7/12/20

CN LAB

AKSHAY MITTUR
1BM18CS010

Dijikstra's Algorithm

```
class Graph():
    def __init__(self, vertices):
        self.V = vertices
        self.graph = [[0 for column in range(vertices)]
                      for row in range(vertices)]
    def print_solution(self, dist):
        for node in range(self.V):
            print(node, "\t", dist[node])
    def min_distance(self, dist, sptSet):
        min = 9999
        for v in range(self.V):
            if dist[v] < min and sptSet[v] == False:
                min = dist[v]
                min_index = v
        return min_index
    def add_edge(self, src, dest, weight):
        self.graph[src][dest] = self.graph[dest][src] = weight
    def dijikstra(self, src):
        dist = [9999] * self.V
        dist[src] = 0
        sptSet = [False] * self.V
        for count in range(self.V):
            u = self.min_distance(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]
```

Dijikstra Algo, CN Lab

AKSHAY MITTU R
1BM18CS010

7/12/20

self.print(dist)

```
g = Graph(int(input()))
e = int(input())
for i in range(e):
    src, dest, cost = [int(_)for _ in input().split(' ')]
    g.add_edge(src,dest,cost)
src=int(input())
g.dijikstra(src)
```