Welcome to CIS 121.

* Efficiency (Running time)

Correctness

**Greatest Common Divisor**

$d$ is a common divisor of $a$ and $b$ if $d | a$ and $d | b$.

\[ \text{Greatest Common Divisor of } g, b \]

then if $d' | a$ and $d' | b$

then $d' \leq d$.

\[ \gcd(g, b) = 6 \]

\[ a = 24 \]

\[ b = 18 \]

\[ \pm 1, \pm 2, \pm 3, \pm 6 \]

\[ \times 18 \]
This naive algorithm will check all common divisors and will take $O(n)$ running steps.

**Proposition:** If $a \geq b$ and $c = a \mod b$ then $\gcd(a, b) = \gcd(b, c)$.

**Example:** $\gcd(34, 9) = \gcd(9, 6)$

**Proof:**

$$a \geq b \quad a = qb + r$$

$$[\gcd(9, 24) = 9 = 0.24 + 9]$$

Let $d = \gcd(a, b) \geq d | b$

Because $c = a - qb$, $d | c$

Let $e = \gcd(b, c)$ $\Rightarrow d \leq e$
\[ a = q_0 b + c \Rightarrow e \mid a \Rightarrow e \leq d \]
\[ \Rightarrow d = e. \]

Example: 
\[ \gcd(683, 234) \]
\[ = \gcd(234, 221) \]
\[ = \gcd(221, 13) \]
\[ = \gcd(13, 0) = 13 \]

Euclid's Algorithm

Input: \( a, b > 0 \)
Output: \( \gcd(a, b) \)
Steps: 
\[ c = a \mod b \]
if \( c = 0 \) return \( b \)
else return gcd(b, c)

\[ a = 63 \quad b = 75 \quad c = 63 \]

\[
\begin{array}{llll}
75 & 63 & 12 & \\
63 & 12 & 3 & \\
12 & 3 & 0 & \\
\end{array}
\]

First call switches a and b if \( a < b \)

return 3

Does Euclid's algorithm indeed compute \( gcd(a, b) \)?

WLOG (without loss of generality) \( a \geq b \). We will prove by contradiction. Assume that \( a, b \) exist and that their sum

10
is the smallest possible.

\[ c \neq 0 \quad a = q_b + c \quad 0 < c < b \leq a \]

\[ c < b \]
\[ b \leq e \]
\[ 5 + c < a + 5 \]

If the algorithm computed \( \gcd(b, c) \) (next step) correctly then \( \gcd(a, b) \) must also be correct. But this isn't per our assumption. Their mean that \( \gcd(b, c) \) fails as well.

But then \( 3b, c \) with smaller sum than \( 3b \)

for which it fails. Contradict our pick of \( a \) and \( b \) with the smallest possible sum \( a + b \). \( \square \)
Summary of proof:
\( \gcd(a, b) \) fails for \( a, b \) with \( e + b \) minimum
\( \gcd(b, c) \) fails for \( b, c \) with \( b + c \) minimum
\[ b + c < a + b \].

Alternative proof:
Induction on the number of steps.
Runtime of Euclid's GCD algorithm

Proposition: If \( a \geq b > 0 \) and \( c = a \mod b \) then \( c < \frac{a}{2} \)

Proof:

(1) \( a < 2b \) (e.g. \( \gcd(36,23) \))

\[
a = 1 \cdot b + c
\]

\[
c = a - b \quad b > \frac{a}{2}
\]

\[
< a - \frac{a}{2} \quad -b < -\frac{c}{2}
\]

\[
= \frac{a}{2}
\]

(2) \( a \geq 2b \quad b \leq \frac{c}{2} \)

\( c < b \) anyway (remainder \(<\) divisor)
\[ c < b \leq \frac{a}{2} \]

Sketch for a runtime proof:

\[ a \geq b \geq c \geq d \geq e \geq \frac{b}{2} \geq \frac{a}{2} \]

After \( 2^n \) recursions:

\[ S \geq t \geq 0 \]

\[ S < \frac{a}{2^n} \quad t < \frac{b}{2^n} \]