BSGS

基础篇

问题:

给出a, b, p, 其中gcd(a, p) = 1, 求x满足

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

思路:

设 $x=A\sqrt{p}-B$ 其中 $A\in(0,\sqrt{p}],B\in[0,\sqrt{p})$,得到问题的变形

$$a^{A\sqrt{p}-B}\equiv b (mod\ p) \ a^{A\sqrt{p}}\equiv b a^B (mod\ p)$$

我们先枚举B,算出每个 $ba^B \ mod \ p$,用 $unordered_map$ 存起来,再枚举A,计算出 $a^{A\sqrt{p}}$,在 $unordered_map$ 中找相同的值,这样的A,B就能恰好凑成一对答案。复杂度 $O(\sqrt{p})$,如果用map的话,就多一个log。

模板:

P2485 [SDOI2011]计算器

[TJOI2007] 可爱的质数/【模板】BSGS

进阶篇

问题:

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

解法一:

设g是p的一个原根,根据原根的性质存在c,满足 $g^c\equiv x$ 成立,同理存在t,满足 $g^t\equiv b$ 成立。

$$\therefore (g^c)^a \equiv b \pmod{p}$$
$$(g^a)^c \equiv b \pmod{p}$$

 g^a 已知,所以我们就能用基础篇直接求解c,也就是一个特解。

解法二:

设g是p的一个原根,根据原根的性质存在c,满足 $g^c \equiv x$ 成立,同理存在t,满足 $g^t \equiv b$ 成立。

$$\therefore g^{ac} \equiv g^t (mod \ p)$$
 根据阶的性质有 $\therefore ac \equiv t (mod \ arphi(p))$ 根据 $exgcd$ 求出 c ,也是一个特解。

求全部的解:

我们在已知一个特解 g^c 的情况下,我们要得到全部解。

$$egin{aligned} arphi g^{arphi(p)} &\equiv 1 (mod \ p) \ dots &orall t \in \mathbb{Z}, x^a \equiv g^{ca + rac{tarphi(p)}{a}} (mod \ p) \ dots &orall t \in \mathbb{Z}$$
且 $a | t arphi(p), x \equiv g^{c + rac{tarphi(p)}{a}} (mod \ p) \ dots & orall t arphi(p) \ dots & rac{a}{gcd(a, arphi(p))} | t \ dots & t = rac{a}{gcd(a, arphi(p))} i \ dots & ext{$\stackrel{\circ}{=}$ } \exists t \in \mathbb{Z}, x \equiv g^{c + rac{arphi(p)}{gcd(a, arphi(p))} i} (mod \ p) \end{aligned}$

题目:

P3306 [SDOI2013] 随机数生成器

$$egin{aligned} x_{i+1} &\equiv ax_i + b (mod\ p) \ x_{i+1} + rac{b}{a-1} &\equiv a(x_i + rac{b}{a-1}) (mod\ p) \ x_n &\equiv t (mod\ p) \ a^{n-1}(x_1 + rac{b}{a-1}) - rac{b}{a-1} &\equiv t (mod\ p) \ a^{n-1} &\equiv (t + rac{b}{a-1}) * inv(x_1 + rac{b}{a-1}) (mod\ p) \end{aligned}$$

```
#include<bits/stdc++.h>
     using namespace std;
     typedef long long 11;
     11 \text{ qpow}(11 \text{ x}, 11 \text{ y}, 11 \text{ mod})  {
          11 \text{ ans} = 1;
6
          while(y) {
7
              if(y & 1) ans = ans * x \% mod;
               x = x * x % mod;
8
9
               y >>= 1;
10
11
          return ans;
```

```
12
    }
13
    unordered_map<11, 11> mp;
14
    11 bsgs(11 a, 11 b, 11 p) {
15
16
         if(a \% p == 0) return -1;
17
         mp.clear();
18
         11 k = ceil(sqrt(p));
19
         for(int i=0; i<=k; i++) {
20
             mp[b] = i;
21
             b = b * a % p;
22
         }
23
         11 aa = qpow(a, k, p), A = aa;
24
         for(int i=1; i<=k; i++) {
25
             if(mp[aa]) {
                 return 1]] * i * k - mp[aa] + 1;
26
27
             }
28
             aa = aa * A % p;
29
         }
30
         return -1;
31
    }
32
    int main() {
33
    #ifndef ONLINE_JUDGE
         freopen("in.txt", "r", stdin);
34
35
         freopen("out.txt", "w", stdout);
36
    #endif
        int t;
37
38
         cin >> t;
39
         while(t--) {
40
            11 p, a, b, x, t;
41
             cin >> p >> a >> b >> x >> t;
42
             if(x == t) {
43
                 cout << 1 << endl;</pre>
44
                 continue;
45
             }
46
             if(a == 0) {
47
                 if(b == t) cout << 2 << end1;
48
                 else cout << -1 << endl;
49
                 continue;
             }
50
51
             if(a == 1) {
52
                 if(b == 0) cout << -1 << end];
53
                 else {
54
                     11 k = qpow(b, p-2, p);
55
                     cout << ((t-x+p)%p*k) % p + 1 << endl;
56
                 }
57
                 continue;
58
             }
59
             11 tmp = b*qpow(a-1, p-2, p) \% p;
60
61
             t = (t + tmp) \% p;
             t = t * qpow((x+tmp)%p, p-2, p) % p;
62
63
             ll ans = bsgs(a, t, p);
             cout << ans << endl;</pre>
64
         }
65
66
         return 0;
67
    }
```

