CISS451: Cryptography

Y. Liow (March 25, 2025)

Contents

Chapter 1

Basic number theory

SUGGESTIONS. For this chapter, state the basic axioms and properties/theorems of $\mathbb{Z}$. Provide proofs. But remember that most of the properties/theorems can be generlized to properties/theorems for rings. It's still a good idea to prove the facts for $\mathbb{Z}$ since $\mathbb{Z}$ is not as abstract as general rings and will prepare you for the general results.

1.1 Axioms of $\mathbb{Z}$ debug: axioms-of-Z.tex

We will assume that $(\mathbb{Z}, +, \cdot, 0, 1)$ satisfies the following axioms.

- Properties of $+$:
  - Closure: If $x, y \in \mathbb{Z}$, then $x + y \in \mathbb{Z}$.
  - Associativity: If $x, y, z \in \mathbb{Z}$, then $(x + y) + z = x + (y + z)$.
  - Inverse: If $x \in \mathbb{Z}$, then there is some y such that $x + y = 0 = y + x$. The y in the above is an **additive inverse** of x.
  - Neutrality: If $x \in \mathbb{Z}$, then $0 + x = x = x + 0$.
  - Commutativity: If $x, y \in \mathbb{Z}$, then $x + y = y + x$.

  (Memory aid for the first four: CAIN.)
- Properties of $\cdot$:
  - Closure: If $x, y \in \mathbb{Z}$, then $x \cdot y \in \mathbb{Z}$.
  - Associativity: If $x, y, z \in \mathbb{Z}$, then $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.
  - Neutrality: If $x \in \mathbb{Z}$, then $1 \cdot x = x = x + 1$.
  - Commutativity: If $x, y \in \mathbb{Z}$, then $x \cdot y = y \cdot x$.

  It is common to write xy instead of $x \cdot y$.
- Distributivity: If $x, y, z \in \mathbb{Z}$, then $x \cdot (y + z) = x \cdot y + x \cdot z$ and $(y + z) \cdot x = y \cdot x + z \cdot x$

additive inverse

(In the Inverse axiom above, we say that y is *an* additive inverse of x. Later we will show that x has only one additive inverse. Therefore we can say y is *the* additive invers of x.) Also, it is conventional to write xy for $x \cdot y$.

A set R with operations $+_R, \cdot_R$ and elements $0_R, 1_R$ satisfying the above properties is called a **commutative ring**. This is an important generalization because there are many very useful commutative rings and we want to prove results about commutative rings so that these results can be applied to all commutative rings, including but not restricted to $\mathbb{Z}$. If the commutativity of multiplication is left out, then we have the concept of a **non-commutative rings**. This is also an important concept since $n \times n$ matrices with $\mathbb{R}$ entries, $\mathrm{M}_{n\times n}(\mathbb{R})$, form a non-commutative ring. In fact, more generally, the set of $n \times n$ matrices with entries in a commutative ring R, $\mathrm{M}_{n\times n}(R)$, is itself a non-commutative ring. By convention, **ring** means commutative ring. We will return to the concept of commutative and non-commutative rings later.

commutative ring

non-commutative rings

ring

The next property of $\mathbb{Z}$, integrality, is very special and does not apply to all communtative rings and is therefore left out of the definition of a commutative ring:

- Integrality: If $x, y \in \mathbb{Z}$, then $xy = 0 \implies x = 0$ or $y = 0$.

In general a ring satisfying the integrality condition is called an **integral domain**. Therefore $\mathbb{Z}$ is not just a commutative ring – it is an integral domain. Another property of $\mathbb{Z}$ that we will assume is

integral domain

- NONTRIVIALITY: $0 \neq 1$

The nontriviality axiom of $\mathbb{Z}$ is extremely simple, but cannot be deduced from the previous axioms.

The above forms the algebraic properties of $\mathbb{Z}$, i.e., properties involving addition and multiplication.

It is actually possible to first define axioms for $\mathbb{N} = \{0, 1, 2, ...\}$ and then define $\mathbb{Z}$ in terms of $\mathbb{N}$. (See my Calculus notes.) We will not do that except to mention that the axioms for $\mathbb{N}$ are called the Peano-Dedekind axioms and that one very important Peano-Dedekind axiom of $\mathbb{N}$

- INDUCTION of $\mathbb{N}$: If X is a subset of $\mathbb{N}$ satisfying
  - $0 \in X$
  - If $n \in X$, then $n + 1 \in X$

  Then $X = \mathbb{N}$.

It can be proven that as a consequence of the induction axiom of $\mathbb{N}$, we have

- WELL-ORDERING PRINCIPLE (WOP) for $\mathbb{N}$: If X is a nonempty subset of $\mathbb{N}$, then X contains a **least element**, i.e., there is some $m \in X$ such that
$$m \leq x$$
  for all $x \in X$.

least element

Without going into details, it can be shown that for $\mathbb{N}$, the WOP is equivalent to each of the following axioms:

- WEAK MATHEMATICAL INDUCTION for $\mathbb{N}$: Let X be a subset of $\mathbb{N}$ satisyfing the following two conditions:
  - $0 \in X$ and
  - Let $n \in \mathbb{N}$. If $n \in X$, then $n + 1 \in X$.

  Then $X = \mathbb{N}$.

and

- STRONG MATHEMATICAL INDUCTION for $\mathbb{N}$: Let X be a subset of $\mathbb{N}$ satisyfing the following two conditions:

- $0 \in X$ and
- Let $n \in \mathbb{N}$. If $k \in X$ for all $0 \leq k \leq n$, then $n+1 \in X$.

Then $X = \mathbb{N}$.

In the above two induction properties, if we write $X = \{n \mid P(n)\}$ where $P(n)$ is a propositional formula, then the induction axioms can be rewritten in the following way:

- WEAK MATHEMATICAL INDUCTION for $\mathbb{N}$: Let $P(n)$ be a proposition for $n \in \mathbb{N}$ satisyfing the following two conditions:
  - $P(0)$ is true and
  - Let $n \in \mathbb{N}$. If $P(n)$ is true, then $P(n+1)$ is true.

  Then $P(n)$ is true for all $n \in \mathbb{N}$.

and

- STRONG MATHEMATICAL INDUCTION for $\mathbb{N}$: Let $P(n)$ be a proposition for $n \in \mathbb{N}$ satisyfing the following two conditions:
  - $P(0)$ is true and
  - Let $n \in \mathbb{N}$. If $k \in X$ for all $0 \leq k \leq n$, then $n+1 \in X$.
  - Let $n \in \mathbb{N}$. If $P(k)$ is true for $0 \leq k \leq n$, then $P(n+1)$ is true.

  Then $P(n)$ is true for all $n \in \mathbb{N}$.

There are statements analogous to the above, but for $\mathbb{Z}$:

- WOP for $\mathbb{Z}$: If X is a nonempty subset of $\mathbb{Z}$ that is bounded below, then X contains a **least element**, i.e., there is some $m \in X$ such that least element

$$m \leq x$$

  for all $x \in X$. Here, X is **bounded below** means there is some $b \in \mathbb{Z}$ such that bounded below

$$b \leq x$$

  for all $x \in X$. Note that b in the above is in $\mathbb{Z}$ and need not be in X. For instance if $X = -1, 1, 42$, then the least element of X is -1 and $b = -3 \in Z$ is a lower bound of X. However the least element of X must be an element of X.

- WEAK MATHEMATICAL INDUCTION for $\mathbb{Z}$: Let $n_0 \in \mathbb{Z}$. Let $P(n)$ be a proposition for $n \in \mathbb{Z}$ and $n \geq n_0$ satisyfing the following two conditions:
  - $P(n_0)$ is true and
  - Let $n \in \mathbb{Z}$ with $n \geq n_0$. If $P(n)$ is true, then $P(n+1)$ is true.

  Then $P(n)$ is true for all $n \in \mathbb{Z}$, $n \geq n_0$.

- Strong Mathematical Induction for $\mathbb{Z}$: Let $P(n)$ be a proposition for $n \in \mathbb{Z}, n \geq n_0 \in Z$ satisisfing the following two conditions:
  - $P(n_0)$ is true and
  - Let $n \in \mathbb{Z}$. If $k \in X$ for all $0 \leq k \leq n$, then $n+1 \in X$.
  - Let $n \in \mathbb{Z}$. If $P(k)$ is true for $0 \leq k \leq n$, then $P(n+1)$ is true.

