Programe de aplicații

IMPLEMENTAREA UNUI ALGORITM PENTRU DETERMINAREA DRUMURILOR MINIME ÎNTR-UN GRAF

Mat. Draga Bălan

Software ITC - Filiala Suceava

Considerăm un graf G(X,U), cu mulțimea vîrfurilor $X=\{x_1,x_2,....,x_n\}$ și $I=\{1,2,...N\}$, mulțimea indicilor vîrfurilor, iar U=mulțimea arcelor. Fiecărui arc (x_k,x_j) din U i se asociază o valoare pozitivă reprezentînd lungimea sa.

În practică, aceasta poate însemna durata între două operații, distanța între două localități, costul unei piese etc.

Notăm $L=(l_{kj})$ matricea definită prin:

O	cînd k=j
$l_{kj}=l_{kj}$	cînd (X_k,X_j) există în U
infinit	cînd (X_k,X_j) nu există în U

Valoarea infinită se consideră pozitivă atunci cînd căutăm drumurile minime și negativă atunci cînd le căutăm pe cele maxime.

Adoptăm următoarele operații:

$a * b = a + b$

$a + b$ = mimimum/maximum (a,b)

Algoritmul pe care-l implementăm, numit algoritmul lui Perth, constă în următoarele:

1. - aplicîndu-se operațiile definite mai sus se calculează puterile succesive ale matricii L;
2. - dacă $L^k = L^{k-1}$, stop, altfel se repetă operația de la pasul 1.

Matricea L care satisface egalitatea $L^{k-1} = L^k$, reprezintă matricea distanțelor minime între oricare două noduri ale grafului.

Practic, foarte rar pot apare surprize în aplicarea acestui algoritm, datorită unor restricții matematice ce trebuiesc impuse, dar a căror tratare nu este oportună aici.

Implementarea algoritmului este făcută cu ajutorul unui program în limbaj de programare C, pe calculatoare compatibile IBM-PC.

Tratarea programului este simplistă pentru a încuraja începătorii în învățarea acestui limbaj.

Pentru valoarea infinit s-a considerat 10 la o putere suficient de mare pentru a fi greu de atins de datele practice, iar graful de prelucrat este prevăzut cu un număr maxim de 30 noduri.

Aceste valori pot fi ușor modificate în program fiind definite ca niște constante simbolice.

Problema a fost considerată pentru determinarea drumurilor minime, în caz contrar, în locul funcției minim(), va trebui apelată o funcție similară pentru determinarea valorilor maxime.

Lansăm aceasta ca exercițiu pentru cei interesați.

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#define    N   30
#define    Q   1000
void main(void) /* algoritm p */
{
  int a=10,b,c,i,j,k,n;
  int 1[N][N], m[N][N], v[N], y[N];
  /*1 matricea drumurilor inițiale */
  /*m matricea drumurilor minime */
  /*v vector cu distanțele unui nod i la celelalte */
  /*y vector manevra*/
  clrscr ();
  gotoxy(1,1);
  do {
      printf("\ n CÎTE NODURI ARE GRAFUL (<30)?\t");
      scanf("%d", &n);
}
while(n>N);
for(i=0; i<n; i++)s/* inițializare matrice */
   for(j=0; j<n; j++){
   1[i] [j] = Q;
   1[i] [i] = 0;

    printf(" n n tINTRODU MATRICEA COSTURILOR
N");
  printf("la sfirșit tastează 99 în extremitate început");
  printf("\n\n EXTREMITĂȚI ARC VALOARE");
  printf(" \n început sfîrșit (cost) \n");
  while (i ! = 99) {
    a++;
    gotoxy(5,a);
    scanf("%d" ,&i);
    if(i == 99)
      break;
    gotoxy(15,a);
    scanf("%d" ,&j);
    gotoxy(25,a);
    scanf("%d" ,&l [ i-l ] [ j-l] );
    if(i==j)
      printf(" \ teroare \ t se ignoră");
      l [ i-l ] [ j-l]  = 0;
      }
    printf(" \ n");
}
  printf(" \ n MATRICEA COSTURILOR ESTE:");
  printf(" \ n    ");
  for(i=0;i<n; i++)
    printf(" %d ",i+l);
  printf(" \ n ---------- n");
```

```
for(i=0; i<n ;i++) {
   printf(" \ n %d  | ",i+1);
   for(j=0; j<n ; j++)
     if(l [ i ] [ j ] ==Q)
        printf("  ");
 else
   printf(" %d " , l [i] [j] ;
}
   printf("\ n")
   b=0;
   for(i=0;i<n ;i++)
     for(j=0;j ;j++)
       m [ i ] [ j ] =0;
       for(k=0;k<n ;k++)
       {  if(l [ i ] [ k ]  ==Q&&l  [ k ] [ j ] ==Q)
                v  k  =Q;
      else
                v  [k]  =l [ i] [ k ] + l [ k ] [ j ];
              }
              }
 *y  =  *v;
  c =minim(v,n);
  m [ i ] [j]  = v  [c ];
}
  for(i=0;i<n ;i++)
    for(j=o;j<n; j++)
      if(l [ i]  [ j ] !=m  [ i]  [ j ] )
           b=1;
  if(b==1)
    for(i=0;i<n; i++)
      for(j=0;j<n; j++)
        l  [i]  [j] ==m [ i] [j];
}
 while(b==1);
 printf(" \ n \ n");
 printf(" MATRICEA DRUMURILOR MINIME ESTE: \n ");
  for(i=0;i<n; i++)
    printf( "  %d  ",i+1);
    printf(" [ \ ----------------------- \ n");
    for(i=o;i<n; i++)
      printf(" \ n %d  | ",i+1);
      for(j=0;j<n; j++)
        if(l  [i]  [ j ] ==Q)
           printf("  ");
        else
           printf("  %d  ",l  [ i]  [ j ]);
      }
}
  minim (y,n)
  int n,y [] ;
{
    int p, min, c=0;
    min=y  [0 ] ;
    for(p=o;p<n; p++)
      if(min>y  [ p ] )
         { min=y  [ p ] ;
         c=p;
         }
     return(c);
}
```