Discrete Algebraic Structures

WiSe 2025/2026

Prof. Dr. Antoine Wiehe
Research Group for Theoretical Computer Science

Theorem. The number of partitions of $\{1, \ldots, n\}$ into k parts is

$$\frac{1}{k!} \sum_{r=0}^{k} (-1)^r \binom{k}{r} (k-r)^n$$

Any idea how big this is?

Theorem. The number of partitions of $\{1, \ldots, n\}$ into k parts is

$$\frac{1}{k!}\sum_{r=0}^{k}(-1)^r\binom{k}{r}(k-r)^n$$

Any idea how big this is?

How about k^n?

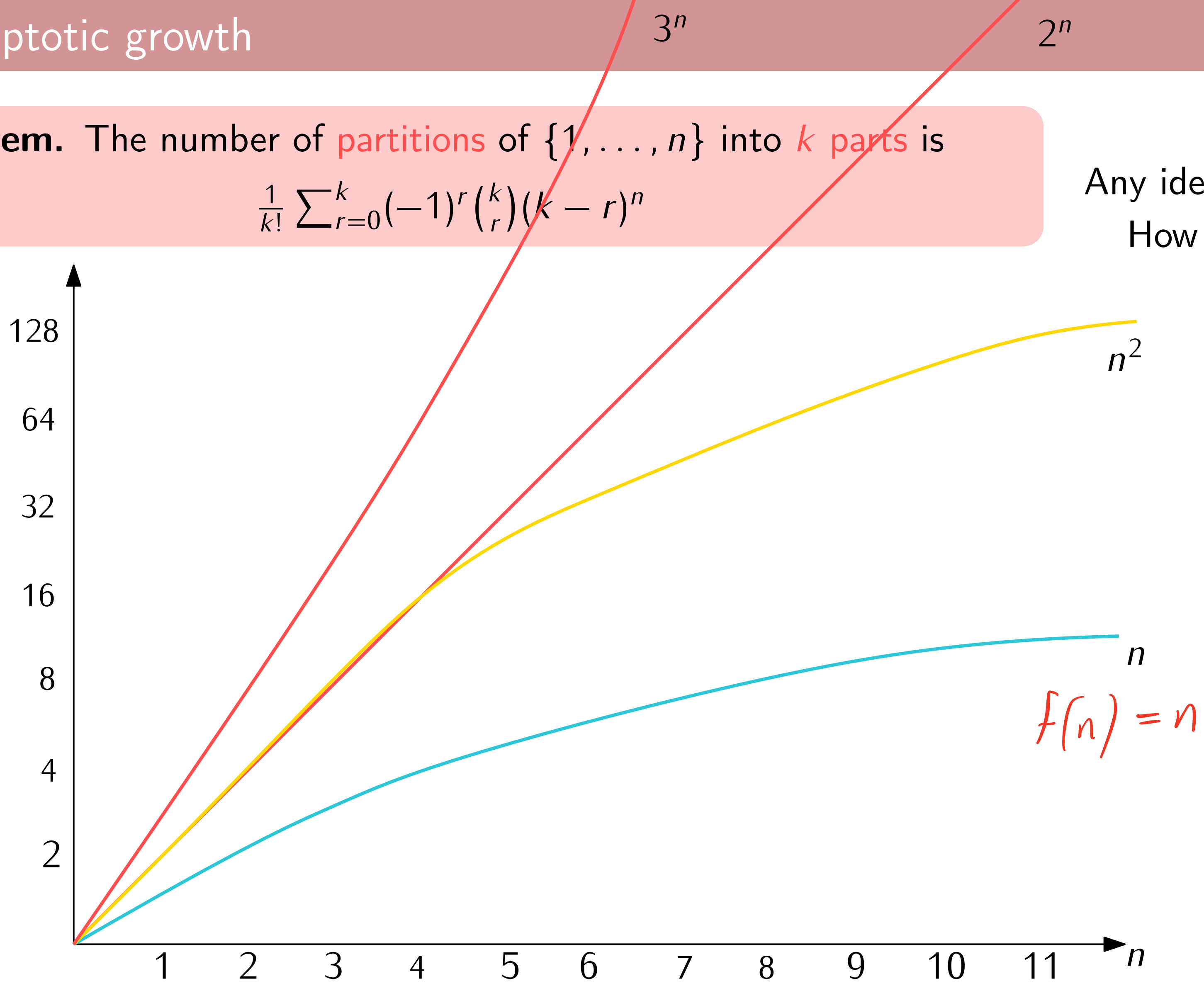
$$(-1)^0 \times \binom{k}{0} \times k^n$$

$$1 \times 1 \times k^n$$

Theorem. The number of partitions of $\{1, \ldots, n\}$ into k parts is
$$\frac{1}{k!} \sum_{r=0}^{k} (-1)^r \binom{k}{r} (k-r)^n$$

Any idea how big this is?

How about k^n?



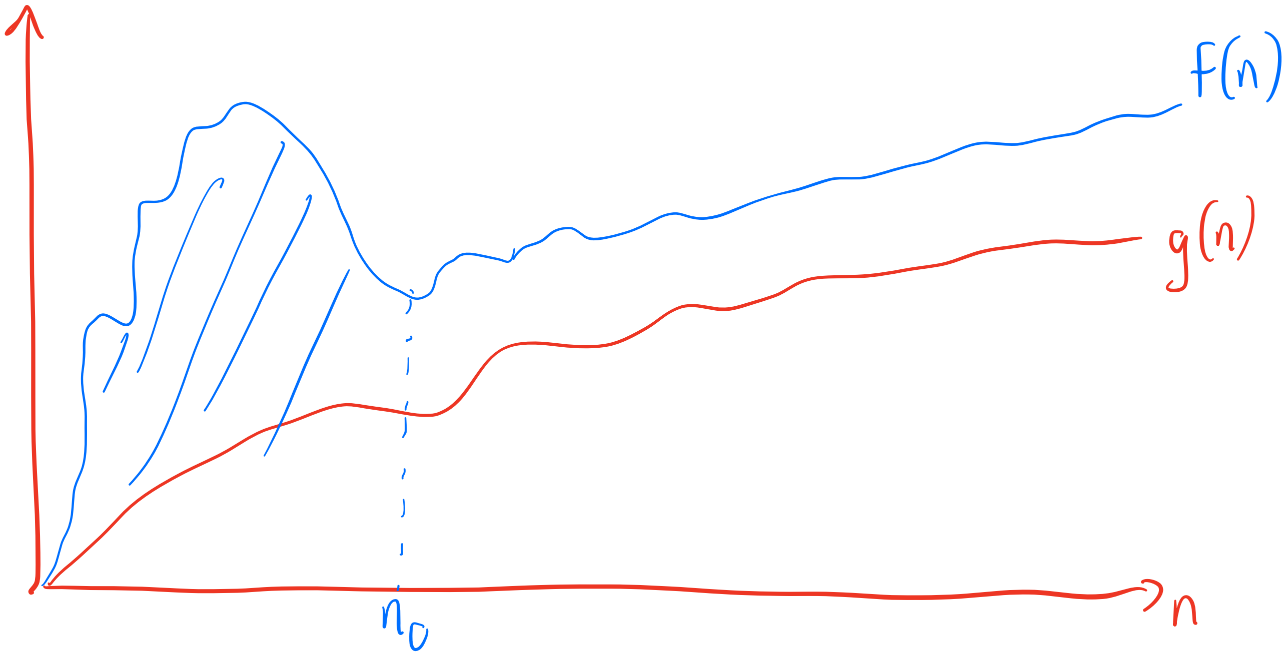
Theorem. The number of partitions of $\{1, \ldots, n\}$ into k parts is

$$\frac{1}{k!} \sum_{r=0}^{k} (-1)^r \binom{k}{r} (k-r)^n$$

Any idea how big this is?

How about k^n?

Definition. Let $f, g \colon \mathbb{N} \to \mathbb{N}$. We say $f \in O(g)$ if there are $C, n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $f(n) \leq C \cdot g(n)$.



Theorem. The number of partitions of $\{1, \dots, n\}$ into k parts is

$$\frac{1}{k!} \sum_{r=0}^{k} (-1)^r \binom{k}{r} (k-r)^n$$

Any idea how big this is?

How about k^n?

Definition. Let $f, g \colon \mathbb{N} \to \mathbb{N}$. We say $f \in O(g)$ if there are $C, n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $f(n) \leq C \cdot g(n)$.

