

---

# 高级算法设计与分析

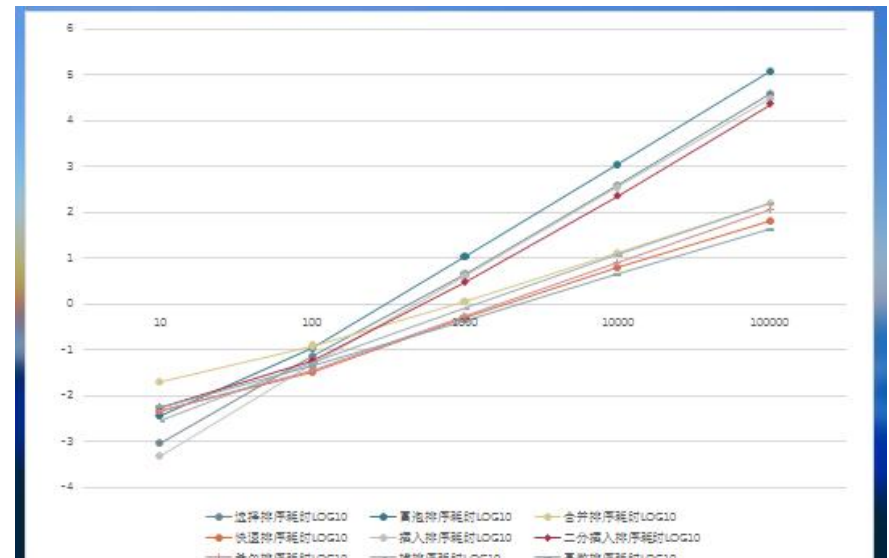
深圳大学计算机与软件学院

卢亚辉

# 增长次数：当n趋向无穷大时的情况

- 当输入规模足够大时，运行时间中的**低阶项**和**最高次项的常数系数**可以忽略；
- 输入规模足够大到只需考虑运行时间的**增长量级**时，研究的算法效率即为**渐近效率**。亦即，我们只关心从极限的角度考虑运行时间如何随输入规模的增长而增长。
- 渐近效率更高的算法，对大规模的输入是更好的。（对小规模的输入则不一定！）
- 算法运行时间的**阶**较简明地刻画了一个算法的效率，并作为不同算法进行比较的工具；

- 如何表达这种概念？



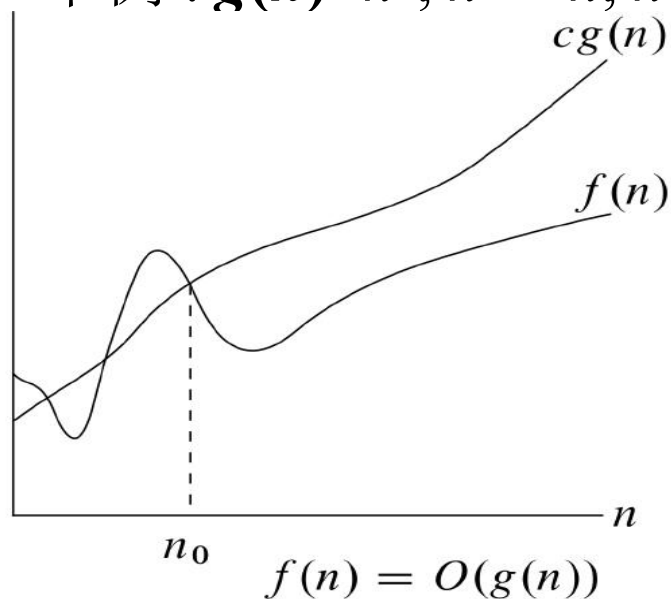
## 1.3 渐近记号 — $O$ 记号 (1)

### 渐近上界记号 $O$ (大 $O$ )

- 渐近地给出了一个函数在常量因子内的上界:

$$O(g(n)) = \{f(n) : \text{存在正常量 } c \text{ 和 } n_0, \text{ 使得对所有 } n \geq n_0, \\ \text{有 } 0 \leq f(n) \leq cg(n)\}$$

举例:  $g(n)=n^2, n^2 + n, n^2 + 1000n, 1000n^2 + 1000n, 5000n, 200n^{1.9}$



- $f(n) = \Theta(g(n))$  蕴含着  $f(n) = O(g(n))$
- $O$  可用于标识最坏情况运行时间

## 1.3 渐近记号 — $O$ 记号 (2)

---

练习：下列哪些关系是正确的？

- $1000000n^2 \in O(n^2)$
- $(n - 1)n / 2 \in O(n^2)$
- $n / 2 \in O(n^2)$
- $n^2 \in O(n)$
- $\lg n^2 \in O(\lg n)$
- $\lg^2 n \in O(\lg n)$

