

Structuri de date - Test restanțe / mărir

filipvalentin66g@gmail.com [Switch accounts](#)



Întrebări

Fie următorul MaxHeap implementat cu tablouri: (42, 39, 27, 10, 4, 21). După o operație de inserare a cheii 30, urmată de o operație de eliminare, tabloul asociat MaxHeap-ului va conține:

- ☐ (39, 27, 30, 10, 4, 21)
- ☐ (39, 21, 30, 10, 4, 27)
- ☐ Niciuna, deoarece tabloul inițial nu satisface proprietatea de MaxHeap.
- ☐ (39, 30, 27, 10, 4, 21)
- ☐ (42, 39, 30, 10, 4, 27)

Se consideră un algoritm a cărui complexitate timp este dată de expresia $T(n) = 10^3 n^2 + n \log n$. Care dintre următoarele afirmații sunt false?

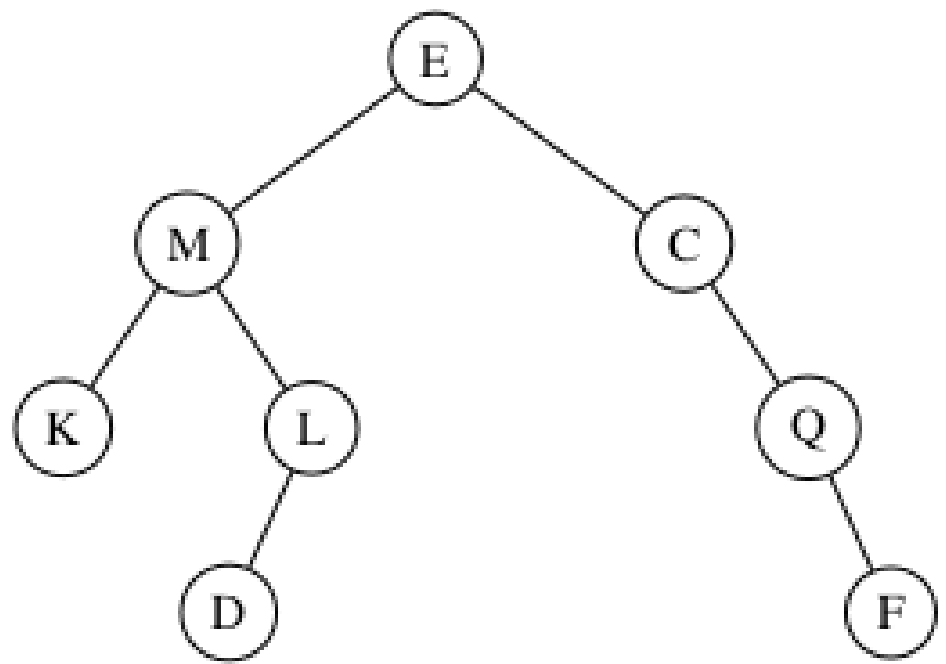
- ☐ $T(n) = \Theta(n^2)$
- ☐ $T(n) = O(n^2)$
- ☐ $T(n) = \Omega(n^2 \log n)$
- ☐ $T(n) = O(n^2 \log n)$
- ☐ $T(n) = \Theta(n \log n)$

Se consideră subprogramul de mai jos unde n este un parametru întreg și strict pozitiv. În care din următoarele clase poate fi încadrată complexitatea timp a acestui subprogram?

```
procedure f(n)
begin
  if (n == 1) then print n
  else {
    f(n-1)
    print n
    f(n-1)
  }
end
```

- ☐ $\Omega(n)$
- ☐ $\Theta(n)$
- ☐ $\Theta(2^n)$
- ☐ $O(n)$
- ☐ $O(2^n)$

Se consideră arborele binar din figura de mai jos. Care din următoarele afirmații sunt false?



