

Structuri de Date și Algoritmi



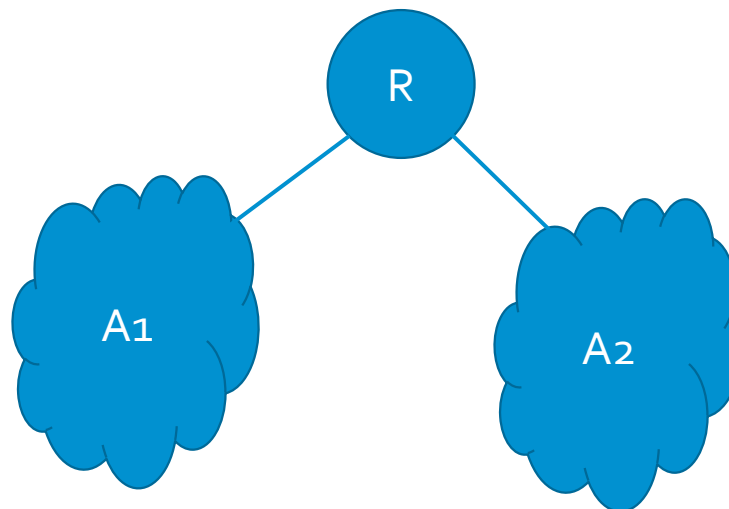
Definiții:

Definiții:

O mulțime vidă este un arbore binar

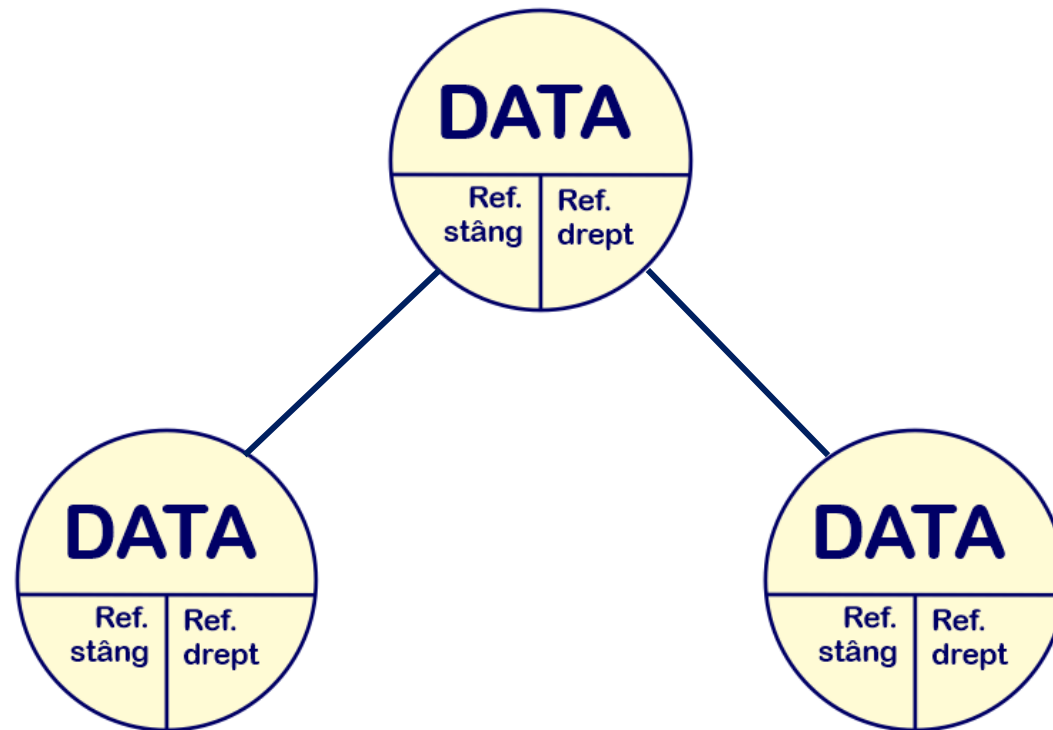
NULL

Dacă A_1 și A_2 sunt arbori binari, atunci structura $A_1 * A_2$ obținută prin adăugarea unei rădăcini R , conectate spre stânga la A_1 și spre dreapta la A_2 este un arbore binar

