Problem 1630: Arithmetic Subarrays

Problem Information

Difficulty: Medium
Acceptance Rate: 83.78%
Paid Only: No
Tags: Array, Hash Table, Sorting

Problem Description

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 `ith` 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` * `-105 <= nums[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
vector<bool> checkArithmeticSubarrays(vector<int>& nums, vector<int>& l,
vector<int>& r) {

}
};
```

Java:

```
class Solution {
public List<Boolean> checkArithmeticSubarrays(int[] nums, int[] l, int[] r) {

}
}
```

Python3:

```
class Solution:
def checkArithmeticSubarrays(self, nums: List[int], l: List[int], r:
List[int]) -> List[bool]:
```