CS30 (Discrete Math in CS), Summer 2021 : Lecture 26

Topic: Numbers: Modular Arithmetic

Disclaimer: These notes have not gone through scrutiny and in all probability contain errors. Please discuss in Piazza/email errors to deeparnab@dartmouth.edu

---

- **Definition.** Given any integer $n > 0$ and another integer a (not necessarily positive), the **division theorem**[1] states that there are *unique* integers q, r such that $a = qn + r$ with $0 \le r < n$. The number r is denoted as $a \bmod n$.

- **Examples.** For example, $17 \bmod 3$ is 2. This is because $17 = 3 \times 5 + 2$. Similarly, $13 \bmod 5 = 3$.

  Slightly more interestingly, $(-1) \bmod 3 = 2$. This is because $-1 = 3 \times (-1) + 2$. Similarly, $(-7) \bmod 5 = 3$ since $-7 = 5 \times (-2) + 3$.

- **The Ring of Integers modulo n.**

  Fix a positive natural number n. The way to think about the $\bmod n$ operation is as a function which takes ***integers*** to the set $\{0, 1, 2, \ldots, n-1\}$ of possible remainders. There is a name for this set of n remainders; it is called the *ring* of integers modulo n and is denoted by $\mathbb{Z}_n$.

  $$\bmod n : \mathbb{Z} \to \mathbb{Z}_n \qquad a \mapsto a \bmod n$$

  Why ring? Well just consider how the numbers map. 0 maps to 0, 1 maps to 1, and so on til $(n-1)$ maps to $(n-1)$. But then n maps to 0, it "rings" around to 0, and the process starts again. $(n+1)$ maps to 1 and so on. It also rings the same way for negative numbers. 1 maps to 1, 0 maps to 0, -1 maps to $n-1$, -2 maps to $n-2$, and so on.

- **An Important Notation.**

  The function $\bmod n$ is clearly not injective. Indeed, any two numbers which map to the same element are called ***equivalent*** modulo n.

  Given two integers a, b, we use the notation

  $$a \equiv_n b$$

  to denote the condition that $a \bmod n = b \bmod n$.

- **Important Properties.** The following simple but important properties are crucial to be comfortable with this new "kind" of math. I would recommend trying to actually prove the facts by yourself and then peeking at the solution.

  1. (Equivalence under addition of multiple of n.) For any natural number n and integers a and b, $a \equiv_n (a + bn)$.

     Suppose $a \bmod n = r$, that is, $a = qn + r$. Then, $a + bn = qn + r + bn = (q + b)n + r$. Thus, $(a + bn) \bmod n = r$ as well.

[1] The division theorem may sound "obvious" to you, for this is probably something you have seen from grade school, but it requires a proof. Why should there be a quotient-remainder pair? And why unique? A UGP from the past explored this.

2. (Transitivity) If $a \equiv_n b$ and $c \equiv_n b$, then $a \equiv_n c$.

   $a \equiv_n b$ implies there is some remainder $0 \le r < n$ and quotients $q_1, q_2 \in \mathbb{Z}$ such that $a = q_1 n + r$ and $b = q_2 n + r$. $c \equiv_n b$ implies there is some q_3 such that $c = q_3 n + r$. Thus, $a \bmod n = r = c \bmod n$ implying $a \equiv_n c$.

3. (Addition OK) Show that if $a \equiv_n b$ and $c \equiv_n d$, then $(a + c) \equiv_n (b + d)$.

   $a \equiv_n b$ means there is some remainder $0 \le r < n$ and quotients $q_1, q_2 \in \mathbb{Z}$ such that $a = q_1 n + r$ and $b = q_2 n + r$.

   Similarly, there is some remainder $0 \le s < n$ and quotients $p_1, p_2 \in \mathbb{Z}$ such that $c = p_1 n + s$ and $d = p_2 n + s$.

   Thus, $(a + c) = (q_1 + p_1)n + (r + s)$ implying $(a + c) \equiv_n (r + s)$ by equivalence under adding a multiple of n. Similarly, $(b + d) = (q_2 + p_2)n + (r + s)$ implying $(b + d) \equiv_n (r + s)$. Transitivity implies $(a + c) \equiv_n (b + d)$.

4. (Multiplication OK) Show that if $a \equiv_n b$ and $c \equiv_n d$, then $(a \cdot c) \equiv_n (b \cdot d)$.

   $a \equiv_n b$ means there is some remainder $0 \le r < n$ and quotients $q_1, q_2 \in \mathbb{Z}$ such that $a = q_1 n + r$ and $b = q_2 n + r$.

   Similarly, there is some remainder $0 \le s < n$ and quotients $p_1, p_2 \in \mathbb{Z}$ such that $c = p_1 n + s$ and $d = p_2 n + s$.

   Thus,

   $$(a \cdot c) = (q_1 n + r) \cdot (p_1 n + s) = (q_1 p_1 n^2 + q_1 n s + p_1 n r + rs) = (q_1 p_1 n + q_1 s + p_1 r)n + rs$$

   and,

   $$(b \cdot d) = (q_2 n + r) \cdot (p_2 n + s) = (q_2 p_2 n^2 + q_2 n s + p_2 n r + rs) = (q_2 p_2 n + q_2 s + p_2 r)n + rs$$

   Therefore, $(a \cdot c) \equiv_n (r \cdot s)$ by equivalence under adding a multiple of n, and so is $(b \cdot d) \equiv_n (r \cdot s)$. Transitivity implies $(a \cdot c) \equiv_n (b \cdot d)$.

5. (Powering with a positive integer OK) Let k be a positive natural number. If $a \equiv_n b$, then $a^k \equiv_n b^k$.

