虚拟

测试点编号 $1 \sim 5$：

按照题目要求暴力建图，dfs 判断连通性，复杂度 $O(n^2)$

期望得分：25分

测试点编号 $6 \sim 12$：

考虑 $lk = rk$ 的情况，我们可以看出来是需要判连通性，但是连边是 n^2 级别的

事实上可以发现 b 那么多其实没用，所以考虑将每一条连续的 b 缩成 1 个 b

这个时候发现每一个点最多连出去 2 条边，可以连边后 DFS 做到 $O(n+q)$

期望得分：35分

测试点编号 $13 \sim 16$：

如果我们对解法一使用线段树优化建图，复杂度 $O(n \log n)$

期望得分：45分

测试点编号 $17 \sim 20$：

设 a 的个数为 cnt 我们分类讨论:

当 $lk \le \left\lfloor \frac{cnt}{2} \right\rfloor$ 时，必然输出 Yes ，证明：考虑最坏情况 $rk = lk + 1$ 且全部字符都为 a （若有 b 则通过 b 可以转移，必然比单纯的 a 优）。而此时，考虑最坏情况 $lk = \left\lfloor \frac{cnt}{2} \right\rfloor$ ，此时可以先一步跳到中间，再将左右一一对应即可

当 $lk > \left\lfloor \frac{cnt}{2} \right\rfloor$ 时，对于任意点前缀有 q 个 a 来说，对于 $q \ge lb$ 和 $q \le cnt - lb$ 的点都是互相联通的，证明：考虑最坏情况 $rk = lk + 1$ ，且所有点都是 a，因为可以一步跳到 $lk+1$ 然后 lk 跳回来，如此往复，还可以一步跳到 lk，停止，所以对于 $q \ge lb$ 和 $q \le cnt - lb$ 的点 1 都可以到达，也就是互相连通

当然，也可以不进行分类讨论，还存在一种简单的做法

由于我们只需要判连通性，有效边数实际是 $O(n)$ 级别的

如果 x 对 $[l, r]$ 存在连边，我们显然能够在 x 和 l 之间连边，然后在 $(l, l+1)$，$(l+1, l+2) \dots$ 之间连边达到期望的效果，这样的边数也是线性的，而每个位置 i 对应连向的区间可以指针扫描求出，复杂度 $O(n+q)$

期望得分：100分

游戏

先考虑若求出小 W 的答案之后小 M 的答案应该怎么求。

设 f_i 为当环长为 i 时的方案数， g_i 为当环长为 i 时本质不同的方案数。

那么套用 Pólya 定理可以得到 $g_n = \frac{1}{n}\sum_{i=1}^{n} f_{\gcd(i,n)}$。

或者枚举循环节直接容斥计算也可行。

设 h_i 为当环长为 i 且循环节为 1 的方案数。

则 $h_i = f_i - \sum_{d|i \wedge d\neq n} h_d$。

因此 $ans_i = \sum_{d|i} \frac{h_d}{d}$。预处理计算即可。

下面的所有预处理皆针对第一种方法。

测试点编号 $1\sim 5$：

对于小 W，先做出链的答案，然后再枚举环上选的最小点即可。

做的好的话时间复杂度 $O(Tn^2)$。

预处理的话可以做到 $O(n^2)$。

期望得分：25分

测试点编号 $6\sim 7$：

送分点，答案显然就是 n 和 1。

期望得分：10分

测试点编号 $8\sim 10$：

假设点集可以为空的话，打个表出来，发现对于小 W 来说有递推式：

设 f_i 为环大小为 i 时的答案。

那么 $f_i = f_{i-1} + f_{i-2}$。

时间复杂度 $O(Tn\log n)$。

预处理一下能做到 $O(n\log n)$。

期望得分：15分

测试点编号 $11\sim 14$：

由上一个 Subtask 的启发，那么递推式是不是 $f_i = f_{i-1} + f_{i-k-1}$ 呢?

确实是这样的。

考虑将选点转换为，每次在一个合法环的情况下加入一条长为 $k+1$，且只选开头的链。且每次把链插在可以插的最末尾。容易发现所有方案都可以转变为这种方式。

那么递推式就不难得到了。要么加入一个空点，要么加入一条只选开头的链。即 $f_i = f_{i-1} + f_{i-k-1}$。

对于点集可以为空的情况，可以改成 $f_i = f_{i-1} + f_{i-k-1} + 1$，或者让所有 f_i 减去 1 也可以。

时间复杂度 $O(Tn\log n)$。

期望得分：60分

测试点编号 $15 \sim 20$：

考虑怎么预处理。

枚举每种环长 i 对长度为 n 的答案的贡献，可以发现贡献为 $\varphi[\frac{n}{i}] \times f_i$。

然后就简单了，对于每种长度 i，直接枚举它的倍数，预处理贡献。

时间复杂度 $O(n \log n)$。

期望得分：100分

吉瀬金鑰

先做一个转换：$B_i = A_i - i$，那么每次操作就是交换 B_i 和 B_{i+1}，代价是 $B_i + B_{i+1} + 2i + 1$。那么最后的序列 B 就必须是严格单调递减的，那 -1 的情况就判好了。

