

Structuri de date – Curs 11

Prof. univ. dr. Cristian CIUREA
Departamentul de Informatică și Cibernetică Economică
Academia de Studii Economice din București
cristian.ciurea@ie.ase.ro

Agenda

- ▶ Grafuri – definire și utilizare
- ▶ Complexitatea McCabe
- ▶ Clasificarea grafurilor
- ▶ Metode de reprezentare
- ▶ Metode de traversare

Grafuri

- ▶ Graful, asemenea arborelui, este o structură de date în care relația dintre nodul părinte și nodul fiu este una ierarhică, dar care este mai puțin restrictivă, în sensul că un nod are mai mulți succesori, dar și mai mulți predecesori.

Grafuri

- ▶ Graful este definit ca o colecție de date reunite în două mulțimi:
 - multimea $N = \{ N_1, N_2, \dots, N_n \mid n - \text{numarul de noduri ale grafului} \}$, ce conține toate nodurile grafului;
 - multimea $A = \{ (N_i, N_j) = A_{ij} \mid N_i, N_j \subset N \text{ si } i, j = 1, n \text{ cu } i \neq j \}$ care conține arcele dintre două noduri vecine.

Grafuri

- ▶ Graful este o pereche ordonată de mulțimi, notată $G=(N, A)$, unde:
 - N este o mulțime finită și nevidă de elemente numite **noduri** sau **vârfuri**;
 - A este o mulțime de perechi (ordonate sau neordonate) de elemente din N , numite **muchii** (dacă sunt perechi neordonate) sau **arce** (dacă sunt perechi ordonate).

Grafuri

- ▶ Utilizare structură de tip graf:
 - informații care au multiple legături între ele;
 - parcurgerea completă a elementelor structurii;
 - localizarea unui element din structură;
 - oferirea de soluții optime de minimizare a costurilor.

Grafuri

- ▶ Graful este larg utilizat în domeniile:
 - ciberneticii;
 - matematicii;
 - cercetărilor operaționale, în vederea optimizării diferitelor activități economice;
 - chimiei, pentru descrierea structurii cristalelor;
 - rețelelor de transport de toate tipurile, pentru optimizarea traseelor;
 - circuitelor electrice, pentru simularea funcționării corecte;
 - inteligenței artificiale;
 - analizei aplicațiilor software.

Grafuri

- ▶ Complexitatea în sens McCabe presupune asocierea fiecărui program a unui graf, în care nodurile corespund instrucțiunilor, iar arcele marchează sensul trecerii execuției de la o instrucțiune la alta;
- ▶ Complexitatea în sens McCabe:

$$CC = n_{arc} - n_{nod} + 2$$

unde:

n_{arc} – numărul de arce;

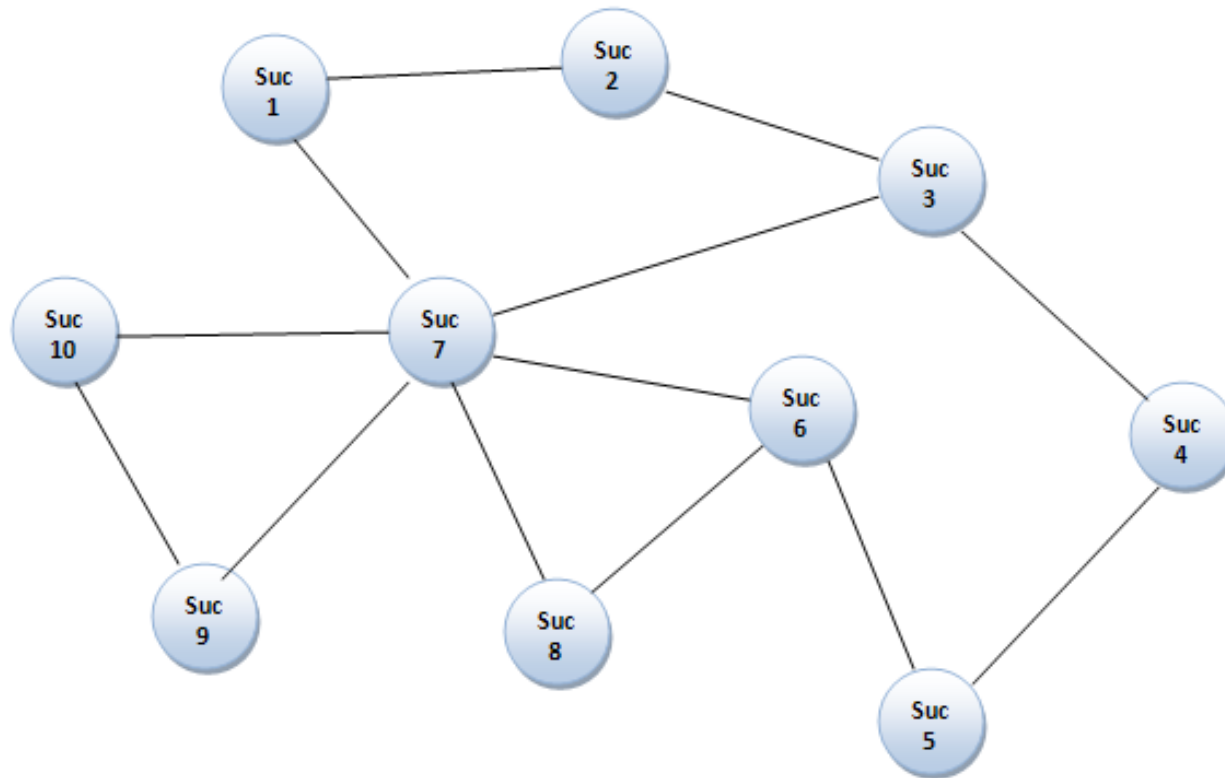
n_{nod} – numărul de noduri.

Grafuri

- ▶ Sistemul colaborativ *bancă* are asociat graful în care nodurile reprezintă sucursalele băncii, iar arcele sunt legăturile dintre sucursale, reprezentate prin clienții care au conturi deschise la mai multe agenții.

Grafuri

- ▶ Graful asociat sistemului colaborativ *bancă*:
 - nodurile reprezentate de orașele din țară în care banca deține sucursale;
 - arcele leagă sucursalele ce dețin clienți comuni.



Grafuri

Structura de tip graf:

- ▶ relație ierarhică între nodul părinte și nodul fiu;
- ▶ relație mai puțin restrictivă: un nod are mai mulți succesori, dar și mai mulți predecesori;
- ▶ colecție de date formată din două mulțimi:
 - mulțimea nodurilor grafului;
 - mulțimea arcelor dintre două noduri vecine.

Grafuri

Criterii de clasificare a grafurilor:

▶ **direcția arcelor:**

- grafuri neorientate (arce nedirecționate);
- grafuri orientate (există sens între două noduri);

▶ **greutatea arcelor:**

- grafuri cu greutate (arce cu valoare numerică);
- grafuri fără greutate (arcele nu au asociate valori numerice);

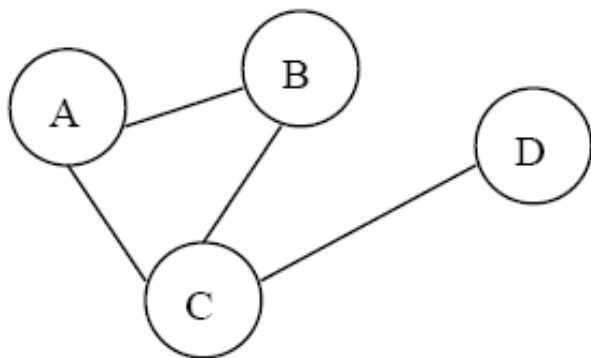
▶ **existența arcelor:**

- grafuri conectate (nu există nici un nod izolat);
- grafuri neconectate (există cel puțin un nod izolat).

