Computer Network

Pratibha.R
1BM18CS074

```
import sys.
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\tDistance from source"
        for node in range (self.V):
            print node , "\t", dist[node]

    def minDistance(self, dist, sptSet):
        min = sys.maxint
        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 = [sys.maxint] * self.V.
        dist[src] = 0.
        sptSet = [False] * self.V.

        for count 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)
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);
```