根号数据结构与根号分治

isaunoya

Hangzhou Dianzi University

目录

序列与值域分块

莫队与二次离线

参数与轻重分类

按时间分块

拓展与练习

讲授安排

主讲依次展开四部分；额外工具与进阶题放在后面的拓展中。

主讲部分	需要掌握的区别	前置知识
序列与值域分块	完整块与边界	前缀和、频次统计
莫队与二次离线	区间移动与事件求值	排序、前缀和；接前一部分
参数与轻重分类	单个代价与对象数量	基础 DP；树上距离与 LCA
按时间分块	旧批次与新修改	BFS、树上距离与 LCA

P4137 为课堂练习；后续专题见 拓展目录。

目录

序列与值域分块
莫队与二次离线
参数与轻重分类
按时间分块
拓展与练习

区间加与区间和

区间加与区间和模型。 维护长度为 n 的数组，支持给区间 $[l, r]$ 的每个数加 v，以及查询区间元素之和。除单独说明外，区间采用闭区间。

区间加与区间和

区间加与区间和模型。 维护长度为 n 的数组，支持给区间 $[l, r]$ 的每个数加 v，以及查询区间元素之和。除单独说明外，区间采用闭区间。

将数组切成长度不超过 B 的块。一次区间操作最多遇到两个不完整块，其余都是完整块。

左边界

l...

逐个处理

中间完整块

.....

每块读取一份汇总

右边界

...r

逐个处理

边界最多处理 $2B$ 个位置，中间最多访问 $\lceil \frac{n}{B} \rceil$ 个块。**要设计的是一份能代表整块的状态。**

序列分块：底值、标记与块和

块 b 保存元素底值 a_i、统一增量 t_b 与真实块和 S_b，始终满足

$$\text{真实值}_i = a_i + t_b, \quad S_b = \sum_{i \in b} a_i + |b| t_b.$$

整块加 v：令 $t_b \leftarrow t_b + v$，$S_b \leftarrow S_b + |b|v$。边界位置加 v：只改底值 a_i，并给 S_b 加 v，无需下传标记。

查询完整块读取 S_b，查询散点读取 $a_i + t_b$。两种操作均为 $O(B + \frac{n}{B})$，取 $B \approx \sqrt{n}$。同块区间只扫描一次。

块长与复杂度

假设某个算法需要选择参数 B，总时间为

$$T(B) = O\left(\alpha B + \frac{\beta}{B}\right).$$

增大 B 会让第一项变大、第二项变小。两项同阶时，得到 $B \approx \sqrt{\frac{\beta}{\alpha}}$。实际实现还要把 B 限制在合法范围，并考虑两部分的常数与内存。

这个式子比“块长取根号”更有用。比如修改、查询次数不同，或者其中一边带一个 $\log n$，最合适的块长就会改变。

序列分块中的 $B + \frac{n}{B}$ 是其中一种情形。后面的值域、参数、时间和二进制位拆分，也会分别得到需要平衡的两项代价。

动态 mex

动态 mex 模型。 维护一个可重复元素的集合，支持插入、删除，以及查询最小未出现的非负整数。

动态 mex

动态 mex 模型。 维护一个可重复元素的集合，支持插入、删除，以及查询最小未出现的非负整数。

设可关注的值域为 $[0, U]$，维护每个值的出现次数，并按值域分块，记录每块有多少个值出现过。

插入和删除只在计数跨过 0 时改变所属块的汇总，时间为 $O(1)$。查询时跳过已经填满的块，在第一个未满块内逐个找缺失值，时间为 $O\left(\frac{U}{B} + B\right)$。

动态 mex：把常数时间留给修改

如果修改有 q_1 次、查询有 q_2 次，总时间是

$$O\left(q_1 + q_2\sqrt{U}\right).$$

这与树状数组的 $O((q_1+q_2)\log U)$ 各有适用场景。修改远多于查询时，常数时间修改可能更重要。值域上界也不必照搬输入最大值：若集合最多有 M 个元素，则 mex 不超过 M，更大的值可以忽略。

P4168 蒲公英

P4168 蒲公英：静态数组的在线区间众数，同频取最小原值。$n \le 40000$、$q \le 50000$、$a_i \le 10^9$；询问依赖上一次答案，不能离线重排。

P4168 蒲公英

P4168 蒲公英：静态数组的在线区间众数，同频取最小原值。$n \le 40000$、$q \le 50000$、$a_i \le 10^9$；询问依赖上一次答案，不能离线重排。

按位置分成 $K = \lceil \frac{n}{B} \rceil$ 个块，并将值按大小离散化为 C 种。预处理任意连续完整块 $[i, j]$ 的众数 $f_{i,j}$：固定起始块，向右扫描并维护频次，在每个块尾记录最优值。

查询只检查**中间完整块的众数，以及两端散块出现的值**。候选至多 $O(B)$ 个；困难从“搜索整个值域”变成“给少量候选精确计数”。

P4168 蒲公英：候选集合的完整性

左边界

枚举所有值

可能增加某个值的频次

中间完整块

只保留众数 y

同频保留最小值

右边界

枚举所有值

可能增加某个值的频次

若 x 不在边界，它的次数没有增加；中间众数 y 原本就不差于 x，加入边界后只会更好。因此没有遗漏能获胜的候选。

若没有完整块，整个区间不足 $2B$，直接扫描即可。**众数不能相减，候选的频次可以。**

P4168 蒲公英：块前缀频次

令 $P_{t,x}$ 为前 t 个块中值 x 的出现次数，$P_{0,x}=0$。扫描两端散块得到临时频次 e_x；中间完整块为 $[s,t]$ 时，候选的真实频次为

$$c_x = e_x + P_{t,x} - P_{s-1,x}.$$

每个候选 $O(1)$ 计数，按“频次更大、原值更小”比较，总查询 $O(B)$。即使候选来自中间众数，也必须加上它在边界的次数。

同块查询只扫描一次；没有完整块时前缀差为零。记录本次碰到的值，只清空这些临时计数，避免一次查询清空整个值域。

P4168 蒲公英：复杂度与在线解密

前缀频次表耗时 $O(n+KC)$，完整块众数表耗时 $O(nK)$。连同离散化与全部查询：

$$T=O\left(n\log n+\frac{n^2}{B}+qB\right),\quad S=O(n+KC+K^2).$$

平衡取 $B\approx\frac{n}{\sqrt{q}}$，截到合法范围。内存不足时可用出现位置表替代频次表，空间降为 $O(n+K^2)$，每次查询变为 $O(B\log n)$。

解密下一次询问时，令 x 为上次输出的**原值**，初始零：

$$l=((l_0+x-1)\bmod n)+1,\quad r=((r_0+x-1)\bmod n)+1.$$

若 $l>r$ 则交换。排名只用来访问表；解密必须用原值，并避免加法溢出。

目录

序列与值域分块
莫队与二次离线
参数与轻重分类
按时间分块
拓展与练习

区间摘要与移动中的状态

序列分块预先保存每块摘要，询问把整块与边界的贡献合并。区间和可以这样计算；众数却不能只由各块众数确定，因而需要额外的候选与计数信息。

如果数组静态、询问次序可以改变，也可以只保留一个区间的完整状态。按合适的次序回答询问，让相邻区间只差少量端点，再用增删操作复用状态。

前者需要设计可供查询使用的块摘要；后者需要设计正确且便宜的端点增量。依赖上次答案的在线询问不能直接重排，静态数据本身并不等于允许离线。

静态区间统计

静态区间统计模型。 给定一个不再修改的数组，以及可以预先读入的若干区间询问。例如，求每个区间中不同值的个数。

静态区间统计

静态区间统计模型。 给定一个不再修改的数组，以及可以预先读入的若干区间询问。例如，求每个区间中不同值的个数。

维护一个当前区间 $[L, R]$，若能快速加入或删除端点，就可以把它逐步移动成下一个询问。

莫队不要求两个区间的答案可以合并；它要求**一个区间的状态可以增删一个元素**。不同询问通过移动复用同一份状态，最后按原编号输出。

例如统计不同值数：加入值 x，若计数从零变一则答案加一；删除时从一变零则答案减一。区间和、频次统计也有类似增量。

莫队：四种端点移动

从空区间 $[1,0]$ 开始，将当前状态移动到目标 $[l,r]$：

需要移动的方向	先后顺序
$L > l$	先令 $L \leftarrow L-1$，再加入左端点
$R < r$	先令 $R \leftarrow R+1$，再加入右端点
$L < l$	先删除左端点，再令 $L \leftarrow L+1$
$R > r$	先删除右端点，再令 $R \leftarrow R-1$

先扩张、后收缩，便不会删除尚未加入的位置。频次类统计的左右接口相同；逆序对等依赖顺序的统计，左加与右加的贡献不同，必须分别实现。

莫队：排序方式与移动次数

令左端点按长度 B 分块。先按左端点块编号排序，同一块内按右端点递增；也可交替使用递增、递减的蛇形顺序减少折返。

同一左块内，相邻左端点距离不超过 B，共 $O(qB)$；跨块的左端点移动总共 $O(n)$。每个左块内右端点单调移动至多 n 步，换块时折返也至多 n 步，共 $O\left(\frac{n^2}{B}\right)$。

总移动次数为 $M = O\left(\frac{n^2}{B} + qB\right)$。若一次增删耗时 T，读取答案耗时 A，总时间是

$$O(q \log q + MT + qA).$$

取 $B \approx \frac{n}{\sqrt{q}}$ 并截到 $[1, n]$。块长只控制移动次数，**不会自动降低一次增删的代价**。

P4137：区间 mex

Rmq Problem / mex 查询静态数组各个区间的 mex。

P4137：区间 mex

Rmq Problem / mex 查询静态数组各个区间的 mex。

把询问按左端点块、再按右端点排序，维护当前区间的值频次。左右指针每移动一步，只对一个值执行插入或删除；采用前面动态 mex 的值域分块，修改只在频次跨过零时更新汇总，耗时 $O(1)$。

对于长度不超过 n 的区间，mex 不超过 n。只维护 0 到 n，较大的值不用进入频次数组。每回答一个询问，按值域块查找第一个缺失值。

P4137：位置块长与值域块长各有职责

若位置块长为 B，指针移动为 $O\left(\frac{n^2}{B}+q\times B\right)$，答案查询为 $O(q\sqrt{n})$，另加 $O(q\log q)$ 排序。位置块长与值域块长是两个不同参数，不能因为都叫分块就混用同一个含义。

例如当前区间的多重集为 $[0,1,1,3]$，mex 为二。删除一个一后仍有一个一，块内“已出现值数”不变；只有删除最后一个一时才需要减少这个汇总。

P4887：区间异或数对

P4887 第十四分块：查询区间内 $i < j$ 且 $\operatorname{popcount}(a_i \oplus a_j) = k$ 的数对数。$n, q \le 10^5$，$0 \le a_i, k < 2^{14}$。

P4887：区间异或数对

P4887 第十四分块：查询区间内 $i < j$ 且 $\operatorname{popcount}(a_i \oplus a_j) = k$ 的数对数。$n, q \le 10^5$，$0 \le a_i, k < 2^{14}$。

若 $k > 14$，答案全为零。记 $w(x, y) = [\operatorname{popcount}(x \oplus y) = k]$。加入值 x 时，要查询当前区间内与它满足 $w = 1$ 的元素数。

枚举所有含 k 个一的 14 位掩码，需要 $D = \binom{14}{k}$ 次访问；普通莫队耗时 $O(MD)$。下面将掩码枚举集中到顺序扫描中，让莫队移动只产生查询事件。

