DISCRETE FOURIER TRANSFORM

Recall: Vector Convolution

Let $\begin{cases} u = (u_0, u_1, \dots, u_{m-1}) \\ v = (v_0, v_1, \dots, v_{n-1}) \end{cases}$ be vectors of length m & n

Vector convolution is an operation defined as:

$$\text{resulting vector} = (u * v)$$

$$= \left[(u*v)_0 , (u*v)_1 , \dots (u*v)_{m+n-2} \right]$$

component k^{th}

$$(u * v)_k = \sum_{s+t=k} u_s v_t \quad \text{or} \quad \sum_{i=0}^{k} u_i v_{k-i}$$

Example:

Given $u = (1, 2, 3)$ and $v = (4, 5)$. Then $(u * v)$:

$u_0 v_0$ $u_0 v_1$

$$(u * v) = (4.1, \ 5.1 + 4.2, \ 5.2 + 4.3, \ 5.3)$$

$$= (4, \ 13, \ 22, \ 15)$$

Visualization:

Using the same example as above, there are 2 ways to visualize the vector convolution operation:

- Sliding window:
  - Flip v: $v_{180°} = (5, 4)$
  - Slide $v_{180°}$ from left to right on u:

$u = (1, 2, 3)$

$v_{180°} = (5, 4)$ ----------> sliding

$(u * v) = (4, \dots, \dots, \dots)$

- Matrix:

Sample u and v as 2D matrix, the diagonals are the resulting components

v \ u	1	2	3
4	4	8	12
5	5	10	15

$(u * v) = (4, 13, 22, 15)$

Connection to Multiplying Polynomials

Consider previous example: $\begin{cases} u = (1, 2, 3) \\ v = (4, 5) \end{cases}$

represents the coefficients of these polynomials: $\begin{cases} A(x) = 1 + 2x + 3x^2 \\ B(x) = 4 + 5x \end{cases}$

Let $C(x)$ be the product of $A(x)$ and $B(x)$, then:

$$C(x) = A(x) \cdot B(x)$$
$$= 4 + 13x + 22x^2 + 15x^3$$

$\Rightarrow$ Notice the coefficients of $C(x)$ is similar to the result of vector convolution of $u * v$

Vector Convolution - Multiplying Polynomials

Let $\begin{cases} A(x) = \sum_{k=0}^{n} a_k x^k \\ B(x) = \sum_{k=0}^{n} b_k x^k \end{cases}$ be two polynomials

The product of $A(x)$ and $B(x)$, denoted $C(x)$ is a polynomial of the form:

$$C(x) = \sum_{k=0}^{2n-1} c_k x^k$$

where each component of polynomial $C(x)$ is:

$$c_k = \sum_{s+t=k} a_s b_t$$

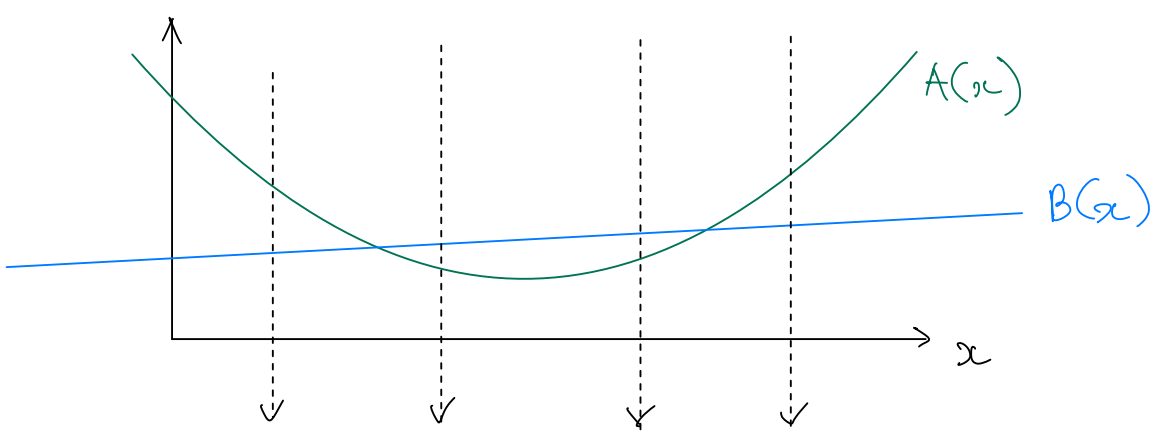
Conclusions

- The task of computing coefficients of $C(x)$ is equivalent to the task of computing $(a * b)$
- Vector convolution, in the context of multiplying polynomials, resulted in polynomial $C(x)$ of higher dimension, up to $m + n$.
  The basis of $C(x)$, given $A(x) \in \mathbb{R}^m$, $B(x) \in \mathbb{R}^n$ is:
  $$\text{basis}_{C(x)} = \{1, x, x^2, \ldots x^{m+n}\}$$
- It can be proven that multiplying polynomials is a linear transformation
  Prove $\begin{cases} T(a + b) = T(a) + T(b) \text{ with } T(Ax) = A(x) + B(x) \\ T(c \cdot a) = cT(a) \end{cases}$
  This leads to vector convolution is also a linear transformation.
- The straightforward method of multiplying 2 polynomials cost $\Theta(n^2)$

Framework for polynomial multiplication (Point-value Representation)

Lets work with the same example: $\begin{cases} A(x) = 1 + 2x + 3x^2 \\ B(x) = 4 + 5x \end{cases}$

Since $C(x) = A(x) \cdot B(x)$ will be of degree 3 $(2+1)$, I need to pick 4 distinct points on x-axis to form 4 equations



solving these equations will resulted in the coefficients of $C(x)$

$\Rightarrow$ This will still cost $\Theta(n^2)$

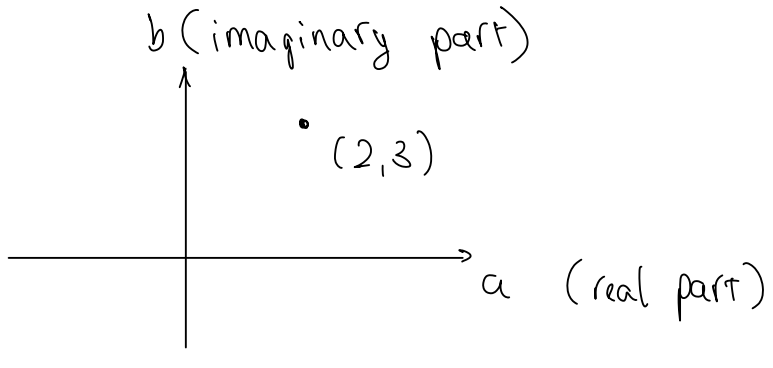
We will later see that if we pick these distinct points cleverly, the cost can reduce time complexity to $\Theta(n \log n)$ time.

Recall: Complex number

Represented as: $z = \underbrace{a}_{\text{real part}} + \underbrace{bi}_{\text{imaginary part}}$

In the complex plane, a complex number can be visualized as a point where:
- The x-axis represents the real part (a)
- The y-axis represents the imaginary part (b)

Example: Given a complex number $c = 2 + 3i$. Then this number will correspond to point $(2, 3)$ on the complex plane.



Recall: Euler's Formula

States that any complex number can be represented in terms of trigonometric functions of this form:

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

where
- $e^{i\theta}$ is a point on the unit circle on complex plane
- angle θ indicates how far counter clockwise you rotate from the positive x-axis
- coordinate $(\cos\theta, \sin\theta)$ gives the position on this circle

Example: Using the same complex number as above

$$c = 2 + 3i$$

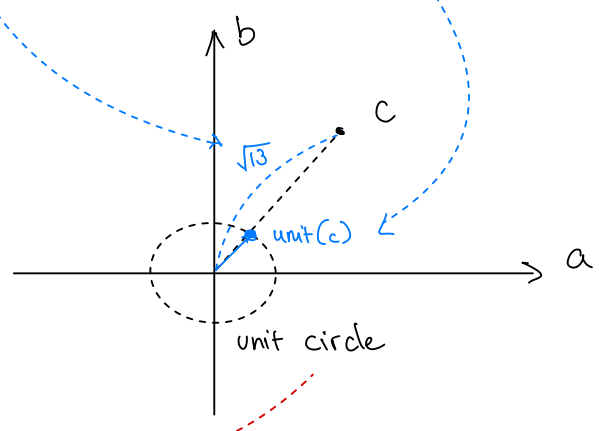
We can show that c can be expressed in trigonometric form:

$$c = |c|(\cos\theta + i\sin\theta)$$

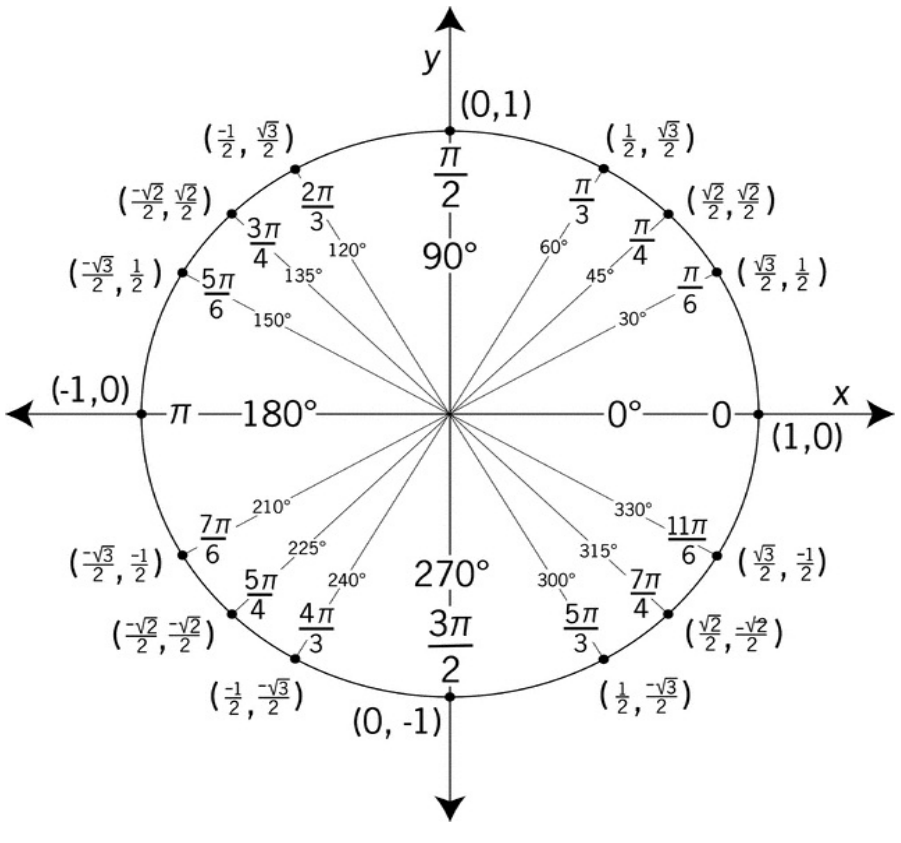
Since
$$|c| = \sqrt{a^2 + b^2} = \sqrt{13}$$
$$\theta = \tan^{-1}\left(\frac{b}{a}\right) \approx 0.9828$$

$$\Rightarrow c = \sqrt{13}(\cos 0.9828 + i \cdot \sin 0.9828)$$

Visualize c:



Recall: Unit Circle



Any points on unit circle has magnitude of 1:

$$|z| = \sqrt{a^2 + b^2} = 1$$

or:

$$\cos^2\theta + \sin^2\theta = 1$$

Recall: Root of Unity

For any positive integer N, let $\omega_N = e^{\frac{i2\pi}{N}}$ be the principal root of unity. The k^{th} root of unity is defined as:

$$\omega_N^k = e^{\left(i \cdot \frac{2\pi}{N} \cdot k\right)}, \text{ where } k = 0, 1, \ldots, N-1$$

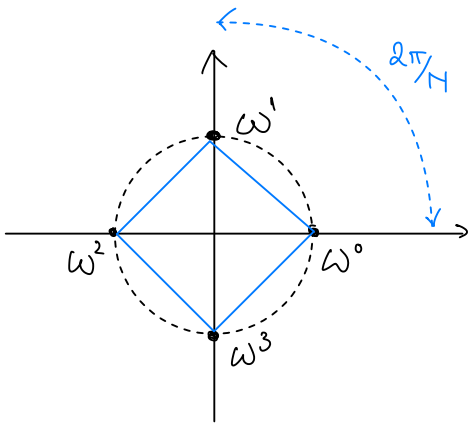
These roots represent points on the unit circle that are evenly spaced in the angle of $\frac{2\pi}{N}$.

