第九届中国大学生程序设计竞赛
桂林
题解

浙江大学

10.23.2023

给 n 张正反面有数字的牌，翻转不超过一个区间使中位数最大。

M. Flipping Cards

Solution

二分答案 w，那么就是要求 $[b_i \geq w] - [a_i \geq w]$ 的最大子段和。
时间复杂度 $\Theta(n \log a)$。

给一个只打左右括号和括号补全打出来的序列，求输入序列

注意到，如果答案存在的话，输出串一定是一个合法输入串。
可以通过模拟或者后缀和来判断无解。
时间复杂度 $\Theta(n)$。

给一个可重集，求有多少子集满足每个元素都可以被异或和整除。

设 $\text{lcm}(S) = f, \text{xor}(S) = g$。
注意到 $f \geq \max\{a\}, g < 2 \cdot \max\{a\}$，所以可能的情况只有
$g = 0, g = \max\{a\}$
$g = 0$ 的情况等价于询问一个集合有多少这子集异或和为 0，设这个集合异或线性基的秩为 r，则答案为 2^{n-r}。
$g = \max\{a\}$ 时，枚举 g，然后枚举所有 g 的因子加入线性基计算即可。
时间复杂度 $\Theta(n \log^2 n)$。

K. Randias permutation task

Description

给 m 个长度为 n 的排列，选出若干个（非零）按原顺序复合，问得到的排列有多少种。$n \cdot m \leq 180$

注意到答案不超过 $\min\{2^m, n!\} \leq 362880$，所以任何复杂度是 $O(ans)$ 级别的爆搜都是正确的。

具体来说，令 f_i 表示前 i 个排列能复合出的排列的集合，每次枚举下一个排列选不选即可。
时间复杂度 $\Theta(nm \cdot \min(2^m, n!) \cdot \log \min(2^m, n!))$。

给一个序列，选择一个区间使得区间颜色数 − 区间 mex 最大。

枚举 mex$= x$，之后我们只需要检查所有 x 不出现的极长区间即可。因为对于答案区间，假设 mex 为 y，那么我们枚举 mex 为 y 时选择的区间一定不劣于答案区间。

使用树状数组扫描线统计颜色数即可，时间复杂度 $\Theta(n \log n)$。

给你两个串集合，都有 n 个长度为 m 的串。问是否存在方案讲两个集合重新排列拼接，使得他们循环同构。

先枚举循环同构的偏移量 x，那么就是要求每个第一个集合中的串长度为 x 的后缀和第二个串中长度为 x 的前缀匹配，第二个集合中长度为 $n-x$ 的后缀和第一个集合中串长度为 $n-x$ 的前缀匹配。
将对应长度的前后缀看成点，字符串看成边，那么就是要求解欧拉回路。
时间复杂度 $\Theta(nm)$。

给定平面上 n 个整点，修建最少数量的地铁（整点折线）使得第 i 个站恰有 a_i 条地铁通过。地铁不能自交。两条地铁不能在地铁站外相交。

最少数量为 $\max(a_i)$ 。构造如下：
按 $(10001, 2)$ 向量的方向排序，在相邻两点之间的垂直于该向量的某条线上一定存在无限个点。每次取 a_i 最大的所有地铁站，在这些点之间连一条 W 形地铁线。重复这个过程即可。
时间复杂度 $\Theta(n \cdot \max a_i)$。

给定一棵无根树，每个点的点权为 0、1 或 2，从树上找到尽量多的不重叠的连通块，使得每个连通块的点权和都为 k。

任选点权和为 k 的连通块，其中 $k>2$，那么该连通块至少包含两个点权非零的叶子，否则可以在不改变 k 的前提下迭代删去点权为 0 的叶子。

- 如果至少有一个叶子的点权为 2，那么删去该叶子即可得到点权和为 $k-2$ 的连通块。
- 否则所有叶子的点权都为 1，那么同时删去两个叶子即可得到点权和为 $k-2$ 的连通块。

