RECURSION – EUCLID'S ALGORITHM

DISCRETE STRUCTURES II

DARRYL HILL

BASED ON THE TEXTBOOK:

DISCRETE STRUCTURES FOR COMPUTER SCIENCE: COUNTING, RECURSION, AND PROBABILITY

BY MICHIEL SMID

Recursion

Greatest common divisor.

$a \geq 1, b \geq 1, \gcd(a,b) =$ largest integer that divides both a and b.

Example: $\gcd(75{,}45) =$

Common divisors: $1, 3, 5, 15$

$\gcd(a,a) = a$

How do we find gcd of large numbers?

$a = 371\,435\,805$
$b = 137\,916\,675$

$\gcd(a,b) =$

Recursion

Greatest common divisor.

$a \geq 1, b \geq 1, \mathrm{gcd}(a, b) =$ largest integer that divides both a and b.

Example: $\mathrm{gcd}(75,45) =$

Common divisors: $1, 3, 5, 15$

$\mathrm{gcd}(a, a) = a$

How do we find gcd of large numbers?

$$a = 371\,435\,805 = \overbrace{3^2 \cdot 5^1 \cdot 13^4 \cdot 17^2}^{\text{Prime Factorization}}$$
$$b = 137\,916\,675 = 3^4 \cdot 5^2 \cdot 13^3 \cdot 31$$

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

Recursion

Greatest common divisor.

$a \geq 1, b \geq 1, \mathrm{gcd}(a,b) =$ largest integer that divides both a and b.

Example: $\mathrm{gcd}(75{,}45) =$

Common divisors: $1, 3, 5, 15$

$\mathrm{gcd}(a,a) = a$

How do we find gcd of large numbers?

$$a = 371\,435\,805 = \overbrace{3^2 \cdot 5^1 \cdot 13^4 \cdot 17^2}^{\text{Prime Factorization}}$$
$$b = 137\,916\,675 = 3^4 \cdot 5^2 \cdot 13^3 \cdot 31$$

$$\mathrm{gcd}(a,b) = 3^2 \cdot 5^1 \cdot 13^3 = 98\,865$$

Recursion

Greatest common divisor.

$a \geq 1, b \geq 1, \gcd(a, b) =$ largest integer that divides both a and b.

Example: $\gcd(75{,}45) =$

Common divisors: $1, 3, 5, 15$

$\gcd(a, a) = a$

How do we find gcd of large numbers?

$$a = 371\,435\,805 = \overbrace{3^2 \cdot 5^1 \cdot 13^4 \cdot 17^2}^{\text{Prime Factorization}}$$
$$b = 137\,916\,675 = 3^4 \cdot 5^2 \cdot 13^3 \cdot 31$$

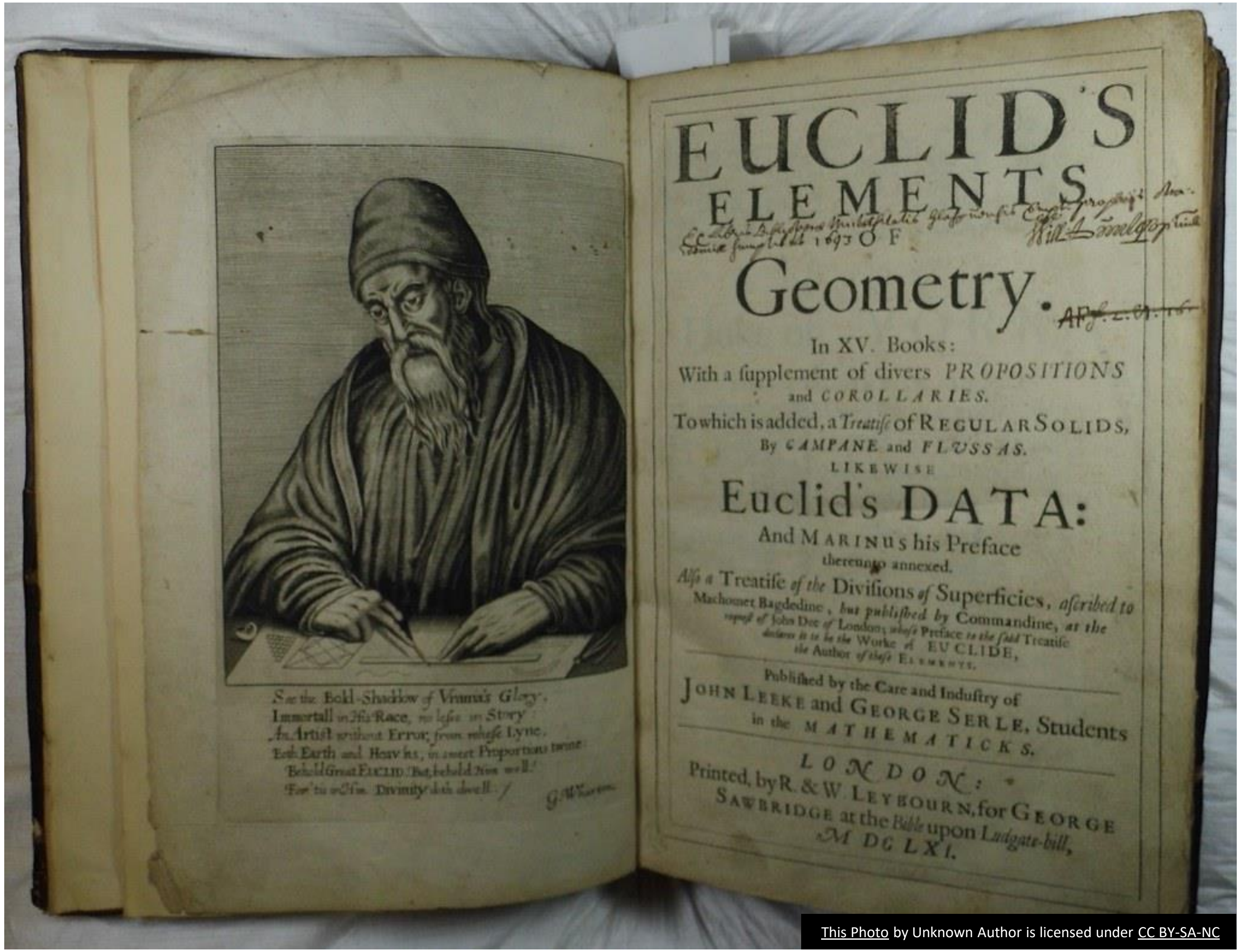
$$\gcd(a, b) = 3^2 \cdot 5^1 \cdot 13^3 = 98\,865$$