Example. $4n^2 + n + 1 \in O(n^2)$

$f(n) = 4n^2 + n + 1$

$g(n) = n^2$

$f \in O(g)$: $n \leqslant n^2$

$1 \leqslant n^2$

$f(n) = 4n^2 + n + 1 \leq 4n^2 + n^2 + n^2 = 6 \cdot n^2 = 6 \cdot g(n)$

Theorem. The number of partitions of $\{1, \ldots, n\}$ into k parts is

$$\frac{1}{k!}\sum_{r=0}^{k}(-1)^r\binom{k}{r}(k-r)^n$$

Any idea how big this is?

How about k^n?

Definition. Let $f, g\colon \mathbb{N} \to \mathbb{N}$. We say $f \in O(g)$ if there are $C, n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $f(n) \leq C \cdot g(n)$.

Exercise. Prove that the relation $\{(f,g) \mid f \in O(g)\}$ is a quasiorder.

Reflexivity: $f \in O(f)$ for every f?

Transitivity: For all $f, g, h\colon \mathbb{N} \to \mathbb{N}$: if $f \in O(g)$ and $g \in O(h)$ then $f \in O(h)$.

Theorem. The number of partitions of $\{1, \dots, n\}$ into k parts is

$$\frac{1}{k!}\sum_{r=0}^{k}(-1)^r\binom{k}{r}(k-r)^n$$

Any idea how big this is?

How about k^n?

Definition. Let $f, g \colon \mathbb{N} \to \mathbb{N}$. We say $f \in O(g)$ if there are $C, n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $f(n) \leq C \cdot g(n)$.

Exercise. Prove that the relation $\{(f, g) \mid f \in O(g)\}$ is a quasiorder.

	Order matters	Order does not matter
Replacement	n^k	$\binom{n+k-1}{n-1}$
No replacement	$n^{\underline{k}}$	$\frac{n!}{k!(n-k)!}$

Number of surjective functions:
(from a set of size n to a set of size k)

$$\sum_{r=0}^{k}(-1)^r\binom{k}{r}(k-r)^n$$

Number of partitions:
(of a set of size n into k parts)

$$\frac{1}{k!}\sum_{r=0}^{k}(-1)^r\binom{k}{r}(k-r)^n$$

Theorem. The number of partitions of $\{1, \dots, n\}$ into k parts is

$$\frac{1}{k!}\sum_{r=0}^{k}(-1)^r\binom{k}{r}(k-r)^n$$

Any idea how big this is?

How about k^n?

Definition. Let $f, g\colon \mathbb{N} \to \mathbb{N}$. We say $f \in O(g)$ if there are $C, n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $f(n) \leq C \cdot g(n)$.

Exercise. Prove that the relation $\{(f, g) \mid f \in O(g)\}$ is a quasiorder.

	Order matters	Order does not matter
Replacement	$n^k \in O(n^k)$	$\binom{n+k-1}{n-1} \in O(n^k)$
No replacement	$n^{\underline{k}} \in O(n^k)$	$\frac{n!}{k!(n-k)!} \in O(n^k)$

Number of surjective functions:
(from a set of size n to a set of size k)

$$O\left(\sum_{r=0}^{k}(-1)^r\binom{k}{r}(k-r)^n\right) \ni k^n$$

Number of partitions:
(of a set of size n into k parts)

$$O\left(\frac{1}{k!}\sum_{r=0}^{k}(-1)^r\binom{k}{r}(k-r)^n\right) \ni k^n$$

Theorem. The number of partitions of $\{1, \ldots, n\}$ into k parts is

$$\frac{1}{k!} \sum_{r=0}^{k} (-1)^r \binom{k}{r} (k-r)^n$$

Any idea how big this is?

How about k^n?

Definition. Let $f, g \colon \mathbb{N} \to \mathbb{N}$. We say $f \in O(g)$ if there are $C, n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $f(n) \leq C \cdot g(n)$.

Exercise. Prove that the relation $\{(f,g) \mid f \in O(g)\}$ is a quasiorder.

	Order matters	Order does not matter
Replacement	n^k	$\binom{n+k-1}{n-1} \in O(n^k)$
No replacement	$n^{\underline{k}} \in O(n^k)$	$\frac{n!}{k!(n-k)!} \in O(n^k)$

Take away message:
$O(n^k)$ is "ok" for algorithms if k small
Don't ever try $O(k^n)$, even if k is small

Number of surjective functions:
(from a set of size n to a set of size k)

$$O\left(\sum_{r=0}^{k} (-1)^r \binom{k}{r} (k-r)^n \right) \ni k^n$$

Number of partitions:
(of a set of size n into k parts)

$$O\left(\frac{1}{k!} \sum_{r=0}^{k} (-1)^r \binom{k}{r} (k-r)^n \right) \ni k^n$$

$f(n)$: runtime (say, in seconds) of a program when the input has size n

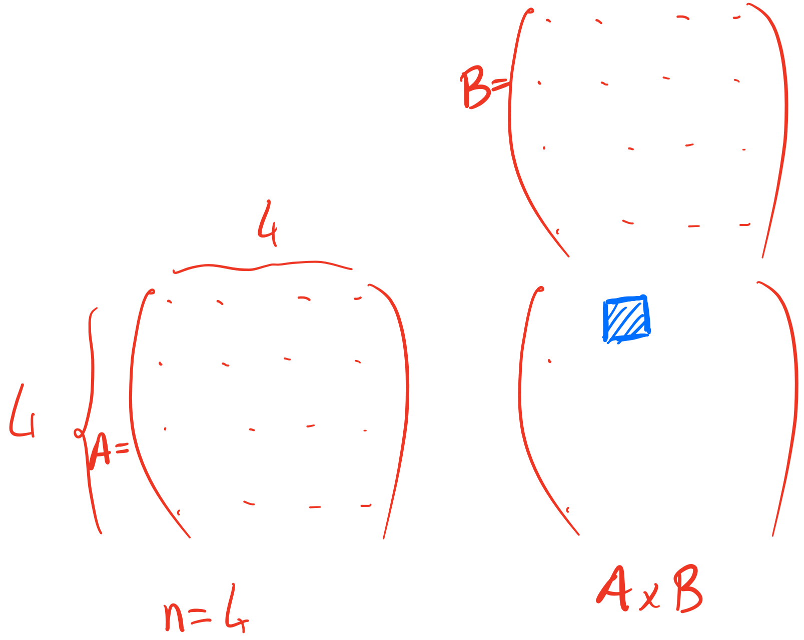
$f(n)$: runtime (say, in seconds) of a program when the input has size n

Efficient algorithm: when $f \in O(n^k)$ for $k \in \mathbb{N}$

- $f \in O(\log(n))$: locate an item in a sorted array

$\log(n) \in O(\sqrt{n})$
$\in O(n^{0.1})$

- $f \in O(n)$: find shortest path in a directed graph, find smallest element/biggest element/median in an array
- $f \in O(n \log(n))$: sort an array $\in O(n^{1.1})$
- $f \in O(n^3)$: compute the multiplication of two matrices of size n



$f(n)$: runtime (say, in seconds) of a program when the input has size n

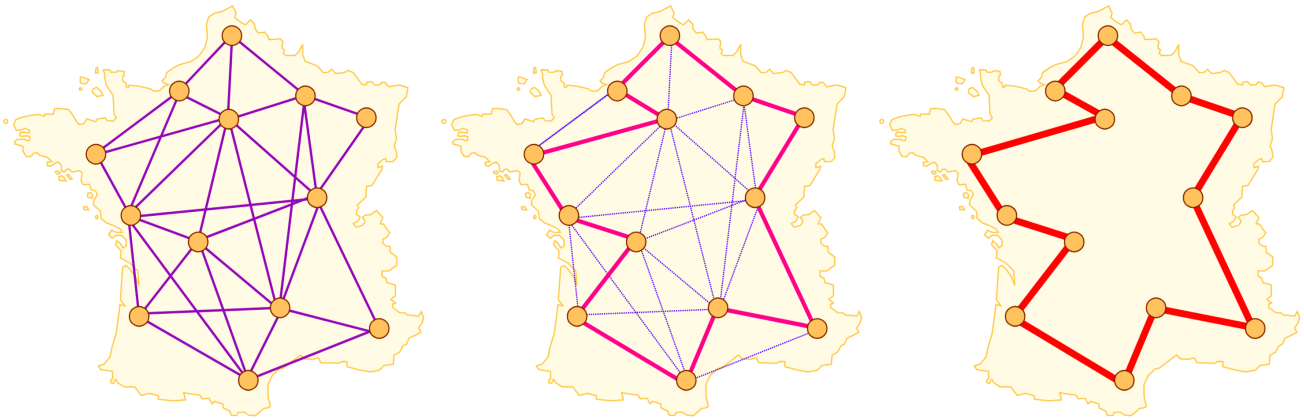
Efficient algorithm: when $f \in O(n^k)$ for $k \in \mathbb{N}$