因此可以通过树形 DP 求出总和为奇数/偶数的点权和最大的连通块来 $O(1)$ 判断是否存在点权和为 k 的连通块。

任选一点为根，自底向上贪心。
假设当前考虑完了 x 的所有儿子，如果 x 子树目前与 x 相连的部分存在一个点权和为 k 的连通块，那么切掉 x 与 x 父亲之间的边不会使得答案变差，对答案的贡献为 1。
时间复杂度 $O(n)$。

B. The Game

Description

给一个集合 A，每次从中选择一个数 $+1$，然后删掉最小的，问能不能变成集合 B。$n, m \leq 3 \cdot 10^5$。

将数字排序，每次执行 $x = x + 1$ 的时候，我们都可以假设操作的是所有 x 中最靠右的一个。故可以认为所有数字的相对顺序不变。则必然是排序后 A 中最大的 m 个数字依次匹配 B 中的数字。

故若对于某个 i，$A_{n-(m-i)} > B_i$，或者 $\sum_{i=1}^{m} B_i - \sum_{i=n-m+1}^{n} A_i > n - m$，均不可行。

否则，称对某个最终会被删除的数进行的操作为冗余操作，我们需要想办法获得尽量多的冗余操作（显然冗余操作随时可以停止，只需要进行对最大的 m 个数操作即可），这样才能消耗掉多余的步数。

考虑每次操作最小的数字，这是最有可能产生冗余操作的情况。我们需要不断地操作最小的数字，直到 $\sum_{i=1}^{m} B_i - \sum_{i=n-m+1}^{n} A_i = n - m$ 为止。需要操作的步数不容易计算，但是可以直接模拟这个过程。

模拟过程使用对顶 set，或者线段树，或者别的数据结构维护均可。如果模拟过程中违反了 $A_{n-(m-i)} \leq B_i$ 的的限制，也为无解。

给一个麻将序列，问每个前缀是不是胡。

E. Prefix Mahjong

Solution

先考虑单个询问，只需要排序后动态规划，状态是考虑到 i 这个数，是否已经有雀头/$i-2$ 为起点有多少个顺子/$i-1$ 为起点有多少个顺子，每个位置为起点的顺子最多只要考虑 2 个，多余的可以变成刻子，因此一共 $2\times3\times3=18$ 个状态。
由于值域比较大，可以只考虑出现过的数和出现过的数 $+1$ 的这些数，因此值域可以压缩到 $2n$。
再考虑对于每个前缀，按值域建线段树进行动态动态规划，每个节点存动态规划转移的矩阵，这是一个关于且和或的 01 矩阵，可以压位优化成平方级别的乘法。复杂度为 $\Theta(18^2 n\log n)$。将状态数优化到 9 或者更小也可以通过。
看起来有点大，但其实 Python 程序可以在二分之一时限通过。

有两个面数为 n, m 的骰子，其面上数字都是一个排列，你需要找到两个不同骰子，使得这两个骰子面上的数字为正数，并且两对骰子和的概率分布相同。在此基础上，最小化你找到的骰子的面数和。

- 设新骰子面数为 s,t。为了方便起见，所有骰子点数从 0 开始。不妨设 $n \le m$。
- 对于 n 个面分别为 $a_1, a_2, \cdots, a_n$ 的骰子，其概率生成函数（PGF）为 $\frac{1}{n}\sum_{i=1}^{n} x^{a_i}$，那么两个骰子和的 PGF 是两个骰子 PGF 的乘积。
- 记 $f_n = \sum_{i=0}^{n-1} x^i = \frac{1-x^n}{1-x}$，那么初始的两个骰子和的 PGF 为 $g_{n,m} = \frac{f_n(x)f_m(x)}{f_n(1)f_m(1)}$。
- 设新骰子的 PGF 分别为 $\frac{1}{s}F_s, \frac{1}{t}F_t$，那么 $\frac{F_s(x)F_t(x)}{st} = g_{n,m} = \frac{f_n(x)f_m(x)}{f_n(1)f_m(1)}$，并要求 $F_i(1) = i$。

