Chap1 Analyzing Algorithms

基礎符號

> 要能默寫

- $O(g(n)) = \{f(n) : \exists c, n_0 > 0 \ such \ that \ 0 \leq f(n) \leq c \cdot g(n), \forall n \geq$
- $\Omega(g(n)) = \{f(n) : \exists c, n_0 > 0 \ such \ that \ 0 \leq c \cdot g(n) \leq f(n), \forall n \geq$
- $\Theta(g(n)) = \{f(n) : \exists c_1, c_2, n_0 > 0 \ such \ that \ 0 \leq c_1 \cdot g(n) \leq f(n) \leq$
    - $f(n) = \Theta(g(n)) \iff \lim\limits_{n \to \infty} \dfrac{f(n)}{g(n)} = L > 0$

- $o(g(n)) = \{f(n) : \forall c > 0, \exists n_0 > 0 \ such \ that \ 0 \leq f(n) < cg(n), \forall n$
    - $f(n) = o(g(n)) \iff \lim\limits_{n \to \infty} \dfrac{f(n)}{g(n)} = 0$
- $\omega(g(n)) = \{f(n) : \forall c > 0, \exists n_0 > 0 \ such \ that \ 0 \leq cg(n) < f(n), \forall n$
    - $f(n) = \omega(g(n)) \iff \lim\limits_{n \to \infty} \dfrac{f(n)}{g(n)} = \infty$

- Stirling's approximation
    - $n! = \Theta(n^{n+\frac{1}{2}} e^{-n})$

- # 證明f(n)是否為poly-bounded

1. 找到常數k使得$f(n) = O(n^k)$
2. $\lg(f(n)) = O(\lg n)$ (上式兩邊同取lg)

複雜度排行

Rank
2^{2^n}
$(n+1)!$
$n!$
2^n
$(\frac{3}{2})$
①$(\log n^{\log n})$
②$(\log n)!$
n^3
$n^2 = 4^{\log n}$
$n = 2^{\log n}$
$(\sqrt{2})^{\log n}$
④$2^{\sqrt{2\log n}}$
$\log^2 n$
$\log n$
$\log\log n$
③$n^{\frac{1}{\log n}}$

①$\log n^{\log n} = n^{\log\log n}$
$n^3 < n^{\log\log n} < 2^n$

$$\text{原理：兩邊取}log$$
$$\log n^{\log\log n} = \log\log n * \log n$$
$$\log 2^n = n\log 2 = n$$

②$\because Stirling's\ approximation$

$$\because n! \approx n^{n+\frac{1}{2}} \cdot e^{-n}$$
$$\therefore \log n! \approx \log n^{\log n+\frac{1}{2}} \cdot \sqrt{\log n} \cdot e^{-\log_e n}$$
$$= \log n^{\log n+\frac{1}{2}} \cdot \sqrt{\log n} \cdot n^{-\log_e e}$$
$$= \frac{\log n^{\log n} \cdot \sqrt{\log n}}{n}$$

③$\because n = 2^{\log n} \therefore n^{\frac{1}{\log n}} = (2^{\log n})^{\frac{1}{\log n}=2}$

④$\because 2 = n^{\frac{1}{\log n}} \therefore 2^{\sqrt{2\log n}} = (n^{\frac{1}{\log n}})^{\sqrt{2\log n}} = n^{(\sqrt{\frac{2}{\log n}})}$

- # asymptotic notation沒有三一律

  - $f(n)=O(g(n)) 與 f(n)=\Omega(g(n))$都不成立是可能的

  “Pasted image 20240817112734.png” could not be found.

ch1常考題型

grow rate 比大小 (ex.)

- 極限比較(可能要羅必達)
- 記得n!永遠在最後(如果沒有n^n的話)
- 代大數(不建議，很可能錯)

求time complexity

- Master theorem
- 直接從定義推導，找出常數($\exists, n_0 such\ that \ldots c$)
- 暴力展開
- Substitution method
  - 要先猜or用master theorem or用tree看出，再驗證
- 離散解法
  - $e.g.\ T(n) = 2T(\sqrt{n}) + \lg n$很常拿來嚇人

$\lg(n!) = \Theta(n \lg n)$

- 看到Θ就知道要證明O和Ω