Structuri de date – Curs 6

Prof. univ. dr. Cristian CIUREA
Departamentul de Informatica si Cibernetica Economica
Academia de Studii Economice din Bucuresti
cristian.ciurea@ie.ase.ro

Agenda

- Arbori oarecare
- Arbori binari
- Arbori binari de căutare

Arbori

- Arborii sunt structuri de date dinamice şi omogene.
- In arborescenţă, există un nod numit rădăcină sau părinte. Acesta are descendenţi. Fiecare descendent poate fi, la rândul său, părinte şi, în acest caz, are descendenţi.

Arbori

- În teoria grafurilor, un arbore este un graf neorientat, conex și fără cicluri.
- Arborii reprezintă grafurile cele mai simple ca structură din clasa grafurilor conexe, ei fiind și cei mai frecvent utilizați în practică.
- Termenul de „arbore” din teoria grafurilor a fost folosit pentru prima dată de *Cayley* în anul 1857. El a plecat de la o analogie cu noțiunea de arbore din botanică.

Arbori

- Domenii de aplicabilitate:
  - în chimia organică, de ex. grafurile chimice;
  - în fizică, de ex. în studiul rețelelor electrice;
  - în informatică;
  - etc.

Arbori

- Dacă *G* este un graf neorientat, atunci:
  - *G* este arbore;
  - *G* este un graf conex minimal (dacă i se elimină orice muchie, se obține un graf neconex);
  - *G* este un graf aciclic maximal (dacă i se adaugă orice muchie, se obține un graf care are măcar un ciclu și atunci nu ar mai fi arbore).
  - dacă *G* are *n* vârfuri, atunci are *n–1* muchii.

Arbori

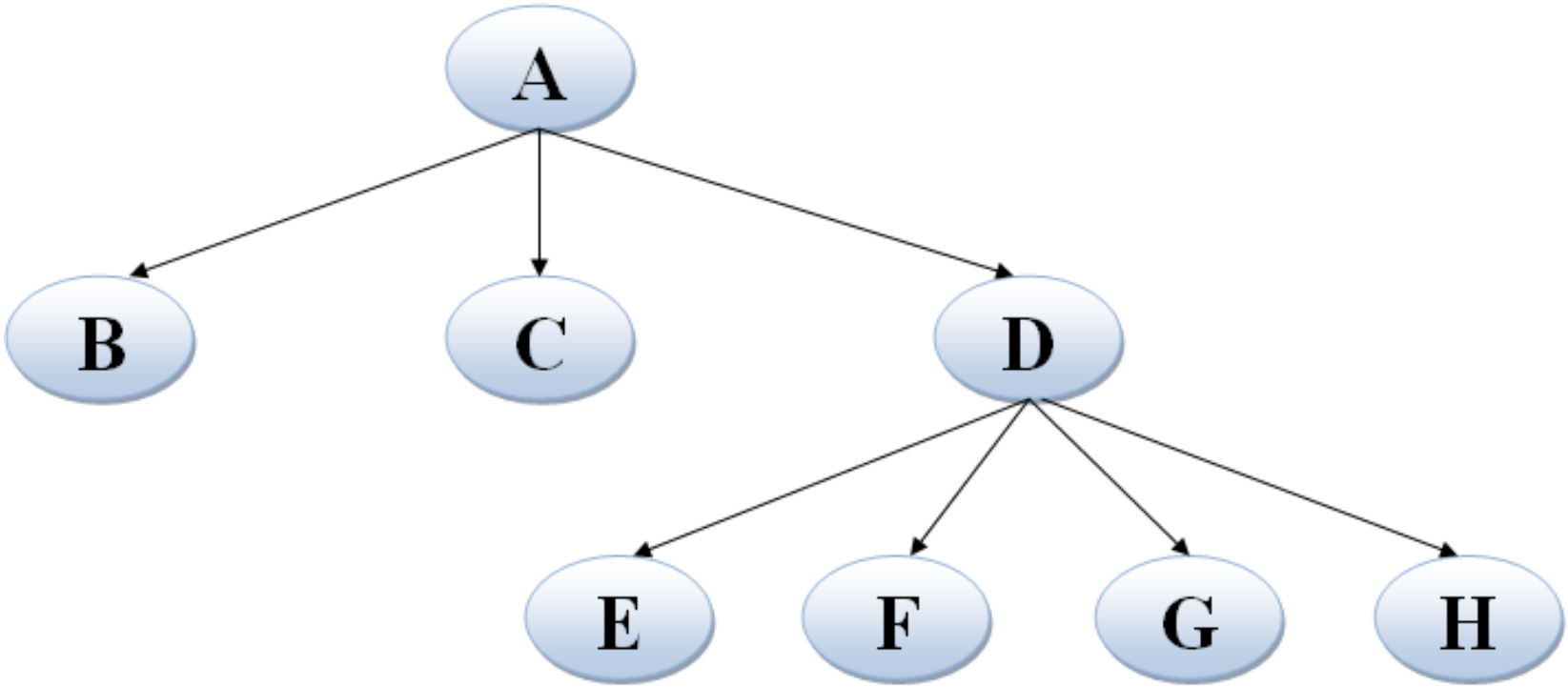
- Categorii de arbori:
  - arbori oarecare;
  - arbori binari;
  - arbori binari de căutare;
  - arbori (perfect) echilibrați;
  - arbori AVL;
  - arbori B;
  - arbori de structură;
  - etc.

Arbori

- Arbore oarecare:
  - graf aciclic, conex si orientat;
  - un nod rădăcină (root);
  - reprezentări mai eficiente față de grafuri: FIU-FRATE, structuri în memoria heap;
  - traversare: regula de vizitare a nodurilor;
  - topologii particulare: binari, de căutare, echilibrați, etc.