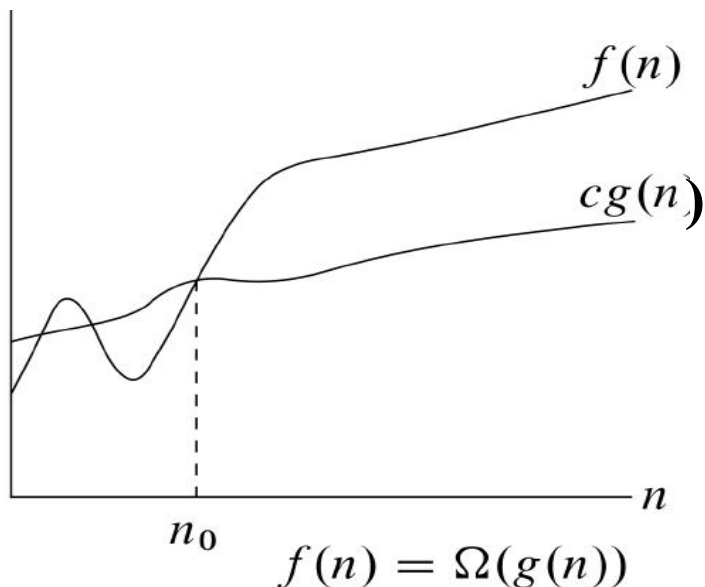
# 1.3 渐近记号 — $\Omega$ 记号 (1)

## 渐近下界记号 $\Omega$ (大 $\Omega$ )

- 渐近地给出了一个函数在常量因子内的下界:

$$\Omega(g(n)) = \{ f(n) : \text{存在正常量 } c \text{ 和 } n_0, \text{ 使得对所有 } n \geq n_0 \\ \text{有 } 0 \leq cg(n) \leq f(n) \text{ for all } n \geq n_0 \}$$

举例:  $g(n)=n^2, 1000n^2 + 1000n, 1000n^2 - 1000n, n^3, 200n^{2.1}$



- $f(n) = \Theta(g(n))$  蕴含着  $f(n) = \Omega(g(n))$
- $\Omega$  可用于标识最佳情况运行时间
- 插入排序的运行时间?
  - $\Omega(n)$  和  $O(n^2)$

## 1.3 渐近记号 — $\Omega$ 记号 (2)

---

练习：下列哪些关系是正确的？

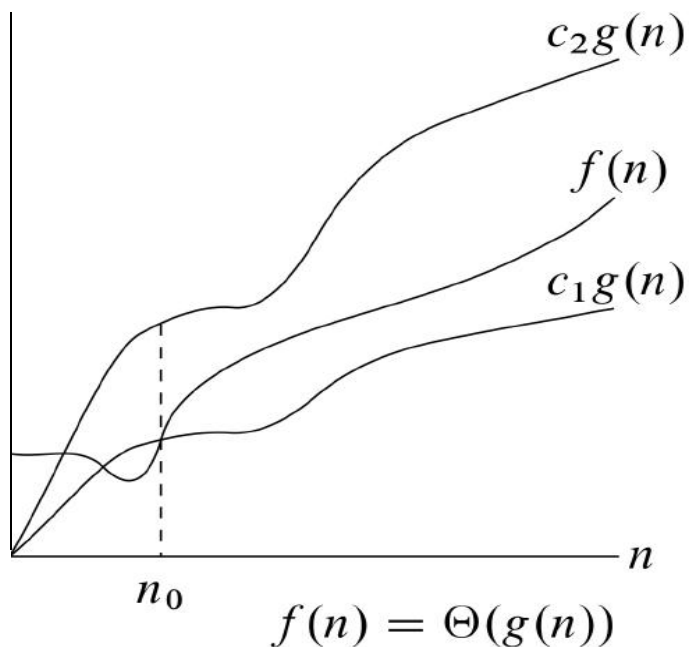
- $1000000n^2 \in \Omega(n^2)$
- $(n - 1)n / 2 \in \Omega(n^2)$
- $n / 2 \in \Omega(n^2)$
- $\lg n^2 \in \Omega(\lg n)$
- $\lg^2 n \in \Omega(\lg n)$
- $n^2 \in \Omega(n)$

