

面试题15：二进制中1的个数

2018/8/10 12:18:56

题目描述

输入一个整数，输出该数二进制表示中1的个数。其中负数用补码表示。

解题思路

1. **Java**，因为题目给出的是int类型，java中的int类型用32位表示，把这个数依次无符号右移（考虑到负数所以选择无符号右移）然后与1作与运算。如果结果为1则count自增。
2. **C++**，如果一个整数不为0，那么这个整数至少有一位是1。如果我们把这个整数减1，那么原来处在整数最右边的1就会变为0，原来在1后面的所有的0都会变成1(如果最右边的1后面还有0的话)。其余所有位将不会受到影响。

举个例子：一个二进制数1100，从右边数起第三位是处于最右边的一个1。减去1后，第三位变成0，它后面的两位0变成了1，而前面的1保持不变，因此得到的结果是1011。我们发现减1的结果是把最右边的一个1开始的所有位都取反了。这个时候如果我们再把原来的整数和减去1之后的结果做与运算，从原来整数最右边一个1那一位开始所有位都会变成0。如 $1100 \& 1011 = 1000$ 。也就是说，把一个整数减去1，再和原整数

做与运算，会把该整数最右边一个1变成0.那么一个整数的二进制有多少个1，就可以进行多少次这样的操作。

Java

```
1 public class Solution {
2     public int NumberOf1(int n) {
3         int count = 0;
4         for(int i = 0; i < 32; i++){
5             if((n >> i & 1) == 1)
6                 ++count;
7         }
8         return count;
9     }
10 }
```

C++

```
1 class Solution {
2 public:
3     int NumberOf1(int n) {
4         int count = 0;
5
6         while(n){
7             ++count;
8             n = (n - 1) & n;
9         }
10        return count;
11    }
12 };
```