

Laborator
Grafuri orientate

Probleme rezolvate

1) Sa se parcurga in adancime (DF) un graf orientat. Graful este dat prin matricea de adiacenta.

```
#include<fstream.h>
#include<iostream.h>
int v[20],a[20][20],n;
void citire()
{
    int i,j;
    fstream f("matrice.txt",ios::in);
    f>>n;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            f>>a[i][j];
}
void df(int nod)
{
    int k;
    cout<<nod<<" ";
    v[nod]=1;
    for(k=1;k<=n;k++)
        if((a[nod][k]==1) && (v[k]==0)) df(k);
}
int main()
{
    citire();
    df(1);
}
```

2) Matricea drumurilor - algoritmul Roy-Warshall

Se citește un graf orientat cu n noduri și m arce, dat prin vectorul arcelor. Să se construiască o matricea existenței drumurilor ($a[i][j]$ este 1 dacă există drum de la i la j și 0 în caz contrar).

Ex: Pentru grafurile alăturate matricea existenței drumurilor este:

```
1 1 0 0 1 0
1 1 0 0 1 0
1 1 1 1 1 1
1 1 1 1 1 1
1 1 0 0 1 0
1 1 1 1 1 1
```

```
#include<fstream.h>
int k,m,n,x[100],a[100][100],p[100];

fstream f("graf.in",ios::in);
fstream g("graf.out",ios::out);

void citire()
{
    int x,y;
    f>>n>>m;
    for(int i=1;i<=m;i++)
    {
        f>>x>>y;
        a[x][y]=1;
    }
}

void roy_warshall()
{
    int i, j, k;
    for(k=1;k<=n;k++)
        for(i=1;i<=n;i++)
            for(j=1;j<=n;j++)
                if(i!=j)
                    if(a[i][j]==0)
                        a[i][j]=a[i][k]*a[k][j];
}

void afisare()
{
    for(int i=1;i<=n;i++)
    {
        g<<endl;
    }
}
```

```
        for(int j=1;j<=n;j++) g<<a[i][j]<<" ";
    }
}

int main()
{
    citire();
    roy_warshall ();
    afisare();
}
```

3) Se citesc 2 grafuri orientate, unul cu n noduri si m arce, iar celalalt cu k varfuri si l arce, ambele date prin vectorul arcelor. Sa se determine daca al doilea graf este subgraf al primului.

```
#include<fstream.h>
fstream f("date1.in",ios::in);
fstream g("date2.in",ios::in);

int a[100][100],b[100][100],n,m,k,l;
void citire()
{
    int x,y,i;
    f>>n>>m;
    for(i=1;i<=m;i++)
    {
        f>>x>>y;
        a[x][y]=1;
    }
    g>>k>>l;
    for(i=1;i<=l;i++)
    {
        g>>x>>y;
        b[x][y]=1;
    }
}

int subgraf()
{
    for(int i=1;i<=k;i++)
        for(int j=1;j<=k;j++)
            if(a[i][j]!=b[i][j]) return 0;
    return 1;
}

int main()
{
```

```
citire();  
if(subgraf()) cout<<"este subgraf";  
else cout<<"NU este subgraf";  
}
```

4) Se da un graf orientat cu n varfuri si m arce prin lista arcelor. Stiind ca graful nu are varfuri izolate, determinati daca este eulerian.

Exemple:

n=5 m=4

arcele:

1 2

2 3

3 4

4 5

nu este eulerian

n=5 m=10

arcele:

1 2

2 3

3 4

4 5

5 1

1 3

3 5

5 2

2 4

4 1

este eulerian

```
#include <fstream.h>  
  
ifstream fin("date.in");  
ofstream fout("date.out");  
int a[50][50],n,m,ok;  
int x[50], p[50], gfd_ext[50], grd_int[50];  
  
void df(int nod)  
{
```

```
int k;
p[nod]=1;
for(k=1;k<=n;k++)
    if((a[nod][k]==1 || a[k][nod]==1) && !p[k]) df(k);
}

int main()
{
    int v1,v2;
    fin>>n>>m;
    for(int i=1;i<=m;i++)
    {
        fin>>v1>>v2;
        a[v1][v2]=1;
        grd_ext[v1]++;
        grd_int[v2]++;
    }
    df(1);
    ok=1;
    for(int i=1;i<=n;i++)
        if(!p[i] || grd_ext[i]!=grd_int[i]) ok=0;
    if(ok) fout<<"da";
    else fout<<"nu";
    fin.close();
    fout.close();
}
```

5) Se da un graf orientat cu n varfuri si m arce prin lista arcelor. Se citeste apoi un numar k si o submultime X cu k varfuri din multimea varfurilor grafului, notata cu V. Afisati arcele din graful citit care au proprietatea ca au o extremitate in multimea X si cealalta in multimea V-X.

Exemplu:

6 7 (n,m)

1 2

2 1

3 4

1 5

4 5

5 6

6 1

3 (k)

1 2 3 (multimea x)

Arcele cerute sunt:

3 4

1 5

6 1

```
#include <fstream.h>
ifstream fin("date.in");
ofstream fout("date.out");
int n,m,k,v[100],a[100][100],p[100];

void citire()
{
    int x,y;
    fin>>n>>m;
    for(int i=1;i<=m;i++)
    {
        fin>>x>>y;
        a[x][y]=1;
    }
    fin>>k;
    for(int i=1;i<=k;i++)
    {
        fin>>v[i];
        p[v[i]]=1;
    }
}

void afisare()
{
    for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++)
    if(i!=j)
    {
        if(a[i][j]==1&& p[i]!=p[j])fout<<i<<" "<<j<<endl;
    }
}

int main()
{
    citire();
    afisare();
    fin.close();
    fout.close();
}
```

Probleme propuse spre rezolvare

1. Să se ruleze programele prezentate mai sus, urmărind apelurile și valorile parametrilor de apel.

2. Celebritate.

Se dă un grup format din n persoane, care se cunosc sau nu între ele. De la tastatură se introduc m perechi de numere întregi (x,y) cu semnificația "persoana x cunoaște pe persoana y ". relația de cunoștință nu este neapărat reciprocă. Numim celebritate, o persoană care este cunoscută de către toate celelalte persoane din grup, dar ea nu cunoaște pe nici un alt membru al grupului. Să se determine dacă din grup există o astfel de celebritate.