

Program-03

Dijkstra's Algorithm.

```
import sys
```

```
class Graph:
```

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

```
        self.V = vertices
```

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

```
    def printSolution(self, dist):
```

```
        print("Vertex \t Distance from Source")
```

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

```
            print(node, "\t", dist[node])
```

```
    def minDistance(self, dist, sptSet):
```

```
        min = sys.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 = [sys.maxsize] * self.v  
    dist[src] = 0  
    sptset = [False] * self.v
```

```
    for cout 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]
```

```
    self.printSolution(dist)
```