Experiment No: 05

Name: Varun Viswanath **SAP ID:** 60004210105

Batch: B1 Comps

Aim: Write a program to implement Dijkstra's Shortest Path Routing Algorithm.
Write a program to implement Bellman Ford Shortest Path Routing Algorithm.

Dijkstra

Code:

```
#include<stdio.h>
#define MAX 9
#define INFINITY 9999

int g[MAX][MAX] = { { 0, 11, 0, 0, 0, 0, 0, 8, 0 },
                    { 11, 0, 8, 0, 0, 0, 0, 11, 0 },
                    { 0, 8, 0, 7, 0, 4, 0, 0, 2 },
                    { 0, 0, 7, 0, 9, 14, 0, 0, 0 },
                    { 0, 0, 0, 9, 0, 10, 0, 0, 0 },
                    { 0, 0, 4, 14, 10, 0, 2, 0, 0 },
                    { 0, 0, 0, 0, 0, 2, 0, 1, 6 },
                    { 8, 11, 0, 0, 0, 0, 1, 0, 7 },
                    { 0, 0, 2, 0, 0, 0, 6, 7, 0 } };

int min1(int d[],int v[],int n){
   int f=0,b[MAX];
   int m ,l=0;
   for(int i=0;i<MAX;i++){
      if(v[i]==0)
      {m=d[i];
         break;
      }
   }
   for(int i=1;i<n;i++)
   {
      if(d[i]<=m && v[i]==0)
      {
         m=d[i];
         {l=i;}
      }
   }
   for(int i=0;i<n;i++)
   {
```

```
        if(v[i]==1)
        {
           f=f+1;
        }
    }
    if(f==n)
    {
        return -1;
    }
    return l;
}

void mindist(int d[], int par[], int u, int v)
{
    if (g[u][v] != 0 && d[v] > d[u] + g[u][v])
    {
        d[v] = d[u] + g[u][v];
        par[v] = u;
    }
    return u;
}


int main()
{
int d[MAX],par[MAX],v[MAX]={0};
int x=0,s=0;//source
d[0]=0;
par[0]=-1;
for(int i=1;i<MAX;i++)
{d[i]=INFINITY;}

do
{
    if(v[x]==0)
    {
    for(int j=0;j<MAX;j++)
    {
        if(g[x][j]!=0)
        {mindist(d,par,x,j);}
    }
    }
    v[x]=1;
}while((x = min1(d,v,MAX))!=-1);
```

```
printf("\nvertex\tdistance from source\tparent");
for(int i=0;i<MAX;i++)
{
  printf("\n%d\t\t%d\t\t%d",i,d[i],par[i]);
}

for(int i=0;i<MAX;i++)
{
   int j;
      if(i!=s)
      {
         j=i;
         printf("\nFrom %d : %d<-",i,i);
         while(par[j]!=s)
         {
            printf("%d<-",par[j]);
            j=par[j];
         }
         printf("%d",s);
      }

}

return 0;

}
```

Output:

```
vertex  distance from source    parent
0               0               -1
1               11              0
2               15              5
3               22              2
4               21              5
5               11              6
6               9               7
7               8               0
8               15              7
From 1  : 1<-0
From 2  : 2<-5<-6<-7<-0
From 3  : 3<-2<-5<-6<-7<-0
From 4  : 4<-5<-6<-7<-0
From 5  : 5<-6<-7<-0
From 6  : 6<-7<-0
From 7  : 7<-0
From 8  : 8<-7<-0
Process returned 0 (0x0)   execution time : 0.018 s
```

Bellman Ford

Code:

```
#include <stdio.h>
#define MAX 5
#define INFINITY 9999

int g[MAX][MAX] ={  {0, 6, 0, 7, 0},
                    {0, 0, 5, 8, -4},
                    {0, -2, 0, 0, 0},
                    {0, 0, -3, 0, 9},
                    {2, 0, 7, 0, 0}};

int checkarr(int a[],int b[],int n){
   int flag = 0;
   for (int i = 0; i < n; i++){
      if (a[i] == b[i]){
         flag++;
      }
   }
   if (flag == n){
      return -1;
   }
   return 0;
}

int mindist(int d[],int par[],int u,int v){

   if (g[u][v] != 0 && d[v] > d[u] + g[u][v]){
      d[v] = d[u] + g[u][v];
      par[v] = u;
   }
   return u;
}

int main(){
   int d1[MAX], d2[MAX],par1[MAX],par2[MAX],c=0;
   d1[0] = 0;
   d2[0] = 0;
   par1[0] = -1;
   par2[0] = -1;
   for (int i = 1; i < MAX; i++){
      d1[i] = INFINITY;
      d2[i] = INFINITY;
   }
   do{
```

```
    for(int i=0;i<MAX;i++){
        for(int j=0;j<MAX;j++){
            mindist(d1,par1,i,j);
        }
    }
    if(checkarr(d1,d2,MAX)==-1){
            break;
    }
    for(int k=0;k<MAX;k++){
            d2[k]=d1[k];
            par2[k]=par1[k];
    }
    c++;
  }while(c <MAX);

  printf("\nvertex\tdistance from source\tparent");
  for(int i=0;i<MAX;i++)
  {
    printf("\n%d\t\t%d\t\t%d",i,d1[i],par1[i]);
  }
  return 0;
}
```

Output:

```
vertex  distance from source    parent
0               0               -1
1               2               2
2               4               3
3               7               0
4               -2              1
Process returned 0 (0x0)   execution time : 0.012 s
```

Conclusion:
Dijkstra and Bellman Ford is implemented.