Integer multiplication
and the
Fast Fourier Transform

$$A^n = A^{n/2} \cdot A^{n/2}$$

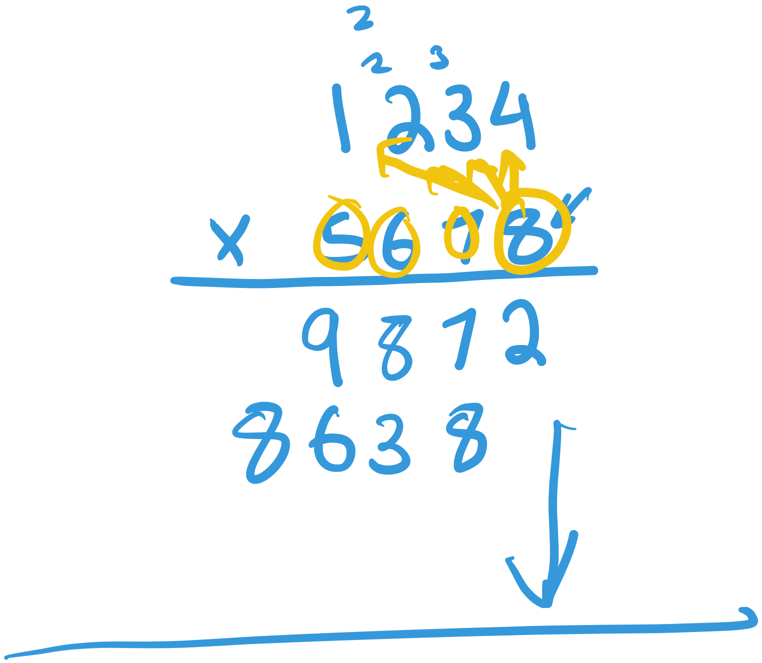
A, A^2, A^4, A^8, A^{16}

Multiplying n-bit integers

Karatsuba

Input: $x, y \in \{0,1\}^n$

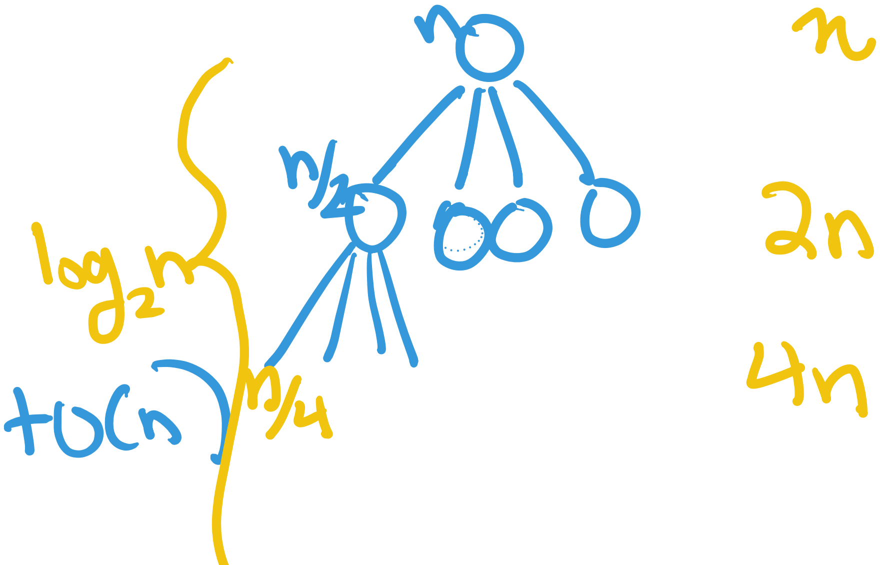
Output: product $(xy) \in \{0,1\}^{2n}$



$$n \sum_{i=0}^{\boxed{\log_2 n}} 2^i$$

$$O(n \cdot 2^{\log_2 n}) = O(n^2)$$

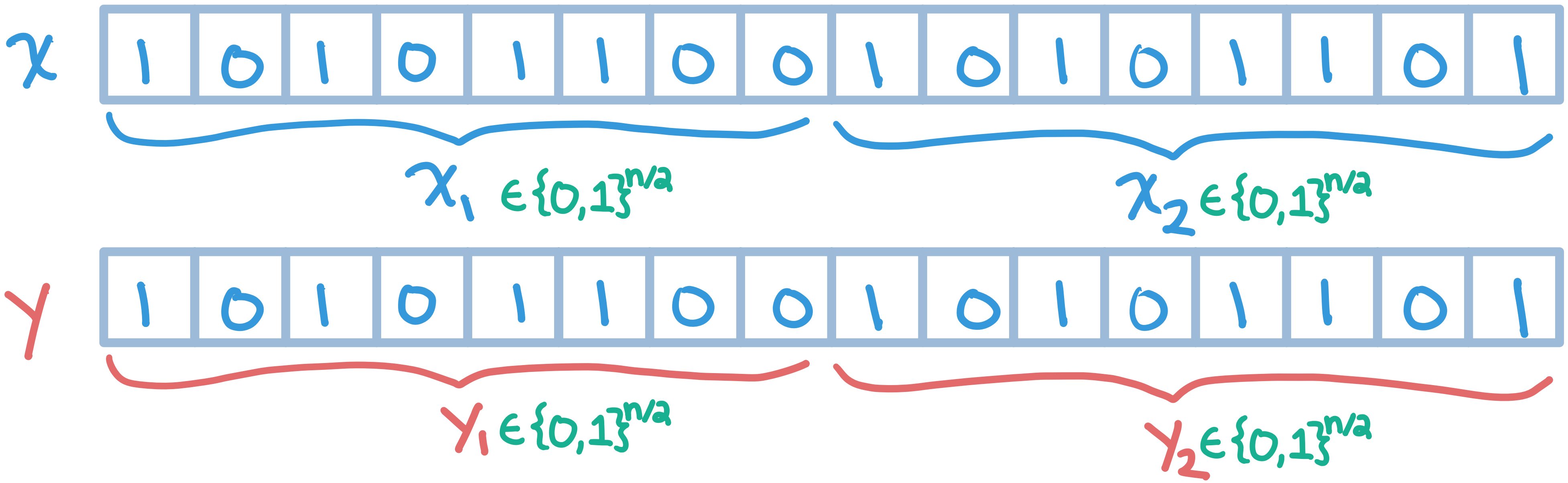
$$O(n^2)$$



$$T(n) = 4T(n/2) + O(n)$$

Better?