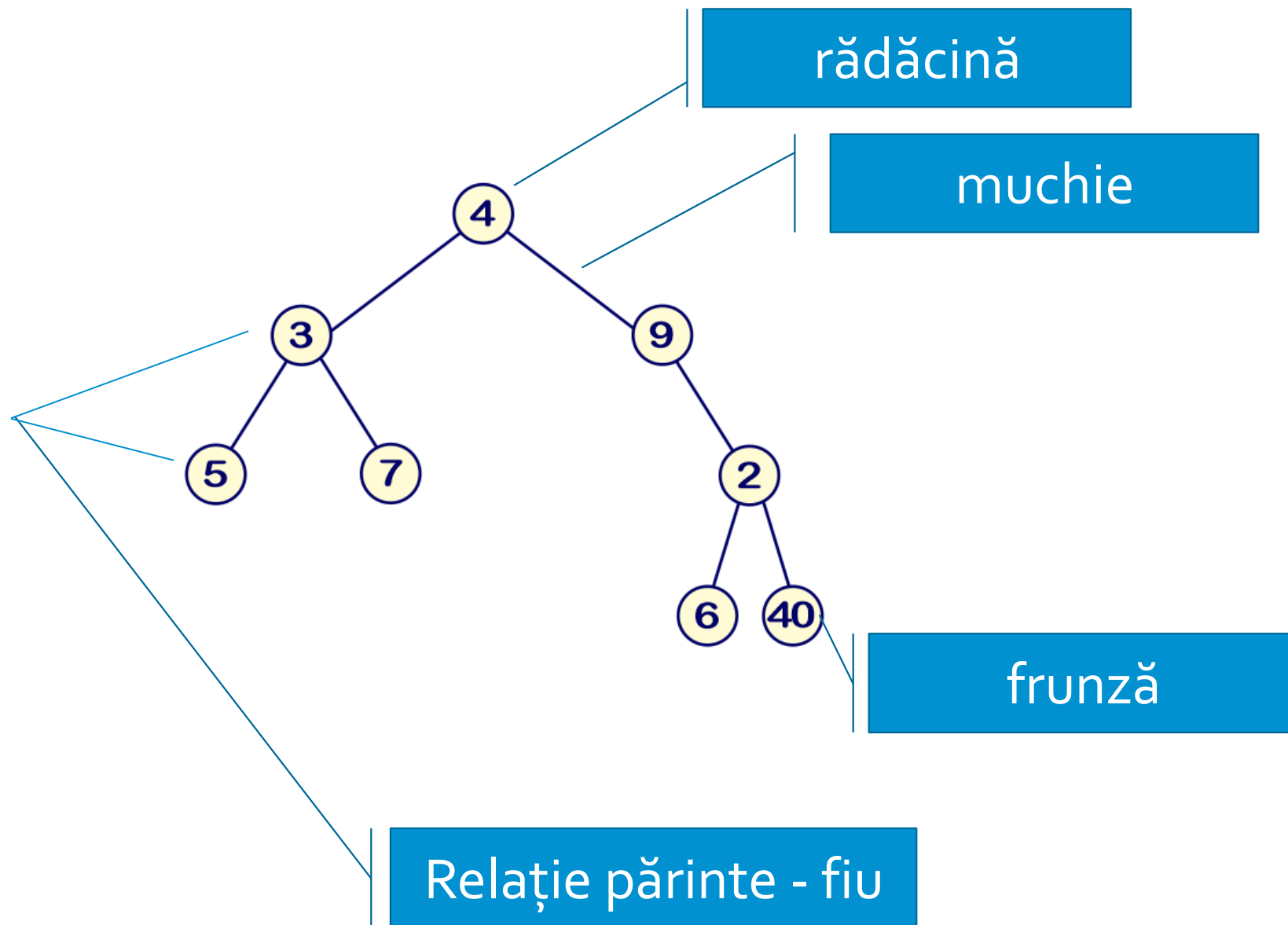


Definiții:

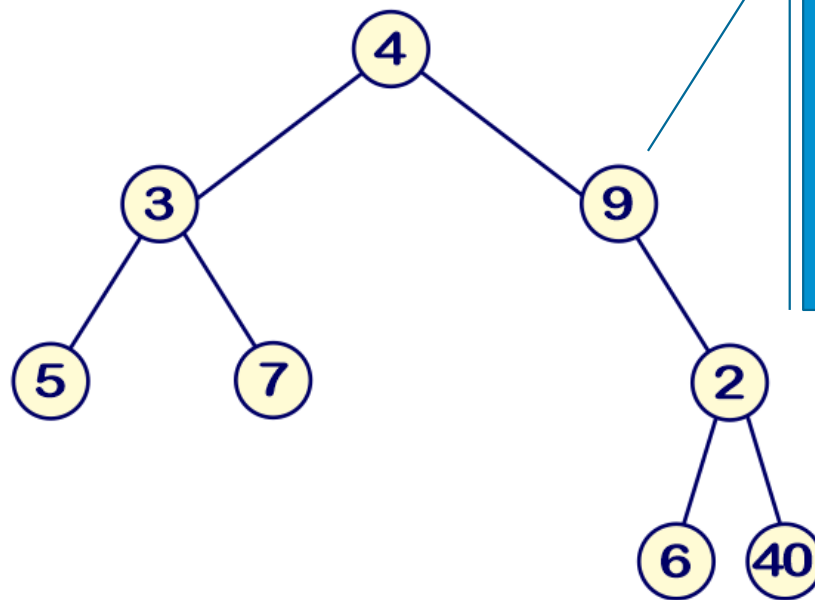
Din punct de vedere a structurilor de date un arbore binar este o structură de date, formată din noduri de date organizate pe nivele, astfel încât fiecare nod are două câmpuri de referințe către nodurile situate pe nivelul următor:



Definiții:



Definiții:



Ancestor (predecesor)

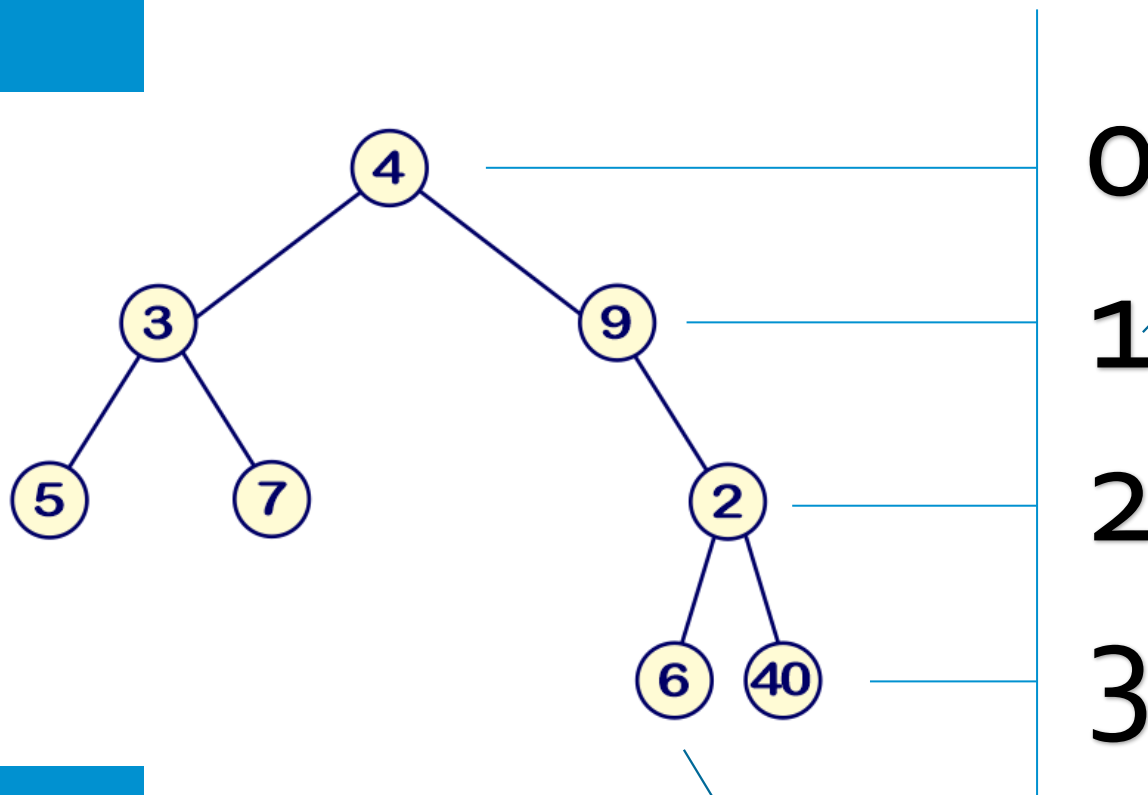
A este ancestor pentru B dacă A apare în calea de la rădăcină spre B. D în acest caz este

descendent

9 – ancestor pentru 2, 6, 40

2, 6, 40 – descendenți pentru 9

Definiții:



0

1

2

3

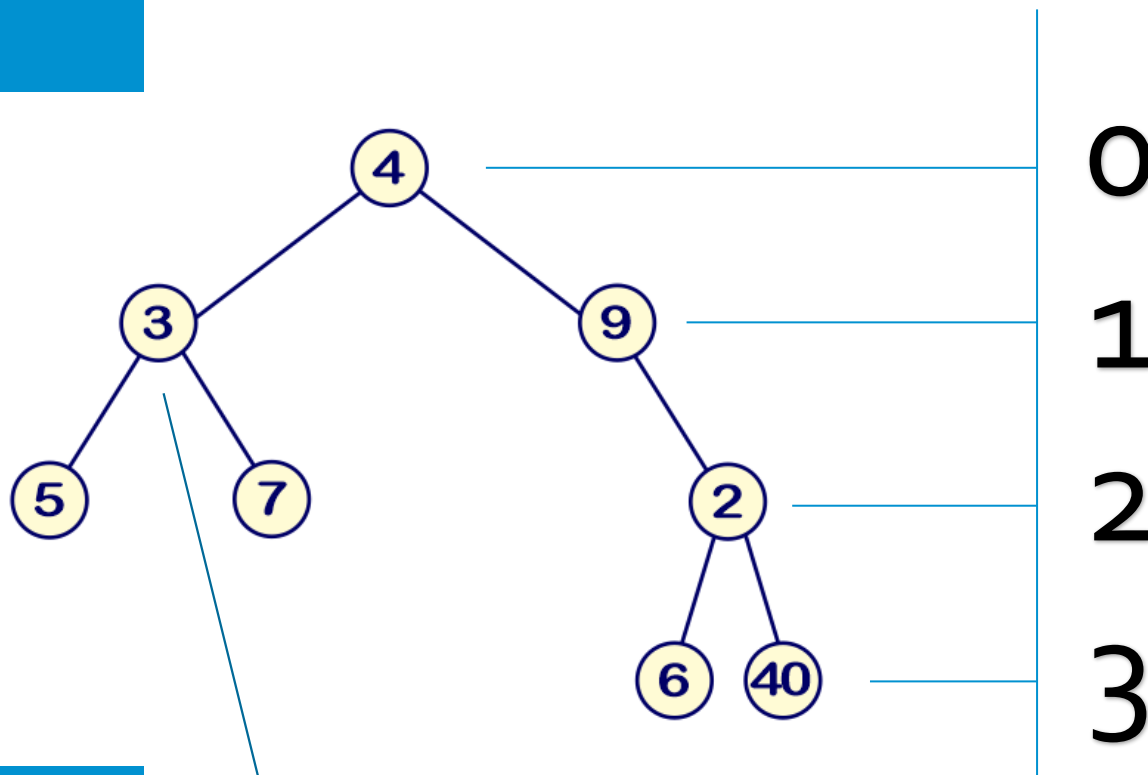
Nivel

Mulțimea de noduri
cu aceeași adâncime

Adâncimea nodului

Numărul de referințe consecutive
de la rădăcină către nod.
6 are adâncimea 3

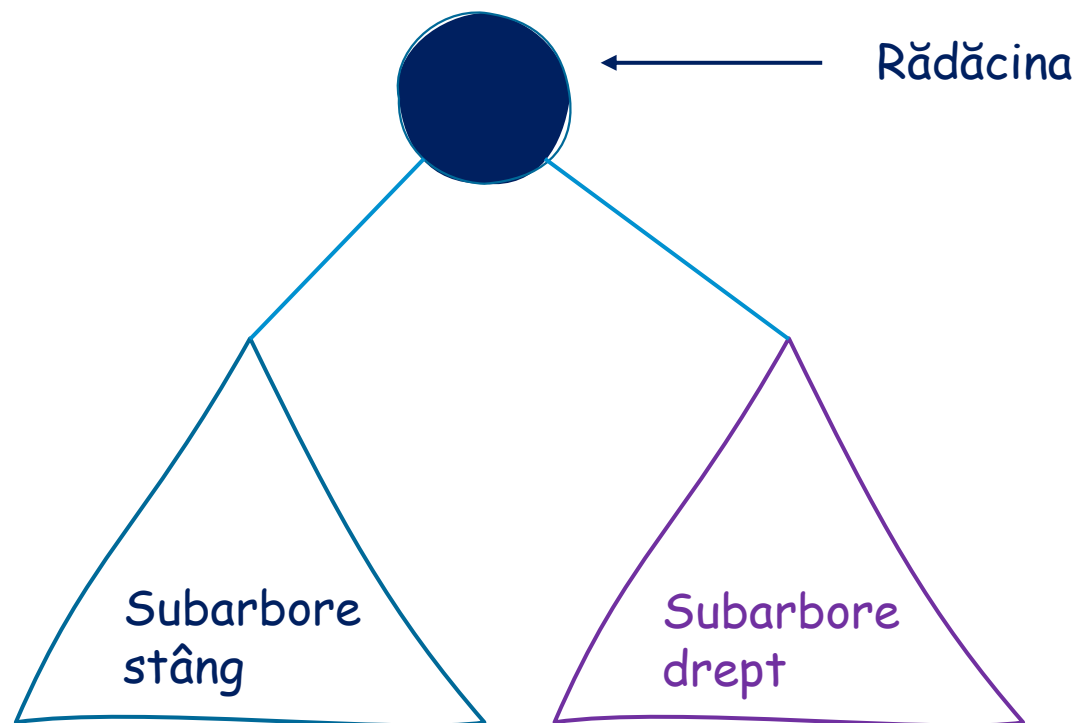
Definiții:



