

# 122-买卖股票的最佳时机II

## 题述

122. 买卖股票的最佳时机 II

难度 中等  1647     

给定一个数组 `prices`，其中 `prices[i]` 表示股票第 `i` 天的价格。

在每一天，你可能会决定购买和/或出售股票。你在任何时候 **最多** 只能持有一股股票。你也可以购买它，然后在 **同一天** 出售。

返回 **你能获得的最大利润**。

示例 1:

输入: `prices = [7,1,5,3,6,4]`

输出: 7

解释: 在第 2 天（股票价格 = 1）的时候买入，在第 3 天（股票价格 = 5）的时候卖出，这笔交易所能获得利润 =  $5 - 1 = 4$ 。

随后，在第 4 天（股票价格 = 3）的时候买入，在第 5 天（股票价格 = 6）的时候卖出，这笔交易所能获得利润 =  $6 - 3 = 3$ 。

示例 2:

输入: `prices = [1,2,3,4,5]`

输出: 4

解释: 在第 1 天（股票价格 = 1）的时候买入，在第 5 天（股票价格 = 5）的时候卖出，这笔交易所能获得利润 =  $5 - 1 = 4$ 。

注意你不能在第 1 天和第 2 天接连购买股票，之后再将它们卖出。因为这样属于同时参与了多笔交易，你必须在再次购买前出售掉之前的股票。

示例 3:

输入: `prices = [7,6,4,3,1]`

输出: 0

解释: 在这种情况下，没有交易完成，所以最大利润为 0。

## 思路

这道题和上一道题的区别就在于可以买卖多次

思路还是动态规划，发生变化的只有递推公式，其他的都一样

重申一下dp数组的含义：

- $dp[i][0]$  表示第  $i$  天持有股票所得现金。
- $dp[i][1]$  表示第  $i$  天不持有股票所得最多现金

## $dp[i][0]$

如果第  $i$  天持有股票即  $dp[i][0]$ ，那么可以由两个状态推出来

- 第  $i-1$  天就持有股票，那么就保持现状，所得现金就是昨天持有股票的所得现金 即：  $dp[i-1][0]$ 
  - 这一点和上一题不同，是因为上一题股票全程只能买卖一次，所以上一题直接就是  $-prices[i]$ ，而本题可能会有之前买卖所得的利润
- 第  $i$  天买入股票，所得现金就是昨天不持有股票的所得现金减去 今天的股票价格 即：  $dp[i-1][1] - prices[i]$

## $dp[i][1]$

第  $i$  天不持有股票即  $dp[i][1]$  的情况，依然可以由两个状态推出来

- 第  $i-1$  天就不持有股票，那么就保持现状，所得现金就是昨天不持有股票的所得现金 即：  $dp[i-1][1]$
- 第  $i$  天卖出股票，所得现金就是按照今天股票佳价格卖出后所得现金即：  $prices[i] + dp[i-1][0]$

## 题解

---

### 动态规划

```
class Solution:
    def maxProfit(self, prices: List[int]) -> int:
        # 动态规划
        #  $dp[i][0] = \max(dp[i-1][0], dp[i-1][1] - prices[i])$ 
        #  $dp[i][1] = \max(dp[i-1][1], dp[i-1][0] + prices[i])$ 
        length = len(prices)
        dp = [[0] * 2 for i in range(length)]
        dp[0][0] = -prices[0]
        dp[0][1] = 0

        for i in range(1, length):
            dp[i][0] = max(dp[i-1][0], dp[i-1][1] - prices[i])
            dp[i][1] = max(dp[i-1][1], dp[i-1][0] + prices[i])
        return dp[-1][1]
```

## 思考

---

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	52 ms	18.2 MB	Python3	2022/04/08 18:16	▶
通过	56 ms	18 MB	Python3	2022/04/08 18:16	▶
通过	40 ms	15.8 MB	Python3	2022/02/08 10:17	▶
通过	4 ms	12.6 MB	C++	2022/02/08 10:02	▶