1 Extended Euclidean Algorithm

1.1 Iterative Approach

Suppose we're trying to find $11^{-1} \bmod 31$. Each color here is a different number we're focusing on; the black numbers are ones that we don't really care too much about—they're just intermediate values. Executing the forward Euclidean algorithm, we have

$$\begin{aligned} 31 &= 11(2) + 9 \\ 11 &= 9(1) + 2 \\ 9 &= 2(4) + \boxed{1} \\ 2 &= 1(2) + 0 \end{aligned}$$

Since we've reached a remainder of 0, the previous remainder is our gcd: 1.

Next, we want to find coefficients a and b such that $11a + 31b = 1$.

This way, when we take the equation in mod 31 space, the $31b$ term gets eliminated (as it's equivalent to 0), leaving us with just $11a \equiv 1 \pmod{31}$, i.e. $11^{-1} \equiv a \pmod{31}$.

If we rearrange all of the above equations (Why? It'll make sense in a bit.), we have

$$\begin{aligned} 9 &= 31 - 11(2) \\ 2 &= 11 - 9(1) \\ 1 &= 9 - 2(4) \end{aligned}$$

Remember that we want to get something of the form $1 = 11a + 31b$. If we look at these new rearranged equations, we have an equation for 1!

$$1 = 9 - 2(4).$$

Now, notice that we also have an equation for 2, immediately above it. (Yes, we also have one for 9, but we want to go in order.) This means that we can plug it in:

$$\begin{aligned} 1 &= 9 - 2(4) \\ &= 9 - (11 - 9(1))(4) \\ &= 9 - 11(4) + 9(4) \\ &= 9(5) - 11(4) \end{aligned}$$

Why did we simplify some expressions and not others? Notice how we kept all of the colored numbers intact—these are the numbers that were used in the forward Euclidean algorithm, and are the numbers that we're actually interested in. Everything else is just along for the ride.

Again, notice that we also have an equation for 9 that we can plug in:

$$\begin{aligned} 1 &= 9(5) - 11(4) \\ &= (31 - 11(2))(5) - 11(4) \\ &= 31(5) - 11(10) - 11(4) \\ &= 31(5) - 11(14) \end{aligned}$$

Oh hey—we've just arrived at our desired equation! This means that we have $1 = 11(-14) + 31(5)$, and

$$11^{-1} \equiv -14 \equiv \boxed{17} \pmod{31}.$$

In general, this approach has two parts. In the first part, we use the Euclidean algorithm to get the gcd, along with some equations that are used in the second part. The second part is where we backtrack from the gcd back up to our original numbers, getting a form of $ax + by = \gcd(x, y)$, by plugging in successive equations, working our way up.

1.2 Tabular Approach

The last approach is arguably quite convoluted, and can lead to a lot of careless mistakes if you're not careful on which numbers are important. The way I personally like to approach these problems is with a tabular approach.

Suppose we're trying to find 11^{-1} mod 31 just like in the last section. In the tabular approach, we make use of the fact that $\gcd(x, y) = \gcd(y, x \bmod y)$. We can apply this successively, giving the following rules:

$$\begin{aligned} x' &= y \\ y' &= x \bmod y \end{aligned}$$

Here, x' and y' represent the values of x and y in the next iteration. If we put this into a table, each iteration we'll be looking at the row above to get x and y, and enter the values of x' and y' into the row below. We'll also keep track of $\lfloor x/y \rfloor$ in the process (why we do this will come clear later). We have the following:

x	y	$\lfloor x/y \rfloor$
31	11	2
11	9	1
9	2	4
2	1	2
$\boxed{1}$	0	

Since we've reached a 0, this means our x value is the gcd: 1.

Now, the second part has us backtrack from this final gcd back up, to get an equation of the form $31a + 11(b) = 1$, much like in the previous section. Here, we'll be using the following rules while backtracking from the bottom up:

$$\begin{aligned} a' &= b \\ b' &= a - \left\lfloor \frac{x}{y} \right\rfloor \cdot b \end{aligned}$$

Here, a' and b' represent the values of a and b in the next iteration. We can add two more columns to this table, and each iteration we'll be looking at the row below for values of a and b, and enter in values of a' and b' into the row above. We get the value of $\lfloor x/y \rfloor$ from the row above as well.

Each row represents values of x, y, a, and b such that $ax + by = \gcd(x, y)$. We always start this second part with these last two rows of the table:

x	y	$\lfloor x/y \rfloor$	a	b
31	11	2		
11	9	1		
9	2	4		
2	1	2	0	1
1	0		1	0

This should make sense, since these last two rows are equivalent to the equations $1(1) + 0(0)$ and $2(0) + 1(1)$, which are both equal to the gcd, 1, as desired.

Alternatively, you can apply the rules for a and b on the last row to get the second to last row manually: $a' = b = 0$, and $b' = a - \lfloor x/y \rfloor \cdot b = 1 - \lfloor x/y \rfloor \cdot 0 = 1$, no matter what $\lfloor x/y \rfloor$ is.

