Problem 123: Best Time to Buy and Sell Stock III

Problem Information

Difficulty: Hard
Acceptance Rate: 52.56%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

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

Find the maximum profit you can achieve. You may complete **at most two transactions**.

Note: You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you buy again).

Example 1:

Input: prices = [3,3,5,0,0,3,1,4] **Output:** 6 **Explanation:** Buy on day 4 (price = 0) and sell on day 6 (price = 3), profit = 3-0 = 3. Then buy on day 7 (price = 1) and sell on day 8 (price = 4), profit = 4-1 = 3.

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. Note that you cannot buy on day 1, buy on day 2 and sell them later, as you are engaging multiple transactions at the same time. You must sell before buying again.

Example 3:

Input: prices = [7,6,4,3,1] **Output:** 0 **Explanation:** In this case, no transaction is done, i.e. max profit = 0.

Constraints:

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

Code Snippets

C++:

```
class Solution {
public:
int maxProfit(vector<int>& prices) {

}
};
```

Java:

```
class Solution {
public int maxProfit(int[] prices) {

}
}
```

Python3:

```
class Solution:
def maxProfit(self, prices: List[int]) -> int:
```