



Laboratorul 8

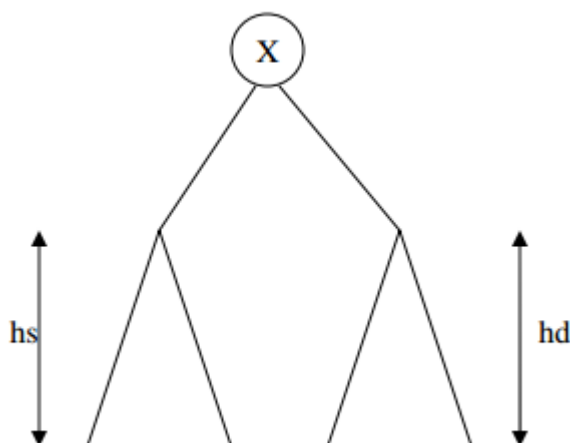
Arbori AVL

Arborii AVL sau **arborii echilibrați** sunt arbori binari ordonați, care au în plus o proprietate de echilibru.

Proprietatea de echilibru e valabilă pentru orice nod al arborelui și spune că: “*înălțimea subarborelui stâng al nodului diferă de înălțimea subarborelui drept al nodului prin cel mult o unitate*”.

Factorul de echilibrare reprezintă diferența dintre înălțimea subarborelui drept și înălțimea subarborelui stâng. Atașând fiecărui nod un câmp care reprezintă factorul de echilibrare al său, se spune că arborele binar este echilibrat atunci când toți factorii de echilibrare ai nodurilor sunt -1,0,+1.

O inserție într-un arbore binar ordonat poate duce la dezechilibrarea anumitor noduri, dezechilibrare manifestată prin nerespectarea formulei $|h_s - h_d| \leq 1$ pentru respectivele noduri.

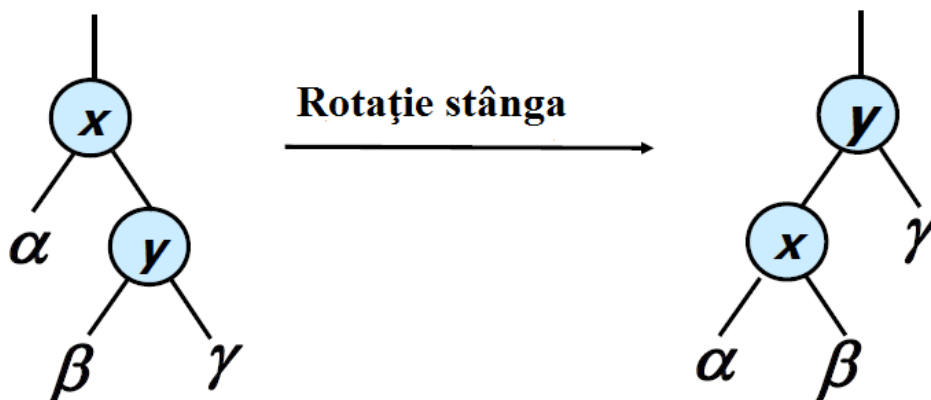


La adăugarea unui nou nod (ca frunză), factorul de echilibru al unui nod interior se poate modifica la **-2** (adăugare în subarborele stânga) sau la **+2** (adăugare în subarborele dreapta), ceea ce va face necesară modificarea structurii arborelui. **Reechilibrarea** se face prin *rotații simple sau duble*, însoțite de recalcularea înălțimii fiecărui nod întâlnit parcurgând arborele de jos în sus, spre rădăcină.

