01 五种渐进符号是什么意思

先懂符号 再填原题的五列 · 默认比较的函数最终为正

n 是输入规模，$f(n), g(n)$ 是随规模变化的量。 例如处理 n 个数的操作次数。渐进分析看 n 越来越大时的增长关系，允许忽略固定倍数与有限个小规模例外；不要求每个 n 都满足。

符号与读法	名称	直观含义	简单例子
O 大 O	渐进上界	f 最终不超过 g 的某个固定倍数	$n = O(n^2)$
o 小 o	严格低阶	f 相比 g 的占比趋于 0	$n = o(n^2)$
Ω 大欧米伽	渐进下界	f 最终不小于 g 的某个正常数倍	$n^2 = \Omega(n)$
ω 小欧米伽	严格高阶	f 相比 g 的倍数趋于无穷	$n^2 = \omega(n)$
Θ 西塔	渐进紧确界	f, g 同阶，彼此只差有界的倍数	$3n + 2 = \Theta(n)$

符号怎么读。 O 是字母，不是数字 0； $\to 0$ 表示趋近 0， $\to \infty$ 表示超过任意固定正常数。比较始终是“前面的 f 相对括号内的 g”。

大 O 大 Ω Θ 用某个常数夹住

$f = O(g)$： 存在常数 $C > 0$ 和阈值 n_0，对所有 $n \geq n_0$ ：

$$0 \leq f(n) \leq Cg(n).$$

$f = \Omega(g)$： 存在常数 $c > 0$ 和 n_0，对所有 $n \geq n_0$：

$$0 \leq cg(n) \leq f(n).$$

$f = \Theta(g)$： 存在 $c_1, c_2 > 0$ 和 n_0，对所有 $n \geq n_0$：

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n).$$

c, C, n_0 可以按函数选，但不能随 n 增长而改变； “存在”是找到一组合适常数即可。

完整例子： 当 $n \geq 1$，

$$3n \leq 3n + 2 \leq 5n.$$

取 $c_1 = 3, c_2 = 5, n_0 = 1$，得到 $3n + 2 = \Theta(n)$，同时也有 $O(n), \Omega(n)$。

宽松上界也成立。 上式还推出 $3n + 2 = O(n^2)$，但不是 $\Theta(n^2)$。因此大 O 本身不表示“最紧的阶”。

小 o 小 ω 要求严格的增长差距

$f = o(g)$： 对任意给定 $C > 0$，都能找到阈值 n_0，使所有 $n \geq n_0$ 时 $0 \leq f(n) < Cg(n)$。等价地：

$$f(n)/g(n) \to 0.$$

$f = \omega(g)$： 对任意给定 $c > 0$，都能找到阈值，使最终 $f(n) > cg(n) \geq 0$。等价地：

$$f(n)/g(n) \to \infty.$$

“任意”比“存在”强：小 o 要连 $0.01g$、$0.0001g$ 都最终超过 f；阈值可随给定常数变化。

小例子： $n/n^2 = 1/n \to 0$，故 $n = o(n^2)$；倒过来 $n^2/n = n \to \infty$，故 $n^2 = \omega(n)$。

数值较小不等于严格低阶。 $n < 2n$，但 $n/(2n) = 1/2$ ，不趋 0。因此 $n = \Theta(2n)$，不是 $o(2n)$。

五列为何可以同时为是

$$o \Rightarrow O, \qquad \omega \Rightarrow \Omega.$$

$$\Theta \Leftrightarrow O \text{ 与 } \Omega \text{ 同时成立.}$$

大 O 与大 Ω 不分别代表最坏、最好情况；应先确定比较的是最好、最坏还是期望运行时间函数。

变式与解析 令 $g = n$。$f = 1$ 时 $f/g \to 0$，成立 O, o； $f = 3n$ 时比值为 3，成立 O, Ω, Θ； $f = n^2$ 时比值趋无穷，成立 Ω, ω。写 $f = O(g)$ 是常用记法，表示 f 满足这一类上界，不是普通代数等式。

02 读题需要的公式与术语

对数 指数 阶乘 求和 · 只保留后面题目实际会用到的内容

先认清“固定常数”。 k, ε, c 固定，表示它们不随 n 变化；任意固定的 n^{100} 仍是多项式。$n^{\lg n}$ 的指数随 n 增长，不是固定次幂。后文“函数增长更快”表示数值随规模增长更快；若函数表示耗时，增长快意味着耗时增长快。

对数是指数的反运算

$$\log_b x = y \iff b^y = x.$$

$\lg x = \log_2 x$，$\ln x = \log_e x$；例如 $2^3 = 8$，故 $\lg 8 = 3$。这里 $e \approx 2.718$。

对数底数 $b > 0, b \neq 1$，真数为正；本资料用于比较大小时底数都大于 1。正数 x, y 满足：

$$\log_b (xy) = \log_b x + \log_b y,$$

$$\log_b (x^a) = a \log_b x, \qquad \log_b x = \frac{\ln x}{\ln b}.$$

换底只差常数。 $\ln n = (\ln 2)(\lg n)$，故同阶。**和不能拆开：** $\lg (x + y)$ 一般不等于 $\lg x + \lg y$。

括号不同 含义就不同

写法	含义或例子
$(\lg n)^2$	先取对数再平方，$n = 16$ 时为 16
$\lg \lg n$	连续取两次对数，$n = 16$ 时为 2
$\lg^* n$	取对数直到不超过 1 的次数，$\lg^* 16 = 3$
$n!$	$1 \cdot 2 \cdots n$，例如 $4! = 24$

