Ajay Mittur
1BM16CS006

Distance Vector Algorithm

```
class Topology:
    def __init__(self, array_of_points):
        self.node = array_of_points
        self.edges = []

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

    def bellman_ford(self):
        import collections
        for node in self.nodes:
            dist = collections.defaultdict(int)
            for other_node in self.nodes:
                if other_node != node:
                    dist[other_node] = 1000000
        # Bellman Ford
        for i in range(len(self.nodes)-1):
            for edge in self.edges:
                dist[edge[1]] = min(dist[edge[1]],
                    dist[edge[0]] + edge[2])

        print_routing_table(node, dist)

def main():
    nodes = input('Enter nodes').split()
    t = Topology(nodes)
    edges = int(input('Enter number of edges:'
    for i in range(edges):
        u, v, cost = input('enter [u] [v] [cost]')
        t.add_edge(u, v, int(cost))
    t.bellman_ford()
```