Assignment 4

Test Plan

*Aria Pahlavan* and *Jett Anderson*

**1. Black Box Testing**: To accomplish functional testing, we created a Junit test file called BlackBoxTest.java in which a very basic test case is taken as an input, passed through the program, and the output word ladder, expected output, is being compared to the actual output. BlackboxTest.java tests that our program executes as expected.

**- testWordLadderSolver()** is the method we used for testing the overall operation of our program.

**2. White Box Testing**: To accomplish complete branch coverage testing, we created a Junit test file called WhiteBoxTest.java in which test the key functionally of the known parts of program. Since we know how every single part of the program operates we can test them separately to ensure that those key parts operate as expected. Below is the list of the test methods used to test makeEdges(), makeLadder(), computeLadder(), validateLadder(), and other smaller parts of the program.

**- testGraph()** is used to test if the Dictionary class properly creates a graph of all dictionary words with their adjacency list.

**- testMakeLadder()** is used to test WordLadderSolver class and whether a correct word ladder is produced in the most optimized way possible.

**3. Invalid Input Test**: To ensure that our program properly catches all errors, handles them in the most efficient way, reports them to the user properly, we decided to make a Junit file called InvalidInputTest.java in which false and invalid inputs are entered and the output errors are being checked to ensure all errors have been handled correctly.

**- testInvalidInput()** will only check the errors received after invalid inputs were entered. The following errors are being checked:

a. One or both of the command line file arguments is missing.

b. At least one of the start word or end word is not a 5-letter word dictionary entry.

c. Invalid input format.