

算法期中复习提纲

2026 课件 × 2023–2025 年期中真题 · 中文整理版

使用说明与复习优先级

这份提纲以本次提供的 8 份课件为知识范围，以 2023、2024、2025 年的试卷和答案为练习依据。重点是能解释结论、补全伪代码、证明复杂度、画出数据结构操作过程。历年出现频次只代表这三份试卷，不代表 2026 年命题承诺。

页码约定：以下“p.”均指 PDF 阅读器显示的实际页码，从 1 开始，包含动画展开页；不是幻灯片页脚编号。“判断 1”指判断题第 1 题，其他题型同理。课件引用简写及完整文件名见文末来源索引。

优先级	模块	三年试卷中的直接依据	复习完成标准
A：先掌握	渐近分析与递归式	三年均有判断 1、简答 1；2025 另有判断 2	能独立求界，并说明主定理是否适用
A：先掌握	排序、线性排序、选择	三年均有多道排序题；2023 综合 1、2024 综合 1、2025 综合 1	会性质比较、稳定计数排序、PARTITION、QuickSelect
A：先掌握	斐波那契堆	三年简答 4 均为抽取最小值后减键与级联切断，每题 10 分	不看答案画完整操作过程，标记正确
A：先掌握	BST、红黑树、数据结构扩充	三年判断 6；2024 简答 3、2025 简答 3 和综合 2	能定义扩充字段、写维护式、说明旋转后如何更新
A：先掌握	双指针、排序后贪心、滑动窗口	2023 简答 3；2024、2025 简答 2	会写伪代码并证明指针移动不会漏解
B：完整覆盖	二项堆、并查集、Shell 排序、区间树	都在本次课件内；这三份真题未直接单独设题	理解机制、复杂度及适用条件，不因未考过而跳过
B：真题补充	AVL、Treap、随机 BST 的期望分析	分别见 2023、2024、2025 综合 2	能从题目给定定义推导；不要求假设已系统讲授
C：冲刺拓展	三年附加题	区间和计数、桶与窗口、受限交换	基础熟练后练建模、分治和连通性

三年试卷均为：判断 6×5 分、简答 4×10 分、综合 2×15 分，共 100 分，另有 10 分附加题。判断题不仅要选对，还要写出理由；练习时按“结论 + 关键理由/反例”作答。

一、算法分析的基础工具

课件定位：T1 p.35–51、57–69；T2 p.4–32。

1.1 必须区分的概念

- **最坏情况**：在所有规模为 n 的输入上取最大代价。
- **平均情况**：指定输入分布后，对输入取期望。例如随机排列中每个排列等概率。
- **随机算法的期望代价**：固定输入，对算法内部的随机选择取期望。随机快速排序不要求输入本身随机。
- **摊还代价**：对一串操作的总代价进行分摊，不需要概率假设。斐波那契堆的摊还 $O(1)$ 减键，不等于每次实际都只花常数时间。
- **时间与空间分别分析**；递归栈也计入辅助空间。

1.2 渐近记号

对最终非负的函数， O 表示渐近上界， Ω 表示渐近下界， Θ 表示上下界同时成立。

$$f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \text{ 且 } f(n) = \Omega(g(n)).$$

$o(g(n))$ 是严格低阶： $f(n)/g(n) \rightarrow 0$ ； $\omega(g(n))$ 是严格高阶：比值趋于无穷。记住定义中的量词： O 是“存在正常数”， o 是“对任意正常数”。函数比值不一定有极限，也不是任意两个函数都能建立同阶或单向上下界关系。

判断题陷阱： $n = O(n^2)$ 是真的，只是不紧；“不是 $\Theta(n^2)$ ” 不能推出 “不是 $O(n^2)$ ”。

1.3 常用增长率与求和

对固定常数 $a > 0, c > 1, k > 0$ ，有 $\lg^k n = o(n^a)$ 、 $n^a = o(c^n)$ 。 $\lg n = \log_2 n$ ， $\lg^2 n = (\lg n)^2$ ， $\lg \lg n$ 是两次取对数，三者不要混淆。

$$\sum_{i=1}^n i = \Theta(n^2), \quad H_n = \sum_{i=1}^n \frac{1}{i} = \Theta(\log n).$$

$$\sum_{i=0}^L q^i = \frac{1 - q^{L+1}}{1 - q} \quad (q \neq 1), \quad \lg(n!) = \Theta(n \lg n).$$

还应认识 $\lg^* n$ 是重复取对数直到不超过 1 的次数；并查集中的 $\alpha(n)$ 是反阿克曼函数，增长极慢，但不是同一个函数。

交换求和顺序的典型题：2025 判断 2。 试卷的 $\sigma(i)$ 表示因子个数，按“每个因子被多少个数字拥有”统计：

$$\sum_{i=1}^n \sigma(i) = \sum_{d=1}^n \left\lfloor \frac{n}{d} \right\rfloor = \Theta(n \log n).$$

上界为 nH_n ，下界为 $nH_n - n$ 。不能把每个数的试除耗时 $O(\sqrt{n})$ ，误当作其因子个数。

1.4 期望分析的通用方法

定义指示变量 I_E ：事件发生时为 1，否则为 0，因此 $\mathbb{E}[I_E] = \Pr(E)$ 。再把总代价写成指示变量之和。

$$\mathbb{E}\left[\sum_i X_i\right] = \sum_i \mathbb{E}[X_i].$$

期望的线性性不要求各变量独立；计算 $\mathbb{E}[X_i X_j]$ 时，若要直接写成乘积，则需要相应的独立性等条件。后续快排比较次数、桶中元素数的平方、随机 BST 祖先个数，都复用这套方法。

二、递归式：求解方法与真题题库

课件定位：T2 p.34–47（代入法）、48–56（递归树）、57–73（主定理与级数）。

2.1 解题顺序

- 1. 写清基本情形与规模如何缩小，识别 $a, b, f(n)$ 。
- 2. 形式符合 $T(n) = aT(n/b) + f(n)$ 时，先比较 $f(n)$ 与 $n^{\log_b a}$ 。
- 3. 不满足条件时，使用递归树、变量替换、展开或代入法。不同规模的多个递归项不能直接套标准主定理。
- 4. 若题目要紧界，必须同时说明上下界；题目只要上界时，不能把上界当成已经证明的紧界。

