A Quick Introduction to Modular Arithmetic

Art Duval
University of Texas at El Paso

November 16, 2004

1 Idea

Here are a few quick motivations for modular arithmetic:

1.1 Sorting integers

Recall how you sort all integers into "odd" and "even". Every number is either odd or even, but not both. This is a "partition" of the integers into two "classes". One way to think of this partition is that we are sorting numbers based on whether or not they are divisible by 2.

If we replace the 2 in the odd/even definition by, say, 3, we could sort numbers based on whether or not they are divisible by 3. It turns out to be better (you'll see why soon, I hope) to sort an integer based on which remainder it leaves when it's divided by 3. In this settting, we think of even numbers as those whose remainder is 0 when divided by 2, and odd numbers as those whose remainder is 1 when divided by 2. And then, when we replace 2 by 3, we'd be sorting the integers into 3 classes, those whose remainder is 0 when divided by 3, those whose remainder is 1 when divided by 3, and those whose remainder is 2 when divided by 3.

From now on, we'll call the number we're dividing by the **modulus**, and denote it by m. So, in the odd and even case, $m = 2$, and the next case we talked about, $m = 3$. We can set m to be any positive integer. (If $m = 1$, something funny happens. Try it!)

When $m = 2$, the integers are sorted into 2 parts, $\{\ldots, -4, -2, 0, 2, 4, 6, 8, \ldots\}$ and $\{\ldots, -3, -1, 1, 3, 5, 7, \ldots\}$. (Note that negative integers are included as well.) When $m = 3$, the integers are sorted into 3 parts, $\{\ldots, -6, -3, 0, 3, 6, 9, 12, \ldots\}$, $\{\ldots, -5, -2, 1, 4, 7, 10, 13, \ldots\}$, $\{\ldots, -4, -1, 2, 5, 8, 11, 14, \ldots\}$.

1.2 Remainders

Closely related to the above idea is the idea of assigning to every integer its remainder when its divided by m. So, for instance, when $m = 5$, we'd assign 17 to 2, since 17 leaves a remainder of 2 when divided by $m = 5$. When $m = 2$, every odd number would be assigned to 1, and every even number would be assigned to 0.

What's the difference between sorting and assigning by remainders? It seems like the same thing, and they are very closely related. When we sort by remainders, we think of all the integers in the same class as being related to one another when they have the same remainder. When we assign, we think of a function assigning to every integer its remainder. These two different perspectives will come up again.

1.3 Last digit

A special case of assigning or sorting by remainder when dividing by m is when $m = 10$. Then, the remainder when dividing a non-negative integer by $m = 10$ is simply its last digit!

1.4 Clock arithmetic

A quick example looking ahead to a simple use of modular arithmetic. When it's 11 o'clock, and you want to know what time it will be 7 hours later, you don't simply add 7 to 11 to get 18 o'clock. We do start with the 18, but then we subtract 12. More generally, if you wanted to know what time it will be 70 hours later, you'd add 70 to 11, get 81, and keep subtracting 12's (six times, as it turns out) until you are left with 9, so it will be 9 o'clock (some days later). In modular arithmetic, using notation we'll get to soon, you are computing $11 + 70 \equiv 9 \pmod{12}$.

Note that here, we are using the function idea of modular arithmetic. Also note that if you are computing on military time, just replace all the 12's by 24's.

2 Definitions

Now let's take some of these ideas and make them more precise.

2.1 Sorting; equivalence relation

The idea is that we want to say that a and b are "equivalent" when they leave the same remainder upon division by m. Say this remainder is r. Then

$$\begin{aligned} a &= ms + r \\ b &= mt + r \end{aligned}$$

for some integers s and t. Subtracting the second equation from the first, we get $a - b = m(s - t)$, which leads to what turns out to be a useful form of the definition of this equivalence:

$$a \equiv b \pmod{m} \text{ when } a - b \text{ is a multiple of } m.$$

(The advantage of this form is that it only involves a, b, m, and does not need to mention r.) We say "a is **congruent** to b mod m". We call $\equiv$ an equivalence relation because it satisfies the following three rules:

- $a \equiv a \pmod{m}$
- if $a \equiv b \pmod{m}$, then $b \equiv a \pmod{m}$
- if $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$, then $a \equiv c \pmod{m}$.

2.2 Remainders; binary operation

Most computer programs are like Mathematica in using "mod" as a function, not a relation:

```
 In[1] :=  Mod[81, 12]
Out[1] =   9
```

The output is "9" because when 81 is divided by 12, the remainder is 9. Note that this means $81 \equiv 9 \pmod{12}$. The difference is that, while 81 is congruent to many numbers $\pmod{12}$, the `Mod` function returns only the special number, 9, from this class that is the unique remainder when you divide by 12.

3 Modular arithmetic

What makes these ideas valuable is how congruence behaves nicely with respect to addition, subtraction, and multiplication (division is a little harder, and beyond the scope of these notes). In short, if

$$\begin{aligned} a &\equiv b \pmod{m} \\ c &\equiv d \pmod{m} \end{aligned}$$

then, as we'd hope,

$$\begin{aligned} a + c &\equiv b + d \pmod{m} \\ a - c &\equiv b - d \pmod{m} \\ a \times c &\equiv b \times d \pmod{m} \end{aligned}$$

Note how, in the special case $m = 10$, this just confirms last-digit arithmetic. For instance, $17 \equiv 7 \pmod{10}$ and $23 \equiv 3 \pmod{10}$, so $17 \times 23 \equiv 7 \times 3 = 21 \equiv 1 \pmod{10}$, which is just a fancy way of saying that the last digit of 17×23 is 1 because the last digit of 7×3 is 1.

We now sketch the details of why these arithmetic facts are true.

3.1 Addition

Since $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, we know that $a - b$ and $c - d$ are multiples of m, so $a - b = ms$ and $c - d = mt$ for some integers s and t. Then

$$\begin{aligned} (a + c) - (b + d) &= (a - b) + (c - d) \\ &= ms + mt \\ &= m(s + t), \end{aligned}$$

so $a + c \equiv b + d \pmod{m}$, since $(a + c) - (b + d)$ is a multiple of m.

3.2 Subtraction

This is entirely similar to addition, and so the details are left to you to work out.

3.3 Multiplication

Since $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, we know that $a - b$ and $c - d$ are multiples of m, so $a - b = ms$ and $c - d = mt$ for some integers s and t. We can rewrite these two equations as $a = ms + b$ and $c = mt + d$. Then

$$\begin{aligned} ac - bd &= (ms + b)(mt + d) - bd \\ &= (m^2st + dms + bmt + bd) - bd \\ &= m(mst + ds + bt) + bd - bd \\ &= m(mst + ds + bt), \end{aligned}$$

so $ac \equiv bd \pmod{m}$, since $ac - bd$ is a multiple of m.