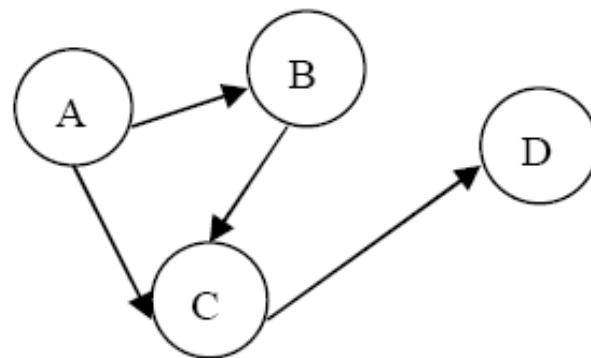
Grafuri

► direcția arcelor:

- în cazul în care arcele dintre nodurile grafului sunt nedirecționate, atunci graful este unul **neorientat**; când există sens între două noduri N_i , N_j și arcul este direcționat, atunci graful este unul **orientat**;



Graf neorientat

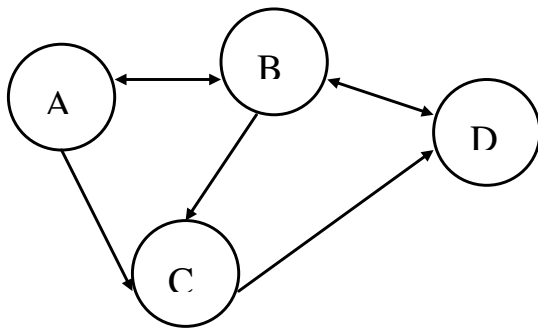


Graf orientat

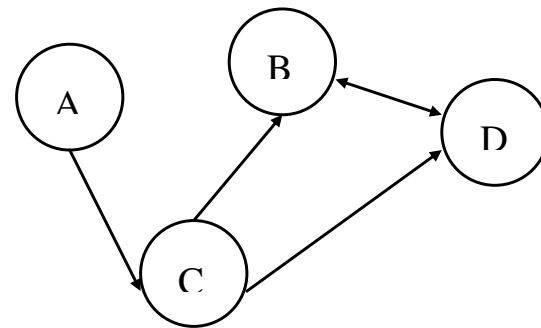
Grafuri

► direcția arcelor:

- graful orientat este **puternic conectat** dacă între oricare două noduri N_i și N_j cu $i, j = 1..n$ există drum (un drum este format din unul sau mai multe arce) orientat de la i la j , $N_i \rightarrow N_j$;
- graful orientat este **slab conectat** dacă între oricare două noduri N_i și N_j cu $i, j = 1..n$ există drum orientat de la i la j , $N_i \rightarrow N_j$ sau de la j la i , $N_i \leftarrow N_j$ (doar unul dintre ele).



Graf orientat puternic conectat

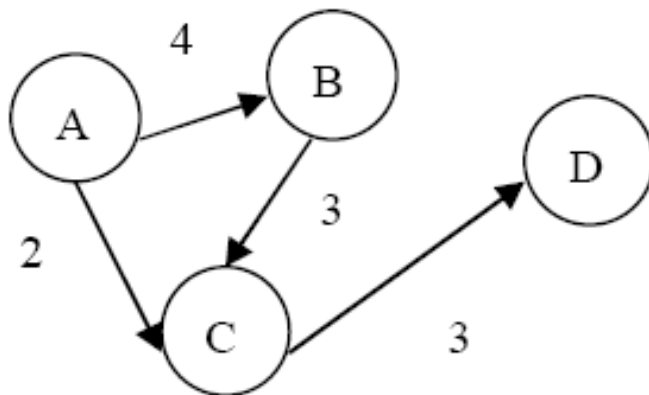


Graf orientat slab conectat

Grafuri

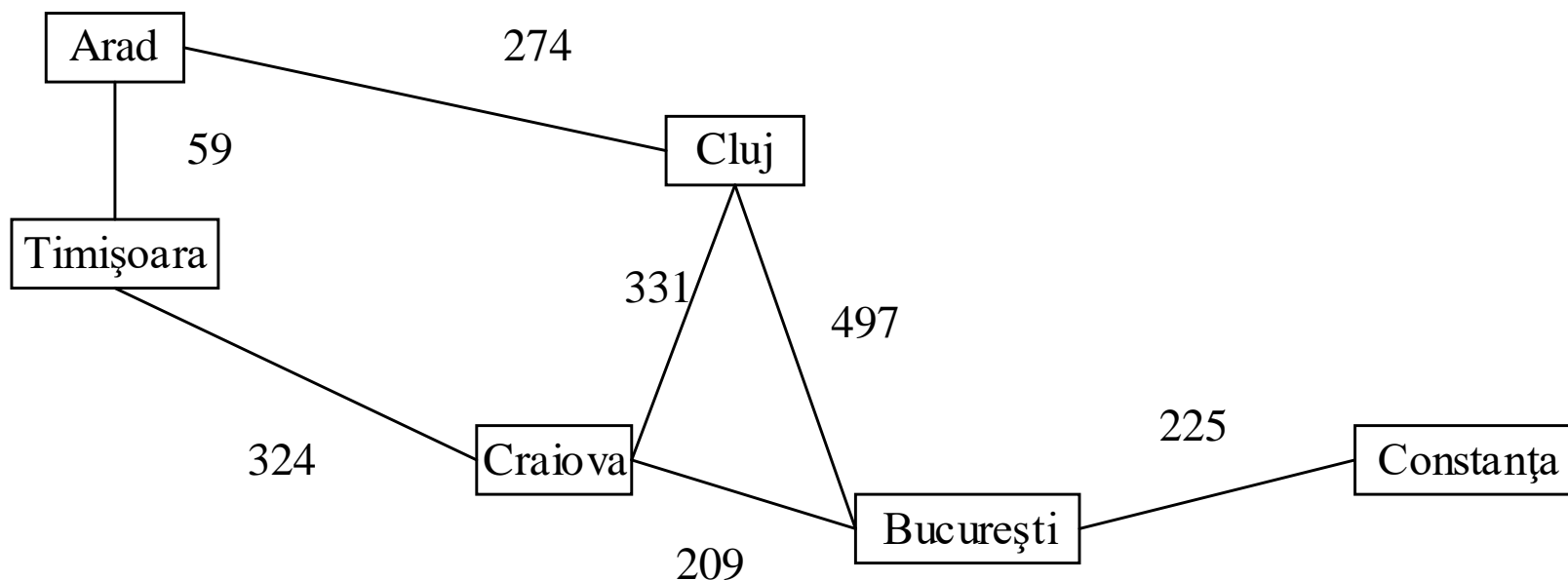
► greutatea arcelor:

- dacă oricare arc dintre două noduri ale grafului are asociată o valoare numerică (care reprezintă de cele mai multe ori distanța, durata de timp sau costul), atunci graful este **cu greutate**; în cazul în care arcele nu au asociate valori numerice, graful este unul **fără greutate**.