- ☐ Parcugerea inordine a arborelui este: K, M, D, L, E, C, Q, F.
- ☐ Primele două vârfuri din parcugerea postordine a arborelui sunt F și D.
- ☐ Ultimele două vârfuri din parcugerea BFS a arborelui sunt D și F.
- ☐ Primele trei vârfuri din parcugerea preordine a arborelui sunt E, M și C.
- ☐ Frontiera arborelui este formată din 2 vârfuri.

Considerăm următorul vector $a=[51, 23, 62, 45, 16, 30, 92, 71]$. Aplicați pasul de partiționare al algoritmului Quick Sort cu pivotul $a[0]$. Care din următoarele afirmații sunt adevărate?

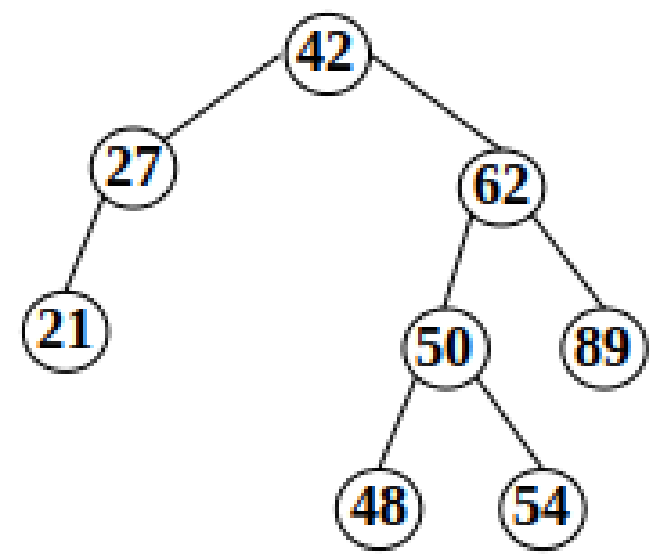
- ☐ Cele două subsecvențe rezultate sunt: $[16, 23, 30, 45]$, $[62, 92, 71]$.
- ☐ După această etapă, pivotul se va afla la indexul 4 în vector.
- ☐ După aceasta etapă, valoarea maximă se află pe ultima poziție în vector.
- ☐ Complexitatea timp a algoritmului Quick Sort pentru un vector cu n elemente, ordonat crescător, în cazul în care pivotul se alege sistemic a fi elementul din mijloc este $O(n \log n)$.

Considerăm funcția de mai jos. Funcția $\text{partitioneaza}(a, n, i, j)$ reprezintă funcția de partiționare folosită în cadrul algoritmului de sortare rapidă Quick Sort și returnează poziția pivotului. Care din următoarele afirmații sunt adevărate?

```
function test(a, n, k)
begin
    i ← 0; j ← n-1
    while i < j do
        x ← partitioneaza(a, n, i, j)
        if x == k then return a[k]
        else
            if x > k then j ← x - 1
            else i ← x + 1
    return a[k]
end
```

- ☐ Funcția calculează al k-lea cel mai mic element dacă verificarea de pe ramura else este $\text{if } x < k$.
- ☐ Funcția calculează al k-lea cel mai mic element.
- ☐ Funcția sortează crescator primele k elemente ale unui vector.
- ☐ Funcția calculează al k-lea cel mai mare element.
- ☐ La finalul fiecărei iterări, valorile $a[0], \dots, a[i]$ sunt ordonate crescător.

Considerați următorul arbore AVL echilibrat. Ștergeți valoarea 27. Care din următoarele afirmații sunt adevărate?



