GRAFURI NEORIENTATE. REPREZENTAREA GRAFURILOR

2. GRAFURI NEORIENTATE. REPREZENTAREA GRAFURILOR

Definitii:

- graf neorientat
- nodurile grafului, muchiile grafului
- gradul vf., nod izolat, graf d-regular
- Lant, lant simplu, elementar
- lant hamiltonian, eulerian
- lungimea unui lant, ciclu
- graf complet de ordin n
- ordin: graf bipartit complet
- graf conex
- distanta intre 2 noduri
- excentricitate nod, raza graf
- centrul grafului, diametrul grafului
- gradul exterior, interior
- matrice de incidenta
- grad exterior, interior, grad nod
- parcurgerea unui graf

- Grafuri neorientate
- Reprezentarea grafurilor (geometrica, matriceala, prin liste)
- Matricea de adiacenta
- Matricea de incidenta
- Reprezentarea vectoriala
  - a) Lista arcelor
  - b) Lista succesorilor
  - c) Lista predecesorilor
- Parcurgerea grafurilor

Algoritmi (descriere+pseudocod+ex.)

- parcurgerea in latime (Breadth First Search BFS)
- parcurgerea in adancime (Depth First Search DFS)

Proprietati. Teoreme

- graful complet de ordin n: nr. muchii
- distanta intre 2 noduri: proprietatile distantei (R,S,tr.)
- suma elementelor matricii de adiacenta = nr. arcelor grafului
- Ap = nr. drumuri (lanturi) de lungime p intre nodurile grafului

Grafuri neorientate

Definiție

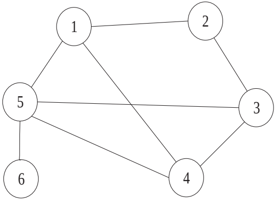
Se numește **graf neorientat** sistemul $G = (X, U)$ unde X este o mulțime de elemente numite **nodurile grafului**, iar U este o mulțime de perechi neordonate de noduri numite **muchiile grafului**.

$u = [x, y] \in U$ muchie

- x, y extremitățile muchiei u
- x, y noduri adiacente
- u muchie incidentă nodurilor x și y
- $u_1 = [x, y],\ u_2 = [x, z] \in U$ - muchii adiacente

Grafuri neorientate

$G = (X, U)$, $X = \{1, 2, 3, 4, 5, 6\}$,
$U = \{(1, 2), (1, 4), (1, 5), (2, 3), (3, 4), (3, 5), (4, 5), (5, 6)\}$



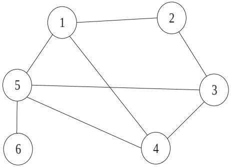
ordinul $|X| = 6 =$ numărul nodurilor

dimensiunea $|U| = 8 =$ numărul muchiilor

muchia $[1, 2]$ adiacentă cu $[1, 4]$, $[1, 5]$, $[2, 3]$

Grafuri neorientate

se numește **gradul** unui vârf x al grafului neorientat $G = (X, U)$ numărul $g(x)$ al muchiilor incidente vârfului $x \in X$



$g(1) = 3,\ g(2) = 2,\ g(3) = 3,\ g(4) = 3,\ g(5) = 4,\ g(6) = 1$

nu există grad exterior și interior al unui vârf

x cu $g(x) = 0$ se numește izolat

graful $G = (X, U)$ în care toate nodurile au același grad d se numește **graf d-regular** drum (graf orientat)= succesiune de arce a.i. extremitatea finala a unui arc din succesiune=extremitatea initiala a arcului urmator.

Definiție

Se numește **lanț** în graful neorientat $G = (X, U)$ o succesiune de muchii cu proprietatea că fiecare muchie din succesiune este adiacentă cu muchiei următoare.

lanțul se poate defini și ca succesiune de noduri adiacente

lanț simplu, elementar

lanț hamiltonian, eulerian

lungimea unui lanț este egală cu numărul muchiilor din succesiune

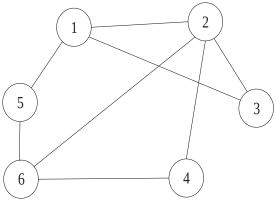
un lanț în care extremitatea inițială coincide cu extremitatea finală se numește **ciclu**

Graf orientat:

-circuit: extremitatea initiala=extremitatea finala

-drum Eulerian=toate arcele (graf eulerian=minim 1 circuit eulerian)