Înălțimea arborelui
 Cea mai mare dintre
 înălțimile nodurilor

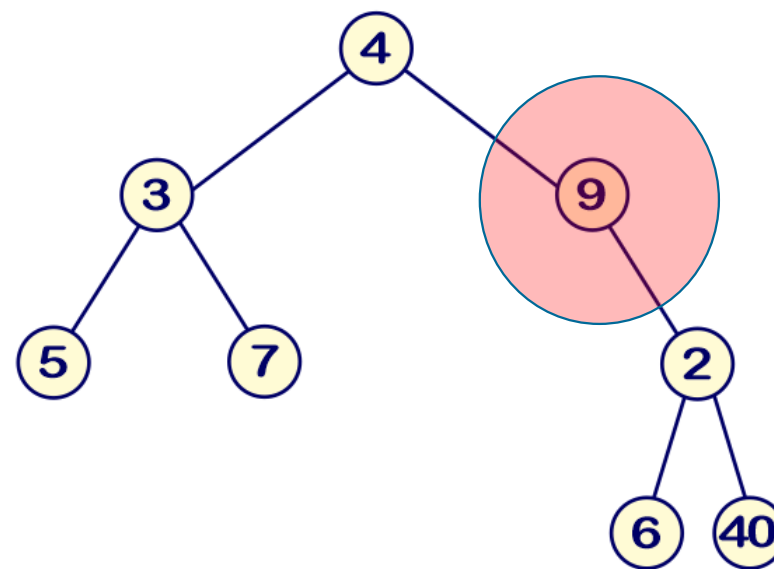
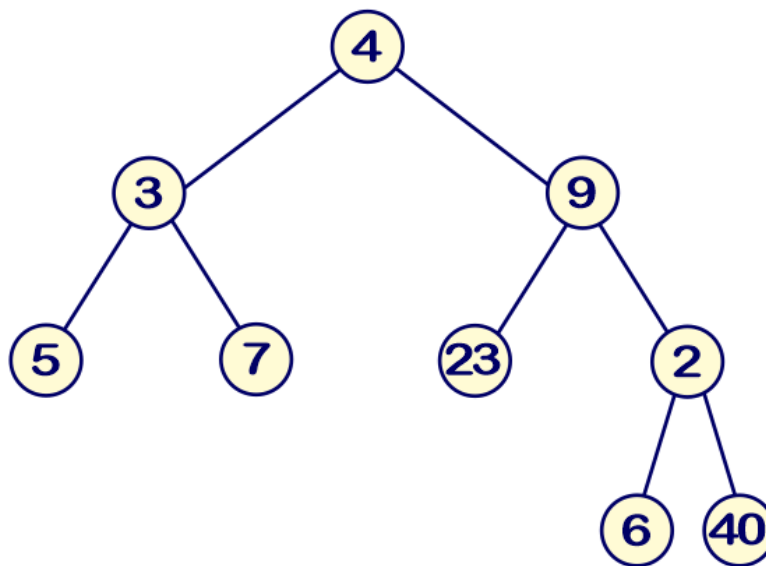
Înălțimea nodului
 Numărul de referințe consecutive de
 la nod către cel mai îndepărtat
 descendent. 3 are înălțimea 1, 9 - 2

Definiții:



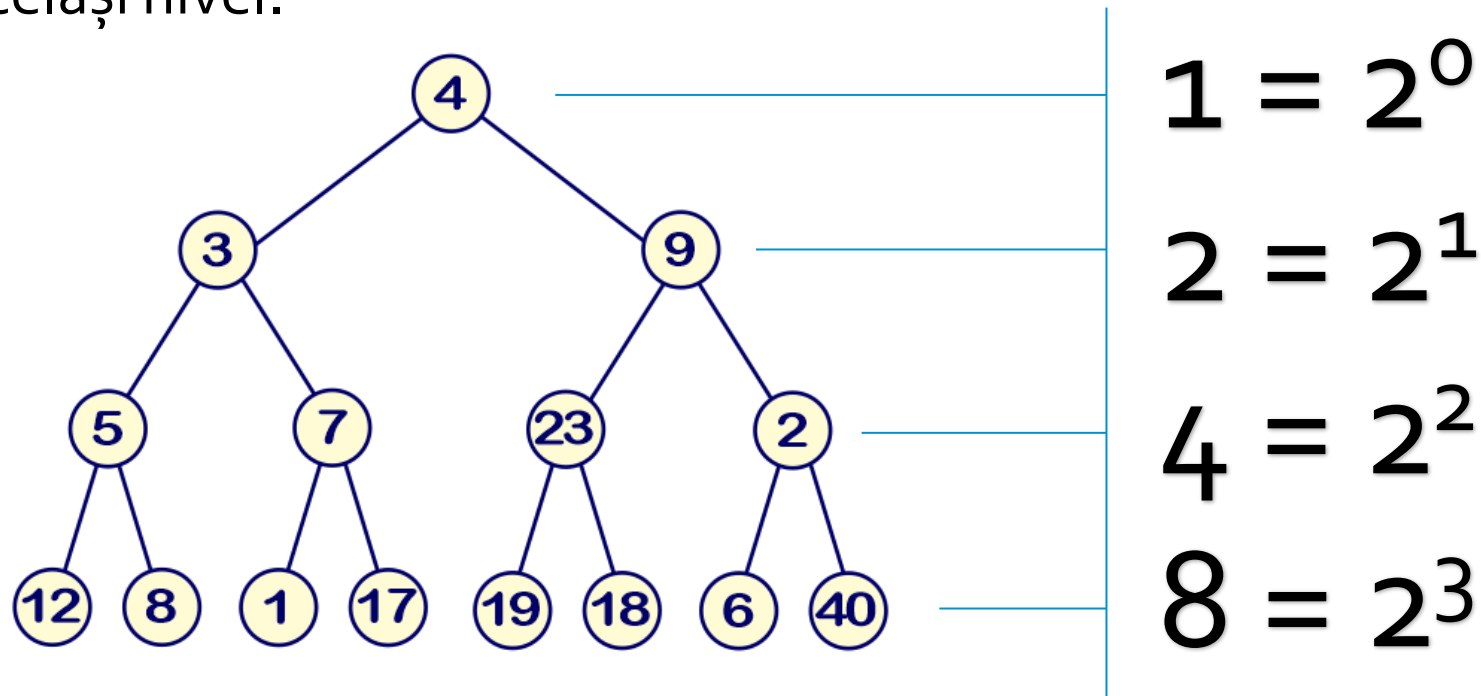
Definiții:

Un arbore este considerat **strict binar**, dacă orice nod al său are fie doi fii, fie nici unul.



