扩展欧几里得算法

Extend- Euclid Algorithm

关键字：不定方程、模线性方程、逆元

重庆南开信竞课程

1.什么是扩展欧几里得算法

欧几里得算法

欧几里德算法又称辗转相除法，用于计算两个整数a，b的最大公约数。

gcd(a , b) == gcd(a % b , b)

```
int  euclid(int a,int b)
{
      if (b==0)return a;
      else  return euclid(b,a%b);
}
```

欧几里得算法 原理

- gcd(a,b) = gcd(b,a % b)
- 证明:
- 若r = a % b，则r可以表示成r = a - k * b, (k=a/b)
- 假设d是a,b的一个公约数，

  则有a % d = 0 , b % d = 0,

  设a = x * d , b = y * d,   x,y为整数

  而r = a - k * b = x * d - k * y * d = (x - k * y) * d

  因为x-k*y是整数，所以r是d的倍数，r%d=0

  因此d是a%b的约数

根据定义d是b的约数，因此d是(b,a % b)的公约数

欧几里得算法 原理

- 依据以上原理：
- 经过一步代换必有 a > b
- 以后的每次代换 将 a 换为 b ,将 b 换为 a % b ,这样 a,b 必定再减小。
- 当 b 减小到 0 时，它们的最大公因数为a

欧几里得算法 代码实现

```
int  euclid(int a,int b)
{
     if (b==0)return a;
     else  return euclid(b,a%b);
}
```

扩展欧几里得算法

扩展欧几里德算法是在已知整数a, b情况下求**a*x+b*y = Gcd(a,b)** 的一组**整数解**x和y

对于整数a，b，必定存在整数对x，y满足a*x+b*y=gcd(a, b)

证明：

设 a*x1+b*y1=gcd(a,b)；　设 b*x2+(a%b)*y2=gcd(b,a%b)；

由欧几里德原理知： gcd(a,b)=gcd(b,a%b)　所以==> a*x1+b*y1=b*x2+(a%b)*y2

因为r=a%b,　设r=a-k*b 所以==> a*x1+b*y1=b*x2+(a-k*b)*y2

因为k=a/b;　所以==> a*x1+b*y1=b*x2+(a-(a/b)*b)*y2

展开得到==> a*x1+b*y1=b*x2+a*y2-b*(a/b)*y2

转换得到==> a*x1+b*y1=a*y2+b*(x2-(a/b)*y2)

观察上式可知,对于(a,b)的不定方程可以转换为(b,a%b)的不定方程**x1=y2，　y1=x2-a/b*y2**

由此可知x1，y1可由x2，y2得出来的，由此类推x2，y2可是由x3，y3得出来的，
那什么时候是终止呢？也就是递归gcd(a,b)中b=0时即gcd(a,0)此时由a*x+b*y==gcd(a,b)
可知a*x+b*y=a　解出x=1, y=0; 此时就是递归终止的地方。
然后根据x1=y2,y1=x2-a/b*y2回溯回去就可求助x和y最初的解了。
所以：a*x+b*y=gcd(a,b)一定有解

扩展欧几里得算法

$a*x+b*y=gcd(a,b)$

$b*x'+(a\%b)*y'=gcd(b,a\%b)$

$x=y'$ ， $y=x'-a/b*y'$

$50x_1+20y_1=gcd(50,20)=gcd(20,10)$ $x_1=y_2=1$ $y1=x_2-a/b*y_2=-2$

$20x_2+10y_2=gcd(20,10)=gcd(10,0)$ $x_2=y_3=0$ $y_2=x_3-a/b*y_3=1$

$10x_3+0y_3=gcd(10,0)=10$ $x_3=1$ $y_3=0$

扩展欧几里得算法

求解一组整数x,y,使得a*x+b*y = Gcd(a,b)

根据我们前边的结论（*对照右边的代码*）：
随着欧几里得算法的进行，
　　a，b 都在减小*(每次运算，a变成b，b变成a % b)*
　　当 b 减小到 0 时,此时a的值就是所求的最大公约数
　　我们就可以得出 x' = 1 ,y' = 0
然后根据x=y', y=x'-a/b*y'**回溯**回去就可以求出最初的 x ,y 了。

```
int euclid(int a,int b)
{
	if (b==0)return a;
	else  return euclid(b,a%b);
}
```

扩展欧几里得算法 代码实现1：

```
//求解a*x1+b*y1=gcd(a,b)
int extended_gcd(int a, int b, int &x1, int &y1)
{
    int d, x2, y2;
    if (b == 0) {  x1 = 1;  y1 = 0;  return a; }
    d = extended_gcd(b, a % b, x2, y2); //求解b*x2+(a%b)*y2=gcd(b,a%b)
    x1 = y2;   y1 = x2 - a / b * y2;
    return d;
}
```

```
int Gcd(int a,int b)
{
    if (b==0)return a;
    else  return Gcd(b,a%b);
}
```

扩展欧几里得算法 代码实现2：

```
int extended_gcd(int a, int b ,int &x,int &y)
{
    int d, tmp;
    if (b==0) {  x = 1;  y = 0;  return a; }
    d = extended_gcd(b, a % b,x,y);
    tmp = x;  x = y;  y = tmp - a / b * y;
    return d;
}
```

扩展欧几里得算法 代码实现3：

```
int main()
{
    int a,b,x,y,z;
    scanf("%d%d",&a,&b);
    z=extended_gcd(a,b,x,y);
    printf("%d %d %d\n",z,x,y);
}
```

