

18M18CS1112

Sumanth.K.V.

5<sup>th</sup> C

Computer Networks.

Pgm: 2

class topology:

```
def __init__(self, array_of_points):
    self.nodes = array_of_points
    self.edges = []
```

```
def add_direct_connection(self, p1, p2, cost):
    self.edges.append((p1, p2, cost))
    self.edges.append((p2, p1, cost))
```

```
def distance_vector_routing(self):
    import collections
    for node in self.nodes:
        dist = collections.defaultdict(int)
        next_hop = {node: node}
        for other_node in self.nodes:
            dist[other_node] = 100000000
        for i in range(len(self.nodes)-1):
            for edges in self.edges:
                src, dest, cost = edges
                if dist[src] + cost < dist[dest]:
                    dist[dest] = dist[src] + cost
                    if src == node:
                        next_hop[dest] = dest
                    if src in next_hop:
                        next_hop[dest] = next_hop[src]
```

Sumanth.K

```
self.print_routing_table (node, dist, next_hop)
print()
```

```
def print_routing_table (self, node, dist, next_hop):
    print("Routing table for {node}")
    print("Dest \t cost \t next hop")
    for dest in dist.keys():
        print(f"{dest} \t {dist[dest]} \t {next_hop[dest]}")
```

```
nodes = input("Enter nodes: ").split()
t = topology(nodes)
edges = int(input("Enter no of connections"))
```

```
for _ in range(edges):
    src, dest, cost = input(" ").split()
    t.add_direct_connection (src, dest, int(cost))
```

```
t.distance_vector_routing()
```

① humanth, cv