实验 2 报告

一、算法说明

1. 核心功能

程序通过模拟多轮实验，统计两种策略下囚犯全员在指定尝试次数内找到自己编号的成功率。具体功能包括：

- 盒子初始化：生成随机排列的盒子（每个盒子存放一个唯一的囚犯编号）。
- 策略模拟：实现两种搜索策略（随机搜索、循环策略），模拟囚犯寻找自己编号的过程。
- 成功率统计：通过多轮实验（用户可指定实验轮次），计算两种策略的总成功率并对比。

2. 关键算法

盒子初始化（box_init 函数）

功能：生成一个随机排列的盒子列表，模拟监狱长随机放置囚犯编号的过程。、

实现逻辑：

输入参数 num 为囚犯数量（如 N=100）。

生成一个初始列表 boxes，元素为 0 到 num-1（对应囚犯编号）。

使用 random.shuffle 随机打乱列表，使每个盒子的编号排列完全随机。

随机搜索策略（random_strategy 函数）

功能：模拟“随机搜索”策略下的逃生成功率。每个囚犯随机选择最多 tries 个盒子，检查是否包含自己的编号。

核心逻辑：

输入参数：囚犯数量 num_prisoners、盒子列表 boxes、尝试次数 tries、实验轮次 epoches。

单轮实验流程：

所有囚犯独立随机选择 tries 个盒子（使用 random.sample 确保不重复选择）。

每个囚犯检查选中的盒子中是否有自己的编号（即 boxes[index] == prisoner）。

若所有囚犯均找到自己的编号，该轮实验成功；否则失败。

统计：累计 epoches 轮中成功的轮次，计算成功率（成功轮次 / 总轮次）。

循环搜索策略（cycle_strategy 函数）

功能：模拟“循环搜索”策略下的逃生成功率。囚犯从自己编号对应的盒子开始，按盒内纸条编号跳转，直到找到自己的编号或超过尝试次数。

输入参数：囚犯数量 num_prisoners、盒子列表 boxes、尝试次数 tries、实验轮次 epoches。

核心逻辑：

输入参数：同 random_strategy。

单轮实验流程：

每个囚犯 prisoner 从自己编号对应的盒子（索引为 prisoner）开始搜索。

检查当前盒子中的编号：若等于 prisoner（找到自己的编号），则成功；否则跳转到该编号对应的盒子（即 box_index = boxes[box_index]）。

重复步骤 2，最多尝试 tries 次；若超过次数未找到，则该囚犯失败，整轮实验失败。

统计：累计 epoches 轮中成功的轮次，计算成功率。

二、 实验结果与分析

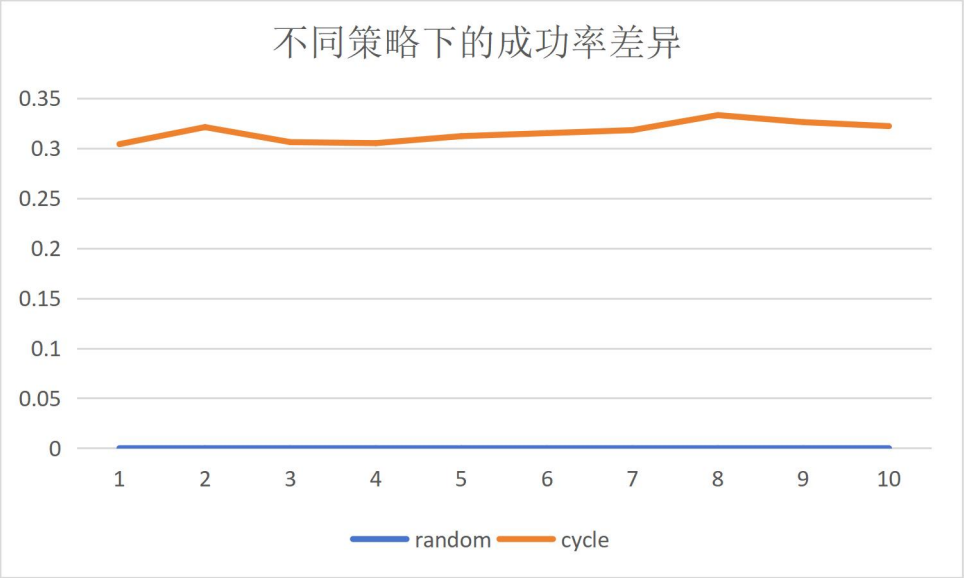
N=100，K=50，T=10000 时：

```
input the number of prisoners100
input the total tries for each prisoner50
input the total epoches for all prisoners10000
正在模拟 100 名囚犯，每人尝试 50 次，共进行 10000 轮实验...
随机策略的成功率为 0.000000
循环策略的成功率为 0.312500
```

