前四课作业解答与方法

一 1-1 渐进关系

1 题目与答案

Day2 第 31 页。判断每对函数的五种渐进关系。$k \geq 1, \varepsilon > 0, c > 1$ 均为固定常数；$\lg = \log_2$，$\ln = \log_e$。

小题	A	B	$O(B)$	$o(B)$	$\Omega(B)$	$\omega(B)$	$\Theta(B)$
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

2 解答分析

a 的理由。 令 $t = \ln n$，则比值是常数倍 $t^k / e^{\varepsilon t}$，趋于 0。固定次幂对数比任意正次幂慢。

b 的理由。 对比值取自然对数得 $k \ln n - n \ln c \to -\infty$，所以原比值趋于 0。固定底数大于 1 的指数比任意固定次幂快。

c 的理由。

$$\frac{A}{B} = n^{1/2 - \sin n}.$$

对于实数规模，分别取 $\sin n = 1$ 和 -1 的序列即可。对于整数规模，分别取最接近 $\pi/2 + 2\pi j$ 和 $3\pi/2 + 2\pi j$ 的整数，其中 j 趋于无穷。取整误差至多 $1/2$，所以两列整数的正弦分别至少为 $\cos(1/2) > 3/4$、至多为 $-\cos(1/2) < 1/4$。因此分别有比值不超过 $n^{-1/4}$ 和不小于 $n^{1/4}$ 的无穷子序列。它既无最终正下界，也无最终常数上界，五种关系均不成立。

d 的理由。 比值为 $2^{n/2} \to \infty$，所以是严格更快的增长。

e 的理由。

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

两者恒等。

f 的理由。 $\lg(n^n) = n\lg n$。由阶乘上下界：

$$\left(\frac{n}{2}\right)^{n/2} \le n! \le n^n,$$

从而对足够大的 n：

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

所以 $\lg(n!) = \Theta(n\lg n)$。更精确地，Stirling 公式给出二者比值趋于 1。

3 判定方法

设 f, g 在足够大的 n 上为正。

关系	定义中的核心不等式	比值有极限时的快速判断
$f = O(g)$	存在 $c > 0, n_0$，$n \ge n_0$ 时 $f \le cg$	$f/g \to L < \infty$ 即成立
$f = o(g)$	对任意 $c > 0$，足够大时 $f < cg$	$f/g \to 0$
$f = \Omega(g)$	存在 $c > 0, n_0$，$n \ge n_0$ 时 $f \ge cg$	$f/g \to L > 0$ 或 ∞
$f = \omega(g)$	对任意 $c > 0$，足够大时 $f > cg$	$f/g \to \infty$
$f = \Theta(g)$	同时有 $c_1 g \le f \le c_2 g$	$f/g \to L \in (0, \infty)$

比值不存在极限时，应回到“最终有界”和“最终有正下界”判断，不能直接宣布所有关系都不成立。例如 $f = n(2 + \sin n), g = n$ 的比值始终在 1 和 3 之间，仍有 Θ 关系。

比值趋于 0 时，O 与 o 同时成立；比值趋于无穷时，Ω 与 ω 同时成立；比值趋于正常数时，O, Ω, Θ 同时成立。这不是五选一。

判定模板。 “先化简原式，令 $R = A/B$；给出极限或上下界；再逐一填写五列。若存在振荡，则找两条趋于无穷的输入子序列分别排除上界和下界。”

二 1-2 函数排序

来源：Day2 第 32 页。原题要求按从快到慢的方向排列，即相邻项满足前者为后者的 Ω，并合并 Θ 等价类。截图中的 30 项如下：

第 1 项	第 2 项	第 3 项	第 4 项	第 5 项	第 6 项
$\lg(\lg^* n)$	$2^{\lg^* n}$	$(\sqrt{2})^{\lg n}$	n^2	$n!$	$(\lg n)!$
$(3/2)^n$	n^3	$(\lg n)^2$	$\lg(n!)$	2^{2^n}	$n^{1/\lg n}$
$\ln\ln n$	$\lg^* n$	$n2^n$	$n^{\lg\lg n}$	$\ln n$	1
$2^{\lg n}$	$(\lg n)^{\lg n}$	e^n	$4^{\lg n}$	$(n+1)!$	$\sqrt{\lg n}$
$\lg^*(\lg n)$	$2^{\sqrt{2\lg n}}$	n	2^n	$n\lg n$	$2^{2^{n+1}}$

先化简，再排序。

$$
\begin{aligned}
n^{1/\lg n} &= 2, & (\sqrt{2})^{\lg n} &= \sqrt{n},\\
2^{\lg n} &= n, & 4^{\lg n} &= n^2,\\
(\lg n)^{\lg n} &= n^{\lg\lg n}, & \lg(n!) &= \Theta(n\lg n).
\end{aligned}
$$

以下表格 **自上而下由快到慢。**同一行内互为 Θ；不同行之间相邻的前一行均为后一行的 ω。共 24 组、30 个函数。

组号	同阶函数	定位
1	$2^{2^{n+1}}$	比下一项多一倍外层指数
2	2^{2^n}	双重指数
3	$(n+1)!$	与 $n!$ 相差因子 $n+1$
4	$n!$	超过任意固定底数指数
5	e^n	固定指数底数最大
6	$n2^n$	严格快于 2^n
7	2^n	底数 2 的指数

组号	同阶函数	定位
8	$(3/2)^n$	底数 1.5 的指数
9	$(\lg n)^{\lg n}$, $n^{\lg\lg n}$	两者恒等，超多项式
10	$(\lg n)!$	快于任意固定次幂，慢于上一组
11	n^3	三次多项式
12	$4^{\lg n}$, n^2	平方级
13	$n\lg n$, $\lg(n!)$	线性乘对数
14	$2^{\lg n}$, n	线性级
15	$(\sqrt{2})^{\lg n}$	平方根级
16	$2^{\sqrt{2\lg n}}$	超过对数的任意固定次幂，低于任意正次幂
17	$(\lg n)^2$	对数平方
18	$\ln n$	对数级，换底只差常数
19	$\sqrt{\lg n}$	对数的平方根
20	$\ln\ln n$	双对数
21	$2^{\lg^* n}$	迭代对数的指数
22	$\lg^* n$, $\lg^*(\lg n)$	最终只差 1
23	$\lg(\lg^* n)$	迭代次数再取对数
24	1, $n^{1/\lg n}$	常数级

其中 6 组各含两个函数，其余 18 组各含一个函数，合计 $6\times 2+18=30$。

四个最容易错的位置。

第一，不能把 $(n+1)!$ 与 $n!$ 合并，因为比值 $n+1$ 无界。相差一个因子 n 会改变 Θ 类。

第二，在这张由快到慢的表中，$(\lg n)!$ 要排在任意固定正次幂 n^c 的上方。令 $x=\lg n$，有：

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

与 $\lg(n^c) = cx$ 比较可知，$x!$ 最终超过任意固定 n^c；但 $x!/x^x \to 0$，所以仍慢于 $(\lg n)^{\lg n}$。若 $\lg n$ 不是整数，可按 $\lfloor \lg n \rfloor !$ 理解，以上跨组的严格顺序保持不变。

第三，令 $x = \lg n$，对 $2^{\sqrt{2x}}$ 和 x^c 取二进制对数，只需比较 $\sqrt{2x}$ 与 $c\lg x$，前者更快；与 $n^{\varepsilon} = 2^{\varepsilon x}$ 比较，前者指数更慢。因此它位于全部固定次幂对数之后、任意正次幂之前。

第四，双重指数不能丢括号：

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

原式是 $2^{2^{n+1}}$，与单层指数 $2^{2n+1} = 2 \cdot 4^n$ 不同。

迭代对数。 $\lg^* n$ 是反复取二进制对数，直到值不超过 1 所需的次数；$\lg^k n = (\lg n)^k$ 是普通幂。对足够大的 n，$\lg^*(\lg n) = \lg^* n - 1$。$2^{\lg^* n}$ 比迭代次数增长快，却远慢于 $\ln\ln n$。

解题方法。 统一对数底数；化成已知形式；把相等或同阶项先合并；先排大类，再比较相邻疑难项；最后逐项数清 30 个原式是否全部出现，并检查题目要求的排序方向。

第 13-24 组逐步比较

以下沿用原表组号，统一 $\lg n = \log_2 n$，并在 $n \to \infty$ 时比较。记 $f \succ g$ 表示 $f/g \to \infty$，即 $f = \omega(g)$；同一组内只要求互为 Θ，数值可以相差固定倍数。

比较的依据是比值。 对最终为正的 f, g：

$$\frac{f}{g} \to \infty \Rightarrow f \succ g, \qquad \frac{f}{g} \to c \in (0, \infty) \Rightarrow f = \Theta(g).$$

当两项都能写成底数 2 的幂时，先计算指数差：

$$\frac{2^{u(n)}}{2^{v(n)}} = 2^{u(n)-v(n)}.$$

因此，只要指数差趋于正无穷，比值就趋于无穷。把两个函数分别取对数后“看起来同阶”，还不足以把原函数合为一组。例如 $\lg n$ 与 $\lg(n^2) = 2\lg n$ 同阶，而 $n^2/n = n \to \infty$。

1 先化简组内函数

第 13 组：为什么 $\lg(n!)$ 与 $n\lg n$ 同阶?

$n! = 1 \cdot 2 \cdots n$。所有因子都不超过 n，所以 $n! \le n^n$；后半段至少有 $n/2$ 个因子不小于 $n/2$，所以：

$$\left(\frac{n}{2}\right)^{n/2} \le n! \le n^n.$$

取二进制对数：

$$\frac{n}{2}(\lg n - 1) \le \lg(n!) \le n\lg n.$$

当 $n \ge 4$ 时，$\lg n - 1 \ge \frac{1}{2}\lg n$，因此：

$$\frac{1}{4}n\lg n \le \lg(n!) \le n\lg n.$$

两边是 $n\lg n$ 的固定正常数倍，故 $\lg(n!) = \Theta(n\lg n)$。这里证明的是同阶；$\lg(n!)$ 并不恒等于 $n\lg n$。

第 14 组： $2^{\lg n} = n$。 对数是指数的反运算，底数一致时直接还原。例如 $2^{\lg 16} = 2^4 = 16$。

第 15 组：把根号底数改写成 2 的幂。

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

这里用的是 $(a^b)^c = a^{bc}$，所以第 15 组就是平方根级。

第 18 组：把自然对数换成二进制对数。

$$\ln n = \frac{\lg n}{\lg e} = (\ln 2)\lg n.$$

$\ln 2$ 是固定正常数，因此 $\ln n = \Theta(\lg n)$。换底不会把对数级变成其他增长级别。

第 24 组：$n^{1/\lg n}$ 为什么是常数?