2.2 主定理的三种常见情况

令 $p = \log_b a$ ，其中 $a \geq 1, b > 1$ 。

$f(n)$ 的条件	结论	检查点
$f(n) = O(n^{p-\epsilon})$ ，某个固定 $\epsilon > 0$	$T(n) = \Theta(n^p)$	必须有多项式量级的差距
$f(n) = \Theta(n^p \lg^k n)$ ，固定 $k \geq 0$	$T(n) = \Theta(n^p \lg^{k+1} n)$	这是课件给出的扩展第二种情况
$f(n) = \Omega(n^{p+\epsilon})$ ，且 $af(n/b) \leq cf(n)$ ，某个 $c < 1$	$T(n) = \Theta(f(n))$	要验证正则条件

不能直接用上述三种情况处理 $f(n) = n^p / \lg n$ ：它与 n^p 只差一个对数因子，不是固定幂次差距，也不符合这里 $k \geq 0$ 的第二种情况。

2.3 六道历年递归式，必须独立做一遍

年份题号	递归式	正确结论与方法
2023 判断 1	$4T(n/4) + n \lg n$	$\Theta(n \lg^2 n)$; $p = 1, k = 1$
2023 简答 1	$T(2n/3) + 2^{\lceil \lg n \rceil}$	$\Theta(n)$; 先用 $n \leq 2^{\lceil \lg n \rceil} < 2n$ ，再求几何和

年份题号	递归式	正确结论与方法
2024 判断 1	$2T(n/5) + \sqrt{n}$	$\Theta(\sqrt{n})$; $\log_5 2 < 1/2$, 根部代价占主导
2024 简答 1	$\frac{1}{2}T(n/3) + \frac{1}{2}T(2n/3) + \lg n$	可证 $O(\lg^2 n)$; 题目要求好的上界与代入证明, 不套标准主定理
2025 判断 1	$2T(2n/3) + n \log n$	$\Theta(n^{\log_{3/2} 2})$; 指数约 1.710, $n \log n$ 比它低固定幂次
2025 简答 1	$2T(n/2) + n/\log_2 n$	$\Theta(n \log \log n)$; 递归树产生调和级数

2025 简答 1 的推导：令 $n = 2^m$, 在规模仍大于常数的第 j 层, 总代价为 $n/(m - j)$ 。因此内部层之和为 nH_m , 叶子再贡献 $\Theta(n)$, 总计 $\Theta(n \log \log n)$ 。分母是底数为 2 的对数, 不是 $(\log n)^2$ 。

2024 简答 1 的代入法框架：设 $L = \lg n$, $u = \lg 3$, $v = \lg(3/2)$ 。尝试上界 cL^2 , 递归右边不超过

$$\frac{c}{2}(L - u)^2 + \frac{c}{2}(L - v)^2 + L = cL^2 - [c(u + v) - 1]L + \frac{c}{2}(u^2 + v^2).$$

取 c 足够大, 余项对足够大的 n 非正, 再选择常数覆盖基本情形, 即完成上界证明。

2.4 代入法最容易丢分的地方

对 $T(n) = 4T(n/2) + n$, 假设 $T(k) \leq ck^2$ 只得到 $cn^2 + n$, **不能写“因此为 $O(n^2)$ ”就宣布归纳闭合**。需要加强假设, 例如 $T(k) \leq c_1k^2 - c_2k$, 用减去的低阶项吸收额外的 n 。这是 T2 p.40–45 的重点。

三、比较排序与二叉堆

课件定位：T1 p.70–85（归并）；T3 p.3–5、7–71（基础排序）、73–85（Shell）、87–134（堆）、136–158（快排）。

3.1 性质对照表

下表针对通常的数组实现；“稳定”要求比较及移动规则正确, 不能只凭算法名称判断任意实现。

算法	最好时间	平均时间	最坏时间	常见辅助空间	稳定性
插入排序	$\Theta(n)$	$\Theta(n^2)$	$\Theta(n^2)$	$O(1)$	稳定, 等键不越过
选择排序（交换版）	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	$O(1)$	不稳定
冒泡排序（有提前终止）	$\Theta(n)$	$\Theta(n^2)$	$\Theta(n^2)$	$O(1)$	稳定, 只交换严格逆序对
归并排序（标准版）	$\Theta(n \lg n)$	$\Theta(n \lg n)$	$\Theta(n \lg n)$	$\Theta(n)$	可稳定实现, 等键先取左边
堆排序	通常写 $O(n \lg n)$	$\Theta(n \lg n)$	$\Theta(n \lg n)$	$O(1)$, 迭代下沉	不稳定

算法	最好时间	平均时间	最坏时间	常见辅助空间	稳定性
快速排序	$\Theta(n \lg n)$	$\Theta(n \lg n)$	$\Theta(n^2)$	栈期望 $O(\lg n)$, 最坏 $O(n)$	常规版不稳定
Shell 排序	依赖增量序列	依赖增量序列	依赖增量序列	$O(1)$	常规版不稳定

堆排序的最好时间与具体实现、是否允许大量相等键有关；考试优先掌握最坏保证与建堆代价。课件给的是最好 $O(n \log n)$ ，不要无条件把它加强成所有输入模型下的 $\Theta(n \log n)$ 。

Shell 排序要会解释“分组做插入排序，逐步减小 gap，最后 gap=1”。T3 p.84 展示不同序列可有不同复杂度，不能把某个序列的界当成整个算法的统一界。

3.2 稳定性与输入顺序的典型判断

- 2023 判断 3：选择排序的比较次数为 $n(n-1)/2$ ，不因初始排列变成线性。
- 2024 判断 3：最后一趟之前，插入排序已保证前 $n-1$ 个数内部有序，但它们未必在最终位置。如 $[2, 3, \dots, n, 1]$ 。
- 2024 判断 4：链表适合顺序合并，归并无需数组式的线性临时缓冲；不能据此声称快排更有优势。
- 2025 判断 3：按标准、稳定合并的归并排序回答。若自行写实现，等键必须优先取左子序列元素，才能保持稳定。

