VYSHNAVI.M.N
1BM18CS125

Distance Vector Algorithm

```
class Routing_table:
    def __init__(self, current):
        self.current = current
        self.table = {'Dest': [current], 'Cost': [0],
                      'NextH': [current]}
        self.neig = []

    def add_direct_connection(self, p, cost):
        self.table['Dest'].append(p)
        self.table['Cost'].append(cost)
        self.table['NextH'].append(p)
        self.neig.append(p)

    def receive_msg(self, dest, cost, sender):
        flag = 0
        index_of_sender = self.table["Dest"].index(sender)
        cost_to_sender = self.table["Cost"][index_of_sender]
        for d, c in zip(dest, cost):
            if d not in self.table["Dest"]:
                self.table["Dest"].append(d)
                self.table["Cost"].append(cost_to_sender
                                          + c)
                self.table["NextH"].append(sender)
                flag = 1
            elif cost_to_sender + c < self.table["Cost"]
                [self.table["Dest"].index(d)]:
                self.table['Cost'][self.table['Dest'].
                index(d)] = cost_to_sender + c
                self.table["NextH"]["self.table["Dest"].
                index(d)] = sender
```

```
        flag = 1
    if flag == 1:
        return self.neig, [self.table['Dest'],
    self.table["Cost"], self.current]
    return [], []

def print_routing_table(self):
    for x, val in self.table.items():
        print(x, val)
```