先写 $n = 2^{\lg n}$，再把两个指数相乘：

$$n^{1/\lg n} = (2^{\lg n})^{1/\lg n} = 2^{(\lg n)(1/\lg n)} = 2.$$

例如 $16^{1/\lg 16} = 16^{1/4} = 2$。所以它与 1 同阶，放在同一组；两者的数值分别是 2 和 1。

2 第 13-19 组：逐对计算比值

第 13 组比第 14 组快。 选各组的代表 $n\lg n$ 与 n，约去共同因子 n：

$$\frac{n\lg n}{n} = \lg n \to \infty.$$

因此 $n\lg n \succ n$。第 13 组另一个函数 $\lg(n!)$ 只差有界倍数，放在相同位置。

第 14 组比第 15 组快。

$$\frac{n}{\sqrt{n}} = n^{1-1/2} = \sqrt{n} \to \infty.$$

因此 $n \succ (\sqrt{2})^{\lg n}$。

第 15 组比第 16 组快：统一成底数 2。

令 $x = \lg n$，则 $x \to \infty$，两项分别变成：

$$\sqrt{n} = 2^{x/2}, \qquad 2^{\sqrt{2\lg n}} = 2^{\sqrt{2x}}.$$

计算比值，指数相减：

$$\frac{\sqrt{n}}{2^{\sqrt{2\lg n}}} = 2^{x/2-\sqrt{2x}}.$$

把指数中的 x 提出来：

$$\frac{x}{2} - \sqrt{2x} = x\left(\frac{1}{2} - \sqrt{\frac{2}{x}}\right).$$

括号趋于 $1/2$。更具体地，当 $x \geq 32$ 时，$\sqrt{2/x} \leq 1/4$，故指数至少为 $x/4 \to \infty$，原比值趋于无穷。因此：

$$\sqrt{n} \succ 2^{\sqrt{2\lg n}}.$$

第 16 组比第 17 组快：把对数幂也改写成指数。

仍令 $x = \lg n$，则 $(\lg n)^2 = x^2 = 2^{2\lg x}$。于是：

$$\frac{2^{\sqrt{2\lg n}}}{(\lg n)^2} = 2^{\sqrt{2x}-2\lg x}.$$

观察指数差：

$$\sqrt{2x}-2\lg x=\sqrt{x}\left(\sqrt{2}-\frac{2\lg x}{\sqrt{x}}\right).$$

这里 $\lg x/\sqrt{x}\to 0$。可再令 $u=\ln x$，将这个比值化成：

$$\frac{\lg x}{\sqrt{x}}=\frac{u}{(\ln 2)e^{u/2}}\to 0.$$

分子是一次方，分母是固定底数指数。因此括号趋于 $\sqrt{2}$，整个指数差趋于无穷，得到：

$$2^{\sqrt{2\lg n}}\succ(\lg n)^2.$$

这一步说明第 16 组为何夹在平方根级与对数平方之间。更一般地，对任意固定 $\varepsilon>0,c>0$：

$$(\lg n)^c=o\left(2^{\sqrt{2\lg n}}\right),\qquad 2^{\sqrt{2\lg n}}=o(n^{\varepsilon}).$$

证明仍是比较指数差：$\sqrt{2x}-c\lg x\to\infty$，$\varepsilon x-\sqrt{2x}\to\infty$。

第 17 组比第 18 组快：先换底，再约分。

$$\frac{(\lg n)^2}{\ln n}=\frac{(\lg n)^2}{(\ln 2)\lg n}=\frac{\lg n}{\ln 2}\to\infty.$$

因此 $(\lg n)^2\succ\ln n$。

第 18 组比第 19 组快。

$$\frac{\ln n}{\sqrt{\lg n}}=\frac{(\ln 2)\lg n}{(\lg n)^{1/2}}=(\ln 2)\sqrt{\lg n}\to\infty.$$

因此 $\ln n\succ\sqrt{\lg n}$。

3 第 19-20 组：再换一次变量

第 19 组是 $\sqrt{\lg n}$，第 20 组是 $\ln\ln n$。令 $x=\lg n$，先把后者写成关于 x 的函数：

$$\ln\ln n=\ln((\ln 2)x)=\ln x+\ln(\ln 2).$$

$\ln(\ln 2)$ 虽然为负数，但它是固定常数。比较较慢项与较快项：

$$\frac{\ln\ln n}{\sqrt{\lg n}} = \frac{\ln x}{\sqrt{x}} + \frac{\ln(\ln 2)}{\sqrt{x}} \to 0.$$

第一项是对数除以正次幂，第二项是常数除以趋于无穷的量，两项都趋于 0。因此：

$$\sqrt{\lg n} \succ \ln\ln n.$$

这里的 $\ln\ln n$ 是连续取两次自然对数；$\sqrt{\lg n}$ 是先取二进制对数，再开平方。

4 先读懂星号：迭代对数在数次数

$\lg^* n$ 表示反复取二进制对数，直到结果不超过 1 所需的次数。它与 $(\lg n)^k$ 这种普通幂不同。例如：

$$16 \to 4 \to 2 \to 1.$$

一共三次，所以 $\lg^* 16 = 3$。沿用这个定义：

n	**取对数的过程**	$\lg^* n$
2	$2 \to 1$	1
4	$4 \to 2 \to 1$	2
16	$16 \to 4 \to 2 \to 1$	3
$65536 = 2^{16}$	$65536 \to 16 \to 4 \to 2 \to 1$	4
2^{65536}	先变成 65536，再做上述四次	5

它是阶梯状函数，但 $n \to \infty$ 时，次数也会趋于无穷。先取一次对数，就少剩一次需要计数的操作，因此对 $n > 1$：

$$\lg^*(\lg n) = \lg^* n - 1.$$

而 $\lg(\lg^* n)$ 是先算次数、再对次数取对数。例如 $n = 16$ 时，前者为 $\lg^* 4 = 2$，后者为 $\lg 3$。

5 第 20-21 组：为什么双对数仍快于迭代次数的指数

要比较 $\ln\ln n$ 与 $2^{\lg^* n}$，不能只看“外面有没有指数”。指数的指数值是迭代次数，增长非常慢。

令 $y = \lg\lg n$。一方面：

$$\ln\ln n = (\ln 2)y + \ln(\ln 2) = \Theta(y).$$

另一方面，前两次对数把 n 变成 y，所以对足够大的 n：

$$\lg^* n = 2 + \lg^* y.$$

下面给出一个可用于严格比较的上界。对 $z \geq 2$：

$$\lg^* z \leq \lceil \lg z \rceil.$$

理由是：当当前值大于 4 时，取一次对数后的值至多是原值的一半；降到不超过 4 后，至多再取两次。令 $h = \lceil \lg z \rceil$，$z \leq 2^h$，所以至多先做 $h - 2$ 次减半，再做两次，就能结束；$2 \leq z \leq 4$ 的情况直接检查即可。

将这个上界用在 $\lg y$ 上，得到：

$$\begin{aligned} \lg^* y &= 1 + \lg^*(\lg y) \\ &\leq 1 + \lceil \lg \lg y \rceil \\ &\leq 2 + \lg \lg y. \end{aligned}$$

因此：

$$\begin{aligned} 2^{\lg^* n} &= 4 \cdot 2^{\lg^* y} \\ &\leq 4 \cdot 2^{2 + \lg \lg y} \\ &= 16 \lg y. \end{aligned}$$

现在较慢项被上界为 $16\lg y$，较快项与 y 同阶。计算比值：

$$0 \leq \frac{2^{\lg^* n}}{\ln \ln n} \leq \frac{16 \lg y}{(\ln 2) y + \ln(\ln 2)} \to 0.$$

所以：

$$\ln \ln n \succ 2^{\lg^* n}.$$

这个结论看 n 足够大时的增长关系，小规模数值的先后可以不同。例如 $n = 16$ 时，$\ln \ln 16 \approx 1.02$，而 $2^{\lg^* 16} = 8$；这一例不会推翻渐进结论。

6 第 21-24 组：把迭代次数整体换元

令 $t = \lg^* n$。虽然 t 增长很慢，但 $n \to \infty$ 时仍有 $t \to \infty$。最后四组变成：

组号	原函数	换元后
21	$2^{\lg^* n}$	2^t
22	$\lg^* n,\ \lg^*(\lg n)$	$t,\ t - 1$
23	$\lg(\lg^* n)$	$\lg t$

组号	原函数	换元后
24	$1,\ n^{1/\lg n}$	$1,\ 2$

第 21 组比第 22 组快。 指数与一次方比较：

$$\frac{2^t}{t} \to \infty.$$

第 22 组的两项同阶。 次数减 1 不改变增长阶：

$$\frac{t-1}{t} = 1 - \frac{1}{t} \to 1.$$

第 22 组比第 23 组快。 一次方与对数比较：

$$\frac{t}{\lg t} \to \infty.$$

第 23 组比第 24 组快。 取对数后的次数仍趋于无穷：

$$\frac{\lg t}{1} = \lg t \to \infty.$$

第 24 组内部的比值为 $2/1 = 2$，所以两项同阶。

7 完整顺序与自测

每个花括号代表一个 Θ 等价类，以下按从快到慢排列：

$$\begin{aligned}
\{n\lg n, \lg(n!)\} &\succ \{2^{\lg n}, n\} \\
&\succ \{(\sqrt{2})^{\lg n}\} \\
&\succ \{2^{\sqrt{2\lg n}}\} \\
&\succ \{(\lg n)^2\} \\
&\succ \{\ln n\} \\
&\succ \{\sqrt{\lg n}\} \\
&\succ \{\ln\ln n\} \\
&\succ \{2^{\lg^* n}\} \\
&\succ \{\lg^* n, \lg^*(\lg n)\} \\
&\succ \{\lg(\lg^* n)\} \\
&\succ \{1, n^{1/\lg n}\}.
\end{aligned}$$

解题时的换元选择。 多项式和“底数为 2 的复杂指数”相遇，令 $x=\lg n$；出现 $\lg^* n$ 的指数、一次方和对数，令 $t=\lg^* n$。比较 2^u 与 2^v 时，写出 $u-v$ 的极限；比较同阶项时，给出比值或固定常数夹逼。

自测一。 化简 $4^{\lg n}$、$n^{3/\lg n}$、$(\sqrt{8})^{\lg n}$。

答案与步骤。

$$4^{\lg n}=2^{2\lg n}=n^2,$$

$$n^{3/\lg n}=(2^{\lg n})^{3/\lg n}=2^3=8,$$

$$(\sqrt{8})^{\lg n}=(2^{3/2})^{\lg n}=n^{3/2}.$$

自测二。 比较 $2^{\sqrt{\lg n}}$ 与 $(\lg n)^{10}$。

答案与步骤。 令 $x=\lg n$，两者比值为 $2^{\sqrt{x}-10\lg x}$。由于 $\lg x/\sqrt{x}\to 0$，指数差趋于无穷，所以前者严格增长更快。

自测三。 $\lg^*(\lg n)$ 与 $\lg(\lg^* n)$ 是否同阶？

答案与步骤。 令 $t=\lg^* n$，两者为 $t-1$ 与 $\lg t$；$(t-1)/\lg t\to\infty$，因此前者严格增长更快。

自测四。 比较 $2^{2\lg^* n}$ 与 $\ln\ln n$。

答案与步骤。 令 $y=\lg\lg n$，沿用上面的界 $2^{\lg^* n}\le 16\lg y$，可得：

$$\frac{2^{2\lg^* n}}{\ln\ln n}\le\frac{256(\lg y)^2}{(\ln 2)y+\ln(\ln 2)}\to 0.$$

