贪心/构造/博弈杂题选讲

wlx

2025 年 2 月 7 日

No resource

- 你初始有 W 块钱，你需要在接下来的 n 天内炒股赚到尽可能多的钱。
- 已知第 i 天的股票值 c_i 块钱，且买卖和持有股票没有数量限制，但必须是整数股。每一天买/卖股票分别需要付 a 块/b 块每股的手续费。
- 问 n 天后你最多能赚多少钱。$n \leq 10^7$，保证答案小于 10^{18}

No resource

$n \le 10^7$ 表明需要 $O(n)$ 做法。反悔贪心似乎可行，但比较暴力。

线性的 DP 比较困难，因为状态是持有的股票数和钱的二元组，难以直接比较优劣，且转移复杂。

一个直觉是，每天要么买股，要么卖股，要么不行动，且一定是全部买入/卖出。前者是显然的，但后者需要讨论一下。

假如某天你只卖出一部分股票，那么你的下一次操作不可能是卖出，否则劣于在价格更高的一天卖出。则下一次操作是买入，且两天价格差大于 $a+b$，不然卖出再买入无意义。此时全部卖出再买入对应股数一定不劣。

No resource

再考虑买入部分股票，下一次操作只能是全部卖出。这说明买入该股票再卖出赚钱，全部买入不劣。

于是操作简化成了全部买入和全部卖出，只需解决状态的比较。由于全部买入的前提，剩余的钱一定少于某一次买入的价格。显然，股票卖出价格高于买入价格加上手续费，因此股票数更多一定更优。

P4704

- 一个平面上有一个圆，圆上等距分布着 $2n$ 个端点。
- 给定这些端点形成的一组大小为 n 的匹配，将匹配的点相连，可得到 n 条线段。
- 问最少需要几条直线，才能使每条线段与至少一条直线在非端点处相交。
- $n \le 2 \times 10^5$

P4704

端点将圆分成 $2n$ 个弧段，显然可以让每条直线经过弧段中点。

进一步发现，我们并不需要关心直线的具体形态，只需要知道直线经过了哪些弧段中点。对于线段 $[l,r]$，只需要满足弧段 (l,r) 和 (r,l) 上都有点被经过即可。答案即为点数除以 2 上取整。

如果是序列问题，可以很轻易地用贪心解决，即每次找当前最靠前的 r_i 即可。事实上，每个位置处对应的下一个位置可以 $O(n)$ 预处理得到。

考虑枚举一个位置断环为链，使用倍增即可在 $O(\log n)$ 时间复杂度内完成一次贪心。

P4704

虽然这一做法足以通过此题，但我们还有一些更聪明的做法。

考虑长度最短的区间，设其长度为 d，显然这 d 个位置中一定存在一个位置被选。

同时，某个位置与其下一个位置间的距离一定不小于 d，意味着我们可以用 $O(n/d)$ 模拟贪心过程。

这样，总时间就是 $O(n)$ 的。

CF1693F

- 给定一个长为 n 的 01 序列 S。
- 你可以选择任意区间 $S[l,r]$，花费 $|c_0 - c_1| + 1$ 的代价将其排序。此处 c_i 为 $S[l,r]$ 中 i 的个数。
- 问使 S 变得有序的最小代价。$n \le 2 \times 10^5$

CF1693F

手玩几个小样例可以发现，最优解可以只操作 $c_0 = c_1$ 的区间。或者说，操作任意 $c_0 \neq c_1$ 的区间一定可以通过操作若干 $c_0 = c_1$ 的区间低代价替代。

不妨设要操作的区间 $c_1 > c_0$，如果最后一位是 1，将右端点向前移动显然更优；反之，我们可以找到一个 01 数量相等的后缀，对其进行操作。两种情况 c_1 都会减少，因此这样操作不劣。

上述贪心可被进一步强化：同样不妨设 S 中 1 更多，如果最后一位是 1 可以将其忽视。当最后一位是 0 时，应选取以该位置为右端点的最大区间进行操作。由于操作为排序，使用线段树维护前/后缀和是简单的。

上述做法还可以进一步优化成线性，因为左端点始终向左移动。预处理出前/后缀和后从后向前扫即可做到 $O(n)$。

CF1329E

- 有一个下标从 0 到 n 的数组 a，初始 $a_0 = a_n = 1$，有 m 个给定位置为 1，其余位置为 0。
- 现在需要再选 k 个为 0 的位置变为 1。记相邻 1 位置之间的最近距离为 l，最远距离为 r，你需要最小化 $r - l$。
- $n \leq 10^{15}, m \leq 4 \times 10^5, k \leq n - m$