Arbori

- arbore oarecare:

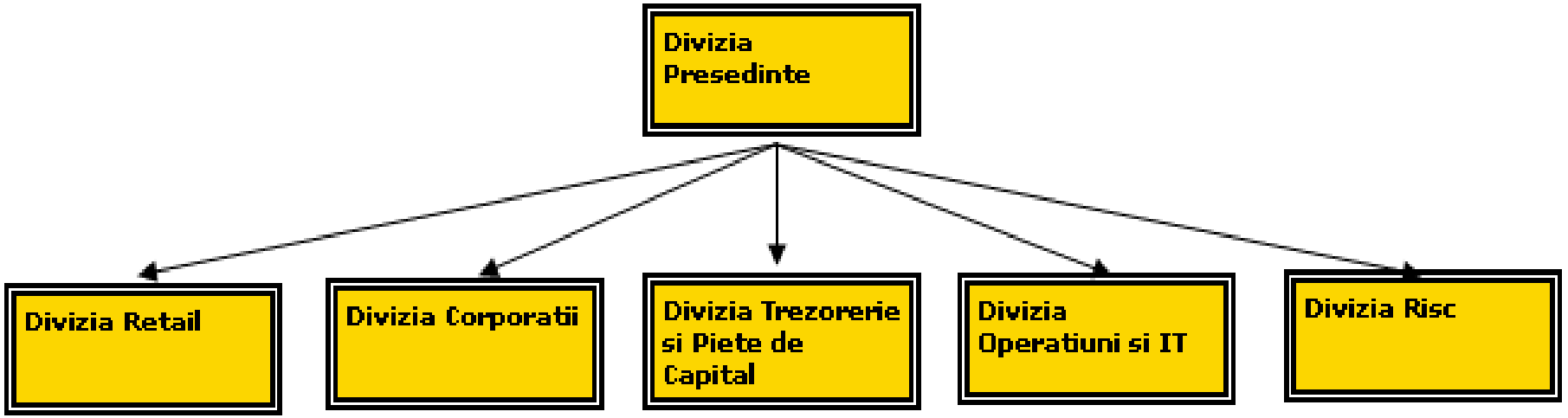


Arbori

- Exemple de arbori oarecare:
  - structura organizatorică (organigrama) a unei companii;
  - arborescența unui website;
  - arborele genealogic;
  - organizarea cărților într-o bibliotecă;
  - etc.

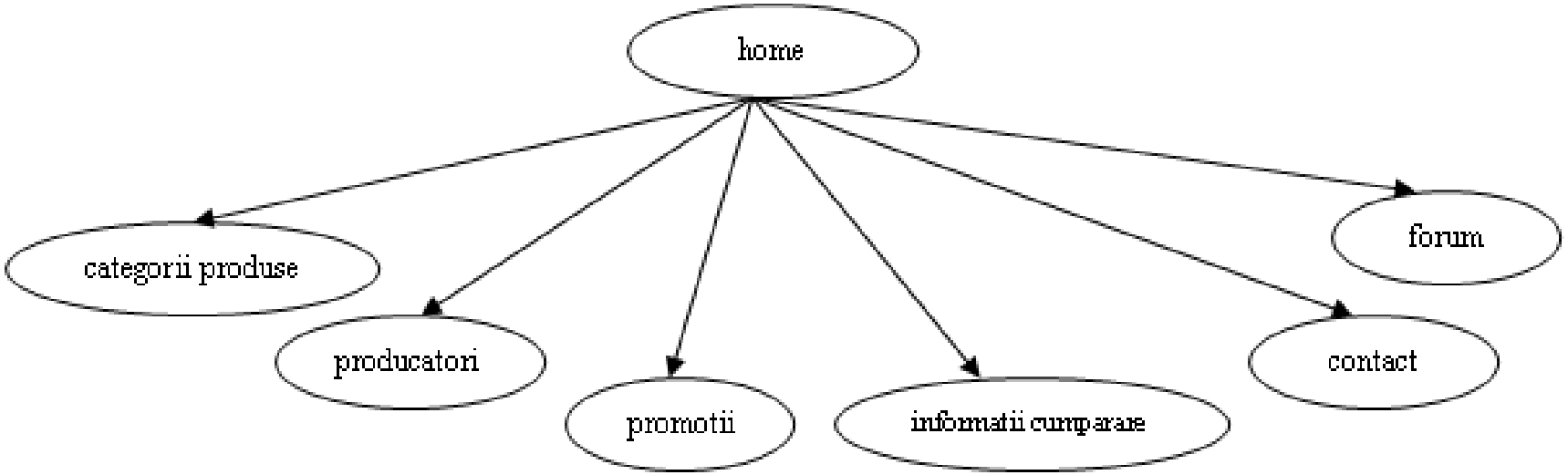
Arbori

- organigrama unei companii (bancă):



Arbori

- arborescența unui website (magazin virtual):