指数先化简：

$$a^x / a^y = a^{x-y}, \qquad (a^x)^y = a^{xy}.$$

因此 $2^{2^{n+1}}$ 不能读成 2^{2n+1}。前者内层还含一次指数增长，称为双重指数。

求和就是把每层或每轮相加

$\sum_{i=1}^{m} i = 1 + 2 + \cdots + m$；下标告诉你从哪一项加到哪一项。

$$\sum_{i=1}^{m} i = \frac{m(m+1)}{2} = \Theta(m^2).$$

$$\sum_{i=1}^{m} i^2 = \frac{m(m+1)(2m+1)}{6} = \Theta(m^3).$$

所以选择排序的 $1 + \cdots + (n-1)$ 是平方级，减常数递推的平方和是三次级。

等比和： 每项是前项的固定 r 倍。

$$\sum_{i=0}^{h} r^i = \frac{1 - r^{h+1}}{1 - r} \quad (r \neq 1).$$

固定 $0 < r < 1$ 时，总和有界；$r = 1$ 时为 $h + 1$；固定 $r > 1$ 时为 $\Theta(r^h)$。这对应递归树的根部、各层、叶部主导。

调和和：

$$H_m = \sum_{i=1}^{m} \frac{1}{i} = \Theta(\lg m).$$

它用于 $n / \lg n$ 的递推变式与随机快排期望公式。

配对数 下标与空间

$$\binom{m}{2} = \frac{m(m-1)}{2}.$$

这是从 m 个不同位置中选两个、不计顺序的数量，也是长度 m 的数组可能拥有的最大逆序对数。

$A[0..j]$ 表示下标从 0 到 j、两端都包含；$\lfloor x \rfloor$ 向下取整，$\lceil x \rceil$ 向上取整，例如 $\lfloor 7/2 \rfloor = 3$、$\lceil 7/2 \rceil = 4$。$\max(a, b)$ 取较大值，$\min(a, b)$ 取较小值。

$O(1)$ 额外空间是只用固定数量的变量；$O(n)$ 表示额外存储至多随输入规模线性增长。时间与空间要分开分析。

变式与解析 $T(n) = T(n/2) + n$ 的层成本为 $n, n/2, n/4, \ldots$，和为 $\Theta(n)$；$T(n) = 2T(n/2) + n$ 每层总成本为 n，共有 $\Theta(\lg n)$ 层，所以是 $\Theta(n \lg n)$。关键在于算“整层”，而非只看一个节点。

03 渐进关系

原题与答案。 $k \geq 1, \varepsilon > 0, c > 1$ 是固定常数； $\lg = \log_2$ 。每一列分别判断，五种关系可以同时成立。

小题	$A(n)$	$B(n)$	O	o	Ω	ω	Θ
a	$(\lg n)^k$	n^ε	是	是	否	否	否
b	n^k	c^n	是	是	否	否	否
c	$\sqrt{n}$	$n^{\sin n}$	否	否	否	否	否
d	2^n	$2^{n/2}$	否	否	是	是	否
e	$n^{\lg c}$	$c^{\lg n}$	是	否	是	否	是
f	$\lg(n!)$	$\lg(n^n)$	是	否	是	否	是

一步选方法 先算比值

设最终 $A, B > 0$，记 $R = A/B$。

比值的表现	成立的关系
$R \to 0$	O, o
$R \to L,\ 0 < L < \infty$	O, Ω, Θ
$R \to \infty$	Ω, ω
无极限	检查最终上下界

a 对数幂与正幂。 令 $t = \ln n$，比值变成常数倍 $t^k/e^{\varepsilon t} \to 0$。固定次幂对数慢于任意正次幂。

b 多项式与指数。 对比值取对数：

$$\ln(n^k/c^n) = k \ln n - n \ln c \to -\infty.$$

所以原比值趋于 0；指数中的 n 压过 $\ln n$。

d 同底数先减指数。

$$2^n/2^{n/2} = 2^{n/2} \to \infty.$$

e 统一写成底数 2。

$$c^{\lg n} = 2^{(\lg c)(\lg n)} = n^{\lg c}.$$

两式恒等，比值为 1。

振荡题 看子序列而非平均值

c 比值为 $R = n^{1/2 - \sin n}$。

沿 $\sin n > 3/4$ 的无穷子序列：

$$0 < R \leq n^{-1/4} \to 0,$$

因此没有最终正下界，排除 Ω, ω, Θ。

沿 $\sin n < 1/4$ 的无穷子序列：

$$R \geq n^{1/4} \to \infty,$$

因此没有最终常数上界，排除 O, o, Θ。

整数规模也成立：分别取最接近 $\pi/2 + 2\pi j$、$3\pi/2 + 2\pi j$ 的整数，误差不超过 $1/2$，正弦分别至少为 $\cos(1/2) > 3/4$、至多为 $-\cos(1/2) < 1/4$。

f 先把分母改写为 $n \lg n$。

$$\frac{n}{2} \lg \frac{n}{2} \leq \lg(n!) \leq n \lg n.$$

足够大时左端不小于 $\frac{1}{4} n \lg n$，已足以夹出 Θ；不必为了同阶结论求精确极限。

定义落笔。 O 是最终 $A \leq CB$；Ω 是最终 $A \geq cB$；Θ 同时满足两者。o, ω 分别要求比值趋于 0、无穷。

变式与解析 $A = n(2 + \sin n), B = n$ 的比值没有极限，但始终满足 $1 \leq A/B \leq 3$，所以 O, Ω, Θ 成立，o, ω 不成立。**技巧：无极限时先找界；有振荡不等于五列全否。**

04 三十个函数完整排序

1-2 · Day2 第 32 页 · 24 个同阶类包含全部 30 项

读取顺序：先左列 1 至 12，再右列 13 至 24，由快到慢。 同格内互为 Θ；相邻两组前者是后者的 ω。保留原题每个函数，不用“大类名称”代替答案。

组	由快到慢的原式	组	接左列继续向下
1	$2^{2^{n+1}}$	13	$n \lg n,\ \lg(n!)$
2	2^{2^n}	14	$2^{\lg n},\ n$
3	$(n+1)!$	15	$(\sqrt{2})^{\lg n}$
4	$n!$	16	$2^{\sqrt{2\lg n}}$
5	e^n	17	$(\lg n)^2$
6	$n2^n$	18	$\ln n$
7	2^n	19	$\sqrt{\lg n}$
8	$(3/2)^n$	20	$\ln \ln n$
9	$(\lg n)^{\lg n},\ n^{\lg\lg n}$	21	$2^{\lg^* n}$
10	$(\lg n)!$	22	$\lg^* n,\ \lg^*(\lg n)$
11	n^3	23	$\lg(\lg^* n)$
12	$4^{\lg n},\ n^2$	24	$1,\ n^{1/\lg n}$

