

# Design and Analysis of Algorithms

## Tutorial 1: Asymptotic Notations and Recurrences



许可 kexu@nlsde.buaa.edu.cn

童咏昕 yxtong@buaa.edu.cn

北京航空航天大学 计算机学院

# 复习：渐近记号

---

近似上界：big-O

- $f(n) = O(g(n))$  : 存在常数  $c > 0$  和  $n_0$  , 使得对于  $n \geq n_0$  有  $f(n) \leq c \cdot g(n)$

近似下界：big-Ω

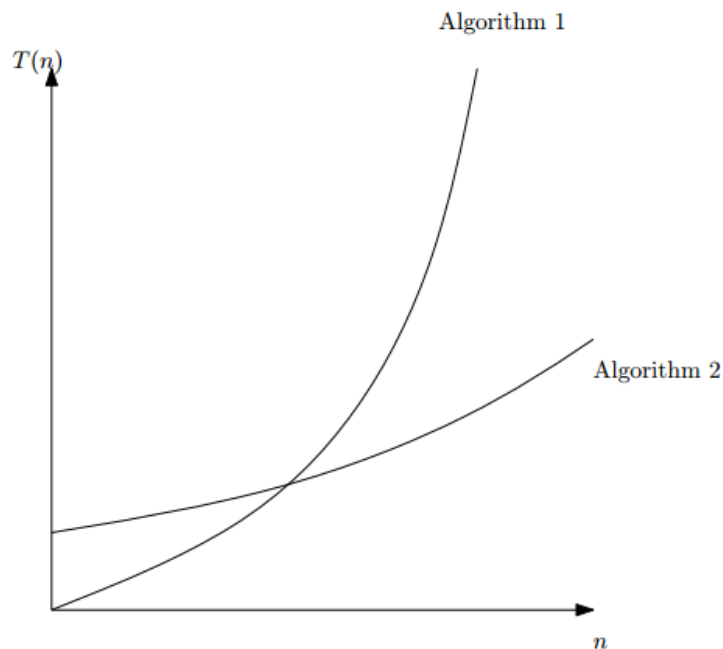
- $f(n) = \Omega(g(n))$  : 存在常数  $c > 0$  和  $n_0$  , 使得对于  $n \geq n_0$  有  $f(n) \geq c \cdot g(n)$ .

近似紧界：big-Θ

- $f(n) = \Theta(g(n))$  :  $f(n) = O(g(n))$  且  $f(n) = \Omega(g(n))$ .

# 复习：时间复杂度

例:



显然Algorithm 2更好

- Algorithm 1 的近似上界为 $O(n^3)$
- Algorithm 2 的近似上界为 $O(n^2)$
- 由于 $n^3$  增长速度更快，我们认为当 $n$ 增加时Algorithm 1 比Algorithm 2花费更多时间。

# 复习：指数函数基础

---

- 对于实数  $a \neq 0$ ,  $m$  以及  $n$ , 有如下性质:

$$a^0 = 1$$

$$a^1 = a$$

$$a^{-1} = \frac{1}{a}$$

$$(a^m)^n = (a^n)^m = a^{mn}$$

$$a^m a^n = a^{m+n}$$

$$a^{\frac{1}{n}} = \sqrt[n]{a}$$

# 复习：对数函数基础

---

- 对于实数  $a > 0$ ,  $b > 0$ ,  $c > 0$  和  $n$ :

$$a = b^{\log_b a}$$

$$\log_c(ab) = \log_c a + \log_c b$$

$$\log_b a^n = n \log_b a$$

$$\log_b a = \frac{\log_c a}{\log_c b}$$

$$\log_b \left( \frac{1}{a} \right) = -\log_b a$$

$$\log_b a = \frac{1}{\log_a b}$$

$$a^{\log_b n} = n^{\log_b a}$$

# 问题1

---

- 判断并证明下面给出的每一个表达式对  $(A, B)$  是否满足  $A=O(B)$ ,  $A=\Omega(B)$  或  $A=\Theta(B)$ . 注意: 一个表达式对可能满足多个关系, 请列出所有可能的结果。

(a)  $A = n^3 + n \log n; B = n^3 + n^2 \log n$

(b)  $A = \log \sqrt{n}; B = \sqrt{\log n}$

(c)  $A = n \log_3 n; B = n \log_4 n$

(d)  $A = 2^n; B = 2^{\frac{n}{2}}$

(e)  $A = \log(2^n); B = \log(3^n)$

## 问题2

---

- 假设  $T_1(n) = O(f(n))$ ,  $T_2(n) = O(f(n))$ . 下列公式是否正确? 请证明你的答案。

(a)  $T_1(n) + T_2(n) = O(f(n))$

(b)  $\frac{T_1(n)}{T_2(n)} = O(1)$

(c)  $T_1(n) = O(T_2(n))$

# 问题3

---

- 使用递归树 (recursion tree) 的方法计算下列递归函数  $T(n)$  的最紧渐近上界：

(a)

$$\begin{aligned} T(1) &= 1 \\ T(n) &= T\left(\frac{n}{2}\right) + n \quad \text{if } n > 1 \end{aligned}$$

(b)

$$\begin{aligned} T(1) &= T(2) = 1 \\ T(n) &= T(n-2) + 1 \quad \text{if } n > 2 \end{aligned}$$

(c)

$$\begin{aligned} T(1) &= 1 \\ T(n) &= T\left(\frac{n}{3}\right) + n \quad \text{if } n > 1 \end{aligned}$$



# 问题3

---

- 使用递归树 (recursion tree) 的方法计算下列递归函数  $T(n)$  的最紧渐近上界:

(d)

$$\begin{aligned} T(1) &= 1 \\ T(n) &= 4T\left(\frac{n}{2}\right) + n \quad \text{if } n > 1 \end{aligned}$$

(e)

$$\begin{aligned} T(1) &= 1 \\ T(n) &= 3T\left(\frac{n}{2}\right) + n^2 \quad \text{if } n > 1 \end{aligned}$$

(f)

$$\begin{aligned} T(1) &= 0, T(2) = 1 \\ T(n) &= T\left(\frac{n}{2}\right) + \log_2 n \quad \text{if } n > 2 \end{aligned}$$