

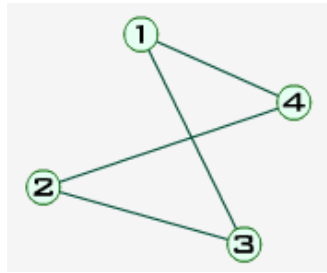
Liste de adiacență

Fie $G=(V, M)$ un graf neorientat cu n vârfuri ($V=\{1,2, \dots, n\}$) și m muchii.

Reprezentarea grafului G prin **liste de adiacență** constă în:

- precizarea **numărului de vârfuri, n** ;
- pentru fiecare vârf i , se precizează **lista L_i a vecinilor săi**, adică lista nodurilor adiacente cu nodul i .

Exemplul 1. Fie graful reprezentat în figura de mai jos:

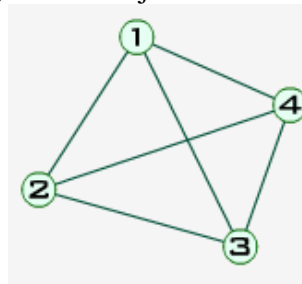


Reprezentarea sa prin liste de adiacente presupune:

- precizarea numărului de vârfuri n , $n=4$;
- precizarea listei vecinilor lui i , pentru $i=1..n$, astfel:

Vârful i	Lista vecinilor lui i
1	3,4
2	3,4
3	1,2
4	1,2

Exemplul 2. Fie graful reprezentat în figura de mai jos:



Reprezentarea sa prin liste de adiacente presupune:

- precizarea numărului de vârfuri n , $n=4$;
- precizarea listei vecinilor lui i , pentru $i= 1..n$, astfel:

Vârful i	Lista vecinilor lui i
1	2,3,4
2	1,3,4
3	1,2,4
4	1,2,3

Comentarii:

Acest mod de reprezentare se poate implementa astfel:

1. Se folosește un **tablou bidimensional T**, caracterizat astfel:

- are $n + 2m$ coloane;
- $T_{1,i}=i$ pentru $i=1..n$;
- Pentru $i=1..n$ $T_{2,i}=k$, dacă $T_{1,k}$ este **primul nod din lista vecinilor lui i**; $T_{2,i}=0$, dacă **nodul i este izolat**;
- Dacă $T_{1,j}=u$, adică u este un nod din lista vecinilor lui i , atunci:
 $T_{2,j}=0$, dacă **u este ultimul nod din lista vecinilor lui i**;
 $T_{2,j}=j+1$, dacă **u nu este ultimul nod din lista vecinilor lui i**.

Exemplu de completare a tabloului pentru graful de la exemplul 1:

Prima etapă. Se numerotează coloanele (1..n+2m) și se trec vârfurile.

1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4								

A doua etapă. Se trec în tabel vecinii lui 1, începând de la coloana 5.

1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	3	4						
5				6	0						

$T_{2,1}=5$, pentru că primul vecin (3) al lui 1 s-a trecut la coloana 5 ($T_{1,5}=3$).

$T_{2,5}=6$, pentru că următorul vecin (4) al lui 1 s-a trecut la coloana 6 ($T_{1,6}=4$).

$T_{2,6}=0$, pentru că vecinul $T_{1,6}$ (4) al lui 1 este ultimul din listă.

A treia etapă. Se trec în tabel vecinii lui 2, începând de la coloana 7.

1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	3	4	3	4				
5	7			6	0	8	0				

$T_{2,2}=7$, pentru că primul vecin (3) al lui 2 s-a trecut la coloana 7 ($T_{1,7}=3$).

$T_{2,7}=8$, pentru că următorul vecin (4) al lui 2 s-a trecut la coloana 8 ($T_{1,8}=4$).

$T_{2,8}=0$, pentru că vecinul $T_{1,8}$ (4) al lui 2 este ultimul din listă.

A patra etapă. Se trec în tabel vecinii lui 3, începând de la coloana 9.

1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	3	4	3	4	1	2		
5	7	9		6	0	8	0	10	0		

$T_{2,3}=9$, pentru că primul vecin (1) al lui 3 s-a trecut la coloana 9 ($T_{1,9}=1$).

$T_{2,9}=10$, pentru că următorul vecin (2) al lui 3 s-a trecut la coloana 10 ($T_{1,10}=2$).

$T_{2,10}=0$, pentru că vecinul $T_{1,10}$ (2) al lui 3 este ultimul din listă.

Ultima etapă. Se trec în tabel vecinii lui 4, începând de la coloana 11.

1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	3	4	3	4	1	2	1	2
5	7	9	11	6	0	8	0	10	0	12	0

$T_{2,4}=11$, pentru că primul vecin (1) al lui 4 s-a trecut la coloana 11 ($T_{1,11}=1$).

$T_{2,11}=12$, pentru că următorul vecin (2) al lui 4 s-a trecut la coloana 12 ($T_{1,12}=2$).

