Chapter 7. Algebraic Algorithms

Topics covered:

- Strassen's matrix multiplication
- Boolean matrix multiplication
- Discrete Fourier transform
- Fast Fourier transform algor.
- Product of polynomials

7.1. Preliminaries.

Def. $(S, \oplus, \odot, \bar{0}, \bar{1})$ is a <u>semiring</u> if

(1) $(S, \oplus, \bar{0})$ and $(S, \odot, \bar{1})$ are monoids with identity elem. $\bar{0}$ and $\bar{1}$, resp.

(2) $\bar{0}$ is an <u>annihilator</u> : $\bar{0} \odot a = a \odot \bar{0} = \bar{0}$

(3) $\oplus$ is commutative

(4) $\odot$ distributes over $\oplus$

Ex. $(\{0,1\}, \vee, \wedge, 0, 1)$, $(\mathbb{N}, +, *, 0, 1)$ are semirings.

Def. A semiring $(S, \oplus, \odot, \bar{0}, \bar{1})$ is a ring if

(5) $\forall a \in S : \exists b \in S : a \oplus b = b \oplus a = \bar{0}$

i.e., w.r.t. $\oplus$ every $a \in S$ has an inverse denoted by $-a$

Ex. (1) $(\mathbb{Z}, +, *, 0, 1)$ is a ring

(2) $(\mathbb{Z}_n, +, *, 0, 1)$ is a ring for any $n > 1$

(3) Let $\mathbb{R}[x]$ = set of polynomials in x with real coefficients.

$(\mathbb{R}[x], +, *, 0, 1)$ is a ring

(4) $(\mathbb{N}, +, *, 0, 1)$ is not a ring

(5) $(\{0,1\}, \vee, \wedge, 0, 1)$ is not a ring. ⊠

Def. A semiring (ring) S is commutative if $\odot$ is commutative.

A commutative ring $(S, \oplus, \odot, \bar{0}, \bar{1})$ is called a field if

(6) $\forall a \in S - \{\bar{0}\} : \exists b \in S : a \odot b = \bar{1} = b \odot a$

i.e. w.r.t. $\odot$ every elem. $a \in S - \{\bar{0}\}$ has an inverse denoted by a^{-1}.

Ex. (1) $(\mathbb{Q}, +, *, 0, 1)$ is a field

(2) $(\mathbb{R}, +, *, 0, 1)$ is a field

(3) $(\mathbb{Z}_p, +, *, 0, 1)$, $p > 1$ prime, is a field.

Def. Given an algebraic structure $(S, \oplus, \odot, \bar{0}, \bar{1})$, $S^{n\times n}$ denotes the set of all $n \times n$ matrices with entries in S.

Fact. Let $\bar{\mathbb{0}}$ = matrix with $\bar{0}$ entries

$\bar{\mathbb{1}}$ = matrix with $\bar{1}$ on main diagonal and $\bar{0}$ otherwise

(1) $(S, \oplus, \odot, \bar{0}, \bar{1})$ semiring $\Rightarrow (S^{n\times n}, \oplus, \odot, \bar{\mathbb{0}}, \bar{\mathbb{1}})$ semiring

(2) $(S, \oplus, \odot, \bar{0}, \bar{1})$ ring $\Rightarrow (S^{n\times n}, \oplus, \odot, \bar{\mathbb{0}}, \bar{\mathbb{1}})$ ring ⊠

Def. Let S be a field and $A \in S^{n\times n}$ be an $n \times n$ matrix. The <u>inverse</u> of A, denoted A^{-1} if exists, is the $n \times n$ matrix s.t.

$$A \odot A^{-1} = \bar{\mathbb{1}}$$

Note A^{-1} is unique and $A^{-1} \odot A = \bar{\mathbb{1}}$

Remark. Matrix product is non-commutative in general.

