Structuri discrete

L. POPOV, dr., lect. univ.
V. ȚÎCĂU, lect. univ.

Grafuri și Arbori

Grafuri

Plan

1. Grafuri. Definiție neformală. Exemple de grafuri
2. Graf neorientat şi orientat.
3. Exemple de grafuri neorientate.
4. Exemple de grafuri orientate.
5. Muchii multiple. Multigraf. Adiacență. Incidență.
6. Reprezentarea grafurilor.
7. Grade. Izomorfism. Exemplu de grafuri izomorfe. Criterii necesare
8. Lanțuri. Conexitate. Cazuri particulare de lanțuri.
9. Grafuri remarcabile.
10. Arbori. Definiția arborilor. Arborescențe. Arbori de acoperire.
11. Grafuri ponderate.
12. Arbori de cost minim.
13. Parcurgerea/traversarea arborilor.
14. Arbori de decizie etc.

Finalități de învățare

La finele studierii temei respective, studentul va fi capabil:

- să definească noțiunea de graf;
- să identifice graful orientat de cel neorientat;
- să definească noțiunea de multigraf și pseudograf;
- să definească noțiunea de grafuri izomorfe;
- să definească noțiunea de lanț, conexitate etc.
- să identifice grafurile remarcabile;
- să definească noțiunea de arbore și arborescență, incidență;
- să definească noțiunea de arbore cu rădăcină;
- să definească noțiunea de arbore de acoperire;
- să definească noțiunea de grafuri ponderate;
- să definească noțiunea de arbore de cost minim;
- să definească noțiunea de arbore de decizie etc.

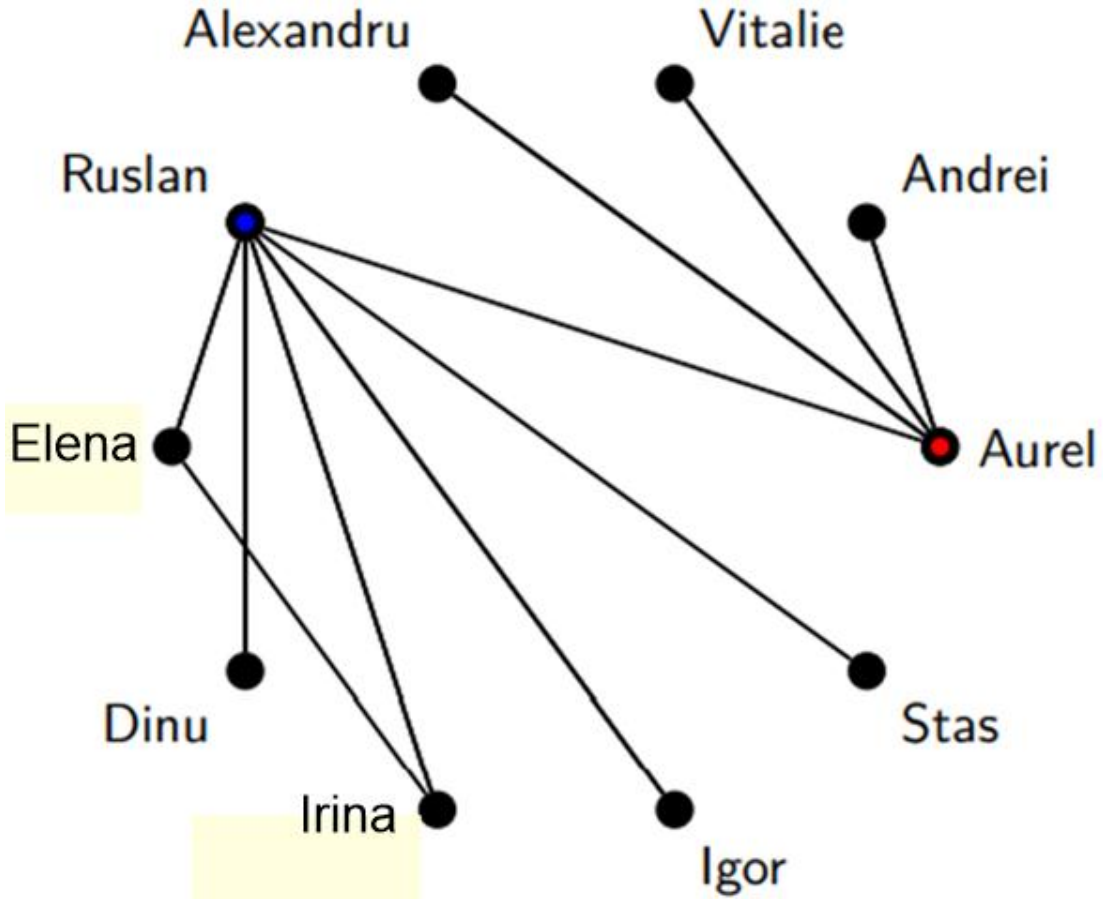
Definiție neformală a unui graf

Un ***graf*** este o structură matematică care constă din obiecte și relațiile dintre aceste obiecte. Obiectele se numesc ***vârfuri***, iar relațiile – ***muchii*** sau ***arce***.

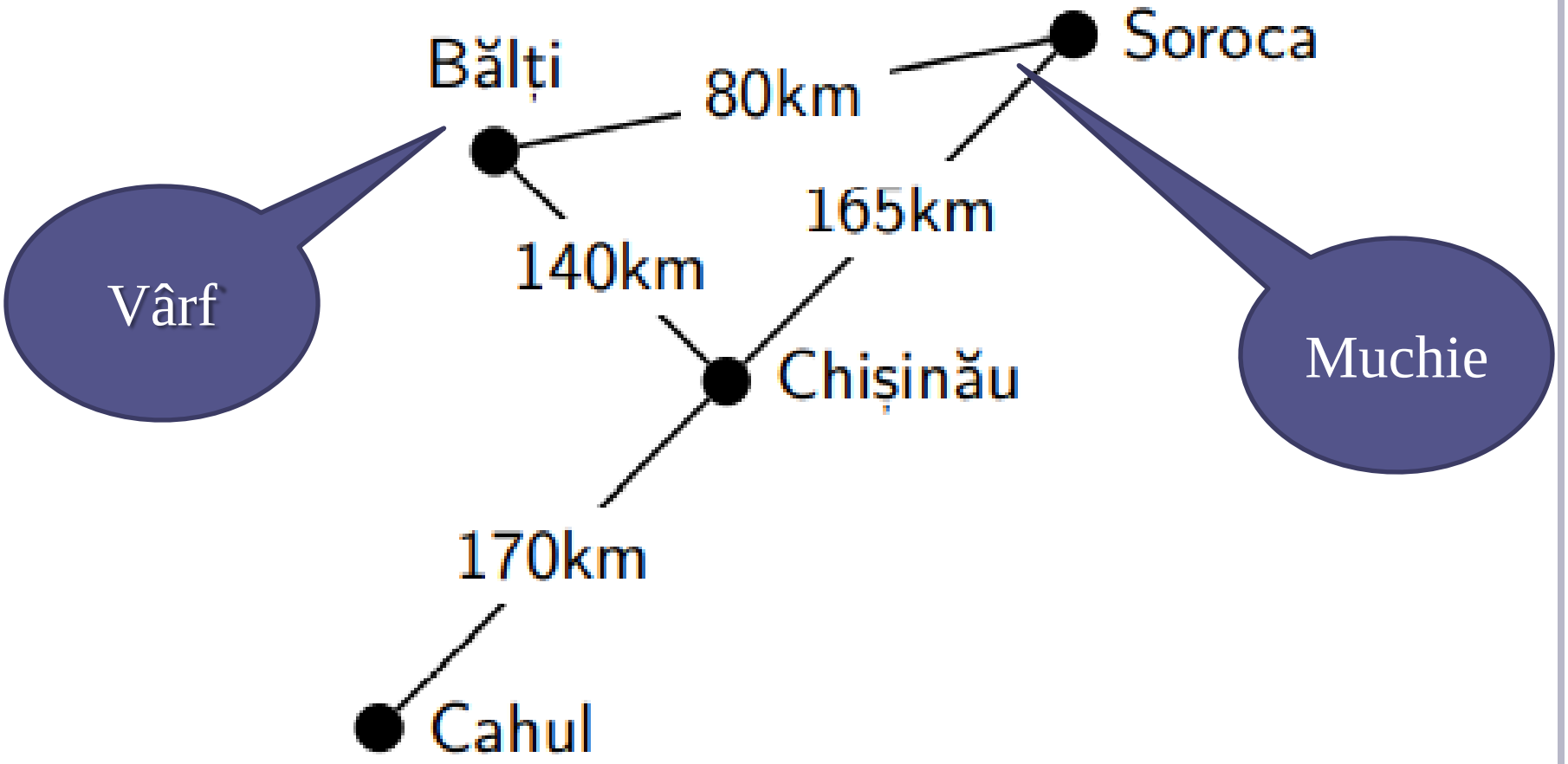
Grafurile pot fi reprezentate geometric: ***vârfurile*** prin puncte, iar ***muchiile*** prin linii drepte sau curbe.

Relații sociale

Aurel este prieten cu Andrei, cu Vitalie, cu Alexandru și cu Ruslan. Ruslan, la rândul lui, evident că este prieten cu Aurel, cu Elena, cu Dinu, cu Irina, cu Igor și cu Stas. Restul persoanelor nu sunt prieteni între ei. cu excepția Elenei și a Irinei.

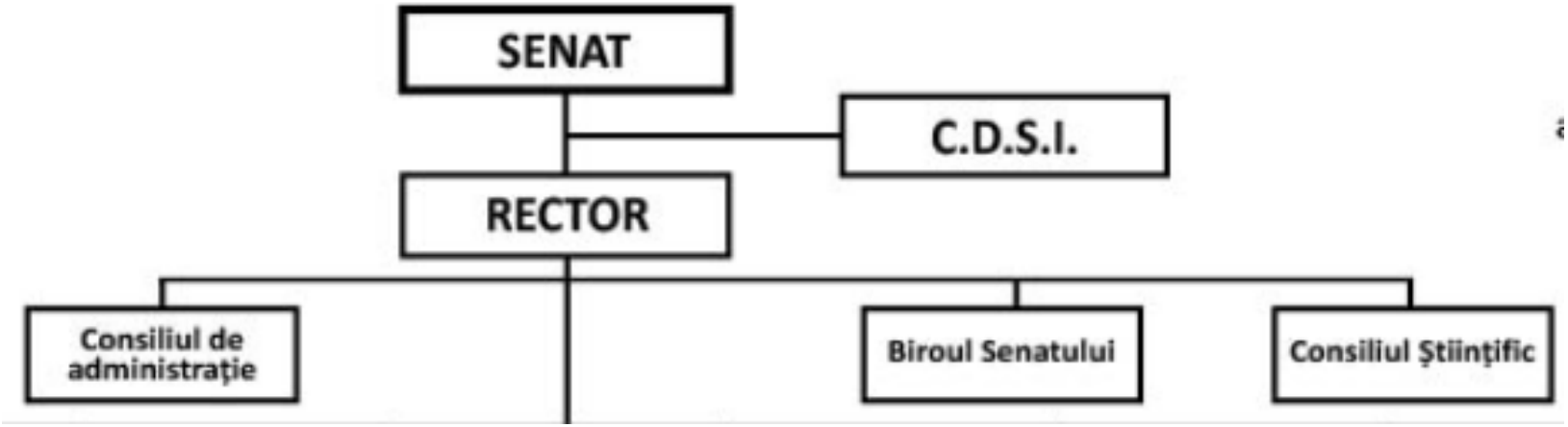


Exemple de grafuri. Hărți



Distanțele sunt aproximative și au fost determinate prin serviciul http://maps.google.com

Exemple de grafuri. Organigrame



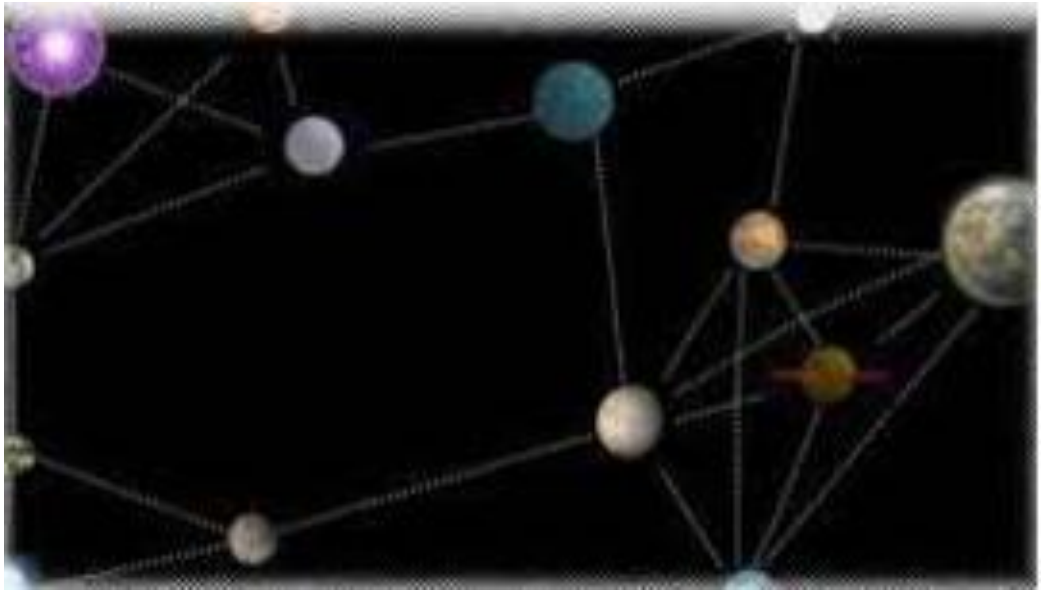
• • •

Fragment din organigrama Universității de Stat „Alecu Russo” din Bălți: https://usarb.md/organigrama/

Grafurile

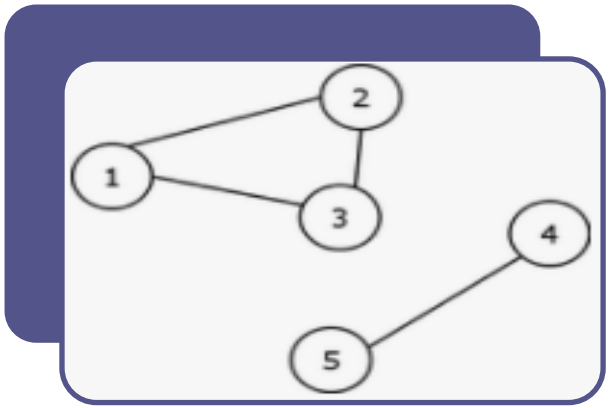
Grafurile au numeroase ***aplicații în diverse domenii***:

- ✓ proiectarea circuitelor electrice;
- ✓ determinarea celui mai scurt drum dintre două localități;
- ✓ rețelele sociale (ex. Facebook) etc.

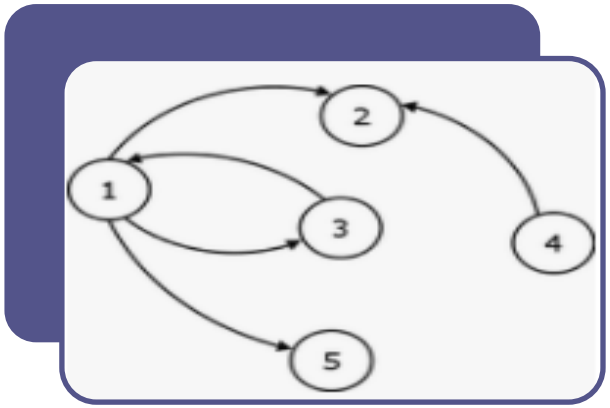


Grafuri

Neorientate



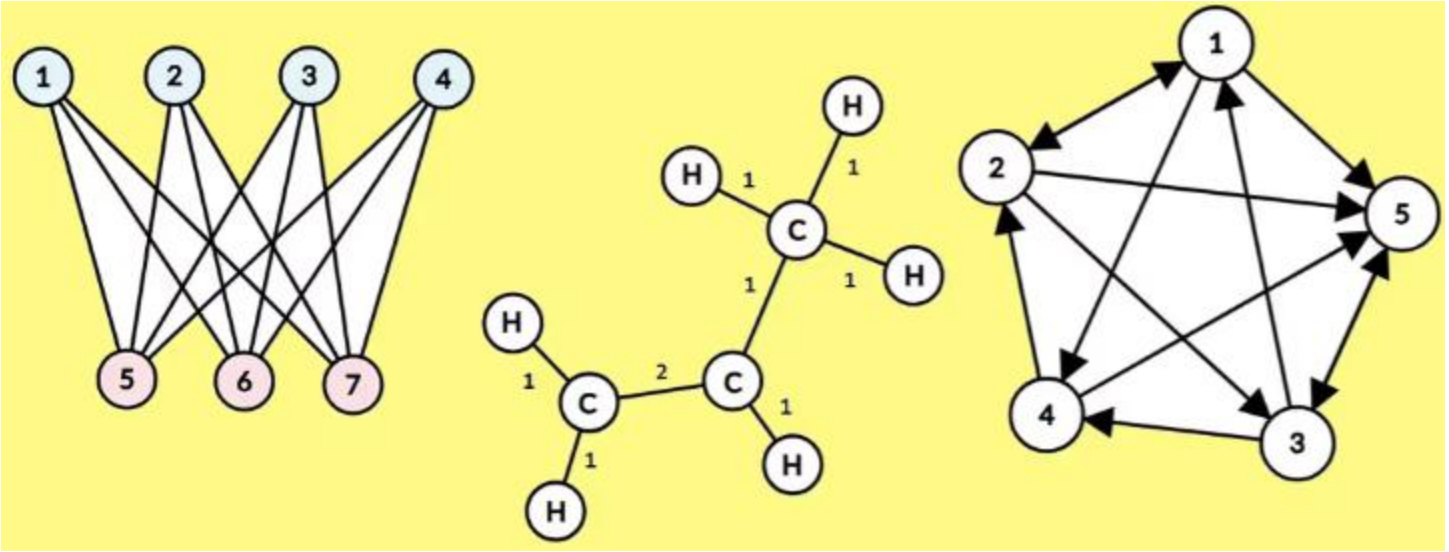
Orientate



Ce este un graf?

Un GRAF este o mulțime de **noduri**, legate între ele printr-o mulțime de **muchii** (la graful *neorientat*) sau **arce** (la graful *orientat*).

Graful este un mod de a reprezenta elementele unei mulțimi și conexiunile dintre ele.



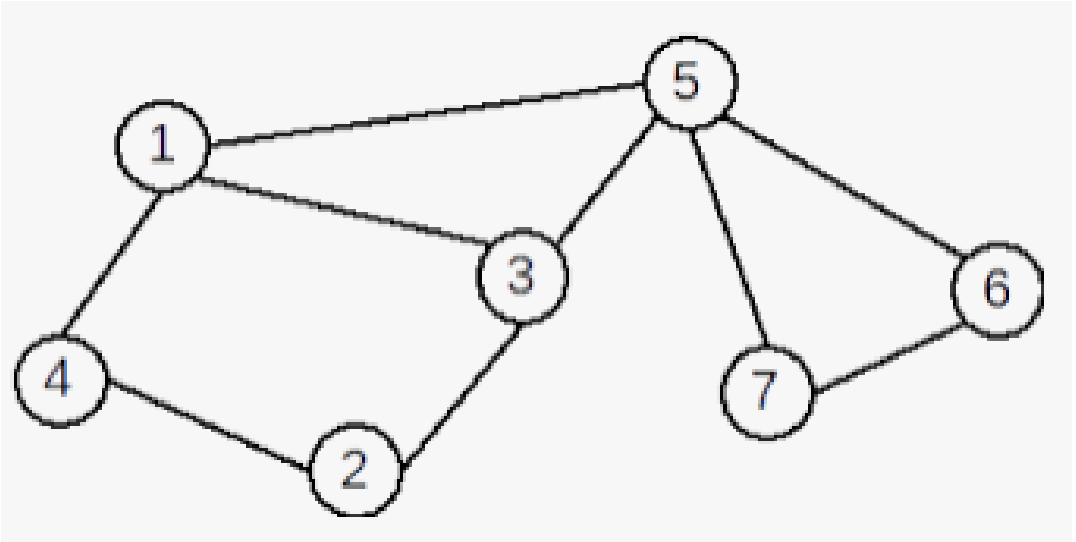
Ce este un graf neorientat?

Un ***graf neorientat*** este o pereche ordonată de mulțimi ***G = (V, E)***, unde:

- ✓ ***V*** este o mulțime finită și nevidă de elemente numită ***vârfuri*** sau ***noduri***;
- ✓ **E** este mulțime finită de submulțimi cu două elemente din **V**, numite muchii.

La graful ***neorientat*** nu contează direcția!