2.扩欧的应用 解不定方程

扩欧应用1 不定方程 无解判定

不定方程ax+by=c 有解？无解？   这里讨论整数解

设d为一整数，**若a%d==0且b%d==0，则(ax+by)%d==0**
由这条定理易推知：
因为ax+by=gcd(a,b) 一定有解，
所以，若a%d==0且b%d==0，则gcd(a,b)%d==0

对于不定方程ax+by=c，设gcd(a,b)=d，
根据上述的结论：若a%d==0且b%d==0，则(ax+by)%d==0
所以，如果ax+by=c有解，则c%d==0。

所以如果 c%d != 0，那么 ax+by=c 一定无解。

扩欧应用1 不定方程 通解

不定方程ax+by=c 通解？

对于不定方程ax+by=c，设gcd(a,b)=d，
当c%d==0时(有解)，
先用扩欧求出ax′+by′= d =gcd(a,b)的一组解x'和y'，
则方程原来的一组解为x=x'*c/d，y=y'*c/d

原方程的解有无限多组，因为满足下面式子：
a*(x+k*b)+b*(y-k*a)=c ，k为整数
于是我们得到ax+by=c的通解：
x=x'*c/d+k*b/d
y=y'*c/d-k*a/d

3.扩欧的应用 解模线性方程

扩展欧几里得算法 解题应用

NKOJ 1886
求关于 x 的同余方程 ax ≡ 1 (mod b)的最小正整数解。

ax ≡ 1 (mod b) 表示 a*x%b == 1%b 即(a*x)%b==1

(a*x)%b==1
即 ax-by==1 y为整数
用扩欧处理即可！

根据前面的定理，ax-by=gcd(a,b)
又1必须能整除1/gcd(a,b)，所以gcd(a,b)一定=1

扩展欧几里得算法 解题应用

```
int extended_gcd(int a,int b,int &x,int &y)
{
       int d,temp;
       if (b==0) {  x=1;  y=0;  return a;   }
       d=extended_gcd(b,a%b,x,y);
       temp=x;  x=y;  y=temp-a/b*y;
       return d;
}
int main()
{
      scanf("%d%d",&a,&b);
      extended_gcd(a,b,x,y);
      if(x>0)x%=b;
      if(x<0)x=(x%b)+b;
      printf("%d",x);
}
```

扩欧应用2 解模线性方程

$$ax \equiv b \pmod{n}$$

$a \equiv b \pmod{n}$的含义是a和b关于模n同余，即**a mod n==b mod n**

$a \equiv b \pmod{n}$的充要条件是a-b是n的整数倍

这样，$ax \equiv b \pmod{n}$可以解释成：ax-b是n的整数倍，设这个倍数为y，则ax-b=ny，移项得ax-ny=b，这恰好是一个二元一次不定方程。

用扩欧就可以解决。

4.扩欧的应用 求乘法逆元

扩欧应用3 乘法逆元

$ax \equiv 1 \pmod{n}$的解x称为a关于模n的乘法逆元

在同余方程的意义下，a的逆元常记为a^{-1},注意，它不是传统的指数概念，只是表示：$a*a^{-1} \equiv 1 \pmod{n}$， a^{-1}可理解为在模n意义下a的“倒数”。

什么情况下a的逆元存在呢？

由前面的讨论可知：不定方程ax-ny=1要有解。

这样1必须是gcd(a,n)的倍数，因此a和n必须互质，即gcd(a,n)=1

在gcd(a,n)=1的前提下，$ax \equiv 1 \pmod{n}$只有唯一解，

此解即为a在模n下的乘法逆元a^{-1}

注意：通过扩欧算出的不定方程有很多解，最终的逆元应该是x%n,也就是逆元的范围是[0,n-1]

扩欧应用3 乘法逆元

求ax ≡ 1 (mod n)的乘法逆元，若不存在，返回-1

```
long long Inverse(long long a,long long n)
{
        long long x,y;
        if(extended_gcd(a,n,x,y)==1)return (x+n)%n;
        else return -1;
}
```

在执行完extended_gcd()后，x可能为负数，加上n后将其变为整数。

求乘法逆元还可以用“欧拉定理”

扩欧应用3 乘法逆元

取模运算对于加、减、乘有分配率，但对除没有。
乘法逆元可在同余式中把除法改为乘法。

```
(a*b) mod p==((a mod p)*(b mod p)) mod p

(a/b) mod p != ((a mod p)/(b mod p)) mod p
```

但是我们可以把 (a/b) mod p 改写成 **$(a*b^{-1})$ mod p**

(a/b) mod p == $(a*b^{-1})$ mod p

$== ((a \bmod p)*(b^{-1} \bmod p)) \bmod p$

扩欧应用3 乘法逆元

求证：(a/b)mod p == (a*b^{-1}) mod p

根据b*x≡1 (mod p)有：b*x-p*y=1

b的逆元x=(p*y+1)/b。

把x代入(a*x) mod p，得：(a*(p*x+1)/b) mod p

=((a*p*x)/b+a/b) mod p

=(((a*p*x)/b) mod p +(a/b)) mod p

=((p*(a*x)/b) mod p +(a/b)) mod p

因为p*((a*x)/b) mod p=0

所以原式等于：(a/b) mod p，得证！

扩展欧几里得算法 课后习题

作业：NKOJ 1886,2733,3676,3675,3681
POJ2115

奋斗吧
少年

巨大的成功需要付出巨大的代价

no sacrifice, no success