- $f \in O(\log(n))$: locate an item in a sorted array
- $f \in O(n)$: find shortest path in a directed graph, find smallest element/biggest element/median in an array
- $f \in O(n\log(n))$: sort an array
- $f \in O(n^3)$: compute the multiplication of two matrices of size n

Problems where **no efficient algorithm** is known:

- $O(2^n)$: problem of finding a satisfying assignment for a propositional formula, optimal clustering of a set of n points into two clusters
- $O(n!)$: finding optimal tour visiting cities

$(p \vee q \vee r) \wedge (\neg p \vee r \vee s) \wedge (\neg r \vee t \vee \neg s)$



What to know: counting

- Countable sets, uncountable sets
- Countable sets = what can be represented exactly on a computer
- Combinatorial proofs as a way to prove equalities/inequalities about numbers using functions

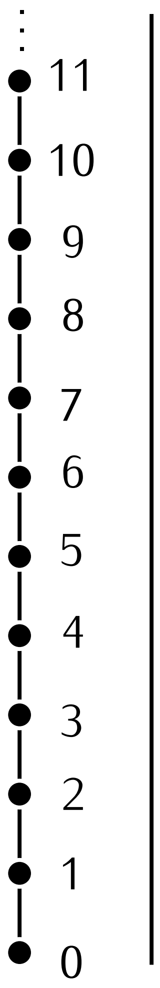
$$\begin{array}{rcl} \text{Injection } A \to B & \leftrightarrow & |A| \leq |B| \\ \text{Surjection } A \to B & \leftrightarrow & |A| \geq |B| \\ \text{Bijection } A \to B & \leftrightarrow & |A| = |B| \end{array}$$

- Drawing a tuple/(multi)set with/without replacement
  n = size of the set we are drawing from
  k = number of draws
- How to use double counting
- The inclusion-exclusion principle for 2 and 3 sets
- How to apply the pigeonhole principle

	Order matters	Order does not matter
Replacement	n^k	$\binom{n+k-1}{n-1}$
No replacement	$n^{\underline{k}}$	$\frac{n!}{k!(n-k)!}$

Elementary Number Theory

$$R = \{(n, m) \in \mathbb{N}_0^2 \mid n \leq m\}$$

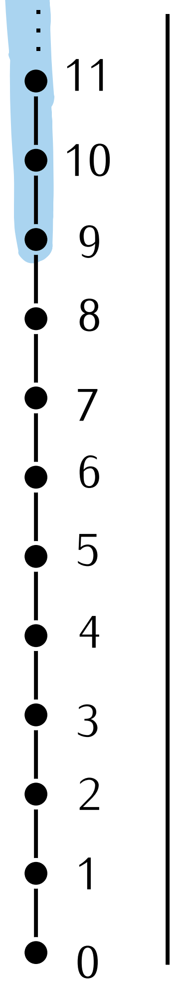


$$R = \{(n, m) \in \mathbb{N}_0^2 \mid n \leq m\}$$

- Linear
- Minimal elements: 0
- Maximal elements: no maximal element
- $n \wedge m$ always exists: $\min(n,m)$
- $n \vee m$ always exists: $\max(n,m)$

$7 \vee 9 = 9$

$7 \vee 10 = 10$

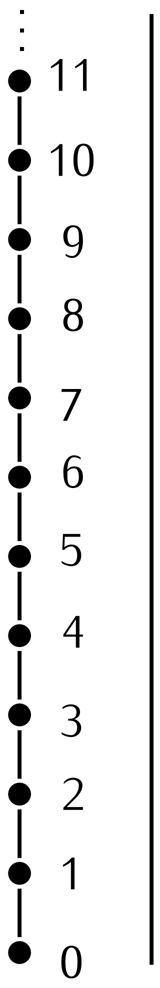


$$R = \{(n, m) \in \mathbb{N}_0^2 \mid n \leq m\}$$

- Linear
- Minimal elements: 0
- Maximal elements: None
- $n \wedge m$ always exists: $\min(n, m)$
- $n \vee m$ always exists: $\max(n, m)$

Every number n can be written (uniquely!) as $1 + 1 + 1 + \dots$

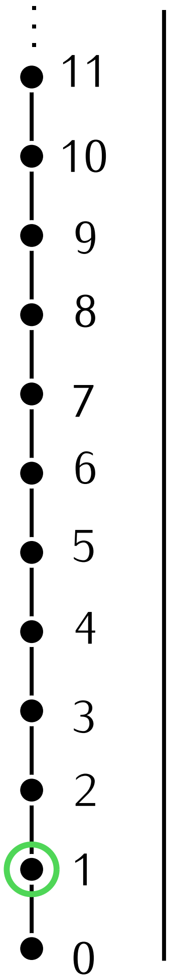
$4 = 1 + 1 + 1 + 1$



$$R = \{(n, m) \in \mathbb{N}_0^2 \mid n \leq m\}$$

- Linear
- Minimal elements: 0
- Maximal elements: None
- $n \wedge m$ always exists: $\min(n, m)$
- $n \vee m$ always exists: $\max(n, m)$

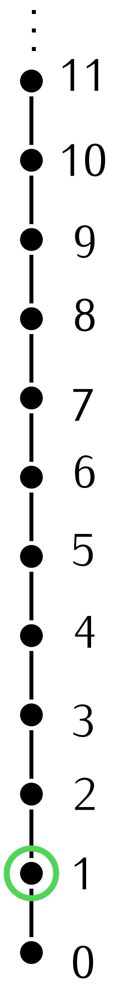
Every number n can be written (uniquely!) as $1 + 1 + 1 + \ldots$



$$R = \{(n,m) \in \mathbb{N}_0^2 \mid n \leq m\}$$

- Linear
- Minimal elements: 0
- Maximal elements: None
- $n \wedge m$ always exists: $\min(n,m)$
- $n \vee m$ always exists: $\max(n,m)$

Every number n can be written (uniquely!) as $1 + 1 + 1 + \ldots$

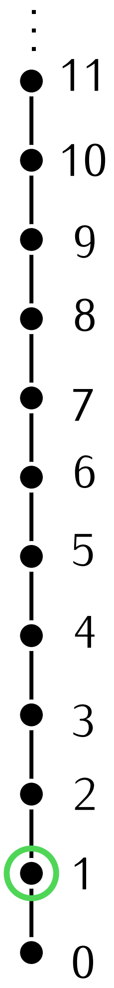


$$R = \{(n,m) \in \mathbb{N}_0^2 \mid n \text{ divides } m\}$$

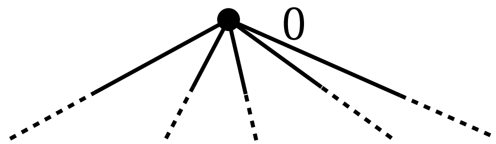
$R = \{(n, m) \in \mathbb{N}_0^2 \mid n \leq m\}$

- Linear
- Minimal elements: 0
- Maximal elements: None
- $n \wedge m$ always exists: $\min(n, m)$
- $n \vee m$ always exists: $\max(n, m)$

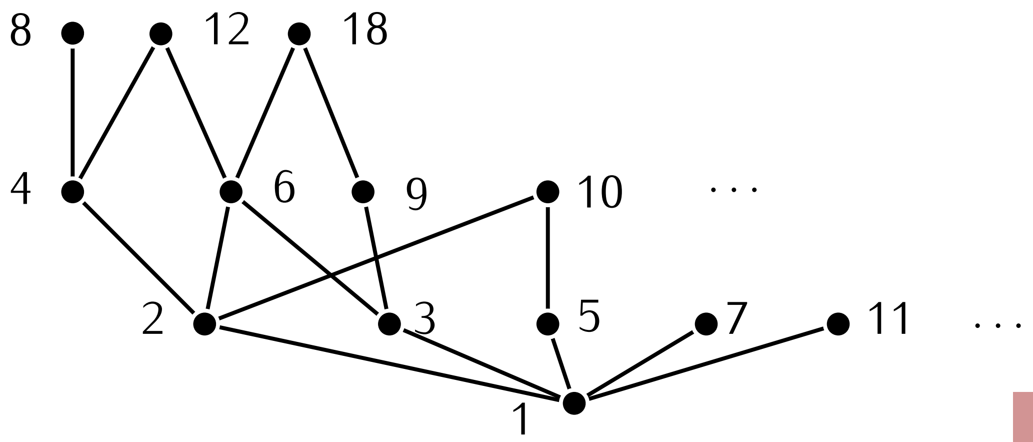
Every number n can be written (uniquely!) as $1 + 1 + 1 + \dots$



$R = \{(n, m) \in \mathbb{N}_0^2 \mid n \text{ divides } m\}$



- Linear? No
- Minimal elements? 1
- Maximal elements? 0
- $n \wedge m$?
- $n \vee m$?



