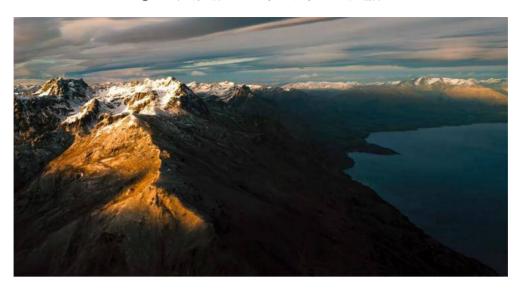
剑指Offer (十一): 二进制中1的个数

⑤ 2017年11月25日 11:06:35 ♀ 4 ◎ 4,618 °C ♣ 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台: 牛客网

书籍下载: 共享资源

二、题目

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。

1、思路

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,从原列整数最右边一个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.系公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。

2、代码

C++:

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

在Python中,由于负数使用补码表示的,对于负数,最高位为1,而负数在计算机是以补码存在的,往右移,符号位不变,符号位1往右移,最终可能给出现全1的情况,导致死循环。与0xffffffff相与,就可以消除负数的影响。

```
Pytho
   # -*- coding:utf-8 -*-
   class Solution
        def NumberOf1(self, n):
4
            # write code here
5
            count = 0
6
            if n<0:
                n = n & 0xffffffff
            while n:
8
                count += 1
10
                n = n \& (n-1)
11
            return count
```

或者可以使用一个更直观的方法,直接位移即可,代码如下:

```
Pytho

1 #-*- coding:utf-8 -*-
2 class Solution:
3 def NumberOf1(self, n):
4 # write code here
5 return sum([(n >> i & 1) for i in range(0,32)])
```



微信公众号

分享技术,乐享生活:微信公众号搜索「JackCui-AI」关注一个在互联网摸爬滚打的潜行者。

哀其不幸, 怒其不争。--- 鲁迅