# 1.3 渐近记号 — $\Theta$ 记号 (1)

## 渐近紧确界记号 $\Theta$

- 渐近地给出了一个函数的上界和下界:

$$\Theta(g(n)) = \{f(n) : \text{存在正常量 } c_1, c_2 \text{ 和 } n_0, \text{ 使得对所有 } n \geq n_0, \\ \text{有 } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\}$$



举例:  $n^2/2 - 2n \in \Theta(n^2)$ , 满足  
 $c_1 = 1/4$ ,  $c_2 = 1/2$  和  $n_0 = 8$ 。

## 1.3 渐近记号 — $\Theta$ 记号 (2)

---

- $f(n) = \Theta(g(n))$ 的确切意义是:  $f(n) \in \Theta(g(n))$ 。  
即对任一函数 $f(n)$ 满足上述条件时, 则 $f(n)$ 属于**集合** $\Theta(g(n))$ ;
- $\Theta(g(n))$ 的定义要求其每个元素**渐近非负** (当 $n$ 趋于无穷大时, $f(n)$ 和 $g(n)$ 都非负), 这也要求 $g(n)$ 本身是渐近非负的 (其他记号也是如此)。
- $\Theta(g(n))$ 中的所有函数具有相同的最高阶项。
- **原理:**  $f(n) \in \Theta(g(n))$ , 当且仅当  $f(n) \in O(g(n))$  且  $f(n) \in \Omega(g(n))$



## 1.3 渐近记号 — $\Theta$ 记号 (3)

---

形式化证明  $n^2/2 - 3n = \Theta(n^2)$

- 即确定正常数  $c_1, c_2$  和  $n_0$ , 使得对所有  $n \geq n_0$ , 有:

$$c_1 n^2 \leq \frac{1}{2} n^2 - 3n \leq c_2 n^2$$

- 用  $n^2$  除不等式得:

$$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2 .$$

- 右边不等式在  $n \geq 1$ ,  $c_2 \geq 1/2$  时成立, 左边不等式在  $n \geq 7$ ,  $c_1 \leq 1/14$  时成立, 选  $c_1 = 1/14$ ,  $c_2 = 1/2$ ,  $n_0 = 7$  时上式即可成立。

## 1.3 渐近记号 — $\Theta$ 记号 (4)

---

练习：下列哪些关系是正确的？

- $1000000n^2 \in \Theta(n^2)$
- $(n - 1)n / 2 \in \Theta(n^2)$
- $n / 2 \in \Theta(n^2)$
- $\lg n^2 \in \Theta(\lg n)$
- $\lg^2 n \in \Theta(\lg n)$
- $n^2 \in \Theta(n)$

# 1.3 渐近记号 — $o$ 记号

---

非渐近紧确上界记号 $o$  (小 $o$  记号)

- $o(g(n)) = \{ f(n) \mid \text{对于任何正常量 } c > 0, \text{ 存在常量 } n_0 > 0 \text{ 使得对所有 } n \geq n_0, \text{ 有 } 0 \leq f(n) < cg(n) \}$
- 举例:  $5n = o(n^2)$ ; 尽管  $5n^2 = O(n^2)$ , 但  $5n^2 \neq o(n^2)$
- 若  $f(n) = o(g(n))$ , 那么  $n$  当趋于无穷时,  $f(n)$  相对于  $g(n)$  变得微不足道

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

## 1.3 渐近记号 — $\omega$ 记号

---

非渐近紧确下界记号  $\omega$  (小  $\omega$  记号)

- $\omega(g(n)) = \{f(n) \mid \text{对于任何正常量 } c > 0, \text{ 存在常量 } n_0 > 0 \text{ 使得对所有 } n \geq n_0, \text{ 有 } 0 \leq cg(n) < f(n)\}$
- 举例:  $5n^2 = \omega(n)$ ; 尽管  $5n^2 = \Omega(n^2)$ , 但  $5n^2 \neq \omega(n^2)$ 。
- 若  $f(n) = \omega(g(n))$ , 那么  $n$  当趋于无穷时,  $f(n)$  相对于  $g(n)$  变得任意大了
- $f(n) \in \omega(g(n)) \iff \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$  且  $g(n) \in o(f(n))$

# 1.3 渐近记号 — 极限 (1)

---

- 用极限去定义渐近符号：

$$\text{若 } \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \begin{cases} c \ (c > 0 \text{ 是常量}), & \text{则 } f(n) = \Theta(g(n)) \\ 0, & \text{则 } f(n) = o(g(n)) \\ \infty, & \text{则 } f(n) = \omega(g(n)) \end{cases}$$

