

# Group Assignment CSC 450 - Computer Networks

## Distance-vector algorithm

---

### Program overview

This program takes a network topology specified in a csv file and calculates the distance vectors for each node in the specified network. The distance vector estimates are calculated using the Bellman-Ford equation.

### Command Line Input

```
python dv_algorithm.py {topology_filepath}
```

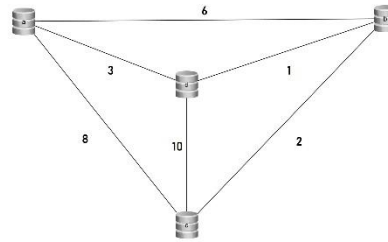
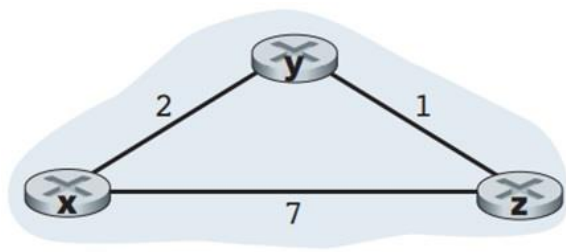
Note: the topology must be a comma separated file with node names on the first row and first column. The cost of links between a row node and a column node are the other cells

Consider the sample format for topology 1:

	<b>X</b>	<b>Y</b>	<b>Z</b>
<b>X</b>	0	2	7
<b>Y</b>	2	0	1
<b>Z</b>	7	1	0

### Outputs

Below are sample runs of the program for sample topologies:



```

distancevector> python .\dv_algorithm.py .\topologies\1\topology.csv
Distance vector for node x: 0 2 3
Distance vector for node y: 2 0 1
Distance vector for node z: 3 1 0
distancevector> python .\dv_algorithm.py .\topologies\2\topology2.csv
Distance vector for node a: 0 4 6 3
Distance vector for node b: 4 0 2 1
Distance vector for node c: 6 2 0 3
Distance vector for node d: 3 1 3 0

```

## Python Version

The program is run using Python 3. Both members used Python 3.6

## Member Responsibilities

Andre Aguiard formatted the output of the file and created the README. Andrew Schoonmaker implemented the dv algorithm and the load topology functions.

Git commit history can be tracked here: [github.com/schoobydrew/distancevector](https://github.com/schoobydrew/distancevector)