# 题目

数组nums包含从0到n的所有整数，但其中缺了一个。请编写代码找出那个缺失的整数。你有办法在O(n)时间内完成吗？

**注意：**本题相对书上原题稍作改动

**示例 1：**

输入：[3,0,1]

输出：2

**示例 2：**

输入：[9,6,4,2,3,5,7,0,1]

输出：8

# 分析

## 方法一：排序+比较

先对原来数组排序，然后对应的下标与数组中的nums[i]比较，如果不相等，则返回这个值，否则返回下一个值。

**代码：**

方法一：

class Solution {

public:

int missingNumber(vector<int>& nums) {

std::sort(nums.begin(),nums.end());

int tmp = 0;

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

{

tmp = i;

if(i != nums.at(i))

return i;

}

return tmp+1; //或者return nums.size();

}

};

## 方法二：位运算

思路：

利用异或的特性：res = res ^ x ^ x。对同一个值异或两次，那么结果等于它本身，所以我们对res从0-nums.length进行异或，同时对nums数组中的值进行异或，出现重复的会消失，所以最后res的值是只出现一次的数字，也就是nums数组中缺失的那个数字。

代码：

class Solution {

public int missingNumber(int[] nums) {

int res = 0;

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

res ^= i;

res ^= nums[i];

}

res ^= nums.length;

return res;

}

}