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 subalgoritm. Justificați rezultatul.

```
Funcția F(n, i) este {:Intreg}
|       {pre: n, i:Intreg}
|       dacă n=1 atunci
|       |       F←1
|       |  altfel
|       |       m ← n div 2
|       |       dacă i mod 2 =0 atunci
|       |       |       F← F(m, i) - i
|       |       |  altfel
|       |       |       F← F(m, i) + i
|       |       Sfdacă
|       Sfdacă
SfF
```

T(n) = T(n/2) + O(1)
T(n/2) = T(n/4) + O(1)
....
T(n) = T(n/2^k) + kO(1)
T(n) = O(log2n)

in cazul timpului meidu consideram cele doua posibilitati ca n sa fie par sau impar, dar complexitatea medie este ca cu cea defavorabila

B. Ilustrați, pe un exemplu concret, o ștergere din AVL care necesită a SRD.

C. Se consideră un vector de numere reale. Alegeți algoritmii de sortare care pot fi folosiți pentru ordonarea vectorului. Justificati

a) MergeSort b) BucketSort c) RadixSort d) HeapSort

Singurul care nu este recomandat pt sortarea numerelor reale este Radix Sort deoarece este un alg de sortare non-comparativ si care functioneaza bine ppt numere intregi. D Utilizarea lui pe numerele reale ar fi complicat deoarece acestea ar trebui transformate in intregi si partea zecimala ar trebui gestionata diferit.

C. Dacă un ansamblu este implementat folosind un vector numit *data*, și acest vector conține *n* elemente ($n > 0$), unde este valoarea cea mai mare într-un ansamblu construit cu relația ≥? Justificati

a) *data*[0] b) *data*[n-1] c) *data*[n] d) *data*[2*n + 1] e) *data*[2*n + 2]

D. Descrieți operația de dublă rotație spre dreapta pentru reechilibrare într-un Arbore Binar de Căutare. Arborele se reprezintă secvențial, pe vector, folosind ca schemă de memorare ansamblul. Indicați grafic situația de rotație, reprezentarea arborelui și descrieți in Pseudocod subalgoritmul. Precizați complexitatea operației. Folosiți comentarii pentru a ușura înțelegerea soluției