Divide bits in half



$$x = x_1 2^{n/2} + x_2, \quad y = y_1 2^{n/2} + y_2$$

$$x \cdot y = \underline{x_1 y_1}\, 2^n + (\underline{x_1 y_2} + \underline{y_1 x_2})\, 2^{n/2} + \underline{x_2 y_2}$$

which multiplies 4 pairs of $(n/2)$-bit numbers

$$\boxed{x \cdot y = x_1 y_1 2^n + (\underline{x_1 y_2} + \underline{y_1 x_2}) 2^{n/2} + \underline{x_2 y_2}}$$

Ideas?

$$\boxed{\begin{array}{cc} x_{1,1} & x_{1,2} \\ y_{1,1} & y_{1,2} \end{array}} \Big\}$$

$$x_{1,1} y_{1,1} 2^{n/2} + (x_{1,2} y_{1,1} + x_{1,1} y_{1,2}) 2^{n/4} + x_{1,2} y_{1,2}$$

$$x_1 y_2 \left\{ \begin{array}{cc} x_{1,1} & x_{1,2} \\ y_{2,1} & y_{2,2} \end{array} \right.$$

$$\boxed{x \cdot y = x_1 y_1 2^{n} + (x_1 y_2 + y_1 x_2) 2^{n/2} + x_2 y_2}$$

Karatsuba's Trick

already computed

$$\underbrace{(x_1+x_2)(y_1+y_2)}_{\text{one multiply}} = x_1 y_1 + \underset{\text{middle term}}{\underline{x_1 y_2 + x_2 y_1}} + x_2 y_2$$

$$-(x_1 y_2 + x_2 y_1) = x_1 y_1 + x_2 y_2 - (x_1+x_2)(y_1+y_2)$$

2 multiplies for the price of 1 (more)!

$$\boxed{x \cdot y = x_1 y_1 2^n + (x_1 y_2 + y_1 x_2) 2^{n/2} + x_2 y_2}$$

Karatsuba's Trick

already computed

$$\underbrace{(x_1 + x_2)(y_1 + y_2)}_{\text{one multiply}} = x_1 y_1 + \underset{\text{middle term}}{\underline{x_1 y_2 + x_2 y_1}} + x_2 y_2$$

$$x_1 y_2 + x_2 y_1 = x_1 y_1 + x_2 y_2 - (x_1 + x_2)(y_1 + y_2)$$

2 multiplies for the price of 1 (more)!

① Recursive specification

Designing a recursive algorithm

① Recursive specification

- declares input/output of algo
- "contract" algo must fulfill
- induction hypothesis for correctness

② Implement spec

③ Prove correctness

argue algo satisfies recursive spec for all n by induction on n

Divide bits in half

$$x \quad \underbrace{1\,0\,1\,0\,1\,1\,0\,0}_{x_1 \in \{0,1\}^{n/2}}\,\underbrace{1\,0\,1\,0\,1\,1\,0\,1}_{x_2 \in \{0,1\}^{n/2}}$$

$$y \quad \underbrace{1\,0\,1\,0\,1\,1\,0\,0}_{y_1 \in \{0,1\}^{n/2}}\,\underbrace{1\,0\,1\,0\,1\,1\,0\,1}_{y_2 \in \{0,1\}^{n/2}}$$

$$x = x_1 2^{n/2} + x_2, \quad y = y_1 2^{n/2} + y_2$$

$$x \cdot y = \underline{x_1 y_1}\, 2^n + (\underline{x_1 y_2} + \underline{y_1 x_2}) 2^{n/2} + \underline{x_2 y_2}$$

which multiplies 4 pairs of (n/2)-bit numbers

Karatsuba's Trick

already computed

$$\underbrace{(x_1+x_2)(y_1+y_2)}_{\text{one multiply}} = x_1y_1 + \underset{\text{middle term}}{\underline{x_1y_2 + x_2y_1}} + x_2y_2$$

$$x_1y_2 + x_2y_1 = x_1y_1 + x_2y_2 - (x_1+x_2)(y_1+y_2)$$

2 multiplies for the price of 1 (more) !

① Recursive specification

`multiply`$(x[1..n], y[1..n])$ = the (2n-bit integer) product of x and y (as n-bit integers).

Designing a recursive algorithm

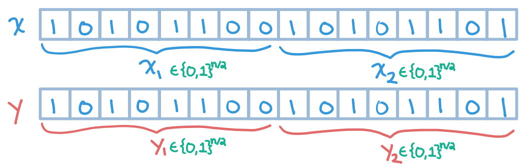
① Recursive specification
- declares input/output of algo
- "contract" algo must fulfill
- induction hypothesis for correctness

② Implement spec

③ Prove correctness

argue algo satisfies recursive spec for all n by induction on n

Divide bits in half



$$x = x_1 2^{n/2} + x_2, \quad y = y_1 2^{n/2} + y_2$$

$$x \cdot y = \underline{x_1 y_1} 2^n + (\underline{x_1 y_2} + \underline{y_1 x_2}) 2^{n/2} + \underline{x_2 y_2}$$

which multiplies 4 pairs of (n/2)-bit numbers

Karatsuba's Trick

already computed

$$\underbrace{(x_1+x_2)(y_1+y_2)}_{\text{one multiply}} = x_1 y_1 + \underbrace{x_1 y_2 + x_2 y_1}_{\text{middle term}} + x_2 y_2$$

$$x_1 y_2 + x_2 y_1 = x_1 y_1 + x_2 y_2 - (x_1+x_2)(y_1+y_2)$$

2 multiplies for the price of 1 (more) !

① Recursive specification

`select`$(k, A[1..n])$: Given an array $A[1..n]$ of comparable elements and an integer $k \in \{1, \ldots, n\}$, returns the kth smallest element out of $A[1..n]$.