P4887：端点增量的前缀分解

定义 $F(t,x)=\sum_{i=1}^{t} w(a_i,x)$。当前区间中与值 x 配对的元素数就是两个前缀之差：

$$\sum_{i=L}^{R} w(a_i,x)=F(R,x)-F(L-1,x).$$

例如向右加入位置 $p=R+1$，它只能与原区间 $[L,p-1]$ 配对，因此新增贡献为

$$F\big(p-1,a_p\big)-F\big(L-1,a_p\big).$$

第一项只依赖原数组和位置 p；第二项还依赖移动时的左端点。将这两部分分开，就能把昂贵的增量查询集中处理。

P4887：静态项与待求前缀

预处理每个位置的静态贡献：

$$c_p = F(p-1, a_p), \quad d_p = F(p, a_p) = c_p + [k = 0].$$

向右加入位置 $p = R + 1$，新增贡献为 $c_p - F(L-1, a_p)$。向左加入位置 $p = L - 1$，新增贡献为 $F(R, a_p) - d_p$。

前一式排除左边界以前的元素，后一式排除新位置及它左边的元素。**即使 $k = 0$，也不能把位置 p 与自己配对。**

先按莫队顺序记录每次答案差的静态项和前缀查询事件，再按前缀位置 t 求出事件值，最后按莫队顺序累计答案差，并按原编号输出。再次重排的是这些增量事件，原有的莫队顺序保持不变。

P4887：事件登记、求值与答案还原

第 j 个询问的所有移动段，共享答案差 Δ_j。

莫队次序 第 j 个询问	$\rightarrow$	静态项加入 Δ_j 留下 $(t,[s,e],\sigma,j)$
前缀次序 扫描到 t	$\rightarrow$	$T[x]=F(t,x)$ $\sigma\sum_{p=s}^{e}T[a_p]$ 加回 Δ_j
莫队次序 再按 j 递增	$\rightarrow$	$A_0=0$ $A_j=A_{j-1}+\Delta_j$

$[s,e]$ 是移动段，σ 是符号，j 是答案差的归属。

P4887：扩张区间的事件

设 C, D' 分别为 c, d 的前缀和，记

$$E(t,s,e)=\sum_{p=s}^{e} F\big(t,a_p\big).$$

右端点从 R 扩到 $R'>R$，左端点 L 不变：

$$\Delta = C(R')-C(R)-E(L-1,R+1,R').$$

左端点从 L 扩到 $L'<L$，右端点 R 不变：

$$\Delta = E(R,L',L-1)-(D'(L-1)-D'(L'-1)).$$

前缀和部分直接计算，另一部分记录为带符号的事件。一次连续移动只存一个 (t,s,e)，不必为每个位置存一条记录。

P4887：收缩区间的事件

右端点从 R 缩到 $R' < R$，左端点仍为 L：

$$\Delta = -(C(R) - C(R')) + E(L-1, R'+1, R).$$

左端点从 L 缩到 $L' > L$，右端点仍为 R：

$$\Delta = -E(R, L, L'-1) + D'(L'-1) - D'(L-1).$$

按普通莫队的“先扩张、后收缩”顺序生成事件，每段移动使用当时的另一个端点。每个询问至多四个事件，并记下它们属于哪一次答案差。

按位置逐个计入增量，c_p, d_p 中的前缀边界保证本次移动段内部的数对也恰好统计一次。

P4887：按前缀位置再次离线

按 t 将事件分桶，重新从左到右扫描原数组。维护值域表 $T[x] = F(t, x)$：插入 $y = a_t$ 时，枚举所有含 k 个一的掩码 z，令 $T[y \oplus z]$ 加一。

扫描到 t 后，处理这一桶的每个事件：遍历 $p = s, ..., e$，累加 $T\left[a_p\right]$，按记录的符号加入相应的答案差。$t = 0$ 时表全为零。

另一次同样的扫描可以取得 c_p（插入前读表）与 d_p（插入后读表）。最后按莫队顺序对答案差做前缀和，就还原了每个询问的答案。

P4887：两次离线的复杂度

两次扫描总共 $O(nD)$ 次表更新。事件遍历的区间长度之和，恰好不超过莫队的总移动量 M；每个位置只需一次 $O(1)$ 读表。

$$T = O\left(nD + \frac{n^2}{B} + qB + q\log q\right), \quad S = O(n + q + 2^{14}).$$

空间中只存每次连续移动的事件，不能将它拆成 M 条记录。答案与答案差都使用 64 位有符号整数。

第一次离线让端点少移动；第二次离线让所有移动共享一份前缀扫描。适用的关键是**增量能拆成静态项与可统一求值的事件**，并非所有莫队都能这样加速。

P4887：事件的归属与答案还原

将莫队顺序中的询问编号为 $j=1,\dots,q$。每次从上一个区间移动到第 j 个区间，静态项加入 Δ_j；前缀项存成 (t,s,e,σ,j)，其中 $\sigma\in\{-1,1\}$。空移动不产生事件。

第二遍扫描到前缀 t 时，先完成位置 t 的插入，再计算该桶事件；将 $\sigma\sum_{p=s}^{e}T[a_p]$ 加入 Δ_j。零号桶在任何插入前处理。

所有事件求值后，才按莫队顺序恢复

$$A_0=0,\quad A_j=A_{j-1}+\Delta_j.$$

起始状态是空区间 $[1,0]$。同一询问的至多四段移动共享编号 j，每段记录的是移动当时的另一端点；最后将 A_j 写回原询问编号。

目录

序列与值域分块
莫队与二次离线
参数与轻重分类
按时间分块
拓展与练习

CF797E：Array Queries

CF797E Array Queries 给出正整数数组 a，每次询问从 p 开始，反复执行 $p \leftarrow p + a_p + k$，直到超过 n，求跳跃次数。$n, q \leq 10^5$，$1 \leq a_i, p, k \leq n$。

CF797E：Array Queries

CF797E Array Queries 给出正整数数组 a，每次询问从 p 开始，反复执行 $p \leftarrow p + a_p + k$，直到超过 n，求跳跃次数。$n, q \le 10^5$，$1 \le a_i, p, k \le n$。

取阈值 B。若 $k > B$，每次至少前进 $B+1$，直接模拟不超过 $O(\frac{n}{B})$ 步。若 $k \le B$，这样的参数只有 B 种，可以分别从后往前预处理：

$$d_k(p) = 1 + d_k\big(p + a_p + k\big),$$

CF797E：块长由询问次数决定

越界状态的值为 0。总时间与空间分别为

$$O\left(n \times B + q \times \frac{n}{B}\right), \quad O(n \times B).$$

平衡得到 $B \approx \sqrt{q}$，在 $q = \Theta(n)$ 时才是熟悉的 $\sqrt{n}$。如果离线按 k 分组处理询问，可以每次只保留一行 DP，把额外 DP 空间降到 $O(n)$。

轻重分类：总规模与阈值

设对象大小为 $s_1, s_2, \ldots$，总和为 S，按阈值 B 分类。

小对象：限制单个代价

$$s_i \le B$$

可以在对象内部枚举点对

大对象：限制对象数量

$$\text{数量} < \frac{S}{B}$$

可以为每个对象扫描全局

小对象的平方代价满足 $\sum_{s_i \le B} s_i^2 \le B \sum_i s_i = SB$；大对象至多 $\frac{S}{B}$ 个。**先对一整类对象求和，再决定能负担什么算法。**

ABC359G：同色点对距离和

ABC359G Sum of Tree Distance 要求树上所有同色点对的距离之和。$2 \leq n \leq 2 \times 10^5$，颜色编号在 $[1, n]$ 内。

ABC359G：同色点对距离和

ABC359G Sum of Tree Distance 要求树上所有同色点对的距离之和。$2 \le n \le 2 \times 10^5$，颜色编号在 $[1, n]$ 内。

设颜色 c 出现 s_c 次。

对于 $s_c \le B$ 的颜色，枚举同色点对，用 LCA 求距离。若单次 LCA 为 $O(1)$，这部分总时间是 $O(n \times B)$；若使用倍增 LCA，则应写成 $O(n \times B \log n)$。

对于 $s_c > B$ 的颜色，遍历整棵树。设某条父子边下方有 t 个颜色为 c 的点，这条边的贡献就是

$$t(s_c - t).$$

ABC359G：每条边贡献两侧计数之积

每个同色点对恰好在其路径上的各条边被统计一次。大颜色至多 $\frac{n}{B}$ 个，时间为 $O\left(\frac{n^2}{B}\right)$。在常数时间 LCA 的前提下，总时间是 $O\left(n \times B + \frac{n^2}{B}\right)$。

这道题也可以建虚树。虚树边代表原树中的一段路径，所以贡献还要乘边长；子树计数只统计目标颜色的点。两种做法分别利用了“只保留有关的点”和“少数大颜色值得整遍扫描”。

目录

序列与值域分块
莫队与二次离线
参数与轻重分类
按时间分块
拓展与练习

按时间分块：固定大部分，只处理少量变化

把操作分成长度为 B 的批次。一个批次内至多有 B 个修改位置，于是可以把数据分成稳定部分与变化部分：稳定部分建静态结构，变化部分逐次枚举。

若要提前剔除本批会修改的位置，就需要预读整批操作。另一种方式是在批首保留完整快照，在线记录并补上后续修改；当修改贡献可以单独计算时，无须预知未来操作。

CF342E：Xenia and Tree

CF342E Xenia and Tree：$n \le 10^5$ 的树，初始只有顶点一为红色。至多 $q \le 10^5$ 次操作：将蓝点染红，或询问某点到最近红点的距离。

CF342E：Xenia and Tree

CF342E Xenia and Tree：$n \le 10^5$ 的树，初始只有顶点一为红色。至多 $q \le 10^5$ 次操作：将蓝点染红，或询问某点到最近红点的距离。

所有红点固定时怎么办？ 把红点同时入队，做一次多源 BFS，便能在 $O(n)$ 时间求出所有答案。

将操作按时间切成每批至多 B 次。批首所有红点称为旧红点，预处理到它们的距离 $d[v]$；本批新染红的点放入列表 S。

CF342E：查询只补最近一批的贡献

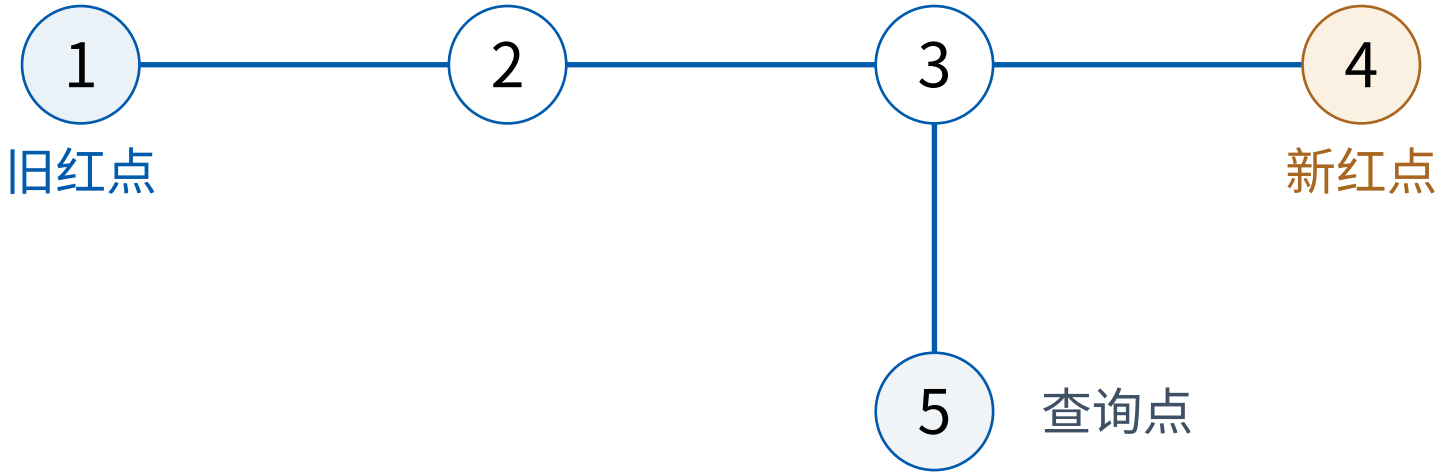
查询 v 时返回

$$\min\left(d[v], \min_{u\in S} \operatorname{dist}(u, v)\right).$$

所有红点恰分为旧红点与新红点，分别取最小值再合并，即为全局最近距离。若 S 为空，只返回 $d[v]$。

批末将 S 并入旧红点，下一批重新多源 BFS。旧红点只会增加；重复染色若出现，只需忽略，避免重复加入列表。初始红点一使答案始终有限。

CF342E：一次查询比较旧答案与新红点



批首只有一为红点，BFS 已得 $d[5]=3$。本批将四染红后，查询五只需再算 $\mathrm{dist}(5,4)=2$，答案为 $\min(3,2)=2$。

