More about gcd's: the <u>extended</u> Euclidean algorithm.

Recall that for $a, b \in \mathbb{Z}$, $\exists\, x, y \in \underline{\mathbb{Z}}$ s.t. $\gcd(a,b) = xa + yb$.

Let's see how to find x, y.

Aside: one application, useful in cryptography, is to find <u>modular inverses</u>.

That is, given $a, n \in \mathbb{Z}^+$, find $x \in \mathbb{Z}^+$ s.t. $ax \,\%\, n = 1$.

(This is possible only when $\gcd(a,n) = 1$.)

If $\gcd(a,n) = 1$, then if $x, y \in \mathbb{Z}$ s.t. $ax + bn = 1$, then note that $ax = 1 - bn$

so $ax \,\%\, n = (1 - bn) \,\%\, n = 1$. ✓

How to compute x, y when given a, b?

Again, we'll use recursion.

Possible prototype:

$$\underline{\texttt{int}}\ \texttt{xgcd(int a, int b, }\overbrace{\texttt{int\& x, int\& y}}^{\text{storage for outputs!}}\texttt{);}$$

Base case? Before, we used b == 0.

If b == 0, what are valid choices of

x, y? $x = 1,\ y = 0$

so that $$xa + yb = 1 \cdot a + 0 \cdot 0$$
$$= a = \gcd(a, 0).$$

Now, assuming our xgcd works for any smaller value of b, how could we find x, y for a, b when $b \neq 0$?

Say $a = qb + r.$, $q \in \mathbb{Z}$, $r \lneqq b$.

As we saw before, $\gcd(a,b) = \gcd(b,r)$.

Assuming xgcd works for smaller values of the second input, we can find x', y' s.t.

$$x'b + y'r = d \qquad (d = \gcd(a,b))$$

But $r = a - qb$ $(q = a/b,\ r = a\%b)$

So, $$d = x'b + y'(a - qb)$$
$$= x'b + y'a - y'qb$$
$$= \underset{\substack{\| \\ x}}{y'} a + \underset{\substack{\| \\ y}}{\underline{(x' - y'q)}}\, b$$

x', y' are the right values for $\underline{b, r}$.
I.e., $x'b + y'r = d$

Yay. ✓

Analysis: how many recursive calls will be made, relative to $|b|$?

Claim: only takes $\approx c \cdot \log_2 |b|$ steps for a small constant c.

How to see this?

If r is "large", the "next r" won't be...

Exercise/TODO: make this argument formal + precise!