所以即使把迭代次数乘以固定常数 2 后再作为指数，仍慢于 $\ln\ln n$。

三 1-3 证明与反例

来源：Day2 第 33 页，f,g 均渐进为正。

a 命题。 $f=O(g)$ 是否推出 $g=O(f)$？

答案：否。 取 $f=n,g=n^2$。前提成立，但 $g/f=n\to\infty$，故结论不成立。若把前提改为 $f=\Theta(g)$，才可以交换双方。

b 命题。 $f+g=\Theta(\min(f,g))$ 是否成立？

答案：否。 取 $f = n^2, g = n$，左侧为 $\Theta(n^2)$，右侧为 $\Theta(n)$。正确的一般结论是：

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

渐进为正的条件保障不存在正负抵消。

c 命题。 若 $f = O(g)$，且足够大时 $\lg g \ge 1, f \ge 1$，能否推出 $\lg f = O(\lg g)$？

答案：能。 选择 $C \ge 1$，使足够大时 $f \le Cg$。则：

$$0 \le \lg f \le \lg C + \lg g \le (1 + \lg C)\lg g.$$

最后一步使用 $\lg g \ge 1$，因此满足 O 的定义。前提中的这两个条件不能在抄题时漏掉。

d 命题。 $f = O(g)$ 是否推出 $2^f = O(2^g)$？

答案：否。 取 $f = 2n, g = n$，前提成立，但：

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

指数位置的常数倍不再是整个函数的常数倍。若知道更强的 $f \le g + C$，则可得 $2^f \le 2^C 2^g$。

解题方法

证明成立。 写出所需量词；展开已知 O, Ω, Θ 的不等式；按目标变形；给出新的常数和充分大的阈值。

证明不成立。 选最简单的正函数；先验证前提；再算比值或矛盾说明结论失败。常用反例材料包括 $1, n, n^2, 2^n$，以及 $f = 2g$。

补充辨析。

命题	结论	一句话依据
$f = o(g) \Rightarrow f = O(g)$	对	比值趋于 0，因而最终有界
$f = O(g)$ 且 $g = O(f) \Rightarrow f = \Theta(g)$	对	上下界合并
$f = \Theta(g) \Rightarrow 2^f = \Theta(2^g)$	错	$f = 2n, g = n$
$f = O(g) \Rightarrow f^2 = O(g^2)$	对	非负不等式平方
$f = O(g) \Rightarrow \lg f = \Theta(\lg g)$	错	$f = 1, g = n$

命题	结论	一句话依据
$f+g=\Theta(\max(f,g))$	对	两个最终非负量夹逼

四 1-4 选择排序

1 题目与解答

完整题意。 Day2 第 34 页：每次在未确定部分找最大值，与其最后一个位置交换；重复直到全数组有序。写伪代码，给出循环不变式，分析最好和最坏运行时间。

```
def selection_sort(a):
    for end in range(len(a) - 1, 0, -1):
        max_index = 0
        for i in range(1, end + 1):
            if a[i] > a[max_index]:
                max_index = i
        a[max_index], a[end] = a[end], a[max_index]
    return a
```

外层循环不变式。 每轮开始时：

1. $A[end+1..n-1]$ 已按非降序排列，恰好包含原输入最大的 $n-1-end$ 个元素。
2. 未处理前缀的任何元素都不大于该已确定后缀中的任何元素。
3. 全数组与原输入的多重集合相同。

初始化。 `end=n-1` 时后缀为空，三个命题均成立。

保持。 内层比较找到前缀最大值；将它放到 `end` 后，新的已确定后缀多了前缀中最大的元素，且它不大于旧后缀中的元素，所以整个新后缀仍非降序。交换也不改变多重集合。

终止。 循环结束时只有位置 0 未单独处理，其元素不大于后缀各元素，因此全数组有序且元素无增删。`end` 严格下降保证终止。

若题目要求内层不变式，可写：“内层检查位置 i 前，`max_index` 指向 $A[0..i-1]$ 中一个最大元素。”

时间。 每轮寻找最大值的元素比较次数依次为 $n-1,n-2,\dots,1$，总计：

$$C(n)=\frac{n(n-1)}{2}.$$

无论输入是否有序，都必须扫描相同位置。故最好和最坏时间均为 $\Theta(n^2)$，额外空间 $O(1)$，最多进行 $n-1$ 次外层交换。

对 $[5,2,4,6,1,3]$，前两轮得到 $[5,2,4,3,1,6]$、$[1,2,4,3,5,6]$，第三轮得到 $[1,2,3,4,5,6]$；余下两轮仍要扫描，不能因为恰好有序就改变本算法的计数。

稳定性反例。 输入 $[2_a, 2_b, 1]$，最大值取首次出现的 2_a 并与末尾交换，得到 $[1, 2_b, 2_a]$，相等键顺序反转，所以标准交换式选择排序不稳定。

2 解题方法

每轮确定一个最大值，维护有序后缀；按初始化、保持、终止证明正确；对未处理前缀的扫描次数求和。代码下标从 0 开始，区间端点均包含。

五 4-1 递推式

1 题目与解答

来源：Day3 第 39 页。按原题，$n \le 2$ 时 $T(n)$ 为正常数；常规向上或向下取整不改变本组渐近结论。

小题	原递推式	参数或方法	紧确答案
a	$2T(n/2)+n^4$	$p=1$，第三种	$\Theta(n^4)$
b	$T(7n/10)+n$	单链递减等比和	$\Theta(n)$
c	$16T(n/4)+n^2$	$p=2$，第二种	$\Theta(n^2\lg n)$
d	$7T(n/3)+n^2$	$p=\log_3 7<2$，第三种	$\Theta(n^2)$
e	$7T(n/2)+n^2$	$p=\lg 7>2$，第一种	$\Theta(n^{\lg 7})$
f	$2T(n/4)+\sqrt{n}$	$p=1/2$，第二种	$\Theta(\sqrt{n}\lg n)$
g	$T(n-2)+n^2$	每次减 2，展开求和	$\Theta(n^3)$

a 完整推导。 $a=2, b=2$，分界函数为 n。$f=n^4$ 比 n 多固定幂次，且：

$$2f(n/2)=2(n/2)^4=\frac{1}{8}n^4.$$

取 $c = 1/8 < 1$，主定理第三种情况得到 $\Theta(n^4)$。递归树每层为 $n^4(1/8)^i$，总和也为 $\Theta(n^4)$。

b 完整推导。 每层只有一个节点，代价依次为：

$$n,\ \frac{7}{10}n,\ \left(\frac{7}{10}\right)^2 n,\ \ldots$$

所以总和不超过 $n/(1-7/10)+O(1)=O(n)$；根层本身就花 $\Omega(n)$，合为 $\Theta(n)$。也可写 $a=1, b=10/7, p=0$，用第三种情况，正则比为 $7/10$。递归深度 $\Theta(\log n)$ 不等于总时间 $\Theta(n\log n)$。

c 完整推导。 $p=\log_4 16=2$，$f=n^2$ 与分界同阶，因此是第二种情况：

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

第 i 层代价为 $16^i(n/4^i)^2=n^2$，共有对数层。不能因为出现 16 次调用就写 n^{16}。

d 完整推导。 因为 $7<9$，$p=\log_3 7<2$，且：

$$7f(n/3)=\frac{7}{9}n^2.$$

满足第三种情况，答案为 $\Theta(n^2)$。每层公比 $7/9$，根部主导。

e 完整推导。 因为 $7>4$，$p=\lg 7>2$，取 $\varepsilon=\lg 7-2>0$，有 $f=n^2=O(n^{p-\varepsilon})$。由第一种情况，答案为 $\Theta(n^{\lg 7})$，其中 $\lg 7\approx 2.807$。每层公比 $7/4$，叶部主导。

f 完整推导。 $p=\log_4 2=1/2$，$f=\sqrt{n}$，属于第二种情况：

$$T(n)=\Theta(\sqrt{n}\lg n).$$

第 i 层成本为 $2^i\sqrt{n/4^i}=\sqrt{n}$。若取 $n=4^m$，并把所有基础代价设为 1，则精确有 $T(4^m)=(m+1)2^m$，与渐近界一致。

g 完整推导。 规模是减去 2，不是除以常数，基本主定理不适用：

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

共有 $O(n)$ 项，每项至多 n^2，给出 $O(n^3)$；前 $\lfloor n/4\rfloor$ 项均不小于 $(n/2)^2$，给出 $\Omega(n^3)$。因此紧确答案为 $\Theta(n^3)$。奇数与偶数最终分别落到基础规模 1 与 2，结论一致。

2 主定理与递归树

$$T(n)=aT(n/b)+f(n).$$

a 是**实际递归调用次数**；n/b 是每次调用的规模；$f(n)$ 包括划分、合并以及当前调用中其他非递归工作；边界代价通常为正常数。

二分查找虽然把搜索区间分成两半，但只递归一半，所以 $a=1$。归并排序递归两半，$a=2$。主定理中的 a 取决于实际发生的递归调用数量，应先读算法，再数调用。

先验证 $a \geq 1, b>1$ 为常数，所有递归子问题大小相同，并设：

$$p=\log_b a, \qquad G(n)=n^p.$$

条件	结论	必须检查
$f(n)=O(n^{p-\varepsilon})$，某常数 $\varepsilon>0$	$T(n)=\Theta(n^p)$	比分界函数小一个多项式因子
$f(n)=\Theta(n^p)$	$T(n)=\Theta(n^p \lg n)$	两者真正同阶
$f(n)=\Omega(n^{p+\varepsilon})$ 且 $af(n/b)\leq cf(n)$，$c<1$	$T(n)=\Theta(f(n))$	多项式差距与正则条件

课件给的是上面的基本版。$f(n)=n/\lg n$ 比 n 慢，但没有小一个固定的多项式因子，不能因为“更小”就套第一种情况。

解题方法。

确定 $a, b, f(n)$；计算 $p=\log_b a$；比较 $f(n)$ 与 n^p；检查多项式差距或第三种的正则条件，写出 Θ 结论。

题目要求紧确上下界时，最后应写 Θ。只有 O 不能表达已经完成了紧确分析。

假设 $n=b^h$，递归在常数规模处停止。第 i 层有 a^i 个节点，每个节点规模为 n/b^i，因此：

$$\text{第 } i \text{ 层代价}=a^i f(n/b^i).$$

总高度为 $\Theta(\log_b n)$，叶子总数为 $\Theta(n^{\log_b a})$。总成本为非叶各层求和再加叶子成本，不能漏掉最后一项。

若 $f(n)=n^d$，每层成本为：

$$n^d\left(\frac{a}{b^d}\right)^i.$$

公比小于 1，根部主导；等于 1，各层相同，要乘对数层数；大于 1，叶部主导。这是主定理三种情况的来源。

3 常见变式

变式一。 $T(n)=2T(n/2)+n\lg n$。

令 $n=2^m$，$U_m=T(2^m)/2^m$，则：

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

因此 $U_m=\Theta(m^2)$，得到 $T(n)=\Theta(n\lg^2 n)$。这是常见的对数扩展情况，但不能硬说 $n\lg n=\Theta(n)$。

变式二。 $T(n)=2T(n/2)+n/\lg n$，规模足够小时设为常数。

仍用 $U_m=T(2^m)/2^m$，得到 $U_m=U_{m-1}+1/m$。调和和为 $\Theta(\log m)$，故：

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

变式三。 $T(n)=T(\sqrt{n})+1$。

递归 h 次后的规模为 $n^{1/2^h}$。到常数规模时 $2^h=\Theta(\lg n)$，所以：

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

变式四。 $T(n)=T(n/3)+T(2n/3)+n$。

子问题不等大，不能直接套基本主定理。每层子问题规模之和不超过 n，最长深度 $O(\log n)$，所以 $O(n\log n)$。前 $\lfloor\log_3 n\rfloor-O(1)$ 层尚无叶子，每层规模和为 n，所以还有 $\Omega(n\log n)$。合为 $\Theta(n\log n)$。