由 $f_n(0)=1$，对 $st=\dfrac{1}{f_n(x)f_m(x)}nmF_s(x)F_t(x)$ 代入 $x=0$ 有

$$\begin{cases} nm \mid st \\ \dfrac{st}{nm} \mid F_s(x)F_t(x) \end{cases}\text{。}$$

- 故对于任意一组 $st \neq nm$ 的解，总有一组 PGF 相等的解 $F_{s'}F_{t'}$ 满足 $s't'=nm$。

- 首先考虑 $st \neq nm$ 的情形。
- 如果不考虑“不和原有骰子相同”这个限制，$\{F_s, F_t\}$ 不优于 $\{F_{s'}, F_{t'}\}$，故只需要考虑 $\{F_{s'}(x), F_{t'}(x)\} = \{f_n, f_m\}$ 的情况。
- 这一情况也是必须被考虑的。例如对于 $n = 1, m = 1$ 没有 $st = nm$ 的解，对 n, m 都是质数时这一情况能给出比 $1 + nm$ 更好的解。
- 显然最优解为 $F_s = 2f_n$，$F_t = f_m$。复杂度 $O(m)$。

- 考虑 $st = nm, \{s, t\} \neq \{n, m\}$ 的情形。
- 有 $F_s(x)F_t(x) = f_n(x)f_m(x)$。因此问题本质是对 $f_n(x)f_m(x)$ 再分解，使 $s + t$ 最小。

引理 1：给定整系数多项式 $f(x) = \sum_{i=0}^{n} a_i x^i$，若存在质数 p 满足：

$$\begin{cases} p \nmid a_n \\ p \mid a_i & (0 \leq i < n) \\ p^2 \nmid a_0 \end{cases}$$

，那么 $f(x)$ 在 $\mathbb{Q}[x]$ 不可约。
证明：众所周知的 Eisenstein 判别法，具体证明不赘述。

- 引理 2：设 n 的唯一分解为 $n = \prod_i p_i^{k_i}$，那么 $f_n(x)$ 的在 $\mathbb{Z}[x]$ 的分解中恰好含有 k_i 个系数和为 p_i 的不可约多项式，且两两不同。

证明：记 $H_{j,p}(x) = \sum_{l=0}^{p-1} x^{lp^j}$，那么对全体 $j < k_i$ 有 $H_{j,p_i} \mid f_n$，且显然系数和为 p_i。

我们证明 $H_{j,p}$ 在 $\mathbb{Z}[x]$ 上不可约：考虑

$$H_{j,p}(x+1) = \sum_{l=0}^{p-1} (x+1)^{lp^j} = \sum_{l=0}^{p-1} \sum_{k=0}^{lp^j} \binom{lp^j}{k} x^k = \sum_{k=0}^{(p-1)p^j} x^k \sum_{l=0}^{p-1} \binom{lp^j}{k}。$$

我们尝试使用 Eisenstein 判别法证明 $H_{j,p}(x+1)$ 在 $\mathbb{Q}[x]$ 上不可约，那么需要考察 $H_{j,p}(x+1) \bmod p$ 的系数：

对于其最高次项 $x^{(p-1)p^j}$，直接考察 $\sum_{l=0}^{p-1} (x+1)^{lp^j}$ 可知它系数为 1，满足条件；

对于其零次项，同样考察 $\sum_{l=0}^{p-1}(x+1)^{lp^j}$ 可知它系数为 p，满足条件；

由 Lucas 定理，对于 p 进制数 $\begin{cases} n = \overline{n_1 n_2 n_3 \cdots n_l} \\ m = \overline{m_1 m_2 m_3 \cdots m_l} \end{cases}$，$\binom{n}{m} \equiv \prod_{i=1}^{l} \binom{n_i}{m_i} \pmod p$。

考察 x^k 的系数，将 k 写成 p 进制数的形式 $k = \overline{k_{p-1} k_{p-2} k_{p-3} \cdots k_0}$，那么 $\sum_{l=0}^{p-1} \binom{lp^j}{k} \equiv \sum_{l=0}^{p-1} \binom{l}{k_j} \prod_{i \neq j} \binom{0}{k_i} \pmod p$。