② Implement spec

```
multiply($x[1..n]$, $y[1..n]$)
1. If $n = 0$ then return 0.
2. If $n = 1$ then return $x[1] * y[1]$.
3. If $n$ is not even then pad $x$ and $y$ with a leading 0, making $n$ even.
   // $O(n)$.
4. Let $x_1 = x[1..n/2]$, $x_2 = [n/2+1..n]$, $y_1 = y[1..n/2]$, and
   $y_2 = y[n/2+1..n]$ (as $n/2$-bit integers).                         // $O(n)$.
5. Let $x_3[1..n/2+1] = x_1 + x_2$ and $y_3[1..n/2+1] = (y_1 + y_2)$ (as
   $(n/2+1)$-bit integers).                                             // $O(n)$.
6. Let $Z_1$ = multiply($x_1$, $y_1$), $Z_2$ = multiply($x_2$, $y_2$), and
   $Z_3$ = multiply($x_3$, $y_3$).                                     // $3T((n+2)/2)$.
/* Note that multiplying by $2^n$ is just a bit-shift and takes $O(n)$ time. */
7. Return $2^n * Z_1 + 2^{n/2}(Z_3 - Z_1 - Z_2) + Z_2$
```

Designing a recursive algorithm

① Recursive specification
- declares input/output of algo
- "contract" algo must fulfill
- induction hypothesis for correctness

② Implement spec

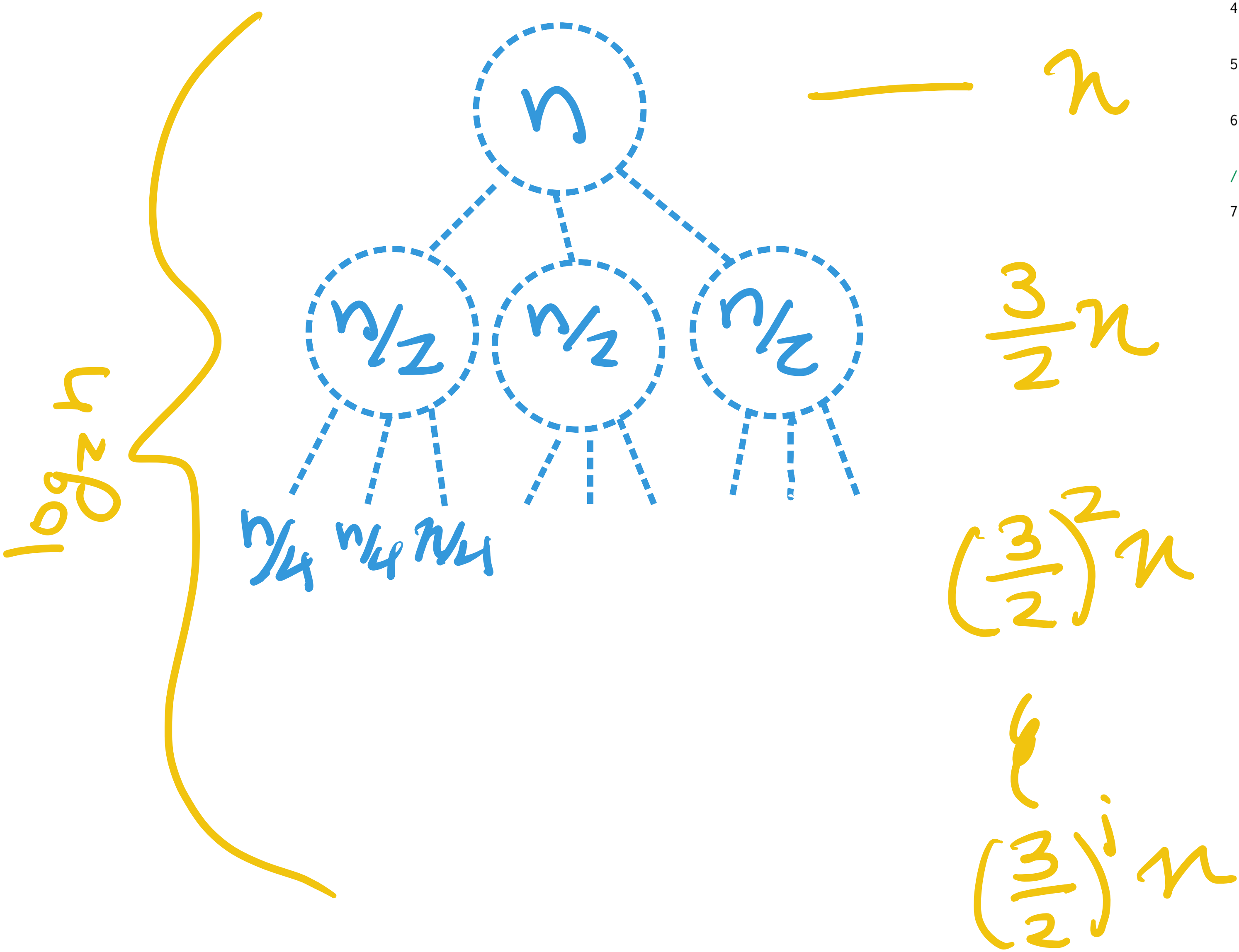
③ Prove correctness

argue algo satisfies recursive spec for all n by induction on n

4. Running time

$$T(n) = 3T(\tfrac{n}{2}) + O(n)$$

```
multiply(x[1..n], y[1..n])
If $n = 0$ then return 0.
If $n = 1$ then return $x[1] * y[1]$.
If $n$ is not even then pad $x$ and $y$ with a leading 0, making $n$ even.
    // $O(n)$.
Let $x_1 = x[1..n/2]$, $x_2 = [n/2+1..n]$, $y_1 = y[1..n/2]$, and
    $y_2 = y[n/2+1..n]$ (as $n/2$-bit integers).                  // $O(n)$.
Let $x_3[1..n/2+1] = x_1 + x_2$ and $y_3[1..n/2+1] = (y_1 + y_2)$ (as
    $(n/2+1)$-bit integers).                                      // $O(n)$.
Let $Z_1$ = multiply($x_1$,$y_1$), $Z_2$ = multiply($x_2$,$y_2$), and
    $Z_3$ = multiply($x_3$,$y_3$).                                // $3T((n+2)/2)$.
/* Note that multiplying by $2^n$ is just a bit-shift and takes $O(n)$ time. */
Return $2^n * Z_1 + 2^{n/2}(Z_3 - Z_1 - Z_2) + Z_2$
```



