







## Dijkstras\_Algorithm.java

```

import java.util.Scanner;
public class Dijkstras_Algorithm {
    public static int path[] = new int[10];
    public static void main(String[] args) {
        int tot_nodes,i,j;
        int cost[][]=new int[10][10];
        int dist[] = new int[10];
        int s[] = new int[10];
        System.out.println("\n\t\t——Creation of Graph——");
        System.out.println("\nEnter Total Number of Nodes:");
        Scanner in = new Scanner(System.in);
        tot_nodes = in.nextInt();
        create(tot_nodes,cost);
        for(i=0;i<tot_nodes;i++) {
            System.out.println("\t\tPress any Key To Continue... ");
            System.out.println("\t\tWhen Source=" + i);
            for(j=0;j<tot_nodes;j++) {
                Dijkstra(tot_nodes,cost,i,dist);
                if(dist[j]==999)
                    System.out.println("\n There is no Path to " + j);
                else
                {
                    display(i,j,dist);
                }
            }
        }
        public static void create(int tot_nodes,int cost[][])
        {
            int i,j,val,tot_edges,count=0;
            for(i=0;i<tot_nodes;i++) {
                for(j=0;j<tot_nodes;j++) {
                    if(i==j)
                        cost[i][j]=0;
                    else
                        cost[i][j]=999;
                }
            }
            System.out.println("\n Total Number of Edges:");
            Scanner in = new Scanner(System.in);
            tot_edges = in.nextInt();
            while(count<tot_edges) {
                System.out.println("\nEnter Vi and Vj:");
                i = in.nextInt();
                j = in.nextInt();
                System.out.println("\nEnter the cost along this edge:");
                val = in.nextInt();
                cost[j][i] = val;
                cost[i][j] = val;
                count++;
            }
        }
        static void Dijkstra(int tot_nodes,int cost[][],int source,int dist[])
        {
            int i,j,k;
            int min;
            for(i=0;i<tot_nodes;i++)
                dist[i] = 999;
            dist[source] = 0;
            for(k=0;k<tot_nodes;k++)
            {
                min = 999;
                for(i=0;i<tot_nodes;i++)
                    if(dist[i]<min)
                        min = dist[i];
                if(min==999)
                    break;
                for(j=0;j<tot_nodes;j++)
                    if(cost[i][j]<999)
                        dist[j] = Math.min(dist[j],dist[i]+cost[i][j]);
            }
        }
        static void display(int i,int j,int dist[])
        {
            System.out.println("The shortest distance from " + i + " to " + j + " is " + dist[j]);
        }
    }
}

```

