More examples w/ recursion.

GCD's

gcd(a,b) = largest integer d s.t. $d | a$ and $d | b$.

(Note: $d | a$ means $\exists\, k \in \mathbb{Z}$ s.t. $a = kd$.)

Earlier, we used "brute force":

```
int d = min(a,b);
while (!(a%d==0 && b%d==0))
    d--;
return d;
```

$\underbrace{\texttt{!(a\%d==0 \&\& b\%d==0)}}$ ⇐ De Morgan's Law.

`a%d != 0 || b%d != 0`

Might take a lot of steps... in the worst case, it would take $\approx$ min(a,b) steps.

Today we'll find a way to do this computation with $\approx \log_2(\min(a,b))$ steps.

Key observation: common divisors of a, b are <u>the same</u> as the common divisors of b, r, where $r = a \% b$.

Proof/reason: Note that for any $a, b \in \mathbb{Z}$, $\exists\, q, r \in \mathbb{Z}$ s.t.

$$a = qb + r, \quad r \lneq b.$$

$q \equiv$ "quotient", $r \equiv$ "remainder".

(in C/C++, $q = a/b$, $r = a\%b$.)

Now suppose $d|a$ & $d|b$.

That is, $\exists\, k_a, k_b \in \mathbb{Z}$ s.t., $a = k_a d$, $b = k_b d$.

But $r = a - qb$

have a factor of d..

$= k_a d - q k_b d$

$= (k_a - q k_b) d \implies d|r.$

$\mathbb{Z}$

And if $d|r$, $d|b$, clearly $d|a$:

$a = qb + r = q k_b d + k_r d = (q k_b + k_r) d$ ✓

(where $r = k_r d$)

$\mathbb{Z}$.

So common divisors of a, b $\equiv$ common divisors of b, r.

This suggests the following recursive procedure:

```
size_t gcd (size_t a, size_t b)
{       if (b == 0) // base case
            return a;
      return gcd(b, a%b); // note: a%b ≨ r
}
```

Note: we can use b as the "size" of the input.

Exercise: think about how many recursive calls our gcd function might make in terms of a, b.

Exercise: what happens when $a = f_k$, $b = f_{k-1}$ where $\{f_k\}_{k=0}^{\infty}$ is the Fibonacci sequence?

Sample trace: $\gcd(12, 18)$:

$$\begin{array}{c} (12, 18) \\ | \\ (18, 12) \\ | \\ (12, 6) \\ | \\ (6, 0) = 6. \end{array}$$