联合省选 2022 模拟赛（题解）

Changzhou Senior High School of Jiangsu Province

Mar 19, 2022

双色砖

以下我们认为白色为 0，红色为 1。

子任务 1

爆搜……

子任务 2

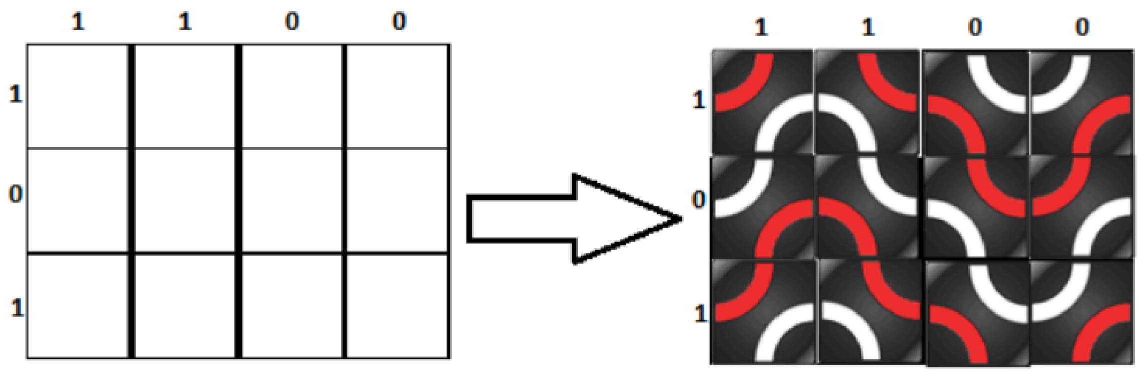
> **这一个子任务是本题的关键，给了很多分。**

注意到给出的双色砖中，**任意两个对边的颜色是不同的。**

于是在只考虑**限制1**时，只需确定第一行和第一列的状态即可还原整个矩形。

具体来说，设 $(R,1)$ 的左端颜色为 $colorx_R$，$(1,C)$ 的上端颜色为 $colory_C$，则只用枚举 $colorx_{1\sim H}$ 和 $colory_{1\sim W}$ 即可还原出整个矩形，然后再判断是否有环即可。

下图是 $colorx=\{1,0,1\},colory=\{1,1,0,0\}$ 的例子：



子任务 $3,4,5$

这些子任务可以放在一起讲。

首先，根据子任务 2 ，每个限制可以看作是强制 $colorx_R和colory_C$ 为某一个值。如果发现矛盾则可报告无解。

接着考虑如何满足第二个限制。

定义 $parityx_i=(colorx_i+i)\ mod\ 2$，$parityy_i=(colory_i+i)\ mod\ 2$

则发现当 $parityx$ **全相同**时，所有同色段都形如一个从左到右的“波浪”，不可能形成环。

这个结论对于 $parityy$ 也是类似的。

则可以算出 $parityx$ 或 $parityy$ 全相同的情况，并统计进答案。

接下来计算 $parityx$ 和 $parityy$ 都不全相同时的可能情况数：

设 x_0 是满足 $parityx_x \neq parityx_{x+1}$ 的某一个 x。

设 y_0 是满足 $parityy_y \neq parityy_{y+1}$ 的某一个 y。

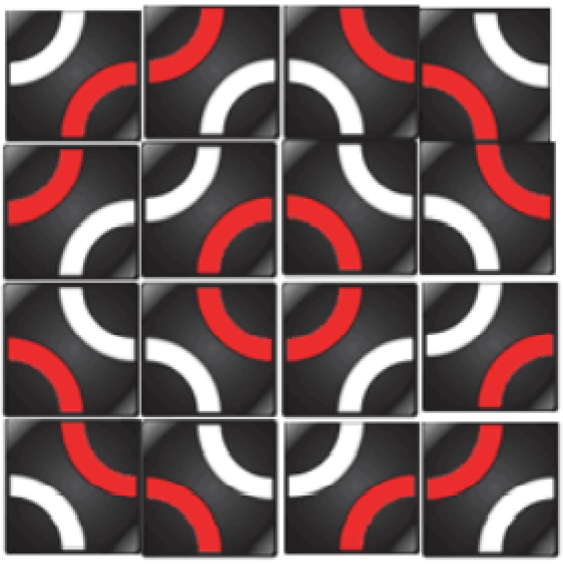
分两种情况：

> $parityx_{x_0} = parityy_{y_0}$

则不难发现此时 $(x_0, y_0), (x_0+1, y_0), (x_0, y_0+1), (x_0+1, y_0+1)$构成了一个环。

下图是一个例子。

其中 $parityx_2 = 1, parityx_3 = 0$，$parityy_2 = 1, parityy_3 = 0$，在 $(2,2), (2,3), (3,2), (3,3)$ 处形成了一个环。



