

EE422C Project 3 (Word Ladder) Test Plan

Numan Gilani ng22457

Miguel Garza Mag8238

Fall 2018

Test Plan Summary:

We wanted to test both algorithms for a variety of cases, including cases where no ladder exists. We used JUNIT to test the entire Main class at once. In our test suite, we covered the cases in DFS where the ladder was significantly long, where it was relatively short, and where a ladder did not exist. We also tested our BFS when the words were relatively far, close to one another, and where a ladder did not exist. We also tested the case where the start and end words were neighbors (to insure proper output). We did not cover cases where a word ladder did not exist but both words had at least one neighbor.

1)

c) Test name: *long_ladder_dfs*

b) What feature does the test cover:

This test covers the case where the existing word ladder is relatively long, to check that stack overflow or out of memory errors do not occur.

c) Set up for the test—initialization.

None.

d) Expected output for a good module.

A word ladder of a few thousand rungs.

e) The pass/fail criterion for the test.

A stack overflow or a no existing ladder output can be considered fails.

f) Any comments, if any.

2)

a) Test name: *no_ladder_dfs*

b) What feature does the test cover

This tests that the DFS algorithm does not have a memory error when traversing the whole dictionary when a ladder does not exist.

c) Set up for the test—initialization.

None.

d) Expected output for a good module.

This should output that no ladder exists.

- e) The pass/fail criterion for the test.
If a faulty ladder is printed or a stack overflow occurs, this test case is considered failed.
- f) Any comments, if any.

3)

- a) Test name: *Case insensitive*
- b) What feature does the test cover
This feature tests for inputs that are construed in lower and uppercase letters.
- c) Set up for the test–initialization.
None.
- d) Expected output for a good module.
*For example the user input “dElvE” and “haLve” should still have an output the same as “delve” and “halve” which is
a 1-rung word ladder exists between delve and halve.
delve
helve
halve*
- e) The pass/fail criterion for the test.
If a faulty ladder is printed or a stack overflow occurs, this test case is considered failed.
- f) Any comments, if any.

4)

- a) Test name: */quit test*
- b) What feature does the test cover
This feature tests for inputs “/quit” wherever they occur
- c) Set up for the test–initialization.
None.
- d) Expected output for a good module.
*For example, the user input “/quit” (ENTER) should terminate the program.
In addition, the user input “delve deave /quit” should terminate the program after the program for “delve deave” finishes.*
- e) The pass/fail criterion for the test.
Faulty ladder, stack overflow, or the program still running should indicate the test has failed.
- f) Any comments, if any.

5)

a) Test name: *repeated inputs*

b) What feature does the test cover

This feature tests for the same inputs running repeatedly 100 times

c) Set up for the test—initialization.

None.

d) Expected output for a good module.

For example, the user input “delve halve” should run the same time without any changes to the dictionary.

e) The pass/fail criterion for the test.

Faulty ladder, stack overflow, or the program still running should indicate the test has failed.

f) Any comments, if any.

Can set up an array or a file with the string inputs so this way, one does not need to type the words onto the keyboard over and over again.