  Then $P(n)$ is true for all $n \in \mathbb{Z}$, $n \geq n_0$.

It can be shown that from the induction axiom of $\mathbb{N}$, we can prove that the WOP of $\mathbb{N}$, the weak induction of $\mathbb{N}$, and the strong induction of $\mathbb{N}$ hold. Furthermore, these implies the WOP for $\mathbb{Z}$, the weak induction of $\mathbb{Z}$, and the strong induction of $\mathbb{Z}$ holds.

Note that obvious statement of "WOP for $\mathbb{R}$" does not hold. For instance the open interval $X = (0,1)$ is bounded below (for instance by -42). However there is no m in X such that $m \leq x$ for all x in X. For instance $m = 0.01 \in X$ is not a least element of X since $0.0001 \in X$ is smaller than m. Also, $m = 0.0000001 \in X$ is also not a least element of X since $0.0000000001 \in X$ is less than m. In fact for any $m \in X$, $(1/2)m$ is in X and is less than m. In other words no value in X can be a minimum value of X.

(See end of this section for exercises on WOP and induction.)

The above are the algebraic and inductive-type axioms of $\mathbb{Z}$. There are also the order relations such as $<$ and $\leq$ of $\mathbb{Z}$. Technically speaking, the order relation should come before the inductive axiom, WOP, and induction (weak and strong) since they use order relations.

The following are the order axioms of $\mathbb{Z}$. We assume there is a subset $\mathbb{Z}^+$ of $\mathbb{Z}$ satisfying the following: $\mathbb{Z}^+$

- Trichotomy for $\mathbb{Z}$: If $x \in \mathbb{Z}$, then exactly one of the following holds:

$$-x \in \mathbb{Z}^+, \;\; x = 0, \;\; x \in \mathbb{Z}^+$$

- Closure of $+$ for $\mathbb{Z}^+$: If $x, y \in \mathbb{Z}^+$, then $x + y \in \mathbb{Z}^+$.
- Closure of $\cdot$ for $\mathbb{Z}^+$: If $x, y \in \mathbb{Z}^+$, then $x \cdot y \in \mathbb{Z}^+$.

With the above, we define $<$ on $\mathbb{Z}$ as follows: If $x, y \in \mathbb{Z}$, then we write $x < y$ if $<$

$$y - x \in \mathbb{Z}^+$$

i.e., there is some $z \in \mathbb{Z}^+$ such that

$$y - x = z$$

i.e.,

$$y = x + z$$

Since $<$ is defined (on $\mathbb{Z}$), we can define $x \leq y$ (on $\mathbb{Z}$) to mean "either $x < y$ or $x = y$". The above order relation is expressed abstractly without referring to the fact that $\mathbb{Z}^+$ is $\{1, 2, 3, ...\}$, i.e., the set of positive integers. In fact, you can prove $\mathbb{Z}^+ = \{1, 2, 3, ...\}$ from the above axioms. Of course you define $x > y$ to be the same as $y < x$ and $x \geq y$ to be $y \leq x$. Also, we say $x \in \mathbb{Z}$ is **positive** is $x > 0$. $\leq$ positive

The three axioms in the order relation on $\mathbb{Z}$ can be derived from the order relation axioms of $\mathbb{N}$. The order relation on $\mathbb{N}$ is defined as follows: For $\mathbb{N} = \{0, 1, 2, 3, ...\}$, we simply define

$$x \geq 0$$

for all $x \in \mathbb{N}$. For $x, y \in \mathbb{N}$, we define

$$x \geq y$$

if there is some $z \in \mathbb{N}$ such that

$$x = y + z$$

Finally $x > y$ is defined to be $x \geq y$ and $x \neq y$. If $x > 0$, we say that x is **positive**. Just like positivity in $\mathbb{Z}$, if $x \in \mathbb{N}$, we say that x if **positive** if $x > 0$. positive

Also, we will define $|x|$ in the usual way:

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{otherwise} \end{cases}$$

There is one more axiom of $\mathbb{Z}$ that is related to the "topology" of $\mathbb{Z}$ and uses the order relation:

- Topology for $\mathbb{Z}$: Given any $x \in \mathbb{Z}$, there is no $y \in \mathbb{Z}$ such that

$$x < y < x + 1$$

This can derived from this:

- Topology for $\mathbb{N}$: There is no $x \in \mathbb{N}$ such that

$$0 < x < 1$$

You can think of topology of a set as study of "closeness" of values in that set. For $\mathbb{Q}$, given any two distinct rational values $x < y$, there is also some $z \in \mathbb{Q}$ such that $x < z < y$. This is the same for $\mathbb{R}$. Therefore the topology of $\mathbb{Z}$ is very different from the topology of $\mathbb{Q}$ and $\mathbb{R}$ because there are "holes" in $\mathbb{Z}$ where there are no $\mathbb{Z}$ values. $\mathbb{Z}$ has what is called a **discrete topology**. On the other hand $\mathbb{Q}$ and $\mathbb{R}$ are "dense". (I won't go further into topology since we won't need it. See my notes on topology.) The above assumes the existence of an order relation of $\mathbb{Z}$, i.e., $<$.

discrete topology

Now let's prove some algebraic facts about $\mathbb{Z}$ that can be deduced from the fact that $\mathbb{Z}$ is a commutative ring.

Proposition 1.1.1. (**Uniqueness of additive inverse**). *The additive inverse for x is unique. In other words, if y, y' satisfy*

Uniqueness of additive inverse

$$x + y = 0 = y + x \tag{1}$$
$$x + y' = 0 = y' + x \tag{2}$$

then $y = y'$.

Proof.

$$\begin{aligned}
y &= 0 + y && \text{by Identity of } + \\
&= (y' + x) + y && \text{by RHS of (2)} \\
&= y' + (x + y) && \text{by Associativity of } + \\
&= y' + 0 && \text{by LHS of (1)} \\
&= y' && \text{by Neutrality of } +
\end{aligned}$$

□

Since the additive inverse of x is unique, we can choose to write the additive inverse of x in terms of x. This is usually written $-x$. (Had it been the case that the additive inverse of x is not unique, we would have to denote the additive inverses of x by $(-x)_1$, $(-x)_2$, etc.)

We define the operator $-$ in terms of the additive inverse:

Definition 1.1.1. Let $x, y \in \mathbb{Z}$. We define the subtraction operator as

$$x - y = x + (-y)$$

Note that every value in $\mathbb{Z}$ has an additive inverse, but we do not require values of $\mathbb{Z}$ to have multiplicative inverses:

Definition 1.1.2. Let $x \in \mathbb{Z}$. Then y is a **multiplicative inverse** of x if

multiplicative inverse

$$x \cdot y = 1 = y \cdot x$$

We say that x is a **unit** if x has a multiplicative inverse. We can also say that x is (multiplicatively) **invertible**.

unit

invertible

Intuitively, you know that the only values of $\mathbb{Z}$ with multiplicative inverses are 1 and -1. We will prove this later.

Proposition 1.1.2. (Uniqueness of multiplicative inverse). *Let $x \in \mathbb{Z}$. If x is a unit, then the multiplicative inverse of x is unique. In other words if y, y' satisfy*

Uniqueness of multiplicative inverse

$$xy = 1 = yx \tag{1}$$
$$xy' = 1 = y'x \tag{2}$$

then $y = y'$.

Proof.

$$\begin{aligned}
y &= 1 \cdot y && \text{by Identity axiom of } \cdot \\
&= (y' \cdot x) \cdot y && \text{by RHS of (1)} \\
&= y' \cdot (x \cdot y) && \text{by Associativity of } \cdot \\
&= y' \cdot 1 && \text{by LHS of (2)} \\
&= y' && \text{by Neutrality of } \cdot
\end{aligned}$$

□

Definition 1.1.3. The multiplicative inverse of x, if is exists, is denoted by x^{-1}. The set of all units of $\mathbb{Z}$ is denoted by either $U(\mathbb{Z})$ or $\mathbb{Z}^\times$. We will show later that $U(\mathbb{Z}) = \{-1, 1\}$.