Properties:

- Periodic nature: raising any root k to its degree will return 1

$$(\omega_N^k)^N = 1$$

- Geometric representation: these roots form a regular polygon inside the unit circle



$N = 4$

$$\Rightarrow \begin{cases} \omega_N^0 = e^0 = 1 \\ \omega_N^1 = e^{i\pi/2} = i \\ \omega_N^2 = e^{i\pi} = -1 \\ \omega_N^3 = e^{i\,3\pi/2} = -i \end{cases}$$

Roots of Unity's properties

- Cancellation Lemma

For any integers $n > 0$, $k \geq 0$, and $d > 0$

$$\omega_{dn}^{dk} = \omega_n^k$$

↳ Corollary 30.4: $\omega_n^{n/2} = \omega_2 = -1$

- Halving Lemma

$$(\omega_n^k)^2 = (\omega_n^{k+n/2})^2, \text{ where } n > 0 \text{ is even}$$

In words: Squares of the n complex n^{th} roots of unity are the $n/2$ complex $(n/2)^{th}$ roots of unity

- Summation Lemma

For integer $n \geq 1$, $k \neq 0$, k not divisible by n

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

Discrete Fourier Transform

If we pick n roots of unity as n distinct points, we can evaluate / interpolate n-degree polynomial in $O(n \log n)$.

What is Discrete Fourier Transform ?

$$\begin{bmatrix} A(\omega_N^0) \\ A(\omega_N^1) \\ A(\omega_N^2) \\ \vdots \\ A(\omega_N^{N-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_N & \omega_N^2 & \dots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \dots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \dots & \omega_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{N-1} \end{bmatrix}$$

$\hat{a}$ = Discrete Fourier Transform Matrix a

Discrete Fourier Transform

Maps any given vector $a = (a_0, a_1, \dots, a_{N-1})$ of complex numbers to a vector $\hat{a} = (A(\omega_N^0), A(\omega_N^1), A(\omega_N^2), \dots A(\omega_N^{N-1}))$

where $A(x) = \sum_{k=0}^{N-1} a_k x^k \Rightarrow \hat{a}_k = A(\omega_N^k) = \sum_{j=0}^{n-1} a_j \omega_n^{jk}$

$\omega_N = e^{\frac{2\pi i}{N}}$

$\Rightarrow$ vector $\hat{a}$ is the DFT of vector a

↳ DFT provides a better way to convert polynomial $A(x)$ between coeffient form and point-value form.

Fast Fourier Transform (Evaluation at complex roots of unity)

- Is the algorithm that leverage charateristic of roots of unity to evaluate a polynomial in $O(n \log n)$.
- It states that: To evaluate polynomial $A(x)$ degree n at n^{th} roots of unity $\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}$. Solve:

$$A(x) = A_{even}(x^2) + x A_{odd}(x^2) \quad (1)$$

where $A_{even}(x) = a_0 + a_2 x + a_4 x^2 + \dots + a_{n-2} x^{n/2-1}$

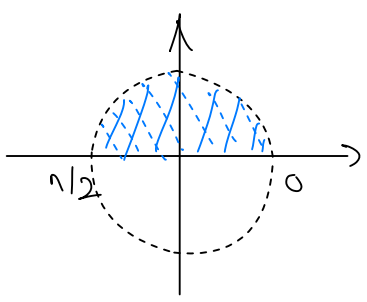
$A_{odd}(x) = a_1 + a_3 x + a_5 x^2 + \dots + a_{n-1} x^{n/2-1}$

- In other words, the problem reduce to:
  1. Evaluate polynomials $A_{even}(x)$ and $A_{odd}(x)$ degree $n/2$ at the points $(\omega_n^0)^2, (\omega_n^1)^2, \dots (\omega_n^{n-1})^2$.
  2. Then combine the result according to equation (1).

