Fall 2019
CS3413
Lab 4

Profiling

One of the most important tasks when writing C code is to ensure that your program is efficient. While debugging can help you gain insight into if your program functions properly, profiling can give you insight into how efficient your program is. One of the most common profiling tools is `gprof`. This unix utility provides information on how your execution is broken down into functions (ie. how much each function is called and how much time is spent executing each function). Here is an online source on how gprof works - http://sourceware.org/binutils/docs/gprof/

Another utility is valgrind. Valgrind is developed to provide more information about how you use (or misuse) your memory allocation in a C program. Yes, it can help you find memory leaks and identify where/why those pesky segmentation faults occur! You can read more about valgrind here - http://valgrind.org/docs/manual/quick-start.html

After you have read about gprof and valgrind from the links above, please try each of the tools on the sample code provided for the lab. Get comfortable with the tool and understanding the output provided.

Next, I want you to pull out your last working assignment where you wrote some C code that works! I suggest that you **not** use a multi-threaded assignment to keep things simple. Using both tools I want you to find the answers to the following:

1. What are the top 3 functions that are called during execution.
2. From the information in #1, do you see any ways you can improve the execution of your program?
3. Are there any memory leaks in your previous work?
4. If yes to #3, do you know how to fix it?

Once you complete the tasks, please submit your solution via D2L.