EE422C Project 3 (Word Ladder) Test Plan

Christian Han cjh3752
Keiran Crain klc3788
Summer 2018

Test plan summary:
When writing our test cases, we made sure that our DFS and BFS function worked for all word ladders. This included cases where there wasn't a valid word ladder and if the two words are already neighbors (different by one letter). We also made sure that our input doesn't consider case sensitivity. We used our five custom test cases as well as the provided JUNIT test cases. We also created a separate validation function to make sure the final word ladder produced was indeed a valid word ladder. Finally, we ran the provided grading scripts to make sure there wasn't any massive errors and that it was able to run the script correctly. While developing the project, we tested each method as we created them. We did not check for invalid inputs.

1.
   a) Test case 1
   b) Checks for correct printing of a word ladder between cream and apple.
   c) None
   d) Ladder between cream and ladder. No duplicates in ladder.
   e) No stack overflow, ladder has no duplicates. Ladder correct, as checked by our validate function.
   f) Test is expected to run in less than 2 seconds. Tests both BFS and DFS

2.
   a) Test case 2
   b) Checks for correct printing of a word ladder between guava and xylyl. (Test for valid words with no word ladder between them.)
   c) None.
   d) No valid word ladder between these two words.
   e) No stack overflow, ladder has no duplicates. Ladder correct, as checked by our validate function.
   f) Test is expected to run in less than 2 seconds. Tests both BFS and DFS

3.
   a) Test case 3
   b) Checks for correct printing of a word ladder between mElOn and ToXiC. (Test for valid words with case sensitivity issues)
   c) None
   d) Ladder between mElOn and ToXiC. No duplicates in ladder.
   e) No stack overflow, ladder has no duplicates. Ladder correct, as checked by our validate function.
   f) Test is expected to run in less than 2 seconds. Tests both BFS and DFS

4.

a) Test case 4
b) Checks for correct printing of a word ladder between lemon and mango.
c) None
d) Ladder between lemon and mango. No duplicates in ladder.
e) No stack overflow, ladder has no duplicates. Ladder correct, as checked by our validate function.
f) Test is expected to run in less than 2 seconds. Tests both BFS and DFS

5.

a) Test case 5
b) Checks for correct printing of a word ladder between peach and prune.
c) None
d) Ladder between peach and prune. No duplicates in ladder.
e) No stack overflow, ladder has no duplicates. Ladder correct, as checked by our validate function.
f) Test is expected to run in less than 2 seconds. Tests both BFS and DFS

6.

a) isNeighbor
b) Checks to makes sure two words are neighbors (String is different by one character)
c) none
d) returns Boolean value (true)
e) No duplicates
f) This method is used as a flag to make sure if two words are neighbors.

7.

g) validate
h) Checks to makes sure the word ladder returned is valid (Valid word ladder, with no duplicates)
i) none
j) returns Boolean value (true)
k) No stack overflow, no duplicates in ladder
l) Helper function to check for a valid word ladder.