3.3 二叉堆：三种代价不能混淆

数组下标从 1 开始时，父结点为 $\lfloor i/2 \rfloor$ ，左右孩子为 $2i, 2i+1$ 。堆是近乎完全的二叉树，保证父子之间的大小关系，不保证中序有序。

1. **一次 MAX-HEAPIFY**：两棵孩子子树已经是堆，沿一条路径下沉，最坏 $O(\lg n)$ 。
2. **自底向上 BUILD-MAX-HEAP**：从 $\lfloor n/2 \rfloor$ 到 1 下沉，总计 $\Theta(n)$ 。
3. **HEAPSORT**：建堆后反复将最大值移到数组末尾，最坏 $\Theta(n \lg n)$ 。

建堆线性证明按高度求和：高度为 h 的结点至多 $O(n/2^{h+1})$ 个，各花 $O(h)$ ，因此

$$\sum_{h \geq 0} O\left(\frac{nh}{2^{h+1}}\right) = O(n).$$

遍历并处理输入又给出 $\Omega(n)$ 。逐个 INSERT 建堆是另一种方法，最坏 $\Theta(n \lg n)$ ，不与自底向上建堆的结论矛盾。对应 2024 判断 2。

3.4 快排必须默写的伪代码

```
PARTITION(A, p, r)
    pivot = A[r]
    i = p - 1
    for j = p to r - 1
        if A[j] <= pivot
            i = i + 1
            swap(A[i], A[j])
    swap(A[i + 1], A[r])
    return i + 1

RANDOMIZED-QUICKSORT(A, p, r)
    if p < r
        s = RANDOM(p, r)
        swap(A[s], A[r])
        q = PARTITION(A, p, r)
        RANDOMIZED-QUICKSORT(A, p, q - 1)
        RANDOMIZED-QUICKSORT(A, q + 1, r)
```

要能说清循环不变式：已扫描部分分成“小于等于 pivot”与“大于 pivot”两段；pivot 最后才放到正确位置。递归不能把 q 再包含进去。

三种问题要分开：最好运行时间为 $\Theta(n \lg n)$ 、最坏为 $\Theta(n^2)$ ；最好递归深度 $\Theta(\lg n)$ 、最坏 $\Theta(n)$ ；而包含基本情形的递归调用总次数在最好和最坏情况下都是 $\Theta(n)$ 。最后一点就是 2023 简答 2 所问，不是问深度或运行时间。

对互异键、均匀随机 pivot，排序后第 i, j 个元素发生比较的概率为 $2/(j - i + 1)$ ；由指示变量得到期望比较次数

$$\mathbb{E}[C_n] = 2(n+1)H_n - 4n = \Theta(n \lg n).$$

四、线性排序与顺序统计量

课件定位：T4.1 p.3-7（下界）、8-20（计数）、21-31（基数）、32-38（桶）；T4.2 p.3-22（选择）。

4.1 比较排序下界的证明

对 n 个互异元素有 $n!$ 种排列。二叉比较决策树必须有至少 $n!$ 个可达叶子；若高度为 h ，则 $2^h \geq n!$ ，因此最坏比较次数至少 $\lg(n!) = \Omega(n \lg n)$ 。

计数、基数、桶排序利用键值范围或分布等额外信息，超出了“只进行两两比较”的模型，所以不违反此下界。

2023 判断 5 的归约：如果能把二叉堆在线性最坏时间转成 BST，就可以先线性建堆、再线性转换、最后线性中序遍历，从而得到线性比较排序，与下界矛盾。

4.2 稳定计数排序

假设整数键在 $[0, k]$ 。三个阶段：统计频数、累计计数、反向扫描输入放入输出数组。

```

COUNTING-SORT(A, k)                                // A[1..n]
  C[0..k] = 0
  for j = 1 to n
    C[A[j]] = C[A[j]] + 1
  for v = 1 to k
    C[v] = C[v] + C[v - 1]
  create B[1..n]
  for j = n downto 1
    B[C[A[j]]] = A[j]
    C[A[j]] = C[A[j]] - 1
  return B

```

时间 $\Theta(n + k)$ ，标准稳定版本的辅助空间 $\Theta(n + k)$ ，不是 $O(1)$ 。若每个元素是记录，应把整条记录放入 B，按记录的键访问 C。

稳定性解释：后出现的等键元素先占较靠后的输出位置，前出现的元素随后占前面的位置。此代码若直接改成从前向后扫描，会颠倒等键记录的相对顺序。

对应：2025 判断 4、综合 1(a)。

4.3 基数排序

LSD 基数排序按最低位到最高位逐位排序；每一轮必须稳定，才能保留低位形成的顺序。

$$T = \Theta(d(n + k)),$$

其中 d 为位数、每位有 k 种取值。对于 b 位整数，每次处理 r 位：

$$T(n, b, r) = \Theta\left(\left\lceil \frac{b}{r} \right\rceil (n + 2^r)\right).$$

若 $b = O(\lg n)$ ，取适当的 $r = \Theta(\lg n)$ 可令轮数为常数且桶数为 $O(n)$ ，得到线性时间；若 b 更大，轮数不能直接当成常数。

2023 综合 1(a)：对 $[1, n^2]$ 中的整数，可先减 1 变为 $[0, n^2 - 1]$ ，用 n 进制的两位稳定计数排序，每轮 $\Theta(n)$ 。若不减 1，端点 n^2 可能需要第三位，但仍是常数轮数；不要漏掉边界。

对应：2023 综合 1(a)、2025 综合 1(b)(c)。

4.4 桶排序：为什么只是期望线性

独立均匀分布于 $[0, 1)$ 时，分到 n 个等宽桶，桶内插入排序后按桶号连接。若第 i 桶有 N_i 个元素：

$$T = \Theta(n) + O\left(\sum_i N_i^2\right), \quad \mathbb{E}[N_i^2] = 2 - 1/n.$$

所以期望 $\Theta(n)$ ；最坏所有元素进同一桶，采用插入排序时可达 $\Theta(n^2)$ 。均匀分布、独立性及桶的设计是分析条件。

2023 综合 1(b) 的非均匀外观： $Z_i = X_i + Y_i$ ，其中 X_i 在 m 个离散位置均匀取值， $Y_i \in (0, 1/n)$ ，且 $m < n$ 为常数。有效区间是 m 段彼此分离、各长 $1/n$ 的短区间，应该在这些短区间内细分桶，而不是把概率几乎为零的大片间隔也均匀分桶。

