文章编号:1005－0523(2005)02－0155－03

# 整数运算中的问题和解决方法

周尚超

(华东交通大学 基础科学学院,江西 南昌 330013)

**摘要**:给出整数运算中存在的问题和解决办法.

**关 键 词**:同余式;模运算;算法

**中图分类号**:O156.1,TP311.1 **文献标识码**:A

## 1 引 言

在各种计算机语言中,整数是在一定范围内运行的.如果超出范围,就会出错.本文在 C 语言中讨论此问题并提出解决的办法.

例 1. 计算组合数 C[m][n],它表示从 m 个元素中取 n 个元素的种数.如果将它定义为 unsigned int 型整数,它表示的最大整数为 m＝4294967295.C[80000][2]＝80000 * 79999/2＝ 3199960000/.计算机的计算结果是错的,这是因为它先计算 80000 * 79999,此值已超过它表示的最大整数 m＝4294967295.如将运算改为(80000/2) * 79999,则计算结果正确.

在数论,组合数学和密码学和计算机科学中经常要用到整数取模运算.各种计算机语言都设置了整数取模运算.C 语言中的取模运算符为%.这些问题可归纳为:

如果参与运算的数和运算结果都小于和等于某最大值 m 的整数.如何使运算结果正确.

例 1 的运算可归结为计算 A＝a*b/s,其中 a,b,s 的值和运算结果 A 都是不超过某最大值 m 的正数.计算方法如下:

```
d＝gcd(a,s); //gcd 表示最大公因数
a＝a/d;
s＝s/d;
b＝b/s;
A＝a*b;
```

在取模运算中,所有参与运算的数和运算结果都小于某数 m.要解决此问题只要有正确的加法运算和乘法运算就行了.下面给出加法运算的函数 add 和乘法运算的函数 mult(m 为模).参与运算的数为 0,1,...,m－1.令 M＝m－1.

计算(a＋b)% m (a,b<m)的算法(函数)如下:

如果 a＋b 的值大于 M,则 运算(a＋b)%m 的值为 a＋b－m. 由于不能使用大于或等于 m 的值,直接写 s＝(a＋b)%m 就会出错,写为 a－(M－b＋1)则正确(b>0).

```
int add(int a,int b,int M)
{int  s;
if(a==0||b==0)s＝a＋b;
else if(a>＝M－b＋1)s＝a－(M－b＋1);
else s＝a＋b;
return s;
}
```

对于乘法运算(m 为模),不能使用大于或等于 m 的数.先以 a＝51, b＝23 , m＝100. 计算计算 a * b%m 方法如下

**收稿日期**:2005－01－03

**基金项目**:江西省自然科学基金项目,华东交通大学科技基金项目.

**作者简介**:周尚超(1948－),男,云南蒙自人,教授.

$51*23=(3+12*4)*23=3*23+12*92=69+12*(-8)=69-96=69+4=73$.

根据此例的方法,我们给出计算 a*b%m, a,b<m 的函数 mult,在此函数中出现的数都是小于 m 的数.

```
int mult(int a,int b,int M)
{ int s,r,d,t,k; //M=m-1
if(a==1)return b;
if(a==0||b==0)return 0;
if(b==1)return a;
if(a==M)return M-b+1;
if(b==M)return M-a+1;
if(a<=M/b)return a*b;
k=0;
if(b>M-b){k=1;b=M-b+1;} //使 d=M/b>1 ,s<a/2
d=M/b; t=M%b; //M=d*b+t d*b=M-t
s=a/d; r=a%d; // a=s*d+r a*b=(s*d+r)*b =r*b+(M-t)*s
if(k==0) add(r*b,mult(s,M-t,M),M); //d*b<=M, r<d, r*b<M
else return return M-add(r*b,mult(s,M-t),M)+1;
}
```

mult 是一个递归函数,每次调用后 s 的值减少 1/2 以上,计算速度是很快的.

运算中不能使用数 m,但有时会遇到运算 m/a.

例 2. 如果已知 a ,要求模 m 的逆元 b,即使 a*b=1(mod m). 由数论,a 存在逆元的充分必要条件是 gcd(a,m)=1. 这里 gcd 表示最大公因数. 计算逆元 b 的算法如下.

```
r=1; b1=0;b2=1;
While(r! =0)
{ q=m/a; r=m%a;
if(r! =0)
{
m=a; a=r; t=b2; b2=b1-b2*q; b1=t;
continue;
}
break;
}
if(a==1)b=b2;
else b=0; //逆元不存在
```

在例 2 中,用到 q=m/a;r=m%a; 在不使用 m 的情况下,要计算 q=m/a 和 r=m%a,程序如下

```
// m/a=(M+1)/a; m%a=(M+1)%a
q=M/a; if(M%a==a-1)q++;
r=M%a; r++; if(r==a)r=0;
```

因此例 2 中,计算逆元 b 的算法如下.

```
M0=M; //M=m-1;
r=1; b1=0;b2=1;
flag=0;
while(r! =0)
{
if(flag==0)
{
q=M/a; //第 1 次计算时 flag=0 ,M 为最大值 m-1,计算的是 m/a,m%a
if(M%a==a-1)q++;
r=M%a; r++;if(r==a)r=0;
flag=1;
}
else {q=M/a; r=M%a;}//M 已不是最大值 m-1
if(r! =0)
{
M=a; a=r; t=b2; d=mult(b2,q,M0);
if(b1>=d)b2=b1-d; else b2=M0-d+b1+1;

b1=t;flag=1;
continue;
}
break;
}
if(a==1)s=b2;
else s=0; //逆元不存
```

以下给出几个计算实例

例 3. 在密码学中,要计算 a^n% m. 当 n 值很大时,使用下列快速算法.

```
b=1;
while(n>1)
{ if(n%==0){n/=2; a=a*a%m;}
else {n--; b=b*a%m;}
}
b=b*a%m;
```

b 的值 即为 a^n%m;

使用 add 和 mult 函数,则程序如下:

```
b=1;
while(n>1)
{ if(n%==0){n/=2; a=mult(a,a,M);}
else {n--; b=mult(b,a,M);}
}
b=mult(b, a,M);
```

例 4. 北京大学在线问题(http://acm.pku.edu.cn/JudgeOnline)第 2115 题. 一个超级时钟有 m 个刻度,称为 1 点, 2 点,..., m 点. m 点又称为零点. 从零点出发,每次走 a 点,问能否走到 b 点,如能,要走多少次. 例如对 m=12,从零点出发,每次走 3 点,走 1 次到 3 点,走 2 次到 6 点,还可走到 9 点和 12 点,但不能走到其他点. 如每次走 5 点则可走到 12 个点中的所有点. 因此这题归结于计算方程

ax=b (mod m) , m<=2^32 的解

这题中的 m 可能为最大值 m=2^32, unsigned int 型整数表示的最大值为 M=2^32-1. 利用例 2 中的方法先计算 a 的逆元 s,则其解为 x=s * b (mod m)

# A New Method for Integer Operation

**ZHOU Shang-chao**

(East China Jiaotong University, Nanchang 330013, China)

**Abstract**: This paper studies some problems in integer opration and gaves its sulution.

**Key words**: congruent; modul operation; algorithm.