[关闭]

@LittleClown 2016-05-06 10:51 字数 5958 阅读 0

- **内容目录**

模方程基础

ACM 学习笔记 数论 Baby-Step-Gaint-Step

---

* **原文链接** *

扩展欧几里得算法

问题描述

已知 a,b 为常整数，求方程 $ax+by=\gcd(a,b)$ 的一对整数解?

问题简析

令 $\begin{cases} A=a(\mod b) \\ B=b \end{cases}$，构造方程

$$AX+BY=\gcd(A,B). \tag{1}$$

由于，$\gcd(a,b)=\gcd\left(a(\mathrm{mod}\ \ b),b\right)=\gcd(A,B)$，

$$
\begin{aligned}
\text{又，}\ ax+by&=\left(a-\left\lfloor\frac{a}{b}\right\rfloor\times b+\left\lfloor\frac{a}{b}\right\rfloor\times b\right)\times x+by &(1)\\
&=\left(a-\left\lfloor\frac{a}{b}\right\rfloor\times b\right)\times x+\left(by+\left\lfloor\frac{a}{b}\right\rfloor\times bx\right) &(2)\\
&=\left(a(\mathrm{mod}\ \ b)\right)\times x+b\times\left(y+\left\lfloor\frac{a}{b}\right\rfloor\times x\right) &(3)
\end{aligned}
$$

所以，$\begin{cases}X=x\\Y=y+\left\lfloor\frac{a}{b}\right\rfloor\times x\end{cases}\Rightarrow\begin{cases}x=X\\y=Y-\left\lfloor\frac{a}{b}\right\rfloor\times X\end{cases}$

方程 (1) 的构造与辗转相除有异曲同工之妙，需要说明的是，$\gcd(x,0)=x$。
所以：

1. 当 $b=0$ 时，我们得到一组解 $\begin{cases}x=1\\y=0\end{cases}$；
2. 当 $b\neq 0$ 时，套用辗转相除的框架求出 (X,Y)，再得到 (x,y)。

程序实现

```
typedef long long LL;
void Extend_Euclid(LL a, LL& x, LL b, LL &y, LL& d) { // d=gcd(a,b)
    if( !b ) { d=a; x=1; y=0; }
    else { Extend_Euclid(b, y, a%b, x, d); y -= (a/b)*x; }
}
```

值得一提的是，扩展欧几里得算法解出的一组解是所有解中 $|x|+|y|$ **最小**的；
另外，$\gcd(a,b)$ 可以一起求。

代码中，将 d=a 的执行顺序放在第一位是一个小技巧，因为这样就可以像这样调用函数：

```
Extend_Euclid(a, x, b, d, d);
// 或者
Extend_Euclid(a, d, b, d, y);
```

问题扩展

求方程 $ax+by+c=0$ 有多少对整数解，满足 $x\in[x_1,x_2]$，$y\in[y_1,y_2]$?

由前文可知，**当 $\gcd(a,b)\mid c$ 时，方程有解**。因为方程 $ax+by=\gcd(a,b)$ 的解
乘上 $\frac{c}{\gcd(a,b)}$ 就是此方程的一组解。
由此，我们可以推出 $ax+by=k\cdot\gcd(a,b)$，所以，**当 $\gcd(a,b)\nmid c$ 时，方程无解**。

那有多少个解呢?
如果 (x_0,y_0),(x_1,y_1) 是方程的两个解，那么 $a(x_1-x_0)+b(y_1-y_0)\equiv 0$。
令 $a'=\dfrac{a}{\gcd(a,b)}$，$b'=\dfrac{b}{\gcd(a,b)}$，显然，$\gcd(a',b')=1$。
所以，$(x_1-x_0)\mid b'$，$(y_0-y_1)\mid a'$，并且 $\dfrac{x_1-x_0}{b'}\equiv\dfrac{y_0-y_1}{a'}$。

一般地，若 (x_0,y_0) 是方程 $ax+by+c=0$ 一组解，那么 (x_0+kb',y_0-ka') 也是一组解。

其中，$b'=\dfrac{b}{\gcd(a,b)}$，$a'=\dfrac{a}{\gcd(a,b)}$，k 为任意整数。

剩下的问题就很简单了，不再赘述。

逆元

问题描述

对于任意整数 $A,\ N$，如果存在一个整数 B 使得 $A\cdot B\equiv 1(\bmod\ C)$，
那么，A 和 B 称作模 C 意义下的互为逆元，记做 $A^{-1}=B$ 或 $B^{-1}=A$。

