TEMA6.**Algoritmi pe grafuri**

6.1. **Determinarea drumului Hamilton intr-un graf orientat fara circuite.**

Consideram graful $G = \langle X; U \rangle$ de ordinul $\boldsymbol{n}$ (card(X)= $\boldsymbol{n}$) redat prin matricea de adiacenţă $A_{nxn} = (a_{ij})$ $(1 \le i \le \boldsymbol{n}, 1 \le j \le \boldsymbol{n})$.

Definiţie.Numim **matrice a drumurilor**, sau matrice a conexiunilor totale matricea $D_{nxn} = (d_{ij})$ $(1 \le i \le \boldsymbol{n}, 1 \le j \le \boldsymbol{n})$, unde

d_{ij}= 1, dacă există drum din x_i în x_j şi d_{ij}= 0, dacă nu există drum din x_i în x_j.

6.1.1. **Algoritmul construirii matricei drumurilor**

Indicăm paşii pentru construirea căreiva linii ***i*** din matricea drumurilor :

Pasul 1.

Dacă linia ***i*** din matricea de adiacenţă conţine unităţi pe locaţiile $\mathbf{a_{ip}}$, $\mathbf{a_{ir}}$,......... $\mathbf{a_{iv}}$, atunci în linia ***i*** a matricei D pe locurile p, r, ...,v înscriem elementele $\mathbf{d_{ip}=d_{ir}= ...= d_{iv}=1}$ şi la elementele liniei ***i*** din matricea drumurilor se adună boolean elementele corespunzătoare din liniile p, r, ...,v din matricea de adiacenţă, generând sau nu unităţi noi.

(REMARCĂ: Suma booleană: 0+0=0
0+1=1
1+0=1
1+1=1)

Pasul 2.

Fie că unităţile noi generate în linia ***i*** din matricea drumurilor sunt: $d_i\alpha$, $d_i\beta$,.... $d_i\lambda$.

Adăugăm boolean liniile $\alpha, \beta,..., \lambda$ din matricea de adiacenţă la linia ***i*** din matricea drumurilor, generând sau nu unităţi noi în această linie.

Pasul 3. Se repeată pasul 2 pînă când se ajunge la una din situaţiile următoare:

a)toate elementele liniei ***i*** din matricea drumurilor sunt unităţi;

b)nu se mai pot genera unităţi noi în linia ***i*** şi toate locurile rămase libere se completează cu zero.

Repetăm algoritmul pentru toate liniile matricei drumurilor, obţinând matricea D.

6.1.2.Algoritmul determinării drumului Hamiltin în graful orientat fără circuite.

Definiţie: Numim putere de atingere a unui vârf X_i numărul de vârfuri care pot fi atinse din X_i şi se notează

$$P(X_i)=\sum_{j=1}^{n} d_{ij}, \quad 1 \le i \le \boldsymbol{n}$$

Din matricea drumurilor ⇒ că dacă toate elementele de pe diagonala principală sunt = 0, atunci graful este fără circuite, iar în caz contrar – există circuite.

Problema determinării **drumului hamiltonian** (drum, care trece o singură data, dar prin fiecare vârf al grafului) se rezolvă în mod diferit în graful fără circuite şi în graful cu circuite.

Pentru graful orientat fără circuite pentru determinarea drumului Hamilton se aplică :

Teorema Y.C.Chen: Fie $G = \langle X; U\rangle$ *un* graf orientat de ordinul $\boldsymbol{n}$ fără circuite. Atunci condiţia necesară şi suficientă ca să existe drum Hamilton în graf este ca suma puterilor de atingere să fie egală cu n(n-1)/2, adică:

$$\sum_{i=1}^{n} P(X_i)=\boldsymbol{n(n-1)/2}$$

Algoritmul Chen:

Pasul I. Determinăm matricea drumurilor D.

Pasul II. Determinăm puterea de atingere a fiecărui vârf x_i:

$P(X_i)=\sum_{i=1}^{n} d_{ij}$, $1 \le i \le \boldsymbol{n}$ şi se calculează suma puterilor de atingere a vârfurilor:

$$\sum_{i=1}^{n} P(X_i)$$

Pasul III. Comparăm $\sum_{i=1}^{n} P(X_i)$ cu $\boldsymbol{n(n-1)/2.}$

Dacă nu-s egale, atunci nu există drum Hamilton.

Dacă are loc $\sum_{i=1}^{n} P(X_i)=\boldsymbol{n(n-1)/2,}$ atunci există drum Hamilton şi pentru determinarea drumului concret se trece la :

Pasul IV. Aranjăm vârfurile în ordinea descreşterei puterii de atingere avirfurilor şi acesta este drumul Hamilton în graful dat.

Exemplul1. De stabilit drumul Hamilton, daca exista în graful $G = (X,F)$, unde $X=\{ X_1, X_2, X_3, X_4, X_5, X_6\}$, iar

$F(X_1)= \{ X_3, X_4, X_6\}$,

$F(X_2)= \{ X_1, X_3, X_6\}$,

$F(X_3)= \{ X_4, X_5,\}$,

$F(X_4)= \{ X_5\}$,

$F(X_5)=\{ \varnothing \}$

$F(X_6)= \{ X_3, X_5\}$

Rezolvare:

Alcătuim lista de adiacenţă

1- 3_4_6_0

2- 1_3_6_0

3- 4_5_0

4- 5_0

5- 0

6- 3_5_0

Alcătuim matricea de adiacenţă:

	X_1	X_2	X_3	X_4	X_5	X_6	
X_1	0	0	1	1	0	1	
X_2	1	0	1	0	0	1	
X_3	0	0	0	1	1	0	
X_4	0	0	0	0	1	0	
X_5	0	0	0	0	0	0	
X_6	0	0	1	0	1	0	

Alcătuim matricea drumurilor:

1)În linia I a matricei D(L_{D1}) înscriem elementele $d_{13}= d_{14}= d_{16}=1$.
La linia L_{D1} adăugăm boolean lilniile 3, 4 şi 6 din matricea de adiacenţă: L_{A3}, L_{A4}, L_{A6}.
A apărut unitate nouă - elementul d_{15}.
Adăugăm boolean linia L_{A5} (din adiacenţă) la elementele corespunzătoare ale liniei L_{D1}(din matricea drumurilor). Aşa cum L_{A5} (din adiacenţă) e formată numai din zerouri ⇒ că nu apar unităţi noi. Restul elementelor din linia L_{D1} le completăm cu zerouri.

2) În linia 2 a matricei D (L_{D2}) înscriem elementele $d_{21}= d_{23}= d_{26}=1$.
La elementele liniei L_{D2} adăugăm boolean lilniile 2, 3 şi 6 din matricea de adiacenţă: L_{A2}, L_{A3}, L_{A6}.
Au apărut unităţi noi - $d_{24} =d_{25}=1$.
Adăugăm boolean liniile L_{A4} şi L_{A5} (din adiacenţă) la elementele corespunzătoare ale liniei L_{D2}(din matricea drumurilor).
Nu apar unităţi noi. Restul elementelor din linia L_{D2} le completăm cu zerouri.

3) În linia 3 a matricei D (L_{D3}) înscriem elementele $d_{34}= d_{35}= 1$.
La elementele liniei L_{D3} adăugăm boolean lilniile 4 şi 5 din matricea de adiacenţă: L_{A4}, L_{A5}.
Nu apar unităţi noi. Restul elementelor din linia L_{D3} le completăm cu zerouri.