变式五。 $T(n)=2T(n-1)+1$。展开后是等比增长，得到 $\Theta(2^n)$，不能因“减去 1”就把项数乘最大单项当成紧确答案。

4 代入法

要证明 $T(n)=2T(n/2)+n$ 的上界，可以猜：

$$T(n)\le Cn\lg n+Dn,$$

其中 D 足够大以覆盖基础代价。代入归纳假设：

$$\begin{aligned}T(n) &\le 2\left(C\frac{n}{2}\lg\frac{n}{2}+D\frac{n}{2}\right)+n\\ &= Cn\lg n+Dn-(C-1)n\\ &\le Cn\lg n+Dn, \qquad C\ge 1.\end{aligned}$$

下界可由每层 $\Theta(n)$ 且有 $\Theta(\lg n)$ 层得到。

六 2.3-7 两数之和

1 题目与算法

来源：Day3 第 40 页。给定 n 个整数组成的集合 S 和整数 x，判断是否存在两个元素之和为 x，要求时间 $\Theta(n\lg n)$。

集合中的两个元素应当不同；若题目变为允许重复值的数组，则要求两个不同位置，允许这两个位置上的值相同。例如数组 [5,5] 可以凑出 10，[5] 不可以。

思路。 先用归并排序得到非降序数组，左右指针从两端开始。和太小就排除左端，和太大就排除右端，相等时找到答案。

```
def two_sum_sorted_search(values, target):
    a = merge_sort(values)
    left, right = 0, len(a) - 1
    while left < right:
        current = a[left] + a[right]
        if current == target:
            return True
        if current < target:
            left += 1
        else:
            right -= 1
    return False
```

2 正确性分析

不变式。 若原输入中存在答案，则在尚未返回 `True` 的每轮开始时，当前区间中仍有一对合法答案。

初始化时区间覆盖所有元素。若 $A[l]+A[r]<x$，则对当前区间中所有 $l<t\le r$：

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

所以 $A[l]$ 不可能与任何仍存活的元素配对，可以排除左端。若 $A[l]+A[r]>x$，则对 $l\le t<r$：

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

所以右端也可以安全排除。已经排除的元素在此前步骤中已证明不可能参与答案，不必重新考虑。

每轮 $r-l$ 严格减少，算法终止。若返回 `True`，两个不同位置和为目标；若退出时 $l\ge r$，当前不足两个位置，根据不变式原问题无解。因此所有返回均正确。

3 手算与复杂度

输入 $S=\{-4,-1,0,2,5,9\}$、$x=7$。指针对应的和依次为：

左值	右值	和	操作
-4	9	5	左指针右移
-1	9	8	右指针左移
-1	5	4	左指针右移
0	5	5	左指针右移
2	5	7	返回真

归并排序无论输入次序如何都用 $\Theta(n\lg n)$，扫描至多 $n-1$ 轮、用 $O(n)$，总时间：

$$\Theta(n\lg n)+O(n)=\Theta(n\lg n).$$

排序及其结果数组需 $O(n)$ 额外空间，扫描本身需 $O(1)$。空集合和单元素集合直接无解。

4 常见变式

要求原数组下标。 排序时保存 (值,原下标)，比较与移动仍按值；找到后返回两个原下标。

输入已经有序。 直接双指针，最坏 $\Theta(n)$，不再额外排序。

使用哈希。 一次遍历可期望 $O(n)$ 时间、$O(n)$ 空间，查询后再插入可防止同一个位置被使用两次。但原题指定 $\Theta(n\lg n)$ 方案时，排序加双指针最直接满足要求；哈希的期望时间和最坏时间需分清。

目标改为最接近给定值。 每次先更新最小绝对差，再依据当前和与目标大小移动同样的指针。正确性仍来自有序性：被排除的一端与更内侧值配对不会得到比当前组合更接近的方向改善。

解题方法。 “先建立单调结构；明确和偏大或偏小时哪一端绝不可能再参与更好的解；证明排除安全；用区间长度下降证明终止；排序成本与扫描成本相加。”

5 归并排序

把数组分成两半，各自递归排好，再合并两个有序序列。每次只需比较两个序列当前尚未取出的最小值。

```python
def merge_sorted(left, right):
    result = []
    i = j = 0
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1
    while i < len(left):
        result.append(left[i])
        i += 1
    while j < len(right):
        result.append(right[j])
        j += 1
    return result
def merge_sort(a):
    if len(a) <= 1:
        return list(a)
    mid = len(a) // 2
    left = merge_sort(a[:mid])
    right = merge_sort(a[mid:])
    return merge_sorted(left, right)
```

合并不变式。 输出已是有序前缀，且包含两序列中最小的若干元素；两个指针各自指向本序列最小的未输出项。因此取两者较小值时，必然取到全体未输出元素的最小值。

递归证明。 长度不超过 1 时有序。假设短数组可以正确排序，则两个递归结果均正确，线性合并又保持有序与多重集合，因此当前数组正确。子问题严格缩小保证终止。

手算。 合并 $[2,4,7]$ 与 $[1,3,6]$，依次比较选出 $1,2,3,4,6$，然后直接复制 7。最多 $3+3-1=5$ 次键比较。一般两段长度为 $p,q>0$ 时，最坏键比较次数为 $p+q-1$，总复制工作为 $\Theta(p+q)$。

递推。

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

常规实现最好、平均、最坏时间均为 $\Theta(n\lg n)$。峰值额外空间 $O(n)$，另有 $O(\log n)$ 层栈。上面的切片实现会产生总计 $O(n\log n)$ 的累计分配量，但各层不同时保留全部历史对象，峰值仍是 $O(n)$。

等值时优先取左序列，保证稳定。题目若增加“已经有序就直接返回”之类优化，最好情况应针对新算法重新分析。

七 4.1-5 最大子数组

1 题目与算法

来源：Day4 第 32 页。按题意从左向右扫描，维护已见部分的最大子数组，并利用“以位置 j 结尾的最大子数组”在常数时间内求出以 $j+1$ 结尾的最大子数组。要求非递归、线性时间。

最关键的状态区别。

$$E_j = \text{必须以 } j \text{ 结尾的最大非空区间和,}$$

$$B_j = \text{整个前缀 } A[0..j] \text{ 中的最大非空区间和.}$$

以 j 结尾的区间，要么只有 $A[j]$，要么接在某个以 $j-1$ 结尾的区间后面。为使后者最优，选 E_{j-1}：

$$E_j = \max\{A[j], E_{j-1} + A[j]\}, \qquad B_j = \max\{B_{j-1}, E_j\}.$$

初始 $E_0 = B_0 = A[0]$。只记住全局最佳 B_{j-1} 无法正确续接，因为那段区间可能不在 $j-1$ 结束。

```
def max_subarray(a):
    ending = best = a[0]
    start = best_left = best_right = 0
    for j in range(1, len(a)):
        if ending < 0:
            ending = a[j]
            start = j
        else:
            ending += a[j]
        if ending > best:
            best = ending
            best_left, best_right = start, j
    return best, best_left, best_right
```

代码前提是数组非空，返回 (最大和,左端点,右端点)。并列最优时返回其中一个合法区间，原题未要求唯一的并列规则。

2 手算过程

输入 $[-2,1,-3,4,-1,2,1,-5,4]$：

下标 j	$A[j]$	E_j	B_j	当前结尾区间起点
0	-2	-2	-2	0
1	1	1	1	1
2	-3	-2	1	1
3	4	4	4	3

下标 j	$A[j]$	E_j	B_j	当前结尾区间起点
4	-1	3	4	3
5	2	5	5	3
6	1	6	6	3
7	-5	1	6	3
8	4	5	6	3

答案为 6，区间 $[3,6]$，内容 $[4,-1,2,1]$。虽然最后的 $E_8 = 5$，全局答案仍是 6，因此不能只返回最后一个 `ending`。

全负例子 $[-5,-2,-7]$：`ending` 依次为 $-5,-2,-7$，`best` 为 $-5,-2,-2$，最终返回 $(-2,1,1)$。单元素数组直接返回该元素及区间 $[0,0]$。

3 正确性与解题方法

初始化。 只有位置 0 时，唯一非空区间是 $[0,0]$，两个状态与端点均正确。

保持。 假设处理完 $j-1$ 后两个状态正确。任何以 j 结尾的区间不是只含当前位置，就是由以 $j-1$ 结尾的区间加上 $A[j]$。两者取最大得到正确的 E_j。前缀中的最优区间不是已经在旧前缀出现，就是以新位置结尾，所以与 B_{j-1} 取最大得到正确的 B_j。重启时更新起点，刷新全局最优时保存端点，因此端点也对应正确区间。

终止。 全数组已处理，`best` 就是所求。每次常数更新，循环 $n-1$ 次，时间 $\Theta(n)$，额外空间 $O(1)$。

解题方法。 “先定义必须以当前位置结尾的最优值，再枚举最后一步的两种来源；写初值与递推；用另一个变量记录所有结尾的最优；需要端点时同步记录起点和刷新时的端点。”

4 常见变式

前缀和方法。 定义 $P[0]=0$、$P[j+1]=P[j]+A[j]$，以 j 结尾的区间和是 $P[j+1]-P[i]$，其中 $0 \le i \le j$。因此：

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

先用此前最小前缀更新答案，再把当前前缀纳入最小值，保证 $s < t$，区间非空。只保存滚动前缀和与最小值，仍为 $\Theta(n)$ 时间、$O(1)$ 空间。

分治返回更多信息。 每段返回四项：总和 `sum`、最大非空前缀和 `pref`、最大非空后缀和 `suff`、最大非空区间和 `best`。左右段分别为 L, R，合并公式是：

$$\begin{aligned}\text{sum} &= L.\text{sum} + R.\text{sum},\\ \text{pref} &= \max(L.\text{pref}, L.\text{sum} + R.\text{pref}),\\ \text{suff} &= \max(R.\text{suff}, R.\text{sum} + L.\text{suff}),\\ \text{best} &= \max(L.\text{best}, R.\text{best}, L.\text{suff} + R.\text{pref}).\end{aligned}$$

单元素四项均等于该元素。合并成本降为 $O(1)$，递推变成 $2T(n/2) + O(1) = \Theta(n)$。这说明不能只背“最大子数组分治一定是 $n\log n$”，要看返回的信息是否足够。

并列最优的处理。 原题要求找到一组最优子数组时，保留任意一组即可；若变式要求最左、最短或最长区间，则在和相等时增加相应的端点比较。先写清规则，再选择严格改善或非严格改善时更新。

八 6.5-9 多路归并

1 题目与算法

来源：Day4 第 33 页。给定 k 个已经排好序的表，共 n 个元素，将它们合并成一个有序表，要求 $O(n\lg k)$ 时间；提示使用最小堆。

关键观察。 每个表自身有序，所以它的最小未输出元素一定是当前表头。全局最小未输出元素一定在这至多 k 个表头中，无须把所有 n 个元素都放进堆。

堆节点保存 (值,表编号,表内位置)。取出某个节点后，只补入**同一张表**的后继元素。表编号与位置也使重复值能被一致处理。

```
import heapq


def merge_k_sorted(lists):
    heap = []
    for table, values in enumerate(lists):
        if values:
            heap.append((values[0], table, 0))
    heapq.heapify(heap)
    result = []
    while heap:
        value, table, index = heapq.heappop(heap)
        result.append(value)
        next_index = index + 1
        if next_index < len(lists[table]):
            heapq.heappush(
                heap,
                (lists[table][next_index], table, next_index),
            )
    return result
```