一种直接设计是：把每个有效区间再分成 n 份，总共 $mn = \Theta(n)$ 个桶（因为 m 是常数）。对元素 z ：

$$j = \lfloor mz \rfloor, \quad y = z - j/m, \quad \text{bucket}(z) = jn + \lfloor n^2 y \rfloor.$$

因为 $m < n$ ，有 $0 \leq my < m/n < 1$ ，故可准确恢复其离散区间编号 j ；每个有效桶的概率为 $1/(mn) = \Theta(1/n)$ 。按桶编号输出且桶内插入排序，期望 $O(n)$ 。桶用半开区间覆盖所有边界；这里使用题目给出的独立分布和常数时间实数运算模型。

4.5 最值与选择算法

问题或算法	关键结论	要能解释的原因
找最小值或最大值	最坏至少 $n - 1$ 次比较	除胜者外，每个候选必须至少输一次
同时找最小值与最大值	$\lceil 3n/2 \rceil - 2$ 次比较， $n \geq 2$	先两两比较，较小者争最小、较大者争最大
RANDOMIZED-SELECT	期望 $\Theta(n)$ ，最坏 $\Theta(n^2)$	partition 后只递归一边
五个一组的 SELECT / BFPRT	最坏 $\Theta(n)$	中位数的中位数保证丢弃常数比例

QuickSelect 在 $A[p..r]$ 中找第 i 小，令 q 为 pivot 下标， $s = q - p + 1$ ： $i = s$ 返回 pivot； $i < s$ 递归左侧；否则右侧找第 $i - s$ 小。注意“子数组中的相对名次”与“整个数组的目标下标”两种参数约定不能混用。

BFPRT 的核心递归界是

$$T(n) \leq T(\lceil n/5 \rceil) + T(7n/10 + O(1)) + O(n) = O(n).$$

要会讲清：先求各组中位数，再递归找中位数的中位数作 pivot；除去常数个边界组后，左右两边各能保证排除约 $3n/10$ 个元素。

2024 综合 1 的计数口径：对题目给定的 Lomuto PARTITION，互异键、均匀随机 pivot 或均匀随机排列下，一次分区有 $n - 1$ 次“元素与 pivot”的键码比较。于是

$$C_{n,m} = n - 1 + \frac{1}{n} \left(\sum_{s=1}^{m-1} C_{n-s,m-s} + \sum_{s=m+1}^n C_{s-1,m} \right).$$

基本情形 $C_{1,1} = 0$ 。当 $m = 1$ 时，有 $C_{n,1} = 2n - 2H_n = \Theta(n)$ 。原答案的分区代价与最终表达式存在口径不一致，复习时以明确的比较模型为准。

五、BST、红黑树与树的扩充

课件定位：T5.1 p.3-26 (BST)、27-52 (红黑树)、53-65 (扩充与区间树)。

5.1 BST 与遍历

BST 满足左子树键不大于根、右子树键不小于根；若允许重复键，应统一放置和区分规则。搜索、插入、删除、最值、前驱、后继均为 $O(h)$ ，其中 h 为树高；普通 BST 最坏退化成链，操作 $O(n)$ 。

- 中序：左、根、右；BST 中序给出非降序序列，耗时 $\Theta(n)$ 。
- 前序：根、左、右。后序：左、右、根。
- 对结点标识互异的二叉树，中序 + 前序、中序 + 后序均可唯一恢复；前序 + 后序一般不行，单孩子在左或右会产生歧义。
- 后序与中序相同时，每个结点都无右孩子，否则根与右子树的相对顺序冲突。
- 删除有两个孩子的结点时，用后继或前驱替代；右子树最小结点没有左孩子，因此实际删除归结为至多一个孩子的情形。

对应：2024 判断 5、2025 判断 5。

5.2 红黑树的五条性质与高度证明

1. 结点是红或黑；2. 根为黑；3. NIL 叶子为黑；4. 红结点的孩子都为黑；5. 从任一结点到其后代 NIL 的所有路径，黑结点数相同。

按本课件定义： $bh(x)$ 不计 x 自身，计末端 NIL (T5.1 p.31)。证明链为

$$n_x \geq 2^{bh(x)} - 1, \quad bh(\text{root}) \geq h/2, \quad h \leq 2 \lg(n+1).$$

因此红黑树提供最坏 $O(\lg n)$ 的动态集合操作。普通高度与黑高不是同一概念；红黑树不要求左右子树普通高度差至多 1，那是 AVL 的要求。

黑高为 b 、根为黑的红黑树，内部结点数满足

$$2^b - 1 \leq n \leq 2^{2b} - 1.$$

下界来自全黑的最小树，上界可由红黑层交替且完全展开达到。对应 2025 判断 6；写证明前必须说明黑高是否计根与 NIL。

2023 判断 6 的思路：若除根与 NIL 外全部变色后仍合法，那么原树中两个相邻的、非根内部结点也不可能同时为黑，否则变色后出现红红相邻。结合原有“无红红相邻”，可知根以下的内部颜色必须交替。原树与变色后的等黑高条件共同迫使各根到 NIL 路径长度相同，从而得到每个结点的左右子树结点数相同。要同时使用两棵树合法这一条件。

5.3 旋转、插入与删除

旋转只改变常数个指针，保持中序顺序，单次 $O(1)$ 。务必检查：孩子父指针、原父结点指向、旋转后子树根，以及扩充字段。

插入修复：新结点先染红。叔叔红时父与叔染黑、祖父染红，向上继续；叔叔黑时先把折线形变直，再旋转祖父并重新染色；最后根染黑。左右对称。

删除修复：先按 BST 删除，关注实际移走结点的原颜色。移走黑色时，用“额外一层黑”理解修复：红兄弟先转为黑兄弟情形；黑兄弟且两个孩子均黑则重新染色、向上传递；近侄红远侄黑时先旋转兄弟；远侄红时旋转父结点、重新染色后结束。此处按“近侄/远侄”理解，课件的 case 编号与某些教材的顺序不同。

