

COMPUTER NETWORKS

LAB 6

NAME - KAPAROTU VENKATA SURYA THARANI
USN - 22BTRAD018
BRANCH - AIDE

Ques - Implement Dijkstra's algorithm to compute the shortest path through a network.

Dijkstra's algorithm is used to find the shortest path from a single source to all destinations in the subnet graph.

Here the code for the implementation is done using Java language.

CODE:

```
public class DijkstraAlgorithm {  
  
    // defining the method to implement Dijkstra's Algorithm  
  
    public void dijkstraAlgorithm(int[ ][ ] graph, int source)  
    {  
        // number of nodes  
        int nodes = graph.length;  
        boolean[] visited_vertex = new boolean[nodes];  
        int[] dist = new int[nodes];  
        for (int i = 0; i < nodes; i++)  
        {  
            visited_vertex[i] = false;  
            dist[i] = Integer.MAX_VALUE;  
        }  
  
        // Distance of self-loop is zero  
        dist[source] = 0;  
        for (int i = 0; i < nodes; i++)  
        {  
  
            // Updating the distance between the neighboring vertex and the source  
            vertex  
  
            int u = find_min_distance(dist, visited_vertex);
```

```

    visited_vertex[u] = true;

    // Updating the distances of all the neighboring vertices

    for (int v = 0; v < nodes; v++)
    {
        if (!visited_vertex[v] && graph[u][v] != 0 && (dist[u] + graph[u][v] < dist[v]))
        {
            dist[v] = dist[u] + graph[u][v];
        }
    }
    for (int i = 0; i < dist.length; i++)
    {
        System.out.println(String.format("Distance from Vertex %s to Vertex %s is %s", source, i, dist[i]));
    }
}

// defining the method to find the minimum distance

private static int find_min_distance(int[] dist, boolean[] visited_vertex)
{
    int minimum_distance = Integer.MAX_VALUE;
    int minimum_distance_vertex = -1;
    for (int i = 0; i < dist.length; i++)
    {
        if (!visited_vertex[i] && dist[i] < minimum_distance)
        {
            minimum_distance = dist[i];
            minimum_distance_vertex = i;
        }
    }
    return minimum_distance_vertex;
}

// main function

public static void main(String[] args)
{
    // declaring the nodes of the graphs
    int graph[][] = new int[5][5]
    {
        { 0, 1, 1, 2, 0, 0, 0 },
        { 0, 0, 2, 0, 0, 3, 0 },
        { 1, 2, 0, 1, 3, 0, 0 },
        { 2, 0, 1, 0, 2, 0, 1 },
        { 0, 0, 3, 0, 0, 2, 0 },
    }
}

```

```

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

    // instantiating the DijkstraAlgorithm() class

    DijkstraAlgorithm Test = new DijkstraAlgorithm();

    // calling the Dijkstra algorithm() method to find the shortest distance from
    the source node to the destination node

    Test.dijkstraAlgorithm(graph, 0);
}
}

```

OUTPUT:

```

Distance from Vertex 0 to Vertex 0 is 0
Distance from Vertex 0 to Vertex 1 is 1
Distance from Vertex 0 to Vertex 2 is 1
Distance from Vertex 0 to Vertex 3 is 2
Distance from Vertex 0 to Vertex 4 is 4
Distance from Vertex 0 to Vertex 5 is 4
Distance from Vertex 0 to Vertex 6 is 3
PS C:\Users\kvsth\Desktop\Term 5\Comp Networks\Lab codes>

```