

问题描述（100 囚犯抽签问题）

100 名囚犯编号为 1 至 100。监狱长准备一个房间，内有 100 个盒子，每个盒子内随机放入一张囚犯编号的纸条（编号不重复）。囚犯依次进入房间，每人可打开最多 50 个盒子寻找自己的编号。若所有囚犯均在 50 次尝试内找到自己的编号，则全体获释；否则全员失败。原问题出自 2003 年《American Mathematical Monthly》

关键点：

1. 囚犯不能交流或修改盒子内容。
2. 每个囚犯的搜索策略影响整体成功率。

算法说明：

随机搜索策略：

随机策略是最直观的方法，每个囚犯独立地随机选择 50 个盒子打开：

核心逻辑：从 100 个盒子中无放回地随机选取 50 个

数学本质：每个囚犯成功概率为 50%，全体成功概率为 $(1/2)^{100} \approx 0$

代码实现：使用 `np.random.choice` 生成随机选择序列

循环搜索策略：

循环策略利用排列的循环结构，是问题的关键突破点：

核心逻辑：囚犯从与自己编号相同的盒子开始，根据盒内纸条编号跳转到下一个盒子，形成循环

成功条件：当且仅当包含该囚犯的循环长度不超过 50

数学原理：排列中最长循环长度 $\leq K$ 的概率，理论上约为 31%

代码实现：通过迭代跳转盒子，记录尝试次数

实验设计：

基本参数设置：

囚犯数量 N : 100（默认）

最大尝试次数 K : 50（默认，即 $N/2$ ）

模拟轮次 T : 10000 次

对比指标：全体成功率、成功囚犯数量分布

仿真流程：

生成 100 个盒子的随机排列（0-99 的随机置换）

对每个囚犯应用指定策略进行搜索

记录单轮实验中成功的囚犯数量

统计全体成功的轮次占比

重复 T 次实验，计算成功率及相关统计量

关键技术实现

排列生成：使用 `np.random.permutation` 确保无重复

向量化操作：利用 NumPy 数组运算提高效率

循环检测：通过标记访问状态分析排列的循环结构

实验结果：

主要发现：

1. 随机策略成功率: 0.00e+00，几乎不可能成功
2. 循环策略成功率: 0.304，约为 30.4%
3. 循环策略的关键洞察：成功当且仅当最长循环长度 ≤ 50

4. 理论成功率 = $P(\text{最长循环} \leq 50) \approx 0.312$

数学原理:

随机策略数学原理:

每个囚犯成功概率为 $p = K/N = 0.5$, 全体成功概率为: $P(\text{全体成功}) = \left(\frac{K}{N}\right)^N = \left(\frac{1}{2}\right)^{100} \approx 0$

这是指数级衰减的概率, 实际中几乎不可能成功。

循环策略数学原理

排列中长度为 l 的循环出现的概率为 $(1/l)$, 最长循环长度 ($> K$) 的概率为: $P(\text{最长循环}$

$$> K) = \sum_{l=K+1}^N \frac{1}{l}$$

当 $N=100, K=50$ 时: $P(\text{最长循环} > 50) = \sum_{l=51}^{100} \frac{1}{l} \approx 0.6882$

因此成功概率为: $P(\text{全体成功}) = 1 - \sum_{l=51}^{100} \frac{1}{l} \approx 0.3118$