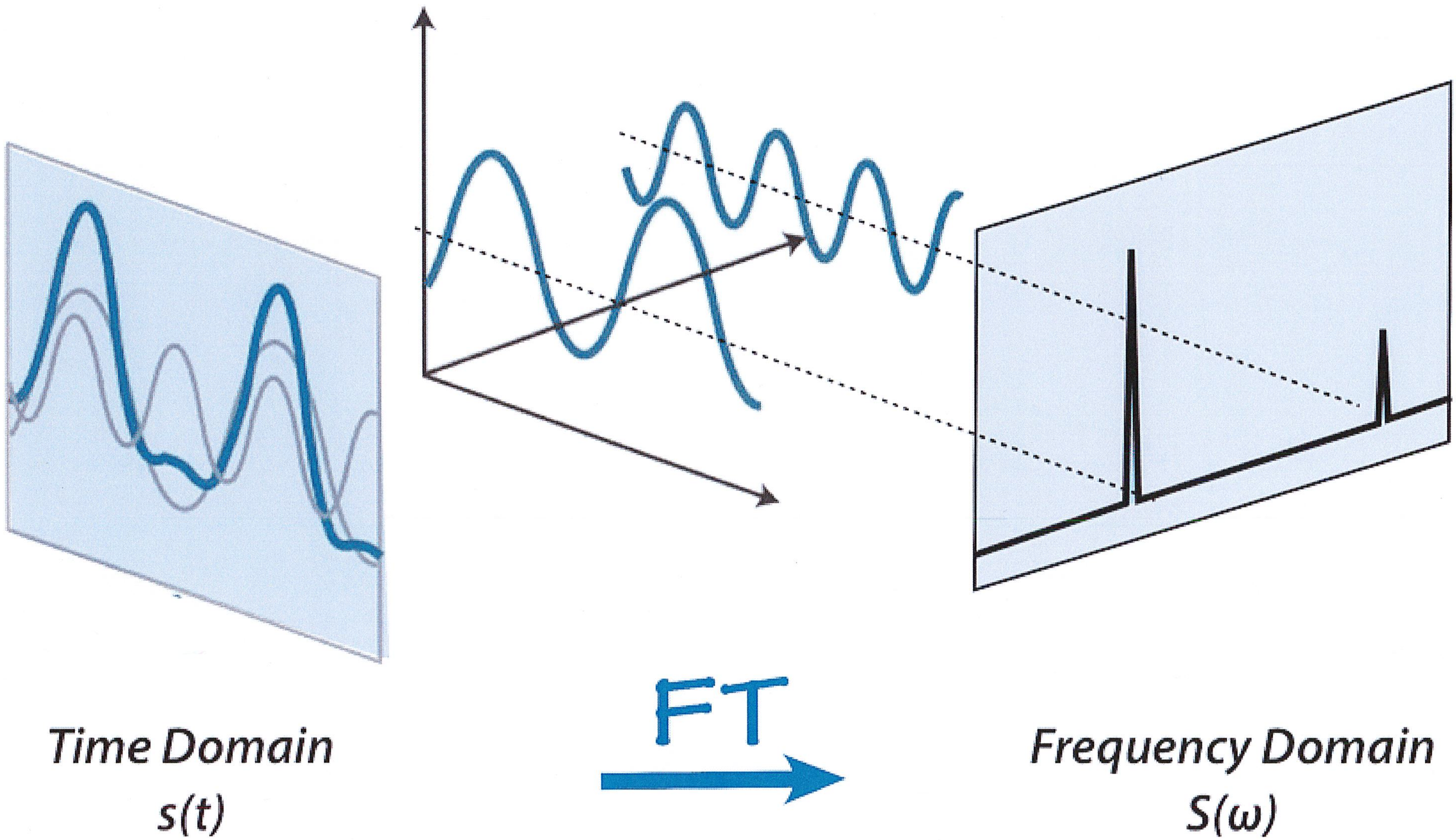
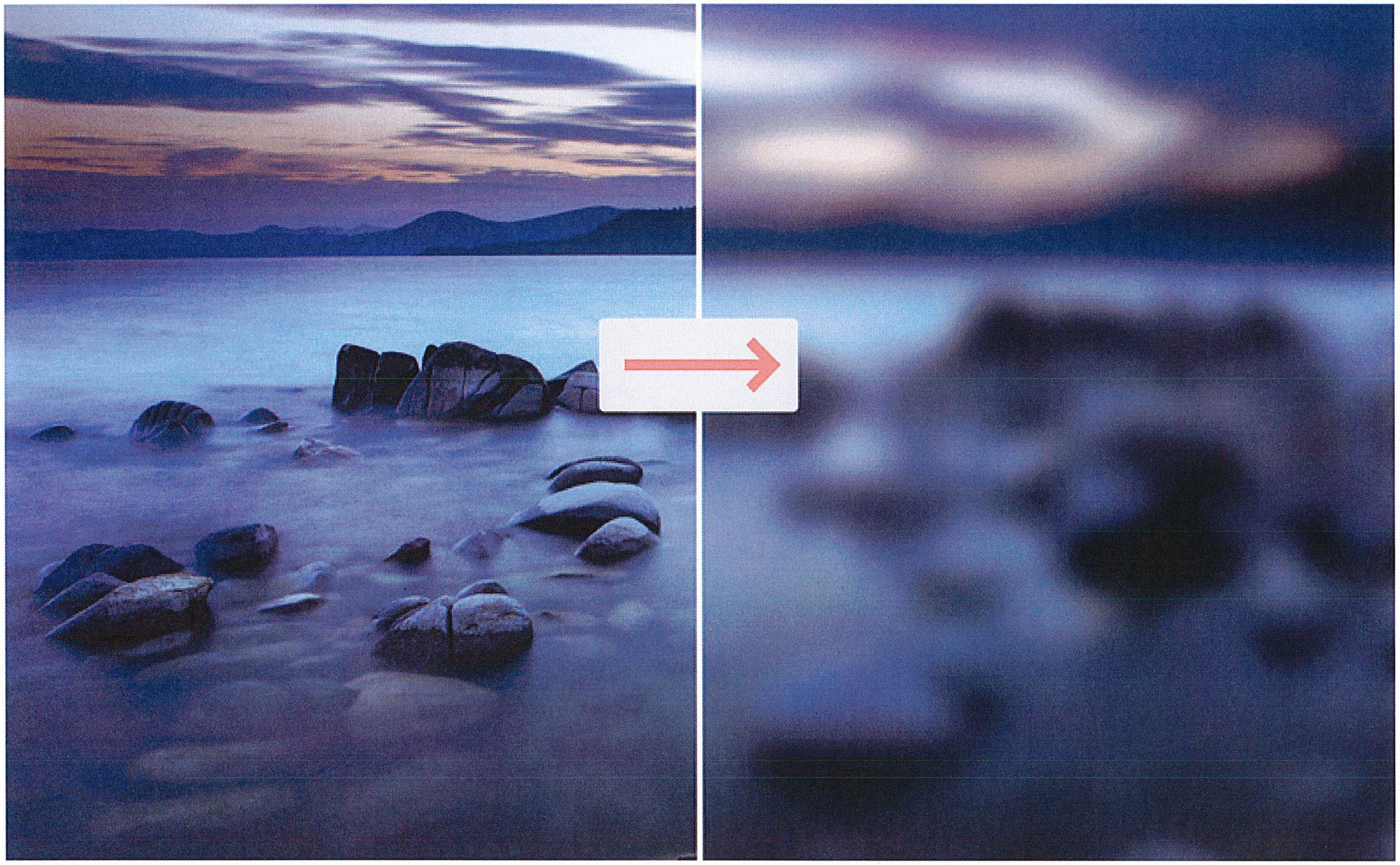
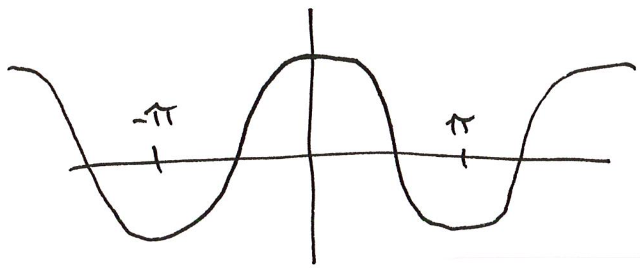