CF1329E

本质上是对 $m+1$ 个段进行切分，显然每一段均分是最优的。设第 i 段段长 c_i，划分 d_i 次，则 $\sum d_i = k+m+1 = K$，且要最小化：

$$\max\lceil\frac{c_i}{d_i}\rceil - \min\lfloor\frac{c_i}{d_i}\rfloor$$

乍一看这个式子没有什么好的性质，需要进一步研究。

假设我们取到了 $R-L$，则有两个显然的要求：

- $\sum\lceil\frac{c_i}{R}\rceil \le K \le \sum\lfloor\frac{c_i}{L}\rfloor$。
- 存在 d_i 使得 $L \le \lfloor\frac{c_i}{d_i}\rfloor \le \lceil\frac{c_i}{d_i}\rceil \le R$，即 $\lceil\frac{c_i}{R}\rceil \le d_i \le \lfloor\frac{c_i}{L}\rfloor$。

CF1329E

所以可取的 d_i 是一个区间，只要有 $\lceil \frac{c_i}{R} \rceil \le \lfloor \frac{c_i}{L} \rfloor$，第一个条件就是充要的。于是可以先二分出 L, R 的上/下界 L_0, R_0，这样条件一就直接满足了。

现在需要调整 L, R 使得 $\forall i, \lceil \frac{c_i}{R} \rceil \le \lfloor \frac{c_i}{L} \rfloor$。显然，我们有 $\lceil \frac{c_i}{R} \rceil \le \lceil \frac{c_i}{L} \rceil \le \lfloor \frac{c_i}{L} \rfloor + 1$ 和 $\lceil \frac{c_i}{R} \rceil - 1 \le \lfloor \frac{c_i}{R} \rfloor \le \lfloor \frac{c_i}{L} \rfloor$。
由于我们的调整必须改变 $\lceil \frac{c_i}{R} \rceil$ 或 $\lfloor \frac{c_i}{L} \rfloor$，那么只需要改变 L, R 中的一个即可。现在我们可以对每个 i 求出 (L_i, R_i)，表示至少需要满足 $L \le L_i, R \ge R_i$ 中的一个。

使用贪心，按 L_i 从大到小排序，对每一段前缀考虑即可。

CF1637H

- 给定一个 $1 \sim n$ 的排列 p，问选择 p 的一个长为 k 的子序列，将其按原顺序移动到 p 的开头后，排列的最小逆序对数。
- 你需要对每个 $k \in [0, n] \cap \mathbb{N}$ 求出答案。$n \leq 2 \times 10^5$

CF1637H

仅移动位置 i 的贡献/逆序对减少量为 $(i-1)-2\sum_{j=1}^{i-1}[p_j<p_i]$，记为 d_i。考虑选出的子序列位置为 $q_1,q_2,\cdots,q_k$，则贡献为 $d_q-2\,\mathrm{inv}(p_q)$，即每一项 d 求和减去两倍逆序对数。

先将贡献写作有利于分析的 $\sum\limits_{i\notin q}\sum\limits_{i<j,j\in q}([p_i>p_j]-[p_i<p_j])$。

考虑原排列 p 的一个逆序对 (i,j)，$i<j$ 且 $p_i>p_j$。感觉或手玩一下，如果 p_i 被选入子序列，p_j 似乎也应被选入，不然将 p_i 换成 p_j 似乎不劣。

尝试证明。为简化 i 到 j 之间的情况，我们可增加一些约束，比如选取 $j-i$ 最小的逆序对。这样可以保证 $\forall l\in[i+1,j-1]$，$p_l>p_i$ 一定不被选，$p_l<p_j$ 一定被选，且 $p_l\notin[p_j,p_i]$。