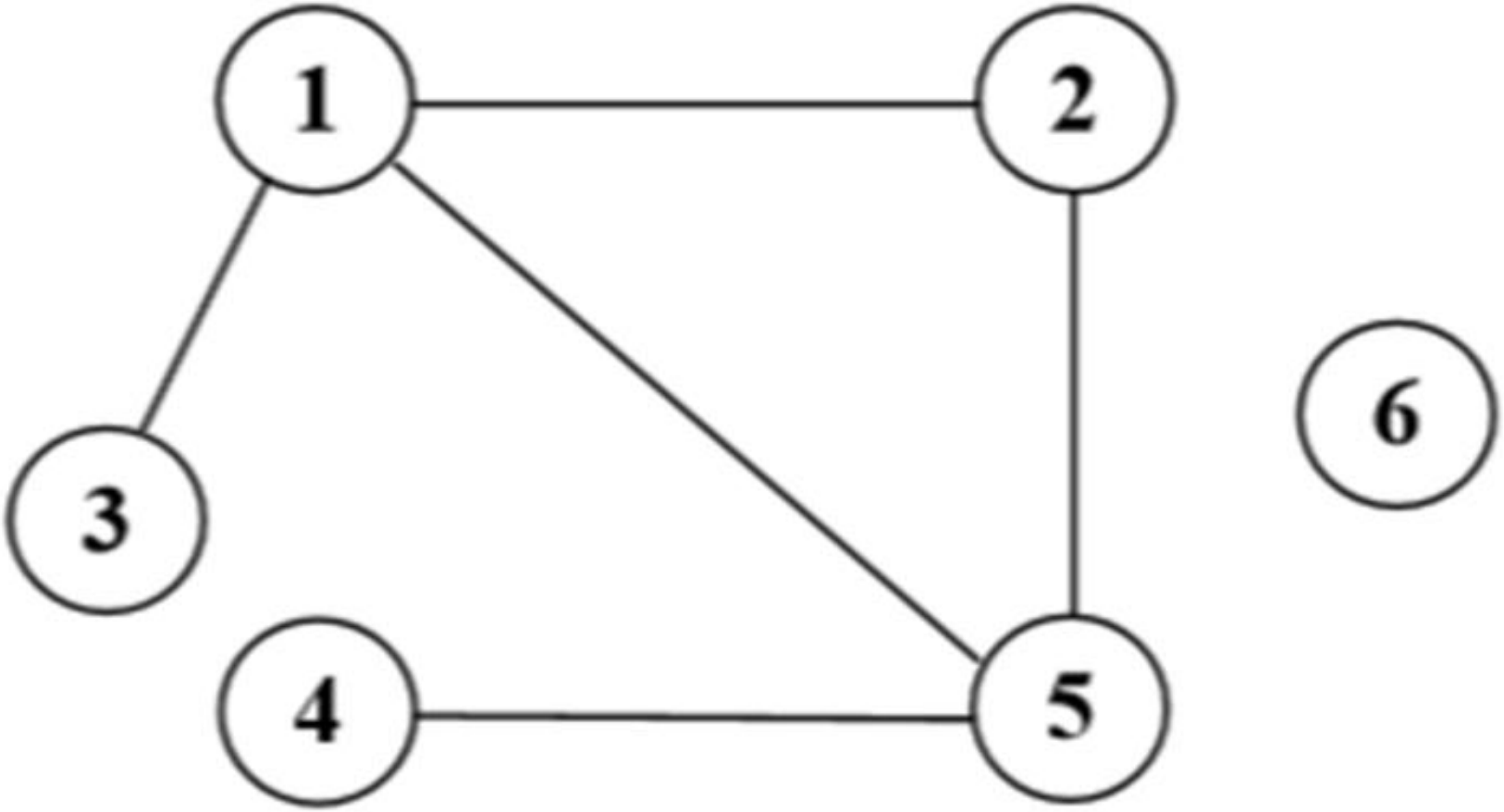
Poate fi considerat și bidirecțional (telefon)!



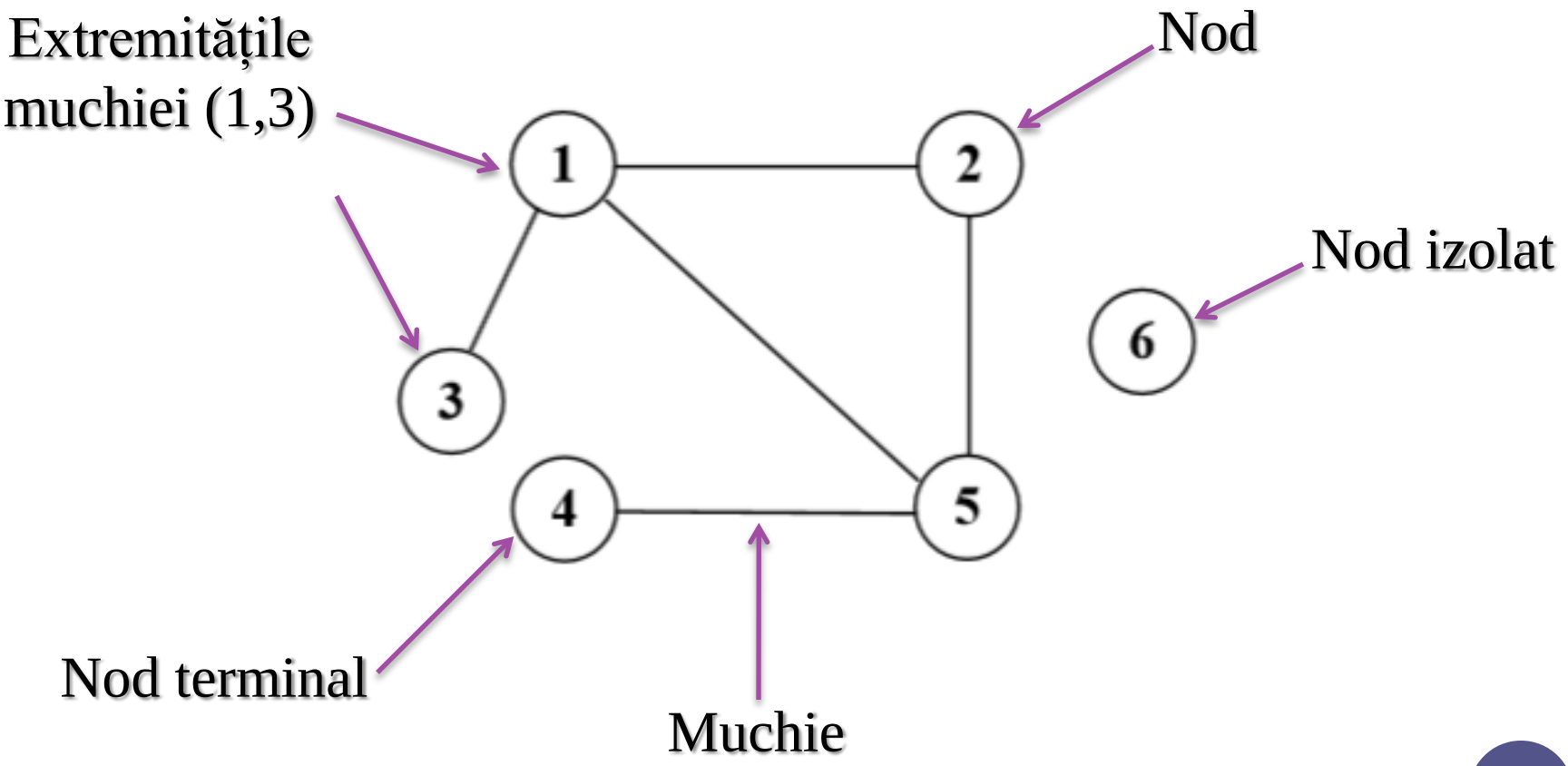
Graf neorientat $G = (V, E)$.

$V = \{1,2,3,4,5,6\}$,

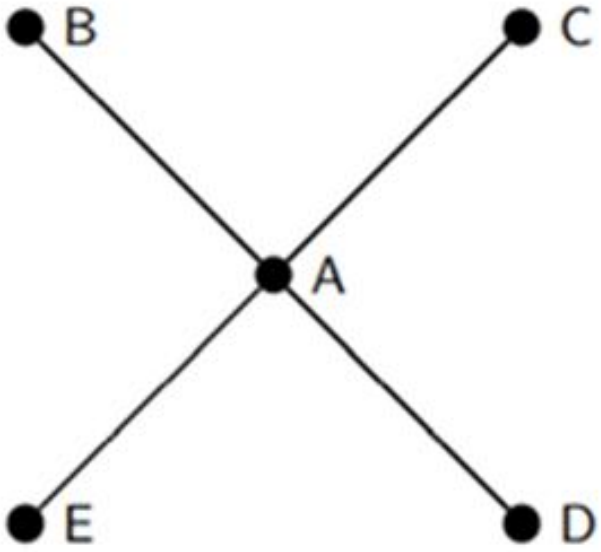
$E = \{(1,2), (1,3,), (1,5), (2,5), (4,5)\}$



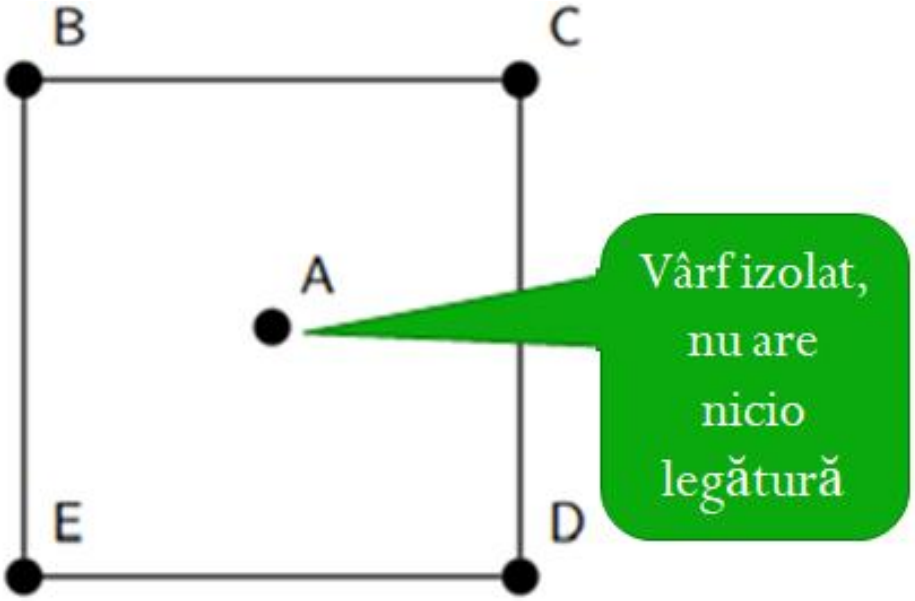
Graf neorientat



Exemple de grafuri neorientate (în ambele direcții)

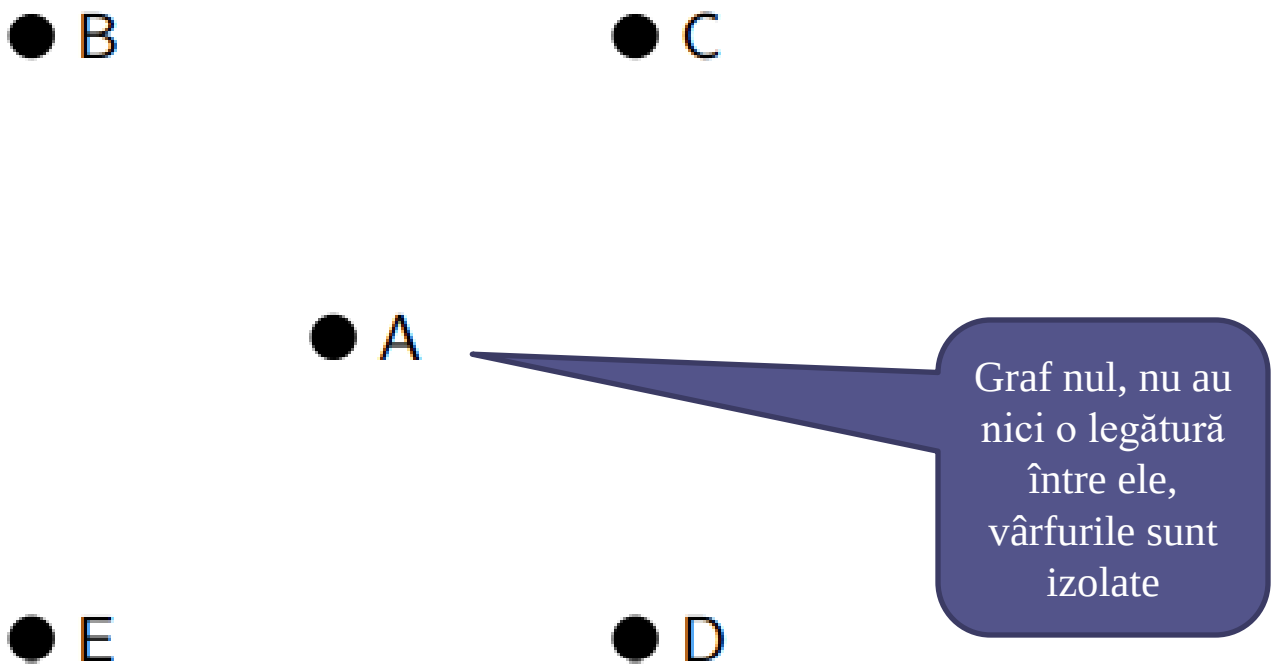


$G_1 = (\{A, B, C, D, E\},$
$\{\{A, B\}, \{A, C\}, \{A, D\}, \{A, E\}\})$



$G_2 = (\{A, B, C, D, E\},$
$\{\{B, C\}, \{C, D\}, \{D, E\}, \{D, B\}\})$

Exemple de grafuri neorientate



$$G_3 = (\{A, B, C, D, E\}, \emptyset)$$

Graf orientat

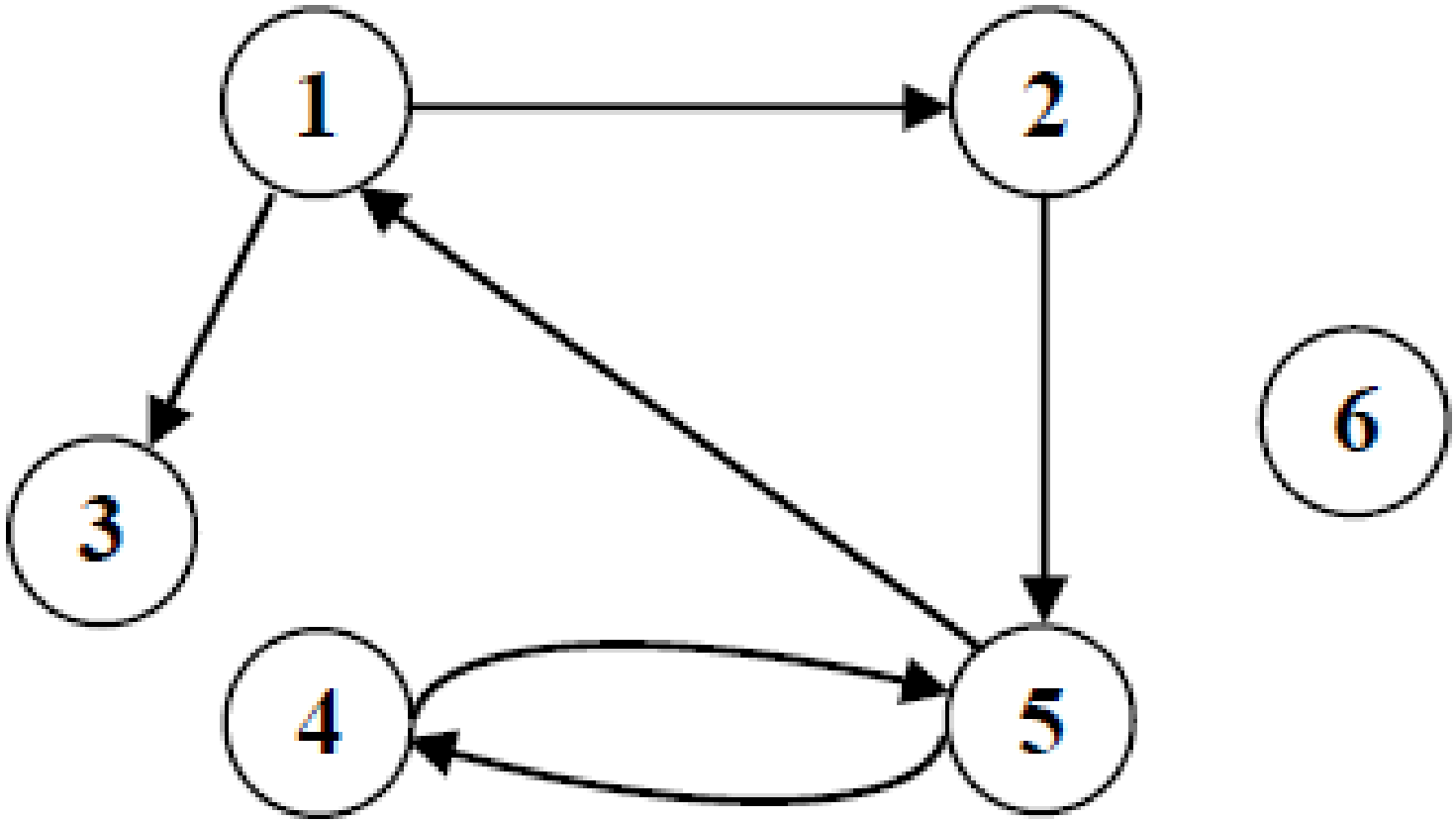
Un ***graf orientat*** este o pereche ***G = (V, E)***, unde ***V*** este o mulțime finită (noduri), iar ***E*** este formată din perechi ordonate (arce) de elemente din ***V***.

Graful orientat este unidirecțional, totdeauna este indicată direcția!

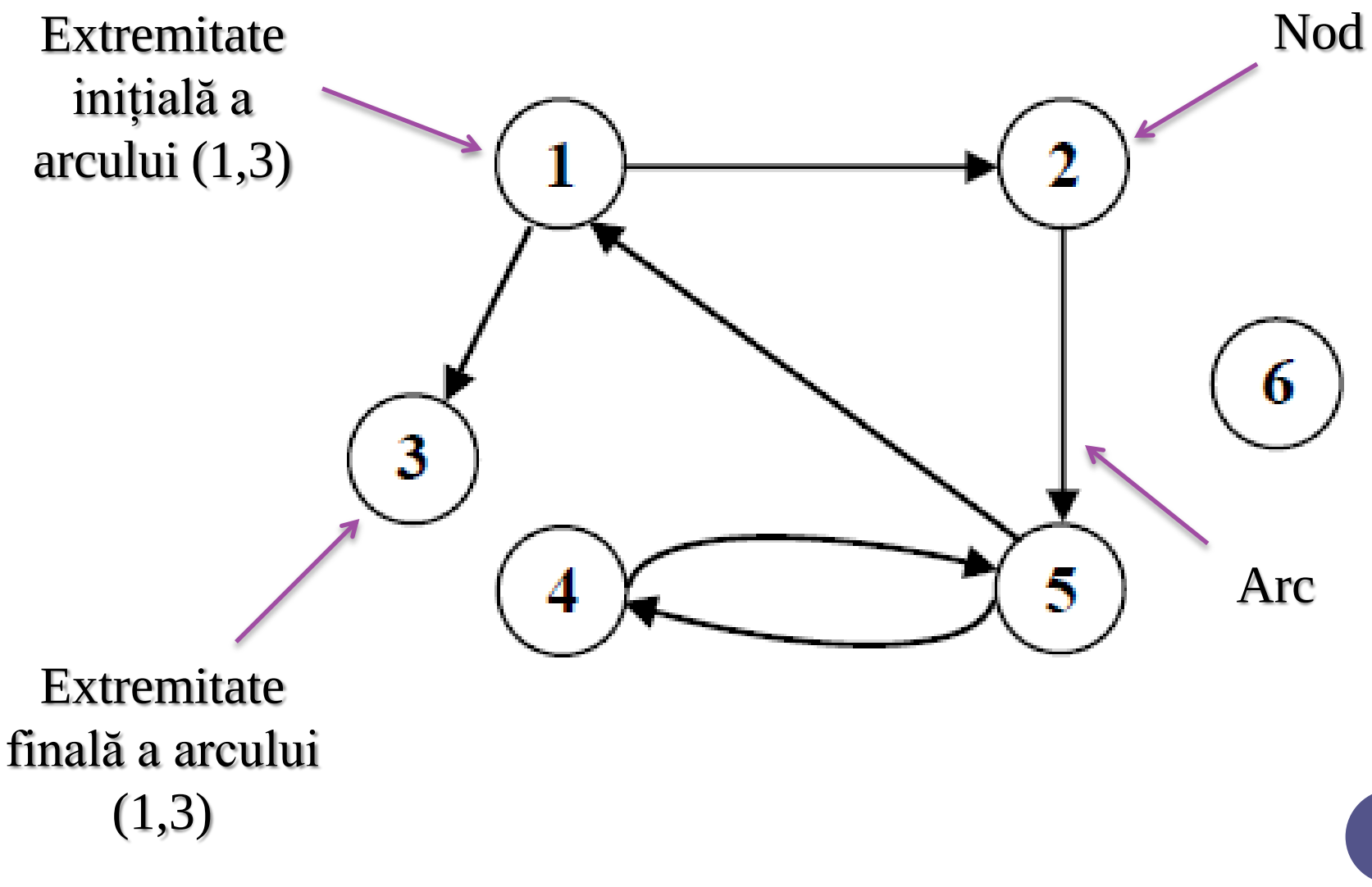
Graf orientat $G = (V, E)$.

$V = \{1,2,3,4,5,6\}$,

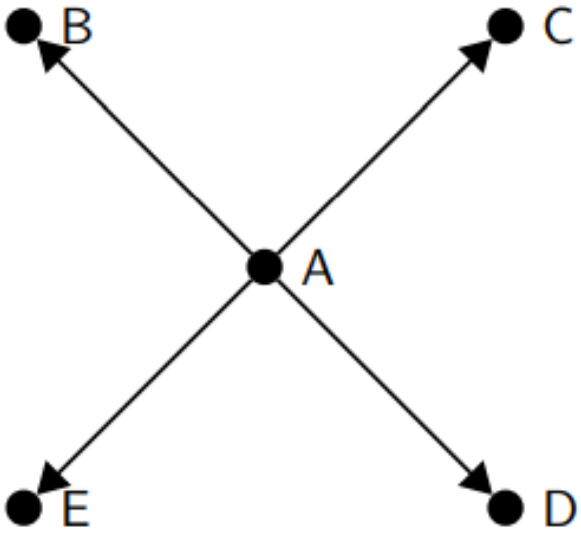
$E = \{(1,2), (1,3), (2,5), (4,5), (5,1), (5,4)\}$



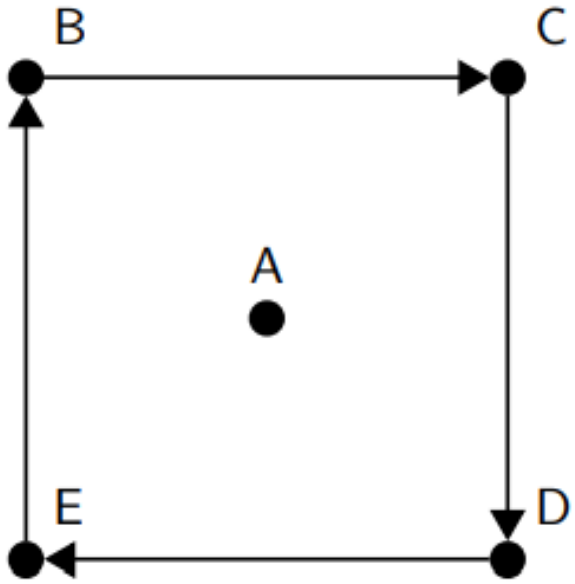
Graf orientat



Exemple de grafuri orientate



$G_1 = (\{A, B, C, D\},$
$\{(A, B), (A, C), (A, D), (A, E)\})$



$G_2 = (\{A, B, C, D\},$
$\{(B, C), (C, D), (D, E), (D, B)\})$

Exemple de Grafuri orientate și neorientate

Fie că avem rețeaua de străzi a unui oraș. Străzile reprezintă ***muchii***, iar intersecțiile reprezintă ***noduri***.

