Polynomials

\[ A(x) = \sum_{i=0}^{n-1} a_i x^i \]

\( n \) - the degree of the polynomial.

\( a_0, \ldots, a_{n-1} \) - the coefficients of the polynomial.

**Coefficient representation:**

The polynomial \( A(x) = \sum_{i=0}^{n-1} a_i x^i \) is represented by the vector \( a = (a_0, a_1, \ldots, a_{n-1}) \).

The value \( A(x_0) \) can be computed in \( O(n) \) time by

\[
A(x_0) = a_0 + x_0 (a_1 + x_0 (a_2 + \ldots + x_0 (a_{n-2} + x_0 x_{n-1}) \ldots))
\]
Summation

Given two polynomials \( A(x) = \sum_{i=0}^{n-1} a_i x^i \) and \( B(x) = \sum_{i=0}^{n-1} b_i x^i \)

\[
C(x) = A(x) + B(x) = \sum_{i=0}^{n-1} (a_i + b_i) x^i
\]

The degree of \( C(x) \) is the max degree of \( A(x) \) and \( B(x) \).

The sum of two degree \( n \) polynomials, given in a coefficient representation, is computed \( O(n) \) time
Given two polynomials $A(x) = \sum_{i=0}^{n-1} a_i x^i$ and $B(x) = \sum_{i=0}^{n-1} b_i x^i$

$$D(x) = A(x)B(x) = \sum_{i=0}^{2(n-1)} d_i x^i$$

where

$$d_i = \sum_{k=0}^{i} a_k b_{i-k}$$

The degree of $D(x)$ is the sum of the degrees of $A(x)$ and $B(x)$.

The product of two degree $n$ polynomials, given in a coefficient representation, is computed $O(n^2)$ time.
Point value representation

A set of $n$ pairs

$$\left\{(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})\right\}$$

such that

- for all $i \neq j$, $x_i \neq x_j$.
- for every $k$, $y_k = A(x_k)$;
Theorem 1. For any set of $n$ point value pairs $(x_i, y_i)$ there is a unique degree $n$ polynomial $A(x)$ such that $A(x_i) = y_i$ for all pairs.

Proof. We need to solve

$$
\begin{pmatrix}
1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\
1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1}
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
\vdots \\
a_{n-1}
\end{pmatrix}
= 
\begin{pmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{n-1}
\end{pmatrix}
$$

The determinant of the Vandermonde matrix is

$$
\prod_{j<k}(x_k - x_j)
$$

If all the $X_i$'s are distinct, the matrix is nonsingular and the linear system has a unique solution. □
Given two polynomials in (same) point value representation \( \{(x_0, y_1^0), (x_1, y_1^1), \ldots, (x_n, y_1^n)\} \) and \( \{(x_0, y_0^2), (x_1, y_1^2), \ldots, (x_n, y_n^2)\} \)

The sum of two degree \( n \) polynomials in point value representation is computed in \( O(n) \) time:

\[
\{(x_0, y_0^1 + y_0^2), (x_1, y_1^1 + y_1^2), \ldots, (x_{n-1}, y_{n-1}^1 + Y_{n-1}^2)\}
\]

To compute the product of two degree \( n \) polynomials we need an “extended” point value representation of \( 2n \) points.

Given such a representation, the product of two polynomials in point value representation is computed in \( O(n) \).

\[
\{(x_0, y_0^1y_0^2), (x_1, y_1^1y_1^2), \ldots, (x_{2n-2}, y_{2n-1}^1y_{2n-1}^2)\}
\]
Fast Polynomial Multiplication

To compute the product of two degree \( n \) polynomials in coefficient representation:

1. Evaluate the polynomials in \( 2n \) points to create an extended \( 2n \) point value representation of the polynomials.

2. Compute the product of the two polynomials in \( O(n) \) time.

3. Convert the point value representation of the product to coefficient representation.

Using the FFT method (1) and (3) can be done in \( O(n \log n) \) time.
Complex roots of unity

A complex number \( w \) is the \( n \)-th root of unity if

\[
w^n = 1
\]

There are \( n \) complex \( n \)-th roots of unity given by

\[
e^{\frac{2\pi ik}{n}} \quad \text{for} \quad k = 0, \ldots, n - 1
\]

were \( e^{iu} = \cos(u) + i \sin(u) \) and \( i = \sqrt{-1} \).

The principal \( n \)-th root of unity is

\[
w_n = e^{\frac{2\pi i}{n}}
\]

the other roots are powers of \( w_n \).
Operations on the roots of unity

For any $j$ and $k$:

$$w_n^kw_n^j = w_n^{j+k}$$

Since $w_n^n = 1$

$$w_n^kw_n^j = w_n^{j+k} = w_n^{(j+k) \mod n}$$

and

$$w_n^{-k} = w_n^{n-k}$$
The Discrete Fourier Transform (DFT) of a coefficient vector \( a = (a_0, a_1, \ldots, a_{n-1}) \) is a vector \( y = (y_0, y_1, \ldots, y_{n-1}) \) such that

\[
y_k = A(w_n^k) = \sum_{j=0}^{n-1} a_j w_n^{kj}.
\]

\[y = DFT_n(a).\]

Using Fast Fourier Transform (FFT) we can compute \( DFT_n(a) \) in \( O(n \log n) \) steps, instead of \( O(n^2) \).
FFT

Assume that $n$ is a power of 2 (otherwise complete to the nearest power of 2).