4) În linia 4 a matricei D (L_{D4}) înscriem elementul $d_{45} =1$.
La elementele liniei L_{D4} adăugăm boolean lilnia 5 din matricea de adiacenţă: L_{A5}.
Nu apar unităţi noi. Restul elementelor din linia L_{D4} le completăm cu zerouri.

5) Linia L_{D5} e alcătuită din zerouri fiindcă în L_{A5} toate elementele sunt zero.

6) În linia 6 a matricei D (L_{D6}) înscriem elementele $d_{63}= d_{65}=1$.
La elementele liniei L_{D6} adăugăm lilniile 3 şi 5 din matricea de adiacenţă: L_{A3}, L_{A5}.
A apărut unitate nouă – $d_{64} = 1$.
Adăugăm boolean linia L_{A4} (din adiacenţă) la elementele corespunzătoare ale liniei L_{D6}(din matricea drumurilor).
Nu apar unităţi noi. Restul elementelor din linia L_{D6} le completăm cu zerouri.

Calculăm puterile de atingere sumând în fiecare linie elementele pe orizontală.
Calculăm suma puterilor de atingere a vârfurilor sumând elementele din ultima coloană
Matricea drumurilor

	X_1	X_2	X_3	X_4	X_5	X_6	$P(X_i) = \sum d_{ij}$
X_1	0	0	1	1	1	1	$P(X_1)$ =4
X_2	1	0	1	1	1	1	$P(X_2)$ =5
X_3	0	0	0	1	1	0	$P(X_3)$ =2
X_4	0	0	0	0	1	0	$P(X_4)$ =1
X_5	0	0	0	0	0	0	$P(X_5)$ =0
X_6	0	0	1	1	1	0	$P(X_6)$ =3
							$\sum P(X_i)$=15

Aşa cum n=6 şi n(n-1)/2=6(6-1)/2=15 si

$\sum P(X_i)=\boldsymbol{n(n-1)/2},$ atunci există drum Hamilton, pe care il obtinem aranjind virfurile in ordinea descresterii puterilor de atingere:

dH=($X_2, X_1, X_6, X_3, X_4, X_5$).

Exemple pentru rezolvare individuala:
De stabilit drumul Hamilton (daca exista) în graful $G = (X,F)$, cu lista de adiacenţă:

a) 1- 2_5_0
2- 3_5_6_0
3- 6_0
4- 3_6_0
5-4_ 0
6- 0

Raspuns: dH=(1,2,5,4,3,6)

b) 1- 2_4_6_0
2- 5_0
3- 4_0
4- 1_5_0
5- 0
6- 3_5_0 Raspuns: dH=(1,6,3,4,2,5).

c) 1- 0
2- 1_3_0
3- 1_0
4- 3_5_6_0
5- 1_2_6_ 0
6- 3_0 Raspuns: dH=(4,5,2,6,3,1)

6.2. Algorimul determinării componentelor tare conexe într-un graf orientat (aplicabil pentru orice graf)

Considerăm graful $G = \langle X; U \rangle$ şi fie A_{nxn} matricea booleană de adiacenţă.

Amintim, ca componenta tare conexa afiliata virfului x_i este multimea tuturor virfurilor grafului situate cu x_i pe un circuit.

Determinam componenta tare conexa afiliata virfului x_1. Selectăm vârful x_1.

Pasul I. Determinăm mulţimea V_1 a tuturor vârfurilor, care pot fi atinse din x_1.

Pentru determinarea mulţimei V_1 se utilizează algoritmul Chen de determinare a primei linii din matricea drumurilor.

Pasul II. Determinăm mulţimea V_1^1 a tuturor vârfurilor, din care poate fi atins x_1.

Pentru determinarea mulţimei V_1^1 se utilizează algoritmul Chen de determinare a primei coloane din matricea drumurilor.

Pasul III. Determinăm componenta tare conexă care conţine x_1 conform formulei:

$C_1 = (V_1 \cap V_1^1) \cup \{x_1\}$.

Dacă $C_1 = X$, atunci graful este tare conex şi procedura s-a terminat.

Dacă $C_1 \subset X$, atunci se trece la :

Pasul IV. Se determină subgraful format prin eliminarea din G a componentei tare conexe C_1.

În matricea A se elimină liniile şi coloanele corespunzătoare vârfurilor din C_1.

În matricea rămasă se caută a doua componentă tare conexă, repetând aceiaşi paşi ai algoritmului, apoi dacă-i cazul – o a treia componentă tare conexă, până ce toate vârfurile din X vor fi grupate în componente tare conexe.

Exemplul1. De determinat componentele tare conexe în graful dat prin lista de adiacenţă:

1- 2_3_5_0

2- 5_6_7_0

3- 1_2_4_7_0

4- 1_2_5_6_0

5- 6_7_0

6- 7_8_0

7- 0

8- 5_6_7_0

Matricea de adiacenţă:

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	V_1^1	V_2^1	V_3^1
X_1	0	1	1	0	1	0	0	0	1		
X_2	0	0	0	0	1	1	1	0	0		
X_3	1	1	0	1	0	0	1	0	1		
X_4	1	1	0	0	1	1	0	0	1		
X_5	0	0	0	0	0	1	1	0	0		
X_6	0	0	0	0	0	0	1	1	0		
X_7	0	0	0	0	0	0	0	0	0		
X_8	0	0	0	0	1	1	1	0	0		
V_1	1	1	1	1	1	1	1	1	0		
V_2											
V_3											

Calculăm V_1 conform algoritmului lui Chen.

Asa cum în linia L_{A1} pe locul 2, 3, 5 se află unitatti le înscriem pe locurile corespunzătoare din linia V_1.

Adunăm boolean la linia V_1 liniile L_{A2}, L_{A3}, L_{A5}, generând 4 unităţi noi pe locurile 1, 4, 6, 7. La linia V_1 adăugăm boolean L_{A1}, L_{A4}, L_{A6}, L_{A7} generând unitate nouă pe locul 8 din V_1. Adaugam linia L_{A8} la linia V_1.

În linia V_1 toate elementele au devenit egale cu 1. Procedura cu linia V_1 este finisată.

Calculăm V_1^1 conform algoritmului lui Chen operând cu coloanele.

Asa cum în coloana C_{A1} pe locul 3 şi 4 se află 1 le înscriem pe locurile corespunzătoare din coloana V_1^1 si la ea adunăm boolean coloanele C_{A3}, C_{A4}, generând unitate nouă pe locul 1. Trebuie de adunat coloana I din adiacenţă cu V_1^1, dar noi cu dânsa am startat. Deci unităţi noi nu se mai pot genera şi restul elementelor din coloana V_1^1 se completează cu zero.

Determinăm componenta tare conexă care conţine x_1 conform formulei:

$C_1 = (V_1 \cap V_1^1) \cup \{x_1\} = \{x_1, x_3, x_4\}$

Asa cum $C_1 \subset X$, atunci se trece la pasul următor. Alcatuim matricea de adiacenţă a subgrafului obţinut suprimmând in matricea A liniile şi coloanele 1, 3, 4.(colorat verde)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	V_1^1	V_2^1	V_3^1
X_1	0	1	1	0	1	0	0	0	1		
X_2	0	0	0	0	1	1	1	0	0	0	
X_3	1	1	0	1	0	0	1	0	1		
X_4	1	1	0	0	1	1	0	0	1		
X_5	0	0	0	0	0	1	1	0	0	0	
X_6	0	0	0	0	0	0	1	1	0	0	
X_7	0	0	0	0	0	0	0	0	0	0	
X_8	0	0	0	0	1	1	1	0	0	0	
V_1	1	1	1	1	1	1	1	1	0		
V_2		0			1	1	1	1	0		
V_3											

Sau exista alternativa de a crea matricea de adiacenta pentru subgraful obtinut

	X_2	X_5	X_6	X_7	X_8	V_2^1
X_2	0	1	1	1	0	0
X_5	0	0	1	1	0	0
X_6	0	0	0	1	1	0
X_7	0	0	0	0	0	0
X_8	0	1	1	1	0	0
V_2	0	1	1	1	1	

Asa cum în linia L_{A2} pe locul 5, 6, 7 se află 1 îl înscriem pe locurile corespunzătoare din linia V_2. Adunăm boolean la linia V_2 liniile L_{A5}, L_{A6}, L_{A7}, generând unitate nouă pe locul 8. Adunam linia 8 din adiacenţă cu V_2, dar unităţi noi nu se mai pot genera şi restul elementelor din coloana linia V_2 se completează cu zero.