Din punct de vedere al pietonilor, orașul este un ***graf neorientat***, deoarece aceștia se pot deplasa pe stradă în ambele sensuri.

Există în orice oraș și străzi cu sens unic. Pentru un șofer acele străzi reprezintă ***grafuri orientate***.

Exemple de Grafuri orientate și neorientate

Multitudinea căilor aeriene reprezintă ***grafuri***.

Nodurile sunt intersecțiile sau aeroporturile, iar ***muchiile*** sunt rutele.

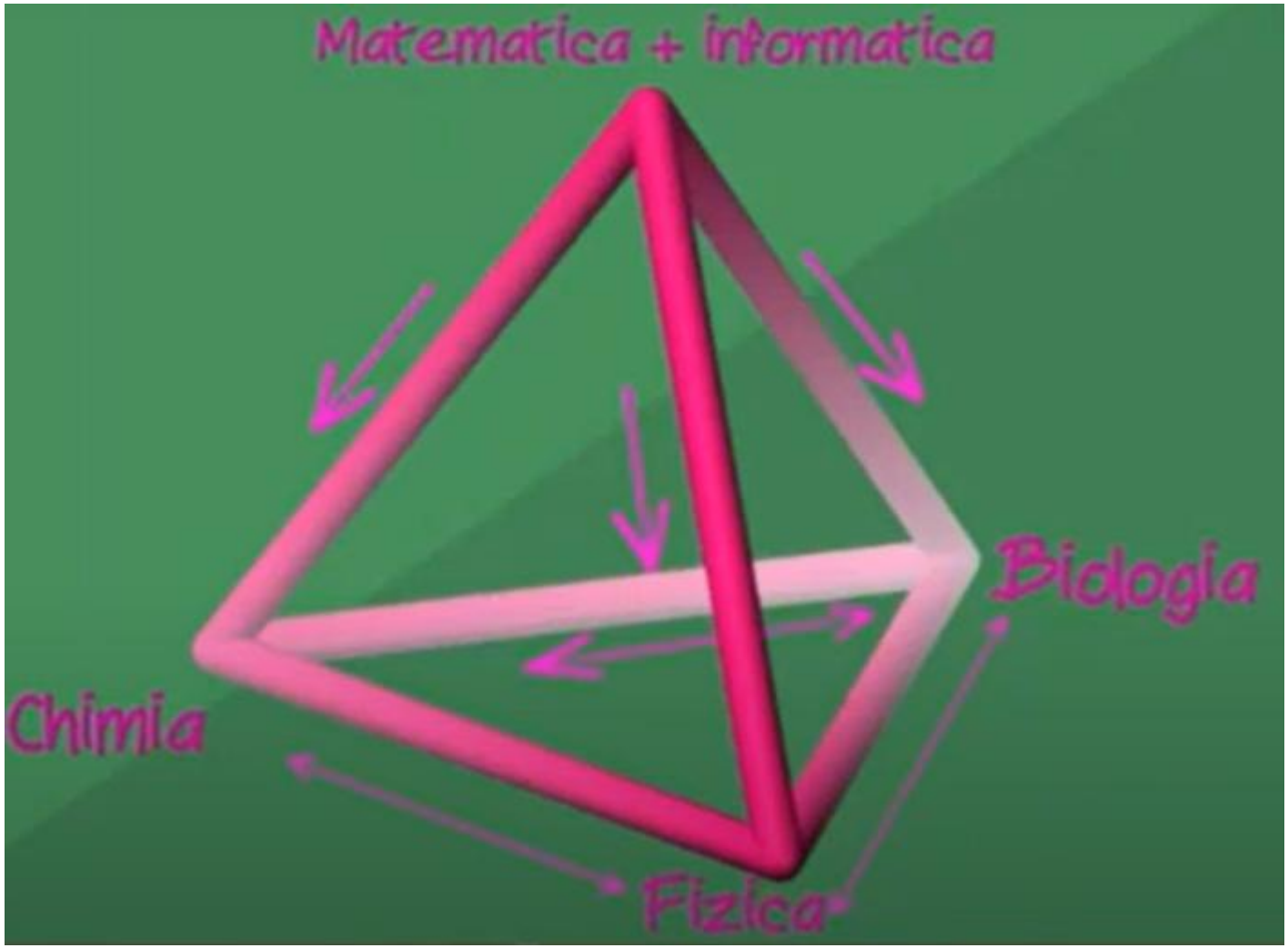
Exemple de Grafuri orientate și neorientate

Cunoștințele sunt un alt tip de graf.

Există:

1. Persoane pe care noi le cunoaștem, dar ele nu ne cunosc pe noi (***graf orientat***).
2. Persoane care ne cunosc pe noi, dar noi nu le cunoaștem (***graf orientat***).
3. Persoane care ne cunosc pe noi și pe care noi le cunoaștem (***graf neorientat***).

Toate științele exacte, la fel reprezintă un Graf



Muchii (arce) multiple

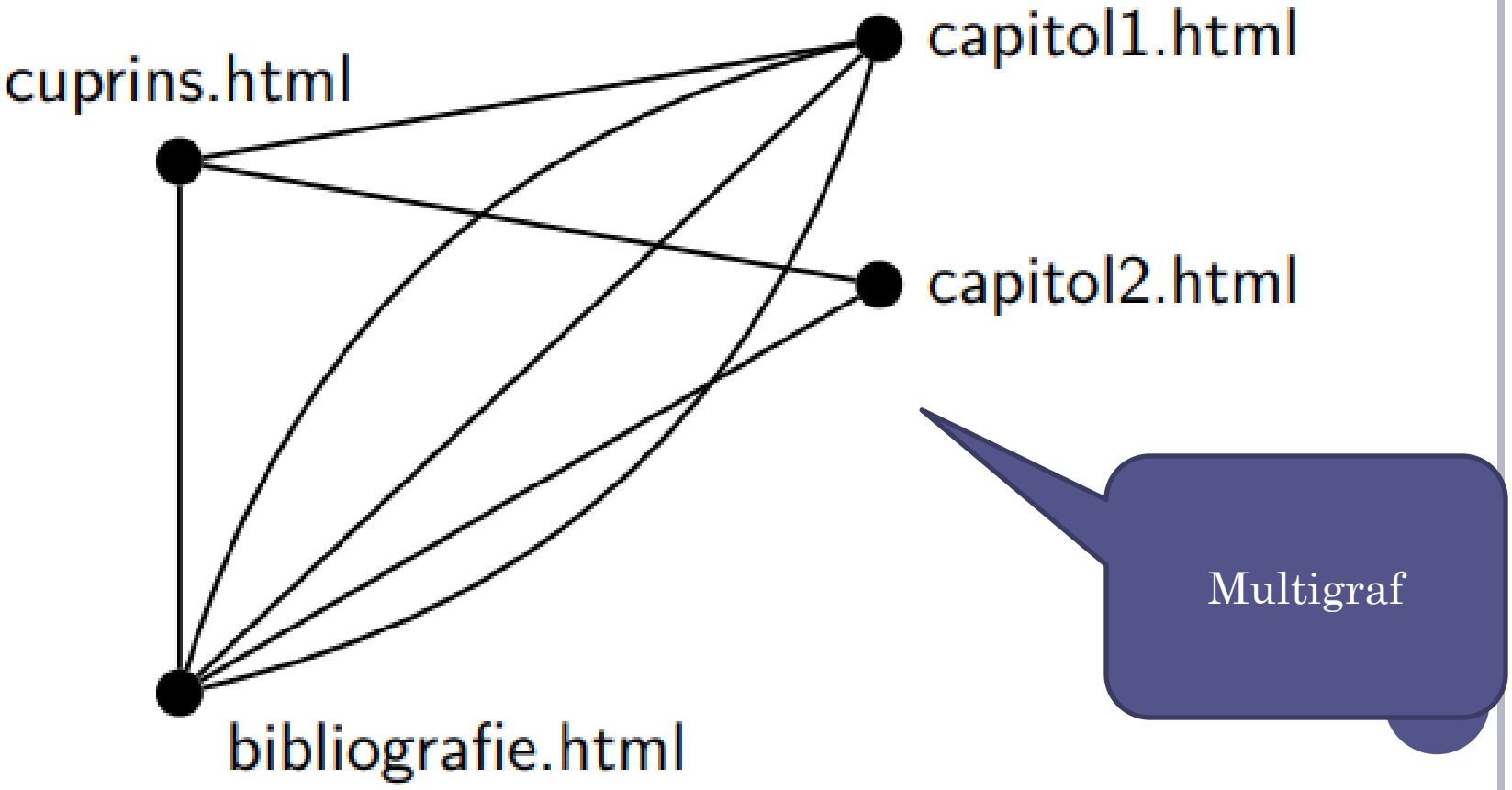
Sunt situații când legăturile dintre obiecte pot fi determinate de mai multe relații.

Exemple:

- ✓ între oameni pot fi relații de prietenie, de afacere etc.
- ✓ între fișiere pot fi relații bazate pe tipul fișierelor, numele, dimensiunea, autorul etc.
- ✓ de la Bălți la Chișinău se poate ajunge folosind mai multe drumuri/trasee naționale, de exemplu: R14 și M2 sau M14.

Exemple de muchii multiple

O muchie între două documente HTML semnifică prezența unei referințe în document care indică către celălalt.



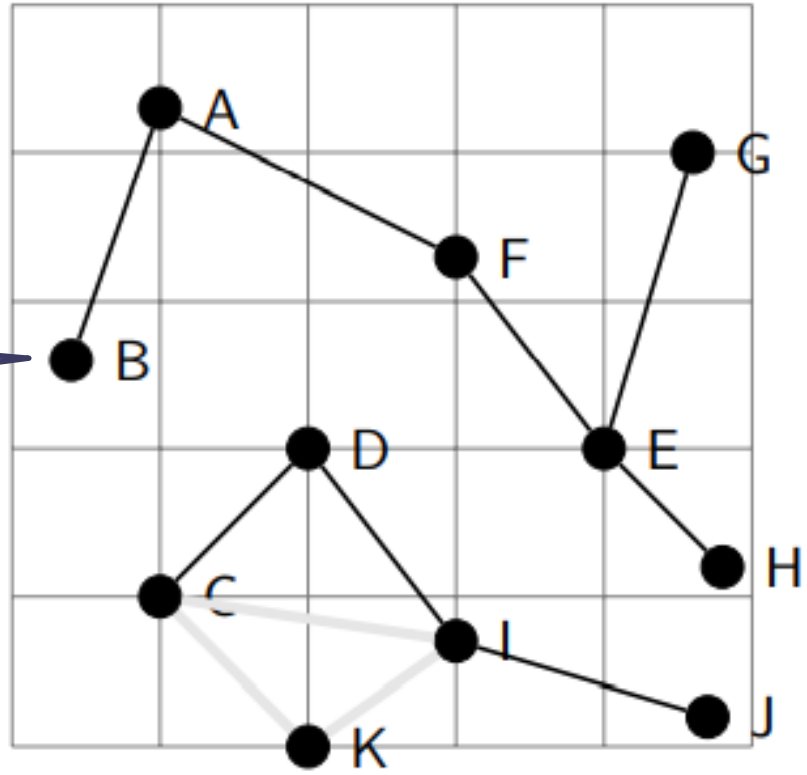
Multigraf

Un graf care conține muchii multiple se numește ***multigraf***.

Un graf care conține bucle se numește ***pseudograf***.

Un graf care nu conține bucle sau muchii multiple se numește ***graf simplu***. ***Bucla*** (figură închisă la care drumul nu se repetă) este o secvență de vârfuri legate între ele la care primul și ultimul vârf coincid.

În cazul în care unim B cu F, obținem o buclă (figură închisă)

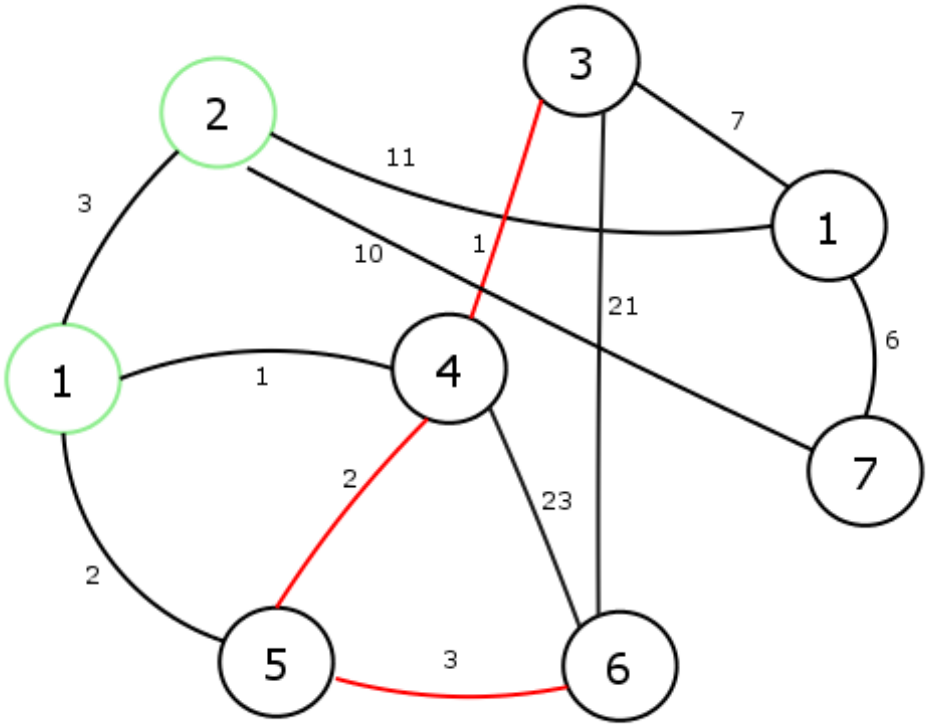


Adiacență

Două ***vârfuri*** se numesc ***adiacente*** (sau ***vecine***) dacă între ele există cel puțin o muchie.

Două ***muchii*** se numesc ***adiacente*** dacă au un vârf comun.

Vârfurile **verzi** sunt adiacente. Muchiile **roșii** formează un drum care leagă nodurile 3 și 6.

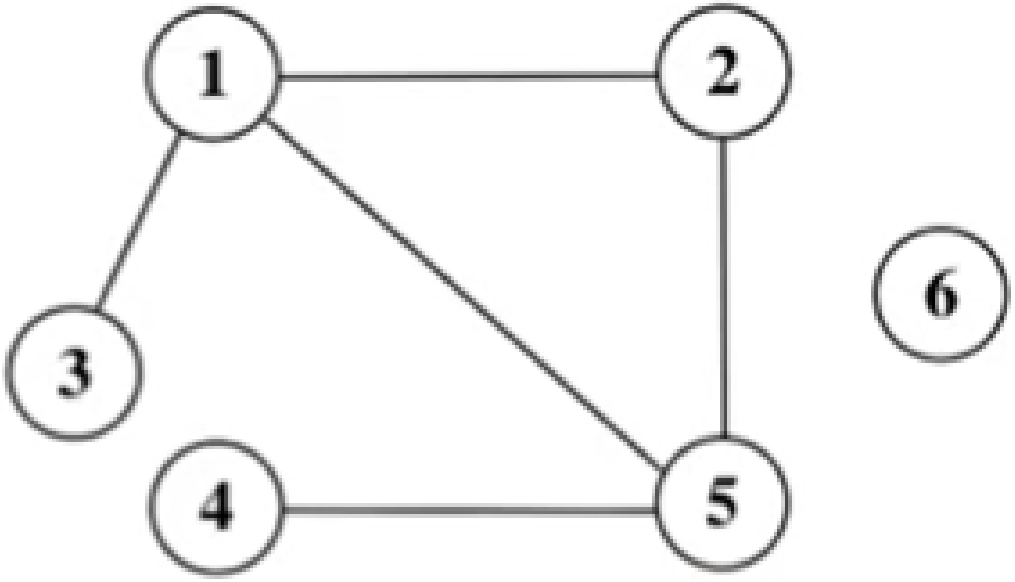


Adiacență și Incidență

În cazul în care există un arc (sau o muchie) cu extremitățile 1 și 2, atunci nodurile 1 și 2 sunt ***adiacente***.

Fiecare extremitate de muchie sau de arc este considerată ***incidentă*** cu muchia sau arcul respectiv.

Exemplu: Nodul 1 și muchia (1,2) sunt ***incidente***.



Incidență

O muchie este incidentă cu un vârf dacă vârful este unul din capetele sale sau altfel spus ***un vârf este incident cu o muchie*** doar în cazul în care vârful este extremitate a acelei muchii.

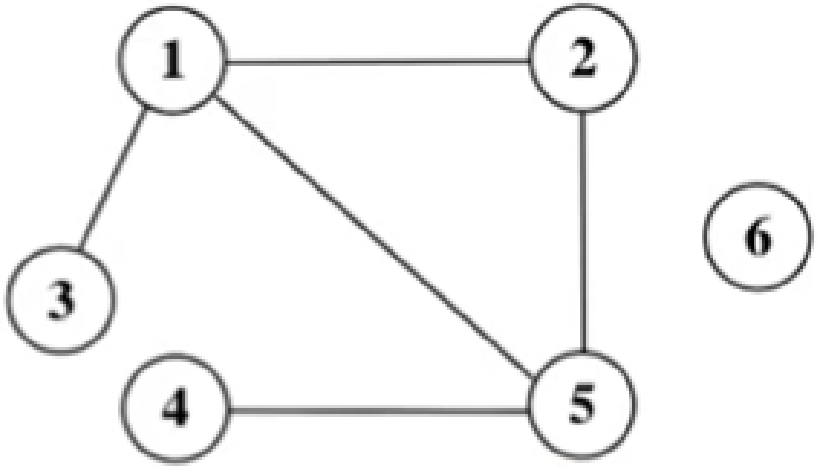
De exemplu:

Muchia dintre A și B este incidentă la vârfurile A și B.

Două muchii sunt ***incidente*** dacă au o extremitate comună.

Mulțimea muchiilor are ***proprietatea de simetrie***: dacă (1,2) este muchie, atunci și (2.1) este muchie.

În continuare putem desprinde următoarele concluzii.



Grafuri neorientate

Conform definiției:

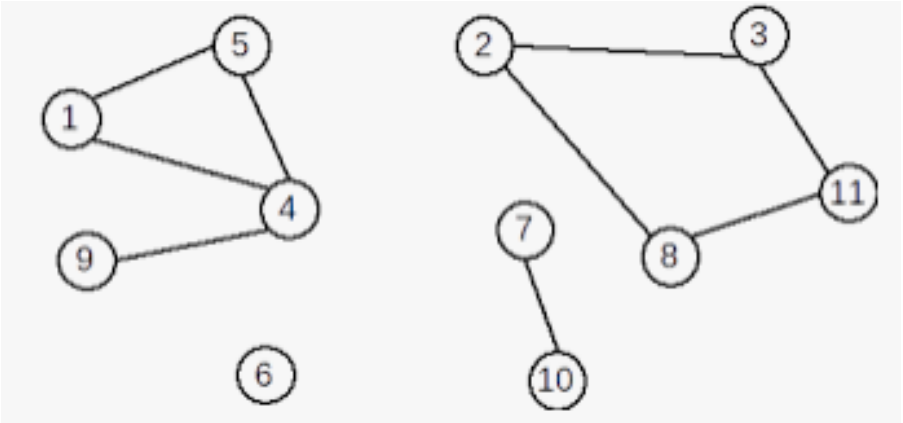
✓ Într-un graf neorientat nu există muchie de la un vârf la el însuși, adică nu există buclă;

✓ Între două vârfuri distincte există cel mult o muchie, adică dacă există legătură de la 1 la 5, muchia poate fi citită (1,5) sau (5,1), nu contează, ambele cazuri sunt corecte.

Exemplu:

V={(1,2,3,4,5,6,7,8,9,10,11)} - mulțimea valorilor de la 1 la 11 care reprezintă ***nodurile*** sau ***vârfurile*** grafului neorientat.

E= {(1,4), (1,5), (2,3), (2,8), (3,11), (4,5), (4,9), (7,10), (8,11)} formată din perechi de noduri în care există muchii. Observăm nodul 6 care nu este legat prin nici o muchie de restul grafului și poartă numele de ***nod izolat***.



