TEORIA GRAFURILOR - DEPTH FIRST SEARCH

DINU ANDREI-RAZVAN,CLASA A XII-A C,COLEGIUL NATIONAL IULIA HASDEU

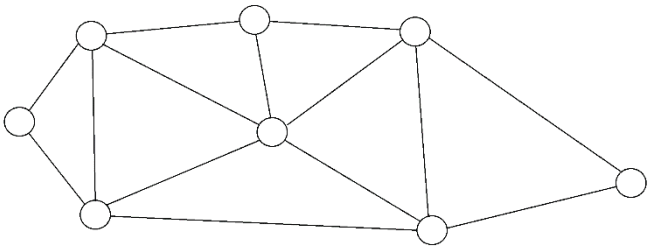
CORINA TEODOR- COLEGIUL NATIONAL IULIA HASDEU

Jocurile și amuzamentele matematice au fost punctul de plecare în ceea ce astăzi numim „teoria grafurilor”. Dezvoltându-se la început paralel cu algebra, această ramură a științei a căpătat în timp atât formă cât și conținut propriu, devenind un tot unitar bine conturat și bine fundamentat teoretic, cu o largă aplicare practică.

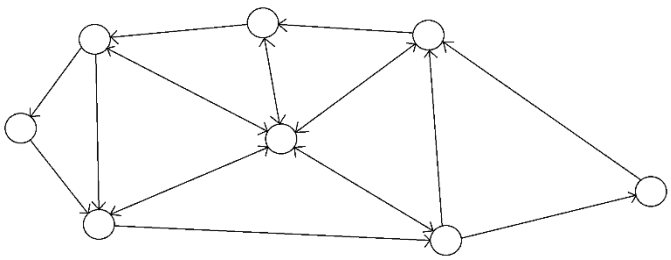
Un graf este o structură care corespunde unui grup de obiecte, în care unele perechi de obiecte sunt într-un anumit sens "legate" reciproc. Mai exact, este o multime de noduri, legate intre ele printr-o multime de muchii(pentru grafuri neorientate) sau arce(pentru grafuri orientate).

Pentru a intelege mai bine acest concept, vom lua ca exemplu reteaua de strazi a unui oras, in care intersectiile sunt reprezentate prin noduri, iar strazile prin muchii sau arce. Pentru pietoni, orasul este un graf neorientat deoarece se pot deplasa pe strada in ambele sensuri. Nu putem spune acelasi lucru si din perspectiva soferilor. Ca in orice oras exista strazi cu sens unic, care limiteaza sensul de deplasare al vehiculelor, acesta devenind un graf orientat. Vom reprezenta o portiune de oras astfel:

- Pentru pietoni:



- Pentru soferi:



Rezolvarile bazate pe teoria grafurilor a unor probleme care se pot ivi in diferite domenii de activitate sunt realizate prin intermediul unor algoritmi specifici. Unul dintre acestea este căutarea sau parcurgerea în adâncime (Depth-first search, abreviat **DFS**). Este un algoritm pentru parcurgerea sau căutarea într-o structură de date de tip arbore sau graf. Se începe de la rădăcină (sau alegând un nod arbitrar ca rădăcină în cazul unui graf) și se explorează cât mai mult posibil de-a lungul fiecărei ramuri înainte de a se intoarce de unde a plecat.

O versiune a DFS-ului a fost cercetată în secolul al XIX-lea de către matematicianul francez Charles Pierre Trémaux ca o strategie pentru rezolvarea de labirinturi. Din acest motiv, fiind un adept al provocarilor, m-am hotarat sa realizez un program in C++ destinat rozolvarii unui labirint.

Am construit un labirint din caractere pe care l-am citit de la tastatura intr-un vector bidimensional. ‘S’ reprezinta punctul de plecare, ‘F’ punctul de final, ‘|‘ reprezinta peretii labirintului, iar ‘ ‘ drumurile din acesta.

In continuare, am privit fiecare element din matricea de caractere ca pe un nod dintr-un graf, in care doua noduri sunt adiacente daca si numai daca corespondentii lor din labirint indeplinesc urmatoarele conditii: se afla unul langa celalat(dar nu pe diagonala) si amandoi retin caracterul ‘ ‘.