261. Discrete Roots模板

```
#include<bits/stdc++.h>
 2
    using namespace std;
 3
    typedef long long 11;
 4
    11 \text{ qpow}(11 \text{ x}, 11 \text{ y}, 11 \text{ mod})  {
         11 \text{ ans} = 1;
 6
         while(y) {
 7
             if(y \& 1) ans = ans * x % mod;
 8
             x = x * x % mod;
 9
             y >>= 1;
10
11
         return ans;
12
    }
13
    11 G(11 p) {
14
         if(p == 2) return 1;
         vector<11> tmp;
15
         int phi = p-1, n = phi;
16
17
         for(int i=2; 111*i*i<=n; i++) {
18
             if(n \% i == 0) {
19
                 tmp.push_back(i);
20
                 while(n \% i == 0) n /= i;
21
             }
22
         }
23
         if(n > 1) tmp.push_back(n);
24
         for(int i=1; i<=p; i++) {
25
             bool flag = 0;
26
             for(auto it : tmp) {
27
                 if(qpow(i, phi/it, p) == 1) {
28
                      flag = 1;
29
                      break;
30
31
             }
32
             if(!flag) return i;
33
         }
34
         return -1;
35
36
    unordered_map<11, 11> mp;
37
    11 bsgs(11 a, 11 b, 11 p) {
38
         if(a % p == 0 && b != 0) return -1;
39
         mp.clear();
         int k = ceil(sqrt(p));
40
41
         for(int i=0; i<=k; i++) {
             mp[b] = i;
42
             b = b * a % p;
43
44
45
         11 aa = qpow(a, k, p), A = aa;
46
         for(int i=1; i<=k; i++) {
             if(mp[A]) return 111 * i * k - mp[A];
47
48
             A = A * aa % p;
49
         }
50
         return -1;
```

```
51 }
52
53
    int main() {
54 #ifndef ONLINE_JUDGE
        freopen("in.txt", "r", stdin);
55
        freopen("out.txt", "w", stdout);
56
57
    #endif
58
        11 p, k, a;
59
        while(cin \gg p \gg k \gg a) {
60
            if(a == 0) {
                 cout << 1\n0\n;
61
62
                 continue;
63
            }
            11 g = G(p), gk = qpow(g, k, p);
64
65
            11 x0 = bsgs(gk, a, p), t = bsgs(g, a, p);
66
            if(x0 == -1) {
67
                 cout << 0\n;
68
                 continue;
            }
69
70
            vector<11> ans;
71
72
            11 d = \underline{gcd(p-1, k)}, mod = (p-1)/d;
73
            x0 = x0 \% mod;
74
            for(int i=0; i<d; i++) {//最多只有d个不同的解。
75
                 ans.push_back(qpow(g, (x0+i*mod\%(p-1))\%(p-1), p));
76
            }
77
            sort(ans.begin(), ans.end());
78
            cout << ans.size() << endl;</pre>
79
            for(auto it : ans) cout << it << ' ';</pre>
80
            cout << endl;</pre>
81
        }
82
        return 0;
83 }
```

