**难度: 简单**

给定一个整数数组和一个整数 k，判断数组中是否存在两个不同的索引 i 和 j，使得 nums [i] = nums [j]，并且 i 和 j 的差的 绝对值 至多为 k。

示例 1:

输入: nums = [1,2,3,1], k = 3

输出: true

示例 2:

输入: nums = [1,0,1,1], k = 1

输出: true

示例 3:

输入: nums = [1,2,3,1,2,3], k = 2

输出: false

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/contains-duplicate-ii

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路1:**

1). 使用HashMap来保存数组的value和下标

2). 然后遇到重复的,就比较以下对应的下标的差是否小于等于k

3). 成立的话就返回true,不成立就返回false

|  |
| --- |
| public boolean containsNearbyDuplicate(int[] nums, int k) {  HashMap<Integer,Integer> map = new HashMap<>();  for (int i = 0; i < nums.length; i++) {  Integer value = map.get(nums[i]);  if(value != null && Math.abs(value - i) <= k) {  return true;  }else {  map.put(nums[i],i);  }  }  return false;  } |

**代码2:**

1). 判断key是否存在,如果存在就比较一下两个下标的差是否小于等于k

2). 不成立的话就添加key和value

|  |
| --- |
| public boolean containsNearbyDuplicate2(int[] nums, int k) {  HashMap<Integer,Integer> map = new HashMap<>();  for (int i = 0; i < nums.length; i++) {  if (map.containsKey(nums[i])) {  if (Math.abs(map.get(nums[i]) - i) <= k) {  return true;  }  }  map.put(nums[i],i);  }  return false;  } |