Problem 801: Minimum Swaps To Make Sequences Increasing

Problem Information

Difficulty: Hard
Acceptance Rate: 41.14%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

You are given two integer arrays of the same length `nums1` and `nums2`. In one operation, you are allowed to swap `nums1[i]` with `nums2[i]`.

* For example, if `nums1 = [1,2,3,_8_]`, and `nums2 = [5,6,7,_4_]`, you can swap the element at `i = 3` to obtain `nums1 = [1,2,3,4]` and `nums2 = [5,6,7,8]`.

Return _the minimum number of needed operations to make_`nums1` _and_`nums2` _**strictly increasing**_. The test cases are generated so that the given input always makes it possible.

An array `arr` is **strictly increasing** if and only if `arr[0] < arr[1] < arr[2] < ... < arr[arr.length - 1]`.

Example 1:

Input: nums1 = [1,3,5,4], nums2 = [1,2,3,7] **Output:** 1 **Explanation:** Swap nums1[3] and nums2[3]. Then the sequences are: nums1 = [1, 3, 5, 7] and nums2 = [1, 2, 3, 4] which are both strictly increasing.

Example 2:

Input: nums1 = [0,3,5,8,9], nums2 = [2,1,4,6,9] **Output:** 1

Constraints:

* `2 <= nums1.length <= 105` * `nums2.length == nums1.length` * `0 <= nums1[i], nums2[i] <= 2 * 105`

Code Snippets

C++:

```
class Solution {
public:
int minSwap(vector<int>& nums1, vector<int>& nums2) {

}
};
```

Java:

```
class Solution {
public int minSwap(int[] nums1, int[] nums2) {

}
}
```

Python3:

```
class Solution:
def minSwap(self, nums1: List[int], nums2: List[int]) -> int:
```