Theorem. For all $a, d \in \mathbb{Z}$ such that $d \neq 0$, there exists a unique pair $(q, r) \in \mathbb{Z}^2$ such that:

- $a = q \cdot d + r$
- $r \in \{0, \ldots, |d| - 1\}$

Theorem. For all $a, d \in \mathbb{Z}$ such that $a \neq 0$, there exists a unique pair $(q, r) \in \mathbb{Z}^2$ such that:

- $a = q \cdot d + r$
- $r \in \{0, \ldots, |d| - 1\}$

$$\begin{array}{r|l}
131 & 9 \\ \cline{2-2}
-(9) & 14 \\
\hline
41 & \\
-(36) & \\
\hline
5 &
\end{array}$$

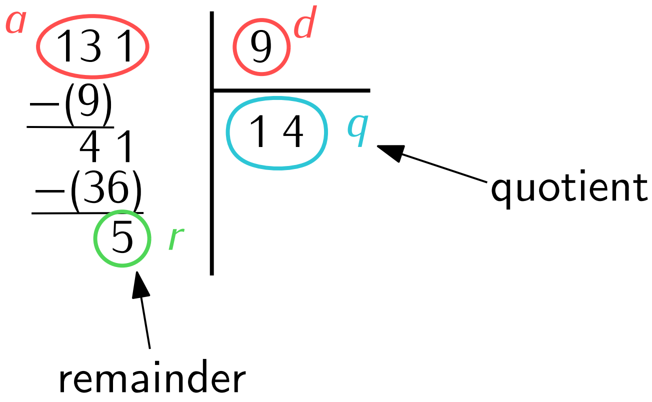
$131 = 14 \times 9 + 5$

q

$r \in \{0, .., 8\}$

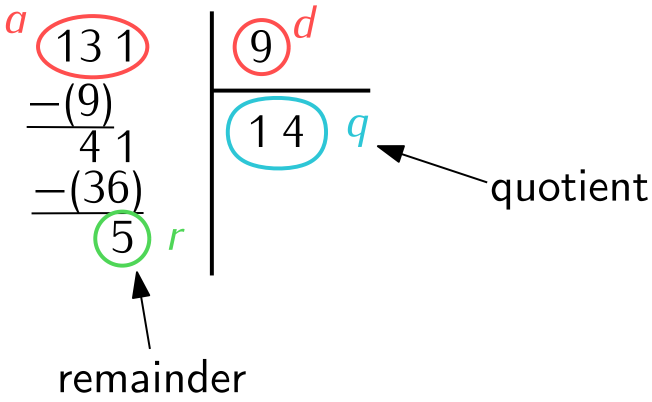
Theorem. For all $a, d \in \mathbb{Z}$ such that $a \neq 0$, there exists a unique pair $(q, r) \in \mathbb{Z}^2$ such that:

- $a = q \cdot d + r$
- $r \in \{0, \ldots, |d| - 1\}$



Theorem. For all $a, d \in \mathbb{Z}$ such that $a \neq 0$, there exists a unique pair $(q, r) \in \mathbb{Z}^2$ such that:

- $a = q \cdot d + r$
- $r \in \{0, \ldots, |d| - 1\}$

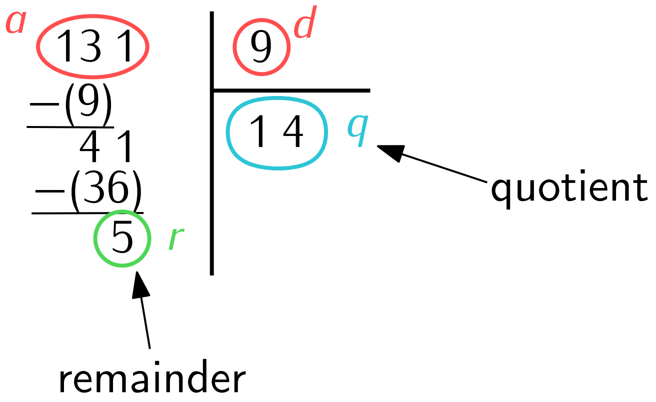


Almost the same as `divmod` in Python:

```
divmod(131,9) # (14,5)
divmod(10,-3) # (-4,-2)
```

Theorem. For all $a, d \in \mathbb{Z}$ such that $a \neq 0$, there exists a unique pair $(q, r) \in \mathbb{Z}^2$ such that:

- $a = q \cdot d + r$
- $r \in \{0, \dots, |d| - 1\}$



Almost the same as `divmod` in Python:

```
divmod(131,9) # (14,5)
divmod(10,-3) # (-4,-2)
```

- If remainder is 0: we say d divides a
  a is a multiple of d

Notation: $d \mid a$

Theorem. For all $a, d \in \mathbb{Z}$ such that $a \neq 0$, there exists a unique pair $(q, r) \in \mathbb{Z}^2$ such that:

- $a = q \cdot d + r$
- $r \in \{0, \ldots, |d| - 1\}$

$$\begin{array}{r|l} \overset{a}{131} & 9^{\,d} \\ \hline -(9) & 14 \;\; q \\ 41 & \\ -(36) & \\ 5 \;\; r & \end{array} \qquad \begin{array}{r|l} 860 & 9^{\,d} \\ \hline -(81) & 95 \\ 50 & \\ -(45) & \\ 5 & \end{array}$$

quotient

remainder

Almost the same as divmod in Python:

```
divmod(131,9) # (14,5)
divmod(10,-3) # (-4,-2)
```

$860 = 95 \times 9 + 5$

$\underbrace{(-3)\times(-4)}_{12}+(-2) = 10$

- If remainder is 0: we say d divides a
  a is a multiple of d

Notation: $d \mid a$

- Define $b \equiv_d b'$ by "they have the same remainder in the division by d"
  $131 \equiv_9 860$

What does 131 really *mean*?

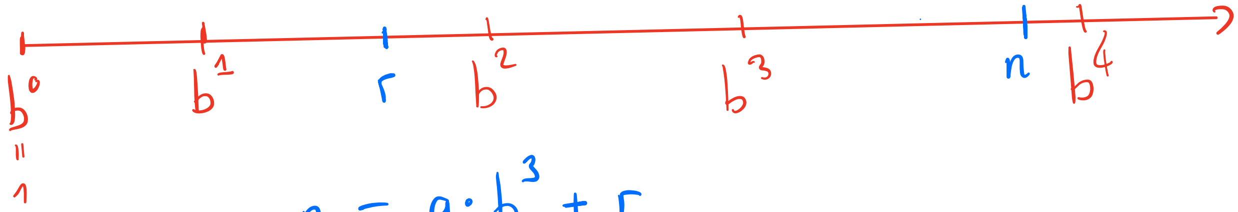
What does 132 really *mean*? $132 = 1 \times \underset{10^2}{\underset{\shortparallel}{100}} + 3 \times \underset{10^1}{\underset{\shortparallel}{10}} + 2 \times \underset{10^0}{\underset{\shortparallel}{1}}$

What does 131 really *mean*? $132 = 1 \times 100 + 3 \times 10 + 2 \times 1 = \sum_{i=0}^{2} n_i \times 10^i$

What does 131 really *mean*? $132 = 1 \times 100 + 3 \times 10 + 2 \times 1 = \sum_{i=0}^{2} n_i \times 10^i$

102 1(11)2

Theorem. Let $b \in \mathbb{N}$ be such that $b \geq 2$. For every $n \in \mathbb{N}_0$, there exists a unique sequence $n_0, \dots, n_k$ of numbers in $\{0, \dots, b-1\}$ such that $n = \sum_{i=0}^{k} n_i b^i$.



$$n = q \cdot b^3 + r$$

$q \in \{1, \dots, b-1\}$ $r \in \{0, \dots, b^3 - 1\}$

What does 131 really *mean*? $132 = 1 \times 100 + 3 \times 10 + 2 \times 1 = \sum_{i=0}^{2} n_i \times 10^i$

Theorem. Let $b \in \mathbb{N}$ be such that $b \geq 2$. For every $n \in \mathbb{N}_0$, there exists a unique sequence $n_0, \ldots, n_k$ of numbers in $\{0, \ldots, b-1\}$ such that $n = \sum_{i=0}^{k} n_i b^i$.

This is the decomposition of n in base b, written $(n_k, \ldots, n_1, n_0)_b$.