F. Lunar New Year and a Recursive Sequence

题意:

给出
$$f_1=f_2=\cdots=f_{k-1}=1$$
和 $b_1,b_2\cdots b_k$,还有递推方程 $f_i=f_{i-1}^{b_1}f_{i-2}^{b_2}\cdots f_{i-k}^{b_k}$

问是否存在一个 f_k 使得 $f_n \equiv m \pmod{p}$ 成立;

思路:

矩阵快速幂+BSGS

因为n很大,所以不能直接推出 f_n 是 f_k 的多少次方,观察递推式全是乘法,我们就想到用矩阵加速。

先推出矩阵。

假设k=4,则可以推出下面的式子。

$$egin{aligned} f_k &= f_k \ f_{k+1} &= f_k^{b_1} f_{k-1}^{b_2} f_{k-2}^{b_3} f_{k-3}^{b_4} = f_k^{b_1} \ f_{k+2} &= f_{k+1}^{b_1} f_k^{b_2} f_{k-1}^{b_3} f_{k-2}^{b_4} = (f_k^{b_1})^{b_1} f_k^{b_2} = f_k^{b_1^2 + b_2} \ f_{k+3} &= f_{k+2}^{b_1} f_{k+1}^{b_2} f_k^{b_3} f_{k-1}^{b_4} = f_k^{b_1^3 + 2b_1 * b_2 + b_3} \ f_{k+4} &= f_{k+3}^{b_1} f_{k+2}^{b_2} f_{k+1}^{b_3} f_k^{b_4} = f_k^{b_1^4 + 3b_1^2 b_2 + 2b_1 b_3 + b_2^2 + b_4} \ f_{k+5} &= f_{k+4}^{b_1} f_{k+3}^{b_2} f_{k+2}^{b_3} f_{k+1}^{b_4} \end{aligned}$$

我们可以看出每个 f_i 都是由前k个f值推出来的,所以我们设 f_i 的对应的 f_k 的系数是 g_i

我们就能找到一个一维的矩阵乘法

$$egin{aligned} \left[\, g_4, g_3, g_2, g_1 \,
ight] imes egin{bmatrix} b_1 \ b_2 \ b_3 \ b_4 \ \end{bmatrix} = g_5 \ \ & \begin{bmatrix} g_5, g_4, g_3, g_2 \,
ight] imes egin{bmatrix} b_1 \ b_2 \ b_3 \ b_4 \ \end{bmatrix} = g_6 \end{aligned}$$

所以我们把矩阵扩展到k维,希望得到一个包含 g_i 的矩阵乘以另一个矩阵得到包含形式相同且包含 g_{i+1} 的矩阵,

推出:

$$\begin{bmatrix} g_7 & g_6 & g_5 & g_4 \\ g_6 & g_5 & g_4 & g_3 \\ g_5 & g_4 & g_3 & g_2 \\ g_4 & g_3 & g_3 & g_1 \end{bmatrix} \times \begin{bmatrix} b_1 & 1 & 0 & 0 \\ b_2 & 0 & 1 & 0 \\ b_3 & 0 & 0 & 1 \\ b_4 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} g_8 & g_7 & g_6 & g_5 \\ g_7 & g_6 & g_5 & g_4 \\ g_6 & g_5 & g_4 & g_3 \\ g_5 & g_4 & g_3 & g_2 \end{bmatrix}$$

