剑指 Offer 56 - I. 数组中数字出现的次数

Tuesday, June 14, 2022 9:48 AM

https://leetcode.cn/problems/shu-zu-zhong-shu-zi-chu-xian-de-ci-shu-lcof/
中等
一个整型数组 nums 里除**两个数字之外**，其他数字都出现了两次。
请写程序找出这两个只出现一次的数字。
要求时间复杂度是**O(n)**，空间复杂度是**O(1)**。

示例 2：
输入：nums = [1,2,10,4,1,4,3,3]
输出：[2,10] 或 [10,2]

我的思路：
O(N)O(N)还能哈希表，O(nlogn)O(1)还能先排序。
算了，我是笨蛋 >_<

看了题解：
算了，我真是笨蛋，
算了算了，不会还有笨蛋看不懂题解吧，：）
（没错，说的就是.. 还有谁在看呢 hhh）

题解：
1. 异或（^）的特性，位操作，同0异1；0与任何数异或，还是那个数； a^b^a=b
2. 设 res=0；res=res ^ 数组内的每一个数，得到的res 就是 目标两个数的异或结果。
3. 1的二进制为 000...0001，根据 异或同0异1，对res & 1操作，
   若结果为1则，目标数 x,y 在末位不同，
   若结果为0，将1左移一位变成 000...0010，再和res &操作，
   不断重复上步操作，直到找到结果为1。
   记录下 1 当时经左移变化后的数为 mark。（即记录下了二进制下x与y的一个不同位，以此区分x, y）
4. 此时，再次遍历数组，每个数先与 mark &操作，得到的数要么是0，要么不是（因为mark二进制只有1位为1，其他位为0）
   是0的为一组，依次和 x 异或操作 x=x^该数（x,y初始化都为0）
   不是0的为一组，和y异或操作 y=y^该数。
5. 最终得到的结果就是 x,y 两个唯一不同的数

简单的吧。 ^_^

```
public int[] singleNumbers(int[] nums) {
    // xor用来计算nums的异或和
    int xor = 0;
    for(int num : nums)
```

```
            xor ^= num;
        // 此时 xor = x^y

        //设置mask为1，二进制为00...0001
        int mark = 1;
        // 因为x，y不同，故异或结果至少有一位为1，求出不同的一位。
        while((xor & mark)==0){
            mark <<= 1;
        }
        // 此时 mark= 0..010..00，标记了不同的那位
        // 初始化为0，其与任何数异或都为该数 0^a=a
        int x=0, y=0;

        for(int num : nums){
            // 根据&是否为0区分将两个数字分区，并分别求异或和
            if((num & mark)==0){
                a ^= num;
            }else{
                b ^= num;
            }
        }
        return new int[]{a,b};
    }
```