Grafuri

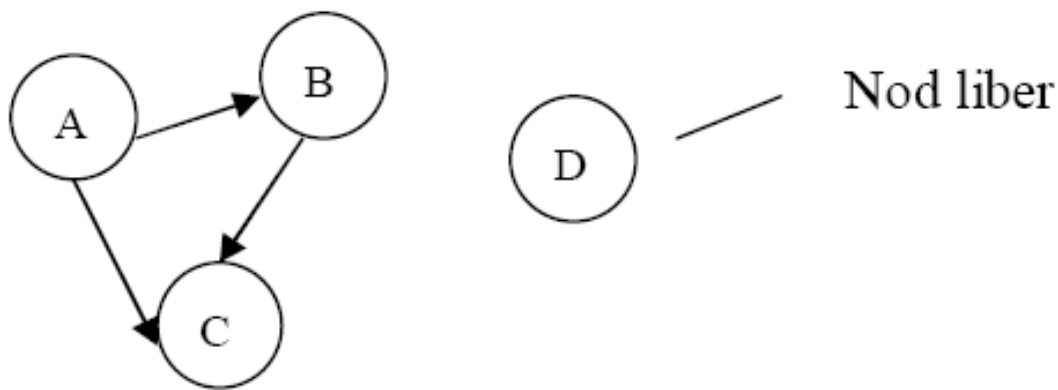
- ▶ $N = \{\text{Arad, București, Cluj, Craiova, Constanța, Timișoara}\}$
- ▶ $A = \{\text{Arad - Timișoara} = 59 \text{ km, Timișoara - Craiova} = 324 \text{ km, Craiova - București} = 209 \text{ km, București - Constanța} = 225 \text{ km, Arad - Cluj} = 274 \text{ km, Cluj - Craiova} = 331 \text{ km, Cluj - București} = 497 \text{ km}\}$



Grafuri

► existența arcelor:

- dacă într-un graf nu există nici un nod izolat, altfel spus pentru oricare nod N_i cu $i=1..n$ exista cel puțin un nod N_j cu $1 \leq j \leq n$ și $i \neq j$ pentru care există arcul A_{ij} asociat, atunci graful este **conectat**; un graf este **neconectat** dacă există cel puțin un nod izolat.



Grafuri

Metode de reprezentare a grafului:

- ▶ matrice de adiacență;
- ▶ liste de adiacență;
- ▶ vector de pointeri la liste simple sau dublu înlănțuite de noduri adiacente;
- ▶ listă simplu sau dublu înlănțuită de pointeri la liste simple sau dublu înlănțuite de noduri adiacente;
- ▶ vector de pointeri la liste simple sau dublu înlănțuite de arce.

Grafuri

- ▶ Reprezentarea prin matrice de adiacență – eficientă:
 - se cunoaște numărul nodurilor;
 - se cunoaște numărul mediu al arcelor – grad de umplere al matricei;
 - matrice pătratică;
 - reprezentare arce: valoarea 1 (graf fără greutate), greutate arc (graf cu greutate).

Grafuri

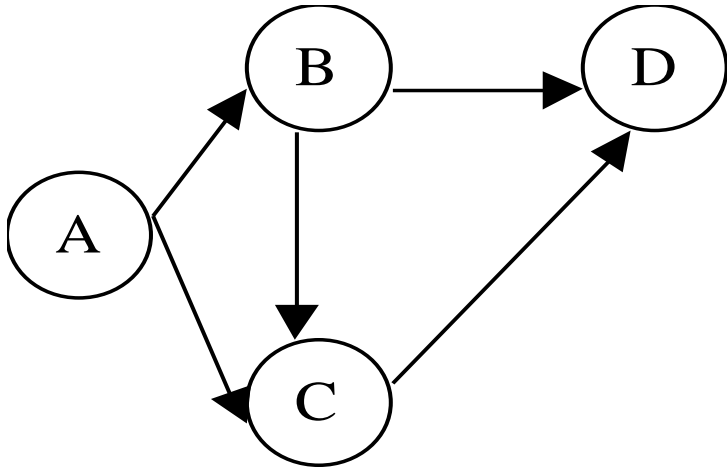
- ▶ Inițial matricea de adiacență are toate elementele egale cu valoarea 0.
- ▶ Pentru a reprezenta arcul A_{ij} dintre nodurile N_i și N_j (orientat de la N_i la N_j) la intersecția liniei i cu coloana j se trece:
 - valoarea 1, în cazul grafului fără greutate, sau
 - greutatea arcului, pentru graful cu greutate.
- ▶ În cazul unui graf neorientat, matricea de adiacență asociată este simetrică.

Grafuri

- ▶ Pentru graful fără greutate:
 - $M[i][j] = 1$, dacă există arc între nodul N_i și N_j
 - $M[i][j] = 0$, dacă nu există arc între nodul N_i și N_j
- ▶ Pentru graful cu greutate:
 - $M[i][j] = a_{ij}$, dacă există arc între nodul N_i și N_j , iar a_{ij} reprezintă greutatea arcului
 - $M[i][j] = 0$, dacă nu există arc între nodul N_i și N_j

Grafuri

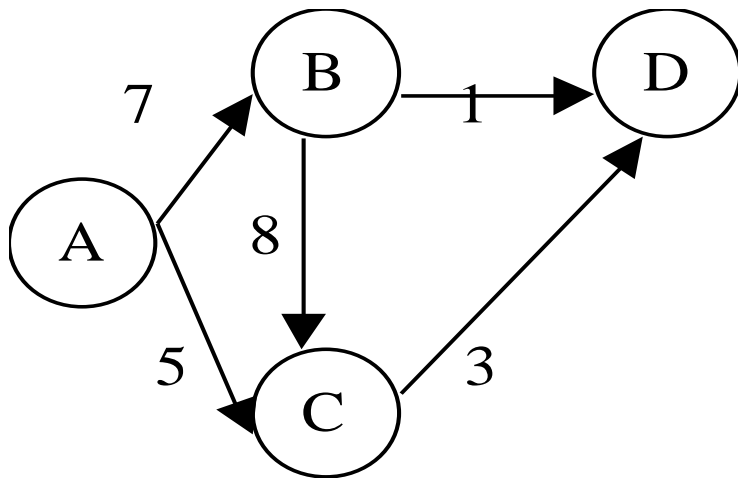
- ▶ Graf orientat fără greutate:



$$M[i][j] = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Grafuri

- ▶ Graf orientat cu greutate:



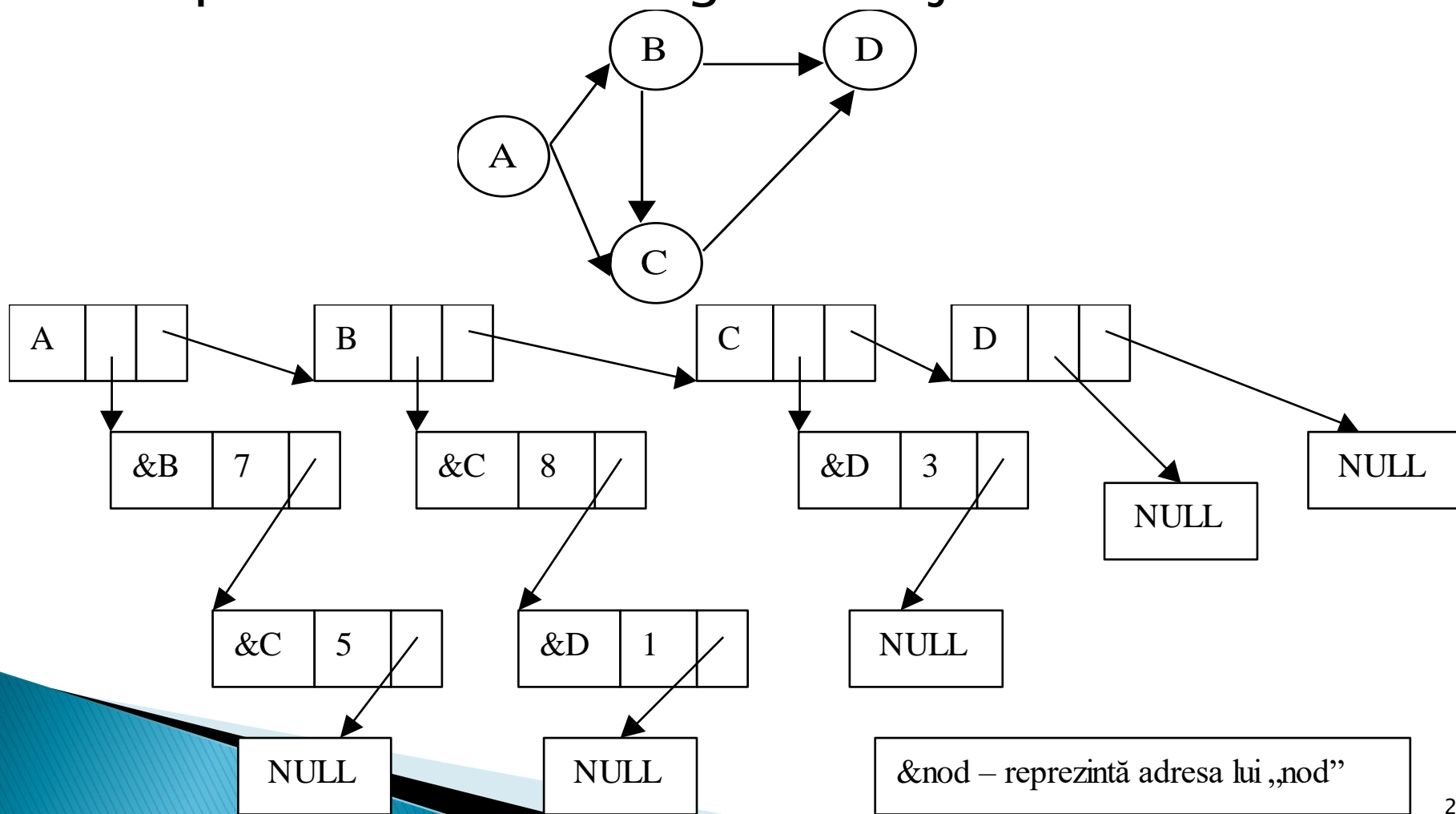
$$M[i][j] = \begin{bmatrix} 0 & 7 & 5 & 0 \\ 0 & 0 & 8 & 1 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Grafuri

- ▶ Reprezentarea prin liste de adiacență:
 - nu se cunoaște numărul de noduri \Rightarrow nu se cunoaște dimensiunea matricei de adiacență;
 - construirea dinamică a structurii de tip graf;
 - rețea de liste înlănțuite;
 - listă a arcelor ce este construită dinamic;
 - descrierea grafului cuprinde mulțimea de noduri și mulțimea de arce, precizând orientarea arcului și greutatea lui.

Grafuri

- ▶ Reprezentarea unui graf cu ajutorul listelor:



Grafuri

Traversarea unui graf:

- ▶ nu este unică;
- ▶ oricare nod al grafului este un posibil punct de start al traversării;
- ▶ se pune problema trecerii o singura dată prin fiecare nod;
- ▶ evitarea revenirii într-un nod vizitat: asocierea unei etichete.

Grafuri

Metode de traversare:

- ▶ *traversarea în adâncime*: Depth – First Traversal;
- ▶ *traversarea în lățime*: Breadth – First Traversal.

Grafuri

Traversarea în adâncime a unui graf:

- ▶ algoritm tip backtracking;
- ▶ algoritm similar cu traversarea în preordine a unui arbore;
- ▶ utilizare structuri de ajutor: vector, listă, stivă;
- ▶ nodurile sunt explorate începând cu ultimul nod selectat.

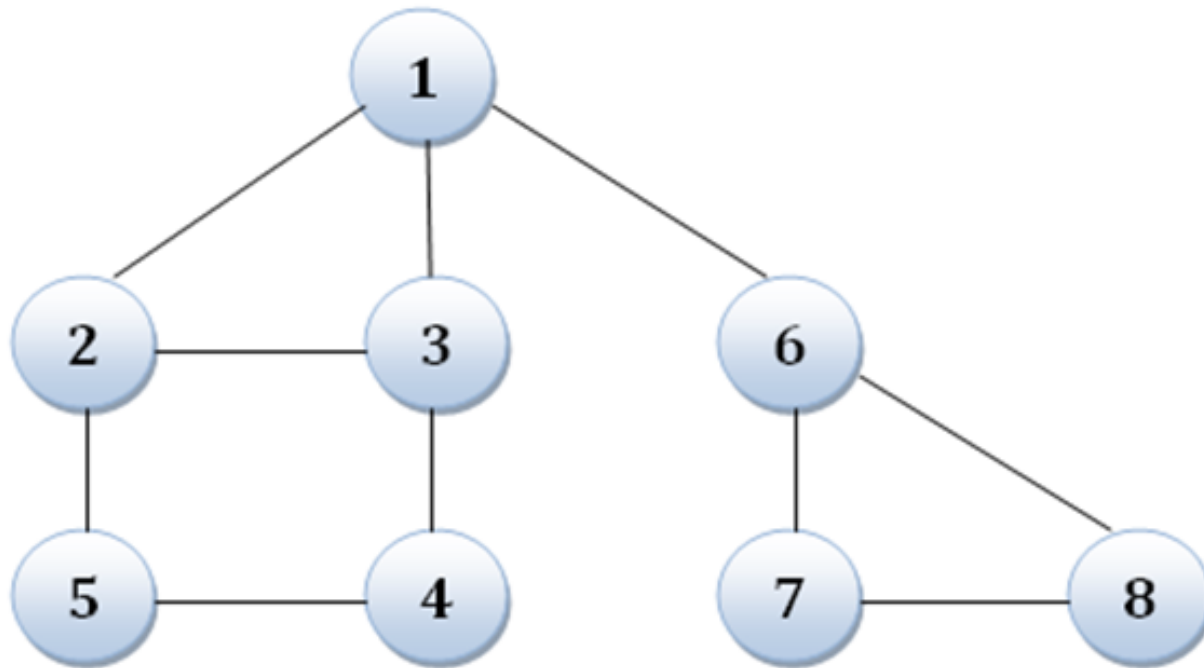
Grafuri

Traversarea în adâncime:

- ▶ se alege un nod oarecare; dacă există un nod adiacent care nu a fost vizitat, atunci el va fi noul punct de plecare;
- ▶ când toate nodurile adiacente au fost marcate, se încheie consultarea începută în nodul inițial;
- ▶ dacă au rămas noduri nevizitate, se alege unul și se reia algoritmul până când toate nodurile sunt marcate.