前两种情况意味着  $f(n) \in O(g(n))$

后两种情况意味着  $f(n) \in \Omega(g(n))$

第二种情况意味着  $f(n) \in \Theta(g(n))$

- 例1

$$\lim_{n \rightarrow \infty} \frac{10n^3 - 3n}{n^3} = \lim_{n \rightarrow \infty} \frac{10n^3}{n^3} - \lim_{n \rightarrow \infty} \frac{3n}{n^3} = 10, \quad \text{因此 } 10n^3 - 3n \in \Theta(n^3)$$

## 1.3 渐近记号 — 极限 (2)

---

**洛必达法则 (L'Hopital's Rule):**

$$\lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = \lim_{x \rightarrow \infty} \frac{f'(x)}{g'(x)}$$

$$\lim_{n \rightarrow \infty} \frac{n \ln n}{n^2} = \lim_{n \rightarrow \infty} \frac{(\ln n)'}{(n)'} = \lim_{n \rightarrow \infty} \frac{1/n}{1} = 0 \quad \Rightarrow \quad n \ln n \in o(n^2)$$

$$\lim_{n \rightarrow \infty} \frac{n^k}{2^n} = \lim_{n \rightarrow \infty} \frac{kn^{k-1}}{2^n \ln 2} = \lim_{n \rightarrow \infty} \frac{k(k-1)n^{k-2}}{2^n \ln^2 2} = \dots = \lim_{n \rightarrow \infty} \frac{k!}{2^n \ln^k 2} = 0$$

$$\Rightarrow n^k \in o(2^n)$$

$$\lim_{n \rightarrow \infty} \frac{\ln n}{n^a} = \lim_{n \rightarrow \infty} \frac{1}{an^a} = 0 \quad \Rightarrow \quad \ln n = o(n^a), \text{ 对任意 } a > 0$$

# 1.3 渐近记号 — 性质 (1)

---

## 传递性:

- $f(n) = O(g(n))$  且  $g(n) = O(h(n))$  蕴含  $f(n) = O(h(n))$
- $f(n) = \Omega(g(n))$  且  $g(n) = \Omega(h(n))$  蕴含  $f(n) = \Omega(h(n))$
- $f(n) = \Theta(g(n))$  且  $g(n) = \Theta(h(n))$  蕴含  $f(n) = \Theta(h(n))$
- $f(n) = o(g(n))$  且  $g(n) = o(h(n))$  蕴含  $f(n) = o(h(n))$
- $f(n) = \omega(g(n))$  且  $g(n) = \omega(h(n))$  蕴含  $f(n) = \omega(h(n))$

## 自反性:

- $f(n) = O(f(n))$
- $f(n) = \Omega(f(n))$       注意:  $o$  和  $\omega$  不具自反性
- $f(n) = \Theta(f(n))$

# 1.3 渐近记号 — 性质 (2)

---

对称性:

- $f(n) = \Theta(g(n))$  当且仅当  $g(n) = \Theta(f(n))$ .

转置对称性:

- $f(n) = O(g(n))$  当且仅当  $g(n) = \Omega(f(n))$ .
- $f(n) = o(g(n))$  当且仅当  $g(n) = \omega(f(n))$ .

渐近符号	算术符号
$O$	$\leq$
$\Omega$	$\geq$
$\Theta$	$=$
$o$	$<$
$\omega$	$>$



# 1.3 渐近记号 — 性质 (3)

---

单调性:

- 单调递增: 若  $m \leq n$  蕴含  $f(m) \leq f(n)$
- 单调递减: 若  $m \leq n$  蕴含  $f(m) \geq f(n)$
- 严格递增: 若  $m < n$  蕴含  $f(m) < f(n)$
- 严格递减: 若  $m < n$  蕴含  $f(m) > f(n)$

