

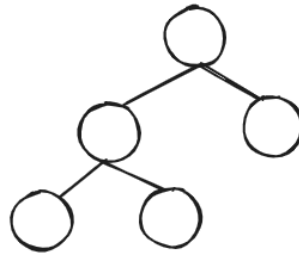
# Model Examen – Seria 13

## Rezolvări

### 1 AVL Trees

**Cerinta:** Care e inaltimea minima a unui arbore AVL cu 5 noduri? Presupunem ca inaltimea unui arbore cu un nod este 0.

Inaltimea minima este 2. Exista mai multe moduri de a desena structura arborelui.



### 2 Minimum Spanning Trees

**Cerinta:** In algoritmul lui Kruskal:

- Sortam muchiile in ordinea crescatoare a ponderilor.
- Mai multi arbori pot coexista in decursul executiei algoritmului.
- O muchie poate fi adaugata si apoi eliminata.
- Raspunsurile nu sunt corecte.

Algoritmul mai intai prelucreaza datele sortand muchiile crescator in functie de ponderi, iar apoi itereaza prin multimea sortata, adaugand muchia curenta numai daca nu formeaza un ciclu. Astfel, raspunsurile **a** si **b** sunt corecte.

### 3 Tree-walks

**Cerinta:** Traversarea in postordine a unui arbore binar de cautare este **15, 10, 23, 25, 20, 35, 42, 39, 30**. Care este traversarea in inordine pentru acelasi arbore?

Cum traversarea in inordine este mereu multimea de valori sortata, obtinem **10, 15, 20, 23, 25, 30, 35, 39, 42**. Observatie: faptul ca traversarea pe care ne-a dat-o este in postordine este o informatie inutila.

## 4 AVL Trees

**Cerinta:** Care dintre urmatoarele afirmatii sunt adevarate intr-un arbore AVL?

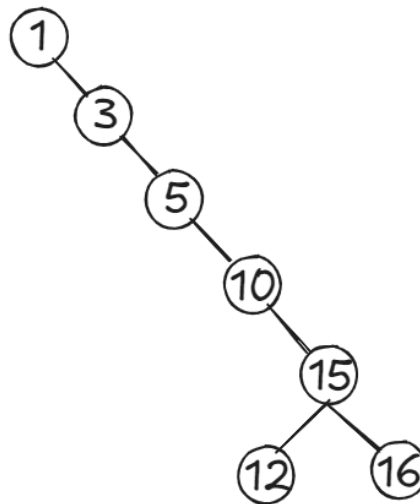
- Succesorul unui nod este intotdeauna un nod frunza.
- Rotatiile simple sunt uneori folosite pentru a restabili echilibrul arborelui.
- Rotatiile duble sunt uneori folosite pentru a restabili echilibrul arborelui.
- Succesorul unui nod este intotdeauna fie un nod frunza, fie un nod fara copil drept.

Pentru un nod oarecare, succesorul poate fi un nod frunza, un nod fara copil stang sau un stramos. Astfel, variantele legate de succesor sunt gresite. Variantele corecte sunt **b** si **c**, deoarece arborii AVL sunt bazati pe rotatii.

## 5 Binary Search Trees

**Cerinta:** Urmatoarele numere sunt inserate succesiv intr-un arbore binar de cautare gol: **1, 3, 5, 10, 15, 12, 16**. Care este inaltimea arborelui la final?

Inaltimea arborelui este 5.



## 6 Hash Tables

**Cerinta:** Sa consideram urmatoarele elemente: **4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199** si functia hash  $h(x) = x \bmod 100$ . Cum sunt mapate valorile si care este incarcarea tablei?

Conform functiei hash, ar trebui ca tabela sa aiba slot-urile  $0 \rightarrow 99$ , deci avem 100 de slot-uri in total. Valorile sunt mapate in felul urmator:

- $22 \rightarrow 4322$
- $34 \rightarrow 1334$
- $71 \rightarrow 1471, 6171$
- $73 \rightarrow 6173$

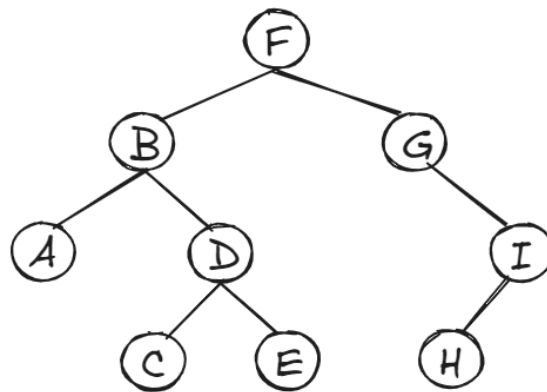
- $79 \rightarrow 9679$
- $89 \rightarrow 1989$
- $99 \rightarrow 4199$

Incarcarea tabelii este  $\frac{m}{n}$ , unde  $m$  reprezinta numarul de elemente din tabela, iar  $n$  este numarul total de slot-uri. In total, sunt 8 elemente  $\rightarrow$  incarcarea tabelii este  $\frac{8}{100} = 8\%$ .

