Problem 2143: Choose Numbers From Two Arrays in Range

Problem Information

Difficulty: Hard
Acceptance Rate: 52.60%
Paid Only: Yes
Tags: Array, Dynamic Programming

Problem Description

You are given two **0-indexed** integer arrays `nums1` and `nums2` of length `n`.

A range `[l, r]` (**inclusive**) where `0 <= l <= r < n` is **balanced** if:

* For every `i` in the range `[l, r]`, you pick either `nums1[i]` or `nums2[i]`. * The sum of the numbers you pick from `nums1` equals to the sum of the numbers you pick from `nums2` (the sum is considered to be `0` if you pick no numbers from an array).

Two **balanced** ranges from `[l1, r1]` and `[l2, r2]` are considered to be **different** if at least one of the following is true:

* `l1 != l2` * `r1 != r2` * `nums1[i]` is picked in the first range, and `nums2[i]` is picked in the second range or **vice versa** for at least one `i`.

Return _the number of**different** ranges that are balanced. _Since the answer may be very large, return it **modulo** `109 + 7` _._

Example 1:

Input: nums1 = [1,2,5], nums2 = [2,6,3] **Output:** 3 **Explanation:** The balanced ranges are: - [0, 1] where we choose nums2[0], and nums1[1]. The sum of the numbers chosen from nums1 equals the sum of the numbers chosen from nums2: 2 = 2. - [0, 2] where we choose nums1[0], nums2[1], and nums1[2]. The sum of the numbers chosen from nums1 equals the sum of the numbers chosen from nums2: 1 + 5 = 6. - [0, 2] where we choose nums1[0], nums1[1], and nums2[2]. The sum of the numbers chosen from nums1 equals the sum of the numbers chosen from nums2: 1 + 2 = 3. Note that the second and third balanced ranges are different. In the second balanced range, we choose nums2[1] and in the third balanced range, we choose nums1[1].

Example 2:

Input: nums1 = [0,1], nums2 = [1,0] **Output:** 4 **Explanation:** The balanced ranges are: - [0, 0] where we choose nums1[0]. The sum of the numbers chosen from nums1 equals the sum of the numbers chosen from nums2: 0 = 0. - [1, 1] where we choose nums2[1]. The sum of the numbers chosen from nums1 equals the sum of the numbers chosen from nums2: 0 = 0. - [0, 1] where we choose nums1[0] and nums2[1]. The sum of the numbers chosen from nums1 equals the sum of the numbers chosen from nums2: 0 = 0. - [0, 1] where we choose nums2[0] and nums1[1]. The sum of the numbers chosen from nums1 equals the sum of the numbers chosen from nums2: 1 = 1.

Constraints:

* `n == nums1.length == nums2.length` * `1 <= n <= 100` * `0 <= nums1[i], nums2[i] <= 100`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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