219. Contains Duplicate II

Description | Hints | Submissions | Discuss | Solution

Pick One

Given an array of integers and an integer *k*, find out whether there are two distinct indices *i* and *j* in the array such that **nums[i]** = **nums[j]** and the **absolute** difference between *i* and *j* is at most *k*.

Example 1:

```
Input: nums = [1,2,3,1], k = 3
Output: true
```

Example 2:

```
Input: nums = [1,0,1,1], k = 1
Output: true
```

Example 3:

```
Input: nums = [1,2,3,1,2,3], k = 2
Output: false
```

Seen this question in a real interview before? Yes No

```java
/*
 * 题意：当且仅当存在两个不同的下标i和j满足nums[i]=nums[j]并且|i - j| <= k时返回true，否则返回false
 */
public class L219 {
    public boolean containsNearbyDuplicate(int[] nums, int k) {
        //输入判断条件
        if(nums == null || nums.length < 2 || k < 1) {
            return false;
        }

        Map<Integer, Integer> map = new HashMap<Integer, Integer>();

        for(int i = 0; i < nums.length; i ++) {
            if(!map.containsKey(nums[i])) {
                map.put(nums[i], i);
            }
            else {
                int value = map.get(nums[i]);
                if(i - value <= k) {
                    return true;
                }else {
                    map.put(nums[i], i);
                }
            }
        }
        return false;
    }
}
```