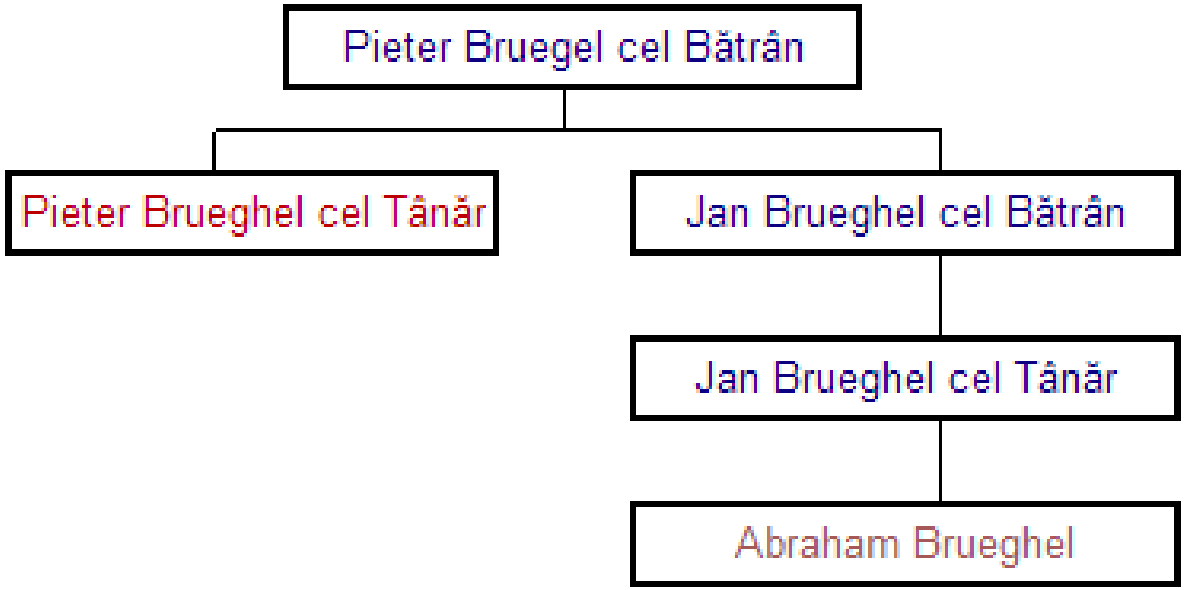
Arbori

- arborescența unui website (magazin virtual):

```
<siteMap>
   <siteMapNode title="Home" url="">
         <siteMapNode title="Categorii" url=""/>
         <siteMapNode title="Producatori" url=""/>
         <siteMapNode title="Promotii" url="">
         <siteMapNode title="Informatii" url="">
                  <siteMapNode title="Reclamatii" url="">
                           <siteMapNode title="Sesizari" url=""/>
                  </siteMapNode>
         </siteMapNode>
         <siteMapNode title="Contact" url=""/>
         <siteMapNode title="Forum" url=""/>
   </siteMapNode>
</siteMap>
```

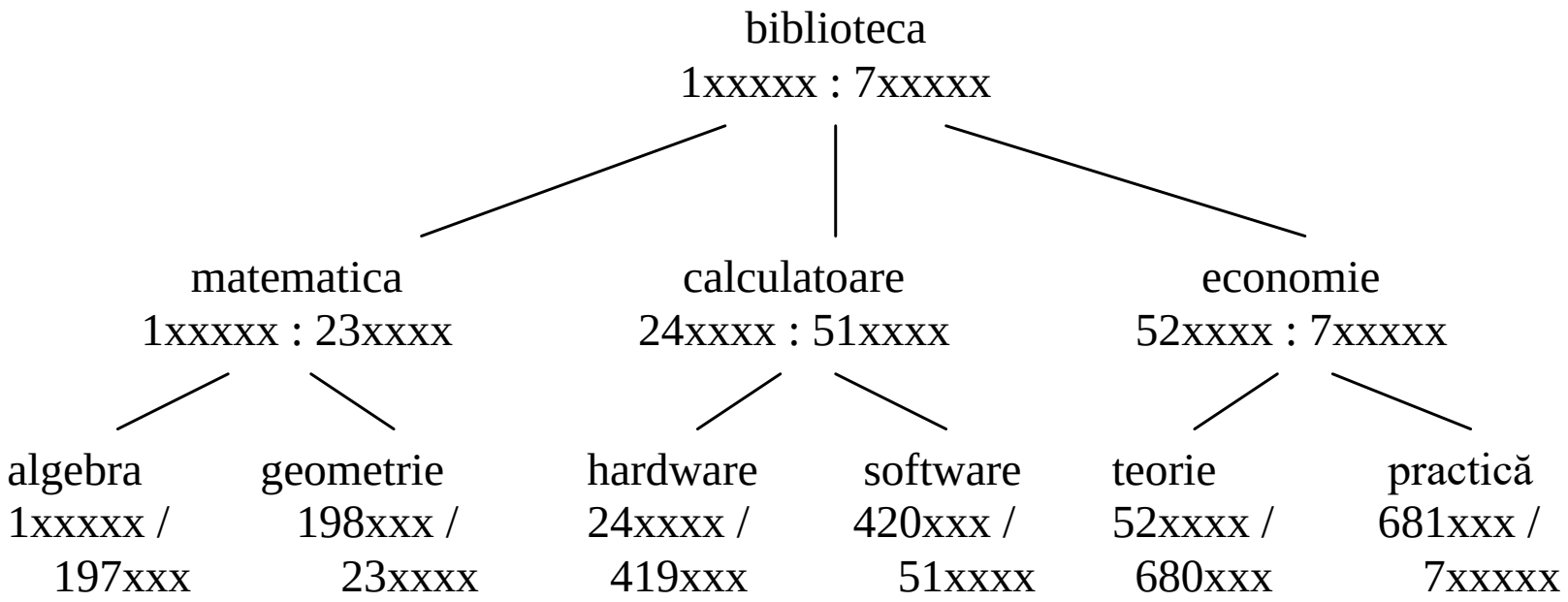
Arbori

- arbore genealogic (familia Bruegel):



Arbori

- organizarea cărților într-o bibliotecă:

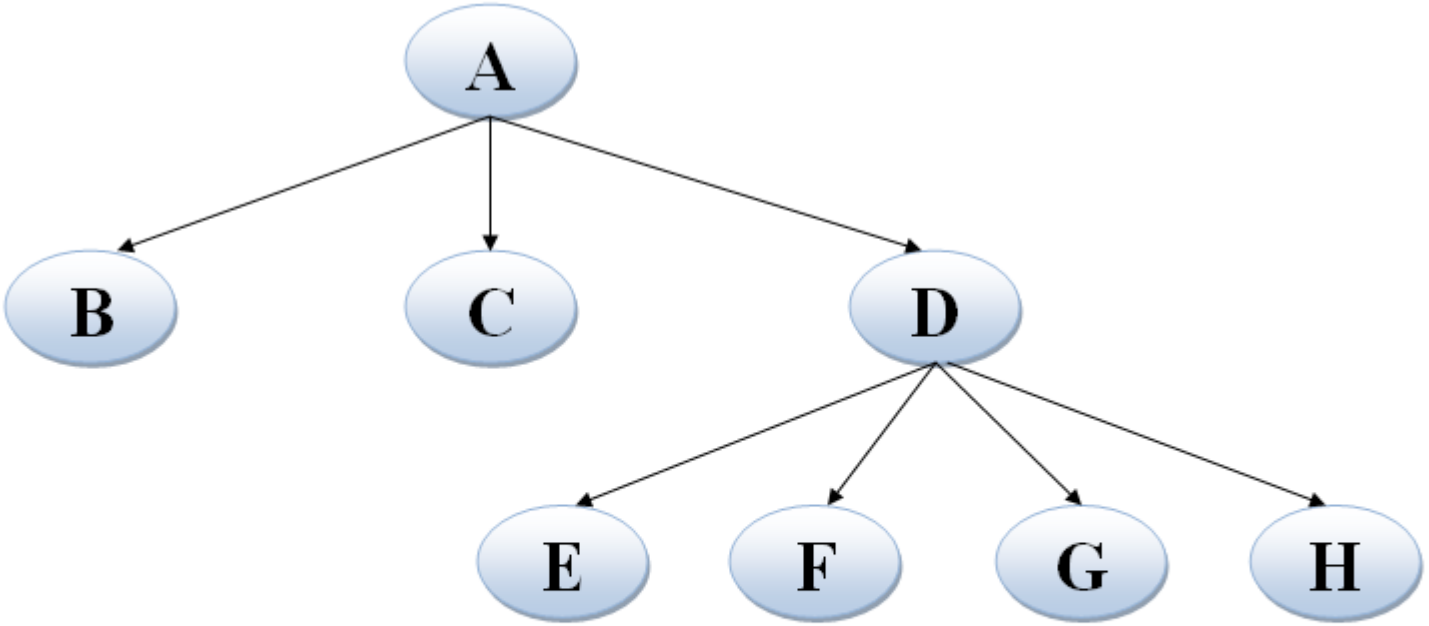


Arbori

- Arbori oarecare:
  - reprezentarea FIU-FRATE a unui arbore;
  - structura unui nod din arbore include:
    - ID nod pentru primul descendent;
    - ID frate al primului descendent;
    - informația stocată de nod.

Arbori

- reprezentare FIU-FRATE:



Nod[ ] = {A, B, C, D, E, F, G, H}
Fiu[ ] = {B, ø, ø, E, ø, ø, ø, ø }
Frate[ ] = {ø, C, D, ø, F, G, H, ø }

Arbori

- Arbori oarecare:
  - Structuri de memorare:
    - numărul de noduri ale arborelui;
    - ID nod rădăcină;
    - vector primi descendenți;
    - vector frați ai primilor descendenți.
  - Reprezentarea prin structuri memorate în heap:
    - ID nod;
    - vector/listă/altă structură liniară cu adrese ale descendenților nodului;
    - eventual, se precizează numărul maxim de descendenți (memorare adrese descendenți în vector).

Arbori

- Arbori oarecare:

```
/* definirea structurii unui nod de arbore oarecare
  cu maxim 10 descendenti */
struct nodArb{
   int inf;
   nodArb* fiu[10];
};

…

/* definirea variabilei de gestionare a structurii
arborescente */

nodArb* rad = NULL;
```

Arbori

- Reprezentarea în memorie a informaţiilor care au multiple legături între ele trebuie să fie efectuată, astfel încât să se poată realiza parcurgerea completă a zonelor sau localizarea unui element din structură.

Arbori

Traversarea arborilor oarecare:

- Preordine:
  - selecția nodului rădăcină ca nod curent;
  - se vizitează nodul curent;
  - pentru nodul curent se vizitează sub-arborii în ordinea dată în structura de stocare a adreselor acestora.

Arbori

Traversarea arborilor oarecare:

- Postordine:
  - se vizitează sub-arborii în ordinea dată în structura de stocare a adreselor acestora;
  - se vizitează nodul rădăcină (după vizitarea tuturor nodurilor din sub-arbori).

Arbori

Traversarea arborilor oarecare:

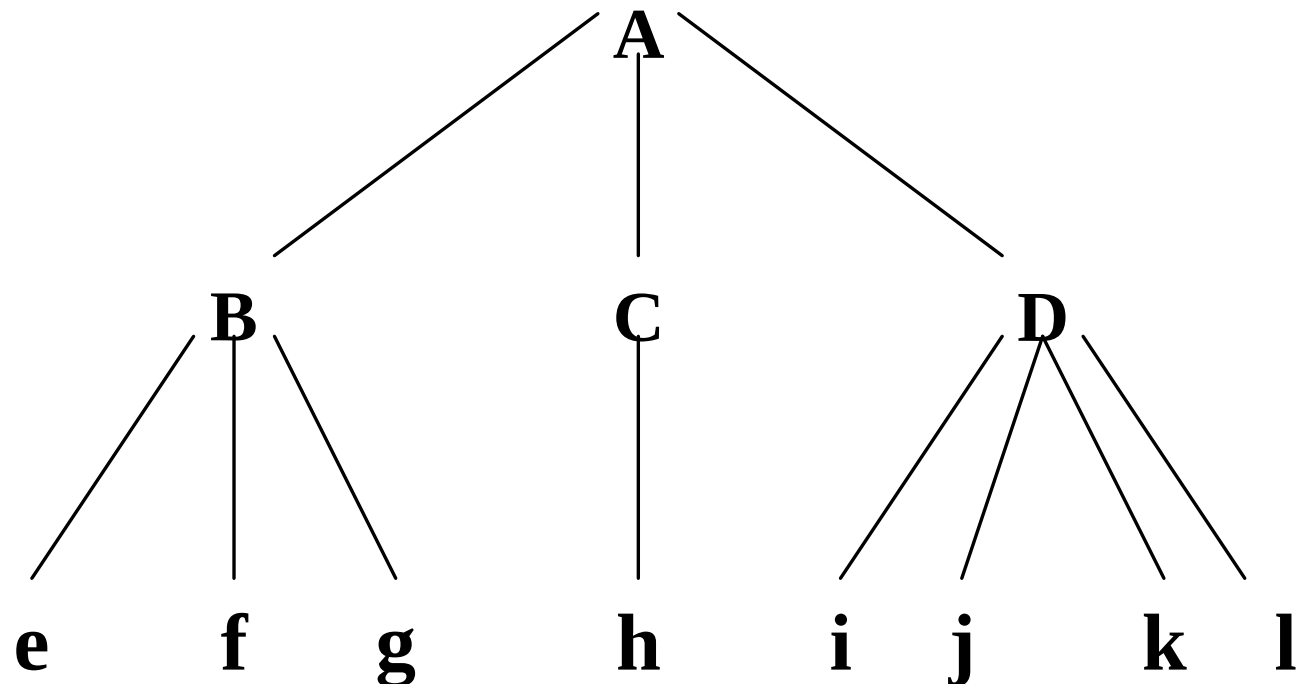
- Pe niveluri:
  - se vizitează nodurile în ordinea crescătoare a distanțelor față de rădăcină.
  - reprezentarea pe niveluri determină să nu mai fie necesar sensul arcelor; fiecare vârf poate fi considerat rădăcină pentru un subarbore.

Arbori

- Problema transformării unei structuri arborescente oarecare într-o structură arborescentă binară este rezolvabilă prin introducerea de noduri fictive.
- Arborele oarecare va avea un număr de noduri mai mic decât arborele binar în urma introducerii nodurilor fictive.

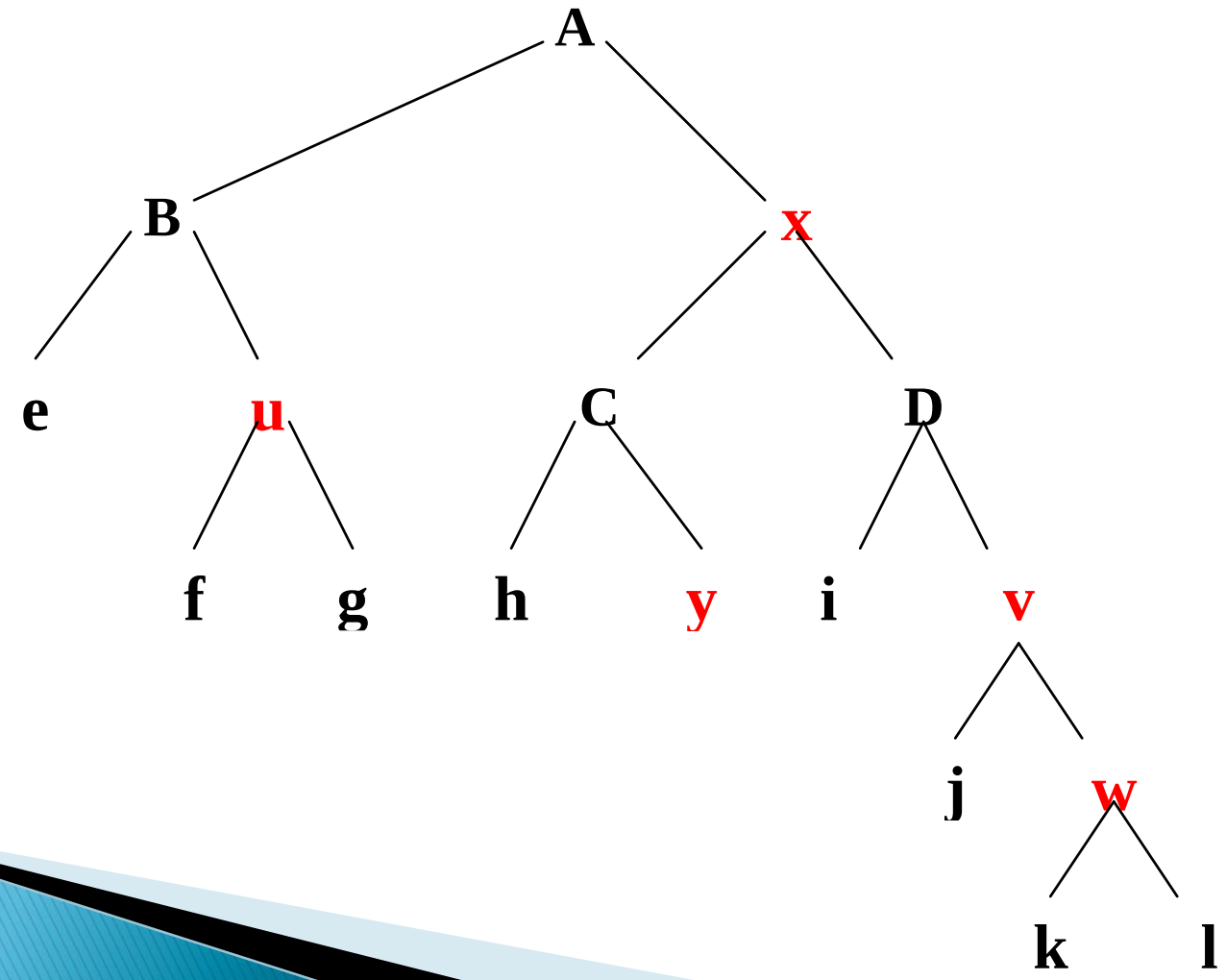
Arbori

- arbore oarecare de transformat în arbore binar:



Arbori

- introducerea nodurilor fictive x, y, u, v, w:



Arbori

- Arbore binar:
  - arbore oarecare cu maxim doi descendenți;
  - orice sub-arbore respectă restricția de la punctul anterior;
  - topologii particulare: arbori binari de căutare, etc;
  - reprezentare: structuri alocate în heap.

Arbori

- Arborele binar: orice nod al său are un singur părinte şi maxim 2 descendenţi/fii.
- Fiecare nod este reprezentat prin trei informaţii:
  - informaţia utilă care face obiectul prelucrării, ea descriind elementul sau fenomenul asociat nodului;
  - informaţia care localizează nodul părinte;
  - informaţiile care localizează nodurile descendente.

Arbori

- **Nod arbore oarecare:**

```
struct arbore_oarecare
{
        int valoare;
        arbore_oarecare **fii;  //vector/lista fiilor unui nod
        int nrfii;              //nr. de fii ai nodului
};
```

- **Nod arbore binar:**

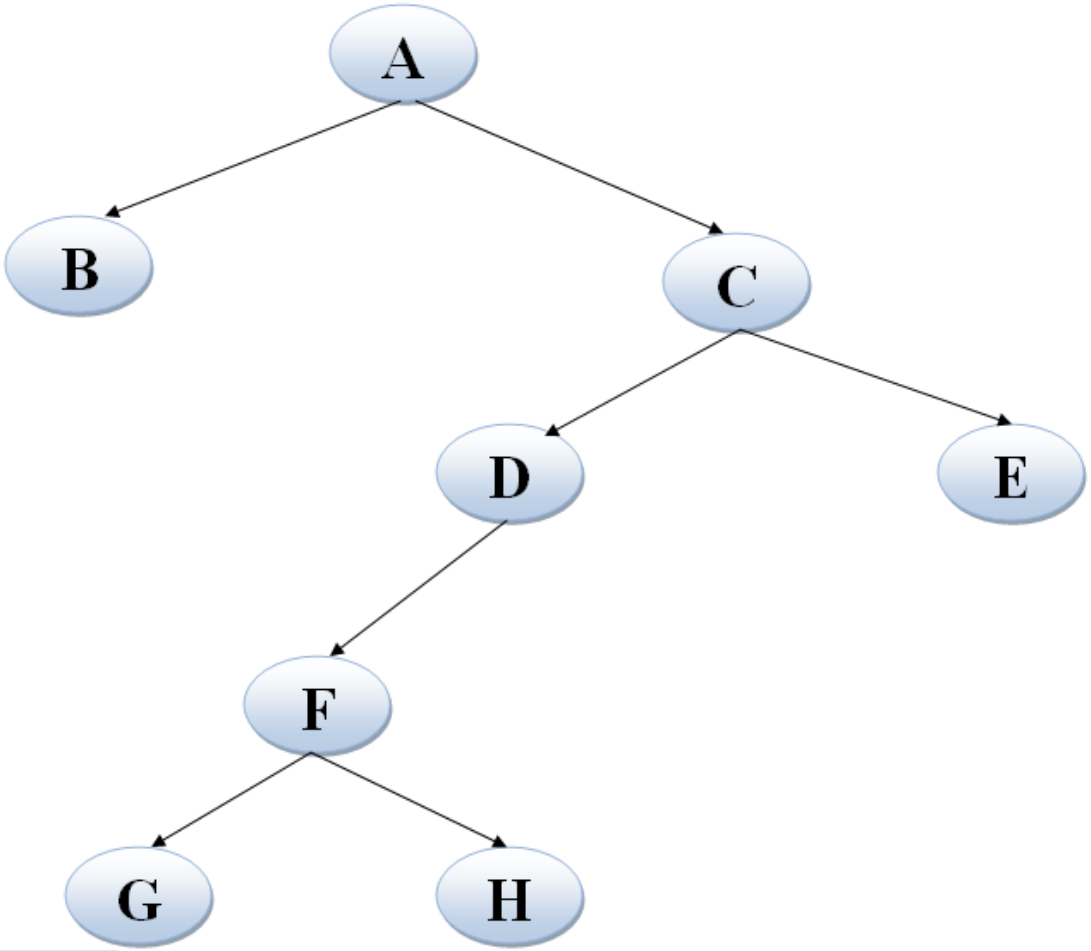
```
struct arbore_binar
{
        int valoare;
        arbore_binar *stanga;
        arbore_binar *dreapta;
};
```

Arbori

- Arborele binar A este o mulţime de chei care poate fi, fie vidă, fie conţine un element numit rădacină şi exact doi arbori binari A_s şi A_d, $A = \{r, A_s, A_d\}$.
- Oricare dintre mulţimile A, A_s şi A_d poate fi vidă, în consecinţă orice nod din arborele binar poate avea 0, 1 sau 2 fii.

Arbori

- arbore binar:



Arbori

- **Rangul** unui nod reprezintă numărul de subarbori asociaţi cu acel nod.
- Un nod care are rangul 0 se numeşte **frunză**.

