Experiment No. 6

Aim: Implement Single source shortest path- Dijkstra algorithm

Theoretical Background:

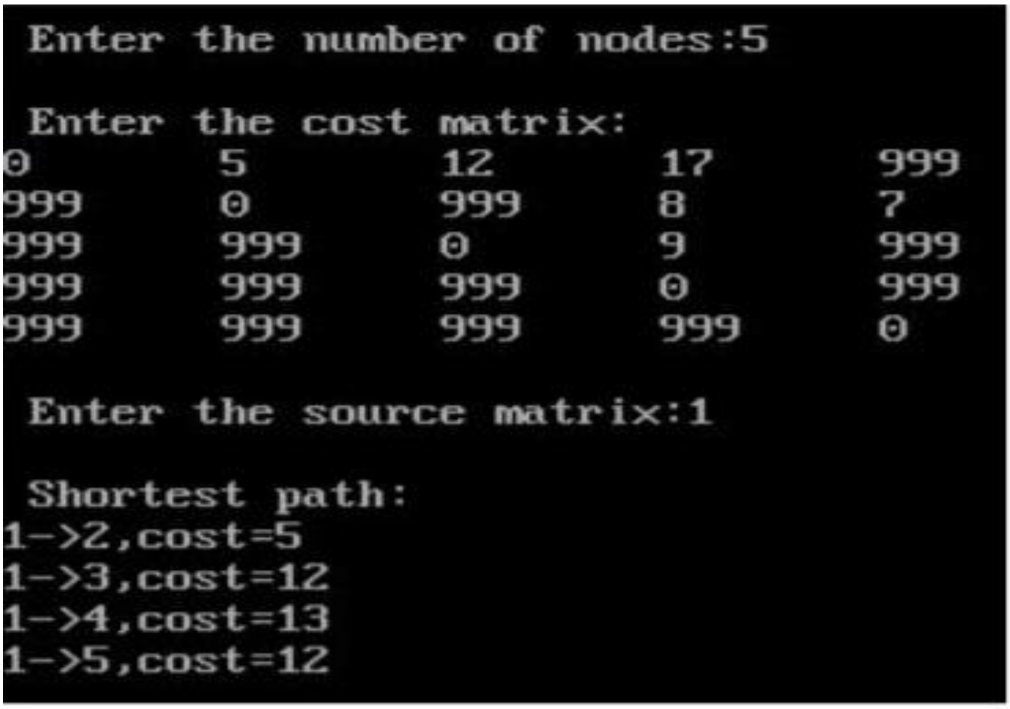
Dijkstra's algorithm solves the single-source shortest-paths problem on a directed weighted graph G = (V, E), where all the edges are non-negative (i.e., w(u, v) ≥ 0 for each edge (u, v) Є E).

In the following algorithm, we will use one function Extract-Min(), which extracts the node with the smallest key

Program:

```
#include<stdio.h>

#include<conio.h>

#define infinity 999

void dij(int n,int v,int cost[10][10],int dist[100])

{

int i,u,count,w,flag[10],min;

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

flag[i]=0,dist[i]=cost[v][i];

count=2;

while(count<=n)

{

min=99;

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

if(dist[w]<min && !flag[w])

min=dist[w],u=w;

flag[u]=1;

count++;

for(w=1;w<=n;w++)
if((dist[u]+cost[u][w]<dist[w]) && !flag[w])
dist[w]=dist[u]+cost[u][w];
}
}
void main()
{
int n,v,i,j,cost[10][10],dist[10];
clrscr();
printf("\n Enter the number of nodes:");
scanf("%d",&n);
printf("\n Enter the cost matrix:\n");
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{
scanf("%d",&cost[i][j]);
if(cost[i][j]==0)
cost[i][j]=infinity;
}
printf("\n Enter the source matrix:");
scanf("%d",&v);
dij(n,v,cost,dist);
printf("\n Shortest path:\n");
for(i=1;i<=n;i++)
if(i!=v)
printf("%d->%d,cost=%d\n",v,i,dist[i]);
getch();
}
```

Output:



```
 Enter the number of nodes:5

 Enter the cost matrix:
0       5       12      17      999
999     0       999     8       7
999     999     0       9       999
999     999     999     0       999
999     999     999     999     0

 Enter the source matrix:1

 Shortest path:
1->2,cost=5
1->3,cost=12
1->4,cost=13
1->5,cost=12
```

Conclusion:

Thus, in this experiment we have studied about Single source shortest path- Dijkstra algorithm