ARAVINDH R
1BM18CS145

CN Lab 9

```
class graph():
    def __init__(self, vertices):
        self.V = vertices
        self.graph = [[0 for column in range(V)]
                      for row in range(V)]

    def print_solution(self, dist):
        print("Vertex \t Distance from Source")
        for node in range(self.V):
            print(node, "\t", dist[node])

    def min_distance(self, dist, sptSet):
        min = 9999
        for v in range(self.V):
            if dist[v] < min and sptSet[v] == False:
                min = dist[v]
                min_in = v
        return min_index

    def add_edge(self, src, dest, weight):
        self.graph[src][dist] = self.graph[dest][src]
                              = weight
```

ARAVINDH R
1BM19CS145

```
def dijstra(self, src):
    dist=[9999] * self.V
    dist[src] = 0
    sptSet = [False] * self.V

    for cout in range(self.V):
        u = self.min_distance(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.print_solution(dist)
```