ANAGHA ACHARYA

1BM19CS224

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

#include<stdio.h>

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

{

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

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

{

visited[i]=0;

dist[i]=cost[v][i];

}

visited[v]=1;

dist[v]=1;

count=2;

while(count<=n)

{

min=999;

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

if((dist[w]<min) && (visited[w]!=1))

{

min=dist[w];

u=w;

}

visited[u]=1;

count++;

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

if((dist[u]+cost[u][w]<dist[w]) && (visited[w]!=1))

dist[w]=dist[u]+cost[u][w];

}

}

void main()

{

int n,v,cost[10][10],dist[10], i, j;

printf("Enter number of vertices: ");

scanf("%d",&n);

printf("\nEnter the cost matrix(for infinity, enter 999):\n");

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

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

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

printf("\nEnter source vertex:");

scanf("%d",&v);

dijkstra(n,v,cost,dist);

printf("\nShortest path from");

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

if(i!=v)

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

}

