Section 4.7: The Euclidean Algorithm*

We mentioned when we talked about the gcd of m and n, that there was an algorithm that helps find the gcd. This algorithm will also help find s and t such that $\gcd(m, n) = sm + tn$. And help us solve the expression

$$x \cdot n \equiv a(mod\, m).$$

Why would we want to solve such an equation?

- applications to public-key cryptosystems for secure transmission of data
- fast implementation if computer arithmetic for very large numbers

So what is this algorithm?

Proposition. *If m and nare integers with $n > 0$, then the common divisors m and n are the same as the common divisors of n and $m \mod n$).*

Let's try a few examples.
Find the common divisors of 90 and 6.

What does this theorem say that about this example?

Proof.
We just need to show that if d is a divisor m and n. Then d is a divisor of m mod n. And then on the reverse side, if d' is a divisor of n and m mod n then d' is a divisor of m mod n.
To start, let's assume d is a divisor of m and n. We know

$$m = \qquad\qquad +m \mod n$$

so

$$m \mod n =$$

And therefore:

*For Math 243, Katie Walsh

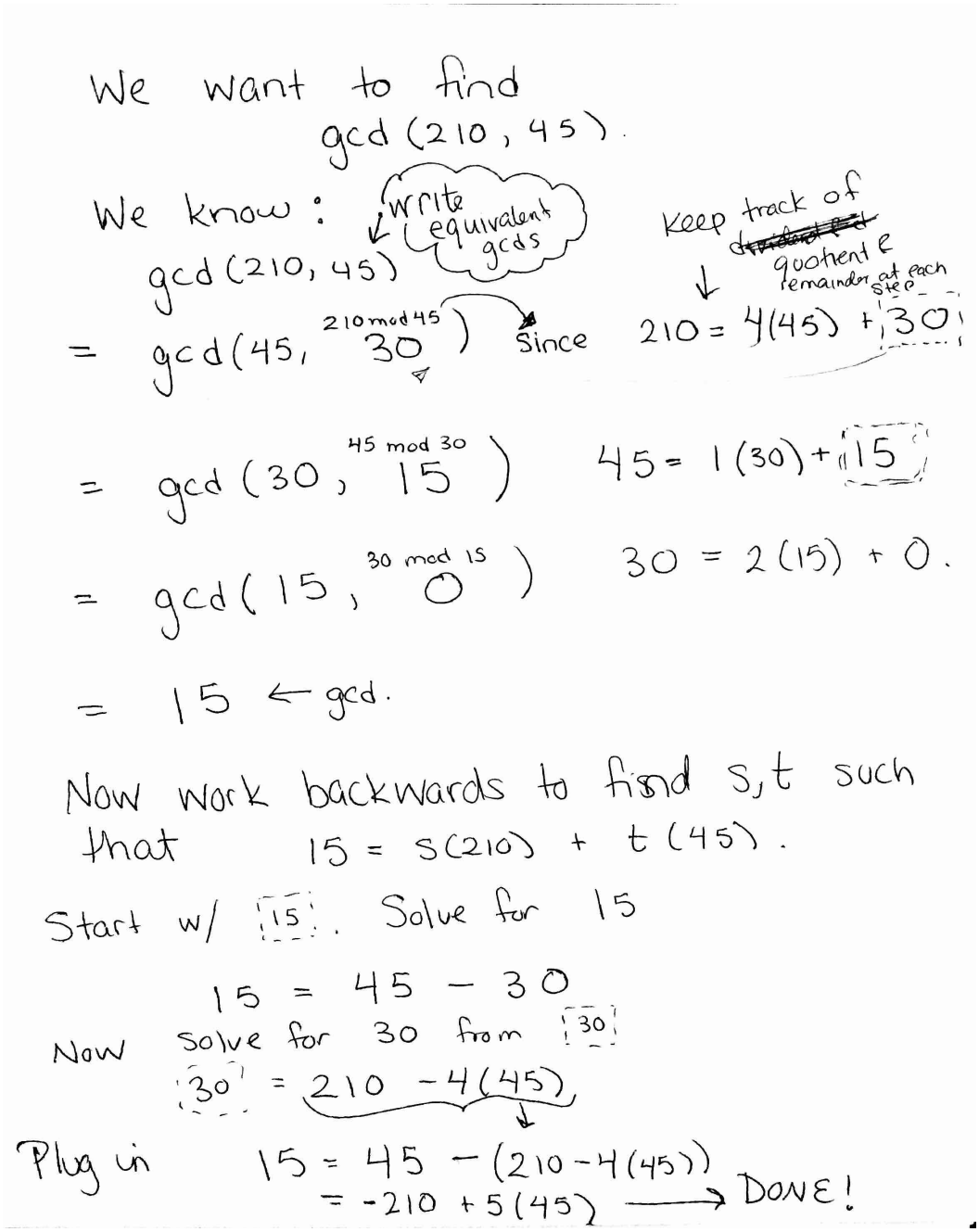
Now, assume d' is a divisor of n and m mod n and show d' is a divisor of m mod n.

