Algoritmi si Structuri de Date

- seminar V -

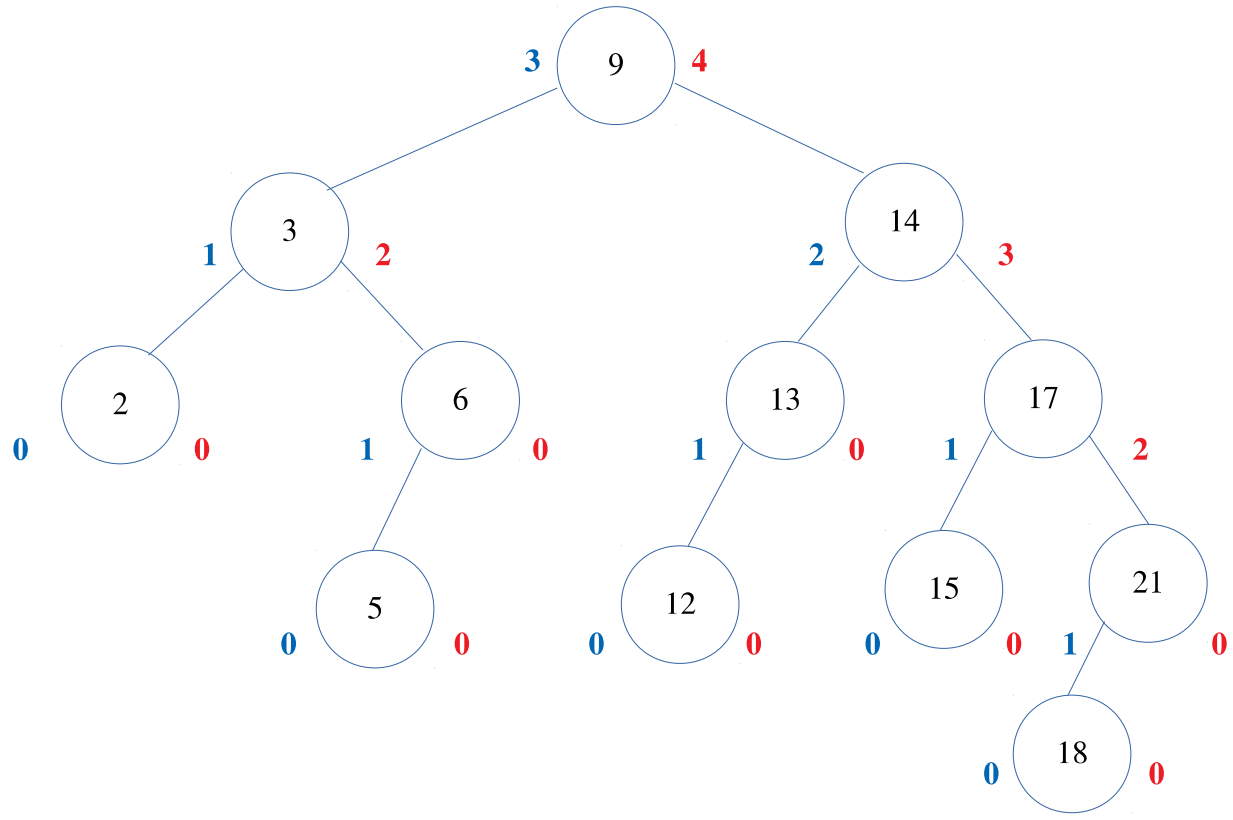
AVL.

Arborii AVL sunt arbori binari de cautare echilibrati:

- arbori binari: fiecare nod are cel mult doi fii (fiul stang si fiul drept);
- de cautare: pentru orice nod, subarborele stang contine elemente mai mici decat nodul respectiv, iar subarborele drept contine elemente mai mari;
- echilibrati: adancimea subarborelui stang este aproximativ egala cu adancimea subarborelui drept.

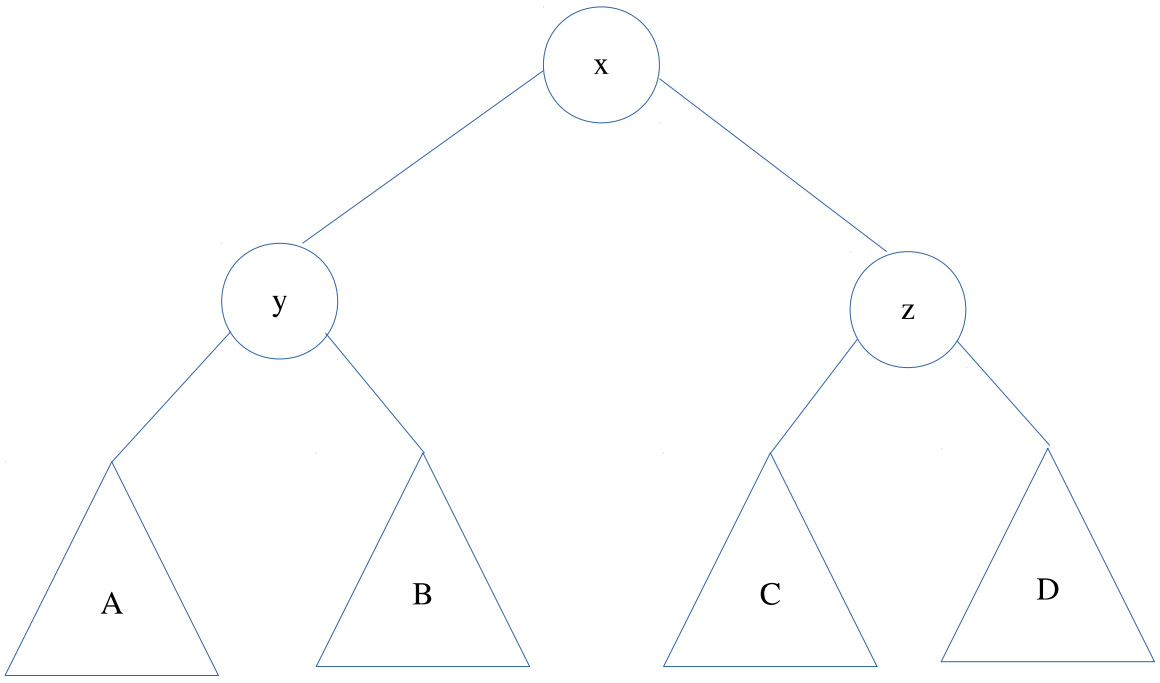
Arborii AVL se deosebesc de arborii binari de cautare datorita proprietatii ca diferenta dintre inaltimea subarborelui stang si cea a subarborelui drept in modul este cel mult 1.