Def. Given an $n \times n$ matrix A, the determinant of A, denoted $\det(A)$, is

$$\det(A) = \sum_{\substack{\text{permutation} \\ \pi \text{ over } \{1,\dots,n\}}} (-1)^{\sigma_\pi} \prod_{i=1}^{n} A(i, \pi(i))$$

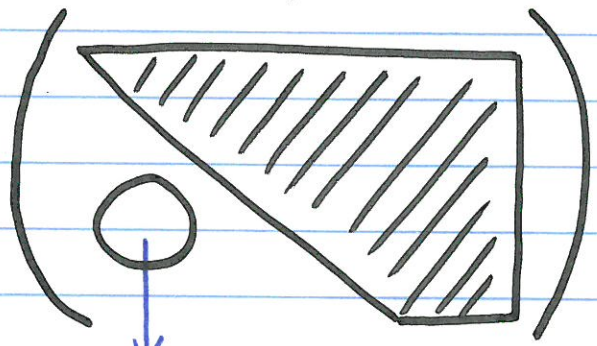
where $\sigma_\pi = \begin{cases} 0 & \text{if } \pi \text{ is even} \\ 1 & \text{if } \pi \text{ is odd} \end{cases}$

(π is even if π is product of even number of interchanges)

Fact. $\det(AB) = \det(A) \cdot \det(B)$

Def. A is nonsingular if $\det(A) \neq \bar{0}$

An $m \times n$ matrix A is upper-triangular if A is of form:

$A =$ 

i.e., $A[i,j] = \bar{0}$

$\forall\, 1 \leq j < i \leq m$

all elements below main diagonal are $\bar{0}$

Similarly we def. lower-triangular matrix

7.2. Strassen's Matrix Multiplication

Let S be a ring and $A, B \in S^{n \times n}$ be $n \times n$ matrices where n is a power of 2.

We can partition A, B into $\frac{n}{2} \times \frac{n}{2}$ sub-matrices

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

where
$$C_{11} = A_{11} B_{11} + A_{12} B_{21}$$
$$C_{12} = A_{11} B_{12} + A_{12} B_{22}$$
$$C_{21} = A_{21} B_{11} + A_{22} B_{21}$$
$$C_{22} = A_{21} B_{12} + A_{22} B_{22}$$

Applying divide-and-conquer this way we obtain for the time $T(n)$ to mult. 2 $n \times n$ matrices the recurrence

$$T(n) = 8 \cdot T\left(\frac{n}{2}\right) + 4\left(\frac{n^2}{4}\right)$$

So, $T(n) = O(n^{\lg 8}) = O(n^3)$

Goal. To reduce the number of $\frac{n}{2} \times \frac{n}{2}$ matrix multiplications from 8 to 7.

Lem. The product of two 2×2 matrices over an arbitrary ring S can be computed with 7 multiplications and 18 add/subtr.

Proof. To compute

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

proceed as follows

Compute

$$m_1 = (a_{12} - a_{22}) \cdot (b_{21} + b_{22})$$
$$m_2 = (a_{11} + a_{22}) \cdot (b_{11} + b_{22})$$
$$m_3 = (a_{11} - a_{21}) \cdot (b_{11} + b_{12})$$
$$m_4 = (a_{11} + a_{12}) \cdot b_{22}$$
$$m_5 = a_{11} \cdot (b_{12} - b_{22})$$
$$m_6 = a_{22} \cdot (b_{21} - b_{11})$$
$$m_7 = (a_{21} + a_{22}) \cdot b_{11}$$

and

$$c_{11} = m_1 + m_2 - m_4 + m_6$$
$$c_{12} = m_4 + m_5$$
$$c_{21} = m_6 + m_7$$
$$c_{22} = m_2 - m_3 + m_5 - m_7$$

⊠

Remark. Subtraction is used in the above lemma, so it does not hold if S is a semiring but not a ring. (e.g. $\mathbb{N}$)

Theorem [Strassen] The product of two $n \times n$ matrices whose entries are from an arbitrary ring can be computed in $O(n^{\lg 7}) = O(n^{2.81})$ arithmetic operations. □

Remark. Based on Strassen's algor, a better upper bound of $O(n^{2.376})$ was derived by Coppersmith & Winograd.

7.3. Boolean Matrix Multiplication

Theorem. The product of two $n \times n$ Boolean matrices can be computed in $O(M(n))$ arithmetic operations, where $M(n)$ is the time to multiply two $n \times n$ matrices over an arbitrary ring.

