

0268. 丢失的数字

👤 ITCharge ⌚ 大约 2 分钟

- 标签：位运算、数组、哈希表、数学、二分查找、排序
- 难度：简单

题目链接

- [0268. 丢失的数字 - 力扣](#)

题目大意

描述： 给定一个包含 $[0, n]$ 中 n 个数的数组 *nums*。

要求： 找出 $[0, n]$ 这个范围内没有出现在数组中的那个数。

说明：

- $n == \text{nums.length}$
- $1 \leq n \leq 10^4$
- $0 \leq \text{nums}[i] \leq n$ 。
- *nums* 中的所有数字都独一无二。

示例：

- 示例 1:

输入: `nums = [3,0,1]`

输出: `2`

解释: $n = 3$ ，因为有 3 个数字，所以所有的数字都在范围 $[0, 3]$ 内。2 是丢失的数字，因为它没有出现在 `nums` 中。

py

- 示例 2:

输入: `nums = [0,1]`

输出: `2`

解释: $n = 2$, 因为有 `2` 个数字, 所以所有的数字都在范围 `[0,2]` 内。`2` 是丢失的数字, 因为它没有出现在 `nums` 中。

解题思路

$[0, n]$ 的范围有 $n + 1$ 个数 (包含 0)。现在给了我们 n 个数, 要求找出其中缺失的那个数。

思路 1: 哈希表

将 `nums` 中所有元素插入到哈希表中, 然后遍历 $[0, n]$, 找到缺失的数字。

这里的哈希表也可以用长度为 $n + 1$ 的数组代替。

思路 1: 代码

```
class Solution:
    def missingNumber(self, nums: List[int]) -> int:
        numSet = set(nums)

        for num in range(len(nums)+1):
            if num not in numSet:
                return num
```

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: 数学计算

已知 $[0, n]$ 的求和公式为: $\sum_{i=0}^n i = \frac{n*(n+1)}{2}$, 则用 $[0, n]$ 的和, 减去数组中所有元素的和, 就得到了缺失数字。

思路 2：代码

```
class Solution:
    def missingNumber(self, nums: List[int]) -> int:
        sum_nums = sum(nums)
        n = len(nums)
        return (n + 1) * n // 2 - sum_nums
```

py

思路 2：复杂度分析

- 时间复杂度： $O(n)$ 。
- 空间复杂度： $O(1)$ 。