

剑指 Offer 53 - II. 0 ~ n-1中缺失的数字

题目

一个长度为 $n-1$ 的递增排序数组中的所有数字都是唯一的，并且每个数字都在范围 $0 \sim n-1$ 之内。在范围 $0 \sim n-1$ 内的 n 个数字中有且只有一个数字不在该数组中，请找出这个数字。

示例 1:

输入: [0,1,3]

输出: 2

示例 2:

输入: [0,1,2,3,4,5,6,7,9]

输出: 8

限制:

$1 \leq \text{数组长度} \leq 10000$

思路一

遍历，一个个比较。

代码

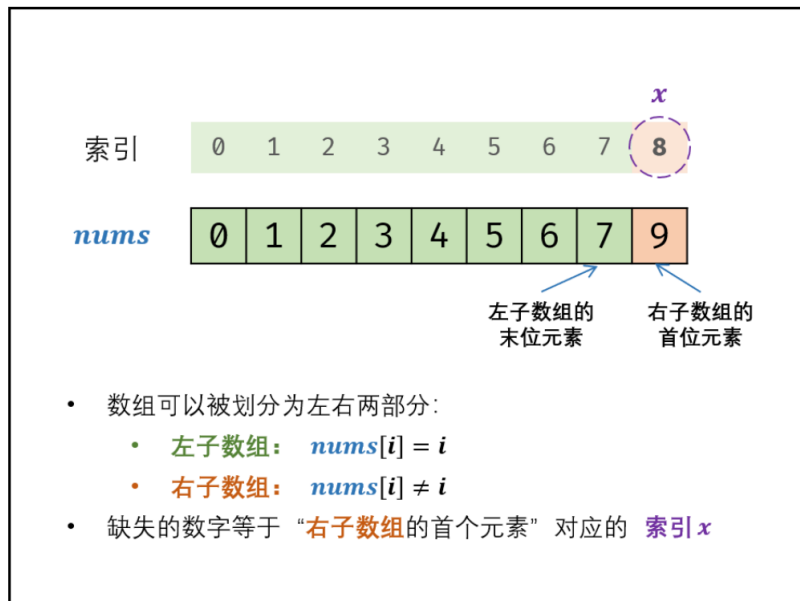
```
class Solution(object):
    def missingNumber(self, nums):
        """
        :type nums: List[int]
        :rtype: int
        """
        if nums[0] == 1:
            return 0
        for i in range(len(nums)):
            if nums[i] != i:
                return i
        return len(nums)
```

复杂度分析

- 时间复杂度： $O(N)$
- 空间复杂度： $O(1)$

思路二

排序数组中的搜索问题，首先想到 **二分法** 解决。



代码

```
class Solution(object):
    def missingNumber(self, nums):
        """
        :type nums: List[int]
        :rtype: int
        """
        i, j = 0, len(nums)-1
        while i <= j:
            m = (i + j) // 2
            if nums[m] == m:
                i = m + 1
            else:
                j = m - 1
        return i
```

复杂度分析

- 时间复杂度: $O(\log N)$, 二分法为对数级别复杂度
- 空间复杂度: $O(1)$