通过k=4的特例,我们能够推出k是范围内的任意矩阵。

我们就能推出 f_n 是 f_k 的多少次方,设 $f_n = f_k^a$ 。

现在的问题是已知a,求是否存在 f_k 使 $f_k^a \equiv m \pmod{p}$ 成立。

这个就是之前的bsgs的进阶篇: 进阶篇

Code:

```
1 #include<bits/stdc++.h>
 2 using namespace std;
 3 typedef long long 11;
4 const int N = 1e2+10;
5 const int mod = 998244353;
  vector<11> a:
 7
   11 b[N], k;
8
   struct matrix {
9
        11 s[N][N];
10
        matrix operator * (const matrix &t) {
11
            matrix tmp;
            for(int i=1; i<=k; i++) {
12
13
                for(int j=1; j<=k; j++) {
14
                    tmp.s[i][j] = 0;
15
                    for(int l=1; l<=k; l++) {
```

```
16
                          tmp.s[i][j] = (tmp.s[i][j] + s[i][l] * t.s[l][j] %
    (mod-1)) \% (mod-1);
17
                      }
                 }
18
19
             }
20
             return tmp;
21
         }
22
    } A, I, B;
23
24
    matrix mqpow(matrix t, 11 y) {
25
        matrix ans = I;
26
         while(y) {
27
             if(y \& 1) ans = ans * t;
28
             t = t * t;
29
             y >>= 1;
30
         }
31
         return ans;
32
    }
33
    11 \text{ qpow}(11 x, 11 y)  {
34
35
        11 \text{ ans} = 1;
36
         while(y) {
37
             if(y \& 1) ans = ans * x % mod;
38
             x = x * x \% mod;
39
             y >>= 1;
40
         }
41
         return ans;
42
    }
43
44
    11 G(11 p) {
45
         vector<int> v;
46
         ll phi = p-1, tmp = phi;
47
         for(int i=2; 111*i*i<=tmp; i++) {</pre>
48
             if(tmp \% i == 0) {
49
                 v.push_back(i);
                 while(tmp \% i == 0) tmp /= i;
50
51
             }
52
         }
53
         if(tmp > 1) v.push_back(tmp);
54
         for(int i=2; i<=phi; i++) {</pre>
55
             int f = 1;
             for(auto it : v) {
56
                 if(i % it == 0) {
57
                      f = 0;
58
59
                      break;
60
                 }
61
                 if(f) return i;
62
             }
         }
63
64
         return -1;
65
    unordered_map<11, 11> mp;
66
    11 bsgs(11 aaa, 11 b, 11 p) {
67
68
         mp.clear();
69
         if(aaa % p == 0 && b % p != 0) return -1;
         int k = ceil(sqrt(p));
70
71
         for(int i=0; i<=k; i++) {
             mp[b] = i, b = b * aaa % mod;
72
```

```
73
         }
 74
         11 aa = qpow(aaa, k), A = aa;
 75
         // cout << aa<< endl;</pre>
 76
         for(int i=1; i<=k; i++) {
 77
              if(mp[A]) return 111*i*k - mp[A];
 78
             A = A * aa % mod;
 79
         }
 80
         return -1;
 81
     }
 82
     int main() {
     #ifndef ONLINE_JUDGE
 83
 84
         freopen("in.txt", "r", stdin);
 85
         freopen("out.txt", "w", stdout);
 86
     #endif
         scanf("%11d", &k);
 87
         for(int i=1; i<=k; i++) I.s[i][i] = 1;
 88
 89
         for(int i=1; i<=k; i++) {
 90
              scanf("%11d", &b[i]);
 91
         }
 92
         for(int i=1; i < k; i++) a.push_back(0);
 93
         a.push_back(1);
         for(int i=1; i<=k; i++) {
 94
 95
              11 tmp = 0;
 96
              for(int j=1; j<=k; j++) {
 97
                  tmp = (tmp + b[j] * a[a.size()-j] % (mod-1)) % (mod-1);
 98
              }
 99
              a.push_back(tmp);
100
         }
101
         11 n, m, aa;
         scanf("%11d%11d", &n, &m);
102
103
         if(n \le 2*k) {
104
              aa = a[n-1] \% (mod-1);
105
         }
106
         else {
107
              for(int j=1, i=a.size()-1; j <= k; j++, i--) {
108
                  int 1 = 1;
109
                  while(1 \le k) {
110
                      A.s[j][1] = a[i-1+1];
                      // cout << i-l+1 << endl;
111
112
                      1++;
113
                  }
114
115
              for(int i=1; i<=k; i++) {
                  for(int j=1; j<=k; j++) {
116
117
                      if(i == 1) B.s[j][i] = b[j];
                      else if(j == i-1) {
118
119
                           B.s[j][i] = 1;
120
                      }
                  }
121
122
              }
123
              A = A * mqpow(B, n-2*k);
124
              aa = A.s[1][1]\%(mod-1);
125
         }
         11 g = G(mod);
126
127
         11 ans = bsgs(qpow(g, aa), m, mod);
128
         if(ans != -1) cout << qpow(g, ans);
129
         else cout << -1 << endl;
130
         return 0;
```

