Consecutive Remainders

Attempted by: **362** / Accuracy: **87%** / Maximum Score: **50** / 4 Votes

Tag(s): Basic Programming, Medium-Hard

Setter: Artsem Zhuk
Tester: Hasan Jaddouh
Editorialist: Saatwik Singh Nagpal

Prerequisites: Group Theory, Number Theory.

Problem:
Given three numbers p, a, b, where p is prime. Find a few different integers $0 \leqslant x < p$ such that $(x+b)^a \equiv x^a$ (mod p).

Solution:
We are given a field of numbers to us with multiplication and addition modulo p defined in it. Now, let us find a generator g of the multiplicative group in this field. A generator is such a number in the group which satisfies the following:
The set of elements formed by $g, g^2, g^3, \ldots, g^{p-1}$ contains all the elements of the group, which here would be $1, 2, 3, \ldots, p-1$.
We need it because we can represent the above equivalence in terms of the generator as each of the elements can be generated using the generator. Let $x \equiv g^i$ (mod p) and $b \equiv g^j$ (mod p), for some i and j. So, our equivalence becomes :
$(g^i + g^j)^a \equiv (g^i)^a$ (mod p)
Let $inv(y)$ be the inverse of y in this field. If we multiply both sides by $inv(g^{i*a})$, we get:
$inv(g^{i*a}) * (g^i(1+g^{j-i}))^a \equiv inv(g^{i*a}) * g^{i*a}$ (mod p)
$\implies inv(g^{i*a}) * g^{i*a} * (1+g^{j-i})^a \equiv 1$ (mod p)
$\implies (1+g^{j-i})^a \equiv 1$ (mod p)

Remember that $1+g^{j-i}$ (mod p) also belongs to the same field, hence it can also be generated by the generator g. Let $1+g^{j-i} \equiv g^k$ (mod p). So, we have the following :
$g^{k*a} \equiv 1$ (mod p)
We claim that $k*a$ is divisible by $p-1$, for the above to be true. This is because as g is a generator, so only the values $g^0, g^{p-1}, g^{2p-2}\ldots$ will be unity. So, $k*a$ is divisible by $p-1$.

Now, we can iterate over k such that $k*a$ is divisible by $p-1$ and $k < p$, and use the value of g^{k*a} to calculate x. We stop when $k > p$ or number of distinct values of $x = 10$.

But how do we find x, given g^{k*a} (mod p)? Let's see.

We know that:
$1+g^{j-i} \equiv g^k$ (mod p)
$\implies g^{j-i} \equiv g^k - 1$ (mod p)
Substitute g^i as x and g^j as b.

$\implies b * inv(x) \equiv g^k - 1 \pmod p$
$\implies x \equiv inv(g^k - 1) * b \pmod p$

IS THIS EDITORIAL HELPFUL?

Yes, it's helpful

No, it's not helpful

Tester Solution

```
#include <iostream>
#include <assert.h>
#include <random>
using namespace std;
int T;
long long a,b,p;

long long gcd(long long a,long long b){
        if(b==0)return a;
        return gcd(b,a%b);
}

long long pw(long long b,long long k,long long md){
        long long ret=1;
        long long p2=b;
        while(k){
                if(k%2){
                        ret *= p2;
                        ret %= md;
                }
                k/=2;
                p2 *= p2;
                p2 %= md;
        }
        return ret;
}
int sol[11],sl=0;

bool is_prime(int x){
    for(int i=2;i*i<=x;i++){
        if(x%i==0){
            return false;
        }
    }
    return true;
}
int main(){
```

```
        cin.sync_with_stdio(false);
        cin>>T;
        assert(1<=T && T<=10000);
        while(T--){
                cin>>p>>a>>b;
                assert(2<=p && p<=1000000000);
                assert(0<=b && b<=p-1);
                assert(0<=a && a<=1000000000);
                assert(is_prime(p));
                if(b==0){
                        int sol=min(p,10ll);
                        cout<<sol<<endl;
                        for(int i=0;i<sol;i++){
                                cout<<i<<" ";
                        }
                        cout<<endl;
                        continue;
                }
                int g=gcd(p-1,a);
                int num= min(10,g-1);
                sl=0;
                int pwr=(p-1)/g;
                for(int i=0;i<1000;i++){
                        if(sl == num)break;
                        int z=(rand() *1ll* rand() + rand() )% (p-1);
              z += p-1;
              z %= p-1;
              z++;
                        int yy= pw (z,pwr,p);

                        if(yy==1)continue;

                        yy = (b * pw(yy-1,p-2,p))%p;
                        bool ok=true;
                        for(int i=0;i<sl;i++){
                                if(sol[i]==yy)ok=false;
                        }
                        if(ok){
                                sol[sl++]=yy;
                        }
                }
                //assert(sl == num);
                cout<<sl<<endl;
                for(int i=0;i<sl;i++){
                        cout<<sol[i]<<" ";
                //      assert(pw( (sol[i] + b)%p,a,p) == pw(sol[i],a,p));
                }
                cout<<endl;
        }
}
```

LIVE EVENTS

COMMENTS (0)

Join Discussion...

Cancel Post

LIVE EVENTS