

Dijkstra's Algorithm

Description Use the source code from last lab as a building block for this lab (whether it was completely finished or not). It contains the definition of a struct `Graph`, which stores the information needed to represent a directed graph using adjacency lists. Just like last assignment, the class provided cannot be changed. You can only add methods, if needed, and variables, but you cannot change the underlying representation used to store the adjacency list.

Questions and input structure

1. If you did not finish last assignment's question 2, you are allowed to finish it and submit it along with this lab. (The challenging part was to implement breadth-first search for operator 3.)
Testcase number to use with `grade_me`: 49.
2. Read the graph from the keyboard as in last assignment's question 1. Then start a cycle where you read a couple of vertices (one per line) and you print out the length of the shortest path connecting the vertices. Print `INFINITY` if the vertices exist but no path is found. Print `ERROR` if one or both the vertices do not exist. Stop your program when the string `END` is read.
Testcase number to use with `grade_me`: 19.

Examples of input and output

Input for Test Case 49

```
Bremen
Hannover
Hamburg
Osnabruck
END
Hannover Bremen 123
Hannover Osnabruck 50
Bremen Hamburg 113
Hamburg Hannover 190
END
1 Bremen
1 Brem
2 Osnabruck Hannover
2 Hannover Osnabruck
3 Hannover Hannover
3 Hannover Hamburg
3 Osnabruck Bremen
0
```

Output for Test Case 49

1
0
-1
50
0
2
-1

Input for Test Case 19

Bremen
Hannover
Hamburg
Osnabruk
END
Hannover Bremen 123
Hannover Osnabruk 50
Bremen Hamburg 113
Hamburg Hannover 190
END
Hannover Bremen
Hannover Paris
Hannover Hamburg
Osnabruk Bremen
Hamburg Hamburg
END

Output for Test Case 19

123
ERROR
236
INFINITY
0

Your solutions Before leaving the lab, submit a zipped tar archive of your program through the assignments page of UCMCROPS. Please use your UCMNetID as the filename for the zipped tar archive. Be careful since UCMCROPS strictly enforces the assignment deadlines (deadlines will be every lab date at either 4:20pm or 7:20pm depending on your lab session.).