
# Algorithm Theory

## 02 - Polynomial Multiplication and Fast Fourier Transform

**Dr. Alexander Souza**

D & C : $O(n \log n)$ FFT

# 1. Polynomials

**Real polynomial** *p* **in one variable** *x*:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x^1 + a_0$$

$a_0, \ldots, a_n \in R$: coefficients of *p*

degree of *p*: highest power of *x* in *p* (= *n*)

**Example:**

$p(x) = 3x^3 - 15x^2 + 18x$

Set of all real polynomials: $R[x]$

# 2. Operations on polynomials

$$p, q \in R[x]$$

$$p(x) = a_n x^n + \ldots + a_1 x^1 + a_0$$
$$q(x) = b_n x^n + \ldots + b_1 x^1 + b_0$$

**1. Addition**

$$p(x) + q(x) = (a_n x^n + \ldots + a_0) + (b_n x^n + \ldots + b_0)$$
$$= (a_n + b_n) x^n + \ldots + (a_1 + b_1) x^1 + (a_0 + b_0)$$

gather by $x^i$

Running time $O(n)$, optimal

# Operations on polynomials

**2. Multiplication:** deg (p q) = deg (p) + deg (q)

$$p(x)q(x) = (a_n x^n + \ldots + a_0)(b_n x^n + \ldots + b_0)$$
$$= c_{2n} x^{2n} + \ldots + c_1 x^1 + c_0$$

$c_i$: What products of monomials have degree $i$? Those whose indices sum up to $i$

$$\Rightarrow c_i = \sum_{j=0}^{i} a_j b_{i-j} \quad i = 0,\ldots,2n. \qquad c_i = a_0 \cdot b_i + a_1 b_{i-1} + \ldots + a_i b_0$$

$$a_{n+1} = \ldots = a_{2n} = 0, b_{n+1} = \ldots = b_{2n} = 0$$

$i+1$ mult
$i$ add

Polynomial ring $R[x]$.

$$\sum_{i=0}^{2n} 2i+1 = O(n^2)$$ running time expensive

Goal: $O(n \log n)$

# Operations on polynomials

**3. Evaluation at a specific point $x_0$: Horner's method**

$$p(x_0) = (\ldots(a_n x_0 + a_{n-1})x_0 + \ldots + a_1)x_0 + a_0$$

Running time: O(n)

Example

$p(x) = 3x^3 + 7x^2 - 4x + 10$

$= (3x^2 + 7x - 4)x + 10$

$= ((3x + 7)x - 4)x + 10$

O(n) running time

# 3. Representation of polynomials

$p(x) \in R[x]$

**Possible representations of** $p(x)$**:**

## 1. Coefficient representation

$$p(x) = a_n x^n + \ldots + a_1 x^1 + a_0$$

**Example:**

$$p(x) = 3x^3 - 15x^2 + 18x$$

# Representation of polynomials

## *2.* Product of linear factors

Polynomial of degree n has exactly n (complex) roots

$p(x) \in R[x]$

$$p(x) = a_n (x - x_1) \ldots (x - x_n)$$

**Example:**

$$p(x) = 3x(x-2)(x-3)$$

# Representation of polynomials

## 3. Point-value representation

**Interpolation lemma:**

Any polynomial p(x)∈ *R[x]* of degree *n* is uniquely defined by *n*+1 pairs $(x_i, p(x_i))$, where $i = 0,\ldots,n$ and $x_i \neq x_j$ for $i \neq j$.

point value

**Example:**

The polynomial

$$p(x) = 3x(x-2)(x-3)$$

is uniquely defined by the point-value pairs (0,0), (1,6), (2,0), (3,0).

$p(0) = 0$ , $p(1) = 3 \cdot 1 \cdot (1-2) \cdot (1-3) = 6$

# Operations on polynomials

$p, q \in R[x]$, degree($p$) = degree($q$) = $n$

- **Coefficient representation**

  Addition: O($n$)

  Multiplication: O($n^2$)

  Evaluation at $x_0$ : O($n$) Horner

- **Point-value representation**

$$p = (x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

$$q = (x_0, z_0), (x_1, z_1), \ldots, (x_n, z_n)$$

# Operations on polynomials

**Addition:**

$$p+q=(x_0, y_0+z_0),(x_1, y_1+z_1),\ldots,(x_n, y_n+z_n)$$

Running time: O(*n*)

**Multiplication:**

$$p\cdot q=(x_0, y_0\cdot z_0),(x_1, y_1\cdot z_1),\ldots,(x_n, y_n\cdot z_n)$$

(Condition: $n \geq \text{degree}(pq)$)

Running time: O(n)

**Evaluation at point** *x´:* ??
Convert polynomial to coefficient representation (interpolation)

# Polynomial multiplication

Compute the product of two polynomials *p, q* of degree < *n*:

initially coefficient representation

$p,q$ of degree *n-1*, *n* coefficients

↓ **Evaluation:** $x_0, x_1, \ldots, x_{2n-1}$ n. Horner $O(n^2)$ ⇝ FFT $O(n \log n)$

2*n* point-value pairs $(x_i, p(x_i))$ und $(x_i, q(x_i))$

↓ **Pointwise multiplication** $O(n)$

2*n* point-value pairs $(x_i, pq(x_i))$

↓ **Interpolation** ?

$pq$ of degree *2n-2*, *2n-1* coefficients

# Divide-and-conquer approach

for polynomial evaluation

**Idea:** (assume *n* is even)

$$p(x) = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1}$$

Separate odd and even indexed terms

$$= a_0 + a_2 x^2 + \ldots + a_{n-2} x^{n-2} + a_1 x + a_3 x^3 + \ldots + a_{n-1} x^{n-1}$$

Factor out $x^2$ ($a_4 \cdot (x^2)^2 + a_6 (x^2)^3 + \ldots$)

$$= a_0 + a_2 x^2 + \ldots + a_{n-2} \left(x^2\right)^{(n-2)/2} + x\left(a_1 + a_3 x^2 + \ldots + a_{n-1} \left(x^2\right)^{(n-2)/2}\right) \quad (a_5 (x^2)^2 + \ldots)$$

Factor out $x$ and $x^2$ in the rest

$$= p_0\left(x^2\right) + x p_1\left(x^2\right)$$

Instead of evaluating one polynomial with degree $n$ we evaluate two polynomials with degree $\frac{n}{2}$

$$p_0(x) = a_0 + a_2 x + \ldots + a_{n-2} x^{(n-2)/2}$$
$$p_1(x) = a_1 + a_3 x + \ldots + a_{n-1} x^{(n-2)/2}$$

$p_0(x^2) = a_0 + a_2 x^2 + a_4 x^4 \ldots$

$p_1(x^2) = a_1 + a_3 x^2 + \ldots$

Select $x_0, \ldots, x_{2n-1}$ such that the computations of $p(x_k)$ and $p(x_{k+n})$ are almost identical.