批末重建后，这个二进入新的 $d[5]$，新红点列表清空。

CF342E：距离查询与块长选择

使用倍增 LCA，预处理 $O(n\log n)$，且

$$\mathrm{dist}(u,v)=\mathrm{depth}[u]+\mathrm{depth}[v]-2\mathrm{depth}[\mathrm{lca}(u,v)].$$

每次距离计算 $O(\log n)$，所以总时间为

$$O\left(n\log n+n\left\lceil\frac{q}{B}\right\rceil+qB\log n\right),$$

空间 $O(n\log n+B)$。平衡主要两项可取 $B\approx\sqrt{\frac{n}{\log}n}$，并截到 $[1,q]$。

若距离查询使用 $O(1)$ 的 LCA，扫描新红点的总时间降为 $O(qB)$，相应地重新平衡块长。

四种处理方式的适用条件

方法	决定它能否使用的条件
序列分块	整块摘要可维护，边界可逐项处理；可在线。
莫队	允许重排，端点可增删；二次离线还需批量求值。
轻重分类	总规模有界；大对象少，小对象单次代价低。
时间分块	旧数据可批量处理，少量新修改可单独补偿。

主讲到此结束。点击方法名进入对应拓展，其余专题见 拓展目录。各组都可返回对应主讲或本页。

目录

序列与值域分块
莫队与二次离线
参数与轻重分类
按时间分块
拓展与练习

拓展目录

按所需知识选讲；各组均保留完整题目与解法。

专题	从哪里接着学
区间统计与回滚	序列分块、莫队
轻重分类的进一步应用	轻重分类
参数分治与批量询问	参数分类
时间分块的进一步应用	时间分块
二进制与批量计算	按所用工具选择
立方根与抽样	分块、摊还与概率

返回讲授安排 · 返回主讲小结

拓展：区间统计与回滚

先完成序列分块与莫队。HDU 7337 还需离线扫描与平方数枚举；回滚部分从“加入容易、删除困难”的区间统计开始，最后一题需要并查集。

HDU 7337

JOISC 历史研究

P5906

CF763E

返回对应主讲 · 主讲小结 · 拓展目录

HDU 7337：区间平方数对

HDU 7337 原题：给定一个 1 到 n 的排列，询问区间内有多少对位置 $i < j$ 满足 $a_i + a_j$ 是完全平方数。至多五组数据，每组 $n, q \leq 10^5$。

HDU 7337：区间平方数对

HDU 7337 原题：给定一个 1 到 n 的排列，询问区间内有多少对位置 $i < j$ 满足 $a_i + a_j$ 是完全平方数。至多五组数据，每组 $n, q \leq 10^5$。

先把询问按右端点排序，扫描到位置 r 时，枚举可能的平方数 s，通过排列的逆映射找到值为 $s - a_r$ 的位置 i。若 $i < r$，就在位置 i 加一。此时，区间 $[l, r]$ 的和正好是对应询问的答案：每一对已经由右端点加入，再由左端点决定是否被统计。

HDU 7337：大量修改值得省掉一个对数

因为 $a_i + a_r \le 2n$，每个右端点只会产生 $O(\sqrt{n})$ 次修改。全程有 $O(n\sqrt{n})$ 次单点修改，却只有 q 次区间查询。

用单点修改 $O(1)$、区间查询 $O(\sqrt{n})$ 的普通分块，总时间为

$$O\left(n\sqrt{n} + q\sqrt{n} + q\log q\right).$$

最后一项来自询问排序，也可以按右端点分桶去掉。这里选择分块的理由，是修改次数明显多于查询次数。

JOISC 2014：历史研究

JOISC 2014 历史研究 要求区间内 $x \times \mathrm{count}(x)$ 的最大值，值 x 为正数。

JOISC 2014：历史研究

JOISC 2014 历史研究 要求区间内 $x \times \mathrm{count}(x)$ 的最大值，值 x 为正数。

加入一个元素时，只需增加计数并尝试更新答案；任意删除一个元素后，原来的最大值却可能失效。

回滚莫队把询问按左端点所在块分组，组内按右端点递增处理。设左端点块为 $[L, R]$：

1. 左右端点同块的询问直接暴力。
2. 其他询问的长期状态只维护 $[R+1, r]$，右端点单调向右移动。
3. 回答询问前保存基础状态，再把 $[l, R]$ 临时加入。
4. 记录答案后，逆序撤销刚才的临时修改，恢复长期状态。

回滚莫队：基础区间只向右走

左端点所在块

$l...R$

临时加入，再撤销

右侧基础区间

$R+1...r$

随询问右端点递增，只加入

① 记录栈高
保存基础状态

② 加左边界
最多 B 个

③ 读答案
含两侧贡献

④ 回滚
恢复原栈高

持久部分不撤销；临时部分只撤销最近的修改。短询问直接在块内暴力。

回滚莫队：撤销后必须恢复同一个状态

回滚必须恢复加入左边界之前的完整状态，包括统计量与答案。只记录本次改动，撤销成本与新增成本同阶；不能为每个询问复制整份状态。

每个左端点块最多让右指针走 n 步，每个询问临时加入至多 B 个元素，移动总量为

$$O\left(\frac{n^2}{B}+q\times B\right).$$

因此 $B\approx\frac{n}{\sqrt{q}}$，再截到合法范围。排序另需 $O(q\log q)$。这项技术适合“加入容易、任意删除困难，但撤销最近修改容易”的状态。

P5906：区间最远等值位置

P5906 原题 查询区间内两个相同数之间的最大下标差。

P5906：区间最远等值位置

P5906 原题 查询区间内两个相同数之间的最大下标差。

把每个值离散化，维护它在当前区间内的最左、最右位置，答案是两者之差的最大值。

长期右端点向右扩展时，更新这个值的最右位置；临时加入左侧位置时，更新最左位置。每次只会增大对应距离，答案直接取最大。记录修改前的端点值与答案，临时加入结束后逆序恢复即可。某个值第一次出现时，两端同时设为当前位置。

P5906：边界修改连同答案一起撤销

同块查询直接扫描并记录最早出现位置。每个询问只处理实际访问过的位置，初始化与清理均计入本次扫描成本。

回滚莫队移动量为 $O\left(\frac{n^2}{B}+q\times B\right)$，每次移动 $O(1)$，空间 $O(n+q)$。这道题与历史研究维护的数值不同，但“加入只让答案变好，撤销恢复旧快照”的结构相同。

CF763E：Timofey and our friends

CF763E Timofey and our friends 查询区间诱导子图的连通块数，$n, q \leq 10^5$、$K \leq 5$，边只连接编号差不超过 K 的点。

CF763E：Timofey and our friends

CF763E Timofey and our friends 查询区间诱导子图的连通块数，$n, q \leq 10^5$、$K \leq 5$，边只连接编号差不超过 K 的点。

加入顶点 u 时先新增一个连通块，再合并它与当前区间内邻点之间的边。

由于图简单，顶点最多有 $2K$ 个邻点。使用按大小合并、无路径压缩的回滚并查集；临时加入左边界后读取答案，再撤销全部临时合并，恢复基础区间的连通关系与连通块数。

CF763E：回滚并查集必须计入树高

只合并当前区间内部的边；换左端点块时重建基础状态，单个询问只撤销临时边。并查集单次合并为 $O(\log n)$。

时间为 $O\left(\left(\frac{n^2}{B}+q\times B\right)K\log n+q\log q\right)$，空间为 $O(n+m+q)$。这里的 K 与并查集树高都应出现，不能把一次新增顶点误认为恒定次数合并。

本组目录 · 对应主讲 · 主讲小结 · 拓展目录

拓展：轻重分类的进一步应用

先掌握总规模与阈值。网格题使用组合计数与 DP；女生赛 J 使用因子求和与扫描线；CF587F 需要 AC 自动机与 fail 树。其余题沿频次或图上度数分类展开。

小度的双色球

AGC001E

P8330

Graph Cuts

CF587F

ABC259Ex

P4062

ABC219G

女生赛 J

返回对应主讲 · 主讲小结 · 拓展目录

小度的双色球

百度之星 2023 决赛 H：小度的双色球 有 n 个盒子，每盒内的小球颜色互不相同。每次有放回地均匀抽一个盒子，直到抽到同时含颜色 x, y 的盒子；询问首次成功的期望抽取次数。

若永远不能成功，输出 -1；否则期望模 998244353 输出。$x \neq y$，盒数、询问数与小球总数 S 均不超过 2×10^5。

小度的双色球

百度之星 2023 决赛 H：小度的双色球 有 n 个盒子，每盒内的小球颜色互不相同。每次有放回地均匀抽一个盒子，直到抽到同时含颜色 x, y 的盒子；询问首次成功的期望抽取次数。

若永远不能成功，输出 -1；否则期望模 998244353 输出。$x \neq y$，盒数、询问数与小球总数 S 均不超过 2×10^5。

设同时含 x, y 的盒子有 c 个。$c = 0$ 时无解，否则每次独立成功的概率为 $\frac{c}{n}$，首次成功期望为 $\frac{n}{c}$，输出 $nc^{-1} \bmod 998244353$。

于是只需求两个颜色的盒子集合交集。若两种颜色均出现不超过 B 次，将位置表排序后双指针求交，单次 $O(B)$。

颜色交集：大集合预处理整行答案

若至少一个颜色是大颜色，则预处理它与其他颜色的交集。固定大颜色 x，先标记它的所有位置，再扫描其他颜色的位置表，总工作量为 $O(S)$。总预处理时间为 $O\left(\frac{S^2}{B}\right)$，小颜色查询共 $O(q \times B)$。

这里还要检查内存。把所有“大颜色 × 全部颜色”的答案存成矩形数组，可能比时间先超限。可以离线按大颜色分组，只回答与它有关的询问，用完临时计数后释放。

ABC259Ex：Yet Another Path Counting

ABC259Ex Yet Another Path Counting 中，网格大小为 $N \times N$，只能向右或向下走。统计起点和终点颜色相同的路径数，包含长度为零的路径；答案模 998244353。$N \leq 400$，颜色编号在 $[1, N^2]$ 内。

ABC259Ex：Yet Another Path Counting

ABC259Ex Yet Another Path Counting 中，网格大小为 $N \times N$，只能向右或向下走。统计起点和终点颜色相同的路径数，包含长度为零的路径；答案模 998244353。$N \le 400$，颜色编号在 $[1, N^2]$ 内。

某颜色只有少量格子时，枚举起终点。若坐标差为非负的 $\Delta x, \Delta y$，路径数为

$$\binom{\Delta x + \Delta y}{\Delta x}.$$

预处理阶乘后，每对格子可以常数时间计算，所有小颜色共 $O(N^2 B)$。

ABC259Ex：大量同色格子整张网格 DP

某颜色格子很多时，扫描整个网格。令 $d_{i,j}$ 为从任意一个该颜色格子出发、走到 (i,j) 的路径数：

$$d_{i,j} = d_{i-1,j} + d_{i,j-1} + [a_{i,j} = c].$$

在颜色为 c 的终点累加 $d_{i,j}$ 即可，其中也包含长度为零的路径。大颜色最多 $\frac{N^2}{B}$ 个，总时间为 $O\left(\frac{N^4}{B}\right)$。取 $B \approx N$，得到 $O(N^3)$。