Pseudo code

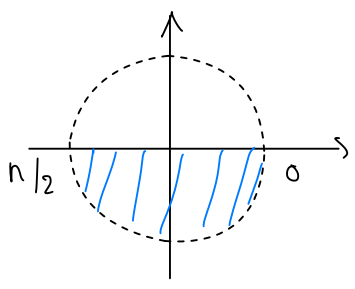
```
FFT ( a , n )
    if  n == 1:                                   } base case:  A(ω_1^0) = a_0 ω_1^0 = a_0
        return a
    ω_n = e^{2πi/n} ,   ω = 1                     } principal n^th root , cummulative ω
    a_even = ( a_0 , a_2 , ... , a_{n-2} )        } index split
    a_odd  = ( a_1 , a_3 , ... , a_{n-1} )
    y_even = FFT ( a_even , n/2 )                 } recursive
    y_odd  = FFT ( a_odd  , n/2 )
    for k : 0 ⟶ n/2 - 1 :        # at this point, ω = ω_n^k
        (1) y_k       = y_k^even + ω · y_k^odd    } combine result
        (2) y_{k+n/2} = y_k^even - ω · y_k^odd
        ω = ω · ω_n
    return →  = [ y_k ; y_{k+n/2} ]
```

(1) Calculate scalars of vector y from $0 \to n/2 - 1$



$$y_k = y_k^{even} + \omega\, y_k^{odd}$$
$$= A^{even}(\omega_{n/2}^k) + \omega_n^k A^{odd}(\omega_{n/2}^k)$$
$$= A^{even}(\omega_n^{2k}) + \omega_n^k A^{odd}(\omega_n^{2k}) \quad \langle \text{cancellation lemma} \rangle$$
$$= A(\omega_n^k) \quad \text{for } k: 0 \to n/2 - 1$$

(2) Calculat scalars of vector y from $n/2 \to n$



$$y_{k+n/2} = y_k^{even} - \omega\, y_k^{odd}$$
$$= A^{even}(\omega_{n/2}^k) + \omega_n^{k+n/2} A^{odd}(\omega_{n/2}^k)$$
$$= A^{even}(\omega_n^{2k}) + \omega_n^{k+n/2} A^{odd}(\omega_n^{2k}) \quad \langle \text{cancellation lemma} \rangle$$
$$= A^{even}(\omega_n^{2k+n}) + \omega_n^{k+n/2} A^{odd}(\omega_n^{2k+n})$$
$$= A(\omega_n^{k+n/2}) \quad \text{for } k: 0 \to n/2 - 1$$

spin 1 full round, land in the same place

With (1) and (2), we have calculated all n^{th} roots of unity:

$$\vec{y} = DFT_n = [\, A(\omega_n^0), A(\omega_n^1), A(\omega_n^2), \dots A(\omega_n^{n-1}) \,]$$

Why this is better?

Because of Halving Lemma, when we "split" to $A_{even}(x)$ and $A_{odd}(x)$ on $(\omega_n^0)^2, (\omega_n^1)^2, \ldots, (\omega_n^{n-1})^2$. This list does not contain n distinct values, but $(n/2)$ distinct values. Hence, reduce the problem size by half

Fast Fourier Transform's Recurrence Relation

$$T(n) = 2T(n/2) + \Theta(n)$$
$$= \Theta(n \log n) \quad \text{< Master Theorem. case 2 >}$$

Inverse Fast Fourier Transform (Interpolation at complex roots of unity)

Recall the DFT matrix:

$\hat{a}$ = Discrete Fourier Transform Matrix a

$$\begin{bmatrix} A(\omega_N^0) \\ A(\omega_N^1) \\ A(\omega_N^2) \\ \vdots \\ A(\omega_N^{N-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{N-1} \end{bmatrix}$$

or: $\hat{a} = V_n \cdot a$

or: $\hat{a} = DTF_n \cdot a$

To interpolate polynomials $A(x)$, we simply inverse the operation:

$$a = V_n^{-1} \cdot \hat{a}$$

We can do this because $V_n^{-1} \cdot V_n = I_n$. Proof in book (p. 914)

Theorem 30.7

For $j, k = 0, 1, \ldots, n-1$. The (j,k) entry of V_n^{-1} is:

$$V_n^{-1}{}_{(j,k)} = \omega_n^{-jk/n}$$

With inverse matrix V_n^{-1} defined, coefficients of polynomial $A(x)$ can be defined as:

$$a_j = \frac{1}{n} \sum_{k=0}^{n-1} \hat{a}_k \cdot \omega_n^{-kj}$$

interpolation formula

Inverse-FFT: same steps as FFT but switch a for $\hat{a}$, ω_n for ω_n^{-1} and divide each element of the result by n.

How do I visualize this evaluation process in a geometrical way?

