Today:

① Fibonacci

② Asymptotic Notation

③ Dividing, conquer. (Master Thm for analyzing recurrences)

Fib sequence:

0, 1, 1, 2, 3, 5, 8 ...

Algo1 (recursion)

$\boxed{\text{Fib}(n)}$

```
    if n ≤ 1 : return n.
    else : return Fib(n-1) + Fib(n-2)
```

will count flops (floating point ops)

$$T(n) = \begin{cases} 0 & \text{if } n \leq 1 \\ T(n-1) + T(n-2) + 1 & \text{if } n \geq 2. \end{cases}$$

$\approx 2^{5n}$ flops

$$F_n = F_{n-1} + F_{n-2} \geq 2 F_{n-2} \geq 2 \cdot 2 F_{n-4} \geq 2^{\frac{n}{2}}$$

Algo2 (iteration)

$\boxed{\text{FasterFib}(n)}$

$n-1$ flops

```
    if n ≤ 1 : return n.
    A ← 0
    B ← 1
    for i = 2 to n.
        tmp ← A+B
        A ← B
        B ← tmp
```

return B.

Algo 3. (fast matrix powering)

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} F_1 + F_0 \\ F_1 \end{bmatrix} = \begin{bmatrix} F_2 \\ F_1 \end{bmatrix}$$

A

$$A^n \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} \Leftarrow O(\log_2 n) \text{ flops}$$

$(A \cdots A)v$

$q^{71} \to \boxed{q^1}\ \boxed{q^2}\ \boxed{q^4} \cdots \boxed{q^{64}}$

$71: 64+4+2+1 = 1000111$ flops $\leq 2\log_2 n$

"repeated squaring"

Algo 4. $O(1)$ time

$\varphi = \frac{1+\sqrt{5}}{2}$, $\psi = \frac{1-\sqrt{5}}{2}$

$A^n v$

$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$: $\lambda_1 = \varphi$, $\lambda_2 = \psi$

$A = Q \Lambda Q^T$

$Q = \begin{bmatrix} \sqrt{\varphi} & -\sqrt{\psi} \\ -\sqrt{\psi} & \sqrt{\varphi} \end{bmatrix} \cdot \frac{1}{\sqrt[4]{5}}$

Q: "orthogonal" ($Q^T Q = I$)

$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$

$A^n = Q \Lambda^n Q^T$

$\Rightarrow F_n = \frac{1}{\sqrt{5}}(\varphi^n - \psi^n)$

exact formula

$F_n \approx \exp(c \cdot n)$

$\Rightarrow \Theta(n)$ digits

Runtime

Algo	Flops	Runtime
recurse	$\exp(cn)$	$\exp(cn)\cdot$small
iter.	n	n^2
matrix powering	$\log n$	$\leq \cancel{n^2 \log n}$ n^2

matrix mult:

when you square A^k.

entries are $\approx ck$ digits long.

Time $\leq c(1^2+2^2+\cdots n^2)$

$= O(n^2)$

Asymptotic Notation

- f, g are functions mapping $\mathbb{Z}^+$ to $\mathbb{Z}^+$ ← positive ints

(Big O): $f = O(g)$ if $\exists\, c < \infty$ s.t. $\forall n \; f(n) \leq c \cdot g(n)$

(little o) $f = o(g)$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

(Big Omega) $f = \Omega(g)$ if $g = O(f)$.

(little omega) $f = \omega(g)$ if $g = o(f)$

(Theta) $f = \Theta(g)$ if both $f = O(g)$

$f = \Omega(g)$

Analogies:

O	$\le$
o	$<$
Ω	$\ge$
ω	$>$
Θ	$=$

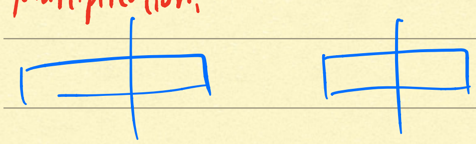
Ex:

① $f(n) = 3n^3 + n^2 + \log n$

$g(n) = n^3$.

$$\lim_{n \to \infty} \frac{3n^3 + n^2 + \log n}{n^3} = 3 \Rightarrow O$$

Multiplication.

