LAB 10: Dijkstra Algorithm

SHASHIDHAR PATE
1BM18CS099

```
import sys
class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.graph = [[0 for coloumn in range(vertices)]
                      for row in range(self.V)]:
                      print(node, "\t", dist[node])
  def minDistance(self, dist, sptSet):
        min = sys.maxsize
        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.maxsize] * 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)
```