Analiza algoritmilor
Arbori generalizaţi

conf. dr. ing. Ciprian-Bogdan Chirila
Departamentul de Calculatoare si Tehnologia Informatiei
Universitatea Politehnica Timisoara

Cuprins

- Definitii
- Parcurgeri
  - preordine, inordine, postordine
- Implementari
  - indicator spre parinte
  - prim fiu frate drept
  - multilista
- Aplicatie

Definitii (1)

- **Arborele generalizat** este o multime de n>=0 noduri de acelasi tip,
- care poate fi vida (arbore vid) sau
- formata dintr-un nod numit radacina,
- restul nodurilor formând un numar finit de arbori (numiti **subarbori**),
- doi câte doi disjuncti.

Definitii (2)

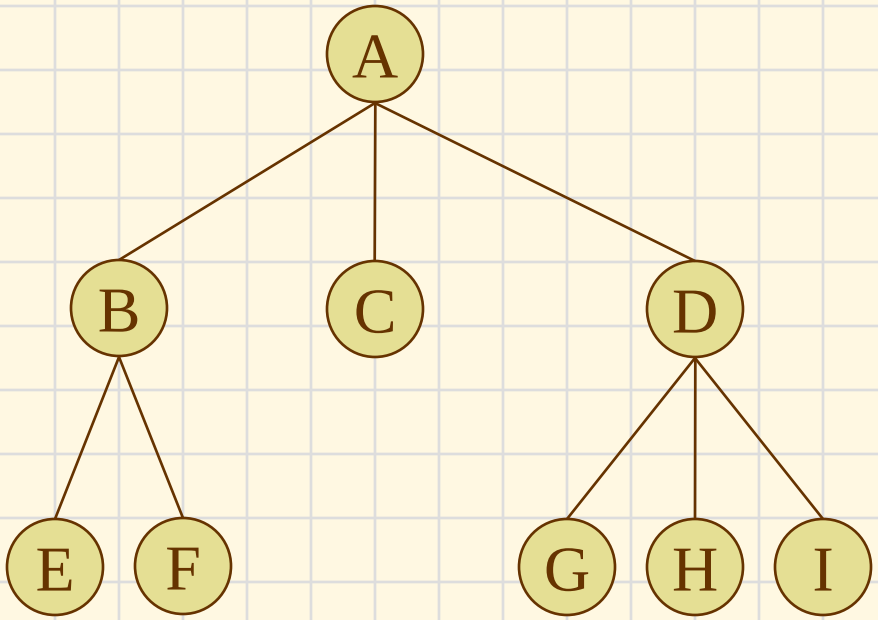
- **Înaltimea unui nod** este lungimea celui mai lung drum de la nodul respectiv la unul terminal.
- **Adâncimea unui nod** e lungimea drumului de la radacina la nodul respectiv.
- **Gradul** unui nod este numarul fiilor unui nod.
- **Gradul arborelui este** gradul maxim al nodurilor acelui arbore.