2 执行过程与正确性

输入三表 $L_0 = [1,4,7]$、$L_1 = [2,5,8]$、$L_2 = [3,6,9]$。

本轮输出	来源表	补入后继	输出完成后的堆中候选值
1	0	4	$\{2,3,4\}$
2	1	5	$\{3,4,5\}$
3	2	6	$\{4,5,6\}$
4	0	7	$\{5,6,7\}$
5	1	8	$\{6,7,8\}$
6	2	9	$\{7,8,9\}$
7	0	无	$\{8,9\}$
8	1	无	$\{9\}$
9	2	无	空

表中展示的是候选集合，不是堆数组的精确内部排列。

不变式。 每轮开始时，输出已经有序且正好包含全体最小的若干元素；每个尚未用完的表恰有一个节点在堆中，代表该表最小未输出项。

初始化把所有非空表头加入堆，不变式成立。每个表剩余项均不小于自己的堆中代表，所以堆顶是全局最小未输出项；输出它并补入本表后继，保持不变式。堆空时所有表均用完，且每个元素恰好输出一次，得到所需合并结果。

3 复杂度

设实际非空表数为 $q \le k$。扫描表目录需要 $O(k)$，建堆需要 $O(q)$。共有 n 次输出，每次删除和至多一次插入，各用 $O(\log(q+1))$，因此完整表达式是：

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

在题目通常采用的 $2 \le k \le n$ 且无大量空表条件下，写成要求的 $O(n\lg k)$。$k = 1$ 时是 $\Theta(n)$，不能把 $n\lg 1 = 0$ 当作实际时间；若有大量空表，必须保留扫描目录的 $O(k)$。

堆占 $O(q)$ 工作空间，输出列表另占 $\Theta(n)$；若边合并边写入外部输出，则无须把全部结果同时保存在内存中。

4 解题方法与变式

逐次扫描所有 k 个表头找最小，每个输出花 $O(k)$，总计 $O(nk)$。把全部元素放进一个堆，堆规模变为 n，总计 $O(n\log n)$，没有利用每张表已经有序的条件。

两两归并也能做到 $O(n\log k)$：按平衡的轮次合并相邻表，每轮全部元素至多处理一次，共 $\lceil \log_2 k \rceil$ 轮。不要把第一个表与后续表依次累积合并；不平衡的重复扫描可能退化。

解题方法。 “每个数据流只保留一个当前候选；用堆选全局最小；输出后补同一来源的后继；证明候选完备；总元素数乘对数堆规模，并单独计初始化与输出空间。”

5 堆操作

最小二叉堆是一棵完全二叉树，父节点键不大于子节点键。它不是全局排好序的数组，但根节点一定是全局最小值。

0-based 数组存储下，位置 i 的左孩子为 $2i+1$、右孩子为 $2i+2$；非根节点的父亲为 $\lfloor (i-1)/2 \rfloor$。堆高 $O(\log q)$，其中 q 为堆内元素数。

操作	时间	理由
查看堆顶	$O(1)$	直接读根

操作	时间	理由
插入	$O(\log q)$	最多沿父链上浮
删除堆顶	$O(\log q)$	末项补根后沿孩子下沉
自底向上建堆	$\Theta(q)$	多数节点高度很小

建堆为何不是紧确的 $q\log q$。 高度约为 h 的节点只有 $O(q/2^h)$ 个，每个下沉至多 $O(h)$，所以：

$$\sum_{h\ge 0} O\left(\frac{q}{2^h}\right) O(h) = O(q),$$

而建堆至少要处理线性数量的位置，故为 $\Theta(q)$。把每个节点都按树的最大高度估计，只得到较松的 $O(q\log q)$ 上界。

九 7.4-5 混合排序

1 题目与算法

来源：Day4 第 34 页。快排递归到**少于 k 个元素**的子数组时，直接返回，暂不排序。顶层快排结束后，对**整个数组**做一次插入排序。证明期望时间：

$$O\left(nk + n\lg\frac{n}{k}\right),$$

并讨论理论与实践如何选择 k。以下分析取 $2 \le k \le n$、元素互异、每次枢轴在当前子数组中均匀随机。

```
def hybrid_sort(a, k, rng=None):
    if rng is None:
        rng = Random()

    def partial(left, right):
        if right - left + 1 < k:
            return
        pivot = randomized_partition(a, left, right, rng)
        partial(left, pivot - 1)
        partial(pivot + 1, right)

    partial(0, len(a) - 1)
    insertion_sort(a)
    return a
```

代码使用本章的 `insertion_sort` 与 `randomized_partition`，调用前提为整数 $k \ge 2$。长度恰好等于 k 仍要划分。最终插入排序保证输出有序，而所有操作均保持元素多重集合。

2 块间有序

快排停止后，小块内部可以无序，但不同终止小块之间没有逆序对。任取两个不同块，找到首次把它们分到两侧的祖先枢轴；左块全部元素小于该枢轴，右块全部元素大于它，所以左块任意值都小于右块任意值。已经就位的枢轴也不会与其余区间形成逆序对。

例如 $k = 4$ 时一种可能结果是：

$$[3,1,2] \quad 4 \quad [6,5] \quad 7 \quad [9,8].$$

三个未排块的长度为 3、2、2，均小于 4。数字 4 和 7 是已就位的枢轴。剩余逆序对为 $(3,1), (3,2), (6,5), (9,8)$，共 4 个，全部在块内部。

3 插入阶段的代价

设小块长度为 $m_1, \dots, m_s$，满足 $m_i < k, \sum m_i \le n$。由于块间无逆序对：

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

这里使用的是便于书写的上界；严格 $m_i < k$ 还可给更小常数，但不改变阶。因此对**全数组**做插入排序的时间：

$$\Theta(n + I) = O(nk).$$

不能把这一阶段直接判成普通最坏 $\Theta(n^2)$，也不能不说明块间有序就假设只有 $O(nk)$。

4 截断快排的期望代价

理想均分时只需要约 $\lg(n/k)$ 层，每层 $O(n)$，可以直观得到目标式。要证明随机情况下的期望界，还需处理不均衡划分。

把一次长度为 m 的划分工作平均摊到其中各元素，每个元素每参与一次划分付常数成本。固定任意元素，跟踪其所在区间。均匀随机枢轴至少有一半概率落在中间一半的秩中；取到这样的好枢轴后，该元素或者直接就位退出，或者进入大小至多 $3m/4$ 的一侧。

从规模 n 降到小于 k，所需好划分次数为 $O(1 + \log(n/k))$。每次等到好划分的期望尝试数至多为 2，因此每个元素参与划分的期望次数为该对数量级。对全体元素求和：

$$\mathbb{E}[T_{\text{partial}}] = O\left(n + n\lg\frac{n}{k}\right).$$

不同元素的路径可以依赖，期望线性性仍成立。再加第二阶段：

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

当 $k = n$ 时 $\lg(n/k) = 0$，第一阶段仍要进行一次线性划分，因此单独写第一阶段界时要保留加法项 n；总界中的 nk 可吸收它。

5 阈值选择

选择	代入上界	结论
固定 $k \geq 2$	$O(n) + O(n\lg n)$	保持 $O(n\lg n)$
$k = \Theta(\lg n)$	$O(n\lg n) + O(n\lg(n/\lg n))$	仍是 $O(n\lg n)$
$k = \Theta(\sqrt{n})$	$O(n^{3/2}) + O(n\lg n)$	上界由插入阶段主导
$k = n$	$O(n^2)$	上界可能退化到平方级

理论上取固定小常数可以保持最优阶。若用带成本系数的模型：

$$F(k) = an\ln(n/k) + bnk, \qquad a, b > 0,$$

则 $F'(k) = -an/k + bn = 0$ 给出连续最优点 $k = a/b$，为常数。若使用 lg，相应常数为 $a/(b\ln 2)$。大 O 隐藏了这些系数，所以不能从渐近式算出适合所有机器的具体整数阈值。

实践中在固定实现、硬件和相同数据集上比较若干小阈值，覆盖随机、有序、逆序、近乎有序与重复值输入，多次运行后选择表现稳定的值。不要把未经测量的 10、16 或 32 当成题目推出的唯一答案。

解题方法。 “先说明第一阶段留下了什么结构；据此界定第二阶段工作；再单独证明第一阶段的期望成本；相加后吸收低阶项；最后代入不同阈值并解释成本常数。”

6 插入排序与逆序对

插入排序维护已经有序的前缀。处理新元素时，暂存该元素，把比它大的前驱右移，再填入空位。下面代码原地修改数组。

```python
def insertion_sort(a):
    for j in range(1, len(a)):
        key = a[j]
        i = j - 1
        while i >= 0 and a[i] > key:
            a[i + 1] = a[i]
            i -= 1
        a[i + 1] = key
    return a
```

每轮插入结果。 对 $[5,2,4,6,1,3]$：

待插入元素	本轮右移的值	本轮结束数组
2	5	$[2,5,4,6,1,3]$
4	5	$[2,4,5,6,1,3]$
6	无	$[2,4,5,6,1,3]$
1	6、5、4、2	$[1,2,4,5,6,3]$
3	6、5、4	$[1,2,3,4,5,6]$

不变式。 第 j 轮开始时，$A[0..j-1]$ 已有序，且恰好包含原输入的前 j 个元素。初始化时只有一个元素。保持时，大于 key 的元素右移，并把 key 放在第一个不大于它的前驱之后；有序性与元素数量均保持。终止时前缀覆盖全数组，所以输出正确。

逆序对。 满足 $i<j$ 且 $A[i]>A[j]$ 的一对位置称为逆序对。插入排序每次右移恰好让当前 key 越过一个这样的元素；每个原始逆序对恰好触发一次右移。因此：

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

样例右移总数 $1+1+0+4+3=9$，恰有 9 个逆序对。已经有序时 $I=0$，时间 $\Theta(n)$；严格逆序时 $I=\binom{n}{2}$，时间 $\Theta(n^2)$。右移次数与“比较次数”不同，失败的停止比较和边界检查需按题目指定口径另计。

稳定性。 代码只移动严格大于 key 的元素，相等元素不越过彼此，因此稳定。若改成 >=，稳定性会被破坏。用二分查找定位只能减少比较，数组仍可能需要二次总移动。

7 划分过程

下面采用 Lomuto 划分：把枢轴放在区间末尾，扫描时把不大于枢轴的值放到左侧。单次划分结束后，枢轴所在位置已最终确定，后续递归必须排除它。

```
from random import Random


def partition(a, left, right):
    pivot = a[right]
    boundary = left - 1
    for scan in range(left, right):
        if a[scan] <= pivot:
            boundary += 1
            a[boundary], a[scan] = a[scan], a[boundary]
    final = boundary + 1
    a[final], a[right] = a[right], a[final]
    return final
def randomized_partition(a, left, right, rng):
    chosen = rng.randrange(left, right + 1)
    a[chosen], a[right] = a[right], a[chosen]
    return partition(a, left, right)
```

扫描不变式。 扫描 `scan` 之前，$A[left..boundary]$ 均不大于枢轴，$A[boundary+1..scan-1]$ 均大于枢轴，$A[scan..right-1]$ 未检查，$A[right]$ 是枢轴。遇到小值时把它交换进左区；遇到大值时只扩展大值区。扫描结束后，把末尾枢轴放在两个区域之间。

