(\log^2 n)$ - (1 punct); $O(\log n)$ - (1,5 puncte).