

二

08 16进制又是个什么鬼？ - 16进制的讲解

十六进制这个词可能你第一次听会觉得很吓人，但是当你了解了它之后，很快你就会发现它是你的好朋友并且会爱上他。现在让我们一起来看看这个小可爱。

我们前面已经讲过了二进制，8进制以及你从小就知道的十进制，让我们来对比一下这些数值。对于从0 到9，对于16进制来说其实是没有区别的。

Binary(二进制)	十进制	16进制
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

复制

看到上面这个ABCDEF是不是感觉很神奇的操作，数字不够，字母来凑。可能你会觉得很奇怪，但是习惯了之后，你就会觉得还可以。那我们来再查看一下，十进制里的16怎么用16进制表示呢。我相信聪明的你一定想到了那就是“10”，当然这个10不是常见的10。

十进制怎么转换成16进制呢

假设我们现在要把数字95转成用16进制表示。和我们前面的方法类似，我们先用95除以16能得到什么呢，是不是等于5然后余数是15。所以答案就是5F。因为F表示的就是15对不对。

16进制怎么转换成10进制呢

我们还使用上面的案例来进行一下相反的操作, $5F = 5 * 16 + 15 * 1 = 95$ 。因为F表示15, 所以使用 $15 * 1$ 。其实你明白原理之后就会明白, 这个很简单。

你可能会问哪里可以使用到呢, 我给你一个鲜活的例子。那就是网页设计, 不知道你有没有做过前端开发, 在HTML和CSS上就是使用16进制来表示网页上的特定颜色。

到这里, 你应该已经完全掌握了各个进制转换的精华, 但是作为一个严谨的工程师, 也为了让你们在将来面试的时候如果遇到这种进制的题可以对答如流。让我们来一起看一下面试可能会问什么类型的题。

面试问题1

给你 $2n + 1$ 个数字, 所有的数都会出现两次除了有一个数只会出现一次。找到这个数?

看一下这个例子

Input: [1,1,2,2,3,4,4]

Output: 3

复制

我希望你有看过我个人的文章, 你就会了解我是怎么讲算法的, 首先我们考虑一下最暴力的解法, 你怎么做, 你可以使用一个hashmap, 把所有的数循环一遍, 然后在查找这个hashmap, 看哪个value是1, 那个key就是你要找的对不对。但是这里你无形中使用了一个额外的数据结构对不对, 也就是你的空间复杂度是 $O(n)$ 。那我们下面来看一下可不可以优化这个。把这个空间复杂度从 $O(n)$ 降为 $O(1)$ 。

我们先来回顾一下小知识, 还记得XOR吗, XOR的特点是什么, 当0遇到0, 或者1遇到1, 就会变成0。就好像消消乐一样, 那你再扩展一下这个概念, 是不是任意两个相同的数字异或 (XOR) 之后是不是都是0, 你想一下比如说数字3, 二进制是011, 那011异或011之后是不是0。所以我们来看一下上面的这个例子。1和1异或之后是0, 2和2之后异或之后是0, 4和4异或之后是0, 最后得到的结果是不是就是我们要找的那个数也就是3。

可能你会有另一个疑问就是说顺序是不是有影响, 让我们来看一下, 3和4异或等于多少 011 和 $100 = 111$, 然后在和3异或之后也就是 111 和 $011 = 100$ 也就是4, 如果和4异或 111 和 100 也就是 $011 = 3$ 。结果是不是和你想的一样。

```
public int findSingleNumber(int[] number) {  
    if(number == null || number.length == 0) {  
        return -1;  
    }  
}
```

```
int rst = 0;
for (int i = 0; i < number.length; i++) {
    rst ^= number[i];
}
return rst;
}
```

复制

面试问题2

给你两个数，你来决定需要翻转几个bits可以使两个数相同

看一下这个例子

输入：n = 30, m = 14

输出：1

因为30的二进制是11110， 14的二进制是01110，所以只需要翻转1个bit就可以使结果相同。

复制

是不是还是需要用到异或的特性，你异或这两个数，然后不断的使用>>>向右移，直到最后结果是0。然后你就可以知道有几位不同了。这里就不给你们写代码了。因为代码很简单，主要是原理。感兴趣的你可以自己挑战一下。

当然这样的例题或者是面试题还有很多，在练习题里，会再给你们列出几道题，其实二进制也好，8进制和16进制也好，并没有什么特别难得问题，主要是转化你的思维，毕竟是计算机的思考方式。是不是你理解了之后，将来变成机器人的时候会更容易点（哈哈，开玩笑）。但是这个思维对你来说很重要，需要把基础还有原理理解清楚，然后根据特性来进行使用。主要的是异或运算，你也能看出来使用异或的地方很多，而且也是最有用的。希望你借助这一章来提升你对二进制的理解。好，那我们今天就到这里。

```
public int bitFilp(int number1, int number2) {
    int count = 0;
    for (int c = number1 ^ number2; c != 0; c = c >>> 1) {
        count += c & 1;
    }
    return count;
}
```

[上一页](#)

[下一页](#)