QTA 笔面试刷题 week1-答案

欧岱松

2025 年 11 月 29 日

1. **简单的抽球问题**

   题目： 一个袋子里有 4 个红球、3 个蓝球和 2 个黄球。随机不放回抽 3 个球，请问抽到的 3 个球至少有两种不同颜色的概率有多少?

   解法： 补集法。总共 $\binom{9}{3} = 84$ 种抽法。只有一种颜色：3 个红球 $\binom{4}{3} = 4$ 种，3 个蓝球 $\binom{3}{3} = 1$ 种，共 5 种。

   答案： $P = 1 - \dfrac{5}{84} = \dfrac{79}{84}$

2. **切线段的极限和**

   题目： 一条长度为 1 的线段，随机在中间切一刀，得到两部分 x 和 y。把它们相乘并加到和里。再把 x 和 y 各自随机切一刀，如此重复操作。请问这个和的极限值是多少?

   解法： 设 $E(L)$ 为长度为 L 的线段切分后的乘积和期望值。根据递归定义，总期望等于当前切分收益 $x(L-x)$ 加上剩余两段的期望收益在 $[0, L]$ 上的平均值，即满足方程：

$$E(L) = \frac{1}{L}\int_0^L [x(L-x) + E(x) + E(L-x)]\, dx \tag{1}$$

   根据量纲分析设 $E(L) = kL^2$，利用积分对称性将通解代入方程，可得：

$$kL^2 = \frac{1}{L}\left[\int_0^L (xL - x^2)dx + 2\int_0^L kx^2 dx\right]$$

   计算积分项得 $\frac{L^3}{6}$ 与 $\frac{2kL^3}{3}$，代入化简得 $k = \frac{1}{6} + \frac{2}{3}k$，解得 $k = \frac{1}{2}$。因此当 $L = 1$ 时，极限值为 $E(1) = \frac{1}{2}$。

3. **年久失修的密码锁**

   题目： 一个 3 位的密码锁，年久失修功能异常，只需要输入任意两位密码就能打开。最少需要尝试多少次?

   解法： 针对三位密码锁只需匹配任意两位即可打开的，其尝试次数的实际下界分析需从理论体积极限出发。首先，由于总状态空间为 10^3，而单次尝试，能覆盖自身及三个维度上

各改变一位的 9 个状态，单次总覆盖体积为 $1+9+9+9=28$；理论上若假设这些覆盖区域完美填充空间，则所需最小次数为 $1000/28 \approx 35.71$，即至少 36 次。然而，因为在 $10\times10\times10$ 的离散网格中，固定的"十字形"覆盖域无法在不发生重叠的情况下完美拼合。所以需要可以找到一些实际可以构造的下界。

为了构建一个能够 100% 覆盖所有可能密码的确定性方案，我们的构造将总尝试集合定义为 6 个子集 S_k 的并集，总计 60 次尝试。对于每个子集，通过遍历首位数字 $x \in \{0,\ldots,9\}$，并依据同余方程：

$$y \equiv (x+2k) \pmod{10} \tag{2}$$

$$z \equiv (x+k) \pmod{10} \tag{3}$$

确定后续两位，生成 10 个互不干涉的空间对角线码字。核心在于参数 k 的选取——我们严格选定：

$$k \in \{0,1,2,3,5,7\}$$

这组特定的参数巧妙利用了三维空间中 xy、xz 及 yz 平面投影差值的互补性，确保了无论目标密码的三位数字间距为何，都能落入覆盖网中。

覆盖原理简述

其覆盖的完备性源于差值集合的无缝拼接：

- $y-x$ 的差值（即 $2k \pmod{10}$）覆盖了偶数集合 $\{0,2,4,6\}$；
- $z-x$ 的差值（即 $k \pmod{10}$）覆盖了奇数集合 $\{1,3,5,7\}$；
- $z-y$ 的差值（即 $-k \pmod{10}$）补足了高位差值 $\{8,9\}$。

三者之并集恰好为 $\{0,\ldots,9\}$ 的全集，从而在数学上证明了覆盖的必然性

4. **把守圆圈**

题目： 有一个 100 米半径的圆。超人站在圆心，速度为 1 米/秒。蝙蝠侠站在圆周上某一点。问：蝙蝠侠速度最低是什么值，才能使得超人无法逃出这个圆?

解法： 狮子与人问题。超人最优策略分两阶段：

(1) 螺旋移动到半径 $r=\frac{100}{v_b}$ 处，使双方角速度相等

(2) 从半径 r 处直线冲向圆周（选择与蝙蝠侠相反方向）

临界条件：超人冲刺距离 $100-\frac{100}{v_b}$ = 蝙蝠侠跑半圆时间 $\frac{100\pi}{v_b}$

解得：$v_b = 1+\pi$

答案： $v_b = 1+\pi \approx 4.14$ 米/秒

5. **快速排序的性质（多选）**

题目： 以下是快速排序的特性有：

A. 第一趟排序后，所有元素都在其最终位置上

B. 最坏情况时间复杂度为 $O(n^2)$

C. 是不稳定的排序算法

D. 空间复杂度为 $O(\log n)$ 到 $O(n)$ 之间

解法：

A 错误：只有 pivot 元素在最终位置，其他元素不一定

B 正确：当每次 pivot 都是极值时（如已排序数组），递归深度为 n，时间 $O(n^2)$

C 正确：长距离交换可能改变相同元素的相对顺序，因此不稳定

D 正确：递归调用栈深度：最好 $O(\log n)$（平衡），最坏 $O(n)$（极度不平衡）

答案：BCD

6. **Expected Distinct Numbers**

题目： Say you have n numbers $\{1, 2, \ldots, n\}$, and you uniformly sample from this distribution with replacement n times. What is the expected number of distinct values you would draw?

解法： 指示随机变量法。定义 $X_i = 1$ 当且仅当数字 i 至少出现一次。不同数字总数 $X = \sum_{i=1}^{n} X_i$。

计算：$P(X_i = 1) = 1 - \left(1 - \frac{1}{n}\right)^n$

由期望的线性性：$\mathbb{E}[X] = n \cdot \left[1 - \left(1 - \frac{1}{n}\right)^n\right]$

当 $n \to \infty$ 时，$\left(1 - \frac{1}{n}\right)^n \to \frac{1}{e}$

答案： $n\left[1 - \left(1 - \frac{1}{n}\right)^n\right] \approx 0.632n$

7. **Arithmetic Subarrays**

题目： Given an integer array `nums`, return the number of arithmetic subarrays of `nums`. (等差子数组：至少 3 个元素，相邻元素差相等)

解法： 动态规划。定义 $dp[i] =$ 以 $\texttt{nums}[i]$ 结尾的等差子数组个数

状态转移：

$$dp[i] = \begin{cases} dp[i-1] + 1, & \text{if } \texttt{nums}[i] - \texttt{nums}[i-1] = \texttt{nums}[i-1] - \texttt{nums}[i-2] \\ 0, & \text{otherwise} \end{cases}$$

Python 代码:

```python
def numberOfArithmeticSlices(nums):
    n = len(nums)
    if n < 3:
        return 0

    total = 0
    dp = 0

    for i in range(2, n):
        if nums[i] - nums[i-1] == nums[i-1] - nums[i-2]:
            dp += 1
            total += dp
        else:
            dp = 0

    return total
```