Ex:



Cu albastru este trecuta adancimea subarborelui stang si cu rosu avem trecuta adancimea subarborelui drept. Pentru orice nod, daca calculam $|h_{stanga} - h_{dreapta}|$ vom observa ca este <= 1.

Pentru a pastra aceasta proprietate (diferenta in modul <= 1) cand modificam arborele (inserari, stergeri) va trebui sa efectuam rotatii asupra arborelui.



Exista doua tipuri de rotatii:

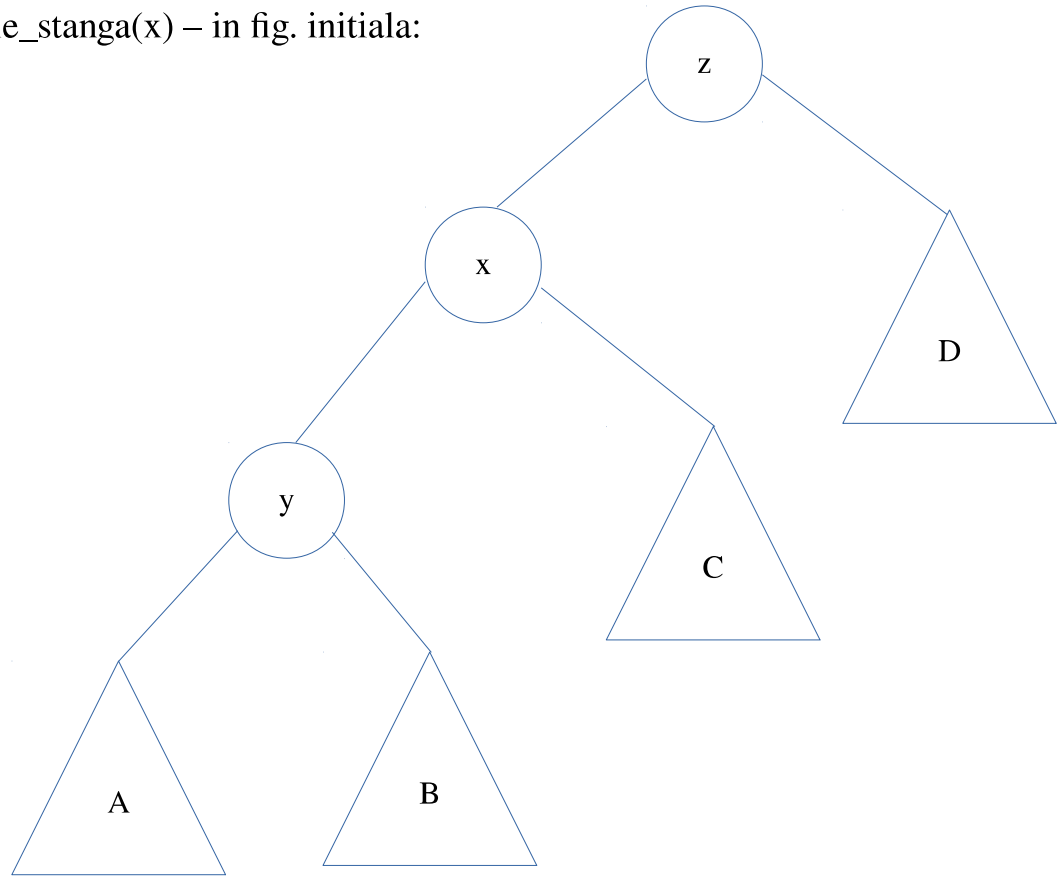
- rotatie stanga: nodul curent trece in locul fiului stang
- rotatie dreapta: nodul curent trece in locul fiului drept

Cele doua tipuri de rotatii sunt inverse una celeilalte:

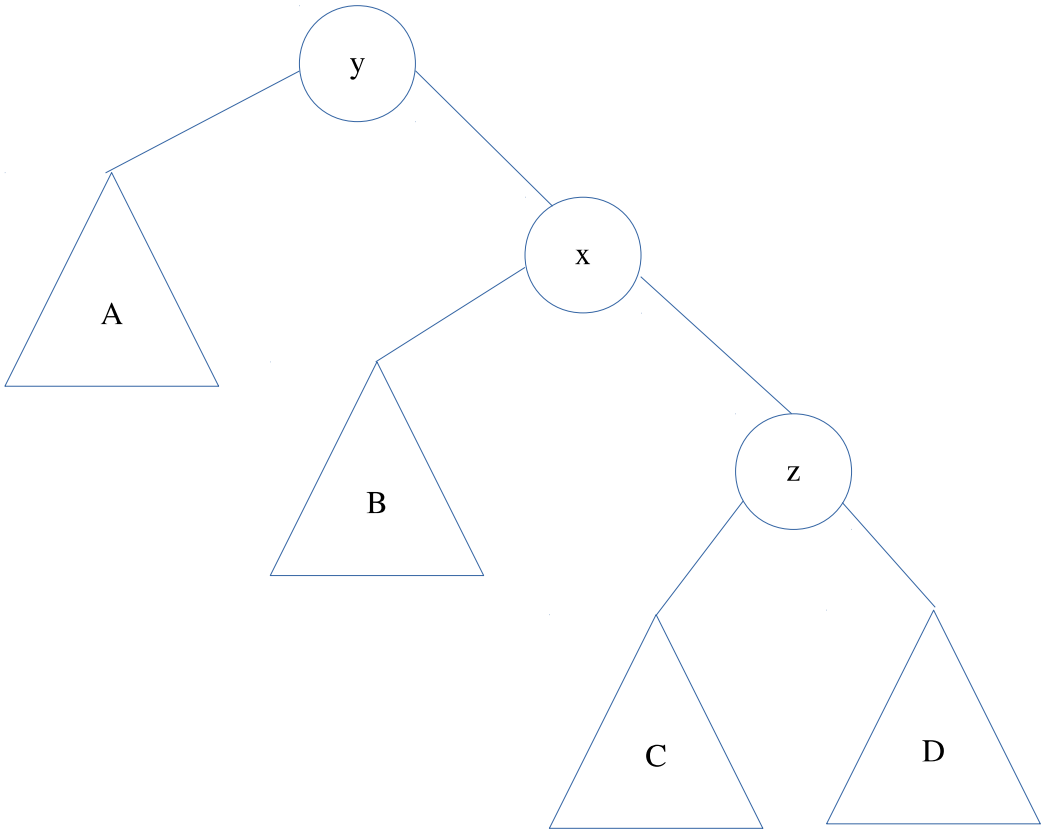
- rotatie_stanga(rotatie_dreapta(x)) = x
- rotatie_dreapta(rotatie_stanga(x)) = x

