**难度: 简单**

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～n-1之内。在范围0～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

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/que-shi-de-shu-zi-lcof

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). 因为本身数组的范围是0 到 n – 1

2). 所以可以循环判断是否nums[i] == i

如果不等于直接返回i

3). 如果循环结束还没有返回,那么最后直接返回nums的长度就行了

|  |
| --- |
| public int missingNumber(int[] nums) {  for (int i = 0; i < nums.length - 1; i++) {  if (nums[i] != i) {  return i;  }  }  return nums.length;  } |

**思路2(使用二分查找):**

1). 先判断中间数组是否和要找的中间点相等

2). 如果相等向右进行二分

3). 如果不相等向左进行二分

4). 最后返回左节点

|  |
| --- |
| public int missingNumber2(int[] nums) {  int left = 0;  int right = nums.length;  while(left <= right) {  int mid = ( left + right )/2;  if (nums[mid] == mid) {  left = mid + 1;  }else {  right = mid - 1;  }  }  return left;  } |