A

对于每个限制条件 (x_i, y_i, z_i)，我们称 x_i 与 y_i 之间有一条权值为 z_i 的边。

考虑对于每一个连通块分别处理出答案。

显然，对于一个连通块，如果某个点的权值被确定了，那么剩下的点的权值都会随之确定。所以我们可以考虑把所有限制条件都转化到这个连通块中编号最小的点上。

不妨设我们正在处理的连通块中的点的编号依次为 $1, 2, \ldots, p$。

我们可以求出这个连通块的一棵以结点 1 为根的生成树，设 w_u 表示结点 u 到根的路径上所有边的权值的异或和，则对于每一条非树边 i，必须满足 $w_{x_i} \oplus w_{y_i} = z_i$，否则这些限制条件之间就矛盾了，显然无解。

如果这些非树边都满足条件，则显然 $a_u = a_1 \oplus w_u$，因为要求序列 a 的字典序最小，我们只需要找到最小的满足条件的 a_1 即可。

考虑到 a_1 只需要满足 $l_u \le a_1 \oplus w_u$ 且 $a_1 \oplus w_u \le r_u$，于是我们可以放到 trie 上考虑：对于每个形如 $x \le a_1 \oplus y$ 或 $x \ge a_1 \oplus y$ 的条件，我们枚举 x 和 $a_1 \oplus y$ 在二进制下第一个不同的位置，这样就可以把条件转化为，ban 掉 trie 树上的 $\mathcal{O}(\log V)$ 个子树。于是我们只需要在 trie 上找到最小的没有被 ban 掉的结点即为 a_1 的最小值，如果所有结点都被 ban 掉了则无解。

时间复杂度 $\mathcal{O}(n \log V + m)$。

B

显然，如果一个串中不存在 `hi`，则对于每次询问，可以直接记录前缀中 `he` 的个数，通过二分答案找到第 x_i 个字符是由哪个字符得到的。

如果一个串中存在 `hi`，那么比较头疼的是形如 `hihihi...his` 和 `hihihi...hihe` 的结构。只有在右侧的 `his` 变成 `her` 再变成 `sher` 后，左侧的 `hi` 才能拿到 `s` 并开始变化。

于是考虑把所有询问按照 k_i 排序，用线段树维护每个前缀增加的字符数量，每次加入新开始变化的 `hi`，线段树二分找到第 x_i 个字符是由哪个字符得到的即可。

细节比较多，时间复杂度 $\mathcal{O}(n \log n)$。

C

$Y = 0$:

容易发现操作是可逆的，所以染色状态之间是等价关系。

我们总可以找到一个足够小的纵坐标 y_0，可以把所有黑点集中到这个纵坐标上。（$(x, y+1)$ 上的黑色可以变成 (x, y) 和 $(x+1, y)$ 上的黑色）。

考察一个黑点在 $y = y_0$ 这条线上的情况，可以发现是组合数在模 2 意义下的值，用 Lucas 定理能够计算。

多个黑点只需要各自计算然后异或即可。

由于此时答案黑点的纵坐标固定在 0 上，如果我们取 $y_0 = 1 - 2^k$，答案黑点在 y_0 上会表现为一段连续的长度为 2^k 的黑色点，且答案黑点的横坐标一定是这段黑点的横坐标最小值（左端点）。

取足够大的 k，然后二分就可以了。

时间复杂度 $O(n\log^2\max\{|x_i|,|y_i|\})$ 或 $O(n\log\max\{|x_i|,|y_i|\})$，瓶颈在于 Lucas 定理部分的实现差异。

考虑满分解法：

首先如果我们能够找到一个 $y=y_0$ 上的黑点，那么根据 Lucas 定理的循环节性质，可以通过从大到小枚举 2 的幂次，每次尝试往左右两边跳，来获得最左边和最右边的黑点，并定位得到答案黑点。

现在的问题是如何找到一个 $y=y_0$ 上的黑点。

一个巧妙的想法是将横坐标对 3 取模，之后观察黑点数量的奇偶性：

$$\begin{matrix}1&0&0\\1&1&0\\1&0&1\\0&1&1\\1&1&0\\&\cdots&\end{matrix}$$

我们发现必然至少有一个 r 使得横坐标模 3 等于 r 的横坐标上黑点数量是奇数！

那么找到这个 r，然后二分，每次找奇数的那一半就可以找到至少一个黑点了。

中间还需要实现一个数位 DP，求区间上横坐标模 3 后等于 r 的黑点数量奇偶性。

时间复杂度 $O(n\log^2\max\{|x_i|,|y_i|\})$。