EE422C Project 3 (Word Ladder) Test Plan

(Replace <...> with your actual data.)
Kevin Tong kyt259
Kevin Pang kwp535
Spring 2022

Test plan summary

What was your goal, methodology, and conclusion? 1 paragraph. Example: Did you use JUNIT? Did you test modules or the whole thing all at once? What areas did you feel you covered in your final test suite? What did you not cover?

We mainly tested our code as we coded it, with the goal that by doing so we would minimize the errors that might occur with actual test cases. We would create dummy variables within the methods of our code and then use the debug feature to step through and see if the code was reading the values we wanted it to read, and to see where the code would go after reading some input. We also used print statements in main to check less intensive code, such as checking to see if parse was reading the inputs correctly and seeing if it would continue the code or stop the program. We used JUNIT to confirm that our code was working as we intended and not really throughout developing a method.

1.
   a) Test name:
      Length_3_Test_BFS
   b) What feature does the test cover – 1-2 phrases or sentences.
      Checks if the BFS works correctly for a word ladder that has three words
   c) Set up for the test – initialization.
      Standard initialization
   d) Expected output for a good module.
      Using smart as the starting word and scars as the ending word should create a word ladder 3 words long
   e) The pass/fail criterion for the test.
      Pass should present "smart, scart, scars" and anything else should fail
   f) Any comments, if any.
      Test is expected to run in 2 seconds or less.

2.
   a) Test name:
      Compare_DFS_with_BFS_3
   b) What feature does the test cover – 1-2 phrases or sentences.
      Checks if the DFS correctly gives the shortest possible ladder, which should be the same length as Length_3_Test_BFS with the same inputs
   c) Set up for the test – initialization.
      Standard initialization
   d) Expected output for a good module.

Should see a ladder consisting of 3 words with the starting and ending word being the same as Length_3_Test_BFS

e) The pass/fail criterion for the test.
Any valid ladder between the starting and ending word that has a length of 3 will pass, anything else should fail

f) Any comments, if any.
Although a DFS ladder is typically longer than a BFS ladder with both of them having the same inputs, our code should give a ladder the same length as the BFS ladder

3.

a) Test name
BFS_SmartMoney

b) What feature does the test cover – 1-2 phrases or sentences.
Checks if BFS will still work with a longer ladder

c) Set up for the test – initialization.
Standard initialization

d) Expected output for a good module.
Should give a ladder consisting of 11 words (9 rungs)

e) The pass/fail criterion for the test.
Only a ladder of 11 words is allowed, first and last words must be the inputs

f) Any comments, if any.

4.

a) Test name
DFS_SmartMoney

b) What feature does the test cover – 1-2 phrases or sentences.
Checks if DFS will work with a longer ladder and also checks if it is the same length as BFS_SmartMoney

c) Set up for the test – initialization.
Standard initialization

d) Expected output for a good module.
Should give a ladder that also has 11 words (9 rungs) like the BFS, but a longer length is acceptable as well as long as it's not too long

e) The pass/fail criterion for the test.
Should not cause a stack overflow, first and last words must be the inputs, length of the ladder should be close to 11 but it can be over

f) Any comments, if any.
Like before, although a DFS ladder is typically longer than a BFS ladder, we made it so that the shortest possible ladder should be the output, so it should also be 11 words long like the BFS but the rungs in between could be different

5.

a) Test name
Compare_BFS_DFS_ShankYouth

b) What feature does the test cover – 1-2 phrases or sentences.

Compare the length of a BFS word ladder for shank to youth to the length of a DFS word ladder for shank to youth to see if we are successfully finding the shortest DFS, the two words can be replaced with any two different words

c) Set up for the test – initialization.
Standard initialization

d) Expected output for a good module.
The length of the BFS word ladder should be the same as the length of the DFS word ladder to show that the DFS is finding the shortest ladder possible

e) The pass/fail criterion for the test.
Pass if the length of the BFS ladder and the DFS ladder are equal, fail otherwise, fail if DFS creates a stack overflow

f) Any comments, if any.