-drum hamiltonian=toate nodurile (graf hamiltonian: min. 1 circuit hamiltonian)



$\mu_1 = \{(1,2),(2,3),(3,1),(1,5),(5,6),(6,2),(2,4),(4,6)\}$ lanț eulerian

$\mu_2 = \{2,3,1,5,6,4,2\}$ ciclu hamiltonian - graful este hamiltonian

Eulerian=toate arcele (graf eulerian=minim 1 ciclu eulerian)
Hamiltonian=toate nodurile (graf hamiltonian: min. 1 ciclu hamiltonian)

Grafuri neorientate

subgraf

-G1 = (Y, V) subgraf al G = (X;U) daca Y=submultime X, multimea de muchii V muchii comune U si produsului cartezian Y x Y.

graf parțial

-G2 = (X; V) graf partial G = (X;U) daca V submultime muchii ale U

subgraf parțial

-G3=(Y,V) subgraf partial al G = (X;U): Y submultime noduri din X, multimea V muchii comune U si produsului cartezian Y x Y.

graf complementar

G4=(X,V) graf complementar al G = (X;U), daca V= diferenta dintre produs (X x X) \ U \ {(x; x)| x din X}

grafuri izomorfe

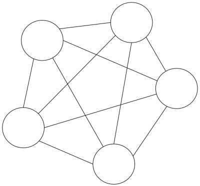
G1 = (X1;U1), G2 = (X2;U2) izomorfe: f. bijectiva (x1; x2) din U1=>(f (x1), f (x2)) din U2

Grafuri neorientate

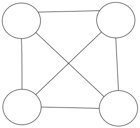
graf complet de ordin n - $G = (X, U)$ cu n noduri $|X| = n$ și $U = \{(x, y)|\ x, y \in X\}$; există muchie între oricare două noduri

graful complet de ordin n - K_n are $m = |U| = C_n^2 = \frac{n \cdot (n-1)}{2}$ muchii

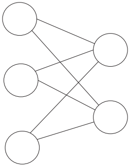
graf bipartit complet - $\exists Y \subset X$ a.i. $U = \{(x, y)|\ x \in Y,\ y \in X \setminus Y\}$
$|Y| = p$, $q = n - p$ se notează $K_{p,q}$ graful bipartit complet, $m = p \cdot q$



K_5



K_4



$K_{3,2}$

Definiție

Un graf neorientat $G = (X, U)$ se numește **graf conex** dacă există lanț între oricare două noduri ale grafului.

se numește **distanță** între două noduri $x, y \in X$ minimul dintre lungimile tuturor lanțurilor dintre nodurile x și y

distanța dintre nodurile x și y ale grafului $G = (X, U)$ se notează cu $d(x, y)$ și are proprietățile unei distanțe, adică

- $d(x, x) = 0 \; \forall \, x \in X$ - reflexivitate
- $d(x, y) = d(y, x) \; \forall \, x, y \in X$ - simetrie
- $d(x, y) \leq d(x, z) + d(z, y) \; \forall \, x, y, z \in X$ - regula triunghiului

se numește **excentricitatea** nodului $x \in X$ al grafului $G = (X, U)$ numărul

$$e(x) = max\{d(x, y) | y \in X\},$$

distanța la nodul cel mai îndepărtat

se numește **raza** grafului G numărul

$$r(G) = min\{e(x) | x \in X\},$$

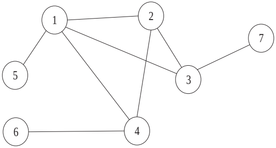
excentricitatea nodului de excentricitate minimă

se numește **centrul** grafului G nodul $x \in X$ pentru care $r(G) = e(x)$, nodul de excentricitate minimă

se numește **diametrul** grafului G numărul

$$d(G) = max\{e(x) | x \in X\}$$

Grafuri neorientate



$e(1) = 2, e(2) = 2, e(3) = 3, e(4) = 3, e(5) = 3, e(6) = 4, e(7) = 4$

- raza grafului este $r(G) = 2$
- diametrul grafului este $d(G) = 4$
- graful are ca centre nodurile 1 și 2 de excentricitate 2

$d(1,2) = 1, d(1,3) = 1, d(1,4) = 1, d(1,5) = 1, d(1,6) = 2, d(1,7) = 2,$
$d(2,1) = 1, d(2,3) = 1, d(2,4) = 1, d(2,5) = 2, d(2,6) = 2, d(2,7) = 2,$
$d(3,1) = 1, d(3,2) = 1, d(3,4) = 2, d(3,5) = 3, d(3,6) = 3, d(3,7) = 1,$

Reprezentarea grafurilor

Fie $G = (X, U)$ un graf de ordin n și dimensiune m cu mulțimea nodurilor $X = \{x_1, x_2, ..., x_n\}$ și mulțimea arcelor $U = \{u_1, u_2, ..., u_m\}$.

