

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

Algoritm recursiv de cautare binara:

```
Subalgoritm bs(v, st, dr, x)
    daca st > dr atunci
        bs ← -1
    m = (st+dr)/2
    if ( v[m] < x)
        bs ← bs(v, st, m, x)
    else
        bs ← bs(v, m+1, dr, x)
```

$$T(n) = T(n/2) + 1$$

$$T(n/2) = T(n/4) + 1$$

...

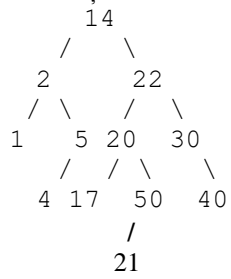
$$T(n/2^{k-1}) = T(n/2^k) + 1$$

$$\Rightarrow n/2^k = 1$$

$$\Rightarrow 2^k = n$$

$$\Rightarrow k = \log_2(n)$$

B. Care este înălțimea nodului 22 în arborele binar de mai jos? Justificați.



Înălțimea nodului este 3 (lungimea drumului maxim de la 22 la un nod frunză -21- este 3).

C. Ștergerea unui element e dintr-un vector ordonat x_1, \dots, x_n se poate face în:

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

$O(\log_2 n)$ este timpul favorabil de gasire al elementului insa dupa gasire elementele trebuie mutate, iar asta se va realiza in $O(n)$. Incluzand ambele complexitati, $\Theta(n)$ este complexitatea exacta a stergerii.

C. Presupunem o implementare înlănțuită a unei Cozi, memorând în 2 variabile referințe la primul, respectiv ultimul nod al cozii.

Care dintre aceste două variabile se modifică la o adăugare într-o coadă VIDĂ? Justificati

- a) niciuna b) doar primul se modifică c) doar ultimul se modifică d) amandoua se modifică

In coada adaugarea se face dupa ultimul element, iar noul element devine ultimul.

D. Fie containerul **Coada cu priorităţi** reprezentat folosind un *ansamblu cuaternar* (în loc de 2 descendenţi, vor fi 4). Se va folosi o relaţie de ordine \mathfrak{R} între priorităţi (dacă $\mathfrak{R}=\leq$, atunci elementul cel mai prioritar este **minimul**). Specificaţi şi implementaţi operaţia de **ştergere** din **CP**. Se va indica reprezentarea folosită şi se va preciza complexitatea operaţiei. Folosiţi comentarii pentru a uşura înţelegerea soluţiei.

Ansamblu

```
max : Integer { capacitatea maxima de memorare }
n : Integer { numarul de elemente din ansamblu }
e : TElem[1...n] { elementele din ansamblu }
```

Deoarece Ansamblul este cuaternar, fiecare nod va intretine minim 0 si maxim 4 fii.

Vizualizam vectorul astfel :

a_1, a_2, \dots, a_n – sub forma unui arbore cuaternar aproape plin

a_1 – radacina arborelui

a_i – are fiul 1 pe $a_{4 \cdot i - 2}$, 2 pe $a_{4 \cdot i - 1}$, 3 pe $a_{4 \cdot i}$, 4 pe $a_{4 \cdot i + 1}$

a_i – are parintele $a_{\lceil i / 4 \rceil}$

Prin urmare, pe nivelul i in arbore sunt 4^i noduri :

$n = 1 + 4 + 4^2 + \dots + 4^h$ { h – inaltimea arborelui }

$$n = (4^{(h+1)} - 1) / (4 - 1) = (4^{(h+1)} - 1) / 3$$

$$4^{(h+1)} - 1 = 3 \cdot n$$

$$4^{(h+1)} = 3 \cdot n + 1$$

$$h + 1 = \log_4(3 \cdot n + 1)$$

$$h = \log_4(3 \cdot n + 1) - 1$$

$$h = \lceil \log_4(3 \cdot n + 1) \rceil - 1$$

prin urmare, inaltimea arborelui cuaternar este $h = \lceil \log_4(3 \cdot n + 1) \rceil - 1$

Subalgoritm Sterge(a, e)

{ complexitate $O(\log_4(3 \cdot n + 1))$ }

{ pre : a – Ansamblu, a nu e vid }

{ post : e – TElem e elementul minim si e sters, a ramane ansamblu dupa stergere }

{ memoram elementul minim, aflat in radacina }

$m \leftarrow a[1]$ // m este minimul din ansamblu

$n \leftarrow n - 1$ // n este numarul de elemente din ansamblu

{ restabilim proprietatea de ansamblu }

@Coboara($a, 1$)

SfSubalgoritm