P THINK FAST S

Competitive Programming

From Problem 2 Solution in O(1)

Number Theory

Modular multiplicative inverse

Mostafa Saad Ibrahim
PhD Student @ Simon Fraser University

Recall

- Mod distributed smoothly over +, -, *
  - (a + b * c) % n = (a%n + (b%n * c%n)%n) % n
- Multiplicative inverse (reciprocal)
  - Of number a: 1/a or $\mathbf{a}^{-1}$ => then a * (1/a) = 1
  - Then for any a * b = 1, then b = 1 / a = a^{-1}
  - And **a / x** => $\mathbf{a} * \mathbf{x}^{-1}$
- Congruence: a ≡ x (% m) => a - x = qm
  - ax ≡ 1 (% m) ?
- what about **(a / x) %n** ? Should equal a * *Multiplicative inverse of x* considering n

Modular multiplicative inverse

- ax ≡ 1 (% m)
  - Which means ax % m = 1 % m
  - m = 11, a = 8, x = 7 => 8 * 7 = 1 (mod 11)
- Then, a is multiplicative inverse of x for % m
- Also **a = 1 / x** (mod m)
- **Exists IFF gcd(a, m) = 1**
- (119 / 7) % 11 => 17 % 11 => 6
  - Recall 8 * 7 = 1 (mod 11) … then **1 / 7 == 8** %11
- (119 * 8) % 11 = (119%11 * 8) % 11 = 6

Solution 1: Extended Euclidean

- $ax \equiv 1$ (% m)
- Then (ax-1) % m = 0, then ax-1 = qm
  - m = 11, a = 8, x = 7 => 8 * 7 = 1 (mod 11)
  - 56 - 1 = 5 * 11
- Rearrange: ax + m(-q) = 1
- This is similar to ax + my = gcd(a, m) = 1
- That is, the solution to extended (a, m) giving that gcd(a, m) = 1
- So just 1 call to extended, x is the answer

Solution 1: Extended Euclidean

- a = 17, m = 43
  - -5 * 17 + 2 * 43 = 1
  - then (1 / 17) % 43 = -5 = 38
- a = 43, m = 17
  - 2 * 43 - 5 * 17 = 1
  - then (1 / 43) % 17 = 2
  - E.g. (559 / 43) % 17 = 13 % 17 = 13
  - Same: (559 * 2) % 17 = 13

Solution 1: Extended Euclidean

```
// ax ==1 %m  IFF a, m coprimes
// return -1 means NO answer
// handle case x may be -ve
ll modInversek(ll a, ll m) {
    ll x, y;

    ll d = extended_euclid(a, m, x, y);

    if(d == 1)
        return -1;

    return (x + m) % m;
}
```

Solution 2: Euler's theorem

- if gcd(a, m) = 1 => $a^{\varphi(m)} \equiv 1 \pmod{m}$
  - φ(m) is Euler's totient function
- As a result (divide both sides by a)
  - $a^{\varphi(m)-1} \equiv a^{-1} \pmod{m}.$
  - $a^{-1} \equiv a^{m-2} \pmod{m}.$ if m is prime
- Computations amount in GCD vs Euler?
- In addition, the theorem can be used to help reducing **large powers** evaluations

Solution 2: Euler's theorem

```
// (a^k) % m
ll pow(ll a, ll k, ll M) {
    if (k == 0)
        return 1;
    ll r = pow(a, k / 2, M);
    r = (r * r) % M;
    if (k % 2)
        r = (r * a) % M;
    return r;
}

//ax ==1 %p  IFF p primes
ll modInversep(ll a, ll p) {
    return pow(a, p-2, p);
}

//ax ==1 %m  IFF a, m coprimes
ll modInverse(ll a, ll m) { //IFF a, m coprimes
    return pow(a, phi(m) - 1, m);
}
```

Modinverse range for prime

- Given P, compute all mod inv for range 1 - (p-1)
- p % i = p - (p / i) * i   => % equation
  - (p%i) % p = p%i
  - p%p = 0
- p % i = -(p / i) * i (mod p)   => % P
- Now, divide by i * (p % i)
- 1 / i = - (p / i) * 1 / (p % i)   % p
- inv[i] = - (p / i) * inv[p % i]   % p
- Add +p to convert to +ve
- inv[i] = p - (p / i) * inv[p % i]   % p