例子 $[40,30,10,20]$，枢轴 20：扫描到 10 时交换成 $[10,30,40,20]$；末次交换得到 $[10,20,40,30]$，返回位置 1。两侧分别递归 $[0,0]$ 与 $[2,3]$，不能把位置 1 再递归进去。

区间长为 m 时，恰有 $m-1$ 次枢轴比较，时间 $\Theta(m)$，单次划分额外空间 $O(1)$。

8 随机快排的复杂度

以下先讨论**互异元素、均匀随机枢轴、没有有序预检查**的普通快排。

情形	递推或结构	时间
每次近似均分	$2T(n/2)+\Theta(n)$	$\Theta(n\log n)$
连续取当前极值	$T(n-1)+\Theta(n)$	$\Theta(n^2)$
对随机枢轴取期望	各枢轴秩等概率	$\Theta(n\log n)$

随机化改善期望，并不会让坏路径变得不可能。若固定取末尾为枢轴，有序或逆序数组可能触发二次时间。若每次划分前先检查是否已经有序，并在有序时直接返回，则该新算法在有序输入上的最好时间为 $\Theta(n)$，应与普通版本分开讨论。

递归栈在坏路径上可深达 $\Theta(n)$。只递归较小一侧、用循环处理较大一侧，可以把工作栈控制到 $O(\log n)$；这属于实现优化，不能把单次划分的常数空间直接写成原始递归快排的总空间。

设 C_n 为排序 n 个互异元素的期望枢轴比较次数，$C_0 = C_1 = 0$。枢轴秩等可能，因此：

$$C_n = n - 1 + \frac{1}{n}\sum_{r=1}^{n}(C_{r-1} + C_{n-r}) = n - 1 + \frac{2}{n}\sum_{i=0}^{n-1} C_i.$$

不能把随机划分直接假设为每次均分。乘以 n，与前一规模式子相减，可得对 $n \geq 2$：

$$nC_n = (n+1)C_{n-1} + 2(n-1),$$

$$\frac{C_n}{n+1} = \frac{C_{n-1}}{n} + \frac{2(n-1)}{n(n+1)}.$$

累加后：

$$C_n = 2(n+1)H_n - 4n = \Theta(n\lg n).$$

例如 $C_2 = 1, C_3 = 8/3$。这是期望比较次数，允许不是整数；一次实际运行的比较次数当然为整数。

指示变量的另一种证明。 将元素按大小记作 $z_1 < \cdots < z_n$。z_i, z_j 被比较，当且仅当在这段秩区间中最先成为枢轴的是其中某个端点，概率为 $2/(j-i+1)$。对所有元素对的指示变量求和，也得到 $\Theta(n\log n)$。期望线性性不要求这些事件独立。

9 逆序对计数变式

这是归并排序的补充变式。递归求左右内部逆序对，合并时计算跨区间逆序对。左右已经有序，当 $L[i] > R[j]$，左边从 i 起的所有剩余值均大于 $R[j]$，所以一次增加：

$$|L| - i$$

个跨区间逆序对，然后取出 $R[j]$。若 $L[i] \leq R[j]$，先取 $L[i]$，不增加逆序对。时间递推仍为 $2T(n/2) + \Theta(n)$，所以 $\Theta(n\log n)$。

对 $[3,1,2]$，逆序对是 $(3,1),(3,2)$，答案 2。相等元素不构成逆序对，比较条件必须相应处理。

十 分层自测题

答案见第十三章。

1 概念与短题

D01 渐进关系。 对以下四对 (A,B) 分别填写五列 $O, o, \Omega, \omega, \Theta$：① $(3n^2 + 2n, n^2)$；② $(n\lg n, n^{1.1})$；③ $(4^n, 2^{2n})$；④ $(n(2 + \sin n), n)$。

D02 指数化简。 化简 $8^{\lg n}$、$n^{3/\lg n}$、$(\lg n)^{\lg n}$、$2^{\lg\lg n}$，指出哪些与 n^3 同阶。

D03 函数排序。 将 $1, \lg\lg n, \sqrt{\lg n}, (\lg n)^3, n^{0.01}, n, 2^n, n!$ 按由慢到快排列，并说明 $(\lg n)^3$ 与 $n^{0.01}$ 的比较理由。

D04 命题证明。 若 $f = O(g)$ 且 $g = O(h)$，证明 $f = O(h)$。写出常数与阈值如何合并。

D05 命题反例。 否定“$f = \Theta(g)$ 推出 $3^f = \Theta(3^g)$”，并给出一个足以推出该指数结论的更强条件。

D06 循环计数。 外层 $i = 1, \ldots, n$，内层 $j = 1, \ldots, i$，求总次数；若每轮内层从 $j = 1$ 开始、执行 `j *= 2` 直到超过 n，求总次数的渐近阶。

D07 稳定性。 对选择排序 $[2_a, 2_b, 1]$ 给出反例轨迹；解释插入排序把 `>` 改为 `>=` 后会发生什么。

D08 作业算法边界。 分别给出：非空最大子数组 $[-4, -2, -7]$ 的和与端点；两数之和输入 $[5]$、目标 10 的判断结果；多路归并输入为空表集合、或只有一个表 $[1,1,3]$ 时的输出。说明各题最容易误用的初始化或循环条件。

D09 随机算法。 判断并说明理由：“随机快排每次划分期望均衡，所以每条实际路径都只有对数层。”

D10 排序方案选择。 原题 2.3-7 要求 $\Theta(n\lg n)$ 时间。为什么归并排序加双指针满足要求，而固定末尾枢轴的快排加双指针不能保证？若改用随机快排，必须怎样表述其时间结论？

D11 堆操作。 堆内有 k 个候选，查看最小值与删除最小值的时间分别是什么？为什么自底向上建堆是 $O(k)$？

D12 比较决策树。 有 20 个必须区分的候选状态，每次比较只有两种结果。最坏至少需要几次比较？为什么叶数容量足够不自动证明存在达到该次数的合法比较算法？

2 计算与完整解题

D13 主定理。 求下列递推式的紧确界并说明情况：① $3T(n/3) + n$；② $9T(n/3) + n$；③ $T(n/2) + n$；④ $4T(n/2) + n^3$。

D14 不适用主定理。 求 $T(n) = T(n-3) + n$ 和 $T(n) = T(\sqrt{n}) + 1$ 的紧确界，给出展开或终止条件。

D15 对数变式。 求 $2T(n/2) + n\lg n$ 和 $2T(n/2) + n/\lg n$ 的紧确界。说明基本主定理哪里不能直接用。

D16 插入过程。 对 $[4,1,3,2]$ 写出每轮插入后的数组，并给出右移总次数和逆序对清单。

D17 归并过程。 合并 $[1,4,6]$ 与 $[2,3,5]$，列出每次键比较和输出，总比较次数是多少？

D18 两数之和。 对 $[-5, -2, 0, 3, 7, 10]$、目标 8，列出双指针轨迹。再说明目标 20 时为什么不能重复使用值 10。

D19 最大子数组。 对 $[2, -5, 3, 1, -2, 4, -6, 2]$ 列出每个位置的 E_j, B_j，给出一组最优端点。

D20 前缀和。 对 $[3, -1, 4, -2]$ 建立前缀和，求闭区间 $[1,3]$ 的和；说明一组下标差 1 的常见错误。

D21 多路归并。 合并 $[1,4]$、空表、$[1,3,5]$、$[2]$。列出输出序列、初始堆节点数和最大堆节点数。

D22 快排划分。 用本讲义的 Lomuto 划分处理 $[9,3,7,1,6]$，末尾 6 为枢轴。写出每次有效交换后的数组及最终枢轴位置。

D23 混合排序。 设 $k=4$，第一阶段结果为 $[2,1,3],4,[6,5],7,[9,8]$。剩余逆序对多少？最终插入排序右移几次？若 $k=\sqrt{n}$，代入总上界。

D24 混合排序阈值。 将 k 的四种取值 $2,\Theta(\lg n),\Theta(\sqrt{n}),n$ 分别代入 $O(nk+n\lg(n/k))$，化简上界。为什么不能推出所有机器通用的最佳整数阈值？

3 迁移与证明

D25 不等规模分治。 证明 $T(n)=T(n/3)+T(2n/3)+n$ 为 $\Theta(n\lg n)$，不得直接套等规模主定理。

D26 逆序对算法。 描述利用归并排序计数逆序对的算法。为什么合并时 $L[i]>R[j]$ 可一次增加 $|L|-i$？

D27 连续区间优化。 最大子数组要同时输出左右端点时，只保存 `best` 为什么不够？还需哪些变量，何时更新？

D28 块间顺序反例。 设 $n=2m,k=3$，数组为 $[2m-1,2m],[2m-3,2m-2],\ldots,[1,2]$，每块内部升序。求跨块逆序对数；为何“小块短于 k”不足以推出 $O(nk)$？截断快排还保证什么？

D29 归并代价比较。 合并 $k=\Theta(\sqrt{n})$ 个非空有序表，总元素数 n。比较扫描表头与最小堆的总时间上界，说明堆初始化成本，以及堆顶查询 $O(1)$ 为什么不代表总时间 $O(n)$。

D30 比较下界。 对 6 个互异元素，求任意比较排序的最坏比较次数下界，再证明一般 n 的界。说明为何此下界不保证存在恰好达到该次数的算法。

十一 模拟卷 A

自拟题，满分 100 分。

A01 渐进关系 12 分。 三对函数分别填 $O, o, \Omega, \omega, \Theta$ 并说明理由，每对 4 分：① $((\lg n)^4, \sqrt{n})$；② $(n^{\lg 5}, 5^{\lg n})$；③ $(2^{n+3}, 2^n)$。

A02 函数排序 8 分。 将 $1, \lg n, n^{1/\lg n}, \sqrt{n}, n\lg n, 3^n, n!, 2^{2^n}$ 按由慢到快排序并同阶分组。

A03 证明与反例 10 分。 每问 5 分：① 证明最终为正的 f, g 满足 $f + g = \Theta(\max(f, g))$；② 否定“$f = O(g) \Rightarrow 2^f = O(2^g)$”。

A04 选择排序 12 分。 对 $[4,1,3,2]$ 使用每轮选最大放末尾的版本：写前两轮数组 4 分；写外层不变式及终止结论 4 分；给出键比较总数、最好和最坏时间 4 分。

A05 递推式 16 分。 各求紧确界并说明方法，每问 4 分：① $3T(n/3) + n$；② $9T(n/3) + n$；③ $T(n-3) + n$；④ $T(n/2) + n$。

A06 两数之和 14 分。 对 $S = \{-3,0,2,4,7,11\}$、目标 9，设计 $\Theta(n\lg n)$ 判定算法：算法与样例轨迹 6 分；证明指针移动不漏解 5 分；时间与空间 3 分。

A07 最大子数组 16 分。 对 $[-1,4,-2,3,-5,2]$：写线性非递归算法或状态递推 6 分；列 E_j, B_j 与答案区间 6 分；说明全负边界、时间与空间 4 分。

A08 比较排序下界 12 分。 从决策树证明 $\Omega(n\lg n)$：候选状态与叶数 4 分；树高不等式 4 分；$\lg(n!)$ 的界与适用范围 4 分。

