EE422C Project 3 (Word Ladder) Test Plan
(Replace <...> with your actual data.)
Peter Yu   pmy89
Jia-luen Yang JY8435
Fall 2016

To test the ‘parse’ method, we entered different combinations of two-word inputs to see if the method returned the correct ArrayList with the words properly separated. JUnit was used to call the parse method and check the resulting ArrayList that it returned.
Throughout BFS and DFS functions, we used the debugger as well as print statements to check the status of the ArrayList.
Certain aspects of DFS that could optimize the process were not checked. We expect DFS to be less efficient than BFS; however, the time taken to run DFS is still within our expected limitations. (Always < 2 seconds)

1.
    a) 0 Rung Ladder Test
    b) If a word has only one letter difference from the other word, then the output should be a 0 rung ladder
    c) Input “thing think” into command line
    d) Console prints ‘0-rung word ladder exists between thing and think’
    e) Must print statement exactly

2.
    a) No Word Ladder Test
    b) Select two words that do not have a word ladder between them ("aldol", "drawl")
    c) Input the selected words and make sure correct dictionary is imported
    d) Console prints: ‘no word ladder can be found between <word1> and <word2>’
    e) Console statement must match
    f) [pink, ponk, bonk, blue], choose words pink and blue

3.
    a) N Word Ladder Test
    b) Select two words that have a word ladder with rungs n>0
    c) Input selected words and make sure correct dictionary is imported
    d) Console prints ‘N word ladder found between <word1> and <word2>’
    e) Console statement must match
    f) F

4.
    a) Time Test
    b) Select two words from the given dictionary that are extremely far away from one another such as “aloof” and any other word
    c) Input selected words and make sure correct dictionary is imported
    d) Console prints ‘N word ladder found between <word1> and <word2>’

e) Console statement must match, time to run must be <30seconds

5.

a) Reverse Order Test

b) Select two words that have at least 1 rung between each other. Run the test case once. Then reverse the order of the words and run it again.

c) Input the selected words and make sure the correct dictionary is imported and reversing the order correctly

d) Console should print “N word ladder found between <word1> and <word2>” and then “N word ladder found between <word2> and <word1>” for BFS. For DFS, the number would usually differ.

e) DFS should still run within <30 seconds and the stack should not overflow for either normal or reverse order test.