

NUME:

PRENUME:

GRUPA:

	Ex1	Ex2	Ex3	Ex4
a				
b				
c				
d				

Punctajele se acorda astfel: $a=1p$, $b=2p$, $c=3p$, $d=4p$, total $4 \times (1+2+3+4) = 40p$

Punctajul pentru d se acorda doar daca ati rezolvat corect punctul c corespunzator

1. Lista Inlantuite vs Vectori

- Cum se aloca static un vector (array)? Dar dinamic? Dati cate un exemplu.
- Enumerati 3 avantaje si 3 dezavantaje ale unei liste inlantuite fata de un vector.
- Fie clasa ResizableArray, ce modeleaza un array cu redimensionare automata, si clasa DoublyLinkedList, ce modeleaza o lista dublu inlantuita. Implementati functia addFirst pentru fiecare din cele doua structuri de date.
- Fie clasa SortedResizableArray si clasa SortedDoublyLinkedList. Implementati operatia `add(T newElem)` si estimati numarul de atribuirii pe care le face aceasta functie in a, respectiv b, pentru:

```
SortedResizableArray<int> a;
SortedDoublyLinkedList<int> b;
for (int i=1; i<10; i++) {
    a.add( (i * 3) % 10 );
    b.add( (i * 3) % 10 );
}
```

2. Arbori Binari de Cautare

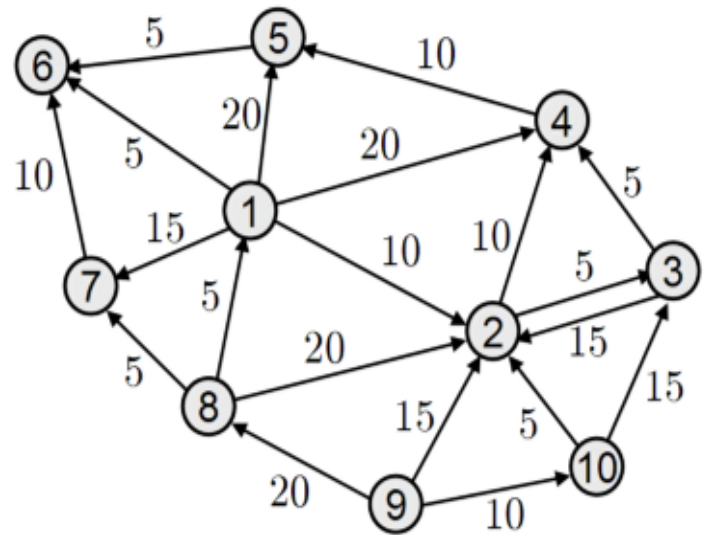
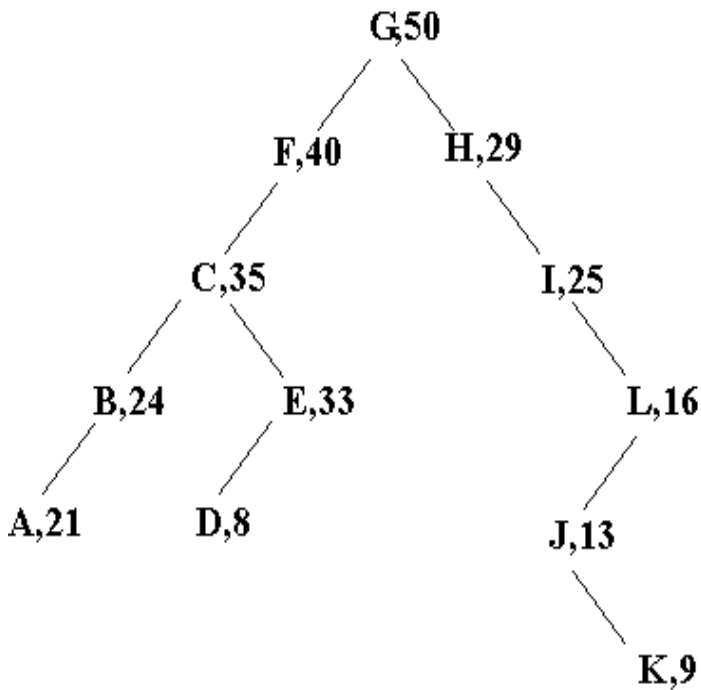
- Fie urmatoarea succesiune de chei: C X D A E G F K Y. Cum arata un arbore binar de cautare in care se introduc aceste chei?
- Functia `maxDepth` trebuie sa calculeze adancimea maxima a unui arbore binar (adica numarul de noduri de pe calea cea mai lunga de la radacina la o frunza). Completati:

```
int maxDepth(struct node* node){
    if (node==NULL)
        return 0;
    else {
        int lDepth = maxDepth(node->left);
        int rDepth = maxDepth(node->right);
        .....
    }
}
```

- Se da un `BinarySearchTree` si o `SortedDoublyLinkedList`. Se cunoaste o succesiune de n numere, care se presupune ca au fost deja adaugate in fiecare din cele doua structuri de date. Explicati cate elemente trebuie sa citeasca sau modifice urmatoarele operatii, in functie de n : `add(T newElem)`, `extractMax()`, `extractElem(T elem)`
- Scrieti cod si explicati prin desene cum functioneaza echilibrarea prin refacere globala (global rebuilding). Folositi punctele a si c sa aratati de ce arborele devine mai eficient.

3. Treap

- Arborele din figura este un treap? Argumentati.
- Fie un treap gol. Se adauga urmatoarele chei 13, 10, 7, 23, 9, 18, 8, carora le corespund urmatoarele prioritati 15, 18, 14, 20, 19, 6, 25. Apoi se elimina elementul cu cheia 10. Desenati cum va arata treapul in cursul acestor operatii.
- Scriti functia de rotatie la dreapta a unui TreapNode.
- Considerand rotatiile gata implementate, scrieti functia de remove pentru TreapNode, inclusiv functia/functiile utilizate de aceasta (cu exceptia rotatiilor).



4. Graf

- Reprezentati graful orientat din figura sub forma de lista de vecini
- Care sunt componentele tare conexe ale grafului?
- Implementati o parcurgere BFS. Ce rezultat va produce pornind din nodul 1, tinand cont ca vecinii se viziteaza in ordinea in care apar in lista de vecini de la punctul a.
- Explicati, prin exemple si cod, cum functioneaza unul dintre algoritmii de sortare topologica (nu uitati sa precizati cum se numeste algoritmul ales).