Justin Gou (jyg2qhc)

3/3/2020

inlab6.pdf

In-Lab 6 Report

My implementation of the hash table and word puzzle solver produced correct results, as the output of my program matched the expected output. I did not need to reformat my output because I wrote a function that takes in parameters row, column, direction, and word and returns an output string in the format shown in the lab guide. This way, my output was able to directly match with the expected output. The only line that produced a difference when using "diff" was the line that printed the time of the program because my program ran in a different amount of time than the sample code, as expected.

The optimization flag (-O2) significantly increased the speed of my program. For the 250x250 grid using words.txt, my program without the -O2 flag took 14.904 seconds, but only took 4.066 seconds when compiled with the -O2 flag. This means it decreased the runtime by 10.838 seconds. The 300x300 grid using words2.txt produced similar results; without the -O2 flag, the program took 13.908 seconds, but with the -O2 flag, it only took 2.837 seconds. This was an decrease of 11.071 seconds, which is a significant amount relative to the runtime.

The big-theta run speed of my word-search program is $\Theta(r*c)$. This was determined by the quadruple "for" loop, which checked every possible word length at every coordinate in the grid in every direction. Fortunately, the number of directions was constant (8 total directions), and so was the maximum word size, which means we can ignore them in our big-theta analysis.

The biggest problem I encountered when implementing this lab was determining a suitable hash function to produce uniformly distributed values. I also had to determine which data structure would be best for the hash table; I chose a vector of lists, which I have never worked with before.

Shell scripting was very straightforward; I was able to do it with no problem. I have been shell scripting for many years now and I think they are incredibly useful, especially when doing tedious tasks, such as running a program many times to determine the average runtime. I like that shell scripts are easy to understand because they are simply commands that could be entered into a terminal, so it's we don't have to learn a new language to do something completely different.