六组同阶的原因

$$n^{1/\lg n}=2,\qquad 2^{\lg n}=n.$$
$$4^{\lg n}=n^2,\qquad (\sqrt{2})^{\lg n}=\sqrt{n}.$$

后一个等式用于定位第 15 组，原题中没有单独的 $\sqrt{n}$ 项需要合并。

$$\lg((\lg n)^{\lg n})=(\lg n)\lg\lg n.$$
$$\lg(n^{\lg\lg n})=(\lg\lg n)\lg n.$$

取对数相等，原式恒等，所以第 9 组可合并。

$$\lg(n!)=\Theta(n\lg n).$$
$$\lg^*(\lg n)=\lg^* n-1\quad (n\text{ 足够大}).$$

迭代次数只差 1，且趋于无穷，故第 22 组同阶。六个双项组加十八个单项组，共 $6\times2+18=30$ 项。

排序技巧与三个易错位置

先化简，再用比值排邻居。 对正函数取同底数对数，有助于比较指数、阶乘；取对数后的“同阶”不能直接当作原函数同阶。

阶乘差一项也不能合并。

$$(n+1)!/n!=n+1\to\infty.$$

$n2^n$ 排在 e^n 后、2^n 前。

$$\frac{e^n}{n2^n}=\frac{(e/2)^n}{n}\to\infty.$$
$$\frac{n2^n}{2^n}=n\to\infty.$$

双重指数按括号读。

$$\frac{2^{2^{n+1}}}{2^{2^n}}=2^{2^n}\to\infty.$$

第 1 项不是 2^{2n+1}；后者只是底数为 4 的普通指数。最外层的幂不能读丢。

验收答案。 方向从快到慢；六对同阶项合并；30 个原式逐项出现一次。

变式与解析 $\lg(2^n)=n$、$\lg(3^n)=n\lg 3$ 同阶，但 $3^n/2^n=(3/2)^n\to\infty$，原函数不属于同一 Θ 类。判断原函数同阶仍要回到**原函数的比值**。

05 函数排序的疑难比较

1-2 · 四个关键位置的推导 · $\prec$ 表示严格慢于

先换元 $x = \lg n$。 多项式 n^c 变成 2^{cx}；$(\lg n)^{\lg n}$ 变成 x^x。把复杂外观换成同一变量，才容易比较。

对数的阶乘为何快于多项式

要证 $(\lg n)! = \omega(n^c)$，其中 $c > 0$ 固定。令 $x = \lg n$，Stirling 量级给出：

$$\lg(x!) = x\lg x - \Theta(x).$$

而 $\lg(n^c) = cx$，故：

$$\lg \frac{x!}{2^{cx}} = x\lg x - \Theta(x) - cx \to \infty.$$

关键分析： $\lg x$ 最终超过任意固定常数，因此 $x\lg x$ 压过所有常数倍 x。即使把 c 改成 100，结论也不变。

若 $\lg n$ 非整数，按 $\lfloor \lg n \rfloor!$ 理解，不改变这里的严格排序。

它为何又慢于准多项式

另一边是 $(\lg n)^{\lg n} = x^x$。对于整数 x：

$$\frac{x!}{x^x} = \prod_{i=1}^{x} \frac{i}{x} \le 2^{-\lfloor x/2 \rfloor} \to 0.$$

因为前一半因子均不超过 $1/2$，其余因子不超过 1。无需完整 Stirling 公式也能证明严格更慢。

$$n^c \prec (\lg n)! \prec n^{\lg\lg n}$$

解题技巧： 阶乘与“每项都取最大值”的幂比较，用乘积拆分；阶乘与固定多项式比较，用取对数。

中间函数为何夹在对数幂与正幂之间

原式 $F = 2^{\sqrt{2\lg n}} = 2^{\sqrt{2x}}$。

与 $(\lg n)^c = x^c$ 比较：

$$\lg(F/x^c) = \sqrt{2x} - c\lg x \to \infty.$$

与 $n^\varepsilon = 2^{\varepsilon x}$ 比较：

$$\lg(F/n^\varepsilon) = \sqrt{2x} - \varepsilon x \to -\infty.$$

所以，对任意固定 $c, \varepsilon > 0$：

$$(\lg n)^c \prec F \prec n^\varepsilon.$$

关键分析： 根号 x 快于 $\lg x$、慢于 x；比较被降成了熟悉的三个函数。

迭代对数的先后

$\lg^* n$ 是反复取 $\lg$ 直到结果不超过 1 的次数；$(\lg n)^k$ 是普通的 k 次幂。

令 $t = \lg^* n \to \infty$，则：

$$\lg t \prec t \prec 2^t.$$

故原题第 23、22、21 组依次更快。还要比较 2^t 与 $\ln\ln n$：后者远大于它。

可令 $a_0 = 1, a_{r+1} = 2^{a_r}$。当 $\lg^* n = t$ 时，$n > a_{t-1}$，于是 $\lg\lg n > a_{t-3}$；而 $a_{t-3}/2^t \to \infty$，得到第 20 组位于第 21 组之前。

比较时可直接使用的三条规则

底数相同，减指数；指数相同，比底数。 $a^n/b^n = (a/b)^n$。**幂与对数幂，令** $x = \ln n$。 比较 n^ε 与 $(\ln n)^k$ 变成 $e^{\varepsilon x}$ 与 x^k。**同阶看原比值。** 证明比值夹在两个正常数之间，比说“它们看起来差不多”更可靠。

变式与解析 排序 $n^{100}, (\lg n)!, n^{\lg\lg n}, 2^n$，由快到慢为 $2^n \succ n^{\lg\lg n} \succ (\lg n)! \succ n^{100}$。其中 $\lg(n^{\lg\lg n}) = (\lg n)\lg\lg n = o(n)$，所以准多项式仍慢于 2^n。

06 证明与反例

1-3 · Day2 第 33 页 · 每个反例都要验证前提和结论

a 上界能否倒过来

命题： $f = O(g) \Rightarrow g = O(f)$?

