Nume prenume: **Grupa:**

Punctaj **SD IUNIE 2024 Seria 15 Numărul 1**

- **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 probleme) 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(15), I(11), I(14), I(25), del (11), I(7), I(11), I(9), I(5), del(16), I(8), I(10), I(6), del(7). Aratati arborele după fiecare 2 operații.**
3. **Care dintre următoarele NU este o caracteristică a unei funcții hash bune?**
    a. **Distribuție uniformă a valorilor hash**
    b. **Cost computațional redus**
    c. **Rată ridicată de coliziuni**
    d. **Rezultat determinist**

Partea 2

4) Care dintre următoarele secvențe de operații este invalidă într-o stivă care are inițial trei 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, PUSH, POP, PUSH, TOP, TOP, TOP**
E. **PUSH, PUSH, PUSH, PUSH, PUSH, PUSH**
F. **POP, POP, POP, POP, POP, POP, POP**

5) Care dintre următoarele afirmații sunt adevărate despre ștergerea unui nod dintr-un arbore binar de căutare (BST)?

a. **Ștergerea unui nod frunză nu necesită nicio rearanjare a arborelui.**
b. **Ștergerea unui nod cu un singur copil implică înlocuirea nodului cu copilul său.**
c. **Ștergerea unui nod cu doi copii implică găsirea succesorului sau a predecesorului sau și înlocuirea nodului cu acesta.**
d. **După ștergerea unui nod, arborele rezultat poate să nu mai fie echilibrat.**
e. **Ștergerea rădăcinii unui BST este imposibilă.**

6) Care din următoarele structuri de date permit operații de Insert, Search, și Delete în O(log n)

A. AVL
B. Red Black Tree
C. Fibonacci Heap
D. Binary Heap
E. Hash
F. Deque

7) Un arbore ternar cu 19 noduri poate avea înălțimea ? (Considerăm ca un arbore cu 1 nod are înălțimea 0)

A. 1
B. 2
C. 3
D. 4
E. 5
F. 6
G. 7
H. 8

8) Dacă vrem sa sortam 10^7 numere naturale mai mici decât 10^6 ce algoritm de sortare ar fi bine sa folosim ?

A. Radix Sort (baza 2^6)
B. Quick Sort
C. Merge Sort
D. Counting Sort
E. Timsort
F. Radix Sort (baza 2^16)

9) Să presupunem că avem numerele între 1 și 1000 inserate într-un arbore binar de căutare și că dorim să căutăm valoarea 363. Care dintre următoarele secvențe nu ar putea fi secventa de noduri examinate?

A. 2,252,401,398,330,344,397,363.
B. 924,220,911,244,898,258,362,363.
C. 925,202,911,240,912,245,363.
D. 2,399,387,219,266,382,381,278,363
E. 935,278,347,621,299,392,358,363

10) În ce complexitate putem construi cat mai eficient un heap dintr-un vector de n elemente ?

A. O(n)
B. O(1)
C. O(nlogn)
D. O(log(n))
E. O(sqrt(n))

11) Care dintre următoarele afirmații sunt adevărate despre un heap binar?

A. **Inserarea și ștergerea se fac în aceeași complexitate.**
B. **Poate avea mai mult de un nivel incomplet**
C. **Poate avea mai multe rădăcini la un moment dat.**
D. **Poate fi folosit pentru sortarea unui vector în complexitate O(nlogn)**
E. **Suporta aceleași operații pe care le suporta un arbore binar echilibrat în aceeași complexitate.**

12) Fie H un max-heap care contine 80 valori distincte. In cate poziții diferite se poate afla elementul minim ?

A. **32**
B. **40**
C. **41**
D. **42**
E. **64**

13) Care este limita inferioară a complexității în timp (în cel mai rău caz) pentru orice algoritm de sortare care folosește doar comparații între elemente, pentru a sorta un șir de n elemente?

A. **(a) $\Omega(n^2)$**
B. **(b) $\Omega(n)$**
C. **(c) $\Omega(n \log n)$**
D. **(d) $\Omega(\log n)$**
E. **(e) $\Omega(1)$**
F. **(f) $\Omega(\sqrt{n})$**

14) Diferența de înălțime dintre doua frunze într-un max-heap poate fi ?

A. **2**
B. **1**
C. **3**
D. **0**
E. **4**
F. **5**

15) Care dintre următoarele afirmații sunt adevărate despre stive și cozi?

A. **(a) Ambele sunt structuri de date liniare.**
B. **(b) Ambele respectă principiul LIFO (Last In, First Out).**
C. **(c) Ambele respectă principiul FIFO (First In, First Out).**
D. **(d) Stiva suportă operațiile PUSH și POP, iar coada suportă operațiile ENQUEUE și DEQUEUE.**
E. **(e) Stiva poate fi folosită pentru a evalua expresii postfixate.**
F. **(f) Stiva poate fi folosită pentru a implementa o coadă de așteptare**

Partea 3

16) Inserati următoarele chei și priorități într-un treap (cu heap de minim): (A, 34), (F, 42), (B, 59), C(87), (J, 77), (L, 10). Aratati arborele după fiecare inserare.

17) Construiți sparse table-ul (matricea din algoritmul RMQ) pentru șirul: 4 1 8 2 5 9 13 10. Presupunem ca în cadrul unui query am vrea sa determinam minimul pe un interval.

18) Inserati într-o trie cuvintele: their, there, this, that, does, did. Explicați cum puteți sorta aceste cuvinte în ordine lexicografică folosind trie-ul construit.

19) Construiți un min-heap cu 11 noduri în care pe ultimul nivel se afla printre altele valorile 11, 5 și 7.

20) Inserati în skip list următoarele valori: 13, 17, 1, 6, 8, 21, 23, 4, 9. La aruncarea banului obțineți următoarele valori: B, S, S, S, B, B, S, S, B, S, B, S, S, B, S, S, S, B, S, B, S, B. Cand obtineti B va opriți și inserați până la acel nivel. Cand obtineti S continuați la nivelul următor.

Partea 4

21) Care este numărul minim 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 succesorul lui x?

23) Care este complexitatea în timp, în cel mai rău caz, pentru a uni (merge) două heap-uri binomiale de dimensiuni m și n, într-un nou heap binomial?

24) Se poate găsi elementul maxim dintr-un min-heap în O(log(n))?. Justificati.

25) Fie T un arbore binar cu n noduri. Este posibil sa verificati în O(n) dacă T este arbore binar de căutare?Justificati.

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

26) Fie P o permutare cu n elemente. Scopul este sa determinam aceasta permutare știind pentru fiecare poziție i (1 <= i <= n) cate elemente aflate înaintea ei sunt mai mici decat P[i]. Dacă exista mai multe permutări posibile, se va afla cea minim lexicografica.

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, sa se raspunda dacă subsecventa de la i la j (S[i…j]) este corect parantezata.

28) Se da un vector A cu N elemente. 1 <= A[i] <= N. pentru fiecare i sa se gaseasca j minim a.i. A[j] < A[i].

29) Se da un vector A cu N elemente. 1 <= A[i] <= N. Să se numere tripletele (i, j, k) a.i. 1 <= i < j < k <= N si A[i] < A[j] < A[k].