Problem 2291: Maximum Profit From Trading Stocks

Problem Information

Difficulty: Medium
Acceptance Rate: 47.33%
Paid Only: Yes
Tags: Array, Dynamic Programming

Problem Description

You are given two **0-indexed** integer arrays of the same length `present` and `future` where `present[i]` is the current price of the `ith` stock and `future[i]` is the price of the `ith` stock a year in the future. You may buy each stock at most **once**. You are also given an integer `budget` representing the amount of money you currently have.

Return _the maximum amount of profit you can make._

Example 1:

Input: present = [5,4,6,2,3], future = [8,5,4,3,5], budget = 10 **Output:** 6 **Explanation:** One possible way to maximize your profit is to: Buy the 0th, 3rd, and 4th stocks for a total of 5 + 2 + 3 = 10. Next year, sell all three stocks for a total of 8 + 3 + 5 = 16. The profit you made is 16 - 10 = 6. It can be shown that the maximum profit you can make is 6.

Example 2:

Input: present = [2,2,5], future = [3,4,10], budget = 6 **Output:** 5 **Explanation:** The only possible way to maximize your profit is to: Buy the 2nd stock, and make a profit of 10 - 5 = 5. It can be shown that the maximum profit you can make is 5.

Example 3:

Input: present = [3,3,12], future = [0,3,15], budget = 10 **Output:** 0 **Explanation:** One possible way to maximize your profit is to: Buy the 1st stock, and make a profit of 3 - 3 = 0. It can be shown that the maximum profit you can make is 0.

Constraints:

* `n == present.length == future.length` * `1 <= n <= 1000` * `0 <= present[i], future[i] <= 100` * `0 <= budget <= 1000`

Code Snippets

C++:

```cpp
class Solution {
public:
    int maximumProfit(vector<int>& present, vector<int>& future, int budget) {


    }
};
```

Java:

```java
class Solution {
    public int maximumProfit(int[] present, int[] future, int budget) {


    }
}
```

Python3:

```python
class Solution:
    def maximumProfit(self, present: List[int], future: List[int], budget: int)
    -> int:
```