Calculăm V_2^1 conform algoritmului lui Chen operând cu coloanele.

Asa cum în coloana C_{A2} nu sunt unităţi în coloana V_2^1 inscriem zerouri.

Determinăm componenta tare conexă C_2 care este afiliata virfului x_2 conform formulei:

$C_2 = (V_2 \cap V_2^1) \cup \{x_2\} = \{x_2\}$.

Asa cum $C_1 \cup C_2 \subset X$, atunci se trece la pasul următor. Alcatuim matricea de adiacenţă a subgrafului obţinut suprimând din matricea prcedentă linia 2 şi coloana 2 (colorat albastru)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	V_1^1	V_2^1	V_3^1
X_1	0	1	1	0	1	0	0	0	1		
X_2	0	0	0	0	1	1	1	0	0	0	
X_3	1	1	0	1	0	0	1	0	1		
X_4	1	1	0	0	1	1	0	0	1		
X_5	0	0	0	0	0	1	1	0	0	0	1
X_6	0	0	0	0	0	0	1	1	0	0	1
X_7	0	0	0	0	0	0	0	0	0	0	0
X_8	0	0	0	0	1	1	1	0	0	0	1
V_1	1	1	1	1	1	1	1	1	0		
V_2		0			1	1	1	1	0		
V_3					1	1	1	1			

Sau in forma mai compacta matricea de adiacenta a subgrafului nou:

	X_5	X_6	X_7	X_8	V_3^1
X_5	0	1	1	0	1
X_6	0	0	1	1	1
X_7	0	0	0	0	0
X_8	1	1	1	0	1
V_3	1	1	1	1	

Asa cum în linia L_{A5} pe locul 6, 7 se află 1 le înscriem pe locurile corespunzătoare din linia V_3.
Adunăm boolean la linia V_3 liniile L_{A6}, L_{A7}, generând unitate nouă pe locul 8.
Adunăm boolean la linia V_3 linia L_{A8}, generând 1 pe locul 5, dar cu linia 5 am startat. Mai multe unitati nu pot fi generate.

Calculăm V_3^1 conform algoritmului lui Chen operând cu coloanele.
Asa cum în coloana C_{A5} pe locul 8 se află 1 îl înscriem pe locul corespunzător din coloana V_3^1 si la ea adunăm boolean coloana C_{A8}, generând unitate nouă pe locul 6. Trebuie de adunat coloana 6 din adiacenţă cu V_3^1, generând unitate nouă pe locul 5. Trebuie de adunat coloana 5 din adiacenţă cu V_3^1, dar noi cu ea am startat. Deci unităţi noi nu se mai pot genera şi restul elementelor din coloana V_3^1 se completează cu zero.

Determinăm componenta tare conexă C_3 afiliata lui x_5 conform formulei:

$C_3 = (V_3 \cap V_3^1) \cup \{x_5\} = \{x_5, x_6, x_8\}$.

Asa cum $X/(C_1 \cup C_2 \cup C_3) = \{x_7\}$ $\Rightarrow$ mai poate exista o singura componentă tare conexă afiliată lui x_7:

$C_4 = \{x_7\}$.

Deci există 4 componente tare conexe:

$C_1 = \{x_1, x_3, x_4\}$,

$C_2 = \{x_2\}$,

$C_3 = \{x_5, x_6, x_8\}$,

$C_4 = \{x_7\}$.

Exemple pentru rezolvare individuala:

De determinat componentele tare conexe în graful dat prin lista de adiacenţă:

a)1- 2_6_0

2- 5_0

3- 1_4_0

4- 2_3_0

5- 1_6_7_0

6- 7_0

7- 6_0 Raspuns: $C_1 = \{x_1, x_2, x_5\}$, $C_2 = \{x_3, x_4\}$, $C_3 = \{x_6, x_7\}$.

b)1- 2_0

2- 3_4_0

3- 1_2_0

4- 5_6_0

5- 4_0

6- 7_0

7- 6_0

Raspuns:

C_1={ x_1, x_2, x_3},

C_2= { x_4, x_5},

C_3= { x_6, x_7},

c)1- 2_0

2- 3_5_7_0

3- 4_6_0

4- 3_5_0

5- 6_0

6- 0

7-1_ 6_0 Raspuns:C_1={ x_1, x_2, x_7},C_2= { x_3, x_4}, C_3= { x_5},C_4= { x_6}.

6.3.Algoritmul determinării drumului Hamilton într-un graf orientat cu circuite.

Considerăm graful orientat $G = (X, F)$ de ordinul n cu circuite.

Fie $C_1, C_2, \ldots, C_p$ componentele tare conexe ale grafului G.

Considerăm mulţimea X^*={ $C_1, C_2, \ldots C_p$} a componentelor tare conexe ale grafului.

Definiţie.Numim **graf condensat** al grafului $G = (X, F)$ notat $G^* = (X^*, F^*)$ aplicatia mulţimii X^* pe ea însăşi, adica graful virfurile caruia sunt componentele tare conexe.

Mulţimea arcelor U^* sunt arcele de forma(C_i,C_j) si un astfel de arc este format dintr-un arc al grafului iniţial, sau din mai multe arce, care leagă vârfuri din componenta tare conexa C_i cu virfuri din componenta tare conexa C_j.

Au loc:

Propoziţia 1. Graful condensat al unui graf orientat cu circuite este un graf orientat fără circuite.

Propoziţia 2. Un graf orientat cu circuite poate avea cel puţin un drum Hamilton atunci când graful său condensat are un drum Hamilton(dar poate să nu aibă nici unul). Adică, dacă ∄ drum Hamilton în graful condensat, atunci ∄ nici în graful iniţial, iar dacă ∃ drum Hamilton în graful condensat, atunci nu întotdeauna poate să ∃ drum Hamilton în graful iniţial.

Algoritmul determinării drumului Hamilton într-un graf orientat cu circuite.

Pasul1. Determinăm componentele tare conexe ale grafului dat:

$C_1, C_2, \ldots, C_p$

Pasul2. Alcătuim o matrice A^*_{nxp} (n- numarul de virfuri, iar p – numarul de componente tare conexe), care se completeaza dupa coloane:

In fiecare coloana C_i se scrie o singura data suma booleenă a elementelor din coloanele respective ale matricei de adiacenta A a grafului G, care formeaza componenta tare conexa C_i .

Pasul3. Alcătuim o matrice patratica A^{**}_{pxp} (p – numarul de componente tare conexe), care se completeaza dupa linii:

In fiecare linie C_i a matricei A^{**}_{pxp} se scriu o singura data suma booleană a elementelor din acele linii din matricea A^{*}_{nxp} , care formeaza componenta tare conexa C_i .

