

0219. 存在重复元素 II

👤 ITCharge 🕒 大约 1 分钟

- 标签：数组、哈希表、滑动窗口
- 难度：简单

题目链接

- [0219. 存在重复元素 II - 力扣](#)

题目大意

描述： 给定一个整数数组 $nums$ 和一个整数 k 。

要求： 判断是否存在 $nums[i] == nums[j]$ ($i \neq j$)，并且 i 和 j 的差绝对值至多为 k 。

说明：

- $1 \leq nums.length \leq 10^5$ 。
- $-10^9 \leq nums[i] \leq 10^9$ 。
- $0 \leq k \leq 10^5$ 。

示例：

- 示例 1:

输入: $nums = [1, 2, 3, 1]$, $k = 3$

输出: $True$

py

解题思路

思路 1：哈希表

维护一个最多有 k 个元素的哈希表。遍历 $nums$ ，对于数组中的每个整数 $nums[i]$ ，判断哈希表中是否存在这个整数。

- 如果存在，则说明出现了两次，且 $i \neq j$ ，直接返回 *True*。
- 如果不存在，则将 *nums[i]* 加入哈希表。
- 判断哈希表长度是否超过了 *k*，如果超过了 *k*，则删除哈希表中最旧的元素 *nums[i - k]*。
- 如果遍历完仍旧找不到，则返回 *False*。

思路 1：代码

```
class Solution:
    def containsNearbyDuplicate(self, nums: List[int], k: int) -> bool:
        nums_dict = dict()
        for i in range(len(nums)):
            if nums[i] in nums_dict:
                return True
            nums_dict[nums[i]] = 1
            if len(nums_dict) > k:
                del nums_dict[nums[i - k]]
        return False
```

py

思路 1：复杂度分析

- 时间复杂度： $O(n)$ 。
- 空间复杂度： $O(n)$ 。