

3355. Zero Array Transformation I

Attempted

Medium Topics Companies Hint

You are given an integer array `nums` of length `n` and a 2D array `queries`, where `queries[i] = [li, ri]`.

For each `queries[i]`:

- Select a **subset** of indices within the range `[li, ri]` in `nums`.
- Decrement the values at the selected indices by 1.

A **Zero Array** is an array where all elements are equal to 0.

Return `true` if it is possible to transform `nums` into a **Zero Array** after processing all the queries sequentially, otherwise return `false`.

Example 1:

Input: `nums = [1,0,1]`, `queries = [[0,2]]`

Output: `true`

Explanation:

- For `i = 0`:
 - Select the subset of indices as `[0, 2]` and decrement the values at these indices by 1.
- The array will become `[0, 0, 0]`, which is a Zero Array.

Python3 Auto

```
1 class Solution:
2     def isZeroArray(self, nums: List[int], queries: List[List[int]]) -> bool:
3         total = [0] * len(nums)
4         for q in queries:
5             l, r = q[0], q[1]
6             for i in range(l, r+1):
7                 total[i] += 1
8
9         for j in range(len(nums)):
10            if nums[j] > total[j]:
11                return False
12
13        return True
14
15
```

TLE

Constraints:

- `1 <= nums.length <= 105`
- `0 <= nums[i] <= 105`
- `1 <= queries.length <= 105`
- `queries[i].length == 2`
- `0 <= li <= ri < nums.length`

All Submissions

Accepted 668 / 668 testcases passed

AndrewC275 submitted at May 20, 2025 09:48

Editorial

Solution

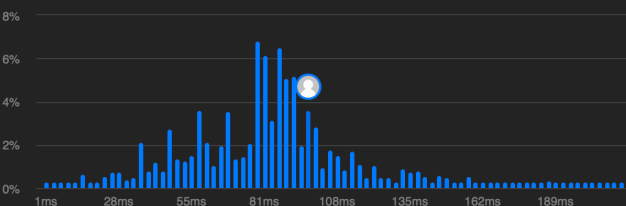
Runtime

98 ms | Beats 31.05%

Analyze Complexity

Memory

54.96 MB | Beats 70.61%



Python3 Auto

```
1 class Solution:
2     def isZeroArray(self, nums: List[int], queries: List[List[int]]) -> bool:
3         total = [0] * (len(nums) + 1)
4
5         for q in queries:
6             l, r = q[0], q[1]
7             total[l] += 1
8             if r+1 < len(nums):
9                 total[r+1] -= 1
10
11        count = 0
12        for i in range(len(nums)):
13            count += total[i]
14            if nums[i] > count:
15                return False
16
17        return True
18
19
```

Approach: Difference Array $O(n+m)$

For each query interval `[left, right]`:

increment `deltaArray[left]` by +1, indicating an increase in the operation count starting from left.

Decrement `deltaArray[right + 1]` by -1, indicating that the operation count returns to its original value after `right + 1`.