Time Domain
s(t)
FT
Frequency Domain
S(ω)



Fourier Transform (FT)

- refers to the transformation of a continuous signal



cosine in Time domain

cosine in frequency domain

Discrete Fourier Transform (DFT)

- refers to the transformation of a discrete signal

Fast Fourier Transform (FFT)

any $\Theta(n \log n)$ algorithm for computing DFT

Polynomials

A polynomial $A(x)$ can be written as

$$A(x) = a_0 + a_1 x^1 + a_2 + x^2 + \ldots + a_{n-1} x^{n-1}$$

$$= \sum_{k=0}^{n-1} a_k x^k$$

$$= \langle a_0, a_1, a_2 \ldots a_{n-1} \rangle$$

coefficient vector

Operations

1. Evaluation: Given a polynomial $A(x)$ and a number x_0 compute $A(x_0)$

2. Addition: Given polynomials $A(x)$ and $B(x)$ compute $C(x)$ $\quad C(x) = A(x) + B(x) \;\; \forall (x)$

3. Multiplication: $\quad C(x) = A(x) \cdot B(x) \;\; \forall x$

Polynomial Evaluation

$$A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_{n-1} x^{n-1}$$

$\Theta(n^2)$ if we compute each x^k naively

Horner's Rule

$$A(x) = a_0 + x(a_1 + x(a_2 + x(\quad \dots \quad x(a_{n-1})\dots)$$

$\Theta(n)$ time

Polynomial Addition

$$C(x) = A(x) + B(x) \quad \forall x$$

$$C(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_{n-1} x^{n-1}$$

$$\forall_k \quad c_k = a_k + b_k \qquad \Theta(n) \text{ time}$$

Polynomial Multiplication

$$C(x) = A(x) \cdot B(x) \quad \forall x$$

$$A(x) = \langle a_0, a_1, a_2, \ldots a_{n-1} \rangle$$

$$B(x) = \langle b_0, b_1, b_2, \ldots b_{n-1} \rangle$$

$$C(x) = \langle a_0 b_0, \; a_1 b_0 + a_0 b_1, \; a_2 b_0 + a_1 b_1 + a_0 b_2, \ldots \rangle$$

$$c_k = \sum_{j=0}^{k} a_j b_{k-j} \quad \text{for} \quad 0 \le k \le 2(n-1)$$

$$\Theta(n^2)$$

Roots and Scale

For some polynomial $A(x)$, a root of that polynomial is a solution to $A(x) = 0$

Fundamental Theorem of Algebra

- any degree n polynomial has exactly n roots

$$A(x) = (x - r_0)(x - r_1)(x - r_2) \dots (x - r_{n-1})\, c$$

Evaluation $\Theta(n)$

Multiplication $O(n)$ concatenate roots

Addition hard / impossible

Point Value Representation (Samples)

We uniquely represent a polynomial as a set of samples

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

with $A(x_k) = y_k$ $\forall$ All x_k must be distinct

- Addition/multiplication - add/multiply the y_k if the x_k match $O(n)$

- Evaluation - requires interpolation $\Theta(n^2)$

	Coefficient vector	roots	Samples
Evaluation	$O(n)$	$O(n)$	$O(n^2)$
Addition	$O(n)$	∞	$O(n)$
Multiplication	$O(n^2)$	$O(n)$	$O(n)$

$O(n \log n)$

Convert Coefficients → Samples

Vandermonde Matrix

$$V \cdot A = Y$$

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & & & & \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Converting Samples → Coefficients

$$V \cdot ? = Y$$

Gaussian Elimination

$$O(n^3)$$

Matrix Vector Product

$$V^{-1} \cdot Y = A$$

Divide and Conquer

V. A

Divide into even and odd coefficients

$$A_{even}(x) = \sum_{k=0}^{\lceil \frac{n}{2}-1 \rceil} a_{2k} x^k = \langle a_0, a_2, a_4, \dots \rangle$$

$$A_{odd}(x) = \sum_{k=0}^{\lfloor n/2 - 1 \rfloor} a_{2k+1} x^k = \langle a_1, a_3, a_5, \dots \rangle$$

