

## WordcountDesignForPerl.pdf

For my program this wordcount.pl is a perl script that tries to take in a input that is either a directory or a file. It will before doing anything check if it received a file/directory, number (would suppose any number that is < then the total lines outputted), and a output file that is not present in the current directory. It has 3 if checks that were specified in the instructions provided.

I have 3 sub functions and a master if elsif else statement. The validwords and printToFile are called once if either a file or directory is inputted and it uses global variables that are declared early in the file (their commented). The wanted function is apart of the find library that you use to set up a rule for what your trying to find. I need it to find all files that are in the directory given and its subdirectories. The master if statement checks for files the else if checks for directories and else is just a catch all if its not one or the other so I suppose like a .exe or some unknown file type maybe a .msi file. Once valid words are found they are put into a hashtable for quick out put and print according to the instructions.

The performance of this perl script is way to quick. I ran your script wordcount.sh and it took to long. I didn't get a exact time for either but I expected as much as a hash table is  $O(n)$  as its just a key value table. Looping through arrays is  $O(n)$  so double looping is  $O(n^2)$  so its just obvious that perl is just really useful. In the past two weeks its got all this stuff I spent the last semester in data structures learning to write.

I really didn't like bash, in perl I only had to use a couple lines of regular expressions and I was able to find a lot online about perl so it was easier to complete this assignment once we had finished doing the perl lectures. I got to use a bunch of simple loops and counters to get around more difficult parts so I was able to cheese some of the work and was able to write my code to where I could reuse the same two master sub functions for either situation. My file went through like 3-4 versions and its grown to this like 200 line thing but I commented everything so I could read it later. This class is still probably the hardest I have and its still difficult but I might survive.