CF1637H

- 对于 $l < i$ 的位置，只有 l 未被选且 $p_l \in [p_j, p_i]$ 时产生 $+2$ 的贡献。
- 对于 $l > j$ 的位置，只有 l 被选且 $p_l \in [p_j, p_i]$ 时产生 $+2$ 的贡献。
- 对于 $i < l < j$ 的位置，$p_l > p_i$ 和 $p_l < p_j$ 时均产生 $+1$ 贡献。

i, j 共同又产生 $+1$ 的贡献，怎么算都是正的！

加入这一限制后，直接有 $\mathrm{inv}(p_q) = \sum_{i=1}^{k} \sum_{j=q_i+1}^{n} [p_i > p_j]$，显然这一步将 $\mathrm{inv}(p_q)$ 放大了。于是令 $d'_i = d_i - 2\sum_{j=i+1}^{n} [p_i > p_j]$，贡献直接转化为 $\sum d'_i$，选最大的几个即可。

d' 可用树状数组轻易求出，总时间 $O(n \log n)$

CF1481F

- 给定一颗大小为 n 以 1 为根的树，有 x 个字符 a 和 $n-x$ 个字符 b。
- 你需要给每个结点分配两种字符中的一个，此时所有从根结点开始到树上某一点结束的路径对应 n 个字符串。
- 请最小化不同的字符串数目。$n \leq 10^5$

CF1481F

设树高为 h，则 $ans \geq h$，且感性认为 $h+1$ 是比较好达到的。

先考虑 $ans = h$ 的情况，此时每一层的结点只能放同一种字符，可以用背包判定。由经典结论，使用二进制分组后时间复杂度是 $O(n\sqrt{n})$ 的。

再考虑 $h+1$ 是什么情况，此时应当是在某一层出现分歧，且有一方全为叶子结点。

假设第 i 层有 k 非叶子节点，前面正常填，到现在两种字符分别剩下了 x_1, x_2 个。显然 $2k \leq x_1 + x_2$，因此这 k 个结点可以全部使用同种字符。现在让叶子结点也使用这种字符。

如果字符不够，我们的构造其实可以直接结束，因为剩余结点全部使用另一种字符即可，答案即为 $h+1$。

GYM101221A

- 有一个下标为任意整数的数组 a，初始 $a_i = 0, \forall i \le 0$ 和 $i > 2n$；$a_i = (i \bmod 2) + 1, \forall i \in [1, 2n]$。
- 每次，你可以选择两个相邻且非零的位置，和两个相邻且为零的位置，整体交换其上的值。
- 最终，你需要将 a 中所有 1 调换到 2 的前面，并保证非零位置连续。
- 请构造方案，使得操作步数最小。$3 \le n \le 10^6$

GYM101221A

即将 a 从 $2121\cdots2121$ 变成 $1111\cdots2222$，先考虑最小操作步数。观察样例和手玩可以发现是 n。

一个解释是，最终有 $2(n-1)$ 个非零位置与下一个位置相同，显然，每次操作最多使这样的位置增加两个，且第一次操作至多增加一个。

这同样指出了每次操作尽可能要满足的条件。

第一下应该形如 $2\underline{00}1\cdots2121\underline{12}$，之后 $2\underline{21}1\cdots\underline{00}2112$。发现先做外侧需要将 2112 的 11 移到前面，再将 2211 的 22 移到后面。好像不太可能，而且不能一直使用最外面的 00，不然次数肯定不够。

GYM101221A

但如果先做 $2211\cdots002112$ 的中间，似乎可以凑成递归的形式。

此时，我们需要将里面的 $2121\cdots2100$ 变成 $00111\cdots222$ 。这样，就可以进行 $2211\underline{11}\cdots2\underline{00}2$ 和 $\underline{00}1111\cdots2\underline{22}2$ 补全递归。

最后考虑边界情况：$n=4$ 没有问题，但 $n=1,2,3$ 都是不可能完成的。所以需要对 $n\leq 7$ 的情况特判，此时可以直接暴搜。

QOJ6511

- 给定两个 $2 \times n$ 的二维数组 a, b，其中的数均组成 1 到 $2n$ 的排列。
- 每次可以选择 $a_{p,i}$ 和 $a_{q,j}$ 进行交换（$p, q \in \{0, 1\}$），要求 $a_{p,i} > a_{1-p,i}, a_{q,j} > a_{1-q,j}$
- 问能否在 $5n$ 步内将 a 变成 b，可以则给出方案。$n \leq 2 \times 10^3$

QOJ6511

由于越大越可能移动，我们从小往大考虑。可以从 1 开始把一定不能动的数判了。扫到需要移动的 x 时，看看 x 能否移动，不能就无解了。

再看目标位置上的数 y 能否移动。如果可以，感性上似乎可以直接移动；反之，我们需要找到一个可移动的中介 z，先换到 y 对面，再将 x 换过去。由于我们优先让更小的数归位，所以 z 应当满足 $z > x$，且 $z < y$。如果 z 不存在，感性上似乎无解了。