## 7 Tree-walks

**Cerinta:** Care din urmatoarele secvente **nu** este una din traversarile in preordine sau postordine ale arborelui din figura de mai jos?

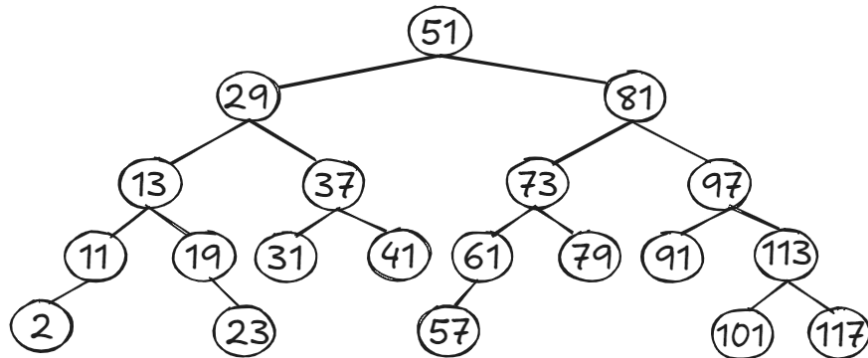
- A, B, C, D, E, F, G, H, I.
- F, B, D, A, C, E, G, H, I.
- F, B, G, A, D, I, C, E, H.
- Raspunsurile nu sunt corecte.



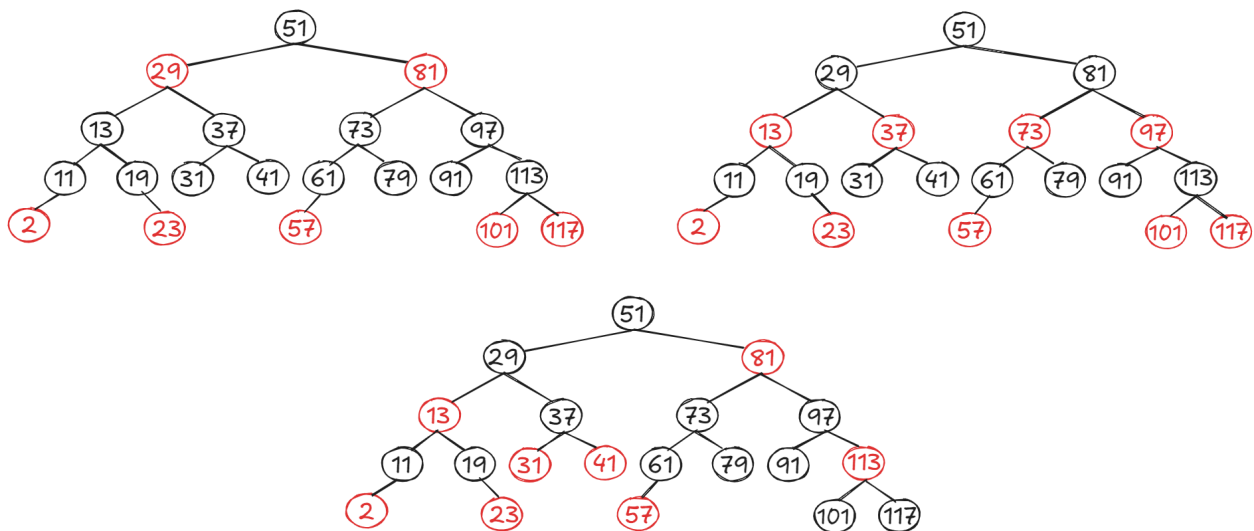
Prima varianta este traversarea in inordine, ceea ce inseamna ca **a** este corect. A doua traversare este una aleatoare  $\rightarrow$  **b** este corect. Urmatoarea traversare este BFS  $\rightarrow$  varianta **c** este corecta.

## 8 Red-Black Trees

**Cerinta:** Care noduri sunt colorate in mod sigur cu **negru**, daca stim ca arborele de mai jos este un red-black tree?



