# 题目

给定一个包含n + 1个整数的数组nums ，其数字都在[1, n]范围内（包括1和n），可知至少存在一个重复的整数。

假设nums只有一个重复的整数，返回这个重复的数。

你设计的解决方案必须不修改数组nums且只用常量级O(1)的额外空间。

示例 1：

输入：nums = [1,3,4,2,2]

输出：2

示例 2：

输入：nums = [3,1,3,4,2]

输出：3

提示：

1 <= n <= 105

nums.length == n + 1

1 <= nums[i] <= n

nums 中只有一个整数出现两次或多次 ，其余整数均只出现一次

进阶：

如何证明 nums 中至少存在一个重复的数字?

你可以设计一个线性级时间复杂度 O(n) 的解决方案吗？

# 分析

## 方法一：二分查找

class Solution {

public:

int findDuplicate(vector<int>& nums) {

int n = nums.size();

int left = 1, right = n - 1, ans = -1;

while (left <= right) {

int mid = (left + right) >> 1;

int cnt = 0;

for (int i = 0; i < n; i++) {

// 判断当前的数值是否小于等于中间位置

cnt += nums[i] <= mid;

}

if (cnt <= mid) {

left = mid + 1;

} else {

right = mid - 1;

ans = mid;

}

}

return ans;

}

};

## 方法二：位运算

## 方法三：双指针/快慢指针