尝试严谨化上述过程。

首先观察到 $\min\{a_{0,i}, a_{1,i}\}$ 是单调不增的，且下降后较小值位置交换。同时，上述做法保证了较大值位置在移动后不变（除了归位的时候）。

QOJ6511

如果 x 不能移动，那么 x 较小，且位置一直不变。由于更小的数先归位，我们有 $\min\{b_{0,i}, b_{1,i}\} \ge x$，简单分析即可推出无解。

如果 x 可以移动，但满足条件的 z 不存在。类似的，可证明每个 $z \in (x, y)$ 的位置都没有移动过。进一步还可归纳得出 z 应当现在就归位了，否则也无解。

那么 z 不可能移动，否则会使 $\min\{b_{0,j}, b_{1,j}\}$ 减小。于是为了让 y 所在列的较小值减小，且位置不变，需要至少两次较小值的改变。

在 $z \in (x, y)$ 全部不能用的前提下，这是不可能的，所以无解。

直接模拟上述过程即可，步数不超过 $4n$。

P3526

- 给定一个长为 n 的字符串 S，请你构造一个长为 n 且字典序最小的 01 串，使两者拥有相同的周期。若不存在报告无解。
- 称 $p \in [1, n)$ 是 S 的周期，如果 $\mathrm{S}_i = \mathrm{S}_{i+p}, \forall i \in [1, n-p]$。
- $n \leq 2 \times 10^5$

P3526

显然 p 是周期当且仅当 S 有长为 $n-p$ 的 border，所以周期相同等价于 border 长度相同。

一个结论是若 p,q 是 S 的周期，且 $p+q\le n$，则 $\gcd(p,q)$ 也是 S 的周期。接下来会多次使用该结论。

可以先用 kmp 处理出 S 的所有 border/周期，一些情况下的构造是显然的：

- 如果 S 只有一种字符，就返回全 0 串
- 如果 S 无周期，就将全 0 串最后一位改为 1

P3526

周期具有很好的性质，所以考虑 S 的最短周期 d。

如果 $2d \leq n$，记 $\mathrm{T} = \mathrm{S}[1, d]$，则 $\mathrm{S} = \mathrm{T} \cdots \mathrm{T}\mathrm{T}'$，此时 d 的倍数都是周期。

注意到构造出的串也一定有 $\mathrm{A} \cdots \mathrm{A}\mathrm{A}'$ 的形式，很自然地想到将 $\mathrm{T}\mathrm{T}'$ 和 $\mathrm{A}\mathrm{A}'$ 对应起来。下面证明 $\mathrm{T}\mathrm{T}'$ 和 $\mathrm{A}\mathrm{A}'$ 同周期与 $\mathrm{T} \cdots \mathrm{T}\mathrm{T}'$ 和 $\mathrm{A} \cdots \mathrm{A}\mathrm{A}'$ 同周期等价。

设 T' 长度为 d'，则 $< d + d'$ 的 border 是立即满足的。而对于所有 $\geq d + d'$ 的 border，两者均保证了 $|\mathrm{A}|$ 是最短周期，此时这样 border 的长度一定形如 $kd + d'$，也是立即显然满足的。

所以二者等价，且都是让 A 的字典序最小，可以递归构造。

P3526

如果 $2d > n$，此时 $\mathrm{S} = \mathrm{TBT}$，T 是最长的 border，但 B 的性质我们并不清楚。

此时答案应当形如 ADA，显然 A 和 T 的 border 集合应当相等，而 ADA 最长的 border 为 A。A 的字典序应当优先最小，所以先递归，再尝试确定 D，设 $l = |\mathrm{B}|$。

D 为全 0 串肯定是最好的，但是这不一定合法，假设此时出现了更长的 border，长度为 $p > d$。

如果 $p \le d + l$，那么 border 的头尾一定是 0，进一步推出 A 是全 0 串。此时直接将 D 最后一位设为 1 即可。

P3526

再考虑只有 $p > d + l$ 的情况，那么串 ADA 存在周期，取最短周期 q，则 q 是 $d + l$ 的因数。

$q \leq l$ 的时候同样可以推出 A 全 0，而 $q > l$ 时可设 $\mathrm{C} = \mathrm{T}[1, q - l]$ 非全 0，那么 A 可表示为 $\mathrm{CD} \cdots \mathrm{CDC}$ 的形式，其中 D 全 0。