Cea mai simpla rezolvare este sa incercam sa desenam arborii. Nodurile care sunt colorate cu rosu in cel putin o reprezentare sunt scoase din ecuatie. De asemenea, implicit stim ca **51** este colorat cu negru, deoarece este radacina.



Nodurile care vor fi colorate in mod sigur cu negru sunt **51, 11, 19, 61, 79, 91**.

## 9 Order Statistics Trees

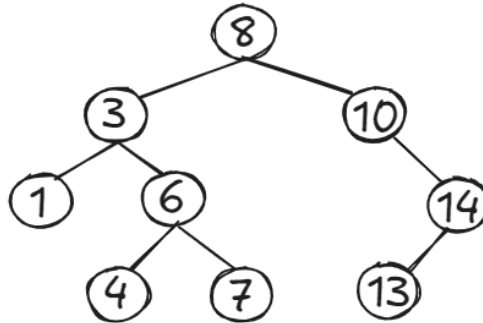
**Cerinta:** De ce adaugam campul "tree-size" la un arbore order statistics?

Pentru a putea face interogari de tipul "care este al k-lea cel mai mic element?" sa ruleze in timp  $O(\log n)$ .

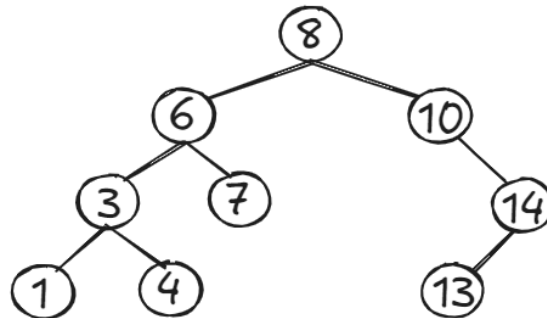
## 10 Rotations

**Cerinta:** Sa consideram urmatorul arbore binar de cautare. Dupa ce am rotit nodul 6 in jurul nodului 3:

- a. Nodul 1 este copil al nodului 3.
- b. Nodul 4 este copil al nodului 3.
- c. Marimea subarborelui nu se schimba, dar radacina sa da.
- d. Raspunsurile nu sunt corecte.



Variantele corecte sunt **a**, **b** si **c**. Am atasat desenul mai jos.



## 11 Stacks

**Cerinta:** Care din urmatoarele secvente de operatii sunt imposibile intr-o stiva cu patru elemente?

- a. PUSH, POP, POP, POP, POP, PUSH.
- b. PUSH, POP, POP, POP, PUSH, POP, POP.
- c. PUSH, POP, POP, POP, POP, POP, PUSH, POP.
- d. POP, POP, POP, POP, POP, POP, POP.
- e. PUSH, PUSH, PUSH, PUSH, PUSH, PUSH, PUSH.

Singura varianta corecta este **d**.

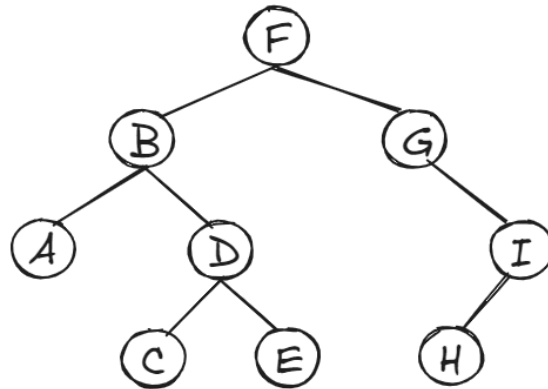
## 12 Hash Tables

**Cerinta:** Cand numarul de slot-uri dintr-o tabela de dispersie se tripleaza, iar numarul de elemente se dubleaza, ce se intampla cu incarcarea tablei?

Incarcarea tablei are formula  $\frac{m}{n}$ , unde  $m$  reprezinta numarul de elemente din tabela, iar  $n$  reprezinta incarcarea tablei. Daca numarul de slot-uri se tripleaza, obtinem  $\frac{m}{3n}$ . Mai departe, daca numarul de elemente se dubleaza, obtinem  $\frac{2m}{3n} \rightarrow$  incarcarea tablei scade la  $\frac{2}{3}$  din cea initiala.

## 13 Binary Search Trees

**Cerinta:** Presupunand ca arborele de mai jos e unul de cautare, unde se afla al treilea element?



Facem parcurgerea in inordine si ne oprim la al treilea element: A, B, C.

