Christian Kinzer

Cfk5ax

10/23/18

9:30 AM Lab

Inlab6.pdf

My implementation did produce the correct results. In order to properly compare with diff, I did have to reformat my output on the final line and sort my output, but after doing that, I found that there were no differences in my output and the correct output. Using words.txt and 300x300.grid.txt and without the -O2 flag, my program was running consistently around 30 seconds. With it, I was able to lower my average time to around 10 seconds. With the 250x250 grid and words.txt, it ran at 8.15 seconds. With the 300x300 grid and words2.txt, it ran at 8.55 seconds. The big theta running speed of my program is rows * columns, since words do not factor in. The program checks every single possible word in the grid, regardless of whether it is an actual word or not. Since the max word size is a small constant, and since there are 8 (another small constant) directions to check, the quad nested for loop runs at a big theta runtime of rows * columns. I found that the hardest part of the prelab was writing wordPuzzle.cpp. Writing the code for the hash table itself was fairly straightforward and didn't trouble me too much. However, with writing the code for wordPuzzle.cpp, the hardest part was finding a way to make sure I wasn't checking out of the bounds of the grid in each different direction. The shell scripting writing was very easy, since the program was quite basic. I was glad to learn about shell scripts because they are very prevalent in any computing system and they weren't really something I had considered before doing this lab. They seem to be a very effective way to run generally any series of commands on the command line.