Now, let's apply these rules for a an b to get the next row above:

$$\begin{aligned} a' &= b = 1 \\ b' &= a - \left\lfloor \frac{x}{y} \right\rfloor \cdot b = 0 - 4 \cdot 1 = -4 \end{aligned}$$

x	y	$\lfloor x/y \rfloor$	a	b
31	11	2		
11	9	1		
9	2	4	1	-4
2	1	2	0	1
1	0		1	0

With these new values, we can apply the rules again:

$$a' = b = -4$$
$$b' = a - \left\lfloor \frac{x}{y} \right\rfloor \cdot b = 1 - 1 \cdot -4 = 5$$

x	y	$\lfloor x/y \rfloor$	a	b
31	11	2		
11	9	1	−4	5
9	2	4	1	−4
2	1	2	0	1
1	0		1	0

We can do this one last time to get our final equation:

$$a' = b = 5$$
$$b' = a - \left\lfloor \frac{x}{y} \right\rfloor \cdot b = -4 - 2 \cdot 5 = -14$$

x	y	$\lfloor x/y \rfloor$	a	b
31	11	2	5	−14
11	9	1	−4	5
9	2	4	1	−4
2	1	2	0	1
1	0		1	0

Reading off the first row, we have $31(5) + 11(-14) = 1$, or $11^{-1} \equiv -14 \equiv 17 \pmod{31}$, just like before.

In general, this tabular method works essentially the same as the iterative approach, but is more algorithmic and organized. All you need to remember are the two sets of rules for x and y, and for a and b (this is something you could put on your cheatsheet!), and the rest of the algorithm is just the application of these rules.

A potential downside of the tabular approach, however, is that in very specific circumstances, finding $\lfloor x/y \rfloor$ may be difficult/confusing (but still possible!). An example of this is question 8 on the summer 2020 midterm—though the chances of this being on another exam is slim.

Every problem can be solved with any of these approaches, so it's up to you which one you prefer and which one you feel most comfortable with.

2 Chinese Remainder Theorem

The Chinese Remainder Theorem gives a systematic way of solving systems of modular equations of the form (and guarantees the uniqueness of a solution):

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &= a_n \pmod{m_n} \end{aligned}$$

An important requirement for using CRT is that the moduli must all be pairwise coprime. This means that any pair of moduli must have a gcd of 1.

In case this is not true, you can always decompose the non-prime moduli into equations for each prime factor (or prime powers). For example, you can decompose

$$x \equiv 3 \pmod{14} \implies \begin{cases} x \equiv 3 \equiv 1 & \pmod{2} \\ x \equiv 3 & \pmod{7} \end{cases}$$

If a contradiction arises from this decomposition, this means that there are no solutions to the system.

Note that going the other way is also possible, and can be proven through CRT (as an exercise for the reader). For example, this means the following is also valid (from the worksheet):

$$\begin{cases} x \equiv -1 & \pmod{3} \\ x \equiv -1 & \pmod{4} \\ x \equiv -1 & \pmod{5} \end{cases} \implies x \equiv -1 \equiv 59 \pmod{60}.$$

This last fact is often very useful when solving some simple systems of equations—it's always the first thing that I check when solving a system, as it simplifies the problem significantly.

Let's get back to CRT. The scary formula for CRT is

$$x \equiv \sum_{i=1}^{n} a_i b_i \pmod{M}, \text{ where } b_i = \frac{M}{m_i}\left(\left(\frac{M}{m_i}\right)^{-1} \bmod m_i\right) \text{ and } M = \prod m_i.$$

Although this looks quite daunting, let's break this up into parts—it's more intuitive than you may think.

The essence of this formula is that we can find a way to sum together different components, each satisfying exactly one of the equations in the system.

$\frac{M}{m_i}$ is the product of all of the moduli *except* the one we're currently focusing on (i.e. the product of every modulus except m_i)—notice that this makes $\frac{M}{m_i} \equiv 0 \pmod{m_j}$ for all $j \neq i$, and that $\frac{M}{m_i} \not\equiv 0 \pmod{m_i}$.

Multiplying by the inverse, $\left(\frac{M}{m_i}\right)^{-1} \pmod{m_i}$, makes it so that we have $b_i \equiv 1 \pmod{m_i}$, and preserving the fact that $b_i \equiv 0 \pmod{m_j}$ for all $j \neq i$.

What have we done so far? If you're familiar with linear algebra, this is just like a standard basis vector—equal to 1 in exactly one coordinate, and 0 in all other coordinates. If you're not familiar with linear algebra, this construction of b_i allows us to get *components* of the solution to the system, one equation at a time.

How do we get these components? We just multiply by the constants a_i. What does this do? We had before that $b_i \equiv 1 \pmod{m_i}$, so multiplying by a_i gives $a_i b_i \equiv a_i \pmod{m_i}$.

This expression is now a solution to the ith equation in the system, while preserving the property that it's equivalent to 0 in every other modulus. As such, we can calculate this value for every equation i, and just add them together.

Since each expression is only nonzero in exactly one modulus (m_i), adding them together gives us a value that satisfies *all* of the equations—the solution x that we're looking for.

The way I like to execute CRT is as follows:

1. Pick an equation to focus on.
2. Calculate the product of the moduli of all *other* equations. (This is $\frac{M}{m_i}$.)
3. Calculate the inverse of this product, mod m_i. (This is $\left(\frac{M}{m_i}\right)^{-1} \bmod m_i$.)
4. Multiply these together, and also multiply the constant in the equation we're currently focusing on. (This is now $a_i b_i$.)
5. Repeat for every single equation in the system.
6. Add everything up, and simplify mod M.