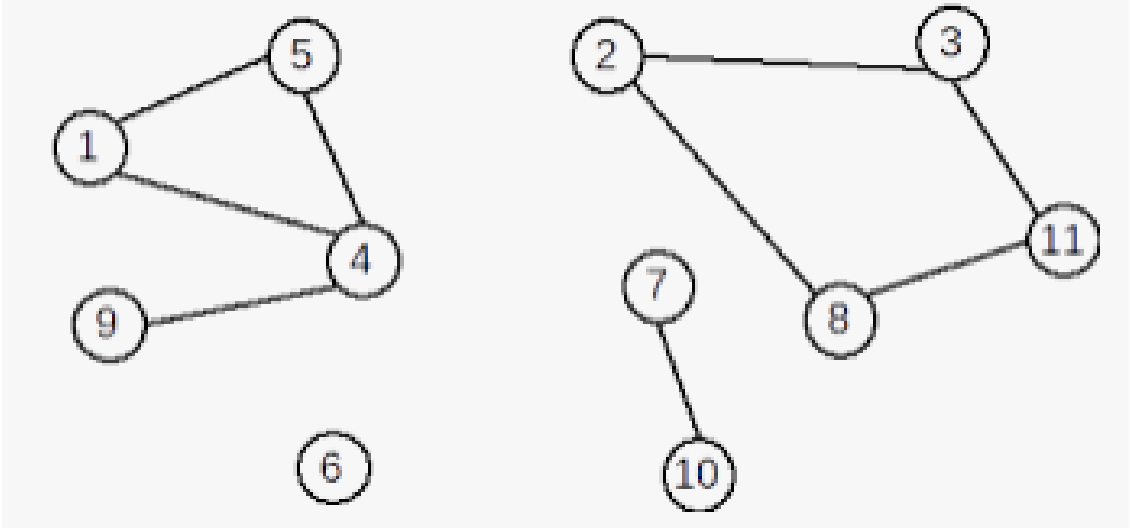
Gradul unui vârf

Pentru fiecare nod din graf se poate stabili gradul nodului respectiv.

Definiție: Gradul unui vârf (nod) este egal cu numărul de muchii incidente în acel nod. Gradul unui vârf (nod) **x** se notează cu **d(x)**. Analizând același graf, să calculăm gradele nodurilor.

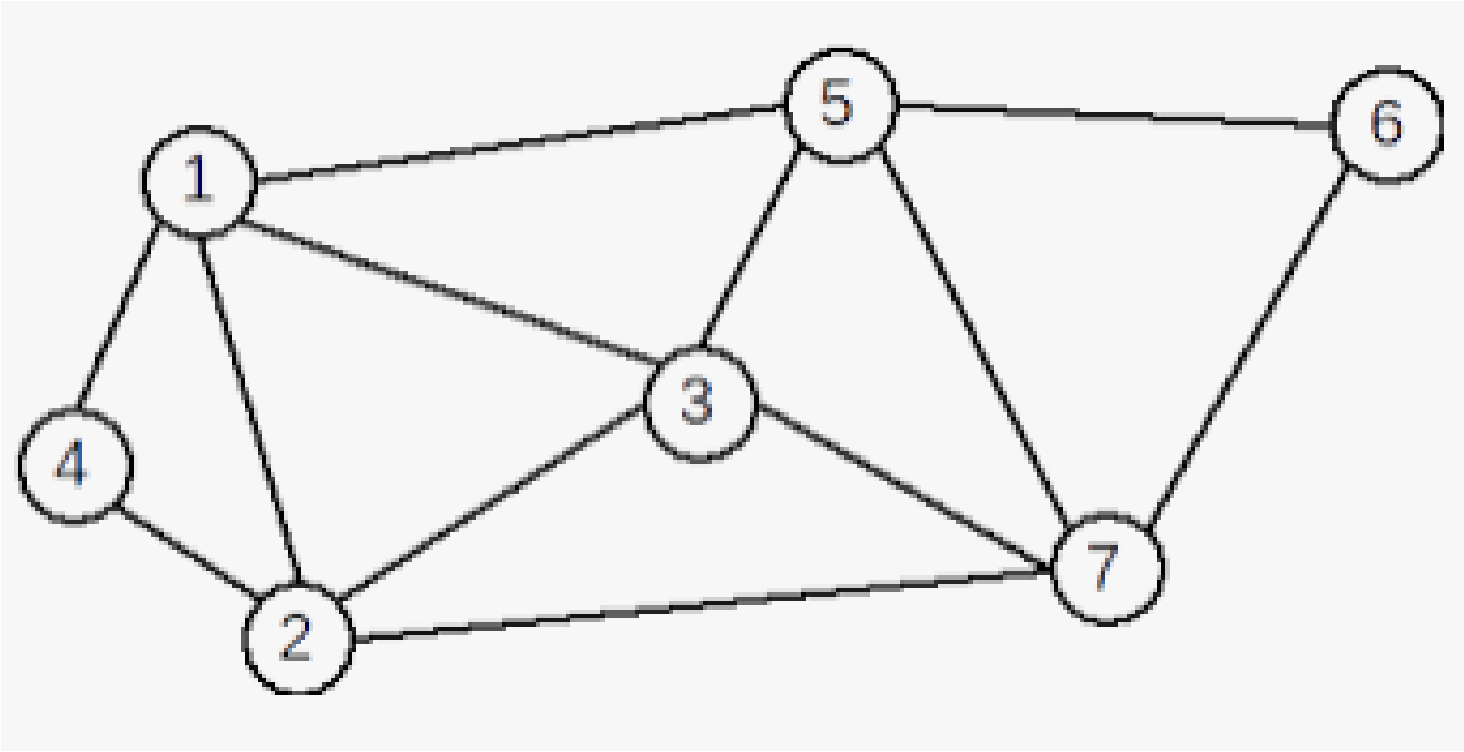
De exemplu: Gradul nodului 1, vedem că sunt două muchii incidente, de unde rezultă că d(1)=2; d(5)=2; d(4)=3; d(9)=1; d(2)=2; d(3)=2; d(11)=2; d(8)=2; d(7)=1; d(10)=1; d(6)=0.

Vârful cu gradul 0 poartă numele de ***nod izola***t, iar vârful cu gradul 1 **-** ***terminal***. Gradul ***maxim*** al unui vârf într-un graf cu **n** vârfuri este egal cu **n-1**.



Gradul unui vârf

În cazul în care ***Gradul*** unui vârf este numărul de muchii incidente cu acest vârf, rezultă că ***Bucla*** are cel puțin gradul doi, adică are două muchii. În cazul buclei te pornești pe un drum și te întorci pe un alt drum în același vârf, din acest motiv spunem că are minimum două muchii. **Calculați gradul nodurilor!**



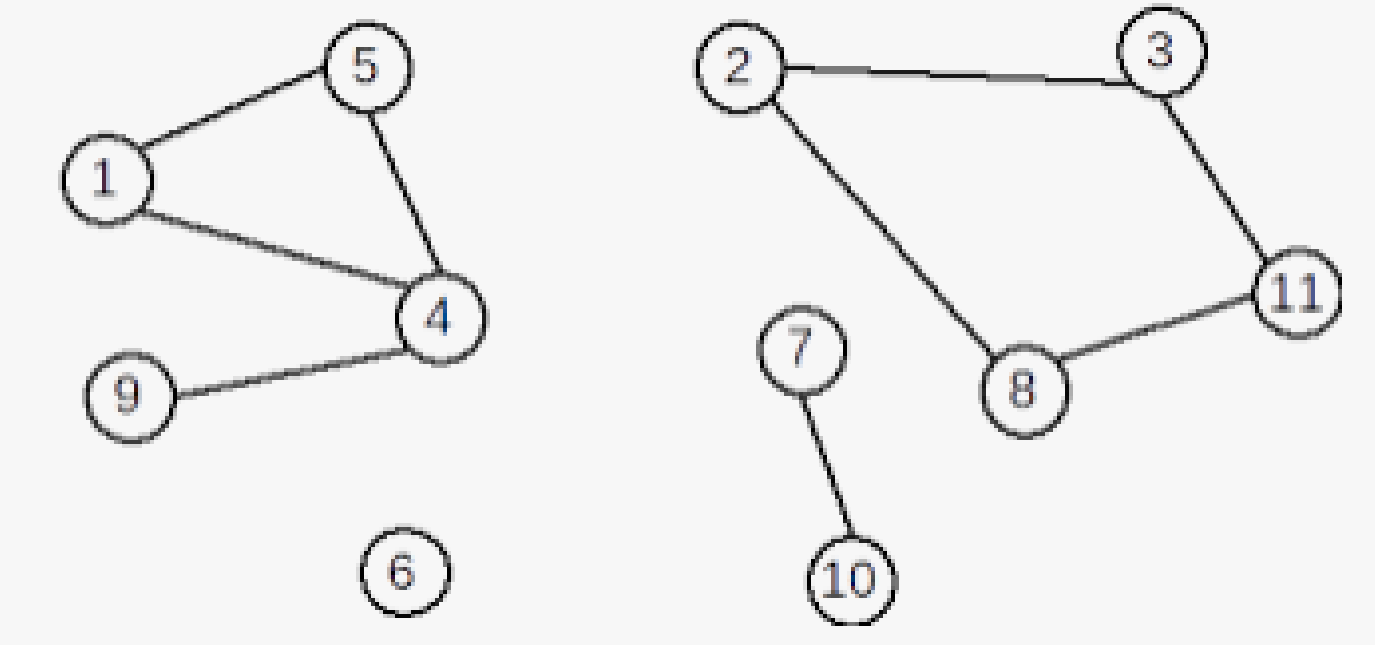
Grafuri neorientate

Într-un graf neorientat, suma gradelor tuturor vârfurilor este ***dublul numărului de muchii***. În cazul nostru avem 18; (9x2=18)

Consecințe:

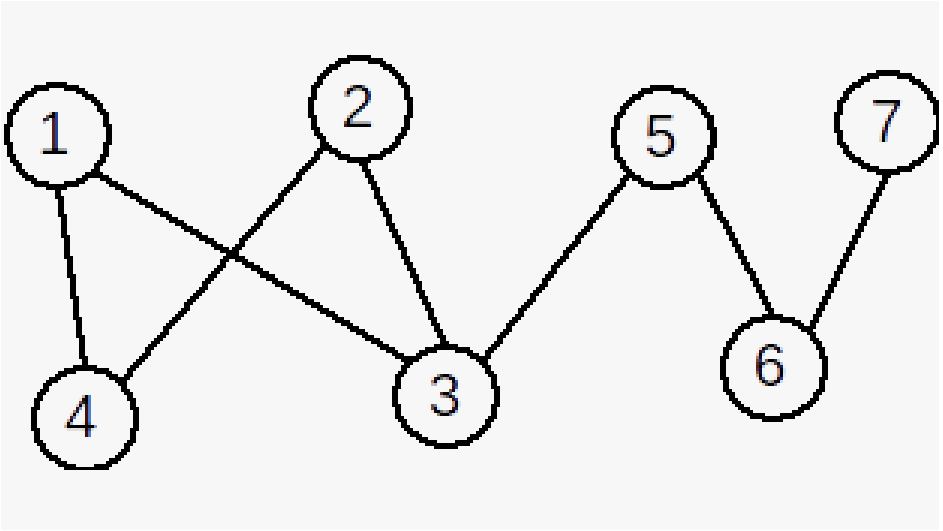
✓ Suma gradelor tuturor vârfurilor este număr par;

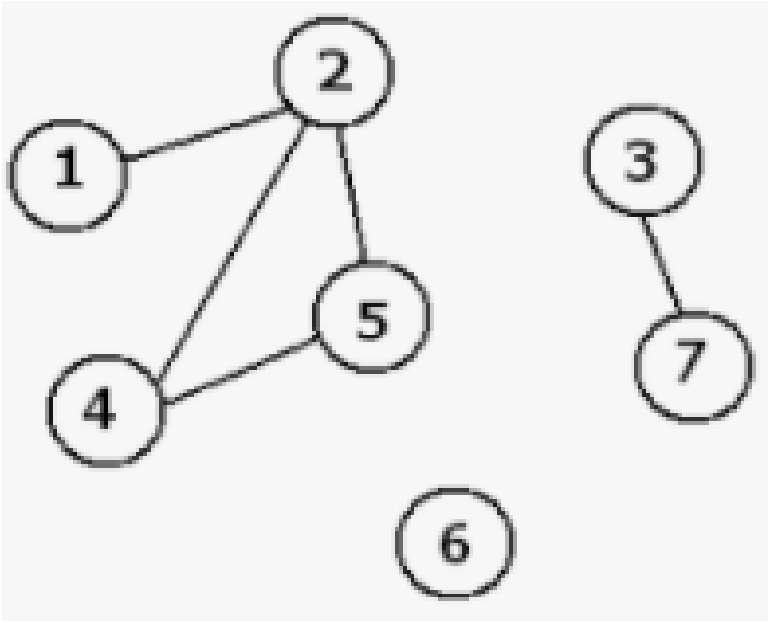
✓ Într-un graf neorientat, numărul de vârfuri de grad impar este întotdeauna par.



Exercițiu: Calculați gradul vârfurilor (nodurilor).

Care Nod este Izolat? Care nod este Terminal?





Reprezentarea grafurilor

1. Matrice de adiacență
2. Matrice de incidență muchie-vârf
3. Matrice de incidență vârf-muchie
4. Liste de adiacență
5. Liste de incidență

Matricea de adiacență

Se pune problema, precum că în cazul în care vrem să prelucrăm un astfel de ***graf*** trebuie să-l implementăm în memoria calculatorului.

O modalitate de a reprezenta în memoria calculatorului un graf neorientat este ***Matricea de adiacență*** (MA).

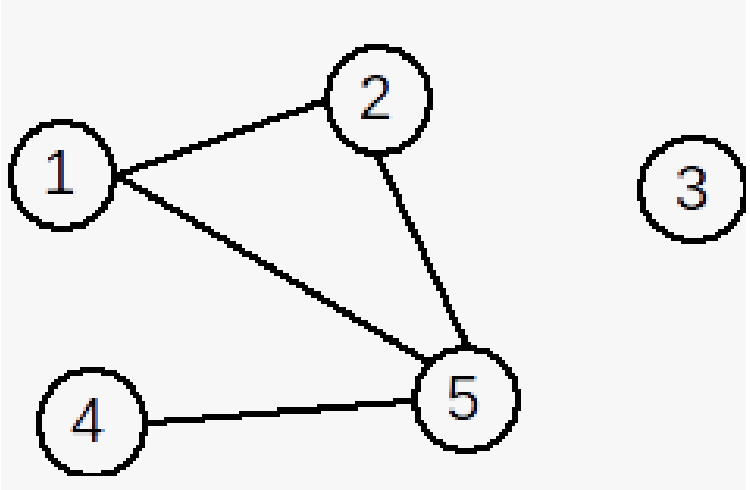
MA pentru un graf neorientat cu **n** vârfuri este o matrice pătrată de dimensiuni **n** ori **n**, unde **n** este egal cu numărul de vârfuri (noduri). MA include elemente de (0,1) și se notează totdeauna cu A, astfel încât oricare element $\mathbf{A_{i,j}}$ din matrice, poate să aibă valoarea **1** în caz că există muchie de la nodul **i l**a nodul **j** sau **0** dacă nu exista muchie de la nodul **i l**a nodul **j**.

$$A_{i,j} = \begin{cases} 1 & \text{dacă } [i,j] \in U \\ 0 & \text{dacă } [i,j] \notin U \end{cases}$$

Matricea de adiacență

Pentru graful din imagine să construim ***matricea de adiacență***. Avem un graf cu 5 noduri și putem și recapitula elementele care deja le cunoaștem. Să vedem care ar fi mulțimea X? Este formată din elementele 1,2,3,4,5 și mulțimea U este formată din perechile de noduri cu muchiile (1,2); (1,5); (2,5); (4,5). Și acum MA poate fi citită de către noi fie de la tastatură sau fie introdusă tot de către noi într-un fișier din care să o citim. Privim la elementul $\mathbf{A_{i,j}}$, apoi ne uităm la graf să vedem dacă există muchie între nodul **i** și **j**, dacă există, atunci valoarea din matrice va fi **1**, în caz contrar va fi **0**.

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

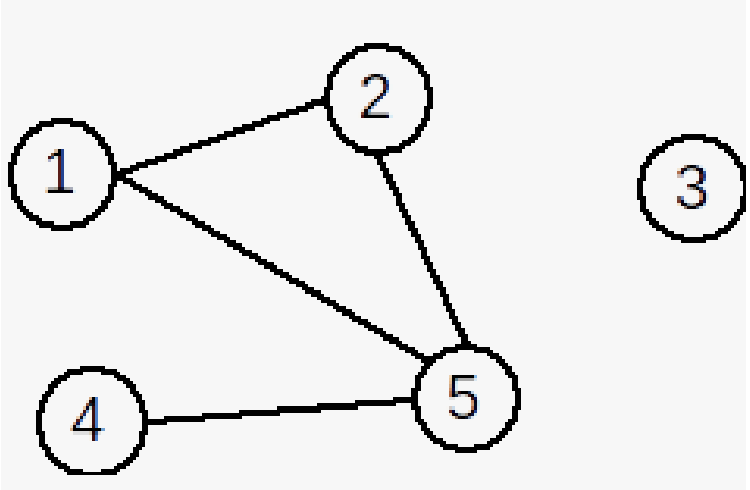


Matricea de adiacență

După ce am construit această matrice putem scoate în evidență următoarele observații: MA este simetrică față de diagonala principală, că toate elementele de pe diagonala principală sunt **0**, exact e despre ceea ce am discutat anterior că nu putem avea bucle și gradul oricărui vârf X este egal cu numărul de elemente de **1** aflate pe linia sau pe coloana X. Acest lucru se verifică destul de ușor. Care ar fi gradul nodului d(1)=2?

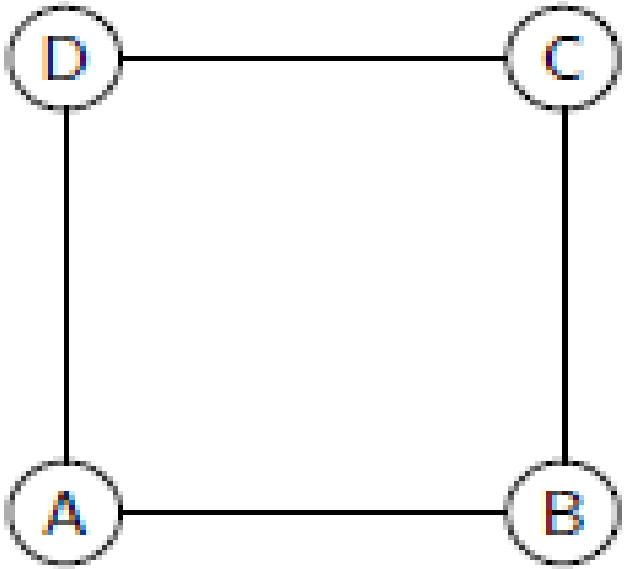
Observăm că pe prima linie în MA avem 2 de 1. Exact în așa mod verificăm mai departe: d(2)=2; d(3)=0; d(4)=1; d(5)=3. Pentru a calcula gradul unui nod trebuie să calculăm suma elementelor de pe linia respectivă.

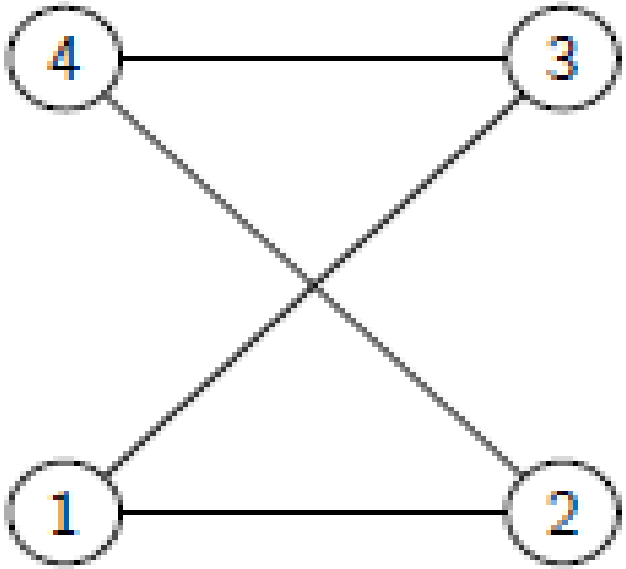
$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$



Izomorfism

Două grafuri ***G*** și ***H*** se numesc ***izomorfe*** dacă există o funcție bijectivă ***f: V(G) → V(H)*** astfel, încât ∀***u, v ∈ V(G)***, ***u*** și ***v*** sunt vecine dacă și numai dacă ***f(u)*** și ***f(v)*** sânt vecine în ***H***.





