

07/12/2020
Monday

Lab 8
Writeup

Arjun A.S
18M18CS019

Dijkstra's Algorithm in Networking

```
class Network:
    # Set of nodes

    def __init__(self, nodes, graph):
        self.V = nodes
        self.graph = graph

    # function to print the network
    def printTable(self, dist, src, path):
        print("Shortest path Table of {}".format(chr(ord('A') + src)))
        for node in range(self.V):
            print("{}\t{}\t{}".format(chr(ord('A') + node), dist[node],
                                      path[node]))

    def minDistance(self, dist, sptSet):
        min = 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 = [MAXSIZE] * self.V
        dist[src] = 0
        sptSet = [False] * self.V
        path = {}
```

Arjun A.S

```
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 u == src:
```

```
                path[v].append(chr(Ord('A') + v))
```

```
            else:
```

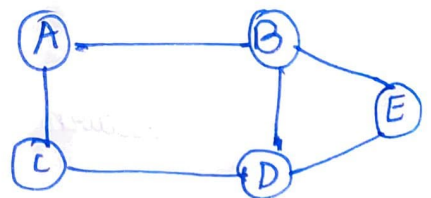
```
                path[v].append(chr(Ord('A') + u))
```

```
                path[v].append(chr(Ord('A') + v))
```

```
    self.printTable(dist, src, path)
```

MAIN FUNCTION

```
graph = [[0, 1, 1, 0, 0],  
         [1, 0, 0, 1, 1],  
         [1, 0, 0, 1, 0],  
         [0, 1, 1, 0, 1],  
         [0, 1, 0, 1, 0],  
         ]
```



```
g = Network(5, graph)
```

```
for _ in range(g.V):
```

```
    g.dijkstra(-)
```

Assign 28