扩展大步小步法解决离散对数问题

May 16, 2015 (http://blog.miskcoo.com/2015/05/discrete-logarithm-problem) miskcoo (http://blog.miskcoo.com/author/miskcoo) Algorithm (http://blog.miskcoo.com/category/algorithm), Math (http://blog.miskcoo.com/category/math) 68 views Edit (http://blog.miskcoo.com/wp-admin/post.php?post=848&action=edit)

离散对数（Discrete Logarithm）问题是这样一个问题，它是要求解模方程

$$a^x \equiv b \pmod{m}$$

这个问题是否存在多项式算法目前还是未知的，这篇文章先从 m 是质数开始介绍大步小步法（Baby Step Giant Step）来解决它，之后再将其应用到 m 是任意数的情况。这个算法可以在 $\mathcal{O}(\sqrt{m})$ 的时间内计算出最小的 x，或者说明不存在这样一个 x

题目链接：BZOJ-2480 (http://www.lydsy.com/JudgeOnline/problem.php?id=2480)、SPOJ-MOD (http://www.spoj.com/problems/MOD)、BZOJ-3239 (http://www.lydsy.com/JudgeOnline/problem.php?id=3239)

首先解决 $m = p$ 是质数的情况，我们可以设 $x = A\lceil\sqrt{p}\rceil + B$，其中 $0 \le B < \lceil\sqrt{p}\rceil$, $0 \le A < \lceil\sqrt{p}\rceil$，这样的话就变成求解 A 和 B 了，方程也变成

$$a^{A\lceil\sqrt{p}\rceil+B} \equiv b \pmod{p}$$

可以在两边同时乘以 a^B 的逆元，由于 p 是质数，这个逆元一定存在，于是方程变成

$$a^{A\lceil\sqrt{p}\rceil} \equiv b \cdot a^{-B} \pmod{p}$$

由于 A, B 都是 $\mathcal{O}(\sqrt{p})$ 级别的数，可以先计算出右边这部分的值，存入 Hash 表，然后计算左边的值，在 Hash 表中查找，只要按照从小到大的顺序如果有解就能够找到最小的解，由于两边都只有 $\mathcal{O}(\sqrt{p})$ 个数，因此时间复杂度是 $\mathcal{O}(\sqrt{p})$ 的，这样 m 是质数的情况就解决了

一个优化：我们可以设 $x = A\lceil\sqrt{p}\rceil - B$，其中 $0 \le B < \lceil\sqrt{p}\rceil$, $0 < A \le \lceil\sqrt{p}\rceil + 1$，这样的话化简后的方程就是

$$a^{A\lceil\sqrt{p}\rceil} \equiv b \cdot a^{B} \pmod{p}$$

就可以不用求出逆元，要注意只是不用求出逆元，而不是没有用到逆元的存在

现在来看 m 不是质数的情况，同样可以设 $x = A\lceil\sqrt{m}\rceil + B$，根据上面的推导，会发现需要用到的性质就是 a^B 的逆元存在，所以当 m 和 a 互质的时候这个方法仍然有效！

如果 $(m, a) \neq 1$ 该怎么办呢？我们要想办法把方程转化为 $(m, a) = 1$ 的情况

把要求的模方程写成另外一种形式

$$a^x + km = b, k \in \mathbb{Z}$$

设 $g = (a, m)$，这样的话可以确定如果 $g \nmid b$ 那么该方程一定无解，所以当 $g \mid b$ 的时候，在方程左右两边同时除以 g

$$\frac{a}{g}a^{x-1}+k\frac{m}{g}=\frac{b}{g},k\in\mathbb{Z}$$

这样便消去了一个因子，得到方程

$$\frac{a}{g}a^{x-1}\equiv\frac{b}{g}\pmod{\frac{m}{g}}$$

令 $m'=\frac{m}{g},b'=\frac{b}{g}\left(\frac{a}{g}\right)^{-1}$（这里不可以把 g 消掉），就可以得到新的方程

$$a^{x'}\equiv b'\pmod{m'}$$

得到解之后原方程的解 $x=x'+1$，不断重复这个过程最后一定会得到一个可以解的方程，套用刚刚的大步小步法解出后即可。要注意的是在这个过程中如果某一步发现 $b'=1$，那么就可以直接退出，因为这时候已经得到了解

