Efficiency

Programming Assignment #8
CS253 Spring 2014
Due 6:00pm Wednesday, April 20th, 2016

Motivation

Donald Knuth said that "Premature optimization is the source of all evil", and I agree. But it is no longer premature. You now have a complex program. Time to make it fast.

Task

The task in PA8 is *exactly the same as PA7.* Same input specification, same output specification. The difference is that it will be graded on efficiency as well as correctness, not just correctness (see below).

You may not use non-standard C++ libraries for this assignment. One way to guard against this is to avoid using the –L command in your makefile (which would allow you to access directories other than the standard library directories). Also, don't make any non-standard libraries in your host directory.

Submitting your program

You will submit your program through Canvas. You should submit a single tar file, and this file should contain all the source files of your program *and a makefile*, but no object or executable files. The makefile should create an executable program called PA8. To grade your programs, the GTAs will write a script that untars your file in a clean directory, runs 'make', and then tests PA8 on novel inputs. If your program does not compile, whether because of an error in your makefile or an error in your code, you will receive no credit on any of the test cases. Note that as always, we will test your code on the department Linux machines.

Grading

Unlike in previous assignments, your program will be graded on a combination of correctness and speed. Your program will be tested on novel inputs, similar to those used to grade PA7. If your program does not produce correct output for a test case, it will receive a score of 0 for that test case. If your program produces correct output for a test case, it will receive at least 50% credit for that case; its performance (as estimated using the unix time command) will determine how much of the other 50% credit it receives.

Hints

1. Let the Valgrind profiler guide you with regard to what to optimize. It will tell you where your time is going.

2. If Valgrind does not suggest any hot spots, look to reduce memory usage.
3. Keep a correct version of PA7 as a reference. Be careful not to introduce errors in the process of optimization.