# 2024 CCPC 女生赛部分题目讲评

isaunoya

Hangzhou Dianzi University

# 目录

D. 优秀的拆分

E. 重心树

I. 字符串复制

J. 最大公因数的平方和

附录

# 目录

D. 优秀的拆分

E. 重心树

I. 字符串复制

J. 最大公因数的平方和

附录

# D：优秀的拆分

正式题目 D。给定长度为 $n$ 的排列，保持相对顺序，将所有元素分到两个子序列 $A, B$，每个元素恰好属于一组。

最大化 $\mathrm{LIS}(A) + \mathrm{LDS}(B)$；空序列的长度为零。

输出最大值。多组数据总长度满足 $\sum n \leq 2 \times 10^5$。

# D：优秀的拆分

正式题目 D。给定长度为 $n$ 的排列，保持相对顺序，将所有元素分到两个子序列 $A, B$，每个元素恰好属于一组。

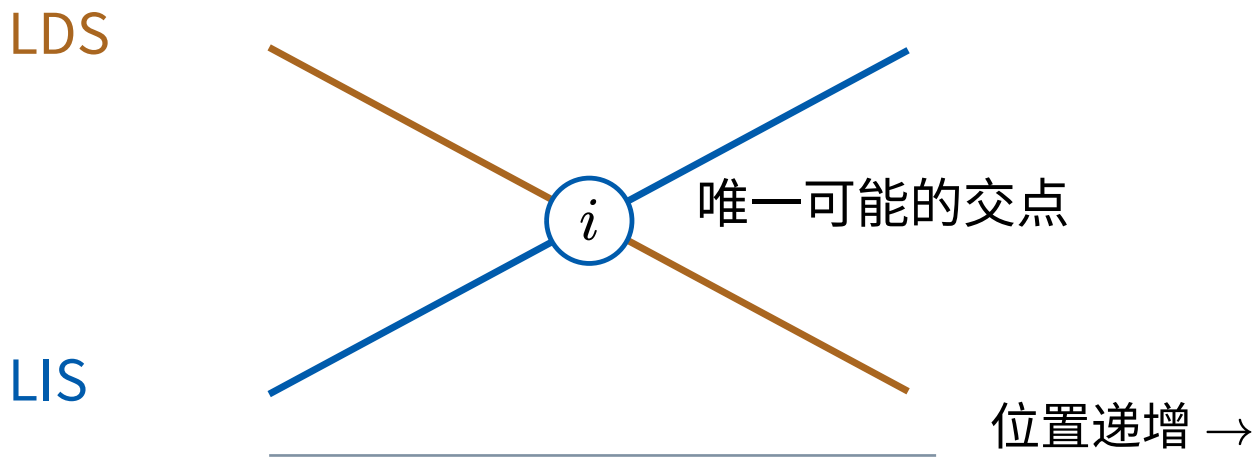
最大化 $\mathrm{LIS}(A) + \mathrm{LDS}(B)$；空序列的长度为零。

输出最大值。多组数据总长度满足 $\sum n \leq 2 \times 10^5$。

先分别求原排列的最长上升、下降子序列长度 $L, D$。答案至多为 $L + D$；距离这个上界，最多会差多少？

# D：两条序列至多相交一次

若上升、下降序列同时经过 $i < j$，就会同时要求 $a_i < a_j$ 与 $a_i > a_j$，矛盾。




任选一对 LIS、LDS；若相交，删去交点即可分组，未选元素任意分配。因此答案只有 $L + D$ 或 $L + D - 1$；达到前者，等价于存在不相交的一对。

# D：按唯一交点统计配对

设 LIS、LDS 总数为 $C, E$，经过位置 $i$ 的方案数为 $h_i, H_i$。

所有配对共 $CE$ 对；相交配对按唯一交点分类，共 $\sum_i h_i H_i$ 对。因此不相交配对数为

$$Z = CE - \sum_i h_i H_i.$$

整数意义下，$Z > 0$ 时答案为 $L + D$，否则减一。

这个等式依赖“至多一个交点”；一般两类子序列不能直接这样扣除。

# D：四遍扫描统计经过每点的方案

对上升子序列，从左、右各扫一遍，得到以 $i$ 结尾、开始的最优长度 $l_i, r_i$ 及方案数 $c_i, e_i$。线段树同时维护长度与达到该长度的计数。