答案：错。 取 $f = n, g = n^2$。

前提：$f/g = 1/n \to 0$，所以 $f = O(g)$。

结论：$g/f = n \to \infty$，不能被一个常数压住，所以 $g \neq O(f)$。

关键分析： O 只限制“不能快太多”，允许严格更慢；倒过来就失去上界。

可改成的真命题：

$$f = \Theta(g) \Leftrightarrow f = O(g) \text{ 且 } g = O(f).$$

b 和由小项还是大项决定

命题： $f + g = \Theta(\min(f, g))$?

答案：错。 取 $f = n^2, g = n$，和为 $\Theta(n^2)$，最小项却为 n。

最终非负时，正确的夹逼是：

$$\max(f, g) \le f + g \le 2\max(f, g).$$

因此：

$$f + g = \Theta(\max(f, g)).$$

关键分析： 用两项之和替换较大项，只差至多常数 2；换成较小项则可能差一个无界因子。

c 取对数能否保留大 O

命题： 已知 $f = O(g)$，且足够大时 $f \ge 1$，$\lg g \ge 1$，是否有 $\lg f = O(\lg g)$?

答案：对。 由前提取 $C \ge 1$，使最终 $f \le Cg$。逐步写：

$$\begin{aligned} 0 \le \lg f &\le \lg C + \lg g \\ &\le (1 + \lg C)\lg g. \end{aligned}$$

两处条件各有用途： $f \ge 1$ 保证 $\lg f \ge 0$；$\lg g \ge 1$ 允许把加法常数 $\lg C$ 吸收到常数倍 $\lg g$ 中。取新常数 $C' = 1 + \lg C$ 即满足定义。

d 指数能否保留大 O

命题： $f = O(g) \Rightarrow 2^f = O(2^g)$?

答案：错。 取 $f = 2n, g = n$。$f/g = 2$，前提成立；但：

$$\frac{2^f}{2^g} = 2^{f-g} = 2^n \to \infty.$$

关键分析： $2^f/2^g$ 由差 $f - g$ 决定，而非比值 f/g。

若最终 $f \le g + C$，则：

$$2^f \le 2^C 2^g,$$

这是足以推出结论的更强条件。

答题时怎样选最短证明

判断为真： 展开前提的定义不等式，沿目标方向变形，明确新常数与共同阈值。**判断为假：** 先试 $1, n, n^2$；涉及指数变换优先试 $f = 2g$。反例只需一组，但必须同时满足原题所有前提。

容易混淆的变式	判定与最短依据
$f = o(g) \Rightarrow f = O(g)$	对，比值趋 0，所以最终有界
$f = O(g) \Rightarrow \lg f = \Theta(\lg g)$	错，$f = 1, g = n$ 时左边为 0
$f = O(g) \Rightarrow f^2 = O(g^2)$	最终非负时对，将 $f \le Cg$ 平方

变式与解析 $f = \Theta(g)$ 仍不能推出 $3^f = \Theta(3^g)$：取 $f = 2n, g = n$，比值为 3^n。若最终 $|f - g| \le C$，则 $3^{-C} \le 3^f/3^g \le 3^C$，这时可推出同阶。

07 选择排序

1-4 · Day2 第 34 页 · 每轮选最大值放到前缀末尾

原题。 写伪代码、循环不变式，分析最好与最坏时间。以 $[4,1,3,2]$ 为例，数组下标从 0 开始；蓝色后缀表示已经确定的位置。

前置概念。 循环不变式是每轮开始都成立、并能被本轮保持的性质；用“初始化、保持、终止”证明。前缀从数组开头起，后缀在末尾；多重集合相同表示每个值及其出现次数都不变。稳定性指相等键的原先次序保持不变。

三轮完整轨迹

初始	4	1	3	2
轮1	2	1	3	4
轮2	2	1	3	4
轮3	1	2	3	4

第 1 轮： 未处理区为 $[0,3]$，最大值 4 在位置 0；与位置 3 的 2 交换。

第 2 轮： 未处理区为 $[0,2]$，最大值 3 已在位置 2；数组不变，比较仍做了 2 次。

第 3 轮： 未处理区为 $[0,1]$，最大值 2 与 1 交换，整个数组有序。

可直接写出的伪代码

```
for end = n-1 down to 1:
    p = 0
    for i = 1 to end:
        if A[i] > A[p]:
            p = i
    swap A[p], A[end]
```

内层不变式： 检查 i 前，p 指向 $A[0..i-1]$ 中一个最大值；检查后扩展到 $A[0..i]$。

正确性证明的三步

不变式： 每轮开始时，$A[end+1..n-1]$ 有序，恰含全体最大的若干元素；前缀每个值均不大于后缀每个值；全数组多重集合不变。

初始化： 后缀为空，不变式成立。

保持： 把前缀最大值放到 end。它大于等于余下前缀，又不大于旧后缀任何值，所以新后缀仍有序；交换不增删元素。

终止： 最后仅位置 0 未单独确定，它不大于后缀各值；故整个数组有序且元素正确。

复杂度的得分式

$$\begin{aligned} C(n) &= (n-1)+(n-2)+\cdots+1 \\ &= \frac{n(n-1)}{2}. \end{aligned}$$