AGC001E：BBQ Hard

BBQ Hard 求

$$\sum_{i<j}\binom{a_i+a_j+b_i+b_j}{a_i+a_j},$$

其中 $n \le 2\times 10^5$，$1 \le a_i, b_i \le A = 2000$，答案模 10^9+7。

AGC001E：BBQ Hard

BBQ Hard 求

$$\sum_{i<j}\binom{a_i+a_j+b_i+b_j}{a_i+a_j},$$

其中 $n\le 2\times 10^5$，$1\le a_i,b_i\le A=2000$，答案模 10^9+7。

单个组合数表示从 $(-a_i,-b_i)$ 向右、向上走到 (a_j,b_j) 的单调路径数。

在包含所有起终点的 $[-A,A]^2$ 网格中，把每个 $(-a_i,-b_i)$ 的初始值加一，重复点也累加。然后按坐标递增做

$$d_{x,y}=d_{x-1,y}+d_{x,y-1}+\mathrm{start}_{x,y}.$$

AGC001E：扣掉自己到自己的路径

每条路径唯一地由最后一步或其起点分类，故在所有 (a_j, b_j) 上求和得到所有有序点对 (i, j) 的贡献。

扣掉对角线

$$\sum_i \binom{2a_i + 2b_i}{2a_i},$$

再乘二的模逆元，即为 $i < j$ 的答案。时间 $O(A^2 + n)$，空间可用完整网格 $O(A^2)$，也可按行滚动并按行收集终点查询。所有运算模 $10^9 + 7$。

P4062：Yazid 的新生舞会

P4062 Yazid 的新生舞会 统计存在严格多数元素的子区间。$n \leq 5 \times 10^5$。

P4062：Yazid 的新生舞会

P4062 Yazid 的新生舞会 统计存在严格多数元素的子区间。$n \leq 5 \times 10^5$。

一个区间的严格多数元素至多有一个，这使得按颜色计数不会重复。

某颜色在整个数组中至多出现 B 次，那么以它为严格多数的区间长度必然小于 $2B$。因此可以枚举所有长度小于 $2B$ 的区间，维护出现次数与最大频率，但只统计多数元素属于小颜色的情况，时间为 $O(n \times B)$。

P4062：大颜色转成正和子区间计数

对于大颜色 c，将等于 c 的位置记为 $+1$，其他位置记为 -1。原问题成为统计和为正的子区间，也就是前缀和中满足 $p_i < p_j$ 的数对 $i < j$。

这里可以做到单颜色 $O(n)$：相邻前缀和只变化 1，维护当前值左侧已有前缀和的数量；当前值上升时加入刚跨过的频次，下降时减去刚跨过的频次。用频次数组即可更新，无须每次在树状数组上查询。

总时间为 $O\left(n \times B + \frac{n^2}{B}\right)$。若直接用树状数组统计数对，则大颜色部分要再乘 $\log n$。

P4062：前缀只移动一格时的计数

固定大颜色。扫描前维护前缀值 p、频次 f，以及已见前缀中小于 p 的个数 z。初始 $p = z = 0$、$f[0] = 1$，其余频次为零。

读入当前的 $+1$ 或 -1，按下式更新：

$$+1: \quad z \leftarrow z + f[p], \quad p \leftarrow p + 1;$$

$$-1: \quad p \leftarrow p - 1, \quad z \leftarrow z - f[p].$$

此时 z 正好是以当前位置结尾的正和区间数。先把 z 加入答案，再令 $f[p]$ 加一，将当前前缀留给后续位置使用。下标加 n 偏移即可使用数组，答案用 64 位整数。

P8330：ZJOI2022 众数

ZJOI2022 众数 选择一个区间整体加上整数 $k \in [-10^9, 10^9]$，最大化最终众数次数，并输出所有可能的众数值。

$2 \le n \le 2 \times 10^5$，$1 \le a_i \le 10^9$，且 a_i 不全相等；所有测试的 $\sum n \le 5 \times 10^5$。

P8330：ZJOI2022 众数

ZJOI2022 众数 选择一个区间整体加上整数 $k \in [-10^9, 10^9]$，最大化最终众数次数，并输出所有可能的众数值。

$2 \le n \le 2 \times 10^5$，$1 \le a_i \le 10^9$，且 a_i 不全相等；所有测试的 $\sum n \le 5 \times 10^5$。

将原值离散化，记颜色 x 的总出现次数为 c_x。

若把区间内的颜色 x 平移成 y，最终 y 的数量为

$$c_y + \mathrm{count}_x(l, r) - \mathrm{count}_y(l, r).$$

P8330：最优众数一定是原有的值

输入不全相等，所以总能让原众数再增加至少一次；最优答案严格大于原最大频率。一个原先不存在的目标值只能获得区间内某种颜色的次数，不能达到这个上界。因此只需枚举原来出现过的目标值 y，且平移量 $y-x$ 自动满足题面的范围。

取阈值 B。若 x 或 y 的总次数超过 B，按重颜色处理；剩下两种颜色都轻，用出现位置之间的区间枚举。

P8330：把其他颜色移成重颜色

固定重颜色 h，建立其出现次数前缀和 $H(t)$。对另一颜色 y 的出现位置 $p_1 < \ldots < p_c$，只扫描这个列表。

把 y 改成 h 的最大增量为

$$\max_{1 \le i \le j \le c} \big((j-i+1) - \big(H(p_j) - H(p_i - 1)\big)\big).$$

扫描 j 时维护 $H(p_i - 1) - (i-1)$ 的历史最大值，就能在 $O(c)$ 内求出，并更新目标 h 的答案。

P8330：把重颜色移成其他颜色

反过来，把 h 改成 y，需要最大化 $H(r)-H(l-1)-\mathrm{count}_y(l,r)$。看前缀差 $D(t)=H(t)-\mathrm{count}_y(1,t)$：它只在 h 位置加一，在 y 位置减一。每段相邻 y 之间，最小值在开头，最大值在结尾。

因此依次考察每个 y 之前的 $H\left(p_j-1\right)-(j-1)$，减去历史最小前缀差；经过该 y 后，用 $H\left(p_j\right)-j$ 更新历史最小值。加上开头零前缀与末尾 n 的检查，即可得到反方向增量并更新目标 y。

所有其他颜色的出现次数之和为 n，一个重颜色总共 $O(n)$。重颜色数至多 $\frac{n}{B}$，这一部分为 $O\left(\frac{n^2}{B}\right)$。

P8330：预处理达到 k 次的最短右端点

固定轻目标颜色 y，出现位置为 $p_1 < \ldots < p_c$，补左哨兵 $p_0 = 0$。区间左端点可以移到某个 $p_i + 1$：从左侧最近的 y 后面开始，只会多得到候选源颜色，不会损失更多 y。

对 $1 \le k \le B$，定义 $R_k(l)$：从 l 开始，第一次出现某种轻颜色达到 k 次的最小右端点；不存在则为 $n + 1$。固定 k，用双指针维护窗口中有多少种轻颜色已达到 k 次，左右端点各走至多 n 步，便能在 $O(n)$ 时间求出整行 R_k。

P8330：内部拿到 k 次，外部损失最少

现在只考虑内部获得 k 个相同元素。对左端点 p_i+1，选择最短的右端点 $R_k(p_i+1)$，损失的目标颜色 y 最少。若这个右端点包含到第 j 个 y，就能更新

$$\mathrm{answer}_y = \max\bigl(\mathrm{answer}_y, c-(j-i)+k\bigr).$$

按 i 递增扫描时，$R_k(p_i+1)$ 也不减，所以指针 j 只需向右移动。一个目标颜色花 $O(c+1)$，所有轻目标合计 $O(n)$。若最短区间不存在，跳过即可。

P8330：轻颜色区间的边界变化

限定源颜色与目标 y 都轻，取该分支的最优区间：

原区间 $[l, r]$	$\rightarrow$	轻源颜色出现 $k \le B$ 次 p_i 是 l 前最后一个 y
扩左界 $[p_i + 1, r]$	$\rightarrow$	新增部分不含 y 原来的 k 个源颜色仍在
缩右界 $[p_i + 1, R_k(p_i + 1)]$	$\rightarrow$	某种轻颜色已达到 k 次 右界不超过 r，损失的 y 不增

候选不减，因此不会漏掉最优值；源颜色若恰为 y，候选只有 c_y。

P8330：逐行处理，把空间降到线性

逐个处理 k，算完一行 R_k 就更新全部轻目标，然后复用这行内存。无需保存整个 $n \times B$ 的表。加上重颜色部分，总时间为 $O\left(n \times B + \frac{n^2}{B}\right)$，额外空间为 $O(n)$，取 $B \approx \sqrt{n}$。

最后取各目标最优值的最大值，输出达到它的所有原颜色值。正式数据 $\sum n \leq 5 \times 10^5$，因此只保留一行预处理结果对内存尤为重要。

ABC219G：Propagation

ABC219G Propagation：无向简单图，顶点 v 初始写着 v。一次操作读取 x 的当前值，并将所有邻点改为这个值，最后输出全部顶点的值。$n, m, q \leq 2 \times 10^5$，允许 $m = 0$。

ABC219G：Propagation

ABC219G Propagation：无向简单图，顶点 v 初始写着 v。一次操作读取 x 的当前值，并将所有邻点改为这个值，最后输出全部顶点的值。$n, m, q \leq 2 \times 10^5$，允许 $m = 0$。

直接扫描邻点，一次最坏 $O(n)$。**若只把度数大的点的广播延后，读取一个点时要补看多少条消息？**

阈值取 $B \geq 1$，度数至少 B 的称为重点。由总度数 $2m$，重点数至多 $2\frac{m}{B}$。

ABC219G：消息按时间戳覆盖

每点保存已写入的值和时间戳；每个重点另存最近一次广播的值和时间戳。预处理每个点的重点邻居表。

读取 v：将已写入记录与所有重点邻居的广播比较，取时间戳最大的值，并缓存它。

第 t 次操作先读取 x。若 x 是轻点，向全部邻点写入 (t, value)；若是重点，仅记录自己的广播 (t, value)。广播值是**操作发生时的快照**，之后 x 改值不应改变旧消息。

ABC219G：每一次有效赋值都能追溯

对任意 v，来自轻点的赋值已经直接写入；来自重点的赋值仍保存在对应广播记录里。当前值就是这些消息中最新的一条，这个不变量在每次操作后成立。

重点后来再次广播时，旧消息已被新消息覆盖，保留最新一条足够。操作结束后，对每个点执行一次读取，再输出；孤立点始终保持初始值。

总时间 $O\left(n+m+q\left(B+\frac{m}{B}\right)\right)$，末次读取总共扫描至多 $2m$ 条邻接关系。空间 $O(n+m)$。可取 $B \approx \sqrt{\max(1,m)}$。

官方题解

Graph Cuts

Graph Cuts 维护顶点集合 U。加入或移除顶点后，查询一条恰有一个端点在 U 中的现存边，并将它永久删除。

Graph Cuts

Graph Cuts 维护顶点集合 U。加入或移除顶点后，查询一条恰有一个端点在 U 中的现存边，并将它永久删除。

按初始度数分轻重，度数大于 B 为重点，重点至多有 $2\frac{m}{B}$ 个。删除边不会增加度数，所以分类始终有效。

只连接两个轻点的边，放进一个可删除的割边集合。轻点切换状态时，扫描至多 B 条原邻边，忽略已经删除的边，更新这些轻轻边是否属于割。

对于每个重点 h，维护两个邻点集合，分别存当前在 U 内外且与 h 有现存边的邻点。任意顶点 v 改变状态时，只需通知它的重点邻居，把 v 从一侧集合移到另一侧。这至多涉及 $2\frac{m}{B}$ 个重点。

Graph Cuts：两类候选覆盖全部割边

查询先看轻轻割边集合；若为空，逐个检查重点，从与它颜色相反的邻点集合取任意一个。每条重边至少有一个重点端点，因此两部分合起来覆盖全部割边。

找到边后标记删除，并从涉及的重点邻点集合或轻轻割边集合移除。重复删除不会发生，因为候选集合始终只存现存边。

用平衡树保存可删除集合，每次操作为 $O\big((B+\frac{m}{B})\log n\big)$，初始化与存储为 $O((n+m)\log n)$ 时间、$O(n+m)$ 空间。取 $B\approx\sqrt{m}$。若用哈希集合，可获得相应的期望常数操作，但这里保留平衡树的确定性复杂度。

