

Dijkstra's algorithm

Shubham R  
18011810034  
veng

import

class Network:

def \_\_init\_\_(self, nodes):

self.nodes = nodes

self.graph = [[0 for col in  
range(nodes)]

for row in range(nodes)]

def print\_table(self):

print("Source \t Destination \t Distance")

for node in range(self.nodes):

print(str(self.src) + "\t" +

str(node) + "\t" + str(dist(node)))

def min\_distance(self, dist, visited):

min\_distance = float("inf")

for v in range(self.nodes):

if dist[v] < min\_distance and  
not visited[v]:

min\_distance = dist[v]

min\_node = v

return min\_node.

def dijkstra(self, src):

self.src = src

self.dist = [float('inf')] \* self.nodes

self.dist[src] = 0

visited = [False] \* self.nodes

for \_ in range(self.nodes):

u = ~~min~~ self.min\_distance(  
self.dist, visited)



visited[v] = True

for v in range(self.nodes):

if self.graph[v][v] > 0 and  
not visited[v] and

self.dist[v] > self.dist[v] +  
self.graph[v][v]:

self.dist[v] = self.dist[v] +  
self.graph[v][v]

self.print\_table()