

Distance Vector Algorithm using Bellmanford Algorithm

class Network:

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

```
    self.matrix = []
```

```
    self.n = n
```

```
def addLink(self, u, v, w):
```

```
    self.matrix.append((u, v, w))
```

```
def printtable(self, dest, src):
```

```
    print('Vector table')
```

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

```
        print(src, dest, dist[i])
```

```
def algo(self, src):
```

```
    dist = [999] * self.n
```

```
    dist[src] = 0
```

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

```
        for u, v, w in self.matrix:
```

```
            if dist[u] != 999 and dist[u] + w < dist[v]:
```

```
                dist[v] = dist[u] + w
```

```
    self.printtable(dist, src)
```

~~Ques~~

In DVR protocol, each node informs its neighbours of topology changes periodically. Each node maintains a distance vector table containing the distance vector to each node and src. Optimization is done using bellman ford algorithm.