

1.

Se consideră algoritmul de mai jos. Care dintre următoarele afirmații sunt adevărate?

```
function f(n)
begin
  s ← 0; k ← 1
  while k < n do {
    s ← s+1; k ← 2*k
  }
  return s
end
```

$$1 < \log m < \sqrt{m} < m < m \log m < m^2 < \dots < 2^m$$

$\bigcirc \rightarrow \limsup$   $\Omega \rightarrow \liminf$   
 $\bigoplus \rightarrow \limmed$

- ☒ Complexitatea timp a algoritmului se încadrează în clasa  $O(2^n)$ .
- ☐ Apelul  $f(9)$  va returna valoarea 3.
- ☒ Apelul  $f(15)$  va returna valoarea 4.
- ☒ Complexitatea timp a algoritmului se încadrează în clasa  $O(\log n)$ .
- ☐ Complexitatea timp a algoritmului se încadrează în clasa  $\Omega(n)$ .

$$\log m < 2^m$$

2.

În care din următoarele clase poate fi încadrată complexitatea următorului algoritmului?

```
function acc(n)
begin
  c ← 0
  for i ← 0 to n*n-1 do
    for j ← i to n-1 do
      for k ← 0 to 4 do
        c ← c+2
      end
    end
  end
```

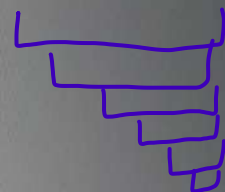
$$i \rightarrow m^2$$

$$i \rightarrow m^2 - 1$$

$$i \rightarrow m - 1$$

$$0 \rightarrow 4$$

$$1+2+3+4+5+\dots+m$$



$$for(1) \sim i \rightarrow m^2 \quad O(m^2)$$

$$for(2) \rightarrow m$$

$$for(3) \rightarrow +5$$

$$\left( \sum_{i=0}^{m-1} \sum_{j=i}^{m-1} \sum_{k=0}^4 1 + \sum_{i=m}^{m^2 \cdot m} \right) = m^2 + m$$

$$O(m^2 + m)$$

$$O(m^2)$$

- ☐  $O(n)$
- ☐  $\Theta(n^3)$
- ☒  $O(n^2)$
- ☒  $\Omega(n)$
- ☒  $O(n^3)$

3. Un algoritm are complexitatea timp definită astfel:  $T(n) = 10^3 n^2 \log n - 7n + \log n$ . Care dintre următoarele afirmații sunt false?

- ☐  $T(n) = O(n^3)$   
☒  $T(n) = \Theta(n^2)$   
☒  $T(n) = O(n \log n)$   
☐  $T(n) = \Omega(1)$   
☐  $T(n) = \Omega(n^2)$

4. Timpul de execuție al unui algoritm recursiv se exprimă prin relația de recurență  $T(n) = 3T(\frac{n}{2}) + n^2$ . În care din următoarele clase de complexitate poate fi încadrată soluția acestei recurențe?

- ☐  $T(n) = \Theta(n^{\log_2 3} \log n)$   
☐  $T(n) = \Theta(n^{\log_2 3})$   
☐  $T(n) = \Theta(\log n)$   
☐  $T(n) = \Theta(n^2 \log n)$   
☒  $T(n) = \Theta(n^2)$

$$a=3, b=2, \log_2 3 < 2, K=2$$

$$\Theta(n^2)$$

5. Se consideră relația de recurență  $T(n) = 2T(\frac{n}{2}) + n \log n$ . Care dintre următoarele afirmații sunt adevărate?

- ☐ Recurența verifică ipoteza cazului 1 al teoremei Master și are soluția  $\Theta(n)$ .  
☐ Recurența verifică ipoteza cazului 2 al teoremei Master și are soluția  $\Theta(n \log^2 n)$ .  
☒ Recurența nu se încadrează în niciunul din cazurile teoremei Master.  
☐ Recurența verifică ipoteza cazului 3 al teoremei Master și are soluția  $\Theta(n \log n)$ .

$$a=2, b=2, \text{nilăgim}$$

$$1=1, K=1$$

6. Se consideră funcția de mai jos cu  $n$  având valori întregi, pozitive. În care din următoarele clase poate fi încadrată complexitatea timp a acesteia?

