GRAFURI

Colegiul Național “Petru Rareş” Suceava
prof. Marius UDUDEC

Sumar

Competenţe 5
I. Noţiunea de graf 6
1. Terminologie 6
2. Clasificarea grafurilor 8
II. Grafuri neorientate 9
1. Terminologie 9
2. Gradul unui nod 12
3. Noţiunile de lanţ şi ciclu 14
4. Reprezentarea grafurilor 16
5. Graf ponderat 23
6. Graf parţial 24
7. Subgraf 25
8. Tipuri speciale de grafuri 26
9. Conexitatea grafurilor 31
10. Parcurgerea grafurilor 33
III. Grafuri orientate 36
1. Terminologie 36
2. Gradul unui vârf 42

3. Noţiunile de lanţ, drum şi circuit 42
4. Reprezentarea grafurilor 45
5. Graf ponderat 52
6. Graf parţial 53
7. Subgraf 54
8. Tipuri speciale de grafuri 55
9. Conexitatea grafurilor 59
10. Parcurgerea grafurilor 63
IV. Arbori 66
1. Terminologie 66
2. Reprezentarea arborilor cu rădăcină 68
3. Arbori binari 73
4. Reprezentarea arborilor binari 75
5. Parcurgerea arborilor binari 78
6. Tipuri speciale de arbori binari 82
7. Arbore parţial 90
8. Arbore parţial de cost minim 91

Aplicaţii 92
Bibliografie şi webografie 93

Competenţe generale

- *identificarea datelor care intervin într-o problemă şi aplicarea algoritmilor fundamentali de prelucrare a acestora*

Competenţe specifice

- *transpunerea unei probleme din limbaj natural în limbaj de grafuri, folosind corect terminologia specifică*
- *analizarea unei probleme în scopul identificării datelor necesare şi alegerea modalităţilor adecvate de structurare a datelor care intervin într-o problemă*
- *descrierea unor algoritmi simpli de verificare a unor proprietăţi specifice grafurilor*
- *descrierea algoritmilor fundamentali de prelucrare a grafurilor şi implementarea acestora într-un limbaj de programare*

I. Noţiunea de graf

1. Terminologie

Definiţie:

Se numeşte graf (G) o pereche ordonată de mulţimi (X,U), unde X este o mulţime finită şi nevidă, iar U o mulţime de perechi formate cu elemente distincte din mulţimea X (familie de submulţimi cu două elemente din mulţimea X).

Elementele mulţimii X se numesc noduri sau vârfuri. Mulţimea X se numeşte mulţimea nodurilor sau mulţimea vârfurilor grafului G.

Mulţimea X este de forma:

$$X=\{x_1,\ x_2,\ x_3,\ ...,\ x_n\}$$

unde x_i reprezintă nodul/vârful i al grafului G care are n noduri.

Elementele mulţimii U sunt perechi de noduri/vârfuri, adică submulţimi cu două elemente distincte din mulţimea X şi se notează cu u_k. Mulţimea U se numeşte mulţimea muchiilor sau mulţimea arcelor grafului G.

Mulţimea U este de forma:

$$U=\{u_1,\ u_2,\ u_3,\ ...,\ u_m\}$$

unde u_i este definit de perechea de forma $\{x_i,x_j\}$ cu x_i şi $x_j \in X$ şi $x_i \neq x_j$.

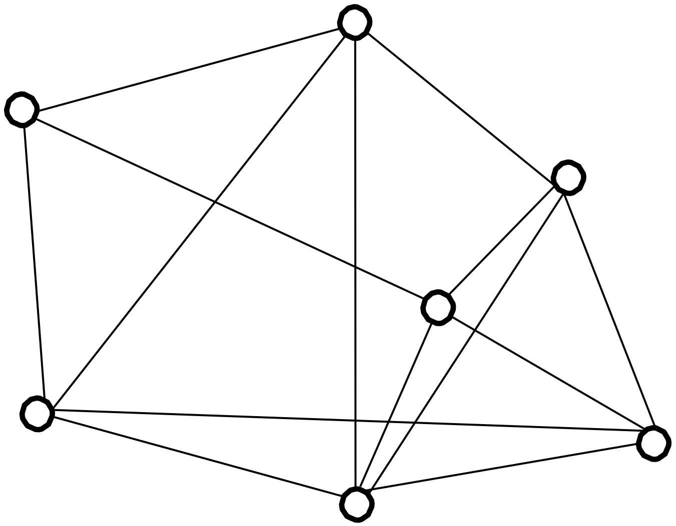
Ordinul grafului (notat cu n) reprezintă numărul de noduri/vârfuri ale grafului:

n=cardinalul mulțimii X(G).

Un **graf** poate fi reprezentat sub forma unei figuri geometrice alcătuite din:

• ***puncte*** (care corespund *nodurilor* sau *vârfurilor*) și

• din ***linii*** drepte sau curbe care unesc aceste puncte (care corespund *muchiilor* sau *arcelor*).



2. Clasificarea grafurilor

Criteriul de clasificare folosit este proprietatea de simetrie a mulţimii **U**.

Mulţimea **U** are *proprietatea de simetrie:* pentru orice pereche de noduri /vârfuri `{x,y}`, dacă `{x,y}∈U`, atunci şi `{y,x}∈U`.

În funcţie de proprietatea de simetrie, grafurile se clasifică în:

- *grafuri neorientate*. Un graf *G=(X,U)* este *graf neorientat* dacă mulţimea *U* are proprietatea de simetrie. Mulţimea U este formată din *perechi neordonate* `[x,y]`.
- *grafuri orientate*. Un graf *G=(X,U)* este un *graf orientat* dacă mulţimea *U* nu are proprietatea de simetrie. Mulţimea *U* este formată din *perechi ordonate* `(x,y)`.

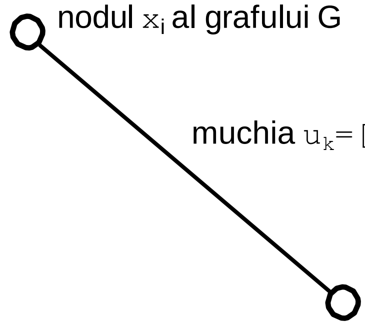
II. Grafuri neorinetate

1. Terminologie

Definiţie

Se numeşte graf neorientat, notat $G=(X,U)$, o pereche ordonată de mulţimi (X,U), unde:
- X este o mulţime finită şi nevidă de elemente, numită *mulțimea noduri*;
- U este o mulţime de perechi neordonate din mulţimea X, numită *mulțimea muchiilor*.

- numim noduri adiacente orice pereche de noduri care formează o muchie $[x,y] \in U$;
- nodurile vecine unui nod x sunt toate nodurile y care sunt adiacente cu el;
- se numeşte nod extrem al unei muchii oricare dintre cele două noduri care se găsesc la capătul muchiei;
- se numesc muchii incidente două muchii u_i şi u_j care au o extremitate comună;
- muchia $[x_i,x_j]$ este aceeaşi cu muchia $[x_j,x_i]$;



nodul x_i al grafului G

muchia $u_k=[x_i,x_j]$ sau $u_k=[x_j,x_i]$ a grafului G

nodul x_j al grafului G

Exemplu

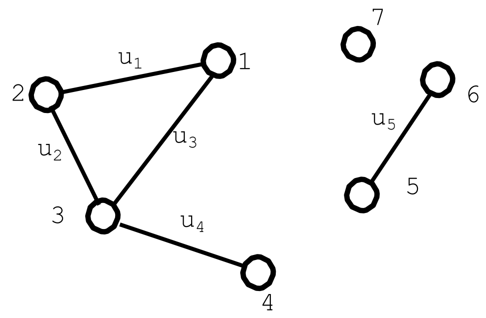
Fie graful G=(X,U).

n=7 numărul de noduri (ordinul grafului G)

m=5 - numărul de muchii

X={1, 2, 3, 4, 5, 6, 7} - mulţimea nodurilor

U={u_1,u_2,u_3,u_4,u_5}={[1,2],[2,3],[1,3],[3,4],[5,6]} - mulţimea muchiilor



- nodul 1 este adiacent cu nodurile 2 şi 3;
- vecinii nodului 1 sunt nodurile 2 şi 3;
- extremităţile muchiei u_1 sunt nodurile 1 şi 2;
- muchia u_1 este incidentă cu muchiile u_2 şi u_3;`

Teoremă

Într-un graf G=(X,U) cu n noduri , numărul total de grafuri neorientate care se pot forma cu aceste noduri este g:

$$g=2^{C_n^2}$$

sau

$$g=2^{\frac{n\cdot(n-1)}{2}}$$

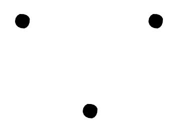
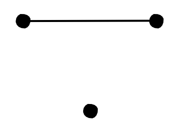
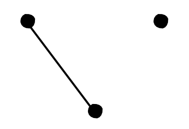
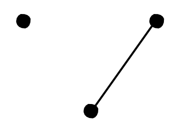
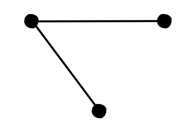
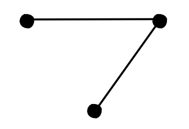
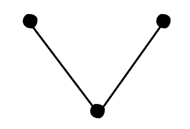
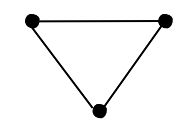
Exemplu

Fie graful `G=(X,U)`.

`n=3` – numărul de noduri

Numărul total de grafuri neorientate care se pot forma cu 3 noduri este:

$g=2^{C_n^2}=2^{C_3^2}=2^3=8$ sau $g=2^{\frac{n\cdot(n-1)}{2}}=2^{\frac{3\cdot(3-1)}{2}}=2^3=8$

1	2	3	4	5	6	7	8
							

2. Gradul unui nod

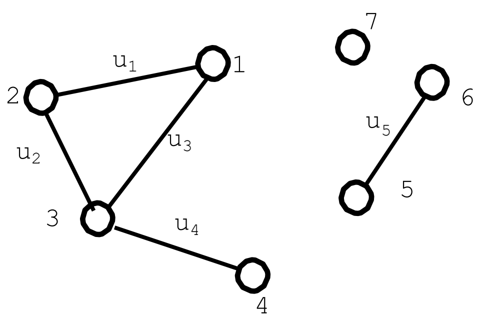
Definiţie:

Gradul unui nod `x` al grafului G este egal cu numărul muchiilor incidente cu nodul `x` şi se notează `d(x)`.

• se numeşte nod terminal un nod care are gradul egal cu 1, `d(x)=1` (este incident cu o singură muchie);
• se numeşte nod izolat un nod care are gradul egal cu 0, `d(x)=0` (nu este adiacent cu niciun alt nod al grafului);

Exemplu:

Fie graful `G=(X,U)`.
`n=7` - numărul de noduri
`m=5` - numărul de muchii



`d(1)=2`
`d(2)=2`
`d(3)=3`
`d(4)=1`
`d(5)=1`
`d(6)=1`
`d(7)=0`

noduri terminale: 4, 5 şi 6
nod izolat: 7

Teoremă

Într-un graf G=(X,U) cu n noduri şi m muchii, suma gradelor tuturor nodurilor grafului este egală cu dublul numărului de muchii:

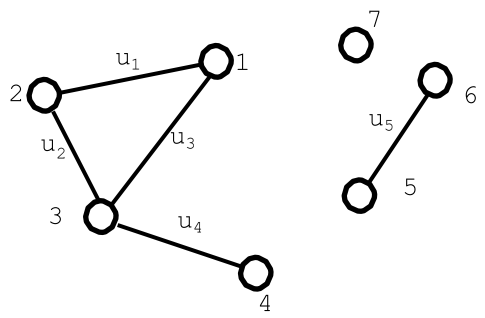
$$\sum_{i=1}^{n} d(x_i) = 2 \cdot m$$

Exemplu

Fie graful G=(X,U).

n=7 - numărul de noduri

m=5 - numărul de muchii



d(1)=2
d(2)=2
d(3)=3
d(4)=1
d(5)=1
d(6)=1
d(7)=0
d(1)+d(2)+d(3)+d(4)+d(5)+d(6)+d(7)=2·5=10

suma gradelor = 10

3. Noţiunile de lanţ şi ciclu

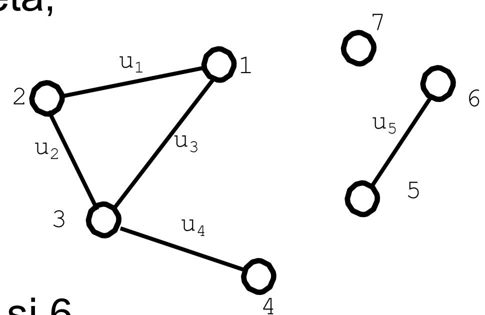
Definiţie

Se numeşte lanţ într-un graf G, o succesiune de noduri care au proprietatea că, oricare ar fi două noduri seccesive, ele sunt adiacente.

