Tim Yoder tjy263
Leo Xia lx939
Fall 2016

The goal with our test cases was to take care of the obvious test cases (no word ladder exists, one-word ladder). In addition, we wanted to test words that return ladders over 10 rungs. In addition, we tested a few random words expecting a ladder of 3-8 rungs. Since we are following an algorithm, we should not have logic errors for unique test cases. If the algorithm proves sufficient for end cases as well as a condition in the middle, the program should suffice for all searches.

1.
- a) One_Away_BFS
- b) This test covers when two words are one away from each other
- c) money->honey
- d) "Money"
  "Honey"
- e) If test runs and prints above output
- f) Test should run quickly since only one away (under 1 second)

2.
- a) Two_Away_3_Letter_BFS
- b) This test covers when two words are two letters away.
  This test covers when words are not five letters.
- c) hit->dot
- d) 2 rung ladder between "hit" and "dot"
- e) Expected output
- f) Under 1 second

3.
- a) No_Match_Long_BFS
- b) This test covers two words that do not have a word ladder
- c)
- d) No ladder exists message
- e) No ladder exists message formatted correctly
- f) Run time will be significantly longer (around a couple minutes)

4.
- a) Long_Match_BFS
- b) This test covers two words that will produce a long ladder
- c) stone -> money
- d) Properly outputted, 10-rung ladder
- e) Runs properly and finds best path
- f) Run time will intermediate to long (about a minute)

5.

a) No_Match_Short_BFS
b) This covers two words that do not have a ladder
c) angst->alula
d) no word ladder
e) runs properly and output formatted well
f) Run time will be short, angst does not have a lot of connections (under 10 seconds)

6.

a) Medium_Match_BFS
b) This covers two words that have a medium-length ladder
c) books->lulls
d) 6 rung latter from books to lulls
e) Run properly and format well
f) Run time will be intermediate (about 30 seconds)

7

a) Long_Match_BFS_2
b) This covers a long word ladder
c) Middy->breed
d) 11 rung ladder from middy to breed
e) Run properly and format well
f) Run time be intermediate to long(1 minute)

Now a caveat for testing the DFS algorithm was that finding a ladder can be extremely long. Thus for DFS, the testing consists of three types of cases: a 0 rung ladder, an n rung ladder, and no ladder found. Performance was also taken into account because DFS can run notoriously slow. Initialization of the dictionary takes a bit of time to complete because adjacent nodes are stored as an arraylist in each node of the dictionary but after initialization the DFS returns values pretty fast.

1

a) Letter_Difference_1_DFS
b) This covers words in the dictionary that have a 1 letter difference
c) start->smart
d) 0 rung ladder from start to smart, ladder checked for no duplicates
e) Run properly and format well, no stack overflow
f) Initialization should take less than 5 seconds and Run time should take less than 5 seconds

2

a) Letter_Difference_All_DFS
b) This covers two different words in the dictionary
c) smart->money
d) n rung ladder from start to smart, ladder checked for no duplicates
e) Run properly and format well, no stack overflow
f) Initialization should take less than 5 seconds and Run time should take less than 5 seconds

3

a) No_Ladder_DFS
b) This covers two words in the dictionary that have no ladder
c) jazzy->leady
d) no ladder found
e) Run properly and format well, no stack overflow
f) Initialization should take less than 5 seconds and Run time should take less than 5 seconds