

Justin Ngo
jmn4fms
3/3/20
inlab.pdf

In my implementation for the hashTable and wordPuzzle I produced the correct results with sorting before running as well as ignoring whitespace with the -w flag. With the -O2 flag my program ran over 8 seconds faster. The runtime of my program without the -O2 flag from just compiling with clang++ was 10.101691 seconds. The runtime of my program with the -O2 flag was 1.792153 seconds. For the 250x250 grid using words.txt as the dictionary file my program ran in 1.689162 seconds. For the words2.txt and the 300x300 grid my program ran in 1.792153 seconds.

The big-theta running speed of the word-search component of my program is n^3 . When iterating through the rows and columns, the big theta runtime is n^2 . Adding the words with a runtime of n on to the runtime of rows and columns produces a big theta of n^3 .

A problem I encountered when implementing the lab was understanding how to bring the different components together in my wordPuzzle.cpp. At first I had trouble understanding how to read in words from the grid and iterate through the grid to check the words. I solved this problem by writing pseudocode and breaking the program down into separate components, especially when iterating through the nested for loops to check the words in the grid. Another problem I encountered was implementing new tools in c++ to read in words from other files such as ifstream. The shell script writing was more challenging than I thought because the syntax requirements are different from what we have learned in c++ and java. I had tried to work with floating point numbers in the shell script but learned that it only liked to work with ints. As a result, I converted seconds to milliseconds and rounded for the shell script writing to test the five

runtimes and take the average. I think that shell scripts are nice because they allow the programmer to combine multiple steps in compiling, running, and testing the code into a single command line.