Documentation for Best Time to Buy and Sell Stock III

Problem Statement

You are given an array prices where prices[i] is the price of a given stock on the ith day. You need to find the maximum profit you can achieve by completing at most two transactions. However, you cannot engage in multiple transactions simultaneously, meaning you must sell the stock before you buy again.

Example Analysis

Let's walk through an example to understand the problem better:

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.
- Total profit = 3 + 3 = 6.

Approach

To solve this problem efficiently, we use a dynamic programming approach:

1. **Define States:** We maintain four variables:

- *buy1:* Maximum profit after the first buy operation.
- *sell1:* Maximum profit after the first sell operation.
- *buy2:* Maximum profit after the second buy operation.
- *sell2:* Maximum profit after the second sell operation.

2. **Initialization:** Initialize these variables based on the first price in the prices array.

3. **State Transitions:** *Iterate through each price in the prices array and update these variables based on the current price:*

- Update buy1 and sell1 considering the profit if the first transaction (buy/sell) were made up to the current day.
- Update buy2 and sell2 considering the profit if the second transaction (buy/sell) were made up to the current day.

4. **Result:** The maximum profit achievable after the two transactions will be stored in sell2.

Constraints and Complexity

Constraints: The length of prices can go up to 100,000, making an O(n) solution crucial.

Time Complexity: O(n), where n is the length of the prices array, since we iterate through it once.

Space Complexity: O(1), since we use only a constant amount of extra space.

This approach ensures that we efficiently compute the maximum profit by leveraging dynamic programming to keep track of the best possible profits after each potential transaction.

By following this approach, we can find the maximum profit achievable with at most two transactions for any given set of stock prices.