Conquer

calculate $A_{even}(y)$ and $A_{odd}(y)$ for $\forall y \in X^2$

Combine

$$A(x) = A_{even}(x^2) + x A_{odd}(x^2)$$

Divide and Conquer

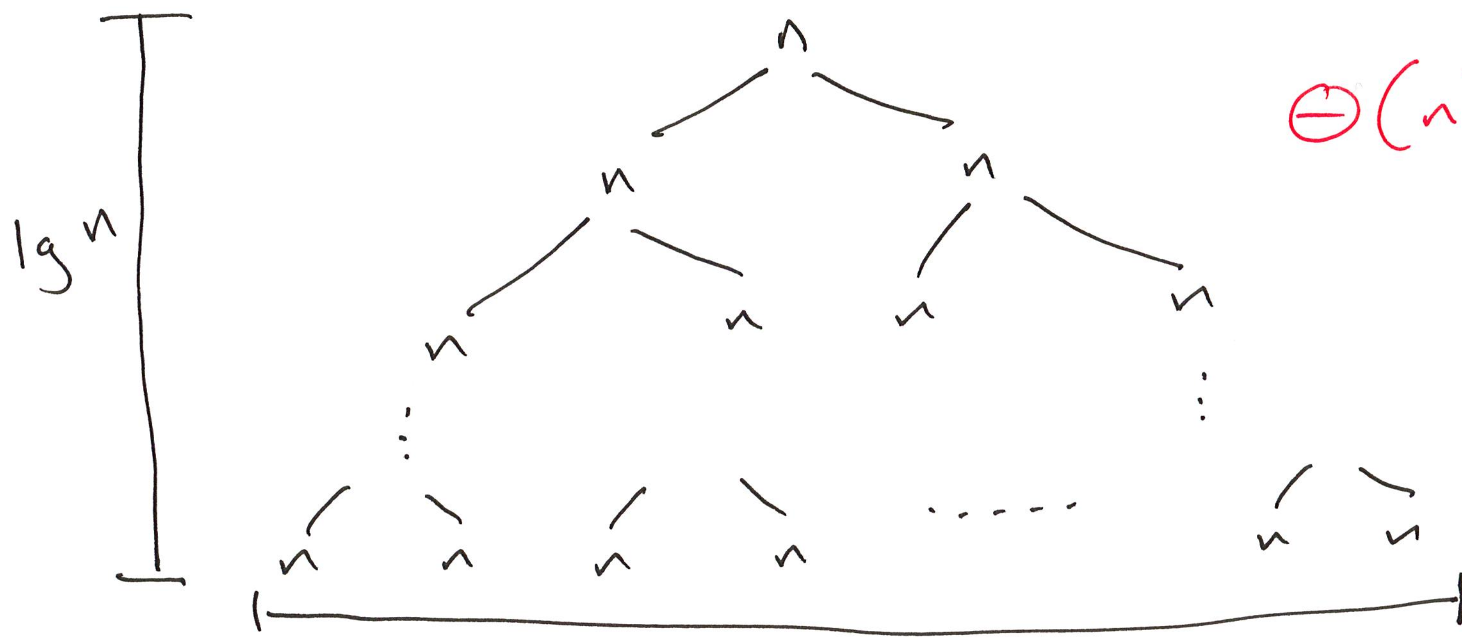
Recurrence relation

$$T(n, |X|) = 2T\left(\frac{n}{2}, |X|\right) + O(n + |X|)$$

number of coefficients

number of samples

initially $|X| = n$



$\Theta(n^2)$

$2^{\lg n} = n$

Collapsing Inputs

- In order to get our desired complexity the number of inputs must also decrease at each step of the recursion.
- We want a set of inputs such that when we square ~~them~~ all of them the total number of unique values is cut in half.

