**EE422C Project 3 (Word Ladder) Test Plan**

*Juan David Bravo*

*Jdb5338*

*Vidhu Appalaraju*

*vsa267*

*Fall 2016*

**Test plan summary**

Our goal was to test our program in several corner cases that we felt that the TA’s would probably check. For this we used JUNIT to test all of our test cases. We tested first a word input that we knew had a specific solution (“smart” to “money”) since our lab example showed what the outcome should be, we compared it with our code solution to verify correctness. Secondly, with tested for outcomes that we knew had caused us stack overflow in the program prior to our implementation of avoiding stack overflow. This guaranteed that our implementation for threating stack overflow worked. Additionally, we covered words that are aloof words, meaning that they can’t reach any other words nor can be reached from any word. Although we didn’t covered every single possible pair in the dictionary, we believe that our test cases covered every instance of corner cases that we could possibly affect our code.

1.

* a)  aloofToOther
* b)  Checks that the ladder is null since, no word can be derived from aloof
* c)  None
* d)  Null ArrayList
* e)  no words in the ladder
* f) checks for both DFS and BFS

2.

* a)  otherToAloof
* b)  Checks that the ladder is null since, no word can be derived from aloof
* c)  None
* d)  Null ArrayList
* e)  no words in the ladder.
* f) checks for both DFS and BFS

3.

* a)  Stackoverflow
* b)  checks a combination of words that previous to our stack overflow handler implementation has producing us stack overflow
* c)  None
* d)  ArrayList
* e)  no null Arraylist, and not Stack overflow error
* f) checks for both DFS and BFS

4.

* a)  GeneralCase
* b)  checks a combination of words that we knew what the outcome should be since it was the example in the PDF
* c)  None
* d)  ArrayList
* e)  no null ArrayList, with 9-word ladder for BFS
* f) checks for both DFS and BFS

5.

* a)  SameWordCase
* b)  checks that the size of a word ladder is 1 for all ladders returned from BFS and DFS with same words
* c)  None
* d)  ArrayList
* e)  ArrayList with size 1
* f) checks for both DFS and BFS