Proposition 1.1.3. (Cancellation law for addition). *Let $x, y, z \in \mathbb{Z}$.*

Cancellation law for addition

(a) *If $x + z = y + z$, then $x = y$.*
(b) *If $z + x = z + y$, then $x = y$.*

Proof. TODO □

Proposition 1.1.4. *Let $x \in \mathbb{Z}$.*

(a) $0x = 0 = x0$
(b) $-0 = 0$
(c) $x - 0 = x$.

Proof. (a) We will first prove $0x = 0$:

$$\begin{aligned}
0x &= (0+0)x && \text{by Neutrality of } + \\
&= 0x + 0x && \text{by Distributivity} \\
\therefore\ 0 + 0x &= 0x + 0x && \text{by Neutrality of } + \\
0 &= 0x && \text{by Cancellation Law of Addition}
\end{aligned}$$

To prove $0 = x0$, from above

$$\begin{aligned}
0 &= 0x \\
&= x0 && \text{by Commutativity of } \cdot
\end{aligned}$$

(b) TODO

(c) TODO □

Proposition 1.1.5. *Let $x, y, c \in \mathbb{Z}$.*

(a) $-(-1) = 1$
(b) $-(-x) = x$
(c) $x(-1) = -x = (-1)x$
(d) $(-1)(-1) = 1$
(e) $(-x)(-y) = xy$
(f) $-(x + y) = -x + -y$
(g) $-(x - y) = -x + y$

Proof. TODO □

Proposition 1.1.6. (Cancellation law for multiplication). *Let $x, y, z \in \mathbb{Z}$.*

Cancellation law for multiplication

(a) *If $xz = yz$ and $z \neq 0$, then $x = y$.*
(b) *If $zx = zy$ and $z \neq 0$, then $x = y$.*

Proof. TODO □

For general expressions involving n terms, instead of writing $x_1 + \cdots + x_n$, we formally define

$\Sigma_{i=1}^{n}$

$$\sum_{i=1}^{n} x_i = \begin{cases} 0 & \text{if } n = 0 \\ \sum_{i=1}^{n-1} x_i + x_n & \text{if } n > 0 \end{cases}$$

Note that the definition above implies that our summation is left associative, i.e.,

$$\begin{aligned} \sum_{i=1}^{3} x_i &= \sum_{i=1}^{2} x_i + x_3 \\ &= \left(\sum_{i=1}^{1} x_i + x_2 \right) + x_3 \\ &= \left(\left(\sum_{i=1}^{0} x_i + x_1 \right) + x_2 \right) + x_3 \\ &= ((0 + x_1) + x_2) + x_3 \\ &= (x_1 + x_2) + x_3 \end{aligned}$$

This conforms with the standard practice. Likewise, we define

$\Pi_{i=1}^{n}$

$$\prod_{i=1}^{n} x_i = \begin{cases} 1 & \text{if } n = 0 \\ \prod_{i=1}^{n-1} x_i + x_n & \text{if } n > 0 \end{cases}$$

Proposition 1.1.7. (General associativity) *Let $x_1, \ldots, x_n \in \mathbb{Z}$.*

General associativity

(a) *Different fully parenthesized expressions of $x_1 + \cdots + x_n$ evaluates to the same value.*
(b) *Different fully parenthesized expressions of $x_1 \cdot \cdots \cdot x_n$ evaluates to the same value.*

Proof. The proof for both statements are similar. We will only prove (b).

TODO ☐

Now for the case when the summation or product involves the same value, we have the following. For convenience, I will write $x^2 = xx$ and in general $()^n$

$$x^n = \begin{cases} 1 & \text{if } n = 0 \\ x^{n-1}x & \text{if } n > 0 \end{cases}$$

Note that

$$x^3 = x^2 x = (x^1 x)x = ((x^0 x)x)x = ((1x)x)x = (xx)x$$

i.e., the above recursive definition of x^n is left recursive. If x has a multiplicative inverse, i.e., if x^{-1} exists, then, for $n \geq 0$, I will define

$$x^{-n} = \left(x^{-1}\right)^n$$

Note that nx has two meanings: nx can be the multiplication of n and x and it can also be $x + \cdots + x$ where the expression contains n copies of x. Of course you would expect them to be the same. For now define

$$[n]x = \begin{cases} 0 & \text{if } n = 0 \\ [n-1]x + x & \text{if } n > 0 \end{cases}$$

$[n]x$ is just a notation that is easier to write than $\sum_{i=1}^n x$. And if n is negative, we define

$$[n]x = -([-n]x)$$

I'll write $[\bullet]$ for the function $\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ as defined above.

Proposition 1.1.8. *Let $x \in \mathbb{Z}$. Then $[n]x = n \cdot x$, i.e., "add n copies of x" is the same as "multiply n and x".*

Proof. We will prove two proofs, using weak induction and WOP.

First we proof the above for $n \geq 0$ using weak induction. For $n \geq 0$, let $P(n)$ be the statement that $[n]x = n \cdot x$.

Base case: We now prove $P(0)$ holds. By definition, $[0]x = 0$. By Proposition 1.1.4 (page 11), $0 \cdot x = 0$. Hence $[0]x = 0 \cdot x$, i.e., $P(0)$ holds.

Inductive case: Now assume $P(n)$ holds, i.e, $[n]x = n \cdot x$. We will prove that $P(n+1)$ holds. We have

$$\begin{aligned}
[n+1]x &= [n]x + x && \text{by definition of } [\bullet] \\
&= n \cdot x + x && \text{by inductive hypothesis } P(n) \\
&= n \cdot x + 1 \cdot x && \text{by Neutrality of } \cdot \\
&= (n+1) \cdot x && \text{by Distributivity}
\end{aligned}$$

Hence $P(n+1)$ holds.

Therefore by weak induction $P(n)$ holds for all integers $n \geq 0$.

We now prove $P(n)$ holds for integers $n < 0$:

$$\begin{aligned}
[n]x &= -([-n]x) && \text{by definition of } [\bullet] \\
&= -((-n) \cdot x) && \text{by } P(-n) \\
&= (-1) \cdot ((-n) \cdot x) && \text{by Proposition 1.1.5(c) (page 11)} \\
&= ((-1) \cdot (-n)) \cdot x) && \text{by Associativity of } \cdot \\
&= (1 \cdot n) \cdot x && \text{by Proposition 1.1.5(e) (page 11)} \\
&= n \cdot x && \text{by Neutrality of } \cdot
\end{aligned}$$

Next, we proof the above for $n \geq 0$ using WOP.

TODO □

Now let us look at order relations on $\mathbb{Z}$.

Proposition 1.1.9. *Let $a, b, c, d \in \mathbb{Z}$ with $x > 0$.*

(a) *If $a < b$, then $a + c < b + c$.*
(b) *If $a < b$, then $ax < bx$.*
(c) *If $a < b$, then $a(-x) > b(-x)$.*
(d) $a < a + c$
(e) $a \leq ac$

There are analogous statements for (a)-(c) where $<$ and $>$ are replaced by $\leq$ and $\geq$.

Proof. (a) Note that

$$\begin{aligned}
(b+c)-(a+c) &= (b+c)+(-(a+c)) && \text{by definition of } - \\
&= b+(c+(-(a+c))) && \text{by Associativity of } + \\
&= b+(c+(-a+-c)) && \text{by Proposition 1.1.5 (page 11)} \\
&= b+(c+(-c+-a)) && \text{by Commutativity of } + \\
&= b+((c+-c)+-a) && \text{by Associativty of } + \\
&= b+(0+-a)) && \text{by Inverse of } + \\
&= b+-a && \text{by Neutrality of } + \\
&= b-a && \text{by definition of } - \qquad (1)
\end{aligned}$$

We have

$$\begin{aligned}
& a < b \\
\therefore\ & b-a \in \mathbb{Z}^+ && \text{by definition of } < \\
\therefore\ & (b+c)-(a+c) = b-a \in \mathbb{Z}^+ && \text{by (1)} \\
\therefore\ & b+c > a+c && \text{by definition of } <
\end{aligned}$$

□

Proposition 1.1.10. $1 \in \mathbb{Z}^+$, *i.e.*, 1 *is positive.*

Proof. TODO □

Proposition 1.1.11. $\mathbb{Z}^+ = \{1, 2, 3, ...\} = \mathbb{N} - \{0\}$.