$$e^{\pi i} = -1$$

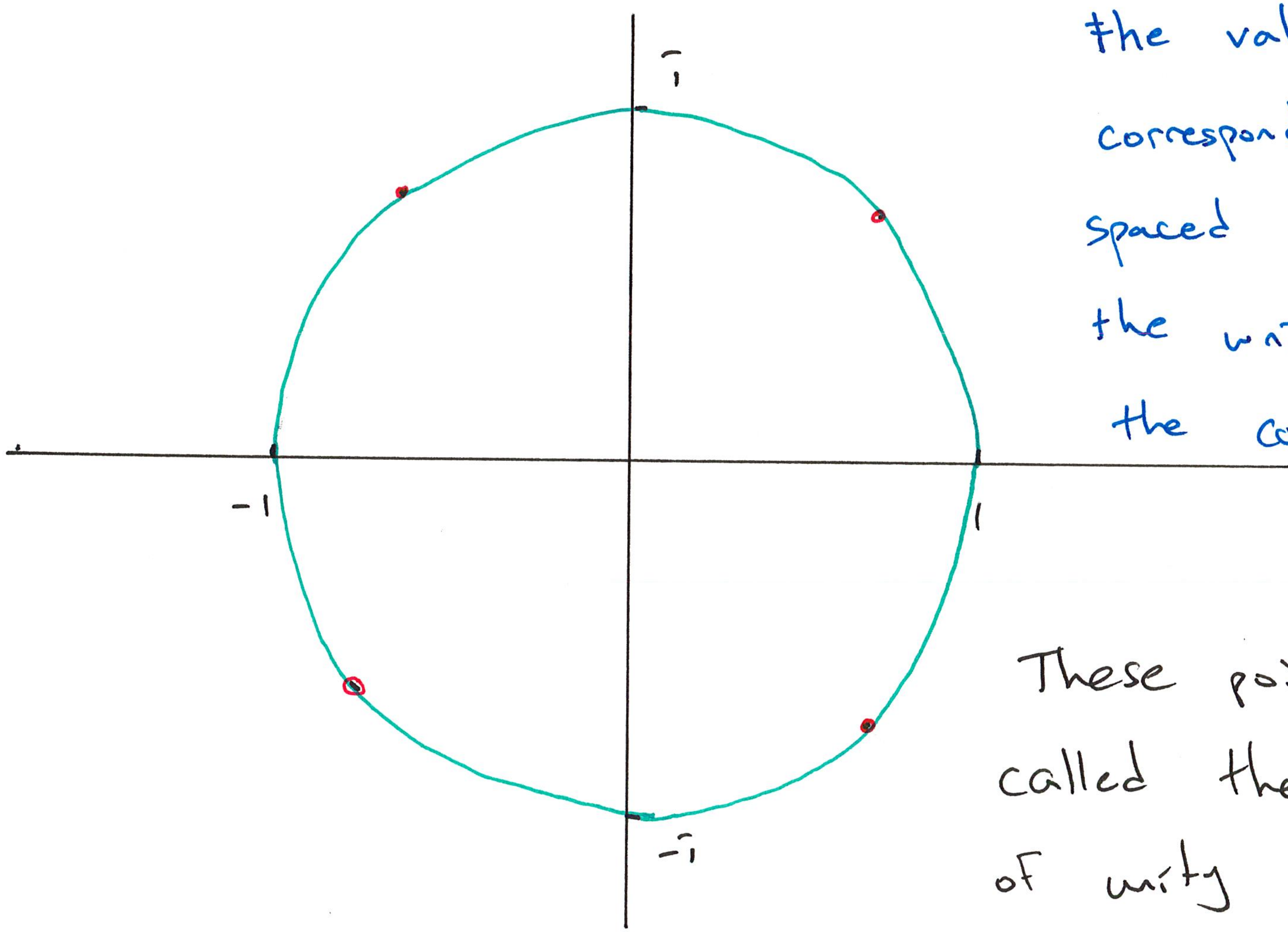
- Consider $\sqrt{}$

  0 - $\{1\}$

  1 - $\{1, -1\}$

  2 - $\{1, -1, i, -i\}$

  3 - $\{1, -1, i, -i, \frac{\sqrt{2}}{2}(1+i), -\frac{\sqrt{2}}{2}(1+i), \pm \frac{\sqrt{2}}{2}(-1+i)\}$



the values we saw correspond to evenly spaced points around the unit circle on the complex plane

These points are called the n^{th} roots of unity

n xs such that $x^n = 1$

- Euler's formula for a general point on a circle

$$(\cos\theta, \sin\theta) = \cos\theta + i\sin\theta = e^{i\theta}$$

- n evenly spaced angles gives us

$$\theta = 0, \frac{1}{n}\tau, \frac{2}{n}\tau, \ldots \frac{n-1}{n}\tau \qquad \tau = 2\pi$$

- The n^{th} roots of unity where $n = 2^{\ell}$ forms a collapsing set

$$(e^{i\theta})^2 = e^{i(2\theta)} = e^{i(2\theta \bmod \tau)}$$

Therefore the even n^{th} roots of unity are equivalent to the $\frac{n}{2}^{th}$ roots of unity

- By using the n^{th} roots of unity for some $n = 2^{\ell}$ the same divide and conquer algorithm now runs in $\Theta(n \log n)$