309. 最佳买卖股票时机含冷冻期

算法 动态规划

> 题目： Best Time to Buy and Sell Stock with Cooldown
> 英文版链接：https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-cooldown
> 中文版链接：https://leetcode-cn.com/problems/best-time-to-buy-and-sell-stock-with-cooldown/

题目分析

举个例子：

```
Input: [1,2,3,0,2]
Output: 3
Explanation: transactions = [buy, sell, cooldown, buy, sell]
```

本题我们需要维护三个一维数组buy, sell，和rest。其中：
buy[i]表示在第i天之前最后一个操作是买，此时的最大收益。
sell[i]表示在第i天之前最后一个操作是卖，此时的最大收益。
rest[i]表示在第i天之前最后一个操作是冷冻期，此时的最大收益。

需要注意，i表示天数也就是时刻的变换。

根据这样我们很容易写出来这样的递推公式：

$$
\begin{aligned}
buy[i] &= max(rest[i-1] - price, buy[i-1]) \\
sell[i] &= max(buy[i-1] + price, sell[i-1]) \\
rest[i] &= max(sell[i-1], buy[i-1], rest[i-1])
\end{aligned}
$$

上述递推式很好的表示了在买之前有冷冻期，买之前要卖掉之前的股票。一个小技巧是如何保证[buy, rest, buy]的情况不会出现，这是由于$buy[i] \le rest[i]$， 即$rest[i] = max(sell[i-1], rest[i-1])$，这保证了[buy, rest, buy]不会出现。

另外，由于冷冻期的存在，我们可以得出$rest[i] = sell[i-1]$，这样，我们可以将上面三个递推式精简到两个：

$$
\begin{aligned}
buy[i] &= max(sell[i-2] - price, buy[i-1]) \\
sell[i] &= max(buy[i-1] + price, sell[i-1])
\end{aligned}
$$

我们还可以做进一步优化，由于i只依赖于i-1和i-2，所以我们可以在O(1)的空间复杂度完成算法。

答案

```python
class Solution:
    def maxProfit(self, prices) -> int:
        if len(prices) <= 1:
            return 0
        buy, pre_buy, sell, pre_sell = -max(prices), 0, 0, 0
        for price in prices:
            pre_buy = buy
            buy = max(pre_sell - price, pre_buy)
            pre_sell = sell
            sell = max(pre_buy + price, pre_sell)

        return sell
```