

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了三次。找出那个只出现了一次的元素。

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： [2,2,3,2]

输出： 3

示例 2:

输入： [0,1,0,1,0,1,99]

输出： 99

解题思路

Single Number系列的第二题，第一题请看 [Leetcode 136 Single Number](#) 仅出现一次的数字 。

实际上136题是137题都是一个特例，更抽象的题目是，有一个数组，其中有一个元素出现了 x 次，其余所有元素都出现了 y 次($x < y$)，找出这个出现了 x 次的元素。关于这个问题，我们在另一篇文章里进行讨论（请看[Leetcode Single Number 扩展](#)）。这里我们只讨论137这个题目。

由于除去目标元素`target`之外，所有元素都出现3次，假设出现3次的元素有 n 个，这样的话假如我们统计所有元素的某一位（比如最后一位），其一共有 $3n+1$ 个二进制位。因为对与同一个元素来说，其所有的二进制位一定是相同的，所以对这些元素的某一位来说一定是以3个1或3个0为单位出现的，即 $3n+1$ 个二进制位中一定是 $3x$ 个1和 $3y$ 个0，其中 $x+y=n$ ，再外加一个`target`对应的二进制位（1或0都有可能）。综上所述，我们可以统计所有数字每一位上1的个数，对3取模，如果为1就说明`target`对应位为1，否则为0。

下面问题就是如何统计每一位上1的个数，一个比较好的方法就是采用位运算来处理，当个数满3时就清零（当然这是参考的网上大神们的思路）。

我们用三个整数`one`，`two`，`three`的二进制位来分别表示32位整数某一位上1出现次数是否为1次、2次、3次，举例，假如：

```
one = 1 --- 0x00000001 --- 00000000 00000000 00000000 00000001
```

则表示当前统计情况下最低位出现1的次数为1次

```
two = 3 --- 0x00000003 --- 00000000 00000000 00000000 00000011
```

则表示当前统计情况下最低位出现1的次数为2次，倒数第二位出现1的次数为2次

```
three = 4 --- 0x00000004 --- 00000000 00000000 00000000 00000100
```

则表示当前统计情况下最倒数第三位出现1的次数为3次

1
2
3
4
5

大神们说one two three可以称之为掩码。

有了如上逻辑后，我们可以遍历所有的数字，对于每个数字，操作其所有的二进制位，来更新one two three三个数字。当我们遍历完所有数字时，由于除去target只出现一次外，其余元素都是以3为单位出现的，所以可以知道one中存储的二进制位代表的数字就是target。

对于实际代码中，one two three三者的更新，有以下两个版本，版本一是我自己想出来的，更通用，而且更接近于上面提出的更一般的问题的解决思路（代码一）；版本二是网上大神们的代码，只能说更巧妙一些，但是实际上都一样（代码二）。详细请看代码部分中的注释。

作者：Orange橘子洲头

来源：CSDN

原文：https://blog.csdn.net/smile_watermelon/article/details/47748227?utm_source=copy

版权声明：本文为博主原创文章，转载请附上博文链接！

代码

```
int singleNumber(int* nums, int numsSize) {
    int one = 0, two = 0, three = 0, tmp;
    while (numsSize) {
        // 获取当前要统计的数字
        tmp = *(nums + --numsSize);
        // 之前已经出现2次1的位且当前数字中又为1，则更新为出现3次
        three = two & tmp;
        // 清空上一步操作中出现3次1的位
        two &= ~three; // 在two中清空
        tmp &= ~three; // 当前数字也要清空，以防重复统计
        // 之前已经出现1次1的位且当前数字中又为1，则更新为出现2次
        two |= one & tmp;
        // 之前没有出现且当前数字中为1，更新为出现1次
        // 或者之前出现次数为1次，且当前数字中没有出现，仍旧为1次
        one ^= tmp;
    }
    return one;
}
```

作者：Orange橘子洲头

来源：CSDN

原文：https://blog.csdn.net/smile_watermelon/article/details/47748227?utm_source=copy

版权声明：本文为博主原创文章，转载请附上博文链接！