扩展篇

问题:

当p. a可能不互质时,求

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

方法:

当 $d_1 = gcd(a,p) \neq 1$, 当 $d_1 \nmid b$ 时, 无解, 则原式变成:

$$rac{a}{d_1}a^{x-1}\equivrac{b}{d_1}(mod\;rac{p}{d_1})$$

当设 $d_2=\gcd(a,rac{p}{d_1})
eq 1$,当 $d_2
mid rac{b}{d_1}$ 时,无解,则原式可变成:

$$rac{a^2}{d_1 d_2} a^{x-2} \equiv rac{b}{d_1 d_2} (mod \ rac{p}{d_1 d_2})$$

重复直到 $d_{cnt}=1$ 。设 $D=\prod_{i=1}^{cnt}d_i$,原式可以写成:

$$rac{a^{cnt}}{D}a^{x-cnt}\equivrac{b}{D}(mod\;rac{p}{D})$$

剩下的 $a^{x-cnt}, rac{b}{D}, rac{p}{D}$,可以构成基础篇的bsgs,修改的地方就是乘了一个系数 $rac{a^{cnt}}{D}$ 。

注意:cnt可能大于x这个解,所以我们在做除法的时候就要判断是不是能够相等,相等直接输出当前的cnt。

题目:

P4195 【模板】扩展BSGS

```
#include<bits/stdc++.h>
     using namespace std;
     typedef long long 11;
     11 \text{ qpow}(11 \text{ x}, 11 \text{ y}, 11 \text{ mod})  {
 5
          11 \text{ ans} = 1;
 6
          while(y) {
 7
               if(y \& 1) ans = ans * x % mod;
               x = x * x % mod;
 8
 9
               y >>= 1;
10
          }
          return ans;
11
12
13
     unordered_map<11, 11> mp;
```

```
11 bsgs(11 a, 11 b, 11 p, 11 ad) {
14
15
         if(a % p == 0 && b%p != 0) return -1;
16
         mp.clear();
17
        int k = ceil(sqrt(p));
18
         for(int i=0; i<k; i++) {
19
             mp[b] = i;
20
             b = b * a % p;
21
         }
22
        11 aa = qpow(a, k, p), A = aa*ad % p;
23
         for(int i=1; i<=k; i++) {
             if(mp[A]) return 1||1*i*k - mp[A];
24
25
             A = A * aa % p;
26
         }
27
         return -1;
28
    }
29
    11 exgcd(11 &x, 11 &y, 11 a, 11 b) {
30
         if(b == 0) {
31
             x = 1, y = 0;
32
             return a;
33
         }
        11 d = exgcd(x, y, b, a\%b);
34
35
        11 tmp = y;
36
        y = x - a/b * y;
37
        x = tmp;
38
         return d;
39
    11 inv(11 a, 11 b) {
40
41
        11 x, y;
42
         11 d = exgcd(x, y, a, b);
         return x;
43
44
    }
45
46
    11 exbsgs(11 a, 11 b, 11 p) {
47
         a \% = p, b \% = p;
48
         if(b == 1 || p == 1) return 0;
49
        11 cnt = 0, d, tmp = 1;
50
        while((d = \underline{gcd}(a, p)) \land 1) {
             if(b % d) return -1;
51
52
             cnt++, b /= d, p /= d;
53
             tmp = tmp * a/d % p;
54
             if(tmp == b) return cnt;
55
         }
56
        ll ans = bsgs(a, b, p, tmp);
         if(ans == -1) return -1;
57
58
         else return ans + cnt;
59
    }
60
    int main() {
61
    #ifndef ONLINE_JUDGE
         freopen("in.txt", "r", stdin);
62
        freopen("out.txt", "w", stdout);
63
    #endif
64
65
         11 a, p, b, d;
        while(cin \gg a \gg p \gg b) {
66
             if(!a || !p || !b) break;
67
68
             ll ans = exbsgs(a, b, p);
69
             if(ans != -1) cout << ans << end];
70
             else cout << "No Solution\n";</pre>
71
         }
```

