Best Time to Buy and Sell Stock II

Medium 11707 2563 Add to List Share

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.
```

In this question we have unlimited buy sell so we can buy every time we see profit.

```
        0  1  2  3  4  5
a = [ 7, 1, 5, 3, 6, 4 ]
      ↑
      i
```

if (a[i] > a[i-1]) profit += a[i] - a[i-1]

i = 1 → 1 > 7 False
i = 2 → 5 > 1 True profit = 0 + 5 - 1 => 4
i = 3 → 3 > 5 False
i = 4 → 6 > 3 True profit = 4 + 6 - 3 => 4 + 3 => 7
i = 5 → 4 > 6 False
i = 6 false

return profit

T.C - O(n)

S.C - O(1)

```cpp
class Solution {
public:
    int maxProfit(vector<int>& p) {
        int maxprofit=0;
        for(int i=1;i<p.size();i++)
        {
            if(p[i]>p[i-1]) maxprofit+=p[i]-p[i-1];
        }
        return maxprofit;
    }
};
```

→ Yesterday was valley and Today is Peak