Proof. TODO □

Proposition 1.1.12. *The only units of* $\mathbb{Z}$ *are* $-1, 1$, $U(\mathbb{Z}) = \{-1, 1\}$.

Proof. TODO □

The following are examples of proofs using WOP and induction.

Exercise 1.1.1. For $n \in \mathbb{N}$,

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$

Here, when $n = 0$, $1 + 2 + \cdot + n$ is defined to be 0.

(a) Prove the above using weak induction.
(b) Prove the above using WOP.

Proof. TODO □

Exercise 1.1.2. For $n \in \mathbb{N}$,

$$1^2 + 2^2 + \cdots + n^2 = \frac{1}{6}n(n+1)(2n+1)$$

Here, when $n = 0$, $1 + 2 + \cdot + n$ is defined to be 0.

(a) Prove the above using weak induction.
(b) Prove the above using WOP.

Proof. TODO □

Exercise 1.1.3. What is wrong with this proof? I claim that every man wears brown pants. I will prove this by weak induction. Let $P(n)$ be the statement that every man in a set of n men wears brown pants. Clearly $P(0)$ is true. For, if $P(0)$ does not hold, then there is a man in an empty set that does not wear brown pants. But an empty set cannot contain a man. Now, let $n \geq 0$ and assume $P(n)$ hold. I will prove that $P(n+1)$ holds. Let X be a set with $n+1$ men. Let Y be a subset of X with exactly n men. Since there are n men in Y, by induction hypothesis, every men in Y wears brown pants. Of course Y misses one man from X, say John. Now form another subset Z of X with exactly n men that contains John. Again by induction hypothesis $P(n)$ hold and therefore every men in Z wears brown pants, including John. Since X is made up of men in Y together with John, we have shown that every men in X wears brown pants. Therefore by weak induction, every men wears brown pants.

(Of course the above cannot possible be true. In fact: you can change "every man wears brown pants" to "every man wears the same pair of pants".)

TODO

Exercise 1.1.4. What is wrong with this proof? I claim that every man has the same favorite number. I will prove this by weak induction. Let $P(n)$ be the statement that every man in a set of n men has the same favorite number as the rest. Clearly $P(0)$ is true vacuously like the previous exercise. Now, let $n \geq 0$ and assume $P(n)$ hold. I will prove that $P(n+1)$ holds. Let X be a set of $n+1$ men. Line the men in X in a line. The set Y of the first n men of X must all share the same favorite number since there are n men in Y. Let Z be the set of n men of X that excludes the first man. All the men in Z must have the same favorite number as well. Clearly every men in Y has the same favorite number as the man in the middle of the line and every men in Z also has the same favorite number as this man in the middle. Therefore all men in X have the same favorite number. Hence every men shared the same favorite number.

TODO

Exercise 1.1.5. What is wrong with this proof? I claim that every day is Monday. I will prove this by strong induction. Let $P(n)$ be the statement that every day in a set of n days is Monday. Clearly $P(0)$ is true. For, if $P(0)$ does not hold, then there is a day in an empty set that is not Monday. But an empty set cannot have a day. Now, let $n \geq 0$ and assume $P(0), P(1), ..., P(n)$ hold. I will prove that $P(n+1)$ holds. Let X be a set with $n+1$ days. Let Y be a subset of X with exactly n days. Since there are n days in Y, by induction hypothesis, every day in Y in Monday. Y misses one day in X, say x. Also by $P(1)$, since $\{x\}$ is a set of size 1, x is also Monday. Since $X = Y \cup \{x\}$, we have shown that every day in X is Monday, Therefore by strong induction, every day is Monday.

TODO

Exercise 1.1.6. What is wrong with this proof? I will prove by strong induction that $e^n = 1$ for all $n \geq 0$. Let $P(n)$ be the statement that $e^n = 1$. Clearly $e^0 = 1$. Therefore $P(0)$ holds. Now assume $P(0), P(1), ..., P(n)$ holds. Since $P(n)$ and $P(1)$ holds, we have $e^{n+1} = e^n \cdot e^1 = 1 \cdot 1 = 1$. Hence $P(n+1)$ holds. Therefore, by strong induction, $P(n)$ holds for all n, i.e., $e^n = 1$ for all $n \geq 0$.

TODO

Exercise 1.1.7. What is wrong with this proof? I will prove by strong induction that given $x \in \mathbb{R}$, $\sum_{i=1}^{n} x = 0$ for $n \geq 0$. Let $P(n)$ be the above statement. $P(0)$ holds since $\sum_{i=1}^{0} x$ is a sum with no terms and by convention is 0. Now assume $P(0), P(1), ..., P(n)$ holds. Let $n+1 = a+b$ with both a, b approximately equal. (For instance if $n+1$ is even, $a = b = (n+1)/2$. And if $n+1$ is odd, $a = \lfloor (n+1)/2 \rfloor$ and $b = n+1-a$.) Since $P(a)$ and $P(b)$ holds, $\sum_{i=1}^{a} x = 0$ and $\sum_{i=1}^{b} x = 0$. Clearly $\sum_{i=1}^{n+1} x = \sum_{i=1}^{a} x + \sum_{i=1}^{b} x$ since the LHS is a sum of $n+1$ x's and there are $a+b$ x's on the RHS. Hence

$$\sum_{i=1}^{n+1} x = \sum_{i=1}^{a} x + \sum_{i=1}^{b} x = 0 + 0 = 0$$

Therefore $P(n+1)$ holds. By strong induction $P(n)$ holds for all $n \geq 0$, i.e., $\sum_{i=1}^{n} x = 0$ for $n \geq 0$.

TODO

1.2 Divisibility debug: divisibility.tex

Definition 1.2.1. Let $a, n \in \mathbb{Z}$ with $a \neq 0$. Then we say that a **divides** b, and we write $a \mid b$, if there is some $x \in \mathbb{Z}$ such that $ax = b$, i.e.,

$$\exists x \in \mathbb{Z} \cdot [ax = b]$$

(The "$\in \mathbb{Z}$" is not necessary as long as it is clear the universe is $\mathbb{Z}$.) Note that when you write $a \mid b$, the a is always nonzero.

Proposition 1.2.1. *Let $a, b, c \in \mathbb{Z}$.*

(a) $1 \mid a$.
(b) $a \mid 0$.
(c) Reflexivity*:* $a \mid a$.
(d) Transitivity*: If $a \mid b$ and $b \mid c$, then $a \mid c$.*
(e) Antisymmetry: *If $a \mid b$ and $b \mid a$, then $a = \pm b$.*
(f) *If $a \mid b$, then $a \mid bc$.*
(g) *If $a \mid b$ and $a \mid c$, then $a \mid b + c$.*
(h) Linearity*: If $a \mid b$, $a \mid c$, then $a \mid bx + cy$ for $x, y \in \mathbb{Z}$.*
(i) *If $a \mid b$, then $|a| \leq |b|$.*

Proof. TODO □

1.3 Congruences debug: congruences.tex

Definition 1.3.1. Let $a, b \in \mathbb{Z}$ and $N \in \mathbb{Z}$ with $N > 0$. Then a is congruent to b mod N and we write

$$a \equiv b \pmod{N}$$

if $N \mid a - b$. In the above expression

$$a \equiv b \pmod{N}$$

we say that N is the modulus of the above relation between a and b.

Proposition 1.3.1. *Let $a, b, c, a', b' \in \mathbb{Z}$ and $N, N' \geq 0$ be in $\mathbb{Z}$.*

(a) Reflexivity: $a \equiv a \pmod{N}$
(b) Symmetry: *If $a \equiv b \pmod{N}$, then $b \equiv a \pmod{N}$*
(c) Transitivity: *If $a \equiv b, b \equiv c \pmod{N}$, then $a \equiv c \pmod{N}$*
(d) Additivity: *If $a \equiv b$, $a' \equiv b' \pmod{N}$, then $a + a' \equiv b + b' \pmod{N}$.*
(e) Multiplicativity: *If $a \equiv b$, $a' \equiv b' \pmod{N}$, then $aa' \equiv bb' \pmod{N}$.*
(f) *If $a \equiv b \pmod{NN'}$, then $a \equiv b \pmod{N}$*

Proof. TODO □

The following connects the Euclidean property and the congruence relation:

Proposition 1.3.2. *Let $a, N \in \mathbb{Z}$ with $N > 0$. Let $q, r \in \mathbb{Z}$ such that*

$$a = Nq + r, \ \ 0 \leq r < N$$

Then $a \equiv r \pmod{N}$.

