Problem 1187: Make Array Strictly Increasing

Problem Information

Difficulty: Hard
Acceptance Rate: 57.86%
Paid Only: No
Tags: Array, Binary Search, Dynamic Programming, Sorting

Problem Description

Given two integer arrays `arr1` and `arr2`, return the minimum number of operations (possibly zero) needed to make `arr1` strictly increasing.

In one operation, you can choose two indices `0 <= i < arr1.length` and `0 <= j < arr2.length` and do the assignment `arr1[i] = arr2[j]`.

If there is no way to make `arr1` strictly increasing, return `-1`.

Example 1:

Input: arr1 = [1,5,3,6,7], arr2 = [1,3,2,4] **Output:** 1 **Explanation:** Replace 5 with 2, then arr1 = [1, 2, 3, 6, 7].

Example 2:

Input: arr1 = [1,5,3,6,7], arr2 = [4,3,1] **Output:** 2 **Explanation:** Replace 5 with 3 and then replace 3 with 4. arr1 = [1, 3, 4, 6, 7].

Example 3:

Input: arr1 = [1,5,3,6,7], arr2 = [1,6,3,3] **Output:** -1 **Explanation:** You can't make arr1 strictly increasing.

Constraints:

* `1 <= arr1.length, arr2.length <= 2000` * `0 <= arr1[i], arr2[i] <= 10^9`

Code Snippets

C++:

```
class Solution {
public:
int makeArrayIncreasing(vector<int>& arr1, vector<int>& arr2) {

}
};
```

Java:

```
class Solution {
public int makeArrayIncreasing(int[] arr1, int[] arr2) {

}
}
```

Python3:

```
class Solution:
def makeArrayIncreasing(self, arr1: List[int], arr2: List[int]) -> int:
```