algorithm to find inverse modulo m

i have two questions how to find inverse of a number modulo m and n! modulo m

2

inverse modulo number theory

asked **26 Jun '12, 13:07**

vamsi2708
31●2●3●5
accept rate: 0%

3

2 Answers: oldest newest most voted

I hope I understood well, I asked same question (inverse modulo) in OLYMPIC tutorial.

10 $\frac{a}{b}\%P = (a\%P * (b^{P-2})\%P)\%P$

Note: P is prime number

n! is simply

n! = (n%MOD * (n-1)%MOD) % MOD

link edited **09 Dec '12, 20:03**

answered **26 Jun '12, 14:18**

betlista ◆◆
16.5k●49●113●220
accept rate: 11%

Suppose we need to calculate nCr, in these cases, n > P. how to handle these cases?

sandipan (29 Dec '13, 20:08)

9

Firstly, the inverse of an element a in the residue classes modulo m exists if and only if: gcd(a,m) = 1 i.e. they are relatively prime

For finding the inverse, use the extended euclidean algorithm (http://en.wikipedia.org/wiki/Extended_Euclidean_algorithm)

It finds the solution(x,y) to the following equation:

ax + by = gcd(a,b)

Taking b = m, the equation becomes:

ax + my = gcd(a,m)

since gcd(a,m) = 1

ax + my = 1

If we use the modulo m operation on both sides:

ax(mod m) + my(mod m) = 1(mod m)

ax(mod m) = 1(mod m)

=> x is the inverse of a modulo m

algorithm:

```
void EE(int a, int b, int& x, int& y)
{
    if(a%b == 0)
    {
        x=0;
        y=1;
```

Follow this question
By Email:
Once you sign in you will be able to subscribe for any updates here

By RSS:
Answers
Answers and Comments

Tags:

modulo **×94**
number **×24**
inverse **×13**
theory **×5**

Asked: **26 Jun '12, 13:07**

Seen: **4,929 times**

Last updated: **29 Dec '13, 20:08**

Related questions

X^a = b (mod 2k + 1)
Editorial : Wilson's theorem
Modulo inverse calculation verification
inverse modulo for non primes
inverse modulo
is this a property of inverse modulo operation?
finding Inverse Modulo of a range of numbers under modulo m
Decimal to binary conversion.
just a simple sum
modulo calculate

```
        return;
    }
    EE(b,a%b,x,y);
    int temp = x;
    x = y;
    y = temp - y*(a/b);
}
```

Using this function and the explanation above, the inverse function can be implemented as follows:

```
int inverse(int a, int m)
{
    int x,y;
    EE(a,m,x,y);
    if(x<0) x += m;
    return x;
}
```

link

answered **10 Dec '12, 15:55**

nihalb
171●**3**●**6**
accept rate: 0%

[hide preview]

community wiki

Post Your Answer