十二 模拟卷 B

自拟题，满分 100 分。

B01 五项辨析 10 分。 每项 2 分，判断并简要说明：① 选择排序有序输入时为 $\Theta(n)$；② 比值无极限就不可能 Θ；③ 普通随机快排最坏为 $\Theta(n^2)$；④ 堆归并中每次取出并补入候选总能按 $O(1)$ 计算；⑤ 截断快排停止后，不同终止小块之间没有逆序对。

B02 递推进阶 12 分。 每问 4 分，求紧确界并说明推导：① $2T(n/2)+n\lg n$；② $T(\sqrt{n})+1$；③ $T(n/3)+T(2n/3)+n$。

B03 堆归并 16 分。 合并 $[0,5,9]$、$[1,3]$、空表、$[2,4,8]$：算法与节点设计 5 分；不变式 4 分；前四个输出及堆中代表 3 分；时间、空表成本、工作空间与输出空间 4 分。

B04 快排分析 14 分。 对 $[8,2,5,1,7,3]$ 取末尾 3 为枢轴，执行本讲义划分：最终数组与枢轴位置 4 分；写互异元素、均匀随机枢轴下的期望比较次数递推 4 分；说明期望 $\Theta(n\lg n)$ 与最坏 $\Theta(n^2)$ 为什么可以同时成立，不能把随机划分直接当成每次均分，6 分。

B05 混合排序 20 分。 证明原题 $O(nk+n\lg(n/k))$ 期望界：块间有序 4 分；逆序对与插入代价 6 分；截断快排期望成本 6 分；理论阈值与实践阈值 4 分。必须说明元素、随机枢轴与截止条件。

B06 最大子数组状态 12 分。 对 $[3,-4,2,-1,6,-5]$：列出每个位置的结尾最优 E_j 与全局最优 B_j，4 分；给出最大和、端点与全负输入的初始化，4 分；解释为什么不能用 $B_{j-1}+A[j]$ 代替 $E_{j-1}+A[j]$，并用本数组指出该错误会虚构什么结果，4 分。

B07 振荡函数 8 分。 对原题中的 $A=\sqrt{n}, B=n^{\sin n}$ 判断 $A=O(B), o(B), \Omega(B), \omega(B), \Theta(B)$ 是否成立。写出比值，并说明怎样用两条子序列排除最终上界和最终正下界；不能只用“振荡”二字作答。

B08 反例与定义证明 8 分。 每问 4 分：① 证明 $(n+1)!$ 与 $n!$ 不同阶；② 在 $f=O(g)$ 且足够大时 $\lg g\geq 1, f\geq 1$ 的条件下，证明 $\lg f=O(\lg g)$，明确所取常数及两个附加条件的作用。

十三 自测答案

1 分层自测答案

D01。 按 $O, o, \Omega, \omega, \Theta$ 顺序，① 是、否、是、否、是，因比值趋于 3；② 是、是、否、否、否，因 $\lg n / n^{0.1} \to 0$；③ 是、否、是、否、是，因 $4^n = 2^{2n}$；④ 是、否、是、否、是，因比值始终在 $[1,3]$，不需要有极限。

D02。 四式依次为 $n^3, 8, n^{\lg\lg n}, \lg n$。只有第一式与 n^3 同阶。第三式的指数 $\lg\lg n$ 随 n 增长，不是固定三次幂。

D03。

$$1 \prec \lg\lg n \prec \sqrt{\lg n} \prec (\lg n)^3 \prec n^{0.01} \prec n \prec 2^n \prec n!.$$

令 $t = \ln n$，$(\lg n)^3 / n^{0.01}$ 为常数倍 $t^3 / e^{0.01t} \to 0$，所以任意固定对数幂最终慢于该正次幂。

D04。 存在 $c_1, c_2 > 0, n_1, n_2$，使足够大时 $f \le c_1 g, g \le c_2 h$。取 $n_0 = \max(n_1, n_2)$、$c = c_1 c_2$，得到 $f \le ch$。原函数最终为非负的阈值也并入 n_0 即可。

D05。 取 $f = 2n, g = n$，有 Θ，但 $3^f / 3^g = 3^n \to \infty$，指数后不同阶。足够强的条件是 $|f - g| \le C$，此时比值夹在 3^{-C} 与 3^C 之间，所以同阶。只知道差为 $O(\log n)$ 仍可能产生无界的多项式比值。

D06。 第一种总数为 $\sum_{i=1}^{n} i = n(n+1)/2$。第二种内层从 1 开始倍增，每轮外层执行 $\lfloor \lg n \rfloor + 1$ 次，合计 $n(\lfloor \lg n \rfloor + 1) = \Theta(n \lg n)$。这里外层的每轮内层上限都是 n。

D07。 首次选择最大值 2_a 与 1 交换，得到 $[1, 2_b, 2_a]$，不稳定。插入排序使用 `>=` 会让新来的等值元素越过前面已经出现的等值元素，例如 $[2_a, 2_b]$ 变为 $[2_b, 2_a]$。稳定版本应只右移严格大于键的前驱。

D08。 非空最大子数组为 $[-2]$，最大和 -2，端点 $[1,1]$；必须用真实首元素初始化，不能用虚构的空区间 0。单元素 $[5]$ 不能凑出 10，双指针要求 `left < right`，禁止重复使用同一位置。没有输入表时输出空列表；只有 $[1,1,3]$ 时原样输出全部三个元素，重复值不能丢。空表不取首元素、不入堆；单表归并仍须输出全部元素。

D09。 错。条件期望不能保证每条路径。每次取极值的随机序列仍有正概率发生，使规模依次为 $n, n-1, \ldots$，总扫描为二次。期望 $\Theta(n \lg n)$ 是对随机选择求平均后的结论。

D10。 标准归并排序在所有输入上为 $\Theta(n\lg n)$，双指针至多线性，合计保证 $\Theta(n\lg n)$。固定末尾枢轴的快排遇到已排序输入时递推为 $T(n)=T(n-1)+\Theta(n)$，最坏 $\Theta(n^2)$，因此不能提供所要求的保证。若用均匀随机枢轴且元素互异，可写"期望 $\Theta(n\lg n)$、最坏 $\Theta(n^2)$"；原题未把要求限定为期望时，使用归并排序即可避免额外条件。

D11。 查看 $O(1)$，删除 $O(\log(k+1))$，对 $k\geq 2$ 常写 $O(\log k)$。建堆按节点高度求和，$\sum_h O\left(kh/2^h\right)=O(k)$，不能将每个节点都按最高高度计算为紧确界。

D12。 最坏至少 $\lceil\lg 20\rceil=5$ 次，因为 $2^4=16<20\leq 32=2^5$。高度为 4 的二叉决策树容纳不了 20 个可区分叶子。高度 5 的叶数容量够用，只说明通过这一必要条件；还需构造每个节点都是合法比较、且能正确区分全部状态的算法，才能证明 5 次足够。

D13。 ① $a=3,b=3,p=1$，第二种，$\Theta(n\lg n)$。② $a=9,b=3,p=2$，$f=n$ 小一个幂，第一种，$\Theta(n^2)$。③ $a=1,b=2,p=0$，$f=n$ 大一个幂且正则比 $1/2$，第三种，$\Theta(n)$。④ $a=4,b=2,p=2$，$f=n^3$ 且正则比 $4/8=1/2$，第三种，$\Theta(n^3)$。

D14。 第一式展开为 $n+(n-3)+(n-6)+\cdots$，共有 $\Theta(n)$ 项，上界 $O(n^2)$，前 $\Theta(n)$ 项均为 $\Omega(n)$，故为 $\Theta(n^2)$。第二式 h 次后的规模为 $n^{1/2^h}$，停止要求 $\lg n/2^h=O(1)$，故 $h=\Theta(\lg\lg n)$，每层常数，答案相同。

D15。 令 $n=2^m$、$U_m=T(2^m)/2^m$，第一式 $U_m=U_{m-1}+m$，得到 $\Theta(n\lg^2 n)$；第二式 $U_m=U_{m-1}+1/m$，得到 $\Theta(n\lg\lg n)$。二者与 n 的差别是对数因子，不符合基本第一、第三种的固定多项式差距，第一式也不等于 $\Theta(n)$。

D16。 各轮数组为 $[1,4,3,2]$、$[1,3,4,2]$、$[1,2,3,4]$。右移次数为 $1+1+2=4$；逆序对为 $(4,1),(4,3),(4,2),(3,2)$。若题目问键比较数，不能把 4 直接当作全部比较数。

D17。 比较与输出为：$1<2$ 输出 1；$4>2$ 输出 2；$4>3$ 输出 3；$4<5$ 输出 4；$6>5$ 输出 5；右表空后复制 6。5 次键比较，输出 $[1,2,3,4,5,6]$。

D18。 先比较 $-5+10=5<8$，左移到 -2；随后 $-2+10=8$，找到答案。目标 20 时最大两个不同元素之和也只有 $7+10=17$，无解；循环使用 `left < right`，所以不能把下标对应的同一个 10 用两次。

D19。

j	0	1	2	3	4	5	6	7
E_j	2	-3	3	4	2	6	0	2

j	0	1	2	3	4	5	6	7
B_j	2	2	3	4	4	6	6	6

最大和 6，端点 $[2,5]$，内容 $[3,1,-2,4]$。

D20。 前缀数组 $P=[0,3,2,6,4]$。闭区间 $[1,3]$ 的和为 $P[4]-P[1]=4-3=1$。$P[3]-P[1]=3$ 对应的是 $[1,2]$，漏掉右端位置 3 的元素。

D21。 输出 $[1,1,2,3,4,5]$。非空表有 3 个，初始化 3 个节点，过程中最多仍为 3 个。空表不入堆；两个相等的 1 均应输出。

D22。 有效交换依次得到 $[3,9,7,1,6]$、$[3,1,7,9,6]$，最后放枢轴得到 $[3,1,6,9,7]$，枢轴下标 2。左右区间分别满足不大于 6 和大于 6，但不要求区间内部已经有序。

D23。 逆序对为 $(2,1),(6,5),(9,8)$，共 3 个，插入排序右移 3 次。若 $k=\sqrt{n}$，总上界 $O(n^{3/2}+n\lg\sqrt{n})=O(n^{3/2})$。这个代入给上界，不能在未另证下界时把它自动改写成 $\Theta(n^{3/2})$。

D24。

阈值	总时间上界	代入依据
$k=2$	$O(n\lg n)$	$nk=2n$，对数项为 $n\lg(n/2)$
$k=\Theta(\lg n)$	$O(n\lg n)$	两项均不超过该阶
$k=\Theta(\sqrt{n})$	$O(n^{3/2})$	nk 的上界主导对数项
$k=n$	$O(n^2)$	对数项为 0，总上界由插入阶段吸收

这些是代入所得的上界，不能无证明地改为紧确界。大 O 隐藏划分、递归、比较与移动的成本系数；具体整数阈值要在固定实现和机器上测量，不能从渐近式直接推出。

D25。 最长路径每次至多保留 $2/3$ 规模，深度 $O(\lg n)$；每层子问题规模之和不超过 n，所以总上界 $O(n\lg n)$。前 $\lfloor\log_3 n\rfloor-O(1)$ 层内，即使每次都走最小的 $1/3$ 分支，规模仍未降到基础值，因此这些层无叶子、总规模均为 n，总下界 $\Omega(n\lg n)$。叶子工作至多线性，不改变结论。

