Pramod O.ys
1BM19CS405
Pramod

Dijkstra Algorithm

```
import sys
class Graph():
    def __init__ (self, Vertices):
        Self . V = Vertexes
        Self. graph = [ [0 for colomn in Range(vertices)]
                        for row in range (Vertices)]

    def print Solution (self, dist):
        print ("Vertex distance from Source:")
        print ("Vertex cost")
        for node in range (Self. V):
            print ("  ", node, "  ", dist [node])

~~tuin~~
    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.
```

Pramod

```
dif dijkstra (self, srs):
    dist = [sy.maxsize] * Seft.V
      dist[src] = 0
      spset = [False] * Self.V.
  for Count 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)

print ("Enter the No. of Vertex ", end=" ")
~~n = input (int~~
  n = int (input ())
  g = Graph (n)
print ("Enter matrix")
  matrix = []
```

Pramod

```
for i in range(n):
    row =[]
     row= list(map.(int, input(), .split (" ")))
      matrix . append(row)
  g.graph =matrix
  print("Enter src vertex")
   src= int(input())
    print()
    g . dijkstra(src)
```

Pramod