

## Clasa 11A–Informatică

### Arbore Binari. Proprietăți, clasificare și reprezentare

#### Definiție

**Arborele binar** este un **arbore cu rădăcină** cu proprietatea că fiecare nod al său are cel mulți 2 descendenți direcți (succesori) denumiți **subarborele stâng As** și **subarborele drept Ad**.

În **Figura 1** apare reprezentarea grafică a unui arbore binar.

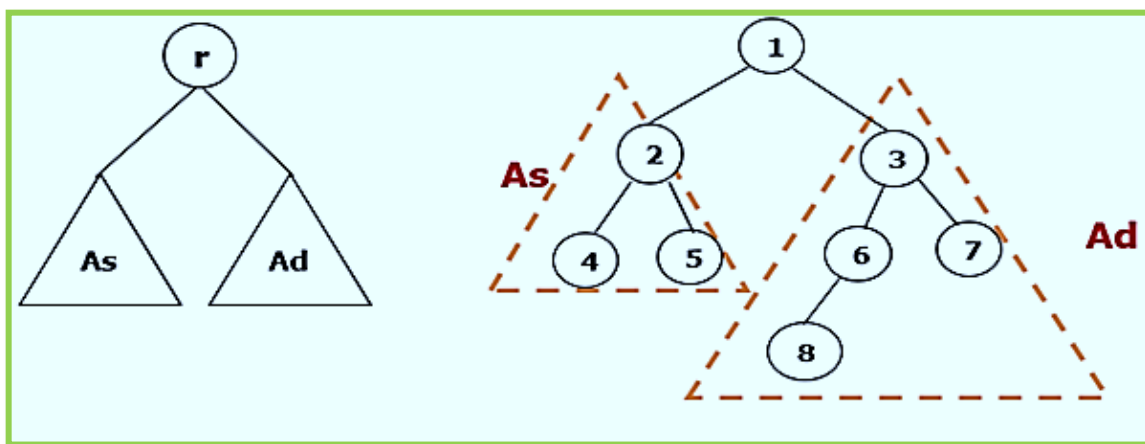


Figura 1 Arbore binar

#### Proprietățile arborilor binari

1. Pentru un arbore binar există un **nod privilegiat** numit **nod rădăcină**, iar celelalte noduri sunt repartizate în 2 grupuri disjuncte, care, fiecare la rândul său formează un arbore binar.
2. Ordinul unui nod dintr-un arbore binar poate fi:
  - 0 pentru nodul terminal
  - 1 dacă unul din subarbori este vid
  - 2

### 1.2.2 Tipuri speciale de arbori binari

Există următoarele **tipuri speciale** de arbori binari:

- **strict;**
- **complet;**
- **aproape complet;**
- **echilibrat;**
- **perfect echilibrat;**
- **degenerat.**

## Arbore binar strict

### Definiție

Se numește **arbore binar strict** un arbore binar cu proprietatea că, fiecare nod, cu excepția nodurilor terminale, are exact 2 succesori.

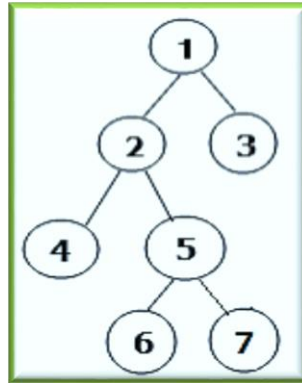


Figura 2 Arbore binar strict

### Proprietăți ale arborelui binar strict

#### Proprietatea1

Un **arbore binar strict** cu  $n$  noduri terminale are  $2*n-1$  noduri

#### Proprietatea2

Un **arbore binar strict** are un număr impar de noduri

## Arbore binar complet

### Definiție

Se numește **arbore binar complet** un **arbore binar strict** care are toate **nodurile terminale** pe același nivel.

În **Figura 3** avem un **arbore binar complet** care are nodurile dispuse pe  $k=3$  niveluri:

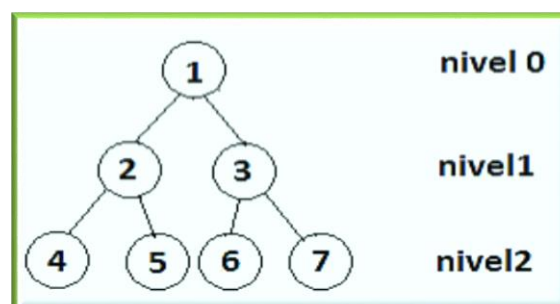


Figura 3 Arbore binar complet

Se observă că, un **arbore binar complet** are  $2^k-1$  noduri dispuse pe  $k$  niveluri:  $0,1,2,...,k-1$  astfel încât pe fiecare nivel  $i$  se găsesc  $2^i$  noduri,  $k \in \mathbb{N}$ .

Pentru exemplul nostru  $k=3$  niveluri și fiecare nivel  $i \in [0,k-1]$

Deci înălțimea arborelui care este nivelul maxim în arbore, va fi 2.

Pe nivelul 0 →  $2^0$  noduri=1 și anume nodul 1

Pe nivelul 1 →  $2^1$  noduri=2 și anume nodurile 2 și 3

Pe nivelul 2 →  $2^2$  noduri=4 și anume nodurile 4, 5, 6 și 7

## Arbore binar aproape complet

### Definiție

Se numește **arbore binar aproape complet** un **arbore binar complet** până la penultimul nivel, la care completarea cu noduri la ultimul nivel se face de la stânga la dreapta.

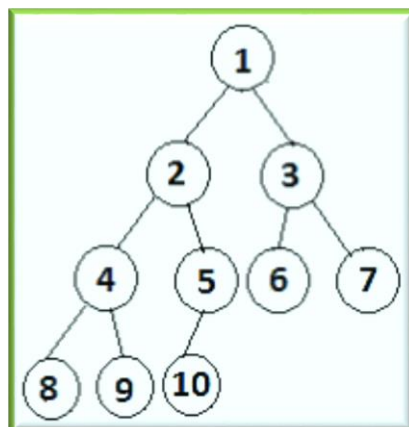


Figura 4 Arbore binar aproape complet

## Arbore binar echilibrat

### Definiție

Un **arbore binar echilibrat** este un **arbore binar** cu proprietatea că **diferența înălțimilor** celor 2 subarbori ai oricărui nod este cel mult 1.

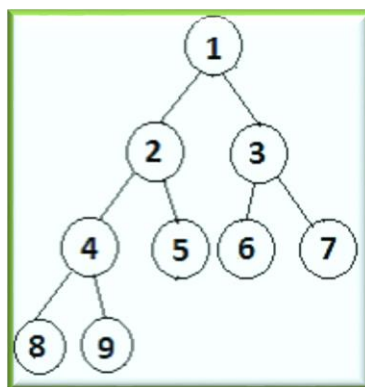


Figura 5 Arbore binar echilibrat

## Arbore binar perfect echilibrat

### Definiție

Un **arbore binar perfect echilibrat** este un **arbore binar** care are proprietatea că, pentru fiecare nod al său, diferența dintre numărul de noduri dintre **subarborele stâng** și **subarborele drept** este *cel mult* 1.

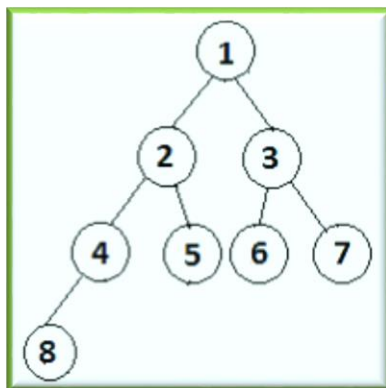


Figura 6 Arbore binar perfect echilibrat

## Arbore binar degenerat

### Definiție

Se numește **arbore binar degenerat** un **arbore binar** care are **n** noduri dispuse pe **n** niveluri.

