

Nume prenume:

Grupa:

Punctaj SD IUNIE 2024 Seria 15 Numarul 2

- **Partea 1 1.5p (0.5 \* 3) MINIM 2 corecte pentru a trece**
- Partea 2 (12 grile cu raspunsuri multiple)  $12 * 0.2 = 2.4p$
- Partea 3 5 întrebări  $5 * 0.2 = 1p$
- Partea 4 Răspuns scurt 5 întrebări  $5 * 0.2 = 1p$
- Partea 5 Probleme (4 inclusiv una mai teoretică) 4.1 p
- Total 10p + Oficiu = 11p
- **Minim 5.00**

### Partea 1

- 1) Într-un min-heap faceți operațiile I(9), I(4), I(10), I(2), delete min, I(17), I(3), I(19), I(26) delete min, delete min. Arată arborele după fiecare operație.
- 2) Într-un arbore binar de căutare faceți operațiile I(8), I(3), I(1), I(6), I(10), I(14), I(4), del(6), del(1), I(7), I(9), del(8). Aratati arborele după fiecare 2 operații.
- 3) Ce se întâmplă dacă două chei diferite au aceeași valoare hash într-un tabel hash?
  - a) Valorile sunt stocate împreună în același slot.
  - b) Una dintre valori este ignorată.
  - c) Apare o coliziune și trebuie rezolvată printr-o metodă adecvată.
  - d) Tabelul hash se redimensionează automat.

### Partea 2

- 4) Care dintre următoarele secvențe de operații este invalidă într-o stivă care are inițial doua elemente?
  - a) PUSH, POP, POP, PUSH, POP, PUSH
  - b) POP, POP, PUSH, PUSH, POP, POP
  - c) POP, POP, POP, POP, POP, PUSH, POP
  - d) PUSH, PUSH, POP, POP, TOP, TOP
  - e) PUSH, PUSH, PUSH, PUSH, PUSH, PUSH
  - f) POP, POP, POP, POP, POP
- 5) Care este complexitatea în timp în cel mai rău caz pentru ștergerea unui nod dintr-un arbore binar de căutare (BST)?
  - a)  $O(1)$
  - b)  $O(\log n)$
  - c)  $O(n)$
  - d)  $O(n \log n)$
  - e)  $O(\sqrt{n})$  (facem cu Batog)

- 6) Care din următoarele structuri de date permit operații de Insert, Search, și Delete în  $O(\log n)$
- a) AVL
  - b) Binomial Heap
  - c) B-Arbori
  - d) Binary Heap
  - e) Hash
  - f) Coadă
- 7) Un arbore ternar în care fiecare nod intern (non-frunză) are exact trei copii, are 15 de frunze. Care este numărul total de noduri din arbore?
- a) 11
  - b) 22
  - c) 15
  - d) 18
  - e) 21
  - f) 22
  - g) 25
  - h) 42
- 8) Doriți să sortați  $5 \times 10^8$  numere întregi mai mici decât  $2^{32}$ . Care algoritm de sortare ar fi cel mai potrivit să utilizați?
- a) Radix Sort (baza 2)
  - b) Radix Sort (baza  $2^{16}$ )
  - c) Heap Sort
  - d) Quick Sort
  - e) Merge Sort
  - f) Counting Sort
- 9) Să presupunem că avem numerele de la 1 la 500 inserate într-un arbore binar de căutare și că dorim să căutăm valoarea 218. Care dintre următoarele secvențe NU ar putea fi secvența de noduri examinate?
- a) 250, 125, 200, 225, 210, 215, 218
  - b) 300, 150, 250, 200, 210, 220, 218
  - c) 100, 200, 250, 225, 210, 218
  - d) 250, 125, 150, 175, 190, 200, 218
  - e) 250, 375, 300, 275, 255, 230, 218
- 10) Dat fiind un arbore binar de căutare cu  $n$  noduri, care este complexitatea în timp optimă pentru a obține un vector sortat în ordine crescătoare conținând toate elementele din arbore?
- a)  $O(1)$
  - b)  $O(n)$
  - c)  $O(n \log n)$
  - d)  $O(\log(n))$
  - e)  $O(\sqrt{n})$