Arbori

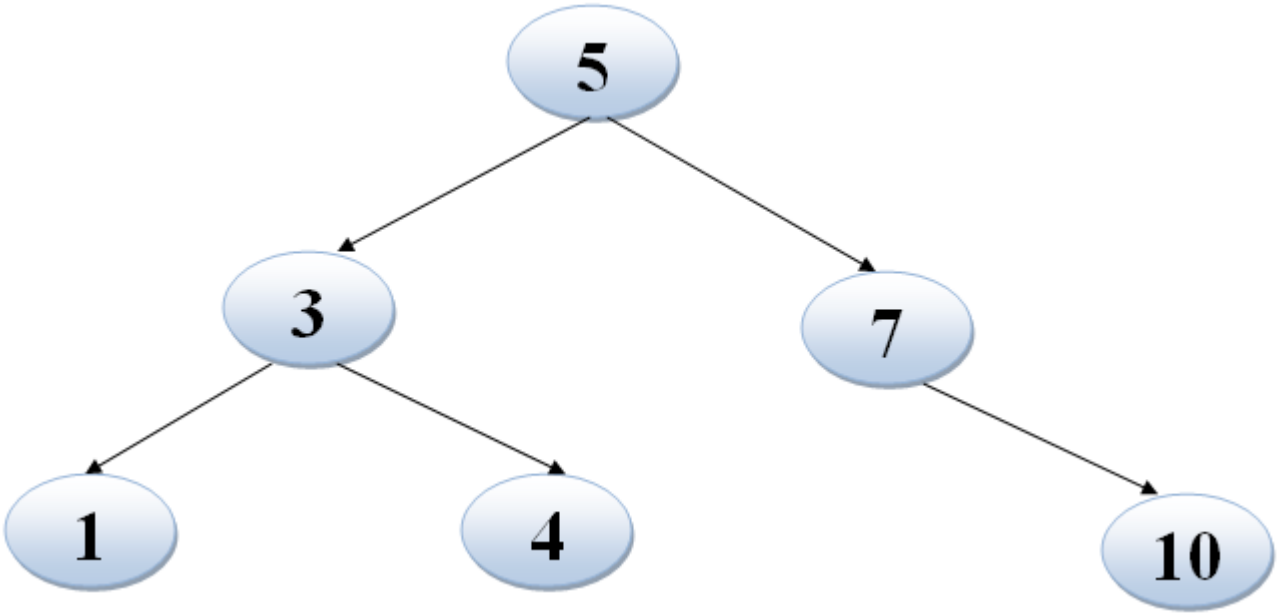
- Un arbore în care operaţiile de căutare sunt foarte rapide se numeşte **arbore de căutare**.
- Arborii de căutare sunt folosiţi pentru a memora o mulţime finită de chei alese dintr-o mulțime total ordonată de chei.
- Fiecare nod din arborii de căutare conţine una sau mai multe chei şi toate cheile din arbore sunt unice.
- Nu se memorează chei duplicate.
- Diferența dintre arborii oarecare şi arborii de căutare este că în arborii de căutare cheile nu sunt memorate în noduri arbitrare din arbore și există un criteriu de ordine care determină unde trebuie memorată o cheie ce se află în relaţie cu alte chei din arbore.

Arbori

- **Arborii binari de căutare** sunt arbori binari în care nodurile sunt memorate ordonat în funcţie de o cheie.
- Toate nodurile din arbore au în subarborele stâng noduri care au chei mai mici şi în subarborele drept chei mai mari.

Arbori

- arbore binar de căutare:



Arbori

- Un arbore binar de căutare are proprietatea că prin parcurgerea în inordine (stânga–rădăcină–dreapta) a nodurilor se obţine o secvenţă monoton crescătoare a cheilor.
- Câmpul cheie este singurul care prezintă interes din punct de vedere al operaţiilor care se pot efectua asupra arborilor de căutare.

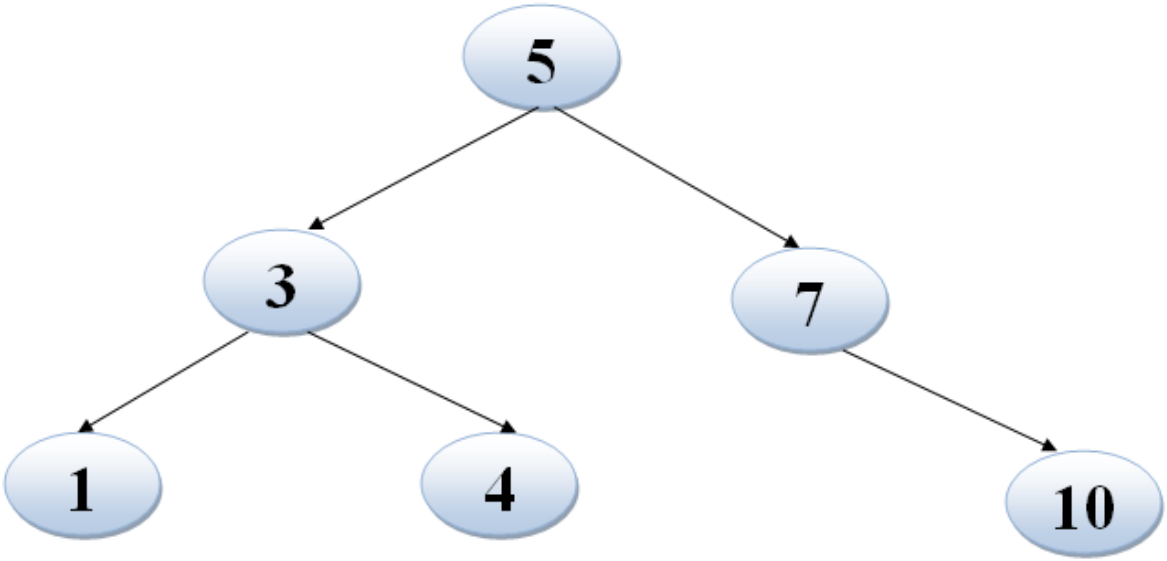
Arbori

Operații pe arbori binari de căutare:

- inserarea unui nod;
- ștergerea unui nod;
- traversarea arborelui: preordine, inordine, postordine, pe niveluri;
- calcul înălțime arbore;
- echilibrare arbore;
- calcul număr noduri frunză;
- etc.