   Apply the above k times. More precisely, $a \equiv_n b$ and $a \equiv_n b$ implies $(a \cdot a) \equiv_n (b \cdot b)$, that is $a^2 \equiv_n b^2$. One proceeds inductively. If we already have shown $a^{k-1} \equiv_n b^{k-1}$, then along with the fact $a \equiv_n b$, we get $(a^{k-1} \cdot a) \equiv_n (b^{k-1} \cdot b)$, that is, $a^k \equiv_n b^k$.

6. (Division usually **not** OK) Show an example of numbers a, b, c, n where $(a \cdot b) \equiv_n (c \cdot b)$ but $a \not\equiv_n c$.

   Let me show ***how*** *I would come up with such an example before telling you the example. If $(ab) \equiv_n (cb)$, we know that $(ab - cb) \equiv_n 0$, that is $(a - c) \cdot b \equiv_n 0$, or n divides $(a - c)b$. And we want an example where $a \not\equiv_n c$ that is n doesn't divide $(a - c)$.*

   Well, if n divides $(a - c)b$ but not $(a - c)$, one simple example would be when $n = b$ and say $a - c = 1$. This leads us to the example $n = 5$, $b = 5$, $a = 2$, $c = 1$. One can check — $(2 \cdot 5) \equiv_5 (1 \cdot 5)$ but $2 \not\equiv_5 1$.

   One may then think – hey, if $b < n$ would this be true. Even in this case, the answer is NO. To see this, again, we want n to divide $(a - c)b$ but n should not divide $(a - c)$. So b could be a factor *of n, and n/b is what divides $(a - c)$ (but not n).*

For instance, $n = 6 = 2 \cdot 3$*,* $b = 3$*,* $a = 7$ *and* $c = 5$ *suffices. Let's check, Is* $21 \equiv_6 15$*? Yes, both give remainder* 3 *when divided by* 6*. Is* $7 \equiv_6 5$*? No,* $7 \bmod 6 = 1$ *which* $5 \bmod 6 = 5$.

Later on, we will see one case when division will be OK. You can perhaps guess (yes, when b *and* n *are relatively prime).*

7. (Taking "roots" **not** OK) Show an example of numbers a, b, n and k, such that $a^k \equiv_n b^k$, but $a \not\equiv_n b$. In fact, show different examples for $k = 2$ and $k = 3$.

   Once again, the method is more important than the specific example.

   Let's start with $k = 2$*.* $a^2 \equiv_n b^2$ *means* $a^2 - b^2 \equiv_n 0$*. That is,* $(a-b)(a+b) \equiv_n 0$*. So, if* n *divides the product of* $(a-b)$ *and* $(a+b)$*. We also want* $a \not\equiv_n b$*, that is, we want* $(a-b) \not\equiv_n 0$*. We want* n not *to divide* $(a-b)$*.*

   Well, if n *divides* $(a-b)(a+b)$ *but not* $(a-b)$*, one simple example would be when* $n = a+b$ *and say* $a - b = 1$*. This leads us to the example* $n = 5$*,* $a = 3$*,* $b = 2$*.*

   Let's check: $3^2 \equiv_5 2^2$ *— yes,* 9 *divided by* 5 *is* 4 *which is* 2^2*. Is* $3 \equiv_5 2$*? Of course not. There's our counterexample. Do you want to do the* $k = 3$ *case on your own? Here's a hint:* $a^3 - b^3 = (a-b)(a^2 + ab + b^2)$*.*

- **Modular Exponentiation Algorithm**

  Suppose we want to figure out what is the remainder when we divide 3^{10} by 7, that is, what is $3^{10} (\bmod 7)$? The hard and often infeasible way would be to compute 3^{10} and then divide by 7 to get the remainder. The above operations allow a much faster way to compute this. Let's first do an example and then give the whole algorithm.

$$
\begin{aligned}
3^{10} \bmod 7 &= (3^2)^5 \bmod 7 \\
&= 9^5 \bmod 7 \\
&= (9 \bmod 7)^5 \bmod 7 && \text{Operation (c) above} \\
&= 2^5 \bmod 7 && \text{Progress! From } 3^{10} \text{ we have moved to } 2^5. \\
&= (2 \cdot 2^4) \bmod 7 && \text{Can't halve 5 as it is odd.} \\
&= \big((2 \bmod 7) \cdot (2^4 \bmod 7)\big) \bmod 7 && \text{We have again halved the exponent by moving to } 2^2 = 4. \\
&= (2 \cdot (4^2 \bmod 7)) \bmod 7 \\
&= 4
\end{aligned}
$$

We get 4 when we divide 3^{10} by 7. The general idea was to keep on reducing the exponent by half by moving to the square, and then replacing the square to a possibly smaller number by taking the mod "inside". The full recursive algorithm is shown below.

procedure ModExp(a, b, n) ▷ Assumes b, n are positive integers.
  ▷ *Returns* $a^b \bmod n$.
  $a \leftarrow a \bmod n$ ▷ We first move a to $a \bmod n$. Always get inside the ring.
  if $b = 1$ **then**:
    return $a \bmod n$. ▷ Nothing to do – base case.
  if b is even **then**:
    return ModExp($a^2, \frac{b}{2}, n$)
  else
    $s =$ ModExp $(a, (b-1), n)$ ▷ $b - 1$ is even.
    ▷ $s = a^{b-1} \bmod n$.
    return $(a \cdot s) \bmod n$.

Remark: *The first line ensures* $a \in \{0, 1, \ldots, n-1\}$. *Note that we compute the mod of* $(a \cdot s) \bmod n$. *The number* $a \cdot s$ *is at most* n^2. *Thus, to compute* $a^b \bmod n$ *one only needs to be "divide" numbers as big as* n^2 *by* n. *Thus* n *is a one or small two-digit number, this all can be done by hand.*

Exercise: *Implement the algorithm up in your favorite language.*