小步大步算法：$BSGS(Baby\ Step\ Giant\ Step)$

~~拔山盖世算法，百度搜索谷歌搜索算法~~

用来求解离散对数(即模意义下的对数)的算法。

给出：$a^x \equiv b \pmod m$中a,b,m值，(a,m互质)，求解x。

由欧拉定理可知：a在模m下有长度为$\varphi(m)$的循环节。

所以朴素的暴力算法就是枚举$O(\varphi(m))$，最坏情况下是$O(m-1)$，即m为素数。

$BSGS$就是暴力算法的优化。

记：$x=At-B$。

有：$a^x \equiv a^{At-B} \equiv b \pmod m \Rightarrow a^{At} \equiv ba^B \pmod m$

B的取值有$\varphi(m) \bmod t$ 个，A的取值有$\lfloor \dfrac{\varphi(m)}{t} \rfloor$个

显然是双钩函数最小值当$t=\lceil \sqrt{\varphi(m)} \rceil$是最优的。

这里为了方便直接取$t=\lceil \sqrt{m} \rceil$就可以了。

然后枚举所有的B对应的右式所有值，储存下来。

然后再对A进行枚举当出现之前的数说明此时的$At-B$就是答案。

代码：

```
ll ksm(ll a,ll n,ll m){
    ll ans=1;
    while(n){
        if(n&1) ans=ans*a%m;
        a=a*a%m;
        n>>=1;
    }
    return ans;
}
ll BSGS(ll a,ll b,ll m){
    unordered_map<ll,ll>mp;
    ll r=b*a%m,t=sqrt(m)+1;
    for(int B=1;B<=t;B++){
        mp[r]=B;
        r=(r*a)%m;
    }
    ll at=ksm(a,t,m),l=at;
    for(int A=1;A<=t;A++){
        if(mp[l]) return A*t-mp[l];
        l=(l*at)%m;
    }
    return -1;
```

```
}
```

模板题：

P3846 [TJOI2007] 可爱的质数/【模板】BSGS

貌似数据太弱了，不用判最小也能过？

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e3+5,M=2e4+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a,b) memset(a,b,sizeof a)
#define PII pair<int,int>
#define fi first
#define se second
#define pb push_back
ll ksm(ll a,ll n,ll m){
    ll ans=1;
    while(n){
        if(n&1) ans=ans*a%m;
        a=a*a%m;
        n>>=1;
    }
    return ans;
}
ll BSGS(ll a,ll b,ll m){
    unordered_map<ll,ll>mp;
    ll r=b*a%m,t=sqrt(m)+1;
    for(int B=1;B<=t;B++){
        mp[r]=B;
        r=(r*a)%m;
    }
    ll at=ksm(a,t,m),l=at;
    ll ans=1e15;
    for(int A=1;A<=t;A++){
        if(mp[l]) return A*t-mp[l];
        l=(l*at)%m;
    }
    return -1;
}
int main(){
    ll p,b,n;cin>>p>>b>>n;
    ll ans=BSGS(b,n,p);
    printf(ans==-1?"no solution\n":"%lld\n",ans);
    return 0;
}
```

扩展$BSGS$待补......