# 题目

在大小为2N的数组A中有N+1个不同的元素，其中有一个元素重复了N次。

返回重复了N次的那个元素。

示例 1：

输入：[1,2,3,3]

输出：3

示例 2：

输入：[2,1,2,5,3,2]

输出：2

示例 3：

输入：[5,1,5,2,5,3,5,4]

输出：5

提示：

4 <= A.length <= 10000

0 <= A[i] < 10000

A.length 为偶数

# 分析

## 方法一：计数

**思路：**

在大小为2N的数组中有N+1个不同的元素，其中有一个元素重复了N次，表明：其他的元素都没有重复，则只要在计数的过程中找到次数== 2就停止即为该元素。

**代码：**

class Solution {

public:

    int repeatedNTimes(vector<int>& nums) {

        vector<int> v(10000);

        for(int i=0;i<nums.size();i++)

        {

            v[nums[i]]++;

            if(v[nums[i]] > 1)

            {

                return nums[i];

            }

        }

        return 0;

    }

};

另一种写法：

class Solution {

public:

    int repeatedNTimes(vector<int>& nums) {

        vector<int> v(10000);

        for(int i=0;i<nums.size();i++)

        {

            if(v[nums[i]] == 1)//第一次循环这里不可能为1            {

                return nums[i];

            }

            v[nums[i]]++;

        }

        return 0;

    }

};

## 方法二：判断相等

**思路：**

逐个比较，出现相等的元素就返回。

**代码：**

int repeatedNTimes(vector<int>& A) {

for(int i = 0;i<A.size();++i)

for(int j = i+1;j<A.size();++j)

if(A[i] == A[j]) return A[i];

return 0;

}