

Distance vector algorithm

Amogh Karliadkar
18M18CS014

```
class Graph():
```

```
    def __init__(self, vert):
```

```
        self.numVert = vert
```

```
        self.graph = [ ]
```

```
    def addEdge(self, fr, to, weight):
```

```
        self.graph.append([fr, to, weight])
```

```
    def printDists(self, x, arr):
```

```
        print("Distances from " + str(x))
```

```
        for i in range(self.numVert):
```

```
            print(str(i) + "\t" * 2 + str(arr[i]))
```

```
    def BellmanFord(self, source):
```

```
        cost = [float('inf')] * self.numVert
```

```
        cost[source] = 0
```

```
        for _ in range(self.numVert - 1):
```

```
            for fr, to, weight in self.graph:
```

```
                if cost[fr] != float('inf') and
```

```
                    cost[fr] + weight < cost[to]:
```

```
                        cost[to] = cost[fr] + weight
```

```
        self.printDists(source, cost)
```

```
n = int(input("Enter number of vertices"))
```

```
g = Graph(0)
```

```
x = 0
```

```
while(x == 0):
```

```
    x = int(input("Enter 1 to add an edge \n 2 to get answer"))
```

```
    if(x == 1):
```

```
        fr = int(input("Enter source: "))
```

```
        to = int(input("Enter destination: "))
```

```
        w = int(input("Enter weight: "))
```

```
        g.addEdge(fr, to, w)
```

```
        x = 0
```

```
    else:
```

```
        for i in range(n):
```

```
            g.BellmanFord(i)
```