Exemplu de arbore

nivel 1

nivel 2

nivel 3

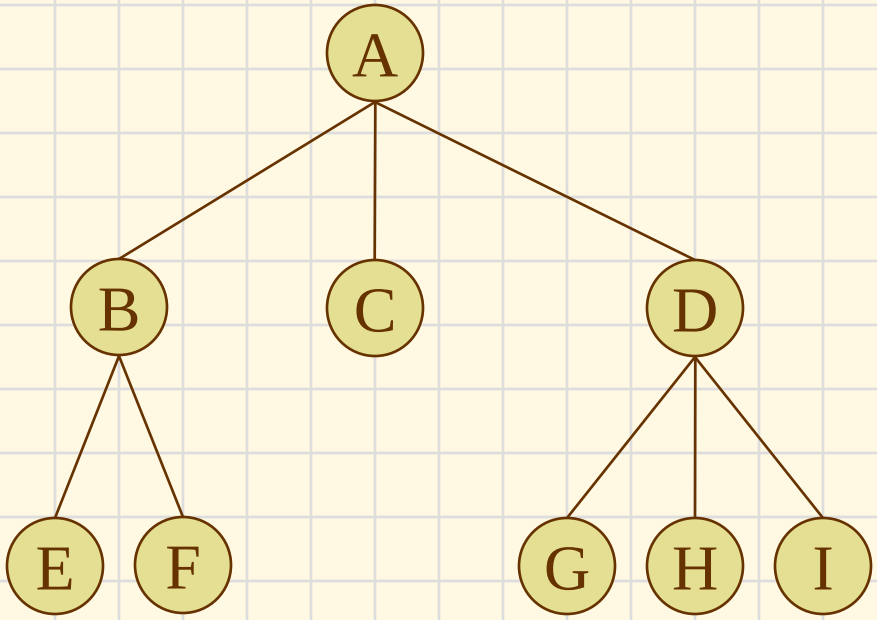


gradul 3

Parcurgerea arborilor

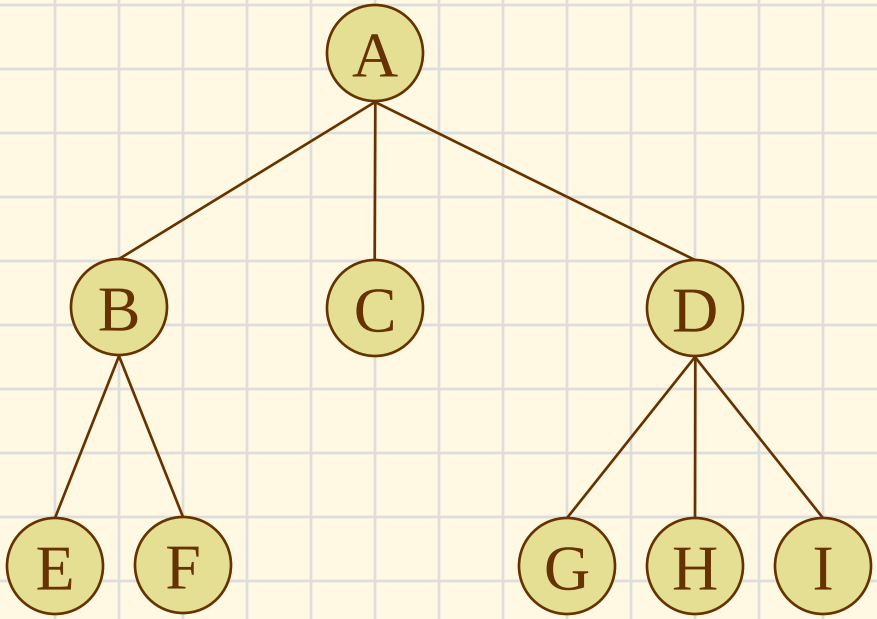
- preordine
  - R, A1, A2, …, An
- inordine
  - A1, R, A2, …, An
- postordine
  - A1, A2, …, An, R

Exemplu de parcurgere



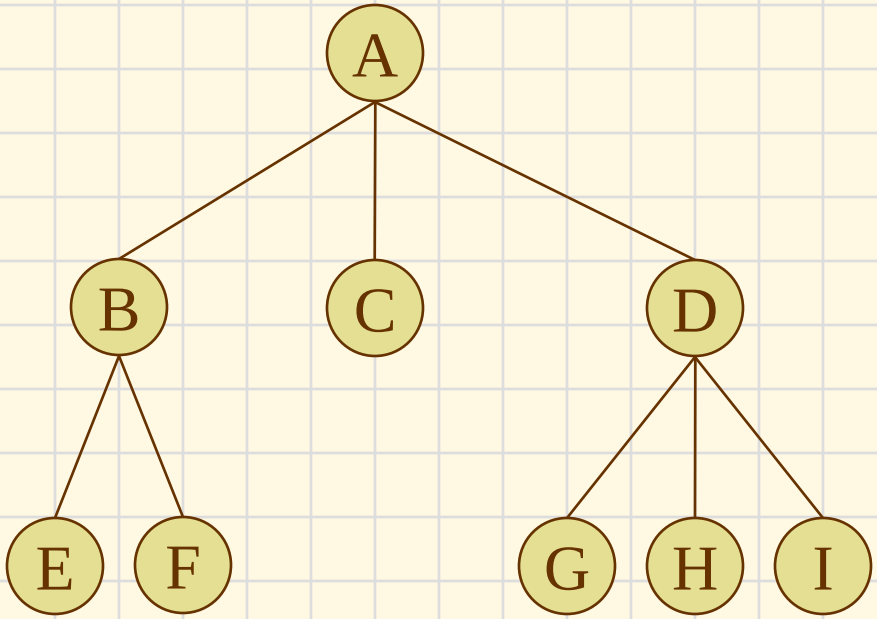
- Preordine: A, B, E, F, C, D, G, H, I
- Inordine: E, B, F, A, C, G, D, H, I
- Postordine: E, F, B, C, G, H, I, D, A

Implementare: Indicator spre parinte



	A	B	C	D	E	F	G	H	I
indice nod:	0	1	2	3	4	5	6	7	8
indice parinte:	-1	0	0	0	1	1	3	3	3

Implementare: Prim fiu, frate drept



	A	B	C	D	E	F	G	H	I
indice nod:	0	1	2	3	4	5	6	7	8
indice prim fiu:	1	4	-1	6	-1	-1	-1	-1	-1
indice frate drept:	-1	2	3	-1	5	-1	7	8	-1

Implementare cu multilista

```
typedeft struct nod
{
  char cheie;
  struct nod *parinte;
  struct nod *prim_fiu;
  struct nod *frate_drept;
} NOD;
```

Aplicatie

- void insereaza_radacina(int cheie_radacina)
  - distruge arborele generalizat daca acesta exista si creaza unul nou avand "cheie_radacina" pe post de radacina;
- void insereaza_cheie(int cheie , int cheie_parinte)
  - insereaza in arbore cheia "cheie" ca si cheie fiu a cheii "cheie_parinte";
- int cauta_cheie (int cheie)
  - cauta cheia "cheie" in arbore; returneaza 1 in caz ca a fost gasita cheia, sau 0 in caz contrar;
- int sterge_cheie(int cheie)
  - cauta cheia "cheie" in arbore si o sterge; returneaza 1 in caz ca a fost gasita si stearsa cheia, sau 0 in caz contrar;

Aplicatie

- void preordine()
  - parcurge arborele in preordine, afisand cheile intalnite in aceasta parcurgere;
- void inordine()
  - parcurge arborele in inordine, afisand cheile intalnite in aceasta parcurgere;
- void postordine()
  - parcurge arborele in postordine, afisand cheile intalnite in aceasta parcurgere;
- int inaltime_arbore()
  - returneaza inaltimea arborelui generalizat;
- int grad_arbore()
  - returneaza gradul arborelui generalizat;

Aplicatie

- int cel_mai_din_stanga_frate(int cheie)
  - returneaza cheia celui mai din stanga frate al cheii "cheie" sau -1 daca acesta nu exista;
- int cel_mai_din_dreapta_frate(int cheie)
  - returneaza cheia celui mai din dreapta frate al cheii "cheie" sau -1 daca acesta nu exista;
- int numar_frati (int cheie)
  - returneaza numarul de frati ai cheii "cheie"