• lungimea lanţului reprezintă numărul de muchii din care este format lanţul;
• extremităţile unui lanţ sunt formate din nodul cu care începe lanţul şi nodul cu care se termină lanţul;
• se numeşte lanţ elementar un lanţ care conţine numai noduri distincte două câte două;
• se numeşte lanţ neelementar un lanţ care conţine noduri care se repetă;
• se numeşte lanţ simplu un lanţ care conţine numai muchii distincte două câte două;
• se numeşte lanţ compus un lanţ care conţine muchii care se repetă;

Exemplu

Fie graful G=(X,U).
n=7 - numărul de noduri
m=5 - numărul de muchii



L_1=(5,6) – lanţ elementar, de lungime 1, cu extremităţile 5 şi 6
L_2=(1,2,3) – lanţ elementar de lungime 2, cu extremităţile 1 şi 3
L_3=(1,2,3,4) – lanţ elementar, de lungime 3, cu extremităţile 1 şi 4
L_4=(1,3,4,3,2) – lanţ neelementar, de lungime 4, cu extremităţile 1 şi 2

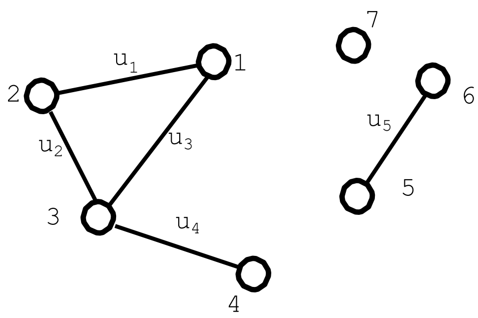
Definiţie

Se numeşte ciclu într-un graf G, un lanţ care are toate muchiile distincte două câte două şi extremităţile coincid.

• lungimea ciclului reprezintă numărul de muchii din care este format ciclul;
• se numeşte ciclu elementar un ciclu care conţine numai noduri distincte două câte două, cu excepţia extremităţilor ciclului;
• se numeşte ciclu neelementar un ciclu care conţine noduri care se repetă, cu excepţia extremităţilor ciclului;

Exemplu:

Fie graful G=(X,U).
n=7 - numărul de noduri
m=5 - numărul de muchii



C=(1,2,3,1) – ciclu elementar de lungime 3

4. Reprezentarea grafurilor neorientate

Printre modurile de reprezentare a unui graf se numără:

- matricea de adiacenţă;
- lista de adiacenţă;
- lista muchiilor;
- matricea costurilor.

a. Reprezentare prin matricea de adiacenţă:

Matricea de adiacenţă a unui graf cu n noduri este o matrice pătratică binară de ordinul `n`, ale cărei elemente $a_{i,j}$ sunt definite astfel:

$$a_{i,j} = \begin{cases} 1, & \text{dacă există muchie de la nodul } i \text{ la nodul } j \\ 0, & \text{dacă nu există muchie de la nodul } i \text{ la nodul } j \end{cases}$$

Proprietăţile matricei de adiacenţă:
• elementele de pe diagonala principală au valoarea 0;
• matricea de adiacenă este simetrică faţă de diagonala principală.

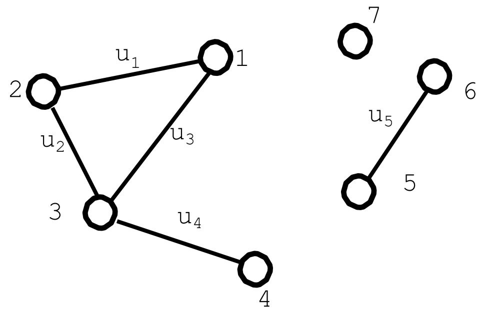
Informaţii obţinute din matricea de adiacenţă:
• suma elementelor matircei de adiacenţă este egal cu dublul numărului de muchii (`2·m`);
• gradul unui nod `i` este egal cu suma elementelor de pe linia `i` a matricei (sau cu suma elementelor de pe coloana `i`);
• nodurile adiacente nodului `i` sunt nodurile `j`, (`j=1,n`) pentru care elementele din linia `i` sunt egale cu 1;
• numărul de vecini ai nodului `i` este egal cu gradul nodului `i`;
• muchia `(i,j)` a grafului reprezintă un element al matricei de adiacenţă care îndeplineşte condiţia: `a[i][j]=1` sau `a[j][i]=1`.

Exemplu

Fie graful `G=(X,U)`.

`n=7` - numărul de noduri

`m=5` - numărul de muchii



Matricea de adiacenţă asociată grafului dat este:

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	1	0	1	0	0	0	0
3	1	1	0	1	0	0	0
4	0	0	1	0	0	0	0
5	0	0	0	0	0	1	0
6	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0

b. Reprezentare prin lista de adiacenţă

Lista de adiacenţă este formată din listele L_i $(i=1,n)$ care conţin toţi vecinii unui nod x_i la care se poate ajunge direct din x_i.

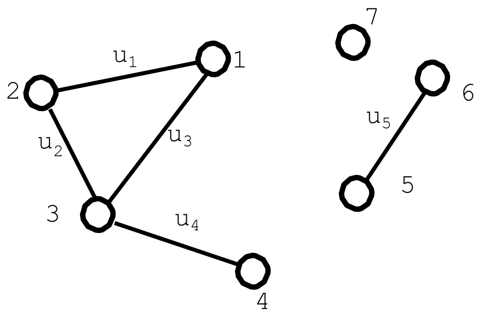
• lista L_i a vecinilor unui nod x_i al unui graf este formată din nodurile x_j adiacente nodului x_i;

Exemplu

Fie graful $G=(X,U)$.

$n=7$ - numărul de noduri

$m=5$ - numărul de muchii



Lista de adiacenţă asociată grafului dat este:

Nod	Lista de adiacenţă
1	2, 3
2	1, 3
3	1, 2, 4
4	3
5	6
6	5
7	-

c. Reprezentare prin lista muchiilor

Lista muchiilor este formată din `m` elemente care conţin, fiecare, câte o pereche de două noduri, x_i şi x_j, care formează o muchie, adică pentru care $[x_i, x_j] \in U$.

Implementarea listelor muchiilor se face folosind una dintre următoarele structuri de date:

- matricea muchiilor;
- vectorul muchiilor.

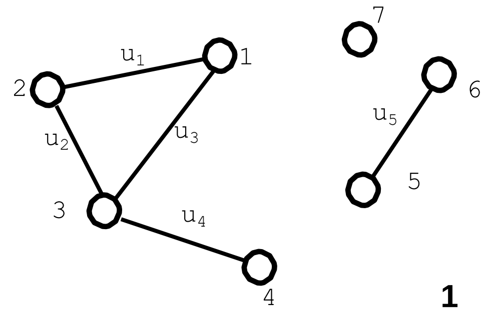
Matricea muchiilor este o matrice cu `m` linii şi 2 coloane, în care fiecare linie corespunde unei muchii.

Exemplu

Fie graful `G=(X,U)`.

`n=7` - numărul de noduri

`m=5` - numărul de muchii



Matricea muchiilor asociată grafului dat este:

	1	2
1	1	2
2	2	3
3	1	3
4	3	4
5	5	6

Vectorul muchiilor este o structură (înregistrare) cu două câmpuri x şi y ce conţin etichetele nodurilor care se găsesc la extremităţile muchiei.

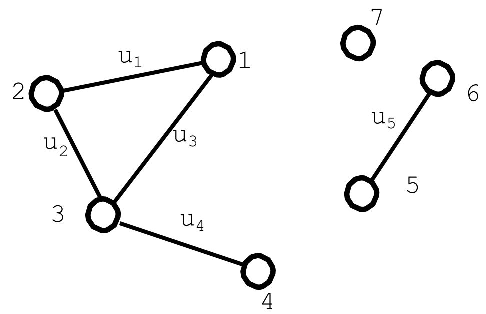
```
struct muchie
{
    int x,z;
}u[<m>];
```

Exemplu

Fie graful `G=(X,U)`.

`n=7` - numărul de noduri

`m=5` - numărul de muchii



Vectorul muchiilor asociat grafului dat este:

	u_1	u_2	u_3	u_4	u_5
x	1	2	1	3	5
y	2	3	3	4	6

d. Reprezentare prin matricea costurilor

Matricea costurilor unui graf cu `n` noduri este o matrice pătratică de ordinul `n`, ale cărei elemente $a_{i,j}$ sunt definite astfel:

$$a_{i,j} = \begin{cases} c, & \text{dacă există o muchie cu cost } c > 0 \text{ între nodurile } i \text{ şi } j, \text{ cu } i \neq j \\ 0, & \text{dacă } i = j \\ \pm\infty, & \text{dacă nu există muchie între nodurile } i \text{ şi } j, \text{ cu } i \neq j \end{cases}$$

Există două forme de reprezentare a matricei costurilor:
- *matricea costurilor minime* – pentru a determina valoarea minimă a funcţiei de cost (+∞);
- *matricea costurilor maxime* – pentru a determina valoarea maximă a funcţiei de cost (-∞);

Exemplu

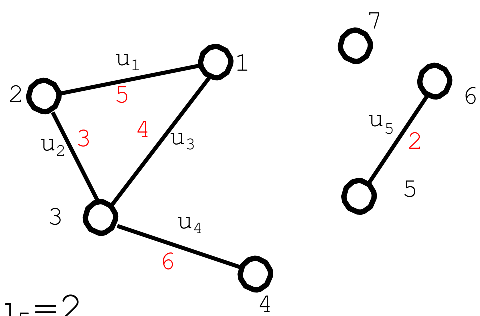
Fie graful `G=(X,U)`.

`n=7` - numărul de noduri

`m=5` - numărul de muchii

şi costurile:

$u_1=5, u_2=3, u_3=4, u_4=6, u_5=2$



Matricea costurilor minime asociată grafului dat este:

	1	2	3	4	5	6	7
1	0	5	4	+∞	+∞	+∞	+∞
2	5	0	3	+∞	+∞	+∞	+∞
3	4	3	0	6	+∞	+∞	+∞
4	+∞	+∞	6	0	+∞	+∞	+∞
5	+∞	+∞	+∞	+∞	0	2	+∞
6	+∞	+∞	+∞	+∞	2	0	+∞
7	+∞	+∞	+∞	+∞	+∞	+∞	0

5. Graf ponderat

Definiţie

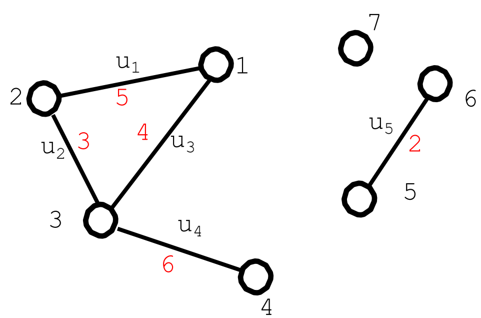
Se numeşte graf ponderat un graf în cadrul căruia fiecărei muchii îi este asociată o valoare.

Un graf ponderat poartă denumirea şi de *graf valoric* sau *graf cu costuri*.
Valoarea asociată muchiei are semnificaţia de “cost” al legăturii între cele două noduri, sau de “distanţa” între cele două vârfuri.
Un graf ponderat poate fi reprezentat utilizând matricea costurilor.

Exemplu

Fie graful G=(X,U).
n=7 - numărul de noduri
m=5 - numărul de muchii
şi costurile:
$u_1=5,\ u_2=3,\ u_3=4,\ u_4=6,\ u_5=2$

Graf ponderat:



6. Graf parţial

Definiţie

Fie graful G=(X,U) şi mulţimea V⊆U. Graful G_p=(X,V) se numeşte graf parţial al grafului G.

Un graf parţial al grafului G este chiar graful G sau se obţine din G păstrând toate nodurile şi eliminând nişte muchii.

Teoremă

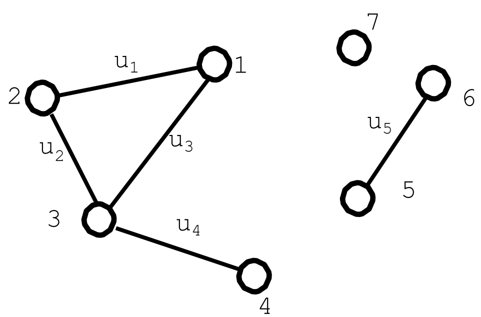
Numărul de grafuri parţiale ale unui graf cu m muchii este egal cu 2^m.

