Experimental Evaluation

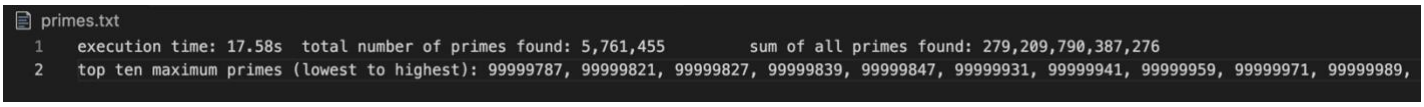
My approach to this assignment closely follows the examples from the class slides. I created a shared counter object that the eight concurrent threads can access. The shared resource is protected by Java's synchronized method. As for my method of calculating a prime number, I followed the primality test algorithm found on Wikipedia. I found it was the most efficient at calculating a prime number. One issue that I ran into was that I kept having long runtimes because I was printing each prime number along with the thread that found it to the console for testing purposes. I had to remove that part of my code because it was slowing it down dramatically. I stored the prime numbers found inside a List variable in the counter object.

I tested the program on my two different machines:

- MacBook Pro
    - Processor - 2.3 GHz 8-Core Intel Core i9
    - Graphics - Intel UHD Graphics 630 1536 MB
    - Memory - 16 GB 2667 MHz DDR4
    - Runtime for program was ~18 seconds

- PC
    - Processor - Intel Core i7-4770K CPU @ 3.50GHz
    - Graphics – NVIDIA GeForce GTX 1070 Ti
    - Memory – 15.94 GB RAM
    - Runtime for program was ~14 seconds

Regarding the correctness and efficiency of my design; I feel my design is decent but perhaps could be optimized further. It gets the job done without any errors and utilizes the specified number of threads with a balanced workload. For my first multithreading project I am satisfied with the results.



Example output from my MacBook Pro