Pasul4. In matricea A^{**}_{pxp} exista elemente pe diagonala principala $a^{**}_{ii}=1$, care indica ca in componenta tare conexa C_i exista circuit. Inlocuim in A^{**}_{pxp} elementele *de* pe diagonala principala *egale cu* 1 prin zerouri si obtinem matricea de adiacenta A^{***}_{pxp} a grafului condensat $G^* = (X^*,F^*)$, care este un graf fara circuite.

Pasul5. Determinam drumul Hamilton dH^* in graful G^*conform algoritmului Chen.
Aranjam componentele tare conexe in ordinea cum apar in dH^* in graful G^*.
Aflam drumul(drumurile) Hamilton in fiecare componenta tare conexa a lui G, apoi le legam cu ajutorul arcelor ce leaga componentele tare conexe intre ele in cite modalitati este posibil(in directia indicate in dH*).

Exemplul1. De determinat drumul Hamilton in graful dat prin lista de adiacenţă:

1- 2_3_5_0

2- 5_6_7_0

3- 1_2_4_7_0

4- 1_2_5_6_0

5- 6_7_0

6- 7_8_0

7- 0

8- 5_6_7_0

Aflam componentele tare conexe (vezi exemplu 1 din 6.2- toate calculele se repeta)

Matricea de adiacenţă:

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	V_1^1	V_2^1	V_3^1
X_1	0	1	1	0	1	0	0	0	1		
X_2	0	0	0	0	1	1	1	0	0		
X_3	1	1	0	1	0	0	1	0	1		
X_4	1	1	0	0	1	1	0	0	1		
X_5	0	0	0	0	0	1	1	0	0		
X_6	0	0	0	0	0	0	1	1	0		
X_7	0	0	0	0	0	0	0	0	0		
X_8	0	0	0	0	1	1	1	0	0		
V_1	1	1	1	1	1	1	1	1	0		
V_2											
V_3											

Calculăm V_1 conform algoritmului lui Chen.
Asa cum în linia L_{A1} pe locul 2, 3, 5 se află 1 le înscriem pe locurile corespunzătoare din linia V_1.
Adunăm boolean la linia V_1 liniile L_{A2}, L_{A3}, L_{A5}, generând 4 unităţi noi pe locurile 1, 4, 6, 7. La linia V_1 adăugăm boolean L_{A1}, L_{A4}, L_{A6}, L_{A7} generând unitate nouă pe locul 8 din V_1.
În linia V_1 toate elementele au devenit egale cu 1. Procedura cu linia V_1 este finisată.
Calculăm V_1^1 conform algoritmului lui Chen operând cu coloanele.
Asa cum în coloana C_{A1} pe locul 3 şi 4 se află 1 le înscriem pe locurile corespunzătoare din coloana V_1^1 si la ea adunăm boolean coloanele C_{A3}, C_{A4}, generând unitate nouă pe locul 1. Trebuie de adunat coloana I din adiacenţă cu V_1^1, dar noi cu dânsa am startat. Deci unităţi noi nu se mai pot genera şi restul elementelor din coloana V_1^1 se completează cu zero.

Determinăm componenta tare conexă care conţine x_1 conform formulei:

$C_1= (V_1 \cap V_1^1) \cup \{ x_1\}=\{ x_1, x_3, x_4\}$

Asa cum $C_1 \subset X$, atunci se trece la pasul următor. Alcatuim matricea de adiacenţă a subgrafului obţinut suprimmând din matricea A liniile şi coloanele 1, 3, 4.(colorat verde)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	V_1^1	V_2^1	V_3^1
X_1	0	1	1	0	1	0	0	0	1		
X_2	0	0	0	0	1	1	1	0	0	0	
X_3	1	1	0	1	0	0	1	0	1		
X_4	1	1	0	0	1	1	0	0	1		
X_5	0	0	0	0	0	1	1	0	0	0	
X_6	0	0	0	0	0	0	1	1	0	0	
X_7	0	0	0	0	0	0	0	0	0	0	
X_8	0	0	0	0	1	1	1	0	0	0	
V_1	1	1	1	1	1	1	1	1	0		
V_2		0			1	1	1	1	0		
V_3											

Sau exista alternativa de acrea matricea de adiacenta pentru subgraful obtinut

	X_2	X_5	X_6	X_7	X_8	V_2^1
X_2	0	1	1	1	0	0
X_5	0	0	1	1	0	0
X_6	0	0	0	1	1	0
X_7	0	0	0	0	0	0
X_8	0	1	1	1	0	0
V_2	0	1	1	1	1	

Asa cum în linia L_{A2} pe locul 5, 6, 7 se află 1 le înscriem pe locurile corespunzătoare din linia V_2. Adunăm boolean la linia V_2 liniile L_{A5}, L_{A6}, L_{A7}, generând unitate nouă pe locul 8. Adunam linia 8 din adiacenţă cu V_2, dar unităţi noi nu se mai pot genera şi restul elementelor din linia V_2 se completează cu zero.

Calculăm V_2^1 conform algoritmului lui Chen operând cu coloanele.

Asa cum în coloana C_{A2} nu sunt unităţi în coloana V_2^1 inscriem zerouri.

Determinăm componenta tare conexă C_2 care conţine x_2 conform formulei:

$C_2 = (V_2 \cap V_2^1) \cup \{x_2\} = \{x_2\}$.

Asa cum $C_1 \cup C_2 \subset X$, atunci se trece la pasul următor. Alcatuim matricea de adiacenţă a subgrafului obţinut suprimmând din matricea prcedentă linia 2 şi coloana 2 (colorat albastru)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	V_1^1	V_2^1	V_3^1
X_1	0	1	1	0	1	0	0	0	1		
X_2	0	0	0	0	1	1	1	0	0	0	
X_3	1	1	0	1	0	0	1	0	1		
X_4	1	1	0	0	1	1	0	0	1		
X_5	0	0	0	0	0	1	1	0	0	0	1
X_6	0	0	0	0	0	0	1	1	0	0	1
X_7	0	0	0	0	0	0	0	0	0	0	0
X_8	0	0	0	0	1	1	1	0	0	0	1
V_1	1	1	1	1	1	1	1	1	0		
V_2		0			1	1	1	1	0		
V_3					1	1	1	1			

Sau in forma mai compacta matricea de adiacenta a subgrafului nou:

	X_5	X_6	X_7	X_8	V_3^1
X_5	0	1	1	0	1
X_6	0	0	1	1	1
X_7	0	0	0	0	0
X_8	1	1	1	0	1
V_3	1	1	1	1	

Asa cum în linia L_{A5} pe locul 6, 7 se află 1 le înscriem pe locurile corespunzătoare din linia V_3.
Adunăm boolean la linia V_3 liniile L_{A6}, L_{A7}, generând unitate nouă pe locul 8.
Adunăm boolean la linia V_3 linia L_{A8} , generând 1 pe locul 5, dar cu linia 5 am startat, sau toate elementele din linia V_3 sunt unitati.

Calculăm V_3^1 conform algoritmului lui Chen operând cu coloanele.
Asa cum în coloana C_{A5} pe locul 8 se află 1 îl înscriem pe locul corespunzătoare din coloana V_3^1 si la ea adunăm boolean coloana C_{A8}, generând unitate nouă pe locul 6. Trebuie de adunat coloana 6 din adiacenţă cu V_3^1, generând unitate nouă pe locul 5. Trebuie de adunat coloana 5 din adiacenţă cu V_3^1 , dar noi cu ea am startat.Deci unităţi noi nu se mai pot genera şi restul elementelor din coloana V_3^1 se completează cu zero.