Proof. View the two $n \times n$ Boolean matrices A, B as matrices over $\mathbb{Z}_{n+1}$ (= ring of

integers modulo $n+1$): $A \cdot B = C$

Observe that

$C[i,j] = 0$ in $\{0,1\} \Rightarrow C[i,j] = 0$ in $\mathbb{Z}_{n+1}$

$C[i,j] = 1$ in $\{0,1\} \Rightarrow C[i,j] \in \mathbb{Z}_{n+1} - \{0\}$.

$\Rightarrow$ Simply perform arith. operations in $\mathbb{Z}_{n+1}$ □

Corollary. Let $m(k)$ denote the number of bitwise operations to multiply two k-bit integers. Then Boolean matrix multiplication can be done in $O(M(n) \cdot m(\log n))$ bitwise operations □

Since $m(k) = O(k \lg k \lg\lg k)$ as shown later, we obtain

Corollary. Boolean matrix multiplication can be done in $O(M(n) \cdot \lg n \cdot \lg\lg n \cdot \lg\lg\lg n)$ bitwise operations. □

7.4. Discrete Fourier Transform.

Def. Let $(S, \oplus, \odot, \bar{0}, \bar{1})$ be a commutative ring. An element $\omega \in S$ is called a **principal n-th root of unity** if

(1) $\omega \neq \bar{1}$

(2) $\omega^n = \bar{1}$

(3) $\sum_{j=0}^{n-1} \omega^{jp} = \bar{0}$ for $p = 1, \ldots, n-1$

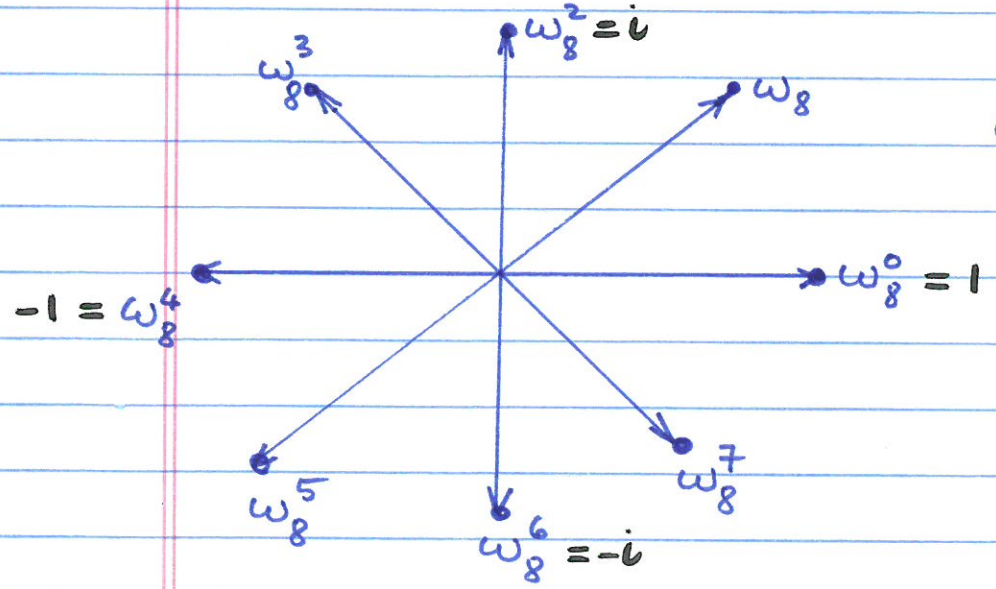
$\omega^0, \omega^1, \ldots, \omega^{n-1}$ are the **n-th roots of unity**

Ex. (Complex roots of unity)

The n complex n-th roots of unity are

$$e^{2\pi i k/n} \qquad k = 0, 1, \ldots, n-1$$

where $e^{iu} = \cos(u) + i\sin(u)$. For $n = 8$:



$$\omega_8 = e^{2\pi i/8}$$

Note: $\omega = e^{2\pi i/n}$ is principal n-th root of unity

The n-th roots are ω^p, $p = 0, \ldots, n-1$.

They form a group under multiplication

(A group is a monoid in which every elem. has an inverse.)

Observe that for $p \neq 0$

$$\sum_{j=0}^{n-1} \omega^{jp} = \sum_{j=0}^{n-1} (\omega^p)^j = \frac{(\omega^p)^n - 1}{\omega^p - 1}$$

$$= \frac{(\omega^n)^p - 1}{\omega^p - 1} = \frac{1^p - 1}{\omega^p - 1} = 0 \quad \square$$

Def. Let $(S, \oplus, \odot, \bar{0}, \bar{1})$ be a commutative ring. Let n denote $\underbrace{\bar{1} \oplus \cdots \oplus \bar{1}}_{n \text{ times}}$.