于是这种情况下无解。

> $parityx_{x_0} \neq parityy_{y_0}$

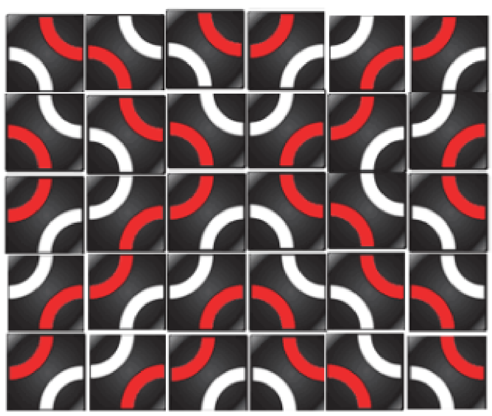
为方便叙述，记 $A = 0, 0, 0, \ldots\ldots, 0, 1, 1, \ldots\ldots, 1$，$B = 1, 1, 1, \ldots\ldots, 1, 0, 0, \ldots\ldots, 0$。

发现 $parityx$ 一定形如 A 或者 B，即不可能同时存在连续的 01 和 10，否则一定有一个位置与 $parityy_{y_0}$ 相同，无解。

对于 $parityy$ 也是类似的。

可以参考下面的例子：（其中右图为完整的矩形，左图为中央四个双色砖的具体形状）

其中 $parityx = \{1, 1, 0, 0, 0\}, parityy = \{0, 0, 0, 1, 1, 1\}$



所以，这种情况下，需要考虑的是 $parityx$ 为 A 且 $parityy$ 为 B，或者 $parityx$ 为 B 且 $parityy$ 为 A 的方案数。

考虑维护出 $parityx$ 和 $parityy$ 的每一个前缀/后缀能否全是 $0/1$，即可 $O(H+W)$ 算出合法的 $01/10$ 交界位置个数。最后把两个位置数相乘即可。

最后，将两部分答案相加即可。

时间复杂度 $O(H+W+N)$

$N=0$ 的部分分你也许可以打表。

$1 \leqslant H,W \leqslant 10^3$ 的部分分是给那些不会 $O(H+W)$ 算最后一个部分的人的，或者如果你会 $O(H*W)$ 的话也能过。

后记

如果你看到了测试数据，会发现有很多点 N 很小。

因为 N 越大答案越小，为了不使太多点答案不用取模就这么造数据了。

当然，为了防止什么 $O(N^2)$ 过题这种东西，还是有一些点 N 很大的，只不过 N 大的答案都很小。

如果你会什么剪枝跑出所有合法解的话，可能可以在没有子任务的情况下获得大量分数。

平衡树

本题中要用到的**AVL**树的性质：

> **性质1.** 树高是 $O(logn)$ 的。
>
> **性质2.** 对于每个点 x，x 的左子树中每个点的值 $< x <$ x 的右子树中每个点的值。

这些性质对你之后的正确性分析和时间复杂度分析是有帮助的。

$O(n^2)$

首先，很容易想到使用逐位确定。首先考虑第一个结点，尝试将其加入答案，假设其高度为h，那么我们至少要选择一棵结点数最少的高度为h的avl树加入答案，为了保证答案的最优性，显然应该选择字典序最小的那一棵加入。

然后，依次考虑每一个点是否能够加入答案，每次加入一个点需要额外再加一些点进答案来维持avl树的结构，只需检查在这之后答案的总点数是否超过k就可以知道这个点是否还能够出现在答案内。那么，应该选择哪些点加入来维持avl树的结构呢？

显然有两种想法，一种是选择需要加入点数最少的方案，一种是选择字典序尽可能靠前的方案（也就是只考虑为了"支撑"起当前一个点所对应的那颗字典序最小的avl树中，还未加入答案的那些）。