只有 $l_i + r_i - 1 = L$ 时，$i$ 才在某条 LIS 上；此时前后两段可独立选择，且只在 $i$ 相交，所以

$$h_i = c_i e_i.$$

其余位置置零。将值换成 $n + 1 - a_i$，再做两遍相同扫描，得到 LDS 的 $D, E, H_i$。

四遍扫描为 $O(n \log n)$ 时间、$O(n)$ 空间。还需解决一个问题：计数可能很大，而固定模数下的零值不能证明不存在合法配对。

# D：给配对加权，保留判零能力

若所有计数只对固定素数取模，$Z$ 可能是该素数的非零倍数。可以保持四遍扫描的结构，改为独立随机加权。

为每个位置取两个独立变量 $x_i, y_i$。上升序列 $I$ 的权为 $\prod_{i\in I} x_i$，下降序列 $J$ 的权为 $\prod_{j\in J} y_j$。

把 $C, E, h_i, H_i$ 全部改成对应的权值和，则

$$Z(x,y) = \sum_{I,J:I\cap J=\varnothing} \left(\prod_{i\in I} x_i\right)\left(\prod_{j\in J} y_j\right).$$

两类序列使用不同变量族，每个配对对应不同单项式，系数均为一。有合法配对时，这在任意素域上都是非零多项式。

# D：判零保证与复杂度

将两个变量族独立均匀取自素域 $\mathbb{F}_p$，其中 $p = 2^{61} - 1$。合法配对存在时，$Z(x, y)$ 是次数不超过 $L + D \le n + 1$ 的非零多项式，随机落在零点上的概率至多 $\frac{n+1}{p}$。

独立做两轮四遍加权扫描。任何一轮 $Z \neq 0$ 就输出 $L + D$，两轮均为零才减一，单组误判概率至多

$$\left(\frac{n+1}{p}\right)^2.$$

时间仍为 $O(n \log n)$，空间 $O(n)$。无合法配对时恒为零，因此随机性只可能漏掉一个合法配对；多组测试的错误率相加。

判零引理的证明 · 加权扫描与零权值的处理

# 目录

D. 优秀的拆分

E. 重心树

I. 字符串复制

J. 最大公因数的平方和

附录

# E：重心树

正式题目 E。有根树以 $1$ 为根，父亲编号严格小于儿子编号。对每个点 $u$，给出其每棵儿子子树的一个重心，但不告诉对应儿子是谁。

重心是删除后最大连通块最小的点；若有两个，给深度较深者。保证有解，要求输出任意一棵符合所有信息的树。多组数据，$\sum n \leq 2 \times 10^5$。

# E：重心树

正式题目 E。有根树以 $1$ 为根，父亲编号严格小于儿子编号。对每个点 $u$，给出其每棵儿子子树的一个重心，但不告诉对应儿子是谁。

重心是删除后最大连通块最小的点；若有两个，给深度较深者。保证有解，要求输出任意一棵符合所有信息的树。多组数据，$\sum n \leq 2 \times 10^5$。

给出的重心一定属于对应儿子子树。能否只用这条归属信息，找到整棵子树的根？

# E：编号给出了处理顺序

按 $n, n-1, ..., 1$ 处理。处理 $u$ 时，所有后代编号都更大，因此每棵儿子子树已经处理完毕。

暂时不连接尚未处理的点。一棵已完成的儿子子树就是一个连通块；输入给出的重心 $c$ 位于这个块内。

沿父子边编号递增，所以块内最小编号就是该子树的根，也就是 $u$ 的儿子。

这里只使用重心的“位于子树内”这一性质。给出子树内任意一个代表点，同样能恢复连接关系。

# E：并查集维护两种不同的根

每个并查集连通块保存最小编号 $\mathrm{root}$。并查集代表元服务于合并，$\mathrm{root}$ 才是恢复树时需要的子树根。

对 $u$ 给出的每个代表点 $c$：

- 找到 $c$ 的并查集代表元，读出该块的 $\mathrm{root} = v$。
- 输出树边 $(u, v)$。
- 合并 $u$ 与该块；合并后最小编号为两块最小编号的较小者。