考虑 $\sum_{l=0}^{p-1} \binom{l}{k_j}$ 的组合意义是从 l 个中选 k_j 个，这可以看作 $l+1$ 个选 k_j+1 个并枚举第一个被选的编号，故 $\sum_{l=0}^{p-1} \binom{lp^j}{k} \equiv \binom{p}{k_j+1} \prod_{i \neq j} \binom{0}{k_i} \pmod p$。

当存在 $i \neq j$ 满足 $k_i > 0$，显然 x^k 系数是 p 的倍数。否则，由于 x^k 不是最高次，故 $k < (p-1)p^j$，因此 $k_j < p-1$，那么 $p \mid \binom{p}{k_j+1}$，系数也是 p 的倍数。

综上，$H_{j,p}(x+1)$ 满足 Eisenstein 判别法，因此它在 $\mathbb{Q}[x]$ 上不可约，故 $H_{j,p}$ 在 $\mathbb{Q}[x]$ 不可约。由 $H_{j,p}$ 为首一多项式，$H_{j,p}$ 在 $\mathbb{Z}[x]$ 也不可约。由于 $H_{j,p}$ 在 $\mathbb{Z}[x]$ 不可约，它们的乘积也是 f_n 的因式。由于乘积的系数和等于系数和的乘积，而已经构造出系数和乘积为 n 的不可约多项式，故其余的不可约多项式的系数和均为 1，引理得证。

- 关于这种情况的构造：
- 注意到对于任意的 s, t $(st = nm)$，设 $p = \dfrac{n}{\gcd(s, n)}$，
  $q = \dfrac{ps}{n} = \dfrac{s}{\gcd(s, \frac{st}{m})} = \dfrac{sm}{\gcd(sm, st)} = \dfrac{m}{\gcd(m, t)}$，则 $p \mid n$，$q \mid m$。
- 有 $f_n(x)f_m(x) = (\sum_{i=0}^{p-1} x^i \sum_{j=0}^{n/p-1} x^{jp})(\sum_{k=0}^{q-1} x^k \sum_{l=0}^{m/q-1} x^{lq})$
- 显然有一个解是 $F_s(x) = \sum_{j=0}^{n/p-1} x^{jp} \sum_{k=0}^{q-1} x^k$，$F_t(x) = \sum_{i=0}^{p-1} x^i \sum_{l=0}^{m/q-1} x^{lq}$，可以自行验证 $F_i(1) = i$。

- 至此，对于 $\{s, t\} \neq \{n, m\}$ 的情况都构造出了解，只需要选其中 $|s - t|$ 最小的一组。
- 找到这组 s, t 的复杂度为 $O(m)$，计算对应的解可以直接枚举 j, k 和枚举 i, l，复杂度为 $O(\frac{nq}{p} + \frac{mp}{q}) = O(s + t)$，但当 $s + t \geq 2n + m$ 时不优于 $\{2f_n, f_m\}$ 这组解，不需要考虑，复杂度为 $O(m)$。

- 当所有解都满足 $|s-t| > m-n$ 时，需要考虑是否有 $\{s, t\} = \{n, m\}$ 的解。不妨令 $s = n, t = m$。
- 由于对 F_i 的要求为 $F_i(1) = i$，故相当于考虑交换 $f_n(x)$ 与 $f_m(x)$ 的因式，并且被交换的因式系数和相等。
- 一种容易想到的情况是 $\gcd(n, m) \neq 1$ 条件下的解（需要具体讨论），但这个解并不是唯一的解法。由于它在完整做法中几乎没有用处，这里不展开。
- 实际上，对解决这个问题更有帮助的，是找到如 $n = 6, m = 7$ 情况的解，并发现它的本质：

