

leetcode Single Number II - 位运算处理数组中的数

题目描述：

给定一个包含n个整数的数组，除了一个数出现一次外所有的整数均出现三次，找出这个只出现一次的整数。

题目来源：

<http://oj.leetcode.com/problems/single-number-ii/>

题目分析：

对于除出现一次之外的所有的整数，其二进制表示中每一位1出现的次数是3的整数倍，将所有这些1清零，剩下的就是最终的数。

用ones记录到当前计算的变量为止，二进制1出现“1次”（mod 3 之后的 1）的数位。用twos记录到当前计算的变量为止，二进制1出现“2次”（mod 3 之后的 2）的数位。当ones和twos中的某一位同时为1时表示二进制1出现3次，此时需要清零。即用二进制模拟三进制计算。最终ones记录的是最终结果。

时间复杂度：O(n)

示例代码：



```
int singleNumber(int A[], int n) {
    int ones = 0, twos = 0, xthrees = 0;
    for(int i = 0; i < n; ++i) {
        twos |= (ones & A[i]);
        ones ^= A[i];
        xthrees = ~(ones & twos);
        ones &= xthrees;
        twos &= xthrees;
    }

    return ones;
}
```



扩展一：

给定一个包含n个整数的数组，除了一个数出现二次外所有的整数均出现三次，找出这个只出现二次的整数。

ones记录1出现一次的数，twos记录1出现2次的数，容易知道twos记录的即是最终结果。

扩展二：

给定一个包含n个整数的数组，有一个整数x出现b次，一个整数y出现c次，其他所有的数均出现a次，其中b和c均不是a的倍数，找出x和y。使用二进制模拟a进制，累计二进制位1出现的次数，当次数达到a时，对其清零，这样可以得到b mod a次x，c mod a次y的累加。遍历剩余结果（用ones、twos、fours...变量表示）中每一位二进制位1出现的次数，如果次数为b mod a 或者 c mod a，可以说明x和y的当前二进制位不同（一个为0，另一个为1），据此二进制位将原数组分成两组，一组该二进制位为1，另一组该二进制位为0。这样问题变成“除了一个整数出现a1次（a1 = b 或 a1 = c）外所有的整数均出现a次”，使用和上面相同的方式计算就可以得到最终结果，假设模拟a进制计算过程中使用的变量为ones、twos、fours...那么最终结果

可以用ones | twos | fours ...表示。

posted @ 2013-10-05 21:35 代金桥 阅读(29) 评论(2) 编辑 收藏