

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.

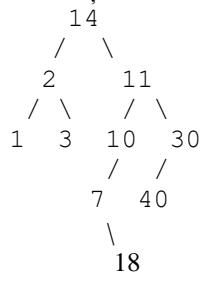
A. Deduceți timpii mediu si defavorabil pentru următorul subalgoritm. Justificați rezultatul.

Subalgoritm **S**(n, i) este

```
{ pre: n:Intreg; i:Intreg }
daca n>1 atunci
    i←2*i; m←[n/2]
    daca i mod 2=0 atunci
        S(m, i-2)
    altfel
        S(m, i-1)
    sfdaca
    altfel
        scrie i
    sfdaca
```

sfOperatie

B. Care este înălțimea nodului 11 în arborele binar de mai jos? Justificați.



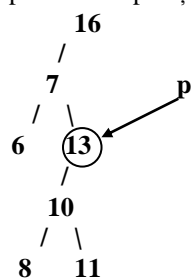
C. Selectați afirmația falsă. Justificați.

- a) orice arbore binar are cel puțin un nod
- b) orice arbore binar nevid are exact un nod rădăcina
- c) orice nod dintr-un arbore are cel mult doi fii
- d) orice nod care nu e rădăcina are exact un părinte

- C. Algoritmul de sortare HeapSort are proprietatea de sortare stabilă. Justificati
- a) adevarat
 - b) fals

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 înlănțuirile reprezentate pe tablou. Indicați reprezentarea arborelui și precizați complexitatea operației. Folosiți comentarii pentru a ușura înțelegerea soluției.

Exemplu:



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