function f(n)

begin



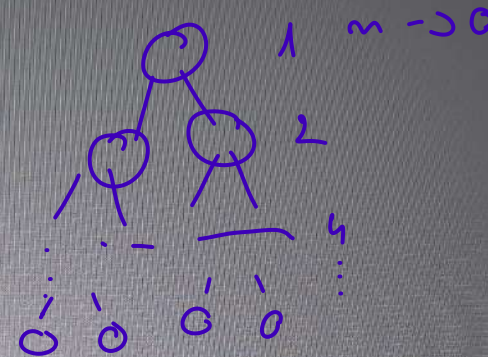
5. Se consideră relația de recurență  $T(n) = 2T\left(\frac{n}{2}\right) + n \log n$ . Care dintre următoarele afirmații sunt adevărate?

- ☐ Recurența verifică ipoteza cazului 1 al teoremei Master și are soluția  $\Theta(n)$ .
- ☐ Recurența verifică ipoteza cazului 2 al teoremei Master și are soluția  $\Theta(n \log^2 n)$ .
- ☒ Recurența nu se încadrează în niciunul din cazurile teoremei Master.
- ☐ Recurența verifică ipoteza cazului 3 al teoremei Master și are soluția  $\Theta(n \log n)$ .

6. Se consideră funcția de mai jos cu  $n$  având valori întregi, pozitive. În care din următoarele clase poate fi încadrată complexitatea timp a acesteia?

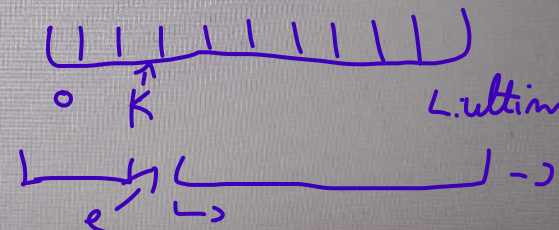
```
function f(n)
begin
  if n == 0 then return 1
  else return 1 + f(n-1) + f(n-1)
end
```

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



7. Fie următorul algoritm pentru inserarea unui element într-o listă liniară implementată cu tablouri. Care dintre următoarele afirmații sunt adevărate?

```
procedure insereaza(L, k, e)
begin
  if (k < 0 or k > L.ultim+1) then throw "Eroare: pozitie incorecta"
  if (L.ultim >= Max-1) then throw "Eroare: spatiu insuficient"
  L.ultim <- L.ultim+1
  for j <- L.ultim downto k+1 do
    L.tab[j] <- L.tab[j-1]
  L.tab[k] <- e
end
```



```

procedure insereaza(L, k, e)
begin
  if (k < 0 or k > L.ultim+1) then throw "Eroare: pozitie incorecta"
  if (L.ultim == Max-1) then throw "Eroare: spatiu insuficient"
  L.ultim <- L.ultim+1
  for j <- L.ultim downto k+1 do
    L.tab[j] <- L.tab[j-1]
  L.tab[k] <- e
end

```

- ☐ Algoritmul este greșit deoarece vechiul element de pe poziția k nu este deplasat pe poziția k+1.
- ☐ Algoritmul este greșit deoarece linia `L.ultim <- L.ultim+1` ar trebui mutată la sfârșit (înainte de instrucțiunea `end`).
- ☒ Algoritmul este corect și inserează elementul e pe poziția k.
- ☐ Algoritmul este greșit deoarece pune pe toate pozițiile de după k aceeași valoare.
- ☐ Algoritmul este corect și inserează elementul k pe poziția e.

