EE 360C — Algorithms — Summer 2013
Programming Assignment #3

Due: July 31, 2013 11:59pm (via Blackboard)

Programming assignments are to be done individually. You may discuss the problem and general concepts with other students, but there should be no sharing of code. You may not submit code other than that which you write yourself or is provided with the assignment. This restriction specifically prohibits downloading code from the Internet.

Problem Description

In this programming assignment, you are required to implemenet multiple algorithms for integer multiplication and analyze their performance over the number of decimal digits in the input integers.

- Brute force integer multiplication with $O(n^2)$ operations.
- Divide and conquer integer multiplication with $O(n^{1.59})$ operations.
- (25% Bonus) Integer multiplication using Fast Fourier Transform with $O(n \log n)$ complex number operations.

The input consists of a single character (B/D/F) specifying the algorithm to use followed by two big integers of up to 10,000 digits on separate lines. For example:

```
B
123456789
987654321
```

should give the output

```
121932631112635269
```

In addition to program source code, you should submit a single plot called "plot.jpeg" with two or three line graphs—one for each algorithm you implement. The y-axis should show time in milliseconds and x-axis should show input size in number of decimal digits. You should have at least ten samples for your algorithms' time for 1000, 2000, 3000, . . . , 10000 decimal digits for both integers.

Implementation Guidelines

- On a Linux machine, you can use the `time` utility to find the running time of your program.
- You can generate test cases using the `bc` utility on a Linux machine for arbitrary big integers.
- You cannot use a builtin library for Complex numbers or for BigInteger.
- Since the input/output have decimal digits, it will be much easier for you to work with the algorithms in base-10. So instead of multiplying bits you multiply decimal digits.
- The grading for this assignment will focus on time complexity i.e. whether you are able to implement the algorithms in their respective time complexity or not.

Submission Instructions

- Make sure your program compiles on LRC Linux machines before you submit it. It should compile using the standard commands `javac *.java`, `gcc *.c`, or `g++ *.cc` without any extra switches or any additional libraries.

- Your solution must take input from console and output to console.

- Your solution must be submitted via Blackboard *before* the deadline. No late submissions will be accepted.