## 14 Splay Trees

**Cerinta:** Care din urmatoarele afirmatii sunt adevarate intr-un arbore splay?

- Subarboarele din stanga si din dreapta radacinii au aceeasi inaltime.
- Subarboarele din stanga si din dreapta fiecarui nod au inaltime care pot diferi cu cel mult 1 in valoare absoluta.
- Fiecare nod este colorat cu negru sau cu rosu.
- Raspunsurile nu sunt corecte.

Raspunsurile nu sunt corecte; prima afirmatie se refera la o proprietate de la **B-Trees**, a doua afirmatie se refera la **AVL Trees**, iar a treia afirmatie se refera la **Red-Black Trees**.

## 15 Dynamic Sets

**Cerinta:** Sa presupunem ca dorim sa listam elementele unei multimi dinamice in ordine descrescatoare. Care din urmatoarele metode sunt potrivite?

- Parcurgem arborele in inordine, punand nodurile intr-o stiva. La final le afisam pe masura ce le scoatem din stiva.
- Parcurgem arborele in preordine, punand nodurile intr-o coada. La final le afisam pe masura ce le scoatem din coada.

- c. Parcurgem arborele recursiv cu ordinea *right – vertex – left*.
- d. Raspunsurile nu sunt corecte.

**Raspuns:** Sa presupunem ca avem un arbore in care radacina este **2**, copilul stang este **1** iar copilul drept este **3**:

- Daca parcurgem arborele in inordine, obtinem stiva **1, 2, 3**. Daca le scoatem de pe stiva, obtinem **3, 2, 1**. Astfel, prima varianta este **corecta**.
- Daca parcurgem arborele in preordine, obtinem in coada **2, 1, 3**. Daca le scoatem din coada, vor fi afisate in ordinea **2, 1, 3**. Astfel, a doua varianta este **gresita**.
- Daca parcurgem arborele in ordinea *right – vertex – left*, obtinem **3, 2, 1**. Aceasta parcurgere este opusul ordinii *left – vertex – right*, care este parcurgerea in inordine; astfel, a treia varianta este corecta.

## 16 Linked Lists

**Cerinta:** Care este complexitatea gasirii elementului median intr-o lista dublu inlantuita data prin pointerul head?

- a.  $\Theta(n)$ .
- b.  $\Theta(n \log n)$ .
- c.  $\Theta(n^2)$ .
- d. Raspunsurile nu sunt corecte.

Deoarece exista algoritmi de cautare a medianei in  $O(n)$ , primul raspuns este corect.

## 17 Dynamic Sets

**Cerinta:** Vrem sa implementam operatia de gasire a elementului median intr-o multime dinamica de elemente. Vrem ca operatia sa aiba complexitate  $O(\log n)$ . De la ce structura de date ar fi cel mai bine sa pornim?

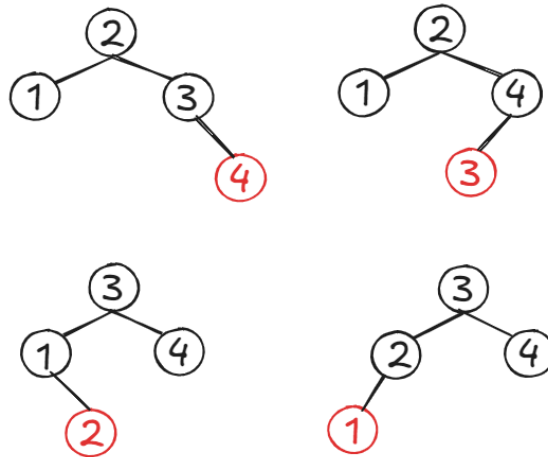
- a. Red-Black Tree.
- b. AVL Tree.
- c. Splay Tree.
- d. Order Statistics Tree.
- e. Raspunsurile nu sunt corecte.

Primii trei arbori sunt nefolositori pentru o asemenea operatie. Aceasta intrebare este una de tipul "order statistics" → varianta **d** este corecta.

## 18 Red-Black Trees

**Cerinta:** Vrem sa reprezentam multimea **1, 2, 3, 4** cu un red-black tree. In cate moduri putem face acest lucru?

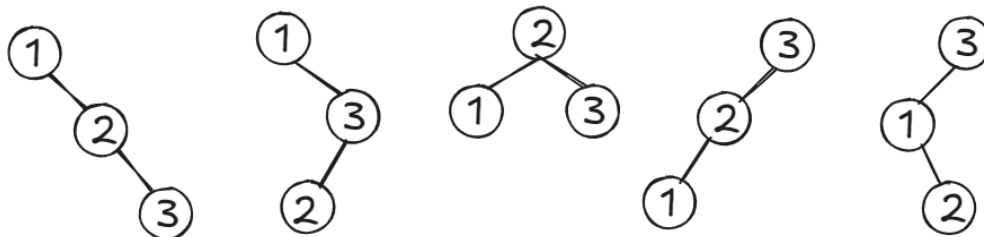
In patru moduri.