Grafuri

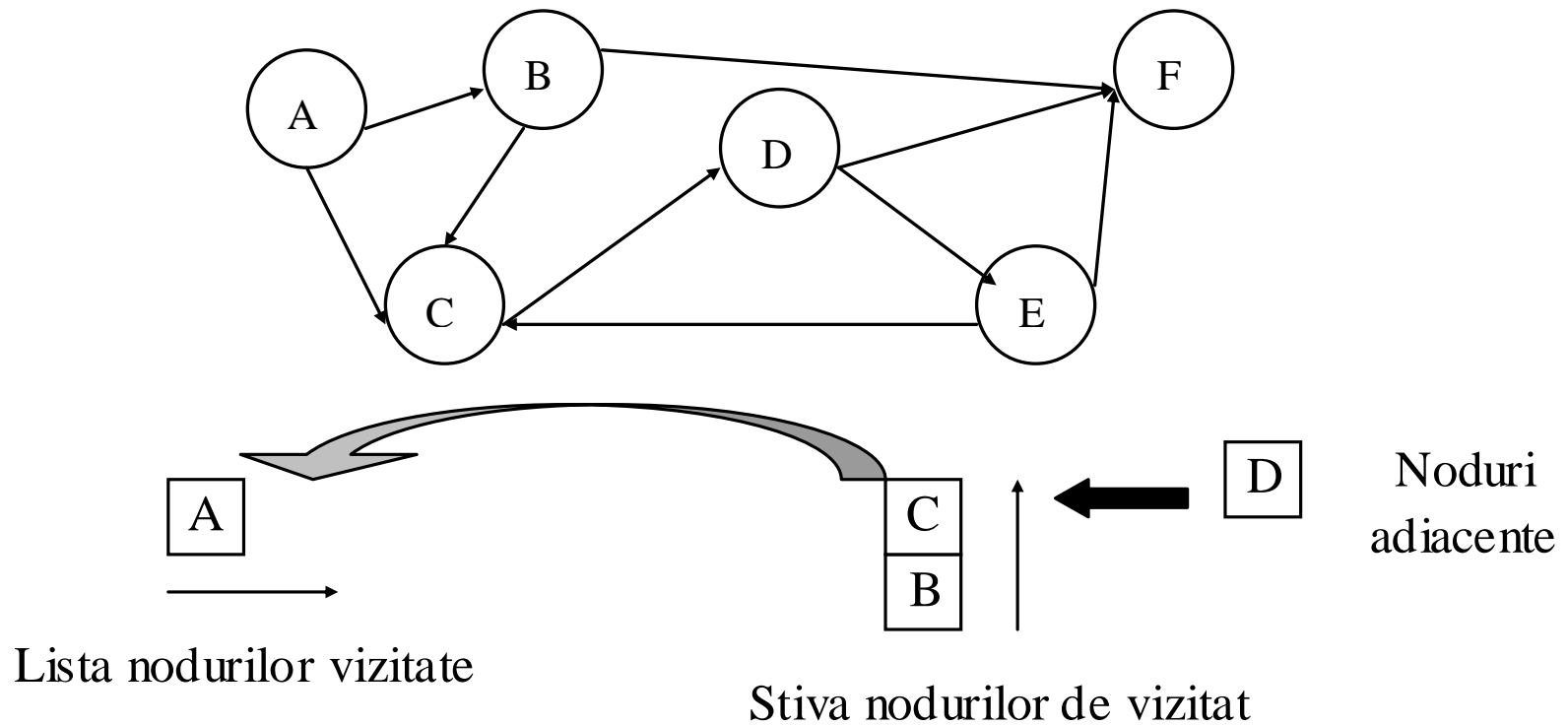
Traversarea în adâncime:



$1 \rightarrow 6 \rightarrow 8 \rightarrow 7 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 2$

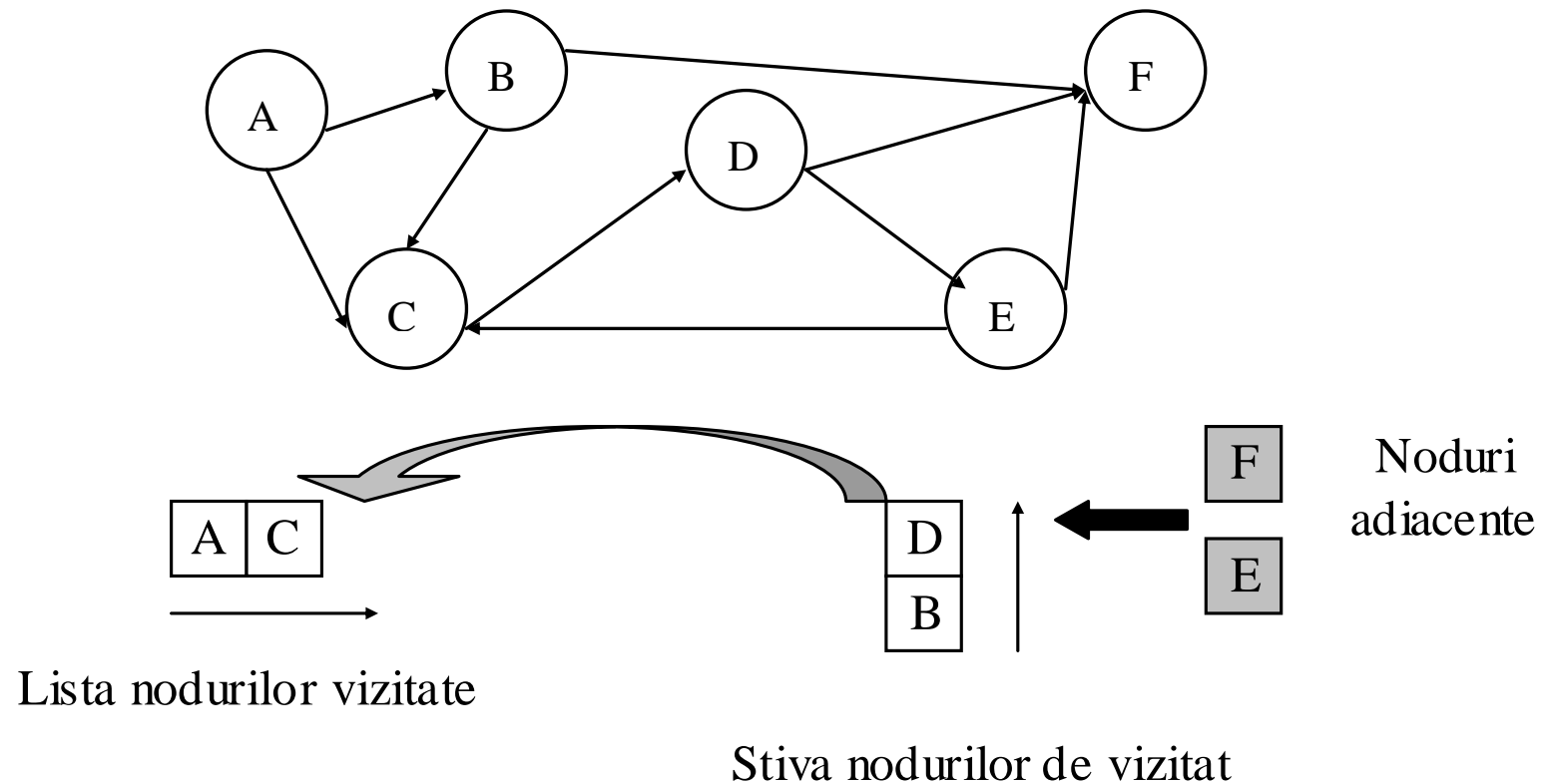
Grafuri

Traversarea în adâncime:



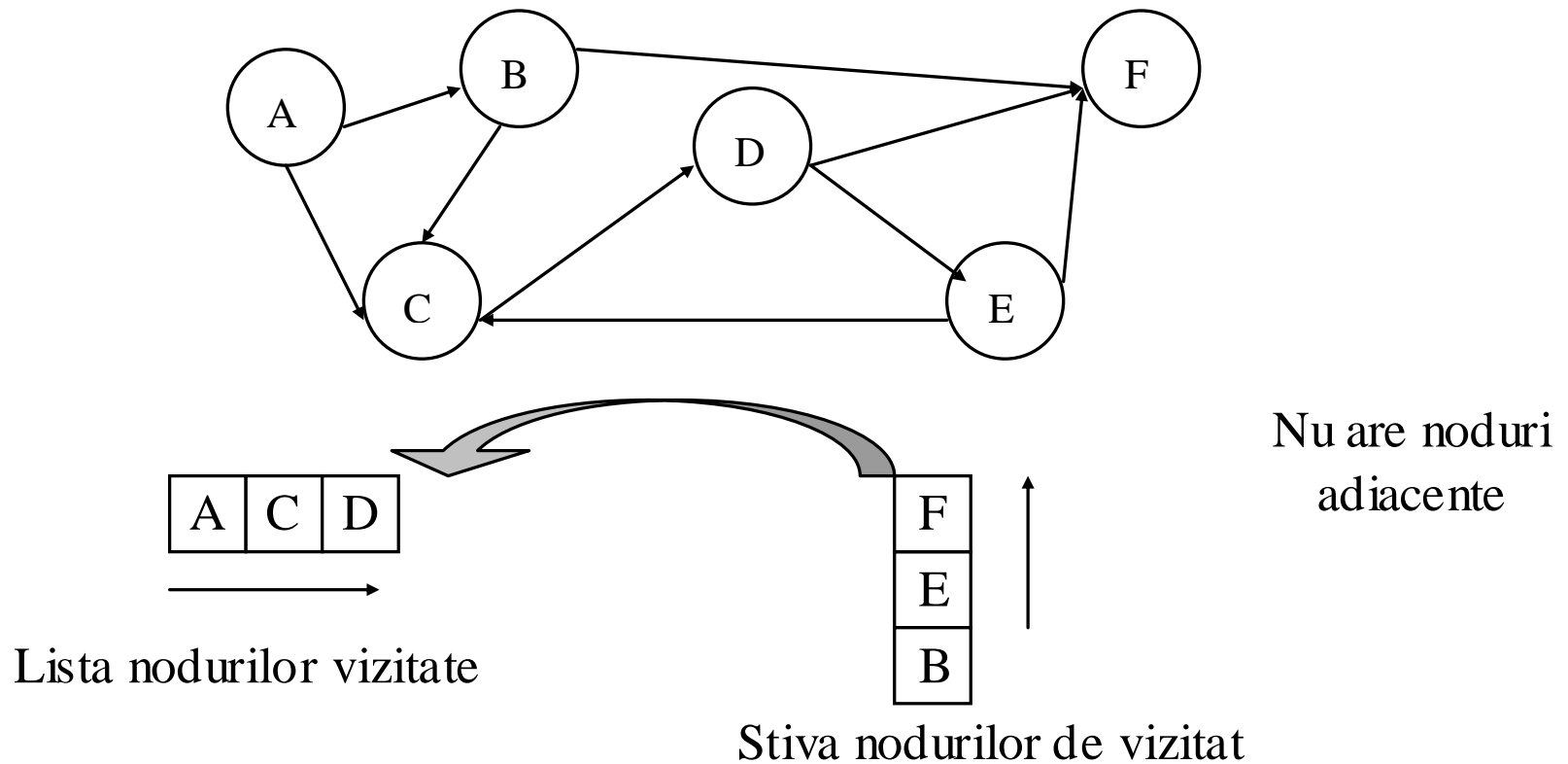
Grafuri

Traversarea în adâncime:



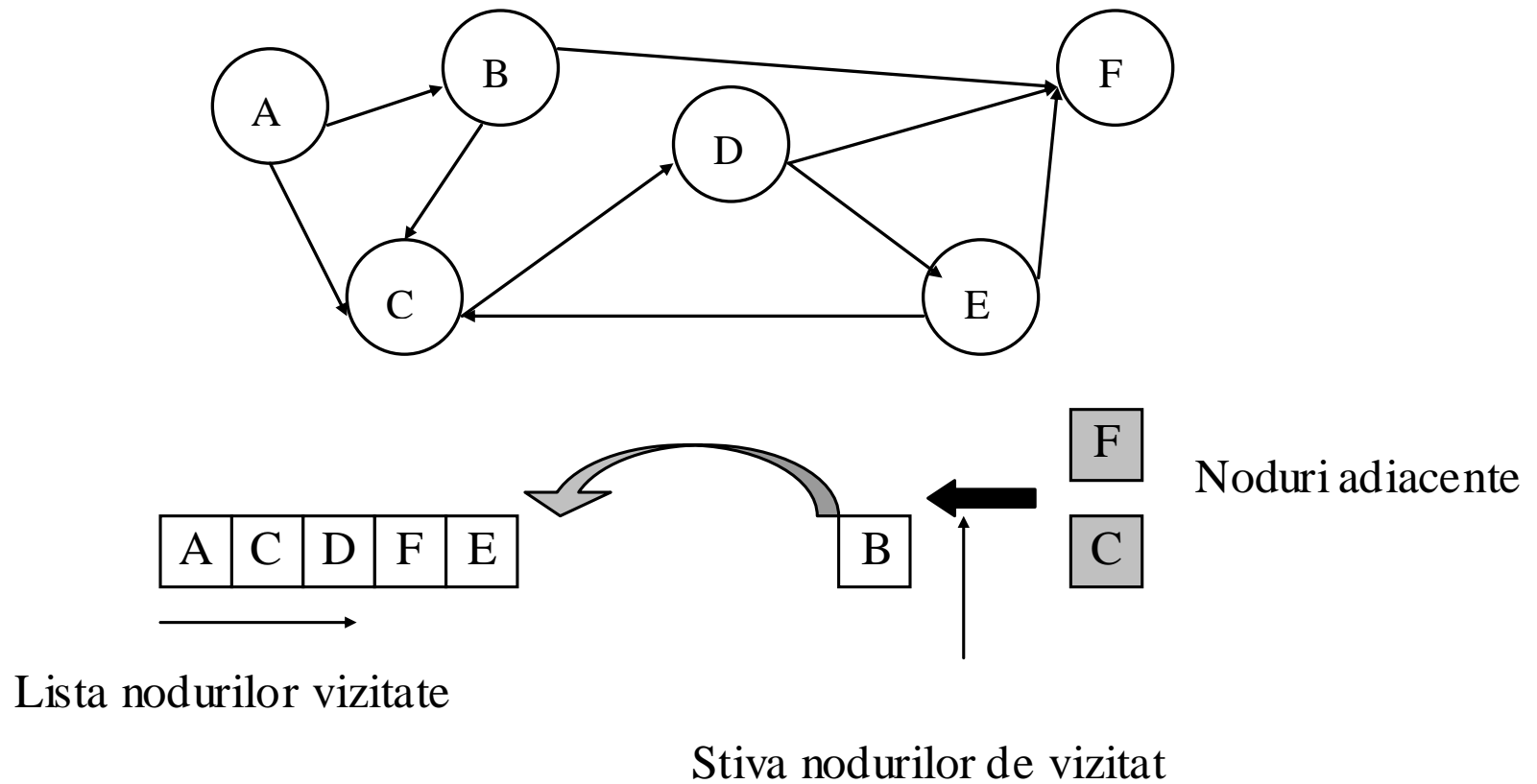
Grafuri

Traversarea în adâncime:



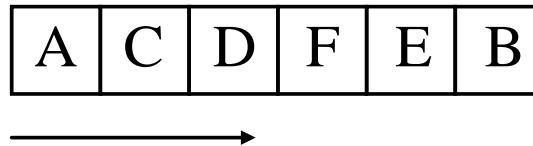
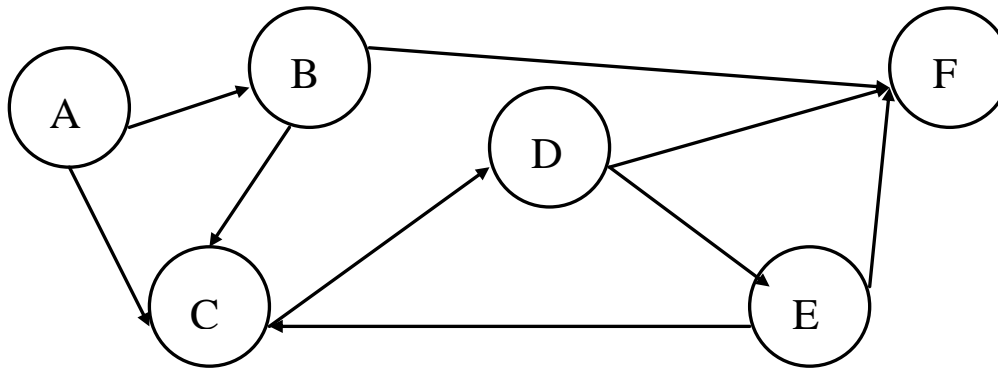
Grafuri

Traversarea în adâncime:



Grafuri

Traversarea în adâncime:



Grafuri

Traversarea în lățime a unui graf:

- ▶ analog procesului de traversare în inordine a unui arbore;
- ▶ folosirea unei structuri de tip coadă pentru nodurile de verificat;
- ▶ algoritmul parcurge sistematic graful pentru a descoperi toate nodurile accesibile din nodul sursă;
- ▶ nodurile se pot afla în una dintre cele trei stări: nevizitat, vizitat sau procesat.

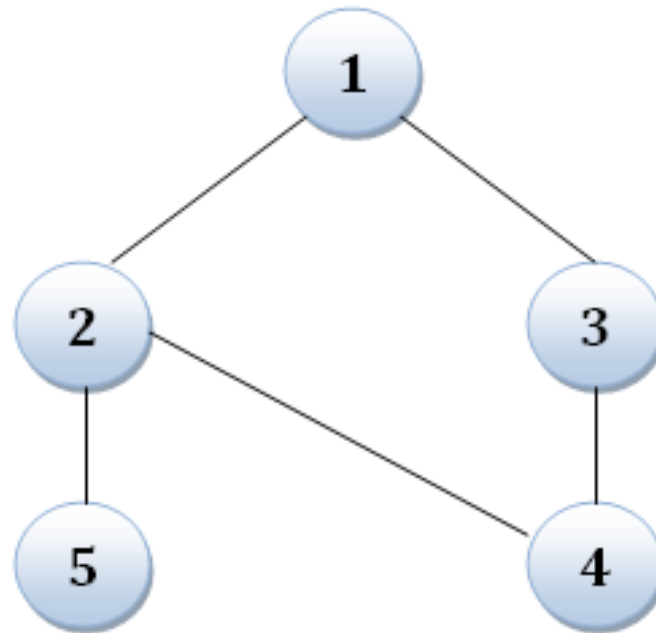
Grafuri

Traversarea în lăţime:

- ▶ se vizitează nodul iniţial, apoi vecinii acestuia, apoi vecinii nevizitaţi ai acestora şi așa mai departe.

Grafuri

Traversarea în lățime:



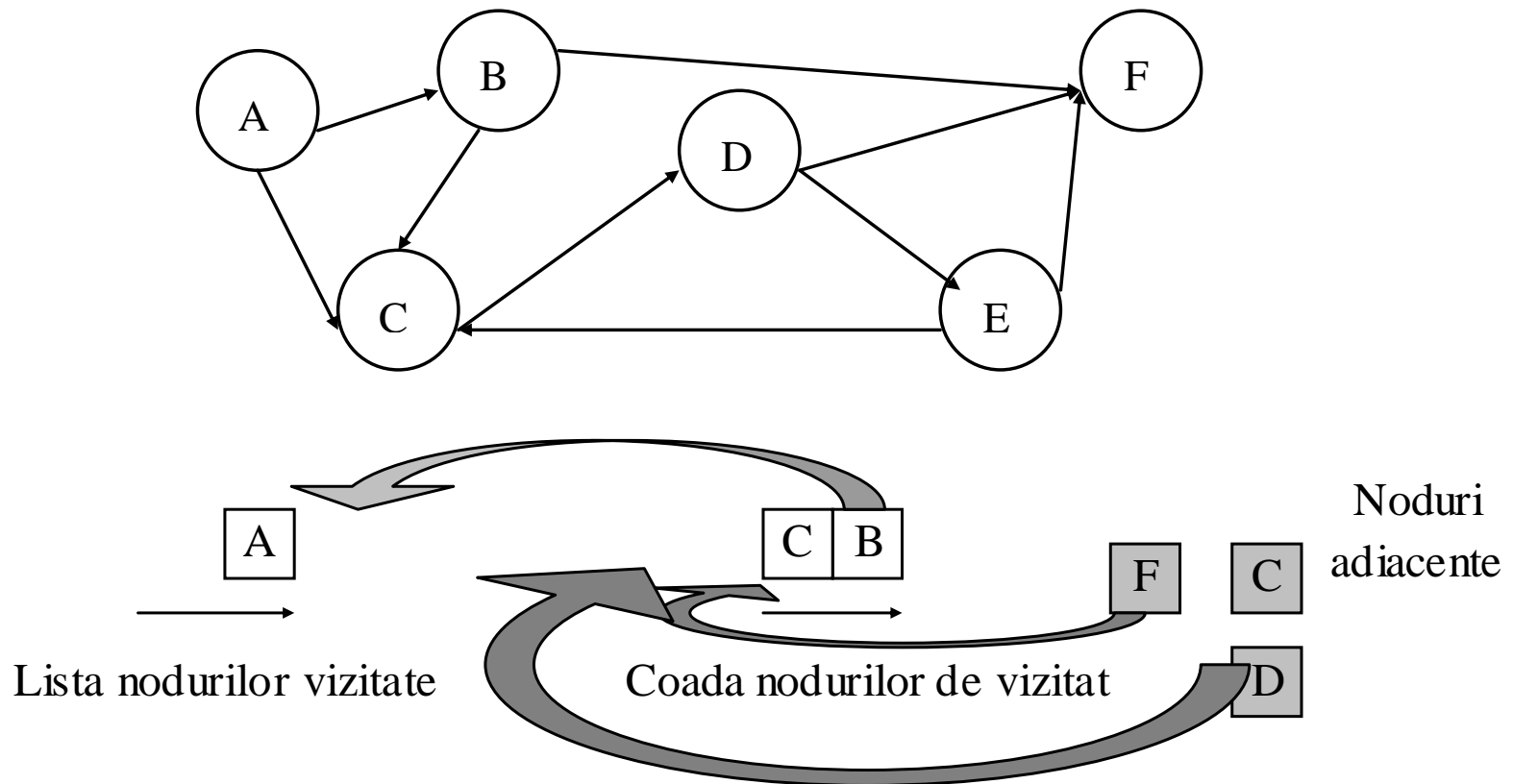
$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

