

剑指Offer（四十八）：不用加减乘除的加法

🕒 2018年1月19日 12:21:20 📄 5 🌡 2,887 °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

写一个函数，求两个整数之和，要求在函数体内不得使用+、-、\*、/四则运算符号。

1、思路

首先看十进制是如何做的：5+7=12，

可以使用三步走：

第一步：相加各位的值，不算进位，得到2。

第二步：计算进位值，得到10。如果这一步的进位值为0，那么第一步得到的值就是最终结果。

第三步：重复上述两步，只是相加的值变成上述两步的得到的结果2和10，得到12。同样我们可以

三步走的方式计算二进制值相加：5-101，7-111

第一步：相加各位的值，不算进位，得到010，二进制每位相加就相当于各位做异或操作，101^111。

第二步：计算进位值，得到1010，相当于各位做与操作得到101，再向左移一位得到1010，(101&111)<<1。

第三步：重复上述两步，各位相加 010^1010=1000，进位值为100=(010&1010)<<1。

继续重复上述两步：1000^100 = 1100，进位值为0，跳出循环，1100为最终结果。

2、代码

C++:

C+

```
1 class Solution {
2 public:
3     int Add(int num1, int num2)
4     {
5         return num2 ? Add(num1 ^ num2, (num1 & num2) << 1) : num1;
6     }
7 };
```

Python:

Pytho

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def Add(self, num1, num2):
4         # write code here
5         MAX = 0xffffffff
```

```
6     mask = 0xffffffff
7     while num2 != 0:
8         num1, num2 = (num1 ^ num2), ((num1 & num2) << 1)
9         num1 = num1 & mask
10        num2 = num2 & mask
11    return num1 if num1 <= MAX else ~(num1 ^ mask)
```

---



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。

领袖和跟风者的区别就在于创新。--- 乔布斯