Determinăm componenta tare conexă C_3 care conţine x_5 conform formulei:

$C_3 = (V_3 \cap V_3^1) \cup \{x_5\} = \{x_5, x_6, x_8\}$.

Asa cum $X/(C_1 \cup C_2 \cup C_3) = \{x_7\}$ ⇒ mai poate exista o componentă tare conexă afiliată lui x_7.

$C_4 = \{x_7\}$.

Deci există 4 componente tare conexe:

$C_1 = \{x_1, x_3, x_4\}$,

$C_2 = \{x_2\}$,

$C_3 = \{x_5, x_6, x_8\}$,

$C_4 = \{x_7\}$.

Alcatuim matricea A^*_{8x4} pe care o completam dupa coloane.

In coloanal C_1 scriem suma booleana(o singura data) a elementelor din coloana 1, 3, 4 a matricei de adiacenta, fiindca $C_1 = \{x_1, x_3, x_4\}$.

In coloanal C_2 scriem elementele din coloana 2 a matricei de adiacenta, fiindca $C_2 = \{x_2\}$.

In coloanal C_3 scriem suma booleana(o singura data) a elementelor din coloana 5, 6, 8 a matricei de adiacenta fiindca $C_3 = \{x_5, x_6, x_8\}$.

In coloanal C_4 scriem elementele din coloana 7 fiindca $C_4 = \{x_7\}$.

	C_1	C_2	C_3	C_4
X_1	1	1	1	0
X_2	0	0	1	1
X_3	1	1	0	1
X_4	1	1	1	0
X_5	0	0	1	1
X_6	0	0	1	1
X_7	0	0	0	0
X_8	0	0	1	1

Alcătuim o matrice patratica A^{**}_{4x4} (p=4 – numarul de componente tare conexe), care se completeaza dupa linii:

In fiecare linie C_i a matricei A^{**}_{4x4} se scriu o singura data sumele booleene a elementelor din acele linii din matricea A^*_{8x4} , care formeaza componenta tare conexa C_i .

In linia C_1 scriem suma booleana(o singura data) a elementelor din liniile 1, 3, 4 a matricei A^*_{8x4} fiindca $C_1=\{ x_1, x_3, x_4\}$.

In linia C_2 scriem elementele din linia 2 a matricei A^*_{8x4} fiindca $C_2=\{ x_2\}$

In linia C_3 scriem suma booleana(o singura data) a elementelor din liniile 5, 6, 8 a matricei A^*_{8x4} fiindca $C_3=\{ x_5, x_6, x_8\}$.

In linia C_4 scriem elementele din linia 7 a matricei A^*_{8x4} fiindca $C_4=\{ x_7\}$.

	C_1	C_2	C_3	C_4
C_1	1	1	1	1
C_2	0	0	1	1
C_3	0	0	1	1
C_4	0	0	0	0

In matricea A^{**}_{4x4} pe diagonala principala exista si unitati pe care le inlocuim cu 0, obtinind matricea de adiacenta A^{***}_{4x4} a grafului condensat:

	C_1	C_2	C_3	C_4
C_1	0	1	1	1
C_2	0	0	1	1
C_3	0	0	0	1
C_4	0	0	0	0

Determinam drumul Hamilton daca exista in graful condensate. Aplicam algoritmul Chen pentru determinarea matricei drumurilor D^*:

La linia I adunam boolean liniile 2 , 3 si 4 ne obtinind unitate noua.

La linia 2 adunam boolean liniile 3 si 4, dar nu se obtin unitati noi.

La linia 3 adunam boolean lini 4, dar nu se obtin unitati noi.

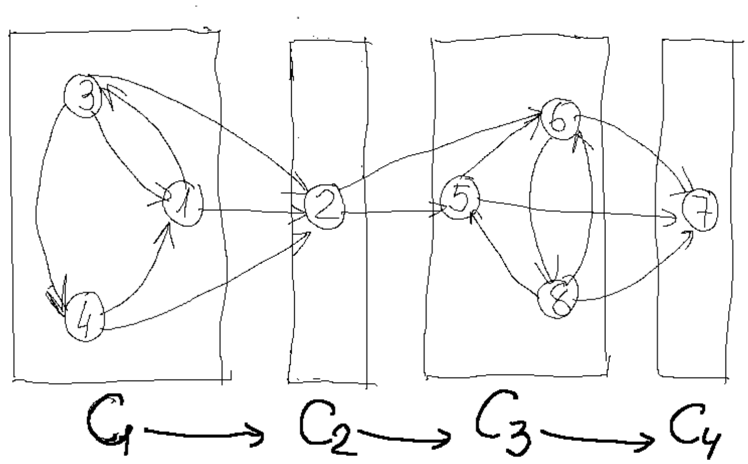
	C_1	C_2	C_3	C_4	$P(C_i)$
C_1	0	1	1	1	3
C_2	0	0	1	1	2
C_3	0	0	0	1	1
C_4	0	0	0	0	0

$\sum P(C_i) =6$

Asa cum $\sum P(C_i)=n(n-1)/2=6 \Rightarrow$ exista drum Hamilton:

$dH^*=(C_1, C_2 , C_3 ,C_4)$

Aranjam componentele tare conexe in ordinea aparitiei lor in dH*si aflam drumul(drumurile) Hamilton in fiecare componenta tare conexa a lui G, apoi le legam cu ajutorul arcelor ce leaga componentele tare conexe intre ele in cite modalitati este posibil(in directia indicate in dH*).



Am obtinut drumurile Hamilton:

dH_1=(1,3,4,2,5,6,8,7), dH_2=(1,3,4,2,6,8,5,7), dH_3=(4,1,3,2,5,6,8,7),

dH_4=(4,1,3,2,6,8,5,7), dH_5=(3,4,1,2,5,6,8,7), dH_6=(3,4,1,2,6,8,5,7).

Exemple pentru rezolvare individuala

De determinat drumul Hamilton (daca el exista) in graful dat prin lista de adiacenţă:

a)1- 2_6_0
2- 5_0
3- 1_4_0
4- 2_3_0
5- 1_6_7_0
6- 7_0
7- 6_0

b)1- 2_0
2- 3_4_0
3- 1_2_0
4- 5_6_0
5- 4_0
6- 7_0
7- 6_0

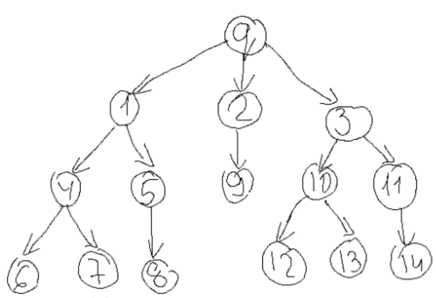
c)1- 2_0
2- 3_5_7_0
3- 4_6_0
4- 3_5_0
5- 6_0
6- 0
7-1_ 6_0

d)1- 2_6_0
2- 5_0
3- 1_4_0
4- 2_3_0
5- 1_6_7_0
6- 7_0
7- 6_0 Raspuns: Nu exista dH*.

6.4 Algoritmul parcurgerii grafului in adincime.

Algoritmul garantează vizitarea fiecărui vârf al grafului in conformitate cu următoarea proceduă recursivă:

Mai întâi e declarată şi vizitată rădăcina arborelui ***q***, apoi dacă ea nu-i frunză (lista subarborilor săi să nu fie vidă) pentru fiecare fiu ***p*** al lui ***q*** ne adresăm recursive procedurii de parcurgere în adâncime pentra vizita vârfurile tuturor subarborilor cu rădăcina ***p*** ordonate ca fii a lui ***q.***



Diferite variante de parcurgere in adincime(se afiseaza ordinea parcurgerii) :
(q,1,4,6,7,5,8,2,9,3,10,1213,11,14)
(3,10,12,13,11,14, q,1,4,6,7,5,8,2,9)e.t.c..