Compute Prime Factorization of a and b

Very slow!

Much computer security is based on the fact that prime factorization is very slow

An integer with 1000 digits will take 1000's of years to compute PF.



Recursion

Greatest common divisor.

$a \geq 1, b \geq 1, \text{gcd}(a, b) =$ largest integer that divides both a and b.

Easy, fast algorithm to compute $\text{gcd}(a, b)$.

Invented by Euclid around 300 BC.

Uses the Modulo operation.

Modulo

Modulo operation:

$a \bmod b$ = remainder of a divided by b

$a = qb + r,$

$0 \le r < b, q \ge 0$

$q =$ quotient
$r =$ remainder

$a \bmod b = r$

$a \bmod b = r$

$17 \bmod 5 = 2$

$17 \bmod 17 = 0$

$17 \bmod 1 = 0$

$17 \bmod 19 = 17$

$a = qb + r$

$17 = 3 \cdot 5 + 2$

$17 = 1 \cdot 17 + 0$

$17 = 17 \cdot 1 + 0$

$17 = 19 \cdot 0 + 17$

Euclid Algorithm

$a \bmod b = r$

$a = qb + r,$

$0 \leq r < b, q \geq 0$

Algorithm Euclid(a, b): $// \; a \geq b \geq 1$
$\quad r = a \bmod b$
$\quad$if $r = 0$: return b
$\quad$if $r \geq 1$:
$\quad\quad$return Euclid(b, r) $// \; b \geq r \geq 1$

Euclid(75, 45):

Using prime factorization:

$$75 = 3 \cdot 5^2$$
$$45 = 3^2 \cdot 5$$

$$\gcd(75,45) = 3 \cdot 5 = 15$$

Euclid Algorithm

$\gcd(75{,}45) = 15$

$a \bmod b = r$

$a = qb + r,$

$0 \le r < b, q \ge 0$

Algorithm Euclid(a, b): $// \ a \ge b \ge 1$
  $r = a \bmod b$
  if $r = 0$: return b
  if $r \ge 1$:
    return Euclid(b, r) $// \ b \ge r \ge 1$

Euclid(75, 45):
  $r = 75 \bmod 45 = 30$
  Euclid(45, 30)
    $r = 45 \bmod 30 = 15$
      Euclid(30, 15)
      $r = 30 \bmod 15 = 0$
      return 15

It is correct for this input.

We have to argue Euclid is correct $\forall a, b$ and also that Euclid terminates.

Euclid Algorithm

Lemma 1: $a \geq b \geq 1, r = a \bmod b$

$\boxed{\text{a) if } r = 0 \text{ then } \gcd(a, b) = b}$

$\boxed{\text{b) if } r \geq 1 \text{ then } \gcd(a, b) = \gcd(b, r)}$

$a = qb + r$

a) if $r = 0$,

$\gcd(a, b) = \gcd(qb, b) = b$,

so a) is true

b) if $r \geq 1$ then $\gcd(a, b) = \gcd(b, r)$ is true if

all common divisors of a and b = all common divisors of b and r.