8. Fie următoarea procedură pentru ștergerea tuturor aparițiilor elementului e dintr-o listă liniară L implementată cu structuri înlanțuite. Care din următoarele afirmații sunt adevărate?

```

procedure elimina(L, e)
begin
  while (L.prim != NULL and L.prim->elt == e) do {
    q ← L.prim; L.prim ← L.prim->succ; delete(q)
  }
  if (L.prim == NULL) then L.ultim ← NULL
  else {
    p ← L.prim
    while (p != NULL) do {
      q ← p->succ;
      if (q != NULL and q->elt == e) then {
        p->succ ← q->succ; delete(q)
        if (L.ultim == q) then L.ultim ← p;
      }
    }
  }
end

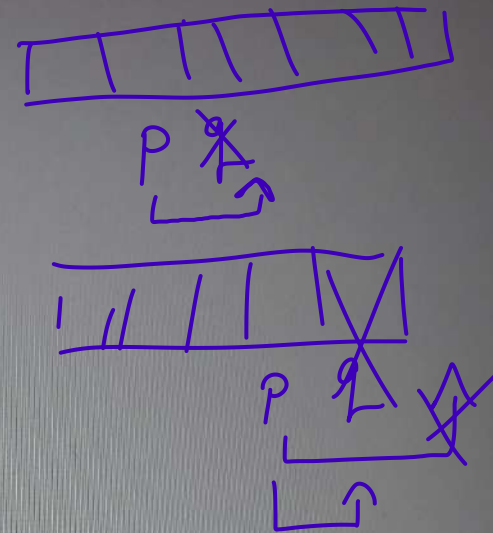
```



```

while (L.prim != NULL and L.prim->elt == e) do {
    q ← L.prim; L.prim ← L.prim->succ; delete(q)
}
if (L.prim == NULL) then L.ultim ← NULL
else {
    p ← L.prim
    while (p != NULL) do {
        q ← p->succ;
        if (q != NULL and q->elt == e) then {
            p->succ ← q->succ; delete(q)
            if (L.ultim == q) then L.ultim ← p;
        }
        p ← p->succ
    }
}
end

```



- ☐ Algoritmul este greșit deoarece șterge elementele succesoare nodurilor ce corespund aparițiilor elementului e din lista L.
- ☐ Algoritmul este greșit deoarece nu parcurge toată lista L.
- ☐ Algoritmul este greșit deoarece nu șterge toate aparițiile elementului e din lista L.
- ☐ Algoritmul este corect și șterge toate aparițiile elementului e din lista L.
- ☒ Algoritmul este greșit deoarece la sfârșit L.ultim nu va conține adresa ultimului element din lista L.

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

```

C <- coadaVida()
while (not esteVida(S)) do {
    aux <- top(S); pop(S)
}

```

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

```

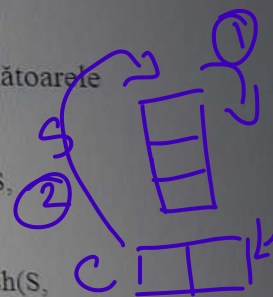
C <- coadaVida()
while (not esteVida(S)) do {
    aux <- top(S); pop(S)
    insereaza(C, aux)
}
while (not esteVida(C)) do {
    aux <- citește(C)
    push(S, aux)
}

```

- ☐ Algoritmul inversează ordinea elementelor în stiva S.
- ☐ Algoritmul lasă stiva S nemodificată.
- ☐ După execuția algoritmului, stiva S este vidă.
- ☒ Algoritmul rulează la nesfârșit.

10. Fie C o coadă ce conține 2 elemente, iar S o stivă ce conține 3 elemente. Care dintre următoarele patru secvențe de operații interschimbă conținutul celor două structuri?

- ☒ insereaza(C, top(S)); pop(S); insereaza(C, top(S)); pop(S); insereaza(C, top(S)); pop(S); push(S, citește(C)); elimina(C); push(S, citește(C)); elimina(C)
- ☐ insereaza(C, top(S)); pop(S); insereaza(C, top(S)); pop(S); push(S, citește(C)); elimina(C); push(S, citește(C)); elimina(C); push(S, citește(C)); elimina(C)
- ? ☒ e <- top(S); pop(S); insereaza(C, top(S)); pop(S); insereaza(C, top(s)); insereaza(C, e); pop(S); push(S, citește(C)); elimina(C); push(S, citește(C)); elimina(C)
- ☐ push(S, citește(C)); elimina(C); insereaza(C, top(S)); pop(S); push(S, citește(C)); elimina(C); insereaza(C, top(S)); pop(S); insereaza(C, top(S)); pop(S)



