Noah French (njf5cu)
Lab 6
9/17/17
Filename: inlab6.pdf
File description: Lab 6 In-lab Report

My implementation produced the correct results! I did not have to reform my output for this in-lab. My outputs were in a different order with slightly different spacing, however, so I had to sort both files and use the -w flag before diff agreed that my output was the same as the expected output. Without the -O2 flag my program got through the program in slightly less than six minutes. (I know, I have a lot of optimizing to do.) With the -O2 flag, my program ran in about 2.5 minutes. It more than cut the time in half.

The big-Theta running speed of my program is n^2. Although the word-search component uses a quad-nested for loop, two of the loops (word length and word orientation) have set maximums. They are not affected by the number or rows, columns, or words. Thus, the big-Theta of the quad-nested for loop is just n^2. My hash table uses linear probing, so the worst case run time for find is linear. Thus, the total bit-Theta running speed is n^2. My implementation (with the added -O2 flag) solves the 250x250 grid in 149.9 seconds using words.txt as the dictionary file. It ran the 300x300 grid in 20.2 seconds using words2.txt as the dictionary file. Both these tests were run on my laptop through the virtual box.

All my issues had to do with lack of knowledge about C++ and syntax. I understood conceptually everything I needed to do, so it was just a matter of figuring out how to make C++ do what I wanted. I had some problems initially with how to use the getWordInGrid.cpp functions in my main method. I experimented with breaking that file up into a .h and .cpp but then just decided to just paste the whole thing into wordPuzzle.cpp. I tried to use the STL hash table (unordered_map) first to get my main method working first (as was recommended in class), but using the STL hash table was confusing and required adding strange flags to my Makefile. I spent over an hour trying to get my program to use unordered_map before I gave up and make my hash table class first.

The shell scripting writing was a bit more intensive that I expected. The provided command for reading in the last line of the a.out had some problems for me; it read the last line and one additional cout from my .cpp. Thus, I was unable to get my bash script working correctly on my computer before the midnight deadline, but hopefully it passes the tests.