- reprezentare geometrică

- reprezentare matriceală

- reprezentare prin liste

Matricea de adiacență

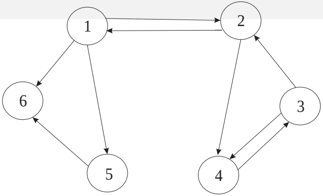
Definiție

Se numește **matrice de adiacență** a grafului $G = (X, U)$ matricea pătratică $A \in M_n(\{0,1\})$ definită prin $A = (a_{ij})_{i,j=\overline{1,n}}$,

$$a_{ij} = \begin{cases} 1, & \text{dacă } (x_i, x_j) \in U \\ 0, & \text{în caz contrar} \end{cases}.$$

G neorientat $\Rightarrow$ matricea de adiacență este simetrică

Matricea de adiacență



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

Matricea de adiacență

$\sum\limits_{j=1}^{n} a_{ij} = g^{+}(i), i = \overline{1,n}$ - suma elementelor de pe linia i a matricii de adiacență este gradul exterior al nodului i,

$\sum\limits_{j=1}^{n} a_{ji} = g^{-}(i), i = \overline{1,n}$ - suma elementelor de pe coloana i a matricii de adiacență este gradul interior al nodului i,

$\sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} a_{ij} = \sum\limits_{i=1}^{n} g^{+}(i) = |U| = m$ - suma elementelor matricii de adiacență este numărul arcelor grafului

G neorientat $\Rightarrow$

$\sum\limits_{j=1}^{n} a_{ij} = g(i), i = \overline{1,n}$, $\quad \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} a_{ij} = \sum\limits_{i=1}^{n} g(i) = 2 \cdot |U| = 2 \cdot m$

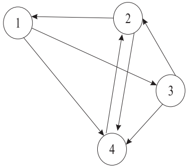
Matricea de adiacență

Teoremă

Dacă A este matricea de adiacență a grafului $G = (X, U)$ de ordin $|X| = n$ atunci matricea A^p, $p \in \mathbb{N}^*$ indică numărul drumurilor (lanțurilor) de lungime p între nodurile grafului,

$$A^p = (a_{ij}^{(p)})_{i,j=\overline{1,n}} \text{ ,}$$

$a_{ij}^{(p)}$ = numărul drumurilor (lanțurilor) de lungime p între nodurile i și j ale grafului.



$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \Rightarrow \; A^4 = \begin{pmatrix} 1 & 2 & 2 & 3 \\ 2 & 2 & 1 & 3 \\ 1 & 3 & 1 & 3 \\ 1 & 2 & 0 & 2 \end{pmatrix}$$

$a_{13}^{(4)} = 2 \Rightarrow$ există 2 drumuri de lungime 4 între nodurile 1 și 3
$a_{14}^{(4)} = 3 \Rightarrow$ 3 drumuri de lungime 4 între nodurile 1 și 4
$a_{33}^{(4)} = 1 \Rightarrow$ există un ciclu de lungime 4 ce conține nodul 3

Matricea de incidență

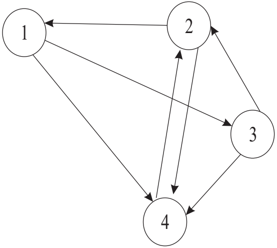
Definiție

Se numește **matrice de incidență** a grafului orientat $G = (X, U)$ de ordin $n = |X|$ și dimensiune $m = |U|$, matricea B cu n linii și m coloane și elemente $-1, 0, 1$, $B \in M_{n \times m}(\{-1, 0, 1\})$

definită prin

$$B = (b_{iu})_{i=\overline{1,n}, u=\overline{1,m}}, \; b_{iu} = \begin{cases} 1, \text{ dacă } u = (x_i, x_j) \in U \\ -1, \text{ dacă } u = (x_j, x_i) \in U \\ 0, \text{ în caz contrar} \end{cases}$$

Matricea de incidență



$$B = \begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 & 0 & -1 \\ -1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & -1 & 1 \end{pmatrix}.$$