What does 131 really *mean*? $132 = {\color{red}1} \times 100 + {\color{red}3} \times 10 + {\color{red}2} \times 1 = \sum_{i=0}^{2} n_i \times 10^i$

Theorem. Let $b \in \mathbb{N}$ be such that $b \geq 2$. For every $n \in \mathbb{N}_0$, there exists a unique sequence $n_0, \ldots, n_k$ of numbers in $\{0, \ldots, b-1\}$ such that $n = \sum_{i=0}^{k} n_i b^i$.

This is the decomposition of n in base b, written $(n_k, \ldots, n_1, n_0)_b$.

- Computers use bits 0 and 1 $\rightsquigarrow$ base 2

  $(131)_{10} = (10000011)_2$

$$131 = 2^7 + 3 = 2^7 + 2^1 + 1 = 2^7 + 2^1 + 2^0$$

0	1	2	3	4	5	6	7
1	2	4	8	16	32	64	128

What does 131 really *mean*? $132 = 1 \times 100 + 3 \times 10 + 2 \times 1 = \sum_{i=0}^{2} n_i \times 10^i$

Theorem. Let $b \in \mathbb{N}$ be such that $b \geq 2$. For every $n \in \mathbb{N}_0$, there exists a unique sequence $n_0, \ldots, n_k$ of numbers in $\{0, \ldots, b-1\}$ such that $n = \sum_{i=0}^{k} n_i b^i$.

This is the decomposition of n in base b, written $(n_k, \ldots, n_1, n_0)_b$.

- Computers use bits 0 and 1 $\rightsquigarrow$ base 2
  $(131)_{10} = (10000011)_2$

0	1	2	3	4	5	6	7
1	2	4	8	16	32	64	128

What is the largest number that can be represented with 8 bits?
(So binary decomposition has length at most 8.)

$(b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)_2$

$(1\,1\,1\,1\,1\,1\,1\,1)_2 = 2^7 + 2^6 + 2^5 + \ldots + 2 + 1 = 255$

Writing numbers in different bases

What does 131 really *mean*? $132 = 1 \times 100 + 3 \times 10 + 2 \times 1 = \sum_{i=0}^{2} n_i \times 10^i$

Theorem. Let $b \in \mathbb{N}$ be such that $b \geq 2$. For every $n \in \mathbb{N}_0$, there exists a unique sequence $n_0, \ldots, n_k$ of numbers in $\{0, \ldots, b-1\}$ such that $n = \sum_{i=0}^{k} n_i b^i$.

This is the decomposition of n in base b, written $(n_k, \ldots, n_1, n_0)_b$.

- Computers use bits 0 and 1 $\rightsquigarrow$ base 2
  $(131)_{10} = (10000011)_2$
- Base 16 also very common: `commit 6d068c500bd1baede277a8c1f62d1a7b5d1a1d12`

$\mathcal{O}$

What does 131 really *mean*? $132 = {\color{red}1} \times 100 + {\color{red}3} \times 10 + {\color{red}2} \times 1 = \sum_{i=0}^{2} n_i \times 10^i$

Theorem. Let $b \in \mathbb{N}$ be such that $b \geq 2$. For every $n \in \mathbb{N}_0$, there exists a unique sequence $n_0, \ldots, n_k$ of numbers in $\{0, \ldots, b-1\}$ such that $n = \sum_{i=0}^{k} n_i b^i$.

This is the decomposition of n in base b, written $(n_k, \ldots, n_1, n_0)_b$.

- Computers use bits 0 and 1 $\rightsquigarrow$ base 2

  $(131)_{10} = (10000011)_2$

- Base 16 also very common: `commit 6d068c500bd1baede277a8c1f62d1a7b5d1a1d12`

  $= 622426021449319981362286421400854052354972589330$

  about 17% shorter to write numbers in base 16 instead of 10, 75% shorter than in base 2

  alphabet $0, \ldots, 9, a, b, c, d, e, f$

  10 13

What does 131 really *mean*? $132 = 1 \times 100 + 3 \times 10 + 2 \times 1 = \sum_{i=0}^{2} n_i \times 10^i$

Theorem. Let $b \in \mathbb{N}$ be such that $b \geq 2$. For every $n \in \mathbb{N}_0$, there exists a unique sequence $n_0, \ldots, n_k$ of numbers in $\{0, \ldots, b-1\}$ such that $n = \sum_{i=0}^{k} n_i b^i$.

This is the decomposition of n in base b, written $(n_k, \ldots, n_1, n_0)_b$.

- Computers use bits 0 and 1 $\rightsquigarrow$ base 2
  $(131)_{10} = (10000011)_2$
- Base 16 also very common: `commit 6d068c500bd1baede277a8c1f62d1a7b5d1a1d12`
  $= 622426021449319981362286421400854052354972589330$

  about 17% shorter to write numbers in base 16 instead of 10, 75% shorter than in base 2
  alphabet $0, \ldots, 9, a, b, c, d, e, f$

$$(bad)_{16} = b \times 16^2 + a \times 16^1 + d \times 16$$
$$= 11 \times 256 + 10 \times 16 + 13 = 2816 + 160 + 13 = 2989$$

What does 131 really *mean*? $132 = 1 \times 100 + 3 \times 10 + 2 \times 1 = \sum_{i=0}^{2} n_i \times 10^i$

Theorem. Let $b \in \mathbb{N}$ be such that $b \geq 2$. For every $n \in \mathbb{N}_0$, there exists a unique sequence $n_0, \ldots, n_k$ of numbers in $\{0, \ldots, b-1\}$ such that $n = \sum_{i=0}^{k} n_i b^i$.

This is the decomposition of n in base b, written $(n_k, \ldots, n_1, n_0)_b$.

- Computers use bits 0 and 1 $\rightsquigarrow$ base 2
  $(131)_{10} = (10000011)_2$
- Base 16 also very common: `commit 6d068c500bd1baede277a8c1f62d1a7b5d1a1d12`
  $= 622426021449319981362286421400854052354972589330$

  about 17% shorter to write numbers in base 16 instead of 10, 75% shorter than in base 2
  alphabet $0, \ldots, 9, a, b, c, d, e, f$

$$(bad)_{16} = b \times 16^2 + a \times 16^1 + d \times 16$$
$$= 11 \times 256 + 10 \times 16 + 13 = 2816 + 160 + 13 = 2989$$

- $\log_b(n)$: number in $[k, k+1)$ when $b^k \leq n < b^{k+1}$
  $$3 < log_{16}(2989) < 4$$

Definition. Let $\leq$ be an order on A, $S \subseteq A$, and $a \in A$. We say:

- a is a lower bound of S if for all $s \in S$, we have $a \leq s$
- a is a greatest lower bound of S if it is a lower bound and for every lower bound b of S, we have $b \leq a$.

We write $a \wedge b$ for the greatest lower bound of $\{a, b\}$, if it exists.

Definition. Let $S \subseteq \mathbb{N}_0$, and $d \in \mathbb{N}_0$. We say:

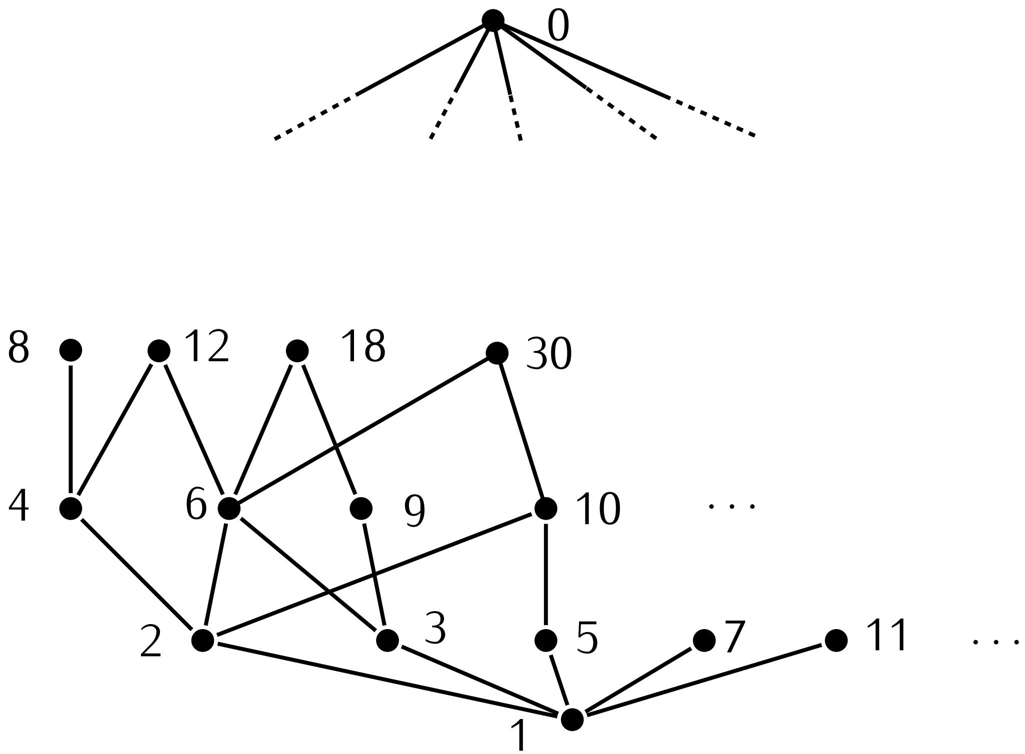
- d is a common divisor of S if for all $s \in S$, we have d divides s
- d is a greatest common divisor of S if it is a common divisor and for every common divisor d' of S, we have d' divides d

We write $a \wedge b$ for the greatest common divisor (gcd) of $\{a, b\}$.

