

GCDs

Recall our naive method (brute force):

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

$d = \min(a, b);$

while $(a \% d \neq 0 \parallel b \% d \neq 0)$ $d--;$

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 $r \leq b$

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

Key observation: common divisors of a, b are
the same as the common divisors of b, r .

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

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

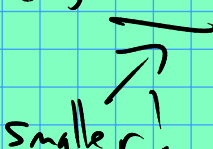
So if we define the "size" of an input as the magnitude of the second param, it is always shrinking...

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:

$d|a$ and $d|b \iff d|b$ and $d|r$
 (where r is from the div. algo).

(\implies) $\nexists d|a \nmid d|b$. Then $a = a' \cdot d$, $b = b' \cdot d$
 for $a', b' \in \mathbb{Z}$.

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

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

Example: $\gcd(12, 18)$

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

$$a = 12, b = 18$$

$$a \% b = a$$

$$\gcd(6,0) = 6. \quad \checkmark$$

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 $\gcd(b, b-\epsilon)$ for ϵ "small".

So the next remainder is small!

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

$$\gcd(1000, 995)$$

$$\gcd(995, 5)$$

$$\boxed{\begin{array}{l} b \% (b-\epsilon) = \epsilon \\ \text{Whenever } \epsilon < b/2 \end{array}}$$

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

$$\text{Then } r = a - b$$

$$(\text{in our situation, } r = b - (b - \epsilon) = \epsilon).$$

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

Maybe of interest: think about what happens when

$$a = f_k, b = f_{k-1} \quad \text{for } \{f_i\}_{i=0}^{\infty} \equiv \text{Fibonacci sequence} \dots$$

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 ?