Observatie! Rotatiile sunt transformari locale. Nodurile care se afla deasupra nodului asupra caruia se efectueaza rotatia nu se modifica si nici cele aflate la doua niveluri mai jos. In imagine triunghiul reprezinta un subarbore.

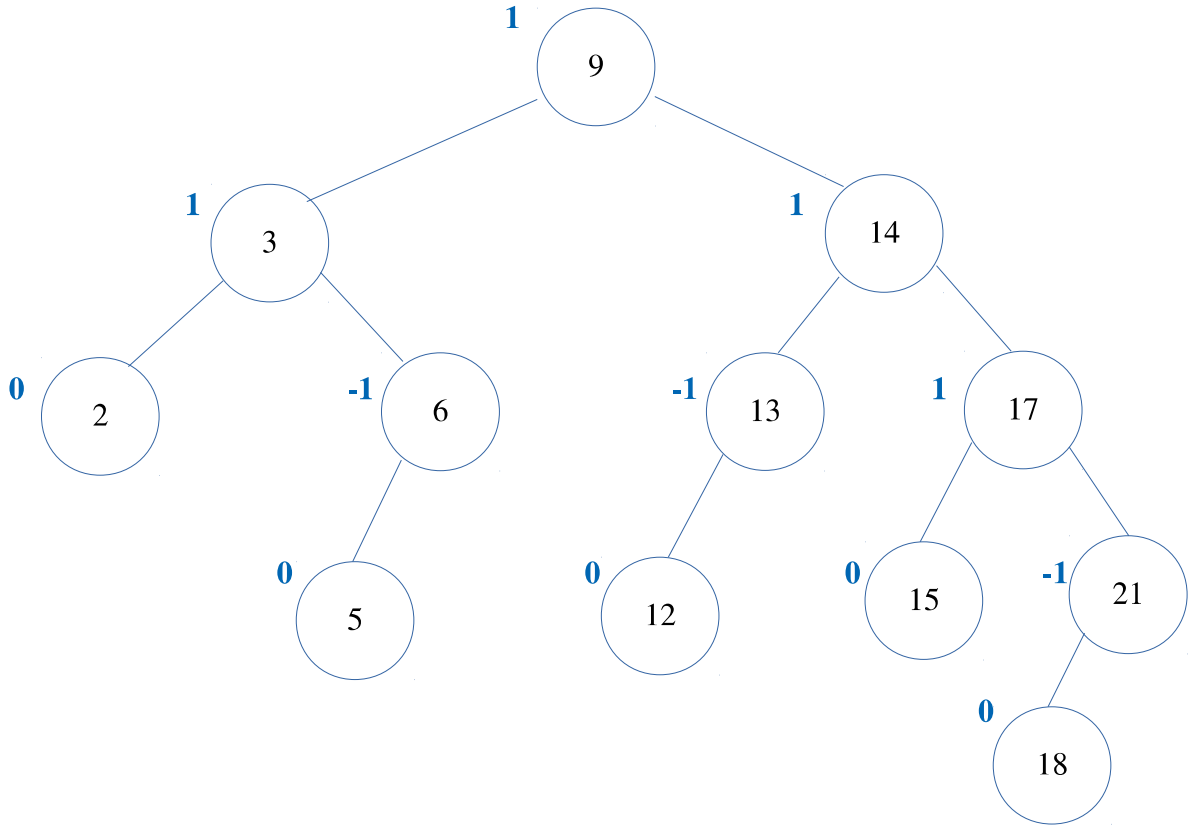
rotatie_stanga(x) – in fig. initiala:



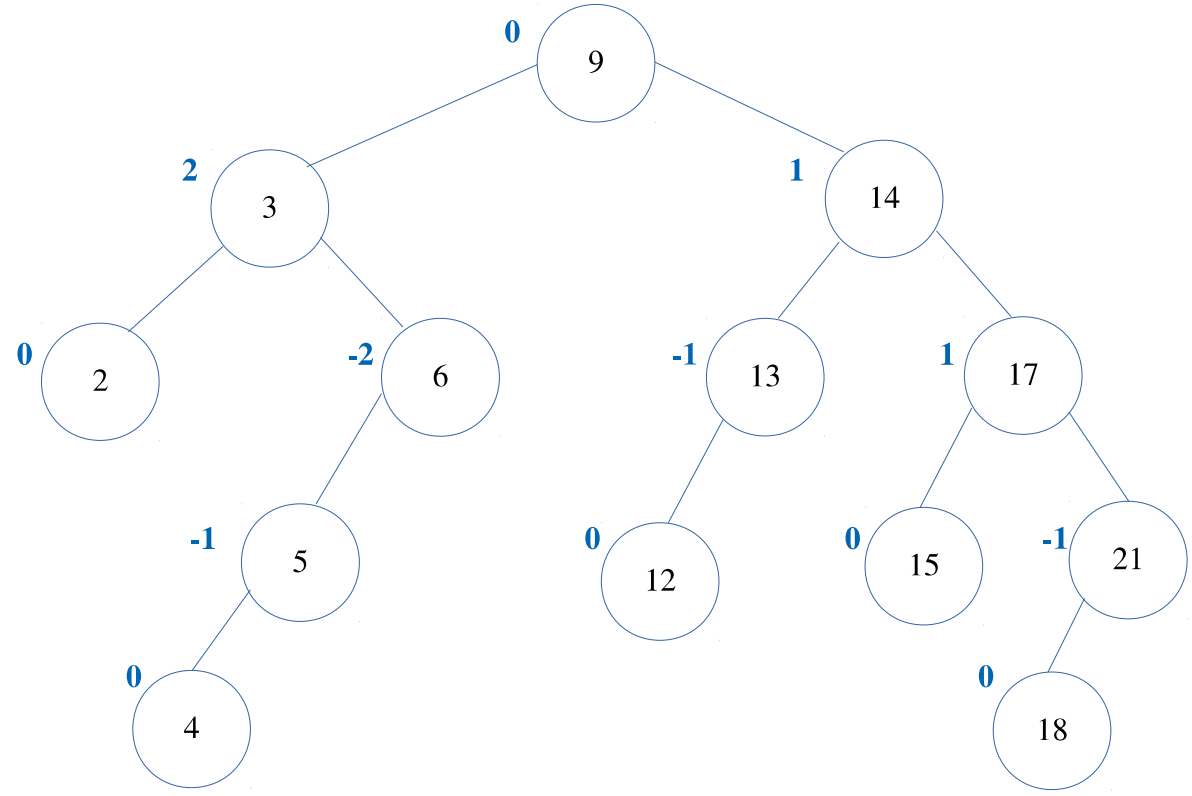
rotatie_dreapta(x) – in fig. initiala:



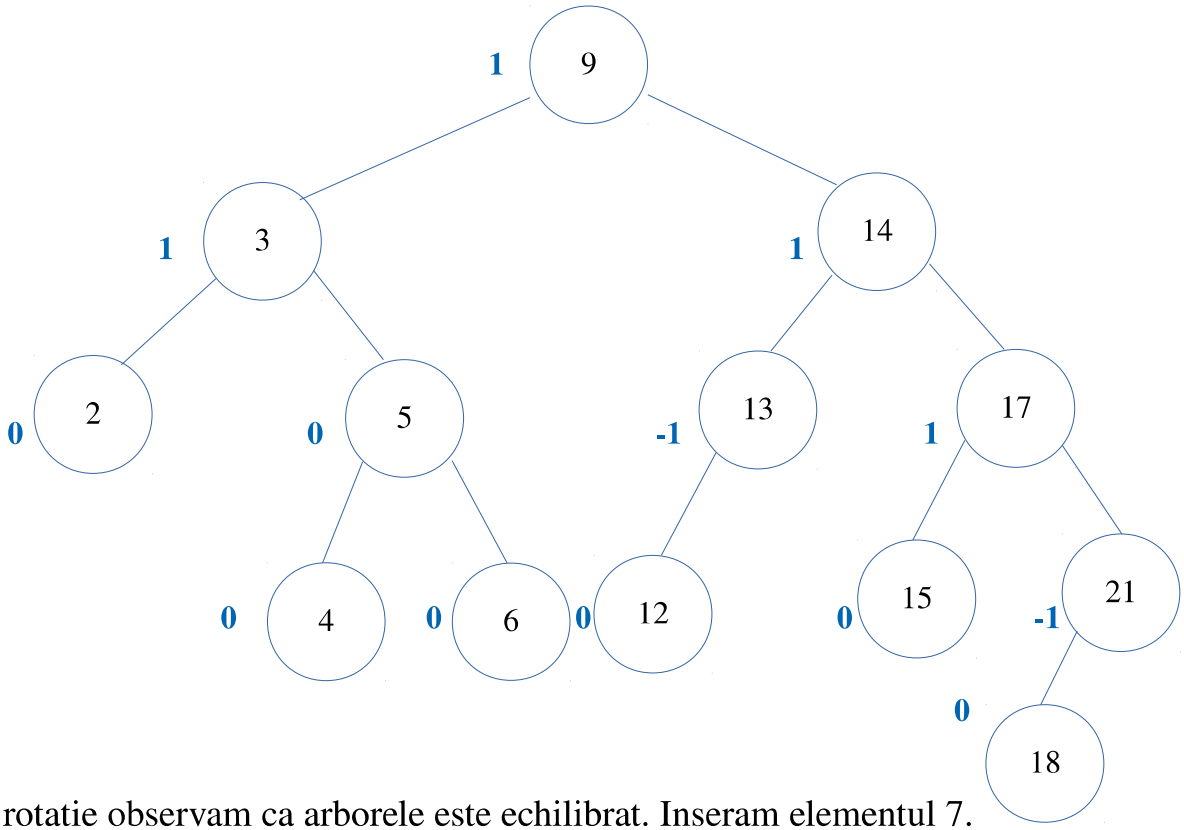
Pentru a simplifica calculele, nu vom salva doua valori in fiecare nod, ci vom salva doar diferenta dreapta – stanga. Frunzele vor avea valoarea 0.



