Structuri de date și algoritmi
- examen scris -

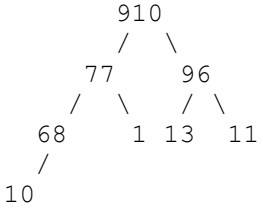
Notă

1. Subiectele se notează astfel: of - 1p; A – 2p; B - 1.5p; C1 - 1p; C2 – 1p; D - 3.5p.
2. Pentru cerința A, justificarea unei complexități presupune deducția acesteia.
3. Pentru cerințele B și C (C1, C2) se cer justificări, care vor fi punctate.
4. Problema de la D se va rezolva în Pseudocod. Se cer și se vor puncta: (1) descrierea ideii de rezolvare și comentarii despre soluția propusă; (2) scrierea reprezentării indicate în enunț; (3) (specificare și) implementare subalgoritm(i); (4) complexitate.
   Nu se acceptă cod C++. Nu se acceptă pseudocod fără comentarii despre soluția propusă.

A. Deduceți timpii mediu si defavorabil pentru următorul algoritm. Justificați rezultatul.

```
Functia f(n) este {:Intreg}
|       {pre: n:Intreg}
|       S←0; m ←n
|       cattimp m≠0 executa
|       |       S←S+[m/10]; m←[m/10]
|       sfcattimp
|       f ←S
Sff
Algoritm A
|       S←0; @citeste n;
|       pentru i=1, n executa
|       |       m←2*i+1; S←S+f(m)
|       sfpentru
|       scrie S
sfA
```

B. Arătați ansamblul rezultat prin aplicarea operației de ștergere din urmatorul ansamblu. Justificati



C. Inserarea unui element într-un vector ordonat $x_1, \ldots, x_n$ se poate face în:

a) $O(\log_2 n)$ b) $O(n)$ c) $\theta(n)$ d) $\theta(\log_2 n)$

Justificati

C. Algoritmul de sortare MergeSort are proprietatea de sortare stabilă. Justificati

a) adevărat b)fals

D. Se consideră un arbore binar conținând în noduri elemente distincte. Se cere sa se scrie în Peudocod operația care să determine înălțimea unui element *e* dat. Arborele se reprezintă înlănțuit, cu înlănțuirile reprezentate pe tablou. Se va folosi o procedura nerecursivă. Se va indica reprezentarea și se va preciza complexitatea operației. Folosiți comentarii pentru a ușura înțelegerea soluției.

Ex: Pentru arborele de mai jos, *e*= 22 ⇒ înălțimea este 3