Lets say we are evaluating polynomial $A(x) = \sum_{k=0}^{n-1} a_k x^k$ for N^{th} roots of unity, lets consider the case of 8^{th} root of unity:

$$A(\omega_8^k) = A_{even}(\omega_8^{2k}) + \omega_8^k A_{odd}(\omega_8^{2k})$$
$$= A_{even}(\omega_4^k) + \omega_8^k A_{odd}(\omega_4^k) \quad \text{(Cancellation Lemma)}$$

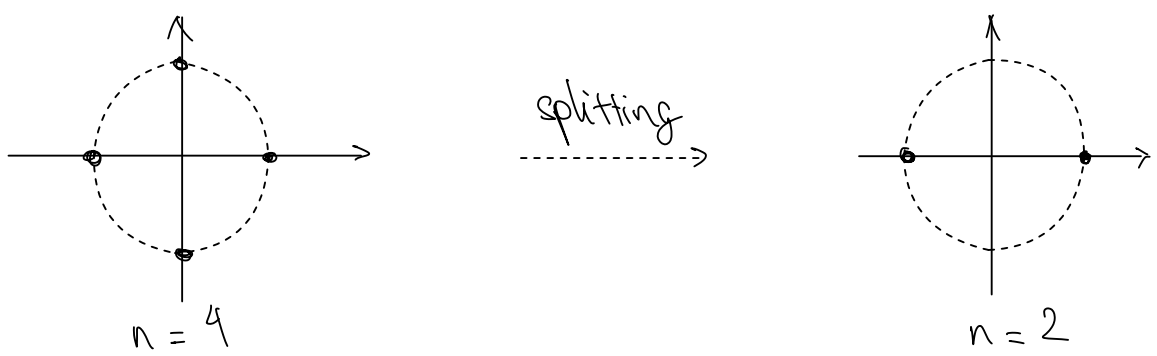
At this point, visually:

Then for each $A_{even}(\omega_4^k)$ and $A_{odd}(\omega_4^k)$, splits again:

or $A_{even}(\omega_4^k) = A_{even}(\omega_4^{2k}) \pm \omega_4^k A_{odd}(\omega_4^{2k})$

$A_{odd}(\omega_4^k) = A_{even}(\omega_2^k) \pm \omega_4^k A_{odd}(\omega_2^k)$ <Cancellation lemma>

At this point, visually:

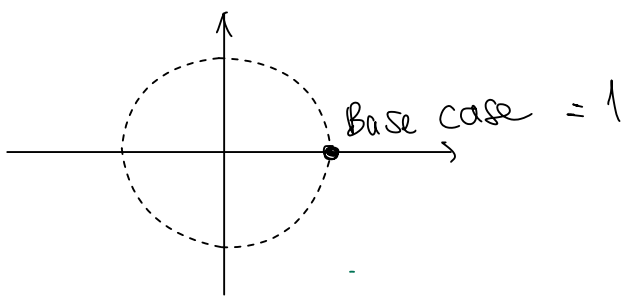


And again, now we reach the base case:

or $A_{even}(\omega_2^k) = A_{even}(\omega_2^{2k}) \pm \omega_2^k A_{odd}(\omega_2^{2k})$

$A_{odd}(\omega_2^k) = A_{even}(\omega_1^k) \pm \omega_2^k A_{odd}(\omega_1^k)$

Visually:



Once reach base case, the recursive calls returns and calculate the DFT's scalars in this sequence:

Base case $\longrightarrow$ $A_{even}(\omega_1^k)$, $A_{odd}(\omega_1^k)$ $\longrightarrow$ $A_{even}(\omega_2^k)$, $A_{odd}(\omega_2^k)$ $\longrightarrow$ $A_{even}(\omega_4^k)$, $A_{odd}(\omega_4^k)$ $\longrightarrow$ $A(\omega_8^k)$

Theorem 30.8 (Convolution Theorem)

For any 2 vectors a and b of length n, where n is an exact power of 2

$$a \otimes b = DFT_{2n}^{-1}\left(DFT_{2n}(a) \cdot DFT_{2n}(b)\right)$$

$$DFT_{2n}(a \otimes b) = DFT_{2n}(a) \cdot DFT_{2n}(b)$$

Convolution in **Real Space** is Products in **Fourier Space**

$$c = a * b \qquad DFT_{2n}(c) = DFT_{2n}(a) \cdot DFT_{2n}(b)$$