- 若 n 至少有两个不同的质因子 p,q，则 $\sum_{i=0}^{pq-1} x^i \mid \sum_{i=0}^{n-1} x^i$，而 $\sum_{i=0}^{pq-1} x^i = \sum_{i=0}^{p-1} x^i \sum_{j=0}^{q-1} x^j \Phi(x)$，其中 $\Phi_{pq}(x)$ 是一个 $\varphi(pq)$ 次多项式，满足 $\Phi_{pq}(1)=1$。我们考虑交换 Φ_{pq} 和 1，即形如 $\begin{cases} F_s = \dfrac{f_n}{\Phi_{pq}} \\ F_t = f_m\Phi_{pq} \end{cases}$ 的一组解，那么只需要证明这组解的系数是非负的。

$F_s = \sum_{k=0}^{n/(pq)-1} x^{kpq} \sum_{i=0}^{p-1} x^i \sum_{j=0}^{q-1} x^j$，系数显然非负。

F_t 则较为棘手。如果对分圆多项式有一定了解，一个结论是 $\Phi_{pq}(x)$ 的非零系数总是 $1,-1$ 交替出现。那么 $f_m\Phi_{pq}(x)$ 系数非负相当于 $\Phi_{pq}(x)$ 系数的每个前后缀和都是非负的，而这是显然的。当然，我们也可以不借助这个结论：

$\Phi_{pq}=\dfrac{(1-x^{pq})(1-x)}{(1-x^p)(1-x^q)}=\dfrac{\sum_{i=0}^{q-1}x^{ip}(x-1)}{x^q-1}$。手动模拟多项式除法可以得到 $x^n=x^{n \bmod q}+(x^q-1)\sum_{i=0}^{\lfloor n/q\rfloor-1}x^{iq+(n \bmod q)}$，故 $x^{ip}-x^{ip \bmod q}=(x^q-1)\sum_{j=0}^{\lfloor ip/q\rfloor-1}x^{(ip \bmod q)+jq}$。注意到 $ip \bmod q$ 在 i 取遍 $[0,q)$ 时恰好也取遍 $[0,q)$ 一次，故有 $\Phi_{pq}=\sum_{i=0}^{q-1}\sum_{j=0}^{\lfloor ip/q\rfloor-1}x^{(ip \bmod q)+jq}(x-1)+\sum_{i=0}^{q-1}x^i\dfrac{x-1}{x^q-1}$，其中 $\sum_{i=0}^{q-1}x^i\dfrac{x-1}{x^q-1}=1$。

设 $G(x) = \dfrac{\Phi_{pq}(x) - 1}{x - 1} = \sum\limits_{i=0}^{q-1} \sum\limits_{j=0}^{\lfloor ip/q \rfloor - 1} x^{(ip \bmod q)+jq}$，显然 $(ip \bmod q) + jq$ 两两不同，故 $G(x)$ 系数只有 $0, 1$。而 $F_t = f_m((x-1)G(x)+1) = f_m + (x^m - 1)G(x)$，由于 $\deg G < pq \le n \le m$，$(x^m - 1)G(x)$ 所有负系数均为 -1 且次数小于 m,，而 f_m 在次数小于 m 处系数均为 1，故加上 f_m 后系数均非负，故 F_t 系数非负。

- 因此 F_s, F_t 构成了一组解。
- 另一个问题是如何求出 Φ_{pq} 并计算 $f_m\Phi_{pq}$。由于

$$\Phi_{pq} = \frac{\sum\limits_{i=0}^{pq-1} x^i}{\sum\limits_{i=0}^{p-1} x^i \sum\limits_{i=0}^{q-1} x^i},$$

问题等价于对某个多项式 g 计算 $\dfrac{g}{f_k}$ 和 gf_k。

- 先考虑如何计算 gf_k，这相当于按照 g 的系数对结果多项式作 $\deg g$ 次长度为 k 的区间加法，可以差分维护。
- 计算除法只需要将这个过程逆序完成。复杂度 $O(m)$。

- 接下来考虑 n 为质数幂的情形。
- 设 $n = p^k$，同上令 $H_{j,p}(x) = \sum_{l=0}^{p-1} x^{lp^j}$，则 $\sum_{i=0}^{n-1} x^i = \prod_{j=0}^{k-1} \sum_{i=0}^{p-1} H_{j,p}(x)$。

