Problem 2907: Maximum Profitable Triplets With Increasing Prices I

Problem Information

Difficulty: Medium
Acceptance Rate: 54.68%
Paid Only: Yes
Tags: Array, Binary Indexed Tree, Segment Tree

Problem Description

Given the **0-indexed** arrays `prices` and `profits` of length `n`. There are `n` items in an store where the `ith` item has a price of `prices[i]` and a profit of `profits[i]`.

We have to pick three items with the following condition:

* `prices[i] < prices[j] < prices[k]` where `i < j < k`.

If we pick items with indices `i`, `j` and `k` satisfying the above condition, the profit would be `profits[i] + profits[j] + profits[k]`.

Return _the**maximum profit** we can get, and _`-1` _if it 's not possible to pick three items with the given condition._

Example 1:

Input: prices = [10,2,3,4], profits = [100,2,7,10] **Output:** 19 **Explanation:** We can't pick the item with index i=0 since there are no indices j and k such that the condition holds. So the only triplet we can pick, are the items with indices 1, 2 and 3 and it's a valid pick since prices[1] < prices[2] < prices[3]. The answer would be sum of their profits which is 2 + 7 + 10 = 19.

Example 2:

Input: prices = [1,2,3,4,5], profits = [1,5,3,4,6] **Output:** 15 **Explanation:** We can select any triplet of items since for each triplet of indices i, j and k such that i < j < k, the

condition holds. Therefore the maximum profit we can get would be the 3 most profitable items which are indices 1, 3 and 4. The answer would be sum of their profits which is 5 + 4 + 6 = 15.

Example 3:

Input: prices = [4,3,2,1], profits = [33,20,19,87] **Output:** -1 **Explanation:** We can't select any triplet of indices such that the condition holds, so we return -1.

Constraints:

* `3 <= prices.length == profits.length <= 2000` * `1 <= prices[i] <= 106` * `1 <= profits[i] <= 106`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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