# 1.3 渐近记号 — 性质 (4)

---

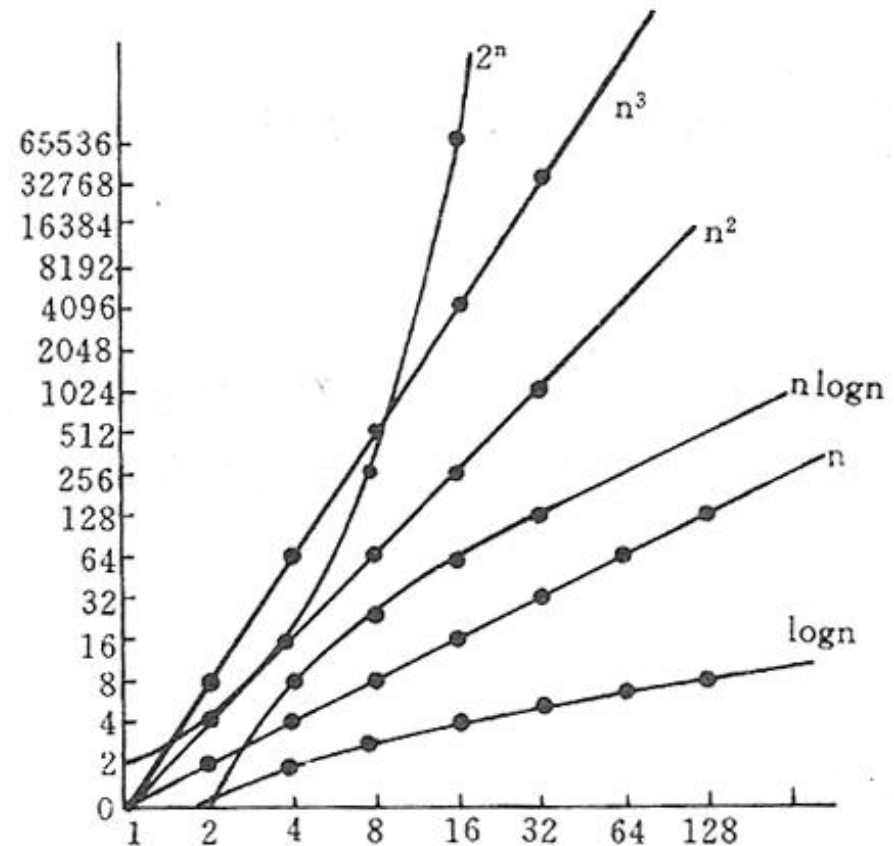
- 定理 如果  $t_1(n) \in O(g_1(n))$  并且  $t_2(n) \in O(g_2(n))$ , 则
$$t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$$
- 对于符号  $\Omega$  和  $\Theta$ , 该定理也成立。
- 该定理表明: 当算法由两个连续执行部分组成时, 该算法的整体效率由具有较大增长次数的那部分所决定, 即效率较差的部分决定

例子:  $0.5(n(n-1)) \in O(n^2)$ ;

$$0.5(n(n-1)) = 0.5n^2 - 0.5n; \quad 0.5n^2 \in O(n^2)$$

# 1.3 渐近记号 — 基本的效率类型 (1)

1	常量
$\log n$	对数
$n$	线性
$n \log n$	$n \log n$
$n^2$	平方
$n^3$	立方
$2^n$	指数
$n!$	阶乘



- 排在前面的复杂性类型是算法设计追求的目标。

# 1.3 渐近记号 — 基本的效率类型 (2)

$\log n$	$n$	$n \log n$	$n^2$	$n^3$	$2^n$
0	1	0	1	1	2
1	2	2	4	8	4
2	4	8	16	64	16
3	8	24	64	512	256
4	16	64	256	4096	65536
5	32	160	1024	32768	4294967296

$f(n) = o(g(n))$ 也可用偏序关系 $f(n) \prec g(n)$ 表示。

常用的复杂性类型有如下的偏序关系：

$$1 \prec \log \log n \prec \log n \prec \sqrt{n} \prec n^{3/4} \prec n \prec n \log n \prec n^2 \\ \prec 2^n \prec n! \prec 2^{n^2}$$

# 1.3 渐近记号 — 常用函数 (1)

---

## 向下取整与向上取整：

- 底：小于或等于实数 $x$ 的最大整数， $\lfloor x \rfloor$
- 顶：大于或等于实数 $x$ 的最小整数， $\lceil x \rceil$
- 对任意实数 $x$ ： $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
- 对任意整数 $n$ ： $\lceil n/2 \rceil + \lfloor n/2 \rfloor = n$
- 对任意实数 $x$ 和整数 $a \neq 0$ ,  $b \neq 0$ :
  - $\lceil \lceil x / a \rceil / b \rceil = \lceil x / ab \rceil$
  - $\lfloor \lfloor x / a \rfloor / b \rfloor = \lfloor x / ab \rfloor$
- 向下取整函数和向上取整函数是单调递增的；

