Powerful tricks with modulo calculation - Boris Sokolov

Trick #1:

`(A / B) % MOD = (A % (MOD * B)) / B`

Conditions: none.

Advices: use this trick only if B can be not coprime with MOD, because new modulus = MOD * B can be large. How to avoid overflow working with large modulus read at the trick #5.

Trick #2:

$(A\ /\ B)\ \%\ MOD = ((A\ \%\ MOD) * (B^{phi(MOD) - 1}\ \%\ MOD))\ \%\ MOD$, where phi is Euler's totient function

Conditions: B and MOD are coprimes.

Proof: $(A\ /\ B)\ \%\ MOD = ((A\ \%\ MOD) * (B^{-1}\ \%\ MOD))\ \%\ MOD$ from Exponentiation properties. And from Euler's theorem follows that $B^{phi(MOD)}\ \%\ MOD = 1$. Let's multiply this equation by B^{-1} and we will get that $B^{-1}\ \%\ MOD = B^{phi(MOD) - 1}\ \%\ MOD$.

Trick #3:

$(A\ /\ B)\ \%\ MOD = ((A\ \%\ MOD) * (B^{MOD - 2}\ \%\ MOD))\ \%\ MOD$

Conditions: B and MOD are coprimes, MOD is a prime number.

Advices: if you're sure that MOD is prime, better use this trick instead of trick #2. Remember that 10^9+7 and 10^9+9 are prime numbers.

Proof: if MOD is prime then phi(MOD) = MOD - 1 from properties of Euler's totient function. As it's just a particular case of trick #2, the rest of proof is similar.

Trick #4:

$A^N\ \%\ MOD = A^{N\ \%\ phi(MOD)}\ \%\ MOD$

Conditions: A and MOD are coprimes.

Advices: use this trick only if N can't be present in any standart data type, otherwise use Fast exponentiation.

Proof: from Euler's theorem follows that $A^{phi(MOD)}\ \%\ MOD = 1$. It's easy to see that $A^{X * phi(MOD)}\ \%\ MOD = 1$ too. So if $N = X * phi(MOD) + Y$ then $A^N\ \%\ MOD = A^Y\ \%\ MOD$ and minimal such $Y = N\ \%\ phi(MOD)$.

Trick #5:

`(A * B) % MOD where MOD can't be present in int data type`

```
function mulmod(A, B, MOD) {
    RES = 0;
    while (B > 0) {
        if (B is odd) {
            RES = (RES + A) %
MOD;
        }
        A = (A * 2) % MOD;
        B = B / 2;
    }
    return RES;
}
```

Conditions: 2 * MOD can be present in a standart data type.

Advices: use this trick only if (A % MOD) * (B % MOD) can't be present in any standart data type because of overflow and you don't want to use BigIntegers. But keep in mind that it works in O(logB) operations, not in O(1) as `(A % MOD) * (B % MOD)`.

Proof: if B is even then A * B = 2 * A * (B / 2), otherwise A * B = A + A * (B - 1).

F.A.Q. (in PM) - Alex Danilyuk (CF handle: Um_nik)

Q1: I'm a newbie. What should I do to become great coder?
A1: ~~Stop doing competitive programming~~ Solve problems.
Q2: I'm doing CP for two months and I'm still not red green. What should I do?
A2: ~~You are lazy and impatient~~ Solve more problems.
Q3: You became a red in less than two years, it is unbelievable!
A3: No, it isn't. ~~You can do it too if you will solve f***ing problems.~~
Q4: You became a red blah-blah-blah such a huge fan blah-blah. Oh, and what should I do to become as great as you?
A4: ... right. You already know the answer. Solve problems. ~~I hate you.~~
Q5: I'm not good at DP [or something]. What can you suggest?
A5: Maybe you should try to ~~stop asking stupid questions and~~ solve some problems on DP? Or read some blogs and editorials.
Q6: I can't solve a problem / understand your code. Can you help me?
(Well, it is not a bad question in general. It is a good (if you really want me to explain something not to write the solution instead of you) question. But...)
A6: Sure. Can you provide the link to the problem / code?
(I'm not joking, it is a real story).
Bonus!
Q0: Hello bro/sir.
A0: Stop doing this please.