Akshay S Bharadwaj

1BM18CS011

CN Lab Writeup (Dijkstra's algorithm)

```
import sys
class Matrix():
    def _init_ (self, n):
        self.n = n;
        self.graph = [[ 0 for column in rage(n)]for row
                        in range(n)]
    def printdistance (self, dist, src, path):
        print(" Shortest path table of {} ".format (chr(ord('A')
                                                    +src)))
        for node in range (self.n):
            print ("{0}\t{1}\t{2}". format (chr (ord ('A')+node
                                          , dist[node], path[node]))
    def minimum( self, dist, visited):
        min = sys.maxsize
        for v in range (self.n):
            if dist [v]< min and visited [v] == False:
                min = dist [v];
                idx = v
        return idx
    def dijkstra (self, src):
        dist = [sys.maxsize ]* self.n
        dist [src] = 0
        visited = [False] * self.n
        path = {}
        for _ in range (self.n):
            path [_] = []
```

[illegible]

Akshay S Bharadwaj

1BM18CS011

```
for i in range(self.n):
    u = self.minimum(dist, visited)
    visited[u] = True
    for v in range(self.n):
        if ~~&~~ self.graph[u][v]>0 and visited[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.printdistance(dist, src, path)
```

AB