考虑一下几种情况，不难发现字典序尽可能靠前的方案就是需要加入点数最少的方案，这也很符合我们的直觉。到现在为止，考虑一个点的复杂度还是 $O(n)$，总复杂度为 $O(n^2)$。

优化

我们没有没有充分利用题目的性质，考虑进行一些优化。

首先，记下答案所组成的**AVL**树内每棵子树的高度，通过合适的方法，我们可以做到在每次尝试时只会遍历那些高度发生变化了的结点，这样一来若是没有失败，由于高度最多为 $O(\log n)$，总复杂度就将为了 $O(n \log n)$。

其次，我们拥有如下性质：

> 假设当前考虑到了点 x，假设其深度为 h，如果它不能够被加入答案，任何之后深度同为 h 的点都必然无法被加入答案，继而之后任何深度**不低于** h 的点都不需要再进行考虑。

考虑如何证明这个性质。对于第一个点，这是显然的。对于之后的每个点 x，若其深度为 h,考虑从根到 x 的路径上，最后一次走进右子树是在点 u 处，那么根据**AVL**树的性质，u 的左子树内一定有一个深度为 $u-1$ 的点 y，且由于点 x 未被跳过，y 一定被选入了答案内。

这样一来，只要记下当前可能加入点的最大深度，每次我们尝试加入一个点失败就会将其更新并减小，并直接跳过那些。由于输入树的高度是 $O(\log n)$ 级别的，这样我们就最多只会失败 $O(\log n)$ 次，花在尝试失败上的时间就也只有 $O(n \log n)$。

这样一来，为了"支撑"点 y，点 x 的二级祖先一定会被选入答案。此时，分加入 x 是否更新了答案对应树的深度，x 的父亲是否被选入答案等情况考虑，可以发现对于任何同深度的点 $z > x$，为了支撑x所需要的新加入的点数不超过为了支撑 z 所需要的新加入的点数，也就证明了上面的性质。

合并两种情况对复杂度的贡献，总时间复杂度依旧为 $O(n \log n)$。

垃圾题

前言

这确实是一道垃圾题。而且十分垃圾。

子任务 1

对于所有 $O(N^2)$ 个区间，统计其是否是好的，询问时直接利用二维前缀和回答即可。

时间复杂度 $O(N^2+Q)$

子任务 $2,3$

可以见 **Luogu P4062**。

对于每次询问，采用那题的做法即可。

时间复杂度为 $O(N*Q*polylog(N))$ 的能过子任务 2，时间复杂度为 $O(N*Q)$ 的能过子任务 $2,3$。

一些重要性质

> **以下开始和正解有关**，默认 N,Q 同阶。

一个区间是好的，当且仅当其**区间众数**出现次数**严格大于**区间的一半。

为了方便叙述，把**区间众数**出现次数**严格大于**区间的一半的众数称作“好的众数”。

而对于一个区间，这样的数最多只有一个，这启发我们可以枚举区间众数 x，然后算有多少区间的“好的众数”是 x。

> **Observation 1**：对于一个出现次数为 x 的数，它能构成合法的以它为区间的左右端点个数是 $O(x)$ 的。

因此，所有端点对应严格众数的数的种类数之和是 $O(N)$。

> **Observation 2**：对于一个端点，能构成严格众数的数的种类是不超过 $\log N$ 的。

我们尝试用莫队去维护答案。我们按端点对应严格众数的**数的种类数**进行分块。

端点左右移动时，需要加上或减去固定一个端点，另一个端点在一个区间的答案。

子任务 4

在这种情况下，你不用考虑左端点的移动，或许容易实现一些。

子任务 5

出题人对于这一档也没有什么巧妙地做法，或许只是提示你要往**数的种类**上去想。

子任务 $6,7,8$

在线地询问有些麻烦，我们可以把这些询问离线下来。**（也就是莫队二次离线）**然后再做一个扫描线，此时有 $O(N)$ 个修改，$O(N\sqrt{N})$ 个询问，如果使用分块平衡一下复杂度就可以做到 $O(N\sqrt{N})$ 的时间复杂度，如果把 $O(N\sqrt{N})$ 个询问压缩成 $O(N)$ 个区间，空间复杂度可以做到 $O(N)$。

此外离线下来使用树状数组的做法也是可以的，时间复杂度为 $O(N\sqrt{N}logN)$，视常数能获得不同的分数。**甚至可以过题。**