EE422C Project 3 (Word Ladder) Test Plan

Bryan Yao byy77
Peter Nguyen ppn229
Spring 2017
Test plan summary

Our goal was to test all types of word ladders. This included cases where there were short ladders, long ladders, and no ladder. Short ladders were tested with different types of words that were close together. Long ones were words further apart. We had to check different pairs of words to find ones with no ladder as well. We mainly tested in main by running the program. However, we also used the given JUNIT tests, and when we tested we made sure that we tested each algorithm separately but still obtained a ladder on both when one existed. We probably covered most possibilities but likely didn't find the single worst-case scenario memory-wise for our algorithm. Also we did not check with words of different length if that will be tested.

1.

a) Name: Short Ladder BFS 1 - boats to moans
b) Checks for a correct ladder between two words that are close to each other in spelling
c) No set-up (just the regular initialization of our graph structure)
d) Expected output: Short ladder of 1 between the two words without duplicates
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1 second

Output: a 1-rung ladder exists between boats and moans
boats
moats
moans
2.

a) Name: Short Ladder BFS 2 - alias to alane
b) Checks for a correct ladder between two words that are close to each other in spelling
c) No set-up (just the regular initialization of our graph structure)
d) Expected output: Short ladder of less than 5 between the two words without duplicates
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1 second

Output: a 4-rung ladder exists between alias and alane
alias
amias
amins
amine
aline
alane

3.

a) Name: Long Ladder BFS 1 - house to apple
b) Checks for a correct ladder between two words that are far from each other in spelling
c) No set-up (just the regular initialization of our graph structure)
d) Expected output: Long ladder of under 10 words between the two words without duplicates
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1 second

Output: a 9-rung ladder exists between house and apple

4.

a) Name: Long Ladder BFS 1 - hello to apple
b) Checks for a correct ladder between two words that are far from each other in spelling
c) No set-up (just the regular initialization of our graph structure)
d) Expected output: Long ladder between the two words without duplicates
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1 second

Output: a 14-rung ladder exists between hello and apple

5.

a) Name: No Ladder BFS - aahed to aalii
b) Checks for a ladder between two words that cannot be reached (no ladder exists)
c) No set-up (just the regular initialization of our graph structure)
d) Expected output: No ladder exists
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1 second

Output: no word ladder can be found between aahed and aalii

6.

a) Name: Short Ladder DFS Unoptimized - boats to moans
b) Checks for a correct ladder between two words that are close to each other in spelling
c) No set-up (just the regular initialization of our graph structure), comment out optimize
d) Expected output: A ladder between the two words without duplicates
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1-2 seconds

Output: a 3236-rung ladder exists between boats and moans

7.

a) Name: Long Ladder DFS Unoptimized - hello to apple
b) Checks for a correct ladder between two words that are far from each other in spelling
c) No set-up (just the regular initialization of our graph structure), comment out optimize
d) Expected output: A ladder between the two words without duplicates
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1-2 seconds

Output: a 4308-rung ladder exists between hello and apple

8.

a) Name: Short Ladder DFS Optimized - boats to moans
b) Checks for a correct ladder between two words that are close to each other in spelling
c) No set-up (just the regular initialization of our graph structure), uncomment optimize
d) Expected output: A short ladder between the two words without duplicates
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1-2 seconds

Output: a 1-rung ladder exists between boats and moans
boats
moats
moans

9.

a) Name: Long Ladder DFS Optimized - hello to apple
b) Checks for a correct ladder between two words that are far from each other in spelling
c) No set-up (just the regular initialization of our graph structure), uncomment optimize
d) Expected output: A ladder between the two words without duplicates
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1-2 seconds

Output: a 782-rung ladder exists between hello and apple

10.

a) Name: No Ladder DFS - aahed to aalii
b) Checks for a ladder between two words that cannot be reached (no ladder exists)
c) No set-up (just the regular initialization of our graph structure)
d) Expected output: No ladder exists
e) Passes if above output conditions are met with no exceptions or overflow
f) Test should run quickly in less than 1-2 seconds

Output: no word ladder can be found between aahed and aalii

11.

a) Name: /quit
b) Checks the functionality of the /quit command
c) No set-up
d) Expected output: None, program terminates
e) Passes if program terminates with no exceptions or overflow
f) Test should run quickly in less than 1 second

Output: program terminated