问题简析

1. 如果 $\gcd(A,N)=1$，根据欧拉定理，有 $A^{-1}=N^{\varphi(N)-1}$，特别地，当 N 是素数时，
   $\varphi(N)=N-1$。
   另外，由于 $\gcd(A,N)=1$，构造方程 $Ax+Ny=1$，用扩展欧几里得算法也可求出 A 在模 N 意义下的逆元 $A^{-1}=x$。
2. 如果 $\gcd(A,N)\neq 1$，由前文可知，方程 $Ax+Ny=1\neq k\cdot\gcd(A,N)$ 不存在整数解，故不存在逆元。

程序实现

1. 欧拉定理求逆元，
   当 N 是素数时，$\phi(N)=N-1$，A 的逆元为 A^{N-2}

```
typedef long long LL;
LL ModPower(LL A, LL x, LL mod) { // 返回 A^x%mod
    LL ans = 1;
    for(; x > 0; x >>=1, A=A*A%mod)
        if( x&1 ) ans = ans*A%mod;
    return ans;
}

LL INV(LL A, LL mod) {
    return ModPower(A, mod-2, mod);
}
```

1. 扩展欧几里得求逆元

```
typedef long long LL;
LL INV(LL A, LL mod) {
    LL x, y, d;
    Extend_Euclid(A, x, N, y, d);
    if( d != 1 ) return -1; // 若 A 和 mod 不互质，则不存在逆元
    return x;
}
```

中国剩余定理

问题描述

若正整数 $m_1,m_2,\cdots,m_k$ 两两互素，$0\leqslant a_1<m_1,0\leqslant a_2<m_2,\cdots,0\leqslant a_k<m_k$。
求整数 x 满足同余方程组：

$$\begin{aligned} x &\equiv a_1 (\mathrm{mod}\ \ m_1) \\ x &\equiv a_2 (\mathrm{mod}\ \ m_2) \\ &\cdots \\ x &\equiv a_k (\mathrm{mod}\ \ m_k) \end{aligned} \tag{2}$$

问题简析

令 $w_i = \prod_{j \neq i} m_j$，显然：

$$\gcd(w_i, m_j) = \begin{cases} 1, & i = j \\ m_j, & i \neq j \end{cases} \tag{3}$$

构造方程 $w_i x_i + m_i y_i = \gcd(w_i, m_i) = 1$。
那么，由方程组 (3) 有，

$$w_i x_i \ (\mathrm{mod}\ \ m_j) \equiv \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{4}$$

结合方程组 (2) 和方程组 (4)，进而有，

$$a_i w_i x_i \ (\mathrm{mod}\ \ m_j) \equiv \begin{cases} a_i, & i = j \\ 0, & i \neq j \end{cases} \tag{5}$$

所以，$x = \sum_{i=1}^{k} (a_i \cdot w_i \cdot x_i)$ 为同余方程组 (2) 的解。

更一般地，方程的解为 $x = \sum_{i=1}^{k} (a_i \cdot w_i \cdot x_i) + k \prod_{i=1}^{k} m_i$。

剩下的问题仅需利用扩展欧几里得算法求出 x_i。

程序实现

```
LL CRT(int N, int* a, int* m) {
    LL M = 1, ans = 0;
    for(int i=0; i < N; ++i) M *= m[i];
    for(int i=0; i < N; ++i) {
        LL w = M/m[i], x, y, d;
        Extend_Euclid(w, x, m[i], y, d);
        ans = (ans+x*w*a[i]) % M;
    }
    return ans;
}
```

Baby Step Gaint Step

问题描述

求一个整数 $x \in [0, C-1]$ 使得 $A^x \equiv B (\mathrm{mod}\ \ C)$；其中，$C$ 为素数。

问题简析

注意到 C 是一个素数，那么对于任意 X 都存在一个模 C 意义下的逆元 X^{-1}，使得
$X \cdot X^{-1} \equiv 1 (\mathrm{mod}\ C)$。
对于 $0 \leqslant x_0 < m$，如果事先知道了所有 $A^{x_0} (\mathrm{mod}\ C)$ 的值，
如何判断是否存在一个 x' 满足 $km \leqslant x' = x_0 + km < (k+1)m$，使得 $A^{x'} \equiv B (\mathrm{mod}\ C)$ 成立呢?

注意到，令方程两边同乘以 A^{km} 的逆元 A^{-km}，得到 $A^{x_0} \equiv B \cdot A^{-km} (\mathrm{mod}\ C)$。
所以，我们仅需将 $f(x_0) = A^{x_0} (\mathrm{mod}\ C)$ 映射到哈希表或平衡树中，再判断哈希表或平衡树中，
是否存在 $B \cdot A^{-km} (\mathrm{mod}\ C)$，并且在存在时返回 x_0 即可。

m 等于多少呢?