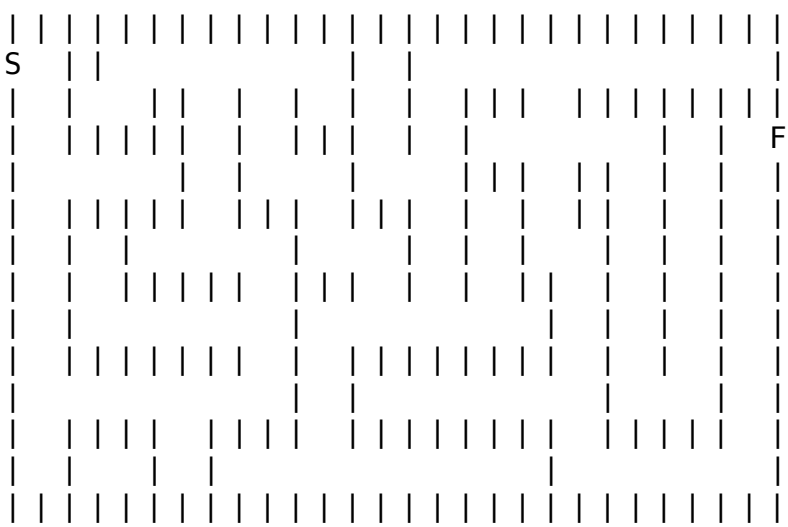
Pentru parcurgerea labirintului si marcarea solutiei acestuia am utilizat un algoritm DFS recursiv care se foloseste de matricea de adiacenta a grafului si de un vector de vizite.

```
#include<iostream>

#include<string.h>

using namespace std;

char a[100][100];
int m,n,F,b[10000][10000],v[10000];

void DFS(int x, int&ok)
{
   if(x!=F-1 && ok==0)
   {
      v[x]=1;
      for(int i=0;i<n*m;i++)
      {
         if(b[x][i]==1 && v[i]==0 && ok==0)
         {
            DFS(i,ok);
         }
      }
   }
   else
   {
      ok=1;
   }
   if(ok==1)
   {
```

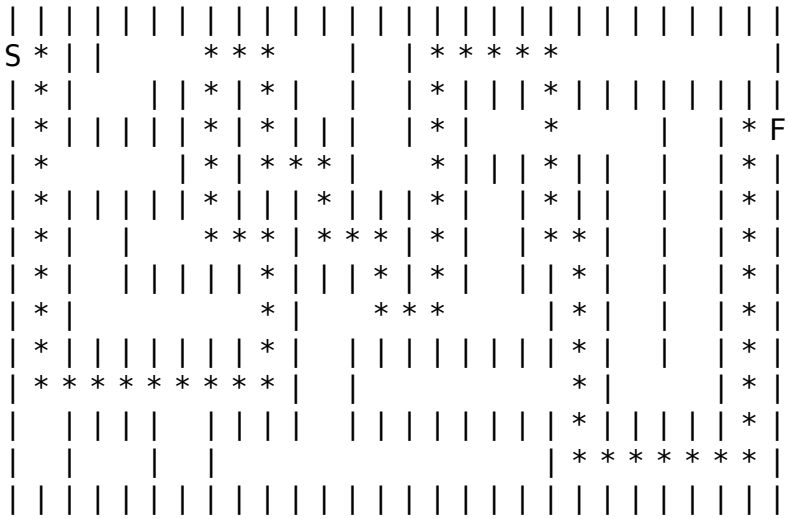
```
        a[x/n][x%n]='*';
    }
}

int main()
{
    int i,j,s;
    cin>>m>>n;
    cin.get();
    for(i=0;i<m;i++)
    {
        cin.getline(a[i],n);
    }

    for(i=0;i<m-1;i++)
    {
        for(j=0;j<n-1;j++)
        {
            if(a[i][j]==' ')
            {
                if(a[i][j+1]==' ' && j+1!=n)
                {
                    b[i*n+j][i*n+j+1]=1;
                    b[i*n+j+1][i*n+j]=1;
                }
                if(a[i+1][j]==' '  && i+1!=m)
                {
                    b[i*n+j][(i+1)*n+j]=1;
                    b[(i+1)*n+j][i*n+j]=1;
                }
            }
            else
            {
                if(a[i][j]=='S')
                {
                    s=i*n+j+1;
                }
                if(a[i][j]=='F')
                {
                    F=i*n+j;
                }
            }
        }
    }

    int ok=0;
    DFS(s,ok);

    cout<<endl<<endl;
    for(i=0;i<m;i++)
    {
```

```
            cout<<a[i]<<endl;
          }
          return 0;
        }
```

```
Date de intrare:
14 29
```



```
Date de iesire:
```



In final, consider ca este momentul sa raspund la cele doaua intrebari.

1. Am ales aceasta tema deoarece am fost surprins de numeroasele situații din viața cotidiană care pot fi modelate cu ajutorul teoriei grafurilor, asa cum a mentionat si marele matematician si informatician roman Grigore Constantin Moisil : “Azi teoria grafurilor a devenit o disciplină majoră, deşi nu-şi găseşte locul într-o clasificare dogmatică a capitolelor matematicii. Folosirea teoriei grafurilor în domenii variate, de la chimie, la economie, de la studiul reţelelor electrice la critica textelor şi la politică, îi dau azi un prestigiu de care cel ce clasifică ştiinţele trebuie să ţină seama”.

2. Prin intermediul acestei lucrari vreau sa raspund la probabil cea mai comuna intrebare in randul elevilor: “Cu ce ma ajuta in viata?”. Recunosc ca si eu mi-am pus aceasta intrebare de cateva ori, insa cu timpul am ajuns la concluzia ca din orice lectie care mi-a fost predata, din orice documentar vizionat, sau din orice carte citita pot extrage informatii care au intradevar aplicabilitate in viata de zi cu zi si care ma indruma spre o mai buna intelegere a lumii in care traim.