Definition. Let $S \subseteq \mathbb{N}_0$, and $d \in \mathbb{N}_0$. We say:

- d is a common divisor of S if for all $s \in S$, we have d divides s
- d is a greatest common divisor of S if it is a common divisor and for every common divisor d' of S, we have d' divides d

We write $a \wedge b$ for the greatest common divisor (gcd) of $\{a, b\}$.



Definition. Let $S \subseteq \mathbb{N}_0$, and $d \in \mathbb{N}_0$. We say:

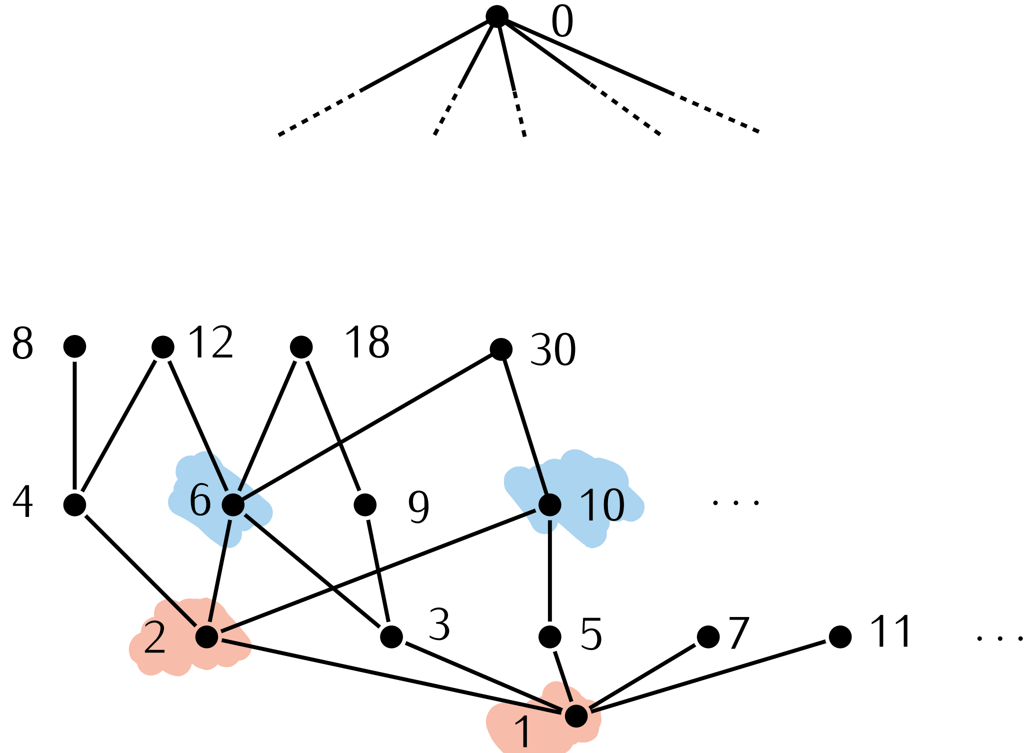
- d is a common divisor of S if for all $s \in S$, we have d divides s
- d is a greatest common divisor of S if it is a common divisor and for every common divisor d' of S, we have d' divides d

We write $a \wedge b$ for the greatest common divisor (gcd) of $\{a, b\}$.

What is $6 \wedge 10$?

- 2
- 4
- 6
- 10
- 16

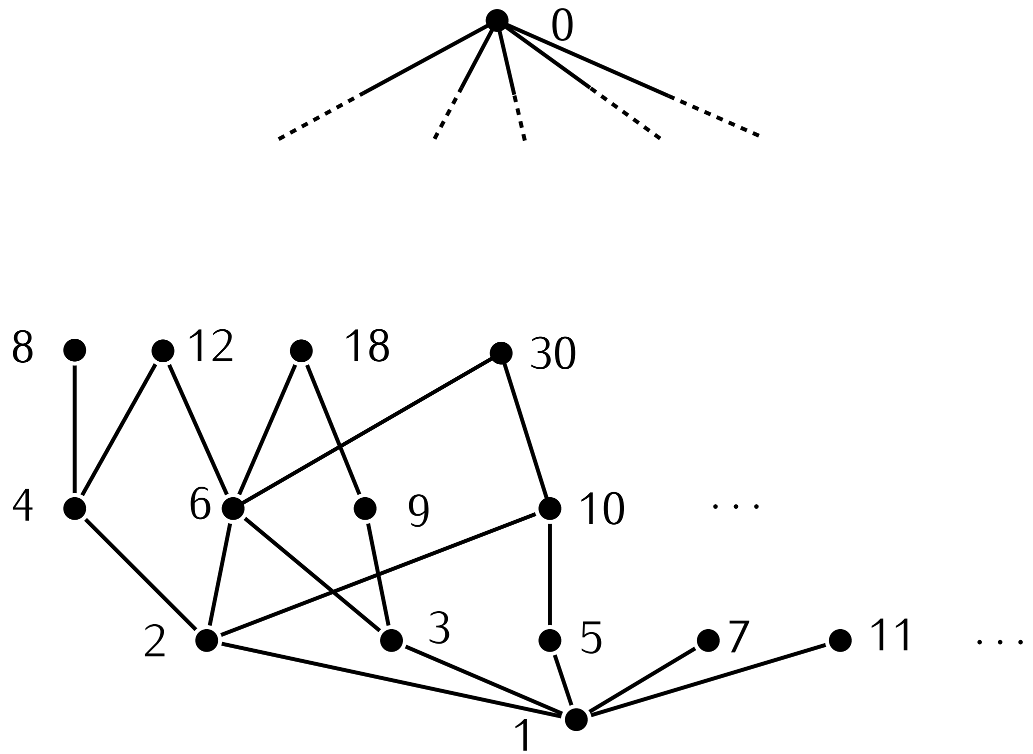
[redacted]



Definition. Let $S \subseteq \mathbb{N}_0$, and $d \in \mathbb{N}_0$. We say:

- m is a common multiple of S if for all $s \in S$, we have m is a multiple of s
- m is a lowest common multiple of S if it is a common multiple and for every common multiple m' of S, we have m divides m'

We write $a \vee b$ for the lowest common multiple (lcm) of $\{a, b\}$.

