

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. Scrieți un subalgoritm recursiv având complexitatea timp $O(n^3)$. Justificați complexitatea (prin deducția acesteia).

B. Ilustrați pe un exemplu concret operația de simplă rotație spre dreapta într-un arbore AVL. Exemplul va fi ales în așa fel încât să justifice necesitatea aplicării rotației. Justificați

C. Care este scopul principal al unui Iterator? Justificati

- a) adăugarea unor noi obiecte la un container
- b) să parcurgă elementele unui container câte unul la moment dat
- c) să permită ștergerea unor obiecte dintr-un container

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

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

D. Descrieți operația de dublă rotație spre stânga pentru reechilibrare într-un Arbore Binar de Căutare. Arborele se reprezintă înlanțuit, cu alocare dinamică a nodurilor. Indicați grafic situația de rotație, reprezentarea arborelui și descrieți în Pseudocod subalgoritmul. Precizați complexitatea operației. Folosiți comentarii pentru a ușura înțelegerea soluției.