Hardhik Agrawal
1BM18CS134

classmate
Date
Page

Cycle-2 program-2

Q) Write a program for distance vector algo to find a suitable path for transmission.

```
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_nodes != nodes:
                dist [other_nodes] = 1000000

        for i in range (len (self.nodes)-1):
            For edge in self.edges:
                src, dist, cost = edge
                if dist[src]+cost < dist[dest]
                    dist[dest] = dist[src]+cost
                if src = node:
                    next_hop [dest] = dest
                elif src in next_hop:
                    next_hop[dest] = next_hop[src]
```

Scanned with CamScanner

```
self.print-routing-table(node, dist, next-hop)
print()

def print-routing-table(self, node, dist, next-hop)
    print (Routing table) for {nodes}:')
    print ('Dest \t cost \t Next hop')
    for dest, cost in dist.ite():
        print (f'{Dest} \t {cost}
            \t ={next-hop [dest]}")

nodes = input ('Enter node:').split()
t = topology (nodes)
Edges = int (input ('Enter number of connections'))

for _ in range (edges)
    src, dest, cost =
        input ('Enter [src] [dest] [cost:')
                                split()]
    t. add-direct = connections (src, dest, int(cost))

t. distance- vector- routing ()
```