Proof. TODO

Definition 1.3.2. Let $a, N \in \mathbb{Z}$ with $N > 0$. By the Euclidean property of $\mathbb{Z}$, there exist unique q, r such that

$$a = Nq + r, \ \ 0 \leq r < N$$

r is called the "**residue** of a mod N" ("residue" = "what is left" after a is divided by N, i.e., the remainder after a is divided by N). It is common to write r as $a \bmod N$.

Sometimes $a \bmod N$ is written as $r_N(a)$. For instance to find the residue of 15 mod 4, there is some q such that

$$15 = 4q + 3, \;\; 0 \leq 1 < 4$$

i.e.

$$15 \equiv 3 \pmod 4$$

where $0 \leq 1 < 4$. Therefore the residue of 15 mod 4 is 1, or we simple write

$$15 \bmod 4 = 3$$

i.e., $r_4(15) = 3$.

WARNING: "mod" now has two meanings. "mod N", where $N > 0$ is an integer, can be used to denote a relation between integers

$$a \equiv b \pmod N$$

and "mod N" can also be used to denote a function

$$a \bmod N = r$$

Exercise 1.3.1. Show that the cancellation law for $\mathbb{Z}$ does not translate to $\mathbb{Z}$ mod N. In other words, find N, a, b, c such that $c \not\equiv 0 \pmod N$ and

debug: exercises/nt-50/question.tex

$$ac \equiv bc \pmod N, \;\; a \not\equiv b \pmod N$$

(Go to solution, page 22) □

Solutions

Solution to Exercise 1.3.1.

debug: exercises/nt-50/answer.tex

Solution not provided.

1.4 Euclidean property debug: euclidean-property.tex

$\mathbb{Z}$ satisfies this very important property:

Theorem 1.4.1. (**Euclidean property**) *If $a, b \in \mathbb{Z}$ with $b \neq 0$, then there are integers q and r satisfying* Euclidean property

$$a = bq + r, \quad 0 \leq |r| < |b|$$

The above theorem is the version that can be generalized to general rings. Below is the version for $\mathbb{Z}$. The only difference is the $|r|$ is replaced by r:

Theorem 1.4.2. (**Euclidean property 2**) *If $a, b \in \mathbb{Z}$ with $b \neq 0$, then there are integers q and r satisfying* Euclidean property 2

$$a = bq + r, \quad 0 \leq r < |b|$$

In many cases, one is interested in the case when $a \geq 0$. So this version is the one found in most textbooks:

Theorem 1.4.3. (**Euclidean property 3**) *If $a, b \in \mathbb{Z}$ with $a \geq 0, b > 0$, then there are integers $q \geq 0$ and $r \geq 0$ satisfying* Euclidean property 3

$$a = bq + r, \quad 0 \leq r < b$$

Note that a, b, q, r are in $\mathbb{N} = \{0, 1, 2, ...\}$. Hence the above can also be called Euclidean property of $\mathbb{N}$.

q is called the **quotient** when a is divided by b; r is the **remainder**. q and r are unique (see proposition below). For instance if $a = 25$ and $b = 3$, then quotient remainder

$$25 = 3 \cdot 8 + 1, \quad 0 \leq 1 < 3$$

The computation

$$a, b \to q, r$$

is called a **division algorithm**. division algorithm

In Python, you can do this:

```
a = 25
b = 8
q, r = divmod(25, 8)
print("%s = %s * %s + %s" % (a, b, q, r))
```

The output is

```
[student@localhost elementary-number-theory] python divmod.py
25 = 8 * 3 + 1
```

Algorithmically, when a and b have a huge number of digits and they are represented using arrays of digits, the division algorithm to compute q, r is basically long division you learnt in middle school. At the hardware level, the same division algorithm occurs but the computation is in terms of bits and not digits.

If we peek ahead and pretend for the time being that fractions such as $\frac{a}{b}$ exists, then for $a > 0$ and $b > 0$, we have

$$q = \left\lfloor \frac{a}{b} \right\rfloor, \quad r = a - bq$$

where $\lfloor x \rfloor$ means the floor of x. If we write `(a/b)` for the *integer* quotient of a by b (i.e. this is the / in C++ for integers) and `(a%b)` for the corresponding remainder, then of course we have

```
a = b * (a/b) + (a%b)
```

Although the above Euclidean property is for $\mathbb{Z}$, we will first prove it for $a \geq 0$ and $b > 0$. The q, r will satisfy $q \geq 0$, $r \geq 0$. (Furthermore in this setup q, r are unique.) Once we have proven the Euclidean property for integer $a \geq 0$, it will not be difficult to extend the result to the whole of $\mathbb{Z}$.

To prove the Euclidean property of $\mathbb{Z}$, we will use WOP. (One can also prove the Euclidean property of $\mathbb{Z}$ using induction.)

Well-ordering principle for $\mathbb{N}$

Well-ordering principle (WOP) for $\mathbb{N}$: Let X be a nonempty subset of $\mathbb{N}$. Then X has a least element. In other words there is some $m \in X$ such that $m \leq x$ for all $x \in X$.

You can prove the following version of well-ordering principle on $\mathbb{Z}$:

Well-ordering principle for $\mathbb{Z}$

Well-ordering principle for $\mathbb{Z}$: Let X be a nonempty subset of $\mathbb{Z}$ that is *bounded below*. Then X has a least element. In other words there is some

$m \in X$ such that $m \leq x$ for all $x \in X$.

$\mathbb{R}$ does not satisfy the second version well-ordering principle with $\mathbb{Z}$ replaced by $\mathbb{R}$. For instance the open interval $X = (0, 1)$ is bounded below (for instance by -42). However there is no m in X such that $m \leq x$ for all x in X. For instance $m = 0.01 \in X$ is not a minimum element of X since $0.0001 \in X$ is smaller than m. Also, $m = 0.0000001 \in X$ is also not a minimum of X since $0.0000000001 \in X$ is less than m. In fact for any $m \in X$, $(1/2)m$ is in X and is less than m. In other words no value in X can be a minimum value of X.

Now we will prove Theorem 1.4.3.

Proof. TODO □

Exercise 1.4.1. Prove Theorem 1.4.3 using induction. (Go to solution, page 27) □

debug: exercises/nt-65/question.tex

Proposition 1.4.1. *Given a, b, the q, r in Theorem 1.4.3 are unique. In other words, if*

$$\begin{aligned} a &= bq + r, \;\; 0 \leq r < |b| \\ a &= bq' + r', \;\; 0 \leq r' < |b| \end{aligned}$$

then

$$q = q', \quad r = r'$$

Proof. TODO □

Now I'm going to prove Theorem 1.4.1 which allows a to be any integer.

Proof of Theorem 1.4.1. TODO □

Exercise 1.4.2. Prove: If $a \in \mathbb{Z}$ and $b \in \mathbb{Z}$ and $b \neq 0$, then there are unique integers q, r such that $a = bq + r$ and $b \leq r < 2b$. (Go to solution, page 29) □

debug: exercises/nt-00/question.tex

Exercise 1.4.3. Using the Euclidean property, prove that every integer is congruent to $0, 1, 2,$ or $3 \mod 4$. (Go to solution, page 30) □

debug: exercises/nt-01/question.tex

Exercise 1.4.4. Prove that squares are 0 or 1 mod 4. In other words if $a \in \mathbb{Z}$, then $a^2 \equiv 0$ or $1 \pmod 4$. (Go to solution, page 31) □

debug: exercises/nt-02/question.tex

Exercise 1.4.5. Solve $4x^3 + y^2 = 5z^2 + 6$ (in $\mathbb{Z}$). (Go to solution, page 32) □

debug: exercises/nt-03/question.tex

Exercise 1.4.6. Prove that $11, 111, 1111, 11111, 111111, \ldots$ are all not perfect squares. (An integer is a perfect square is it's of the form a^2 where a is an integer.) (Go to solution, page 33) □

debug: exercises/nt-04/question.tex

Exercise 1.4.7. How many of $3, 23, 123, 1123, 11123, 111123, 1111123, \ldots$ are perfect squares? (Go to solution, page 34) □

debug: exercises/nt-05/question.tex

Solutions

Solution to Exercise 1.4.1.

debug: exercises/nt-65/answer.tex

We'll prove (a) by mathematical induction. (We'll prove (b) later.) So let's form our $P(n)$. Let b be fixed. Let $P(n)$ be the statement:

$$P(n) : \text{There are integer } q, r \text{ such that } n = bq + r,\ 0 \leq r < b$$

(I'm thinking of the original a as a variable in the $P(n)$.)

