**AIM:** From a given vertex in a weighted connected graph, find shortest paths to other vertices using **Dijkstra’s algorithm**.

Note: In the record book students should

- Handwrite the Algorithm

- Handwrite the Program

- Pasting of the printout of the Output or handwrite the Output

**ALGORITHM**: dijkstras(c[1….n,1….n],src)

//To compute shortest distance from given source node to all nodes of a weighted undirected graph

//Input: An nXn cost matrix c[1…n,1….n] with source node *src*

//Output: The length *dist[j]* of a shortest path from *src* to *j*

**for** j🡨1 to n **do**

dist[j]🡨c[src,[j]

**end for**

**for** j🡨1 to n **do**

vis[j]🡨0

**end for**

dist[src]🡨0

vis[src]🡨1

count🡨1

**while** count!=n **do**

min🡨9999

**for** j🡨1 to n **do**

**if** dist[j]<min **and** vis[j]!=1

min🡨dist[j]

u🡨j

**end if**

**end for**

vis[u]🡨1

count🡨count+1

**for** j🡨1 to n **do**

**if** min+c[u,j]<dist[j] **and** vis[j]!=1

dist[j]🡨min+c[u,j]

**end if**

**end for**

**end while**

write ‘shortest distance is’

**for** j🡨1 to n **do**

write src,j,dist[j]

**end for**

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

void dijkstras();

int c[10][10],n,src;

void main()

{

int i,j;

clrscr();

printf("\nenter the no of vertices:\t");

scanf("%d",&n);

printf("\nenter the cost matrix:\n");

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

{

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

{

scanf("%d",&c[i][j]);

}

}

printf("\nenter the source node:\t");

scanf("%d",&src);

dijkstras();

getch();

}

void dijkstras()

{

int vis[10],dist[10],u,j,count,min;

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

{

dist[j]=c[src][j];

}

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

{

vis[j]=0;

}

dist[src]=0;

vis[src]=1;

count=1;

while(count!=n)

{

min=9999;

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

{

if(dist[j]<min&&vis[j]!=1)

{

min=dist[j];

u=j;

}

}

vis[u]=1;

count++;

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

{

if(min+c[u][j]<dist[j]&&vis[j]!=1)

{

dist[j]=min+c[u][j];

}

}

}

printf("\nthe shortest distance is:\n");

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

{

printf("\n%d----->%d=%d",src,j,dist[j]);

}

}

==========Output=============

Enter the no. of vertices: 5

Enter the cost matrix:

9999 3 9999 7 9999

3 9999 4 2 9999

9999 4 9999 5 6

7 2 5 9999 4

9999 9999 6 4 9999

Enter the source node: 1

The shortest distance is:

1-----------> 1 = 0

1-----------> 2 = 3

1-----------> 3 = 7

1-----------> 4 = 5

1-----------> 5 = 9