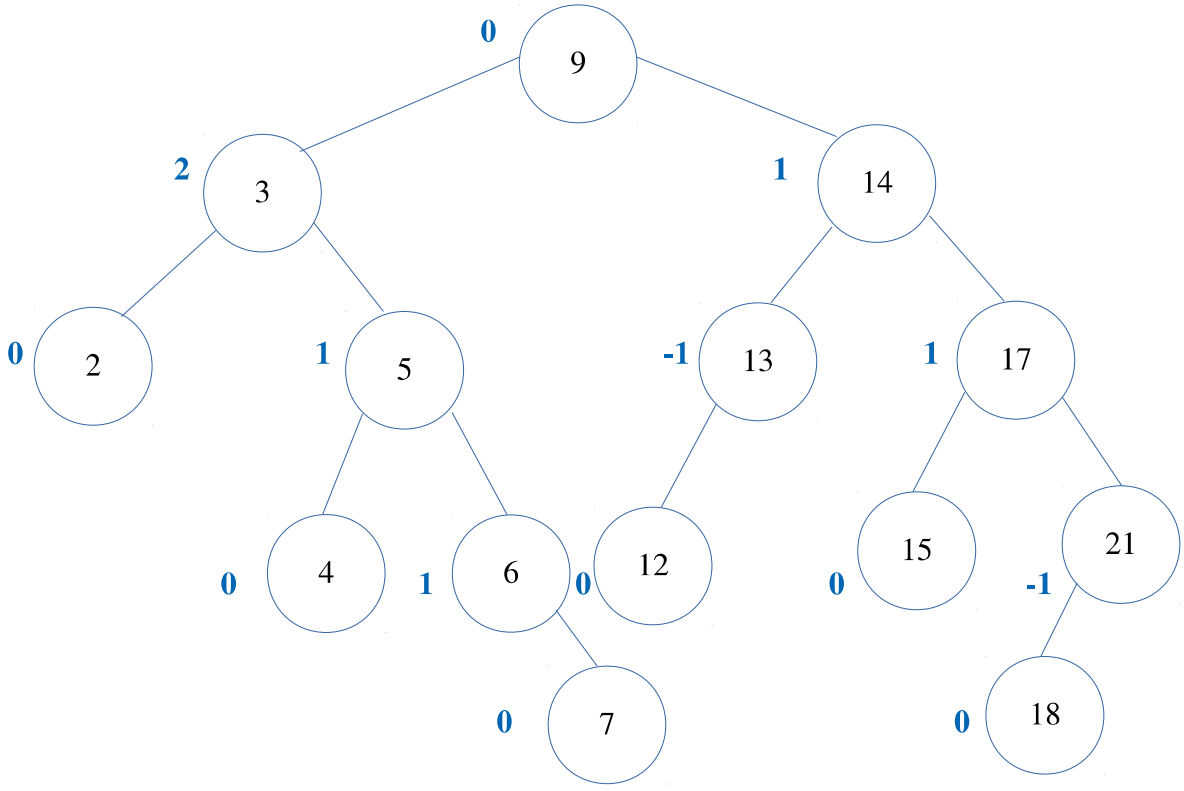
Cand inseram un element nou, vom folosi metoda uzuala de inserare pentru arborii binari de cautare. Inseram elementul 4 si recalculam diferentele.



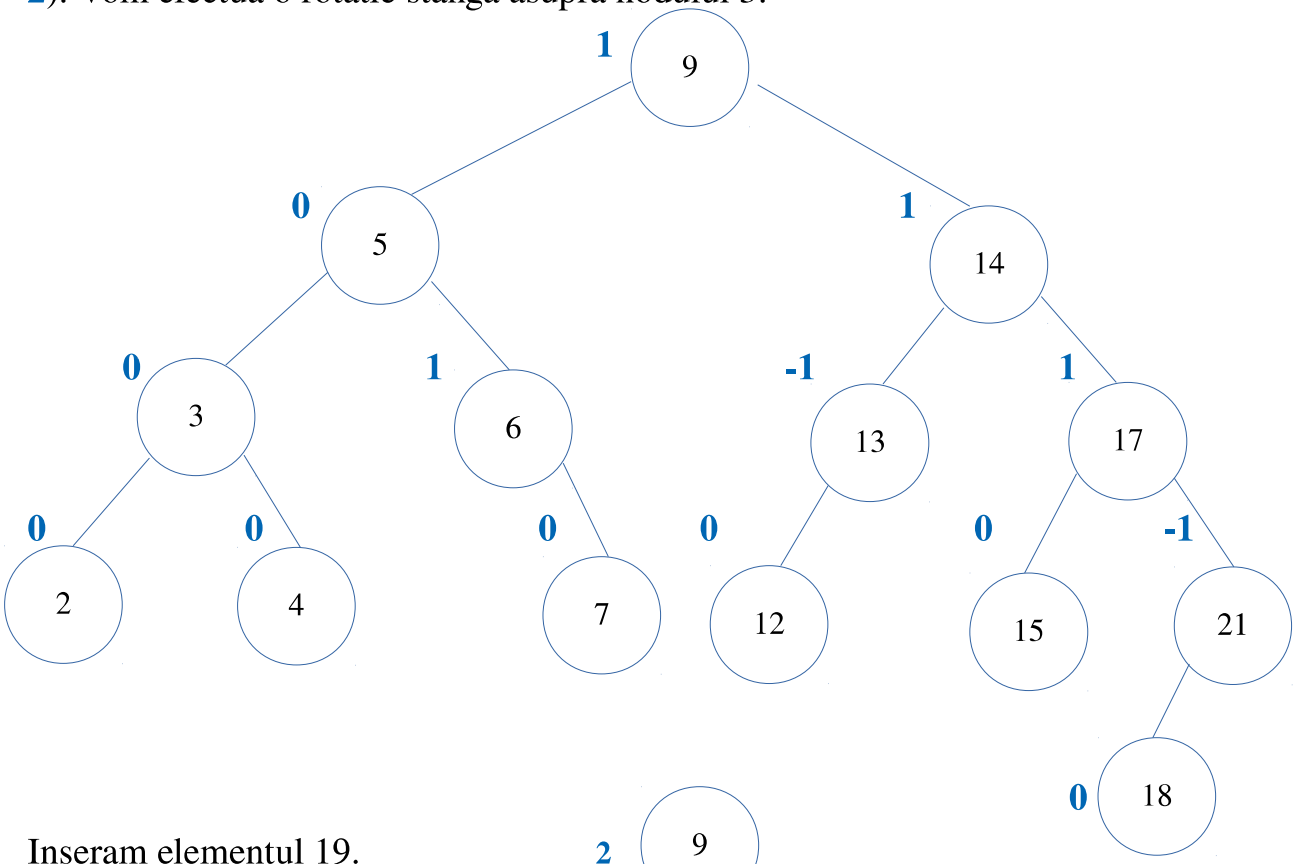
Observam ca a aparut o problema la nodul 6 deoarece diferenta intre subarborele drept si cel stang este 2 (avem valorile **0 -1 -2**). Vom efectua o rotatie dreapta asupra nodului 6 pentru a repara eroarea.



