FAST FOURIER TRANSFORM (FFT)

Given two vectors $a = (a_0, a_1, a_2, \dots, a_{n-1})$ & $b = (b_0, b_1, b_2, \dots, b_{n-1})$.

- $$a + b = (a_0 + b_0,\ a_1 + b_1,\ a_2 + b_2,\ \dots,\ \dots,\ a_{n-1} + b_{n-1})$$

- $$a \cdot b = (a_0 \cdot b_0,\ a_1 \cdot b_1,\ a_2 \cdot b_2,\ \dots,\ \dots,\ a_{n-1} \cdot b_{n-1})$$

- $$a * b = (a_0 b_0,\ a_0 b_1 + \overset{a_1 b_0}{\cancel{b_1 a_0}},\ a_0 b_2 + a_1 \cdot b_1 + a_2 b_0,\ \dots,\ \dots,\ a_{n-1} \cdot b_{n-1})$$

$a * b$ is called the Convolution of a and b

$$
\begin{array}{ccccc}
a_0 b_0 & a_0 b_1 & a_0 b_2 & \cdots & a_0 b_{n-1} \\
a_1 b_0 & a_1 b_1 & a_1 b_2 & \cdots & a_1 b_{n-1} \\
a_2 b_0 & a_2 b_1 & a_2 b_2 & \cdots & a_2 b_{n-1} \\
\vdots & \vdots & \vdots & & \vdots \\
a_{n-1} b_0 & a_{n-1} b_1 & a_{n-1} b_2 & \cdots & a_{n-1} \cdot b_{n-1}
\end{array}
$$

$$
a * b = \left(a_0 b_0 \, , \; a_0 b_1 + \overset{a_1 b_0}{\cancel{b_1 a_0}} \, , \; a_0 b_2 + a_1 b_1 + a_2 b_0 \, , \; \ldots \, , \; a_{n-1} \cdot b_{n-1} \right)
$$

If length of a and b is n each, then length of $a * b$ is ______

Motivation

$$A(x) = a_0 + a_1 \cdot x + a_2 x^2 + \dots + a_{m-1} \cdot x^{m-1}$$

$$B(x) = b_0 + b_1 \cdot x + b_2 \cdot x^2 + \dots + b_{n-1} \cdot x^{n-1}$$

Cofficient vectors of $A(x)$ & $B(x)$:

$a = (a_0, a_1, a_2, \dots, a_{m-1})$

$b = (b_0, b_1, b_2, \dots, b_{n-1})$

If $C(x) = A(x) \cdot B(x)$, then

cofficient vector of $C(x)$ is

$$c = a * b$$

<u>Complexity</u> : c can be Computed

in $O(m \cdot n)$ time.

If $m = n$, then $O(n^2)$ time

Using Divide & Conquer we can find $c = a * b$ in $O(n \cdot \log n)$ time.

<u>Algo</u>:

- Choose $2n$ values and evaluate $A(x)$ and $B(x)$ on each of the chosen values

- If $x_1, x_2, x_3, \dots, x_{2n}$ are the values, then
$$C(x_j) = A(x_j) \cdot B(x_j)$$
for each j.

- C can be recovered from its values on $x_1, x_2, \dots, x_{2n}$

This is called POLYNOMIAL INTERPOLATION.