$T_{2,12}=0$, pentru că vecinul $T_{1,12}$ (2) al lui 4 este ultimul din listă.

2. Se foloseste un **tablou unidimensional**, cu numele **cap**, si un **tablou bidimensional**, cu numele **L** (care retine listele de vecini pentru fiecare nod), caracterizate astfel:

Tabloul **cap**:

- are **n componente**;

- $cap_i = c$, dacă **primul nod** din lista **vecinilor lui i** este trecut in tabloul **L** la coloana **c**, adică $L_{1,c}$ este **primul vecin al lui i**,

si $cap_i = 0$, dacă **nodul i este izolat**

Tabloul **L**:

- are **2m componente**;

- dacă **k** este un **vecin al nodului i**, atunci:

$L_{1,k} = k$ si $L_{2,p} = 0$, dacă **k este ultimul vecin din listă**, sau

$L_{1,p} = k$ si $L_{2,p} = p+1$ dacă **k nu este ultimul vecin din listă** (**p** este **coloana la care s-a ajuns** în tabloul **L**).

• **Exemplu** de completare a tablourilor **cap** si **L**, pentru graful de la exemplul 1.

Tabloul **cap**

1	2	3	4
1	3	5	7

Tabloul **L**

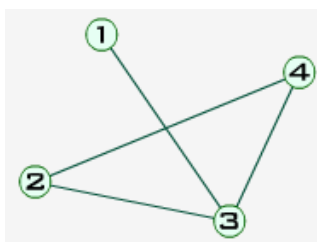
1	2	3	4	5	6	7	8
3	4	3	4	1	2	1	2
2	0	4	0	6	0	8	0

3. Se foloseste un **tablou bidimensional**, cu numele **L**, caracterizat astfel:

- are **n linii**;

- **pe linia i se trec vecini nodului i**.

• **Exemplu** de completare a tabloului **L**, pentru graful:



Tabloul **L**

3		
3	4	
1	2	4
2	3	

Implementarea în limbajul C++, a ideii prezentate mai sus, se realizează conform secvenței de program prezentată mai jos.

```

.....
int L[20][20];
int nr_vec[20];
cout<<"n="; cin>>n;
for (i=1;i<=n;i++)
{ cout<<"Dati numarul veciniilor nodului "<<i; cin>>nr_vec[i];
for (j=1;j<=nr_vec[i];j++)
cin>>L[i][j];}
.....

```

Construirea matricei de adiacență când se cunoaște L (listele vecinilor fiecărui nod).

```
.....  
for (i=1;i<=n;i++)  
{ for (j=1;j<=nr_vec[i];j++)  
a[i][L[i][j]]=1;}
```

Construirea tabloului L (listele vecinilor nodurilor) **când se cunoaște matricea de adiacență.**

```
.....  
for (i=1;i<=n;i++)  
{k=0;  
for (j=1;j<=n;j++)  
if (a[i][j]==1)  
{k=k+1 ;  
L[i][k]=j;}}
```

Sirul muchiilor

Fie $G=(V, M)$ un graf neorientat cu n vârfuri ($V=\{1,2,\dots, n\}$) si m muchii. Reprezentarea grafului G constă în precizarea numărului n de noduri si numărului m de muchii, precum si în precizarea extremităților pentru fiecare muchie în parte.

Comentarii:

Acest mod de reprezentare se implementează astfel:

1. Se dă numărul n de noduri, numărul m de muchii si extremitățile fiecărei muchii, care sunt trecute în vectorii $e1$ si $e2$ astfel:

- extremitățile primei muchii sunt $e1[i]$ si $e2[1]$;
- extremitățile celei de-a doua muchie sunt $e1[2]$ si $e2[2]$;

.....
deci $M=\{ [e1[1],e2[1]] , [e1[2],e2[2]] , \dots, [e1[m],e2[m]] \}$.

Secventa C++ corespunzătoare este:

```
int e1[100],e2[100];
int n, m, i;
.....
cout<<"n="; cin>>n;
cout<<"m="; cin>>m;
for (i=1;i<=m;i++)
{cout<<"Dati extremitatile muchiei cu numarul "<<i;
cin>>e1[i]>>e2[i];}
.....
```

Construirea matricei de adiacență, când se cunoaste sirul muchiilor ca mai sus.

```
.....
cout<<"n="; cin>>n;
for (i=1;i<=m;i++)
{ a[e1[i]][e2[i]]=1;
a[e2[i]][e1[i]]=1;}
.....
```

Construirea sirului muchiilor, ca mai sus, când se dă matricea de adiacență.

```
.....
k=0;
for (i=1;i<=n-1;i++)
for (j=i+1;j<=n;j++)
if (a[i][j]=1)
{k=k+1;
e1[k]=i;
e2[k]=j;}
m=k;
.....
```