# UIT2511---Software Development Project – II

**Optimised Source to Destination path**

**Name:** Vasundhara

**Roll No:** 3122 21 5002 119

**DIJKSTRA’S ALGORITHM:**

Given a graph and a source vertex in the graph, find the shortest paths from source to all vertices in the given graph.  
Dijkstra’s algorithm generates a *SPT (shortest path tree)* with a given source as root. It maintains two sets, one set contains vertices included in the shortest-path tree, another set includes vertices not yet included in the shortest-path tree. At every step of the algorithm, we find a vertex that is in the other set (set of not yet included) and has a minimum distance from the source.

class Graph():

def \_\_init\_\_(self, vertices):

        self.V = vertices

        self.graph = [[0 for column in range(vertices)]

                    for row in range(vertices)]

    def printSolution(self, dist):

        print("Vertex \t Distance from Source")

        for node in range(self.V):

            print(node, "\t\t", dist[node])

    def minDistance(self, dist, sptSet):

        min = 1e7

        for v in range(self.V):

            if dist[v] < min and sptSet[v] == False:

                min = dist[v]

                min\_index = v

        return min\_index

    def dijkstra(self, src):

        dist = [1e7] \* self.V

        dist[src] = 0

        sptSet = [False] \* self.V

        for cout in range(self.V):

            u = self.minDistance(dist, sptSet)

            sptSet[u] = True

            for v in range(self.V):

                if (self.graph[u][v] > 0 and

                sptSet[v] == False and

                dist[v] > dist[u] + self.graph[u][v]):

                    dist[v] = dist[u] + self.graph[u][v]

        self.printSolution(dist)

# Driver program

g = Graph(9)

g.graph = [[0, 4, 0, 0, 0, 0, 0, 8, 0],

        [4, 0, 8, 0, 0, 0, 0, 11, 0],

        [0, 8, 0, 7, 0, 4, 0, 0, 2],

        [0, 0, 7, 0, 9, 14, 0, 0, 0],

        [0, 0, 0, 9, 0, 10, 0, 0, 0],

        [0, 0, 4, 14, 10, 0, 2, 0, 0],

        [0, 0, 0, 0, 0, 2, 0, 1, 6],

        [8, 11, 0, 0, 0, 0, 1, 0, 7],

        [0, 0, 2, 0, 0, 0, 6, 7, 0]

        ]

g.dijkstra(0)

**OUTPUT:**

A black screen with white text

Description automatically generated