Exemplu

Fie graful G=(X,U).

n=7 - numărul de noduri

m=5 - numărul de muchii

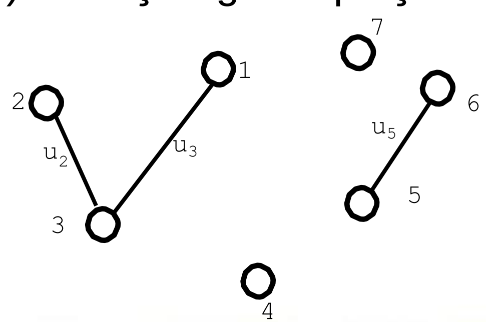


Prin eliminarea muchiilor (1,2) şi (3,4) se obţine graful parţial:

G_p=(X,V)

n=7 - numărul de noduri

m=3 - numărul de muchii



7. Subgraf

Definiţie

Fie graful G=(X,U). Graful G_s=(Y,V) se numeşte subgraf al grafului G dacă Y⊆X şi muchiile din mulţimea V sunt toate muchiile din mulţimea U care au ambele extremităţi în mulţimea Y.

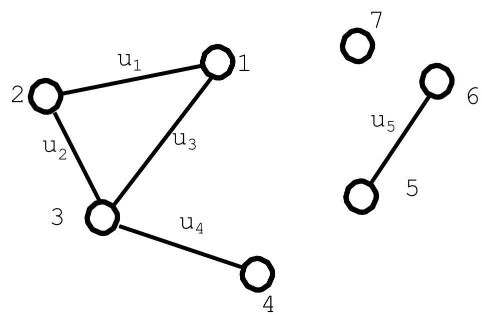
Un subgraf al grafului G este chiar graful G sau se obţine din G eliminând nişte noduri şi păstrând doar acele muchii care au ambele extremităţi în mulţimea nodurilor rămase.

Teoremă

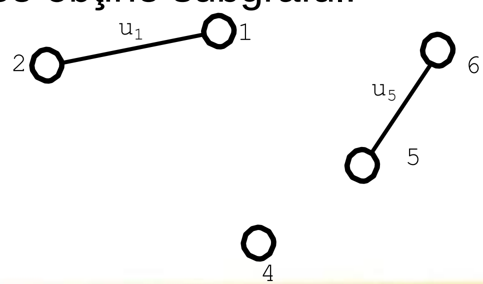
Numărul de subgrafuri ale unui graf cu n noduri este egal cu 2^n-1.

Exemplu

Fie graful `G=(X,U)`.
`n=7` - numărul de noduri
`m=5` - numărul de muchii



Prin eliminarea nodurilor 3 şi 7 se obţine subgraful:
G_s=(Y,V).
`n=5` - numărul de noduri
`m=2` - numărul de muchii



8. Tipuri speciale de grafuri

- graf complet;
- graf hamiltonian;
- graf eulerian;
- graf bipartit;

a. Graf complet

Definiţie

Un graf cu n noduri pentru care oricare două noduri sunt adiacente se numeşte graf complet cu n noduri şi se notează k_n.

Se poate construi un singur graf neorientat complet, cu n noduri, deoarece între oricare două noduri, x şi y există o singură muchie $[x,y]$.

Teoremă

Numărul m de muchii ale unui graf neorientat complet cu n noduri este:

$$m = n \cdot \frac{n-1}{2}$$

sau

$$m = C_n^2$$

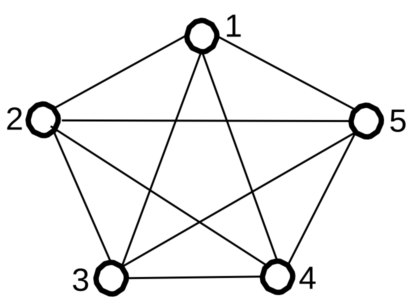
Exemplu

Fie graful G=(X,U).

n=5 - numărul de noduri

m=5·4/2=10 - numărul de muchii

Graful complet k_5:



Colegiul Naţional "Petru Rareş" Suceava
prof. Marius UDUDEC

b. Graf hamiltonian

Definiţie

Un ciclu elementar al unui graf G care trece prin toate nodurile grafului se numeşte ciclu hamiltonian.

Definiţie

Un graf G care conţine un ciclu hamiltonian se numeşte graf hamiltonian.

Un graf hamiltonian este un graf în care pornind de la un nod oarecare se pot parcurge o singură dată toate nodurile grafului, revenind la nodul iniţial.

Teoremă

Dacă G este un graf cu $n \geq 3$ noduri, astfel încât gradul oricărui nod x verifică inegalitatea:

$d(x) \geq \frac{n}{2}$, rezultă că G este hamiltonian.

Această teoremă este o condiţie suficientă, dar nu şi necesară ca un graf să fie hamiltonian.

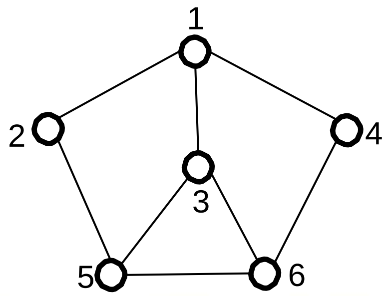
Exemplu

Fie graful G=(X,U).

$n=6$ - numărul de noduri

$m=8$ - numărul de muchii

Graf hamiltonian:



c. Graf eulerian

Definiţie

Un ciclu al unui graf G care conţine toate muchiile grafului se numeşte ciclu eulerian.

Definiţie

Un graf G care conţine un ciclu eulerian se numeşte graf eulerian.

Un graf eulerian este un graf în care pornind de la un nod oarecare se pot parcurge o singură dată toate muchiile grafului, revenind la nodul iniţial.

Teoremă

Un graf fără noduri izolate este eulerian, dacă şi numai dacă există un lanţ între oricare două noduri, şi gradele tuturor nodurilor sunt numere pare.

Această teoremă este o condiţie necesară şi suficientă ca un graf să fie eulerian.

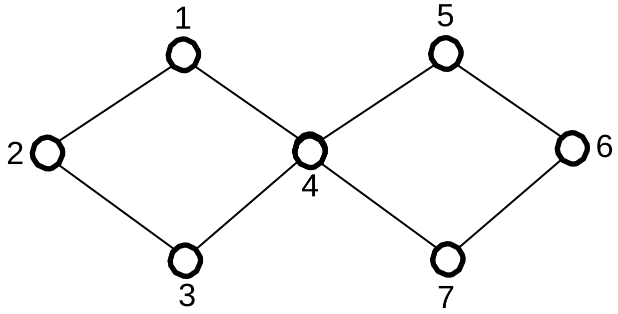
Exemplu

Fie graful G=(X,U).

`n=7` - numărul de noduri

`m=8` - numărul de muchii

Graf eulerian:



d. Graf bipartit

Definiţie

Graful G=(X,U) se numeşte graf bipartit dacă există două mulţimi nevide de noduri `A` şi `B` care au următoarele proprietăţi:
- $A \cup B = X$
- $A \cap B = \emptyset$
- orice muchie din mulţimea U are o extremitate în mulţimea de noduri `A` şi cealaltă extremitate în mulţimea de noduri `B`.

Definiţie

Se numeşte graf bipartit complet, un graf bipartit cu proprietatea că pentru orice nod `x` din `A` şi orice nod `y` din `B`, există muchia `[x,y]`.

Exemple

Fie graful G=(X,U).

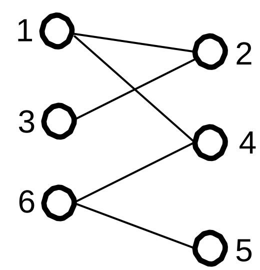
`n=6` - numărul de noduri

`X={1, 2, 3, 4, 5, 6}`

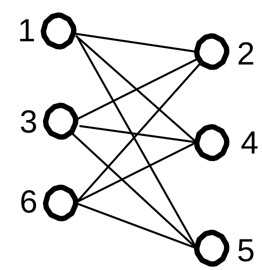
`A={1, 3, 6}`

`B={2, 4, 5}`

Graf bipartit:



Graf bipartit complet:



9. Conexitatea grafurilor

Graf conex

Definiţie

Un graf G se numeşte conex dacă pentru orice pereche de noduri (x, y) cu x≠y, există un lanţ de la x la y.

Definiţie

Se numeşte componentă conexă C a grafului G, un subgraf conex al lui G, care este maximal în raport cu această proprietate. Cu alte cuvinte nu există niciun lanţ al lui G care să unească un nod din C cu un nod care nu aparţine lui C.

Un graf conex are o singură componentă conexă, care cuprinde toate nodurile grafului.

Teoremă

Numărul minim de muchii necesar ca un graf neorientat, cu n noduri, să fie conex este n-1.

Un graf neorientat conex cu n noduri şi n-1 muchii este aciclic (nu conţine cicluri).

Exemple

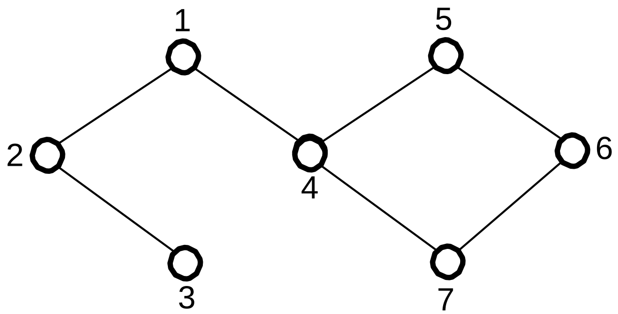
Fie graful G=(X,U).

`n` - numărul de noduri

`m` - numărul de muchii

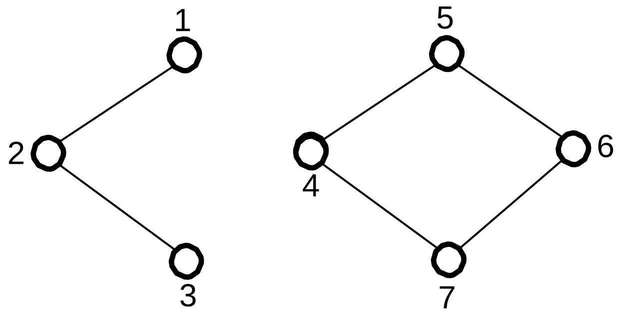
1. Graf conex:

`n=7, m=7`



2. Graf neconex (are 2 componente conexe):

`n=7, m=6`

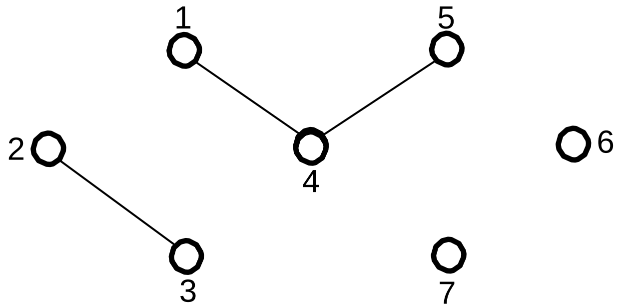


3. Graf conex:

`n=1, m=0`

4. Graf neconex (are 4 componente conexe):

`n=7, m=3`



10. Parcurgerea grafurilor

Parcurgerea unui graf înseamnă vizitarea nodurilor sale într-o anumită ordine, trecându-se de la un nod `i` (nodul curent) la nodurile vecine lui, în vederea prelucrării informaţiilor asociate nodurilor.

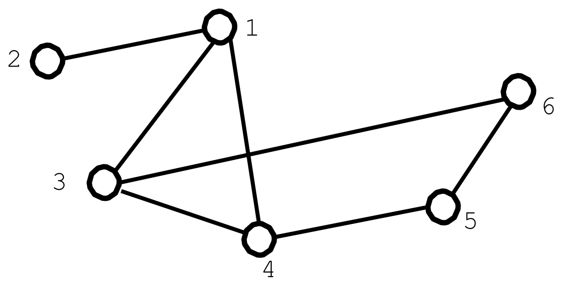
Există două metode de parcurgere a grafurilor:
- *metoda de parcurgere "în lăţime"* (BF – Breadth First);
- *metoda de parcurgere "în adâncime"* (DF – Depth First).

***a. Metoda de parcurgere "în lăţime"* (BF – Breadth First)**

Se vizitează mai întâi un nod iniţial `i`, apoi vecinii acestuia, apoi vecinii nevizitaţi ai acestora, şi aşa mai departe până când se parcurg toate nodurile grafului.

Exemple:

Fie graful G=(X,U).
`n=6` - numărul de noduri
`m=8` - numărul de muchii



Parcurgeri în lăţime:
- 1, 2, 3, 4, 6, 5
- 2, 1, 3, 4, 6, 5
- 3, 1, 4, 6, 2, 5
- 4, 3, 5, 1, 6, 2
- 5, 4, 6, 3, 1, 2
- 6, 3, 5, 1, 4, 2

Observaţie: vecinii unui nod pot fi vizitaţi în orice ordine.

b. Metoda de parcurgere "în adâncime" (DF – Depth First)

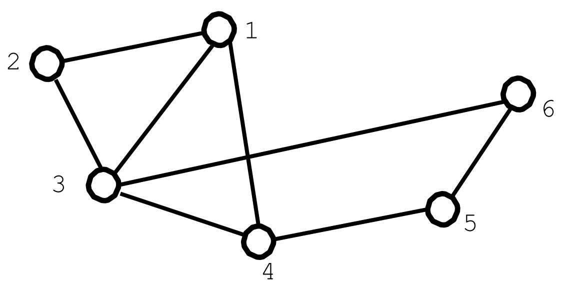
Se vizitează mai întâi un nod iniţial `i`, după care se parcurge unul dintre vecinii săi nevizitaţi, de exemplu `j`, după care se trece la alt vecin nevizitat al lui `j`, şi aşa mai departe până când se parcurge în adâncime ramura respectivă. Când s-a ajuns la capătul ei, se revine la nodul din care s-a plecat ultima dată şi se parcurge următorul vecin nevizitat.

