Divisibility

and Euclid’s Algorithm

1 Divisors

Definitions

Let $a, b, d \in \mathbb{Z}$ (most of the time we will have $a, b, d \in \mathbb{N}$)

- d is *divisor* of a , d *divides* a

$$d|a \leftrightarrow \exists k \in \mathbb{Z}.\ a = k \cdot d$$

- a positive divisor d of a different from a and 1 is called a *factor* of a
- $p \in \mathbb{N}$ is a *prime number* if it has no factors.
- $a \in \mathbb{N}$ is *composite* if it is not a prime number
- d is *common divisor* of a and b iff $d|a \wedge d|b$
- *greatest common divisor*

$$gcd(a, b) = \max\{d \in \mathbb{N} : d|a \wedge d|b\}$$

characterisation of gcd: extremely useful!!

Lemma 1. *Let* $a, b \in \mathbb{N}$ *and let* $L(a,b)$ *be the set of linear combinations of* a *and* b *with coefficients in* $\mathbb{Z}$.

$$L(a,b) = \{ax + by \ : \ x, y \in \mathbb{Z}\}$$

Then $gcd(a,b)$ *is the smallest positive integer in this set.*

$$gcd(a,b) = \min\{z \in L(a,b) \ : \ z > 0\}$$

characterisation of gcd: extremely useful!!

Lemma 1. *Let $a,b \in \mathbb{N}$ and let $L(a,b)$ be the set of linear combinations of a and b with coefficients in $\mathbb{Z}$.*

$$L(a,b) = \{ax+by \ : \ x,y \in \mathbb{Z}\}$$

Then $gcd(a,b)$ is the smallest positive integer in this set.

$$gcd(a,b) = \min\{z \in L(a,b) \ : \ z > 0\}$$

- $$0 < s = ax+by \quad \text{minimal}$$

$$\begin{aligned} a \bmod s &= a - qs \text{ with } q \in \mathbb{Z} \\ &= a - q(ax+by) \\ &= a(1-qx) + b(-qy) \\ &< s \end{aligned}$$

$$a \bmod s = 0 \quad (\text{minimality of } s)$$

$$s|a \quad , \quad s|b \text{ similarly} \quad , \quad gcd(a,b) \geq s$$

characterisation of gcd: extremely useful!!

Lemma 1. *Let* $a, b \in \mathbb{N}$ *and let* $L(a,b)$ *be the set of linear combinations of* a *and* b *with coefficients in* $\mathbb{Z}$.

$$L(a,b) = \{ax + by \ : \ x, y \in \mathbb{Z}\}$$

Then $gcd(a,b)$ *is the smallest positive integer in this set.*

$$gcd(a,b) = \min\{z \in L(a,b) \ : \ z > 0\}$$

- Let $d = gcd(a,b)$

$$d|a \ , \ d|b \ , \ s = ax + by \ \rightarrow d|s \ , \ gcd(a,b) \leq s$$

- $$0 < s = ax + by \quad \text{minimal}$$

$$\begin{aligned} a \bmod s &= a - qs \text{ with } q \in \mathbb{Z} \\ &= a - q(ax + by) \\ &= a(1 - qx) + b(-qy) \\ &< s \end{aligned}$$

$$a \bmod s = 0 \quad (\text{minimality of } s)$$

$$s|a \quad , \quad s|b \text{ similarly} \quad , \quad gcd(a,b) \geq s$$

characterisation of gcd: extremely useful!!

Lemma 1. *Let* $a,b \in \mathbb{N}$ *and let* $L(a,b)$ *be the set of linear combinations of* a *and* b *with coefficients in* $\mathbb{Z}$*.*

$$L(a,b) = \{ax+by \ : \ x,y \in \mathbb{Z}\}$$

Then $gcd(a,b)$ *is the smallest positive integer in this set.*

$$gcd(a,b) = \min\{z \in L(a,b) \ : \ z > 0\}$$

- $$0 < s = ax+by \quad \text{minimal}$$

$$\begin{aligned} a \bmod s &= a - qs \text{ with } q \in \mathbb{Z} \\ &= a - q(ax+by) \\ &= a(1-qx) + b(-qy) \\ &< s \end{aligned}$$

$$a \bmod s = 0 \quad (\text{minimality of } s)$$

$$s|a \quad , \quad s|b \text{ similarly} \quad , \quad gcd(a,b) \geq s$$

- Let $d = gcd(a,b)$

$$d|a \ , \ d|b \ , \ s = ax+by \ \rightarrow d|s \ , \ gcd(a,b) \leq s$$

Lemma 2.

