

# 解谜游戏 题解

CTT 2022.12.7

# 题目大意

- 交互库有一个长度为 $n$ 的排列
- 你每次可以猜一个长度为 $n$ 的排列，交互库返回你猜对了多少个位置
- 使用尽可能少的猜测次数猜出排列
- $n \leq 1000$ ，猜测次数不超过9500

# 得分情况 & 选手交流

- 出题人std的实现：平均约9285次，标准差约35
- 另外还实现了一些诸如89分、76分、61分不等的算法
- 验题人设计了另一个能获得89分的算法
- 有一众AC选手平均在8800~9300之间
- 有2个人干翻了std，最优解为平均约7600次

# sol

- 不妨假设排列是随机的
- 如果不是，你可以自己复合上一个随机排列

# sol

- 直接暴力
- 基于决策树的暴力
- 搜索剪枝
- 爬山

sol

- 核心思想：当 $n$ 足够大时，错排（随机生成一个排列，与原排列一个相同元素也没有）的概率约为 $1/e$

# sol

- 一个正经的多项式级别做法：
- 考虑每次确定一个位置的数
- 如何检验 $x$ 位置上的数是否为 $y$ ？
- 可以将 $x$ 位置固定为 $y$ ，剩余位置随机打乱，当打乱次数足够多时，如果还没有发生错排，就可以相信 $x$ 位置确实是 $y$ 。
- 这个“足够多”要自己把握
- 枚举 $x$ 和 $y$ 进行上述判断，“足够多”可以设得大一些（因为只有 $n$ 次命中，其余平均 $e$ 次就会结束）
- 再加一些精细实现可以做到20分

# sol

- 实际上， $O(n^2)$ 次询问有一个不需要基于随机的做法：
- 尝试如下排列 $(0,1,\dots,n-1)$ 、 $(1,0,2,\dots,n-1)$ 、 $(2,1,0,3,\dots,n-1)$ 、 $\dots$ 、 $(n-1,1,2,\dots,n-2,0)$ ，即把0分别和每个位置交换。
- 有两种可能：
  - 1、某一次的答案比其他都大，设为第 $x$ 次（ $x$ 可能为0），那么0一定在 $x$ 位置；
  - 2、某两次的答案相同且比其他都大，设为 $x$ 和 $y$ ，则0、 $x$ 、 $y$ 三个数一定不同，且0在 $x$ 位置、 $y$ 在0位置或相反，尝试一次以判断。
- 这样可以花 $n+1$ 次判断出一个或两个数的位置，实际能得30分

# sol

- 从“按住一个数”到“按住一半数”：考虑二分的思想
- 首先，尝试 $(0,1,2,\dots,n-1)$ 、 $(1,2,\dots,n-1,0)$ 、 $\dots$ 、 $(n-1,0,1,2,\dots,n-2)$
- 从以上每一个答案非0的排列分别出发，尝试寻找正确的位置
- 设初始排列中有 $d(d \geq 1)$ 个数的位置是对的
- 按住左半边的所有数，每次把右半边随机打乱
- 于是，在每次随机中，右半边“存在正确元素”的概率不超过 $1-1/e$ ，实际上比这要低，因为右半边元素的真实位置并不一定还在右半边

# sol

- 随机打乱中，如果出现某一次询问的答案小于 $d$ ，这说明最开始右半边一定是存在正确元素的，就递归到右半边；否则，如果在进行足够多次的打乱后仍然得不到比 $d$ 小的答案，就有理由相信原先右半边是没有正确元素的，就可以递归到左半边，如此直到确定一个元素为止。
- 注意最后区间较小时建议改为暴力以避免边界情况
- 问题还是一样的：“足够多”该如何把握
- 由于二分要做 $O(n \log n)$ 次，每次大概需要尝试 $O(\log n)$ 次才能较为稳妥，这样反而可能不如 $O(n^2)$ 做法。

# sol

- 更高明的想法：二分时轮流按住一侧不动，随机打乱另一侧。
- 由于总有一侧是有正确元素的，因此无需对于“足够多”的把握，只需要一直随机到某一次询问答案 $< d$ 为止即可
- 这一过程只需要常数次随机打乱（期望不超过 $2e$ 次），因此总询问次数 $O(n \log n)$ 。
- 出题人的实现中，这个做法大约需要15500次，可以获得61分。
- 这一方法距离满分还差一些细节上的优化，可供优化的方向很多，以下为std的一个思路。

# sol

- 优化1：
- 充分利用已被确定的元素
- 为了快速使得某一次询问的答案减小，其实是希望打乱过程中尽量避免有正确的元素出现
- 用一些已经被确定的元素去替换被打乱侧的元素，可以使得打乱后出现正确的概率更小，从而减少随机次数
- 出题人的实现中，这个做法大约需要13600次，可以获得76分。

# sol

- 优化2:
- 当已确定的元素相对于区间长度足够多时，其实就没必要随机了
- 设已确定的元素数量为 $b$ ，当前区间长度为 $a$ ，如果 $b \geq a/2$ ，可以直接变成真二分（用已确定的元素替换一侧的所有数）。
- 出题人的实现中，这个做法大约需要9550次，可以获得89分。

# sol

- 优化3:
- 设已确定的元素数量为 $b$ ，当前区间长度为 $a$
- 实际上，可以确定一个常数 $k$ ，一旦 $b \geq ka$ 就去掉随机打乱，而是用已确定元素替换区间中的元素
- 当 $b < a/2$ 时，这可能造成两个递归子区间不等长，不过只要选取合适的 $k$ ，相比于随机打乱而言就会更优
- 经测试， $k=0.1 \sim 0.2$ 之间均为不错的选择，此时平均大约需要9290次询问，可以获得100分。

# sol

- 还有很多可供优化的思路，就不一一赘述。
- 目前来看，9500的上界还是挺宽松的，一个实现优秀的算法可以达到9000次以内。

THANKS

祝大家身体健康