**ASSIGNMENT NO.3**

**Dijktra’s Algorithm:-**

public class Dijkstra {

    public static void dijkstra(int[][] graph, int source) {

      int count = graph.length;

      boolean[] visitedVertex = new boolean[count];

      int[] distance = new int[count];

      for (int i = 0; i < count; i++) {

        visitedVertex[i] = false;

        distance[i] = Integer.MAX\_VALUE;

      }

      distance[source] = 0;

      for (int i = 0; i < count; i++) {

        int u = findMinDistance(distance, visitedVertex);

        visitedVertex[u] = true;

for (int v = 0; v < count; v++) {

          if (!visitedVertex[v] && graph[u][v] != 0 && (distance[u] + graph[u][v] < distance[v])) {

            distance[v] = distance[u] + graph[u][v];

          }

        }

      }

      for (int i = 0; i < distance.length; i++) {

        System.out.println(String.format("Distance from %s to %s is %s", source, i, distance[i]));

      }

    }

    private static int findMinDistance(int[] distance, boolean[] visitedVertex) {

      int minDistance = Integer.MAX\_VALUE;

      int minDistanceVertex = -1;

      for (int i = 0; i < distance.length; i++) {

        if (!visitedVertex[i] && distance[i] < minDistance) {

          minDistance = distance[i];

          minDistanceVertex = i;

        }

      }

      return minDistanceVertex;

    }

    public static void main(String[] args) {

      int graph[][] = new int[][] { { 0, 0, 1, 2, 0, 0, 0 }, { 0, 0, 2, 0, 0, 3, 0 }, { 1, 2, 0, 1, 3, 0, 0 },

          { 2, 0, 1, 0, 0, 0, 1 }, { 0, 0, 3, 0, 0, 2, 0 }, { 0, 3, 0, 0, 2, 0, 1 }, { 0, 0, 0, 1, 0, 1, 0 } };

      Dijkstra T = new Dijkstra();

      T.dijkstra(graph, 0);

    }

  }

**output:-**

Distance from 0 to 0 is 0

Distance from 0 to 1 is 3

Distance from 0 to 2 is 1

Distance from 0 to 3 is 2

Distance from 0 to 4 is 4

Distance from 0 to 5 is 4

Distance from 0 to 6 is 3