女生赛 J：区间 gcd 平方和

2024 CCPC 女生赛 J：最大公因数的平方和：给两个 1 到 n 的排列 a, b，询问两个下标区间，求

$$\sum_{i=l}^{r}\sum_{j=L}^{R}\gcd(a_i,b_j)^2 \bmod 2^{32},\quad n,q\le 10^5.$$

女生赛 J：区间 gcd 平方和

2024 CCPC 女生赛 J：最大公因数的平方和：给两个 1 到 n 的排列 a, b，询问两个下标区间，求

$$\sum_{i=l}^{r}\sum_{j=L}^{R}\gcd\left(a_i, b_j\right)^2 \bmod 2^{32}, \quad n, q \le 10^5.$$

直接枚举两段中的数对，单次可能达到 $O(n^2)$。两个区间独立变化，无法只靠维护一段序列的统计量解决。

希望找到一个分解：每个公因子 d 贡献一个固定权值，两边各自只需统计 d 的倍数。这样就能把“一对数的关系”拆成“两份独立的计数”。

女生赛 J：按公因子展开平方

定义二阶 Jordan 函数 J_2，满足

$$\sum_{d|x} J_2(d) = x^2.$$

因此 $\gcd(x,y)^2 = \sum_{d|x,d|y} J_2(d)$。设 A_d, B_d 分别为两个询问区间内 d 的倍数个数，答案就是

$$\sum_{d=1}^{n} J_2(d) A_d B_d.$$

初始化 $J_2(d) = d^2$；按 $d = 1, \ldots, n$ 递增，将已确定的 $J_2(d)$ 从每个严格倍数中减去，筛法共 $O(n \log n)$。

女生赛 J：小因子逐个算整批询问

取阈值 K。对于每个 $d \le K$，扫描两个排列，分别建立“是否为 d 的倍数”的前缀计数 P_d, Q_d。

每次询问的这一项可在常数时间求出：

$$J_2(d)(P_d(r) - P_d(l-1))(Q_d(R) - Q_d(L-1)).$$

处理完一个 d 就复用两条前缀数组。总时间 $O(K(n+q))$，空间 $O(n+q)$。

若所有 d 都这样处理，仍需 $O(n(n+q))$。**大因子的优势不是数量少，而是每个因子的倍数少。**

女生赛 J：大因子与稀疏点集

固定 $d > K$。将 a 中所有 d 的倍数位置记为 X_d，b 中对应位置记为 Y_d；在 $X_d \times Y_d$ 的每个点放权值 $J_2(d)$。

一个查询就是取矩形 $[l, r] \times [L, R]$ 内的点权和。它恰好选出 $A_d B_d$ 个点，因此与按因子求和一致。

$$M = \sum_{d>K} \left\lfloor \frac{n}{d} \right\rfloor^2 \le n^2 \sum_{d>K} \frac{1}{d^2} = O\left(\frac{n^2}{K}\right).$$

同一点可以由多个公因子产生，权值必须相加：这正是 $\gcd(a_i, b_j)^2$ 的分解。减少的是**待处理点的总数**，无需保存完整的 $n \times n$ 矩阵。

女生赛 J：扫描线里修改多，查询少

记 $F(x,y)$ 为前缀矩形的权值和。每个询问拆成

$$F(r,R)-F(l-1,R)-F(r,L-1)+F(l-1,L-1).$$

按第一维分桶这些前缀询问。扫描到 $a_i=x$ 时，枚举 x 的大因子 $d>K$，再枚举 b 中 d 的倍数 y，向位置 $\mathrm{pos}_b(y)$ 加 $J_2(d)$；加入后再回答第一维为 i 的询问。

数组此时恰好包含横坐标不超过 i 的全部权值，前缀和就是 $F(i,y)$；第一维或第二维为零时返回零。

这里有 M 次修改，只有 $4q$ 次查询。第二维用块长 B 的数组分块：单点加 $O(1)$，前缀和 $O\left(B+\frac{n}{B}\right)$，取 $B\approx\sqrt{n}$。

女生赛 J：两次分块分别平衡不同代价

用筛法预存每个数的大因子表；总空间 $O(n \log n + q)$。扫描时即时产生更新，无须存下 M 个带权点。

加上筛法、小因子和扫描线，总时间为

$$O\left(n \log n + K(n+q) + \frac{n^2}{K} + q\sqrt{n}\right).$$

前两类因子代价平衡时 $K \approx \frac{n}{\sqrt{n+q}}$，截到 $[1, n]$；当 $q = \Theta(n)$，才有 $K \approx \sqrt{n}$，总时间 $O(n\sqrt{n})$。

模 2^{32} 的乘法用无符号 64 位中间值，减法按模处理。排列性质保证每个大因子在两边各有 $\left\lfloor \frac{n}{d} \right\rfloor$ 个位置，故点数上界成立。

CF587F：Duff is Mad

CF587F Duff is Mad 给出若干字符串，询问编号在 $[l, r]$ 内的模式串在文本 s_k 中总共出现多少次。字符串数、询问数与总长度 L 均不超过 10^5。

CF587F：Duff is Mad

CF587F Duff is Mad 给出若干字符串，询问编号在 $[l,r]$ 内的模式串在文本 s_k 中总共出现多少次。字符串数、询问数与总长度 L 均不超过 10^5。

建立 AC 自动机和 fail 树。扫描文本的每个字符，会到达一个自动机状态。模式串的终止节点是这个状态在 fail 树上的祖先，当且仅当该模式串在当前位置结束。

对长文本 $|s_k|>B$，单独扫描它，记录每个到达状态的次数，再沿 fail 树向父亲累加。这样得到每个模式串在该文本中的出现次数；按模式串编号做前缀和，就能常数时间回答与这个文本有关的询问。长文本最多 $\frac{L}{B}$ 个，这部分总时间为 $O\left(\frac{L^2}{B}+q\right)$。

CF587F：短文本查询扫描所有前缀状态

对短文本，把 $[l,r]$ 拆成两个前缀询问，扫描模式串编号。加入一个模式串时，把它的终止节点在 fail 树上的整个子树加一。查询文本时，沿文本扫描自动机，对每个到达状态做单点查询，再求和。

注意要扫描文本的所有前缀状态，只查询文本的最终状态会漏掉中间出现的匹配。fail 树的 DFS 序加上树状数组，可实现子树加、单点查。这部分时间为 $O((L+q\times B)\log L)$。

长文本值得预先算出所有模式的答案；短文本值得在每次询问时重新扫描。这里平衡的是文本长度与询问次数。

本组目录 · 对应主讲 · 主讲小结 · 拓展目录

拓展：参数分治与批量询问

接参数分类：P5309 与 CF1207F 按模数大小处理；P7811 再结合 RMQ。猫树的两侧摘要会在题内说明。可先独立完成 CF1207F，再阅读解法。

P5309　　CF1207F（练习）

P7811

返回对应主讲 · 主讲小结 · 拓展目录

P5309：初始化

初始化 支持将下标 $y, y+x, y+2x, \ldots$ 的元素加 z，其中 $1 \le y \le x$；查询区间和，答案模 10^9+7。$n, q \le 2 \times 10^5$。

P5309：初始化

初始化 支持将下标 $y, y+x, y+2x, \ldots$ 的元素加 z，其中 $1 \le y \le x$；查询区间和，答案模 10^9+7。$n, q \le 2 \times 10^5$。

将初始数组和大模数修改放进普通位置分块。若 $x > B$，受影响位置至多 $\frac{n}{B}$ 个，逐个修改元素与块和即可，每个位置 $O(1)$。

若 $x \le B$，不去修改实际元素，而是维护余数标记。把余数写成 $1, \ldots, x$，余数 x 对应通常的零余数。令 $P_x(t)$ 是前 t 个余数标记之和，$P_x(0) = 0$。

P5309：余数标记也要支持前缀求和

给余数 y 加 z 时，将 $P_x(y), ..., P_x(x)$ 全部加 z，只需 $O(x) \le O(B)$。这样，前 r 个下标从模数 x 的修改得到的总贡献为

$$\left\lfloor \frac{r}{x} \right\rfloor P_x(x) + P_x(r \bmod x).$$

每个小模数能在常数时间求贡献，枚举所有小模数并做两个前缀相减，再加上普通分块的区间和即可。查询 $O\left(B + \sqrt{n}\right)$，修改 $O\left(B + \frac{n}{B}\right)$，取 $B \approx \sqrt{n}$ 后每次为 $O\left(\sqrt{n}\right)$，空间 $O\left(n + B^2\right)$。

如果只存每个余数的单独标记，查询时再把所有余数扫一遍，就会退化成 $O\left(B^2\right)$。把便宜的前缀更新放在修改里，才是这里平衡两类工作的关键。

CF1207F：Remainder Problem

Remainder Problem 在长度为 $V = 500000$ 的零数组上做单点加，查询下标模 x 为 y 的元素和。

CF1207F：Remainder Problem

Remainder Problem 在长度为 $V = 500000$ 的零数组上做单点加，查询下标模 x 为 y 的元素和。

对每个 $x \le B$，预存 $\mathrm{sum}[x][r]$。位置 p 加 v 时，逐个小模数更新 $\mathrm{sum}[x][p \bmod x]$，同时修改原数组，时间 $O(B)$。小模数查询直接查表；大模数查询枚举 $y, y+x, y+2x, \ldots$，跳过下标零，最多 $O\left(\frac{V}{B}\right)$ 项。

总时间 $O\left(q\left(B + \frac{V}{B}\right)\right)$、空间 $O(V + B^2)$，取 $B \approx \sqrt{V}$。这与 P5309 的区别是：这里修改一个位置，查询一个余数类；P5309 则修改一个余数类，查询一个连续区间。

P7811：区间取模最小值

P7811 你的名字。 给定数组 a，查询 $\min_{l\le i\le r}(a_i \bmod k)$。

$n,q\le 3\times 10^5$，$1\le a_i,k\le V=10^5$；完整数据的空间限制为 128 MiB。

P7811：区间取模最小值

P7811 你的名字。 给定数组 a，查询 $\min_{l\le i\le r}(a_i \bmod k)$。

$n,q\le 3\times 10^5$，$1\le a_i,k\le V=10^5$；完整数据的空间限制为 128 MiB。

下面按模数大小分类，复杂度同时依赖数组长度 n 与值域上界 V。

若 $k\le B$，对每个实际被询问的小 k，建立数组 $a_i \bmod k$ 的静态 RMQ，直接回答这一组询问。用稀疏表时，预处理共 $O(n\times B\log n)$；各个 k 分组处理并复用表，空间为 $O(n\log n)$。

区间模 k：大模数只枚举少量倍数

若 $k > B$，枚举 $t = 0, k, 2k, ..., V$，求区间中不小于 t 的最小元素 x，用 $x - t$ 更新答案。每个元素自己的下取整倍数必在枚举中，因此不会漏解；若查出的 $x \geq t + k$，这次候选不会优于同一元素在其真正倍数处的余数。

按 t 从大到小扫描，逐次加入值不小于 t 的数组元素。未加入位置为正无穷，已加入位置为原值。每个子询问就变成一次区间最小值。修改只有 n 次，查询却可能有 $O\left(q \times \frac{V}{B}\right)$ 次，适合用慢修改、快查询的 RMQ。

P7811：按阈值轮转询问

不必一次存下全部子询问。建立阈值 $0, \ldots, V$ 的桶，每个原询问只占一个链表节点，最初放在 $t = \left\lfloor \frac{V}{k} \right\rfloor k$ 的桶中。

