Inlab6 Report

Alan Zheng (az4xfp)

CS 2150, Section 104, March 3 2020

1 Analysis

My implementation of the hash table did produce the correct results. However, my initial formatting of the output made me think I had it wrong, but I confirmed that it was just in a different order, since the number of words found were the same and after sorting the output, it ended up being the same.

The speed of my implementation on the 250x250 grid with words.txt for the prelab was on average 4.980 seconds over 5 separate trials. The speed on the 300x300 grid with words2.txt for the prelab was on average 6.978 seconds over 5 separate trials. The timer was started after the hash table was created and stopped right after all words were found and before they were printed.

Since the word-search component of my program is just a bunch of nested for loops, the max word size is a small constant, and hash table lookup is pretty much constant, the big-theta running speed of the word-search should be $\Theta(r*c)$. The only thing that is dependent on w is the building of the hash table, which the total time does not take into account.

After adding the -O2 flag, the average runtime for the 250x250 grid with words.txt went from 4.980 seconds to 4.881 seconds, which was not a huge improvement. For the 300x300 grid with words2.txt, it went from 6.978 to 5.887 seconds.

Some problems that I encountered when implementing this lab were picking a decent hash function that was fast but also unique for different strings and also selecting what data structures to represent my hash table.

Writing the shell script was confusing at first since bash is a bit different from other programming languages that I know. I think shell scripts will be extremely useful in the future so I don't have to write a bunch of the same successive Unix commands to run my code.