- 先考虑被交换多项式系数和非 1 的情形。由于 $H_{j,p}(x)$ 在 $\mathbb{Z}[x]$ 不可约，故如果需交换出多项式系数和非 1 只能和同样系数和为 p 的交换，即 $p \mid m$。
- 若 $p^2 \nmid \operatorname{lcm}(n, m)$，那么 f_n, f_m 拥有的系数和为 p 的多项式为同一个（即 $H_{0,p}$），不能交换；
- 否则，f_n, f_m 至少拥有两种系数和为 p 的多项式（$H_{0,p}, H_{1,p}$），可以交换这两个多项式，构成了一组解。
- 复杂度 $O(m)$。

- 再考虑系数和为 1 的情形。首先讨论：m 有至少两个质因子。
- 一个令人惊讶的结果是：尽管 n 至少有两个质因子时总是有解，m 至少有两个质因子时却不一定有解。反例：$n=7, m=15$ 是无解的。
- 假设 m 的最小两个质因子为 p,q，一组可能的解为 $\begin{cases} F_s=f_n\Phi_{pq} \\ F_t=\dfrac{f_m}{\Phi_{pq}} \end{cases}$ 。
- 我们证明这也是最好的一组解：即，若这组解不成立，无其他解。
- 同 n 有两个质因子的证明可知，若 $n\geq pq$ 这一定是一组解。
- 否则，由于不存在 $n<s,t<m$ 的解，那么 $s=\dfrac{m}{p}$ 时一定满足 $s\leq n$，有 $n\geq\dfrac{m}{p}$。而 $n\geq pq$ 已经找到解了，故只需要考虑 $pq>n\geq\dfrac{m}{p}$，即 $m<p^2q$。由于 p,q 是最小的质因子，有 $m=pq$。可以证明 Φ_{pq} 在 $\mathbb{Z}[x]$ 不可约，而 $f_p=H_{0,p}$ 已经证明是不可约的，故 $f_m=f_pf_q\Phi_{pq}$ 是 f_m 的分解，没有其他系数和为 1 的多项式。

L. Alea Iacta Est

$\{s, t\} = \{n, m\}$，n 为质数幂，被交换多项式系数和为 1，m 为质数幂

- 当 m 为质数幂时，其所有分解的多项式系数和都不是 1，故无解。
- 以上是所有情况。总复杂度 $O(m)$。

一棵树，每次可以选择直径的一个端点删除，直到把树删空，问有多少种不同的删除序列。$n \le 300$。

所有直径一定过同一个中心点，这个中心点可能是点或者边的中点。定义半径为直径长度的一半。不妨假设现在的中心点为 u，半径为 r。则在原树中，只有 $dis(i, u) \leq r$ 的点 i 可能在这个状态被保留。任意一对不在 u 的同一个子树，且 $dis(i, u) = r$ 的点，都构成一条直径。下文中，我们将 $dis(i, u) = r$ 的点称为边界点，边界点构成的集合称为边界点集。不加说明的话，所有讨论都是基于某个时刻中心为 u，半径为 r。

可以证明，所有 $dis(i, u) < r$ 的点 i 必然被保留。只有边界点可能被删除了。

现在考虑中点从 u 转移到了 v 会发生什么：必然是 u 的除了 v 方向上的所有子树中，其中的边界点已经被删除，而 u 的 v 方向的子树上的边界点没有被完全删除。

我们将 (u, r) 的边界点称为原边界点，$(v, r-\frac{1}{2})$ 的边界点称为新边界点。原边界点和新边界点的交即为 u 的 v 方向子树的全体边界点。可以发现，除去两个边界点集的交集，剩下的点的情况都很好计算。原边界点集中全被删除，新边界点集中全部存在。而对于交集部分，它可以插入在抵达 (u, r) 之后的任意一个位置（但不能早于 $(v, r-\frac{1}{2})$）删完。

这启发我们一件事：不妨假设所有" 可能存在" 的点全部存在，直到这个点被删除的时候插入删除序列的合适的位置。

令 $f_{u,r,k,l}$ 表示当前中心为 u，半径为 r，上一个中心是子树 k 的方向（这是为了确定所谓的 "边界点集的交集" 是哪个集合），且对于这些" 边界点集的交集"，能够被插入的后缀长度为 l（插入到之前的删除序列中）。