Complexity

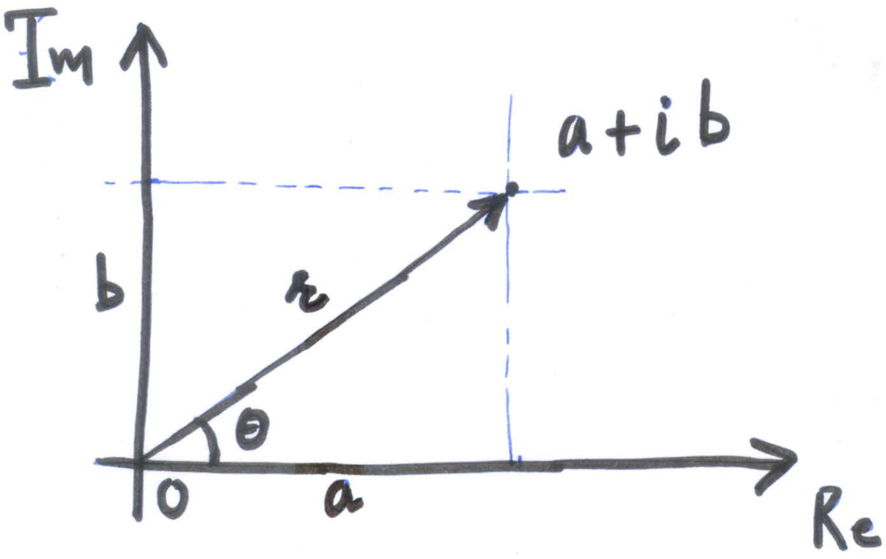
- Evaluation of polynomial $A(x)$ or $B(x)$ on single value takes $\Omega(n)$ time. So total time is still $O(n^2)$.

- But we promised $O(n \cdot \log n)$ time

- Find the set of $2n$ values $x_1, x_2, \ldots, x_{2n}$ that are intimately related in some way. This will ensure that work done in evaluation of $A(x)$ or $B(x)$ on different values overlap.

The 2n values are not any arbitrary numbers. They are rather $\boxed{\text{Complex roots of Unity}}$

Complex Nos (Recall)



$a+ib \quad , \quad i^2 = -1$

$r(\cos\theta + i\sin\theta) \quad , \quad r = \sqrt{a^2+b^2}$

$r \cdot e^{i\theta}$

(ARGAND DIAGRAM)

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

$$e^{2\pi i} = 1$$

Kth roots of Unity

All Complex nos x that satisfy

$$x^k = 1$$

Each $\omega_{j,k} = e^{2\pi j i/k}$

For $j = 0, 1, 2, \ldots, k-1$

is a Complex kth root of Unity

Pictorically : Set of k equally spaced pts lying on the Unit Circle in a Complex plane.

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

$$A_{even} = a_0 + a_2 x + a_4 x^2 + \ldots + a_{n-2}\, x^{\frac{n-2}{2}}$$

~~Aodd~~

$$A_{odd} = a_1 + a_3 x + a_5 x^2 + \ldots + a_{n-1}\, x^{\frac{n-2}{2}}$$

Now we have : $A(x) = A_{even}(x^2) + x \cdot A_{odd}(x^2)$

<u>Recall</u> : We want $2n$ points over which $A(x)$ can be evaluated.

<u>Key Point</u> : We choose $2n$ th roots of Unity

$$\omega_{j,2n} = e^{2\pi j i/2n}$$

$$\omega_{j,2n}^2 = e^{2\pi j i/n} = \omega_{j,n}$$

[nth root of Unity]

$$A(\omega_{j,2n}) = A_{even}(\omega_{j,2n}^2) + \omega_{j,2n} \cdot A_{odd}(\omega_{j,2n}^2)$$

If $T(n)$ is the time taken to evaluate $A(x)$ on each of the $2n$ th roots of Unity, then $T\left(\frac{n}{2}\right)$ is the time to evaluate $A_{even}(x)$ or $A_{odd}(x)$ on each of the nth root of Unity

We have $T(n) \leq 2T\left(\frac{n}{2}\right) + O(n)$

$\therefore T(n)$ is $O(n \cdot \log n)$

Similarly we can evaluate $B(x)$ on each of the $2n$ th roots of Unity in time $O(n \cdot \log n)$.

Hence we can obtain the value of $C(x)$ on ~~each of~~ $2n$ different points in time $O(n \cdot \log n) + O(n \cdot \log n) + O(n)$
$= O(n \cdot \log n)$.