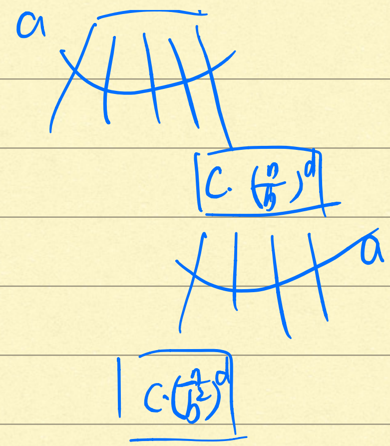


$$T(n) \le 3T\left(\frac{n}{2}\right) + cn$$

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

General: $T(n) \le aT\left(\frac{n}{b}\right) + c \cdot n^d$

	Time per level
$\boxed{c \cdot n^d}$	$c \cdot n^d$.



$C\cdot\frac{a}{b^d}\cdot n^d$

$C\cdot\left(\frac{a}{b^d}\right)^2\cdot n^d$

$\vdots$

$C\cdot\left(\frac{a}{b^d}\right)^k\cdot n^d$

$\frac{n}{b^k}\le 1 \qquad k\geq \log_b n$

$\boxed{\text{"Master theorm"}}$

Time: $C\cdot n^d\left[1+\left(\frac{a}{b^d}\right)+\dots\left(\frac{a}{b^d}\right)^k\right]$

$$= \begin{cases} O(n^d\cdot\log n) & \text{if } a=b^d \\ O(n^d) & \text{if } a<b^d \\ O(n^{\log_b a}) & \text{if } a>b^d \end{cases}$$

$$C\cdot n^d\cdot\frac{\left(\frac{a}{b^d}\right)^{\log_b n+1}-1}{\frac{a}{b^d}-1} = C\cdot\cancel{n^d}\cdot\frac{\frac{a}{b^d}\left(\frac{(b^{\log_b a})^{\log_b n}}{\cancel{(b^{\log_b n})^d}}\right)-1}{\frac{a}{b^d}-1}$$

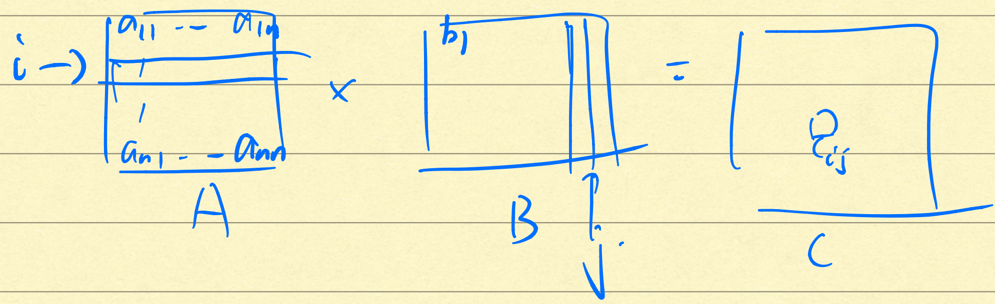
$$= C\cdot\frac{\frac{a}{b^d}\cdot n^{\log_b a}-n^d}{\frac{a}{b^d}-1}$$

case 1: $d > \log_b a \iff b^d > a \iff \frac{a}{b^d} < 1$

time $\leq \dfrac{c \cdot n^d - \cdots}{1 - \cdots} = O(n^d)$

case 2: $d < \log_b a$

Matrix Mult.



Algo: · init $C \leftarrow 0$'s

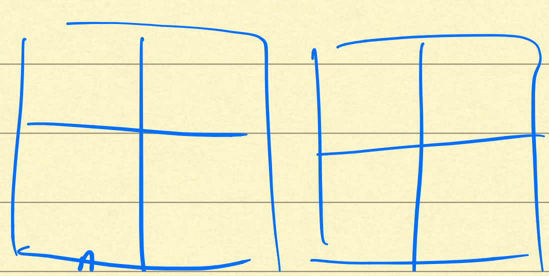
for $i = 1$ to n

for $j = 1$ to n.

for $k = 1$ to n

$c_{ij} = a_{ik} \cdot b_{kj}$

$= O(n^3)$



[illegible]