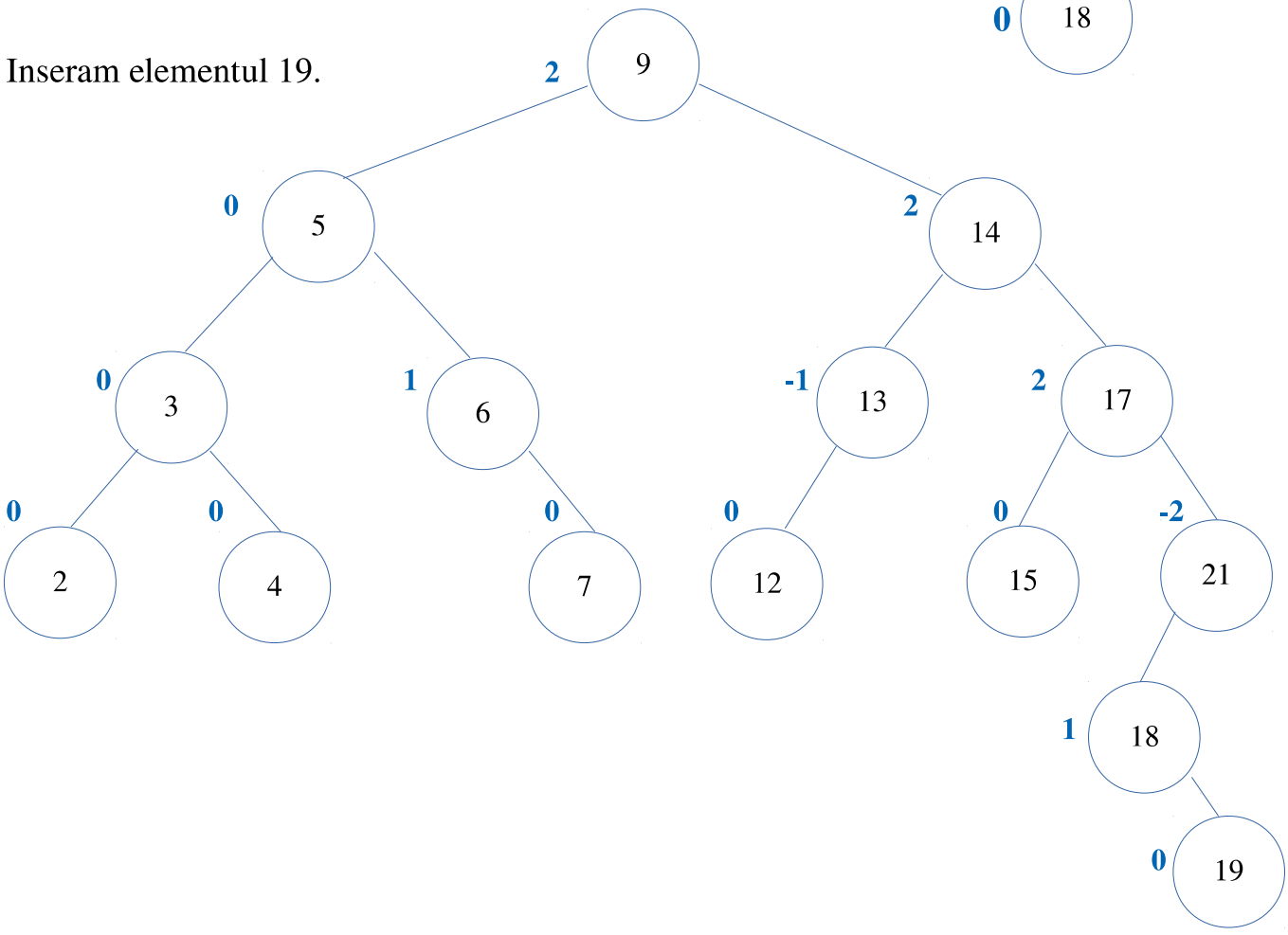
Dupa rotatie observam ca arborele este echilibrat. Inseram elementul 7.



