

# [LeetCode] Number of 1 Bits & Reverse Integer - 整数问题系列

原创

Eastmount

2015-09-14 05:15:15

2768

☆ 收藏

版权

分类专栏: [LeetCode](#)

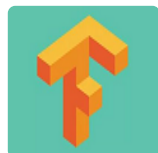
文章标签: [leetcode](#)

[整数处理](#)

[整数翻转](#)

[整数求1](#)

[整数判断重复](#)



## Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、神...



Eastmount

¥9.90

[订阅博主](#)

### 目录:

- 1.Number of 1 Bits - 计算二进制1的个数 [与运算]
- 2.Contains Duplicate - 是否存在重复数字 [遍历]
- 3.Reverse Integer - 翻转整数 [int边界问题]
- 4.Excel Sheet Column Number - Excel字符串转整数 [简单]
- 5.Power of Two & Happy Number - 计算各个位数字 [%10 /10]

## 一.Number of 1 Bits

### 题目概述:

Write a function that takes an unsigned integer and returns the number of '1' bits it has (also known as the [Hamming weight](#)).

For example, the 32-bit integer '11' has binary representation

00000000000000000000000000001011

so the function should return 3.

### 解题方法:

三种方法包括:

- 1.依次和0x1进行&与运算，若结果为1则加1，表示1个数，再右移；
- 2.推荐的方法， $n \& (n-1)$ ，直到为0，次数为1的个数；
3. $n$ 取2模，依次判断个位是否为1，在 $n/2$ 移位，常规方法。

其中uint32\_t为32位无符号类型数据，[参考地址](#)

**Power of Two**题目也可以通过`return (n > 0) && !(n & (n - 1))`一句话实现。

**Reverse Bits**题目也可以<<移位实现。

## 我的代码：

```
/*
 * uint32_t为32位无符号类型数据 思路：数字移位
 */
int hammingWeight(uint32_t n) {
    //第一种方法 考查移位及与运算&
    int result=0, left=0;
    while(0 != n)
    {
        left = n & 0x1;
        result += left;
        n = n >> 1;
    }
    return result;

    //第二种方法
    int re = 0;
    while(0 != n)
    {
        n = n&(n - 1);
        ++re;
    }
    return re;

    //第三种方法 求2模
    int count = 0;
    while (n)
    {
        if (n % 2 == 1)
        {
            ++count;
        }
        n /= 2;
    }
    return count;
}
```

## 二.Contains Duplicate

### 题目概述：