LeetCode 3355. Zero Array Transformation I

May 20 2025

3355. Zero Array Transformation I

Solved

Medium | Topics | Companies | Hint

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

For each `queries[i]`:

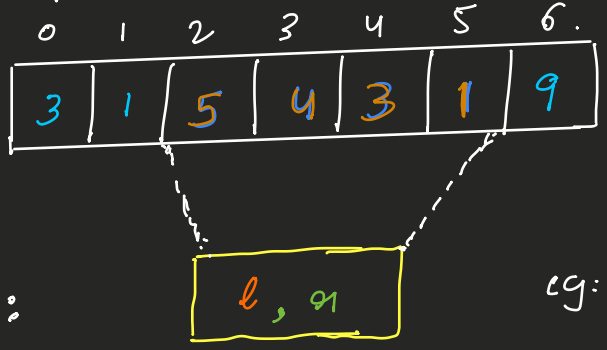
- Select a subset of indices within the range `[l_i, r_i]` in `nums`.
- Decrement the values at the selected indices by 1.

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`.

Question: 1) Given an array of +ve numbers.

2) Given q - queries each is a range [l, r] ← index range.



query 1 : eg: (2, 5)

3) In this range, you can decrease any set of numbers by 1 for a query.

4) Finally return true if whole array can be made to zeros.

Thought Process :

Approach : Naive

- we can just simulate the exact question.

- So, the idea is - for every Query Range, Go over the array and decrement a value if it is > 0.

- This results in TLE as Time Complexity is (N x Queries)

⭐ Approach - 2 (Optimized)

Range Operations (+/-)
- Segment Tree (Over work here)
- Difference Array Technique ✓✓

Forget about the Terminology,

Ask Yourself :-

1) If it is of any help if you know how many times a position in the array has been affected by the Query Range ?

2) Can you get to something, ??

When, for a particular number / position in the array, there is information on how many times it can be decreased at most ....

3) This is the core intuition. If you know

this information, the idea is simple.

You have this original number $x = a$

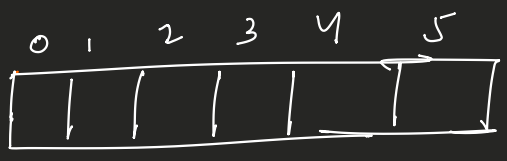
Information you have — — — d — — —

d - Max number of decrements.

i.e, how many times the position occured in queries ranges.

Now you can basically tell if x can be made to 0 by decreasing at most d times.

→ To achieve this you can use Difference Array Technique



queries = { [1,3], [0,2], [1,4] }

step 1 : Make an empty array of size n. → diff name.

step 2 : For each query → [l, r]

diff [l] ↑ by 1

diff [r+1] ↓ by 1 (if valid).

why decrement r+1 by 1 ?

- To Nullify the affect of query past the range while doing the cumulative sum finally.

Got stuck ?? watch this ☺

https://www.youtube.com/watch?v=ZHNVmtm08WY&list=PLplkg8OmuX-Kqkb8DqDe_4-Tiav6ilS_L

Hopefully this helps to solve the problem.