

# 0217. 存在重复元素

👤 [ITCharge](#) ⌚ 大约 2 分钟

- 标签：数组、哈希表、排序
- 难度：简单

## 题目链接

- [0217. 存在重复元素 - 力扣](#)

## 题目大意

**描述：** 给定一个整数数组 `nums` 。

**要求：** 判断是否存在重复元素。如果有元素在数组中出现至少两次，返回 `True` ； 否则返回 `False` 。

**说明：**

- $1 \leq \text{nums.length} \leq 10^5$ 。
- $-10^9 \leq \text{nums}[i] \leq 10^9$ 。

**示例：**

- 示例 1:

```
输入: nums = [1,2,3,1]
输出: True
```

py

- 示例 2:

```
输入: nums = [1,2,3,4]
输出: False
```

py

# 解题思路

---

## 思路 1：哈希表

- 使用一个哈希表存储元素和对应元素数量。
- 遍历元素，如果哈希表中出现了该元素，则直接输出 `True`。如果没有出现，则向哈希表中插入该元素。
- 如果遍历完也没发现重复元素，则输出 `False`。

## 思路 1：代码

```
class Solution:
    def containsDuplicate(self, nums: List[int]) -> bool:
        numDict = dict()
        for num in nums:
            if num in numDict:
                return True
            else:
                numDict[num] = num
        return False
```

py

## 思路 1：复杂度分析

- 时间复杂度： $O(n)$ 。
- 空间复杂度： $O(n)$ 。

## 思路 2：集合

- 使用一个 `set` 集合存储数组中所有元素。
- 如果集合中元素个数与数组元素个数不同，则说明出现了重复元素，返回 `True`。
- 如果集合中元素个数与数组元素个数相同，则说明没有出现了重复元素，返回 `False`。

## 思路 2：集合代码

```
class Solution:
    def containsDuplicate(self, nums: List[int]) -> bool:
        return len(set(nums)) != len(nums)
```

py

## 思路 2：复杂度分析

- 时间复杂度： $O(n)$ 。
- 空间复杂度： $O(n)$ 。

## 思路 3：排序

- 对数组进行排序。
- 排序之后，遍历数组，判断相邻元素之间是否出现重复元素。
- 如果相邻元素相同，则说明出现了重复元素，返回 `True`。
- 如果遍历完也没发现重复元素，则 `False`。

## 思路 3：排序代码

```
class Solution:
    def containsDuplicate(self, nums: List[int]) -> bool:
        nums.sort()
        for i in range(1, len(nums)):
            if nums[i - 1] == nums[i]:
                return True
        return False
```

py

## 思路 3：复杂度分析

- 时间复杂度： $O(n \times \log n)$ 。

- 空间复杂度:  $O(1)$ 。