GCDs

Recall our naive method (brute force):

to find $\gcd(a,b)$: (say neither $a,b = 0$)

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

Cost? (in terms of a,b)

Might take $\approx \min(a,b)$ steps.

Smarter recursive solution

Recall the "division algorithm":

$\forall\, a,b \in \mathbb{Z},\ \exists\, q,r \in \mathbb{Z}$ with $r \lneq b$

s.t. $a = qb + r$. ($q \equiv$ quotient, $r \equiv$ remainder)

Key observation: common divisors of a,b are <u>the</u> <u>same</u> as the common divisors of b,r.

Might be useful: $\boxed{\gcd(a,b) = \gcd(b,r)}$

but $r \lneq b$, so the second input is smaller.

So if we define the "size" of an input as the magnitude of the second param, it is always shrinking...

Given the above observation as a fact, we could do this:

```
size_t gcd(size_t a, size_t b)
{   if (b == 0) return a;
    return gcd(b, a % b);
}
```

smaller!

Let's prove the key observation:

$$d \mid a \text{ and } d \mid b \iff d \mid b \text{ and } d \mid r$$

(where r is from the div. algo).

($\Longrightarrow$) Suppose $d \mid a$ & $d \mid b$. Then $a = a' \cdot d$, $b = b' \cdot d$ for $a', b' \in \mathbb{Z}$.

Since $a = qb + r$,
$$\begin{aligned} r &= a - qb \\ &= a'd - qb' \cdot d \\ &= (a' - qb') \cdot d \end{aligned}$$
$$\Longrightarrow d \mid r. \quad \checkmark$$

($\Longleftarrow$)
$$\begin{aligned} a &= qb + r \\ &= qb'd + r'd \\ &= (qb' + r')d. \quad \checkmark \end{aligned}$$

<u>Example</u>:

$a = 12$, $b = 18$

$a \% b = a$

$\gcd(12, 18)$

|

$\gcd(18, 12)$

|

$\gcd(12, 6)$

|

$\gcd(6,0) = 6.$ ✓

Claim: this is *way* faster than the brute force algo.

Why? Need to think about total # of recursive calls before we hit $b == 0$.

How big could r be? (In terms of a & b)

$r \leq \underline{b-1}$.

But if r was large (close to b), then the *next* call looks like $\gcd(b, b-\varepsilon)$ for ε "small".

So the *next* remainder is small!

More formal claim: after 2 calls, second param $\leq b/2$.

$\gcd(1000, 995)$

$\gcd(995, 5)$

$$\boxed{\begin{gathered} b \,\%\, (b-\varepsilon) = \varepsilon \\ \text{whenever } \varepsilon < b/2 \end{gathered}}$$

Say $\boxed{q=1}$ in $a = qb + r$

Then $r = a - b$

(in our situation, $r = b - (b-\varepsilon) = \varepsilon$).

Corollary: $\gcd(a,b)$ only takes $\approx \log_2 b$ steps!!

Maybe of interest: think about what happens when $a = f_k$, $b = f_{k-1}$ for $\{f_i\}_{i=0}^{\infty}$ = Fibonacci sequence...

Note: $\gcd(a,b) = xa + yb$ for $x, y \in \mathbb{Z}$.

Question: can we modify our gcd algo to compute such x and y?