matricea distantelor:

#include"stdafx.h"

#include<fstream>

usingnamespace std;

ifstream fin("dateg2.txt");

ofstream fout("g.out");

constint maxx = 100000;

int a[100][100], k, n, m, p[100];

void citire()

{

int i, j, x;

fin >> n >> m;

for (i = 1; i <= n; i++)

for (j = i + 1; j <= n; j++)

a[i][j] = a[j][i] = maxx;

for (x = 1; x <= m; x++)

{

fin >> i >> j;

a[i][j] = 1;

a[j][i] = 1;

}

}

void rf()

{

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]>a[i][k] + a[k][j])

a[i][j] = a[i][k] + a[k][j];

}

void afis()

{

int i, j;

for (i = 1; i <= n; i++)

{

for (j = 1; j <= n; j++)

if (a[i][j] != maxx) fout << a[i][j] <<" ";

else fout <<"inf ";

fout <<"\n";

}

}

int main()

{

citire();

rf();

afis();

fin.close();

fout.close();

return 0;

}

conexitate

#include"stdafx.h"

#include<fstream>

#include<iostream>

usingnamespace std;

ifstream fin("dateg2.txt");

ofstream fout("g.out");

int a[100][100], k, n, m, viz[100];

void citire()

{

int i, j, x;

fin >> n >> m;

for (x = 1; x <= m; x++)

{

fin >> i >> j;

a[i][j] = 1;

a[j][i] = 1;

}

}

void dfs(intk)

{

int i;

fout <<k<<" ";

viz[k] = 1;

for (i = 1; i <= n; i++)

if (a[k][i] == 1 && viz[i] == 0)

dfs(i);

}

// verificare daca graful este conex

int conex()

{

int i;

dfs(1);

for (i = 1; i <= n; i++)

if (viz[i] == 0) return 0;

return 1;

}

int main()

{

citire();

if (conex() == 1) cout <<"graful este conex";

else cout <<" graful este NEconex";

// afisare componentele conexe ale grafului

for (k = 1; k <= n; k++)

if (!viz[k])

{

dfs(k);

fout << endl;

}

fin.close();

fout.close();

return 0;

}