Matricea de incidență

$| \{ b_{iu} | \ b_{iu} = 1, \ u = \overline{1, m} \} | = \ g^{+}(i), \ i = \overline{1, n}$, numărul elementelor 1 de pe linia i este gradul exterior al nodului i ,

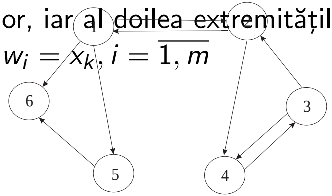
$| \{ b_{iu} | \ b_{iu} = -1, \ u = \overline{1, m} \} | = \ g^{-}(i), \ i = \overline{1, n}$, numărul elementelor -1 de pe linia i este gradul interior al nodului i ,

$\sum_{u=1}^{m} | \ b_{iu} \ | = \ g(i), \ i = \overline{1, n}$, numărul elementelor nenule de pe linia i este gradul nodului i

a) Lista arcelor

- se folosesc doi vectori v și w de dimensiune m, primul conținând extremitățile inițiale ale arcelor, iar al doilea extremitățile finale

$u_i = (x_j, x_k) \in U \Rightarrow v_i = x_j, w_i = x_k, i = \overline{1, m}$



v :	1	1	1	2	2	3	3	4	5
w :	2	5	6	1	4	2	4	3	6

b) Lista succesorilor

doi vectori: *poz* de dimensiune $n+1$ și *succ* de dimensiune m

- *succ* conține succesorii fiecărui nod
- *poz* memorează pozițiile din vectorul *succ* unde începe memorarea succesorilor pentru fiecare nod

$poz_i = j \Leftrightarrow succ_j$ este primul succesor al nodului $x_i, i = \overline{1, n}$

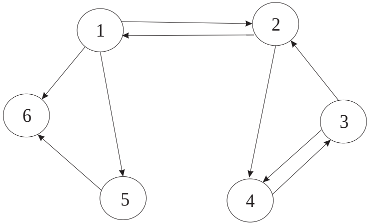
succesorii nodului x_i sunt $succ_{poz_i}, succ_{poz_i+1}, \ldots, succ_{poz_{i+1}-1}$

pe poziția poz_{i+1} în *succ* încep succesorii nodului x_{i+1}

$poz_{n+1} = m+1$, m numărul arcelor grafului doi vectori: *poz* de dimensiune $n+1$ si *succ* de dimensiune m

succ contine succesorii fiecarui nod

b) Lista succesorilor



poz memoreaza pozitiile din vectorul *succ* unde incepe memorarea succesorilor pentru fiecare nod

$\boldsymbol{poz_i = j \Leftrightarrow succ_j}$ **este primul succesor al nodului** $\boldsymbol{x_i}$ **,** $\boldsymbol{i = 1, n}$

nod:	1	2	3	4	5	6			
poz:	1	4	6	8	9	10	10		
succ:	2	5	6	1	4	2	4	3	6

poz (n+1)=m+1, nr. arce graf

succesorii 1: 2 5 6 succ 2: 1 4 succ 3: 2 4 succ 4: 3 succ 5: 6

deci succ: 2 5 6 1 4 2 4 3 6

pozitia: 1 2 3 4 5 6 7 8 9

poz(i)=j , succ(j)=primul succesor nod x(i), i = 1; n

succesorii x(i)=succ(poz(i)), succ(poz(i)+1),..., succ(poz(i+1)-1)

c) **Lista predecesorilor**

doi vectori: *poz* de dimensiune $n+1$ și *pred* de dimensiune m

- *pred* conține predecesorii fiecărui nod
- *poz* memorează pozițiile din vectorul *pred* unde începe memorarea predecesorilor pentru fiecare nod

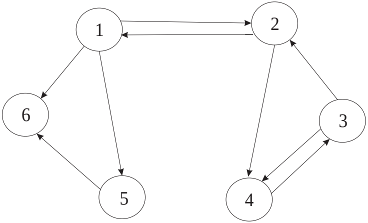
$poz_i = j \Leftrightarrow pred_j$ este primul predecesor al nodului x_i, $i = \overline{1, n}$

predecesorii nodului x_i sunt $pred_{poz_i}, pred_{poz_i+1}, \ldots, pred_{poz_{i+1}-1}$