Given the polynomial $A(x) = \sum_{j=0}^{n-1} a_j x^j$ we define two polynomials

$$A^{[0]}(x) = a_0 + a_2 x + a_4 x^2 + \ldots + a_{n-2} x^{n/2-1}$$

$$A^{[1]}(x) = a_1 + a_3 x + a_5 x^2 + \ldots + a_{n-1} x^{n/2-1}$$

Then

$$A(x) = A^{[0]}(x^2) + xA^{[1]}(x^2)$$

To compute $DFT_n(a)$ we need to compute the polynomials $A^{[0]}(y)$ and $A^{[1]}(y)$ in the $n$ points

$$(w_n^0)^2, (w_n^1)^2, \ldots, (w_n^{n-1})^2$$
Theorem 2. The set \((w_n^0)^2, (w_n^1)^2, \ldots, (w_n^{n-1})^2\) contains only \(n/2\) distinct points.

Proof. We’ll show that the squares of \(n\) complex \(n\)-th roots of unity are the \(n/2\) complex \(n/2\)-th roots of unity. Assume that \(k \leq \frac{n}{2}\).

\[
(w_n^k)^2 = (e^{2\pi i k/n})^2 = e^{(2\pi i k)/(n/2)} = w_n^{k/2}
\]

\[
(w_n^{k+n/2})^2 = (e^{2\pi i (k+n/2)/n})^2 = e^{2\pi i n/k} = e^{(2\pi i k)/(n/2)} = (w_n^1)^n w_n^k = w_n^{k/2}
\]
Computing the $DFT_n(a)$ is reduced to:

1. Computing two $DFT_{n/2}$

2. combining the results:

Given $y_k^{[0]} = A^{[0]}(w_{n/2}^k) = A^{[0]}((w_n^k)^2)$ and $y_k^{[1]} = A^{[1]}(w_{n/2}^k) = A^{[1]}((w_n^k)^2)$, for $k \leq n/2$

\[
\begin{align*}
y_k &= y_k^{[0]} + w_n^k y_k^{[1]} \\
y_{k+n/2} &= y_k^{[0]} - w_n^k y_k^{[1]} \\
&= y_k^{[0]} + w_n^{k+n/2} y_k^{[1]}
\end{align*}
\]

Since $w_n^{k+n/2} = -w_n^{n/2} w_n^k = -1 w_n^k$
Complexity

\[ T(n) = 2T(n/2) + O(n) = O(n \log n) \]

**Theorem 3.** A point value representation of an \( n \) degree polynomial given in a coefficient representation can be generated in \( O(n \log n) \) time.
Given the DFT $y = (y_0, \ldots, y_{n-1})$ of a degree $n$ polynomial we want to generate the coefficient representation $a = (a_0, \ldots, a_{n-1})$ of the polynomial.

We need to solve

$$
\begin{pmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & w_n & w_n^2 & \cdots & w_n^{n-1} \\
1 & w_n^2 & w_n^4 & \cdots & w_n^{2(n-1)} \\
1 & w_n^3 & w_n^6 & \cdots & w_n^{3(n-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & w_n^{n-1} & w_n^{2(n-1)} & \cdots & w_n^{(n-1)(n-1)}
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
\vdots \\
a_{n-1}
\end{pmatrix}
= 
\begin{pmatrix}
y
\end{pmatrix}
$$

or $y = V_n a$. 
Theorem 4. The \((i, j)\) entry in \(V_n^{-1}\) is \(\frac{w_{ij}^{-1}}{n}\).

Proof. We show that \(V_n^{-1}V_n = I_n\):

The \((j, j')\) entry of \(V_n^{-1}V_n\)

\[
[V_n^{-1}V_n]_{j,j'} = \sum_{k=0}^{n-1} \frac{w_n^{-kj}}{n} (w_n^{kj'})
\]

\[
= \frac{1}{n} \sum_{k=0}^{n-1} w_n^{-k(j-j')}
\]

If \(j = j'\) the summation is 1.
If $j \neq j'$

$$\sum_{k=0}^{n-1} w^{-k(j-j')} = \sum_{k=0}^{n-1} (w^{j-j'})^k$$

$$= \frac{(w_n^{j-j'})^n - 1}{w_n^{j-j'} - 1}$$

$$= \frac{(w_n^n)^{j-j'} - 1}{w_n^{j-j'} - 1}$$

$$= \frac{(1)^{j-j'} - 1}{w_n^{j-j'} - 1}$$

$$= 0$$
Thus, we need to compute

\[ a_i = \frac{1}{n} \sum_{k=0}^{n-1} y_k w_n^{-ki} \]

which can be computed by the FFT algorithm in \( O(n \log n) \).

**Theorem 5.** Given a point value representation of an \( n \) degree polynomial in \( n \)-th roots of unity, the coefficient representation of that polynomial can be computed in \( O(n \log n) \) time.

**Theorem 6.** The product of two \( n \) degree polynomials can be computed in \( O(n \log n) \) time.