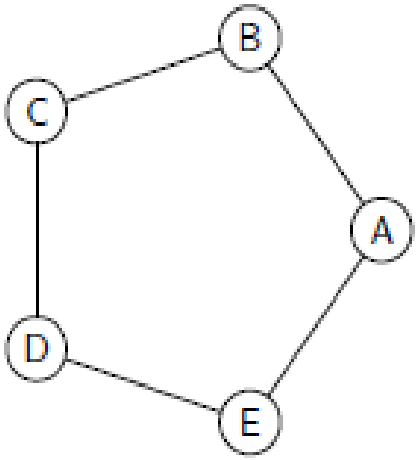
Funcția bijectivă ***f: V(G) → V(H)***

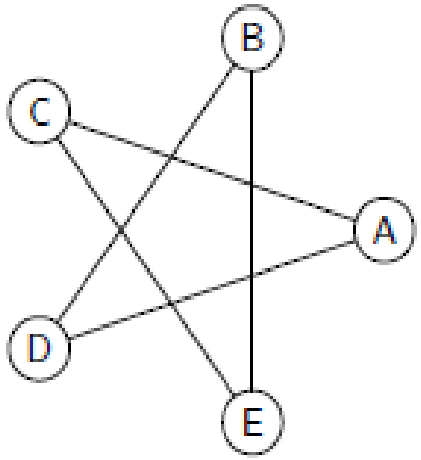
Fie A → 1; B → 2; D → 3; C → 4

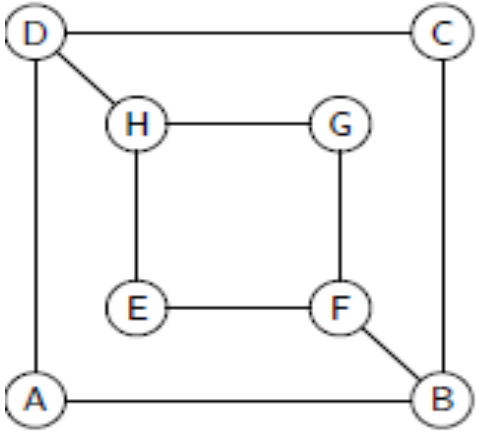
g(A) = {D, B}

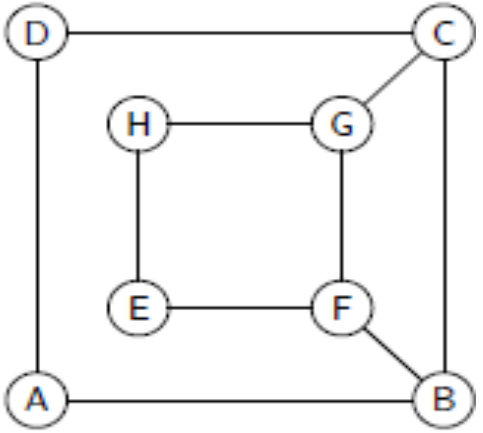
g(B)={A, C}

Exemplu de grafuri izomorfe









Criterii necesare

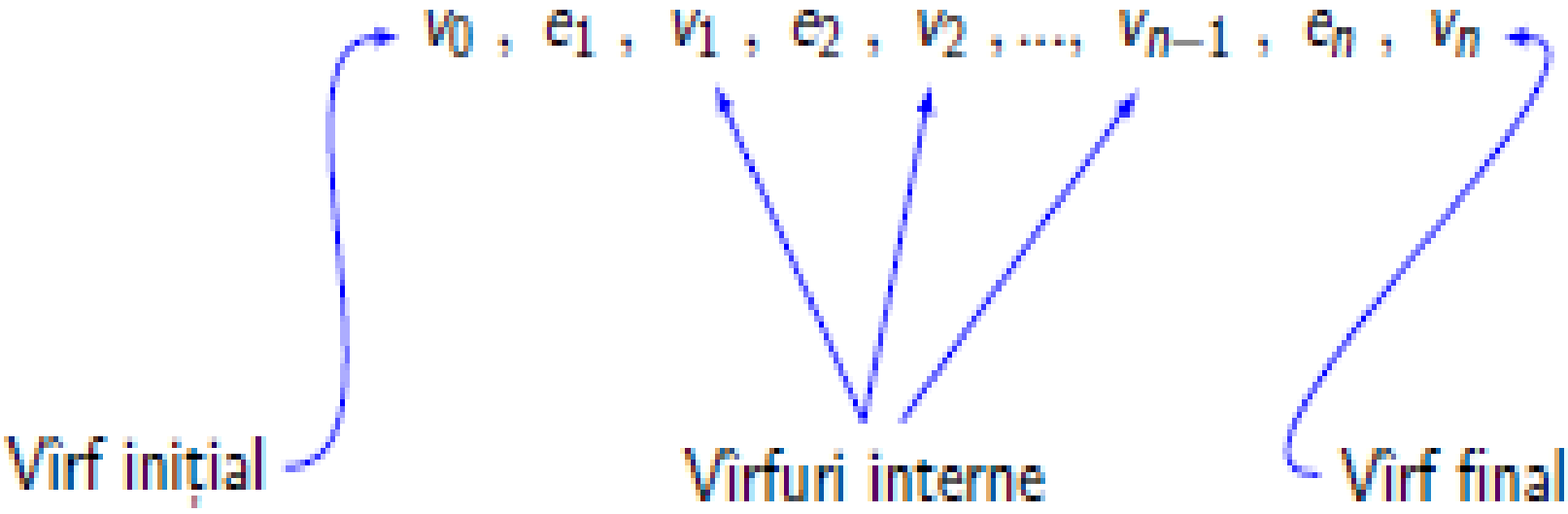
Dacă ***G*** și ***H*** sunt două grafuri ***izomorfe*** atunci:

- ✓ au același număr de vârfuri;
- ✓ au același număr de muchii;
- ✓ au aceleași secvențe de grade.

Dacă la două grafuri diferă numărul de vârfuri sau diferă numărul de muchii sau diferă secvențele de grade atunci aceste grafuri ***nu sunt considerate izomorfe***.

Lanțuri

Un ***lanț*** este o secvență de vârfuri și muchii de forma următoare: $v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$ unde orice muchie e_i este incidentă cu predecesorul v_{i-1} și succesorul său v_i.



Lungimea lanțului este numărul de muchii. Un lanț de lungimea ***k*** se numește ***k*-lanț**.

Cazuri particulare de lanțuri

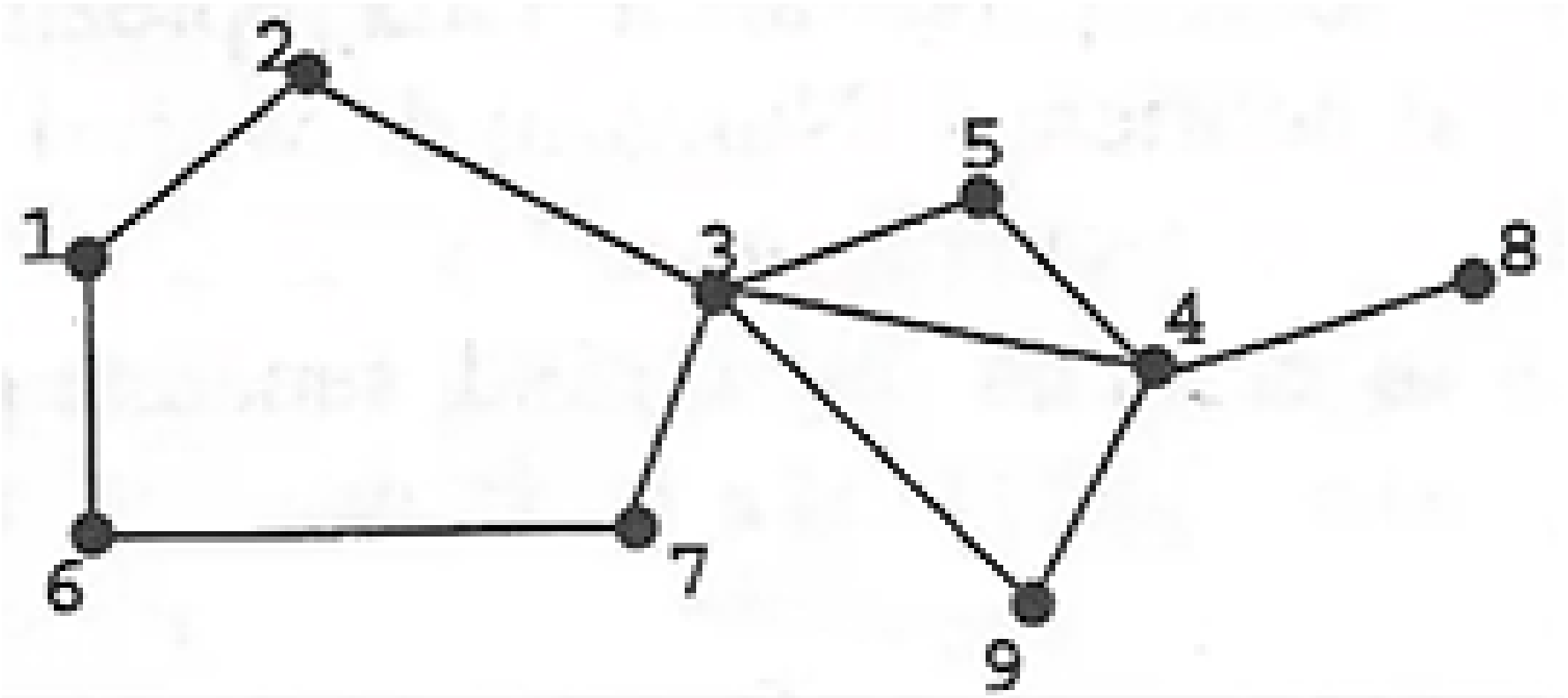
- ✓ Lanț elementar (trece prin același nod o singură dată, adică *vârfurile nu se repetă*);
- ✓ Lanț simplu (*muchiile nu se repetă*);
- ✓ Ciclu (Buclă – primul și ultimul vârf coincid);
- ✓ Ciclu elementar (*vârfurile nu se repetă*);
- ✓ Ciclu simplu (*muchiile nu se repetă*).

Dacă ***G*** și ***H*** sânt grafuri izomorfe și $W = v_0, e_1, v_1, ..., v_{n-1}, e_n, v_n$ lanț în ***G***, atunci $f(W) = f(v_0), f(e_1), f(v_1), ..., f(v_{n-1}), f(e_n), f(v_n)$ este lanț în **H**.

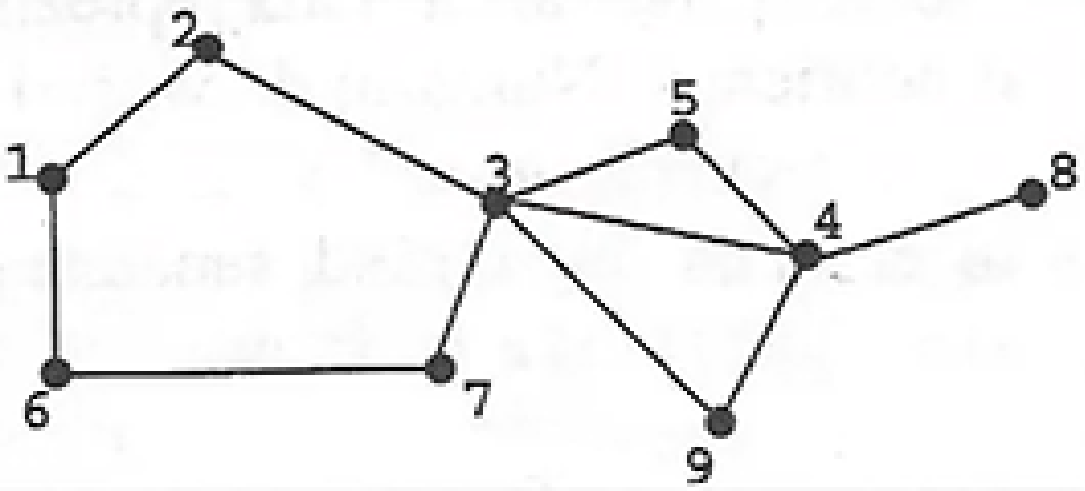
Dacă **G** și **H** sunt grafuri izomorfe și **C** este un ciclu în **G**, atunci **f(C)** este un ciclu în **H**.

Cazuri particulare de lanțuri

Exemplu: Pentru graful din figură se pot distinge mai multe lanțuri printre care: Lanțul 1 format din nodurile: L1=(1,2,3,5,4,8); L2=(1,2,3,2); L3=(9,3,5,4,3,2); L4=(6,7,3,5,4,8)



Lanțuri



Definiţie Fie un graf neorientat G=(X, U) şi L=[x1, x2, ..., xk] un lanţ al grafului G. Dacă nodurile x1, x2, ..., xk sunt distincte două câte două, atunci lanţul se numeşte ***elementar***. În caz contrar, lanţul se numeşte ***ne-elementar***.

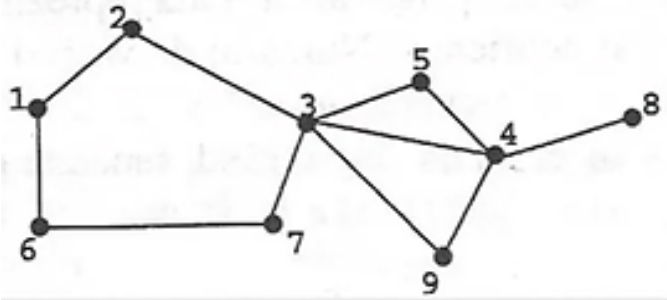
Exemplu: Lanţurile L1 şi L4 din exemplul anterior sunt elementare.

Lanţurile L2 şi L3 din exemplul anterior sunt ne-elementare.

L1=[1, 2, 3, 5, 4, 8], L2=[1, 2, 3, 2], L3=[9, 3, 5, 4, 3, 2],

L4=[6, 7, 3, 5, 4, 8]

Lanțuri



Definiţie Fie un graf neorientat G=(X, U) şi L=[x1, x2, ..., xk] un lanţ al grafului G. Dacă muchiile [x1, x2], [x2, x3], ..., [xk-1, xk] sunt distincte două câte două, atunci lanţul se numeşte ***simplu***. În caz contrar , lanţul se numeşte ***compus***.

Exemplu: Lanţurile L1, L3, L4 din exemplul anterior sunt simple. Lanţul L2 din exemplul anterior este compus.

L1=[1, 2, 3, 5, 4, 8], L2=[1, 2, 3, 2], L3=[9, 3, 5, 4, 3, 2],

L4=[6, 7, 3, 5, 4, 8]

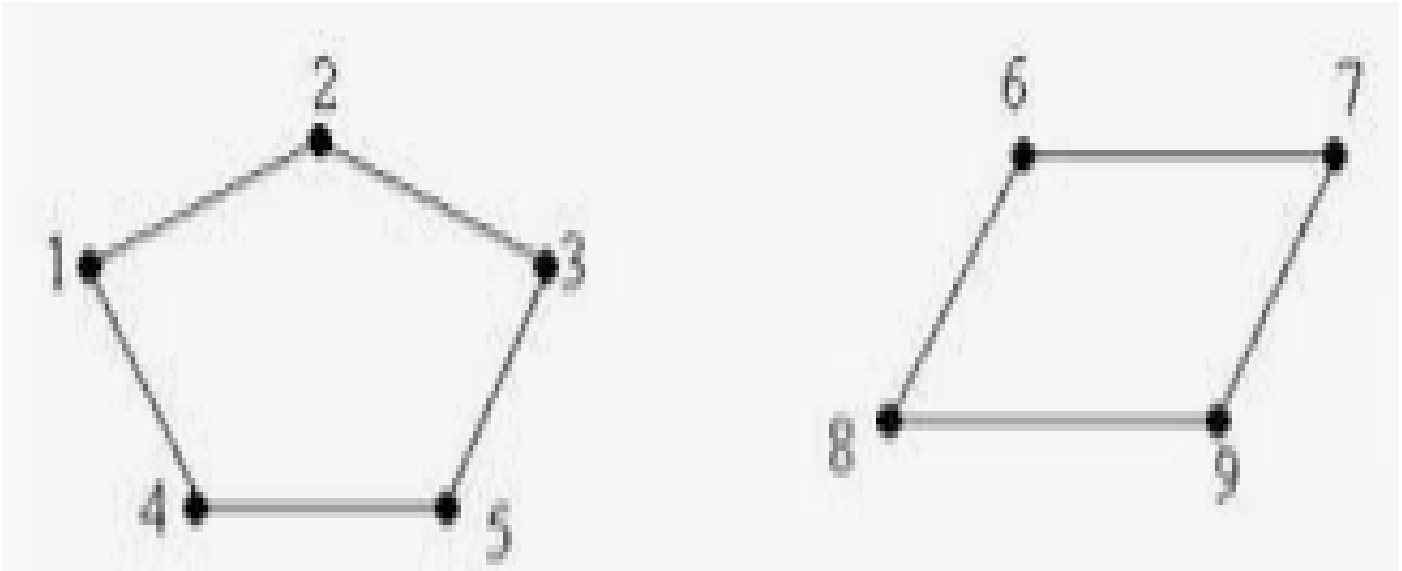
Observaţie: Orice lanţ elementar este simplu. Reciproca nu este valabilă (se poate întâmpla ca lanţul să fie simplu, dar să fie ne-elementar – vezi lanţul L3 din exemplul lanţurilor).

Conexitate

Luând oarecare două vârfuri, putem găsi cel puțin un traseu care pornește dintr-un vârf și ajunge în celălalt.

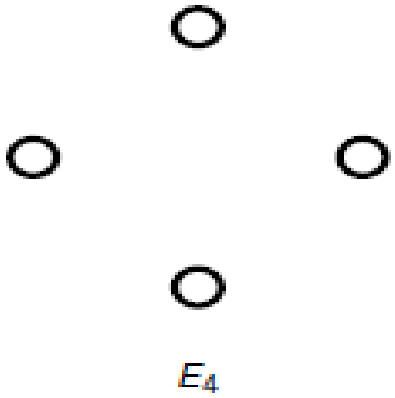
Luând oarecare două vârfuri, ele pot fi legate printr-un lanț. Nu toate grafurile sunt conexe, în schimb, putem desprinde din el „porțiuni” care, fiecare porțiune luată separat, este un graf conex.

Def: Se numește componenta conexă a grafului G(X,U), un subgraf G1=(X1,U1) a lui G, conex, cu proprietatea că nu există nici un lanț care să lege un vârf din X1 cu un vârf din X-X1.

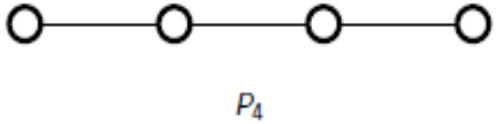


Grafuri remarcabile

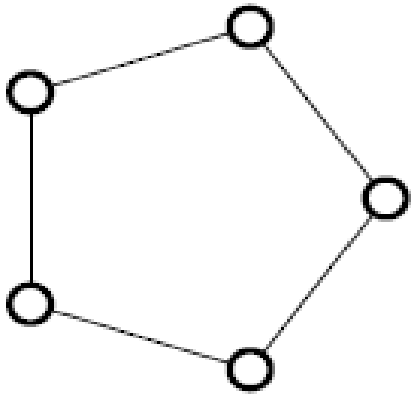
✓ Graful nul E_n, $n > 0$



✓ Graful lanț P_n, $n > 0$



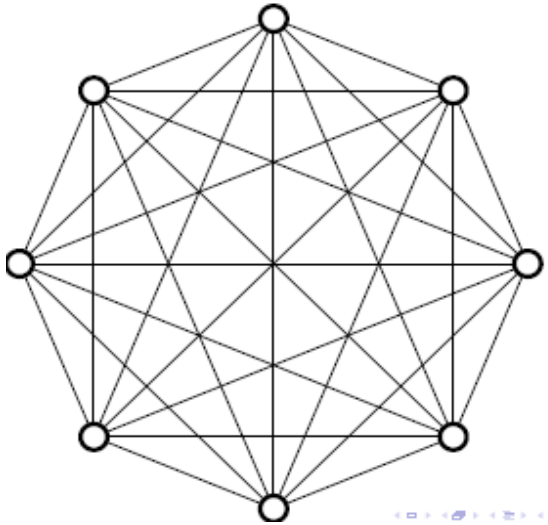
✓ Graful ciclu C_n, $n > 2$



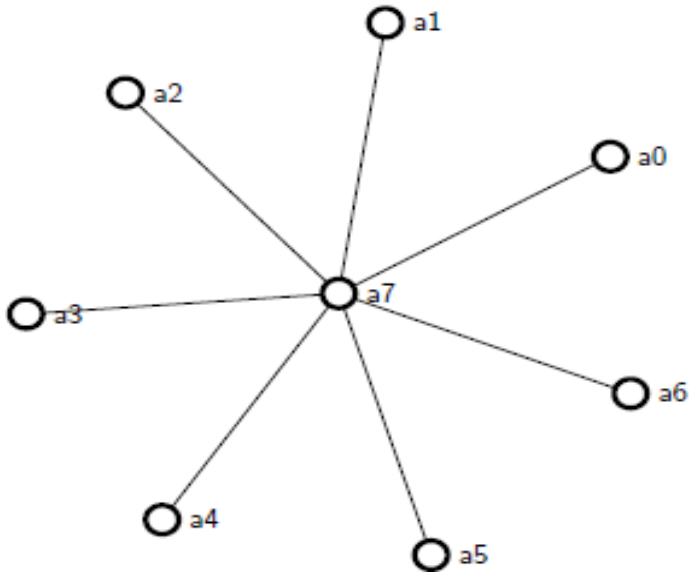
Grafuri remarcabile

✓ Graful complet K_n, $n > 2$

(*orice vârf este legat cu toate vârfurile*)

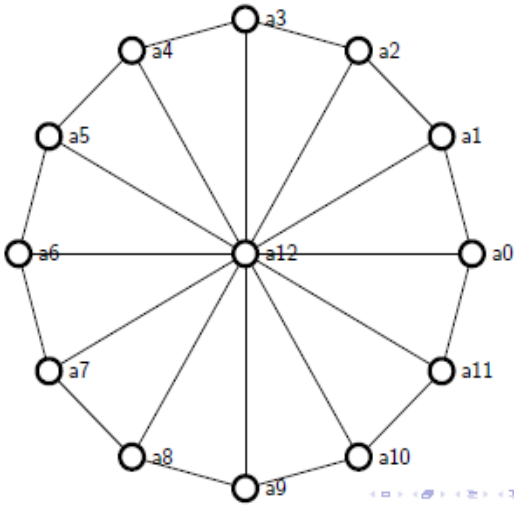


✓ Graful stea S_n, $n > 2$

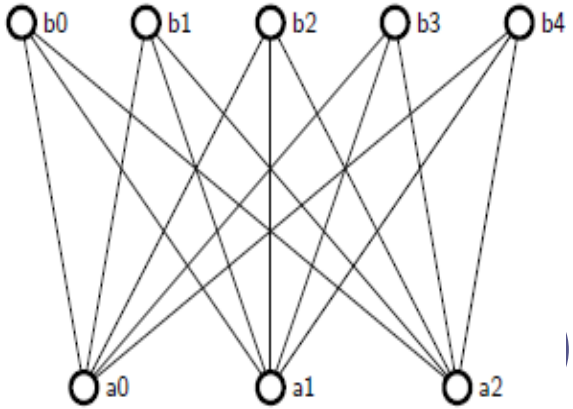


Grafuri remarcabile

✓ Graful roată W_n, $n > 4$
(*fiecare are câte trei vecini, cu excepția centrului*)