采用按大小合并与路径压缩，不依赖代表元恰好等于最小编号。每条边对应一次查询和合并。

# E：为什么恢复的一定是合法树

固定一棵与输入相容的原树，作从大编号到小编号的归纳。

叶子没有儿子，单点块就是它的完整子树。处理非叶子 $u$ 时，归纳假设保证其每棵儿子子树均已恢复，并且彼此仍是不同的块。

每个代表点都落在正确的块里，块的最小编号就是正确儿子。逐个连边后，恰好恢复 $u$ 的整棵子树。

因此算法恢复的边与这棵相容树一致；重心及“双重心取深者”的要求自动满足，不必另算一次重心。

# E：实现边界与复杂度

读入每个点的代表点列表后再倒序处理。叶子的列表为空；$n = 1$ 时不输出边。

同一节点的代表点分别来自不同儿子子树。处理一个代表点后合并进 $u$，不会影响其他尚未处理儿子块的根。

所有列表长度之和为 $n - 1$。按大小合并与路径压缩给出 $O(n\alpha(n))$ 时间，列表、并查集和输出边共 $O(n)$ 空间。

每次合并前读出儿子块的根并记录树边；合并后该块的最小编号已经变为 $u$，不能再用它恢复刚才的儿子。

# 目录

D. 优秀的拆分

E. 重心树

I. 字符串复制

J. 最大公因数的平方和

附录

# I：字符串复制

正式题目 I。给定长度 $1 \le n \le 3 \times 10^5$ 的小写字母串 $S$，将它重复拼接 $1 \le m \le 10^9$ 次。

求结果中本质不同的非空子串数，模 $998244353$。不同出现位置但内容相同的子串只算一次。

# I：字符串复制

正式题目 I。给定长度 $1 \le n \le 3 \times 10^5$ 的小写字母串 $S$，将它重复拼接 $1 \le m \le 10^9$ 次。

求结果中本质不同的非空子串数，模 $998244353$。不同出现位置但内容相同的子串只算一次。

记答案为 $F(m)$。不能显式生成全部 $S^m$；需要说明继续拼接时，哪些长度的子串还会增加。

# I：短子串在两份中已经齐全

把 $S^m$ 看作无限周期串 $SSS...$ 的前缀。子串内容由起点模 $n$ 的余数和长度决定。

对任意 $\ell < n$，串 $SS$ 已允许每种起点余数都出现：取起点 $0, ..., n-1$，终点仍在 $SS$ 内。

因此 $m \geq 2$ 后，所有长度小于 $n$ 的子串种类固定。新增内容只能来自长度至少为 $n$ 的部分。

接下来需要判断：长子串的不同起点，何时会表示同一种内容？

# I：最短整重复节

设 $S = T^{\frac{n}{p}}$，其中 $p = |T|$ 最小，且 $p$ 整除 $n$。这是整重复节，不是任意有限串的最短周期。

长度为 $n$ 的循环窗口，若从两个起点出发内容相同，则对应的两个循环移位相同。

若移位差为 $d$，整个长度为 $n$ 的循环在平移 $d$ 下不变，也在平移 $\gcd(n, d)$ 下不变。于是它的最短循环周期 $p$ 必须整除 $d$。

