1. Implement Karatsuba’s algorithm. You can either use polynomials and then evaluate or do it directly on integers.

2. Implement the multiplication using Fast Fourier Transform FFT (Three prime FFT integer multiplication, Algorithm, 8.25, page 241) and the necessary functions required by this algorithm, which includes Chinese remainder, modular inverse and more. Primes and n-th root of unity for different n are provided (see cs810a_primes.h).

3. Make sure that your implementations are correct. Please provide a test program that reads n pairs of integers and outputs just the result of the multiplication for each pair. Compare the performance of classical, Karatsuba and FFT multiplication for random multi-precision integers of bit length \( l \in \{256, 512, 1024, 1536, 2048, 2560, 3072, 3584, 4096\} \).

Let \( n \in \mathbb{N} \) be the number of iterations performed per multiplication implementation and \( l \) be the bit length of the integers \( a, b \). Tests are performed the following way.

(a) Generate \( n \) pairs of integers of bit length \( l \) and store them in the arrays \( a[i] \) and \( b[i] \) (use program "cs810a_rand").

(b) Evaluate the performance of classical, Karatsuba and FFT multiplication using the following scheme:

1. start timer
2. for \( (j = 0; j < n; j++) \)
3. for \( (i = 0; i < n; i++) \)
4. multiply\( (c[i], a[i], b[j]) \);
5. stop timer
6. Output: \( \frac{\text{time difference}}{n} \).

We are using a nested loop in order to keep the size of the arrays at a reasonable size. Ensure that the multiplication algorithms use the same input data!

The number of iterations \( n \) per multiplication is dependent on the machine type and the bit length of the inputs. You should repeat steps (a) and (b) multiple times.

4. Compare your results with the program ”cs810a_gmp”, which use the multi-precision library GMP (http://www.gmplib.org).

The programs ”cs810a_rand” and ”cs810a_gmp” need GMP (http://www.gmplib.org) in order to compile. You might have to install this library including the developer files (check your package manager). See ”cs810a_gmp.c” for details on how to measure CPU time. Please document your code.

This project is due by October 23.