Programming Project, Part I

General Description:
The programming project for this semester will be to implement an efficient algorithm for the multiplication of two polynomials given in coefficient representation. This is done by applying the Discrete Fourier Transform (DFT) to obtain the point-value representation of the polynomials, multiplying the two polynomials in that representation, and applying the inverse Discrete Fourier Transform to obtain the coefficient representation of the result.

In part I of this project, you will compute the DFT and its inverse using the straightforward $\Theta(n^2)$ algorithms. You will perform timing experiments on your implementation of the DFT.

Detailed Description:
Write a procedure or method that computes the DFT. The DFT method receives the coefficients ($a_0, ..., a_{n-1}$) and computes ($y_0, ..., y_{n-1}$) according to the formula

$$y_k = \sum_{j=0}^{n-1} a_j \omega_n^{kj}.$$  

Also implement the inverse DFT method that receives ($y_0, ..., y_{n-1}$) and computes ($a_0, ..., a_{n-1}$) according to the formula

$$a_k = 1/n \sum_{j=0}^{n-1} y_j \omega_n^{-kj}.$$  

Chapter 30 in our textbook covers this problem.

Write a program that tests your procedure. You may use the programming language of your choice, but take into consideration that you will need to perform operations on complex numbers, and that you will have to perform timing experiments.

Do a sequence of timing experiments of your DFT method or procedure with polynomials of various degrees. Your program should work for any value of $n$. But since we will only use powers of 2 for $n$, so it’s ok to perform timing experiment on inputs of size powers of 2. Draw a graph with the
results of your timing experiments (time versus input size). Then, use your experimental data to find a formula for the running time of your algorithm. Assume the algorithm takes time $an^2 + bn + c$. Find the $a$, $b$, and $c$ that best approximate your experiments.

Also write a program that performs the multiplication of polynomials by applying the DFT on them, multiplying the resulting point-value representation, and applying the inverse DFT. You do not need to do timing experiments on this program.

**Turn in:**

- A report that explains the experiment you performed, how data was gathered, how you made the analysis and the results,
- A listing of your program,
- Some sample runs with input and output illustrating the correctness of your procedures.

**Due date:** October 6th, 11:59pm. The penalty is 10% for each day late up to one week late. No homework accepted after one week.