NOTE：上面这个过程是可能执行多次的比如说
$(a,m)=(6,16)\to(6,8)\to(6,4)\to(6,2)$

下面的是代码，题目是文章开头给的链接

```cpp
/* BZOJ-2480: Spoj3105 Mod
 *   扩展大步小步  */
#include <cstdio>
#include <cmath>
#include <map>

int mod_pow(long long x, long long p, long long mod_v)
{
    long long v = 1;
    while(p)
    {
        if(p & 1) v = x * v % mod_v;
        x = x * x % mod_v;
        p >>= 1;
    }

    return v;
}

int gcd(int a, int b)
{
    return b ? gcd(b, a % b) : a;
}

int baby_step_giant_step(int a, int b, int p)
{
    a %= p, b %= p;
    if(b == 1) return 0;
    int cnt = 0;
    long long t = 1;
    for(int g = gcd(a, p); g != 1; g = gcd(a, p))
    {
        if(b % g) return -1;
        p /= g, b /= g, t = t * a / g % p;
        ++cnt;
        if(b == t) return cnt;
    }

    std::map<int, int> hash;
    int m = int(sqrt(1.0 * p) + 1);
    long long base = b;
    for(int i = 0; i != m; ++i)
    {
        hash[base] = i;
        base = base * a % p;
    }

    base = mod_pow(a, m, p);
    long long now = t;
    for(int i = 1; i <= m + 1; ++i)
    {
        now = now * base % p;
        if(hash.count(now))
            return i * m - hash[now] + cnt;
    }

    return -1;
}

int main()
{
    int a, b, p;
    while(std::scanf("%d %d %d", &a, &p, &b), p)
    {
        int ans = baby_step_giant_step(a, b, p);
        if(ans == -1) std::puts("No Solution");
        else std::printf("%d\n", ans);
    }
    return 0;
}
```

Related Posts:

1. **[数论]线性求所有逆元的方法 (3) (3.000000 is the YARPP match score between the current entry and this related entry. You are seeing this value because you are logged in to WordPress as an administrator. It is not shown to regular visitors.) (http://blog.miskcoo.com/2014/09/linear-find-all-invert)**
2. **NOI2012. 迷失游乐园 (3) (3.000000 is the YARPP match score between the current entry and this related entry. You are seeing this value because you are logged in to WordPress as an administrator. It is not shown to regular visitors.) (http://blog.miskcoo.com/2014/10/bzoj-2878)**
3. **BZOJ-3509. [CodeChef] COUNTARI (3) (3.000000 is the YARPP match score between the current entry and this related entry. You are seeing this value because you are logged in to WordPress as an administrator. It is not shown to regular visitors.) (http://blog.miskcoo.com/2015/04/bzoj-3509)**
4. **BZOJ-4001. [TJOI2015]概率论 (3) (3.000000 is the YARPP match score between the current entry and this related entry. You are seeing this value because you are logged in to WordPress as an administrator. It is not shown to regular visitors.) (http://blog.miskcoo.com/2015/04/bzoj-4001)**
5. **BZOJ-3557. [Ctsc2014]随机数 (3) (3.000000 is the YARPP match score between the current entry and this related entry. You are seeing this value because you are logged in to WordPress as an administrator. It is not shown to regular visitors.) (http://blog.miskcoo.com/2015/05/bzoj-3557)**

bzoj (http://blog.miskcoo.com/tag/tag-bzoj)　数论 (http://blog.miskcoo.com/tag/tag-number-theory)

离散对数 (http://blog.miskcoo.com/tag/%e7%a6%bb%e6%95%a3%e5%af%b9%e6%95%b0)

❮ BZOJ-3812. 主旋律 (http://blog.miskcoo.com/2015/05/bzoj-3812)

BZOJ-3557. [Ctsc2014]随机数 ❯ (http://blog.miskcoo.com/2015/05/bzoj-3557)

MISKCOO (HTTP://BLOG.MISKCOO.COM/AUTHOR/MISKCOO)

某只高二的 Oier，然后顺便卖个萌 > _ < !

LEAVE A REPLY

Logged in as miskcoo (http://blog.miskcoo.com/wp-admin/profile.php). Log out? (http://blog.miskcoo.com/wp-login.php?action=logout&redirect_to=http%3A%2F%2Fblog.miskcoo.com%2F2015%2F05%2Fdiscrete-logarithm-problem&_wpnonce=f23535e01b)

Comment

You may use these HTML (HyperText Markup Language) tags and attributes: `<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code class="" title="" data-url=""> <del datetime=""> <em> <i> <q cite=""> <s> <strike> <strong> <pre class="" title="" data-url=""> <span class="" title="" data-url="">`

POST COMMENT

Search...

