

Rândul 1

1. Descrieți în 2-3 propoziții ideea pentru a rezolva problema de mai jos. Implementați soluția (în pseudocod sau în cod Java) și specificați complexitatea funcției.

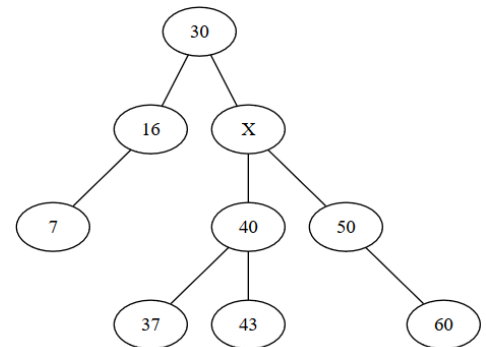
Byteman e un tâmplar, care a primit o comandă ca să construiască s mese din lemn de pin. El are suficient lemn-de-pin în atelierul lui, dar nu mai are șuruburi. Trebuie să se ducă în depozit ca să cumpere niște cutii cu șuruburi. În depozit au mai multe cutii cu număr diferit de șuruburi. Cât este numărul minim de cutii pe care trebuie să le cumpere Byteman pentru a construi mesele?

Scrieți o funcție care primește 4 parametri: s (numărul de mese de construit), k (numărul de șuruburi necesare pentru o masă), nrc (numărul de cutii cu șuruburi din depozit) și ns (un vector/o listă cu nrc elemente, care conține numărul de șuruburi pentru fiecare cutie) și returnează numărul minim de cutii necesare.

De exemplu, dacă $s = 4$, $k = 10$, $nrc = 6$ și $ns = [15, 9, 8, 20, 15, 9]$ numărul minim de cutii necesare este 3 (20, 15 și 15, sau 20, 15, și 9 sau 20, 15 și 8, sunt mai multe combinații, dar cu mai puțin de 3 cutii nu se poate).

2. Calculați complexitatea pentru subalgoritmul următor:

```
subalgoritm magie(n: întreg) este:
    i, j, k: întreg
    pentru i = 0, n, 1 execută
        pentru j = 0, i, 1, execută
            scrie "*"
        sf_pentru
    sf_pentru
    sf_pentru
    k = n*n
    cât timp k > n execută
        k = k - 2
    sf_cât timp
sf_subalgoritm
```



3. Răspundeți la următoarele întrebări cu un desen și justificări scurte.

1. Considerați arborele binar de căutare din dreapta (sus). Presupunând că în arbore avem doar numere întregi, enumerați toate valorile posibile care pot fi în nodul marcat cu X.
2. Arătați procesul de codificare Huffman pentru textul **ABRACADABRA**. Arătați arborele binar construit (și pașii de construcție) și specificați codul pentru fiecare literă.
3. Este vectorul [56, 11, 40, 3, 5, 20, 10, 4, 2, 0, 1, 8] un ansamblu binar? Dacă da, adăugați elementul 52 în el. Dacă nu, transformați-l într-un ansamblu binar interschimbând 2 elemente, și adăugați elementul 52 în el, după interschimbare. Desenați rezultatul în forma de arbore.
4. Avem o tabelă de dispersie cu $m = 9$ poziții și rezolvare coliziuni cu liste independente. Adăugați în tabela de dispersie următoarele elemente: 7, 19, 66, 3, 23, 37, 51, 90, 70, 11.

4. Alegeți răspunsul corect la următoarele întrebări și justificați alegerea făcută. La fiecare întrebare există un singur răspuns corect (exceptând prima întrebare).

