$\text{EVALUATE}(\langle p_0 \ldots p_d \rangle, \{\alpha_1 \ldots \alpha_n\})$:

GOAL: $(p(\alpha_1) \ldots p(\alpha_n))$

1) SPLIT

$$P_{odd}(z) = p_1 + p_3 z + p_5 z^2 + \ldots$$

$$P_{even}(z) = p_0 + p_2 z + p_4 z^2 \ldots$$

$$\boxed{P(x) = P_{even}(x^2) + x \cdot P_{odd}(x^2)}$$

2) $\text{Evaluate}(P_{odd}, \{\alpha_1^2 \ldots \alpha_n^2\})$

$\text{Evaluate}(P_{even}, \{\alpha_1^2 \ldots \alpha_n^2\})$

3) $\forall\, i = 1 \ldots n$

$$p(\alpha_i) = P_{even}(\alpha_i^2) + \alpha_i P_{odd}(\alpha_i^2)$$

$$p(x) = 0 + 1x + 2x^2 + 3x^3 + 4x^4$$

$$\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\} = \{5, 6, 7, 8\}$$

$p(5)$ $p(6)$ $p(7)$ $p(8)$

$P_{even}(5^2)$ $P_{even}(6^2)$ $P_{even}(7^2)$ $P_{even}(8^2)$

$P_{odd}(5^2)$ $P_{odd}(6^2)$ $P_{odd}(7^2)$ $P_{odd}(8^2)$

$$p(x) = 0 + 1x + 2x^2 + 3x^3 + 4x^4$$

$$P_{odd} = 1 + 3z$$

$$P_{even} = 0 + 2z + 4z^2$$

Evaluate $p(x)$ at $\{\alpha_1, \ldots, \alpha_n\}$

$\deg(p) = d$

$\#$ of points $= n$

Evaluate $P_{odd}(z)$ at $\{\alpha_1^2, \ldots, \alpha_n^2\}$

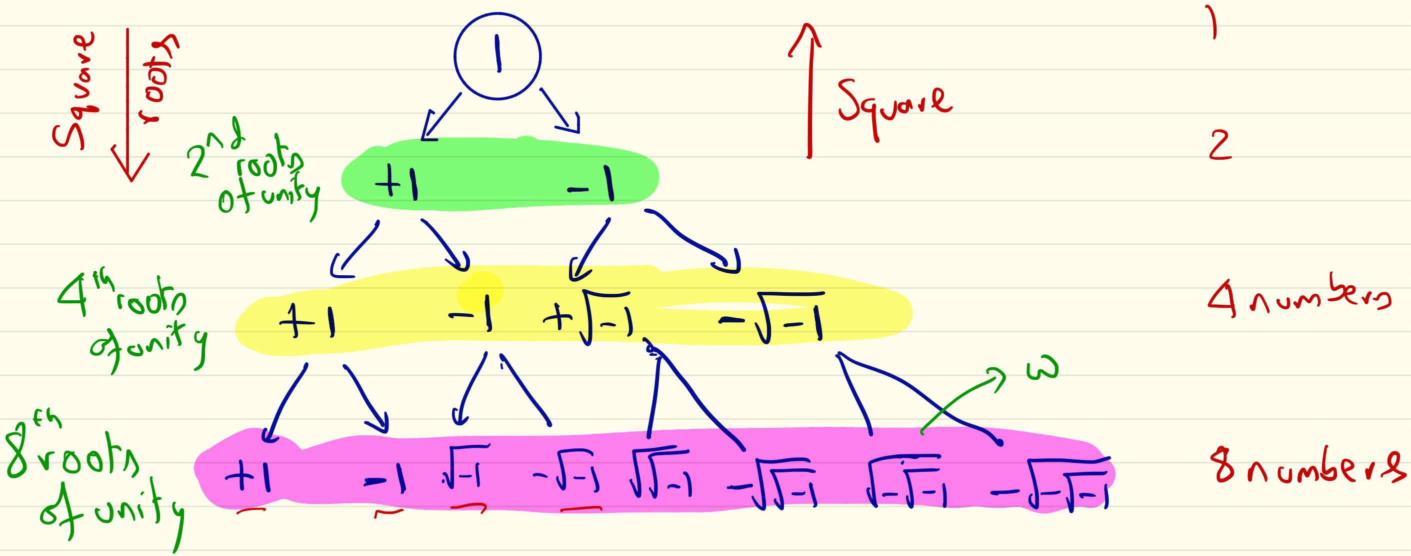
$\deg(p) = d/2$

$\#$ points $= n$

Evaluate $P_{even}(z)$ at $\{\alpha_1^2, \ldots, \alpha_n^2\}$,

$\deg = d/2$

$\#$ points $= n$



1) Every number a has $(+\sqrt{a}, -\sqrt{a})$

2) Squaring numbers in layer i $\Longrightarrow$ numbers in layer $i-1$

Evaluate $p(x)$
at n^{th} roots of unity

Evaluate $P_{odd}(z)$
at Squares of n^{th} roots of unity $=$ $n/2^{th}$ roots of unity

Evaluate $P_{even}(z)$
at $n/2^{th}$ roots of unity

Set $\{\alpha_1 \ldots \alpha_n\} = n^{th}$ roots of unity

n^{th} roots of unity $=$ { solutions to $x^n = 1$ }

$= \{ \omega_0, \omega_1, \ldots \omega_{n-1} \}$

Discrete Fourier Transform

INPUT: $\langle p_0, p_1 \ldots p_{n-1} \rangle$ coefficients of a poly p of deg $n-1$

OUTPUT: $\langle p(\omega_0), p(\omega_1) \ldots p(\omega_{n-1}) \rangle$

evaluations at n^{th} roots of unity.

Thm: $\exists$ an $\Theta(n \log n)$ time alg for Fourier transform

$\langle p_0 \dots p_{n-1} \rangle : \; p(x) = \sum p_i x^i$ find $p(\omega_0) \dots p(\omega_{n-1})$

$$\begin{bmatrix} 1 & \omega_0 & \omega_0^2 & \cdots & \omega_0^{n-1} \\ 1 & \omega_1 & \omega_1^2 & & \omega_1^{n-1} \\ & & \boxed{V} & & \\ \vdots & & & & \\ 1 & \omega_{n-1} & \omega_{n-1}^2 & & \omega_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} p_0 \\ \vdots \\ p_{n-1} \end{bmatrix} = \begin{bmatrix} p(\omega_0) \\ p(\omega_i) \\ p(\omega_{n-1}) \end{bmatrix}$$

$$p(\omega_i) = p_0 \cdot 1 + p_1 \cdot \omega_i^1 + p_2 \cdot \omega_i^2 + \dots + p_{n-1}\omega_i^{n-1}$$

Inverse Fourier Transform

Given $p(\omega_0) \ldots p(\omega_{n-1})$

Compute: $\langle p_0 \ldots p_{n-1} \rangle$

Inverse Fourier transform is given by

$$\begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{bmatrix} = V^{-1} \cdot \begin{bmatrix} p(\omega_0) \\ \vdots \\ p(\omega_{n-1}) \end{bmatrix}$$

Thm:

$$V^{-1} = \frac{1}{n}\left[\, V \text{ with } \omega_i \text{ replaced by } 1/\omega_i \,\right]$$

$$= \frac{1}{n} V^{*} \quad \leftarrow \text{complex conjugate}$$

Corollary: Replacing ω_i by $1/\omega_i$ in FFT yields an algorithm for Inverse Fourier Tromform