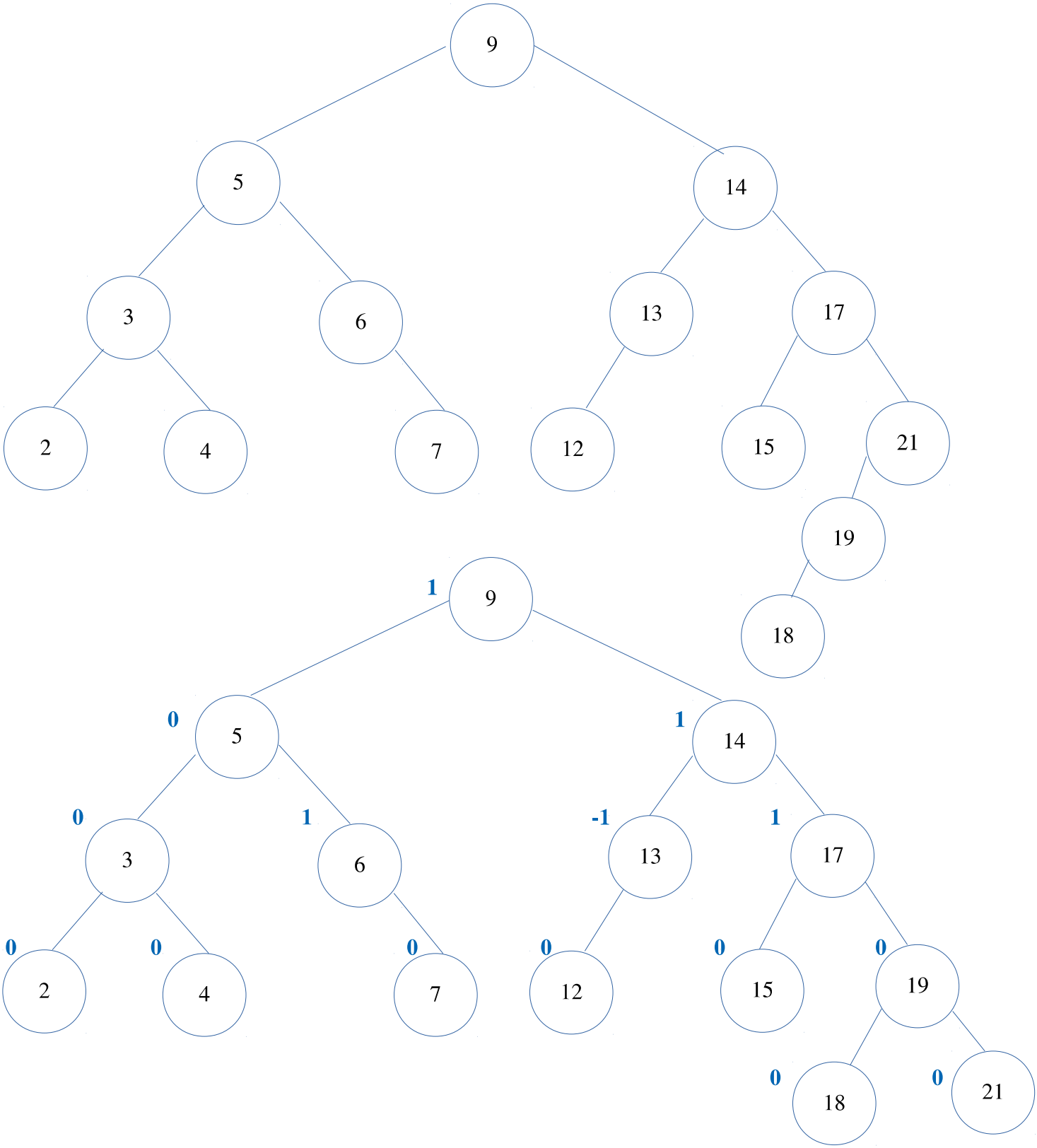
Observam ca a aparut o problema la nodul 3 deoarece diferenta este 2 (avem valorile **1 1 2**). Vom efectua o rotatie stanga asupra nodului 3.



Inseram elementul 19.

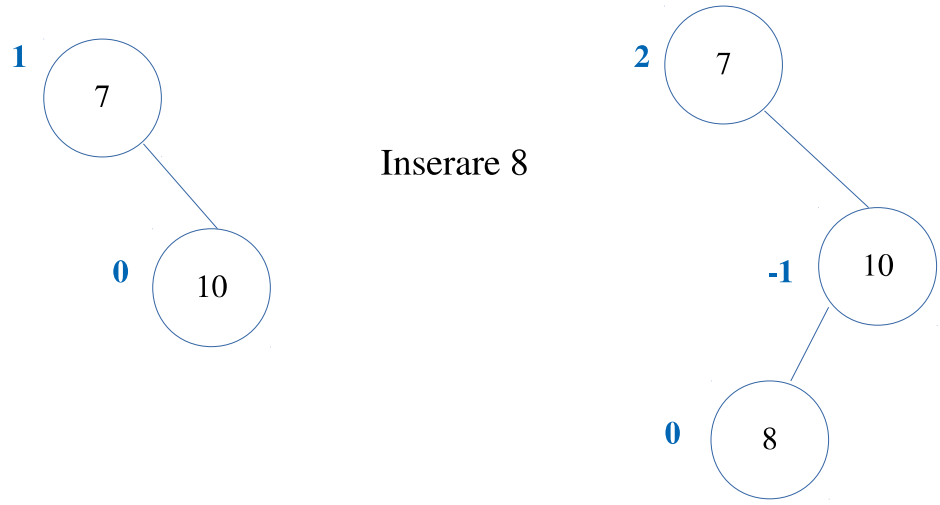


Observam ca a aparut o problema la nodul 21 deoarece diferenta este -2 (avem valorile **0 1 -2**). Vom efectua doua rotatii: o rotatie stanga asupra nodului 18 urmata de o rotatie dreapta asupra nodului 21.

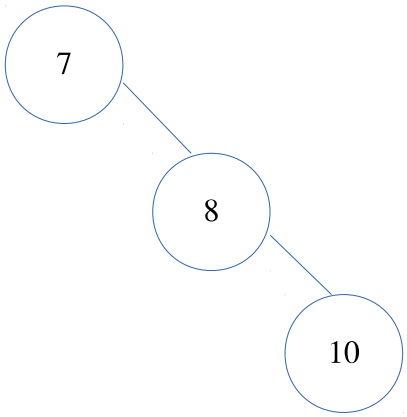


Daca la o inserare vom avea valorile **0 -1 2**, vom efectua tot doua rotatii: o rotatie dreapta asupra nodului care are valoarea **-1** urmata de o rotatie stanga asupra nodului care are valoarea **2**.

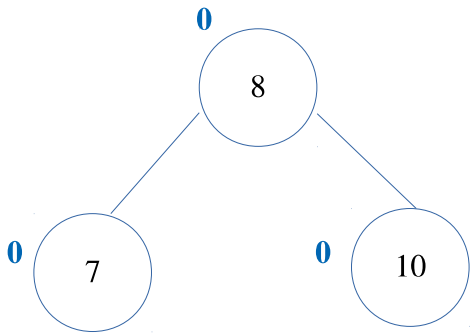
De exemplu:



Rotatie dreapta asupra lui 10:



Rotatie stanga asupra lui 7:



Exercitiu

Inserati pe rand valorile urmatoare intr-un arbore AVL: 8 5 1 6 7 10 9.

Huffman.

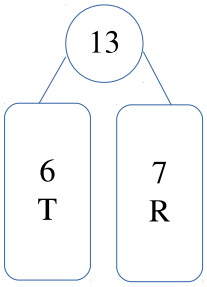
Codurile Huffman reprezinta o modalitate de a compresa un text astfel incat daca o litera apare foarte des in text, aceasta o sa fie codificata cu un numar mic de caractere, iar daca aceasta apare rar, o sa fie codificata cu un numar mai mare de caractere.

