# 6.7 BSGS与扩展BSGS

**6.7.1 概述**

主要用于解决an=b(mod p) 问题，求解最小非负整数n，使其满足左式

普通BSGS求解a与p互质的情况，扩展BSGS可求解不互质情况

**6.7.2 普通BSGS**

**6.7.2.1解题步骤**

1.取m=ceil(sqrt(p))（向上取整），若解存在，可令n=i\*m-j，即数对(i,j)存在

2.an = b ( mod p )⬄ai\*m-j = b( mod p )⬄ai\*m = b\*aj ( mod p )

3.再0-m范围内枚举b\*aj的值，并用hash表存储j

4.在1-m范围内枚举ai\*m的值，在hash表中查找，若找到，则输出i\*m-j，反之无答案

**6.7.2.2代码模板**

1. #include<bits/stdc++.h>
2. #define int long long
3. **using** **namespace** std;
4. **int** a,p,b;
5. unordered\_map<**int**,**int**> mp;
6. **int** qpow(**int** a,**int** n,**int** mod)
7. {
8. **int** ans=1;
9. **while**(n)
10. {
11. **if**(n&1) ans=ans\*a%mod;
12. a=a\*a%mod;
13. n>>=1;
14. }
15. **return** ans;
16. }
17. **int** bsgs(**int** a,**int** b,**int** p)
18. {
19. a%=p;
20. b%=p;
21. **if**(1==b%p) **return** 0;  //特判结果为0的情况
22. **int** m=ceil(sqrt(p));
23. **int** baj=1;
24. **for**(**int** i=0;i<=m;i++)
25. {
26. **if**(i==0)
27. {
28. baj=b%p;
29. mp[baj]=i;
30. **continue**;
31. }
32. baj\*=a;
33. baj%=p;
34. mp[baj]=i;
35. }
36. **int** am=qpow(a,m,p);
37. **int** tmp=1;
38. **for**(**int** i=1;i<=m;i++)//由于假设n为i\*m-j，若i从0开始枚举则有可能输出负数
39. {
40. tmp\*=am;
41. tmp%=p;
42. **if**(mp.count(tmp)) **return** i\*m-mp[tmp];
43. }
44. **return** -1;
45. }
46. **signed** main()
47. {
48. cin.tie(0);
49. cout.tie(0);
50. ios::sync\_with\_stdio(0);
51. **while**(cin>>a>>p>>b)
52. {
53. mp.clear();
54. **if**(a==0&&p==0&&b==0) **break**;
55. **int** ans=bsgs(a,b,p);
56. **if**(ans==-1) cout<<"No Solution"<<endl;
57. **else** cout<<ans<<endl;
58. }
59. **return** 0;
60. }

**6.7.3 扩展BSGS**

**6.7.3.1 解题步骤**

1.当a与p互质时，套用普通BSGS即可

2.当a与p不互质时，设gcd为a与p的最大公约数，对于方程an=b ( mod p ) ，一定有 a/gcd \* an-1 = b/gcd ( mod p/gcd )存在，因此若b不为gcd的倍数，则方程无解

3.对于新方程 令 b/ gcd /( a / gcd) = b’ , p/gcd = p’ , 用逆元处理b’ , 递归求解即可

**6.7.3.2代码模板**

1. #include<bits/stdc++.h>
2. **using** **namespace** std;
3. #define int long long
4. **const** **int** inf=1e8;
5. **int** a,p,b,x,y;
6. **int** qpow(**int** a,**int** n,**int** mod)
7. {
8. **int** ans=1;
9. **while**(n)
10. {
11. **if**(n&1) ans=ans\*a%mod;
12. a=a\*a%mod;
13. n>>=1;
14. }
15. **return** ans;
16. }
17. **int** exgcd(**int** a,**int** b,**int** &x,**int** &y)
18. {
19. **if**(!b)
20. {
21. x=1;
22. y=0;
23. **return** a;
24. }
25. **int** ans=exgcd(b,a%b,x,y);
26. **int** tmp=x;
27. x=y;
28. y=tmp-(a/b)\*y;
29. **return** ans;
30. }
31. unordered\_map<**int**,**int**> mp;
32. **int** bsgs(**int** a,**int** b,**int** p)
33. {
34. mp.clear();
35. a=a%p;
36. //b=(b%p+p)%p;
37. **if**(1%p==b%p) **return** 0;//特判结果为0的情况
38. **int** m=ceil(sqrt(p));
39. **int** baj=1;
40. **for**(**int** i=0;i<=m;i++)
41. {
42. **if**(i==0)
43. {
44. baj=b%p;
45. mp[baj]=i;
46. **continue**;
47. }
48. baj\*=a;
49. baj%=p;
50. mp[baj]=i;
51. }
52. **int** am=qpow(a,m,p);
53. **int** tmp=1;
54. **for**(**int** i=1;i<=m;i++)
55. {
56. tmp\*=am;
57. tmp%=p;
58. **if**(mp.count(tmp)) **return** i\*m-mp[tmp];
59. }
60. **return** -1;
61. }
62. **int** exbsgs(**int** a,**int** b,**int** p)
63. {
64. b=(b%p+p)%p;  //令b为正
65. **if**(1%p==b%p) **return** 0;
66. **int** gcd=exgcd(a,p,x,y);
67. **if**(gcd>1)
68. {
69. **if**(b%gcd) **return** -inf;
70. exgcd(a/gcd,p/gcd,x,y);
71. **int** a2=a;
72. **int** b2=b/gcd\*x%(p/gcd);
73. **int** p2=p/gcd;
74. **return** exbsgs(a2,b2,p2)+1;
75. }
76. **return** bsgs(a,b,p);
77. }
78. **signed** main()
79. {
80. ios::sync\_with\_stdio(0);
81. **while**(cin>>a>>p>>b)
82. {
83. **if**(!a&&!b&&!p) **break**;
84. **int** ans=exbsgs(a,b,p);
85. **if**(ans<0) cout<<"No Solution"<<endl;
86. **else** cout<<ans<<endl;
87. }
88. **return** 0;
89. }

