122. Best Time to Buy and Sell Stock II

Difficulty	medium
☑	✓
Finished	@July 9, 2023
Problem	array
Previously asked company	Google Walmart
website	leetcode

Question:

You are given an integer array `prices` where `prices[i]` is the price of a given stock on the `i th` day.

On each day, you may decide to buy and/or sell the stock. You can only hold **at most one** share of the stock at any time. However, you can buy it then immediately sell it on the **same day**.

Find and return *the **maximum** profit you can achieve*.

Example 1:

```
Input: prices = [7,1,5,3,6,4]
Output: 7
Explanation: Buy on day 2 (price = 1) and sell on day 3 (price = 5), profit = 5-1 = 4.
Then buy on day 4 (price = 3) and sell on day 5 (price = 6), profit = 6-3 = 3.
Total profit is 4 + 3 = 7.
```

Example 2:

```
Input: prices = [1,2,3,4,5]
Output: 4
Explanation: Buy on day 1 (price = 1) and sell on day 5 (price = 5), profit = 5-1 = 4.
Total profit is 4.
```

Example 3:

```
Input: prices = [7,6,4,3,1]
Output: 0
Explanation: There is no way to make a positive profit, so we never buy the stock to achie
ve the maximum profit of 0.
```

My Solution:

This solution is done buy using 3 pointers.

Time complexity: O(n)

Space complexity: O(1)

```
class Solution(object):
    def maxProfit(self, prices):
        n = len(prices)
        a, b, r = 0, 1, 1
        profit = 0

        while r < n:
            if prices[a] > prices[b]:
                a += 1
                b += 1
                r += 1
            elif prices[a] < prices[r] and r == b:
                r += 1
            else:
                if prices[r] > prices[b]:
                    b += 1
                    r += 1
                else:
                    profit += prices[b] - prices[a]
                    a = r
                    r += 1
                    b = r

        if b < n:
            profit += prices[b] - prices[a]

        return profit
```

Optimal Solution:

Time complexity: O(n)

Space complexity: O(1)

This method is easy to code. But technically it is not the desired way to solve this problem but eventually we get the same answer that is expected by following this approach.

```
class Solution(object):
    def maxProfit(self, prices):
        n = len(prices)
        profit = 0
        for i in range(1,n):
            if prices[i] > prices[i-1]:
                profit += prices[i] - prices[i-1]
        return profit
```