11. Fie L o listă liniară ordonată. Dacă utilizăm cele două implementări prezentate la curs (cu

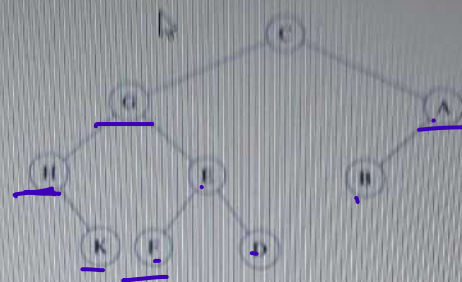


$T$   $L$   
 $I: O(n) \quad O(n) \checkmark$   
 $E: O(n) \quad O(n) \checkmark$   
 $C: O(\log n) \quad O(n)$   
 $O(1) \text{ pt}$   
 $\in \text{lgm lista tota}$

... și cu liste înlănțuite. Care din următoarele afirmații sunt adevărate?

- ☐ Inserarea unui nou element în lista  $L$  are complexitatea  $O(1)$  în cazul tablourilor și  $O(n)$  în cazul listelor înlănțuite  $O(1)$   $O(n)$
- ☒ Eliminarea elementului minim din lista  $L$  are complexitatea  $O(n)$  în cazul tablourilor și  $O(1)$  în cazul listelor înlănțuite
- ☒ Inserarea unui nou element în lista  $L$  are complexitatea  $O(n)$  în cazul tablourilor și  $O(n)$  în cazul listelor înlănțuite
- ☐ Inserarea unui nou element în lista  $L$  are complexitatea  $O(n)$  în cazul tablourilor și  $O(1)$  în cazul listelor înlănțuite
- ☐ Eliminarea elementului minim din lista  $L$  are complexitatea  $O(1)$  în cazul tablourilor și  $O(n)$  în cazul listelor înlănțuite

12. Se consideră arbore binar de mai jos. Care dintre următoarele afirmații sunt false?



$in: SRD$   
 $post: SDR$   
 $\leftarrow HFD E G$   
 $pre: RSD$

- ~~A~~ ☐ Primul și ultimul nod din parcurgerea inordine sunt, respectiv, H și A.
- ~~F~~ ☒ Parcurgerea BFS este K H G F E D C B A.
- ~~F~~ ☒ Primele trei noduri din parcurgerea postordine sunt K, H și G.
- ~~F~~ ☒ Ultimele două noduri din parcurgerea preordine sunt D și B.

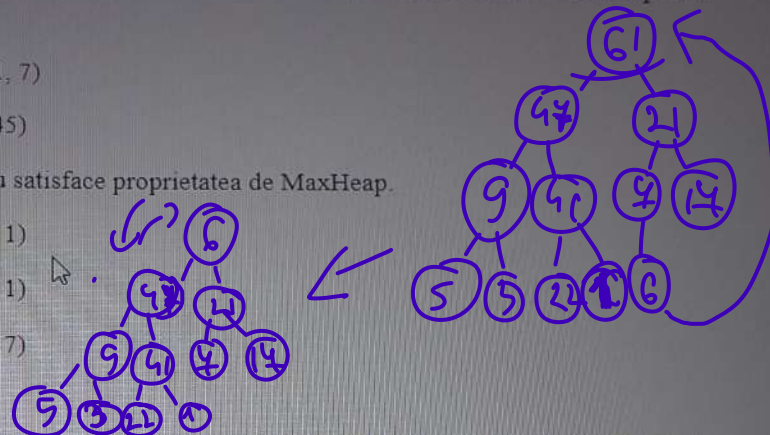
13. Fie următorul MaxHeap implementat cu tablouri: (61, 47, 21, 9, 41, 7, 17, 5, 3, 22, 1, 6). După o operație de eliminare, urmată de o operație de inserare a cheii 45, tabloul asociat MaxHeap-ului va conține:

- ☐ (61, 47, 45, 9, 41, 21, 17, 5, 3, 22, 1, 7)
- ☐ (47, 41, 21, 9, 22, 7, 17, 5, 3, 6, 1, 45)
- ☐ Niciunul, deoarece tabloul initial nu satisface proprietatea de MaxHeap.