$5 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 3$

$2 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 3$

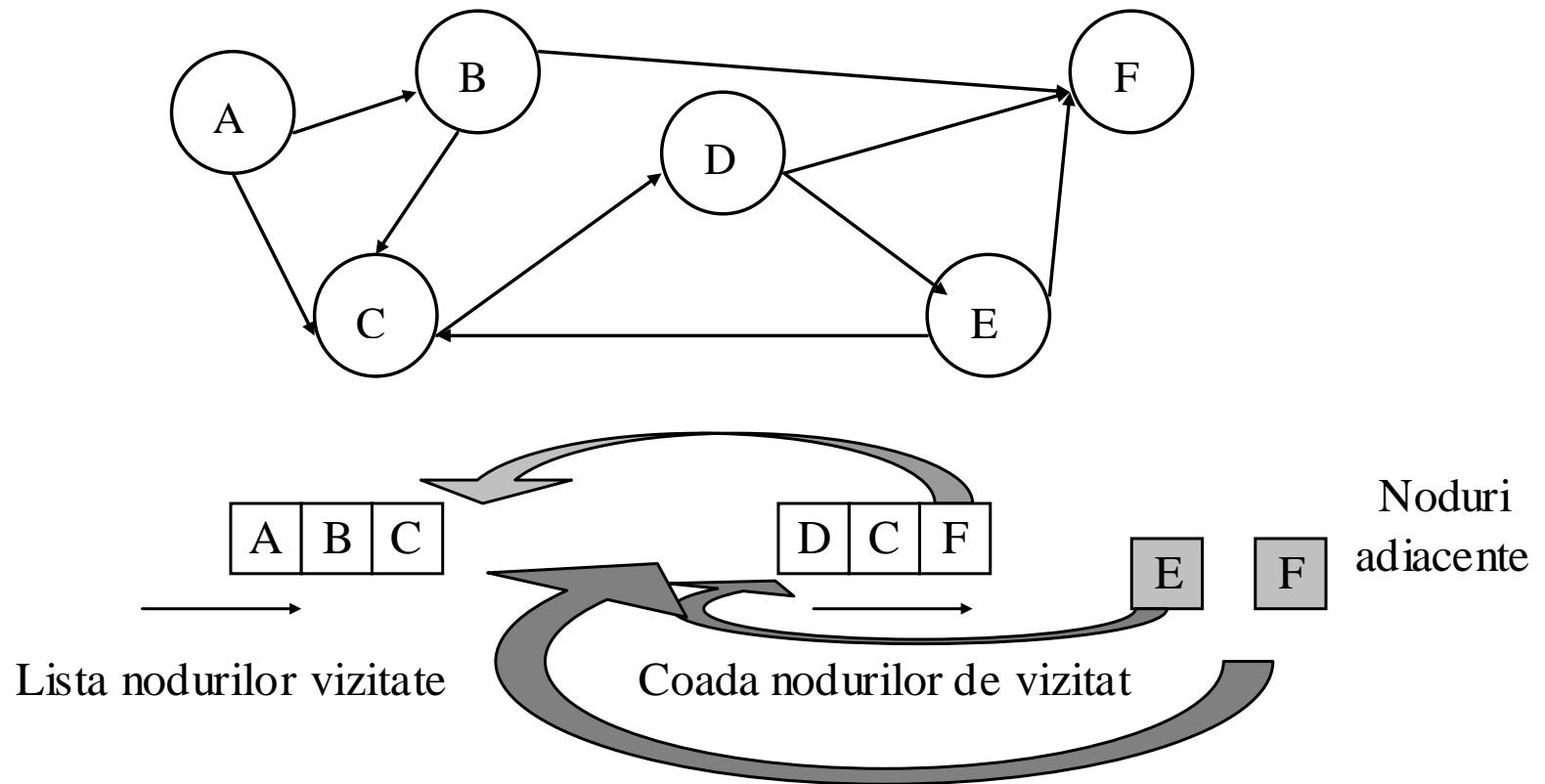
Grafuri

Traversarea în lățime:



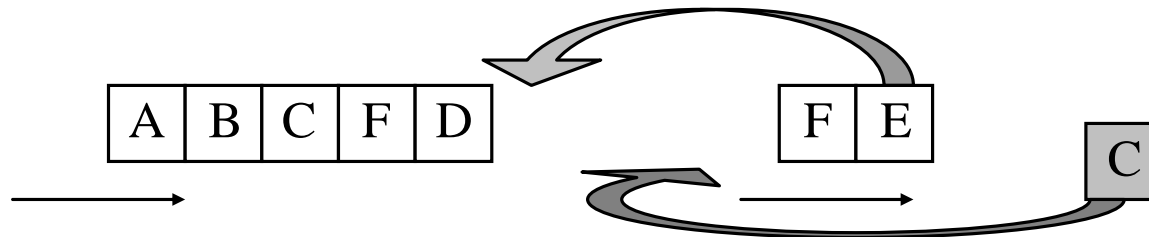
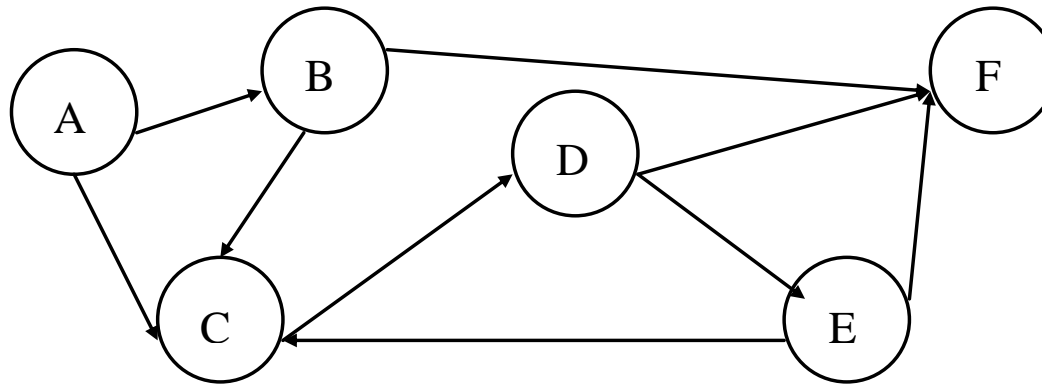
Grafuri

Traversarea în lățime:



Grafuri

Traversarea în lățime:



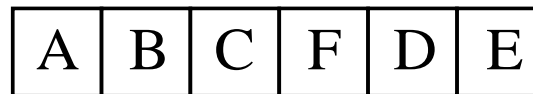
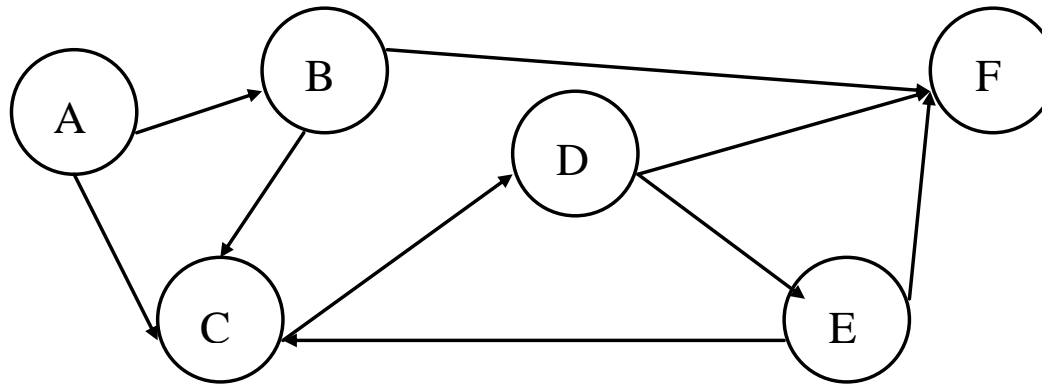
Lista nodurilor vizitate

Coada nodurilor de vizitat

Noduri
adiacente

Grafuri

Traversarea în lățime:



Bibliografie

- ▶ Marius Popa, Cristian Ciurea, Mihai Doinea, Alin Zamfiroiu – *Structuri de date: teorie și practică*, Editura ASE, București, 2023, 280 pg.
 - Cap. 8. Structura de tip graf