$$d|a \ , \ d|b \ \rightarrow d|gcd(a,b)$$

$$\text{lemma 1} \ \rightarrow \ gcd(a,b) = ax+by$$

2 Relatively Prime Integers

Lemma 3. *If a and b are relatively prime to p, then so is their product.*

$$gcd(a,p) = 1\ ,\ gcd(b,p) = 1\ \rightarrow\ gcd(ab,p) = 1$$

2 Relatively Prime Integers

Lemma 3. *If a and b are relatively prime to p, then so is their product.*

$$gcd(a,p) = 1\ ,\ gcd(b,p) = 1\ \rightarrow\ gcd(ab,p) = 1$$

Lemma 1 $\rightarrow$

$$\begin{aligned} ax + py &= 1 \\ bx' + py' &= 1 \end{aligned}$$

2 Relatively Prime Integers

Lemma 3. *If a and b are relatively prime to p, then so is their product.*

$$gcd(a,p) = 1\ ,\ gcd(b,p) = 1\ \rightarrow\ gcd(ab,p) = 1$$

Lemma 1 $\rightarrow$

$$\begin{aligned} ax + py &= 1 \\ bx' + py' &= 1 \end{aligned}$$

multiply:

$$ab(xx') + p(ybx' + y'ax + pyy') = 1$$

Lemma 1 $\rightarrow$

$$gcd(ab,p) = 1$$

3 Euclids Algorithm

gcd recursion theorem:

Lemma 4.

$$a \geq 0\ ,\ b > 0 \rightarrow gcd(a,b) = gcd(b, a \mod b)$$

3 Euclids Algorithm

gcd recursion theorem:

Lemma 4.

$$a \geq 0\ ,\ b > 0 \rightarrow gcd(a,b) = gcd(b, a \mod b)$$

- claim: $gcd(a,b) | gcd(b, a \mod b)$. Let $d = gcd(a,b)$

$$a \text{ mod } b = a - qb \text{ with } q = \lfloor a/b \rfloor$$

$$d|a\ ,\ d|b\ \rightarrow gcd(a,b)|a \text{ mod } b$$

$$\text{lemma2}\ \rightarrow d|gcd(b, a \mod b)$$

3 Euclids Algorithm

gcd recursion theorem:

Lemma 4.

$$a \geq 0\ ,\ b > 0 \rightarrow gcd(a,b) = gcd(b, a \mod b)$$

- claim: $gcd(a,b)|gcd(b,a \mod b)$. Let $d = gcd(a,b)$

$$a \text{ mod } b = a - qb \text{ with } q = \lfloor a/b \rfloor$$

$$d|a\ ,\ d|b\ \rightarrow gcd(a,b)|a \text{ mod } b$$

$$\text{lemma2 } \rightarrow d|gcd(b, a \mod b)$$

- claim: $gcd(b,a \mod b)|gcd(a,b)$. Let $d = gcd(b,a \mod b)$

$$a = a \text{ mod } b + qb \text{ with } q = \lfloor a/b \rfloor$$

$$d|b\ ,\ d|a \text{ mod } b\ \rightarrow d|a$$

$$\text{lemma2 } \rightarrow d|gcd(a,b)$$

3 Euclids Algorithm

gcd recursion theorem:

Lemma 4.
$$a \geq 0\ ,\ b > 0 \rightarrow gcd(a,b) = gcd(b, a \mod b)$$

- claim: $gcd(a,b)|gcd(b,a \mod b)$. Let $d = gcd(a,b)$

$$a \bmod b = a - qb \text{ with } q = \lfloor a/b \rfloor$$

$$d|a\ ,\ d|b\ \rightarrow gcd(a,b)|a \bmod b$$

$$\text{lemma2}\ \rightarrow d|gcd(b,a \mod b)$$

- claim: $gcd(b,a \mod b)|gcd(a,b)$. Let $d = gcd(b,a \mod b)$

$$a = a \bmod b + qb \text{ with } q = \lfloor a/b \rfloor$$

$$d|b\ ,\ d|a \bmod b\ \rightarrow d|a$$

$$\text{lemma2}\ \rightarrow d|gcd(a,b)$$

Euclid's algorithm

```
eucl(a,b):
if b==0 {return a} else {return eucl(b, a amod b)}
```

