

Ministerul Educației, Culturii și Cercetării  
al Republicii Moldova  
Universitatea Tehnică a Moldovei  
Departamentul Ingineria Software și Automatică

# RAPORT

Lucrarea de laborator Nr.3  
la Matematica specială

**Tema :** Algoritmul lui Bellman-Calaba și Ford pentru  
determinarea drumului minim (maxim) .

A efectuat :

gr. SI – 201 , Ivanova Evghenia

A verificat :

asis. univ. Popovici Nadejda

Chișinău – 2021

## Scopul Lucrării :

- Studierea algoritmilor de determinare a drumurilor minime într-un graf.
- Studierea algoritmilor de determinare a drumurilor maxime într-un graf.
- Elaborarea programelor de determinare a drumului minim și maxim într-un graf ponderat cu ajutorul algoritmului Ford și algoritmului Bellman-Calaba.

## Sarcina de bază :

1. Elaborați un program cu următoarele posibilități :
  - introducerea grafului în calculator,
  - determinarea drumului minim și maxim în graful ponderat cu ajutorul algoritmului Ford și algoritmului Bellman-Calaba;
  - vizualizarea rezultatelor la display și imprimantă.

## Codul programului :

```
#include <stdio.h>

int main()
{
    int i, j, k=1, v, min, max;
    printf("\tNr de varfuri "); scanf("%d",&v);
    int H[v], x[v][v], V[5][v], l[v];
    for(i=1; i<=v; i++)
        for(j=1; j<=v; j++)
        {
            if(i==j){ x[i][j]=0; printf("x%d -> x%d= %d\n",i,j,x[i][j]); }
            else{ printf("x%d -> x%d= ",i,j); scanf("%d",&x[i][j]); }
        }
    printf("\n\t1.Drumul minim Ford\n\n");
    H[1]=0; for(i=2;i<=v;i++) H[i]=100;
    for(i=1; i<=v; i++) printf(" x%d ",i); printf("\n");
    for(j=1; j<v; j++)
    {
        for(i=1;i<=v;i++)
        {
            if((x[j][i]!=0) && (x[j][i]!=100) && (H[i]-H[j]>x[j][i]))
            {
                H[i]=H[j]+x[j][i];
                printf(" %d ",H[i]);
            }
        }
    }
}
```

```

    }
    else printf(" %d ",H[i]);
}
printf("\n");
}
printf("\n! min = %d\t",H[v]); l[v]=v;
if(H[v]-H[v-1]==x[v-1][v]) l[v-1]=v-1;
else if(H[v]-H[v-2]==x[v-2][v]) l[v-1]=v-2;
else if(H[v]-H[v-3]==x[v-3][v]) l[v-1]=v-3;

if(H[v-1]-H[v-2]==x[v-2][v-1]) l[v-2]=v-2;
else if(H[v-1]-H[v-3]==x[v-3][v-1]) l[v-2]=v-3;
else if(H[v-1]-H[v-4]==x[v-4][v-1]) l[v-2]=v-4;

if(H[v-2]-H[v-3]==x[v-3][v-2]) l[v-3]=v-3;
else if(H[v-2]-H[v-4]==x[v-4][v-2]) l[v-3]=v-4;
else if(H[v-2]-H[v-5]==x[v-5][v-2]) l[v-3]=v-5;

if(H[v-3]-H[v-4]==x[v-4][v-3]) l[v-4]=v-4;
else if(H[v-3]-H[v-5]==x[v-5][v-3]) l[v-4]=v-5;
else if(H[v-3]-H[v-6]==x[v-6][v-3]) l[v-4]=v-6;

if(H[v-4]-H[v-5]==x[v-5][v-4]) l[v-5]=v-5;
else if(H[v-4]-H[v-6]==x[v-6][v-4]) l[v-5]=v-6;

```

```

if(H[v-5]-H[v-6]==x[v-6][v-5]) l[v-6]=v-6; l[v-7]=1;
for(i=1;i<=v;i++) if((l[i]!=0) && (l[i]!=l[i+1]) && (l[i])<=v) printf(" %d ",l[i]);
printf("\n\n\t2.Drumul maxim Ford\n\n");
H[1]=0; for(i=2;i<=v;i++) H[i]=-100;
for(j=1; j<v; j++)
{
    for(i=1;i<=v;i++)
    {
        if((x[j][i]!=0) && (x[j][i]!=100) && (H[i]-H[j]<x[j][i]))
        {
            H[i]=H[j]+x[j][i];
            printf(" %d ",H[i]);
        }
        else printf(" %d ",H[i]);
    }
    printf("\n");
}
printf("\n\n\t max = %d\t",H[v]); l[v]=v;
if(H[v]-H[v-1]==x[v-1][v]) l[v-1]=v-1;
else if(H[v]-H[v-2]==x[v-2][v]) l[v-1]=v-2;
else if(H[v]-H[v-3]==x[v-3][v]) l[v-1]=v-3;

if(H[v-1]-H[v-2]==x[v-2][v-1]) l[v-2]=v-2;
else if(H[v-1]-H[v-3]==x[v-3][v-1]) l[v-2]=v-3;
else if(H[v-1]-H[v-4]==x[v-4][v-1]) l[v-2]=v-4;