比较次数由循环范围决定，与初始顺序无关。因此最好、最坏均为 $\Theta(n^2)$；工作空间 $O(1)$。

外层有 $n-1$ 轮，按代码执行 $n-1$ 次交换语句，可能包含自己与自己交换。不能把交换次数当成比较次数。

稳定性反例

$[2_a, 2_b, 1]$ 中按代码选首次最大值 2_a，首轮得到 $[1, 2_b, 2_a]$，相等键次序反转，故不稳定。

变式与解析 输入 $[3,1,2]$：$[2,1,3] \to [1,2,3]$，键比较 $2+1=3$ 次。输入本来是 $[1,2,3]$ 也比较 3 次。**技巧：看循环是否提前停止，再判断有序输入是否更快。**

08 七个递推式与主定理

4-1 · Day3 第 39 页 · 基础规模 $n \le 2$ 的代价为正常数

原题七问的紧确答案。 分界指数 $p = \log_b a$；取整不改变以下渐近结论。

前置概念。 $T(n)$ 表示规模为 n 的总成本；$aT(n/b)$ 是 a 次子问题的成本，$f(n)$ 是当前调用的非递归工作。递归树根节点代表原问题、孩子代表递归调用、叶子代表不再递归的基础情况。二分查找只调用一半，$a = 1$；归并排序调用两半，$a = 2$。

题	$T(n)$ 的递推右端	关键计算	答案
a	$2T(n/2) + n^4$	$p = 1$，正则比 $1/8$	$\Theta(n^4)$
b	$T(7n/10) + n$	层成本 $n(7/10)^i$	$\Theta(n)$
c	$16T(n/4) + n^2$	$p = 2$，每层 n^2	$\Theta(n^2 \lg n)$
d	$7T(n/3) + n^2$	$p = \log_3 7 < 2$，正则比 $7/9$	$\Theta(n^2)$
e	$7T(n/2) + n^2$	$p = \lg 7 > 2$，叶部主导	$\Theta(n^{\lg 7})$
f	$2T(n/4) + \sqrt{n}$	$p = 1/2$，每层 $\sqrt{n}$	$\Theta(\sqrt{n} \lg n)$
g	$T(n-2) + n^2$	线性项数的平方和	$\Theta(n^3)$

主定理必须写的条件

对 $T(n) = aT(n/b) + f(n)$，$a \ge 1, b > 1$ 是常数，子问题等大。设 $p = \log_b a$。

第一种： 存在固定 $\varepsilon > 0$，

$$f(n) = O(n^{p-\varepsilon})$$

则 $T(n) = \Theta(n^p)$。差距必须是固定幂次。

第二种： $f(n) = \Theta(n^p)$，

$$T(n) = \Theta(n^p \lg n).$$

第三种： $f(n) = \Omega(n^{p+\varepsilon})$，还要验证正则条件

$$af(n/b) \le cf(n), \quad c < 1,$$

才得 $T(n) = \Theta(f(n))$。

例 a： $2(n/2)^4 = n^4/8$，正则常数可取 $1/8$。

例 d： $7(n/3)^2 = 7n^2/9$，正则常数可取 $7/9$。

递归树为什么得到这三种情况

若 $f(n) = n^d$，第 i 层有 a^i 个规模为 n/b^i 的节点，整层成本是：

$$W_i = a^i (n/b^i)^d = n^d (a/b^d)^i.$$

a：$n^4 \to n^4/8 \to n^4/64$

逐层缩小，根部主导

c：$n^2 \to n^2 \to n^2$

各层相同，乘上对数层数

e：$n^2 \to 7n^2/4 \to 49n^2/16$

逐层增大，叶部主导

公比 $a/b^d < 1$： 逐层衰减，等比和由根层决定。

公比 $= 1$： 各层相同，乘 $\Theta(\lg n)$ 层。

公比 > 1： 逐层增大，叶部主导，叶子数为 $\Theta(n^{\log_b a})$。

别把层数直接当时间。 b 虽然有 $\Theta(\lg n)$ 层，但每层越来越便宜，总和仍是 $\Theta(n)$。

别把 g 套进主定理。 $n - 2$ 是减常数，不能写成固定 n/b；要展开求和。

变式与解析 $4T(n/2) + n^2$ 中 $p = 2$，每层成本 n^2，所以 $\Theta(n^2 \lg n)$。若改成 $4T(n/2) + n$，n 比分界 n^2 小一固定幂次，答案变为 $\Theta(n^2)$。

09 递推式的关键展开与变式

4-1 · 平方和 等比和 换元与非等大分治

g 减去 2 为何是三次而非平方

$$T(n) = n^2 + (n-2)^2 + (n-4)^2 + \cdots + \Theta(1).$$

上界： 至多 $O(n)$ 项，每项不超过 n^2，故 $O(n^3)$。

下界： 前 $\lfloor n/4 \rfloor$ 项各不小于 $(n/2)^2$，故：

$$T(n) \geq \lfloor n/4 \rfloor (n/2)^2 = \Omega(n^3).$$

上下界同阶，得到 $\Theta(n^3)$。只写“项数乘最大项”一般只证明了上界。

b 单链等比和

$$T(n) = n + \frac{7}{10}n + (\frac{7}{10})^2 n + \cdots.$$

根层给 $\Omega(n)$，整棵树不超过：

$$n \sum_{i \geq 0} (7/10)^i + O(1) = \frac{10}{3}n + O(1).$$

故 $\Theta(n)$。**深度是对数，成本是线性，二者不矛盾。**

f 每层相同要乘层数

第 i 层有 2^i 个节点，每个节点成本 $\sqrt{n/4^i}$：

$$2^i \sqrt{n/4^i} = \sqrt{n}.$$

深度为 $\log_4 n$；若 $n = 4^m, T(1) = 1$，则 $T(n) = (m+1)2^m$，即 $\Theta(\sqrt{n} \lg n)$。