1. Avem un arbore binar pentru care parcurgerea în inordine este BADECF și parcurgerea în preordine este ABCDEF. Care dintre următoarele noduri sunt frunze în arbore?
 - a. A
 - b. B
 - c. C
 - d. D
 - e. E
 - f. F
2. Dacă avem o tabelă de dispersie cu 100 de poziții și rezolvare coliziuni prin adresare deschisă, cât este numărul maxim de elemente care pot fi stocate în tabelă?
 - a. 100
 - b. 50
 - c. 99
 - d. 49
 - e. oricâte
3. Dacă avem o Stivă implementată pe un Vector Dinamic, care dintre următoarele operații are complexitatea $\Theta(n)$ în caz defavorabil?
 - a. adaugă (push) când numărul de elemente e mai mic decât capacitatea
 - b. șterge (pop)
 - c. element (top/peek)
 - d. vidă (isEmpty)
 - e. niciuna dintre cele de mai sus
4. Rezultatul evaluării expresiei în forma postfixată: $2\ 3\ 4\ * + 6\ 2\ 1\ + / +$ este o valoare între:
 - a. -100 și -15
 - b. -15 și -5
 - c. -5 și 5
 - d. 5 și 15
 - e. 15 și 100
5. Avem o implementare pentru o Coadă pe o listă simplu înlănțuită în care reținem 2 noduri: unul pentru *front* și unul pentru *end*. Care dintre aceste 2 noduri se schimbă dacă adăugăm un element într-o coada VIDĂ?
 - a. doar *front*
 - b. doar *end*
 - c. ambele
 - d. niciuna
6. Care dintre următoarele operații are complexitate mai bună la o listă simplu înlănțuită decât la un vector dinamic?
 - a. Returnare element de pe o poziție i
 - b. Adăugare la sfârșit
 - c. Adăugare la început
 - d. Căutare
 - e. Dimensiune

5. Un MultiDicționar reprezentat pe o Listă Dublu Înlănțuită, în care fiecare nod reține o cheie și o singură valoare (dacă o cheie are mai multe valori, avem mai multe noduri cu cheia respectivă). Dați reprezentarea MultiDicționarului (ce structuri și ce câmpuri sunt folosite). Specificați și implementați operația de ștergere. Cât este complexitatea operației? Dați reprezentarea iteratorului și implementați la alegere 2 operații de la iterator.
6. Avem containerul Colecție, reprezentată pe o tabelă de dispersie, rezolvare coliziuni prin liste independente, în care în fiecare nod avem un element unic și frecvența lui. Dați reprezentarea Colecției (ce structuri și ce câmpuri sunt folosite). Specificați și implementați operația de adăugare. Cât este complexitatea operației?

Punctaj: 1 – 1p; 2 – 1p; 3 – 3p (4*0.75); 4 – 2p (6*0.33); 5 – 1.5p; 6 – 1.5p; Of - 1p.

Rândul 2

1. Descrieți în 2-3 propoziții ideea pentru a rezolva problema de mai jos. Implementați soluția (în pseudocod sau în cod Java) și specificați complexitatea funcției.

Bytewoman este educatoare în grădiniță și pregătește un spectacol cu grupa ei. Copii sunt foarte entuziasmați, dar Bytewoman trebuie să lucreze foarte mult, pentru că are nevoie de K costume de soldat. Ea vrea să cumpere K costume de aceeași mărime, ca după aceea părinții copiilor să mai facă mici modificări la lungime dacă e necesar. Deci, Bytewoman a înregistrat înălțimea fiecărui copil, iar voi trebuie să o ajutați ca să aleagă dintre cei N copii, cei K care vor juca rolul soldaților astfel încât diferența de înălțime dintre cel mai scund și cel mai înalt dintre cei K copii este minimul posibil.

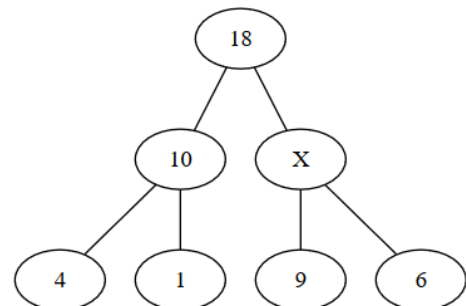
Scrieți o funcție care primește 3 parametri: k , numărul de soldați, n , numărul de copii și un vector/listă cu n elemente, reprezentând înălțimea copiilor. Calculați și returnați diferența minimă posibilă dacă selectăm K copii pentru rolul de soldați.

De exemplu, dacă avem $k = 3$, $n = 7$, și vectorul $[153, 165, 166, 141, 138, 159, 155]$ diferența minimă este 6 (dacă îi alege pe copiii cu înălțimea 153, 159, 155).

