

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 $O(\log_2 n)$. Justificați complexitatea (prin deducția acesteia). Nu se va folosi funcția matematică **logaritm**.

B. Fie U o mulțime de chei (univers de chei), n numărul cheilor ce trebuie dispersate și m numărul de locații din tabela de dispersie.
Arătați că dacă $|U| > n \cdot m$, există o submulțime a lui U de mărime n , care conține chei ce se dispersează toate în aceeași locație.

C. Care este numărul minim de noduri într-un arbore binar aproape plin de adâncime 4? Justificati

- a) 17 b) 15 c) 16 d) 18 e) 10

C. Una dintre diferențele dintre coadă și stivă este: Justificați

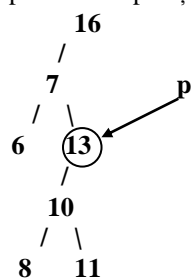
a) cozile necesită liste înlănțuite, iar stivele nu
ambele capete ale structurii, stivele doar un capăt

b) stivele necesită liste înlănțuite, iar cozile nu

c) cozile memorează
d) stivele memorează ambele capete ale structurii, cozile doar un capăt

D. Scrieți în Pseudocod subalgoritmul care, într-un Arbore Binar de Căutare, efectuează următoarea operație: pentru un nod **p** găsește nodul cu cheia imediat mai mică decât cheia din nodul **p**. Arborele se reprezintă înlănțuit cu alocare dinamică a nodurilor. Indicați reprezentarea arborelui și precizați complexitatea operației. Folosiți comentarii pentru a ușura înțelegerea soluției.

Exemplu:



pentru **p**=nodul cu cheia 13 \Rightarrow 11