AYAN DAS
1BM18CS022

Date: Page:

Distance Vector Algorithm

```
class Graph():
    def __init__(self, vert):
        self.numvert = vert
        self.graph = []

    def addEdge(self, from, to, weight):
        self.graph.append([from, to, weight])

    def printDists(self, x, arr):
        print("Dist from:" + str(x))
        for i in range(self.numVert):
            print(str(i) + "\t"*2 + str(arr[i]))

    def bfAlgo(self, source):
        cost = [float('inf')] * self.numVert
        cost[source] = 0

        for _ in range(self.numVert - 1):
            for fr, to, wt in self.graph:
                if cost[fr] != float("inf") and
                    cost[fr]+wt < cost[to]:
                    cost[to] = cost[fr]+wt

        self.printDists(source, cost)
```

→

@ayan...

Date: _____ Page: _____

```
def startNon_mat():
    n = int(input("Enter num of vertices: "))
    g = Graph(n)
    x = 0
    while x == 0:
        x = int(input("1 to insert edge \n 2 to display
                       Distance Vector"))
        if x == 1:
            fr = int(input("Enter Source :"))
            to = int(input("Enter dest :"))
            wt = int(input("Enter weight :"))
            g.addEdge(fr, to, wt)
            x = 0
        else:
            for i in range(5):
                g.bfAlgo(i)
```

~~def startMat()~~

@ryan