Assume that

- n has a multiplicative inverse $\frac{1}{n}$
- S has a principal n-th root of unity ω with multiplicative inverse ω^{-1}.

Let $A = (a_{ij})$, $i, j = 0, 1, \ldots, n-1$ be an $n \times n$ matrix s.t. $a_{ij} = \omega^{ij}$ $i, j = 0, \ldots, n-1$

For a vector $a = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} \in S^n$

the vector $F(a) = A \cdot a$ is called the <u>discrete Fourier transform</u> of a

<u>Lem.</u> A^{-1} exists and $A^{-1}[i,j] = \frac{1}{n}\omega^{-ij}$

<u>Proof.</u> We show that $A \cdot A^{-1} = \mathbb{1}$, i.e.,

$$(A \cdot A^{-1})_{ij} = \frac{1}{n}\sum_{k=0}^{n-1} \omega^{ik}\omega^{-kj} = \delta_{ij}$$

where $\delta_{ij} = 1$ if $i=j$, otherwise $\delta_{ij} = 0$, $i \neq j$.

Case $i = j$: obvious.

Case $i \neq j$: Let $q = i - j$. Then

$$\frac{1}{n}\sum_{k=0}^{n-1} \omega^{ik}\omega^{-kj} = \frac{1}{n}\sum_{k=0}^{n-1} \omega^{qk}, \quad -n < q < n, \; q \neq 0$$

If $q > 0$, then $\frac{1}{n}\sum_{k=0}^{n-1} \omega^{qk} = 0$ since ω is principal nth root

If $q < 0$, then

$$\left(\frac{1}{n}\sum_{k=0}^{n-1} \omega^{qk}\right) \cdot \omega^{-q(n-1)} = \frac{1}{n}\sum_{k=0}^{n-1} \omega^{-q(n-k-1)}$$

$$= \frac{1}{n}\sum_{k=0}^{n-1} \omega^{-qk}$$

$= 0$ since ω is principal n-th root $\square$

Def. The vector $F^{-1}(a) = A^{-1} \cdot a$ is the inverse Fourier transform of a.

Polynomial Evaluation/Interpolation and the Fourier transform.

A polynomial $p(x) = \sum_{i=0}^{n-1} a_i x^i$ of degree $n-1$ can be represented in 2 ways:

(a) by coefficient vector $(a_0, a_1, \ldots, a_{n-1})$, or

(b) by its values at n distinct pts $x_0, \ldots, x_{n-1}$.

Polynomial interpolation is the process of finding for $p(x)$ repr. (a) from repr. (b).

Now consider evaluation of $p(x)$ at the points $\omega^0, \omega^1, \ldots, \omega^{n-1}$.

Let $$y_j = p(\omega^j) = \sum_{i=0}^{n-1} a_i \cdot \omega^{ji}$$

$$= \begin{pmatrix} \omega^{j\cdot 0} & \omega^{j\cdot 1} & \cdots & \omega^{j\cdot(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Thus, $$y = \begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix} = A \cdot a = F(a)$$

Conversely, $F^{-1}(y) = a$

DFT EXAMPLE

Example: $n = 4$, 4-th root $= \omega =$

$= e^{2\pi i/4} = i$

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

Let $p(x) = 4 - 3x + 5x^2 - 2x^3$

$\Rightarrow (a_0, a_1, a_2, a_3) = (4, -3, 5, -2)$

$= a$

$$F(a) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \begin{pmatrix} 4 \\ -3 \\ 5 \\ -2 \end{pmatrix}$$

$$= \begin{pmatrix} 4 \\ 4 - 3i - 5 + 2i \\ 4 + 3 + 5 + 2 \\ 4 + 3i - 5 - 2i \end{pmatrix} = \begin{pmatrix} 4 \\ -1 - i \\ 14 \\ -1 + i \end{pmatrix}$$

i.e., $p(1) = 4$

$p(\omega) = -1 - i$

$p(\omega^2) = 14$

$p(\omega^3) = -1 + i$

□

Thus, polynomial eval $\leftrightarrow$ Fourier transf
polynomial interp. $\leftrightarrow$ Inverse Fourier trans.