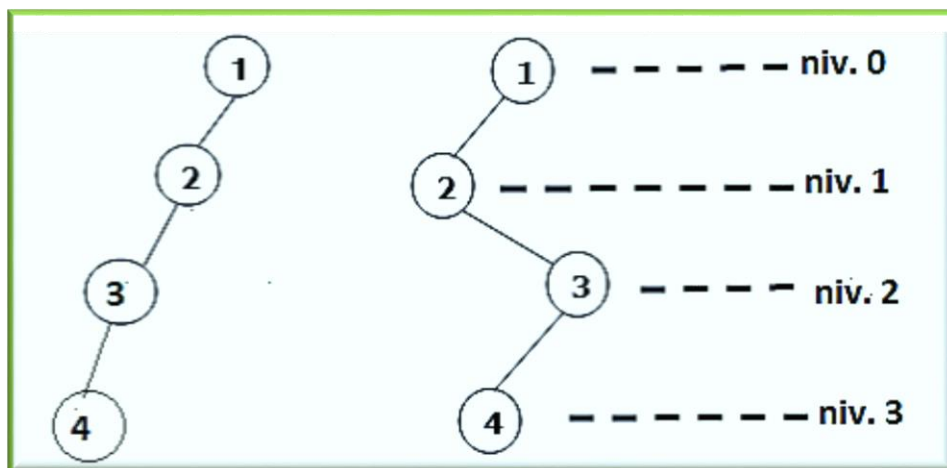


Figura 7 Arbori binari degenerați

Cei doi arbori binari de mai sus au 4 noduri aranjate pe 4 niveluri.

### Definiție

Pentru un arbore binar se numește **cheie(etichetă) a unui nod** câmpul de informație utilă care poate fi folosit pentru a identifica unic nodurile arborelui.

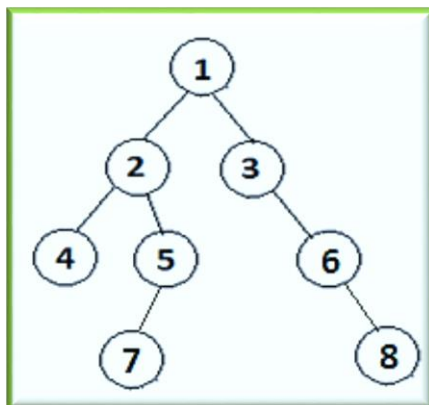
## Metode de reprezentare a arborilor binari

Implementarea structurii de date de tip arbore binar se poate face în două moduri:

1. Implementare statică utilizând vectori
2. Implementare dinamică folosind pointeri

### 1. Implementarea statică

Fie un arbore binar reprezentat în **Figura 8** de mai sus:



Sunt două metode pentru implementarea statică a unui arbore binar:

- a. La prima metodă se folosesc 2 vectori cu numele **tata** și **fii**

- ✚ elementul **tata[i]** memorează eticheta **nodului părinte** al nodului **i**;
- ✚ elementul **fii[i]** poate lua una dintre valorile: **-1** dacă **nodul i** este **successor stâng** al tatălui său sau **1** dacă **nodul i** este **successor drept** al tatălui său.

Pentru arborele din **Figura 8**, iar valorile elementelor celor 2 vectori sunt:

Vectorul tata

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 2 | 3 | 5 | 6 |
|---|---|---|---|---|---|---|---|

Nodul i

1   2   3   4   5   6   7   8

Vectorul fii

|   |    |   |    |   |   |    |   |
|---|----|---|----|---|---|----|---|
| 0 | -1 | 1 | -1 | 1 | 1 | -1 | 1 |
|---|----|---|----|---|---|----|---|

### Observații

- ✚ Nodul rădăcină este **nodul i** pentru care **tata[i]=0** și **fii[i]=0**
- ✚ Nodurile frunză sunt nodurile a căror **etichetă nu există** în vectorul **tata**

Se observă că:

- **tata[1]=0** și **fii[1]=0** deci **rădăcina r** are **eticheta 1**
- Nodurile a căror **etichetă** nu se găsește printre **valorile vectorului tata** sunt **4, 7 și 8**.

Atunci, **frunzele arborelui** sunt nodurile **4, 7 și 8** așa cum se poate observa și din figură.

- b. Pentru a doua metodă se specifică rădăcina **r** și se utilizează 2 vectori cu numele **stg** și **dr** unde:
- elementul **stg[i]** memorează eticheta **nodului successor stâng** al nodului **i**

