Project 2A. Do Prime Factorization on Large Integers

(Optional, Due 8/7/2021 Saturday)

Description:

In this project, we will calculate the *prime factorization* on large integers. Computing the prime factorization could be very *time consuming* if the given integer is *huge*. In this project, we just work on those integers that our program can handle in reasonable amount of time.

Requirements:

1. (*Avoid Overflow*)

   Since the largest primitive integer data type, (e.g. in Java, the `long` data type), can suffer the *overflow* problem if we work on *sufficiently large* integers, we have to find a way to get around this problem. To overcome this *overflow* problem, we need to go to a much larger data type in this situation. In Java, we can use the `BigInteger` class to do the calculation. A Java example using the `BigInteger` class is provided to show you how to do basic mathematical operations on the `BigInteger` operands. While in Python, you do not need to worry about this problem, you just do your usual mathematical operations.

   I do not have the information on the way to solve this problem in C/C++. If you cannot find a good way in C/C++, try to use Java or Python in this project.

2. (*Generate Random Integer Input*)

   Use a random integer generator to create an input integer with certain number of digits. You can set the number of digits to 10, 20, 30, etc. and see if the execution time is too slow or not. If you can reach an integer with `100` digits with acceptable response time, it is good enough for us.

3. (*Find an Algorithm*)

   You can search the web to find a relatively simple algorithm. We do not require that it is the most efficient algorithm. We only need that it is simple to implement.

4. (*Display Output*)

   When you display your prime factorization output, you just find your own way to display the critical information: all the prime factors in ascending order, and for each prime factor, the number of times (or the power of this prime) that appears in the product. The bottom line is, when I see your display, I can understand that this integer can be represented as the product of what prime numbers including the powers of the prime numbers. After you display the factorization, also display the response time in milliseconds.

5. (*Work independently*)

   Everyone should work on the project independently. When I do grading, if I see similar behaviors, I will compare those programs line-by-line. Although you are allowed to discuss with your classmates about the ideas, the final code should be written by yourself.

Any updates about this project will be posted in Canvas as announcements.