二次剩余入门

2018年01月02日 17:40:42 Eiffel,,,, 阅读数：393 标签： 二次域 二次剩余 二次同余方程 更多

昨天训练的时候遇到一道题怎么也不会做，在网上搜了题解之后第一次听说二次剩余，看了一天各种dalao的博客，在这里总结一下自己所理解的二次剩余及其用法。

1，什么是二次剩余?

从定义上来解释，现在给出一个式子 $x^2 \equiv n(\mathrm{mod}\ p)$，再给出 n 和 p，如果能求得一个 x 满足该式子，即 x 满足 $x^2 = n + kp, k \in Z$，那么我们称 **n 是模 p 的二次剩余**。若不存在这样的 x，那么我们称 **n 是模 p 的非二次剩余**。同时我们称 **x 为该二次同余方程的解**。

2，二次剩余有什么用?

对于一个数 n，如果我们要求 $\sqrt{n}(\mathrm{mod}\ p)$ 的值，那么我们可以看 n 是否是模 p 的二次剩余，如果是的话就会满足 $x^2 \equiv n(\mathrm{mod}\ p) \rightarrow x \equiv \sqrt{n}(\mathrm{mod}\ p)$，那么我们就可以用 x 来代替 $\sqrt{n}$，即只需要求该二次同余方程的解即可。

※说白了就是如果该二次同余方程有解，那么n可以在模p的意义下开根号。

3，二次同余方程如何求解?

这一部分内容参考了这两篇文章：

http://blog.csdn.net/a_crazy_czy/article/details/51959546

http://blog.csdn.net/acdreamers/article/details/10182281

首先，以下解法有一个前提，就是p必须要是奇素数。

定理①：对于二次同余方程$x^2 \equiv n(\bmod p)$，总共有$\frac{p-1}{2}+1$个 n 的值可以让方程有解，其中"+1"所代表的就是 n=0 的情况，此时 x=0 方程显然成立。

证明：

假设$u^2 = ap + k$即$u^2 \equiv k(\bmod p)$

$\because \forall n \in Z, (u+np)^2 = u^2 + n^2p^2 + 2unp = (a + n^2p + 2un)p + k$

$\therefore u^2 \equiv (u+np)^2(\bmod p)$

那么由此可知我们只需要考虑$[0, p-1]$范围内的x^2即可。

若存在两个不同的数$u, v \in [0, p-1]$，满足$u^2 \equiv v^2(\bmod p)$，那么显然有$u^2 - v^2 \equiv 0(\bmod p)$，根据平方差公式可知$(u+v)(u-v) \equiv 0(\bmod p)$，由$u, v$的范围和$p$是奇素数可知，$(u-v)$不可能是$p$的倍数，则必满足$(u+v) \equiv 0(\bmod p)$即$u+v=p$。所以共有$\frac{p-1}{2}$组$u, v$满足答案。

再算上 n=0 的情况，所以答案是$\frac{p-1}{2}+1$。

接下来我们引入一个新概念：勒让德符号(legender symbol)

它的定义如下

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & a\text{在模}p\text{意义下是二次剩余} \\ -1, & a\text{在模}p\text{意义下是非二次剩余} \\ 0, & a \equiv 0 \pmod p \end{cases}$$

由此再引出一个定理

定理②：$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} (\bmod p)$

证明：

当 a 是模 p 的二次剩余时，令$x^2 \equiv a(\bmod p)$，那么就有$x^{p-1} \equiv 1(\bmod p)$，由费马小定理可知 x 存在。

当 a 是模 p 的非二次剩余时，令$x^2 \equiv a(\bmod p)$，那么就有$x^{p-1} \equiv -1(\bmod p)$，由费马小定理可知 x 不存在。

当$a \equiv 0(\bmod p)$时显然满足。

接下来是最后一个定理

定理③：设 a 满足 $\omega=a^2-n$ 是模 p 的非二次剩余，即 $x^2\equiv\omega(\mathrm{mod}\ p)$ 无解，那么 $x\equiv(a+\sqrt{\omega})^{\frac{p+1}{2}}$ 是二次同余方程 $x^2\equiv n(\mathrm{mod}\ p)$ 的解。

