Lecture 2. Dec. 13. 2020 Sunday.

1. Fibonacci Sequence

(1) Alg 1. Recursion

$$Fib(n) = \begin{cases} n & n \le 1 \\ Fib(n-1) + Fib(n-2) & n > 1 \end{cases}$$

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

$T(n)$: flops number (floating point operations)

$$T(n) = \begin{cases} 0, & n \le 1 \\ T(n-1) + T(n-2) + \underbrace{1}_{\text{addition}} & n > 1 \end{cases} \qquad \boxed{2^n}$$

(2) Alg 2. Iteration

$$\text{FasterFib}(n) \begin{cases} n & n \le 1 \\ n > 1: & \underset{0}{\overset{A}{\uparrow}} \quad \underset{1}{\overset{B}{\uparrow}} \quad \text{for loop} \begin{cases} temp = A + B \\ A = B \\ B = \cancel{A+B}\ temp \end{cases} \end{cases}$$

$\boxed{n-1 \text{ flops}}$

(3) Alg 3. fast matrix powering

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} F_2 \\ F_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_1 \\ F_0 \end{pmatrix} \qquad \boxed{\log n}$$

$$X^n \begin{pmatrix} F_1 \\ F_0 \end{pmatrix} = \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}$$

$$q^{71} \quad \boxed{q^1} \to \boxed{q^2} \to \boxed{q^4} \to q^8 \to q^{16} \to q^{32} \to \boxed{q^{64}}$$

$71 = 64 + 4 + 2 + 1$ repeated squaring

(4) Alg 4.

$$A^n v \qquad A = Q \Lambda Q^T \quad (Q^T Q = I)$$

$$\begin{cases} Q: \text{orthogonal} \\ \Lambda: \text{diagonal} \end{cases}$$

$$A^n = Q \Lambda^n Q^T \qquad \varphi = \frac{1+\sqrt{5}}{2} \quad \psi = \frac{1-\sqrt{5}}{2}$$

$\cancel{R = \sqrt{\varphi}\ \sqrt{-\psi}} \to 0 \qquad \Omega = \sqrt{\varphi}\ \sqrt{-\psi}\ \ 1$

$$a \cdot \left(\frac{1}{\sqrt{-\psi}} \quad \frac{1}{\sqrt{\varphi}}\right) \frac{1}{\sqrt[4]{5}}$$

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

Run times for Alg 1~4:

$F_n \approx e^{cn} \Rightarrow \Theta(n)$ digits

Alg	Flops	Runtime
recursion	e^{cn}	e^{cn} · small
iter	n	n^2
matrix	$\log n$	$\leq n^2 \log n$ (n^2 actually)

$$1^2 + 2^2 + 4^2 + \cdots + n^2 = O(n^2)$$

2. Asymptotic Notation

f, g are functions mapping $\mathbb{Z}^+$ to $\mathbb{Z}^+$

$f = O(g)$: if $\exists\, c > 0$ s.t. $\forall n$ $f(n) \leq c \cdot g(n)$ Big O

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

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

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

$f = \Theta(g)$ if $f = O(g)$ + $f = \Omega(g)$

O	$\leq$
o	$<$
Ω	$\geq$
ω	$>$
Θ	$=$

Difference of O and o:

$$h(n) = \begin{cases} 2n^3 & , n \text{ odd} \\ n^3 & , n \text{ even} \end{cases}$$

$h(n) = O(n^3)$ ($c = 2$)

but $h(n) \neq o(n^3)$ (no limit)

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

$c \cdot n^d$ $c \cdot n^d$

a

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

a

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

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

$$c \cdot n^d \left(1 + \frac{a}{b^d} + \frac{a^2}{b^{2d}} + \cdots + \frac{a^k}{b^{kd}}\right) \quad k = \log_b n$$

① $\frac{a}{b^d} = 1$ $c \cdot n^d (\log_b n + 1)$ $O(n^d \log n)$

$\left(\frac{a}{\ldots}\right)^{\log_b n + 1} \cdot n^d$ ... $\log$ ...

② $\frac{a}{b^d} \neq 1$ $\quad c \cdot n^d \cdot \frac{\left(\frac{a}{b^d}\right)^{\dots} - 1}{\frac{a}{b^d} - 1} = c \cdot \cancel{n^d} \cdot \frac{1}{\frac{a}{b^d} - 1} \cdot \frac{\frac{a}{b^d} \cdot n^{\log_b a} - n^d}{\cancel{n^d}}$

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

case 1: $d > \log_b a$ $\quad$ time $= O(n^d)$

$b^d > a$

Master Theorem

case 2: $d < \log_b a$ $\quad$ time $= O(n^{\log_b a})$

$b^d < a$