5.4 扩充题的统一答题模板

四步：底层结构 → 附加字段 → 更新维护 → 新查询。 如果字段可由当前结点及左右孩子的信息在 $O(1)$ 时间内重算，就可以沿祖先链维护，并在每次旋转后先更新下层、再更新上层，保持红黑树插入删除的 $O(\lg n)$ 时间。

A. 顺序统计树 OST

$$\text{size}[x] = \text{size}[\text{left}[x]] + \text{size}[\text{right}[x]] + 1, \quad \text{size}[\text{NIL}] = 0.$$

```
OS-SELECT(x, i)
    if x == NIL or i < 1 or i > size[x]
        return NIL
    r = size[left[x]] + 1
    if i == r
        return x
    if i < r
        return OS-SELECT(left[x], i)
    return OS-SELECT(right[x], i - r)
```

OS-RANK 从当前结点向上走；每次从某祖先的右孩子上来，就加上该祖先左子树 size 与该祖先本身。红黑树上的 SELECT、RANK 均为最坏 $O(\lg n)$ 。对应 2025 综合 2(a)(b)：左 size=3、右 size=5，则当前 size=9，第 4 小为根。

B. MIN-GAP

每个子树维护最小键 mn 、最大键 mx 、最小相邻键差 gap 。对非空孩子参与下式：

$$\text{gap}[x] = \min\{\text{gap}[L], \text{gap}[R], \text{key}[x] - \text{mx}[L], \text{mn}[R] - \text{key}[x]\}.$$

同时更新 mn, mx 。不存在的跨边界项应直接跳过；空树与单结点的 gap 可定义为 $+\infty$ ，集合不足两个元素时对外返回“无可差值”。若允许重复数，最小差可以为 0。根的 gap 即答案：查询 $O(1)$ 、更新 $O(\lg n)$ 。对应 2024 简答 3。

C. 动态最大重叠点

把端点按顺序放入红黑树：左端点事件 +1，右端点事件 -1。维护每个子树事件总和 sum 、允许空前缀的最大前缀和 $pref$ 及其位置，NIL 的两值均为 0。

$$sum[x] = sum[L] + v[x] + sum[R],$$

$$pref[x] = \max\{pref[L], sum[L] + v[x], sum[L] + v[x] + pref[R]\}.$$

同步记录最优值由左子树、当前事件还是右子树贡献，即可恢复端点位置。对非空区间集最大值为正，会对应真实事件；空集合单独返回无结果。

闭区间同坐标时，左端点必须排在右端点之前，例如 $[1, 2]$ 与 $[2, 3]$ 在 2 处重叠；可用“坐标、事件类型、唯一 ID”作为键。插入/删除一个区间各涉及两个事件， $O(\lg n)$ ；读取根的最优位置 $O(1)$ 。对应 2025 简答 3。

D. 课件中的区间树

以区间左端点为 BST 键，每个结点保存本子树所有区间的最大右端点：

$$maxHigh[x] = \max\{high[x], maxHigh[L], maxHigh[R]\}.$$

若当前区间不与目标重叠，且左子树非空、 $maxHigh[L] \geq low[query]$ ，优先去左侧，否则去右侧。查询任意一个相交区间耗时 $O(\lg n)$ 。查询任意相交区间与查询最大重叠点需求不同，需要不同的扩充字段。

六、二项堆、斐波那契堆与摊还分析

课件定位：T5.2 p.3-27（二项堆）、28-71（斐波那契堆）。

6.1 二项堆

二项树 B_k 有 2^k 个结点，高度与根的度数均为 k ，深度 i 有 $\binom{k}{i}$ 个结点；两个 B_k 可把较大根接到较小根下，形成一个 B_{k+1} 。

二项堆中每种阶数至多一棵，根链表对应结点数的二进制表示。合并像二进制加法，按阶数归并，并链接同阶树；不是只拼接链表就完成。根链表最多 $\lfloor \lg n \rfloor + 1$ 棵树。按课件标准实现，合并、插入、取最小、抽取最小、减键、删除的最坏时间均为 $O(\lg n)$ ；若额外维护最小指针，取最小可改进为 $O(1)$ ，需说明实现。

6.2 斐波那契堆：结构与复杂度

它是满足最小堆序的树的集合，根与孩子用循环双向链表组织，维护 min 指针；不要求每种度数永远只保留一棵，也不要求树始终平衡。结点维护 parent、child、degree、mark 等字段。

操作	斐波那契堆摊还代价	关键动作
MAKE-HEAP、MINIMUM	$O(1)$	初始化或读 min
INSERT	$O(1)$	新单结点树加入根表，更新 min
UNION	$O(1)$	拼接两个根链表，更新 min
EXTRACT-MIN	$O(\lg n)$	孩子提升为根，再合并同度根

操作	斐波那契堆摊还代价	关键动作
DECREASE-KEY	$O(1)$	必要时切断并级联切断
DELETE	$O(\lg n)$	减键到 $-\infty$ 后抽取最小

以上是**摊还界**。例如大量 INSERT 后，单次 EXTRACT-MIN 可能实际处理线性多个根，不能声称每次实际最坏 $O(\lg n)$ 。

6.3 必练：EXTRACT-MIN 的画图步骤

1. 删除 min 结点；把它的所有孩子加入根链表，parent 置 NIL。
2. 用按 degree 索引的辅助数组进行 CONSOLIDATE。
3. 若已有同度根，比较键，把较大根挂到较小根下面，作为孩子时清 mark，更新父结点的 degree；新度数可能继续碰撞。
4. 重建根链表，重新确定 min。最后同度根至多一个。

检查清单： 结点只少了一个；所有父子满足堆序；degree 等于实际孩子数；根的 parent 为 NIL；合并未漏掉“进位”。如果题目规定根的扫描顺序，按该顺序画；未规定时合并次序可能导致不同的合法树形，说明采用的顺序。

