

0122. 买卖股票的最佳时机 II

👤 ITCharge ⌚ 大约 2 分钟

- 标签：贪心、数组、动态规划
- 难度：中等

题目链接

- [0122. 买卖股票的最佳时机 II - 力扣](#)

题目大意

描述：给定一个整数数组 `prices`，其中 `prices[i]` 表示某支股票第 `i` 天的价格。在每一天，你可以决定是否购买 / 出售股票。你在任何时候最多只能持有一股股票。你也可以先购买，然后在同一天出售。

要求：计算出能获取的最大利润。

说明：

- $1 \leq \text{prices.length} \leq 3 * 10^4$ 。
- $0 \leq \text{prices}[i] \leq 10^4$ 。

示例：

- 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 = $6 - 3 = 3$ 。

总利润为 $4 + 3 = 7$ 。

py

- 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` 。

总利润为 `4` 。

解题思路

思路 1: 贪心算法

股票买卖获取利润主要是看差价, 必然是低点买入, 高点卖出才会赚钱。而要想获取最大利润, 就要在跌入谷底的时候买入, 在涨到波峰的时候卖出利益才会最大化。所以我们购买股票的策略变为了:

1. 连续跌的时候不买。
2. 跌到最低点买入。
3. 涨到最高点卖出。

在这种策略下, 只要计算波峰和谷底的差值即可。而波峰和谷底的差值可以通过两两相减所得的差值来累加计算。

思路 1: 代码

```
class Solution:
    def maxProfit(self, prices: List[int]) -> int:
        ans = 0
        for i in range(1, len(prices)):
            ans += max(0, prices[i]-prices[i-1])
        return ans
```

思路 1: 复杂度分析

- 时间复杂度: $O(n)$, 其中 n 是数组 `prices` 的元素个数。
- 空间复杂度: $O(1)$ 。

