Lab 6 Lab Report

Name: Peter Haynes
Student id: prh7yc
Date: 10/22/19
Section: Tuesdays 12:30

1. Yes, my programs did output the correct result. However, the results were outputted in the wrong order so I did have to sort them to compare correctly. Additionally, I had to change the formatting of my output to math the output files - namely changing the direction output to add spaces after N,E,S,W and making the space between the colon and the outputted word a tab rather than a number of spaces
2. After adding in the -02 flag to the makefile my program ran approximately three times faster.
3. All tests were ran on my laptop (a 2017 Macbook Pro) not the computers in Olsson 101
    a. Before adding the -O2 flag
        i. 300x300 with Words2: 16.0558
        ii. 250x250 with Words: 11.7486
        iii. 140x70 with Words: 1.59983
    b. After adding the -O2 flag
        i. 300x300 with Words2: 5.88966
        ii. 250x250 with Words: 4.80317
        iii. 140x70 with Words: 0.642722
4. For every word search, the file must run through every row. For every row it must run through every column. For every row and column it must check against all the words in the dictionary. Since the length and direction factors always run through a specific range they can be treated as constants. Thus, the big-theta runtime is $r*c*w$.
5. When implementing this lab I think the most difficult part was figuring out where to start. I had a sense of what the goal of the lab was but wasn't exactly sure how hashtables played into it or how to reconcile the hashtable with the wordsearch.cpp file. During office hours on Sunday night a TA helped me figure out a jumping off point and from there I figured out which methods I needed (and added a few helper methods). As far as the actual implementation of the methods is concerned that wasn't super difficult. My biggest problems were just trying to find out what the correct functions to use were and some syntax errors. Additionally, originally I was using a vector of list pointers to store my values, but found it was much easier to use a vector of lists so I changed that. Furthermore, I wasn't sure what the correct load factor to use was so I kind of arbitrarily set it to 0.75. In the word puzzle file, the hardest part was figuring out what the WordInGrid function did so that I could implement the quad for loop. Once I figured that out, I was having an issue where my program would print out the same word multiple

times. After discussing with my friends, I found this was happening because the word showed up for every length I checked if the word ends at the edge of the grid, so I implemented an if statement that breaks the loop if this occurs. Overall, I'd say the hardest part of the lab was just finding direction and figuring out what I was supposed to be doing rather than actual implementing things.

6. Shell script writing went pretty well. I read through the tutorial and got an accurate solution pretty quickly. I don't really mind shell writing, but I'm also not really sure when I would use shell writing over coding in C++ or Java or another programming language.