

## 【SPOJ】Power Modulo Inverted 扩展BSGS

2019-09-10 20:20:00 我是一只计算鸡 阅读数 11 更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/giftedpanda/article/details/100711144>

$$a^x \equiv b \pmod n$$

当 $\gcd(a, n) \neq 1$ 时，我们就不能用BSGS算法求解。既然不互质时，我们无法求解，那我们就想办法让 $\gcd(a, n) = 1$

$$a * a^{x-1} \equiv b \pmod n$$

$$\frac{a}{g} * a^{x-1} \equiv \frac{b}{g} \pmod{\frac{n}{g}}, g = \gcd(a, n)$$

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

如果 $\gcd(a, \frac{n}{g}) = 1$ ，就成了我们熟悉的BSGS算法了。

$$\text{如果 } \gcd(a, \frac{n}{g}) \neq 1, a * a^{x-2} = b * a^{-1} \pmod{\frac{n}{g}}$$

$$\frac{a}{g'} * a^{x-2} \equiv \frac{b}{g'} * a^{-1} \pmod{\frac{n}{g * g'}}, g' = \gcd(a, \frac{n}{g})$$

如果 $b \% g \neq 1$ ，则方程无解

否者一直除到互质为止

$$\text{假设进行了} t \text{次除法, } a^{x-t} \equiv b * \text{inv}\left(\frac{a}{\prod_{i=1}^t g}\right) \pmod{\frac{n}{\prod_{i=1}^t g}}$$

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  const int MOD = 76543;
4  typedef long long ll;
5  ll hs[MOD], id[MOD], _next[MOD], head[MOD], top;
6  // hs hash表 记录存的值
7  // id 值所对应的id
8  // _next 相同hash值的上一个序号
9  // head hash 值相同的最后一个元素的序号
10 // 其实就是用链式前向星实现的hash表
11 ll gcd(ll a, ll b) //最大公约数
12 {
13     if(b == 0) return a;
14     while(b) {
15         ll t = a;
16         a = b;
17         b = t % b;
18     }
19     return a;
20 }
21 ll exgcd(ll a, ll b, ll &x, ll &y) // 扩展欧几里得
22 {
23     if(b == 0) {
24         x = 1;
25         y = 0;
26         return a;
27     }
28     ll d = exgcd(b, a % b, x, y);
29     ll t = x;
30     x = y;
31     y = t - a / b * y;
32     return d;
33 }
34 ll inv(ll a, ll p) // 求解逆元
35 {

```

1024

程序员节，  
为程序员加油！

关闭

```
36 |         ll x, y; 37 |         exgcd(a, p, x, y);
38 |         return (x % p + p) % p; // 不能直接 return x, 因为 x 可能为负数
39 |     }
40 | void insert(ll x, ll y) // 插入hash表
41 | {
42 |     ll k = x % MOD;
43 |     hs[top] = x;
44 |     id[top] = y;
45 |     _next[top] = head[k];
46 |     head[k] = top++;
47 |     return ;
48 | }
49 | ll find(ll x) // 查找
50 | {
51 |     ll k = x % MOD;
52 |     for(ll i = head[k]; i != -1; i = _next[i]) if(hs[i] == x) return id[i];
53 |     return -1;
54 | }
55 | ll EXBSGS(ll a, ll b, ll n) // 扩展大步小步算法, gcd(a, n) != 1
56 | {
57 |     memset(head, -1, sizeof(head)); // hash表初始化
58 |     top = 1;
59 |     if(b == 1) return 0; // a^0 = 1 mod n
60 |     ll cnt = 0, tmp = 1, d;
61 |     while((d = gcd(a, n)) != 1) { // a, n 不互质
62 |         if(b % d != 0) return -1; // 无解
63 |         b /= d;
64 |         n /= d;
65 |         tmp = tmp * (a / d) % n;
66 |         cnt++;
67 |         if(b == tmp) return cnt; // a^{x - cnt} = 1 mod n
68 |     }
69 |     b = b * inv(tmp, n) % n;
70 |     ll m = sqrt(n * 1.0), j;
71 |     ll x = 1, p = 1;
72 |     for(ll i = 0; i < m; i++, p = p * a % n) insert(b * p % n, i); // 枚举右边 1 ~ sqrt(n)
73 |     for(ll i = m; ; i += m) { // 枚举左边sqrt(n) * {1 ~ sqrt(n)}
74 |         if((j = find(x = x * p % n)) != -1) return i - j + cnt;
75 |         if(i > n) break;
76 |     }
77 |     return -1;
78 | }
79 | int main()
80 | {
81 |     ll a, b, n;
82 |     while(scanf("%lld %lld %lld", &a, &n, &b) == 3 && (a || n || b)) {
83 |         ll ans = EXBSGS(a, b, n);
84 |         if(ans == -1) printf("No Solution\n");
85 |         else printf("%lld\n", ans);
86 |     }
87 |     return 0;
88 | }
```

有 0 个人打赏

文章最后发布于: 2019-09-10 20:26:38

©2019 CSDN 皮肤主题: 终极编程指南 设计师: CSDN官方博客

