

CN lab 9

```
class graph():
```

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

```
        self.v = vertices
```

```
        self.graph = [[0 for column in range(v)]  
                        for row in range(v)]
```

```
    def print-solution(self, dist):
```

```
        print("Vertex It Distance from Source")
```

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

```
            print(node, "It", 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-in = v
```

```
        return min-in
```

```
    def add-edge(self, src, dest, weight):
```

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

```
def dijkstra(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]
```

```
    self.print_solution(dist)
```