Running time

$$T(n) = 3T(\tfrac{n}{2}) + O(n)$$

$$\sum_{i=0}^{\log_2 n} \left(\tfrac{3}{2}\right)^i n = n \left(\tfrac{3}{2}\right)^{\log_2 n}$$

$$= n\left(2^{\log_2 \frac{3}{2}}\right)^{\log_2 n}$$

$$2^{\log_2 n} h =$$
$$= n^{\log_2 h}$$

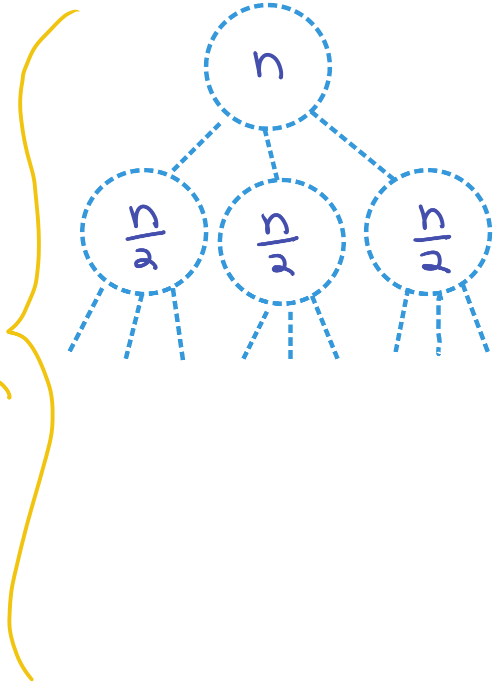
$$= n \cdot n^{\log_2 3 - \log_2 2}$$

$$n^{1.58} \qquad n^{\log_2 3}$$

$$n \log^c n$$

4. Running time

```
multiply(x[1..n], y[1..n])
If $n = 0$ then return 0.
If $n = 1$ then return $x[1] * y[1]$.
If $n$ is not even then pad $x$ and $y$ with a leading 0, making $n$ even.   // $O(n)$.
Let $x_1 = x[1..n/2]$, $x_2 = [n/2+1..n]$, $y_1 = y[1..n/2]$, and $y_2 = y[n/2+1..n]$ (as $n/2$-bit integers).   // $O(n)$.
Let $x_3[1..n/2+1] = x_1 + x_2$ and $y_3[1..n/2+1] = (y_1 + y_2)$ (as $(n/2+1)$-bit integers).   // $O(n)$.
Let $Z_1$ = multiply($x_1$, $y_1$), $Z_2$ = multiply($x_2$, $y_2$), and $Z_3$ = multiply($x_3$, $y_3$).   // $3T((n+2)/2)$.
/* Note that multiplying by $2^n$ is just a bit-shift and takes $O(n)$ time. */
Return $2^n * Z_1 + 2^{n/2}(Z_3 - Z_1 - Z_2) + Z_2$
```



n

$3 \times \frac{n}{2} = \frac{3}{2} n$

$\left(\frac{3}{2}\right)^2 n$

$\vdots$

$\left(\frac{3}{2}\right)^h n$

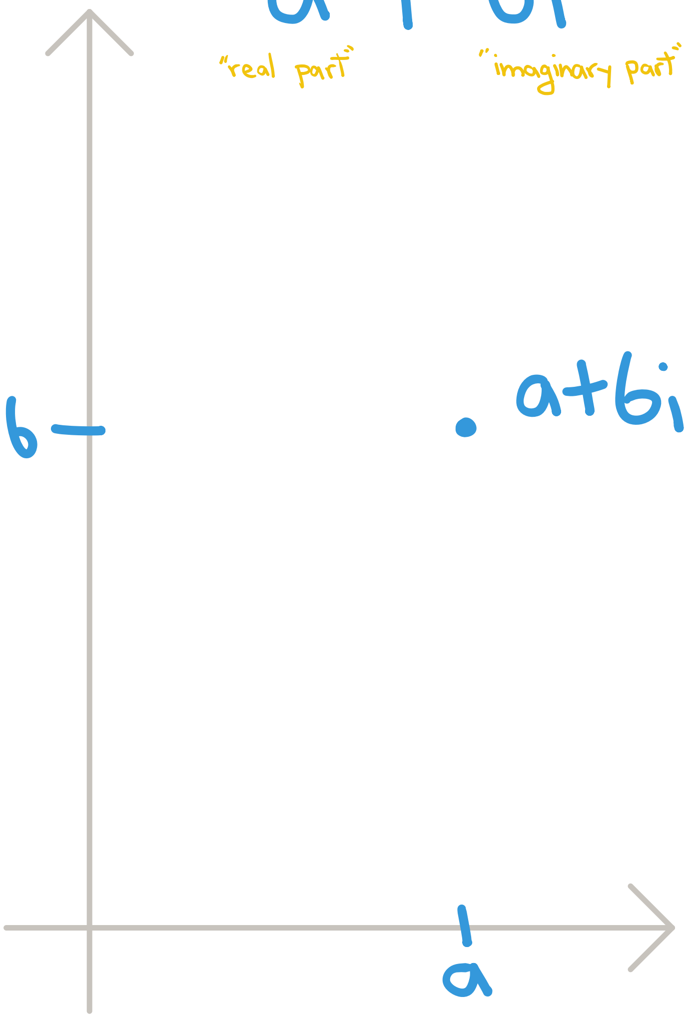
Integer multiplication and the Fast Fourier Transform

Complex numbers $\mathbb{C}$

"imaginary i" satisfying $i^2 = -1$

$$a + bi$$

"real part" "imaginary part"



multiplication:

$$(a+bi)(c+di)$$
$$= ac + (bc+ad)i - bd$$

⇒ polynomials over $\mathbb{C}$

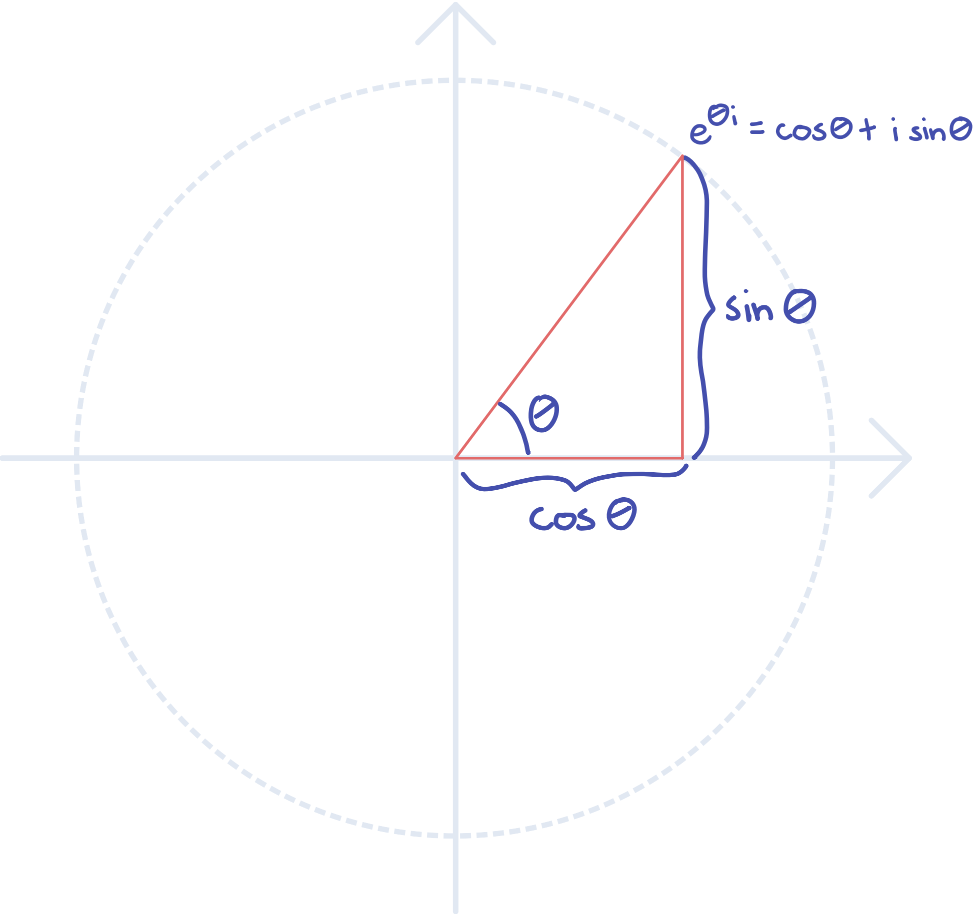
$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$$

⇒ power series over $\mathbb{C}$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Euler's Formula

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



$a + bi$

"real part" "imaginary part"

"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

$\Rightarrow$ Euler's identity:

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

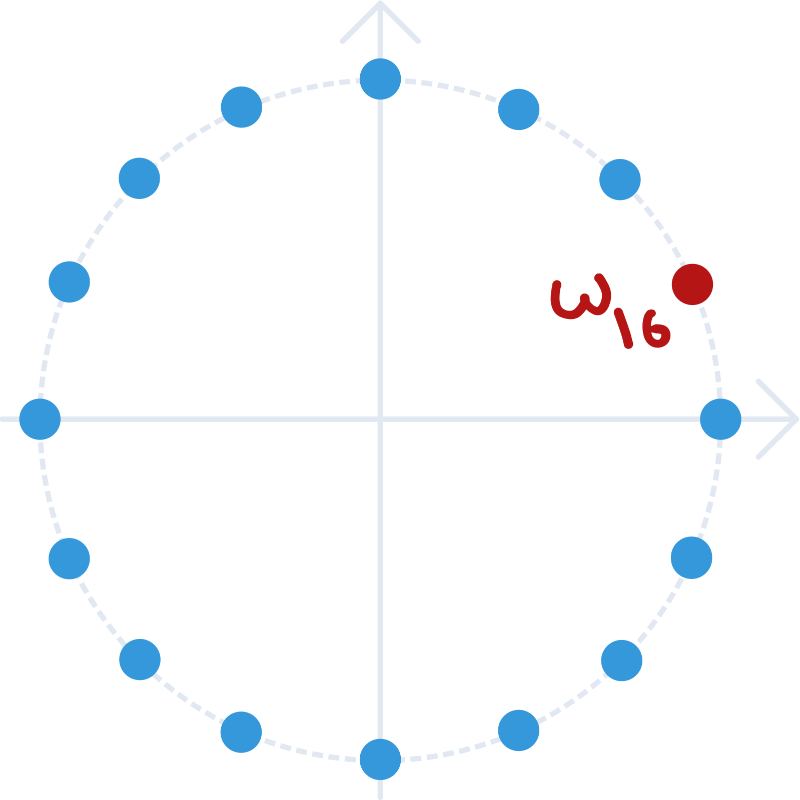
nth roots of unity:

any $\omega \in \mathbb{C}$ s.t. $\omega^n = 1$

e.g. $\omega = 1$ for all n

$\omega = -1$ for n even

$\omega = i$ for $n = 4$



let

$$\omega_n \overset{\text{def}}{=} \boxed{e^{2\pi i/n}}$$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$

are all nth roots of unity

$a + bi$

"real part" "imaginary part"

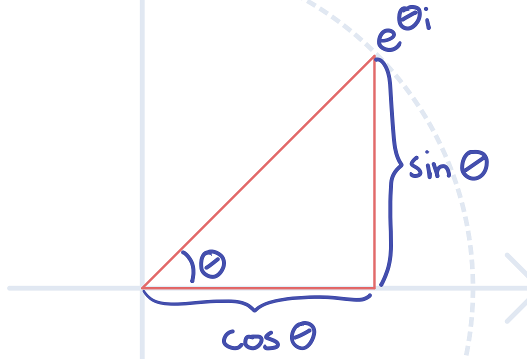
"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Euler's formula:

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



Euler's identity:

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

(nth) discrete Fourier transform F_n

Input: $a_0, a_1, a_2, \cdots, a_{n-1} \in \mathbb{C}$

Output: n values

$$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$$

where $p(x) = a_0 + a_1 x + a_1 x^2 + \cdots + a_n x^{n-1}$

---

conjugate Fourier transform $F_n^\star$

Input: $a_0, a_1, a_2, \cdots, a_{n-1} \in \mathbb{C}$

Output: n values

$$F_n^\star a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$$

where $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^{n-1}$

$$\underset{\text{"real part"}}{a} + \underset{\text{"imaginary part"}}{bi}$$

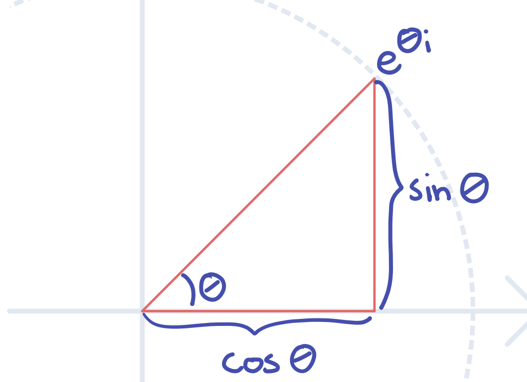
"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Euler's formula:

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



Euler's identity:

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

$$\omega_n \overset{\text{def}}{=} e^{2\pi i/n}$$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

Input: $a_0, a_1, a_2, \cdots, a_{n-1} \in \mathbb{C}$

Output: n values

$$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$$

$$F_n^\star a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$$

where $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^{n-1}$

Fact: $F_n^\star F_n a = F_n F_n^\star a = na$

(i.e., F_n and $F_n^\star$ are essentially inverse)

$$\underset{\text{"real part"}}{a} + \underset{\text{"imaginary part"}}{bi}$$

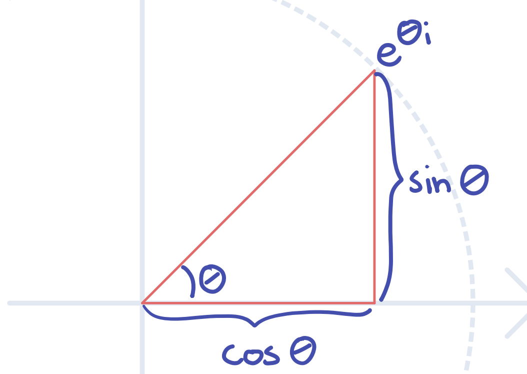
"imaginary i" s.t. $i^2 = -1$

$$(a+bi)(c+di) = ac + (bc+ad)i - bd$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Euler's formula:

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



Euler's identity:

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

$$\omega_n \overset{\text{def}}{=} e^{2\pi i/n}$$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

Input: $a_0, a_1, a_2, \cdots, a_{n-1} \in \mathbb{C}$

Output: n values

$$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$$

$$F_n^* a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$$

where $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^{n-1}$

Fact: $F_n^* F_n a = F_n F_n^* a = na$

Theorem: $F_n a$ and $F_n^* a$ can be computed in $O(n \log n)$ time.

$$\underset{\text{"real part"}}{a} + \underset{\text{"imaginary part"}}{b} i$$

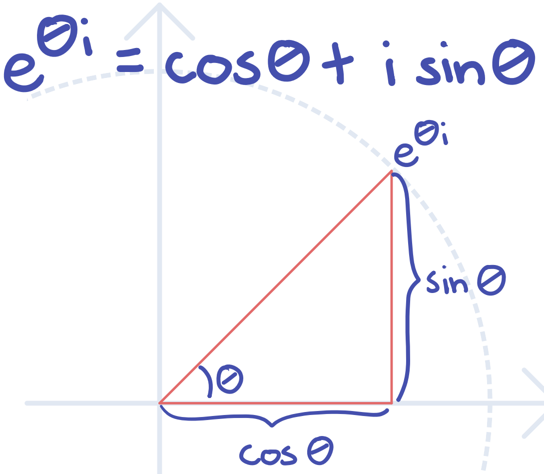
"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$

Euler's formula:

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



Euler's identity:

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

$$\omega_n \overset{\text{def}}{=} e^{2\pi i/n}$$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

Application: multiplying polynomials

$$p(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$$

$$q(x) = b_0 + b_1 x + \cdots + b_{n-1} x^{n-1}$$

goal: compute

$$r(x) = p(x)\,q(x)$$
$$= c_0 + c_1 x + \cdots + c_{2n-2} x^{n-2}$$

where $c_k = \sum_{i=0}^{k} a_i b_{k-i}$

Ideas?

$$a + bi$$

"real part" "imaginary part"

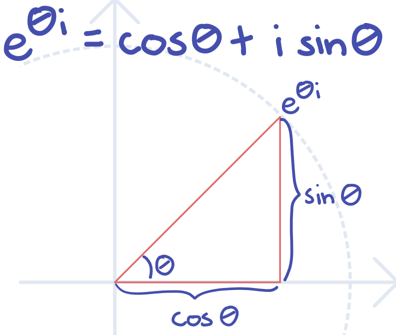
"imaginary i" s.t. $i^2 = -1$

$$(a+bi)(c+di) = ac + (bc+ad)i - bd$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Euler's formula:

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



Euler's identity:

- $e^{\pi i} = -1$ • $e^{2\pi i} = 1$

$$\omega_n \overset{\text{def}}{=} e^{2\pi i/n}$$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

$$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$$

$$F_n^{\star} a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$$

where $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^{n-1}$

Fact: $F_n^{\star} F_n a = F_n F_n^{\star} a = na$

Theorem: $F_n a$, $F_n^{\star} a$ in $O(n \log n)$

Input: $p(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}$

$q(x) = b_0 + b_1 x + \cdots + b_{n-1}x^{n-1}$

goal: $r(x) = p(x)\,q(x) = c_0 + c_1 x + \cdots + c_{2n-2}x^{n-2}$

Idea: $r(\omega_{2n}^i) = p(\omega_{2n}^i)\,q(\omega_{2n}^i)$

$$\begin{tikzcd}
(p,q) \arrow[d, "O(n\log n)"] \\
(F_{2n}a, F_{2n}b) \arrow[d, "O(n) \text{ by mult. pairwise}"] \\
F_{2n}c \arrow[d, "O(n\log n)"] \\
c = \frac{1}{2n}F^{\star}_{2n}F_{2n}c
\end{tikzcd}$$

$$\underset{\text{"real part"}}{a} + \underset{\text{"imaginary part"}}{bi}$$

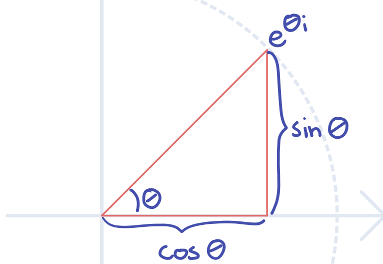
"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$

Euler's formula:

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



Euler's identity:

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

$\omega_n \overset{\text{def}}{=} e^{2\pi i/n}$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