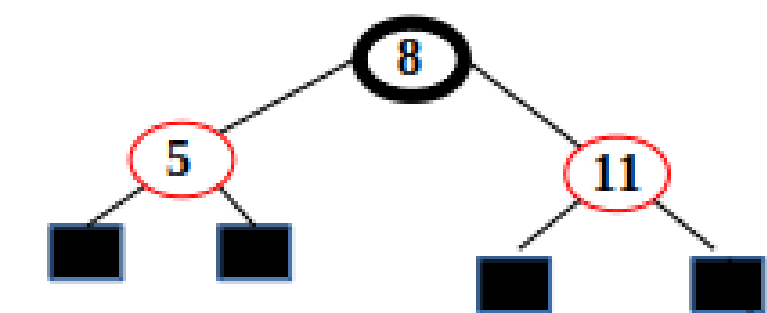
- ☐ Este necesară o operație de rotație simplă la dreapta în rădăcină pentru a restabili proprietatea AVL.
- ☐ Parcurgerea pe nivele a arborelui final este 50, 42, 62, 21, 48, 54, 89.
- ☐ Este necesară o operație de rotație dublă stânga pentru a restabili proprietatea AVL.
- ☐ Nu sunt necesare operații de rotație pentru a restabili proprietatea AVL.
- ☐ Factorul de echilibare al nodului 27 înainte de ștergere este 0.

Fie următoarea secvență de pseudocod în care S este o stivă, iar C este o coadă nevidă. Care dintre următoarele afirmații sunt adevărate?

```
S <- stivaVida()
while (not esteVida(C)) do {
    aux <- citeste(C); elimina(C); push(S, aux)
}
while (not esteVida(S)) do {
    aux <- top(S); insereaza(C, aux)
}
```

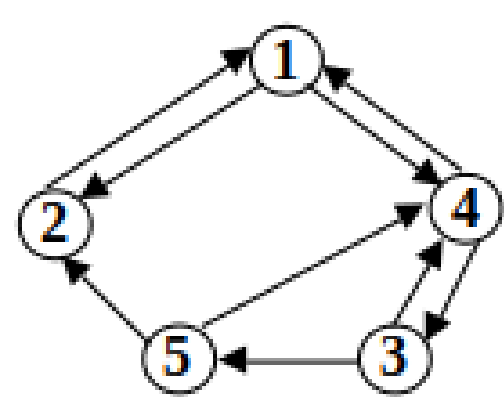
- ☐ Algoritmul lasă coada C nemodificată.
- ☐ După execuția algoritmului, coada C este vidă.
- ☐ Algoritmul rulează la nesfârșit.
- ☐ Algoritmul inversează ordinea elementelor în coada C.

Fie următorul arbore bicolor. Inserăm valoarea 15, și apoi 14. Care din următoarele afirmații sunt adevărate?



- ☐ După inserarea valorii 14, este necesară o operație de rotație și recolorări.
- ☐ După cele două inserări, nodul 5 este colorat în roșu.
- ☐ După inserarea valorii 15, este necesară o operație de rotație și recolorări.
- ☐ Înălțimea neagră a nodului rădăcină după inserare este 1.
- ☐ După cele două inserări, nodurile 11 și 15 sunt colorate în roșu.

Fie digraful $D=(V,A)$ de mai jos, reprezentat prin liste de adiacență (nodurile sunt memorate în ordine crescătoare în cadrul listelor de adiacență). Care dintre următoarele afirmații sunt adevărate?



- ☐ Componentele tare conexe ale digrafului D sunt {1,3,4,5}, {2}.
- ☐ Complexitatea timp a algoritmului de parcurgere BFS a digrafului D este $\Theta(|V|\cdot|A|)$.
- ☐ Parcurgerea în adâncime a digrafului D din nodul 1 este 1,2,4,3,5.
- ☐ Suma gradelor interioare ale vârfurilor unui digraf oarecare este egală cu numărul de vârfuri ale digrafului.
- ☐ Digraful D este tare conex.

Se consideră funcția f de mai jos, unde x și y sunt doi parametri întregi și strict pozitivi. Care dintre următoarele afirmații sunt adevărate?

```
function f(x, y)
begin
  z <- 0;
  while (x > 0) do {
    if x%2 == 1 then z <- z+y
    x <- x/2; y <- y*2
  }
  return z
end
```

- ☐ Apelul $f(25,4)$ returnează valoarea 124.
- ☐ Complexitatea timp a algoritmului poate fi încadrată în clasa $O(\log x)$.
- ☐ Apelul $f(25,4)$ returnează valoarea 24.
- ☐ Complexitatea timp algoritmului poate fi încadrată în clasa $\Theta(x)$.
- ☐ Apelul $f(25,4)$ returnează valoarea 100.