2. Calculați complexitatea pentru subalgoritmul următor:

```
subalgoritm magie(n):  
    k, index, i : întreg  
    k = n*n  
    index = 0  
    cât timp k > 0 execută:  
        k = k / 2  
        index = index + 1  
    sf_cât timp  
    dacă n % 2 == 0:  
        pentru i = 0, index, 1 execută:  
            scrie "*"   
        sf_pentru  
    altfel  
        pentru i = 0, n, 1, execută:  
            scrie "*"   
        sf_pentru  
    sf_dacă  
sf_subalgoritm
```

3. Răspundeți la următoarele întrebări cu un desen și justificări scurte.



1. Considerați ansamblul binar din dreapta. Presupunând că în ansamblu avem doar numere întregi, enumerați toate valorile posibile care pot fi în nodul marcat cu X.
2. Pornind de la un arbore binar de căutare inițial vid, adăugați în el pe rând, în această ordine, următoarele elemente: 40, 8, 1, 50, 43, 73, 5. Desenați arborele final, după cele 7 adăugări.
3. Considerați o tabela de dispersie cu $m = 13$ poziții și adresare deschisă cu funcția de dispersie $d(c, i) = ((c \% m) + 1 * i + 2 * i^2) \% m$. Arătați cum se pot adăuga în tabela de dispersie următoarele elemente, în ordinea specificată: 30, 19, 42, 50, 16, 136, 32.
4. Arătați procesul de codificare Huffman pentru textul **MISSISSIPPI**. Arătați arborele binar construit (și pașii de construcție) și specificați codul pentru fiecare literă.

- 4. Alegeți răspunsul corect la următoarele întrebări și justificați alegerea făcută. La fiecare întrebare există un singur răspuns corect.**
- Avem un arbore binar pentru care parcurgerea în inordine este HKGADFMB și parcurgerea în preordine este AGHKFDBM. Cum arată parcurgerea pe niveluri (breadth-first traversal)?
 - AGFHDBKM
 - KHGADFMB
 - AFGDBHMK
 - GFAHKBDM
 - AGFBDHMK
 - Care dintre următoarele operații NU există pentru un Dicționar implementat pe un Vector Dinamic?
 - adaugă o pereche cheie, valoare
 - șterge o pereche de pe o poziție
 - caută valoarea asociată cu o cheie
 - crează un iterator
 - returnează numărul de perechi
 - Rezultatul evaluării expresiei în forma postfixată: $4\ 5\ 1\ * - 6\ 3 + 2\ *$ este o valoare între:
 - 100 și -15
 - 15 și -5
 - 5 și 5
 - 5 și 15
 - 15 și 100
 - O diferență importantă între Stiva și Coadă este:
 - Pentru a implementa o Stivă avem nevoie de o listă înlănțuită, pentru Coadă nu
 - Pentru a implementa o Coadă avem nevoie de o listă înlănțuită, pentru Stivă nu
 - Coadă folosește 2 capete ale structurii de date, Stiva folosește doar una
 - Stiva folosește 2 capete ale structurii de date, Coadă folosește doar una
 - Care dintre următoarele vectori reprezintă un ansamblu binar?
 - [4, 33, 6, 90, 34, 32, 31, 91, 92, 89, 50, 35]
 - [3, 10, 8, 4, 7, 5, 6, 30, 25, 15, 31, 65]
 - [3, 10, 8, 30, 25, 15, 32, 100, 39, 26, 24, 40]
 - [1, 3, 20, 21, 65, 54, 67, 41, 30, 83, 52]
 - Când ștergem un element din mijlocul unei liste dublu înlănțuite (deci nu primul și nu ultimul element) avem de modificat/setat 4 legături în total.
 - Adevărat
 - Fals
- 5. Avem o Mulțime, reprezentată pe un VectorDinamic. Dați reprezentarea Mulțimii (ce structuri și ce câmpuri sunt folosite). Specificați și implementați operația de adăugare. Cât este complexitatea operației? Dați reprezentarea iteratorului și implementați la alegere 2 operații de la iterator.**
- 6. Avem containerul DicționarOrdonat reprezentat pe un Arbore Binar de Căutare. Dați reprezentarea Dicționarului (ce structuri și ce câmpuri sunt folosite). Specificați și implementați operația de căutare. Cât este complexitatea operației?**

Punctaj: 1 – 1p; 2 – 1p; 3 – 3p (4*0.75); 4 – 2p (6*0.33); 5 – 1.5p; 6 – 1.5p; Of - 1p.