To argue this we must show a bijection between all common divisors of a and b and all common divisors of b and r.

i) First we show that if d is a common divisor of a and b then d is also a common divisor of b and r.
ii) Second we show that if d is a common divisor of b and r then d is also a common divisor of a and b.

Euclid Algorithm

Lemma 1: $a \geq b \geq 1, r = a \bmod b$

a) if $r = 0$ then $\gcd(a, b) = b$
$\boxed{\text{b) if } r \geq 1 \text{ then } \gcd(a, b) = \gcd(b, r)}$

$a = qb + r$

To show: i) if d is a common divisor of a and b then d is also a common divisor of b and r.

$a = qb + r$

$r = a - qb$ where a is a multiple of d and qb is a multiple of d, therefore so is r.

Now we must argue the other direction

Euclid Algorithm

Lemma 1: $a \geq b \geq 1, r = a \bmod b$

a) if $r = 0$ then $\gcd(a, b) = b$

b) if $r \geq 1$ then $\gcd(a, b) = \gcd(b, r)$

$a = qb + r$

To show: i) if d is a common divisor of a and b then d is also a common divisor of b and r.

ii) if d is a common divisor of b and r then d is also a common divisor of a and b.

$a = qb + r$ where r is a multiple of d and qb is a multiple of d, therefore so is a.

If d is a common divisor of a and b then d is also a common divisor of b and $r \rightarrow$ True .

If d is a common divisor of b and r then d is also a common divisor of a and $b \rightarrow$ True.

Therefore all common divisors are the same.

Therefore it must be that $\gcd(a, b) = \gcd(b, r)$. Thus b) is true.

So Lemma 1 is True

Euclid Algorithm

a mod $b = r$

$a = qb + r, 0 \leq r < b, q \geq 0$

Algorithm Euclid(a, b): $// \, a \geq b \geq 1$
  $r = a$ mod b
  if $r = 0$: return b
  if $r \geq 1$:
    return Euclid(b, r) $// \, b \geq r \geq 1$

Lemma 1:
a) if $r = 0$ then $\gcd(a, b) = b$
b) if $r \geq 1$ then $\gcd(a, b) = \gcd(b, r)$

Now we can prove that Euclid is correct using induction. To successfully use induction we require that Euclid terminates, which means we can rank the calls to Euclid.

Can frame induction on size of b. The base case is when $b = 1$, which returns 1 which is true.

Euclid(a,b):
  returns b which is true
  return Euclid(b, r), where $b \leq a - 1$

Euclid(b, r) is correct by induction (that is, it returns $\gcd(b, r)$). Therefore Euclid(a,b) is correct.

Euclid Algorithm

$a \bmod b = r$

$a = qb + r, 0 \leq r < b, q \geq 0$

Algorithm Euclid(a, b): $// \, a \geq b \geq 1$
    $r = a \bmod b$
    if $r = 0$: return b
    if $r \geq 1$:
        return Euclid(b, r) $// \, b \geq r \geq 1$

Lemma 1:
a) if $r = 0$ then $\gcd(a, b) = b$
b) if $r \geq 1$ then $\gcd(a, b) = \gcd(b, r)$

How efficient is Euclid(a,b)?

$M(a, b) =$ number of times line * is executed.

Euclid(75, 45):
    $r = 75 \bmod 45 = 30*$
    Euclid(45, 30)
        $r = 45 \bmod 30 = 15*$
            Euclid(30, 15)
            $r = 30 \bmod 15 = 0*$
            return 15

$M(75,45) = 3$

Euclid Algorithm

$a \bmod b = r$

$a = qb + r, 0 \le r < b, q \ge 0$

Algorithm Euclid(a, b): $// \; a \ge b \ge 1$
   $r = a \bmod b$
   if $r = 0$: return b
   if $r \ge 1$:
      return Euclid(b, r) $// \; b \ge r \ge 1$

Lemma 1:
a) if $r = 0$ then $\gcd(a, b) = b$
b) if $r \ge 1$ then $\gcd(a, b) = \gcd(b, r)$

How efficient is Euclid(a,b)?

$M(a, b)$ = number of times line * is executed.

Start with easy analysis:

Euclid(a, b):
   always $b \ge 1$
   decreases by ≥ 1

$M(a, b) \le b$

But we can do a better analysis based on the Fibonnacci sequence

Euclid Algorithm

$a \bmod b = r, \quad a = qb + r$

Algorithm Euclid(a, b): // $a \geq b \geq 1$
  $r = a \bmod b$ *
  if $r = 0$: return b
  if $r \geq 1$:
    return Euclid(b, r) // $b \geq r \geq 1$

$M(a, b)$ = number of times line * is executed.