The base case is easy: If $n = 0$, then if we set $q = 0$ and $r = 0$, then we do have

$$0 = b \cdot 0 + 0, \quad 0 \leq 0 < b$$

Hence $P(0)$ holds.

Now for the inductive case. Assume that $P(n)$ holds. Now we consider the statement $P(n+1)$. Note that $P(n)$ is true. Therefore there are integers q, r satisfying

$$n = bq + r, \quad 0 \leq r < b$$

Therefore

$$n + 1 = bq + r + 1$$

Either $r = b - 1$ or $r < b - 1$. (Note that r cannot be greater than $b - 1$ since $r < b$.)

CASE: $r = b - 1$. Then we have

$$\begin{aligned} n &= bq + r \\ \therefore \quad n + 1 &= bq + r + 1 \\ \therefore \quad n + 1 &= bq + b \\ \therefore \quad n + 1 &= b(q + 1) \\ \therefore \quad n + 1 &= b(q + 1) + 0, 0 \leq 0 < b \end{aligned}$$

So if we set $q' = q + 1$ and $r' = 0$ we do have

$$n + 1 = bq' + r', \quad 0 \leq r' < b$$

In other words $P(n+1)$ holds.

<u>CASE: $r < b - 1$.</u> Then we have

$$\begin{aligned} & n = bq + r, \quad 0 \leq r, \quad r < b - 1 \\ \therefore \quad & n + 1 = bq + r + 1, \quad 0 \leq r, \quad r < b - 1 \\ \therefore \quad & n + 1 = bq + r + 1, \quad 0 \leq r, \quad r + 1 < b \\ \therefore \quad & n + 1 = bq + r + 1, \quad 0 \leq r + 1, \quad r + 1 < b \\ \therefore \quad & n + 1 = bq + r + 1, \quad 0 \leq r + 1 < b \end{aligned}$$

So if we set $q' = q$ and $r' = r + 1$, we have

$$n + 1 = bq' + r', \quad 0 \leq r' < b$$

<u>CASE: $r > b - 1$.</u> This case cannot occur since we have the fact that $r < b$.

Therefore in all cases, $P(n + 1)$ holds.

Since $P(0)$ is true and if $P(n)$ holds, then so does $P(n + 1)$ for $n \geq 0$, by mathematical induction, $P(n)$ must be true for all $n \geq 0$.

Solution to Exercise 1.4.2.

debug: exercises/nt-00/answer.tex

Solution not provided.

Solution to Exercise 1.4.3.

debug: exercises/nt-01/answer.tex

Solution not provided.

Solution to Exercise 1.4.4.

debug: exercises/nt-02/answer.tex

Solution not provided.

Solution to Exercise 1.4.5.

debug: exercises/nt-03/answer.tex

Solution not provided.

Solution to Exercise 1.4.6.

debug: exercises/nt-04/answer.tex

Solution not provided.

Solution to Exercise 1.4.7.

debug: exercises/nt-05/answer.tex

1.5 Bézout's identity and the Extended Euclidean Algorithm debug: extended-euclidean-algorithm.tex

The following theorem is very important and plays a role in Euclid's lemma on primes. First we will define GCD.

Definition 1.5.1. Let $a, b \in \mathbb{Z}$ such that not both a, b are 0.

(a) Let $din\mathbb{Z}$ where $d \neq 0$. d is a **common divisor** of a and b if $d \mid a$ and $d \mid b$.

(b) Let $g \in \mathbb{Z}$. Then g is the **greatest common divisor** of a and b if g is a common divisor of a, b and g is the largest among all common divisors of a, b. The greatest common divisor is an integer if a, b are not both zero since in this case $\gcd(a, b)$ divides $\max(|a|, |b|)$ and hence $\gcd(a, b)$ is finite.

greatest common divisor

(Note that is $a = 0 = b$, then all integers are common divisors of a, b. In this case the greatest common divisor of a, b is not defined.)

Theorem 1.5.1. (**Bézout's identity**) *If a and b be integers which are not both zero, then there are integers x, y such that*

Bézout's identity

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

The x, y in the above theorem are called **Bézout's coefficients** of a, b. They are not unique.

Bézout's coefficients

Exercise 1.5.1. Prove that if $a \neq 0$, then there are many possible x, y such that $ax + by = \gcd(a, b)$.

debug: exercises/nt-51/question.tex

(Go to solution, page 52) □

Bézout's identity

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

does not provide an algorithm to compute the Bézout's coefficients x, y. Then I'll do a computational example that compute the gcd of a, b as a linear combination of a and b. The example actually contains the idea behind the algorithm to compute the Bézout's coefficients. The algorithm is called the Extended Euclidean Algorithm.

Proof. For convenience, let me write (a, b) as the set of linear combinations of a and b, i.e.,

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

We will also write (g) for the linear combination of g, i.e.,

$$(g) = \{gx \mid x \in \mathbb{Z}\}$$

(Such linear combinations are called ideals. They are extremely important in of themselves. Historically, they were created to study Fermat's last theorem. Since then they are crucial in the the study of ring theory.)

TODO $\square$

The above does not give you an algorithm to compute the x and y. First let me do an example to show you that it's possible to compute $\gcd(a, b)$ as a linear combination of a and b. Then I'll give you the actual algorithm.

Recall that we have computed $\gcd(514, 24) = 2$. Bézout's theorem says that it's possible to find x and y such that

$$2 = \gcd(514, 24) = 514x + 24y$$

How do we compute the x and y? Just like the gcd computation (the Euclidean Algorithm), the x, y are computed using the Euclidean property. First we have

$$514 = 21 \cdot 24 + 10$$

This implies that

$$514 \cdot 1 + 24 \cdot (-21) = 10$$

Now it would be nice if the pesky 10 goes away and is replaced by 2. How would we do that? Well look at 24 and 10 now. We have

$$24 = 2 \cdot 10 + 4$$

again by Euclidean algorithm. Multiplying the equation

$$514 \cdot 1 + 24 \cdot (-21) = 10$$

throughout by 2 gives us

$$514 \cdot 2 + 24 \cdot (-42) = 2 \cdot 10$$

The previous equation

$$24 = 2 \cdot 10 + 4$$

say that $2 \cdot 10$ can be replaced by $24 - 4$. This means that

$$514 \cdot 2 + 24 \cdot (-42) = 24 - 4$$

Hmmm ... this says that we have now

$$514 \cdot 2 + 24 \cdot (-43) = -4$$

or

$$514 \cdot (-2) + 24 \cdot 43 = 4$$

What about 4? Well, if we look at 10 and 4 just like what we did with 24 and 10 we would get

$$10 = 2 \cdot 4 + 2$$

and the remainder 2 gives us the GCD!!! Rearranging it a bit we have

$$1 \cdot 10 + (-2) \cdot 4 = 2$$

i.e. 2 is a linear combination of 10 and 4. But earlier we say that 4 is a linear combination of 514 and 24 ...

$$514 \cdot (-2) + 24 \cdot 43 = 4$$

and even earlier we saw that 10 is also a linear combination of 514 and 24 ...

$$514 \cdot 1 + 24 \cdot (-21) = 10$$

Surely if we substitute all these values into the equation

$$1 \cdot 10 + (-2) \cdot 4 = 2$$

we would get 2 as a linear combination of 514 and 24. Let's do it ...

$$\begin{aligned}
2 &= 1 \cdot 10 + (-2) \cdot 4 \\
&= 1 \cdot (514 \cdot 1 + 24 \cdot (-21)) + (-2)(514 \cdot (-2) + 24 \cdot 43) \\
&= 514 \cdot 1 + 24 \cdot (-21) + 514 \cdot 4 + 24 \cdot (-86) \\
&= 514 \cdot 5 + 24 \cdot (-107)
\end{aligned}$$

Vóila!