```

```
if(H[v-2]-H[v-3]==x[v-3][v-2]) l[v-3]=v-3;
else if(H[v-2]-H[v-4]==x[v-4][v-2]) l[v-3]=v-4;
else if(H[v-2]-H[v-5]==x[v-5][v-2]) l[v-3]=v-5;
```

```
if(H[v-3]-H[v-4]==x[v-4][v-3]) l[v-4]=v-4;
else if(H[v-3]-H[v-5]==x[v-5][v-3]) l[v-4]=v-5;
else if(H[v-3]-H[v-6]==x[v-6][v-3]) l[v-4]=v-6;
```

```
if(H[v-4]-H[v-5]==x[v-5][v-4]) l[v-5]=v-5;
else if(H[v-4]-H[v-6]==x[v-6][v-4]) l[v-5]=v-6;
```

```
if(H[v-5]-H[v-6]==x[v-6][v-5]) l[v-6]=v-6; l[v-7]=1;
for(i=1;i<=v;i++) if((l[i]!=0) && (l[i]!=l[i+1]) && (l[i])<=v) printf(" %d ",l[i]);
printf("\n\n\t3.Drumul minim BK\n\n");
for(i=1; i<=v; i++) printf("\tx%d",i); printf("\n");
for(i=1; i<=v; i++)
{
    printf("x%d | ",i);
    for(j=1; j<=v; j++)
    {
        x[7][4]=100; x[7][5]=100; x[7][6]=100;
        if(i==j){ x[i][j]=0; printf("%d\t",x[i][j]); }
        else printf("%d\t",x[i][j]);
    }
}
```

```

    printf("\n");
}
printf("V | ");
for(i=1; i<=v; i++)
{
    V[1][i]=x[i][v];
    if(i==v) V[1][i]=0;
    else printf("%d\t",V[1][i]);
}
for(k=2; k<=5; k++)
{
    for(i=1; i<=v; i++) V[k][i]=100;
    for(i=1; i<=v; i++)
        for(j=1; j<=v; j++)
        {
            min=V[k-1][i]+x[j][i];
            if(min<V[k][j]) V[k][j]=min;
        }
    printf("\nV%d | ",k-1);
    for(i=1; i<=v; i++) printf("%d\t",V[k][i]);
}
printf("\n\nl min = %d",V[5][1]); l[1]=1; printf("\t%d ",l[1]);
for(i=2; i<=v; i++)
{
    if(V[5][i-1]-V[5][i]==x[i-1][i]){ l[i]=i; printf(" %d ",l[i]);}

```

```

        else if(V[5][i-1]-V[5][i+1]==x[i-1][i+1]){ l[i]=i+1; printf(" %d ",l[i]); i++;}
    }
    printf("\n\n\t4.Drumul maxim BK\n\n");
    for(i=1; i<=v; i++) printf("  x%d",i); printf("\n");
    for(i=1; i<=v; i++)
    {
        printf("x%d | ",i);
        for(j=1; j<=v; j++)
        {
            if(x[i][j]==100) printf("%f  ", 1/0.0);
            else printf("%d  ",x[i][j]);
        }
        printf("\n");
    }
    printf("V  | ");
    for(i=1; i<=v; i++)
    {
        V[1][i]=x[i][v];
        if(i==v) V[1][i]=0;
        if(V[1][i]==100) printf("%f  ", 1/0.0);
        else printf("%d  ",V[1][i]);
    }
    for(k=2; k<=5; k++)
    {
        for(i=1; i<=v; i++) V[k][i]=0;
    }