注意到这样做的时间复杂度（使用哈希表，查询均摊 $O(1)$），是 $O\left(m + \frac{C}{m}\right)$ 的，

不难证明，当 m 取 $\sqrt{C}$ 时复杂度最优。

程序实现

```cpp
typedef long long LL;
unordered_map<LL, int> emp;cnt+

int EBSGS(LL A, LL B, LL C) {
    emp.clear();
    LL M = ceil(sqrt(1.0*C));
    LL AA = 1;
    for(int i=0; i < M; ++i) {
        if( !emp.count(AA) ) emp[AA] = i;
        AA = AA*A % C;
    }
    LL Am = INV(ModPower(A, M, C), C);
    for(int i=0; i < M; ++i) {
        if( emp.count(B) ) return emp[B]+i*M;
        B = B*Am % C;
    }
    return -1;
}
```

Extend Baby Step Gaint Step

问题描述

求一个整数 $x \in [0, C-1]$ 使得 $A^x \equiv B (\mathrm{mod}\ C)$；其中，$C$ 为合数。

问题简析

【定理1】　若 $k \cdot A \equiv k \cdot B (\mathrm{mod}\ k \cdot C)$，则有，$A \equiv B (\mathrm{mod}\ C)$。
这个定理不难证明，事实上，仅需构造方程 $k \cdot A - k \cdot B = k' \cdot k \cdot C$，再等式两边同除以 k 即可得证。

再回过头来看普通版的 Baby Step Gaint Step 是如何工作的。
注意到，Gaint Step 能够成功走出去的前提是 A^m 存在模 C 意义下的逆元 A^{-m}。
由前文可知，A^{-m} 存在的充要条件为 $\gcd(A^m, C) = \gcd(A, C) = 1$，也就是要满足 A 和 C 互质即可。

设 $A_g = \gcd(A^\infty, C)$，并且 $A_g \mid A^t$；再令 $C' = \frac{C}{A_g}$，$D = \frac{A^t}{A_g}$。
所以，　$A^x = A_g \times \frac{A^t}{A_g} \times A^{x-t} = A_g \times D \times A^{x-t}$，　$C = A_g \times \frac{C}{A_g} = A_g \times C'$。

不难发现，D 和 C' 是互质的，A^{x-t} 和 C' 也是互质的。

1. 若 $A_g \mid B$，结合【定理1】，有 $A^x \equiv B(\bmod\ C) \ \Rightarrow\ D \cdot A^{x-t} \equiv \frac{B}{A_g} (\bmod\ C')$。
   又因为 D 和 C' 是互质的，所以存在模 C' 意义下的逆元 D^{-1}；进一步将同余方程化为 $A^{x-t} \equiv \left(D^{-1} \cdot \frac{B}{A_g}\right)(\bmod\ C')$。
   由于 A^{x-t} 和 C' 是互质的，根据前面的分析，可知直接套用普通版的 Baby Step Gaint Step 即可解决。
2. 若 $A_g \nmid B$，由于 $A^x(\bmod\ C) = K \cdot A_g$，所以无解。

程序实现

```cpp
typedef long long LL;
unordered_map<LL, int> emp;

int EBSGS(LL A, LL B, LL C) {
    LL G, D = 1, cnt = 0;
    while( (G=GCD(A, C)) != 1 ) {
        if( B % G ) return -1;
        B /= G; C /= G;
        D = D*A/G % C;
        ++cnt;
        if( D == B ) return cnt;
    }

    emp.clear();
    LL M = ceil(sqrt(1.0*C));
    LL AA = 1;
    for(int i=0; i < M; ++i) {
        if( !emp.count(AA) ) emp[AA] = i;
        AA = AA*A % C;
    }
    B = B*INV(D, C) % C;
    LL Am = INV(ModPower(A, M, C), C);
    for(int i=0; i < M; ++i) {
        if( emp.count(B) ) return emp[B]+cnt+i*M;
        B = B*Am % C;
    }
    return -1;
}
```

练习

题目	分类	程序实现
SPJ/MOD	扩展 BSGS	*Code*

Hint

拖沓了很久，终于耐下性子整理了一下。
鉴于本人能力有限，有误之处欢迎指正。

参考链接

- ACM 1
  - 模方程基础
- Baby-Step-Gaint-Step 1
  - 模方程基础
- 学习笔记 1
  - 模方程基础
- 数论 1