12|1|23 LAB -11

Djistra's

```
from collections import defaultdict
class Graph():
    def init_(self):
    self".edges = defaultdict(list)
    self. ~~edges~~ wt = { }

def addEdge(self, frmnode, tonode, wt):
self.edges[fromnode].append(tonode)
self.edges[tonode].append(fromnode)
self.weights[(frmnode, tonode)] = wt
self.weights[(tonode, frmnode)] = wt

def dijsktra(graph, initial, end):
shortest path = {initial:(None, 0)}
curr_node = initial
vis = set()

while currnode ! = end:
    vis.add(currnode)
    dest = graph.edges[currnode]
    wt_tocur = shortest_path[currnode][1]

for next_node in dest:
  wt = graph.wts[(currnode, nxtnode)] + wt_tocur
  if nxtnode not in shortestpath:
     shortestpath[nxtnode] = (currnode, wt)
  else:
     curr_short_wt = shortestpath[nxtnode][1]
     if curr_short_wt > wt:
        shortestpath[next_node] = (curr_node, wt)
```

```
nxt_dest = { node : shortestpath [node] for node in
shortestpath if node not in visited }
if not nxt_dest:
    return "Route not possible"
currnode = min (nxt_dest, key = lambda K: nxt_dest[K][1])

path = []

while currnode is not None:
    path. append (currnode)
    nxt_node = shortestpath [currnode] [0]
    currnode = nxtnode
```

~~O/P:~~

```
# Test case
g = Graph()

g.addEdge ('a', 'b', 4)
g. addEdge ('a', 'c', 2)
g. addEdge ('b', 'c', 1)
g. addEdge ('b', 'd', 5)
g. addEdge ('c', 'd', 8)
g. addEdge ('c', 'e', 10)
g. addEdge ('d', 'e', 2)
g. addEdge ('d', 'z', 6)
g. addEdge ('e', 'z', 5)
dijkstra(g, 'a', 'z')
```

o/p:
```
Shortest weight = 14
['a', 'c', 'b', 'd', 'z']
```

12/1/2023