多一个对数 换元更直观

对 $T(n) = 2T(n/2) + n \lg n$，令 $n = 2^m$、$U_m = T(2^m)/2^m$：

$$U_m = U_{m-1} + m.$$

U_m 是等差和 $\Theta(m^2)$，乘回 n 得：

$$T(n) = \Theta(n \lg^2 n).$$

关键分析： $n \lg n$ 既不是 $\Theta(n)$，也未比 n 大固定幂次，不能硬套基本三种情况。

若非递归项换成 $n/\lg n$，则：

$$U_m = U_{m-1} + 1/m,$$

调和和为 $\Theta(\lg m)$，故：

$$T(n) = \Theta(n \lg \lg n).$$

规模开根号的层数

对 $T(n) = T(\sqrt{n}) + 1$，递归 h 次后规模为 $n^{1/2^h}$。到常数规模要求：

$$\frac{\lg n}{2^h} = \Theta(1) \Rightarrow h = \Theta(\lg \lg n).$$

每层常数成本，所以总时间同阶。

子问题不等大

$T(n) = T(n/3) + T(2n/3) + n$：每层规模总和不超过 n，最深 $O(\lg n)$；前 $\Theta(\lg n)$ 层无叶子，每层成本恰为 n。上下界合得 $\Theta(n \lg n)$。

减常数不等于只剩多项式

$T(n) = 2T(n-1) + 1$ 每展开一层，子问题数量翻倍：$T(n) = 2^h T(n-h) + (2^h - 1)$。取 $h = n-1$ 得 $\Theta(2^n)$。决定成本的不只是路径长度，还有每层节点数量。

变式与解析 $T(n) = T(n-3) + n$ 有 $\Theta(n)$ 个等差项，总和 $\Theta(n^2)$；$T(n) = T(n-3) + 1$ 则为 $\Theta(n)$。答题先数项，再对每项成本求和，最后给出同阶下界。

10 两数之和

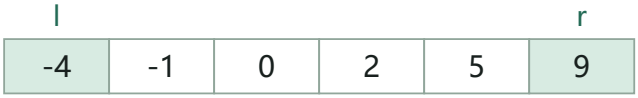
2.3-7 · Day3 第 40 页 · 排序加双指针

原题。 n 个整数构成集合 S，判断是否有两个元素之和为 x，要求 $\Theta(n \lg n)$ 时间。原集合中的两个元素不同；若改成数组，要求不同位置，允许值相等。

前置概念。 双指针是两个表示数组位置的下标，此题 l 指左端、r 指右端；归并排序先递归排好两半，再线性合并，提供后续安全排除所需的有序性。

完整手算

$S = \{-4, -1, 0, 2, 5, 9\}, x = 7$。先归并排序，再从两端开始。



太小：l 右移；太大：r 左移

l, r	**两端相加**	**判断与移动**
$0, 5$	$-4 + 9 = 5$	太小，$l \leftarrow 1$
$1, 5$	$-1 + 9 = 8$	太大，$r \leftarrow 4$
$1, 4$	$-1 + 5 = 4$	太小，$l \leftarrow 2$
$2, 4$	$0 + 5 = 5$	太小，$l \leftarrow 3$
$3, 4$	$2 + 5 = 7$	找到，返回真

```
A = MergeSort(S)
l = 0; r = n-1
while l < r:
    s = A[l] + A[r]
    if s == x: return True
    if s < x: l = l+1
    else: r = r-1
return False
```

不同位置靠 $l < r$ 保证。 单元素 [5] 不能与自己凑出 10；数组 $[5, 5]$ 可以。

为什么移动不会漏解

不变式： 若原输入有合法答案，未返回时当前区间始终保留一组合法答案。

和太小，排除左端。 对所有仍在区间中的 $t > l$：

$$A[l] + A[t] \leq A[l] + A[r] < x.$$

右端已是左值能配到的最大值，仍不够大，所以左值与任何存活项都不可能成功。

和太大，排除右端。 对所有 $t < r$：

$$A[t] + A[r] \geq A[l] + A[r] > x.$$

左端已是右值能配到的最小值，仍过大，所以右值可以丢弃。

终止： 每轮 $r - l$ 至少减少 1；相遇后不足两个位置，根据不变式可判无解。

复杂度的来源

归并排序：

$$T(n) = 2T(n/2) + \Theta(n) = \Theta(n \lg n).$$

双指针至多 $n - 1$ 轮，成本 $O(n)$；相加为 $\Theta(n \lg n)$，排序额外空间 $O(n)$。

已经有序： 省去排序，最坏 $\Theta(n)$。

要求原下标： 排序元素改存“值、原下标”，最后返回两个原下标。

哈希变式： 查询 $x - v$ 后再插入 v，防止复用当前位置；时间通常按期望 $O(n)$ 表述。

变式与解析 $[-3, 0, 2, 4, 7, 11]$、目标 9：两端和依次 $8, 11, 7, 9$，找到 $2 + 7$。**技巧：先证明该丢哪一端，再写移动；“和小左移、和大右移”必须建立在已排序上。**

