README

This code was implemented using *Python version 3.8* and makes use of the *sys* and *csv* libraries. This file includes instructions on how to run this program as well as a screenshot from a sample program run. It also includes the tasks each group member took in order to make this program work.

I. Responsibilities

a. *Parsing the Input File:* For parsing the CSV, it was unanimously decided to use Owen Sutka's parser from the program previously written to perform Dijkstra's algorithm. Therefore, credit for the csv parser function used in this program goes to Owen Sutka.

b. *Distance-Vector Algorithm:* Since this part of the code needed to be a cohesive whole, it was implemented during a pair programming session between Eboni Williams and Yinghao Lin.

   i. *Initializing the Node Graphs:* Eboni Williams

   ii. *Creating DV Updates:* Eboni Williams (during initialization), Yinghao Lin (after applying new cost path)

   iii. *Updating the Node Graphs:* Eboni Williams

   iv. *Finding a New Cost Path:* Yinghao Lin

   v. *Applying New Cost Path:* Yinghao Lin

c. *Displaying Final Distance Vectors:* Because the code relied heavily on dictionaries (which have no specific order in Python), it was very important that the dictionary be parsed, and the distance vectors be displayed correctly. Owen Sutka was responsible for making sure the dictionary was parsed correctly and the distance vectors were displayed correctly.

II. Instructions

a. At the run-time, the name of the csv file must be specified after the program call. Acceptable usage format is

   i. python dv_algorithm.py topology.csv

b. For this program to work, a comma-separated-values (csv) file containing the math topology must be provided. Please note that the csv file must be structured as a matrix with a list of nodes designated in the first row and column. The intersection of the first row and column should be left empty. The remaining cells should contain integers. Furthermore, any nodes that

aren't neighbors should have an integer value significantly higher than other values that indicate distance to neighbors.
NOTE: *Failure to conform to this format will result in an error. An example has been provided (note: in this example, all nodes are neighbors):*

```
,x,y,z
x,0,2,7
y,2,0,1
z,7,1,0
```

Figure 1 *csv as viewed in text editor*

III. Sample Run

```
$ python dv_algorithm.py 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
```