Definiții:

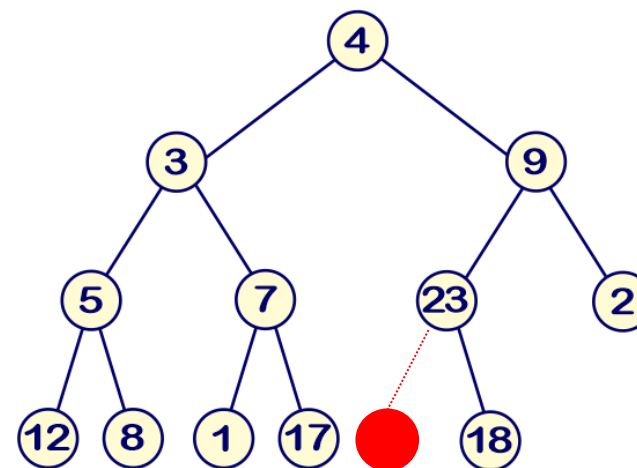
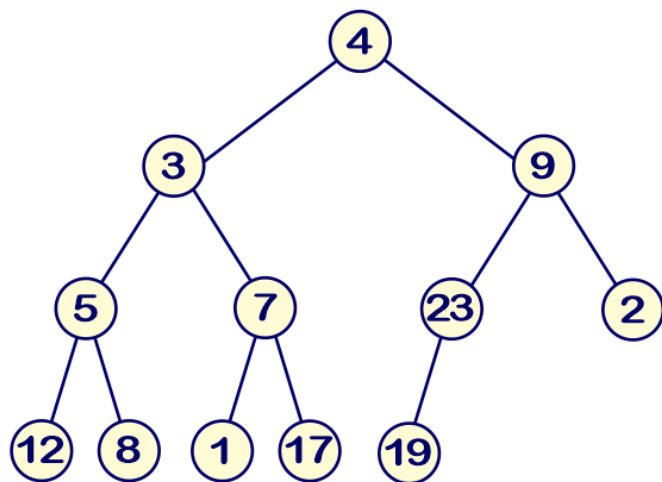
Un **arboare binar** este considerat **plin**, dacă orice nod al său are exact doi fii și toate nodurile – frunze sunt situate la același nivel.



Definiții:

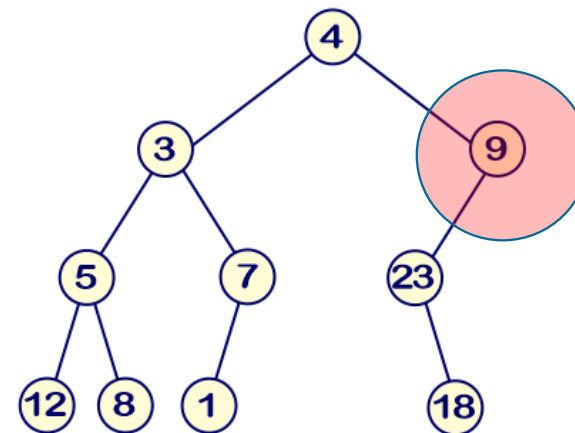
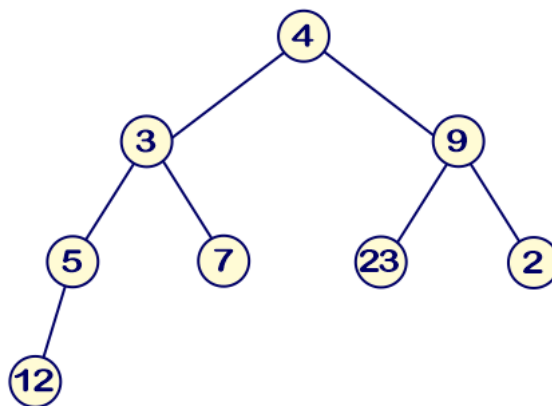
Un **arboare binar** cu înălțimea h este considerat **complet**, dacă:

1. Toate nodurile frunze sunt situate la nivelul h sau $h-1$.
2. Orice nivel $i = 0, \dots, h-1$ conține exact 2^i noduri
3. Nodurile – frunze apar consecutiv, fără goluri de la stânga la dreapta



Definiții:

Un **arbore binar** este considerat **balansat**, dacă pentru orice nod al său diferența dintre înălțimea subarborelui stâng și drept nu depășește 1.



Structura nodurilor:

Structura clasică

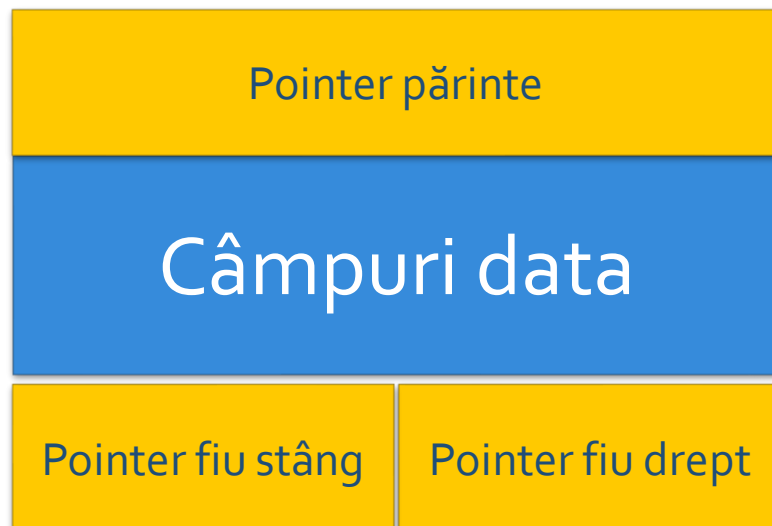
Câmpuri data

Pointer fiu stâng

Pointer fiu drept

```
struct node {  
    int data;  
    struct node *left;  
    struct node *right;  
};
```

