

Greatest Common Divisor

- Since $\gcd(a, b) = \gcd(|a|, |b|)$, we consider $\gcd(a, b)$ with $a, b \in \mathcal{N}$. By prime factorization

$$a = \prod p_i^{l_{a_i}},$$

$$b = \prod p_i^{l_{b_i}}.$$

Then $\gcd(a, b) = \prod p_i^{\min(l_{a_i}, l_{b_i})}$.

However, factoring integers is **HARD**, so this is impractical. Instead we will derive a “fast” algorithm: Euclid’s algorithm (recall an old algorithm).

- **Theorem** (GCD recursion theorem):

$\forall a \in \mathcal{N}$ and positive integers b ,

$$\gcd(a, b) = \gcd(b, a \pmod{b})$$

Proof: we will show

1. $\gcd(a, b) \mid \gcd(b, a \pmod{b})$

$$2. \gcd(b, a \pmod{b}) \mid \gcd(a, b) \implies$$

$\gcd(a, b) = \pm \gcd(b, a \pmod{b})$ but since both are nonnegative, they are equal.

1. Let $d = \gcd(a, b)$, then $d \mid a$ and $d \mid b$, now

$a \pmod{b} = a - \lfloor \frac{a}{b} \rfloor b$. so it is an integer combination of a and b , and so $d \mid a \pmod{b}$ and $d \mid \gcd(b, a \pmod{b})$.

2. Let $d = \gcd(b, a \pmod{b})$, then $d \mid b$ and $d \mid a \pmod{b}$.

Since

$a = a \pmod{b} + \lfloor \frac{a}{b} \rfloor b$ is an integer combination of b and $a \pmod{b}$, $d \mid a$ so $d \mid \gcd(a, b)$. ■

• Euclid's Algorithm

This algorithm was described by Euclid in his "Elements" written about 300 B.C.

It is based on the previous theorem.

EUCLID (a, b)

1. **if** $b = 0$
2. **then return** a
3. **else return** EUCLID $(b, a \pmod{b})$.

This algorithm computes $\gcd(a, b)$

by transforming the gcd into equivalent gcd with progressively smaller arguments.

$$\begin{aligned}
& \text{gcd}(30, 21) \\
&= \text{gcd}(21, 30 \pmod{21}) \\
&= \text{gcd}(21, 9) \\
&= \text{gcd}(9, 21 \pmod{9}) \\
&= \text{gcd}(9, 3) \\
&= \text{gcd}(3, 9 \pmod{3}) \\
&= \text{gcd}(3, 0) \\
&= 3
\end{aligned}$$

Since the recursive calls in EUCLID terminate after finitely many calls (the halting problem?), but how many?

- **The Running Time of EUCLID**

What is the worst case for EUCLID ?

We can assume $a > b \geq 0$, since it not, the first pass of EUCLID fixes this!

Also, if $b = a > 0$, then it returns b after one call. So given this what are the WORST a, b pairs to put into this: successive Fibonacci numbers!

- **Lemma:** IF $a > b \geq 0$ and EUCLID (a, b) performs $k \geq 1$ recursive calls, then

$$\begin{aligned} a &\geq F_{k+2} && \text{and} \\ b &\geq F_{k+1} \end{aligned}$$

Proof: We prove by induction on k .

$k = 1$: If $b = 0$, then $k = 0$, so $b \geq 1 = F_2$.

Now $a > b$, so $a \geq b + 1 \geq 2 = F_3$.

Assume true for $k - 1$. Assume that a, b are such that EUCLID (a, b) takes k recursive calls.

The first of these is EUCLID $(b, a \pmod{b})$. By assumption, this terminates in $k - 1$ calls so that

$$\begin{aligned} b &\geq F_{k-1+2} = F_{k+1} \\ a \pmod{b} &\geq F_{k+1-1} = F_k \end{aligned}$$

We want to prove that $a \geq F_{k+2}$ as well.

$$\begin{aligned} b + a \pmod{b} &= b + a - \lfloor \frac{a}{b} \rfloor b \\ &= a + b(1 - \lfloor \frac{a}{b} \rfloor) \end{aligned}$$

Since $a > b > 0$, we know $\lfloor \frac{a}{b} \rfloor \geq 1$

$$= a \quad \text{if } \lfloor \frac{a}{b} \rfloor = 1$$

$$= a - b \quad \text{if } \lfloor \frac{a}{b} \rfloor = 2$$

$$\leq a \quad \text{in all cases.}$$

So $a \geq b + a \pmod{b}$

$$\geq F_{k+1} + F_k = F_{k+2}. \quad \blacksquare$$

- **Theorem (Lame's Theorem):** \forall integers $k \geq 1$,
if $a > b \geq 0$ and $b < F_{k+1}$, then

EUCLID (a, b) takes fewer than k recursive calls.

This theorem follows from the lemma.

The upper bound on Lame's theorem is the best possible as seen with:

$$\gcd(F_{k+2}, F_{k+1}) =$$

$$\gcd(F_{k+1}, F_{k+2} \pmod{F_{k+1}})$$

But $F_{k+2} = F_{k+1} + F_k$ so

$$\begin{aligned}
F_{k+2} \pmod{F_{k+1}} &= F_k \\
&= \gcd(F_{k+1}, F_k) \\
&\dots \\
&\dots \\
&= \gcd(F_1, F_0) = \gcd(1, 0) = 1
\end{aligned}$$

This takes exactly $k + 1$ recursive calls

This is what the lemma states as

$$b = F_{k+1} < F_{k+2}$$

Recall $F_k = \text{rnd}(\phi^k / \sqrt{5})$, $\phi = \frac{1+\sqrt{5}}{2}$.

Since there are k calls it, $b < F_{k+1}$

$$b \approx \frac{\phi^k}{\sqrt{5}} \text{ or } k = O(\lg b). \blacksquare$$

- If a and b are β -bit numbers than you can show it takes only $O(\beta^2)$ bit operations to compute $\gcd(a, b)$, see problem 31-2.

Figure 1: An example of the operation EXTENDED-EUCLID on the inputs 99 and 78

- **The Extended Euclidean Algorithm**

Since $\gcd(a, b) = \min\{ax + by > 0 : x, y \in \mathbb{Z}\}$

One can try to find x, y so that

$$\gcd(a, b) = ax + by$$

The extended Euclidean algorithm does this by “carrying along” valid x and y values until the gcd is computed.

- **EXTENDED-EUCLID**(a, b)

1. **if** $b=0$
2. **then** $(a, 1, 0)$
3. $(d', x', y') \leftarrow \text{EXTENDED-EUCLID}(b, a \pmod{b})$
4. $(d, x, y) \leftarrow (d', y', x' - \lfloor \frac{a}{b} \rfloor y')$
5. **return** (d, x, y)

This example shows how this works on $\gcd(99, 78)$

Note : $d = \gcd(a, b) = ax + by$

$$d' = \gcd(b, a \pmod{b}) = bx' + a \pmod{b}y'$$

they are equal, so

$$d = bx' + a \pmod{b}y'$$

$$= bx' + (a - \lfloor \frac{a}{b} \rfloor b)y'$$

$$= ay' + b(x' - \lfloor \frac{a}{b} \rfloor y')$$

Thus if $x = y'$ and $y = x' - \lfloor \frac{a}{b} \rfloor y'$

we get consistency, and this proves why it works.

- This clearly has the same number of recursive calls as before, so Lamé's theorem holds for the extended version as well.