- 11) Care dintre următoarele afirmații sunt adevărate despre un arbore binar de căutare (BST)?
- a) Într-un BST, fiecare nod are cel mult doi copii.
  - b) Un BST poate avea mai multe niveluri incomplete.
  - c) Un BST poate avea mai multe rădăcini la un moment dat.
  - d) Un BST poate fi folosit pentru sortarea unui vector în complexitate  $O(n)$ .
  - e) Într-un BST, subarborele stâng al oricărui nod conține doar valori mai mici decât nodul respectiv, iar subarborele drept conține doar valori mai mari.
- 12) Fie  $H$  un max-heap care conține 128 de valori distincte. În câte poziții diferite se poate afla al doilea cel mai mic element?
- a) 32
  - b) 63
  - c) 64
  - d) 65
  - e) 127
- 13) Se dă un algoritm de sortare care folosește doar comparații între elemente. În cel mai bun caz, acest algoritm sortează un șir de  $n$  elemente în timp liniar  $O(n)$ . Care dintre următoarele afirmații este adevărată despre complexitatea în timp în cel mai rău caz a acestui algoritm?
- a) Complexitatea în cel mai rău caz trebuie să fie, de asemenea,  $O(n)$ .
  - b) Complexitatea în cel mai rău caz sigur este  $O(n \log n)$ .
  - c) Complexitatea în cel mai rău caz sigur este  $O(n^2)$ .
  - d) Complexitatea în cel mai rău caz poate fi  $O(1)$ .
  - e) Nu putem determina complexitatea în cel mai rău caz pe baza informației date.
- 14) Care este diferența maximă posibilă de înălțime dintre două noduri frunză într-un arbore binar complet?
- a) (a) 0
  - b) (b) 1
  - c) (c) 2
  - d) (d)  $\log_2 n$  (unde  $n$  este numărul de noduri)
  - e) (e)  $n - 1$  (unde  $n$  este numărul de noduri)
- 15) Care dintre următoarele afirmații sunt adevărate despre deque (double-ended queue)?
- a) Este o structură de date liniară.
  - b) Respectă principiul LIFO (Last In, First Out).
  - c) Respectă principiul FIFO (First In, First Out).
  - d) Poate fi implementat folosind un tablou dinamic.
  - e) Poate fi folosit pentru a implementa atât o stivă, cât și o coadă.
  - f) Operațiile de inserare și ștergere la ambele capete au complexitate  $O(1)$  în cel mai rău caz.

### Partea 3:

- 16) Insearați următoarele chei și priorități într-un treap(cu heap de maxim): (M, 85), (T, 21), (E, 102), (R, 58), (P, 16), (S, 93). Aratați arborele după fiecare inserare.
- 17) Construiți sparse table-ul (matricea din algoritmul RMQ) pentru șirul: 7, 3, 9, 1, 4, 6, 10, 2. Presupunem că în cadrul unui query am vrea să determinăm minimul pe un interval.
- 18) Insearați într-un trie cuvintele: casa, casă, mașină, masă, mare, mar. Explicați cum puteți găsi prefixul maxim comun al cuvântului "cascada" cu oricare cuvânt din lista inițială folosind trie-ul construit.
- 19) Construiți un min-heap cu 13 noduri în care pe penultimul nivel se află printre altele valorile 12, 8 și 3.
- 20) Insearați în skip list următoarele valori:  
1, 9, 3, 14, 6, 11, 19, 15, 22 La aruncarea monedei obțineți următoarele valori: B, S, S, B, S, B, S, S, B, S, S, S, B, S, B, S, S, B, B, S, B. Când obțineți B vă opriți și insearați până la acel nivel. Când obțineți S continuați la nivelul următor.

### Partea 4

- 21) Care este numărul maxim de elemente dintr-un heap binar de înălțime  $k$ ? Vom presupune ca rădăcina heap-ului se afla la înălțime 0.
- 22) Fie  $T$  un arbore binar de căutare și  $x$  un nod din  $T$  cu 2 fii. Care este numărul maxim de fii pe care îi poate avea predecesorul lui  $x$ ?
- 23) Care este complexitatea în timp, în cel mai rău caz, pentru a uni (merge) două heap-uri binare de dimensiuni  $m$  și  $n$ , într-un nou heap binar?
- 24) Se poate găsi elementul minim dintr-un max-heap în  $O(\log(n))$ ? Justificați.
- 25) O parcurgere în ordine a unui max-heap produce mereu un șir crescător? Justificați.

### Partea 5 (fiecare problema are 1p, mai puțin problema 4 care are 1.1p)

- 26) Fie  $P$  o permutare cu  $n$  elemente. Scopul este să determinăm această permutare știind pentru fiecare poziție  $i$  ( $1 \leq i \leq n$ ) câte elemente aflate înaintea ei sunt mai mici decât  $P[i]$ . Dacă există mai multe permutări posibile, se va afla cea minim lexicografică.  
Input:  $N = 4 \ 0 \ 1 \ 1 \ 0$   
Output:  $2 \ 4 \ 3 \ 1$
- 27) Se considera un șir  $S$  de paranteze rotunde (închise sau deschise) de lungime  $n$  și  $q$  query-uri  $(i, j)$ . Pentru fiecare query, să se răspundă dacă subsecvența de la  $i$  la  $j$  ( $S[i..j]$ ) este corect parantezată.
- 28) Se da un vector  $A$  cu  $N$  elemente.  $1 \leq A[i] \leq N$ . Pentru fiecare  $i$  să se găsească  $j$  minim a.i.  $A[j] < A[i]$ .
- 29) Se da un vector  $A$  cu  $N$  elemente.  $1 \leq A[i] \leq N$ . Să se numere tripletele  $(i, j, k)$  a.i.  $1 \leq i < j < k \leq N$  și  $A[i] < A[j] < A[k]$ .