Problem 714: Best Time to Buy and Sell Stock with Transaction Fee

Problem Information

Difficulty: Medium
Acceptance Rate: 71.35%
Paid Only: No
Tags: Array, Dynamic Programming, Greedy

Problem Description

You are given an array `prices` where `prices[i]` is the price of a given stock on the `ith` day, and an integer `fee` representing a transaction fee.

Find the maximum profit you can achieve. You may complete as many transactions as you like, but you need to pay the transaction fee for each transaction.

Note:

* You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you buy again). * The transaction fee is only charged once for each stock purchase and sale.

Example 1:

Input: prices = [1,3,2,8,4,9], fee = 2 **Output:** 8 **Explanation:** The maximum profit can be achieved by: - Buying at prices[0] = 1 - Selling at prices[3] = 8 - Buying at prices[4] = 4 - Selling at prices[5] = 9 The total profit is ((8 - 1) - 2) + ((9 - 4) - 2) = 8.

Example 2:

Input: prices = [1,3,7,5,10,3], fee = 3 **Output:** 6

Constraints:

* `1 <= prices.length <= 5 * 104` * `1 <= prices[i] < 5 * 104` * `0 <= fee < 5 * 104`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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