Exercise 1.5.2. Using the above idea, compute the gcd and Bézout's coefficients of 210 and 78, i.e., compute x and y such that $210x+78y = \gcd(210, 78)$. (Go to solution, page 53) ☐

debug: exercises/nt-52/question.tex

Exercise 1.5.3. Analyze the above and design an algorithm so that when given a and b, the algorithm computes x and y such that $ax + by = \gcd(a, b)$. (Go to solution, page 54) ☐

debug: exercises/nt-53/question.tex

To help you analyze the above computation, let me organize our computations a little. If we can make the process systematic, then there is hope that we can make the idea work for all a and b, i.e., then we would have an algorithm and hence can program it and compute it's runtime performance.

We know for sure that we have to continually use Euclidean property on pairs of numbers. So here we go:

$$\begin{aligned} 514 &= 21 \cdot 24 + 10 \\ 24 &= 2 \cdot 10 + 4 \\ 10 &= 2 \cdot 4 + 2 \\ 4 &= 2 \cdot 2 + 0 \end{aligned}$$

Note that this corresponds to the gcd computation

$$\begin{aligned} \gcd(514, 24) &= \gcd(24, 514 - 21 \cdot 24) = \gcd(24, 10) \\ &= \gcd(10, 24 - 2 \cdot 10) = \gcd(10, 4) \\ &= \gcd(4, 10 - 2 \cdot 4) = \gcd(4, 2) \\ &= \gcd(2, 4 - 2 \cdot 2) = \gcd(2, 0) \\ &= 2 \end{aligned}$$

So in the computation

$$\begin{aligned} 514 &= 21 \cdot 24 + 10 \\ 24 &= 2 \cdot 10 + 4 \\ 10 &= 2 \cdot 4 + 2 \\ 4 &= 2 \cdot 2 + 0 \end{aligned}$$

if the remainder is 0 (see the last line), then the previous line's remainder must be the gcd.

Let's look at our computation of the gcd of 514 and 24:

$$\begin{aligned} 514 &= 21 \cdot 24 + 10 \\ 24 &= 2 \cdot 10 + 4 \\ 10 &= 2 \cdot 4 + 2 \\ 4 &= 2 \cdot 2 + 0 \end{aligned}$$

Recall that the above computation means that the gcd is 2. Note only that through backward substitution, we can rewrite 2 as a linear combination of 514 and 24.

Let's try to do this in a more organized way. So here's our facts again:

$$\begin{aligned} 514 &= 21 \cdot 24 + 10 \\ 24 &= 2 \cdot 10 + 4 \\ 10 &= 2 \cdot 4 + 2 \end{aligned}$$

Let me put the remainders on one side:

$$\begin{aligned} 10 &= 514 - 21 \cdot 24 && (1) \\ 4 &= 24 - 2 \cdot 10 && (2) \\ 2 &= 10 - 2 \cdot 4 && (3) \end{aligned}$$

Note that (1) tells you that 10 is a linear combination of $514, 24$. (2) tells you that 4 is a linear combination of $24, 10$. If we substitute (1) into (2), 4 will become a linear combination of $514, 24$. (3) says that 2 is a linear combination of $10, 4$. But 10 is a linear combination of $514, 24$ and 4 is a linear combination of $514, 24$. Hence 2 is also a linear combination of $514, 24$. See it?

OK. Let's do it. From

$$\begin{aligned} 10 &= 514 - 21 \cdot 24 && (1) \\ 4 &= 24 - 2 \cdot 10 && (2) \\ 2 &= 10 - 2 \cdot 4 && (3) \end{aligned}$$

if we substitute (1) into (2) and (3) (i.e., rewrite 10 as a linear combination of

$514, 24)$:

$$\begin{aligned} 10 &= 514 - 21 \cdot 24 & (1) \\ 4 &= 24 - 2 \cdot (514 - 21 \cdot 24) & (2) \\ 2 &= (514 - 21 \cdot 24) - 2 \cdot 4 & (3) \end{aligned}$$

Collecting the multiples of 514 and 24:

$$\begin{aligned} 10 &= 514 - 21 \cdot 24 & (1) \\ 4 &= (-2) \cdot 514 + (1 + (-2)(-21)) \cdot 24 & (2\text{'}) \\ 2 &= (1) \cdot 514 + (-21) \cdot 24 - 2 \cdot 4 & (3\text{'}) \end{aligned}$$

and simplifying:

$$\begin{aligned} 10 &= 514 - 21 \cdot 24 & (1) \\ 4 &= (-2) \cdot 514 + (43) \cdot 24 & (2') \\ 2 &= (1) \cdot 514 + (-21) \cdot 24 - 2 \cdot 4 & (3') \end{aligned}$$

Substituting $(2')$ into $(3')$:

$$\begin{aligned} 10 &= 514 - 21 \cdot 24 & (1) \\ 4 &= (-2) \cdot 514 + (43) \cdot 24 & (2') \\ 2 &= (1) \cdot 514 + (-21) \cdot 24 - 2 \cdot ((-2) \cdot 514 + (43) \cdot 24) & (3') \end{aligned}$$

Tidying up:

$$\begin{aligned} 10 &= 514 - 21 \cdot 24 & (1) \\ 4 &= (-2) \cdot 514 + (43) \cdot 24 & (2\text{'}) \\ 2 &= (1 - 2(-2)) \cdot 514 + (-21 - 2(43)) \cdot 24 & (3\text{''}) \end{aligned}$$

Simplifying:

$$\begin{aligned} 10 &= 514 - 21 \cdot 24 & (1) \\ 4 &= (-2) \cdot 514 + (43) \cdot 24 & (2\text{'}) \\ 2 &= (5) \cdot 514 + (-107) \cdot 24 & (3\text{''}) \end{aligned}$$

(It's a good idea to check after each substitution that the equalities still hold. We all make mistakes, right?)

OK. That's great. It looks more organized now. So much so that you can now easily write a program to compute the above.

Now let's look at the general case. Suppose instead of 514 and 24, we write a and b. The computation will look like this:

$$\begin{aligned} a &= q_1 \cdot b + r_1 \\ b &= q_2 \cdot r_1 + r_2 \\ r_1 &= q_3 \cdot r_2 + r_3 \\ r_2 &= q_4 \cdot r_3 + 0 \end{aligned}$$

To make things even more regular and uniform, let me rewrite it this way:

$$\begin{aligned} r_0 &= q_1 \cdot r_1 + r_2 \\ r_1 &= q_2 \cdot r_2 + r_3 \\ r_2 &= q_3 \cdot r_3 + r_4 \\ r_3 &= q_4 \cdot r_4 + 0 \end{aligned}$$

In other words let $r_0 = a$ and $r_1 = b$ and do the obvious. A lot nicer, right? Let me write it this way with the remainder term on the lefts:

$$\begin{aligned} r_2 &= (1) \cdot r_0 + (-q_1) \cdot r_1 \\ r_3 &= (1) \cdot r_1 + (-q_2) \cdot r_2 \\ r_4 &= (1) \cdot r_2 + (-q_3) \cdot r_3 \end{aligned}$$

(Remember that r_4 is the gcd of $r_0 = 514$ and $r_1 = 24$, right?) Organized this way, we have the gcd on one side of the equation. Now if we substitute the first equation into the second we get

$$\begin{aligned} r_2 &= (1) \cdot r_0 + (-q_1) \cdot r_1 \text{ ... USED} \\ r_3 &= (1) \cdot r_1 + (-q_2) \cdot ((1) \cdot r_0 + (-q_1) \cdot r_1) \\ r_4 &= (1) \cdot r_2 + (-q_3) \cdot r_3 \end{aligned}$$

i.e.,

$$\begin{aligned} r_2 &= (1) \cdot r_0 + (-q_1) \cdot r_1 \text{ ... USED} \\ r_3 &= (-q_2) \cdot r_0 + (1 + q_1 q_2) \cdot r_1 \\ r_4 &= (1) \cdot r_2 + (-q_3) \cdot r_3 \end{aligned}$$

Note that we cannot throw away the first equation yet. We need to keep r_2 around since it appears in the third equation! So when can we throw the first equation away? Look at the general case. Suppose we have

$$\begin{aligned}
r_2 &= (1) \cdot r_0 + (-q_1) \cdot r_1 \\
r_3 &= (1) \cdot r_1 + (-q_2) \cdot r_2 \\
r_4 &= (1) \cdot r_2 + (-q_3) \cdot r_3 \\
r_5 &= (1) \cdot r_3 + (-q_4) \cdot r_4 \\
r_6 &= (1) \cdot r_4 + (-q_5) \cdot r_5 \\
&\ldots
\end{aligned}$$

Aha! r_2 is used only in the next *two* equations.