RECENT POSTS

BZOJ-2001. [HNOI2010]City城市建设 (http://blog.miskcoo.com/2015/04/bzoj-2001)

区间K小问题 (http://blog.miskcoo.com/2015/04/kth-problem)

CATEGORIES

- Algorithm (http://blog.miskcoo.com/category/algorithm) (26)
- Linux (http://blog.miskcoo.com/category/linux) (5)
- Math (http://blog.miskcoo.com/category/math) (23)
- Programming (http://blog.miskcoo.com/category/programming) (18)
  - C++ (http://blog.miskcoo.com/category/programming/cxx) (4)
  - Web (http://blog.miskcoo.com/category/programming/web) (1)
- Uncategorized (http://blog.miskcoo.com/category/uncategorized) (2)

RECENT COMMENTS

多项式的多点求值与快速插值 | Miskcoo's Space (http://blog.miskcoo.com/2015/05/polynomial-multipoint-eval-and-interpolation) on 多项式除法及求模 (http://blog.miskcoo.com/2015/05/polynomial-division#comment-2528)

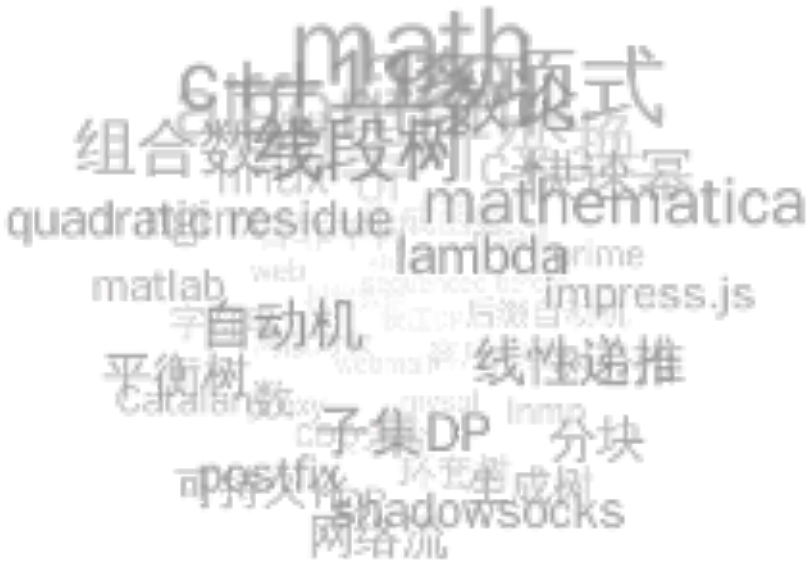
多项式除法及求模 | Miskcoo's Space (http://blog.miskcoo.com/2015/05/polynomial-division) on 多项式求逆元 (http://blog.miskcoo.com/2015/05/polynomial-inverse#comment-2291)

qscqesze on BZOJ-4001. [TJOI2015]概率论 (http://blog.miskcoo.com/2015/04/bzoj-4001#comment-2070)

SCaffrey (http://imcaffrey.github.io) on BZOJ-4002. [JLOI2015]有意义的字符串 (http://blog.miskcoo.com/2015/05/bzoj-4002#comment-1790)

miskcoo (http://www.miskcoo.com) on BZOJ-4001. [TJOI2015]概率论 (http://blog.miskcoo.com/2015/04/bzoj-4001#comment-1717)

TAGS



May 2015

M	T	W	T	F
				1 (http://blog /2015/05/
4	5	6	7	8
11	12	13 (http://blog.miskcoo.com /2015/05/13)	14	15
18	19	20	21 (http://blog.miskcoo.com /2015/05/21)	22 (http://blog /2015/05/
25 (http://blog.miskcoo.com /2015/05/25)	26	27	28	29

« Apr (http://blog.miskcoo.com/2015/04)

FRIEND LINKS

trinkle (http://trinklee.blog.163.com)

97littleleaf11 (http://97littleleaf11.xyz)

hzwer (http://hzwer.com)

META

Site Admin (http://blog.miskcoo.com/wp-admin/)

Log out (http://blog.miskcoo.com/wp-login.php?action=logout&_wpnonce=f23535e01b)

Entries RSS (Really Simple Syndication) (http://blog.miskcoo.com/feed)

Comments RSS (Really Simple Syndication) (http://blog.miskcoo.com/comments/feed)

WordPress.org (https://wordpress.org/)

(https://github.com/miskcoo/)