D26。 递归计数两半内部逆序对，合并时累计跨区间逆序对。若 $L[i]>R[j]$，左半有序使得所有 $L[i],\dots,L[|L|-1]$ 都大于 $R[j]$，而这些元素在原数组又位于右半之前，所以增加 $|L|-i$。输出 $R[j]$ 后其跨

区间贡献已全部计算，不重计。若相等，先取左边不加计数。每层线性、共对数层，时间 $\Theta(n\lg n)$，空间 $O(n)$。

D27。 `best` 只记录和，无法恢复区间。再保存当前结尾最优区间的起点 `start`，以及全局 `best_left,best_right`。若旧结尾和为负、选择从当前元素重新开始，就令 `start=j`；每次新结尾和严格改善全局最佳时，同时保存 `(start,j)`。相等时按题目需要的规则保留或比较端点。

D28。 任取两个不同块，前块的两个元素都大于后块的两个元素，产生 $2\times 2=4$ 个逆序对。块内部没有逆序对，因此总数为：

$$4\binom{m}{2}=2m(m-1)=\frac{n(n-2)}{2}=\Theta(n^2).$$

固定 $k=3$ 时，$O(nk)=O(n)$，与此反例的平方级不符。因此块短还不够。截断快排额外保证不同终止块按值域有序，前块任意元素均小于后块任意元素，逆序对只留在块内，才可用 $\sum_i\binom{m_i}{2}=O(nk)$。

D29。 每输出一个元素都扫描至多 k 个表头，总上界 $O(nk)=O(n^{3/2})$。最小堆自底向上初始化为 $O(k)$，随后 n 次取出及至多 n 次补入各为 $O(\lg k)$，合计 $O(k+n\lg k)=O(n\lg n)$，这里 $\lg k=\Theta(\lg n)$。查看堆顶虽为 $O(1)$，删除堆顶、补入后继仍须恢复堆性质，这些更新成本不能省略。

D30。 $6!=720$，$2^9=512<720\le 1024=2^{10}$，最坏至少 10 次。一般 n 个互异元素有 $n!$ 种相对次序，二叉比较决策树需至少 $n!$ 个叶子。树高为 h 时叶数至多 2^h，故 $h\ge\lceil\lg(n!)\rceil$。由 $(n/2)^{n/2}\le n!\le n^n$ 得 $\lg(n!)=\Theta(n\lg n)$，所以最坏下界为 $\Omega(n\lg n)$。叶数容量不保证存在恰好达到该高度的合法比较树；此界表示每个正确比较排序都存在难输入。

2 模拟卷 A 参考答案

A01 12 分。 ① O,o 成立，其余否，因 $(\lg n)^4/\sqrt{n}\to 0$。② O,Ω,Θ 成立，其余否，因两式恒等。③ O,Ω,Θ 成立，其余否，因比值为 8。每对关系填写 3 分、理由 1 分。

A02 8 分。

$$\{1,n^{1/\lg n}\}\prec\{\lg n\}\prec\{\sqrt{n}\}\prec\{n\lg n\}\prec\{3^n\}\prec\{n!\}\prec\{2^{2^n}\}.$$

常数组合并 2 分，其余严格顺序 6 分。原题的 8 个函数必须全部出现，不能漏掉与 1 同阶但不恒等于 1 的 $n^{1/\lg n}=2$。

A03 10 分。 ① 因最终为正，有 $\max(f,g) \le f+g \le 2\max(f,g)$，直接取 Θ 常数 1 与 2 即可，5 分。② 取 $f=2n, g=n$，前提成立 2 分，比值 $2^n \to \infty$ 排除结论 3 分。

A04 12 分。 第一轮最大值 4 与末尾 2 交换，得到 $[2,1,3,4]$；第二轮最大值 3 已在当前末尾，数组仍为 $[2,1,3,4]$，各 2 分。不变式是“每轮开始时已确定后缀有序，包含全体最大若干元素，且剩余前缀元素不大于后缀元素”；终止时只剩一个最小值，连同后缀全局有序，4 分。键比较数 $3+2+1=6$，最好与最坏均 $\Theta(n^2)$，4 分。

A05 16 分。 ① $p=1, f=n$，第二种，$\Theta(n\lg n)$。② $p=2, f=n$，第一种，$\Theta(n^2)$。③ 展开等差和，共 $\Theta(n)$ 项，和为 $\Theta(n^2)$。④ 单链成本 $n+n/2+n/4+\cdots$，$\Theta(n)$，也可主定理第三种并检查正则比 $1/2$。每问答案 2 分、符合条件的论证 2 分。

A06 14 分。 归并排序后双指针，和小则左移、和大则右移。轨迹是 $-3+11=8$、$0+11=11$、$0+7=7$、$2+7=9$，找到 2 与 7，6 分。若和小，左值与当前任何其他值配对都不大于与右端配对的和，因此左值无解；和大时同理排除右值。不变式保留所有可能答案，最终不足两个位置则无解，5 分。时间 $\Theta(n\lg n)$，额外空间 $O(n)$，3 分。

A07 16 分。 状态 $E_j = \max(A[j], E_{j-1}+A[j])$、$B_j = \max(B_{j-1}, E_j)$，初值为首元素；给出逐位置循环即可，6 分。状态表：

j	0	1	2	3	4	5
$A[j]$	-1	4	-2	3	-5	2
E_j	-1	4	2	5	0	2
B_j	-1	4	4	5	5	5

最大和 5，区间 $[1,3]$，即 $[4,-2,3]$；状态与区间共 6 分。全负时返回最大负元素，不初始化为虚构的 0；一遍扫描时间 $\Theta(n)$、只保留滚动状态及端点空间 $O(1)$，4 分。

A08 12 分。 n 个互异元素共有 $n!$ 种相对次序，算法需正确区分，叶数至少 $n!$，4 分。每次比较两个结果，树高 h 最多 2^h 个叶子，所以 $h \ge \lg(n!)$，4 分。用后半个阶乘因子给 $\Omega(n\lg n)$、用 $n! \le n^n$ 给 $O(n\lg n)$，得到所需下界；说明是只通过比较的排序的最坏界，4 分。

3 模拟卷 B 参考答案

B01 10 分。 ① 错，仍完整扫描，$\Theta(n^2)$。② 错，如比值 $2+\sin n$ 始终在 $[1,3]$，仍同阶。③ 对，连续极端枢轴可造成二次扫描。④ 错，查看堆顶为常数，但删除及补入的最坏成本为 $O(\lg k)$，不能统一按常数计。⑤ 对，不同终止块首次被某个祖先枢轴分离，左块元素小于右块元素，故块间无逆序对。每项判断 1 分、依据 1 分。

B02 12 分。 ① 令 $U_m = T(2^m)/2^m$，每轮增量 m，所以 $\Theta(n\lg^2 n)$。② $n^{1/2^h}$ 降到常数要求 $h=\Theta(\lg\lg n)$，答案相同。③ 每层总规模不超过 n、最深 $O(\lg n)$ 得上界；前 $\Theta(\lg n)$ 层无叶且层成本为 n 得下界，结果 $\Theta(n\lg n)$。每问结论 2 分、证明 2 分。

B03 16 分。 每个非空表取一个三元组 (值,表号,位置) 形成最小堆；每次输出堆顶后补本表后继，5 分。不变式为输出前缀正确且各未用完表恰有最小未输出项在堆中；堆顶因此是全局最小，4 分。初始代表为 $\{0,1,2\}$，输出及之后代表依次是：0 后 $\{1,2,5\}$；1 后 $\{2,3,5\}$；2 后 $\{3,4,5\}$；3 后 $\{4,5\}$，3 分。$n=8,k=4,q=3$，完整时间 $O(k+n\log(q+1))$，通常写 $O(n\lg k)$；堆工作空间 $O(q)$，结果列表另为 $\Theta(n)$，4 分。

B04 14 分。 扫描 2 时得到 $[2,8,5,1,7,3]$，扫描 1 时得到 $[2,1,5,8,7,3]$，最后枢轴交换得到 $[2,1,3,8,7,5]$，枢轴下标 2，4 分。期望比较递推：

$$C_n = n-1+\frac{2}{n}\sum_{i=0}^{n-1} C_i\,,\qquad C_0=C_1=0,$$

4 分。期望是固定输入后对枢轴随机选择取平均，最坏则可包含连续选到当前极值的随机路径。互异元素下极值路径的概率虽小但为正，子问题规模依次降为 $n-1,n-2,\ldots$，总成本为 $\sum_{m=2}^{n}\Theta(m)=\Theta(n^2)$，3 分。按秩等概率求和得到 $C_n=2(n+1)H_n-4n=\Theta(n\lg n)$，两种结论针对的取值方式不同，可以同时成立；只画均分树没有处理全部枢轴秩，不能代替期望证明，3 分。

B05 20 分。 需写明互异元素、均匀随机枢轴、$2\le k\le n$、长度严格小于 k 才停止。两终止块首次被某祖先枢轴分离，故前块任何元素均小于后块元素，块间无逆序，4 分。令块长为 $m_i<k$，剩余逆序数：

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

全数组插入排序 $\Theta(n+I)=O(nk)$，6 分。按元素分摊划分成本，好枢轴概率至少 $1/2$，缩小到 k 前期望参与 $O(1+\lg(n/k))$ 次，故第一阶段 $O(n+n\lg(n/k))$；相加得到要求的期望界，6 分。理论取常数保持 $O(n\lg n)$，若考虑系数模型则常数由成本比决定；实践对固定实现和硬件做重复测量，4 分。

B06 12 分。 状态如下，4 分：

j	0	1	2	3	4	5
E_j	3	-1	2	1	7	2
B_j	3	3	3	3	7	7

最大和为 7，区间 $[2,4]$，内容 $[2,-1,6]$。全负输入时 $E_0 = B_0 = A[0]$，最终返回最大负元素，4 分。B_{j-1} 对应的区间可能早已结束，未必与位置 j 相邻，不能直接续接。这里 $j = 4$ 时若计算 $B_3 + A[4] = 3 + 6 = 9$，就是把位置 0 的 3 与位置 4 的 6 拼在一起并跳过中间元素；真实结尾最优为 $E_3 + A[4] = 1 + 6 = 7$，4 分。

B07 8 分。 比值为 $R(n) = n^{1/2-\sin n}$，2 分。分别取最接近 $\pi/2 + 2\pi j$ 与 $3\pi/2 + 2\pi j$ 的整数作为两条趋于无穷的子序列；取整误差至多 $1/2$，前者满足 $\sin n > 3/4$，后者满足 $\sin n < 1/4$。于是前者上 $R(n) \le n^{-1/4} \to 0$，后者上 $R(n) \ge n^{1/4} \to \infty$，4 分。因此不存在最终常数上界，也不存在最终正下界，五种关系全部不成立，2 分。

B08 8 分。 ① 比值 $(n+1)!/n! = n + 1 \to \infty$，所以前者是后者的 ω，并非 Θ，4 分。② 由 $f = O(g)$ 取 $C \ge 1$ 和阈值 n_0，使 $n \ge n_0$ 时 $f \le Cg$，并把两个附加条件成立的阈值并入 n_0。于是：

$$0 \le \lg f \le \lg C + \lg g \le (1 + \lg C)\lg g.$$

取新的常数 $C' = 1 + \lg C > 0$，即得目标 O 界，2 分。$f \ge 1$ 保证 $\lg f \ge 0$，$\lg g \ge 1$ 允许把加法常数 $\lg C$ 吸收到 $\lg g$ 的常数倍中，2 分。