11 最大子数组

4.1-5 · Day4 第 32 页 · 非空区间 下标从 0 开始

原题。 从左到右一次扫描，在常数时间内由“以 $j-1$ 结尾的最优值”求出“以 j 结尾的最优值”，再维护全局最优。时间要求线性，算法非递归。

前置概念。 子数组必须连续，不能跳选元素；非空要求至少取一项。“状态”是对已处理部分保留的信息，E 记录当前结尾的最优值，B 记录整个已见前缀的最优值。

两个状态 各管一件事

E_j：**必须以 j 结尾**的最大非空区间和。

B_j：前缀 $A[0..j]$ 中任意位置结束的最大非空区间和。

$$E_j = \max(A[j], E_{j-1} + A[j]).$$

$$B_j = \max(B_{j-1}, E_j).$$

$$E_0 = B_0 = A[0].$$

两种来源已经穷尽： 区间要么只含 $A[j]$，要么由一个以 $j-1$ 结尾的区间续接。后者选最大的 E_{j-1}，才能得到最大和。

下标	0	1	2	3	4	5
A	3	-4	2	-1	6	-5
E	3	-1	2	1	7	2
B	3	3	3	3	7	7

E 必须在当前列结束

B 记录已经见过的最好答案

在 $j=2$ 重启： 旧 $E_1 = -1$，接上会比单独取 2 更小，所以 $E_2 = 2$、当前起点改为 2。

在 $j=4$ 刷新全局最优： $E_4 = 1+6 = 7 > B_3$，保存端点 $[2,4]$。

答案是 $[2,-1,6]$，最大和 7；最后 $E_5 = 2$，不能只返回最后的 E。

为什么不能从全局最优续接

本例 $B_3 = 3$ 对应位置 0 的单个 3；它没有在位置 3 结束。若错误计算：

$$B_3 + A[4] = 3 + 6 = 9,$$

就跳过了中间的 $-4, 2, -1$，拼成不连续的区间。

关键技巧： “可续接”的状态必须与当前位置相邻；全局最优只用于记录答案。

端点与线性实现

```
ending = best = A[0]
start = left = right = 0
for j = 1 to n-1:
    if ending < 0:
        ending = A[j]; start = j
    else:
        ending = ending + A[j]
    if ending > best:
        best = ending
        left = start; right = j
return best, left, right
```

每轮常数工作，时间 $\Theta(n)$；只保存几个变量，空间 $O(1)$。表中的全部 E、B 用来手算，代码无需保存。

全负与前缀和变式

全负： $[-4,-2,-7]$ 答案是 $-2, [1,1]$。非空题不能以 0 初始化全局答案。

前缀和： $P[t] = A[0] + \cdots + A[t-1]$、$P[0] = 0$，所以 $P[t] - P[s]$ 是区间 $[s, t-1]$ 的和。最优值为：

$$\max_{1 \le t \le n} (P[t] - \min_{0 \le s < t} P[s]).$$

先用“此前最小前缀”更新答案，再纳入当前前缀，保证 $s < t$、区间非空。

变式与解析 数组 $[2,-5,3,1,-2,4,-6,2]$ 的 E 为 $2,-3,3,4,2,6,0,2$，B 为 $2,2,3,4,4,6,6,6$。最优区间 $[2,5]$，和为 6；从位置 2 重启，在位置 5 保存端点。

12 最小堆多路归并

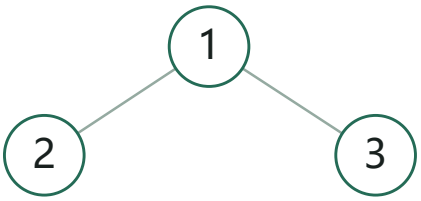
6.5-9 · Day4 第 33 页 · 每张有序表只保留一个候选

原题。 k 张有序表共有 n 个元素，合并成有序表，要求 $O(n \lg k)$。每表最小未输出值就是当前表头；全局最小值必在这些表头中。

前置概念。 完全二叉树除最底层外均填满，最底层从左到右紧靠排列；最小堆还要求父值不大于孩子，故根是最小值。兄弟之间无需有序。删根后下沉、插入后上浮以恢复堆性质，路径长度为对数级。

候选堆与来源

L0	1	4	7
L1	2	5	8
L2	3	6	9



堆节点存 (值, 表号, 位置)。输出某节点后，**只补同一张表的下一个元素**；若该表已空，不补。

输出	补入	剩余候选集合
1	L_0 的 4	$\{2, 3, 4\}$
2	L_1 的 5	$\{3, 4, 5\}$
3	L_2 的 6	$\{4, 5, 6\}$
4	L_0 的 7	$\{5, 6, 7\}$
5	L_1 的 8	$\{6, 7, 8\}$
6	L_2 的 9	$\{7, 8, 9\}$
7	无	$\{8, 9\}$
8	无	$\{9\}$
9	无	空

表中写的是集合；二叉堆的内部数组不要求整体有序。

四步算法与正确性

1. 扫描所有表，取非空表头，统一建最小堆。
2. 删除堆顶，将该值加入输出。
3. 如来源表有后继，把后继加入堆。
4. 重复直到堆空。

不变式： 每张未用完的表恰有一个最小未输出项在堆中；输出已经有序，且包含全局最小的若干项。

保持为何成立： 每表其他未输出项都不小于自己的代表；因此所有代表中的最小值就是全局最小。补入同来源后继，又恢复每表一代表。

把每项成本写清楚

设非空表数 $q \le k$。扫描目录 $O(k)$，统一建堆 $O(q)$，共 n 次删除、至多 n 次插入：

$$O(k + n \log (q + 1)).$$

当 $2 \le k \le n$，可写成题目要求的 $O(n \lg k)$。$k = 1$ 时为 $\Theta(n)$；大量空表存在时保留 $O(k)$。

堆操作	成本
查看最小值	$O(1)$
删除堆顶或插入	$O(\log (q + 1))$
自底向上建堆	$O(q)$

堆工作空间 $O(q)$；若存下完整输出，另用 $O(n)$。

易错： 把全部 n 项放进堆得到 $O(n \lg n)$；每次扫描 k 个表头则为 $O(nk)$。

变式与解析 合并 $[1, 4], [], [1, 3, 5], [2]$，输出 $[1, 1, 2, 3, 4, 5]$。初始与最大堆节点数都为 3；空表不入堆，两次出现的 1 都必须输出。表号与位置保留来源，重复值不会丢失。

13 混合排序的剩余工作量

7.4-5 · Day4 第 34 页 · 截断快排后对全数组插入排序

原题规则。 长度**严格小于** k 时停止快排，暂不排小块；所有划分结束后，对**整个数组**进行一次插入排序。本页分析第二阶段，$2 \le k \le n$。