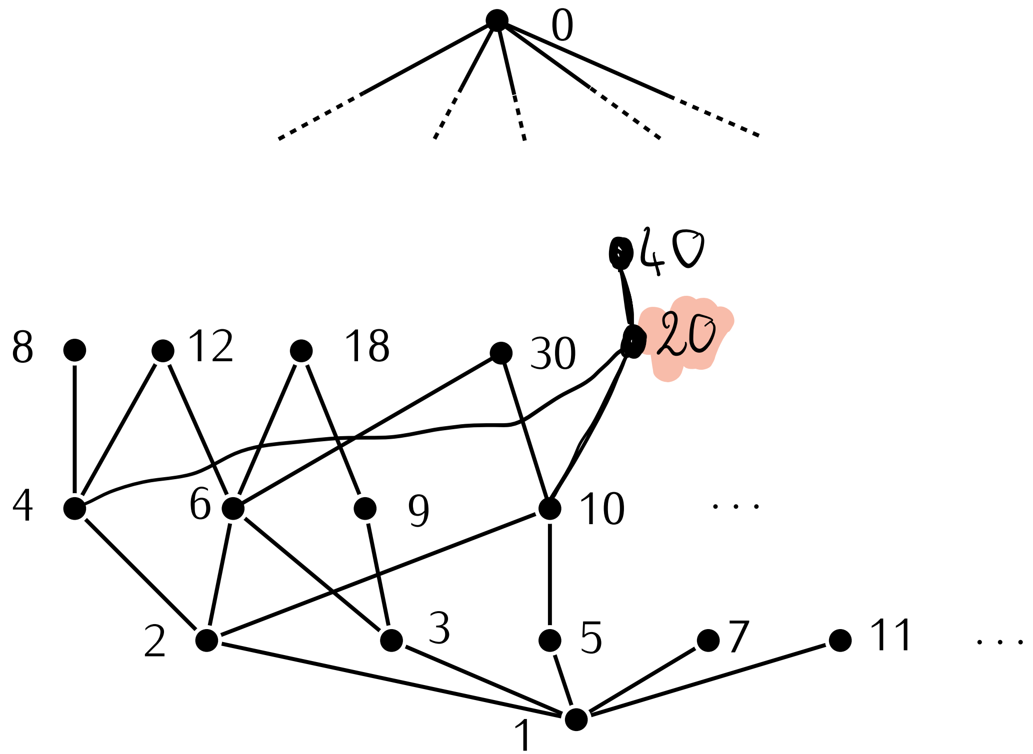


Definition. Let $S \subseteq \mathbb{N}_0$, and $d \in \mathbb{N}_0$. We say:

- m is a common multiple of S if for all $s \in S$, we have m is a multiple of s
- m is a lowest common multiple of S if it is a common multiple and for every common multiple m' of S, we have m divides m'

We write $a \vee b$ for the lowest common multiple (lcm) of $\{a, b\}$.

What is $4 \vee 10$?
(might not be on the picture!)



Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$)

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$) $\longrightarrow$ In particular, $\gcd(a, b)$ **always** exists!

Euclid
(probably lived around -300)

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Input: two numbers $a, b \in \mathbb{N}_0$

Output: $a \wedge b$ (or $\gcd(a, b)$) $\longrightarrow$ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 24$:

$\texttt{remainders} = (24, 7)$

b a

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$) $\longrightarrow$ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 24$:

$$24 = 3 \times 7 + 3$$

$$\texttt{remainders} = (24, 7)$$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$) $\longrightarrow$ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 24$:

$$24 = 3 \times 7 + 3$$

$$\texttt{remainders} = (24, \underset{b}{7}, \underset{a}{3})$$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a,b)$) $\longrightarrow$ In particular, $\gcd(a,b)$ **always** exists!

On input $a = 7, b = 24$:

$$24 = 3 \times 7 + 3$$
$$7 = 2 \times 3 + 1$$

$$\texttt{remainders} = (24, 7, 3)$$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$) ⟶ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 24$:

$$24 = 3 \times 7 + 3$$
$$7 = 2 \times 3 + 1$$

$$\texttt{remainders} = (24, 7, \underset{b}{3}, \underset{a}{1})$$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$) ⟶ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 24$:

$$24 = 3 \times 7 + 3$$
$$7 = 2 \times 3 + 1$$
$$3 = 3 \times 1 + 0$$
$$\texttt{remainders} = (24, 7, 3, 1)$$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$) $\longrightarrow$ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 24$:

$$\begin{aligned} 24 &= 3 \times 7 + 3 \\ 7 &= 2 \times 3 + 1 \\ 3 &= 3 \times 1 + 0 \end{aligned}$$

$\texttt{remainders} = (24, 7, 3, 1, 0)$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$) $\longrightarrow$ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 24$:

$$24 = 3 \times 7 + 3$$
$$7 = 2 \times 3 + 1$$
$$3 = 3 \times 1 + 0$$
$$\texttt{remainders} = (24, 7, 3, 1, 0)$$

this is $\gcd(7, 24)$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

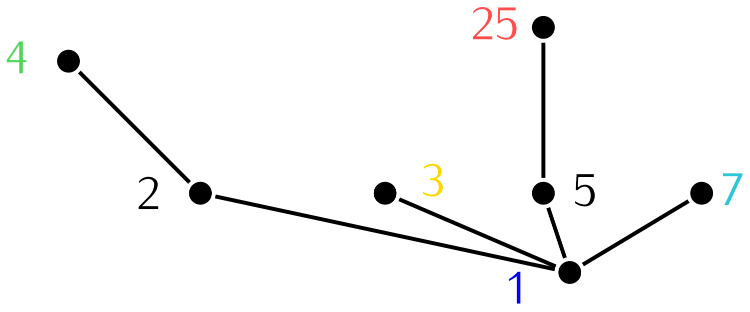
Input: two numbers $a, b \in \mathbb{N}_0$

Output: $a \wedge b$ (or $\gcd(a, b)$) $\longrightarrow$ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 25$:

$$
\begin{aligned}
25 &= 3 \times 7 + 4 \\
7 &= 1 \times 4 + 3 \\
4 &= 1 \times 3 + 1 \\
3 &= 3 \times 1 + 0
\end{aligned}
$$

$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$




```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$) ⟶ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 25$:

$$\begin{aligned} 25 &= 3 \times 7 + 4 \\ 7 &= 1 \times 4 + 3 \\ 4 &= 1 \times 3 + 1 \\ 3 &= 3 \times 1 + 0 \end{aligned}$$

$$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Remark. The algorithm only calls `divmod`.
It would work for any "thing" that also has `divmod`.

Input: two numbers $a, b \in \mathbb{N}_0$
Output: $a \wedge b$ (or $\gcd(a, b)$) $\longrightarrow$ In particular, $\gcd(a, b)$ **always** exists!

On input $a = 7, b = 25$:

$$\begin{aligned} 25 &= 3 \times 7 + 4 \\ 7 &= 1 \times 4 + 3 \\ 4 &= 1 \times 3 + 1 \\ 3 &= 3 \times 1 + 0 \end{aligned}$$

$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Remark. The algorithm only calls `divmod`.
It would work for any "thing" that also has `divmod`.

Things to do when we see an algorithm:

- Is it correct?
- Is it fast?

Runtime of Euclid's algorithm

How many times does the `while` loop run?

$$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$$

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

How many times does the `while` loop run?

$$\texttt{remainders} = ({\color{red}25}, {\color{cyan}7}, {\color{green}4}, {\color{yellow}3}, {\color{blue}1}, 0)$$

Observation 1: the sequence is decreasing

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

Runtime of Euclid's algorithm

How many times does the `while` loop run?

$$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$$

Observation 1: the sequence is decreasing

```
In [9]: mygcd(46368, 75025)
Out[9]:
[75025,
 46368,
 28657,
 17711,
 10946,
 6765,
 4181,
 2584,
 1597,
 987,
 610,
 377,
 233,
 144,
 89,
 55,
 34,
 21,
 13,
 8,
 5,
 3,
 2,
 1,
 0]
```

Runtime of Euclid's algorithm

How many times does the `while` loop run?

$$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$$

Observation 1: the sequence is decreasing
Observation 2: $r_{i+2} < r_i/2$

```
In [9]: mygcd(46368, 75025)
Out[9]:
[75025,
 46368,
 28657,
 17711,
 10946,
 6765,
 4181,
 2584,
 1597,
 987,
 610,
 377,
 233,
 144,
 89,
 55,
 34,
 21,
 13,
 8,
 5,
 3,
 2,
 1,
 0]
```

Runtime of Euclid's algorithm

How many times does the `while` loop run?

$$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$$

Observation 1: the sequence is decreasing
Observation 2: $r_{i+2} < r_i/2$

If true: the sequence divides by 2 every 2 steps
$\rightsquigarrow 2\log_2(b)$ steps at most

$\Uparrow$

$O(\log_2(b))$

```
In [9]: mygcd(46368, 75025)
Out[9]:
[75025,
 46368,
 28657,
 17711,
 10946,
 6765,
 4181,
 2584,
 1597,
 987,
 610,
 377,
 233,
 144,
 89,
 55,
 34,
 21,
 13,
 8,
 5,
 3,
 2,
 1,
 0]
```

Runtime of Euclid's algorithm

How many times does the `while` loop run?

$$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$$

Observation 1: the sequence is decreasing
Observation 2: $r_{i+2} < r_i/2$

If true: the sequence divides by 2 every 2 steps
$\rightsquigarrow 2\log_2(b)$ steps at most

Theorem. Let $(r_0, r_1, \dots, r_k, 0)$ be the sequence of remainders computed by `euclid`.
Then $r_{i+2} < r_i/2$ holds for all $i \in \{0, \dots, k-1\}$.

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

How many times does the `while` loop run?

$$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$$

Observation 1: the sequence is decreasing
Observation 2: $r_{i+2} < r_i/2$

If true: the sequence divides by 2 every 2 steps
$\rightsquigarrow 2\log_2(b)$ steps at most

Theorem. Let $(r_0, r_1, \ldots, r_k, 0)$ be the sequence of remainders computed by `euclid`.
Then $r_{i+2} < r_i/2$ holds for all $i \in \{0, \ldots, k-1\}$.