---

$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$

$F_n^{\star} a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$

where $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^{n-1}$

Fact: $F_n^{\star}F_n a = F_n F_n^{\star} a = na$

Theorem: $F_n a$, $F_n^{\star} a$ in $O(n\log n)$

Input: $a_0, a_1, a_2, \cdots, a_{n-1} \in \mathbb{C}$

Output: n values

$$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$$

$$F_n^* a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$$

where $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^{n-1}$

Fact: $F_n^* F_n a = F_n F_n^* a = na$

Theorem: $F_n a$ and $F_n^* a$ can be computed in $O(n \log n)$ time.

$$\underbrace{a}_{\text{"real part"}} + \underbrace{b}_{\text{"imaginary part"}}i$$

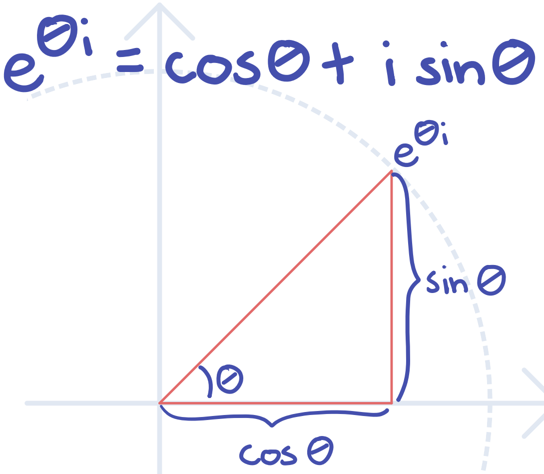
"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$

Euler's formula:

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



Euler's identity:

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

$$\omega_n \overset{\text{def}}{=} e^{2\pi i/n}$$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

Idea: $\omega_{2n}^2 = \omega_n$

let $a = (a_0, a_1, \cdots, a_{2n-1})$. $p(\omega_{2n}^k)$

$$\underline{(F_{2n}a)}_k = a_0 + a_1\underline{\omega_{2n}^k} + \cdots + a_{2n-1}(\omega_{2n}^k)^{2n-1}$$

$$= (a_0 + a_2(\omega_{2n}^k)^2 + \cdots + a_{2n-2}(\omega_{2n}^k)^{2n-2}) \quad \text{(even)}$$

$$+ (a_1\omega_{2n}^k + a_3(\omega_{2n}^k)^3 + \cdots + a_{2n-1}(\omega_{2n}^k)^{2n-1}) \quad \text{(odd)}$$

$$\underset{\text{"real part"}}{a} + \underset{\text{"imaginary part"}}{bi}$$

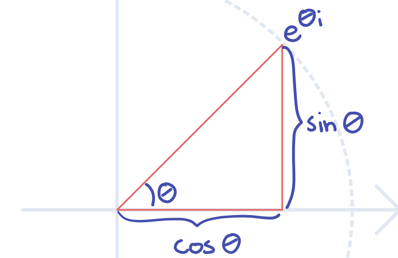
"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$

Euler's formula:

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



Euler's identity:

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

$$\omega_n \overset{\text{def}}{=} e^{2\pi i/n}$$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$

$F_n^* a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$

where $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^{n-1}$

Fact: $F_n^* F_n a = F_n F_n^* a = na$

Theorem: $F_n a$, $F_n^* a$ in $O(n \log n)$

Idea: $\omega_{2n}^2 = \omega_n$

---

let $a = (a_0, a_1, \cdots, a_{2n-1})$.

$$(F_{2n}a)_k = a_0 + a_1\omega_{2n}^k + \cdots + a_{2n-1}(\omega_{2n}^k)^{2n-1}$$

$$= \left(a_0 + a_2(\omega_{2n}^k)^2 + \cdots + a_{2n-2}(\omega_{2n}^k)^{2n-2}\right) \quad \text{(even)}$$

$$+ \left(a_1\omega_{2n}^k + a_3(\omega_{2n}^k)^2 + \cdots + a_{2n-1}(\omega_{2n}^k)^{2n-1}\right) \quad \text{(odd)}$$

$$\left(a_0 + a_2(\omega_{2n}^k)^2 + \cdots + a_{2n-2}(\omega_{2n}^k)^{2n-2}\right)$$

$$= a_0 + a_2\omega_n^k + \cdots + a_{2n-2}(\omega_n^k)^{n-1}$$

$$= (F_n a_{even})_k$$

w/ $a_{even} = (a_0, a_2, \cdots, a_{2n-2})$

$a + bi$

"real part" "imaginary part"

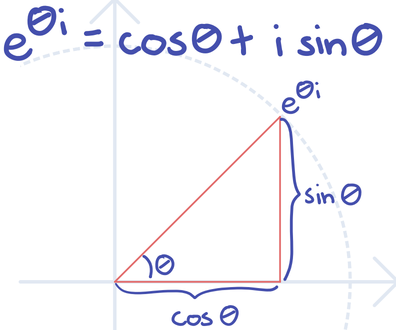
"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$

Euler's formula:

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



Euler's identity:

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

$\omega_n \overset{def}{=} e^{2\pi i/n}$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

---

$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$

$F_n^{\star} a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$

where $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^{n-1}$

Fact: $F_n^{\star} F_n a = F_n F_n^{\star} a = na$

Theorem: $F_n a$, $F_n^{\star} a$ in $O(n\log n)$

Idea: $\omega_{2n}^2 = \omega_n$

---

let $a = (a_0, a_1, \cdots, a_{2n-1})$.

$$(F_{2n}a)_k = a_0 + a_1\omega_{2n}^k + \cdots + a_{2n-1}(\omega_{2n}^k)^{2n-1}$$

$$= \left(a_0 + a_2(\omega_{2n}^k)^2 + \cdots + a_{2n-2}(\omega_{2n}^k)^{2n-2}\right)$$ (even)

$$+ \left(a_1\omega_{2n}^k + a_3(\omega_{2n}^k)^3 + \cdots + a_{2n-1}(\omega_{2n}^k)^{2n-1}\right)$$ (odd)

$$\left(a_1\omega_{2n}^k + a_3(\omega_{2n}^k)^3 + \cdots + a_{2n-1}(\omega_{2n}^k)^{2n-1}\right)$$

$$= \omega_{2n}^k\left(a_1 + a_3\omega_n^k + \cdots + a_{2n-1}(\omega_n^k)^{n-1}\right)$$

$$= \omega_{2n}^k(F_n a_{odd})_k$$

w/ $a_{odd} = (a_1, a_3, \cdots, a_{2n-1})$

$$a + bi$$

"real part" "imaginary part"

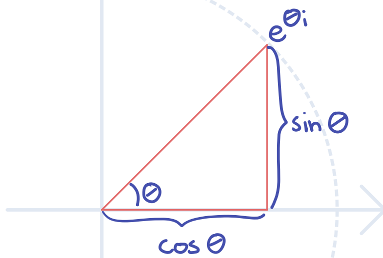
"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$

Euler's formula:

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



Euler's identity:

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

$\omega_n \overset{def}{=} e^{2\pi i/n}$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

---

$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$

$F_n^\star a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$

where $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^{n-1}$

Fact: $F_n^\star F_n a = F_n F_n^\star a = na$

Theorem: $F_n a, F_n^\star a$ in $O(n\log n)$

Idea: $\omega_{2n}^2 = \omega_n$

let $a = (a_0, a_1, \dots, a_{2n-1})$.

$$(F_{2n}a)_k = a_0 + a_1\omega_{2n}^k + \dots + a_{2n-1}(\omega_{2n}^k)^{2n-1}$$

$$= (a_0 + a_2(\omega_{2n}^k)^2 + \dots + a_{2n-2}(\omega_{2n}^k)^{2n-2}) \quad \text{(even)}$$

$$+ (a_1\omega_{2n}^k + a_3(\omega_{2n}^k)^2 + \dots + a_{2n-1}(\omega_{2n}^k)^{2n-1}) \quad \text{(odd)}$$

$$= (F_n a_{even})_k + \omega_{2n}^k (F_n a_{odd})_k$$

recurse!

$a + bi$

"real part" "imaginary part"

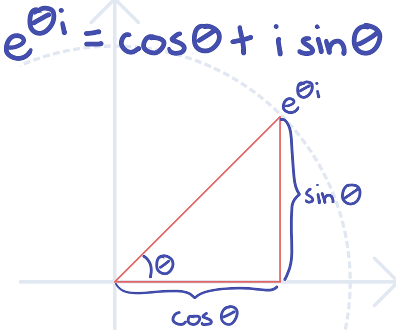
"imaginary i" s.t. $i^2 = -1$

$(a+bi)(c+di) = ac + (bc+ad)i - bd$

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$

Euler's formula:

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



Euler's identity:

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

$\omega_n \overset{def}{=} e^{2\pi i/n}$

$1, \omega_n, \omega_n^2, \dots, \omega_n^{n-1}$ are all nth roots of unity

$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \dots, p(\omega_n^{n-1}))$

$F_n^* a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \dots, p(\omega_n))$

where $p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^{n-1}$

Fact: $F_n^* F_n a = F_n F_n^* a = na$

Theorem: $F_n a, F_n^* a$ in $O(n \log n)$

Fast Fourier Transform($a_0, \cdots, a_n$):

```
1. if $n=0$   return $\emptyset$
2. if $n=1$   return $(a_0)$
3. let $(y_0, \cdots, y_{\lceil n/2 \rceil -1}) = FFT(a_{even})$
4. let $(z_0, \cdots, z_{\lfloor n/2 \rfloor -1}) = FFT(a_{odd})$
5. return $(x_0, \cdots, x_{n-1})$

     where   $x_k = y_k + \omega_n^k z_k$   $k=0,..,n-1$
```

$$T(n) = 2T(\tfrac{n}{2}) + O(n)$$

$$= O(n \log n)$$

$$a + bi$$

"real part" "imaginary part"

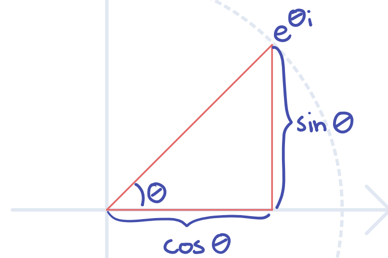
"imaginary i" s.t. $i^2 = -1$

$$(a+bi)(c+di) = ac + (bc+ad)i - bd$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Euler's formula:

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



Euler's identity:

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

$$\omega_n \overset{def}{=} e^{2\pi i/n}$$

$1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}$ are all nth roots of unity

$$F_n a = (p(1), p(\omega_n), p(\omega_n^2), \cdots, p(\omega_n^{n-1}))$$

$$F_n^\star a = (p(1), p(\omega_n^{n-1}), p(\omega_n^{n-2}), \cdots, p(\omega_n))$$

where $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^{n-1}$

Fact: $F_n^\star F_n a = F_n F_n^\star a = na$

Theorem: $F_n a$, $F_n^\star a$ in $O(n \log n)$

$p(x) = a_0 + a_1 x + \sim + a_{n-1} x^{n-1}$ $\qquad$ $q(x) = b_0 + b_1 x + \sim + a_{k-1} x^{k-1}$

Q: $O(n \log k)$ time?

$$p(x) = \underbrace{a_0 + a_1 x}_{k} + \underbrace{\sim}_{k} + \underbrace{a_{n-1} x^{n-1}}_{k}$$

$$\left(\underline{a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}}\right) + x^k \left(a_k + a_{k+1} x + \sim + a_{2k-1} x^{k-1}\right)$$

$$+ (\sim) + (\sim)$$

$$\frac{n}{k} \cdot k \log k = n \log k$$

$p(x)=a_0+a_1x+\cdots+a_{n-1}x^{n-1}$ $\quad$ $q(x)=b_0+b_1x+\cdots+a_{k-1}x^{k-1}$

Q: $O(n\log k)$ time?

Division?

$$p(x)=\underset{\uparrow}{a(x)}\,q(x)+b(x)$$

$$\underbrace{\underset{\overset{\uparrow}{n-2k}}{a(x)}\;\underset{\overset{\uparrow}{2k}}{\underbrace{q^2(x)}}}_{\deg k\qquad \deg \ell}+b(x)\,q(x)$$

$p(x,y)$ $q(x,y)$

$a_{00} + a_{10}x + a_{01}y + a_{11}xy$

Naive: $(k\ell)(k\ell)$

$O(k\ell \log k\ell)$

$y = x^k$