## 19 Binary Search Trees

**Cerinta:** Vrem sa reprezentam multimea **1, 2, 3** cu un arbore binar de cautare. In cate moduri putem face acest lucru?

In cinci moduri.



## 20 Skip Lists

**Cerinta:** Un skip list contine elementele **1, 2, 3, 5, 8, 13, 21, 44**. In al catelea nod vom gasi elementul 8? Deoarece skip list-urile sunt sortate crescator, vom gasi elementul 8 in al cincilea nod.



## 21 B-Trees

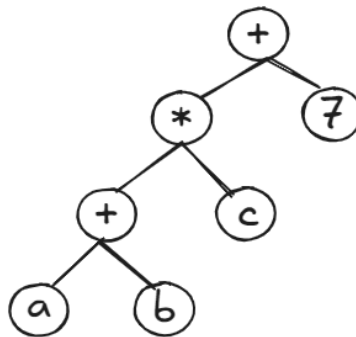
**Cerinta:** Un B-Tree de grad minim  $n$  este un arbore in care orice nod (in afara de radacina) contine:

- a. Cel mult  $2n - 1$  chei.
- b. Exact  $\frac{n-1}{2}$  chei.
- c. Cel mult  $2n$  chei.
- d. Cel mult  $\frac{n-1}{2}$  chei.

Varianta **a** este corecta.

## 22 Syntax Trees

**Cerinta:** Se da o expresie aritmetica reprezentata ca un arbore sintactic, precum in imaginea de mai jos. Care este ordinea in care sunt evaluate nodurile pentru a calcula valoarea expresiei?



Ordinea este **a, b, +, c, \*, 7, root(+)**.

## 23 Sorting Algorithms

**Cerinta:** Cand rulam algoritmul heap-sort, numarul minim de swap-uri se atinge cand:

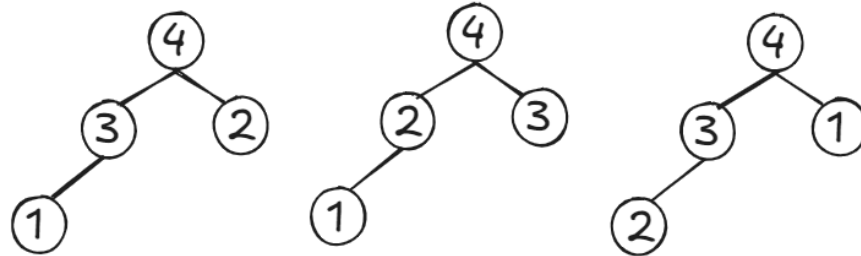
- a. Secventa initiala este sortata crescator.
- b. Secventa initiala este sortata descrescator.
- c. Secventa este una aleatoare.
- d. Raspunsurile nu sunt corecte.

Algoritmul heap-sort utilizeaza un **min-heap**: un arbore binar complet, in care fiecare nod are valoarea mai mica decat copiii sai. De fiecare data cand rulam algoritmul pe un vector de numere, mai intai trebuie sa construiasca min-heap-ul. Daca secventa este deja **sortata descrescator**, proprietatea de min-heap este indeplinita, ceea ce inseamna ca in aceasta etapa nu mai trebuie sa se efectueze swap-uri. Daca secventa este deja **sortata crescator**, ea nu indeplineste proprietatea de min-heap, iar algoritmul va trebui sa rearanjeze elementele si sa sorteze din nou. In concluzie, numarul minim de swap-uri se atinge cand secventa este sortata descrescator → varianta **b**.

## 24 Heaps

**Cerinta:** In cate moduri putem pune elementele 1, 2, 3, 4, astfel incat vectorul rezultat sa fie vazut drept un **max-heap**?

In trei moduri.



## 25 Tree-walks

**Cerinta:** Sa presupunem ca numerele 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 sunt inserate in aceasta ordine intr-un arbore binar de cautare. Care este lista elementelor in postordine?

Lista elementelor in postordine este 0, 2, 4, 3, 1, 6, 5, 9, 8, 7.