```



```

for(i=1; i<=v; i++)
    for(j=1; j<=v; j++)
    {
        max=V[k-1][i]+x[j][i];
        if((max>V[k][j]) && (max<100)) V[k][j]=max;
    }
printf("\nV%d | ",k-1);
for(i=1; i<=v; i++) printf("%d   ",V[k][i]);
}
printf("\n\nl max = %d",V[5][1]); l[1]=1; printf("\t%d ",l[1]);
for(i=2; i<=v; i++)
{
    if(V[5][i-1]-V[5][i]==x[i-1][i]){ l[i]=i; printf(" %d ",l[i]);}
    else if(V[5][i-1]-V[5][i+1]==x[i-1][i+1]){ l[i]=i+1; printf(" %d ",l[i]); i++;}
}
}

```

## Execuția Programului :

Nr de varfuri 7

```
x1 -> x1= 0
x1 -> x2= 5
x1 -> x3= 3
x1 -> x4= 5
x1 -> x5= 6
x1 -> x6= 8
x1 -> x7= 100
x2 -> x1= 100
x2 -> x2= 0
x2 -> x3= 100
x2 -> x4= 1
x2 -> x5= 4
x2 -> x6= 100
x2 -> x7= 100
x3 -> x1= 100
x3 -> x2= 100
x3 -> x3= 0
x3 -> x4= 100
x3 -> x5= 2
x3 -> x6= 100
x3 -> x7= 100
x4 -> x1= 100
x4 -> x2= 100
x4 -> x3= 100
x4 -> x4= 0
```

```
x4 -> x5= 3
x4 -> x6= 5
x4 -> x7= 100
x5 -> x1= 100
x5 -> x2= 100
x5 -> x3= 100
x5 -> x4= 100
x5 -> x5= 0
x5 -> x6= 4
x5 -> x7= 6
x6 -> x1= 100
x6 -> x2= 100
x6 -> x3= 100
x6 -> x4= 100
x6 -> x5= 100
x6 -> x6= 0
x6 -> x7= 5
x7 -> x1= 100
x7 -> x2= 100
x7 -> x3= 100
x7 -> x4= 100
x7 -> x5= 100
x7 -> x6= 100
x7 -> x7= 0
```

### 1.Drumul minim Ford

x1	x2	x3	x4	x5	x6	x7
0	5	3	5	6	8	100
0	5	3	5	6	8	100
0	5	3	5	5	8	100
0	5	3	5	5	8	100
0	5	3	5	5	8	11
0	5	3	5	5	8	11

1 min = 11                      1   3   5   7

### 2.Drumul maxim Ford

0	5	3	5	6	8	-100
0	5	3	6	9	8	-100
0	5	3	6	9	8	-100
0	5	3	6	9	11	-100
0	5	3	6	9	13	15
0	5	3	6	9	13	18

1 max = 18                      1   2   4   5   6   7

...Program finished with exit code 0

Press ENTER to exit console.

### 1.Drumul minim BK

	x1	x2	x3	x4	x5	x6	x7
x1	0	5	3	5	6	8	inf
x2	inf	0	inf	1	4	inf	inf
x3	inf	inf	0	inf	2	inf	inf
x4	inf	inf	inf	0	3	5	inf
x5	inf	inf	inf	inf	0	4	6
x6	inf	inf	inf	inf	inf	0	5
x7	inf	inf	inf	inf	inf	inf	0
V	inf	inf	inf	inf	6	5	0
V1	12	10	8	9	6	5	0
V2	11	10	8	9	6	5	0
V3	11	10	8	9	6	5	0
V4	11	10	8	9	6	5	0

l min = 11            1   3   5   7

### 2.Drumul maxim BK

	x1	x2	x3	x4	x5	x6	x7
x1	0	5	3	5	6	8	inf
x2	inf	0	inf	1	4	inf	inf
x3	inf	inf	0	inf	2	inf	inf
x4	inf	inf	inf	0	3	5	inf
x5	inf	inf	inf	inf	0	4	6
x6	inf	inf	inf	inf	inf	0	5
x7	inf	inf	inf	inf	inf	inf	0
V	inf	inf	inf	inf	6	5	0
V1	13	10	8	10	9	5	0
V2	15	13	11	12	9	5	0
V3	18	13	11	12	9	5	0
V4	18	13	11	12	9	5	0

l max = 18            1   2   4   5   6   7

...Program finished with exit code 0

Press ENTER to exit console.

## Concluzia :

Efectuind aceasta lucrare , neam familiarizat cu algoritmul aflarii drumului minim.Acest algoritm ne permite de a afla drumul minim intre orce doua vurfuri prin metoda lui **Ford** si prin metoda lui **Bellman-Kalaba**.Acest algoritm se aplică pe larg în practică de exemplu la proiectarea șoselelor sau a diferitor tipuri de comunicații,deci studiind teoretic acum acest algoritm pe viitotr e posibil să-l aplicăm pentru un caz real.