6.4 必练：DECREASE-KEY 与 CASCADING-CUT

- 减键后仍不小于父结点：无需切断，但可能更新 min。
- 若小于父结点：切下该结点，加入根表，parent 置 NIL，清 mark，父结点 degree 减 1。
- 检查父结点：若它不是根且此前未标记，标记并停止；若它不是根且此前已标记，则也切下它，清标记，并继续向上。
- 到根停止；根无需因丢失孩子而标记。mark 表示结点自最近一次成为另一个结点的孩子之后，是否已经丢失过一个孩子，不表示颜色意义上的堆序或大小关系。

三年固定练习链：2023 简答 4 是先抽取再 $25 \rightarrow 3$ ；2024 为 $35 \rightarrow 4$ ；2025 为 $30 \rightarrow 4$ 。第二问必须接着第一问的结果操作，不能回到原图。

6.5 为什么有这些摊还界

势能函数：

$$\Phi(H) = t(H) + 2m(H), \quad \hat{c}_i = c_i + \Phi(H_i) - \Phi(H_{i-1}).$$

$t(H)$ 为根树数， $m(H)$ 为标记数。插入会增加少量势能；抽取最小通过合并消耗根数所积累的势能；级联切断会去掉已标记结点，释放的势能支付多次切断。

最大度数的证明路线：按挂接顺序排列某结点的孩子，第 i 个孩子当前度数至少为 $i - 2$ ($i \geq 2$)，因为它在仍为孩子期间最多丢一个孩子。度数为 k 的子树规模至少 $F_{k+2} \geq \varphi^k$ ，因此

$$D(n) \leq \log_{\varphi} n = O(\log n).$$

这限制的是**度数**，不要把它写成斐波那契堆每棵树的高度都为 $O(\log n)$ 。

七、并查集与课件中的新增应用

7.1 并查集不能遗漏

课件定位：T5.2 p.72–85。 三份试卷未直接单独考查，不影响它属于本次课件范围。

必须掌握 MAKE-SET、FIND-SET、UNION 的语义：集合互不相交，用代表元标识集合；UNION 应合并两个代表元，不能直接把任意非根结点当根链接。

- 链表表示：每个成员保留代表指针，查找 $O(1)$ ，合并时更新一条链的所有代表指针；朴素实现可能累计 $\Theta(n^2)$ 。
- **小并大**：只更新较小集合中成员的指针，每个元素被更新后所在集合规模至少翻倍，所以最多更新 $O(\lg n)$ 次。 m 次操作、其中 n 次 MAKE-SET，总计 $O(m + n \lg n)$ 。
- 森林表示：按秩合并 + 路径压缩，操作序列的总时间 $O(m\alpha(n))$ （通常取 $m \geq n$ ），单次摊还 $O(\alpha(n))$ 。路径压缩后 rank 不等于真实高度；rank 是控制合并的上界信息，压缩时不需要重新计算。

```
FIND(x)
    if parent[x] != x
        parent[x] = FIND(parent[x])
    return parent[x]

UNION(x, y)
    rx = FIND(x); ry = FIND(y)
    if rx == ry
        return
    if rank[rx] < rank[ry]
        parent[rx] = ry
    else
        parent[ry] = rx
        if rank[rx] == rank[ry]
            rank[rx] = rank[rx] + 1
```

7.2 2026 课件中的 AI 与复杂度例子

课件定位：T0 p.6–8；T1 p.52–56。 三年旧题无法覆盖这部分新材料，可作为“会做规模分析”的补充，不能据此断言必考。

对 n 个 token、维度 d ， $Q, K, V \in \mathbb{R}^{n \times d}$ ：

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V.$$

- $QK^\top$ ：时间 $\Theta(n^2d)$ ，分数矩阵大小 $n \times n$ 。
- softmax： $\Theta(n^2)$ ；再乘 V ： $\Theta(n^2d)$ 。
- 固定 d 时，稠密注意力核心时间为 $\Theta(n^2)$ ；显式存储分数、权重矩阵需要 $\Theta(n^2)$ 空间， $Q/K/V$ /输出各需 $\Theta(nd)$ 。

- 投影还带来 $\Theta(nd^2)$ 运算。序列长度翻倍，成对分数数量变成四倍；“更高效的实现”与“改变注意力计算模式”要区分，并说明精度、稀疏性或模型表达方面的取舍。
- T0 的 RAG 例子把检索、Top-k 选择、生成连起来：前面所学的排序与选择算法可用于解释为什么不必对全部候选完整排序。

八、算法设计题：双指针与滑动窗口

这是从三年简答题归纳的练习模块，使用了排序、复杂度与正确性证明的基础，并不是声称课件设有一个独立的“双指针”章节。

8.1 有序数组找相反数对：2023 简答 3

左右指针 $l = 0, r = n - 1$ 。若 $A[l] + A[r] = 0$ 返回存在；小于 0 则 $l++$ ，大于 0 则 $r--$ ，循环条件为 $l < r$ 。每轮排除一个不可能再参与有效解的端点，两指针总移动 $O(n)$ 次，辅助空间 $O(1)$ 。

原答案伪代码中出现了循环边界及变量笔误，按上面的正确边界复习。 $l < r$ 还保证不能用同一个 0 和自身配对。

8.2 木块配对填满容器：2024 简答 2

先升序排序。维护最小与最大未处理木块：若二者之和不超过 V ，配成一对并移动两指针；否则丢弃最大的木块。统计可形成的最大对数是否至少为 m 。

正确性分两步：若最小值加最大值仍超限，最大值无法与任何剩余木块配对；若可配，可用交换论证说明存在一个最优方案也把这两块配在一起。排序 $O(n \lg n)$ ，扫描 $O(n)$ 。需要的是选出至少 m 对，允许剩余木块不使用，不是要求所有木块必须装入。

8.3 至少 k 种颜色的最短连续区间：2025 简答 2

维护窗口各颜色出现次数、不同颜色数 *distinct*。右端点每加入一种原来计数为 0 的颜色，就增加 *distinct*；窗口合法时不断缩小左边界并更新最短长度。

```
l = 0; distinct = 0; best = +infinity
count[1..n] = 0
for r = 0 to n - 1
    if count[A[r]] == 0
        distinct = distinct + 1
    count[A[r]] = count[A[r]] + 1
    while distinct >= k
        best = min(best, r - l + 1)
        count[A[l]] = count[A[l]] - 1
        if count[A[l]] == 0
            distinct = distinct - 1
        l = l + 1
return best
```

