

1. 题目描述：

给定一个整数数组 `prices`，表示连续几天的股票价格。返回一个数组 `answer`，其中 `answer[i]` 是指对于第 `i` 天，股价下一次上涨是在几天后。如果在这之后股价都不会上涨，请在该位置用 `0` 来代替。

示例 1:

```
输入: prices = [73,74,75,71,69,72,76,73]
输出: [1,1,4,2,1,1,0,0]
```

示例 2:

```
输入: prices = [30,40,50,60]
输出: [1,1,1,0]
```

示例 3:

```
输入: prices = [30,60,90]
输出: [1,1,0]
```

提示：

- `1 <= prices.length <= 105`
- `30 <= prices[i] <= 100`

2. 题目描述

给定一个数组 `prices`，它的第 `i` 个元素 `prices[i]` 表示一支给定股票第 `i` 天的价格。

你只能选择 **某一天** 买入这只股票，并选择在 **未来的某一个不同的日子** 卖出该股票。设计一个算法来计算你能获取的最大利润。

返回你可以从这笔交易中获取的最大利润。如果你不能获取任何利润，返回 `0`。

示例 1:

输入: [7,1,5,3,6,4]

输出: 5

解释: 在第 2 天 (股票价格 = 1) 的时候买入, 在第 5 天 (股票价格 = 6) 的时候卖出, 最大利润 = 6-1 = 5 。

注意利润不能是 7-1 = 6, 因为卖出价格需要大于买入价格; 同时, 你不能在买入前卖出股票。

示例 2:

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。

提示:

- `1 <= prices.length <= 105`
- `0 <= prices[i] <= 104`

3. 题目描述

给定一个整数数组 `prices` , 其中第 `prices[i]` 表示第 `i` 天的股票价格 。

设计一个算法计算出最大利润。在满足以下约束条件下, 你可以尽可能地完成更多的交易 (多次买卖一支股票) :

- 卖出股票后, 你无法在第二天买入股票 (即冷冻期为 1 天)。

注意: 你不能同时参与多笔交易 (你必须在再次购买前出售掉之前的股票) 。

示例 1:

输入: prices = [1,2,3,0,2]

输出: 3

解释: 对应的交易状态为: [买入, 卖出, 冷冻期, 买入, 卖出]

示例 2:

输入: prices = [1]

输出: 0

