

# Problem 3251: Find the Count of Monotonic Pairs II

## Problem Information

Difficulty: **Hard**

Acceptance Rate: 23.94%

Paid Only: No

Tags: Array, Math, Dynamic Programming, Combinatorics, Prefix Sum

## Problem Description

You are given an array of **positive** integers `nums` of length `n`.

We call a pair of **non-negative** integer arrays `(arr1, arr2)` **monotonic** if:

\* The lengths of both arrays are `n`. \* `arr1` is monotonically **non-decreasing**, in other words, `arr1[0] ≤ arr1[1] ≤ ... ≤ arr1[n - 1]`. \* `arr2` is monotonically **non-increasing**, in other words, `arr2[0] ≥ arr2[1] ≥ ... ≥ arr2[n - 1]`. \* `arr1[i] + arr2[i] == nums[i]` for all `0 ≤ i ≤ n - 1`.

Return the count of **monotonic** pairs.

Since the answer may be very large, return it **modulo** `109 + 7`.

**Example 1:**

**Input:** `nums = [2,3,2]`

**Output:** 4

**Explanation:**

The good pairs are:

1. `([0, 1, 1], [2, 2, 1])` 2. `([0, 1, 2], [2, 2, 0])` 3. `([0, 2, 2], [2, 1, 0])` 4. `([1, 2, 2], [1, 1, 0])`

**\*\*Example 2:\*\***

**\*\*Input:\*\*** nums = [5,5,5,5]

**\*\*Output:\*\*** 126

**\*\*Constraints:\*\***

**\*`1` <= n == nums.length <= 2000` \*`1` <= nums[i] <= 1000`**

## Code Snippets

### C++:

```
class Solution {
public:
    int countOfPairs(vector<int>& nums) {

    }
};
```

### Java:

```
class Solution {
    public int countOfPairs(int[] nums) {

    }
}
```

### Python3:

```
class Solution:
    def countOfPairs(self, nums: List[int]) -> int:
```