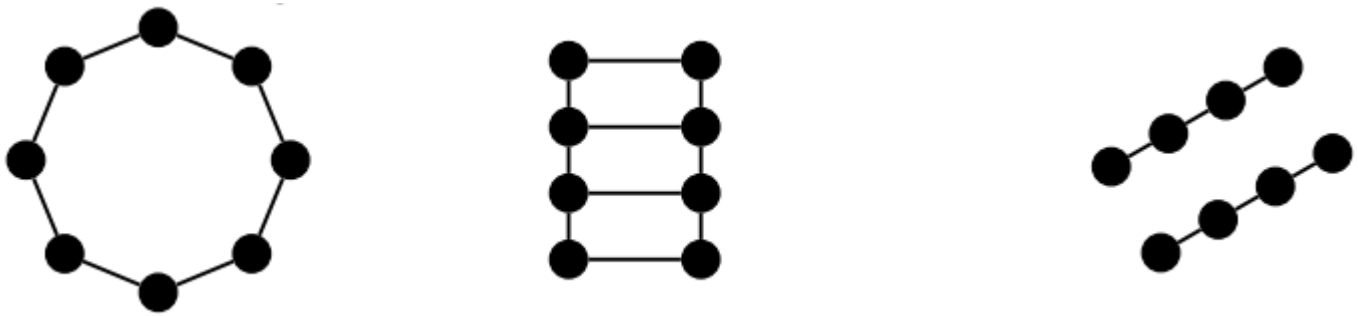
✓ Graful bipartit complet $K_{p,q}$, $p, q > 0$



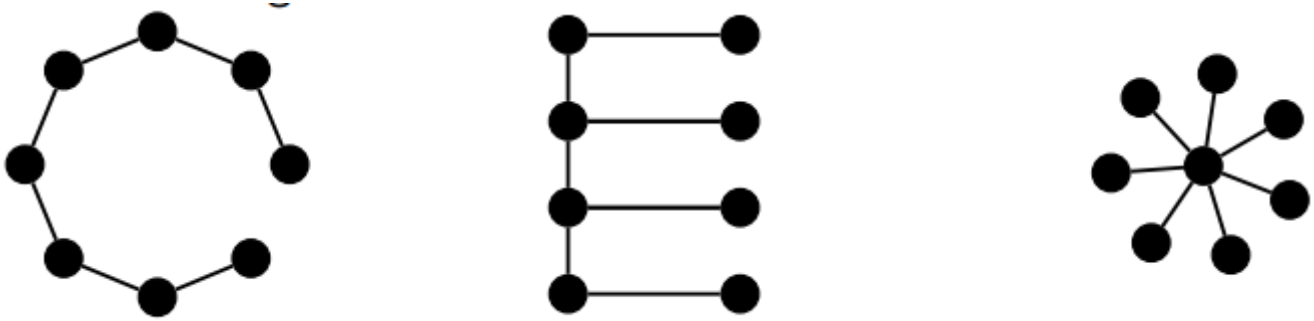
Arbori

Definiția arborilor

Un ***arbore*** este un graf neorientat conex și fără cicluri (aciclic).
Sunt următoarele grafuri arbori? (Nu)



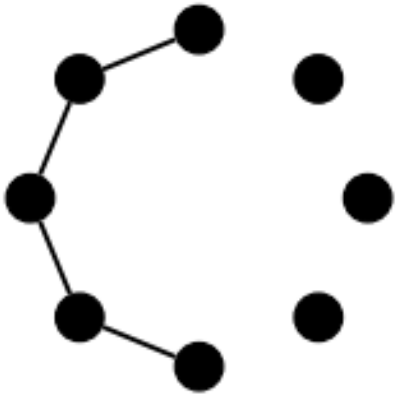
Sunt următoarele arbori grafuri? (Da)

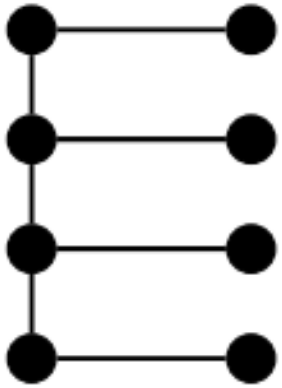


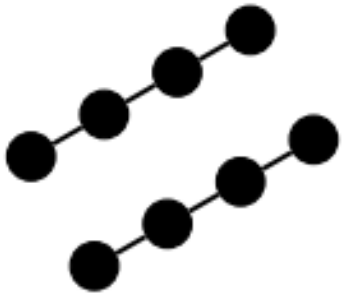
Arborescențe

Un ***graf*** fără cicluri se numește ***arborescență***.

Sunt următoarele grafuri arborescențe? (Da)







Arborescența este o reuniune de arbori disjuncți.

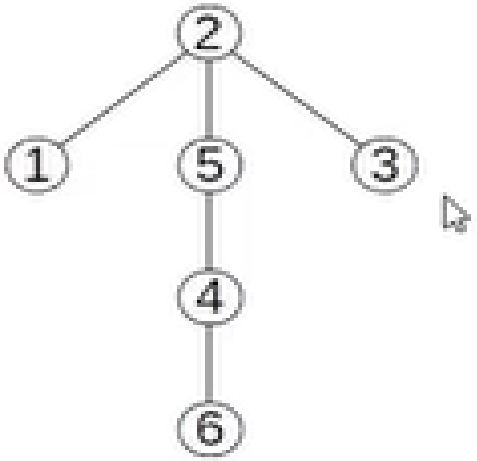
Arbori cu rădăcină

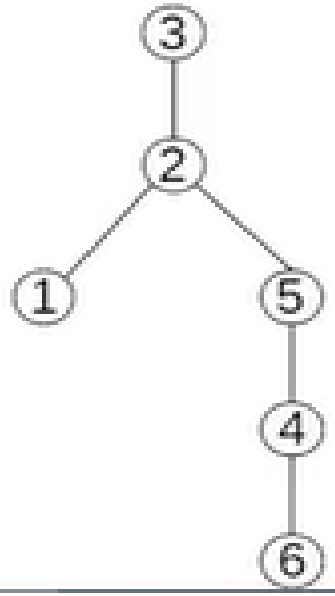
Un arbore în care un vârf a fost desemnat drept ***rădăcină*** se numește ***arbore cu rădăcină*** și este de nivelul **0**.

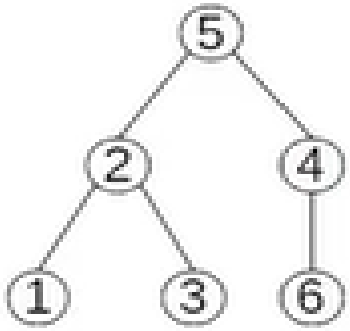
Exemple:

1. Părinte, fiu
2. Arbori binari, ternari etc.;
3. Arbori strict binari, strict ternari etc.;
4. Nivel, adâncimea arborelui.

Arbori cu rădăcină



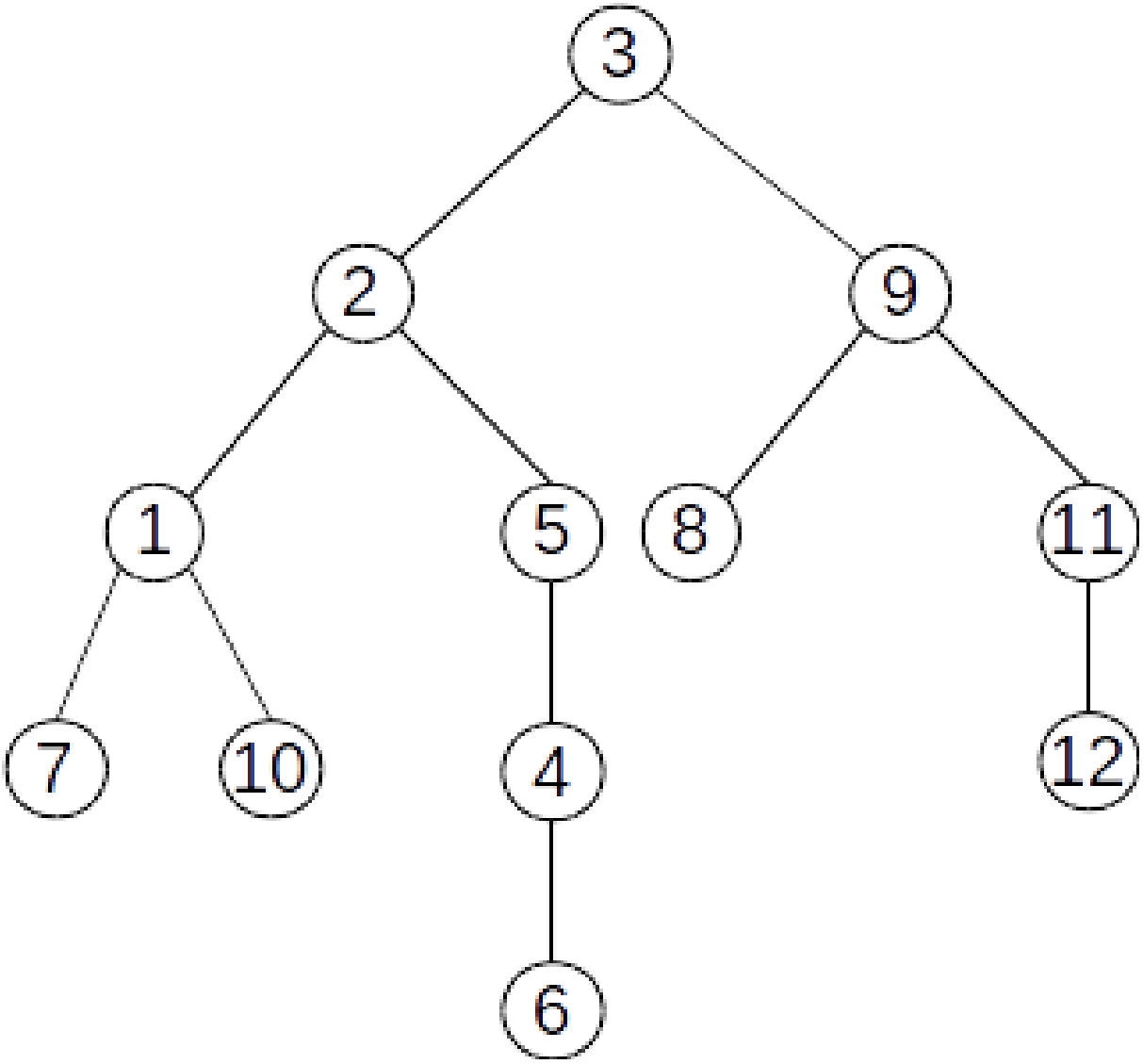


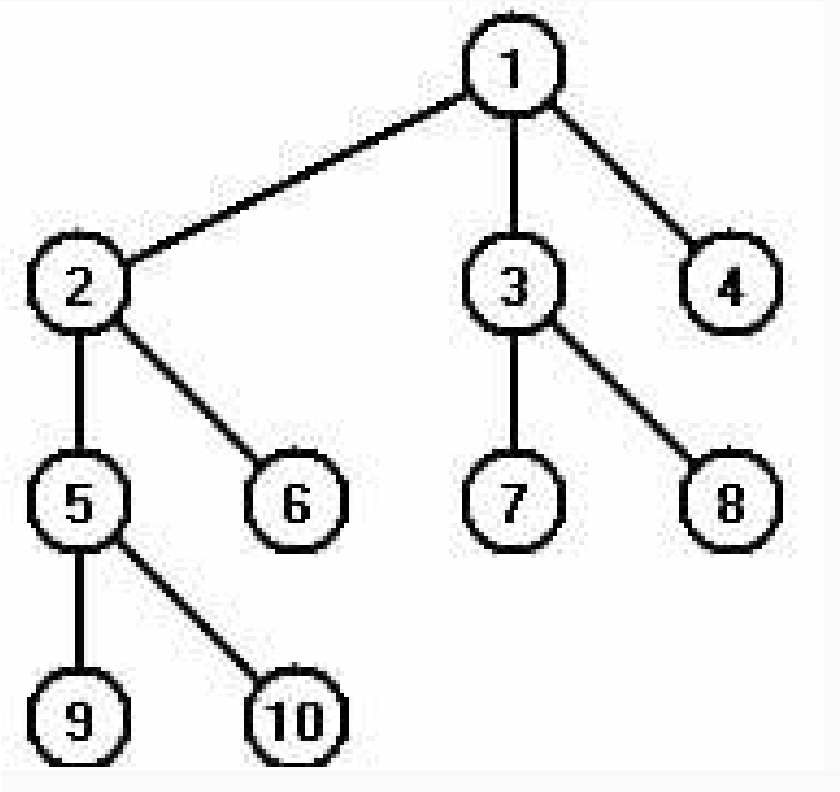


2- ***Nod rădăcină***! Nodul 1, 6 și 3 din prima imagine se numesc ***noduri terminale***, deoarece nu au descendenți, au numai ascendenți. Nodul 5 și 4 sunt ***noduri intermediare***, au și ascendenți și descendenți!

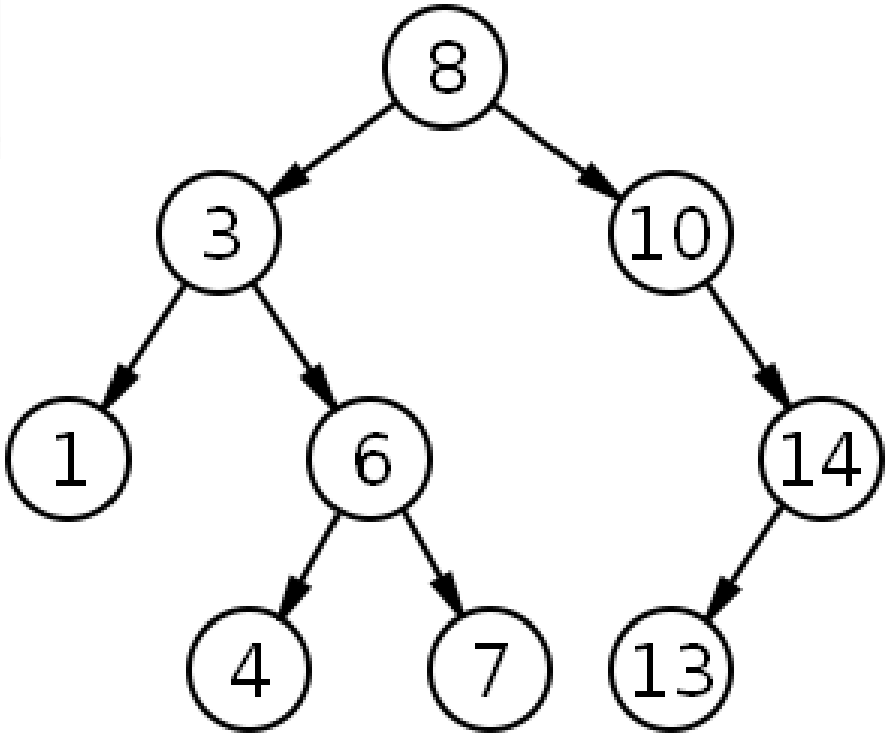
1,5,3 – descendenți direcți ai nodului 2!

Arbori cu rădăcină



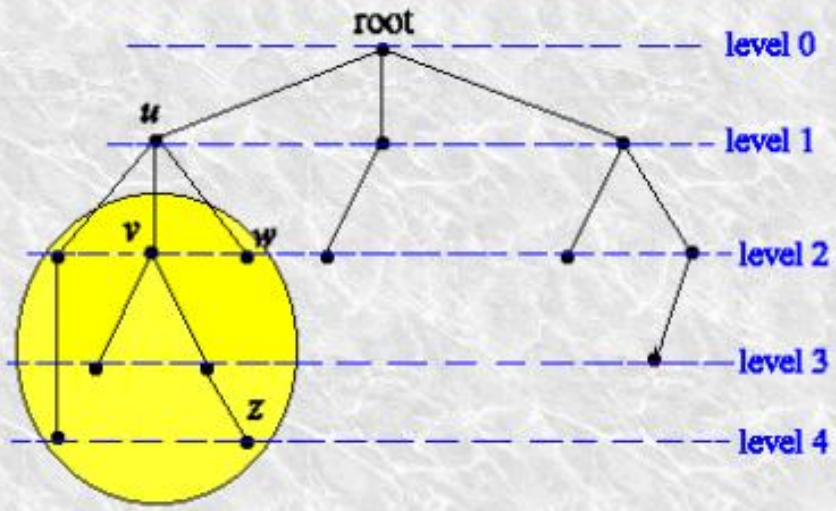


Descrieți arborii rădăcină!



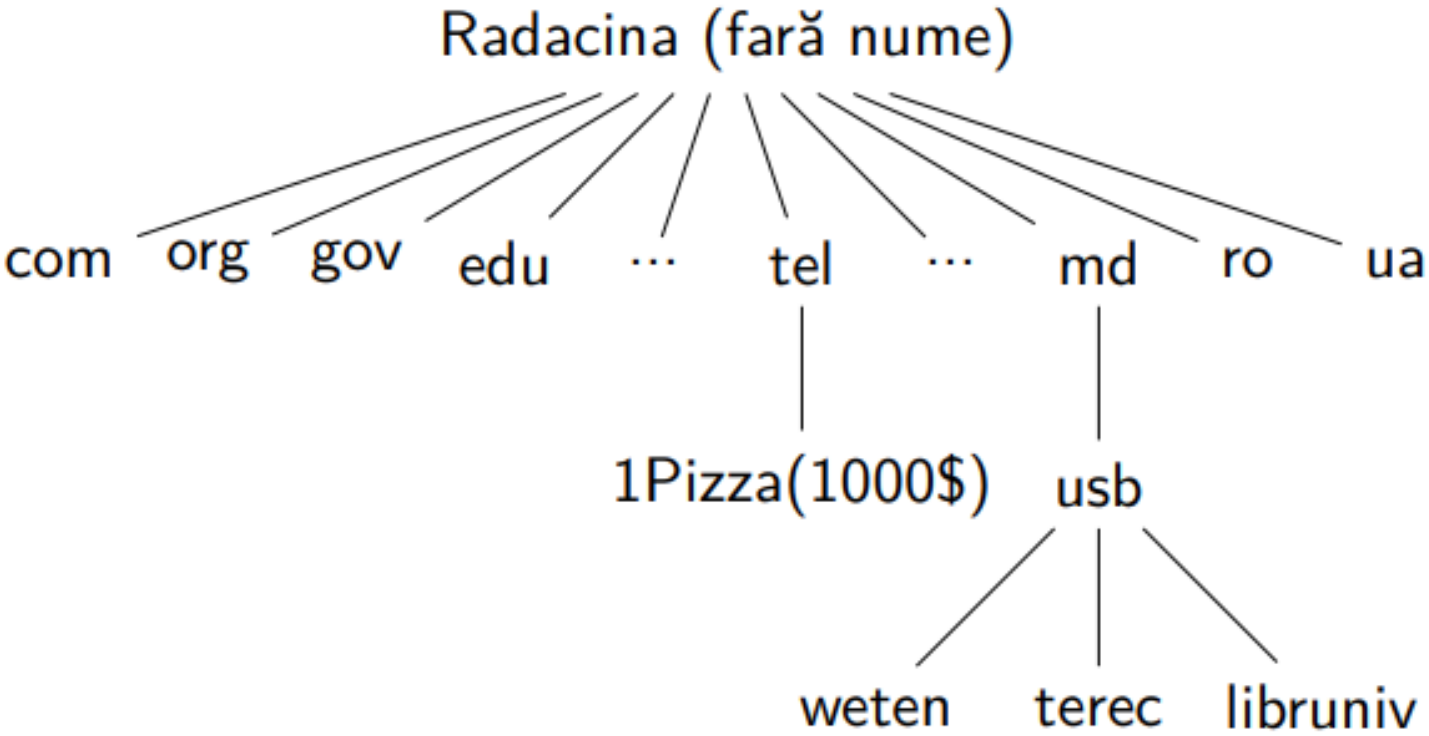
O imagine valorează cât ar valora o mie de cuvinte …

1. The **level** of u is 1, the level of z is 4.
2. The **height** of this tree is 4.
3. v and w are **children** of u, u is a **child** of the *root*.
4. u is the **parent** of v and w, the *root* is the **parent** of u.
5. v and w are **siblings**.
6. u and v are **ancestors** of z, u is an **ancestor** of w.
7. z is a **descendant** of u and v, w is a **descendant** of u.



(C) http://cpsc.ualr.edu/srini/DM/chapters/review5.4.html

Exemple de arbori I



Spațiul/sistemul numelor de domeniu

De ce „nume de domeniu” și nu simplu „nume”?

Din motiv că orice nume simbolic de site vine cu posibilitatea de ai asocia subdomenii.

