Alex Yang ahy3nz
11AM Lab, 104

No, my implementation did not produce the correct results. In wordPuzzle.cpp, I print out a line with words and numbers. I had to print out another line as the last line that would display just an integer that represented the time. Furthermore, since we were only using integers, I had to multiply my running times by 1,000,000 in order to obtain a valid integer and a time in microseconds.

Without the -O2 flag, resolving 50x50.grid.txt took 165018 microseconds. With the -O2 flag, it took 165688 microseconds. Without the -O2 flag, resolving 250x250.grid.txt took 4,636,746 microseconds.

With the -O2 flag, it took 4,614,081 microseconds. Interestingly, there isn't a significant difference in running time with the -O2 flag (using the machines in Ols 001)

Using my own personal computer, without the -O2 flag, resolving 250x250.grid.txt took 2,823,052 microseconds. With the -O2 flag, resolving 250x250.grid.txt took 2,760,184 microseconds. These results indicate there is a difference between using the O2 flag and not using the O2 flag.

Using words2.txt and the 300x300 grid with the -O2 flag, it took 2,667,249 microseconds on the machines in Ols 001

The big-Theta running speed of the running time is theta(r*c*w). In each row and each column (r*c) elements, we have to look for every word w that is in our hashtable. Furthermore, we are looking for words in 8 directions, but calculating big-Theta running times ignores constants. We can optimize this code, however, by simplifying how many words we look for by checking for prefixes first, and terminating some loops early if no valid prefixes were found (no prefixes means we won't find any words).

Some problems I encountered were designing the overall structure of this project. I had no idea where to start or what the purpose of some of the programs were. However, I first started designing the hashtable and then implemented that in the main() method of wordPuzzle.cpp, and then the necessary code just filled in where it needed to be. Some of the syntax for creating vectors and a vector of string lists was new, but not a daunting task. Lastly, considering prefixes was a problem when I had to make sure the prefixes weren't being counted as words, but I resolved that by finding prefixes of length 2, but only counting words that had length 3 (so prefixes were never counted). The actual nitty-gritty of finding words in tables or reading in files was, fortunately, already coded for, so we just had to adapt the code into our own programming.

The shell script writing was fine. It was just like writing in another language, just with a little different syntax for variables and expressions. Shell scripts are very convenient for running commands in terminal in bulk