反过来，起点模 $p$ 相同，任意长度的内容都相同。因此长度至少为 $n$ 时，恰好按起点模 $p$ 区分内容。

# I：每种长度的子串数

在 $S^m$ 中，长度 $\ell \geq n$ 的子串有 $mn - \ell + 1$ 个连续可选起点。

连续起点能覆盖的模 $p$ 余数个数为

$$\min(p, mn - \ell + 1).$$

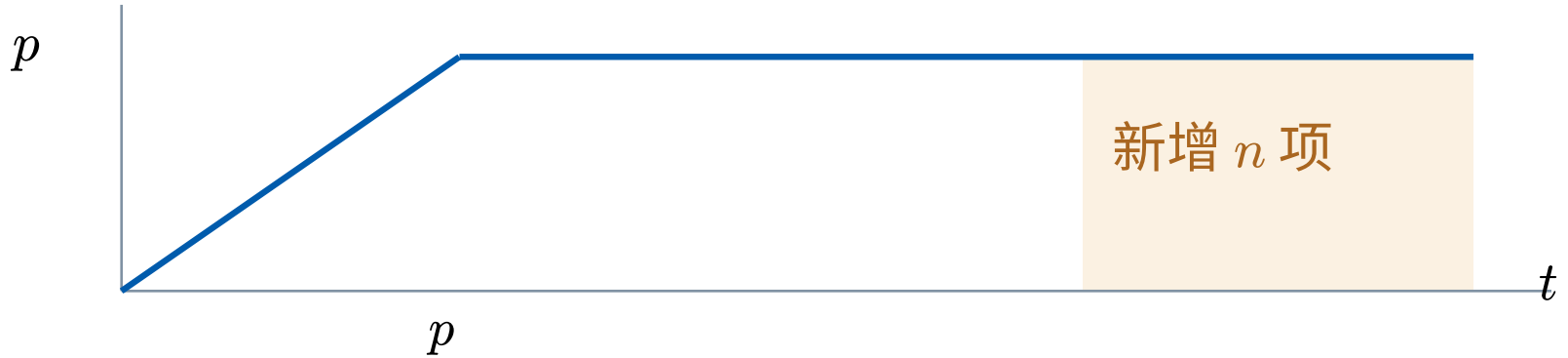
设短子串总数为常量 $C$，则 $m \geq 2$ 时

$$F(m) = C + \sum_{t=1}^{(m-1)n+1} \min(p, t).$$

这个表达式同时计入“起点足够多时饱和为 $p$”和末尾较长子串的不足，不需要逐个长度构造内容。

# I：稳定增量由求和直接得到

从 $m$ 到 $m+1$，求和上界增加 $n$。当 $m \geq 2$，新项的下标 $t > p$，因此它们全部落在 $\min(p, t)$ 的饱和段。




$$F(m+1) - F(m) = np, \quad m \geq 2.$$

故 $F(m) = F(2) + (m-2)(F(3) - F(2))$。按原做法统计 $SS$ 与 $SSS$ 即可；周期 $p$ 用于证明，不一定需要另算。

# I：后缀自动机为什么能计数

后缀自动机将子串按相同的结束位置集合归类。每个状态 $v$ 对应一个长度区间：

$$\text{len}(\text{link}(v)) + 1, \ldots, \text{len}(v).$$

同一状态中的字符串都是最长代表串的后缀，每种长度恰好一个；不同状态不会表示同一个子串。

因此不同非空子串总数为

$$\sum_{v \neq \text{ root}} (\text{len}(v) - \text{len}(\text{link}(v))).$$

根表示空串，不加入答案。

选讲：自动机扩展与克隆

# I：计数、空间与可选简化

每次扩展至多增加两个状态。固定 26 个字母的转移数组下，构造长度 $k$ 的自动机用 $O(k)$ 时间、$O(k)$ 空间。

$m = 1$ 时只处理 $S$。否则依原实现分别统计 $SS$ 与 $SSS$，或在同一个自动机到达 $2n, 3n$ 时扫描状态求和。

也可以只统计 $SS$，用前缀函数求 $p$：令 $q = n - \pi_{n-1}$，若 $q \mid n$ 则 $p = q$，否则 $p = n$，直接使用增量 $np$。

