

Ahmed Ali

IBM18CS135

Dijkstra's Algo.

①

```
class Graph():
```

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

```
        self.v = v
```

```
        self.g = [[0 for c in range(v)] for r in range(v)]
```

```
    def min_dist(self, dist, sptset):
```

```
        min = 99
```

```
        for v in range(self.v):
```

```
            if dist[v] < min and not sptset[v]:
```

```
                min = dist[v]
```

```
                i = v
```

```
        return i
```

```
    def add_edge(self, src, dest, weight):
```

```
        self.g[src][dest] = self.g[dest][src] = weight
```

②

```
def dijkstra (self, src):  
    dist = [99] * self.v  
    dist[src] = 0  
    sptSet = [False] * self.v  
    for v in range (self.v):  
        u = self.min_dist (dist, sptSet)  
        sptSet[u] = True  
        for v in range (self.v):  
            if self.g[v][v] > 0 and sptSet[v] ==  
                False and  
                dist[v] > dist[u] + self.graph[u][v]:  
                dist[v] = dist[u] + self.g[u][v]  
    for node in range (self.v):  
        print (node, '+', dist[node])
```

Munil