两指针各最多走 n 次，即使有嵌套循环，总时间仍为 $O(n)$ ；颜色在 $[1, n]$ 时直接用计数数组，空间 $O(n)$ 。关键单调性：向右扩张不会减少颜色种数，向左缩小不会增加颜色种数。

九、综合题补充：AVL、Treap 与随机 BST

T5.1 系统讲解的是 BST、红黑树和扩充；AVL、Treap 在本次材料中没有独立完整章节。这里为覆盖真题而补充，练习时应先读题目给出的定义。

9.1 AVL：2023 综合 2

维护 $BF[x] = h(left[x]) - h(right[x])$ ，合法值为 $-1, 0, 1$ 。已知左右孩子各自的 BF，不能唯一推出父结点 BF：孩子的 BF 只描述它们各自内部差值，未给出其绝对高度。题设两个孩子 BF 分别为 0 与 1 时，在整体仍为 AVL 的前提下，父结点 BF 的可能值为 $-1, 0, 1$ 。

以空树高度 -1、叶子高度 0 为例，最少结点数满足

$$N(-1) = 0, \quad N(0) = 1, \quad N(h) = 1 + N(h-1) + N(h-2).$$

因此 $N(h) = F_{h+3} - 1 = \Omega(\varphi^h)$ ，从而 $h = O(\log n)$ 。若采用叶子高度 1，初值及 Fibonacci 下标相应平移，结论不变。

旋转规则：LL 右旋、RR 左旋、LR 先左后右、RL 先右后左。写通用平衡函数时，左重且左孩子 $BF \geq 0$ 可直接右旋，右重且右孩子 $BF \leq 0$ 可直接左旋；使用“等于”可兼容删除后出现孩子 $BF=0$ 的情形。旋转后先更新下层结点高度，再更新新根高度。

9.2 Treap：2024 综合 2

key 满足 BST，priority 满足最小堆。若 key 与 priority 各自互异，给定结点集合只对应**唯一一棵 Treap**：priority 最小者必须为根，再按 key 分成左右集合递归确定。

插入先按 key 插成 BST 叶子，再在父 priority 大于新结点 priority 时向上旋转：新结点是左孩子就右旋父结点，是右孩子就左旋父结点。既保持中序顺序，也修复堆序。

当 priority 独立连续且无并列，按 key 排列的内部结点成为叶子，当且仅当它的 priority 大于两个相邻 key 的 priority，概率为 $1/3$ ；两端结点叶子概率为 $1/2$ 。由线性性：

$$\mathbb{E}[L] = \frac{n-2}{3} + \frac{1}{2} + \frac{1}{2} = \frac{n+1}{3} \quad (n \geq 2).$$

$n = 1$ 时单独为 1。随机优先级给出的是期望平衡，不是每次操作最坏都保证 $O(\log n)$ 。

9.3 随机 BST 的祖先概率：2025 综合 2(c)

假设将互异键 $1, \dots, n$ 按均匀随机排列插入普通 BST，根深度为 0。对 $j < i$ ，键 j 成为 i 的祖先，当且仅当 j 是 $[j, i]$ 中第一个被插入的键，概率为 $1/(i-j+1)$ ； $j > i$ 对称。

$$\mathbb{E}[D_i] = \sum_{j < i} \frac{1}{i-j+1} + \sum_{j > i} \frac{1}{j-i+1} = H_i + H_{n-i+1} - 2.$$

OS-SELECT 每层耗时常数，因此在这样的随机 BST 上期望时间 $\Theta(\log n)$ ，最坏仍可能 $O(n)$ 。若底层始终是红黑树，则是最坏 $O(\log n)$ 保证；**不能把“随机插入普通 BST 的证明”直接说成红黑树旋转后的深度精确公式。**

十、附加题拓展与答题策略

10.1 2023：连续刷题方案 → 区间和计数

把每题净收益写成 $b_i = s_i - \alpha$ ，前缀和为 P_j 。目标转为统计

$$l - S \leq P_j - P_i \leq r - S, \quad 0 \leq i < j \leq n.$$

虽然 $s_i \geq 0$ ，净收益 b_i 可能为负，因此不能直接用仅适用于非负数组的单调滑动窗口。可用归并分治，在归并时用两个单调指针统计跨半区的合法前缀差，得到最坏 $O(n \log n)$ 。

10.2 2024：股票价值区间 → 下标窗口 + 值域桶

扫描 j 时只保留前面下标落在 $[j - W, j - 1]$ 的价格。查询同桶与相邻桶是否有价差 $\leq A$ 的元素，再插入当前价格。过期元素及时删除，查询在插入前进行以避免与自身配对。

整数价格与整数阈值可用桶宽 $A + 1$ ；题目只给正数而未明确整数时，不应无条件照搬。对实数且 $A > 0$ 可用宽度 A ，并检查邻桶的实际差值； $A = 0$ 单独查相等值。使用散列表时，整体 $O(n)$ 是标准哈希假设下的**期望时间**，不是无需条件的最坏线性保证。

10.3 2025：受限交换 → 按 k 分类

关键先看清：可交换的是**数值差**受限的两块，它们的位置可以不相邻。

- $k = 0$ ：只有原排列已递增时可完成，否则不可达。
- $k = 1$ ：相邻数值交换的最少次数等于逆序数，可在逆排列中理解为相邻位置交换；归并计数 $O(n \log n)$ 。
- $k = 2$ ：数值差 2 的免费交换把奇数值与偶数值各自连成一组。令 c 为“所在位置奇偶性与数值奇偶性不同”的元素个数，最少代价为 $c/2$ ，扫描 $O(n)$ 。
- $k \geq 3$ 且 $n \geq 4$ ：差 2 或 3 的免费交换图连通，免费边上的交换可生成任意排列，因此答案为 0。较小的 n 应单独讨论，不能直接套此结论。