Modinverse range for prime

```
// p % i = p - (p / i) * i              : Mod Equ
// p % i = -(p / i) * i (mod p)         : %p 2 sides
// inv[i] = - (p / i) * inv[p % i]  : / i * (p % i)
vector<int> ModInvRange(int p)
{
    vector<int> inv(p-1, 1);

    for (int i = 2; i < p; ++i)
        inv[i] = (p - (p/i) * inv[p%i] % p) % p;

    return inv;
}
```

Euler's theorem and large powers

- $\mathbf{a^{\varphi(m)} = 1}$ and $a^{\varphi(m)-1} = a^{-1}$ if $\gcd(a, m) = 1$
- $\mathbf{a^{p-1} = 1}$ and $a^{p-2} = a^{-1}$ if p is prime number
- 7^{222} % 10.
  - $\gcd(7, 10) = 1$ and $\varphi(10) = 4$
  - From Euler's theorem $7^4 \equiv 1$ (% 10)
- $7^{222} \equiv 7^{4 \times 55 + 2} \equiv (7^4)^{55} \times 7^2 \equiv 1^{55} \times 7^2$
- $7^{222} \equiv 49 \equiv 9 \pmod{10}$
- Or shortly, $7^{222} \equiv 7^{222\%4} \equiv 7^{2} = 9 \pmod{10}$

Euler's theorem and large powers

- Compute $(1/a^m)\%\ p$ .. where p is prime
- Same as $((1/a)\%\ p)^m\ \%\ p$
- $(a^{p-2}\ \%\ p)^m\ \%\ p$ use inverse modular
- $a^{m(p-2)}\ \%\ p$
- What about using euler to **reduce** the power?
- $a^{(m(p-2))\ \%(p-1)}\ \%\ p$ or $a^{(m\%(p-1)\ *\ (p-2)\%(p-1)\ )\%(p-1)}\ \%\ p$
- Simil

```
ll modInverse_am(ll a, ll m, ll p) {
    //return pow(a, (m * (p - 2))%(p-1), p);
    return pow(a, (m%(p-1) * (p - 2)%(p-1))%(p-1), p);
}
```

Euler's theorem and large powers

- Let’s learn one more trick for previous issue
- (p-2) % (p-1) = **-1** [use -ve mode]
- It now turns to be: $a^{-m\%(p-1)}$ % p … recall:
  - if m is +ve, its mode: m%a
  - if m is +ve, then -m is: (a + (-m)%a) % a
  - Or more directly a - m%a
- Then turns to be: $a^{p-1-(m\%(p-1))}$ % p
- Moral of that, is we get rid of p-2 with a constant -1 .. this helps in some advanced problems

Euler's theorem and large powers

- What about $a^x \% n$ where $gcd(a, n) > 1$?
- Let's factorize a to p1 * p2 * p3...pk
  - e.g. 12 = 2 * 2 * 3 (p1 = p2 = 2)
  - Then answer = $(p1^x \% n * p2^x \% n....)\%n$
- Our problem = new sub-problems: $p^x \% n$
  - p is a prime number
  - if gcd(p, n) = 1, direct euler...otherwise m % p = 0
- Find largest g such that: $p^g \% n = 0$
  - Then $gcd(p, t = n/p^g) = 1$ ... using euler rule
  - $p^{\varphi(t)} = 1$ (%t) multiply all terms by p^g
  - $p^g p^{\varphi(t)} = p^g$ (%n) and generally: $p^g p^{k\varphi(t)} = p^g$ (%n)

Euler's theorem and large powers

- Now: $p^g p^{k\varphi(t)} = p^g$ (%n)
  - means multiple of has $p^{\varphi(t)}$ no effect
- Back to p^x
  - if x <= g, then it was actually small power. Forget euler
  - if x > g, let's embed it in equation: x = x - g + g
  - $p^x = p^g p^{x-g}$ …. using modified euler
  - p^x (%n) = $p^g p^{(x-g)\ \%\varphi(t)}\ p^{k(x-g)}$(%n) [recall: d = qk + r]
  - p^x (%n) = $p^g p^{(x-g)\ \%\varphi(t)}$(%n)
- Your turn: use above to compute: (8^2^6^4^2^5^8^9) % 10000
  - Hint think: 8^x % 10000 … use recursion for the tower

Finally

- In many problems it asks your for solution % prime (e.g. 10^9+7). They select it prime to allow some euler/inverse solutions
- Readings
  - Euler (link 1, link 2)
  - Fermat's little theorem
  - Carmichael function
  - Discrete logarithm problem (Practice problem)

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً

Problems

- UVA (11440, 11174), UVA (10692), LiveArchive (3343 - Last Digits)