关于同余与模运算的总结 - CSDN博客

<1>　123456789*987654321 = ()

A: 121932631112635266　　　　B: 121932621112635267

C: 121932631112635268　　　　D: 121932631112635269

解答: 利用公式(ab)mod n ＝(a mod n)（b mod n）mod n，可以得到

123456789 X 987654321 mod 10 = (123456789 % 10) X (987654321%10) %10 = 9

在这里我们介绍以下三个公式:

(a+b)mod n = ((a mod n)+ (b mod n))mod n;

(a-b) mod n = ((a mod n)- (b mod n)+n)mod n;

ab mod n = (a mod n) (b mod n) mod n

注意，在减法中，由于a mod n 可能小于b mod n，需要在结果上加上n，而在乘法中，需要注意a mod n 和 b mod n相乘是否会溢出，因此这里要注意用long 型保存中间结果。像这样：

[cpp]view plaincopy

```cpp
int mul_mod(int a,int b,int n)
{
    a %= n; b %= n;
return (int) ((long)a * b %n);
}
```

<2>大整数取模（sicily 1020）

这里要利用到公式：(a+b) mod (n) = (a mod n) + (b mod n) mod (n);

把大整数写成自左向右的形式：1234 = （（1*10+2）*10+3）*10+4，然后利用前面这个公式，每步取模，算法如下

[cpp]view plaincopy

```cpp
// source code of submission 784219, Zhongshan University Online Judge System
#include
#include
usingnamespace std;
int main()
{
int test,n,i,k,ans,temp;
int bas[106],res[106];
char oper[600];                //oper保存大数
    cin>>test;
while(test--)
    {
        cin>>n;
for(i = 0;i < n;i++)
            cin>>bas[i];
        cin>>oper;
int len = strlen(oper);
for(k = 0;k < n;k++)
        {
            temp = bas[k],ans = 0;
for(i = 0;i < len;i++)      //这里运用公式(a+b)%n = ((a%n)+(b%n))%n;
                ans = (int)(((long)ans*10+(oper[i]-'0'))%temp);
            res[k] = ans;
        }
        cout<<"("<
for(i = 1;i < n;i++)
            cout<<","<
        cout<<")"<
    }
return 0;
}
```

<3> 幂取模(sicily 1294)

这也是用到了同余的性质：xy mod c = (x mod c)* (y mod c) mod c

[cpp]view plaincopy

```
参考自郭嵩山老师的算法课件：
d = 1;
for (i = 1; i <= b; ++i) {
    d = d * a % c;
  }
  cout << d;
```

算法如下：

[cpp]view plaincopy

```
// source code of submission 692835, Zhongshan University Online Judge System
#include
usingnamespace std;
int main()
{
int a,b,c,i,d = 1;
    cin>>a>>b>>c;
for(i = 1;i<=b;i++)
        d = d*a%c;
    cout<<
}
```

<4>模线性方程

题意：输入正整数a,b,n,解方程ax ≡ b (mod n) a,b,n<=10^9 。

解答：

a ≡ b(mod n)的意思是说“a 和 b关于模n 同余 ”，即a mod n = b mod n。而a ≡ b mod n 的充要条件是: （a-b） 是n 的整数倍。 这样，这个问题就变成了ax-b是n的正整数倍。设这个"倍数"是y,则ax - b = ny,即ax - ny= b，因此，这个就回到了解不定方程的问题。

比如给定方程ax +by +c = 0,求出满足这个方程的整数解（x,y）.这里，我们首先来学习扩展欧几里德算法——找出一对整数(x,y)，使得ax+by = gcd(a,b),这里的x,y不一定是整数，也可能是负数或者0，例如gcd(6,15) = 3,6*3 - 15*1 = 3,其中x = 3,y=-1;这个方程还有其他解，比如x = -2,y = 1;以下是一个扩展欧几里德算法的程序：

[cpp]view plaincopy

```
#include
usingnamespace std;
int x,y;
void gcd(int a,int b,int& d,int& x,int& y)
{
if (!b)     { d = a; x = 1; y = 0; }
else         { gcd(b,a%b,d,y,x); y -= x*(a/b); }
}
int main()
{
int a,b,x,y,d;
    cin>>a>>b;
    gcd(a,b,d,x,y);
    cout<<<" "<
return 0;
}
```

可以证明：设a,b,c为任意整数。若方程ax+by = c 的一组整数解为x_0,y_0则它的任意整数解都可以写成（x_0+kb',y_0+ka'），其中a' =a/gcd(a,b),b' = b/gcd(a,b),k为任意整数。

假设对于ax - ny= b，其中a = 6,n = -15,b = 9，即6x+15y = 9,根据欧几里德算法，我们得到6X(-2)+15X1 = 3,两边同时乘以3，即可得到6X(-6)+15X3 = 9，即x = -6,y = 3是6x+15y = 9的一组解。

最后，还有这样一个结论：设a,b,c为任意整数，g = gcd(a,b),方程ax+by = g的一组解是(x_0,y_0),则当c是g的倍数时，ax+by=c的一组解是(x_0c/g,y_0c/g);当c不是g的倍数时无整数解。

——参考文献：《算法竞赛入门经典》，刘汝佳