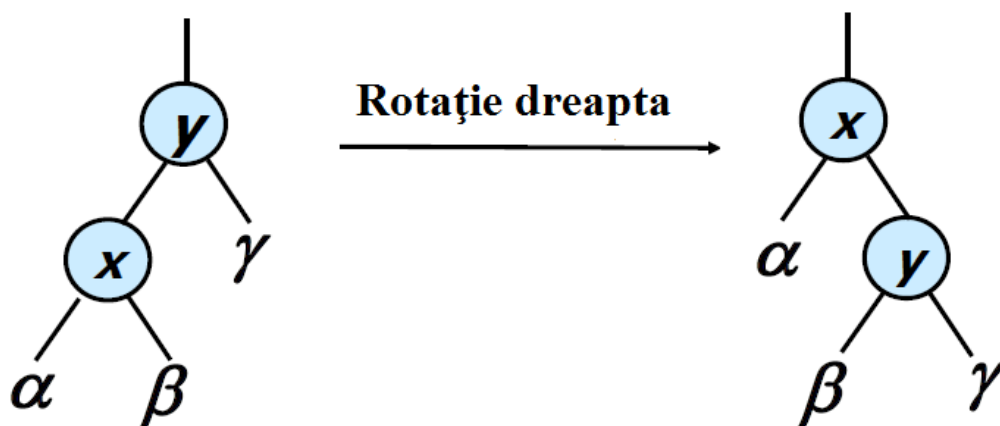


Tipuri de rotații

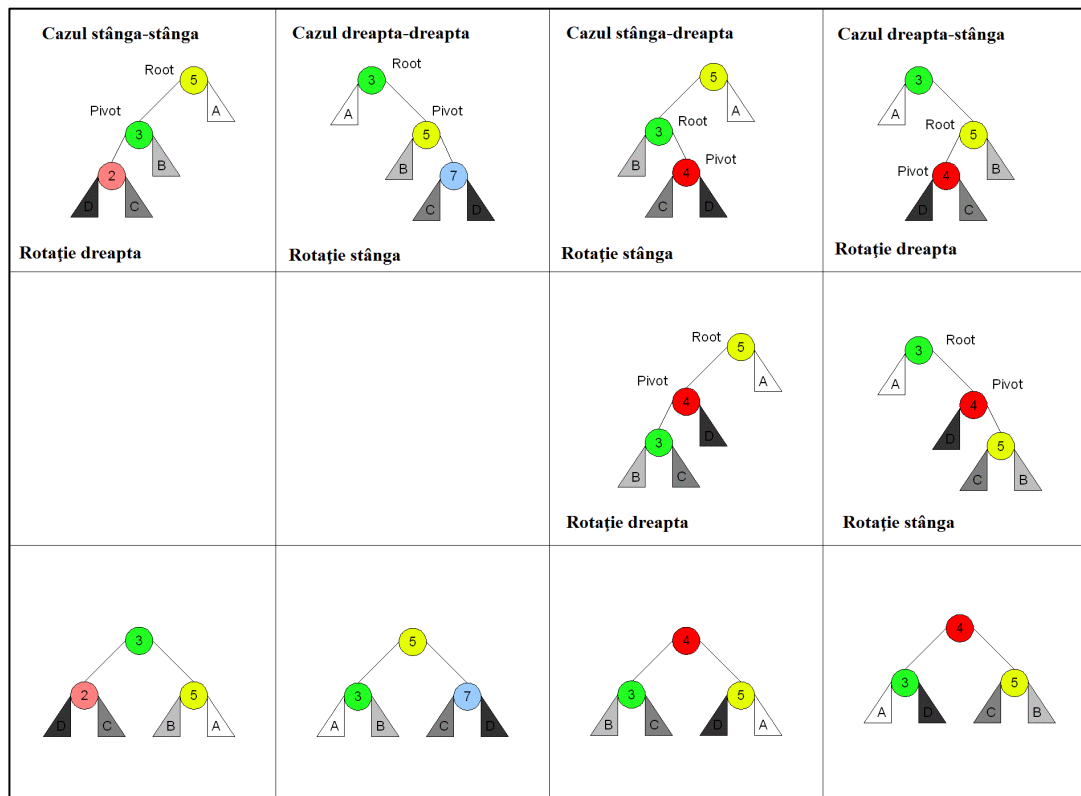
Rotație stânga



Rotație dreapta



În principiu, o cheie se **inserează** într-o primă fază, ca și într-un arbore binar ordonat obișnuit, adică se pornește de la rădăcină și se urmează fiul stâng sau fiul drept, în funcție de relația dintre cheia de inserat și cheia nodurilor prin care se trece, până se ajunge la un fiu nul, unde se realizează inserția propriu-zisă. În acest moment se parcurge drumul invers (care este unic) și se caută pe acest drum *primul nod care nu este echilibrat*, adică primul nod ai cărui subarbori diferă ca înălțime prin 2 unități. **Acest nod trebuie echilibrat** și el se va afla întotdeauna într-unul din cele 4 cazuri prezentate în continuare:



Pentru laboratorul de astăzi, vom folosi următoarele definiții:

```
typedef struct AVLNode{
    Item elem;
    int height;
    struct AVLNode *l;
    struct AVLNode *r;
    struct AVLNode *p;
}AVLNode;
```

```
typedef struct AVLTree{
    long size;
    AVLNode* root;
    AVLNode* nil;
    int (*comp)(Item a, Item b);
}AVLTree;
```



CERINȚE

1.În fișierul *AVLTree.h*, să se definească funcția **avlLeftRotate**, ce primește ca parametru adresa unui arbore și a unui nod *x* și realizează o rotație la stânga a subarborelui care are vârful în *x*. Funcția are urmatorul prototip:

```
void avlLeftRotate (AVLTree *tree, AVLNode *x)
```

(2.5 puncte)

2.În fișierul *AVLTree.h*, să se definească funcția **avlRightRotate**, ce primește ca parametru adresa unui arbore și a unui nod *y* și realizează o rotație la dreapta a subarborelui care are vârful în *y*. Funcția are urmatorul prototip:

```
void avlRightRotate (AVLTree *tree, AVLNode *y)
```

(2.5 puncte)

3.În fișierul *AVLTree.h*, să se definească funcția **avlInsert**, ce inserează un nou nod în într-un arbore AVL. Se va asigura reechilibrarea arborelui. Funcția are urmatorul prototip:

```
void avlInsert (struct AVLTree* tree, Item elem)
```

(4 puncte)



Rulați cu comenzile `runTree` și `runValgrindTree`, care se găsesc în fișierul *makefile*.

build:

```
gcc -std=c9x testAVL.c -o testAVL -w
```

clean:

```
rm testAVL *~
```

runTree:

```
./testAVL
```

runValgrindTree:

```
valgrind ./testAVL
```

Dacă inserarea se face corect, veți primi următorul mesaj:

./testAVL

. Testul Init&IsEmpty a fost trecut cu succes!

. Testul Insert a fost trecut cu succes!

. Testul Free a fost trecut cu succes!