QTA 笔面试刷题 Week 5 解析

QTA

December 2025

目录

1 相关系数为 ρ 的取值范围

1.1 题目描述

设随机变量 X,Y,Z 两两相关系数均为 ρ，相关矩阵

$$R=\begin{pmatrix}1&\rho&\rho\\ \rho&1&\rho\\ \rho&\rho&1\end{pmatrix}$$

要求矩阵半正定，求 ρ 的范围。

1.2 解答与推导

R 可写成 $R=(1-\rho)I+\rho J$，其中 I 为单位阵、J 为全 1 阵。全 1 阵在三维下的特征值是 3（重数 1）和 0（重数 2），因此

$$\begin{aligned}\lambda_1&=(1-\rho)+3\rho=1+2\rho,\\ \lambda_2&=1-\rho,\\ \lambda_3&=1-\rho。\end{aligned}$$

半正定要求所有特征值非负，于是 $1+2\rho\geq 0$ 且 $1-\rho\geq 0$，得到 $-\frac{1}{2}\leq\rho\leq 1$。

结论： $\rho\in[-\frac{1}{2},1]$。

2 连续 6 个 6 的期望投掷次数

2.1 题目描述

公平骰子掷到连续 6 个 6 的期望投掷次数。

2.2 解答与推导

设 E_n 为达成连续 n 个 6 的期望掷数。已连到 $n-1$ 个 6 时再掷一次，先付出 1 次尝试；若掷出 6（概率 1/6）则直接完成，若不是 6（概率 5/6）连击清零，又要重新经历期望为 E_n 的过程。全期望给出

$$E_n=E_{n-1}+1+\frac{5}{6}E_n\quad\Rightarrow\quad E_n=6E_{n-1}+6,$$

且 $E_0=0$。递推可得 $E_n=\sum_{i=1}^{n}6^i$，于是

$$E_6=\frac{6(1-6^6)}{1-6}=55986.$$

结论： 期望掷数为 55986。

3 随机落座（疯子坐飞机）

3.1 题目描述

第 1 人随机就座，其余人如果自己的座位空则入座，否则随机。求第 N 人坐对的概率。

3.2 解答与推导

混乱只在第 1 人被迫随机时出现，此后每当有人随机就座，空位集合只会在“1 号座位”和“N 号座位”之间传递，直到其中之一被选中才结束：若先被选中的是 1 号，则后续人人归位，N 人坐对；若先被选中的是 N 号，则 N 人坐错；落在其他编号只是在把同样的局面递给下一个人。1 号与 N 号被率先选中的概率全程对称，因此各为 $\frac{1}{2}$。

结论： 第 N 个人坐对的概率为 0.5。

4 外星生物颜色

4.1 题目描述

初始红 20、绿 21、蓝 22。规则：两种不同颜色变为两个第三种颜色。问能否最终全为同一种颜色。

4.2 解答与推导

设三色数量为 n_1, n_2, n_3。一次操作让其中两色各减 1、第三色加 2，模 3 等价于所有颜色数同时加 2，因此任意两色数量差在模 3 下保持不变。初始

$$n_{\text{红}} \equiv 2, \quad n_{\text{绿}} \equiv 0, \quad n_{\text{蓝}} \equiv 1 \pmod 3,$$

两两之差模 3 分别为 $2, 1, 1$，都不为 0。若最终全为同色则任意差为 0（模 3 也为 0），与不变量冲突，故无法实现。

结论： 不可能变成同一种颜色。

5 比较排序的时间复杂度下界

5.1 题目描述

证明任意基于比较的排序算法，最坏情况时间复杂度下界为 $O(n \log n)$。

5.2 证明

用决策树刻画比较排序，对 n 个元素共有 $n!$ 种排列，必须对应 $n!$ 个叶子。二叉决策树高度为 h 时叶子最多 2^h 个，因此需 $2^h \geq n!$，得 $h \geq \log_2(n!)$。由斯特林公式 $\ln(n!) \approx n \ln n - n$ 可知 $\log_2(n!)$ 量级为 $n \log n$，故最坏情况比较次数下界为 $\Omega(n \log n)$。

结论： 最坏时间下界 $\Omega(n \log n)$。

6 交易逆序对总数

6.1 题目描述

给定股票价格序列，若前一天价格高于后一天则构成逆序对。$N \leq 50000$，求逆序对总数。

6.2 解答与算法

暴力为 $O(N^2)$，需用 $O(N\log N)$。在归并排序的合并阶段，左右子数组有序；当遇到左侧元素 $L[i] > R[j]$，则 $L[i]$ 以及其后的 $(\mathrm{mid} - i + 1)$ 个元素都大于 $R[j]$，一次性贡献同样数量的逆序对，然后写入 $R[j]$ 继续归并。累积计数即可得到总数。

6.3 代码实现 (Python)

```python
class Solution:
    def reversePairs(self, record: list[int]) -> int:
        self.count = 0
        self.merge_sort(record, 0, len(record) - 1)
        return self.count

    def merge_sort(self, nums, left, right):
        if left >= right:
            return
        mid = (left + right) // 2
        self.merge_sort(nums, left, mid)
        self.merge_sort(nums, mid + 1, right)
        self.merge(nums, left, mid, right)

    def merge(self, nums, left, mid, right):
        temp = []
        i, j = left, mid + 1
        while i <= mid and j <= right:
            if nums[i] <= nums[j]:
                temp.append(nums[i]); i += 1
            else:
                self.count += (mid - i + 1)
                temp.append(nums[j]); j += 1
        while i <= mid:
            temp.append(nums[i]); i += 1
        while j <= right:
            temp.append(nums[j]); j += 1
        for k, v in enumerate(temp):
            nums[left + k] = v
```