# 1.3 渐近记号 — 常用函数 (1)

---

## 多项式：

- 给定一个非负整数 $d$ ， $n$ 的 $d$ 次多项式是具有如下形式的函数：

$$p(n) = \sum_{i=0}^d a_i n^i$$

其中常量 $a_i$ 是多项式的系数且 $a_d \neq 0$ 。

- 一个多项式是渐近正的，当且仅当 $a_d > 0$
- 对一个 $d$ 次的渐近正多项式 $p(n)$ ，有 $p(n) = \Theta(n^d)$ 。
- 对函数 $n^a$ ，当 $a \geq 0$ 时单调增， $a \leq 0$ 时单调减。
- 若对某个常量 $k$ ，使 $f(n) = O(n^k)$ ，则称函数 $f(n)$ 是**多项式有界的**。

# 1.3 渐近记号 — 常用函数 (1)

---

## 指数式：

- 对任意 $a > 0$ ， $m$ 和 $n$ ，有恒等式：

$$a^0 = 1 \text{ (并假设 } 0^0 = 1 \text{)}$$

$$a^1 = a$$

$$a^{-1} = 1/a$$

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

$$(a^m)^n = (a^n)^m$$

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

- 对所有 $n$ 和 $a \geq 1$ ，函数 $a^n$ 单调递增

# 1.3 渐近记号 — 常用函数 (1)

## 指数式（续）：

- 对任意常数 $a$ 和 $b$ ，且 $a > 1$ ，有： $\lim_{n \rightarrow \infty} \frac{n^b}{a^n} = 0$

据此可得： $n^b = o(a^n)$ ，即任意正的指数函数较任意的多项式增长更快。

- 对所有实数 $x$ ，有（ $e$ 表示2.71828， $!$ 表示阶乘函数）

$$e = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

- 由上，得到不等式： $e^x \geq 1 + x$ （ $x = 0$  时等号成立）
- 当 $x \rightarrow 0$ 时， $e^x \rightarrow 1 + x$ ，表示为： $e^x = 1 + x + \Theta(x^2)$
- 对所有 $x$ ，有  $\lim_{n \rightarrow \infty} (1 + \frac{x}{n})^n = e^x$



# 1.3 渐近记号 — 常用函数 (1)

---

## 对数

- 记号：
  - $\lg n = \log_2 n$  (以2为底的对数)
  - $\ln n = \log_e n$  (自然对数)
  - $\lg^k n = (\lg n)^k$  (取幂)
  - $\lg \lg n = \lg(\lg n)$  (复合)
  - $\lg n + k$  表示  $(\lg n) + k$  而不是  $\lg(n+k)$
- 对于  $n > 0$  和  $b > 1$ ,  $\log_b n$  严格递增

# 1.3 渐近记号 — 常用函数 (1)

---

## 对数式（续）：

- 对任意的实数  $a > 0$ ,  $b > 0$ ,  $c > 0$  和  $n$ :

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

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

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

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

$$\log_b(1/a) = -\log_b a,$$

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

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

- 因为改变对数的底只是相差一个常数因子，故当系数不重要时即写为  $\lg n$ ，类似于  $O$  记号

# 1.3 渐近记号 — 常用函数 (1)

---

## 对数式（续）：

- 当 $|x| < 1$ 时， $\ln(1+x)$ 的一个简单的级数展开为：

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} \dots$$

- 当 $x > -1$ 时，有不等式  $\frac{x}{1+x} \leq \ln(1+x) \leq x$  （ $x = 0$ 时等号成立）
- 对于某个常量 $k$ ，如果 $f(n) = O(\lg^k n)$ ，则称函数**多对数有界的**
- 将  $\lim_{n \rightarrow \infty} \frac{n^b}{a^n} = 0$  中用 $\lg n$ 替代 $n$ 和 $2a$ 替代 $a$ ，得：

$$\lim_{n \rightarrow \infty} \frac{\lg^b n}{2^{a \lg n}} = \lim_{n \rightarrow \infty} \frac{\lg^b n}{n^a} = 0 .$$

- 据此有： **$\lg^b n = o(n^a)$** ，即任意正的多项式函数较多项对数函数增长更快。

# 1.3 渐近记号 — 常用函数 (1)

## 阶乘:

- $n!$  定义为对所有整数  $n \geq 0$ :
$$n! = \begin{cases} 1 & \text{if } n = 0, \\ n \cdot (n-1)! & \text{if } n > 0. \end{cases}$$

因此,  $n! = 1 \cdot 2 \cdot \cdots \cdot n$

- $n!$  的一个弱上界为:  $n! \leq n^n$
- 斯特林(Stirling)近似公式:

公式给出了  $n!$  的更紧确的上下界

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

- 可以推出

$$n! > \left(\frac{n}{e}\right)^n \quad \lim_{n \rightarrow \infty} \frac{b^n}{(n/e)^n} = \lim_{n \rightarrow \infty} \left(\frac{b}{n/e}\right)^n = \lim_{n \rightarrow \infty} \left(\frac{be}{n}\right)^n = 0$$

➔  $b^n = o(n!)$ , 其中  $b > 1$

# 1.3 渐近记号 — 常用函数 (1)

---

## 阶乘 (续)

- 根据斯特林近似公式有:
  - $n! = o(n^n)$
  - $n! = \omega(2^n)$
  - $\lg(n!) = \Theta(n \lg n)$
- 对所有  $n$ , 下列界也成立:

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{a_n} \quad \text{其中} \quad \frac{1}{12n+1} < a_n < \frac{1}{12n}$$

# 1.3 渐近记号 — 常用函数 (1)

---

## 多重对数函数:

- 用记号 $\lg^*n$ 来表示多重对数, 其定义为:

$$\lg^* n = \min \{i \geq 0 : \lg^{(i)} n \leq 1\}$$

请注意区分 $\lg^{(i)}n$  与  $\lg^i n$  :

$$\lg^*2=1$$

$$\lg^*4=2$$

$$\lg^*16=3$$

$$\lg^*65536 = 4$$

$$\lg^*(2^{65536}) = 5 \quad (\text{很少能遇到} \lg^*n > 5 \text{的} n)$$

...

- $\lg^*n$ 是一种增长很慢的函数

# 附录：标准记号和常用函数

## 斐波那契数：

- 递归定义：（兔子繁殖问题）

$$F_0 = 0$$

$$F_1 = 1$$

$$F_i = F_{i-1} + F_{i-2}, \quad i \geq 2$$

- 斐波那契序列：0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

- 斐波那契数与黄金分割率

$$F_i = \frac{\phi^i - \hat{\phi}^i}{\sqrt{5}}, \quad \text{其中} \quad \phi = \frac{1 + \sqrt{5}}{2} = 1.61803\dots, \quad \hat{\phi} = \frac{1 - \sqrt{5}}{2} = -0.61803\dots$$

- 第*i*个斐波那契数等于和  $\phi^i / \sqrt{5}$  最近的整数，说明斐波那契数是呈**指数增长**的。

# 附录：有用的求和公式 (1)

---

$$\sum_{j=1}^n j = \frac{n(n+1)}{2} = \Theta(n^2)$$

$$\sum_{j=1}^n j^2 = \frac{n(n+1)(2n+1)}{6} = \Theta(n^3)$$

$$\sum_{j=0}^n c^j = \frac{c^{n+1}-1}{c-1} = \Theta(c^n), c \neq 1$$

$$\sum_{j=0}^n jc^j = \Theta(nc^n), c \neq 1$$

$$\sum_{j=0}^n 2^j = \frac{2^{n+1}-1}{2-1} = \Theta(2^n)$$

$$\sum_{j=0}^{\infty} c^j = \frac{1}{1-c} = \Theta(1), |c| < 1$$

$$\sum_{j=0}^{\infty} \frac{1}{2^j} = 2 = \Theta(1)$$

$$\sum_{j=0}^{\infty} jc^j = \frac{c}{(1-c)^2} = \Theta(1), |c| < 1$$



## 附录：有用的求和公式 (2)

---

$$\sum_{j=1}^n j^k = \Theta(n^{k+1}), k \geq 1$$

$$\sum_{j=1}^n \frac{1}{j} = \Theta(\log n)$$

$$\sum_{j=1}^n \log j = \Theta(n \log n)$$

$$\ln x = \int_1^x \frac{1}{t} dt$$

$$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n \quad n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \text{ (Stirling's formula)}$$

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots = 2.7182818\dots$$

# 例：确定是否所有元素相互不同

---

**ALGORITHM** *UniqueElements*( $A[0..n - 1]$ )

//Determines whether all the elements in a given array are distinct

//Input: An array  $A[0..n - 1]$

//Output: Returns “true” if all the elements in  $A$  are distinct

//           and “false” otherwise

**for**  $i \leftarrow 0$  **to**  $n - 2$  **do**

**for**  $j \leftarrow i + 1$  **to**  $n - 1$  **do**

**if**  $A[i] = A[j]$  **return false**

**return true**

# 例2：考虑元素唯一性问题的效率

---

- 输入规模：数组元素 $n$
- 基本操作：比较
- 除了和 $n$ 有关外，还取决于数组中是否有相同的元素，以及它们在数组中的位置
- 必须研究其最优，平均和最差效率
- 对最内层的循环，有两种类型的最差输入：不包括相同元素的数组，或者最后两个元素是唯一一对相同元素的数组。