Example

$$\begin{aligned} eucl(30,21) &= eucl(21,9) \\ &= eucl(9,3) \\ &= 3 \end{aligned}$$

3 Euclids Algorithm

Euclid's algorithm

```
eucl(a,b):
if b==0 {return a} else {return eucl(b, a amod b)}
```

Example

$$\begin{aligned} eucl(30,21) &= eucl(21,9) \\ &= eucl(9,3) \\ &= 3 \end{aligned}$$

run time: Fibonacci numbers

$$\begin{aligned} F_1 &= 1 \\ F_2 &= 1 \\ F_{k+2} &= F_k + F_{k+1} \end{aligned}$$

Lemma 5. *Let $a > b \geq 1$ and eucl(a,b) makes $k \geq 1$ recursive calls. Then*

$$a \geq F_{k+2}\ ,\ b \geq F_{k+1}$$

3 Euclids Algorithm

Euclid's algorithm

```
eucl(a,b):
if b==0 {return a} else {return eucl(b, a amod b)}
```

Example

$$\begin{aligned} eucl(30,21) &= eucl(21,9) \\ &= eucl(9,3) \\ &= 3 \end{aligned}$$

run time: Fibonacci numbers

$$\begin{aligned} F_1 &= 1 \\ F_2 &= 1 \\ F_{k+2} &= F_k + F_{k+1} \end{aligned}$$

Lemma 5. *Let $a > b \geq 1$ and eucl(a,b) makes $k \geq 1$ recursive calls. Then*

$$a \geq F_{k+2} \text{ , } b \geq F_{k+1}$$

- $k = 1$

$$b \geq 1 = F_2 \text{ ,} a > b \geq 1 \text{ , } a \geq 2 = F_3$$

3 Euclids Algorithm

Euclid's algorithm

```
eucl(a,b):
if b==0 {return a} else {return eucl(b, a amod b)}
```

Example

$$\begin{aligned} eucl(30,21) &= eucl(21,9) \\ &= eucl(9,3) \\ &= 3 \end{aligned}$$

run time: Fibonacci numbers

$$\begin{aligned} F_1 &= 1 \\ F_2 &= 1 \\ F_{k+2} &= F_k + F_{k+1} \end{aligned}$$

Lemma 5. *Let $a > b \geq 1$ and eucl(a,b) makes $k \geq 1$ recursive calls. Then*

$$a \geq F_{k+2} \ , \ b \geq F_{k+1}$$

- $k = 1$

$$b \geq 1 = F_2 \ , a > b \geq 1 \ , \ a \geq 2 = F_3$$

- $k - 1 \to k$

$eucl(a,b)$ calls $eucl(b, a \bmod b)$, which makes $k-1$ recursive calls

$$\to b > 0$$

$$a \bmod b < b \ , \ IH \to a \bmod b \geq F_k \ , \ b > F_{k+1}$$

$$\begin{aligned} a \bmod b &= a - qb \quad \text{with } q = \lfloor a/b \rfloor \geq 1 \\ a &= a \bmod b + qb \\ &\geq a \bmod b + b \\ &\geq F_k + F_{k+1} \\ &= F_{k+2} \end{aligned}$$

3 Euclids Algorithm

Euclid's algorithm

```
eucl(a,b):
if b==0 {return a} else {return eucl(b, a amod b)}
```

Example

$$\begin{aligned} eucl(30,21) &= eucl(21,9) \\ &= eucl(9,3) \\ &= 3 \end{aligned}$$

run time: Fibonacci numbers

$$\begin{aligned} F_1 &= 1 \\ F_2 &= 1 \\ F_{k+2} &= F_k + F_{k+1} \end{aligned}$$

