476 Number Complement

2018年4月12日 15:51

Given a positive integer, output its complement number. The complement strategy is to flip the bits of its binary representation.

Note:

1. The given integer is guaranteed to fit within the range of a 32-bit signed integer.
2. You could assume no leading zero bit in the integer’ s binary representation.

Example 1:

Input: 5

Output: 2

Explanation: The binary representation of 5 is 101 (no leading zero bits), and its complement is 010. So you need to output 2.

Example 2:

Input: 1

Output: 0

Explanation: The binary representation of 1 is 1 (no leading zero bits), and its complement is 0. So you need to output 0.

来自 <https://leetcode.com/problems/number-complement/description/>

给定一个正整数，输出它的补数。补数是对该数的二进制表示取反。

注意:

1. 给定的整数保证在32位带符号整数的范围内。
2. 你可以假定二进制数不包含前导零位。

Solution for Python3:

```
class Solution1:
    def findComplement(self, num):
        """
        :type num: int
        :rtype: int
        """
        mask = ~0
        while num & mask:
            mask <<= 1
        return ~mask & ~num

class Solution2:
    def findComplement(self, num):
        """
        :type num: int
        :rtype: int
        """
        mask = num
        mask |= mask >> 1
        mask |= mask >> 2
        mask |= mask >> 4
        mask |= mask >> 8
        mask |= mask >> 16
        return num ^ mask
```

Solution for C++:

```
class Solution {
public:
    int findComplement(int num) {
        int mask = ~0;
        while (num & mask)
            mask <<= 1;
        return ~mask & ~num;
    }
};
```

Appendix:

二进制数忽略前导0取反操作:

1) '0101' 在忽略前导0取反可以用'0101'^'0111',即找到一个二进制数'0111'，该数是忽略前导0后所有位取1。
2) 即传播最高位的1到后面所有位:

n = 0000 0101 1010 1001

要得到　m = 0000 0111 1111 1111

mask = n	mask=0000 0101 1010 1001
mask |= mask >> 1	0000 0101 1010 1001 0000 0010 1101 0100 mask=0000 0111 1111 1101
mask |= mask >> 2	0000 0111 1111 1101 0000 0001 1111 1111 mask=0000 0111 1111 1111
mask |= mask >> 4	
mask |= mask >> 8	
mask |= mask >> 16	mask=0000 0111 1111 1111