pe poziția poz_{i+1} în *pred* încep predecesorii nodului x_{i+1}

$poz_{n+1} = m+1$, m numărul arcelor grafului

b) Lista predecesorilor



	1	2	3	4	5	6			
poz:	1	2	4	5	7	8	10		
pred:	2	1	3	4	2	3	1	1	5

predesorii 1:	2	pred 2: 1 3	pred 3: 4	pred 4: 2 3	pred 5: 1	pred 6: 1 5
pred:	2	1 3	4	2 3	1	1 5
pozitii:	**1**	**2** 3	4	5 6	**7**	8 9

prima pozitie a predecesorului in poz, poz(n+1)=m+1=10

Parcurgerea grafurilor

- parcurgerea unui graf $G = (X, U)$ presupune examinarea nodurilor sale, o singură dată fiecare nod, în vederea prelucrării acestora

- parcurgerea în lățime (Breadth First Search BFS)

- parcurgerea în adâncime (Depth First Search DFS)

Parcurgerea grafurilor

parcurgerea unui graf $G = (X, U)$ presupune examinarea nodurilor sale, o singură dată fiecare nod, în vederea prelucrării acestora

parcurgerea în lățime (Breadth First Search BFS)

parcurgerea în adâncime (Depth First Search DFS)

Parcurgerea în lățime (BFS)

$G = (X, U)$ un graf și $s \in X$ un nod sursă de la care pornește parcurgerea

- se vizitează toți succesorii/vecinii sursei nevizitați încă și primul succesor/vecin vizitat devine noul nod sursă; se continuă procedeul până nu mai există noduri nevizitate

- dacă există noduri care nu sunt accesibile din s (nu există drum/lanț de la s la acestea), acestea rămân nevizitate

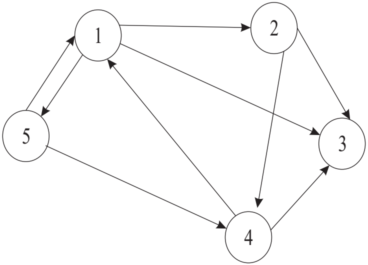
- se rețin nodurile ai căror succesori/vecini nu au fost toți vizitați într-o coadă

- *BFS* se poate utiliza pentru determinarea drumului/lanțului cel mai scurt dintre două noduri. Primul nod va fi nodul sursă, iar parcurgerea se poate opri atunci când nodul vizitat este nodul destinație (al doilea nod)

Adaugă s într-o coadă Q inițial vidă;
Pentru $x \in X$ execută $visit[x] = 0$; // nodurile nu au fost încă vizitate
$visit[s] = 1$; // nodul s se marchează vizitat
Cât timp ($Q \neq \emptyset$) execută // mai există noduri nevizitate
    $x = cap(Q)$;
    Pentru toți $y \in X$ cu $(x, y) \in U$ execută
        Dacă ($visit[y] = 0$) atunci // vecinul y al lui x nevizitat încă
            visit[y]=1; // y e vizitat
            adaugă y în coada Q; // y ultimul nod vizitat
        Sfârșit dacă;
    Sfârșit pentru;
    Se scoate x din coada Q;
Sfârșit cât timp

Parcurgerea în lățime (BFS)

- se viziteaza toti succesorii/vecinii sursei nevizitati inca si primul succesor/vecin vizitat devine noul nod sursa;
-se continua procedeul pana nu mai exista noduri nevizitate
vecinii lui 1: 2,3,5
- 2 devine nod sursa, vecinii nevizitati a lui 2: 4 ce se se adauga in lista
- 3 devine nod sursa; nu are vecini nevizitati
- 5 devine nod sursa; nu are vecini nevizitati
- 4 devine nod sursa; nu mai exista noduri nevizitate
--- nod start=4 similar
--- nod start 3, graful nu se poate parcurge, 3 nu are vecini



$s = 1 \Rightarrow 1, 2, 3, 5, 4;$

$Q : 1; \quad \underline{1}, 2, 3, 5; \quad 2, 3, 5; \quad \underline{2}, 3, 5, 4; \quad \underline{3}, 5, 4; \quad \underline{5}, 4; \quad \underline{4}; \emptyset$

$s = 4 \Rightarrow 4, 1, 3, 2, 5$

$s = 3 \Rightarrow$ graful nu se poate parcurge

Parcurgerea în lățime (BFS)

- colorare a nodurilor unui graf în timpul parcurgerii

    nodurile nevizitate încă sunt albe

    la prima vizitare un nod devine gri (în momentul introducerii în coadă) și rămâne gri cât timp are succesori/vecini nevizitați încă