Exemple:

Fie graful G=(X,U).

`n=6` - numărul de noduri

`m=8` - numărul de muchii



Parcurgeri în adâncime:

- 1, 2, 3, 4, 5, 6
- 2, 1, 3, 4, 5, 6
- 3, 1, 2, 4, 5, 6
- 4, 3, 1, 2, 6, 5
- 5, 4, 3, 1, 2, 6
- 6, 3, 1, 2, 4, 5

Observaţie: vecinii unui nod pot fi vizitaţi în orice ordine.

III. Grafuri orientate

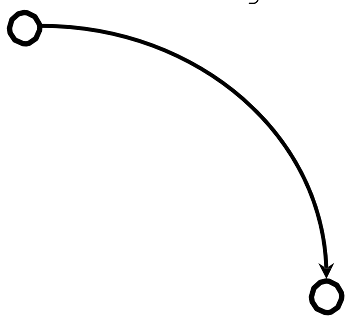
1. Terminologie

Definiţie

Se numeşte graf orientat (digraf), notat G=(X,U), o pereche ordonată de mulţimi (X,U), unde:
- X este o mulţime finită şi nevidă de elemente numite *vârfuri*;
- U este o mulţime de perechi ordonate din mulţimea X, numite *arce*.

- numim vârfuri adiacente orice pereche de vârfuri care formează un arc $(x,y) \in U$ sau $(y,x) \in U$;
- fiecare din cele două vârfuri (x sau y) este vârf incident cu arcul $u_k=(x,y)$ sau cu arcul $u_k=(y,x)$;
- vârfurile x şi y se numesc extremităţile arcului (x,y);
- vârful x este extremitatea iniţială a arcului (x,y);
- vârful y este extremitatea finală a arcului (x,y);
- se numesc arce incidente două arce u_i şi u_j care au o extremitate comună, vârful x;
- se numeşte succesor al vârfului x orice vârf la care ajunge arcul care iese din vârful x;
- se numeşte predecesor al vârfului x orice vârf de la care intră un arc în vârful x;
- mulţimea succesorilor vârfului x, notată $\Gamma^{+}(x)$ este formată din mulţimea vârfurilor la care ajung arcele care ies din vârful x;
- mulţimea predecesorilor vârfului x, notată $\Gamma^{-}(x)$ este formată din mulţimea vârfurilor la care ajung arcele care intră în vârful x;

- mulţimea arcelor care ies din vârful x, notată $\omega^{+}(x)$ este formată din arcele care au extremitatea iniţială în vârful x;
- mulţimea arcelor care intră în vârful x, notată $\omega^{-}(x)$ este formată din arcele care au extremitatea finală în vârful x;



Exemplu

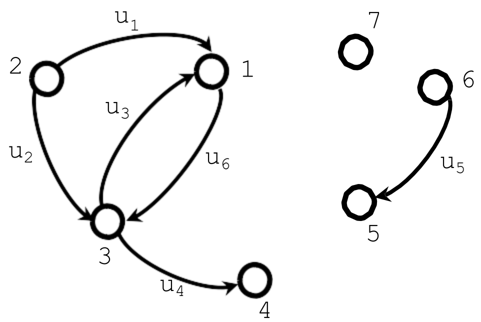
Fie graful G=(X,U).

n=7 - numărul de vârfuri (ordinul grafului G)

m=6 - numărul de arce

X={1, 2, 3, 4, 5, 6, 7} - mulţimea vârfurilor

$U=\{u_1,u_2,u_3,u_4,u_{,5},u_6\}$={(2,1),(2,3),(3,1),(3,4),(6,5),(1,3)} - mulţimea arcelor



- vârful 3 este adiacent cu vârfurile 1, 2 şi 4;
- vârful 3 este vârf incident cu arcele u_1, u_2, u_3 şi u_6;
- vârful 3 este extremitatea iniţială a arcelor u_3 şi u_4;
- vârful 3 este extremitatea finală a arcelor u_2 şi u_6;
- arcele u_1 şi u_2 sunt incidente deoarece au un vârf comun, vârful 2;
- mulţimea succesorilor vârfului 3 are forma: $\Gamma^+(3)=\{1,4\}$;
- mulţimea predecesorilor vârfului 1 are forma: $\Gamma^-(x)=\{2,3\}$;
- mulţimea arcelor care ies din vârful 3 are forma: $\omega^+(3)=\{u_2,u_4\}$;
- mulţimea arcelor care intră în vârful 1 are forma: $\omega^-(3)=\{u_1,u_3\}$;

Teoremă

Într-un graf G=(X,U) cu `n` vârfuri, numărul total de grafuri orientate care se pot forma cu aceste vârfuri este g:

$$g = 4^{C_n^2}$$

sau

$$g = 4^{\frac{n\cdot(n-1)}{2}}$$

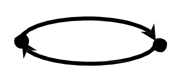
Exemplu

Fie graful `G=(X,U)`.

`n=2` - numărul de vârfuri

Numărul total de grafuri orientate care se pot forma cu 2 vârfuri este:

$$g = 4^{\frac{n\cdot(n-1)}{2}} = 4^{\frac{2\cdot(2-1)}{2}} = 4$$

1	2	3	4
			

2. Gradul unui vârf

Gradul unui vârf este caracterizat prin gradul intern şi gradul extern.

Definiţie

Gradul intern al unui vârf x_k al grafului G este egal cu numărul arcelor care intră în vârful x_k şi se notează cu $d^-(x_k)$.

Definiţie

Gradul extern al unui vârf x_k al grafului G este egal cu numărul arcelor care ies din vârful x_k şi se notează cu $d^+(x_k)$.

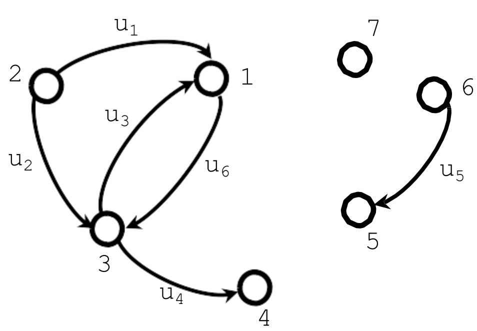
- se numeşte vârf terminal un vârf care are suma gradelor egală cu 1 (este incident cu un singur arc);
- se numeşte vârf izolat un vârf care are suma gradelor egală cu 0 (nu este adiacent cu niciun alt vârf al grafului);

Exemplu

Fie graful $G=(X,U)$.

$n=7$ - numărul de vârfuri

$m=6$ - numărul de arce



$d^-(1)=2$	$d^+(1)=1$
$d^-(2)=0$	$d^+(2)=2$
$d^-(3)=2$	$d^+(3)=2$
$d^-(4)=1$	$d^+(4)=0$
$d^-(5)=1$	$d^+(5)=0$
$d^-(6)=0$	$d^+(6)=1$
$d^-(7)=0$	$d^+(7)=0$

vârfuri terminale: 4, 5 şi 6

vârfuri izolate: 7

Teoremă

Într-un graf G=(X,U) cu `n` vârfuri şi `m` arce, suma gradelor interne ale tuturor vârfurilor este egală cu suma gradelor externe ale tuturor vârfurilor şi cu numărul de arce:

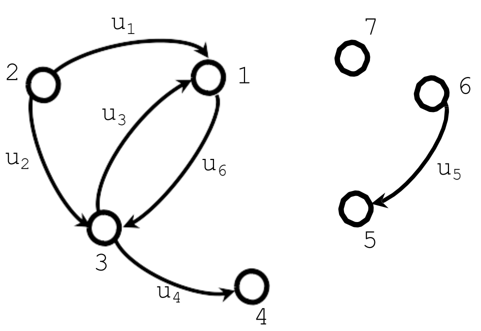
$$\sum_{i=1}^{n} d^{-}(x_i) = \sum_{i=1}^{n} d^{+}(x_i) = m$$

Exemplu

Fie graful G=(X,U).

`n=7` - numărul de vârfuri

`m=6` - numărul de arce



$d^{-}(1)=2$	$d^{+}(1)=1$
$d^{-}(2)=0$	$d^{+}(2)=2$
$d^{-}(3)=2$	$d^{+}(3)=2$
$d^{-}(4)=1$	$d^{+}(4)=0$
$d^{-}(5)=1$	$d^{+}(5)=0$
$d^{-}(6)=0$	$d^{+}(6)=1$
$d^{-}(7)=0$	$d^{+}(7)=0$

$d^{-}(1)+d^{-}(2)+d^{-}(3)+d^{-}(4)+d^{-}(5)+d^{-}(6)+d^{-}(7)=$

$=d^{+}(1)+d^{+}(2)+d^{+}(3)+d^{+}(4)+d^{+}(5)+d^{+}(6)+d^{+}(7)=6$

suma gradelor interne = 6

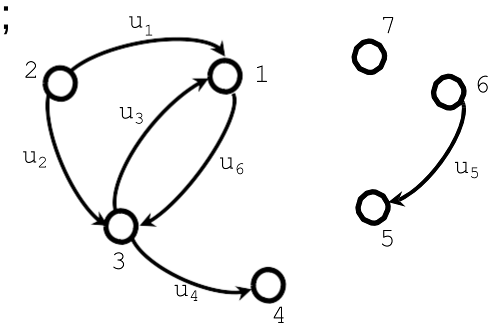
suma gradelor externe = 6

3. Noţiunile de lanţ, drum şi circuit

Definiţie

Se numeşte lanţ într-un graf G, o succesiune de vârfuri care au proprietatea că, oricare ar fi două vârfuri succesive, ele sunt adiacente şi nu respectă sensul săgeţilor.

- la definirea unui lanţ nu se ţine cont de orientarea arcelor;
- lungimea lanţului reprezintă numărul de arce din care este format lanţul;
- extremităţile unui lanţ sunt formate din vârful cu care începe şi vârf cu care se termină lanţul;
- se numeşte lanţ elementar un lanţ care conţine numai vârfuri distincte două câte două;
- se numeşte lanţ neelementar un lanţ care conţine vârfuri care se repetă;
- se numeşte lanţ simplu un lanţ care conţine numai arce distincte două câte două;
- se numeşte lanţ compus un lanţ care conţine arce care se repetă;



Exemplu

Fie graful G=(X,U).

n=7 - numărul de vârfuri

m=6 - numărul de arce

L_1=(1,2,3,4) – lanţ elementar, de lungime 3, cu extremităţile 1 şi 4

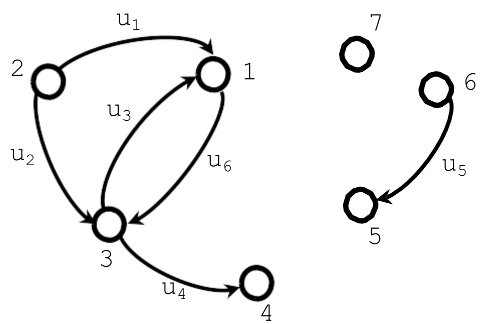
L_2=(4,3,1,2,3) – lanţ neelementar de lungime 4, cu extremităţile 4 şi 3

Definiţie

Se numeşte drum într-un graf G, o succesiune de vârfuri care au proprietatea că, oricare ar fi două vârfuri succesive, ele sunt adiacente şi respectă sensul săgeţilor.

- lungimea drumului reprezintă numărul de arce din care este format drumul;
- extremităţile unui drum sunt formate din vârful cu care începe drumul şi vârful cu care se termină drumul;
- se numeşte drum elementar un drum care conţine numai vârfuri distincte două câte două;
- se numeşte drum neelementar un drum care conţine vârfuri care se repetă;
- se numeşte drum simplu un drum care conţine numai arce distincte două câte două;
- se numeşte drum compus un drum care conţine arce care se repetă;
- pentru reprezentare se utilizează o matrice `d` cu `n` linii şi `n` coloane în care fiecare element `d[i][j]` este definit astfel:

$$d[i][j]=\begin{cases}1, & dacă \quad există \quad drum \quad de \quad la \quad vârful \quad i \quad la \quad vârful \quad j\\ 0, & în \quad caz \quad contrar\end{cases}$$



Exemplu

Fie graful `G=(X,U)`.

