

In this assignment, you are to implement Dijkstra's algorithm to compute the shortest path tree and the cost of least-cost paths for a given node in the network.

Requirements:

1. First, your program has to read the costs of all links in a given network from "topology.csv" file (provided on Moodle). In this file, the first row and the first column refer to the names of the nodes (from *u* to *z*). Other cells show the cost of links between the row node and the column node (meaning two nodes are neighbors). The cost value "9999" indicates an infinity cost (meaning two nodes are not neighbors).

Your program has to take the name of the topology file as a command line argument.

```
python dijkstras_algorithm.py topology.csv
```

2. Next, your program has to take a node's name as an input from a command line and print the shortest path tree and the cost of least-cost paths for this node.

Sample run:

```
at@at:~$ python dijkstras_algorithm.py topology.csv
Please, provide the node's name: u
Shortest path tree for node u:
ux, uw, uwv, uwvy, uwvyz
Costs of least-cost paths for node u:
u:0, v:6, w:3, x:5, y:10, z:12
```

3. The computation of the shortest path tree and the costs of least-cost paths has to be implemented with Dijkstra's algorithm.
4. Please, comment your code thoroughly, explaining the steps of Dijkstra's algorithm and the overall flow of your program (e.g. parsing of input file, generating shortest path tree output, etc.) In addition, please, properly specify any sources you have used.
5. Create a "readme.pdf" in which specify Python version you have used, instructions of how to run your program, and screenshots of some sample program runs.
6. Submit zipped folder with your Python source code and the readme.pdf file on Moodle.