

小米2013校园招聘笔试题

题目：一个数组里，除了三个数是唯一出现的，其余的都出现偶数个，找出这三个数中的任一个。比如数组元素为【1, 2,4,5,6,4,2】，只有1,5,6这三个数字是唯一出现的，我们只需要输出1,5,6中的一个就行。

下面是我的思路

：这个数组元素个数一定为奇数，而且那要求的三个数一定不可能每一bit位都相同，所以我们可以找到其中一个bit位不同，可以把那三个数字分出来，而且可以很推出三个数肯定可以分到两组不同的数组里面去，基于这样的思路就可以找出这三个不同的数字。

找到三个数字一个数的第一个bit位（这里是从右到左算）和其它二个不一样的数就行

如1,5,6的二进制分别为0001,0101,0110。因为6的第一位为0，而其他的为1，用我的程序第一个输出的就是6了。程序如下：

```
#include <stdio.h>
//得到第i位的二进制
#define isON(n, i) ((n) & 1 << (i))

// Author: 397090770
// E-mail: wyphao.2007@163.com
// 转载请注明出处

void findTheSingleNumber(int *arr, int size){
    int tempA = 0, tempB = 0;
    int countA = 0, countB = 0;//用来计数用
    int i = 0, j = 0;

    for(i = 0; i < 32; i++){ //32位平台上，int只有32位
        tempA = tempB = countA = countB = 0;
        for(j = 0; j < size; j++){//遍历数组
            if(isON(arr[j], i)){
                tempA ^= arr[j];
                countA++;
            }else{
                tempB ^= arr[j];
                countB++;
            }
        }
    }

    if(countA & 0x1){//奇数
        if(0 == tempB){
```

```
    continue;
}else{
    printf("%d\n", tempA);//肯定是不同的数字
    break;
}
}else{
    if(0 == tempA){
        continue;
    }else{
        printf("%d\n", tempB);
        break;
    }
}
}

int main(){
    int arr[] = {
        /*1, 3, -9, 2, 1, 2, -10*/
        1, 2, 4, 5, 6, 4, 2
    };

    findTheSingleNumber(arr, sizeof(arr) / sizeof(arr[0]));
    return 0;
}
```

时间复杂度为O(N)。输出为
6。（如果要全部输出不同的数，方法和上面的类似，这里就不给出了）

哪位大牛还有别的解法吗？

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接：[【】（）](#)