La parcurgerea in adincime se utilizeaza principiul LIFO(Last Input First Output), adica ultimul inscris – primul deservit. Pentru aceasta se utilizeaza din stricturi de date **STIVA**, care realizeaza principiul LIFO.

Pasul 1. Declaram o stiva vida:**S**={∅} si o lista L={∅}.

Pasul 2. Alegem radacina din virfurile nevizitate(nemarcate) o marcam si o introducem in stiva **S** si in lista L.

Pasul 3. Verificam daca **S**={∅}   Daca da, atunci ⇒Pasul 9.

Daca nu, atunci:Fie ***P*** virful din topul stivei(unica accesibila la moment).

Pasul 4. Verificam daca lista subarborilor nevizitati lui ***P*** este vida. Daca da, atunci ⇒Pasul 7.

Daca nu, atunci

Pasul 5. Vizitam, marcam si introducem in **S** si **L** fiul mai mare, nevizitat a lui ***P.***

Pasul 6. Repetam pasii P.4-P.5.

Pasul 7. Eliminam virful ***P*** din topul stivei **S.**

Pasul 8. Repetam pasii P.3 - P.7.

Pasul 9. Verificam daca toate virfurile au fost vizitate. Daca da , atunci se afiseaza **L** si **STOP.**

Daca nu, atunci trecem la Pasul 2.

Exemplul1. De determinat ordinea parcurgerii in adincime pentru graful redat prin lista de adiacenta:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

Rezolvare:

Declaram o stiva vida:**S**={∅} si o lista vida L={∅}.

Alegem radacina din virfurile nevizitate(nemarcate) (Fie virvul1) o marcam si o introducem in stiva **S** si in lista L:

S={1}

L={1}.

1- 2_3_4_0

2- 5_6_0

3- 7_8_0

4- 9_0

5- 10_0

6-0

7- 0

8- 11_0

9- 12_13_0

10- 0

11- 0

12- 0

13- 0

Verificam daca **S**={∅}.

Nu, si 1 este virful din topul stivei(unica accesibila la moment).

Verificam daca lista subarborilor nevizitati a lui ***1*** este vida..

Nu ⇒

Vizitam, marcam si introducem in **S** si **L** fiul mai mare 2, nevizitat a lui ***1.***

1- 2_3_4_0

2- 5_6_0

3- 7_8_0

4- 9_0

5- 10_0

6-0

7- 0

8- 11_0

9- 12_13_0

10- 0

11- 0

12- 0

13- 0

S={1,2}

L={1,2}.

Verificam daca lista subarborilor nevizitati a lui ***2*** este vida..

Nu ⇒

Vizitam, marcam si introducem in **S** si **L** fiul mai mare 5, nevizitat a lui ***2.***

1- 2_3_4_0

2- 5_6_0

3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

S={1,2,5}
L={1,2,5}.
Verificam daca lista subarborilor nevizitati a lui ***5*** este vida.
Nu ⇒

Vizitam, marcam si introducem in **S** si **L** fiul mai mare 10, nevizitat a lui 5***.***

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

S={1,2,5,10}
L={1,2,5,10}.
Verificam daca lista subarborilor nevizitati a lui ***10*** este vida. Da si virful 10 se elimina din topul stivei.
S={1,2,5,10}

L={1,2,5,10}.
Verificam daca lista subarborilor nevizitati a lui *5* este vida. Da si atunci 5 se elimina din topul stivei trecem la

S={1,2,5,10}

L={1,2,5,10}.

Verificam daca lista subarborilor nevizitati a lui ***2*** este vida.

Nu ⇒

Vizitam, marcam si introducem in **S** si **L** fiul mai mare 6, nevizitat a lui 2.

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

S={1,2,5,10, 6}

L={1,2,5,10,6}.

Verificam daca lista subarborilor nevizitati a lui ***6*** este vida. Da si virful 6 se elimina din topul stivei.

S={1,2,5,10, 6}

L={1,2,5,10,6}.

Verificam daca lista subarborilor nevizitati a lui ***2*** este vida. Da si virful 2 se elimina din topul stivei

S={1,2,5,10, 6}

L={1,2,5,10,6}.

Verificam daca lista subarborilor nevizitati a lui ***1*** este vida.

Nu ⇒

Vizitam, marcam si introducem in **S** si **L** fiul mai mare 3, nevizitat a lui 1.

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0

7- 0

8- 11_0

9- 12_13_0

10- 0

11- 0

12- 0

13- 0

S={1,2,5,10, 6,3}

L={1,2,5,10,6,3}.

Verificam daca lista subarborilor nevizitati a lui ***3*** este vida.

Nu ⇒

Vizitam, marcam si introducem in **S** si **L** fiul mai mare 7, nevizitat a lui 3.

1- 2_3_4_0

2- 5_6_0

3- 7_8_0

4- 9_0

5- 10_0

6-0

7- 0

8- 11_0

9- 12_13_0

10- 0

11- 0

12- 0

13- 0

S={1,2,5,10, 6,3, 7}

L={1, 2, 5,10, 6, 3, 7}.

Verificam daca lista subarborilor nevizitati a lui ***7*** este vida. Da si virful 7 se elimina din topul stivei.

S={1,2,5,10, 6,3, 7}

L={1, 2, 5,10, 6, 3, 7}.

Verificam daca lista subarborilor nevizitati a lui ***3*** este vida.

Nu ⇒

Vizitam, marcam si introducem in **S** si **L** fiul mai mare 8, nevizitat a lui 3.

1- 2_3_4_0

2- 5_6_0

3- 7_8_0

4- 9_0

5- 10_0

6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

S={1,2,5,10, 6,3, 7, 8}
L={1, 2, 5,10, 6, 3, 7,8}.
Verificam daca lista subarborilor nevizitati a lui ***8*** este vida.
Nu ⇒
Vizitam, marcam si introducem in **S** si **L** fiul mai mare 11, nevizitat a lui 8*.*

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

S={1,2,5,10, 6,3, 7, 8 ,11}
L={1, 2, 5,10, 6, 3, 7,8,11}.

Verificam daca lista subarborilor nevizitati a lui ***11*** este vida. Da si virful 11 se elimina din topul stivei.
S={1,2,5,10, 6,3, 7, 8 ,11}
L={1, 2, 5,10, 6, 3, 7,8,11}.
Verificam daca lista subarborilor nevizitati a lui ***8*** este vida. Da si virful 8 se elimina din topul stivei.
S={1,2,5,10, 6,3, 7, 8 ,11}
L={1, 2, 5,10, 6, 3, 7,8,11}.
Verificam daca lista subarborilor nevizitati a lui 3 este vida. Da si virful 3 se elimina din topul stivei.

S={1,2,5,10, 6, 3, 7, 8 ,11}
L={1, 2, 5,10, 6, 3, 7,8,11}.
Verificam daca lista subarborilor nevizitati a lui ***1*** este vida.
Nu ⇒
Vizitam, marcam si introducem in **S** si **L** fiul mai mare 4, nevizitat a lui 1.

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

S={1,2,5,10, 6, 3, 7, 8 ,11,4}
L={1, 2, 5,10, 6, 3, 7,8,11,4}.
Verificam daca lista subarborilor nevizitati a lui ***4*** este vida.
Nu ⇒
Vizitam, marcam si introducem in **S** si **L** fiul mai mare 9, nevizitat a lui 4.

