CN LAB 8

```
class Network ():

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

    def printTable(self, dist, src, path):
        print("Shortest path:")
        for node in range(self.V):
            print(dist[node], 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
        path = {}
        for - in range(self.V):
            path[-] = []
        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]
            if v == src:
                path[v].append(chr(ord('A') + v))
            else
                path[v].append(chr(ord('A') + u))
                path[v].append(chr(ord('A') + v))
        self.printTable(dist, src, path)
```