N=50, K=25, T=10000 时：

```
input the number of prisoners50
input the total tries for each prisoner25
input the total epoches for all prisoners10000
正在模拟 50 名囚犯，每人尝试 25 次，共进行 10000 轮实验...
随机策略的成功率为 0.000000
循环策略的成功率为 0.312900
```

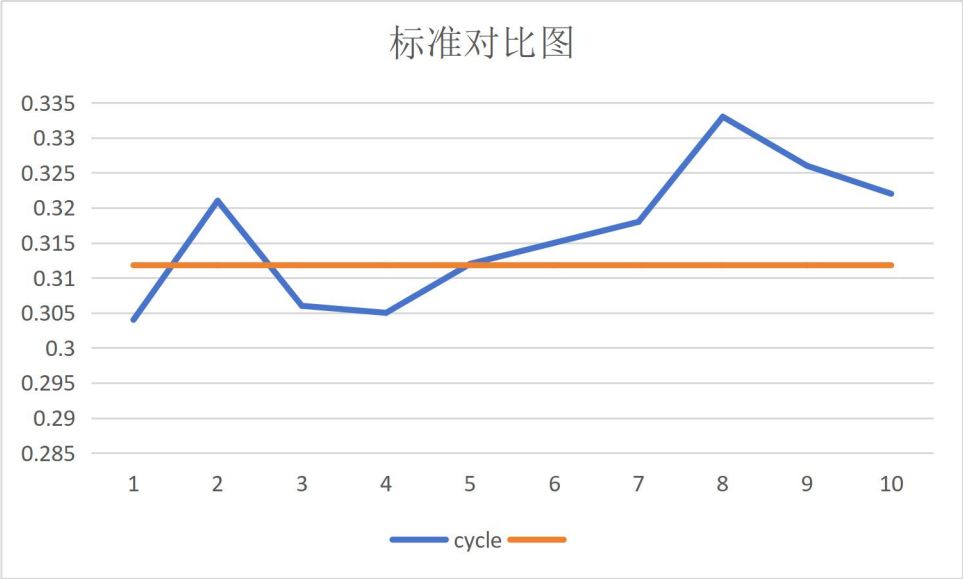
10 次模拟实验下，成功率对比图



三、 理论计算

随机策略：P=$\left(\frac{1}{2}\right)^{100}$

循环策略：设置换π存在长度大于 50 的轮换σ，其长度为 l。则σ 中元素有$\binom{100}{l}$种可能的选择。选定 l 个元素后，可以形成$(l-1)!$种不同的轮换。所以选自 1~100 的、长为 l 的轮换个数是：$\binom{100}{l}(l-1)!$而剩余的个元素可以形成$(100-l)!$种不同的置换。因此这样的置换一共有$\binom{100}{l}(l-1)!\,(100-l)! = \frac{1}{l} * 100!$个。因犯遇到这样的置换的概率是 $\mathrm{P}(\bar{\mathrm{A}}) = \frac{1}{100!}\sum_{l=51}^{100}\left(\frac{1}{l} * 100!\right) = 0.6882$。则成功概率是：$\mathrm{P}(\mathrm{A}) = 1 - \mathrm{P}(\bar{\mathrm{A}}) = 0.3118$



四、 优化思路

在计算速度上，发现当 epoch 很大时（例如 1000000），计算机将花费较长时间进行模拟计算。可以采用并行计算方式，将囚犯分组，分别进行实验。此方法可以减少计算时间。