Another Application

Computing the convolution of 2 vectors:

Let $\mathbf{a} = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix}$ be 2 vectors.

The **convolution** $\mathbf{a} \circledast \mathbf{b}$ is $\mathbf{c} = \begin{pmatrix} c_0 \\ \vdots \\ c_{2n-1} \end{pmatrix}$

where $c_i = \sum_{j=0}^{n-1} a_j b_{i-j}$ $\quad i = 0, \ldots, 2n-1$

($a_k = b_k = 0$ if $k < 0$ or $k \geq n$)

Let $\mathbf{a}, \mathbf{b}$ be the coefficient represent. of polynomials $p(x), q(x)$. Then $\mathbf{a} \circledast \mathbf{b}$ is the coefficient representation of the product $p(x)\,q(x) = \sum_{i=0}^{2n-2} \left(\sum_{j=0}^{i} a_j b_{i-j} \right) x^i$.

Question. How fast can we compute $\mathbf{a} \circledast \mathbf{b}$?

Remark. Straightforward method yields $\Theta(n^2)$ algor. Using FFT we can achieve $O(n \lg n)$.

Theorem. (Convolution Theorem)

Let $a = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$, $b = \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in S^{2n}$.

Then $a \circledast b = F^{-1}(F(a) \cdot F(b))$

($\cdot$: componentwise product)

7.5. Fast Fourier Transform Algor. (FFT)

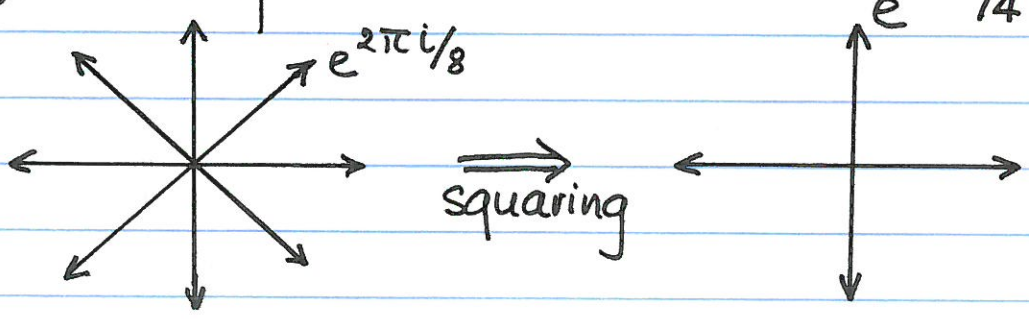
Fact. Let ω_n be principal complex n-th root of unity.

(1) $\omega_n^{n/2} = \omega_2 = -1$ for all even n.

(2) If $n > 0$ is even, then the squares of the n complex n-th roots of unity are the $\frac{n}{2}$ complex $(\frac{n}{2})$-th roots of unity

Pf. (1) : Trivial.

(2) Example : $n = 8$



Observe that for any $d>0$, $n, k \geq 0$:

$$\omega_{dn}^{dk} = \left(e^{2\pi i/dn}\right)^{dk}$$

$$= \left(e^{2\pi i/n}\right)^{k}$$

$$= \omega_n^k \quad \Rightarrow (\omega_{2n}^k)^2 = \omega_n^k .$$

Therefore, for even n:

$$(\omega_n^k)^2 = \omega_{n/2}^k \qquad \square$$

Theorem. For $\mathbf{a} = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix} \in \mathbb{C}^n$, $F(\mathbf{a})$ can be computed in $O(n \lg n)$ operations.

Proof. Assume n is a power of 2.

Let $p(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} = \sum_{i=0}^{n-1} a_i x^i$

Define:

$$p_0(y) = a_0 + a_2 y + \cdots + a_{n-2} y^{\frac{n}{2}-1}$$

$$p_1(y) = a_1 + a_3 y + \cdots + a_{n-1} y^{\frac{n}{2}-1}$$

Then

$$p(x) = p_0(x^2) + x \cdot p_1(x^2) \qquad (\S)$$

That is we "divide" p into 2 "sub"-poly. of degree $\frac{n}{2} - 1$ each.