10.4 各题型的作答骨架

判断题： 真/假 → 精确正确结论 → 一句话证明或最小反例。带有两个分句的命题，只要一处错整个命题就是假，但解释应指出错在哪里。

递归题： 确定方法 → 写关键比较/层代价 → 求和或归纳 → 基本情形 → 给出所需的 O 或 Θ 。

算法设计题： 输入与参数约定 → 核心思路 → 伪代码 → 不变式/交换论证 → 时间与空间 → 重复值、边界、无解条件。

数据结构题： 底层结构 → 字段含义与 NIL 初值 → 合并维护式 → 插入/删除/旋转维护 → 查询如何读取 → 各操作复杂度。只说“扩充红黑树”而不写维护式是不完整的。

画图题： 分步骤画或记录根表 → 更新父指针、degree、mark、min → 说明扫描/合并次序 → 检查结点数与堆序。

十一、复习安排与自测清单

11.1 七轮复习顺序

轮次	任务	结束时要交付给自己的成果
1	渐近记号、求和、六个递归式	一页公式表；不看答案完成六题并写证明
2	比较排序、二叉堆、随机快排	默写性质表、PARTITION、线性建堆证明
3	计数/基数/桶、选择算法	默写稳定计数排序；完成三年综合 1
4	BST、红黑树、扩充	写出 size、MIN-GAP、最大前缀和、maxHigh 四套字段
5	二项堆、斐波那契堆、并查集	完成三年简答 4；讲清势能与小并大
6	双指针、窗口、AVL/Treap/随机 BST	完成三道设计简答与三年综合 2
7	按试卷原顺序整套演练，回看错题	错因分为定义、边界、伪代码、复杂度、证明五类

时间紧时，优先完成 A 类模块与三年简答 4，再补综合题和 B 类课件内容；附加题放在最后。计时练习按教师公布的考试时长进行，本提纲不推测考试时长。

11.2 考前口述自测

1. O 、 Ω 、 Θ 与最好/平均/最坏情况有何区别？
2. 为什么 $n/\lg n$ 不能直接套课件的主定理第二种情况？
3. 为什么自底向上建堆是线性，而不是 $n \log n$ ？
4. 快排的运行时间、递归深度、调用总次数为何不同？
5. 计数排序为什么反向扫描？基数排序为什么需要稳定性？
6. 桶排序的线性结论依赖什么？如何为混合分布设计有效桶？
7. Randomized-Select 与 BFPRT 的线性时间分别是什么保证？
8. 黑高是否计根与 NIL？红黑树为何不满足 AVL 的高度差条件？
9. 扩充字段为何能在旋转后常数时间重算？应先更新谁？
10. MIN-GAP、最大重叠点、区间相交查询分别维护什么？

- 11. 斐波那契堆何时标记、何时切断、何时停止？为什么减键只是摊还常数？
- 12. 并查集的 rank 在路径压缩后是否等于高度？小并大为何每个元素只更新对数次？
- 13. 滑动窗口嵌套 while 为何仍线性？什么时候因为负数而失去单调性？
- 14. 随机 BST 的祖先概率如何写？期望保证能否替代最坏保证？
- 15. 稠密 self-attention 中 n 翻倍时，哪部分代价增为四倍？

能在不看笔记的情况下解释上述问题，再进行整卷练习，比只背复杂度表更能检验掌握情况。

十二、真题定位与来源索引

12.1 三年全部大题速查

题号	2023	2024	2025
判断 1	递归式 $4T(n/4) + n \lg n$	递归式 $2T(n/5) + \sqrt{n}$	递归式 $2T(2n/3) + n \log n$
判断 2	堆排与快排最坏界	堆排序与建堆	因子个数求和
判断 3	选择排序与初始排列	插入排序最后一趟	归并稳定性与复杂度
判断 4	随机选择期望线性	链表上的快排与归并	计数排序空间
判断 5	二叉堆转 BST 与排序下界	后序与中序相同	两种遍历确定第三种
判断 6	红黑树颜色反转	红黑树普通高度差	红黑树结点数上界
简答 1	取整指数递归	加权递归与代入法	$n / \log_2 n$ 递归
简答 2	随机快排伪代码、调用次数	木块配对	至少 k 色最短区间
简答 3	相反数对双指针	扩充 MIN-GAP	动态最大重叠点
简答 4	斐波那契堆	斐波那契堆	斐波那契堆
综合 1	基数排序与分布桶排序	QuickSelect 与期望比较	计数排序、分组基数排序
综合 2	AVL	Treap	OST 与随机深度
附加 1	连续区间得分计数	股票窗口与值差	受限数值交换

三年试卷的实际页码一致：判断 1–3 在 p.1，判断 4–6 与简答 1 在 p.2，简答 2–3 在 p.3，简答 4 在 p.4，综合 1 在 p.5，综合 2 在 p.6，附加题在 p.7。2024 年原答案的简答题前 3 题顺序与试卷不同，必须按内容配对，不能按答案页的题号机械对应。

12.2 课件索引

简写	用户提供的原始文件名	PDF 页数	本提纲采用的内容
T0	Algorithm_2026_Topic_0.pdf	15	课程结构、AI 与算法应用
T1	Algorithm_2026_Topic_1.pdf	85	分析模型、期望、插入与归并、注意力复杂度
T2	Algorithm_2026_Topic_2.pdf	73	渐近记号、常用函数、递归求解
T3	Algorithm_2026_Topic_3.pdf	158	比较排序、Shell、二叉堆、随机快排
T4.1	Algorithm_2026__Topic_4_1.pdf	38	比较下界、计数/基数/桶排序
T4.2	Algorithm_2026__Topic_4_2.pdf	22	最值、随机选择、最坏线性选择
T5.1	Algorithm_2026__Topic_5_1.pdf	65	BST、红黑树、OST、区间树与扩充
T5.2	Algorithm_2026__Topic_5_2.pdf	85	二项堆、斐波那契堆、并查集

真题来源为用户先前提供的《2023–2025 期中考试试题及答案》压缩包中的三年试卷与参考答案。本文把课件定义、真题题干和参考答案分开核对；对答案顺序、伪代码或计数模型的问题，已在对应知识点注明。本文中的归纳、证明补充与纠错为整理内容，不等同于教师发布的勘误。

依据用户提供的 2026 课程课件与 2023–2025 年试卷整理。页码均为 PDF 实际页码。公式为文本排版，可复制、缩放与检索。