Daca s-ar folosi codurile ASCII fiecare litera ar avea aceeasi lungime si codul final ar avea o lungime mai mare decat daca am folosi un cod Huffman.

Pentru a folosi un cod Huffman trebuie sa construiti un arbore folosind ponderile date.

Ex: Se dau urmatoarele litere impreuna cu ponderile lor si se cere constructia arborelui: A – 38%, E – 22%, C – 15%, D – 12%, R – 7%, T – 6%.

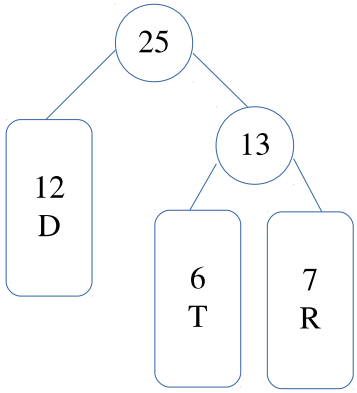
Vom alege cele mai mici doua elemente si vom crea un nod care va avea ca valoare suma acestora. Vom face o conventie: elementul mai mic se va pune in stanga. Cele mai mici ponderi sunt: R – 7%, T – 6%.



Au mai ramas elementele: A – 38%, E – 22%, C – 15%, D – 12%

Si noul nod: 13.

Se vor alege urmatoarele cele mai mici doua elemente: D – 12%, 13

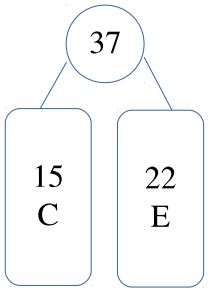


Au mai ramas elementele: A – 38%, E – 22%, C – 15%

Si noul nod: 25.

Se vor alege urmatoarele cele mai mici doua elemente:

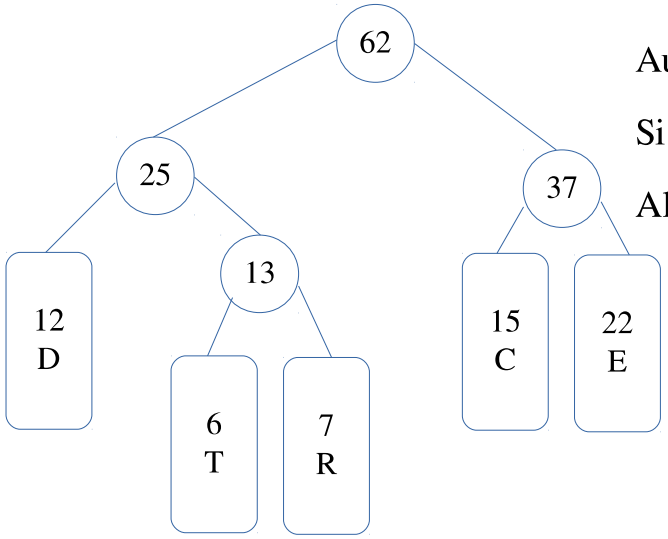
E – 22%, C – 15%



In momentul de fata avem doi arbori separati (cel din stanga si cel de deasupra). Au mai ramas elementele: A – 38%

Si nodurile: 25, 37

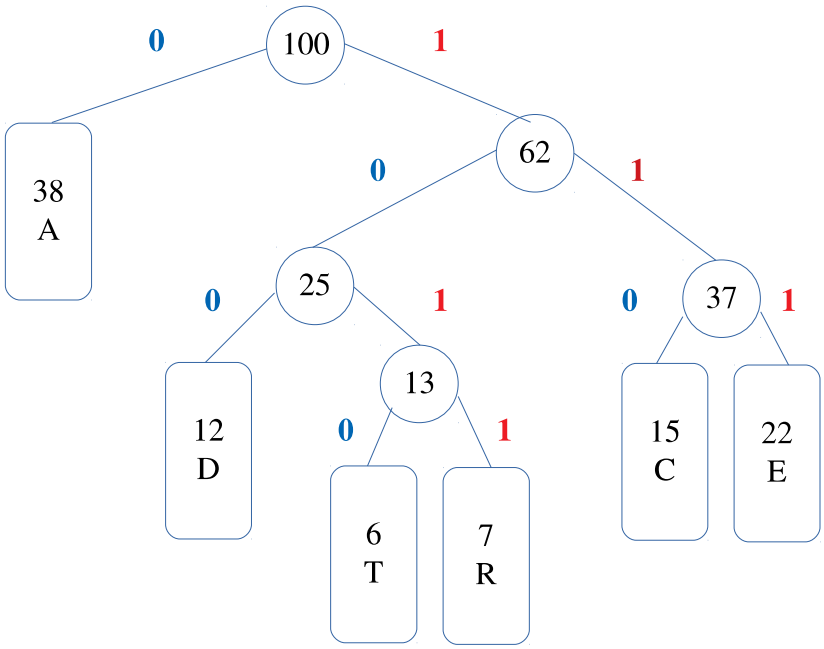
Alegem cele mai mici doua elemente: 25 si 37.



Au mai ramas elementele: A – 38%

Si noul nod: 62

Alegem ultimele elemente ramase: 38 si 62.



Dupa ce construim arborele etichetam fiecare muchie stanga cu **0** si fiecare muchie dreapta cu **1** si scriem codificarea fiecarei litere (drumul parcurs de la radacina la frunze)

A	0
D	100
C	110
E	111
T	1010
R	1011

Exercitii:

1. Construiti arborele Huffman (folosind conventia cu ponderea mica in stanga) pentru ponderile: A – 38%, E – 22%, C – 15%, D – 12%, R – 7%, T – 6%.
2. Care poate fi codificarea cuvantului CARTE:
   - 110010110101110
   - 110011111010111
   - 110010111010111
   - 110010110011100
3. Decodificati, daca e posibil:
   - 1101111011111110101111
   - 1000101111
   - 10111110
4. Gasiti alte cuvinte si scrieti si codificarile lor.