CS 3343 Analysis of Algorithms – Fall 11

9/20/11

Programming Project 1

Due **10/4/11** before class

Polynomial Multiplication (20 points)

Consider computing the following product of two polynomials of degree 3:

$$(2x^3 - 3x^2 + 4x - 5) \cdot (3x^3 - 2x^2 + x - 7)$$

Straight-forward multiplication computes this product as
$2x^3 \cdot 3x^3 - 2x^3 \cdot 2x^2 + 2x^3 \cdot x - 2x^3 \cdot 7 - 3x^2 \cdot 3x^3 + 3x^2 \cdot 2x^2 - 3x^2 \cdot x + 3x^2 \cdot 7 + 4x \cdot 3x^3 - 4x \cdot 2x^2 + 4x \cdot x - 4x \cdot 7 - 5 \cdot 3x^3 + 5 \cdot 2x^2 - 5 \cdot x + 5 \cdot 7$
which simplifies to
$6x^6 - 4x^5 + 2x^4 - 14x^3 - 9x^5 + 6x^4 - 3x^3 + 21x^2 + 12x^4 - 8x^3 + 4x^2 - 28x - 15x^3 + 10x^2 - 5x + 35$
which simplifies further to
$6x^6 - (4+5)x^5 + (2+6+12)x^4 - (14+3+8+15)x^3 + (21+4+10)x^2 - (28+5)x + 35$
and finally to $6x^6 - 9x^5 + 20x^4 - 40x^3 + 35x^2 - 33x + 35$.

Assume each polynomial is given by its sequence of coefficients (so, $2, -3, 4, 5$ and $3, -2, 1, -7$ in the example). The algorithm used above could be implemented with two nested loops.

The task of this programming project is to develop and implement a divide-and-conquer algorithm for multiplying two polynomials. So, consider breaking each of the polynomials into two parts of equal length:

$$[(2x^3 - 3x^2) + (4x - 5)] \cdot [(3x^3 - 2x^2) + (x - 7)]$$

which can be rewritten as

$$[x^2(2x - 3) + (4x - 5)] \cdot [x^2(3x - 2) + (x - 7)]$$

Then the product can be computed as

$$x^2(2x-3) \cdot x^2(3x-2) + x^2(2x-3) \cdot (x-7) + (4x-5) \cdot x^2(3x-2) + (4x-5) \cdot (x-7)$$

This essentially just requires four multiplications of polynomials of degree 1: Note that $x^2(2x-3) \cdot x^2(3x-2)$ can be rewritten as $x^4 \cdot (2x-3) \cdot (3x-2)$, where the second multiplication is between polynomials of degree 1, and the first multiplication with x^4 is a shift of the coefficients by four places to the left. So, we reduced the problem of computing the product of two polynomials of degree 3 to four multiplications of polynomials of degree 1, plus some shifts in the coefficients.

Use this to develop and implement a divide-and-conquer algorithm that computes the product of two polynomials of degree d. Assume d has the form $2^k - 1$, so $d = 1, 3, 7, 15, 31, \ldots$. Test your code with some test cases and document these tests.

FLIP OVER TO BACK PAGE $\Longrightarrow$

Turnin instructions

- You are allowed to turn in this programming project in groups of two.
- You can use Java, C, or C++ for this project. If you want to use a different programming language, check with our TA first.
- **The name of your project directory should be** `project1_<lastName1><firstName1><lastName2><firstName2>`
- Zip up a directory with your entire project (source code and test case report). Turn in the zip file by uploading it to Blackboard. In the comments section during the upload to Blackboard please add instructions on how to compile the program and on how to run the test cases.
- All projects need to compile. If your program does not compile you will receive 0 points on this project.
- Do not use any fancy libraries. We should be able to compile it under standard installs of Java, C, or C++ under linux and/or windows.