前置概念。 枢轴是快排用来把小值与大值分开的元素；划分后它已就位。逆序对是 $i < j$ 却 $A[i] > A[j]$ 的位置对；插入排序把新元素前面比它大的值右移，再填入空位。



k = 4

l = 4

框内：未排小块　　圆圈：已就位枢轴

块间没有逆序对 → 剩余工作集中在块内

块间有序是证明的起点

取两个不同终止块，找到第一次将它们分开的祖先枢轴。前块来自左侧，后块来自右侧。

元素互异时，前块任意值小于枢轴，后块任意值大于枢轴；故**前块任何值都小于后块任何值**。已就位枢轴也不与其他区间形成逆序对。

因此，剩余逆序对全部在块内部。

上图取 $k = 4$，块长分别为 3、2、2。未排块的逆序对是：

$$(3,1),(3,2),(6,5),(9,8),$$

共 $I = 4$。最终插入恰好右移 4 次，不能笼统按最坏逆序数组估成实际 n^2 次移动。

为什么右移次数等于逆序数

插入一个 key 时，每右移一个比它大的前驱，恰好消去一对逆序。每个原始逆序对被跨越一次，所以：

$$T_{insert}(n) = \Theta(n + I).$$

n 项用于逐个访问，即使 $I = 0$，仍需线性时间。

从块长推出整体上界

设终止块长 $m_1, \ldots, m_s$，有 $m_i < k$、$\sum_i m_i \le n$。

$$\begin{aligned} I &\le \sum_i \binom{m_i}{2} \\ &\le \frac{k-1}{2} \sum_i m_i \\ &\le \frac{n(k-1)}{2} = O(nk). \end{aligned}$$

第一个不等式： 一个长度 m_i 的块最多有 $\binom{m_i}{2}$ 对逆序。

第二个不等式： $m_i - 1 \le k - 1$。这里取便于书写的宽松常数，不影响阶。

由此：

$$T_{insert} = \Theta(n + I) = O(nk).$$

只有短块还不够

反例为 [5, 6] [3, 4] [1, 2]。每块长度只有 2，内部甚至有序；但块间逆序共 $4 + 4 + 4 = 12$ 个。

一般令 $n/2$ 个内部有序二元素块按值由大到小排列，跨块逆序数为：

$$4\binom{n/2}{2} = \frac{n(n-2)}{2} = \Theta(n^2).$$

所以“短块”必须结合“快排划分保证的块间有序”，才能推出 $O(nk)$。

变式与解析 [2, 1, 3] 4 [6, 5] 7 [9, 8] 有逆序对 $(2,1),(6,5),(9,8)$，插入右移 3 次。若把比较条件由 > 改为 >=，相等项也会右移，原来的稳定性与“严格逆序对恰好对应右移”都要重新讨论。

14 混合排序的期望时间与阈值

7.4-5 · 元素互异 枢轴均匀随机 $2 \le k \le n$

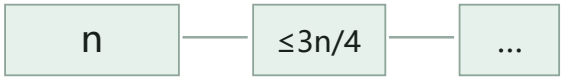
要证明的完整结论： $\mathbb{E}[T(n,k)] = O(nk + n \lg(n/k))$。第一阶段只负责划分，第二阶段负责补齐块内次序；两部分分别计时。

前置概念。 $\mathbb{E}[T]$ 是对算法内部随机选择按概率加权的平均；均匀随机指每个候选元素被选为枢轴的概率相同。秩是元素在当前子数组按值排序后的位置，阈值 k 决定何时停止划分。

第一阶段为何是这个期望界

固定一个元素，跟踪它所在的当前子数组。长度为 m 的一次划分成本 $\Theta(m)$，可给其中每个元素分摊常数成本。

好枢轴： 秩位于中间约一半。均匀随机选择时，概率至少为一个固定正常数；取到后，当前元素要么成为枢轴退出，要么进入大小至多 $3m/4$ 的子问题。



每次好划分，规模按固定比例缩小

经过 h 次好划分：

$$m \le n(3/4)^h.$$

规模降到 k 以下所需好划分次数为 $O(1 + \lg(n/k))$。每次等到好枢轴的期望尝试数是常数，故每个元素期望参与同阶次数的划分。

对 n 个元素求和：

$$\mathbb{E}[T_{partial}] = O(n + n \lg(n/k)).$$

关键分析： 期望线性性不要求各元素路径独立。只画一棵每次均分的树，只能给直觉，不能证明随机期望。

两阶段相加

$$\begin{aligned}\mathbb{E}[T] &= O(n + n \lg(n/k) + nk) \\ &= O(nk + n \lg(n/k)).\end{aligned}$$

当 $k = n$，第一阶段还做一次划分，不能单独写成 $n \lg(n/k) = 0$；总式中 nk 吸收该线性项。

阈值怎样选

k 的选法	代入后的总上界
固定小常数	$O(n \lg n)$
$\Theta(\lg n)$	$O(n \lg n)$
$\Theta(\sqrt{n})$	$O(n^{3/2})$
n	$O(n^2)$

理论： 固定小常数能保持比较排序的最优阶，但大 O 不能确定机器上最快的具体整数。

带常数的模型：

$$F(k) = an \ln(n/k) + bnk, \quad a, b > 0.$$
$$F'(k) = -an/k + bn = 0 \Rightarrow k = a/b.$$

这只解释最优阈值为何常为常数；还要取合法整数、限制在 $[2,n]$，并测量实际成本。常数 a, b 不来自渐近符号。

期望与最坏为什么能并存

随机快排仍可能连续选极值：

$$T(n) = T(n-1) + \Theta(n) = \Theta(n^2).$$

这条坏路径有正概率；“期望快”不等于“每次都快”。

普通随机快排的期望比较次数满足：

$$C_n = n - 1 + \frac{2}{n}\sum_{i=0}^{n-1} C_i.$$
$$C_n = 2(n+1)H_n - 4n = \Theta(n \lg n).$$

例如 $C_3 = 8/3$ 是期望，一次运行的实际比较数仍为整数。混合排序分析使用的是截断后的路径长度，不能把普通快排的完整成本直接当作更紧答案。

变式与解析 取 $k = \sqrt{n}$，则 $nk = n^{3/2}$、$n \lg(n/k) = \frac{1}{2} n \lg n$，总上界为 $O(n^{3/2})$；这一步证明的是上界，不能凭代入把 O 自动改为 Θ。长度恰好为 k 时仍要划分。