```
def euclid(a,b):
  if a > b:
    a,b = b,a # swap a and b
  if a == 0:
    return b

  remainders = [b,a]
  while remainders[-1] != 0:
    b = remainders[-2]
    a = remainders[-1]
    q,r = divmod(b,a)
    remainders.append(r)

  return remainders[-2]
```

$$r_i = q \times r_{i+1} + r_{i+2} \text{ with } r_{i+2} \in \{0, \ldots, r_{i+1} - 1\}$$

Case 1 If $r_{i+1} \leq r_i/2$: since $r_{i+2} < r_{i+1}$, by transitivity we have $r_{i+2} < r_i/2$.

Case 2: If $r_{i+1} > r_i/2$: then $q = 1$ so $r_{i+2} = r_i - r_{i+1} < r_i/2$.

Theorem (Bézout)**.** For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Theorem (Bézout). For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Goal. Find $u, v \in \mathbb{Z}$ such that $25u + 7v = 1$.

$$\begin{aligned} 25 &= 3 \times 7 + 4 \\ 7 &= 1 \times 4 + 3 \\ 4 &= 1 \times 3 + 1 \\ 3 &= 3 \times 1 + 0 \\ \texttt{remainders} &= (25, 7, 4, 3, 1, 0) \end{aligned}$$

Theorem (Bézout). For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Goal. Find $u, v \in \mathbb{Z}$ such that $25u + 7v = 1$.

$$
\begin{aligned}
25 &= 3 \times 7 + 4 & \longrightarrow \quad 4 &= 25 - 3 \times 7 \\
7 &= 1 \times 4 + 3 & \longrightarrow \quad 3 &= 7 - 1 \times 4 \\
4 &= 1 \times 3 + 1 & \longrightarrow \quad 1 &= 4 - 1 \times 3 \\
3 &= 3 \times 1 + 0 & & \\
\texttt{remainders} &= (25, 7, 4, 3, 1, 0) & &
\end{aligned}
$$

Theorem (Bézout). For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Goal. Find $u, v \in \mathbb{Z}$ such that $25u + 7v = 1$.

$$
\begin{aligned}
25 &= 3 \times 7 + 4 & \longrightarrow && 4 &= 25 - 3 \times 7 \\
7 &= 1 \times 4 + 3 & \longrightarrow && 3 &= 7 - 1 \times 4 \\
4 &= 1 \times 3 + 1 & \longrightarrow && 1 &= 4 - 1 \times \boxed{3} \\
3 &= 3 \times 1 + 0 & && &= 4 - 1 \times (7 - 1 \times 4)
\end{aligned}
$$

$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$

Theorem (Bézout). For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Goal. Find $u, v \in \mathbb{Z}$ such that $25u + 7v = 1$.

$$\begin{aligned} 25 &= 3 \times 7 + 4 & \longrightarrow\quad 4 &= \boxed{25 - 3 \times 7} \\ 7 &= 1 \times 4 + 3 & \longrightarrow\quad 3 &= 7 - 1 \times 4 \\ 4 &= 1 \times 3 + 1 & \longrightarrow\quad 1 &= 4 - 1 \times 3 \\ 3 &= 3 \times 1 + 0 & &= 4 - 1 \times (7 - 1 \times 4) \\ \texttt{remainders} &= (25, 7, 4, 3, 1, 0) & &= -1 \times 7 + 2 \times \boxed{4} \end{aligned}$$

Theorem (Bézout). For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Goal. Find $u, v \in \mathbb{Z}$ such that $25u + 7v = 1$.

$$\begin{aligned} 25 &= 3 \times 7 + 4 \\ 7 &= 1 \times 4 + 3 \\ 4 &= 1 \times 3 + 1 \\ 3 &= 3 \times 1 + 0 \\ \texttt{remainders} &= (25, 7, 4, 3, 1, 0) \end{aligned}$$

$$\begin{aligned} 4 &= 25 - 3 \times 7 \\ 3 &= 7 - 1 \times 4 \\ 1 &= 4 - 1 \times 3 \\ &= 4 - 1 \times (7 - 1 \times 4) \\ &= -1 \times 7 + 2 \times 4 \\ &= -1 \times 7 + 2 \times (25 - 3 \times 7) \end{aligned}$$

Theorem (Bézout). For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Goal. Find $u, v \in \mathbb{Z}$ such that $25u + 7v = 1$.

$$\begin{aligned}
25 &= 3 \times 7 + 4 & \longrightarrow \quad 4 &= 25 - 3 \times 7 \\
7 &= 1 \times 4 + 3 & \longrightarrow \quad 3 &= 7 - 1 \times 4 \\
4 &= 1 \times 3 + 1 & \longrightarrow \quad 1 &= 4 - 1 \times 3 \\
3 &= 3 \times 1 + 0 & &= 4 - 1 \times (7 - 1 \times 4) \\
\texttt{remainders} &= (25, 7, 4, 3, 1, 0) & &= -1 \times 7 + 2 \times 4 \\
& & &= -1 \times 7 + 2 \times (25 - 3 \times 7) \\
& & &= 2 \times 25 - 7 \times 7
\end{aligned}$$

Theorem (Bézout). For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Goal. Find $u, v \in \mathbb{Z}$ such that $25u + 7v = 1$.

$$
\begin{aligned}
25 &= 3 \times 7 + 4 &\longrightarrow\quad 4 &= 25 - 3 \times 7\\
7 &= 1 \times 4 + 3 &\longrightarrow\quad 3 &= 7 - 1 \times 4\\
4 &= 1 \times 3 + 1 &\longrightarrow\quad 1 &= 4 - 1 \times 3\\
3 &= 3 \times 1 + 0 & &= 4 - 1 \times (7 - 1 \times 4)\\
 & & &= -1 \times 7 + 2 \times 4\\
 & & &= -1 \times 7 + 2 \times (25 - 3 \times 7)\\
 & & &= 2 \times 25 - 7 \times 7
\end{aligned}
$$

$$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$$

Theorem (Bézout)**.** For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Goal. Find $u, v \in \mathbb{Z}$ such that $25u + 7v = 1$.

$$25 = 3 \times 7 + 4 \longrightarrow 4 = 25 - 3 \times 7$$
$$7 = 1 \times 4 + 3 \longrightarrow 3 = 7 - 1 \times 4$$
$$4 = 1 \times 3 + 1 \longrightarrow 1 = 4 - 1 \times 3$$
$$3 = 3 \times 1 + 0$$
$$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$$

$$\begin{aligned} 1 &= 4 - 1 \times 3 \\ &= 4 - 1 \times (7 - 1 \times 4) \\ &= -1 \times 7 + 2 \times 4 \\ &= -1 \times 7 + 2 \times (25 - 3 \times 7) \\ &= 2 \times 25 - 7 \times 7 \end{aligned}$$

But.. why?

- Any common divisor of $\{7, 25\}$ must divide every number of the form $25u + 7v$.
- So if there exist u, v such that $25u + 7v = 1$, the gcd must be 1!

Theorem (Bézout)**.** For all $a, b \in \mathbb{N}_0$, there exists $u, v \in \mathbb{Z}$ such that $u \cdot a + v \cdot b = \gcd(a, b)$.

Goal. Find $u, v \in \mathbb{Z}$ such that $25u + 7v = 1$.

$$
\begin{aligned}
25 &= 3 \times 7 + 4 \\
7 &= 1 \times 4 + 3 \\
4 &= 1 \times 3 + 1 \\
3 &= 3 \times 1 + 0
\end{aligned}
$$

$\texttt{remainders} = (25, 7, 4, 3, 1, 0)$

$$
\begin{aligned}
4 &= 25 - 3 \times 7 \\
3 &= 7 - 1 \times 4 \\
1 &= 4 - 1 \times 3 \\
&= 4 - 1 \times (7 - 1 \times 4) \\
&= -1 \times 7 + 2 \times 4 \\
&= -1 \times 7 + 2 \times (25 - 3 \times 7) \\
&= 2 \times 25 - 7 \times 7
\end{aligned}
$$

But.. why?

- Any common divisor of $\{7, 25\}$ must divide every number of the form $25u + 7v$.
- So if there exist u, v such that $25u + 7v = 1$, the gcd must be 1!
- We will soon want to compute the modular inverse of some numbers.
  Computing inverses = computing Bézout coefficients