阶和原根

阶

定义:

对于m>1且(a,m)=1,使 $a^n\equiv 1 \pmod m$ 成立的最小的n,称为n模m的阶,记作 $\delta_m(a)$ 。根据欧拉定理,可以证明至少存在一个n使 $a^n\equiv 1 \pmod p$ 成立。

性质:

1. $a, a^2, \ldots, a^{\delta_m(a)}$ 在模m下互不同余。

证明:

假设存在 $i,\ j\in[1,\delta_m(a)]$),使 $a^i\equiv a^j(mod\ m)$,则可以变形成 $a^{|i-j|}\equiv 1(mod\ m)$,易知 $|i-j|<\delta_m(a)$,与阶的定义相矛盾,所以假设不成立,故原命题成立。

2. 若
$$a^n \equiv 1 (mod \ m)$$
,则 $\delta_m(a) | n$ 。

证明:

假设 $\delta_m(a) \nmid n$,则 $n=k\delta_m(a)+r$ (r>0), $a^{k\delta_m(a)}a^r\equiv a^r\equiv 1 (mod\ m)$,根据性质1,可知假设不成立,原命题成立。

可以推出: $a^p \equiv a^q \pmod{m}$, 则 $p \equiv q \pmod{\delta_m(a)}$ 。

3. 设
$$m>1$$
, a , $b\in\mathbb{Z}$, $(a,m)=(b,m)=1$, 则 $\delta_m(ab)=\delta_m(a)\delta_m(b)$ 的充要条件是: $(\delta_m(a),\delta_m(b))=1$

证明:

必要性:

$$egin{aligned} \therefore a^{\delta_m(a)} &\equiv 1 (mod \ p)$$
和 $b^{\delta_m(b)} \equiv 1 (mod \ m) \ &\therefore (ab)^{[\delta_m(a),\delta_m(b)]} \equiv 1 (mod \ m) \ &dots$ 生质 $2 \ &\therefore \delta_m(ab) \mid [\delta_m(a),\delta_m(b)] \ &dots \delta_m(a)\delta_m(b) \mid = \delta_m(ab) \ &\therefore \delta_m(a)\delta_m(b) \mid [\delta_m(a),\delta_m(b)] \ &\therefore (\delta_m(a),\delta_m(b)) = 1 \end{aligned}$

充分性:

$$egin{aligned} egin{aligned} egin{aligned} & egin{aligned} \cdot (ab)^{\delta_m(ab)} & \equiv 1 (mod \ m) \\ & egin{aligned} \cdot (ab)^{\delta_m(ab)} \delta_m(b) & \equiv a^{\delta_m(ab)} \equiv 1 (mod \ m) \\ & egin{aligned} \cdot \delta_m(a) \mid \delta_m(ab) \\ & \equiv \delta_m(a) \delta_m(b) \mid \delta_m(ab) \\ & egin{aligned} \cdot \delta_m(a) \delta_m(b) \mid \delta_m(ab) \\ & \vdots a^{\delta_m(a)} b^{\delta_m(b)} \equiv (ab)^{\delta_m(a)\delta_m(b)} \\ & egin{aligned} \cdot \delta_m(ab) \mid \delta_m(a)\delta_m(b) \\ & \Leftrightarrow \mathbb{E} \colon \delta_m(ab) = \delta_m(a)\delta_m(b) \end{aligned}$$

4. 设 $k\in\mathbb{N}, m>1, a\in\mathbb{Z}, (a,m)=1$,则: $\delta_m(a^k)=rac{\delta_m(a)}{(\delta_m(a),k)}.$

证明:

$$egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} & egi & egin{aligned} & egin{aligned} & egin{aligned} & egin{align$$

原根

定义:

设 $m\in\mathbb{N}^*$, $a\in\mathbb{Z}$.若(a,m)=1,且 $\delta_m(a)=\phi(m)$,则称a为模m的原根(注:a,m互质).