    în momentul în care toți vecinii unui nod gri sunt vizitați (la scoaterea din coadă) acesta devine negru

Parcurgerea în lățime (BFS) cu distanțe

alb=nod nevizitat
gri=prima vizita nod,
cat timp are vecini nevizitati
negru=scoatere din coada

Pentru $x \in X$ execută
$color[x] = alb$; $p[x] = nil$; $d[x] = inf$;
Sfârșit pentru;
Adaugă s într-o coadă Q inițial vidă, $Q = \{s\}$;
$color[s] = gri$; $d[s] = 0$;
Cât timp ($Q \neq \emptyset$) execută
$x = cap(Q)$;
$color[x] = negru$; // x nu mai are succesori nevizitați
Pentru toți $y \in X$ cu $(x, y) \in U$ execută
Dacă ($color[y] = alb$) atunci // y nevizitat
$color[y] = gri$; // y e vizitat/prelucrat
$d[y] = d[x] + 1$; // distanța la y cu 1 mai mare decât la x
$p[y] = x$; // y vizitat ca succesor/vecin al lui x
Adaugă y în coada Q; // y e ultimul nod vizitat
Sfârșit dacă;
Sfârșit pentru;
Scoate x din coada Q;
Sfârșit cât timp;

Parcurgerea în adâncime (DFS)

se pornește de la un nod sursă s și se continuă parcurgerea cu primul succesor/vecin nevizitat al ultimului nod vizitat

dacă ultimul nod vizitat x nu mai are succesori/vecini nevizitați atunci se continuă parcurgerea cu următorul succesor/vecin nevizitat al nodul din care s-a ajuns la nodul x

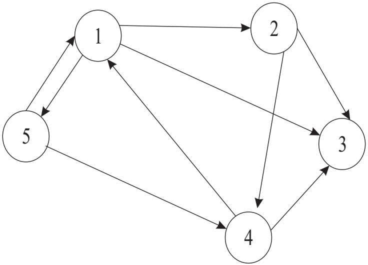
se folosește o stivă pentru stocarea nodurilor nevizitate încă

Parcurgerea în adâncime (DFS)

Adaugă s într-o stivă ST inițial vidă;
Pentru $x \in X$ execută $visit[x] = 0$; // nodurile nu au fost încă vizitate
$visit[s] = 1$; // nodul s se marchează vizitat
Cât timp ($ST \neq \emptyset$) execută // mai există noduri nevizitate
    $x = cap(ST)$;
    Dacă ($\exists y \in X, (x, y) \in U$ and $visit[y] = 0$) atunci
        visit[y]=1;
        adaugă y în stiva ST; // y devine vârful stivei
    altfel
        elimină x (vârful stivei) din ST;
    Sfârșit dacă;
Sfârșit cât timp;

Parcurgerea în adâncime (DFS)

-start: nod sursa
-continua parcurgerea cu primul succesor/ vecin nevizitat al ultimului nod vizitat
-daca ultimul nod vizitat x nu mai are vecini nevizitati atunci se continua parcurgerea cu urmatorul vecin nevizitat al nodul din care s-a ajuns la nodul x
(stiva stocare noduri nevizitate)



-start 1: vecinii 1: 2 3 5
prim vecin 1: 2
deci: 1, 2
-start 2: vecinii 2: 3 4
prim vecin 2: 3
deci: 1, 2, 3
-start 3: vecinii 3: nu exista
al doilea vecin a lui 2: 4 nevizitat
deci: 1,2,3,4
-start 4: vecin 4: 1 3 deja vizitati
al doilea vecin 1: 3 deja vizitat
urmatorul vecin 1: 5
5 nu mai are vecini nevizitati
deci: 1,2,3,4,5

$s = 1 \quad 1, 2, 3, 4, 5$

$s = 4 \quad 4, 1, 2, 3, 5.$

Parcurgerea în adâncime (DFS)

alb=nod nevizitat
gri=prima vizita nod,
cat timp are vecini nevizitati
negru=scoatere din coada

DepthFirstSearch (DFS)
Pentru $x \in X$ execută
$color[x] = alb$;
Pentru $x \in X$ execută
Dacă ($color[x] = alb$) atunci
$DFS(x)$;

$DFS(x)$
$color[x] = gri$;
Pentru fiecare $y \in X$ cu $(x, y) \in U$ execută
Dacă $color[y] = alb$ atunci
DFS(y);
Sfârșit dacă;
Sfârșit pentru;