SCT Modular Math

Andrew Wang

February 2021

1 Introduction

Modular Arithmetic works with remainders instead of integers. For example,

$$9 \bmod 5 = 4$$

since the remainder of 9 divided by 5 is 4. In contests, you'll often see modular arithmetic used to avoid dealing with large numbers that overflow.

1.1 Modular Arithmetic Properties

Some common properties in modular arithmetic:

$$(a + b) \bmod m = (a \bmod m + b \bmod m) \bmod m$$

$$(a - b) \bmod m = (a \bmod m - b \bmod m) \bmod m$$

$$(a \cdot b) \pmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$$

$$a^b \bmod m = (a \bmod m)^b \bmod m$$

2 Optimizations

Taking the modulo of a number over and over again can be very time-costly due to the high constant factor.

2.1 Pre-Calculation

If we're taking the modulo of some set of numbers over and over again(i.e powers of numbers), it may be faster to pre-calculate the modulo of each of these numbers beforehand.

2.2 Addition and Subtraction

Since the % operator has a much higher constant factor compared to more elementary operators like addition or subtraction, always use addition and subtraction when you have the choice. For example,

$$9 \bmod 5 = 9 - 5.$$

3 Tips & Tricks

- Take the modulo of each number before performing operations to prevent overflow.
- While debugging, if you come across a negative number it almost always means overflow.

4 Euler's Totient Theorem

Euler's Totient Function is commonly seen in number theory. It states:

$$a^{\varphi(n)} = 1 \bmod n.$$

4.1 Fermat's Little Theorem

Fermat's Little Theorem is an extensions of Euler's Totient where n is a prime number. This simplifies to:

$$a^{p-1} = 1 \bmod p$$

where p is any prime.

4.2 Modular Inverse

Dividing can be very difficult while in some mod n. For example,

$$(9/3) \bmod 5 \neq ((9 \bmod 5)/(3 \bmod 5)) \bmod 5$$

Luckily, using modular inverses, we can safely divide numbers without worrying about mistakes during division. The modular inverse of a number is equivalent to the reciprocal but in a certain mod.

$$a/b \bmod m = a * i \bmod m$$

where i is the modular inverse of b. The modular inverse of $b \bmod m$, is equivalent to $b^{m-1} \bmod m$ since by Fermat's:

$$b^{m-1} \bmod m \equiv 1 \bmod m$$

if m is prime. Finding b^{m-1} is a much simpler task which can be solved using binary exponentiation or pre-calculating the powers of a number as mentioned before.

4.3 Example Problem: Binomial Coefficients

1. Pre-calculate factorials in an array
2. $\frac{a!}{b!(a-b)!} \bmod m = a! \cdot inverse(b!) \cdot inverse((a-b)!) \bmod m$
3. calculate the inverses using binary exponentation

5 Miscellaneous

Sometimes, problem-setters mess up!

5.1 Challenge Problem: USACO Gold walk

Although the intended solution is an $O(N^2)$ MST, this can be easily fake-solved in $O(1)$ using some modular arithmetic. The general idea is to maximize the function:

$$\min_{x,y \text{ in different groups}} (2019201997 - 84x - 48y).$$

This can be done by placing the smallest $k-1$ in their own distinct group and the other larger numbers together in one group. Simplifying gives us the equation:

$$2019201997 - 84(k-1) - 48n$$

(For a full proof check out the full solution)

6 Resources

6.1 References

- USACO Guide Modular Arithmetic
- CPH Modular Math

6.2 Problens

- USACO Guide Problem set (At the end of the page)