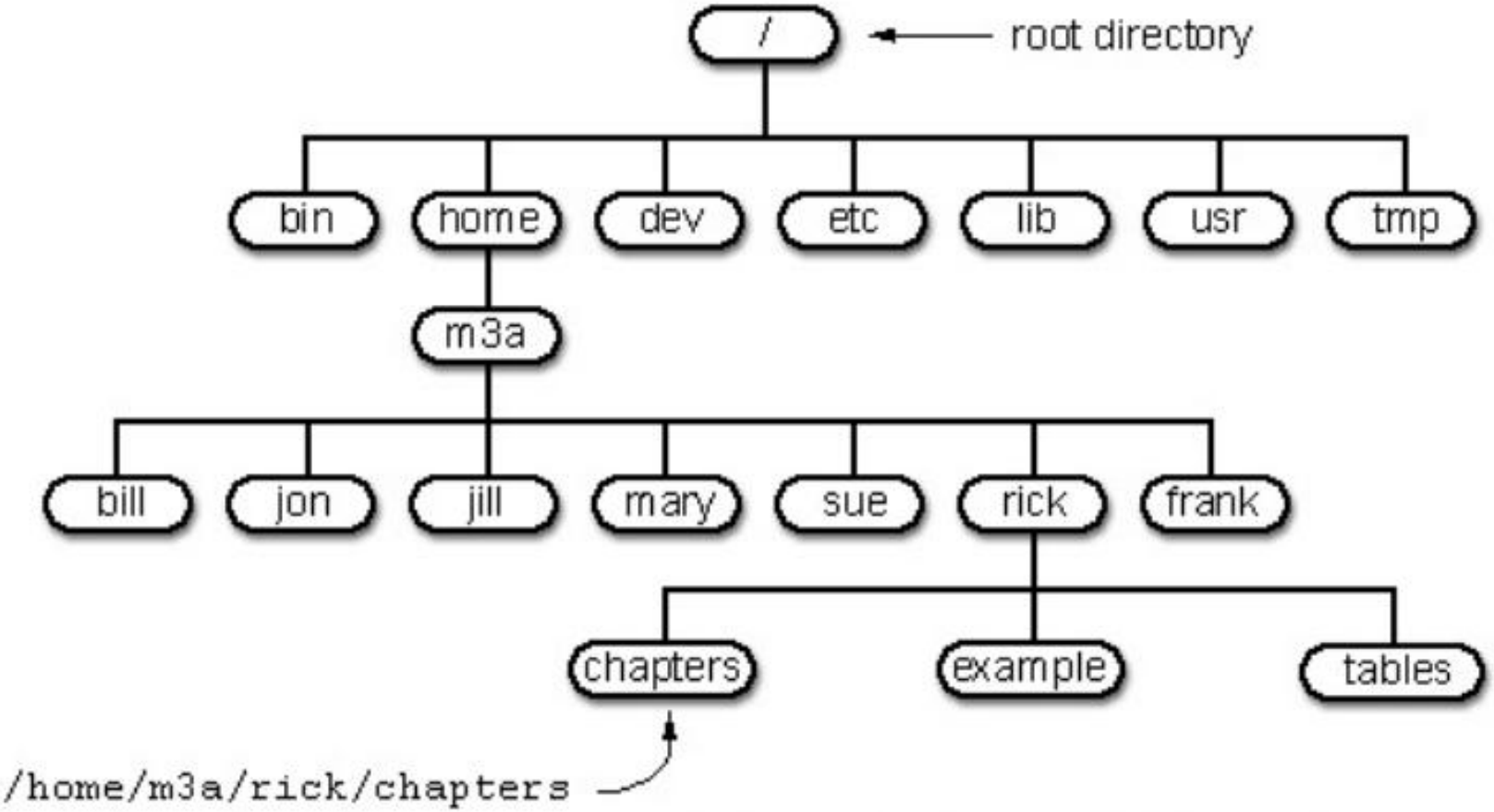
Exemple de arbori II

http://www.incognito.com/resources/nc/wp_understanding_dns.pdf

http://www.pcvr.nl/tcpip/dns_the.htm

Sistemul de directorii în SO Windows

Exemple de arbori III

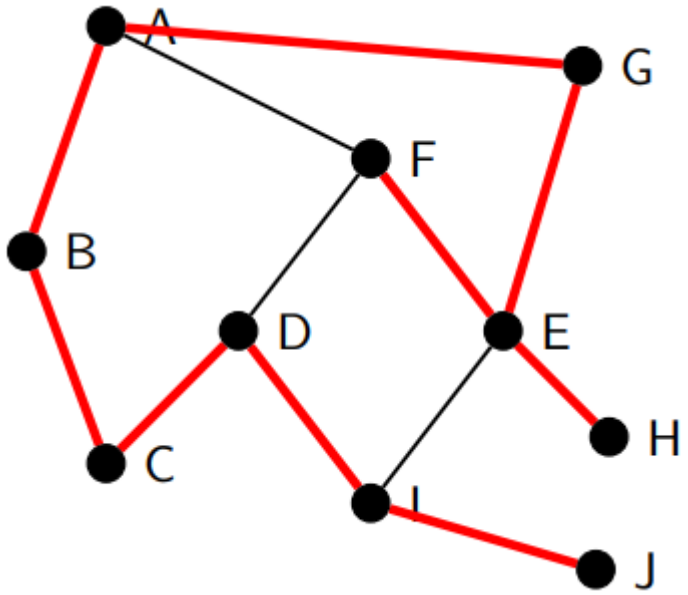


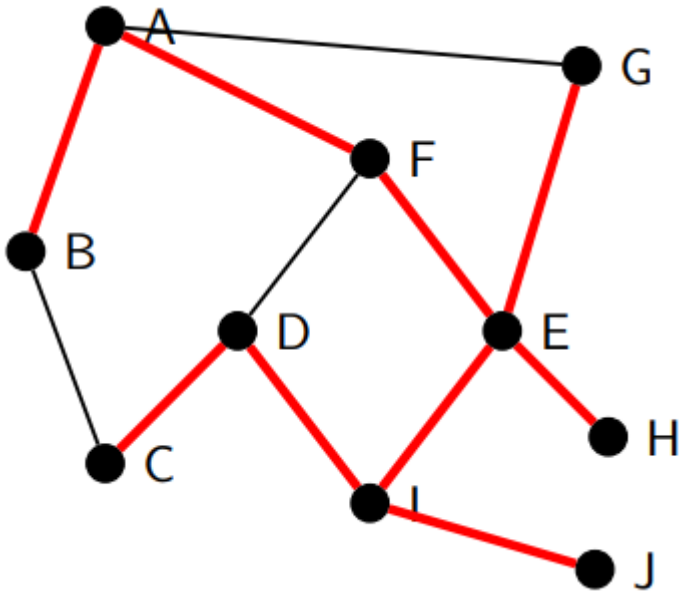
(C) http://www.ryerson.ca/acs/usersguide/learning/unix/commands/index.html

Sistemul de directorii în SO GNU/Linux

Arbori de acoperire

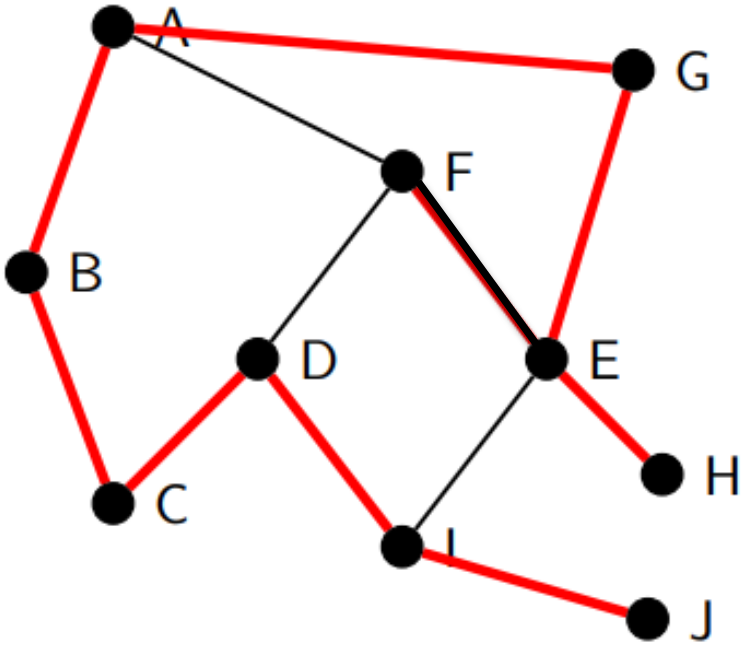
Un ***arbore de acoperire*** al unui graf conex ***G*** este un subgraf conex, fără cicluri ce conține toate vârfurile grafului **G** (Conectarea la Internet).

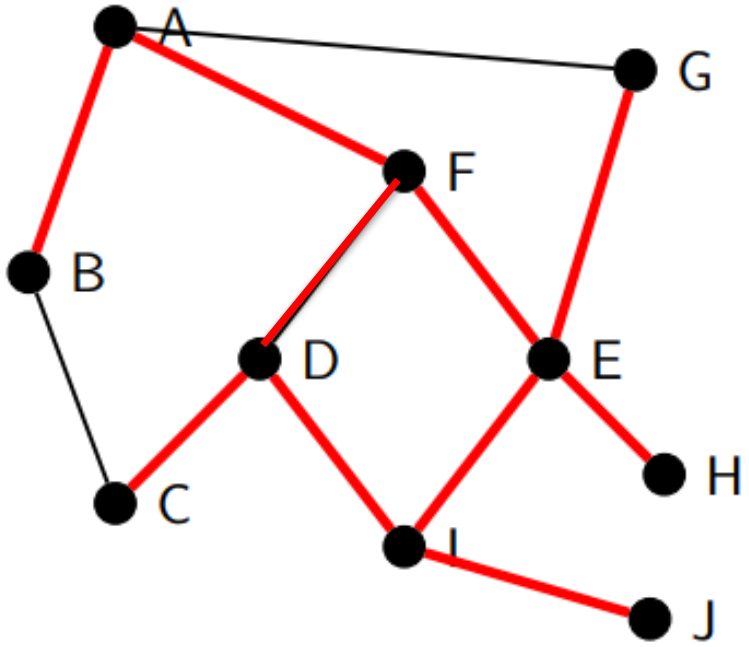




Arbori de acoperire

De ce următoarele subgrafuri ***nu sunt arbori de acoperire***?





Arbori minimi/maximi de acoperire. Grafuri ponderate

Problema determinării arborelui ***minim*** sau ***maxim de acoperire*** are sens în cazul grafurilor ponderate.

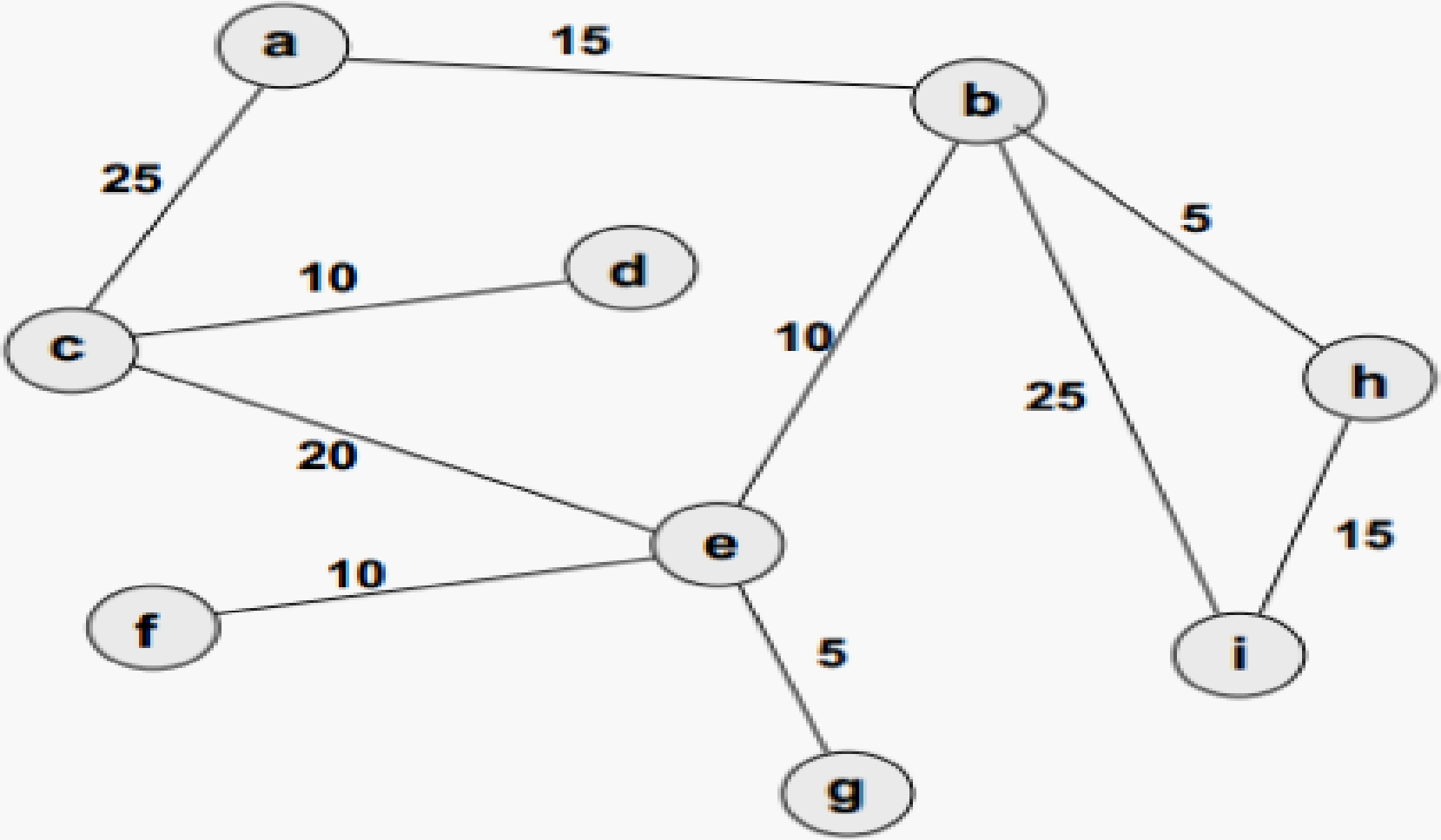
Un ***graf*** se numește ***ponderat*** dacă pe mulțimea muchiilor este definită o funcție cu valori într-o mulțime numerică. Adică fiecărei muchii i se asociază o valoare numerică.

Prezentări ale algoritmilor de determinare a arborelui de acoperire minim le găsiți pe următoarele situri:

http://algs4.cs.princeton.edu/43mst/

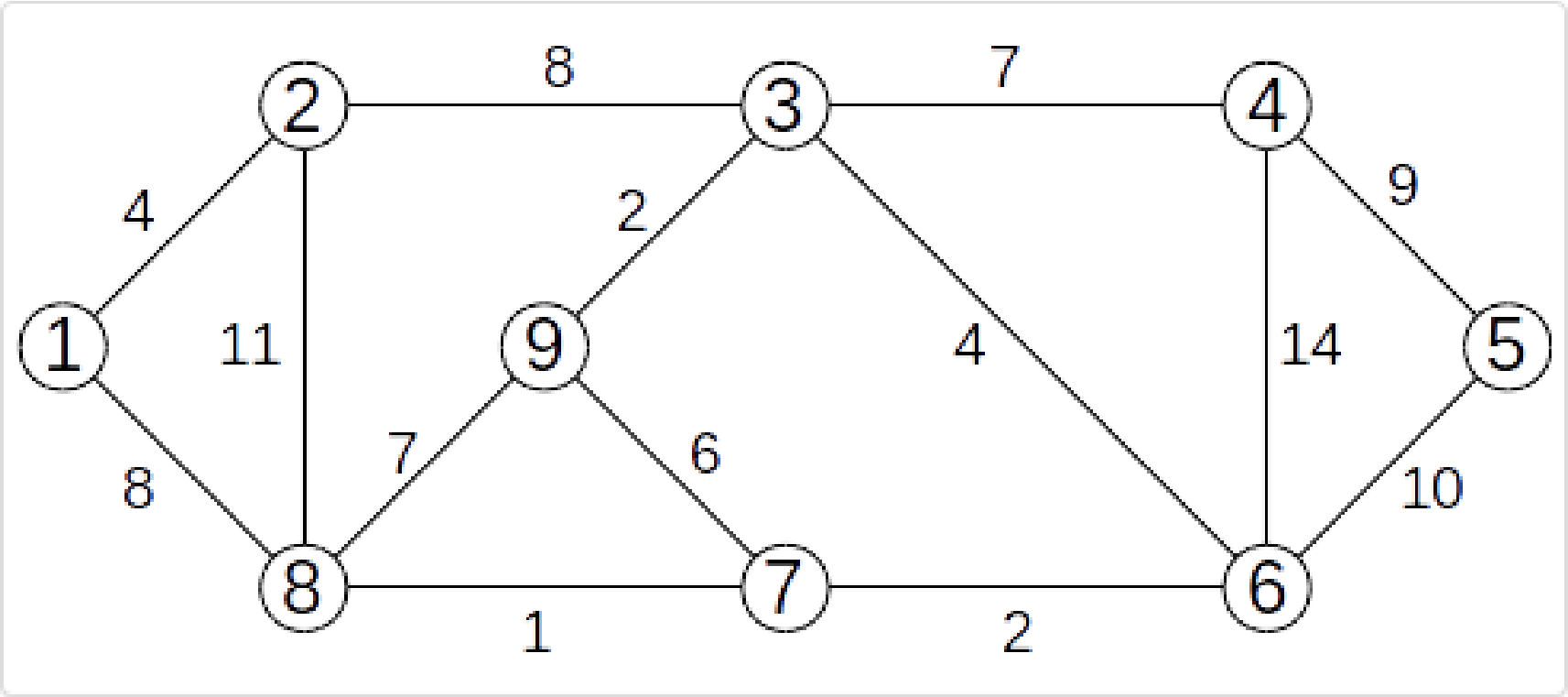
http://www.cs.princeton.edu/~wayne/kleinberg-tardos/04mst-2x2.pdf

Arbori de cost minim



Algoritmii de determinare a ***arborelui de cost minim*** sunt a lui Prim și Kruskal.

Algoritmul lui Kruskal



Considerăm un graf neorientat ponderat (cu costuri) conex **G**. Se numește **arbore parțial** un graf parțial al lui **G** care este arbore. Se numește **arbore parțial de cost minim** un arbore parțial pentru care suma costurilor muchiilor este minimă.

Dacă graful nu este conex, vorbim despre o **pădure parțială de cost minim**.

Algoritmul lui Kruskal permite determinarea unui arbore parțial de cost minim (APM) într-un graf ponderat cu **N** noduri.

Descrierea algoritmului lui Kruskal

Pentru a determina APM se pleacă de la o pădure formată din **N** subarbori. Fiecare nod al grafului reprezintă inițial un subarbore. Aceștia vor fi reuniți succesiv prin muchii, până când se obține un singur arbore (dacă graful este conex) sau până când acest lucru nu mai este posibil (dacă graful nu este conex).

Algoritmul:

- ✓ se ordonează muchiile grafului crescător după cost;
- ✓ se analizează pe rând muchiile grafului, în ordinea crescătoare a costurilor;
- ✓ pentru fiecare muchie analizată:
  - dacă extremitățile muchiei fac parte din același subarbore, muchia se ignoră;
  - dacă extremitățile muchiei fac parte din subarbori diferiți, aceștia se vor reuni, iar muchia respectivă face parte din APM.

Descrierea algoritmului lui Kruskal

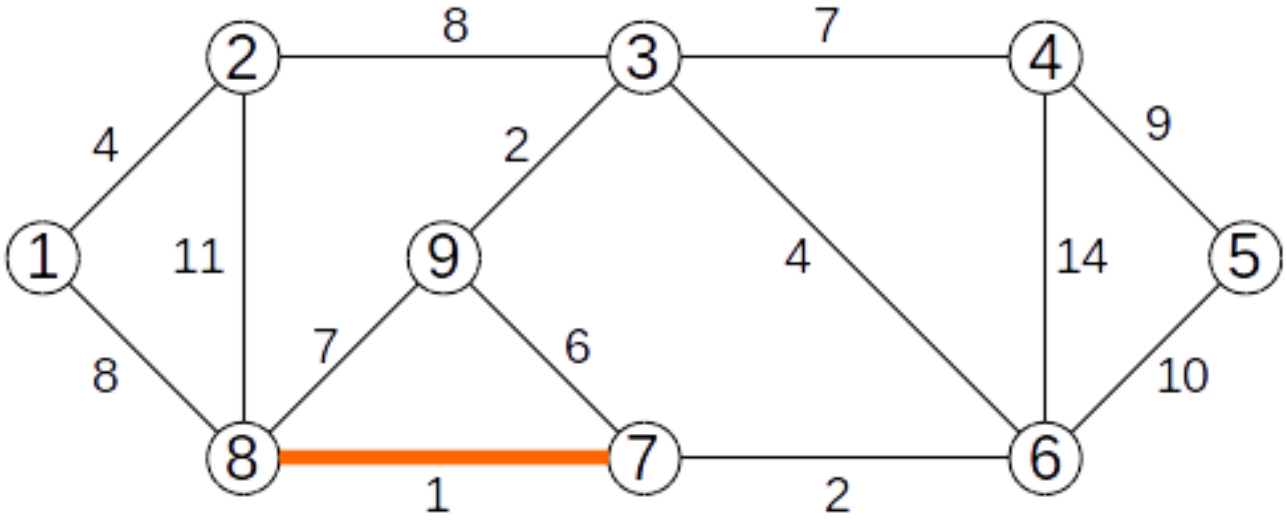
Principala dificultate în algoritmul descris mai sus este stabilirea faptului că extremitățile muchiei curente fac sau nu parte din același subarbore. În acest scop vom stabili pentru fiecare subarbore un nod special, numit ***reprezentant al (sub)arborelui*** și pentru fiecare nod din graf vom memora reprezentantul său (de fapt al subarborelui din care face parte acesta) într-un tablou unidimensional.

Pentru a stabili dacă 2 noduri fac sau nu parte din același subarbore se verifică dacă ele au același reprezentant. Pentru a reuni doi subarbori se înlocuiesc pentru toate nodurile din subarborele **B** cu reprezentantul subarborelui **A**, care sunt simple dar lente.

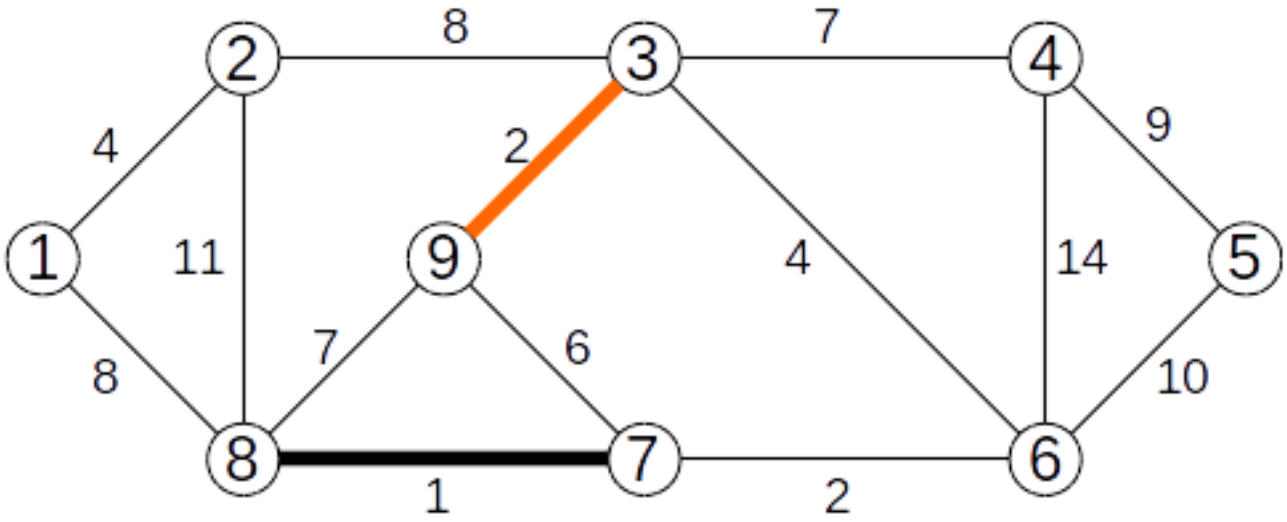
Pentru o implementare mai eficientă a algoritmului se folosește conceptul de **Padure de mulțimi disjuncte**.