## 例：考虑元素唯一性问题的效率

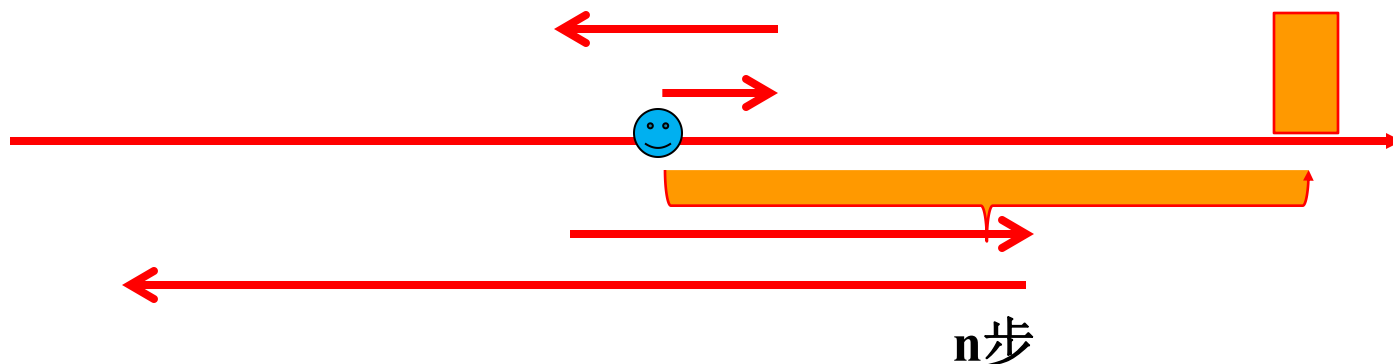
---

$$\begin{aligned} C_{\text{worst}}(n) &= \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} [(n-1) - (i+1) + 1] = \sum_{i=0}^{n-2} (n-1-i) \\ &= \sum_{i=0}^{n-2} (n-1) - \sum_{i=0}^{n-2} i = (n-1) \sum_{i=0}^{n-2} 1 - \frac{(n-2)(n-1)}{2} \\ &= (n-1)^2 - \frac{(n-2)(n-1)}{2} = \frac{(n-1)n}{2} \approx \frac{1}{2}n^2 \in \Theta(n^2) \end{aligned}$$

这个结果是完全可以预测的：在最坏的情况下，对于 $n$ 个元素的所有 $n(n-1)/2$ 对两两组合，该算法都要比较一遍。

## 例：找到我的门

假如你面前有一堵朝两个方向无限延伸的墙，墙上有一扇门，但你不知道门离你有多远，也不知道门在哪个方向。你只有走到门前才能看到它。假设从当前位置到门要走 $n$ （ $n$ 未知）步，请设计一个算法，使你最多走 $O(n)$ 步就能遇到门。



先向右走1步；  
如果途中没遇到门，再向左走2步；  
如果途中没遇到门，再向右走4步；  
如果途中没遇到门，再向左走8步；

.....

直到找到门为止。

# 解:

---

- 1.以向右的方向为正，左向的方向为负；
- 2.用 $f(i)$ 表示变向之前（第 $i$ 次行走）朝某个方向连续行走的距离，则 $f(i)=(-2)^{(i-1)}$ ，其中 $i=1, 2, \dots, k$ ； $f(1)=1, f(2)=-2, f(3)=4, f(4)=-8, \dots$
- 3.用 $d(i)$ 表示行走之后与起始位置的距离，则 $d(i)=f(1)+f(2)+\dots+f(i)$ ，这是个简单的等比级数求和。
- 4.假设在第 $k$ 次朝某个方向行走之后找到了门，于是有 $|d(k-2)| < n \leq |d(k)|$
- 5.进一步化简之后，有  $2^{(k-2)} < 3*n+1 < 2^k$
- 6.行走的总的步数为

$$S=|f(1)|+|f(2)|+\dots+|f(k)|=2^k-1 = 4*[2^{(k-2)}]-1 < 12*n+3$$

因此该算法的复杂度是 $O(n)$

- 
- P8 1-1
  - P30 3.1-2 3.1-3 3.1-4
  - P34 3.2-3
  - P35 3-2 3-3