`n=7` - numărul de vârfuri

`m=6` - numărul de arce

D_1=(2,3,4) – drum elementar, de lungime 2, cu extremităţile 2 şi 4

D_2=(3,1) – drum elementar de lungime 1, cu extremităţile 3 şi 1

D_3=(2,1,3,1) – drum neelementar, de lungime 3, cu extremităţile 2 şi 1

D_4=(2,3,1,3,4) – drum neelementar, de lungime 4, cu extremităţile 2 şi 4

Definiţie

Se numeşte circuit într-un graf G, un drum care are toate arcele distincte două câte două şi extremităţile coincid.

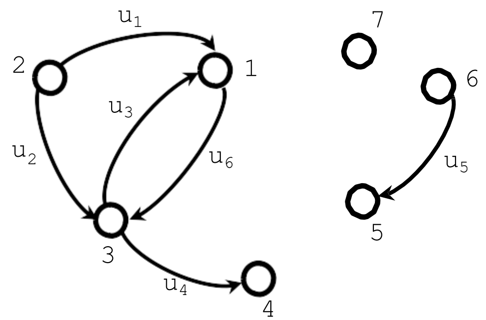
- lungimea circuitului reprezintă numărul de arce din care este format circuitul;
- se numeşte circuit elementar un circuit care conţine numai vârfuri distincte două câte două, cu excepţia extremităţilor circuitului;
- se numeşte circuit neelementar un circuit care conţine vârfuri care se repetă, cu excepţia extremităţilor circuitului;

Exemplu

Fie graful `G=(X,U)`.

`n=7` - numărul de vârfuri

`m=6` - numărul de arce



`C=(1,3,1)` – circuit elementar de lungime 2

4. Reprezentarea grafurilor

Printre modurile de reprezentare a unui graf se numără:

- matricea de adiacenţă;
- lista de adiacenţă;
- lista arcelor;
- matricea costurilor.

a. Reprezentare prin matricea de adiacenţă

Matricea de adiacenţă a unui graf cu n vârfuri este o matrice pătratică binară de ordinul `n`, ale cărei elemete $a_{i,j}$ sunt definite astfel:

$$a_{i,j} = \begin{cases} 1, & dacă\ există\ arc\ de\ la\ vârful\ i\ la\ vârful\ j \\ 0, & dacă\ nu\ există\ arc\ de\ la\ vârful\ i\ la\ vârful\ j \end{cases}$$

Proprietăţile matricei de adiacenţă:
- elementele de pe diagonala principală au valoarea 0;
- matricea de adiacenă nu este simetrică faţă de diagonala principală.

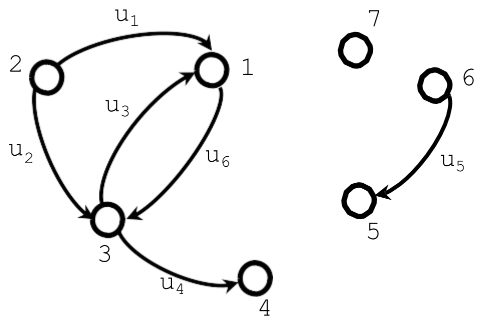
Informaţii obţinute din matricea de adiacenţă:
- suma elementelor matricei de adiacenţă este egal cu numărul de arce (`m`);
- gradul extern al unui vârf `i` este egal cu suma elementelor de pe linia `i` a matricei;
- gradul intern al unui vârf `i` este egal cu suma elementelor de pe coloana `i` a matricei;
- succesorii vârfului `i` sunt vârfurile `j` pentru care elementele de pe linia `i` sunt egale cu 1;
- predecesorii vârfului `i` sunt vârfurile `j` pentru care elementele de pe coloana `i` sunt egale cu 1;
- vârfurile adiacente vârfului `i` sunt vârfurile `j` pentru care elementele din linia `i` sau coloana `i` sunt egale cu 1;
- numărul de vecini ai vârfului `i` este egal cu numărul de vârfuri adiacente vârfului `i`;
- arcul `(i,j)` a grafului reprezintă un element al matricei de adiacenţă care îndeplineşte condiţia: `a[i][j]=1`.

Exemplu

Fie graful `G=(X,U)`.

`n=7` - numărul de vârfuri

`m=6` - numărul de arce



Matricea de adiacenţă asociată grafului dat este:

	1	2	3	4	5	6	7
1	0	0	1	0	0	0	0
2	1	0	1	0	0	0	0
3	1	0	0	1	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0

b. Reprezentare prin lista de adiacenţă (lista succesorilor)

Lista de adiacenţă este formată din listele L_i (i=1,n) care conţin toţi vecinii unui vârf x la care se poate ajunge direct din x, ţinând cont de sensul arcului.

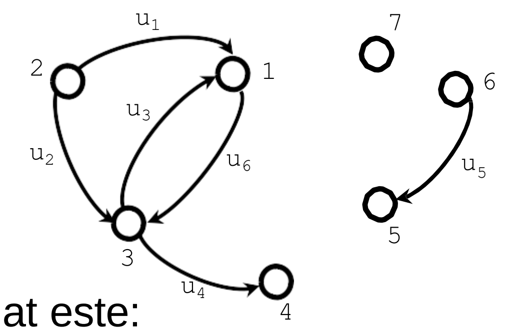
• lista L_i a vecinilor unui vârf x al unui graf este formată din vârfurile y adiacente vârfului x;

Exemplu

Fie graful G=(X,U).

n=7 - numărul de vârfuri

m=6 - numărul de arce



Lista de adiacenţă asociată grafului dat este:

Vârf	Lista de adiacenţă
1	3
2	1, 3
3	1, 4
4	-
5	-
6	5
7	-

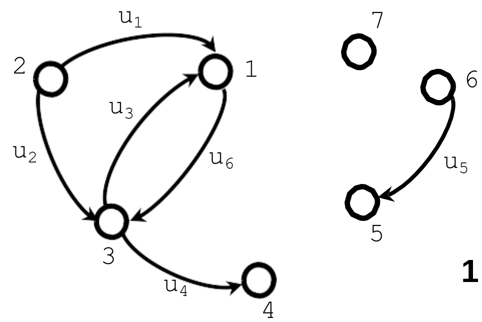
c. Reprezentare prin lista arcelor

Lista arcelor este formată din m elemente care conţin, fiecare, câte o pereche de două vârfuri, x_i şi x_j, care formează un arc, adică pentru care $(x_i, x_j) \in U$.

Implementarea listelor arcelor se face folosind una dintre următoarele structuri de date:
• matricea arcelor;
• vectorul arcelor.

Matricea arcelor este o matrice cu m linii şi 2 coloane, în care fiecare linie corespunde unui arc.

Exemplu
Fie graful $G=(X,U)$.
$n=7$ - numărul de vârfuri
$m=6$ - numărul de arce



Matricea arcelor asociată grafului dat este:

	1	2
1	2	1
2	2	3
3	3	1
4	3	4
5	6	5
6	1	3

Vectorul arcelor este o structură (înregistrare) cu două câmpuri x şi y ce conţin etichetele vârfurilor care se găsesc la extremităţile arcului.

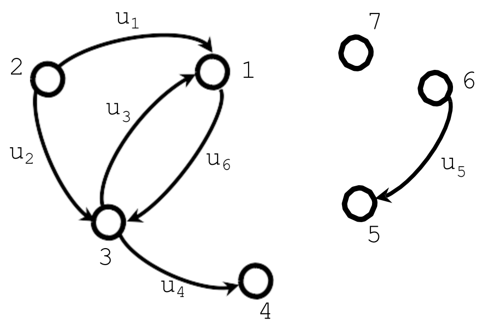
```
struct arc
{
    int x,y;
}u[<m>];
```

Exemplu

Fie graful `G=(X,U)`.

`n=7` - numărul de vârfuri

`m=6` - numărul de arce



Vectorul arcelor asociat grafului dat este:

	u_1	u_2	u_3	U_4	u_5	u_6
x	2	2	3	3	6	1
y	1	3	1	4	5	3

d. Reprezentare prin matricea costurilor

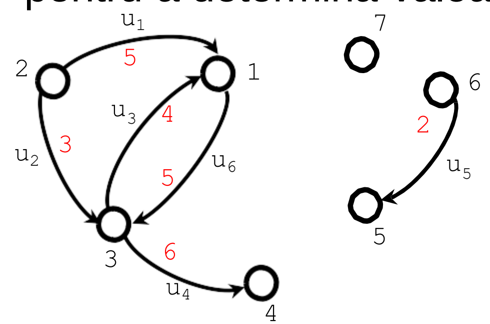
Matricea costurilor unui graf cu n vârfuri este o matrice pătratică de ordinul n, ale cărei elemente $a_{i,j}$ sunt definite astfel:

$$a_{i,j} = \begin{cases} c, & dacă \ \ nu \ \ există \ \ un \ \ arc \ \ cu \ \ costul \ \ c > 0 \ \ între \ \ vârfurile \ \ i \ \ şi \ \ j, \ \ cu \ \ i \neq j \\ 0, & dacă \ \ i = j \\ \pm\infty, & dacă \ \ nu \ \ există \ \ arc \ \ între \ \ vârfurile \ \ i \ \ şi \ \ j, \ \ cu \ \ i \neq j \end{cases}$$

Există două forme de reprezentare a matricei costurilor:
- *matricea costurilor minime* – pentru a determina valoarea minimă a funcţiei de cost (+∞);
- *matricea costurilor maxime* – pentru a determina valoarea maximă a funcţiei de cost (-∞);

Exemplu

Fie graful G=(X,U).
n=7 - numărul de vârfuri
m=6 - numărul de arce
şi costurile:
u_1=5, u_2=3, u_3=4, u_4=6, u_5=2, u_6=5



Matricea costurilor minime asociată grafului dat este:

	1	2	3	4	5	6	7
1	0	+∞	5	+∞	+∞	+∞	+∞
2	5	0	3	+∞	+∞	+∞	+∞
3	4	+∞	0	6	+∞	+∞	+∞
4	+∞	+∞	+∞	0	+∞	+∞	+∞
5	+∞	+∞	+∞	+∞	0	+∞	+∞
6	+∞	+∞	+∞	+∞	2	0	+∞
7	+∞	+∞	+∞	+∞	+∞	+∞	0

5. Graf ponderat

Definiţie

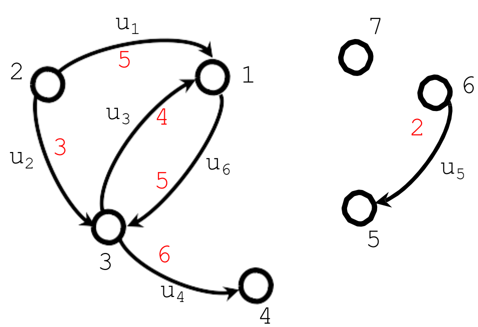
Se numeşte graf ponderat un graf în cadrul căruia fiecăruărui arc îi este asociat o valoare.

Un graf ponderat poartă denumirea şi de *graf valoric* sau *graf cu costuri*.
Valoarea asociată arcului are semnificaţia de "cost" al legăturii între cele două vârfuri, sau de "distanţa" între cele două vârfuri.
Un graf ponderat poate fi reprezentat utilizând matricea costurilor.

Exemplu

Fie graful G=(X,U).
n=7 - numărul de vârfuri
m=6 - numărul de arce
şi costurile:
$u_1=5,\ u_2=3,\ u_3=4,\ u_4=6,\ u_5=2,\ u_6=5$

Graf ponderat:



6. Graf parţial

Definiţie

Fie graful G=(X,U) şi mulţimea $V \subseteq U$. Graful G_p=(X,V) se numeşte graf parţial al grafului G.

Un graf parţial al grafului G este chiar graful G sau se obţine din G păstrând toate vârfurile şi eliminând nişte arce.

Teoremă

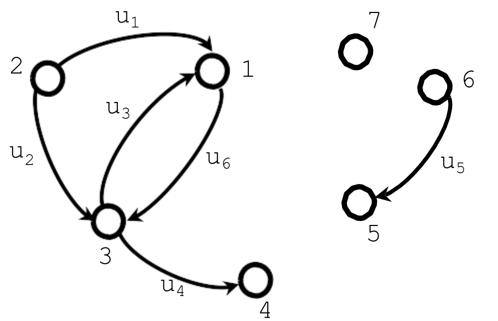
Numărul de grafuri parţiale ale unui graf cu m arce este egal cu 2^m.

Exemplu

Fie graful G=(X,U).

n=7 - numărul de vârfuri

m=6 - numărul de arce

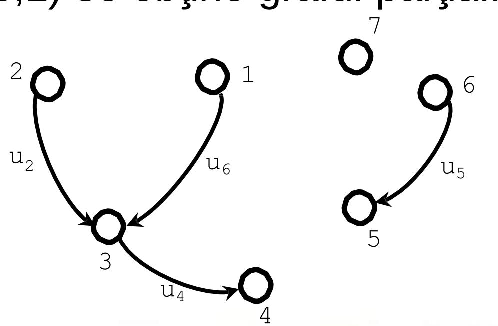


Prin eliminarea arcelor (2,1) şi (3,1) se obţine graful parţial:

G_p=(X,V)

n=7 - numărul de vârfuri

m=4 - numărul de arce



7. Subgraf

Definiţie

Fie graful G=(X,U). Graful G_s=(Y,V) se numeşte subgraf al grafului G dacă $Y \subseteq X$ şi arcele din mulţimea V sunt toate arcele din mulţimea U care au ambele extremităţi în mulţimea Y.

