Java方向编程题答案

day35

[编程题]23742-百万富翁问题

https://www.nowcoder.com/questionTerminal/9fe25b6cf93e46dcb09ba67aeef2c4cc

【题目解析】：

题目非常简单

【解题思路】：

思路也很简单，重点想给大家演示的是，能用 O(1) 一步计算解决的问题，就不要 O(n) 的时间复杂度去计算

【示例代码】：

```
import java.lang.Math;
public class Main{
    public static void main(String[] args){
        long sumRich = 30 * 10;
        long sumStranger = (1L << 30) - 1;
        System.out.print(sumRich + " " + sumStranger);
    }
}
```

[编程题]23496-风口的猪-中国牛市

https://www.nowcoder.com/questionTerminal/9370d298b8894f48b523931d40a9a4aa

【题目解析】：

题目比较容易懂

【解题思路】：

重点还是在求解思路上，这同样是道动态规划的题目。对照代码去反推思路其实还是比较容易明白的

但一定要掌握动态规划的推导思路。如何找到状态，如何确定状态转移方程。个人比较喜欢的方式是从后往前推导的方式。

【示例代码】：

```
public class Solution {
    public static int calculateMax(int[] prices) {
        int firstBuy = Integer.MAX_VALUE;  // 第一次买入最好的价格，越低越好
        int firstSell = 0;                 // 第一次卖出后的最高收益，越高越好
        int secondBuy = Integer.MIN_VALUE; // 第二次买入时还剩余的最高收益，越高越好
        int secondSell = 0;                // 第二次卖出时总的最高收益，越高越好

        for (int price : prices) {

            // 当前价格下第一次买入的价格
```

```
            firstBuy = Math.min(firstBuy, price);
            // 当前价格 - 买入价格 就是当前价格下第一次买卖的收益
            firstSell = Math.max(firstSell, price - firstBuy);
            // 第一次卖出的收益 - 当前价格，即当前价格下，第二次买入后还剩余收益
            secondBuy = Math.max(secondBuy, firstSell - price);
            // 剩余收益 + 当前价格，即当前价格下，第二次买卖的收益
            secondSell = Math.max(secondSell, secondBuy + price);
        }

        return secondSell;
    }
}
```