

EE422C Project 3 (Word Ladder) Test Plan

Jessica Slaughter jts3329

*Rebecca Jiang **

Fall 2016

Test plan summary:

Our goal was to test both DFS and BFS thoroughly to ensure that no stack overflows occur and that both methods run in minimal time. This consisted of testing a wide range of ladder lengths and setting a strict standard for how long these methods should take finding a ladder. We used JUNIT to test these methods individually, providing input as direct Strings as well as through the parse method's return value. We also tested the parse method by inputting every valid input through the keyboard to ensure it ran correctly. In our final test suite, we covered DFS, BFS, parse, and printLadder, but didn't test initialize thoroughly.

Tests:

1.

- a) testDFSLong
- b) This test checks for a correct long ladder of over 2000 words using "hello" and "world"
- c) Initialized before testing.
- d) A ladder between "hello" and "world". The ladder was checked for duplicate words.
- e) No stack overflow, no duplicates in the ladder.
- f) Expected to run in less than 30 seconds.

2.

- a) testDFSShort
- b) This test checks for a correct short ladder of less than 10 words using "hello" and "sails".
- c) Initialized before testing.
- d) A ladder between "hello" and "sails." The ladder was checked for duplicate words.

- e) No stack overflow, no duplicates in ladder.
- f) Expected to run in less than 30 seconds.

3.

- a) testDFSZeroRung()
- b) This test checks for a correct ladder of zero length using “smart” and “start”.
- c) Initialized before testing.
- d) A ladder between “smart” and “start” with no words in between, printed with an N value of 0.
- e) Includes only start and end words.
- f) Expected to run in less than 30 seconds.

4.

- a) testDFSNoRung()
- b) This test checks for a correct output for a ladder that doesn’t exist using “jazzy” and “leady”.
- c) Initialized before testing.
- d) Should print “no word ladder can be found between jazzy and leady.”
- e) Prints proper dialogue for a ladder that doesn’t exist.
- f) Expected to run in less than 30 seconds.

5.

- a) testParse
- b) This test checks for a correct return value from parse based on keyboard input.
- c) Initialized before testing.
- d) An ArrayList containing the Strings “HELLO” and “SAILS” is expected after inputting “hello sails” through the keyboard.
- e) Capitalized words, no extra spaces, whitespace ignored.
- f) Only considered lowercase words that are in the dictionary, as according to the project pdf.

6.

- a) testQuit
- b) This test checks that the program quits after inputting “/quit” to the parse method.
- c) Initialized before testing.
- d) The program should quit.
- e) The program must quit for this test to pass.
- f) Only tested with the exact input of “/quit” as specified by the project pdf.

7.

- a) testBFS()
- b) This test checks for a correct ladder using words “hello” and “sails”.
- c) Initialized before testing.
- d) A ladder between “hello” and “sails.” The ladder was checked for duplicate words.
- e) No stack overflow, no duplicates in ladder.
- f) Expected to run in less than 30 seconds.

8.

- a) testBFSZeroRung()
- b) This test checks for a correct ladder of zero length using “smart” and “start”.
- c) Initialized before testing.
- d) A ladder between “smart” and “start” with no words in between, printed with an N value of 0.
- e) Includes only start and end words.
- f) Expected to run in less than 30 seconds.

9.

- a) testBFSNoRung()
- b) This test checks for a correct output for a ladder that doesn't exist using “jazzy” and “leady”.

- c) Initialized before testing.
- d) Should print “no word ladder can be found between jazzy and leady.”
- e) Prints proper dialogue for a ladder that doesn’t exist.
- f) Expected to run in less than 30 seconds.

10)

- a) testBFSABBA()
- b) This test checks that a BFS ladder from word A to word B is the same length as the BFS ladder from word B to word A.
- c) Initialized before testing.
- d) The first ladder should be the same length as the second ladder.
- e) Passes if both ladders are the same length.
- f) Should work for any pair of words.