子串总数先用 64 位整数。模意义下计算外推式，乘法的两个因子先取模；状态数组按实际处理长度预留。

# 目录

D. 优秀的拆分

E. 重心树

I. 字符串复制

J. 最大公因数的平方和

附录

# J：最大公因数的平方和

正式题目 J。给定两个 $1$ 到 $n$ 的排列 $a, b$，每次给出两个闭区间 $[l, r]$ 与 $[L, R]$，求

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

$n, q \le 10^5$。询问彼此独立，可以离线。

# J：最大公因数的平方和

正式题目 J。给定两个 1 到 $n$ 的排列 $a, b$，每次给出两个闭区间 $[l, r]$ 与 $[L, R]$，求

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

$n, q \le 10^5$。询问彼此独立，可以离线。

区间产生一个二维矩形；gcd 同时取决于两个值。先把每对数的贡献改写成它们共有因子的贡献。

# J：构造需要的约数权值

定义函数 $w$，使对每个正整数 $x$ 都有

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

从小到大求解即可：先令 $w(x) = x^2$，确定 $w(d)$ 后，从所有真倍数的当前值中减去 $w(d)$。

处理到 $x$ 时，它的所有真因子均已处理，因此余下的正是所需 $w(x)$。总次数为 $O(n \log n)$。

这个函数叫二阶 Jordan 函数 $J_2$。此处直接按约数和定义筛出即可，不需要预先掌握它的乘积表达式。

# J：从 gcd 变成可分离的计数

由于 $d \mid \gcd(x, y)$ 等价于 $d$ 同时整除 $x, y$，有

$$\gcd(x, y)^2 = \sum_{d|x, d|y} w(d).$$

交换求和次序，令 $A_d, B_d$ 为两个查询区间内的 $d$ 的倍数个数，答案成为

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

每个因子的两个区间计数可以独立求出。难点只剩下：如何避免对每个询问都遍历全部因子？

# J：小因子使用前缀计数

设分界为 $K$。对每个 $d \le K$，分别扫描两个排列，计算 $d$ 的倍数的前缀计数。

将输入区间转为零基半开 $[l, r), [L, R)$，则

$$A_d = S_a(r) - S_a(l), \quad B_d = S_b(R) - S_b(L).$$

一次扫描后遍历所有询问，加上 $w(d)A_dB_d$。处理完当前 $d$ 即可复用前缀数组。

这部分时间 $O(K(n+q))$，额外空间 $O(n+q)$。无需为所有小因子同时保存前缀表。

# J：大因子只出现少量倍数

对 $d > K$，每个排列中恰有 $\left\lfloor \frac{n}{d} \right\rfloor$ 个倍数。每对倍数 $x, y$ 对应一个带权点

$$(\mathrm{pos}_a(x), \mathrm{pos}_b(y)), \quad \mathrm{weight} = w(d).$$

该点进入查询矩形时贡献一次。不同因子生成相同坐标时，权值仍要累加。所有点的总数满足

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

**排列保证每个值只有一个位置**，这是点数上界的关键。

# J：矩形拆成四个前缀事件

记 $P(x,y)$ 为第一坐标小于 $x$、第二坐标小于 $y$ 的点权和。则矩形答案为

$$P(r,R)-P(l,R)-P(r,L)+P(l,L).$$

将四个事件挂在第一坐标 $x$ 的桶中。扫描完前 $x$ 个 $a$ 的位置后，处理该桶中的第二坐标前缀询问。

初始先处理 $x=0$ 的事件，值均为零。随后插入位置 $x-1$ 产生的点，再处理 $x$ 桶。

这样不会混淆闭区间输入、半开存储和事件的插入时机。

# J：逐行生成点，省去点表

扫描到 $a_i = x$ 时，枚举 $x$ 的所有大因子 $d > K$，再枚举 $d, 2d, ..., \left\lfloor \frac{n}{d} \right\rfloor d$。

每个倍数 $y$ 在第二维的位置为 $\mathrm{pos}_b(y)$，向该位置加 $w(d)$。这样恰好生成第一坐标为 $i$ 的全部点。

大因子列表可预先用筛法加入各个倍数。所有列表共 $O(n \log n)$ 个元素；点本身按需生成，不保存 $M$ 份记录。

要严格保持线性额外空间，可用最小质因子分解每个值，再临时生成其约数；总生成数仍为 $O(n \log n)$。

# J：把常数时间留给修改

第二维有 $M$ 次单点加，只有 $4q$ 次前缀查询。用块长 $B$ 的数组分块维护点值与块和：

- 单点加同时更新点值与块和，$O(1)$。
- 前缀查询先加完整块，再扫描最后一块，$O\left(\frac{n}{B}+B\right)$。

大因子部分时间为

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

若改用树状数组，修改会多出 $\log n$ 因子。这里正是修改远多于查询的场景。

# J：参数与总复杂度

合并筛法、小因子、大因子和前缀查询：

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

取 $B \approx \sqrt{n}$；平衡前两种因子处理，取 $K \approx \frac{n}{\sqrt{n+q}}$ 并限制到 $[1, n]$。