原根判定定理

若g是模m的一个原根,则对于 $\phi(m)$ 的任何大于1且不为自身的因数p,都有 $g^{\frac{\phi(m)}{p}}\equiv 1 \pmod{m}$. 证明:

假设存在一个 $t < \mu(m)$ 使得 $a^t \equiv 1 (mod \ m)$ 且 $\forall i \in [1,k] : a^{\frac{\phi(m)}{d_i}} \equiv 1 (mod \ m).$

由裴蜀定理得,一定存在一组k。x满足 $kt=x\phi(m)+(t,\mu(m))$; 由欧拉定理/费马小定理得 $a^{\phi(p)}\equiv 1 \pmod{p}$;

$$egin{aligned} \therefore 1 &\equiv a^{kt} \equiv a^{x\phi(m)} a^{(t,\phi(m))} \equiv a^{(t,\phi(m))} (mod\ m) \ &\because t < \phi(m)\ \therefore (t,\phi(m)) \leq t < \phi(m) \ &\because (t,\phi(m)) | \phi(m),\ \therefore (t,\phi(m)) -$$
定至少整除 $rac{\phi(m)}{d_i}$ 中的至少一个。
$$& \mathcal{L}(t,\phi(m)) | rac{\phi(m)}{d_i}, \mathbb{M} a^{rac{\phi(m)}{d_i}} \equiv a^{(t,\phi(m))} \equiv 1 (mod\ m). \end{aligned}$$

.:.假设不成立,原命题成立

原根的个数

如果一个数m有原根g,则它的原根个数为 $\phi(\phi(m))$ 。

证明:

如果m存在原根g,则

$$\delta_m(g^k) = rac{\delta_m(g)}{(\delta_m(g),k)}$$
 (阶的性质) $if(k,\delta_m(g)) = 1$ 且 $1 < k < \phi(m)$ 的 k 有 $\phi(\phi(m))$ 个。

原根一定是 $\phi(\phi(m))$ 个,在模m的情况下。

原根的存在定理

一个数m存在原根当且仅当m=2, 4, p^{α} , $2p^{\alpha}$, 其中p为奇素数。 $\alpha\in\mathbb{N}^*$ 。

O(1)快速乘

```
1 inline ll multi(ll x, ll y, ll mod) {
2     ll tmp=(x*y-(l)((long double)x/mod*y+1.0e-8)*mod);
3     return tmp<0 ? tmp+mod : tmp;
4 }</pre>
```

python进制转化

```
def C(a, b, c):
        s = ''
 2
 3
        k, I = 0, 1
        for i in reversed(c):
4
 5
            if '0' <= i <= '9':
 6
               k += (ord(i) - ord('0')) * I
            elif 'A' <= i <= 'Z':
 7
8
               k += (ord(i) - ord('A') + 10) * I
9
            else:
                k += (ord(i) - ord('a') + 36) * I
10
11
            I = I * a
        if k == 0:
12
           return '0'
13
14
        while k:
           m = k \% b
15
16
           if 0 <= m <= 9:
17
               s = s + str(chr(ord('0') + m))
           if 10 <= m <= 35:
18
               s = s + str(chr(ord('A') + m-10))
19
20
            if 36 <= m <= 61:
21
              s = s + str(chr(ord('a') + m-36))
            k //= b
22
23
            # reversed(s)
24
        return s
25
26
27 \mid x = input()
28 \ y = []
29 tmp = ''
30 for i in x:
31
      if i == ' ':
          y.append(tmp)
32
           tmp = ''
33
34
        else:
35
            tmp += i
36 y.append(tmp)
|a| = eval(y[0])
38 b = eval(y[1])
39 cc = C(a, b, y[2])
40 ans = ''
41 for i in reversed(cc):
42
      ans += i
43
      print(i, end='')
44 # print()
45 | # for i in reversed(C(b, a, ans)):
46 # print(i, end='')
```