Un subgraf al grafului G este chiar graful G sau se obţine din G eliminând nişte vârfuri şi păstrând doar acele arce care au ambele extremităţi în mulţimea vârfurilor rămase.

Teoremă

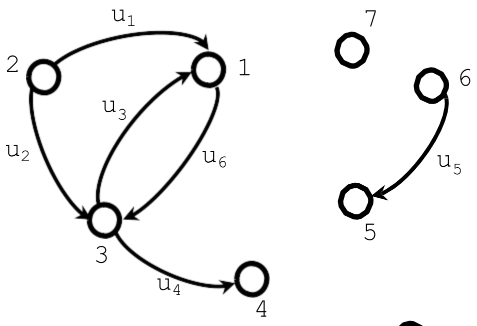
Numărul de subgrafuri ale unui graf cu n vârfuri este egal cu 2^n-1.

Exemplu

Fie graful `G=(X,U)`.

`n=7` - numărul de vârfuri

`m=6` - numărul de arce

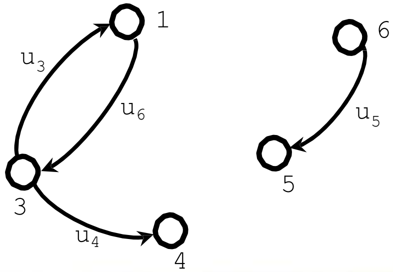


Prin eliminarea vârfurilor 2 şi 7 se obţine subgraful:

G_s=(Y,V)

`n=5` - numărul de vârfuri

`m=4` - numărul de arce



8. Tipuri speciale de grafuri

- graf complet;
- graf bipartit;
- graf turneu;

a. *Graf complet*

Definiţie

Un graf cu `n` vârfuri pentru care oricare două vârfuri sunt adiacente se numeşte graf complet cu `n` vârfuri şi se notează k_n.

Se pot construi mai multe grafuri orientat complete, cu `n` vârfuri, deoarece două vârfuri, `x` şi `y` pot fi adiacente în trei situaţii: există arcul `(x,y)`, există arcul `(y,x)` sau există arcele `(x,y)` şi `(x,y)`.

Teoremă

Numărul de grafuri orientat complete care se pot construi cu `n` vârfuri este:

$$n_k = 3^{C_n^2}$$

Exemplu:

Fie graful G=(X,U).

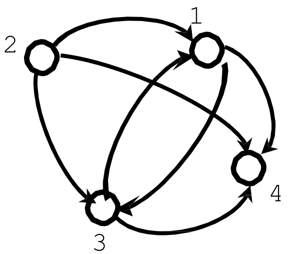
`n=4` - numărul de vârfuri

`m=6` - numărul de arce

Numărul de grafuri complete cu 4 vârfuri este:

$$n_k = 3^{C_4^2} = 3^6 = 729$$

Graf complet:



b. Graf bipartit

Definiţie

Graful G=(X,U) se numeşte graf bipartit dacă există două mulţimi nevide de vârfuri A şi B care au următoarele proprietăţi:
- A∪B=X
- A∩B=∅
- orice arc din mulţimea U are o extremitate în mulţimea de vârfuri A şi cealaltă extremitate în mulţimea de vârfuri B.

Definiţie

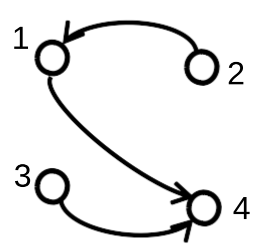
Se numeşte graf bipartit complet, un graf bipartit cu proprietatea că pentru orice vârf x din A şi orice vârf y din B, există un arc format din cele două vârfuri care aparţin mulţimii U.

Pentru un graf orientat bipartit complet fiecare vârf din A trebuie să fie fie legat cu cel puţin un arc de fiecare vârf din mulţimea B.

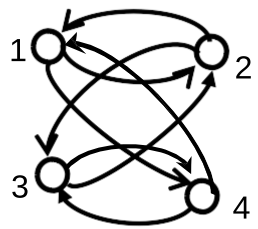
Exemple

Fie graful G=(X,U).
n=4 - numărul de vârfuri
X={1,2,3,4}
A={1,3}
B={2,4}

Graf bipartit:



Graf bipartit complet:



c. Graf turneu:

Definiţie

Un graf orientat în care, între oricare două vârfuri există un singur arc şi numai unul, se numeşte graf turneu.

- arcul dintre două vârfuri poate avea oricare dintre cele două orientări;
- graful turneu este un graf complet;

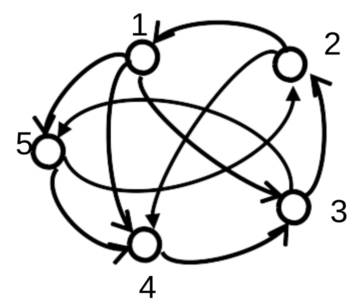
Exemple:

Fie graful G=(X,U).

`n=5` – numărul de vârfuri

`m=10` – numărul de arce

Graf turneu:



9. Conexitatea grafurilor

Graf conex

Definiţie

Un graf G se numeşte conex dacă pentru orice pereche de vârfuri (x, y) cu $x \neq y$, există un lanţ de la x la y.

Definiţie

Se numeşte componentă conexă C a grafului G, un subgraf conex al lui G, care este maximal în raport cu această proprietate. Cu alte cuvinte nu există niciun lanţ al lui G care să unească un vârf din C cu un vârf care nu aparţine lui C.

Un graf conex are o singură componentă conexă, care cuprinde toate vârfurile grafului.

Exemple

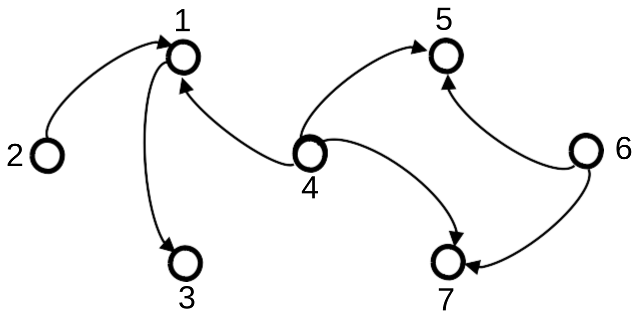
Fie graful G=(X,U).

`n` - numărul de vârfuri

`m` - numărul de arce

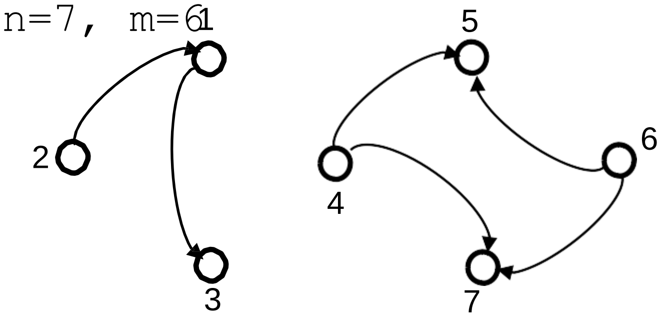
1. Graf conex:

`n=7, m=7`



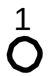
2. Graful nu este conex (are 2 componente conexe):

`n=7, m=6`



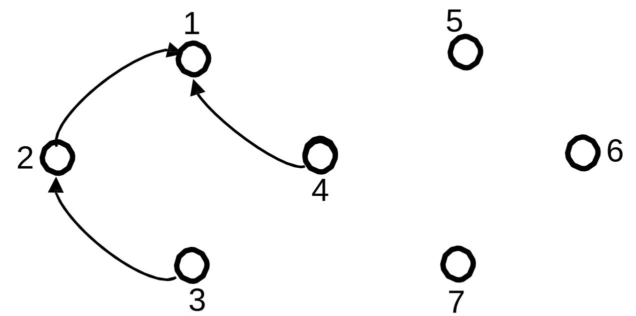
3. Graf conex:

`n=1, m=0`



4. Graful nu este conex (are 4 componente conexe):

`n=7, m=3`



Graf tare conex

Definiţie

Un graf orientat G se numeşte graf tare conex dacă are proprietatea că, pentru orice pereche de vârfuri distincte, există un drum care să le lege.

Definiţie:

Se numeşte componentă tare conexă a unui graf G un subgfraf tare conex C=(Y,V) al său, maximal în raport cu această proprietate (conţine numărul maxim de vârfuri din G care au proprietatea că sunt legate printr-un drum).

Exemple

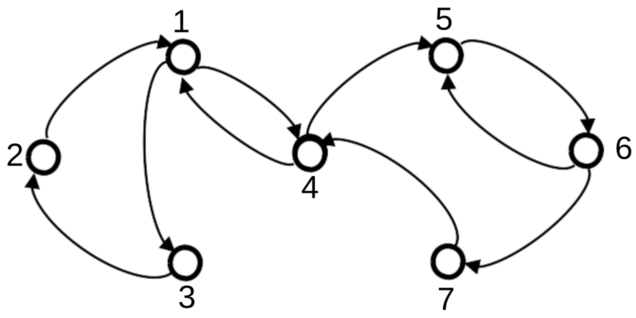
Fie graful G=(X,U).

`n` - numărul de vârfuri

`m` - numărul de arce

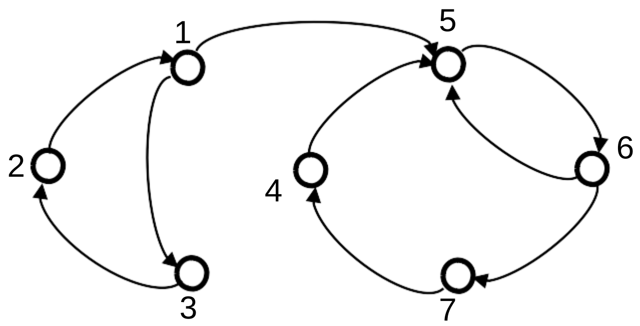
1. Graf tare conex:

`n=7, m=10`



2. Graful nu este tare conex (are 2 componente tare conexe):

`n=7, m=9`



10. Parcurgerea grafurilor

Parcurgerea unui graf înseamnă vizitarea vârfurilor sale într-o anumită ordine, trecându-se de la un vârf `i` (vârful curent) la vârfurile vecine lui, în vederea prelucrării informaţiilor asociate vârfurilor.

Există două metode de parcurgere a grafurilor:

- *metoda de parcurgere "în lăţime"* (BF – Breadth First);
- *metoda de parcurgere "în adâncime"* (DF – Depth First).

a. Metoda de parcurgere "în lăţime" (BF – Breadth First)

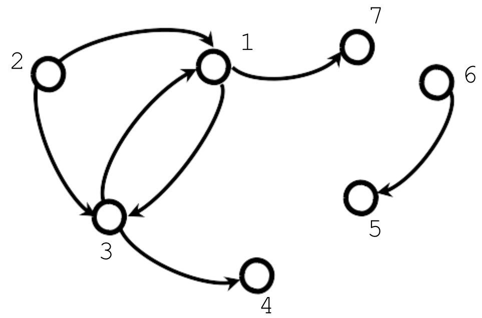
Se vizitează mai întâi un vârf iniţia `i`, apoi vecinii acestuia, apoi vecinii nevizitaţi ai acestora, şi aşa mai departe până când se parcurg toate vârfurile grafului.

Exemple:

Fie graful G=(X,U).

`n=7` - numărul de vârfuri

`m=7` - numărul de arce



Parcurgeri în lăţime:

- 1, 3, 7, 4
- 2, 1, 3, 7, 4
- 3, 1, 4, 7
- 4
- 5
- 6, 5
- 7

b. *Metoda de parcurgere "în adâncime"* (DF – Depth First)

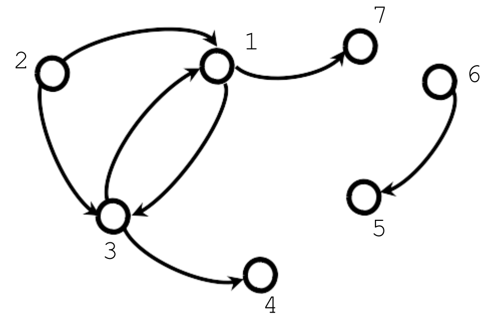
Se vizitează mai întâi un vârf iniţia `i`, după care se percurge unul dintre vecinii săi nevizitaţi, de exemplu `j`, după care se trece la alt vecin nevizitat al lui `j`, şi aşa mai departe până când se parcurge în adâncime ramura respectivă. Când s-a ajuns la capătul ei, se revine la vârful din care s-a plecat ultima dată şi se parcurge alt vecin nevizitat.

