

23/11/2020

Abyan A.S

Cycle 2

IBMI8CS019

CN Lab-

class Network:

```
def __init__(self, n):
```

```
    self.matrix = [[]]
```

```
    self.n = n
```

```
def addlink(self, u, v, w):
```

```
    self.matrix.append((u, v, w))
```

```
def printtable(self, dist, src):
```

```
    print("Vector Table of { } ".format(chr(ord('A')+src)))
```

```
    for i in range(self.n):
```

```
        print("{ } \t { } ".format(chr(ord('A')+i), dist[i]));
```

```
# returns table for given src
```

```
def Dijkstra(self, src):
```

```
    dist = [99] * self.n
```

```
    dist[src] = 0
```

```
    for _ in range(self.n-1):
```

```
        for u, v, w in self.matrix:
```

```
            if dist[u] != 99 and dist[u] + w < dist[v]:
```

```
                dist[v] = dist[u] + w
```

```
    self.printtable(dist, src)
```

```
def main():
```

```
    matrix = []
```

```
    print('Enter No. of Nodes: ') - V
```

```
    n = int(input())
```

```
    print('Enter the Adjacency matrix: "')
```

```
    g = Network(n)
```

```
    for i in range(n):
```

```
        for j in range(n):
```

```
            if matrix[i][j] == 1:
```

```
                g.addLink(i, j, 1)
```

```
    for _ in range(n):
```

```
        g.algo(-)
```

Given input:

