5547. Arithmetic Subarrays

My Submissions (/contest/weekly-contest-212/problems/arithmetic-subarrays/submissions/) Back to Contest (/contest/weekly-contest-212/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

A sequence of numbers is called **arithmetic** if it consists of at least two elements, and the difference between every two consecutive elements is the same. More formally, a sequence `s` is arithmetic if and only if `s[i+1] - s[i] == s[1] - s[0]` for all valid `i`.

For example, these are **arithmetic** sequences:

```
1, 3, 5, 7, 9
7, 7, 7, 7
3, -1, -5, -9
```

The following sequence is not **arithmetic**:

```
1, 1, 2, 5, 7
```

You are given an array of `n` integers, `nums`, and two arrays of `m` integers each, `l` and `r`, representing the `m` range queries, where the i^{th} query is the range `[l[i], r[i]]`. All the arrays are **0-indexed**.

Return *a list of* `boolean` *elements* `answer`, *where* `answer[i]` *is* `true` *if the subarray* `nums[l[i]], nums[l[i]+1], ... , nums[r[i]]` *can be* ***rearranged*** *to form an* ***arithmetic*** *sequence, and* `false` *otherwise.*

Example 1:

```
Input: nums = [4,6,5,9,3,7], l = [0,0,2], r = [2,3,5]
Output: [true,false,true]
Explanation:
In the 0th query, the subarray is [4,6,5]. This can be rearranged as [6,5,4], which is an arithmetic sequence.
In the 1st query, the subarray is [4,6,5,9]. This cannot be rearranged as an arithmetic sequence.
In the 2nd query, the subarray is [5,9,3,7]. This can be rearranged as [3,5,7,9], which is an arithmetic sequence.
```

Example 2:

```
Input: nums = [-12,-9,-3,-12,-6,15,20,-25,-20,-15,-10], l = [0,1,6,4,8,7], r = [4,4,9,7,9,10]
Output: [false,true,false,false,true,true]
```

Constraints:

- `n == nums.length`
- `m == l.length`
- `m == r.length`
- `2 <= n <= 500`
- `1 <= m <= 500`
- `0 <= l[i] < r[i] < n`
- -10^5 `<= nums[i] <=` 10^5

JavaScript

```
/**
 * @param {number[]} nums
 * @param {number[]} l
 * @param {number[]} r
 * @return {boolean[]}
 */
const checkArithmeticSubarrays = (nums, l, r) => {
    let m = l.length;
    let res = [];
    for (let i = 0; i < m; i++) {
        let range = nums.slice(l[i], r[i] + 1);
```

```
        if (ok(range)) {
            res.push(true);
        } else {
            res.push(false);
        }
    }
    return res;
};

const ok = (arr) => {
    let n = arr.length;
    let tmp = [...arr].sort((a, b) => a - b);
    for (let i = 1; i + 1 < n; i++) {
        if (tmp[i + 1] - tmp[i] != tmp[i] - tmp[i - 1]) {
            return false;
        }
    }
    return true;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/412813945/) More Details > (/submissions/detail/412813945/)

Share your acceptance!