



链滴

只出现一次的数字

作者: [someone31851](#)

原文链接: <https://ld246.com/article/1546340481648>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



给定一个**非空**整数数组，除了某个元素只出现一次以外，其余每个元素均出现两次。找出那个只出现一次的元素。

说明：

你的算法应该具有线性时间复杂度。你可以不使用额外空间来实现吗？

示例 1:

输入: [2,2,1]

输出: 1

示例 2:

输入: [4,1,2,1,2]

输出: 4

解法1

使用Hash表，建立一个元素 - 出现次数的映射关系，然后再遍历一遍数组找出出现次数唯一的元素。

```
public int singleNumber(int[] nums) {  
    Map<Integer, Integer> map = new HashMap<>();  
    for (int i = 0; i < nums.length; i++) {  
        Integer value = map.get(nums[i]);  
        map.put(nums[i], (value == null ? 0 : value) + 1);  
    }  
}
```

```

    for (Map.Entry<Integer, Integer> entry : map.entrySet()) {
        if (entry.getValue() == 1) {
            return entry.getKey();
        }
    }

    return 0;
}

```

解法2

在解题时应该充分考虑题目所给的条件。

比如“给定一个整数数组，除了某个元素外其余元素均出现两次”，我们由此可以知道，若该数组有，且有一个元素只出现一次，以步数2向后遍历，那么一定会存在 $a[i] \neq a[i+1]$ 。

```

public int singleNumber1(int[] nums) {

    Arrays.sort(nums);

    for (int i = 0; i < nums.length; i = i + 2) {
        if (i + 1 >= nums.length) {
            return nums[i];
        }
        if (nums[i] != nums[i + 1]) {
            return nums[i];
        }
    }

    return -1;
}

```

解法3

除了上面两个较为普通的解法，还有一个比较不容易想到的解法。

根据计算机基础可以知道：

& 两者同时为真才为真

```

0010 0100
&
0010 0100
=
0010 0100

```

由以上可得知，相同数字做&运算，会得到相同的数字。

| 两者一者为真就为真

```

0010 0100
|
0010 0100
=
0010 0100

```

由以上可得知，相同数字做|运算，会得到相同的数字。

^ 相同为假，不同为真

```
0010 0100
^
0010 0100
=
0000 0000
```

由以上可得知，相同数字做^异或运算，会得到0。

由此延伸到题目中，可以得知，若存在一个数字只出现一次，那么该数组所有元素异或结果大于0.

```
public int singleNumber2(int[] nums) {
    int res = 0;
    for (int i : nums) {
        res ^= i;
    }
    return res;
}
```