CSE2100 Data Structures and Introduction to Algorithms
Divide-and-Conquer Algorithm for Polynomial Multiplication

A (univariate) polynomial is a mathematical expression involving the sum of powers of one variable, usually denoted by $x$, multiplied by constant coefficients. The degree of a polynomial is the highest power of $x$ which has a non-zero coefficient is non-zero. For example, a generic polynomial of degree $n$ can be written as

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

The standard polynomial multiplication algorithm works by multiplying one of the polynomials by the terms (monomials) of the other and adding the results. This requires $O(n^2)$ time for multiplying two polynomials of degree $n$. The following recursive algorithm, based on the divide-and-conquer method, is asymptotically faster.

Let $A(x) = a_n x^n + \ldots + a_2 x^2 + a_1 x + a_0$ and $B(x) = b_n x^n + \ldots + b_2 x^2 + b_1 x + b_0$ denote two polynomials, each of degree $n$, and $m = \lceil n/2 \rceil$. By grouping monomials of degree higher (resp. lower) than $m$ in $A(x)$, we can write

$$A(x) = x^m A_{hi}(x) + A_{lo}(x)$$

where $A_{hi}(x) = a_n x^{n-m} + \ldots + a_{m+1} x + a_m$, where $A_{lo}(x) = a_{m-1} x^{m-1} + \ldots + a_2 x^2 + a_1 x + a_0$. Similarly,

$$B(x) = x^m B_{hi}(x) + B_{lo}(x)$$

where $B_{hi}$ and $B_{lo}$ are two polynomials of degree (at most) $n-m$ and $m-1$, respectively. By using standard properties of polynomial arithmetic it follows that

$$A(x)B(x) = x^{2m} [A_{hi}(x)B_{hi}(x)] + x^m [A_{hi}(x)B_{lo}(x) + A_{lo}(x)B_{hi}(x)] + [A_{lo}(x)B_{lo}(x)]$$

Let

$$P_1(x) = A_{hi}(x)B_{hi}(x)$$

$$P_2(x) = A_{hi}(x)B_{lo}(x) + A_{lo}(x)B_{hi}(x)$$

$$P_3(x) = A_{lo}(x)B_{lo}(x)$$

From (1), computing the product $A(x)B(x)$ can thus be done by first computing polynomials $P_1(x)$, $P_2(x)$ and $P_3(x)$, then multiplying $P_1(x)$ by $x^{2m}$ and $P_2(x)$ by $x^m$, and finally adding the results with $P_3(x)$. According to (2) and (4), $P_1(x)$ and $P_3(x)$ can be computed using two recursive calls to the polynomial multiplication method. However, instead of computing $P_2(x)$ using (4) directly (which would require two recursive calls to the polynomial multiplication method), we can compute it with a single recursive call by observing that

$$P_2(x) = [A_{hi}(x) + A_{lo}(x)][B_{hi}(x) + B_{lo}(x)] - [P_1(x) + P_3(x)]$$

This scheme results in computing the product of two polynomials of degree $n$ with an asymptotic worst-case running time of $O(n \log_2 3) = O(n^{1.58\ldots})$. 

1