再次考虑将 ADA 中间 D 的最后一位更改为 1，即 $\mathrm{AD'A}$，并假设存在 $q' > d$ 的 border。

由 1 的数量可以推出 $q' \geq d + l$，留意到 $n - q'$ 是周期，可对其长度进行讨论。

P3526

- 当 $d+l \le d' \le d+l+|\mathrm{C}|$ 时，可推出 CD 等于 CD$'$ 的一个 rotate，但两者 1 个数不同，矛盾。
- 当 $d' > d+l+|\mathrm{C}|$ 时，考虑串 $\mathrm{AD}', \mathrm{AD}$ 的公共前缀，长度为 $d+l-1$。显然 $n-d' < d-|\mathrm{C}|$ 为周期，长为 $|\mathrm{C}|+l$ 的 CD 也是周期，推出 $\mathrm{AD} = \mathrm{AD}'$，矛盾。

综上，直接返回 AD$'$A 即可。

每次递归至少将长度减少 $\frac{1}{3}$，且一轮是 $O(n)$ 的，因此总时间也是 $O(n)$

AGC010F

- 给你一棵 n 个结点树，第 i 个结点上有 a_i 个石子。Alice 和 Bob 要轮流移动一个标记。
- 每次需要从标记所在结点取走一颗石子，再将标记移动到相邻结点，标记无法移动时判负。
- 问标记初始在哪些结点时先手必胜。$n \leq 3000$

AGC010F

一个观察是，如果标记初始结点已经确定，那么标记在每个节点时轮到谁也是确定的。也就是说，将某个 a_i 减小一定有利于某一方而不利于另一方。同时，断掉一条边也减小一定有利于某一方而不利于另一方。

怎么断掉一条边？如果标记在 u 而 $a_u \le a_v$，它不可能通过 v 了，因为另一方可以跳回 u 和他对拼。由于这一步双方对拼变相减小了 a_u，而 (u, v) 又相当于断掉了，这对于操作者肯定不利。

于是移动标记时只会向石子更少的结点移动，后面就简单了，做到 $O(n)$ 都可以。

CF1776M

- 给定一棵 n 个结点的树，Alice 和 Bob 轮流取点，每次可取走一个叶子（度数 ≤ 1），Alice 先手。
- Alice 想让自己取走的结点的编号的最大值尽可能大，而 Bob 想要这个值尽可能小。
- 问两人都取最优策略时 Alice 取到结点的最大编号。$n \leq 10^5$

CF1776M

开始想到二分，好像没有什么实质性帮助，但启示我们 Bob 必须保证取走所有大于 ans 的点。

注意到 n 为偶数时 Alice 可以取走任意一个结点，所以答案为 n，因此只用考虑 n 为奇数的情况。

先对初始的叶子取 max，接下来考虑非叶子结点。直接考虑 Bob 一定能取走的点集有点困难，可以先试着找出哪些点集 Bob 一定不能全部取走。

玩几个样例也许能注意到，距离为奇数的点对 (u, v) 是满足条件的。如果 Bob 想取走这两个点，就需要避免这两个点在他操作后成为叶子。此时，Alice 也可采取这种策略，那么恰好有奇数个点可以取，一定会轮到 Bob 开出叶子，然后 Alice 取走。

CF1776M

而如果 (u, v) 之间距离为偶数，Bob 一定可以把 u, v 都取走，因为 Bob 一定可以使 Alice 先开出一个叶子，再交换先后手转为 n 为偶数的情况。

对于链而言，这一结论可强化成一个两两距离为偶数的点集，因为可以只考虑最外侧的两个点，然后归纳。

在树上继续手玩，发现如果有三个点 (u, v, w) ，能找到一个点 m，使得 u, v, w 在 m 的不同子树内，且到 m 的距离都为偶数，则 Alice 也能取走 u, v, w 中的至少一个，证明和点对是同样的思路。

CF1776M

加入这种情况的判断后，链上的结论也适用于树上了，证明可以考虑虚树大小加叶子数一定为奇数，然后归纳。

也就是说，Bob 一定能取走的点集 S 满足两两距离为偶数，且不包含上面所说的三元点对。

到这里已经可以有相当多做法了，最朴素的就是倒序加点，判断是否合法。以 n 为根维护虚树可以暴力跳 father 做到 $O(n)$

CF1033G

- 有 n 堆石子，第 i 堆大小为 a_i，Alice 和 Bob 要玩取石子游戏。
- Alice 每次可以从一堆中拿走 A 个石子，Bob 每次可以拿走 B 个。石子数量不够不能拿，某回合不能操作的一方判负。
- 有四种可能的局面：Alice 必胜，Bob 必胜，先手必胜，后手必胜。问对于所有 $A, B \in [1, m]$ 的 (A, B) 对，每种局面各有几个。
- $n \leq 100, m \leq 10^5$