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9}.
Verificam daca lista subarborilor nevizitati a lui ***9*** este vida.
Nu ⇒
Vizitam, marcam si introducem in **S** si **L** fiul mai mare 12, nevizitat a lui 9*.*

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12}.
Verificam daca lista subarborilor nevizitati a lui 12 este vida. Da si virful 12 se elimina din topul stivei.
S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12}.
Verificam daca lista subarborilor nevizitati a lui ***9*** este vida.
Nu ⇒
Vizitam, marcam si introducem in **S** si **L** fiul mai mare 13, nevizitat a lui 9

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0

12- 0
13- 0

S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
Verificam daca lista subarborilor nevizitati a lui 13 este vida. Da si virful 13 se elimina din topul stivei.
S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
Verificam daca lista subarborilor nevizitati a lui 9 este vida. Da si virful 9 se elimina din topul stivei.
S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
Verificam daca lista subarborilor nevizitati a lui 4 este vida. Da si virful 4 se elimina din topul stivei.
S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
Verificam daca lista subarborilor nevizitati a lui 1 este vida. Da si virful 1 se elimina din topul stivei.
S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
Stiva a devenit vida.
Deci, ordinea parcurgerii in adincime este
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.

Exemplul1. De determinat ordinea parcurgerii in adincime pentru graful redat prin lista de adiacenta(**varianta laconica**):

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

Rezolvare:
1)**S**={∅}
L={∅}.

2)S={1}

L={1}.

3)S={1,2}

L={1,2}.

4)S={1,2,5}

L={1,2,5}.

5)S={1,2,5,10}

L={1,2,5,10}.

6)S={1,2,5,10}

L={1,2,5,10}.

7)S={1,2,5,10}

L={1,2,5,10}.

8)S={1,2,5,10, 6}

L={1,2,5,10,6}.

9)S={1,2,5,10, 6}

L={1,2,5,10,6}.

10)S={1,2,5,10, 6}

L={1,2,5,10,6}.

11)S={1,2,5,10, 6,3}

L={1,2,5,10,6,3}.

12)S={1,2,5,10, 6,3, 7}

L={1, 2, 5,10, 6, 3, 7}.

13)S={1,2,5,10, 6,3, 7}

L={1, 2, 5,10, 6, 3, 7}.

14)S={1,2,5,10, 6,3, 7, 8}

L={1, 2, 5,10, 6, 3, 7,8}.

15)S={1,2,5,10, 6,3, 7, 8 ,11}

L={1, 2, 5,10, 6, 3, 7,8,11}.

16)S={1,2,5,10, 6,3, 7, 8 ,11}

L={1, 2, 5,10, 6, 3, 7,8,11}.

17)S={1,2,5,10, 6,3, 7, 8 ,11}

L={1, 2, 5,10, 6, 3, 7,8,11}.

18)S={1,2,5,10, 6, 3, 7, 8 ,11}

L={1, 2, 5,10, 6, 3, 7,8,11}.

19)S={1,2,5,10, 6, 3, 7, 8 ,11,4}

L={1, 2, 5,10, 6, 3, 7,8,11,4}.

20)S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9}

L={1, 2, 5,10, 6, 3, 7,8,11,4,9}.

21)S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12}

L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12}.

22)S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12}

L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12}.
23)S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
24)S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
25)S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
26)S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
27)S={1,2,5,10, 6, 3, 7, 8 ,11,4 , 9, 12,13}
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.
Stiva a devenit vida.
Deci, ordinea parcurgerii in adincime este
L={1, 2, 5,10, 6, 3, 7,8,11,4,9,12,13}.

Exemplul2. De determinat ordinea parcurgerii in adincime pentru graful redat prin lista de adiacenta:

1- 2_3_0
2- 0
3- 2_0
4- 2_5_0
5- 2_0

Rezolvare:
1)S={∅}
L={∅}.
2)S={1}
L={1}.

1- 2_3_0
2- 0
3- 2_0
4- 2_5_0
5- 2_0

3)S={1,2}
L={1,2}.

1- 2_3_0
2- 0
3- 2_0
4- 2_5_0
5- 2_0

4)S={1,2}

L={1,2}.

1- 2_3_0
2- 0
3- 2_0
4- 2_5_0
5- 2_0

5)S={1,2,3}

L={1,2,3}.

6)S={1,2,3}

L={1,2,3}.

7)S={1,2,3}

L={1,2,3}.

Lista subarborilor nevizitati este vida, dar nu toate virfurile au fost vizitate. Mergem la pasul doi si declaram radacina unul din virvurile nemarcate fie 4.

1- 2_3_0
2- 0
3- 2_0
4- 2_5_0
5- 2_0

8)S={1,2,3,4}

L={1,2,3,4}.

1- 2_3_0
2- 0
3- 2_0
4- 2_5_0
5- 2_0

9)S={1,2,3,4,5}

L={1,2,3,4,5}.

10)S={1,2,3,4,5}

L={1,2,3,4,5}.

11)S={1,2,3,4,5}

L={1,2,3,4,5}. Deci, ordinea parcurgerii in adincime este: L={1,2,3,4,5}.

6.5 Algoritmul parcurgerii grafului in lărgime.

Algoritmul la fel garantează vizitarea exact o singură data a fiecărui vârf, dar după un alt principiu:

1)După declararea si vizitarea rădăcinii se viziteayă fiecare vârf adiacent cu rădăcina aleasă.

2) Apoi se vizitează toate vârfurile adiacente cu aceste ultime vârfuri e.t.c. până vor fi vizitate toate vârfurile. Se mai numeste parcurgere în ordine orizontală (postordine) şi realizează parcurgerea vârfurilor de la stânga la dreapta - nivel după nivel.

Algoritmul:

Pasul 1. Declarăm două fire de aşteptare (FA) vide:

$FA_1=\{\emptyset\}$, $FA_2=\{\emptyset\}$ şi o lista $L=\{\emptyset\}$.

Pasul 2. Alegem dintre vârfurile nevizitate rădăcina şi o întroducem în FA_1.

Pasul 3. Verificăm dacă FA_1 este vid. Dacă da, atunci trecem la Pasul 7.

Dacă nu, atunci:

Pasul 4. Marcăm vârful ***p*** din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1.

Pasul 5. În FA_2 întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p*** începând cu cel mai mare (de la stânga la dreapta) şi care n-au fost întroduşi în FA_1.

Pasul 6. Repetăm P.3 – P.5

Pasul 7. Schimbăm cu denumirea FA_1 şi FA_2 (FA_2 devine vid). Verificăm dacă FA_1 este vid. Dacă nu, atunci $\Rightarrow$ Pasul 3.

Dacă FA_1 este vid (adică şi $FA_1=\{\emptyset\}$şi $FA_2=\{\emptyset\}$), atunci:

Pasul 8. Dacă sunt vârfuri nevizitate, atunci $\Rightarrow$ Pasul 2.

Dacă nu sunt, atunci afişăm lista L (ordinea parcurgerii) şi STOP.

Exemplul1. De determinat ordinea parcurgerii in lărgime pentru graful redat prin lista de adiacenta:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

Rezolvare:

1) Declarăm două fire de aşteptare (FA) vide:

$FA_1=\{\emptyset\}$, $FA_2=\{\emptyset\}$ şi o lista $L=\{\emptyset\}$.

2) Alegem dintre vârfurile nevizitate rădăcina şi o întroducem în FA_1. Fie rădăcina $\boldsymbol{x_1}$.