Structura extinsă



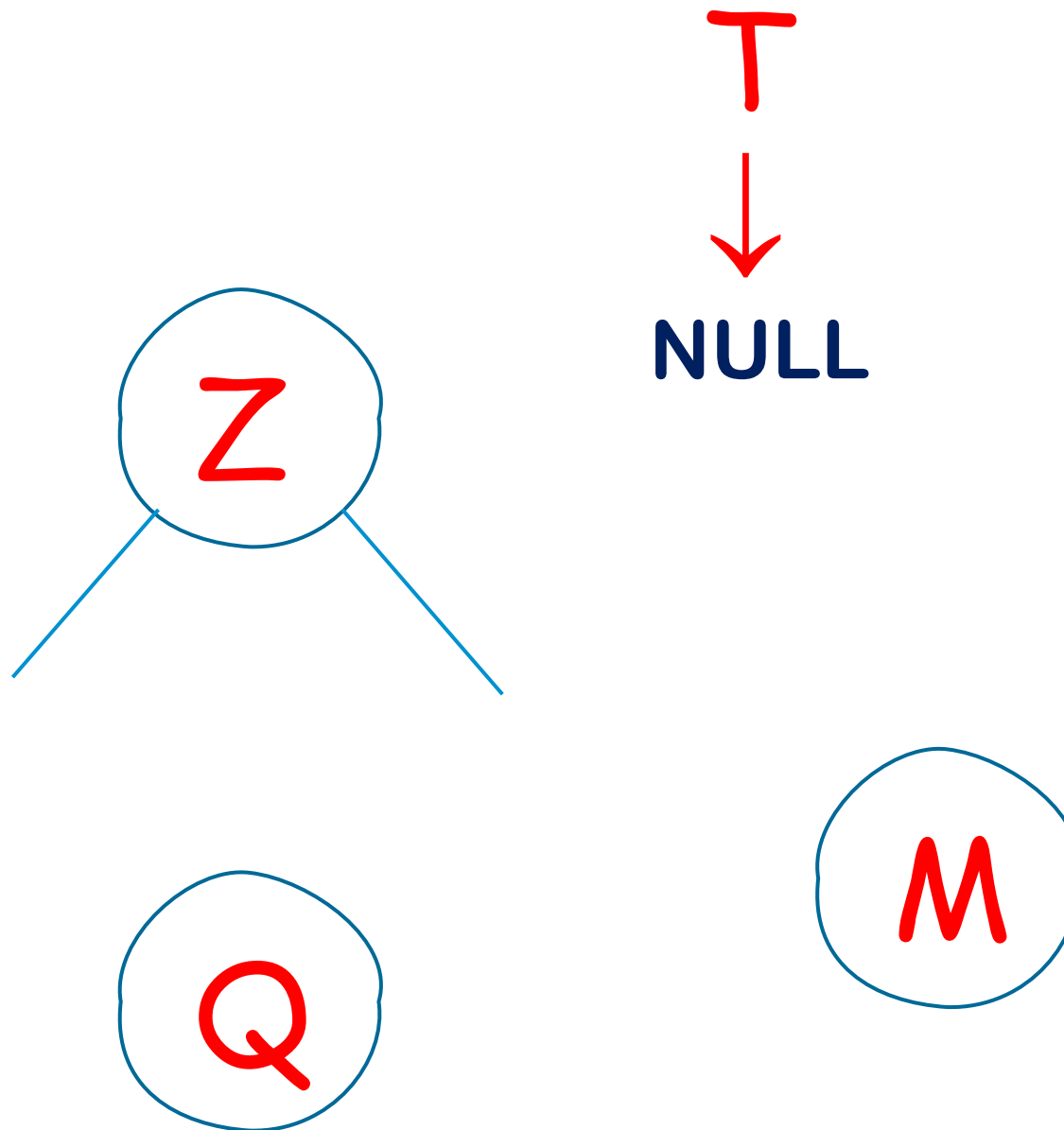
```
struct node {  
    int data;  
    struct node *up;  
    struct node *left;  
    struct node *right;  
};
```


Operații pe arbori:

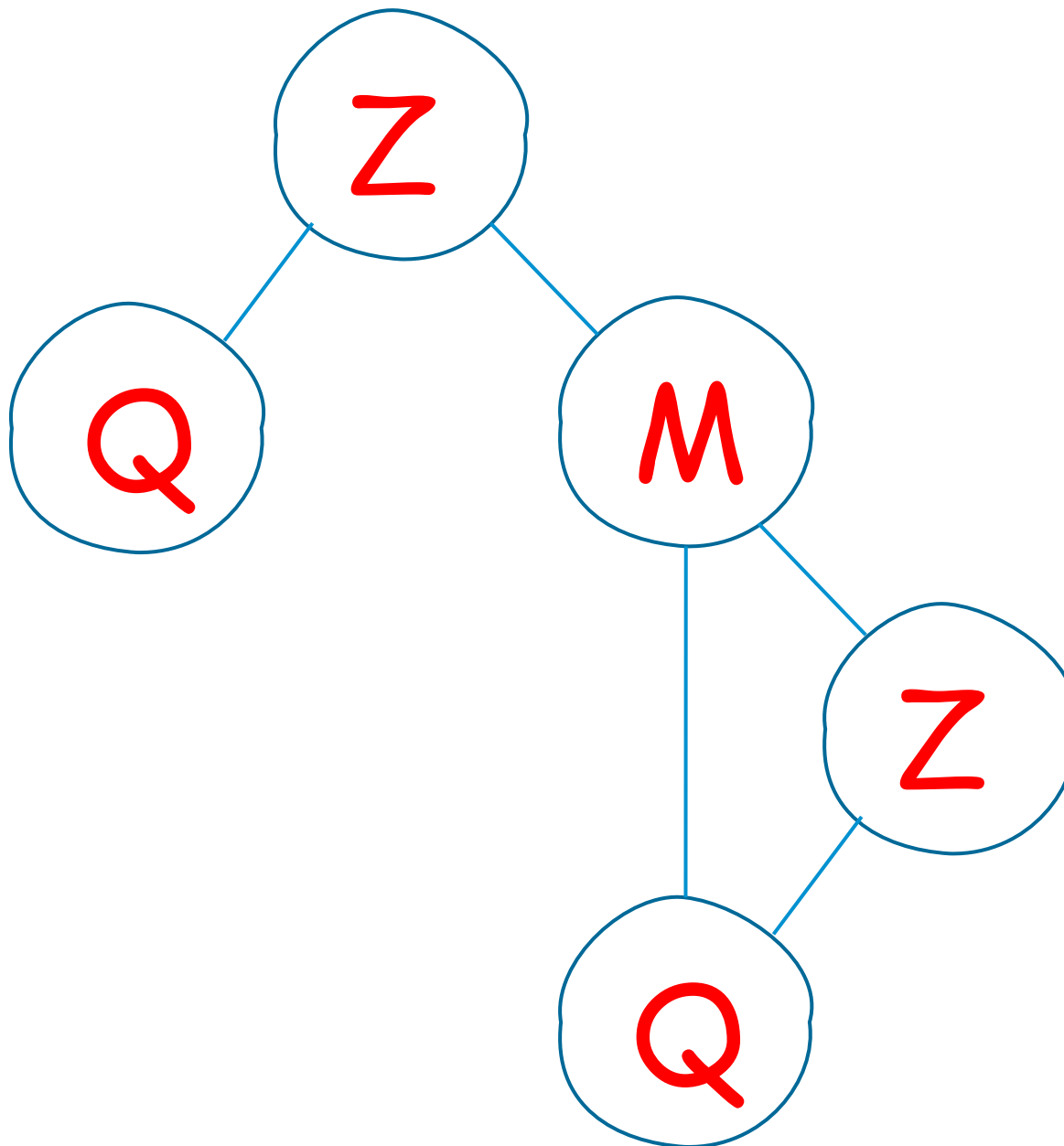
Operații care implică modificarea structurii:

- Inserarea unui element (nod)
- Excluderea unui element (nod)

Adăugarea unui element



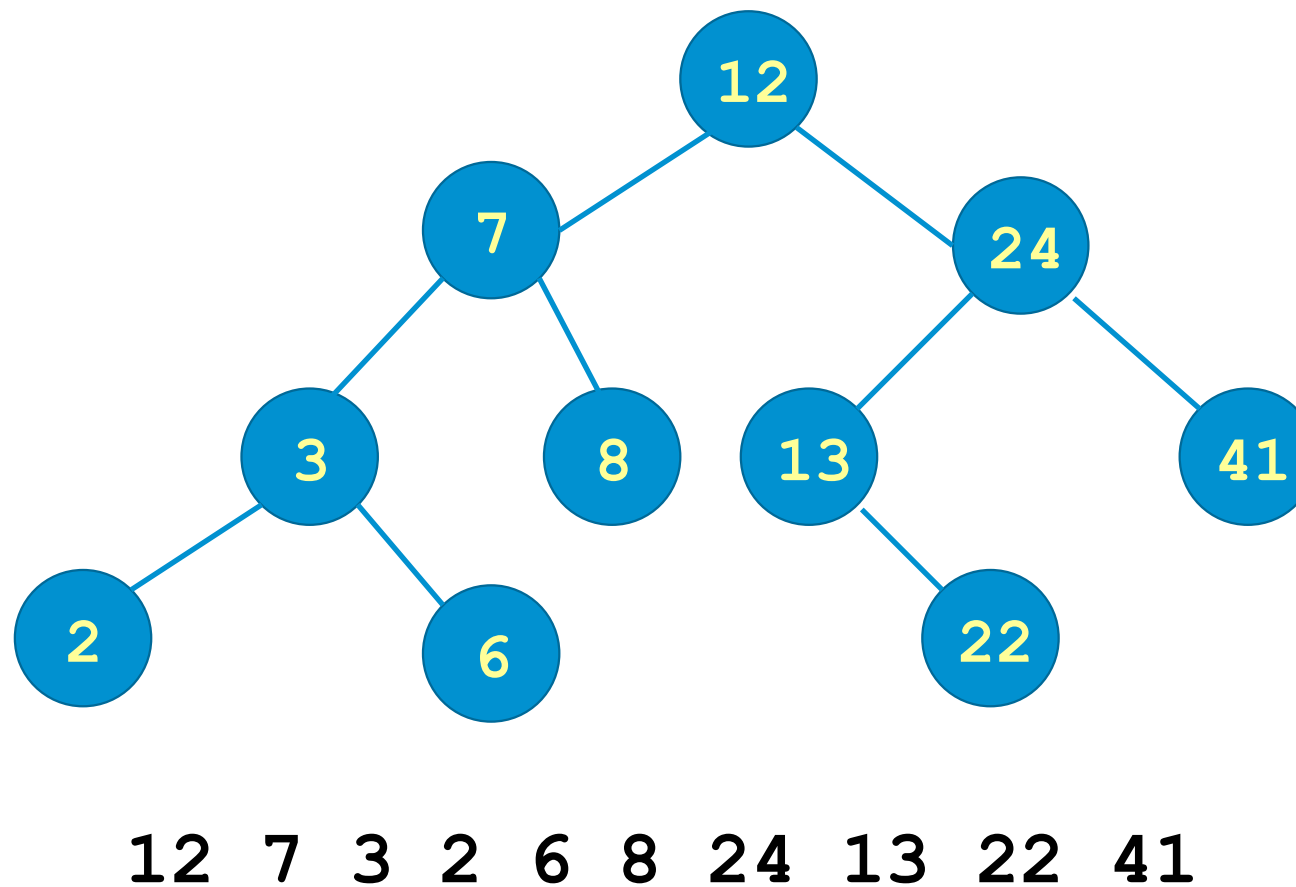
Excluderea unui element



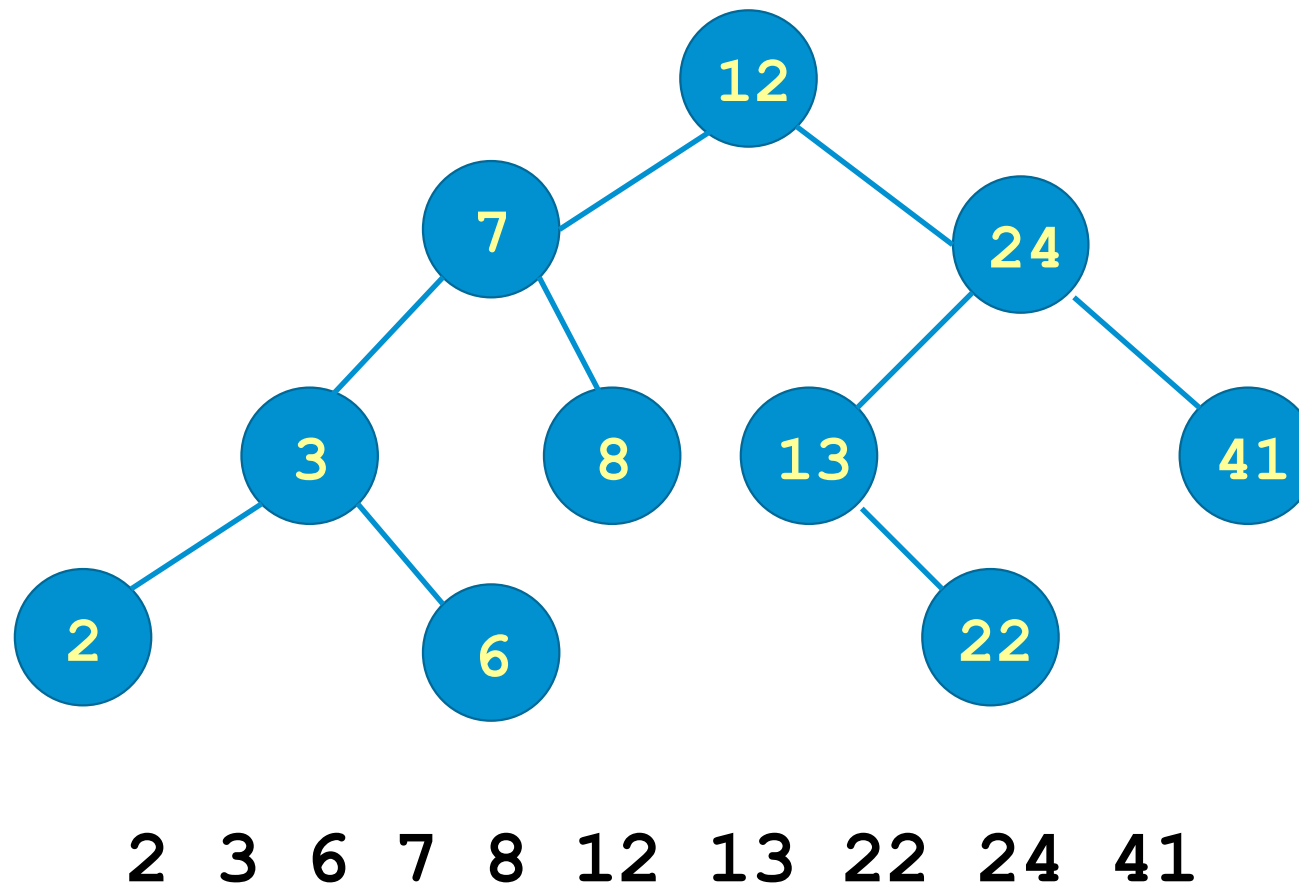
Operații care
NU implică
modificarea
structurii:

- Parcurgerea arborelui
- Căutarea unui element
- Determinarea înălțimii arborelui

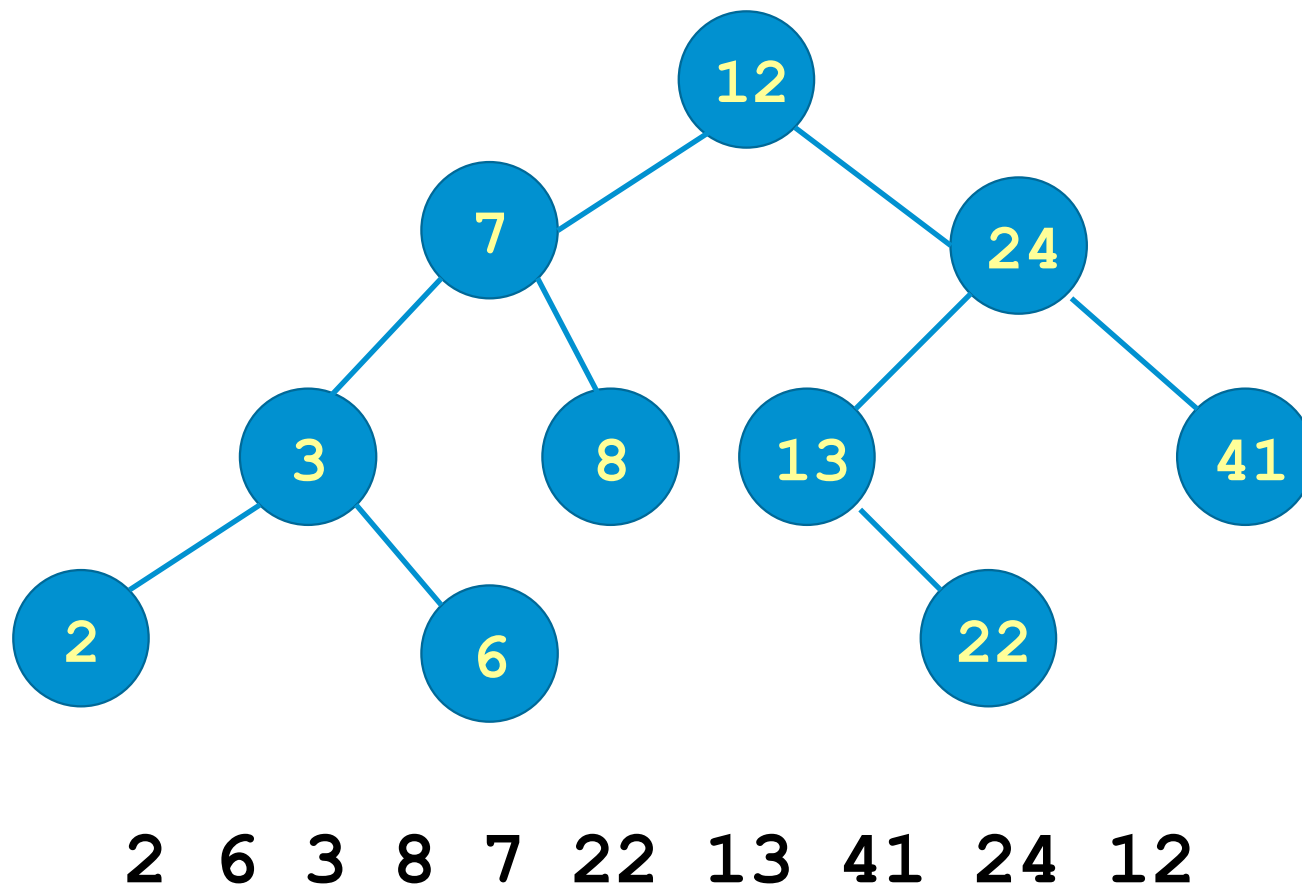
Parcurgerea în PREordine



Parcurgerea în Inordine



Parcurgerea în POSTordine



Implementări!

Lucrul individual :

Modificați exemplul prezentat astfel încât lichidarea să fie efectuată pentru cel mai din stânga nod

Scrieți o funcție pentru calcularea înălțimii arborelui binar

În următoarea sesiune:

- Arbori binari de căutare