CF1033G

由于某个 (A, B) 对应 Alice 必胜时可推出 (B, A) 对应 Bob 必胜，两者数量应该相等。

思考后发现，a_i 减去 $A+B$ 似乎不会影响最后的结果，因为必胜方可在必败方第一次操作这一堆后操作。

那么先枚举 $A+B$，再进行讨论应该比较可行，则所有 $a_i < A+B$。不妨设 $A \leq B$，那么一堆石子只要被 B 取过就等于废了，被 A 取过就不能被 B 取了。

CF1033G

如果存在 $A \le a_i < B$，这堆只能被 A 取，Alice 必胜。类似的，如果存在 $2A \le a_i$，Alice 先取一次这堆就可保证必胜，只要这种堆的数量 ≥ 2 Alice 也必胜。

由于 $a_i < A$ 的堆也是废的，现在只剩下 $B \le a_i < 2A$ 的堆和至多一个 $2A \le a_i$ 的堆。

如果不存在 $2A \le a_i$，就只需要拼 $B \le a_i < 2A$ 的奇偶性了。反之，Bob 必须先干掉 $2A \le a_i$ 的堆，再去拼奇偶性。

讨论完之后的计数并不困难，总时间 $O(nm \log n)$

CF1659F

- 给定一棵 n 个结点的树和一个 $1 \sim n$ 的排列 p，初始结点 k 上有一个棋子。
- Alice 和 Bob 正在博弈。Alice 需要将 p 排序，而 Bob 需要阻止 Alice。
- 每一轮，Alice 可以选择 p 的两个位置交换，然后 Bob 将棋子移动到一个相邻结点。交换时，Alice 不能选择其上的值等于棋子所在结点标号的位置。
- 问 Alice 能否达成目标。$3 \le n \le 2 \times 10^5$

CF1659F

直观上看，Alice 的胜率应该远高于 Bob，因为 Bob 每一轮只能 ban 掉一个数。

注意到 Alice 很多时候只需一轮就可以将一个数归位。假设存在三个数没有归位，显然 Alice 一定能找到其中的一个一步归位。于是，最终一定最多剩下一对 x, y 未归位，满足 $p_x = y, p_y = x$，且 Alice 不能直接交换它们。

不妨设此时标记在 x 上。手玩几个小数据发现，只要图不是菊花图，Alice 都有办法交换 x, y。

首先，如果 x, y 间的距离为偶数，整棵树可以被黑白染色，且 x, y 均为白色。此时，只要交换任意一对黑点，再交换 x, y，再把黑点换回来即可。

CF1659F

类似的，如果 x, y 不是邻居，我们可以选择 y 的一个邻居 z，先交换 y, z，再交换 x, y，则 $p_x = z, p_z = x$。由于 x, y 距离 ≥ 3 为奇数，这些操作一定可以进行，且 x, z 距离为偶数，套用偶数情况即可。

x, y 相邻时，同样找一个 z，先交换 y, z，再交换 x, z。发现只要 z 与 x 或 y 的距离 ≥ 3 ，就一定可以操作，并转化为上述情形。

这样的 z 不存在时，取 x 的邻居 y'，交换 y, y' ，此时 Bob 只能移到 y'。再交换 x, y，转为 $p_x = y', p_{y'} = x$ ，此时 z 一定存在。

接着来看菊花图，假设 Alice 不能在第一轮就直接获胜。

设中心结点为 x，若棋子在 x 处时 x 未归位，则 Bob 胜，因为 Bob 可以始终避免 x 归位。

CF1659F

若 x 已归位，Alice 显然不会再动 x，不妨设此时棋子也在 x 上。

回忆一个经典结论：交换一个排列中的任意两个数必定使置换环数改变 1。

考虑 $n-$ 置换环数，记为 c，若 c 为偶数，Bob 必胜，因为 Alice 操作后 c 为奇数。当 $c=1$ 时，Bob 永远可以移动到置换环上的一位，阻止 c 变为 0 。

反之，Alice 一定获胜，因为 $c=1$ 时 Alice 可以直接操作，而 $c \geq 3$ 时存在至少四个错位元素。Alice 接下来两轮都可以让一个元素复位，使 c 减少 2，且还是奇数。

显然棋子最多一轮后就会在 x 上，增加一些小讨论即可。时间复杂度 $O(n)$