Muchiile se vor analiza în ordinea următoare:

1.

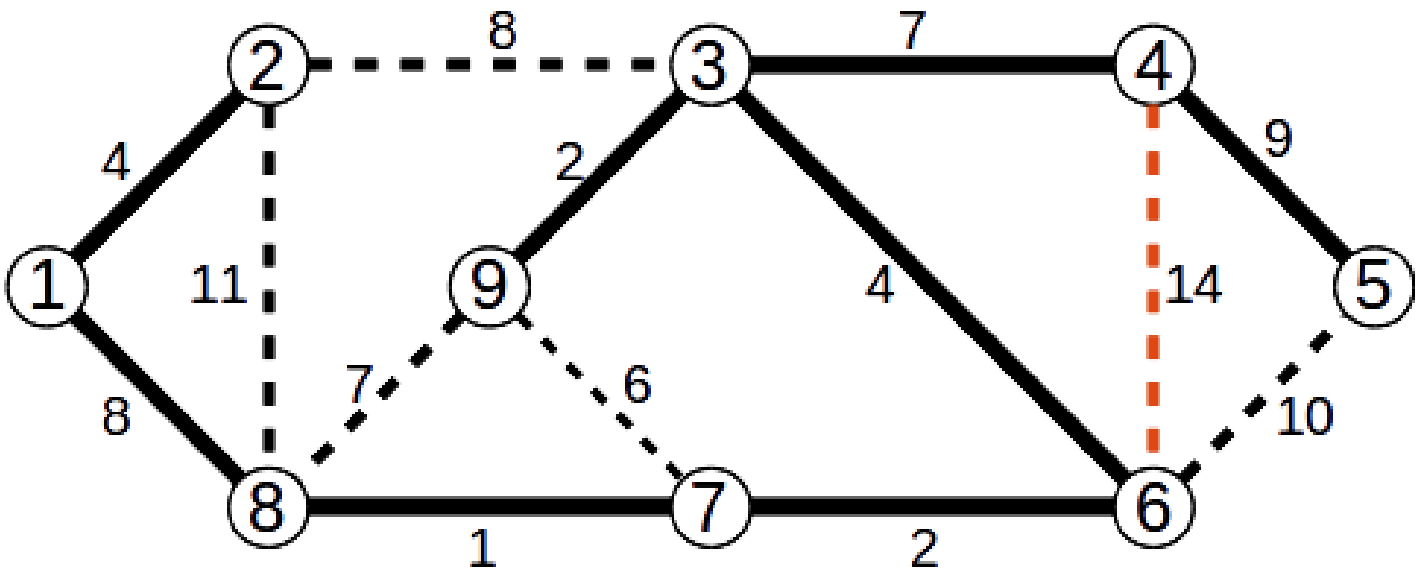


2.

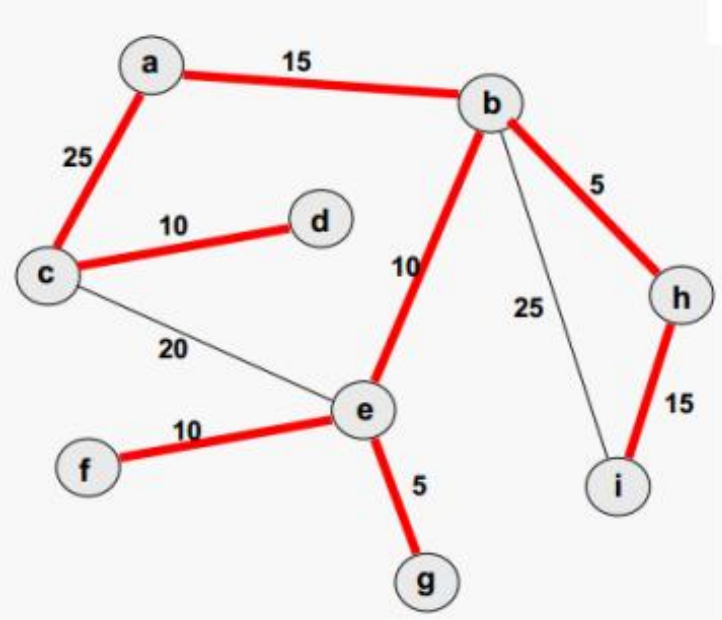


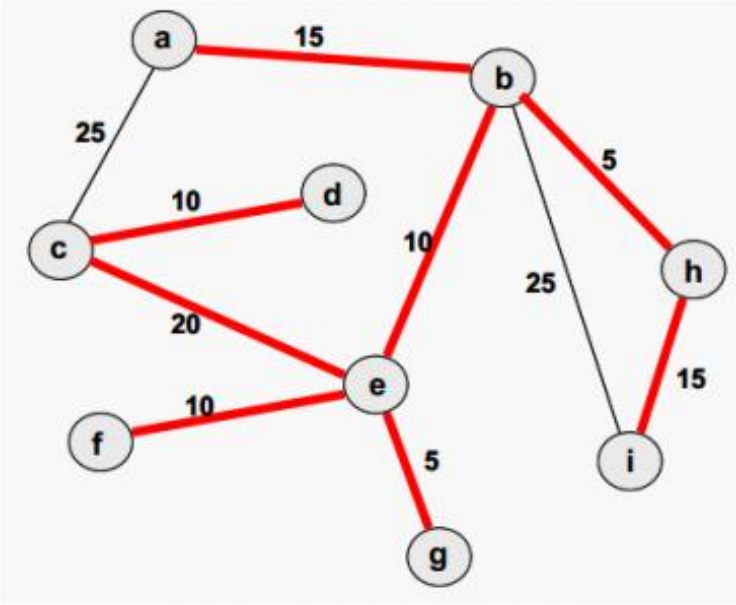
Muchiile se vor analiza în ordinea următoare:

14.



Arbori de cost minim





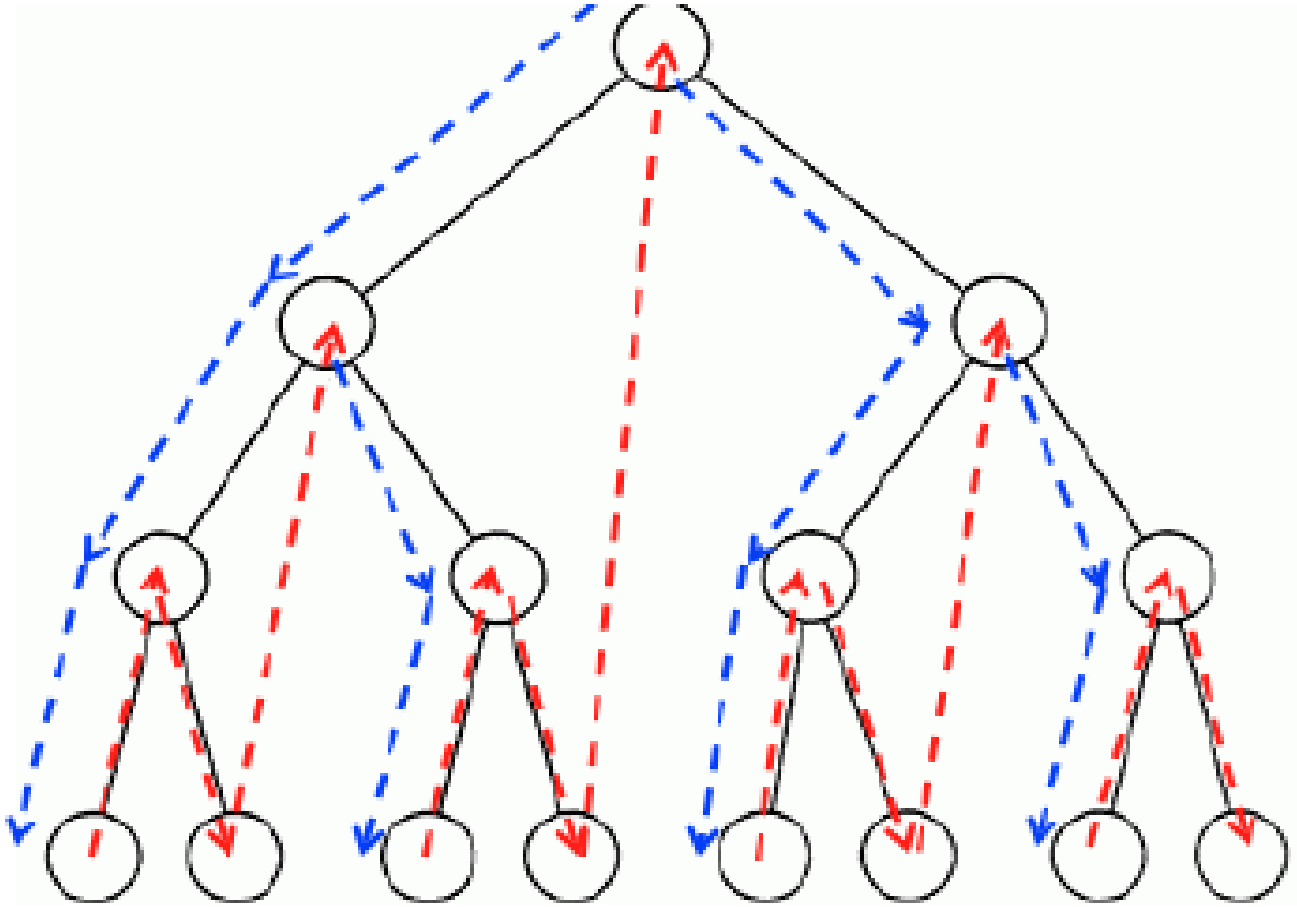
Parcurgerea/traversarea arborilor

- ✓ ***Preordine*:** se parcurge mai întâi rădăcina, apoi consecutiv primul, al doilea, ..., ultimul subarbore;
- ✓ ***Postordine*:** se parcurg consecutiv primul, al doilea, ..., ultimul subarbore și în ultimul rând rădăcina;
- ✓ ***Inordine*:** Se parcurge mai întâi primul subarbore, apoi rădăcina, apoi al doilea, al treilea ... ultimul subarbore. În cazul arborilor binari această traversare se numește ***simetrică*** întrucât se parcurge mai întâi subarborele stâng, apoi vârful și numai după aceasta subarborele drept.

În cazul arborilor binari aceste parcurgeri se fac din raportul poziției rădăcinii față de subarborele stâng și drept (rsd, sdr, srd).

http://www.mhhe.com/math/advmath/rosen/r5/instructor/shared/transparencies5/data/rdm9_4.pdf
http://workshop.rosedu.org/wiki/_media/sesiuni/algoritmica/algoritmi_pe_grafuri.pdf

Parcurgerea/traversarea arborilor



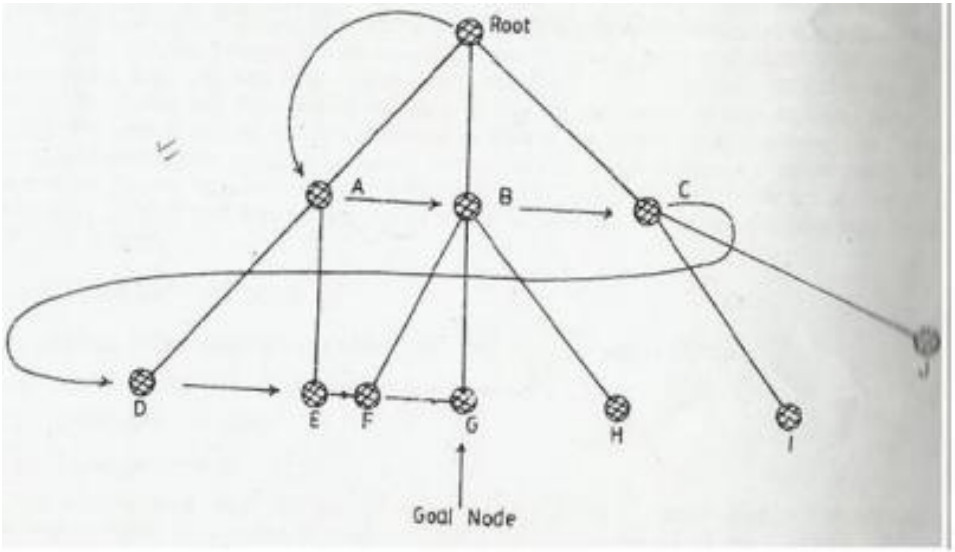
Parcurgerea/traversarea arborilor

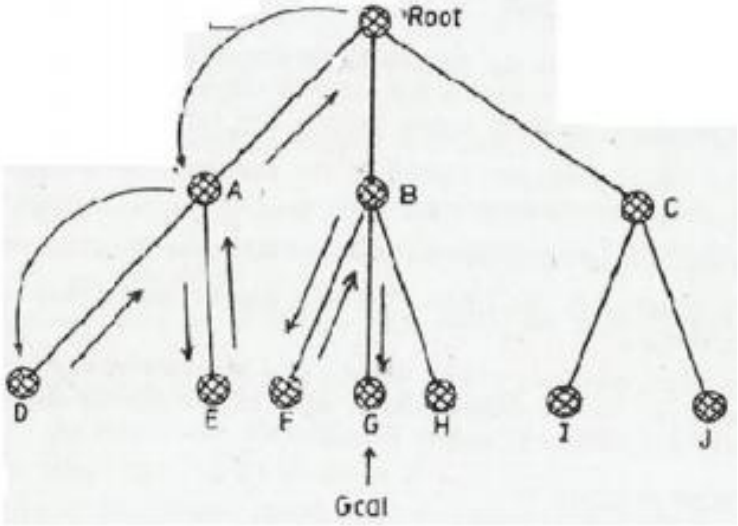
Parcurgerea în lățime: se tipăreşte mai întâi informaţia din nodul rădăcină, după care sunt prelucrate, de la stânga la dreapta, nodurile aflate pe primul nivel, apoi pe cel de-al doilea ș. a. m. d.

Parcurgerea în adâncime: fiii unui nod sunt vizitați tot de la stânga spre dreapta, dar trecerea de la nodul curent la fratele din dreapta se realizează numai după vizitarea tuturor descendenților nodului curent, deci a întregului subarbore nodului respectiv.

Parcurgerea/traversarea arborilor

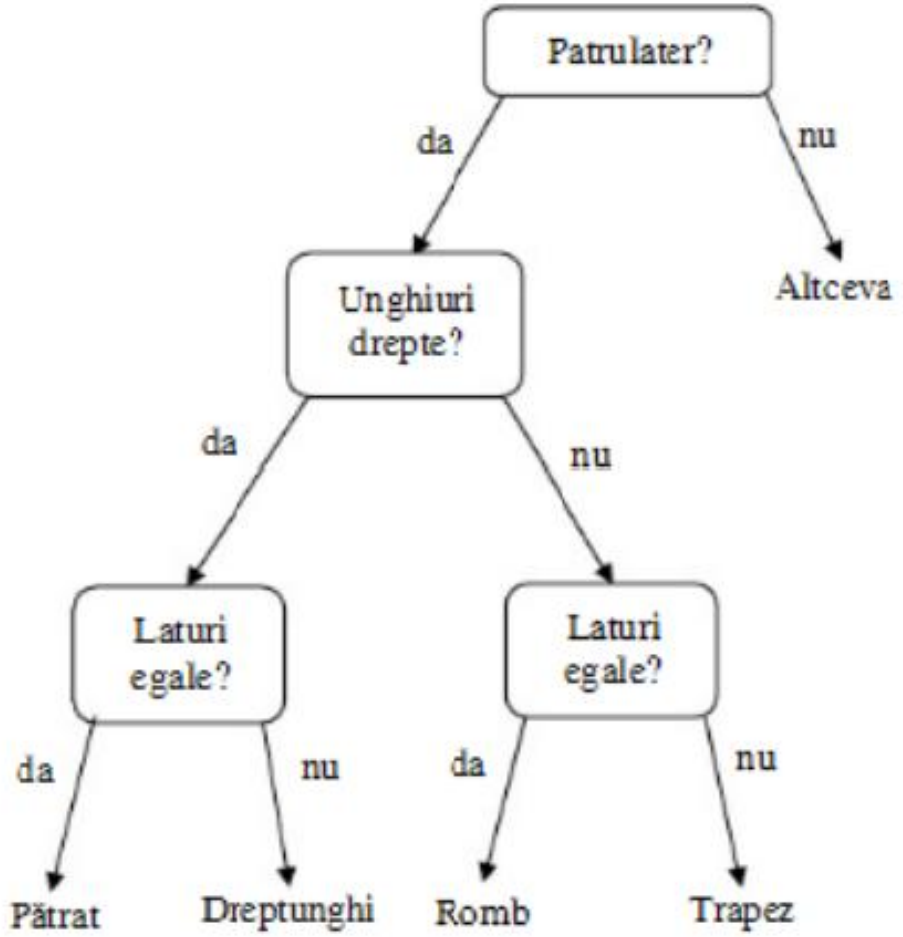
Următoarele imagini reprezintă exemple de parcurgere a arborilor. Care imagine corespunde parcurgerii ***în adâncime*** și care imagine corespunde ***parcurgerii în lățime***?



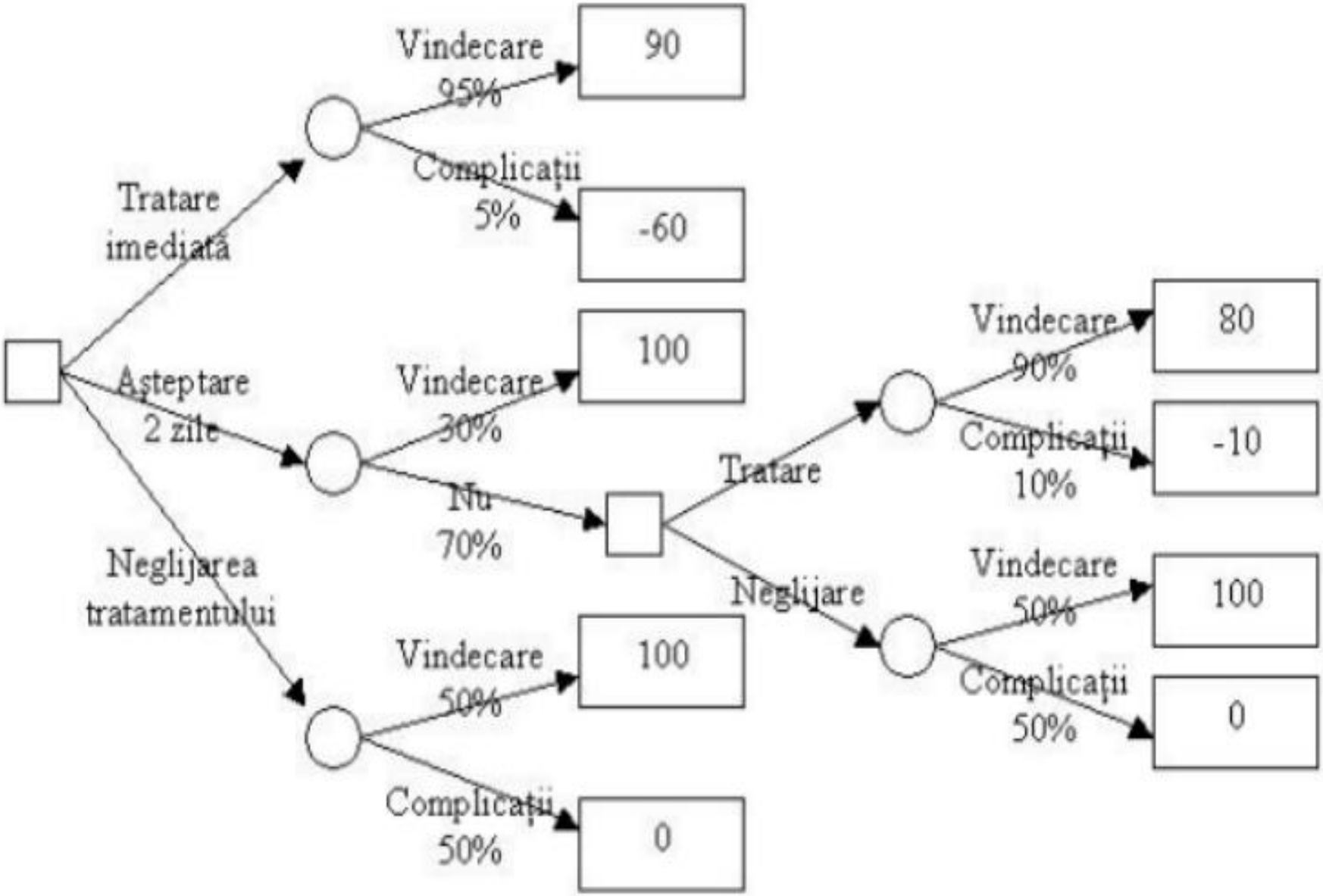


Arbore de decizie

Arborele de decizie este un arbore în care vârfurile interne reprezintă acțiuni, muchiile – rezultatul acțiunii, iar frunzele – rezultatele finale.



Arbore de decizie pentru situația tratării unei gripe



MULȚUMIM PENTRU ATENȚIE!