CS6326 Human-Computer Interactions

Fall 2020

Assignment: Multithreaded Computation

In cryptography, factoring large numbers is important, since encryption keys are typically very large primes however, finding the prime factors can be time-consuming. Thus such a factorization is a prime candidate for running things in parallel on multiple threads. Write a program that does the following:

1. Show two textboxes that allow you to enter a lower bound and an upper bound for a range of numbers for which your program will find prime factors. Let me caution you to follow good user experience design guidelines even for something so seemingly simple. If necessary, go back through the “Practical Design” presentation to see what I mean. You can lose points here. For example, the upper bound must greater than or equal to the lower bound.
2. Show a button that starts the computation. There are two parts to this.
3. Compute and store all primes less than or equal to the square root of the upper bound. This can take a long time for large numbers, so display a progress indicator. I realize there is no formula for computing how many primes there are between two numbers, but there are ways to approximate this. (I tested my version with 1,000,000 and 1,000,100.)
4. Compute and display the numbers in your range, along with their unique prime factors. This can also take a long time, so report progress. The numbers and factors can go into a scrolling ListView. Once again, pay attention to your screen layout and appearance.
5. Both computations should run on a thread separate from the user interface. If you are ambitious, the factoring can use multiple threads, which on most machines will make it run significantly faster. This is not a requirement, but running computation on a separate thread from the user interface is.
6. Use the “long” data type, 64 bits, for this, since we should be able to test on numbers large enough to be interesting.

You have your choice of language, C# or Java. I have covered C# BackgroundWorker in class, but you can also use SwingWorker in Java, which is very similar.

**To hand in via eLearning:** Hand in a Zip file named with your NetID followed by Asg4. For example, mine would be jxc064000Asg4. If you wrote in C#, it should contain your entire Visual Studio project. If you wrote in java, hand in only your java source files.

|  |  |
| --- | --- |
| **Grading: 100 points possible** | |
| Program is truly multithreaded and works according to the above specification | 50 |
| Program follows other UI design rules and guidelines | 30 |
| Input handling | 10 |
| Program comments and good variable names | 10 |

Grading guidelines:

1. The program waits until it finishes to display prime factors. -50
2. No progress indicators. -20
3. Screen is fixed size, not sized to show as much information as possible. -10
4. The program can be made to crash for invalid input such as a bad file name. -10