Lemma 5. *Let $a > b \geq 1$ and eucl(a,b) makes $k \geq 1$ recursive calls. Then*

$$a \geq F_{k+2} \text{ , } b \geq F_{k+1}$$

- $k = 1$

$$b \geq 1 = F_2 \text{ ,} a > b \geq 1 \text{ , } a \geq 2 = F_3$$

- $k - 1 \to k$

$eucl(a,b)$ calls $eucl(b, a \bmod b)$, which makes $k-1$ recursive calls

$$\to b > 0$$

$$a \bmod b < b \text{ , } IH \to a \bmod b \geq F_k \text{ , } b > F_{k+1}$$

$$\begin{aligned} a \bmod b &= a - qb \quad \text{with } q = \lfloor a/b \rfloor \geq 1 \\ a &= a \bmod b + qb \\ &\geq a \bmod b + b \\ &\geq F_k + F_{k+1} \\ &= F_{k+2} \end{aligned}$$

Cor: $b < F_{k+1}$ => fewer than k calls.

$$a = \langle u \rangle \text{ ,} b = \langle v \rangle \text{ , } u, v \in \mathbb{B}^\beta$$

Since $F_k \approx \frac{\left(\frac{1+\sqrt{5}}{2}\right)^k}{\sqrt{5}}$

=> # of recursive calls is $O(\lg \beta)$

$$\to O(\beta) \quad \text{basic arithmetic operations}$$

extended Euclidean algorithm

computes also indices *x* and *y*

ext-eucl(a,b):

if b==0 {return (a,1,0) }

else { (d’,x’,y) = ext-eucl(b, a mod b) };

(d,x,y) = (d’,y’,x’- $\lfloor a/b \rfloor y'$);

return (d,x,y)

extended Euclidean algorithm

computes also indices x and y

ext-eucl(a,b):

if b==0 {return (a,1,0) }

else { (d',x',y) = ext-eucl(b, a mod b) };

(d,x,y) = (d',y',x'- $\lfloor a/b \rfloor y'$);

return (d,x,y)

correctness: to show (by induction of number of recursive calls)

$$(d,x,y) = ext - eucl(a,b) \;\rightarrow\; d = ax + by$$

- $b = 0$:

$$ext - eucl(a,0) = (a,1,0)\;,\; d = a = a \cdot 1 + b \cdot 0$$

extended Euclidean algorithm

computes also indices x and y

ext-eucl(a,b):

if b==0 {return (a,1,0) }

else { (d',x',y) = ext-eucl(b, a mod b) };

(d,x,y) = (d',y',x'- $\lfloor a/b \rfloor y'$);

return (d,x,y)

correctness: to show (by induction of number of recursive calls)

$$(d,x,y) = ext-eucl(a,b) \;\rightarrow\; d = ax+by$$

- $b = 0$:

$$ext-eucl(a,0) = (a,1,0)\;,\; d = a = a\cdot 1 + b\cdot 0$$

- $b \neq 0$:

$$Lemma4 \;\rightarrow\; d = gcd(a,b) = d' = gcd(b, a \bmod b)$$

$$\begin{aligned} d &= d' \\ &= bx' + (a \bmod b)y' \\ &= bx' + (a - b\lfloor a/b \rfloor)y' \\ &= ay' + b(x' - \lfloor a/b \rfloor y') \\ &= ax + by \end{aligned}$$

extended Euclidean algorithm

computes also indices x and y

ext-eucl(a,b):

if b==0 {return (a,1,0) }

else { (d',x',y') = ext-eucl(b, a mod b) };

(d,x,y) = (d',y',x'- $\lfloor a/b \rfloor y'$);

return (d,x,y)

correctness: to show (by induction of number of recursive calls)

$$(d,x,y) = ext-eucl(a,b) \rightarrow d = ax+by$$

- $b = 0$:

$$ext-eucl(a,0) = (a,1,0) \text{ , } d = a = a \cdot 1 + b \cdot 0$$

- $b \neq 0$:

$$Lemma4 \rightarrow d = gcd(a,b) = d' = gcd(b, a \bmod b)$$

$$\begin{aligned} d &= d' \\ &= bx' + (a \bmod b)y' \\ &= bx' + (a - b\lfloor a/b \rfloor)y' \\ &= ay' + b(x' - \lfloor a/b \rfloor y') \\ &= ax + by \end{aligned}$$

example:

a	b	$\lfloor a/b \rfloor$	d	x	y
99	78	1	3	-11	14
78	21	3	3	3	-11
21	15	1	3	-2	3
15	6	2	3	1	-2
6	3	2	3	0	1
3	0		3	1	0

Table 1: Example of ext-eucl(99,78). To 'run' the algorithm by hand first fill the left three columns downward, then fill the right three columns upwards.