$FA_1=\{ \boldsymbol{x_1}\}$,

3) Verificăm dacă FA_1 este vid.

Nu si atunci marcăm vârful ***p,*** adică $\boldsymbol{x_1}$ din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1.

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

$L=\{ \boldsymbol{x_1}\}$, $FA_1=\{\emptyset\}$.

4)În FA_2 întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p,*** adică $\boldsymbol{x_1}$ începând cu cel mai mare (de la stânga la dreapta) şi care n-au fost întroduşi în FA_1, adică $FA_2=\{ \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}\}$.

5) Verificăm dacă FA_1 este vid. Da si atunci trecem la Pasul 7, adica schimbam cu denumirile :

$FA_1==\{ x_2, x_3, x_4\}$ şi $FA_2=\{\emptyset\}$.

Verificăm dacă FA_1 este vid. Nu, atunci $\Rightarrow$ Marcăm vârful $\boldsymbol{p} = x_2$ din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1

1- 2_3_4_0
2- 5_6_0

3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

L={ x_1, x_2},

FA_1=={x_3, x_4} .

În FA_2 întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p,*** adică x_2 începând cu cel mai mare (de la stânga la dreapta) şi care n-au fost întroduşi în FA_1, adică FA_2={ x_5, x_6}.

Verificăm dacă FA_1 este vid. Nu, atunci ⇒ Marcăm vârful ***p*** = x_3 din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

L={ x_1, x_2, x_3},

FA_1=={ x_4} .

În FA_2 întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p,*** adică x_3 începând cu cel mai mare (de la stânga la dreapta) şi care n-au fost întroduşi în FA_1, adică x_7, x_8 si FA_2={x_5, x_6, x_7, x_8}.

Verificăm dacă FA_1 este vid. Nu, atunci $\Rightarrow$ Marcăm vârful ***p*** = x_4 din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

L={ x_1, x_2, x_3, x_4},
FA_1=={ ∅} .

În FA_2 întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p,*** adică x_4 începând cu cel mai mare (de la stânga la dreapta) şi care n-au fost întroduşi în FA_1, adică x_9 si FA_2={x_5, x_6, x_7, x_8, x_9}.

Verificăm dacă FA_1 este vid. Da si atunci trecem la Pasul 7, adica schimbam cu denumirile :
FA_1={x_5, x_6, x_7, x_8, x_9} şi FA_2={∅}.
Verificăm dacă FA_1 este vid. Nu, atunci $\Rightarrow$ Marcăm vârful ***p*** = x_5 din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0

12- 0

13- 0

L={ $x_1, x_2, x_3, x_4, , x_5$},

FA_1={ x_6, x_7, x_8, x_9}

În FA_2 întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p,*** adică x_5 începând cu cel mai mare (de la stânga la dreapta) şi care n-au fost întroduşi în FA_1, adică FA_2={x_{10}}.

Verificăm dacă FA_1 este vid. Nu, atunci ⇒ Marcăm vârful $p = x_6$ din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0

2- 5_6_0

3- 7_8_0

4- 9_0

5- 10_0

6-0

7- 0

8- 11_0

9- 12_13_0

10- 0

11- 0

12- 0

13- 0

L={ $x_1, x_2, x_3, x_4, x_5, x_6$},

FA_1={ x_7, x_8, x_9}

În FA_2 trebuie sa întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p,*** adică x_6 **,** dar x_6 nu are subarbori.

Verificăm dacă FA_1 este vid. Nu, atunci ⇒ Marcăm vârful $p = x_7$ din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0

2- 5_6_0

3- 7_8_0

4- 9_0

5- 10_0

6-0

7- 0

8- 11_0

9- 12_13_0

10- 0

11- 0

12- 0
13- 0

L={ $x_1, x_2, x_3, x_4, x_5, x_6, x_7$},

FA_1={x_8, x_9}

În FA_2 trebuie sa întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p,*** adică x_7 , dar x_7 nu are subarbori.

Verificăm dacă FA_1 este vid. Nu, atunci ⇒ Marcăm vârful ***p*** = x_8 din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

L={ $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$},

FA_1={x_9}

În FA_2 întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p,*** adică x_8 începând cu cel mai mare (de la stânga la dreapta) şi care n-au fost întroduşi în FA_1, adică x_{11} si

FA_2={ x_{10}, x_{11}}.

Verificăm dacă FA_1 este vid. Nu, atunci ⇒ Marcăm vârful ***p*** = x_9 din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0

9- 12_13_0
10- 0
11- 0
12- 0
13- 0

L={ x_1, x_2, x_3, x_4 , x_5, x_6, x_7, x_8, x_9},
FA_1={∅}

În FA_2 întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p***, adică x_9 începând cu cel mai mare (de la stânga la dreapta) şi care n-au fost întroduşi în FA_1, adică x_{12} si x_{13} *si*
FA_2={ x_{10}, x_{11}, x_{12}, x_{13}}.
Verificăm dacă FA_1 este vid. Da si atunci trecem la Pasul 7, adica schimbam cu denumirile :
FA_1={ x_{10}, x_{11}, x_{12}, x_{13}} şi FA_2={∅}.
Verificăm dacă FA_1 este vid. Nu, atunci ⇒ Marcăm vârful ***p*** = x_{10} din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

L={ x_1, x_2, x_3, x_4 , x_5, x_6, x_7, x_8, x_9, x_{10}},
FA_1={ x_{11}, x_{12}, x_{13}}
În FA_2 trebuie sa întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p***, adică x_{10} , dar x_{10} nu are subarbori.
Verificăm dacă FA_1 este vid. Nu, atunci ⇒ Marcăm vârful ***p*** = x_{11} din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0

2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

L={ $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}$},
FA$_1$={ x_{12} x_{13}}
Ȋn FA$_2$ trebuie sa ȋntroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p***, adică x_{11} **,** dar x_{11} nu are subarbori.
Verificăm dacă FA$_1$ este vid. Nu, atunci ⇒ Marcăm vârful ***p*** **=** x_{12} din topul FA$_1$ ȋl ȋntroducem ȋn lista L şi-l eliminăm din FA$_1$:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

L={ $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}$},
FA$_1$={x_{13}}
Ȋn FA$_2$ trebuie sa ȋntroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p***, adică x_{12} **,** dar x_{12} nu are subarbori.

Verificăm dacă FA_1 este vid. Nu, atunci $\Rightarrow$ Marcăm vârful $\boldsymbol{p} = \boldsymbol{x_{13}}$ din topul FA_1 îl întroducem în lista L şi-l eliminăm din FA_1:

1- 2_3_4_0
2- 5_6_0
3- 7_8_0
4- 9_0
5- 10_0
6-0
7- 0
8- 11_0
9- 12_13_0
10- 0
11- 0
12- 0
13- 0

L={ $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}$},
FA_1={∅}
În FA_2 trebuie sa întroducem toţi fiii nevizitaţi (lista subarborilor) a lui ***p***, adică x_{13} , dar x_{13} nu are subarbori.
Deci FA_1={∅} si FA_2={∅}.
Asa cum toate virfurile au fost vizitate se afiseaza ordinea parcurgerii:
L={ $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}$}.

Exemplul2. De determinat ordinea parcurgerii in largime pentru graful redat prin lista de adiacenta:

1- 2_3_0
2- 0
3- 2_0
4- 2_5_0
5- 2_0

Exemplul3. De determinat ordinea parcurgerii in largime pentru graful redat prin lista de adiacenta:

1- 2_4_0
2-5_0
3- 2_4_0
4- 0
5- 4_0