Exemple:

Fie graful G=(X,U).

`n=7` - numărul de vârfuri

`m=7` - numărul de arce



Parcurgeri în adâncime:

- 1, 3, 4, 7
- 2, 1, 3, 4, 7
- 3, 1, 7, 4
- 4
- 5
- 6, 5
- 7

IV. Arbori

1. Terminologie

Definiţie

Se numeşte arbore, un graf neorientat conex şi fără cicluri.

Teoremă

Fie **G** un graf neorientat cu n noduri. **G** este arbore dacă şi numai dacă are **n-1** muchii şi nu conţine cicluri.

Definiţie

Se numeşte arbore cu rădăcină, un arbore în care există un nod privilegiat numit nod rădăcină.

Definiţie

Se numeşte arborescenţă sau structură arborescentă, un arbore cu rădăcină, în care s-a stabilit nodul rădăcină.

- rădăcina arborelui este un nod în care nu intră nicio muchie;
- cu excepţia rădăcinii, fiecare nod are propietatea că în el intră o singură muchie; acesta leagă nodul respectiv de un alt nod numit predecesor direct sau părinte;
- dintr-un nod pot ieşi una sau mai multe muchii; fiecare astfel de muchie leagă nodul respectiv de un alt nod numit succesor direct sau fiu;
- nodurile arborelului sunt organizate pe nivele, primul nivel fiind ocupat de nodul rădăcină;
- nodurile care se caracterizează prin faptul că nu au niciun fiu se numesc noduri terminale sau frunze;
- nodurile pot conţine o aşa-numită informaţie utilă (de orice tip), numită şi cheia nodului;

Exemplu

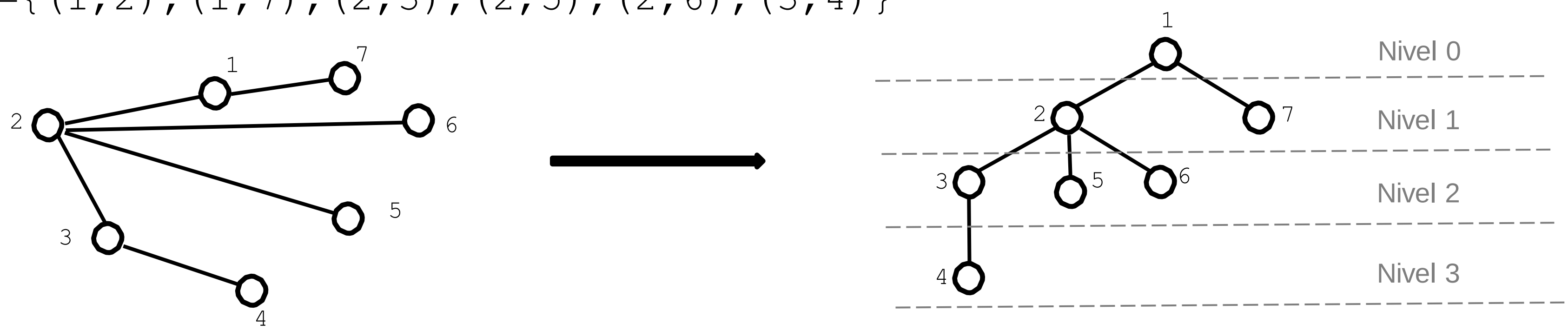
Fie graful G=(X,U).

n=7 - numărul de noduri

m=6 - numărul de muchii

X={1,2,3,4,5,6,7}

U={(1,2),(1,7),(2,3),(2,5),(2,6),(3,4)}



Graful G este arbore cu rădăcină pentru că are o singură componentă conexă şi este aciclic.

Orice nod al grafului poate fi considerat nod rădăcină.

Construirea arborelui având ca rădăcină nodul 1:

- rădăcina arborelui este nodul 1;
- ascendenţii nodului 3 sunt nodurile 1 şi 2;
- ascendentul direct (predecesor) al nodului 3 este nodul 2;
- descendenţii nodului 2 sunt nodurile 3, 4, 5 şi 6;
- descendenţii direcţi (succesor) ai nodului 2 sunt nodurile 3, 5 şi 6;
- nodurile terminale (frunze) sunt: 4, 5, 6 şi 7;

2. Reprezentarea arborilor cu rădăcină

Metode de reprezentare a arborilor cu rădăcină:

- matrice de adiacenţă;
- listă de adiacenţă;
- referinţe descendente – legătură de tip tată (vectorul de taţi);
- referinţe ascendente – legătură de tip părinte nod terminal (vectori pentru noduri terminale şi părinţi).

Colegiul Național “Petru Rareş” Suceava
prof. Marius UDUDEC

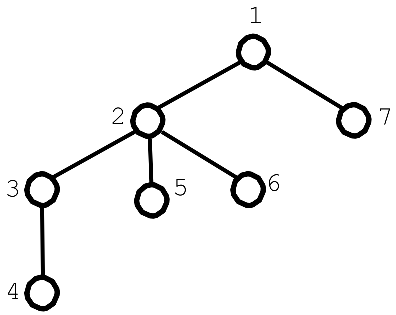
a. Reprezentare prin matricea de adiacenţă

Exemplu

Fie arborele cu rădăcină alăturat.

n=7 - numărul de noduri

m=6 - numărul de muchii



Matricea de adiacenţă asociată arborelui cu rădăcină este:

	1	2	3	4	5	6	7
1	0	1	0	0	0	0	1
2	1	0	1	0	1	1	0
3	0	1	0	1	0	0	0
4	0	0	1	0	0	0	0
5	0	1	0	0	0	0	0
6	0	1	0	0	0	0	0
7	1	0	0	0	0	0	0

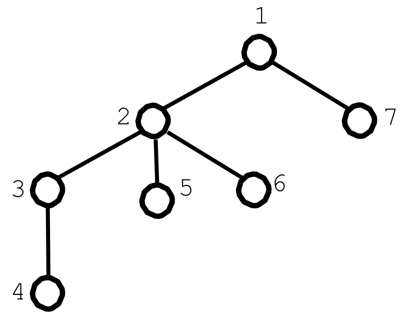
b. Reprezentare prin listă de adiacenţă

Exemplu:

Fie arborele cu rădăcină alăturat.

n=7 - numărul de noduri

m=6 - numărul de muchii



Lista de adiacenţă asociată grafului dat este:

Nod	Lista de adiacenţă
1	2, 7
2	1, 3, 5, 6
3	2, 4
4	3
5	2
6	2
7	1

c. Reprezentare prin referinţe descendente

Legătura de tip tată – se utilizează un vector `t` cu `n` componente, definit astfel:

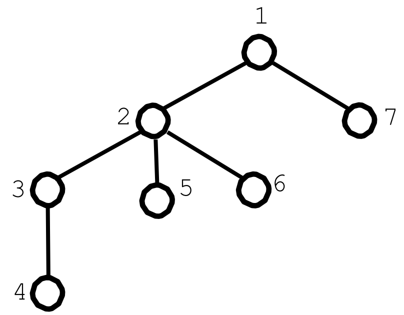
`t[i]`= eticheta părintelui nodului `i`

Exemplu:

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de vârfuri

`m=6` - numărul de muchii



Legătura de tip tată (vectorul de taţi):

t	**0**	**1**	**2**	**3**	**2**	**2**	**1**
	1	2	3	4	5	6	7

d. Reprezentare prin referinţe ascendente

Legătura de tip părinte nod terminal – se utilizează doi vectori cu `n-1` componente, definiţi astfel:

`t[i]`= nodul terminal cu valoarea cea mai mică

`pt[i]`= părintele nodului terminal `t[i]`

Construirea vectorilor:

Pasul 1: Pentru fiecare indice `i` de la `1` la `n-1` execută:

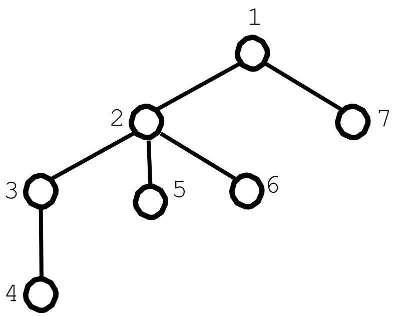
- Pasul 2: se caută nodul terminal cu eticheta cea mai mică
- Pasul 3: se atribuie această etichetă lui `t[i]`
- Pasul 4: se atribuie lui `pt[i]` eticheta nodului părinte al nodului terminal `t[i]`
- Pasul 5: se elimină din arbore nodul terminal `t[i]`

Exemplu:

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de vârfuri

`m=6` - numărul de muchii



Legătura de tip tată (vectorul de taţi):

t	4	3	5	6	2	1
pt	3	2	2	2	1	7
	1	2	3	4	5	6

3. Arbori binari

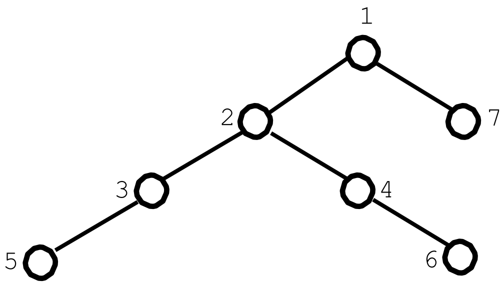
Definiţie

Se numeşte arbore binar, un arbore cu proprietatea că fiecare nod, cu excepţia nodurilor terminale, are cel mult doi descendenţi direcţi.

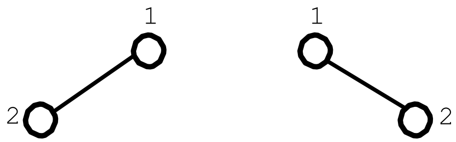
- într-un arbore binar, cei doi succesori ai unui nod (dacă există), se numesc succesorul stâng sau subarborele stâng respectiv succesorul drept sau subarborele drept;
- numărul de niveluri, mai puţin cel al rădăcinii, ocupate de nodurile arborelului, se numeşte adâncimea arborelului binar (lungimea celui mai lung lanţ care porneşte din rădăcină);

Exemple

Arbore binar:



- arborii de mai jos sunt distincţi:



4. Reprezentarea arborilor binari

Metode de reprezentare a arborilor cu rădăcină:

- matrice de adiacenţă;
- listă de adiacenţă;
- referinţe descendente – legătură de tip tată;
- vectori pentru succesor stâng şi succesor drept.

a. Reprezentare prin referinţe descendente

Legătura de tip tată – se utilizează un vector `t` cu `n` componente, definit astfel:

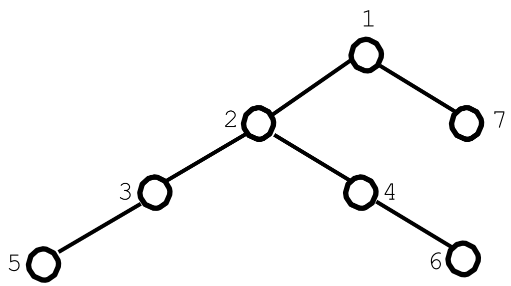
`t[i]`= eticheta părintelui nodului `i`

Exemplu

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de vârfuri

`m=6` - numărul de muchii



Legătura de tip tată (vectorul de taţi):

t	**0**	**1**	**2**	**2**	**3**	**4**	**1**
	1	2	3	4	5	6	7

Colegiul Naţional "Petru Rareş" Suceava
prof. Marius UDUDEC

b. Reprezentare prin vectori pentru succesor stâng şi succesor drept

Se utilizează doi vectori cu `n` componente fiecare, definiţi astfel:

$$st[i]=\begin{cases} j, & dacă\ \ j\ \ este\ \ succesor\ \ stâng\ \ al\ \ nodului\ \ i \\ 0, & dacă\ \ nodul\ \ i\ \ nu\ \ are\ \ succesor\ \ stâng \end{cases}$$

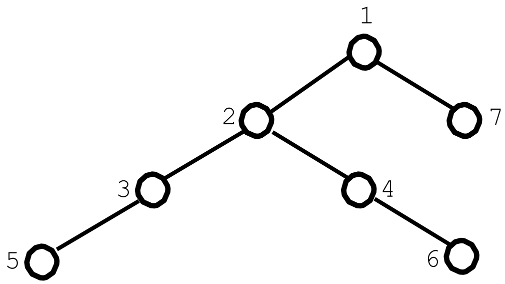
$$dr[i]=\begin{cases} j, & dacă\ \ j\ \ este\ \ succesor\ \ drept\ \ al\ \ nodului\ \ i \\ 0, & dacă\ \ nodul\ \ i\ \ nu\ \ are\ \ succesor\ \ drept \end{cases}$$

Exemplu

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de noduri

`m=6` - numărul de muchii



Legătura de tip tată (vectorul de taţi):

	1	2	3	4	5	6	7
st	2	3	5	0	0	0	0
dr	7	4	0	6	0	0	0

5. Parcurgerea arborilor binari

Metode de parcurgere a arborilor binari:

- parcurgerea în lăţime (la fel ca la grafuri)
- parcurgerea în adâncime (specific arborilor binari):
    - parcurgerea în preordine
    - parcurgerea în inordine
    - parcurgerea în postordine

a. parcurgerea în preordine – RSD

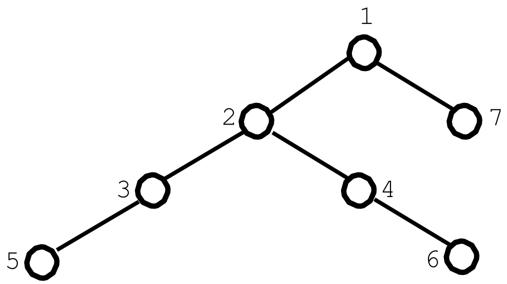
- se prelucrează rădăcina, subarborele stând, subarborele drept;

Exemplu:

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de vârfuri

`m=6` - numărul de muchii



RSD: 1, 2, 3, 5, 4, 6, 7

b. parcurgerea în inordine – SRD

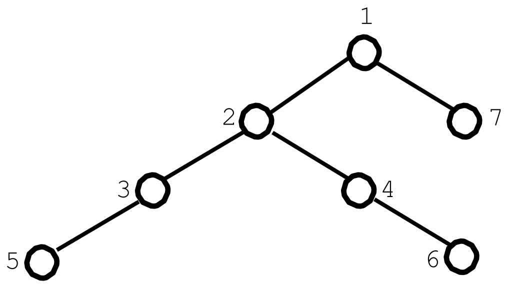
- se prelucrează subarborele stâng, rădăcina, subarborele drept;

Exemplu:

Fie arborele cu rădăcină alăturat.

n=7 - numărul de noduri

m=6 - numărul de muchii



SRD: 5, 3, 2, 4, 6, 1, 7

c. parcurgerea în postordine – SDR

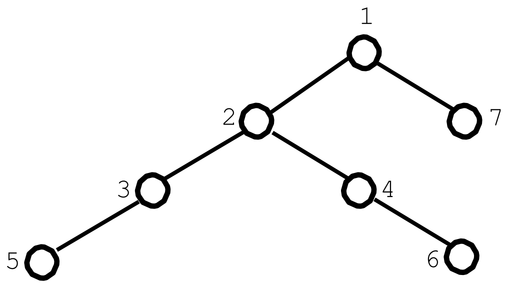
- se prelucrează subarborele stândg, rădăcina, subarborele drept;

Exemplu:

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de noduri

`m=6` - numărul de muchii



SDR: 5, 3, 6, 4, 2, 7, 1

6. Tipuri speciale de arbori binari

- arbore binar complet
- arbore binar de căutare
- heap-uri

a. arbore binar complet

Definiţie

Un arbore binar cu proprietatea că toate nodurile terminale sunt pe acelaşi nivel, se numeşte arbore binar complet.

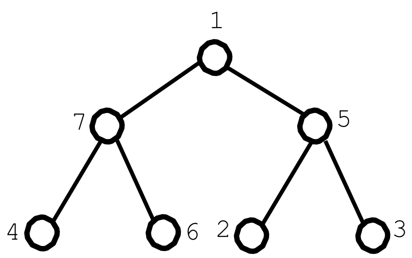
- un arbore binar complet are un număr impar de noduri;

Exemplu

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de noduri

`m=6` - numărul de muchii



Reprezentarea secvenţială a arborelului binar complet:

Se numerotează nodurile unui arbore binar complet începând cu 1 de la rădăcină şi se continuă pe niveluri, de la stânga la dreapta.
Ţinând cont de această numerotare, se observă că între nodurile arborelui există următoarele relaţii implicite:

$$\text{tata}(x) = \begin{cases} \frac{x}{2}, & \text{dacă } x > 1 \\ \text{nu există} & \text{dacă } x = 1 \end{cases}$$

$$\text{fiu stâng}(x) = \begin{cases} 2x, & \text{dacă } 2x \le n \\ \text{nu există} & \text{dacă } 2x > n \end{cases}$$

$$\text{fiu drept}(x) = \begin{cases} 2x+1, & \text{dacă } 2x+1 \le n \\ \text{nu există} & \text{dacă } 2x+1 > n \end{cases}$$

Datorită existenţei acestor relaţii, pentru a reprezenta un arbore binar complet este suficient să se reţină într-un vector informaţiile asociate nodurilor.
Această reprezentare, denumită ***secvenţială***, este optimă din punctul de vedere al complexităţii spaţiu.

b. arbore binar de căutare

Definiţie

Se numeşte arbore binar de căutare un arbore binar care are proprietatea că, pentru fiecare nod, cheia din succesorul stâng este mai mică decât cheia din nod, iar cheia din succesorul drept este mai mare decât cheia din nod.

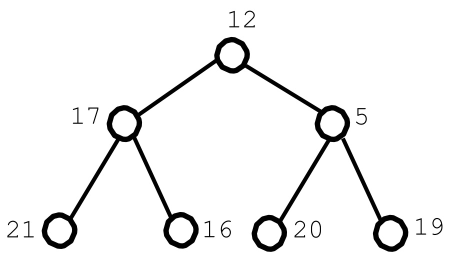
- se numeşte cheia unui nod câmpul de informaţie utilţă a nodului care poate fi folosit pentru a identifica unic nodurile arborelui;
- într-un arbore binar de căutare nu există două noduri cu aceeaşi valoare a cheii;

Exemplu:

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de noduri

`m=6` - numărul de muchii



Operaţii specifice arborilor binari de căutare:

- inserare nod
- ştergere nod
- căutare element

c. Heap-uri

Definiţie

Se numeşte Heap sau ansamblu Heap un arbore binar care îndeplineşte următoarele condiţii:

1. este un arbore aporape complet;
2. în orice pereche de noduri tată-fiu cheile sunt într-o relaţie de ordine prestabilită.

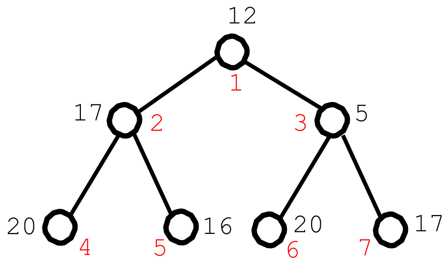
- ansamblul Heap se numeşte şi arbore de selecţie sau arbore parţial ordonat;
- un ansamblu Heap maxim este un ansamblu Heap în care cheia părintelui este mai mare sau egală cu cheia fiului;
- un ansamblu Heap maxim este un ansamblu Heap în care cheia părintelui este mai mică sau egală cu cheia fiului;
- într-un ansamblu Heap pot exista chei cu aceeaşi valoare;

Exemplu

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de noduri

`m=6` - numărul de muchii



Reprezentarea secvenţială a ansamblului Heap:

- vârfurile sunt numerotate în ordinea nivelelor iar, pe fiecare nivel numerotarea se face de la stânga la dreapta
- informaţiile relative la arbore se deduc în următoarele moduri:

$$tata[i]=\begin{cases}[i/2], & dacă \quad i\geq 2\\ nu \quad există, & dacă \quad i=1\end{cases}$$

$$st[i]=\begin{cases}i*2, & dacă \quad i*2\leq n\\ nu \quad există, & dacă \quad i*2>n\end{cases}$$

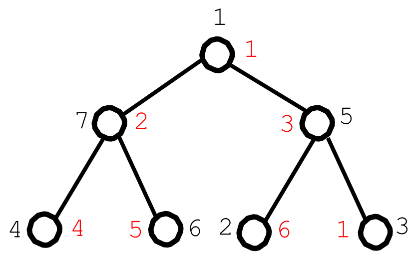
$$dr[i]=\begin{cases}i*2+1, & dacă \quad i*2+1\leq n\\ nu \quad există, & dacă \quad i*2>n\end{cases}$$

Exemplu

Fie arborele cu rădăcină alăturat.

`n=7` - numărul de noduri

`m=6` - numărul de muchii



Operaţii specifice ansamblurilor Heap:

- inserare nod
- extragerea nodului cu cheie maximă/minimă

7. Arbore parţial

Definiţie

Se numeşte arbore parţial al unui graf G un graf parţial al unui graf G care este arbore.

Teoremă

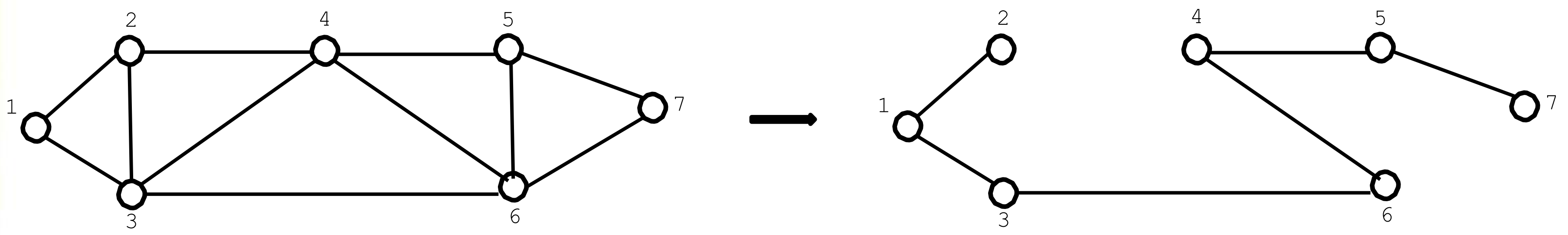
Un graf G conţine un arbore parţial dacă şi nuami dacă este un graf conex.

Exemplu

Fie graful G=(X,U)

n=7 – numărul de noduri

m=11 – numărul de muchii



8. Arbore parţial de cost minim

Definiţie

Se numeşte graf parţial de cost minim al unui graf G conex, cu funcţia de cost c, un graf parţial conex H care are costul minim.

Teoremă

Graful parţia parţial de cost minim al unui graf conex G, cu funcţia de cost c, este un arbore.

Definiţie

Arborele care este un graf parţial de cost minim al grafului conex G, cu funcţia de cost c, se numeşte arbore parţial de cost minim.

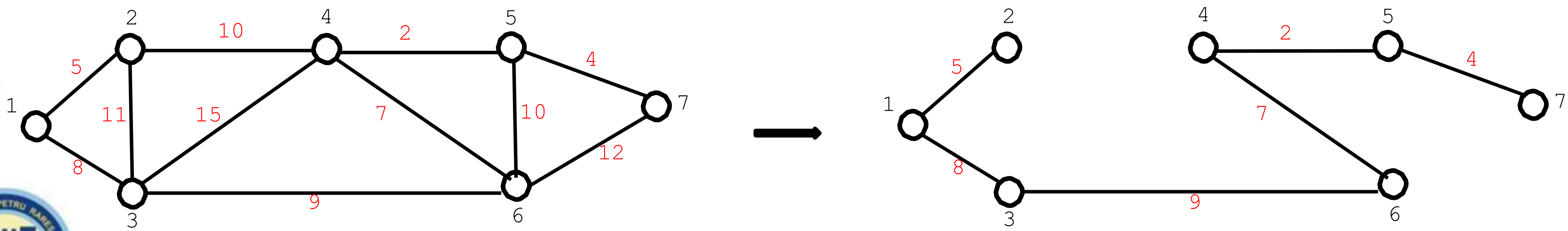
- costul grafului este suma costurilor

Exemplu

Fie graful `G=(X,U)`

`n=7` – numărul de noduri

`m=11` – numărul de muchii



Colegiul Naţional "Petru Rareş" Suceava
prof. Marius UDUDEC

Fişe de lucru

- Întrebări grafuri
- Aplicaţii grafuri
- Întrebări arbori
- Aplicaţii arbori

1. Miloşecsu M., *Informatică. Manual pentru clasa a XI*, Editura Didactică şi Pedagogică, Bucureşti, 2006
2. Mateescu G., Moraru P., *Informatica pentru liceu şi bacalaureat, clasa a XI*, Editura Donaris, Sibiu, 2006
3. Popescu C., *Culegere de probleme de informatică*, Editura Donaris-Info, Sibiu, 2002
4. Tomescu I., *Bazele informaticii*, Editura Didactică şi Pedagogică, Bucureşti, 1994
5. Ministerul Educaţiei, Cercetării şi Tineretului, Centrul Naţional pentru Curriculum şi Evaluare în Învăţământul Preuniversitar, *Proba scrisă la informatică. Examenul de bacalaureat – Variante (1-100)* , Bucureşti 2008
6. http://corina.doit.ro/graf/
7. http://www.graf.go.ro/index1.html
8. http://ro.wikipedia.org/wiki/Graf
9. http://ro.wikipedia.org/wiki/Arbore_(teoria_grafurilor)
10. http://ro.wikipedia.org/wiki/Algoritmul_lui_Kruskal
11. http://ro.wikipedia.org/wiki/Algoritmul_lui_Prim