Fie o tabelă de dispersie externă de dimensiune $m=10$ și $h(k) = (2k+1) \bmod m$. Inerați în ordine valorile 8, 122, 127, 42, 13, 15. Care din următoarele afirmații sunt adevărate?

- ☐ Valorile 8 și 13 sunt inserate în lista asociată indicelui 7.
- ☐ Numărul maxim de elemente care pot fi inserate în tabela de dispersie de mai sus este egal cu 10.
- ☐ Complexitatea timp în cazul cel mai nefavorabil pentru a insera n valori într-o tabelă de dispersie externă inițial vidă, unde elementele în cadrul listelor sunt păstrate în ordine crescătoare, este $O(n)$.
- ☐ Factorul de încărcare al tabelii după ce au fost inserate valorile este egal cu 1.
- ☐ Cea mai mare listă înlănțuită are dimensiunea 3 și se află la indexul 5.

Timpul de execuție al unui algoritm recursiv se exprimă prin relația de recurență $T(n) = 2 T(n/3) + \log_3 n$, unde $\log_3 n$ reprezintă logaritmul în baza 3 din n. În care din următoarele clase de complexitate poate fi încadrată soluția acestei recurențe? (Toți logaritmii de mai jos sunt exprimați în baza 3.)

- ☐ $T(n) = \Theta(\log n)$
- ☐ $T(n) = \Theta(n)$
- ☐ $T(n) = \Theta((\log n)^2)$
- ☐ $T(n) = \Theta(n^{\log 2})$
- ☐ $T(n) = \Theta(n \log n)$

Fie următorul algoritm pentru inserarea unui element e într-o listă liniară L implementată cu structuri înlănțuite. Vom considera poziția k o valoare între 0 și n-1 (n fiind numărul de elemente din listă), iar lista nevidă. Care din următoarele afirmații sunt adevărate?

```
procedure insereaza(L, k, e)
begin
  new(p); p->elt <- e
  if k == 0 then { p->succ <- L.prim; L.prim <- p }
  else
    if k == 1 then { p->succ <- L.prim->succ; L.prim->succ <- p }
    else
      {
        q <- L.prim; i <- 0
        while (i < k-2 and q != L.ultim ) do
          { q <- q->succ; i <- i+1 }
        p->succ <- q->succ; q->succ <- p
        if (q == L.ultim) then L.ultim <- p
      }
  end
end
```

- ☐ Algoritmul este greșit deoarece inserează elementul e pe poziția k-2, atunci când k>1.
- ☐ Algoritmul este greșit deoarece inserează elementul e pe poziția k-1, atunci când k>1.
- ☐ Algoritmul este corect și inserează elementul k pe poziția e.
- ☐ Algoritmul este greșit deoarece atunci când k = n-1 actualizarea câmpului L.ultim nu se face corect.
- ☐ Algoritmul este corect și inserează elementul e pe poziția k.

Fie funcția de mai jos, unde x este adresa unui nod dintr-un arbore AVL echilibrat, iar x, x->drp, x->drp->stg sunt diferiți de NULL. Care din următoarele afirmații sunt adevărate?

```
function rotate(x)
begin
  y <- x->drp
  z <- y->stg
  x->drp <- z
  y->stg <- z->drp
  z->drp <- y
  x->drp <- z->stg
  z->stg <- x
  return z
end
```

- ☐ Funcția realizează o rotație simplă la dreapta în x.
- ☐ Funcția realizează o rotație simplă la dreapta în fiul stâng al lui x.
- ☐ Funcția realizează o rotație simplă la stânga în x.
- ☐ Funcția realizează o rotație dublă la stânga (dreapta, stânga) în x.
- ☐ Funcția realizează o rotație dublă la dreapta (stânga, dreapta) în x.