扫描到 t，先加入值为 t 的元素，再处理本桶：查询区间最小值 x，用 $x - t$ 更新答案；随后把同一节点移到 $t - k$ 的桶。处理完 $t = 0$ 后结束该询问。没有有限的 x 时跳过本次更新。

每个询问在任意时刻只驻留一个桶，存储仍为 $O(q + V)$。元素也可按值分桶，整个阈值扫描额外花费 $O(n + V)$，无需排序或展开所有子询问。

猫树：分界两侧的前后缀

下标从零开始，用正无穷将长度 m 补到二的幂，再按中点递归二分。每个中点保存左半段的后缀最小值与右半段的前缀最小值。

$l \dots \mathrm{mid}$
左半段的后缀最小值

$\mathrm{mid}+1 \dots r$
右半段的前缀最小值

不同端点第一次分开时，查询区间恰由图中两段组成。由 $l \oplus r$ 的最高置位找到这一层，两个预存值取最小即可，时间 $O(1)$；单点区间直接读原值。

建表与存储均为 $O(m \log m)$。修改一个位置时，重算各层受影响半边的前后缀，长度之和为 $O\left(m + \frac{m}{2} + \dots\right) = O(m)$。

两层猫树：块内与完整块

按位置切成长 C 的块，每块建一棵猫树；所有块最小值组成的数组，再建一棵外层猫树。

查询同块区间只查块内。跨块区间由三项取最小：

左边界块后缀 块内猫树	中间完整块 外层猫树	右边界块前缀 块内猫树

单点修改先重建块内相关前后缀，再更新外层的一个块最小值。两层的序列长度分别为 $C, \frac{n}{C}$，因此修改 $O\left(C+\frac{n}{C}\right)$、查询 $O(1)$。

P7811：时间与空间的总代价

取 $C \approx \sqrt{n}$，n 次修改共 $O(n\sqrt{n})$，大模数子询问每次 $O(1)$。
计入小模数预处理、全部询问和阈值扫描，总时间与空间为

$$T = O\left(nB\log n + n\sqrt{n} + q\frac{V}{B} + V\right),$$

$$S = O(n\log n + q + V).$$

其中 $1 \le B \le V$。小模数按组复用 RMQ，内外层猫树的总空间为 $O(n\log n)$；阈值桶复用询问节点，空间不再随子询问总数增长。

本组目录 · 对应主讲 · 主讲小结 · 拓展目录

拓展：时间分块的进一步应用

先完成 CF342E。动态 Fourier 需要快速 Fourier 变换；桥梁需要回滚并查集；MONSTER 使用超集和变换；最后一题同时按时间与位置分块。

动态 Fourier

MONSTER

P5443 桥梁

Multiply Then Plus

返回对应主讲 • 主讲小结 • 拓展目录

动态 Fourier 系数查询

锅 · QOJ 镜像。设

$$\hat{A}_k = \sum_{j=0}^{n-1} A_j \omega^{kj}, \quad \omega = e^{\frac{2\pi \mathrm{i}}{n}}.$$

支持单点加法和查询某个变换系数。

动态 Fourier 系数查询

锅 · QOJ 镜像。设

$$\hat{A}_k = \sum_{j=0}^{n-1} A_j \omega^{kj}, \quad \omega = e^{\frac{2\pi \mathrm{i}}{n}}.$$

支持单点加法和查询某个变换系数。

一次完整 FFT 可以在 $O(n \log n)$ 时间重算全部系数，但一次单点修改 (j, Δ) 对某个系数的影响只有 $\Delta\omega^{k\times j}$。

动态 Fourier：积累一批修改后重建

每累计 B 次修改就重建一次。在两次重建之间，查询从上次重建的答案出发，枚举尚未合并的修改，时间 $O(B)$。若修改、查询分别有 u, v 次，总时间为

$$O\left(n \log n + \left(\frac{u}{B}\right) n \log n + v \times B\right).$$

当两类操作数量同阶时，取 $B \approx \sqrt{n \log n}$。复数版本还需要另行控制数值误差；相同的分批思路也适用于相应的有限域变换。

P5443：桥梁

P5443 桥梁 支持修改无向图边权，并查询仅允许经过权值至少为 y 的边时，从顶点 x 能到达多少点。$n \leq 5 \times 10^4$，$m, q \leq 10^5$，允许 $m = 0$。

P5443：桥梁

P5443 桥梁 支持修改无向图边权，并查询仅允许经过权值至少为 y 的边时，从顶点 x 能到达多少点。$n \le 5 \times 10^4$，$m, q \le 10^5$，允许 $m = 0$。

预读一批操作，将本批中会被修改的边全部标为变化边，其他边为稳定边。稳定边按权值降序排列，本批询问也按阈值降序处理。随阈值下降，逐渐把符合条件的稳定边加入并查集。

对于当前询问，再按它发生时的权值检查所有变化边，临时加入达标的边，读取 x 所在连通块的大小，然后撤销这些临时合并。变化边在各个询问时刻的权值，可按原时间顺序扫一遍该批操作得到。

桥梁：稳定边只加入，变化边才回滚

每批初始化并查集为 $O(n)$；稳定边排序与处理为 $O(m\log(m+1)+m\log n)$。变化边检查和临时合并为 $O\left(B^2\log n\right)$，询问排序为 $O(B\log(B+1))$。

记批数 $H=\lceil\frac{q}{B}\rceil$，总时间为

$$O(H(n+m\log(m+1))+(Hm+qB)\log n+q\log(B+1)).$$

稳定边在本批内只加入一次，变化边才需要回滚。两部分的职责不能混在一起。

桥梁：阈值排序后仍须保留原时间

批首记下所有变化边的权值，并为每次询问保留它在原操作序列中的位置。处理某次询问时，先恢复这份短快照，再按原次序重放该询问以前的修改；同一条边最后一次赋值生效。

这一重放每次至多 B 步，只复用一份长度至多 B 的临时数组，不必存下每次询问的整份图。

加入达标稳定边后保存栈高，再临时加入变化边；读取连通块大小后只回滚到这个栈高。并查集按大小合并、不作路径压缩，成功合并记录父指针与原大小。

批末按原时间执行全部修改，作为下一批的初始边权。空间为 $O(n+m+q+B)$；孤立点也应初始化成大小一的连通块。

桥梁：同一批询问共享稳定边

稳定边

本批从未修改

按权值降序加入并查集

变化边

本批至多 B 条

按询问时刻的权值逐条检查

阈值从大到小：稳定边只会增加，变化边每次临时合并。

$$\text{稳定状态} \to \text{保存栈高} \to \text{临时合并} \to \text{查询} \to \text{回滚}$$

回滚只恢复变化边造成的合并，保留已经满足阈值的稳定边。

CodeChef MONSTER

CodeChef MONSTER 中，一次操作 (x, y) 对所有满足 $k \text{ AND } x = k$ 的怪物 k 造成 y 点伤害。伤害非负，求每次操作后存活的怪物数量。

CodeChef MONSTER

CodeChef MONSTER 中，一次操作 (x, y) 对所有满足 $k \text{ AND } x = k$ 的怪物 k 造成 y 点伤害。伤害非负，求每次操作后存活的怪物数量。

可以改成求每只怪物第一次死亡的时间。把操作按时间分块，设掩码值域为 $U = 2^d$。对于一个操作块，将伤害按掩码累加，再做高维后缀和，得到每个 k 从其所有超集掩码受到的总伤害。

若某只怪物承受完整块后仍存活，就直接扣除这一块的伤害。若它在本块死亡，再枚举块内操作，找到准确的死亡时刻。每只怪物只会进入一次这样的细查。

MONSTER：超集掩码的伤害怎样汇总

先令 $f[x]$ 为本批掩码恰好为 x 的伤害和。依次处理每个二进制位 b；对于第 b 位为零的掩码 s，执行

$$f[s] \leftarrow f[s] + f\left[s \text{ OR } 2^b\right].$$

已处理的位可以任意补一，未处理的位必须与 s 相同。按位归纳，完成后 $f[s]$ 恰为所有超集掩码的伤害总和，即编号 s 的怪物在本批受到的伤害。

死亡当批的细查从批首剩余血量开始，按原顺序扣除命中的伤害，第一次不大于零就是死亡时刻。伤害非负保证它以后不会复活；累加与剩余血量使用足够宽的整数。

MONSTER：每只怪物只细查死亡的那一批

总时间可写为

$$O\left(q+\left\lceil\frac{q}{B}\right\rceil(d\times U+n)+n\times B\right).$$

批次数为 $\lceil\frac{q}{B}\rceil$，每批高维变换的规模由掩码值域 U 决定。得到死亡时间后做前缀计数，即可输出每次操作后的存活数。

Multiply Then Plus

Multiply Then Plus 维护直线 $a_i x + b_i$，支持单点替换直线，以及查询编号区间 $[l, r]$ 内在给定 x 处的最大值。

Multiply Then Plus

Multiply Then Plus 维护直线 $a_i x + b_i$，支持单点替换直线，以及查询编号区间 $[l, r]$ 内在给定 x 处的最大值。

设时间块长为 T，位置块长为 B。在时间块开始时，把本批会修改的位置剔除，给剩余位置的每个块建立静态直线查询结构，例如李超树。

Multiply Then Plus：时间和位置分别平衡

预读所有询问横坐标并离散化，记不同横坐标数为 V，排序花费 $O(q\log q)$。每个时间块重建至多 n 条稳定直线，耗时 $O(n\log(V+1))$，共 $\lceil \frac{q}{T} \rceil$ 批。

一次查询访问至多 $\frac{n}{B}$ 个完整位置块，扫描两端至多 $2B$ 个稳定位置，再检查变化列表中落在 $[l,r]$ 内的当前直线；最后一项至多有 T 个位置。因此总时间为

$$O\left(q\log q+\left\lceil\frac{q}{T}\right\rceil n\log(V+1)+q\left(\left(\frac{n}{B}\right)\log(V+1)+B+T\right)\right).$$

固定 V 后，B 平衡完整块查询与边界扫描，T 平衡批首重建与变化列表扫描。两个块长不必相同。

本组目录 · 对应主讲 · 主讲小结 · 拓展目录

拓展：二进制与批量计算

位拆分部分需要异或、子集枚举与 bitset；树题需要树上 DP；COUNTARI 使用卷积；近似匹配使用字符串指纹。P5021 是树上判定的独立练习，保留完整证明。

CF1716E

CF914F

CF1039D

COUNTARI

P7738

Subset

CF840E

P5021（练习）

Ucup Shenzhen G

返回对应主讲 · 主讲小结 · 拓展目录

CF1716E：Swap and Maximum Block

CF1716E Swap and Maximum Block 给出长度 $N = 2^n$ 的数组。每次将下标二进制第 k 位不同、其他位相同的元素交换，然后查询允许空段的最大子段和。$n \le 18$，$q \le 2 \times 10^5$，$|a_i| \le 10^9$。

CF1716E：Swap and Maximum Block

CF1716E Swap and Maximum Block 给出长度 $N = 2^n$ 的数组。每次将下标二进制第 k 位不同、其他位相同的元素交换，然后查询允许空段的最大子段和。$n \le 18$，$q \le 2 \times 10^5$，$|a_i| \le 10^9$。

使用从零开始的下标后，这次交换就是将位置 i 变成 $i \oplus 2^k$。所有操作叠加后，当前位置 i 读到的原元素为 $a_{i \oplus m}$，其中 m 是累计异或掩码。

CF1716E：预处理每种块内排列

取块长 $B = 2^b$。掩码的高位只改变块的顺序，低位只改变每个块内部的顺序。对每个块预处理全部 B 个低位掩码下的四个信息：总和、最大前缀和、最大后缀和、最大子段和。

设左右两段的信息为 (s_L, p_L, t_L, v_L)、(s_R, p_R, t_R, v_R)，合并公式是

$$s = s_L + s_R, \quad p = \max(p_L, s_L + p_R),$$

$$t = \max(t_R, s_R + t_L), \quad v = \max(v_L, v_R, t_L + p_R).$$

CF1716E：按高位顺序合并块摘要

查询时按高位掩码确定块顺序，取出对应低位状态，依次合并，时间为 $O(\frac{N}{B})$。最直接的预处理枚举每个块的每种排布，时间 $O(N \times B)$，最终存储 $O(N)$ 份四元组。若按二叉结构递归合并不同掩码，还可以进一步减少预处理时间。