证明：

给出一个式子 $(a+\sqrt{\omega})^p$，将其二项式展开得 $\sum_{r=0}^{p}C_p^r a^{p-r}\sqrt{\omega}^r$。我们知道，当 p 为奇素数时，若 $i\neq 0, i\neq p$，则有 $C_p^i\equiv 0(\mathrm{mod}\ p)$（因为 p 永远是因子，不会被除掉）。所以该二项式除了第一项以及最后一项，其它项模 p 都为 0。那么就有 $(a+\sqrt{\omega})^p\equiv a^p+\sqrt{\omega}^p(\mathrm{mod}\ p)$ 成立。由费马小定理可知 $a^p\equiv a^{p-1}a\equiv a(\mathrm{mod}\ p)$。因为 $\sqrt{\omega}^p\equiv\omega^{\frac{p-1}{2}}\sqrt{\omega}\ (\mathrm{mod}\ p)$，根据我们之前给出的定理②可知，$\omega^{\frac{p-1}{2}}=-1$。那么就有 $a^p+\sqrt{\omega}^p\equiv a-\sqrt{\omega}(\mathrm{mod}\ p)$。所以综上所述 $(a+\sqrt{\omega})^p\equiv a-\sqrt{\omega}(\mathrm{mod}\ p)$。然后就有：

$$\begin{aligned}(a+\sqrt{\omega})^{p+1}&\equiv(a+\sqrt{\omega})^p(a+\sqrt{\omega})\\&\equiv(a-\sqrt{\omega})(a+\sqrt{\omega})\\&\equiv a^2-\omega\\&\equiv n(mod\ p)\end{aligned}$$

有了最后一个定理，我们就可以通过随机选择a的值来找到一个满足条件的解。之前的链接里有详细地解释为何可以随机取a的值，总的来说就是找到正解所需的次数的期望只有2。所以随机取a的值可以很快地找到一个解，代码如下。

```
#include <iostream>
#include <ctime>
using namespace std;
typedef long long LL;
#define random(a,b) (rand()%(b-a+1)+a)
LL quick_mod(LL a, LL b, LL c) { LL ans = 1; while (b) { if (b % 2 == 1)ans = (ans*a) % c; b /= 2; a = (a*a) % c; }return ans; }

LL p;
LL w;//二次域的D值
bool ok;//是否有解

struct QuadraticField//二次域
{
	LL x, y;
	QuadraticField operator*(QuadraticField T)//二次域乘法重载
	{
		QuadraticField ans;
		ans.x = (this->x*T.x%p + this->y*T.y%p*w%p) % p;
		ans.y = (this->x*T.y%p + this->y*T.x%p) % p;
		return ans;
	}
	QuadraticField operator^(LL b)//二次域快速幂
	{
		QuadraticField ans;
		QuadraticField a = *this;
		ans.x = 1;
		ans.y = 0;
		while (b)
```

```
                                ans = ans*a;
                                b--;
                        }
                        b /= 2;
                        a = a*a;
                }
                return ans;
        }
};

LL Legender(LL a)//求勒让德符号
{
        LL ans=quick_mod(a, (p - 1) / 2, p);
        if (ans + 1 == p)//如果ans的值为-1，%p之后会变成p-1。
                return -1;
        else
                return ans;
}

LL Getw(LL n, LL a)//根据随机出来a的值确定对应w的值
{
        return ((a*a - n) % p + p) % p;//防爆处理
}

LL Solve(LL n)
{
        LL a;
        if (p == 2)//当p为2的时候，n只会是0或1，然后0和1就是对应的解
                return n;
        if (Legender(n) == -1)//无解
                ok = false;
        srand((unsigned)time(NULL));
        while (1)//随机a的值直到有解
        {
                a = random(0, p - 1);
                w = Getw(n, a);
                if (Legender(w) == -1)
                        break;
        }
        QuadraticField ans,res;
        res.x = a;
        res.y = 1;//res的值就是a+根号w
        ans = res ^ ((p + 1) / 2);
        return ans.x;
}

int main()
{
        LL n,ans1,ans2;
        while (scanf("%lld%lld",&n,&p)!=EOF)
        {
                ok = true;
                n %= p;
                ans1 = Solve(n);
                ans2 = p - ans1;//一组解的和是p
                if (!ok)
                {
                        printf("No root\n");
                        continue;
                }
                if (ans1 == ans2)
                        printf("%lld\n", ans1);
                else
                        printf("%lld %lld\n", ans1, ans2);
        }
}
```