转移有两种情况：

- 新的状态 $(v, r-\frac{1}{2})$，v 不在子树 k 的方向上。此时所有 u 的非 k 非 v 子树方向上的全部边界点都要删除，删除时插入长度为 l 的后缀，使用可重组合数计算方案即可。新的状态为 $f_{v,r-\frac{1}{2},u,l'}$，其中 l' 是 l 加上新插入的点的个数。
- 新的状态 $(v, r-\frac{1}{2})$，v 在子树 k 的方向上。此时上一步残留的边界点集全部被删除，插入到长度为 l 的后缀中。注意此时序列尾部至少要有一个元素（为了保证这颗子树是最晚被删空的），并且我们需要枚举序列尾部的元素个数 m，因为新的可插入长度就是这个 m。仍然需要可重组合数计算答案，此时新的状态为 $f_{v,r-\frac{1}{2},u,m}$。

虽然看起来复杂度很高，但是可以证明复杂度是 $O(n^3)$ 的。证明将会另附。

给定平面上 n 个横纵坐标两两都不同的点，两个点之间如果欧几里得距离不超过 R 则连边。每次在线地激活或者屏蔽一个点，查询全局非割点的数量。

F. Redundant Towers

Solution

根据横坐标建立线段树。对于线段树上一个节点，它代表横坐标在 $[low, high]$ 的所有点构成的诱导子图。

由于横坐标互不相同，显然只有区间内最左最右 $O(R)$ 个点可能往区间外连边，且每个点度数不超过 $O(R)$。

考虑直接在每个线段树节点记录对应区间的诱导子图，父节点的信息由子节点的信息合并得到，并附加 $O(R^2)$ 条边。那么对于单点修改，只需更新叶子到根的 $O(\log n)$ 个节点的信息；对于查询，直接在根节点记录的图中跑 Tarjan 算法统计非割点的数量。

不幸的是，上层节点的信息量将达到 $O(nR)$ 级别，不能接受。

考虑对线段树上指定节点的图结构进行压缩，构造一张等效的图替代它。
首先求出该图的点双连通分量，并建立 Block Forest，那么一个点不是割点当且仅当它是 Block Forest 上的叶子节点。
我们将该区间最左最右 $O(R)$ 个可能往外连边的点视作关键点，考察 Block Forest：

- 如果一个点是叶子，并且不是关键点，那么可以从 Block Forest 中移除。移除时需要给它的邻居附加 “该点不是叶子” 的信息。
- 如果一个点度数为 2，并且不是关键点，那么它对应的两条边可以合并，并在边上记录诸如 “这条边上被压缩了一个点” 的信息。

反复迭代上述两步，可得压缩 Block Forest 的方法。

压缩 Block Forest 的方法:

- 将区间最左最右 $O(R)$ 个可能往外连边的点视作关键点。
- 找出关键点在 Block Forest 上对应的最小斯坦纳树。
- 删除斯坦纳树未覆盖的点，并在斯坦纳树对应节点上附加信息。
- 压缩斯坦纳树上度数为 2 的点得到所有关键点对应的规模为 $O(R)$ 的虚树。

压缩完 Block Forest 得到规模 $O(R)$ 的虚树后，再根据它逆向构造出一张点数边数均为 $O(R)$ 的等效图，作为该线段树节点记录的信息。

从 Block Forest 的虚树构造等效图的方法：

- 对于每条虚树边，根据两端点是否代表点双，共 4 种情况分别构造大小为 $O(1)$ 的树链，树链中间穿插的点赋予点权表示 “该点对应原图中多少个点，它们要么同时都是割点，要么同时都不是割点”。
- 对于每个点双，构造一个环来等价表示点双连通性。

综上所述，可以在 $O(R^2)$ 的时间内完成线段树信息的合并，故时间复杂度为 $O((n + q\log n)R^2)$。

Thanks!