这一题的拆分依据是二进制位，而不是对每次交换逐个模拟。

CF1716E：递归构造全部低位状态

把一个块二分。设半块长度为 H，左右半块的状态表为 $L[x], R[x]$，$0 \le x < H$；令 $M(A, B)$ 表示按前述公式依次合并两段。

$$F[x] = M(L[x], R[x]), \quad F[H + x] = M(R[x], L[x]).$$

掩码的最高位决定两半是否互换，其余位在两半内采用相同排列。叶子值为 a 时，摘要为 $(a, \max(0, a), \max(0, a), \max(0, a))$；空段摘要为 $(0, 0, 0, 0)$。

每层合并 B 个状态，一个块预处理 $O(B \log B)$，所有块共 $O(N \log B)$。生成父表后释放子表，保留最终表共 $O(N)$ 空间；后续查询仍为 $O(\frac{N}{B})$。所有和用 64 位整数。

CF1716E：把异或拆成选块与块内重排

$$i = p \times 2^b + t, \quad m = h \times 2^b + x$$

高位

$$p \to p \oplus h$$

整块换位置

低位

$$t \to t \oplus x$$

每块执行相同的内部排列

例如 $B = 4$，位置 $6 = (01|10)_2$，掩码 $5 = (01|01)_2$。

$$6 \oplus 5 = (00|11)_2 = 3.$$

预处理低位变化；查询时按高位顺序合并块信息。

Subset：动态子集计数

Subset 支持插入、删除 d 位非负整数，并查询满足 $a \text{ AND } s = a$ 的元素个数。原题 $d = 16$，操作数不超过 2×10^5。

Subset：动态子集计数

Subset 支持插入、删除 d 位非负整数，并查询满足 $a \text{ AND } s = a$ 的元素个数。原题 $d = 16$，操作数不超过 2×10^5。

把高 h 位和低 $d - h$ 位拆开。

维护 $F[X][y]$：低位恰好为 y，高位是 X 的子集的元素个数。

插入 (x, y) 时，枚举高位 x 的所有超集 X，令 $F[X][y]$ 加一；删除则减一。查询 (X, Y) 时，枚举 Y 的所有子集 y，累加 $F[X][y]$。

修改最坏为 $O(2^h)$，查询最坏为 $O(2^{d-h})$，空间为 $O(2^d)$。当 $d = 16$ 且修改、查询数量接近时，拆成八位和八位很自然。

“更新枚举超集”与“查询枚举子集”容易写反。先明确 F 的定义，再推两种操作，会比记结论稳妥。

CF914F：Substrings in a String

Substrings in a String 支持单点改字符，查询模式串 T 在 $S[l,r]$ 中的出现次数。所有模式长度之和不超过 10^5。

CF914F：Substrings in a String

Substrings in a String 支持单点改字符，查询模式串 T 在 $S[l,r]$ 中的出现次数。所有模式长度之和不超过 10^5。

为每个字符 c 维护位集 B_c，第 i 位表示 $S_i = c$。单点修改只清除旧字符对应的一位，再设置新字符对应的一位。

模式长度为 m 时，合法起点集合为

$$R = \bigwedge_{j=0}^{m-1} \left(B_{T_j} \gg j \right).$$

CF914F：查询区间与串尾都要截断

第 i 位同时保留，当且仅当每个 $S_{i+j}=T_j$。最后只统计 $[l,r-m+1]$ 内的置位数；若 $m>r-l+1$ 直接返回零。匹配可以重叠。

机器字宽为 w，一次查询为 $O\left(m\left\lceil\frac{n}{w}\right\rceil\right)$，所有查询为 $O\left(\left(\sum|T|\right)\left\lceil\frac{n}{w}\right\rceil+q\right)$，空间为 $O\left(26\left\lceil\frac{n}{w}\right\rceil\right)$ 个机器字。移位需处理跨机器字的拼接，末尾越过串长的位必须清零。

这道题的“块”是机器字。它不靠根号平衡，而是将一次标量检查变成 w 个并行检查，适合作为分块与位运算结合的例子。

CF840E：In a Trap

In a Trap 的查询给出祖先 u 与后代 v，求路径上 $a_x \oplus \mathrm{dist}(x, v)$ 的最大值。

$n \leq 50000$，且 $a_x \leq n$。

CF840E：In a Trap

In a Trap 的查询给出祖先 u 与后代 v，求路径上 $a_x \oplus \mathrm{dist}(x,v)$ 的最大值。$n \le 50000$，且 $a_x \le n$。

数值和距离都不超过 16 位。

异或不能按普通加法拆贡献，但按位拆分没有进位。取 $B = 2^8$，将距离写成 $hB + j$，$0 \le j < B$；同一段内高八位 h 不变，低八位 j 恰好是段内距离。

对每个点 x，预处理它及向上至多 $B-1$ 个祖先组成的一段。按 a_y 的高八位分组，维护

$$g_x(t) = \max_{y:\mathrm{high}(a_y)=t} \big(\mathrm{low}(a_y) \oplus \mathrm{dist}(x,y)\big).$$

CF840E：块内低位先合并，高位再匹配

对一个高位偏移 h，整段答案为

$$F_x(h) = \max_t(((t \oplus h) \times B) + g_x(t)).$$

比较时高位优先，先在所有非空高位组中找到使 $t \oplus h$ 最大的组，再读取它的 $g_x(t)$。在八层二进制 Trie 上贪心，每个 h 花八步。枚举全部 $h = 0, \ldots, 255$，得到这段的完整查询表。

CF840E：预处理与查询边界

为每个点保存向上 B 步的跳转点。查询从 v 开始，累计完整段数 h；只要剩余路径至少有 B 个点，就取 $F_x(h)$ 并跳过这一段。最后不足 B 个点，逐点计算与原始 v 的真实距离，不再套整段表。

预处理每点至多 B 个祖先及 B 次八位 Trie 查询，时间 $O(n \times B \log B)$、空间 $O(n \times B)$；每次查询为 $O(\frac{n}{B} + B)$。本题将 16 位拆成两个八位部分，表的规模与段长同阶。

不要将每段距离都从零开始异或：h 保存的高位偏移正是前面已经跳过的距离。路径两端都包含，处理完整段的条件应按点数而非边数判断。

CF1039D：You Are Given a Tree

CF1039D You Are Given a Tree 对每个 k，求最多选多少条互不交点、恰有 k 个顶点的路径。$n \leq 10^5$。

CF1039D：You Are Given a Tree

CF1039D You Are Given a Tree 对每个 k，求最多选多少条互不交点、恰有 k 个顶点的路径。$n \le 10^5$。

固定 k，可以树形贪心计算：每个子树只向父亲保留一条未使用的最长链；若当前顶点能与最大的两条子链组成长度至少为 k 的路径，就完成一条路径，并停止向上提供这条链。实现中要把当前顶点计入长度，$k = 1$ 单独处理或统一进边界。

CF1039D：能在子树完成，就不必留给祖先

这个贪心维护“已完成路径数最大；在此基础上，向父亲提供的链尽量长”。若两条最长链加当前点仍不足 k，任何经过当前点的路径都无法在这里完成，只需保留最长链。若足够长，就截取一条恰有 k 个点的路径。

为什么可以立即完成？经过当前点、向子树外延伸的路径至多有一条。若某个最优方案选择把这里的链留给祖先，可以用当前完成的路径替换那条跨出子树的路径：路径数不减，祖先部分反而被释放。已在更深处完成的路径都不受影响。归纳应用这一交换，就得到贪心最优性。

CF1039D：大参数只有少量不同答案

记答案为 $f(k)$，则

$$f(k+1) \le f(k), \quad f(k) \le \left\lfloor \frac{n}{k} \right\rfloor.$$

对 $k \le B$，逐个计算，花费 $O(n \times B)$。对 $k > B$，答案至多有 $\frac{n}{B}$ 个正值，可以利用单调性二分每个答案区间的边界；每次判定仍为 $O(n)$。

这样得到 $O\left(n \times B + \frac{n^2 \times \log(n)}{B}\right)$，理论平衡点约为 $\sqrt{n \log n}$。这里变少的是答案种类，参数本身仍有很多取值。

P5021：赛道修建（练习）

NOIP2018 赛道修建：给定一棵有正边权的树，选择 m 条两两不共用边的路径，使其中最短路径的长度尽可能大。

$n \leq 5 \times 10^4$，$1 \leq m \leq n-1$，边权不超过 10^4。路径允许共用顶点，但不能共用边。

P5021：赛道修建（练习）

NOIP2018 赛道修建：给定一棵有正边权的树，选择 m 条两两不共用边的路径，使其中最短路径的长度尽可能大。

$n \leq 5 \times 10^4$，$1 \leq m \leq n-1$，边权不超过 10^4。路径允许共用顶点，但不能共用边。

提示：二分最短长度，将问题改成能否完成至少 m 条路径。子树需要将剩余链配对，并保留合适的剩余链给父亲；边不交允许多条路径经过同一顶点，不能照搬点不交问题的两条最长链规则。

完整解答 • 继续下一题

P5021：判定状态与向上的剩余链

二分最短长度 X。在有根树中，每个子树返回已完成的路径数，以及在不减少这个数量时，能从子树根向上延伸的最长剩余链；空链长度为零。

在顶点 u 汇总所有子树答案。孩子 v 返回长度 d_v，连同边权组成 $s_v = d_v + w_{u,v}$。若 $s_v \geq X$，就独立完成一条路径；其余短链收集起来，两条长度之和至少为 X 时可以配成一条路径。

每条通向孩子的边只能用一次，因而这些链只能两两配对或单独向父亲延伸。点可以共用，所以这里允许在同一个顶点完成多条路径。

P5021：短链的最大配对数

将剩余的 d 条短链排序为 $s_1 \le \ldots \le s_d < X$。双指针从两端开始：若最短与最长之和达到 X，配对并同时移动；否则放弃最短链，只移动左指针。

若 $s_l + s_r < X$，最短链与任何链都不能配对，丢弃它不会减少最优配对数。

若 $s_l + s_r \ge X$，可以要求某个最优方案包含这一对：若二者原来分别配对，让原先与 s_l 配对的链改配另一位伙伴，长度只会增加；若至多一者已配对，替换这一对也不减少数量。

因此两端扫描得到最大配对数 c。但最后留下的链未必最长，不能直接把扫描中最后丢弃的一条交给父亲。

P5021：配对数不变时保留最长链

若 $d = 2c$，所有短链都需要参加配对，向上返回零。否则，尝试保留某条 s_i：跳过它，对其余短链再次双指针，检查能否仍组成 c 对。

可行性关于 i 单调。若能保留较大的 s_j，改为保留更小的 s_i 时，可把配对中的 s_i 替换成 s_j，配对仍然合法。因此可行下标构成前缀。

二分最大的可行下标，返回对应链长。一次检查 $O(d)$，连同排序为 $O(d \log(d+1))$。当 $c = 0$ 时，所有链都可保留，返回最长链即可。

P5021：局部选择为何不损失全局最优

连接父亲的边只有一条，任何合法方案至多让一条链离开当前子树。已经达到 X 的长链立即完成，至多替换原方案中使用它的一条外部路径，路径总数不会减少。

对短链先最大化配对数。若保留一条链会少完成一条路径，它通过父边至多再带来一条，无法得到更多路径；在配对数相同时，保留更长链只会让祖先更容易完成路径。

从叶子向上应用这一交换，便能将最优方案转成上述贪心形式，而不减少路径数。

P5021：根处判定与二分范围

根处不再延伸剩余链，完成数至少为 m 就可行。最优判定若完成了更多路径，任取其中 m 条即可。

可行性关于 X 单调：降低要求不会让原来合法的路径失效。二分范围可取 $\left[0, \left\lfloor \frac{S}{m} \right\rfloor\right]$，$S$ 为总边权；边不交保证 m 条路径的总长不超过 S。