考虑这样一个贪心：从前往后考虑每个数，在这之前有个单调递减的序列，每次相当于在末尾新加入一个数，然后通过交换把这个数挪到让这个序列继续递减的位置。可以证明这是最优的。

然后使用数据结构模拟一下就好。

矩阵

先考虑怎么判定是否有解：对于一个点对，只有两种情况：在两个序列中的前后关系相同或者不同。那么后者会对两个序列中的一个产生 1 的贡献，前者要么两个序列都产生 1 的贡献，要么两个序列都不产生贡献。

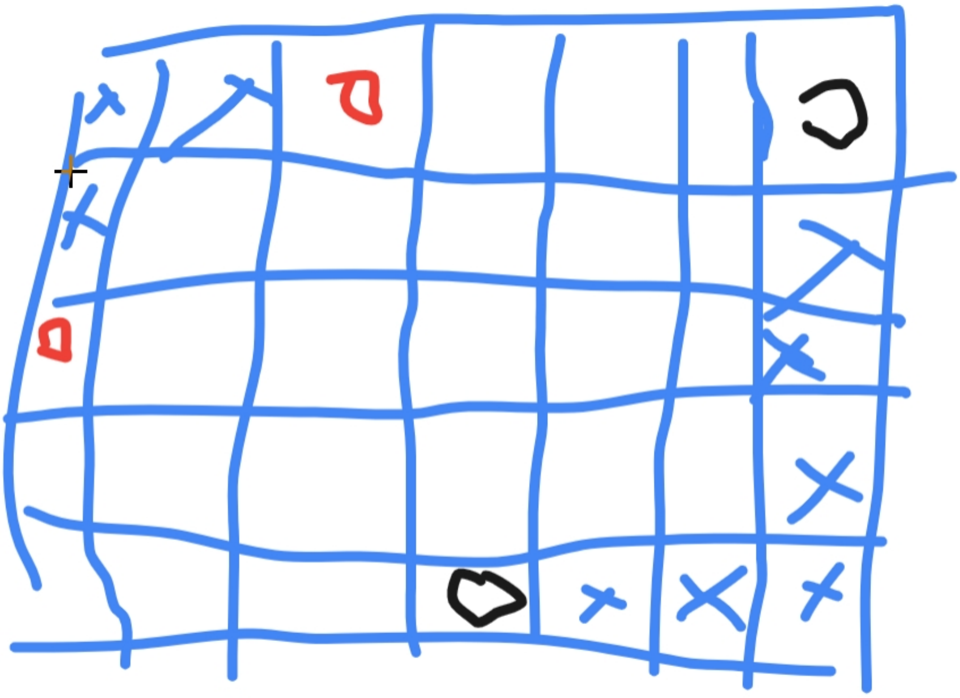
可以算出上述两种点对的逆序对数，如果算不出来或者算出来非常大，那就是无解。

设前者当前的需要逆序对是 W，后者是 Z，最多可能的分别是 $Wmax$ 和 $Zmax$。这意味着所有满足 $0 \le W \le Wmax$ 并且 $0 \le Z \le Zmax$ 的所有 (W, Z) 都是可行的。

然后考虑从小到大填数，然后就删掉这个位置，这样会对于剩下点集的 $W, Z, Wmax, Zmax$ 都会产生影响。可以归纳证明对于任何一个点集，只要满足 $0 \le W \le Wmax, 0 \le Z \le Zmax$ 都是可行的，这样就有个多项式级别的做法。

那么考虑如何证明：那考虑弄出一个更加强大的结论：对于当前满足条件的 W, Z，在当前点集中，(i, j) 最大/最小，(j, i) 最大/最小的四个（可能更少）的点中，必然有一个可以归纳下去。

先从 W 的角度考虑：把四个点分成两类：先证明对于颜色不同的任何两个点都至少有一个不会让 W 不符合限制。



红点会让 W 和 $Wmax$ 同时减去它右下角没有被删的点数 $x1$，黑点会让 $Wmax$ 减少它左上角没有被删的点数 $x2$。如果这两个都不可以，说明 $Wmax \le x1 + x2 - 2$，但是 $Wmax$ 是当前可能的最大逆序对数，显然会有 $Wmax \ge x1 + x2 - 1$，那就矛盾了。

这样的话，如果有某个颜色的点不行，那么另外一个颜色的两个点一定都可行，这是在只考虑 W 的情况下。

在考虑 Z 的情况下，也能得出类似的结论，只不过染色的情况不同而已。画出来之后发现两个集合一定有一个集合重合，那就满足了。

那么同样依靠这个结论，只要对于这几个点判定就好，使用简单的数据结构数点就能够做到 $\mathcal{O}(Tn^2 \log^2 n)$ 的复杂度。实际上如果每次只删这四个方向的点，那么删点的结构是很好看的，那就可以做到 $\mathcal{O}(Tn^2)$ 的复杂度，不过没有特意卡。