QTA 笔面试刷题 week2-题目

背景说明

在金融交易中，我们监控一个二元信号流 $S = \{x_1, x_2, ...\}$，其中 x_i 独立服从伯努利分布 $x_i \sim \text{Bernoulli}(0.5)$（即 $P(H) = P(T) = 0.5$）。

问题一

假设我们需要检测长度为 2 的短信号。我们定义以下随机变量：

- N_{HH}：序列中首次连续出现两个" 正面"（HH）所需的观测次数。
- N_{HT}：序列中首次出现" 正面" 紧接" 反面"（HT）所需的观测次数。

问题： 请分别计算期望值 $\mathbb{E}[N_{HH}]$ 和 $\mathbb{E}[N_{HT}]$。

问题二

现在考虑一个长度为 L 的任意目标序列 pattern $A = (a_1, a_2, \ldots, a_L)$。为了求解 $\mathbb{E}[N_A]$，我们引入一个**公平赌场模型（Fair Casino Model）**：

> 假设每一时刻 $t = 1, 2, \ldots$ 都有一个新的赌徒进场。第 t 个赌徒在进场时押注 1 美元，赌 $x_t = a_1$。如果是公平赌局（赔率为 1:1），赢了变成 2 美元，输了变成 0。如果赢了，他将 2 美元全押在下一时刻 $x_{t+1} = a_2$ 上，以此类推。一旦输掉任何一次，赌徒破产离开。如果连续赢了 L 次（即匹配了整个序列 A），赌场关闭。

提示 (Hint)

利用鞅（Martingale）的性质，可以证明对于任意长度为 L 的二进制序列 A，其首次出现的期望时间公式为：

$$\mathbb{E}[N_A] = \sum_{k=1}^{L} \delta_k \cdot 2^k$$

其中 δ_k 是一个指示函数：当序列 A 的长度为 k 的**前缀**同时也是它的**后缀**时，$\delta_k = 1$，否则 $\delta_k = 0$。

问题：利用上述公式，快速计算模式 **HTHH** 的期望等待时间，并可以简单写出上述 hint 的证明思路。

问题三

现在有两个交易员 Alice 和 Bob 分别选择不同的模式进行" 抢跑" 游戏。一旦信号流中出现了某人的模式，该人获胜，游戏结束。

- Alice 选择了模式 $A = \text{H H}$。
- Bob 选择了模式 $B = \text{T H}$。

问题：请问 Bob 获胜的概率（即 B 先于 A 出现的概率）是多少?

*（注意：这里不再是计算"期望时间"，而是计算**获胜概率**。）*

问题四

一个硬币 0.6 概率为正面，0.4 概率为反面，如何构造一个发生概率为 20% 的随机事件?

问题五

1,0,10,2,4,24 能组成多少个不同的八位数?

问题六

阅读以下 Python 代码，推断最后一行 `print` 语句的输出结果是哪一项?

代码片段

```python
from functools import wraps

def token_check(func):
    """Decorator to check user tokens."""
    @wraps(func)
    def inner_check(*args, **kwargs):
        return func(*args, **kwargs)
    return inner_check

def log_execution(func):
    """Decorator to log function calls."""
    # 注意: 此处故意未使用 @wraps
    def wrapper_log(*args, **kwargs):
        """Wrapper for logging."""
        return func(*args, **kwargs)
    return wrapper_log

@log_execution
@token_check
def fetch_data():
    """Fetches data from database."""
    pass

print(f"{fetch_data.__name__} | {fetch_data.__doc__}")
```

选项:

A. `fetch_data | Fetches data from database.`

B. `inner_check | Fetches data from database.`

C. `wrapper_log | Wrapper for logging.`

D. `wrapper_log | Fetches data from database.`

问题七

题目背景

你正在为一个高频交易系统编写回测引擎。给定一个整数数组 `prices`，其中 prices$[i]$ 表示某支股票在第 i 天的价格。
你需要设计一个算法来计算最大利润，并严格遵守交易所的结算制度与佣金模型。

题目描述

请计算在满足以下所有规则的情况下，你能获得的最大利润：

1. **无限次交易**：你可以尽可能多地进行买卖交易（买入和卖出），但你必须在再次买入前卖出掉之前的股票（即手中最多只能持有一股）。
2. **交易成本**：每次**卖出**股票时，需要支付一笔固定的手续费 `fee`。
3. **结算延迟（冷却期）**：卖出股票后，需要等待 D 天的资金结算期才能再次买入。
   - 例如：如果你在第 i 天卖出股票，那么你在第 $i+1,...,i+D$ 天都无法买入股票。
   - 你最早可以在第 $i+D+1$ 天再次买入。

输入输出示例

示例 1:

Input: `prices = [1, 2, 3, 0, 2]`, `D = 1`, `fee = 0`
Output: `3`

Explanation:

- 第 0 天买入 (price 1)，第 1 天卖出 (price 2)，利润 $= 2 - 1 - 0 = 1$。
- 第 2 天处于冷冻期 $(D = 1)$，无法操作。
- 第 3 天买入 (price 0)，第 4 天卖出 (price 2)，利润 $= 2 - 0 - 0 = 2$。
- 总利润 $= 1 + 2 = 3$。

示例 2:

Input: `prices = [1, 5, 2, 4, 3]`, `D = 2`, `fee = 1`
Output: `3`

Explanation:

- 第 0 天买入 (price 1)，第 1 天卖出 (price 5)。
- 支付 fee = 1，本次净赚 $5 - 1 - 1 = 3$。
- 卖出后进入 2 天冷却（第 2、3 天禁买）。
- 之后无更优操作，总利润 3。

约束条件

- $1 \leq \text{prices.length} \leq 5 \times 10^4$
- $0 \leq \text{prices}[i] \leq 5 \times 10^4$
- $0 \leq D \leq \text{prices.length}$
- $0 \leq \text{fee} \leq 5 \times 10^4$