各点孩子数总和为 $n-1$，一次判定 $O(n \log n)$，二分总时间 $O(n \log n \log(S+1))$，空间 $O(n)$。边权累加使用 64 位整数，深树可按逆 DFS 序处理。

返回练习题面

CodeChef COUNTARI

CodeChef COUNTARI 统计 $i < j < k$ 且 $a_i + a_k = 2a_j$ 的三元组。

$3 \le n \le 10^5$，$1 \le a_i \le 3 \times 10^4$。

CodeChef COUNTARI

CodeChef COUNTARI 统计 $i < j < k$ 且 $a_i + a_k = 2a_j$ 的三元组。

$3 \le n \le 10^5$，$1 \le a_i \le 3 \times 10^4$。

设值域大小为 V，按位置分块。

固定中间点 j 所在块。若 i 在之前的块、k 在之后的块，对两侧的值域频次做卷积，下标 $2a_j$ 的系数就是该 j 的贡献。

需要为每个块维护这两份频次，而不是每换一个块就重扫整个数组。

COUNTARI：左右频次随块移动

按块从左到右处理。右表最初统计全数组，左表为空。处理一个块时：

1. 先从右表删去本块，使两张表只统计当前块之前、之后的位置。
2. 对左右频次做卷积，读取本块各中间点 j 对应的 $2a_j$ 系数。
3. 处理完后，将本块加入左表，再移到下一块。

每个位置只从右表删除一次、加入左表一次，频次维护共 $O(n)$。每块做一次卷积，这一类的总时间是

$$O\left(n + \left(\frac{n}{B}\right) V \log V\right).$$

剩下的三元组至少有两个相邻下标在同一块，再按下一页的两类分别计数。

COUNTARI：剩下两类局部枚举互不重复

1. j,k 同块时，枚举这两个位置，用 j 之前的频次数组统计 i。
2. i,j 同块而 k 不在该块时，枚举 i,j，用块后的频次数组统计 k。

这两类互不重叠，各花费 $O(n\times B)$。总时间为

$$O\left(n\times B+\frac{n\times V\times\log(V)}{B}\right),$$

取 $B\approx\sqrt{V\log V}$。卷积系数可能超过单个 NTT 模数，若要求精确计数，应使用足够的模数组合或经过误差分析的 FFT；答案本身也要使用足够宽的整数类型。

Ucup Shenzhen G：近似匹配

在 Ucup Shenzhen G 中，给定 N 个长度为 M 的串。每次询问给出一个同长串，求已有串中与它的 Hamming 距离不超过 K 的个数。$N, Q \leq 300$、$M \leq 6 \times 10^5$、$K \leq 10$。

Ucup Shenzhen G：近似匹配

在 Ucup Shenzhen G 中，给定 N 个长度为 M 的串。每次询问给出一个同长串，求已有串中与它的 Hamming 距离不超过 K 的个数。$N, Q \le 300$、$M \le 6 \times 10^5$、$K \le 10$。

先考虑比较一对字符串，只关心 Hamming 距离是否不超过 K。按长度 B 分块，先比较每块哈希，只在哈希不同时逐字符检查，并在差异数超过 K 后立即停止。

每个需要进入的块至少提供一个不匹配位置，因此最多细查 $K+1$ 个块，时间为

$$O\left(\frac{M}{B} + (K+1)B\right).$$

Ucup Shenzhen G：近似匹配的总代价

每次询问逐个检查全部 N 个串，统计通过的个数；总时间为

$$O\left(NQ\left(\frac{M}{B}+(K+1)B\right)\right),$$

另加输入与哈希预处理。取 $B\approx\sqrt{\frac{M}{K+1}}$。其中 $K+1$ 包含最后导致超限的一块，因此公式也适用于 $K=0$。

哈希碰撞的误差分析见 随机化选讲。

P7738：量子通信

NOI2021 量子通信：字典中有 n 个由题目生成器产生的 256 位零一串。每次在线给出一个同长串和阈值 $k \leq 15$，判断字典中是否存在与它至多 k 位不同的串。

$n \leq 4 \times 10^5$，$q \leq 1.2 \times 10^5$；询问串需要按上一次答案解密。

P7738：按相同小块筛选候选

两个 256 位串若至多有 15 位不同，将它们切成 16 个 16 位块，至少有一个对应块完全相同。

为每个“块编号、块内容”建立候选列表，查询时取这 16 个列表的并集，再用异或与 popcount 精确验证。鸽巢原理保证不会漏掉真解，精确验证也不会接受假解，所以答案确定正确。

在字典小块均匀、查询串固定且独立于字典的模型下，每个“字典串、块位置”匹配的概率为 2^{-16}。对 $16n$ 项用期望线性性，候选出现次数的期望为 $16\frac{n}{2^{16}}$；去重只会更少。

查询若依赖字典或过去回答，不能直接沿用这个期望。该模型只解释随机数据上的工作量；答案始终精确，最坏仍可能检查全部 n 个候选。

本组目录 · 对应主讲 · 主讲小结 · 拓展目录

拓展：立方根与抽样

Vines 在位置分块之上使用摊还分析；杭州 I 还需要条件概率与失败概率估计。两题分别展示三项代价和抽样后剩余区间，不作为基础分块的前置。

Vines　　ICPC 杭州 I

返回对应主讲 · 主讲小结 · 拓展目录

Vines

CerealCodes III Advanced I 有 0 到 n 层，地面为 n。在某层新建长度一的藤蔓会替换旧藤蔓；给定固定模数 C，可以让所有下标模 C 等于 k 的现存藤蔓长一；查询从某层不断沿藤蔓向下，最终停在哪一层。

Vines

CerealCodes III Advanced I 有 0 到 n 层，地面为 n。在某层新建长度一的藤蔓会替换旧藤蔓；给定固定模数 C，可以让所有下标模 C 等于 k 的现存藤蔓长一；查询从某层不断沿藤蔓向下，最终停在哪一层。

按位置分成长 B 的块。维护 last_i，表示从 i 出发，在离开当前块之前最后到达的位置；若它没有藤蔓，或下一跳已经出块，则 $\mathrm{last}_i = i$。从右往左扫描一个块，若下一跳 j 仍在块内，就令 $\mathrm{last}_i = \mathrm{last}_j$，一次重建为 $O(B)$。

查询先跳到 last_i，再执行它的一次出块跳跃。每轮要么停止，要么进入更靠后的块，总共 $O(\frac{n}{B})$ 轮。地面必须设为没有藤蔓的终止点，越界跳跃截到 n。

Vines：增长标记与活动位置

为每个余数维护全局增长标记 tag_k。藤蔓真实长度写成 $\mathrm{base}_i + \mathrm{tag}_{i \bmod C}$。新建时设置 $\mathrm{base}_i = 1 - \mathrm{tag}_{i \bmod C}$，所以它不会继承新建之前的增长。没有藤蔓的位置用独立标记判断，不能仅看长度表达式。

再为每个余数保存仍然“下一跳可能留在本块”的活动位置集合。增长操作只令对应 tag 加一，然后遍历其活动位置，重建相关块；若某位置下一跳已出块，将它从活动集合删除。

已出块藤蔓以后继续增长，也不会改变块内的最后停留位置，因此不再触发重建。但查询执行出块一步时，仍然按最新标记计算真实长度。

Vines：把增长成本计到新建操作

一次新建将长度重置为一。此后它至多经历 B 次影响块内结构的增长，就一定跳出块。每次触发重建花 $O(B)$，所以所有增长引起的成本都能计到最近一次新建，总计每次新建至多 $O(B^2)$。

活动集合的每次访问都对应这一次有效增长或最后的移除，枚举成本也被包含。新建本身另有 $O(B)$ 重建；查询为 $O(\frac{n}{B})$。故总时间为

$$O\left(n + q\left(B^2 + \frac{n}{B}\right)\right),$$

取 $B \approx n^{\frac{1}{3}}$。集合若用平衡树，还有插入删除的 $O(\log n)$，不改变上述主项。空间为 $O(n + C)$。

ICPC 2022 杭州 I

ICPC 2022 杭州 I 的未知有向环有 $n \leq U = 10^9$ 个顶点，标签恰为 1 到 n 的某种排列。每次可以向前走指定步数并获知新标签，要求不超过 10^4 次询问求出 n。

ICPC 2022 杭州 I

ICPC 2022 杭州 I 的未知有向环有 $n \le U = 10^9$ 个顶点，标签恰为 1 到 n 的某种排列。每次可以向前走指定步数并获知新标签，要求不超过 10^4 次询问求出 n。

标签本身给出下界：看到的最大标签 M 一定满足 $M \le n$。先做 s 次随机跳跃，每次步长独立均匀选自 1 到 U，记录最大标签。

杭州 I：给抽样后剩余区间一个概率界

虽然 U 未必被 n 整除，但每个环上顶点在一次跳跃后被命中的条件概率至少为 $\frac{\lfloor \frac{U}{n} \rfloor}{U} \geq \frac{1}{2n}$。所以漏掉最大 d 个标签的概率至多

$$\left(1-\frac{d}{2n}\right)^{s} \leq \exp\left(-s\frac{d}{2n}\right).$$

这给出 $n-M$ 的尾界，而不仅是最大值的期望。

杭州 I：记录连续小步，再跨越长区间

把采样结束时的位置当作新原点。连续走一步共 B 次，保存走过的标签与相对步数 $0, 1, ..., B$。若这时已经出现重复，第一次重复的步数差就是 n，直接结束。

否则 $n > B$。当前相对步数为 B，再走 M 步，到达相对步数 $M + B$；之后每次再走 B 步。每次检查标签是否在小步表中，若它对应小步位置 r，则

$$n = t - r,$$

其中 t 是当前大步的累计相对步数。

杭州 I：第一次相遇恰好绕环一周

为什么第一次相遇得到的是一个环长而非若干倍？大步从 $M+B \le n+B < 2n$ 开始；长度为 B 的步长和覆盖 $[0,B]$ 的小步表保证一旦跨过 n 就会相遇，不能跳过整段相遇窗口。因此第一次正差恰好为 n。

大步的累计位置依次为 $M+B, M+2B, \ldots$，因此大步次数恰为

$$\max\left(1, \left\lceil \frac{n-M}{B} \right\rceil\right).$$

第一次走 M 步也包含在这个计数中。若 $n \le B$，则已在连续小步阶段提前结束。

杭州 I：抽样次数与块长的平衡

记 $L=\ln\left(\frac{1}{\varepsilon}\right)$，取 $d=\left\lceil 2U\frac{L}{s}\right\rceil$。由前页尾界，$\Pr[n-M>d]\le\varepsilon$。在 $n-M\le d$ 时，总询问数为

$$O\left(s+B+\frac{UL}{sB}\right).$$

固定乘积 sB 时，$s+B$ 在 $s=B$ 处最小。令 $s=B=x$，就要平衡 x 与 $U\frac{L}{x^2}$，即 $x^3\approx UL$。因此

$$s\approx B\approx\left(U\ln\left(\frac{1}{\varepsilon}\right)\right)^{\frac{1}{3}}.$$

这说明立方根来自三项代价的共同平衡；实际还要把参数放回整数询问预算。

杭州 I：一万次询问的预算

预留一次初始读位置的询问，取 $s = B = 3000$，则还可以发出

$$G = 10000 - 1 - s - B = 3999$$

次大步，包括第一次走 M 步。当 $n - M \le GB = 11997000$ 时，一定在预算内完成。由 $n \le 10^9$，超出这段宽度的概率至多

$$\exp\left(-\frac{3000 \times 11997000}{2 \times 10^9}\right) < 1.54 \times 10^{-8}.$$

每次发问前检查总计数；已用满预算且仍需询问时才停止。每次交互后刷新输出。

本组目录 · 对应主讲 · 主讲小结 · 拓展目录