Problem 220: Contains Duplicate III

Problem Information

Difficulty: Hard
Acceptance Rate: 24.19%
Paid Only: No
Tags: Array, Sliding Window, Sorting, Bucket Sort, Ordered Set

Problem Description

You are given an integer array `nums` and two integers `indexDiff` and `valueDiff`.

Find a pair of indices `(i, j)` such that:

* `i != j`, * `abs(i - j) <= indexDiff`. * `abs(nums[i] - nums[j]) <= valueDiff`, and

Return `true` _if such pair exists or_`false` _otherwise_.

Example 1:

Input: nums = [1,2,3,1], indexDiff = 3, valueDiff = 0 **Output:** true **Explanation:** We can choose (i, j) = (0, 3). We satisfy the three conditions: i != j --> 0 != 3 abs(i - j) <= indexDiff --> abs(0 - 3) <= 3 abs(nums[i] - nums[j]) <= valueDiff --> abs(1 - 1) <= 0

Example 2:

Input: nums = [1,5,9,1,5,9], indexDiff = 2, valueDiff = 3 **Output:** false **Explanation:** After trying all the possible pairs (i, j), we cannot satisfy the three conditions, so we return false.

Constraints:

* `2 <= nums.length <= 105` * `-109 <= nums[i] <= 109` * `1 <= indexDiff <= nums.length` * `0 <= valueDiff <= 109`

Code Snippets

C++:

```
class Solution {
public:
bool containsNearbyAlmostDuplicate(vector<int>& nums, int indexDiff, int
valueDiff) {

}
};
```

Java:

```
class Solution {
public boolean containsNearbyAlmostDuplicate(int[] nums, int indexDiff, int
valueDiff) {

}
}
```

Python3:

```
class Solution:
def containsNearbyAlmostDuplicate(self, nums: List[int], indexDiff: int,
valueDiff: int) -> bool:
```