Fibonacci: $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2$, etc

Lemma2: $a \geq b \geq 1, m = M(a, b)$
Then $a \geq f_{m+2}, b \geq f_{m+1}$

(Does this mean m is large or small compared to b?)

Euclid Algorithm

$a \bmod b = r, \quad a = qb + r$

Algorithm Euclid(a, b): $// \; a \geq b \geq 1$

$r = a \bmod b$ *

if $r = 0$: return b

if $r \geq 1$:

return Euclid(b, r) $// \; b \geq r \geq 1$

$M(a, b) =$ number of times line * is executed.

Fibonacci: $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2$, etc

Lemma2: $a \geq b \geq 1, m = M(a, b)$

Then $a \geq f_{m+2}, b \geq f_{m+1}$

The idea behind it is this: $a \geq b + r$

Which means if we look at all the values that we use in calls to Euclid(a, b), they grow like which numbers?

Euclid Algorithm

$a \bmod b = r, \quad a = qb + r$

Algorithm Euclid(a, b): $// \, a \geq b \geq 1$
  $r = a \bmod b$ *
  if $r = 0$: return b
  if $r \geq 1$:
    return Euclid(b, r) $// \, b \geq r \geq 1$

$M(a, b) =$ number of times line * is executed.

Fibonacci: $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2$, etc

Lemma2: $a \geq b \geq 1, m = M(a, b)$
Then $a \geq f_{m+2}, b \geq f_{m+1}$

The idea behind it is this: $a \geq b + r$

Which means if we look at all the values that we use in calls to Euclid(a, b), they grow like which numbers?

$$a \geq b + r$$
$$f_n = f_{n-1} + f_{n-2}$$

So if $r \geq f_{n-2}$ and $b \geq f_{n-1}$ then $a \geq f_n$
Then the numbers in Euclid(a, b) grow *at least* as fast as the fibonnacci sequence

Euclid Algorithm

$a \bmod b = r, \quad a = qb + r$

Algorithm Euclid(a, b): // $a \geq b \geq 1$
  $r = a \bmod b$ *
  if $r = 0$: return b
  if $r \geq 1$:
    return Euclid(b, r) // $b \geq r \geq 1$

$M(a, b)$ = number of times line * is executed.

Fibonacci: $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2$, etc

Lemma2: $a \geq b \geq 1, m = M(a, b)$
Then $a \geq f_{m+2}, b \geq f_{m+1}$

Induction on m: Base case $m = 1$, no recursive call, $r = a \bmod b = 0$.

$a \geq b + 1 \geq 2 = f_3$ which is true
$b \geq 1 = f_2$ which is true

Inductive Step: $m \geq 2$
Euclid(a,b):
  $r = a \bmod b \geq 1$
  Euclid(b, r)
  …recursive calls…

Euclid Algorithm

$a \bmod b = r, \quad a = qb + r$

Algorithm Euclid(a, b): // $a \geq b \geq 1$
  $r = a \bmod b$ *
  if $r = 0$: return b
  if $r \geq 1$:
    return Euclid(b, r) // $b \geq r \geq 1$

$M(a, b)$ = number of times line * is executed.

Fibonacci: $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2$, etc

Lemma2: $a \geq b \geq 1, m = M(a, b)$
Then $a \geq f_{m+2}, b \geq f_{m+1}$

Inductive Step: $m \geq 2$
Euclid(a,b):
  $r = a \bmod b \geq 1$ — 1 call to $a \bmod b$
  Euclid(b, r)
  ...recursive calls... — $m - 1$ calls to $a \bmod b$

Inductive Hypothesis:
$b \geq f_{m+1}$
$r \geq f_m$

$$a = qb + r \geq b + r \geq f_{m+1} + f_m = f_{m+2}$$

Euclid Algorithm

$a \bmod b = r, \quad a = qb + r$

Algorithm Euclid(a, b): // $a \geq b \geq 1$
  $r = a \bmod b$ *
  if $r = 0$: return b
  if $r \geq 1$:
    return Euclid(b, r) // $b \geq r \geq 1$

$M(a, b) =$ number of times line * is executed.

Fibonacci: $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2$, etc

Lemma2: $a \geq b \geq 1, m = M(a, b)$
Then $a \geq f_{m+2}, b \geq f_{m+1}$ → True

Lemma 3: $a \geq b \geq 1, M(a, b) \leq 1 + \log_\phi b$

$$\phi = \frac{1 + \sqrt{5}}{2}$$

if $a = b, M(a, b) = 1 \leq 1 + \log_\phi b$
if $a > b, M(a, b) = m$

$b \geq f_{m+1} \geq \phi^{m-1}$ (exercise using $\phi^2 = \phi + 1$)

$$\log_\phi b \geq m - 1$$
$$m \leq 1 + \log_\phi b$$