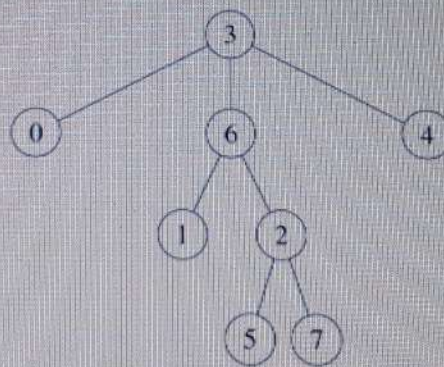
Ultimele două noduri din parcurgerea preordine sunt D și B.

13. Fie următorul MaxHeap implementat cu tablouri: (61, 47, 21, 9, 41, 7, 17, 5, 3, 22, 1, 6). După o operație de eliminare, urmată de o operație de inserare a cheii 45, tabloul asociat MaxHeap-ului va conține:

- ☐ (61, 47, 45, 9, 41, 21, 17, 5, 3, 22, 1, 7)
- ☐ (47, 41, 21, 9, 22, 7, 17, 5, 3, 6, 1, 45)
- ☐ Niciunul, deoarece tabloul initial nu satisface proprietatea de MaxHeap.
- ☐ (47, 45, 41, 22, 21, 17, 9, 7, 6, 5, 3, 1)
- ☐ (47, 45, 41, 21, 9, 22, 7, 17, 5, 3, 6, 1)
- ☒ (47, 41, 45, 9, 22, 21, 17, 5, 3, 6, 1, 7)



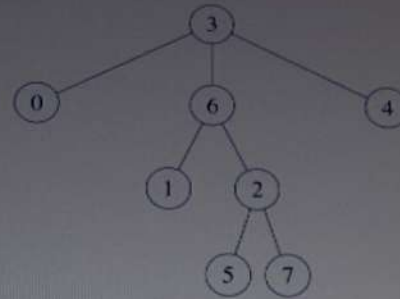
14. Fie structura union-find ponderată de mai jos ( $n = 8$ ). După apelul funcției `find(2)`, urmată de aplatizare, care dintre următoarele afirmații sunt adevărate?



- ☒ Rezultatul unei operații `find` va fi același pentru toate elementele din mulțimea univers.
- ☐ Părintele lui 1 este 3.
- ☐ Vectorul de părinți este (3, 6, 3, -1, 3, 2, 3, 2).
- ☒ Numărul de fii ai rădăcinii este 4.

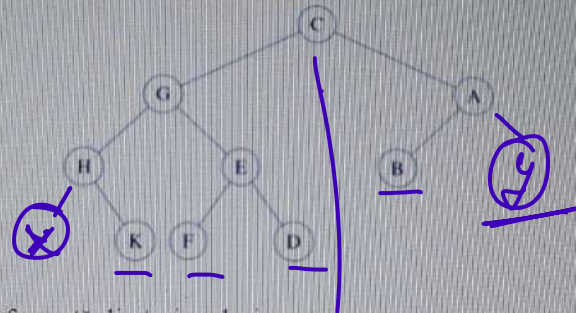


14. Fie structura union-find ponderată de mai jos ( $n = 8$ ). După apelul funcției `find(2)`, urmată de aplatizare, care dintre următoarele afirmații sunt adevărate?



- ☐ Rezultatul unei operații `find` va fi același pentru toate elementele din mulțimea univers.
- ☐ Părintele lui 1 este 3.
- ☐ Vectorul de părinți este (3, 6, 3, -1, 3, 2, 3, 2).
- ☐ Numărul de fii ai rădăcinii este 4.

15. Se consideră arborele binar de mai jos. Care dintre următoarele afirmații sunt false?



- F ☒ Frontiera arborelui este formată din trei noduri.
- A ☐ Numărul minim de noduri ce trebuie adăugate astfel încât arborele rezultat să fie arbore binar propriu este 2.
- F ☒ În urma inserării nodului X ca fiu stânga a lui H și a nodului Y ca fiu dreapta a lui A arborele rezultat este arbore binar complet.
- F ☒ În urma eliminării nodurilor K, F și D arborele rezultat este arbore binar propriu.