Idea: Evaluating $p(x)$ at $\omega_n^0, \ldots, \omega_n^{n-1}$ is reduced to eval. p_0, p_1 at $(\omega_n^0)^2, \ldots, (\omega_n^{n-1})^2$

and $(\omega_n^0)^2, \ldots, (\omega_n^{n-1})^2$ are the squares of $\omega_n^0, \ldots, \omega_n^{n-1}$.

Thus, to eval. $p(x)$ at $\omega_n^0, \ldots, \omega_n^{n-1}$ we:

(1) Eval. $p_0(y), p_1(y)$ at $(\omega_n^0)^2, \ldots, (\omega_n^{n-1})^2$

(2) Combine results in (1) using (§) to obtain value for $p(x)$.

FAST-Fourier-Transform (a)

if $n = 1$ then return a

$\omega_n = e^{2\pi i/n}$; $\omega = 1$

$$a^{[0]} = \begin{pmatrix} a_0 \\ a_2 \\ \vdots \\ a_{n-2} \end{pmatrix}; \quad a^{[1]} = \begin{pmatrix} a_1 \\ a_3 \\ \vdots \\ a_{n-1} \end{pmatrix};$$

recursively compute

$y^{[0]} = \text{FFT}(a^{[0]})$

$y^{[1]} = \text{FFT}(a^{[1]})$

for $k = 0$ to $n/2 - 1$ do

$y_k = y_k^{[0]} + \omega\, y_k^{[1]}$

$y_{k+n/2} = y_k^{[0]} - \omega\, y_k^{[1]}$

$\omega = \omega \cdot \omega_n$

return y

The recurrence for the running time is:

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$

which has solution $\Theta(n \lg n)$ ⊠

Remark. Using a similar divide-and-conquer strategy, the (general) discrete Fourier transform can be computed in $O(n \lg n)$ time.

Corollary. The product of two n-th degree polynomials can be computed in $O(n \lg n)$ arithm. operations.

Example.

$$p(x) = 4 - 3x + 5x^2 - 2x^3$$

$$\Rightarrow a = (a_0, a_1, a_2, a_3) = (4, -3, 5, -2)$$

$$p_0(y) = 4 + 5y \; ; \; p_1(y) = -3 - 2y$$

$$\omega = i \implies \omega^2 = -1$$

$$\omega^0 = 1 \; , \; \omega^1 = i \; , \; \omega^2 = -1 \; , \; \omega^3 = -i$$

$\omega^0 = 1$: $p(\omega^0) = p_0(1) + \omega\, p_1(1)$

$= (4 + 5 \cdot 1) + 1 \cdot (-3 - 2 \cdot 1)$

$= 4$

$\omega = i$: $p(\omega) = p_0(\omega^2) + \omega\, p_1(\omega^2)$

$= p_0(-1) + i\, p_1(-1)$

$= (4 - 5) + i \cdot (-3 + 2)$

$= -1 - i$

$\omega^2 = -1$: $p(\omega^2) = p_0(1) + (-1)\, p_1(1)$

$= (4 + 5) + (-1)(-3 - 2)$

$= 14$

$\omega^3 = -i$: $p(\omega^3) = p_0(-1) + (-i)\, p_1(-1)$

$= (4 - 5) + (-i)(-3 + 2)$

$= -1 + i$

FFT (example).

$n = 4 \qquad a = (4, -3, 5, -2)$

$a_0^{[0]} = (4, 5) \qquad a_1^{[1]} = (-3, -2)$

$y^{[0]} = FFT(a^{[0]}) = \begin{pmatrix} 9 \\ -1 \end{pmatrix}$

$y^{[1]} = FFT(a^{[1]}) = \begin{pmatrix} -5 \\ -1 \end{pmatrix}$

$y_0 = y_0^{[0]} + (1)\, y_0^{[1]}$

$9 + (1)(-5) = 4$

$y_1 = y_1^{[0]} + (i).y_1^{[1]} = -1 + i(-1) = -1 - i$

$y_2 = y_0^{[0]} + (-1)\, y_0^{[1]} = 9 + (-1)(-5) = 14$

$y_3 = y_1^{[0]} + (-i)\, y_1^{[1]} = -1 + (-i)(-1)$

$= -1 + i$