This leads us to the an algorithm for finding the gcd.
AlgorithmGCD(integer, integer)
{Input $m, n \in \mathbb{N}$ not both 0}
{Output: gcd(m,n)}
{Auxiliary Variables: integers a and b}
$a := m; b := n$
{The pairs (a, b) and (m, n) have the same gcd.}
while $b \neq 0$ **do**
: $(a, b) := b, a \mod b)$
return a
Run through this algorithm to find $\gcd(20, 63)$.

Run through this algorithm to find $\gcd(45, 12)$.

Theorem. *For input integers $m > n \geq 0$, AlgorithmGCD makes at most $2log_2(m + n)$ passes through the loop.*

Now, we can also extend the Euclidean Algorithm, to run in the same number of passes through the loop as before, but to remember a bit more information so that we can find s and t such that $\gcd(m, n) = sm+tn$. Here's an example of how that works:



Use the algorithm to find $s$and t such that $\gcd(m, n) = sm + tn$ for the pairs below. You may need extra paper.

$m = 20, n = 63$

$m = 120, n = 162$

$m = 17, n = 123$

And now, we can solve problems of the form

$$n \cdot x \equiv a (mod\, m)$$

The trick is if we can find t such that

$$n \cdot t \equiv 1 (mod\, m)$$

Then we can just multiply both sides by a. So now we just need to find t. Reorganizing this (using the definition of mod) we get (for some s)

$$n \cdot t = 1 + s \cdot m$$

Or that

$$1 = n \cdot t - s \cdot m$$

This is exactly (up to sign) what we found from the Euclidean algorithm. Look at the example below:

Find x such that

$123x \equiv 5 \mod 17$

We'll find t such that

$123t \equiv 1 \mod 17$

then $x = 5t$.

By Euclidean Algorithm (you did this)

gcd(123, 17)

$= \gcd(17, {}^{123 \bmod 17}4)$ $\quad 123 = 17(7) + \boxed{4}$

$= \gcd(4, {}^{17 \bmod 4}1)$ $\quad 17 = 4(4) + \boxed{1}$

$= \gcd(1, 0)$

$= 1$

Now $1 = 17 - 4(4)$

& $4 = 123 - 17(7)$

so $1 = 17 - 4(123 - 17(7))$

$1 = -4(123) + 29(17)$

Thus $t = -4$, so $x = -20$

Check $123(-20) = -2460 = -145 \cdot 17 + 5$

$\equiv 5 \mod 17.$

Note that this only works when the $\gcd(m, n) = 1$. If the gcd is not 1, then

$$n \cdot x \equiv a (mod\, m)$$

only has a solution when a is a multiple of the gcd so we can adapt the algorithm to find it. Try a few examples of this.

Solve the following:

$10x \equiv 3 (mod\, 37)$

$120x \equiv 12 (mod\, 162)$

$20x \equiv 5 (mod\, 63)$