Arbori

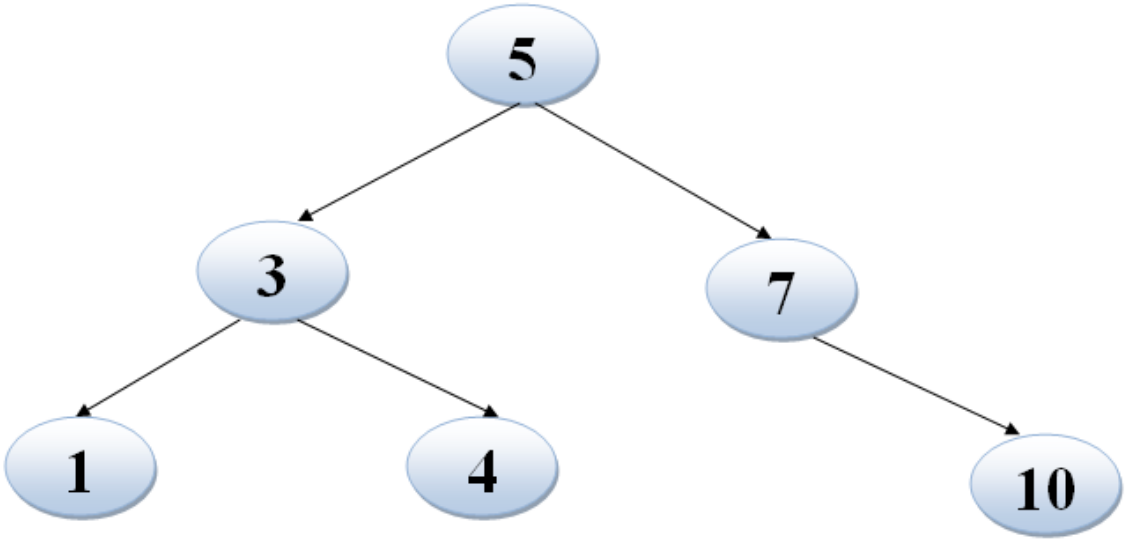
- Traversare arbore binar de căutare:
  - **preordine (RSD)**: traversarea se face prin rădăcina arborelui, apoi se traversează subarborele stâng, iar apoi subarborele drept;



$$5 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 10$$

Arbori

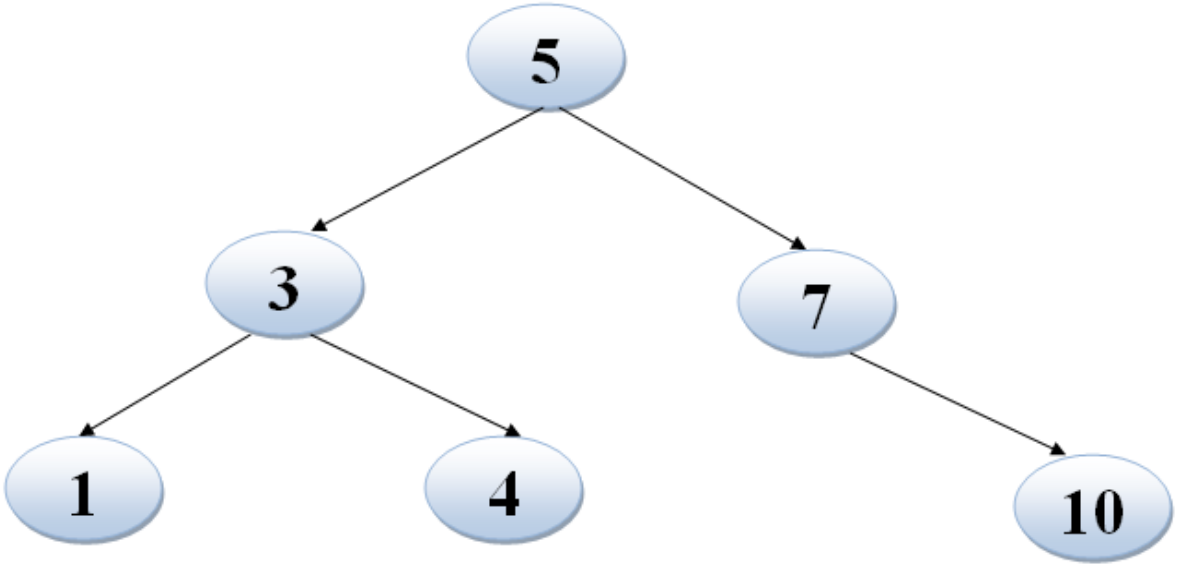
- Traversare arbore binar de căutare:
  - **inordine (SRD)**: traversarea se face începând cu subarborele stâng, apoi prin rădăcină, iar apoi se traversează subarborele drept;



$$1 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 10$$

Arbori

- Traversare arbore binar de căutare:
  - **postordine (SDR)**: traversarea se face începând cu subarborele stâng, apoi se traversează subarborele drept, iar apoi rădăcina;



$$1 \rightarrow 4 \rightarrow 3 \rightarrow 10 \rightarrow 7 \rightarrow 5$$

Bibliografie

- Ion Ivan, Marius Popa, Paul Pocatilu (coordonatori) – *Structuri de date*, Editura ASE, Bucureşti, 2008.
  - Cap. 12. Arbori binari şi arbori de căutare