当 $n, q$ 同阶时，$K \approx \sqrt{n}$，总时间为 $O(n\sqrt{n})$。

存约数列表时空间 $O(n\log n + q)$；临时生成约数时空间 $O(n+q)$。两种实现对应不同内存代价，点表都不需要保存。模运算约定

# 目录

D. 优秀的拆分

E. 重心树

I. 字符串复制

J. 最大公因数的平方和

附录

# 补充证明与实现约定

- D：非零多项式的判零引理
- D：加权扫描不需要除法
- D：空状态与有限域运算
- I：后缀自动机扩展与克隆
- J：模运算约定

返回 D 的判零结论 · 返回 I 的稳定增量

# D：非零多项式的判零引理

非零多项式 $P$ 的总次数为 $d$。将每个变量独立均匀取自素域 $\mathbb{F}_p$，有

$$\Pr[P = 0] \le \frac{d}{p}.$$

一元情形来自“$d$ 次非零多项式至多有 $d$ 个根”。多元情形按最后一个变量展开：最高次系数不为零时，用一元根数界；该系数意外为零时，对系数递归使用同一结论。两部分次数相加不超过 $d$。

这里 $d = L + D \le n + 1$。无合法配对时恒为零；有合法配对时，只有随机取值碰巧落在零点上才会误判。

返回 D 的判零保证

# D：加权扫描不需要除法

前向扫描时，查询最优前驱的权值和 $c$，存入 $x_i c$；记结果为 $u_i$，其中已经含有 $x_i$。

后向扫描时，查询最优后继的权值和，记为 $v_i$；向树中存入 $x_i v_i$，但额外保留未乘 $x_i$ 的 $v_i$。

只有 $l_i + r_i - 1 = L$ 时置 $h_i = u_i v_i$，否则置零。交点权值恰好出现一次。空前驱或空后继的权为一；LDS 使用 $y_i$ 同理。

**长度决定可达性，权值和不决定可达性。** 权值和取模后等于零时，最长长度仍须保留。

返回 D 的判零保证

# D：空状态与有限域运算

树上状态为 $(\ell, c)$：保留较长长度，相等时累加权值和。数据结构的空节点为 $(0, 0)$；没有前驱时，另加入唯一空序列 $(0, 1)$。不能让每个空节点各贡献一份。

排列中每个值只出现一次，点赋值即可。长度决定可达性；模意义下权值和为零，也必须保留该长度。

全部变量独立均匀取自 $[0, p)$，可用拒绝采样避免直接取模的偏差。乘法先转为 `unsigned __int128`，加减规范到 $[0, p)$；随机引擎使用运行时种子。

返回 D 的判零保证

# I：在线扩展与克隆

加入字符 $c$ 时，新建状态 $\mathrm{cur}$，最长长度为旧 $\mathrm{last}$ 的长度加一。沿后缀链接补齐缺失的 $c$ 转移，指向 $\mathrm{cur}$。

若一直到根外，令 $\mathrm{link}(\mathrm{cur}) = \mathrm{root}$。否则遇到已有转移 $p \to q$：

- 若 $\mathrm{len}(p) + 1 = \mathrm{len}(q)$，令 $\mathrm{link}(\mathrm{cur}) = q$。
- 否则复制 $q$ 的转移与链接为克隆点，长度设为 $\mathrm{len}(p) + 1$。从 $p$ 沿后缀链接上行，将仍指向 $q$ 的 $c$ 转移改向克隆，直到转移不再指向 $q$。令 $q$ 与 $\mathrm{cur}$ 的链接均指向克隆。

克隆将一个长度区间分成两段，不产生虚假的子串。最后令 $\mathrm{last} = \mathrm{cur}$。

返回 I 的自动机计数

# J：模数与实现收尾

所有加减都在模 $2^{32}$ 的环中进行。用明确的 `uint32_t` 保存权值、块和与答案，无符号溢出正好完成取模。

两项相乘先转 `uint64_t`，再截取低 32 位；三项乘法每乘两项就截取一次，避免依赖有符号溢出。

平方初始化也先用无符号 64 位计算。前缀次数可用普通整数，减法得到的区间计数非负。

每个矩形的四个符号事件、小因子贡献和大因子贡献均进入同一答案数组；最后按输入顺序输出无符号数值。

返回 J 的复杂度