**6.7.4 经典例题**

**6.7.4.1 洛谷P4861 按钮**

**题目描述**

Ada被关在了一个房间里。房间的铁门上有一个按钮，还有一个显示屏显示着“1”。

旁边还有一行小字：“这是一个高精度M进制计算器，每按一次按钮，屏幕上的数便会乘以K。当个位数再次变为1时，门就开了。”

由于Ada急于出去，所以你要在1s之内求出她的最小按键次数**。**

**输入格式**

一行，两个整数m和k

**输出格式**

一行一个数字，表示最小按键次数。

如果无论Ada按多少次都无法让门打开，输出"Let's go Blue Jays!"（不含引号）。

**输入样例1**

11 2

**输出样例1**

10

**输入样例2**

6 26

**输出样例2**

Let's go Blue Jays!

**思路：若最后答案存在，则有两种表达方式，kn和m\*x+1，于是可以列出方程**

**kn = 1 ( mod m ) ,套用由于答案不能为0，于是特判一下原模板中输出0的情况即可**

**题解**

1. #include<bits/stdc++.h>
2. **using** **namespace** std;
3. #define int long long
4. **int** m,k;
5. unordered\_map<**int**,**int**> mp;
6. **int** exgcd(**int** a,**int** b,**int** &x,**int** &y)
7. {
8. **if**(b==0)
9. {
10. x=1;
11. y=0;
12. **return** a;
13. }
14. **int** ans=exgcd(b,a%b,x,y);
15. **int** tmp=x;
16. x=y;
17. y=tmp-(a/b)\*y;
18. **return** ans;
19. }
20. **int** qpow(**int** a,**int** n,**int** mod)
21. {
22. **int** ans=1;
23. **while**(n)
24. {
25. **if**(n&1) ans=ans%mod\*a%mod;
26. a=a%mod\*a%mod;
27. n>>=1;
28. }
29. **return** ans;
30. }
31. **int** bsgs(**int** a,**int** b,**int** p)
32. {
33. a%=p;
34. b%=p;
35. **int** m=ceil(sqrt(p));
36. **int** tmp;
37. **for**(**int** i=0;i<=m;i++)
38. {
39. **if**(i==0)
40. {
41. tmp=b%p;
42. mp[tmp]=i;
43. **continue**;
44. }
45. tmp\*=a;
46. tmp%=p;
47. mp[tmp]=i;
48. }
49. **int** am=qpow(a,m,p);
50. **int** now=1;
51. **for**(**int** i=1;i<=m;i++)
52. {
53. now\*=am;
54. now%=p;
55. //cout<<"now="<<now<<endl;
56. **if**(mp.count(now)&&i\*m-mp[now]==0) **continue**;  //特判结果为0的情况
57. **else** **if**(mp.count(now)) **return** i\*m-mp[now];
58. }
59. **return** -1;
60. }
61. **int** x,y;
62. **signed** main()
63. {
64. cin>>m>>k;
65. **int** gcd=exgcd(m,k,x,y);
66. **if**(gcd!=1)
67. {
68. cout<<"Let's go Blue Jays!"<<endl;
69. **return** 0;
70. }
71. **int** ans=bsgs(k,1,m);
72. **if**(ans==-1) cout<<"Let's go Blue Jays!"<<endl;
73. **else** cout<<ans<<endl;
74. **return** 0;
75. }