CN LAB

Chandan C Bagan
1BM18CS026

```
# Dijkstra's Algorithm

import sys
class Network():
 def __init__(seft, 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 Table of {}".format(chr(ord('A')
                                                     +src)))
       for node in range(self.v):
           print("{0}\t{1}\t{2}".format(chr(ord('A'+node),
                                 dist[node], path[node])).

  def minDistance(self, dist, spt set):
       min = sys.maxsize
       for v in range(self.v):
           if dist[v] < min and spt set[v] == False:
                min = dist[v]
                min-index = v
       return min_index

 def dijktra(self, src):
      dist = [sys.maxsize] * self.V
      dist[src] = 0
      sptset = [False] * self.v
      path = {}
      for _ in range(self.v):
          path[_] = []
```

```
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]

          if u == src',
             path[v]. append (chr (ord('A') + v) )

          else:
              path (v). append (chr (ord ['A']+u))
              path [v] . append (chr (ord ('A') +v))

     self.printable (dist, src, path)

g = Network (5)
 g.graph = [[0,1,1,0,0],[1,0,0,1,1],[1,0,0,1,0],[0,1,1,0
                [0,1,0,1,0]]
 for i in range(g.v):
       g. dijkstra(i)
```