Modular Inverse from 1 to N

forthright48 on September 29, 2015

We already learned how to find Modular Inverse for a particular number in a previous post, "Modular Multiplicative Inverse". Today we will look into finding Modular Inverse in a bulk.

Problem

Given N and M ($N < M$ and M is prime), find modular inverse of all numbers between 1 to N with respect to M.

Since M is prime and N is less than M, we can be sure that Modular Inverse exists for all numbers. Why? Cause prime numbers are coprime to all numbers less than them.

We will look into two methods. Later one is better than the first one.

$O(NlogM)$ Solution

Using Fermat's little theorem, we can easily find Modular Inverse for a particular number.

$A^{-1} \% M = bigmod(A, M-2, M)$, where $bigmod()$ is a function from the post "Repeated Squaring Method for Modular Exponentiation". The function has complexity of $O(logM)$. Since we are trying to find inverse for all numbers from 1 to N, we can find them in $O(NlogM)$ complexity by running a loop.

```
int inv[SIZE]; ///inv[x] contains value of (x^-1 % m)
for ( int i = 1; i <= n; i++ ) {
    inv[i] = bigmod ( i, m - 2, m );
}
```

But it's possible to do better.

$O(N)$ Solution

This solution is derived using some clever manipulation of Modular Arithmetic.

Suppose we are trying to find the modular inverse for a number a, $a < M$, with respect to M. Now divide M by a. This will be the starting point.

$M = Q \times a + r$, (where Q is the quotient and r is the remainder)
$M = \lfloor \frac{M}{a} \rfloor \times a + (M \% a)$

Now take modulo M on both sides.

$0 \equiv \lfloor \frac{M}{a} \rfloor \times a + (M \% a) \pmod M$
$(M \% a) \equiv -\lfloor \frac{M}{a} \rfloor \times a \pmod M$

Now divide both side by $a \times (M \% a)$.

$$\frac{M \% a}{a \times (M \% a)} \equiv \frac{-\lfloor \frac{M}{a} \rfloor \times a}{a \times (M \% a)} \pmod M$$
$$\therefore a^{-1} \equiv -\lfloor \frac{M}{a} \rfloor \times (M \% a)^{-1} \pmod M$$

The formula establishes a recurrence relation. The formula says that, in order to find the modular inverse of a, we need to find the modular inverse of $b = M \% a$ first.

Since $b = M \% a$, we can say that its value lies between 0 and $a-1$. But, a and M are coprime. So a will never fully divide M. Hence we can ignore the possibility that b will be 0. So possible values of b is between 1 and $a-1$.

Therefore, if we have all modular inverse from 1 to $a-1$ already calculated, then we can find the modular inverse of a in $O(1)$.

Code

We can now formulate our code.

```
1 int inv[SIZE];
2 inv[1] = 1;
3 for ( int i = 2; i <= n; i++ ) {
4     inv[i] = (-(m/i) * inv[m%i] ) % m;
5     inv[i] = inv[i] + m;
6 }
```

In line 2, we set the base case. Modular inverse of 1 is always 1. Then we start calculating inverse from 2 to N. When $i = 2$, all modular inverse from 1 to $i-1 = 1$ is already calculated in array `inv[]`. So we can calculate it in $O(1)$ using the formula above at line 4.

At line 5, we make sure the modular inverse is non-negative.

Next, when $i = 3$, all modular inverse from 1 to $i-1 = 2$ is already calculated. This is process is repeated until we reach N.

Since we calculated each inverse in $O(1)$, the complexity of this code is $O(N)$.

Conclusion

I saw this code first time on CodeChef forum. I didn't know how it worked back then. I added it to my notebook and have been using it since then. Recently, while searching over the net for resources on Pollard Rho's algorithm, I stumbled on an article from Come On Code On which had the explanation. Thanks, fR0DDY, I have been looking for the proof.

Reference

1. forthright48 - Modular Multiplicative Inverse
2. forthright48 - Repeated Squaring Method for Modular Exponentiation
3. Come On Code On - Modular Multiplicative Inverse

Post Views: 219
Category: CPPS, Number Theory

Previous:
Euler Phi Extension and Divisor Sum Theorem

Next:
Chinese Remainder Theorem Part 1 – Coprime Moduli

Your comment...

Login with

Send

Comments: 5 Sort by newest

Add Anycomment to your site

Archives

May 2019 (1)
April 2019 (1)
March 2019 (1)
December 2018 (2)
November 2018 (4)
September 2018 (2)
February 2018 (1)
January 2018 (1)
November 2017 (2)
September 2015 (7)
August 2015 (13)
July 2015 (15)

Categories

CPPS (45)
- Combinatorics (4)
- Data Structure (1)
- Number Theory (36)

Meta (1)
Misc (4)

Recent Comments

My Shopee Interview – Shadman Protik on My Interview Experience with Shopee / Garena / Sea Group

Istiad Hossain Akib on SPOJ LCMSUM – LCM Sum

Rifat Chowdhury on MyStoryit#02 – Deciding Where to Study CS

Salman Farsi on Leading Digits of Factorial

Learning Notes on Multiplicative Functions, Dirichlet Convolution, Mobius Inversion, etc – RobeZH's Blog on SPOJ LCMSUM – LCM Sum

© 2015-2019 Mohammad Samiul Islam