**难度: 简单**

找出数组中重复的数字。

在一个长度为 n 的数组 nums 里的所有数字都在 0～n-1 的范围内。数组中某些数字是重复的，但不知道有几个数字重复了，也不知道每个数字重复了几次。请找出数组中任意一个重复的数字。

示例 1：

输入：

[2, 3, 1, 0, 2, 5, 3]

输出：2 或 3

限制：

2 <= n <= 100000

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/shu-zu-zhong-zhong-fu-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). 创建一个set集合

2). 循环数组,把数组所以的值全部放进去

3). 定义一个判断,如果添加进去为false就直接return

4). 最后如果都没有重复,直接返回-1

|  |
| --- |
| public int findRepeatNumber(int[] nums) {  HashSet<Integer> set = new HashSet<>();  for (int i = 0; i < nums.length; i++) {  if (!set.add(nums[i])) {  return nums[i];  }  }  return -1;  } |

**思路2:**

1). 创建一个和nums一样长度的临时数组

2). 因为nums里面的值是 0～n-1 的范围内,所以每次把nums[i]的值当作下标放入到临时数组

3). 放入每次判断这个临时数组是否等于2,如果等于2说明有重复的,直接返回nums[i]

4). 当循环一次没有之后,说明没有重复的值,直接返回-1

|  |
| --- |
| public int findRepeatNumber2(int[] nums) {  int numsLen = nums.length;  int[] temp = new int[numsLen];  for (int i=0; i < numsLen; i++) {  temp[nums[i]]++;  if (temp[nums[i]] == 2) {  return nums[i];  }  }  return -1;  } |