Suppose we are at equation 3:

$$\begin{aligned}
r_3 &= c_1 \cdot r_0 + d_1 \cdot r_1 \\
r_4 &= c_2 \cdot r_0 + d_2 \cdot r_1
\end{aligned}$$

We have to compute the next equation: This requires r_3, r_4. Then we have

$$r_5 = (1) \cdot r_3 + (-q_4) \cdot r_4$$

where

$$q_4 = \lfloor r_3/r_4 \rfloor, \quad r_5 = r_3 - q_4 r_4$$

Altogether we have

$$\begin{aligned}
r_3 &= c_1 \cdot r_0 + d_1 \cdot r_1 \\
r_4 &= c_2 \cdot r_0 + d_2 \cdot r_1 \\
r_5 &= (1) \cdot r_3 + (-q_4) \cdot r_4
\end{aligned}$$

The last equation becomes

$$r_5 = c_1 \cdot r_0 + d_1 \cdot r_1 + (-q_4) \cdot (c_2 \cdot r_0 + d_2 \cdot r_1)$$

i.e.

$$r_5 = (c_1 - q_4 c_2) \cdot r_0 + (d_1 - q_4 d_2) \cdot r_1$$

Let me repeat that in a slightly more general context. If we have

$$\begin{aligned}
r_3 &= c_1 \cdot r_0 + d_1 \cdot r_1 \\
r_4 &= c_2 \cdot r_0 + d_2 \cdot r_1
\end{aligned}$$

then we get (throwing away the first equation):

$$\begin{aligned} r_4 &= c_2 \cdot r_0 + d_2 \cdot r_1 \\ r_5 &= (c_1 - q_4 c_2) \cdot r_0 + (d_1 - q_4 d_2) \cdot r_1 \end{aligned}$$

To put it in terms of numbers instead of equations this is what we get: If we have

$$c_1, d_1, c_2, d_2, r_3, r_4$$

then we get

$$c_2, d_2, c_1 - \lfloor r_3/r_4 \rfloor c_2, d_1 - \lfloor r_3/r_4 \rfloor d_2, r_4, r_3 - \lfloor r_3/r_4 \rfloor r_4$$

In general, if we have

$$c, d, c', d', r, r'$$

then we get

$$c', d', c - \lfloor r/r' \rfloor c', d - \lfloor r/r' \rfloor d', r', r - \lfloor r/r' \rfloor r'$$

Of course since we start off with r_0, r_1 (i.e. what we call a and b above), we have

$$\begin{aligned} r_0 &= 1 \cdot r_0 + 0 \cdot r_1 \\ r_1 &= 0 \cdot r_0 + 1 \cdot r_1 \end{aligned}$$

i.e., you would start off with

$$c = 1, d = 0, c' = 0, d' = 1, r = r_0, r' = r_1$$

Let's check this algorithm with our $r_0 = 514, r_1 = 24$.

STEP 1: The initial numbers are

$$c = 1, d = 0, c' = 0, d' = 1, r = 514, r' = 24$$

Again this corresponds to

$$\begin{aligned} r_3 &= 1 \cdot 514 + 0 \cdot 24 \\ r_4 &= 0 \cdot 514 + 1 \cdot 24 \end{aligned}$$

STEP 2: The new numbers (6 of them) are

$$\begin{aligned}
c' &= 0 \\
d' &= 1 \\
c - \lfloor r/r' \rfloor c' &= 1 - \lfloor 514/24 \rfloor 0 = 1 \\
d - \lfloor r/r' \rfloor d' &= 0 - \lfloor 514/24 \rfloor 1 = 0 - 21 = -21 \\
r' &= 24 \\
r - \lfloor r/r' \rfloor r' &= 514 - \lfloor 514/24 \rfloor 24 = 514 - 504 = 10
\end{aligned}$$

So the new numbers (we reset the variables in the algorithm):

$$c = 0, d = 1, c' = 1, d' = -21, r = 24, r' = 10$$

These corresponds to the data on the second and third line of the following:

$$\begin{aligned}
514 &= 1 \cdot 514 + 0 \cdot 24 \\
24 &= 0 \cdot 514 + 1 \cdot 24 \\
10 &= 1 \cdot 514 + (-21) \cdot 24
\end{aligned}$$

STEP 3: From the 6 numbers from STEP 2 we get

$$\begin{aligned}
c' &= 1 \\
d' &= -21 \\
c - \lfloor r/r' \rfloor c' &= 0 - \lfloor 24/10 \rfloor 1 = -2 \\
d - \lfloor r/r' \rfloor d' &= 1 - \lfloor 24/10 \rfloor (-21) = 1 + 42 = 43 \\
r' &= 10 \\
r - \lfloor r/r' \rfloor r' &= 24 - \lfloor 24/10 \rfloor 10 = 24 - 20 = 4
\end{aligned}$$

So the new numbers (we reset the variables in the algorithm):

$$c = 1, d = -21, c' = -2, d' = 43, r = 10, r' = 4$$

These corresponds to the data on the third and fourth line of the following:

$$\begin{aligned}
514 &= 1 \cdot 514 + 0 \cdot 24 \\
24 &= 0 \cdot 514 + 1 \cdot 24 \\
10 &= 1 \cdot 514 + (-21) \cdot 24 \\
4 &= (-2) \cdot 514 + 43 \cdot 24
\end{aligned}$$

STEP 4: From the 6 numbers from STEP 3 we get

$$\begin{aligned}
c' &= -2 \\
d' &= 43 \\
c - \lfloor r/r' \rfloor c' &= 1 - \lfloor 10/4 \rfloor (-2) = 1 + 4 = 5 \\
d - \lfloor r/r' \rfloor d' &= -21 - \lfloor 10/4 \rfloor (43) = -21 - 86 = -107 \\
r' &= 4 \\
r - \lfloor r/r' \rfloor r' &= 10 - \lfloor 10/4 \rfloor 4 = 10 - 8 = 2
\end{aligned}$$

So the new numbers (we reset the variables in the algorithm):

$$c = -2, d = 43, c' = 5, d' = -107, r = 4, r' = 2$$

These corresponds to the data on the fourth and fifth line of the following:

$$\begin{aligned}
514 &= 1 \cdot 514 + 0 \cdot 24 \\
24 &= 0 \cdot 514 + 1 \cdot 24 \\
10 &= 1 \cdot 514 + (-21) \cdot 24 \\
4 &= (-2) \cdot 514 + 43 \cdot 24 \\
2 &= 5 \cdot 514 + (-107) \cdot 24
\end{aligned}$$

STEP 5: From the 6 numbers from STEP 4 we get

$$\begin{aligned}
c' &= 5 \\
d' &= -107 \\
c - \lfloor r/r' \rfloor c' &= -2 - \lfloor 4/2 \rfloor 5 = -2 - 10 = -12 \\
d - \lfloor r/r' \rfloor d' &= 43 - \lfloor 4/2 \rfloor (-107) = 43 + 214 = 257 \\
r' &= 2 \\
r - \lfloor r/r' \rfloor r' &= 4 - \lfloor 4/2 \rfloor 2 = 4 - 4 = 0
\end{aligned}$$

So the new numbers (we reset the variables in the algorithm):

$$c = 5, d = -107, c' = -12, d' = 257, r = 2, r' = 0$$

These corresponds to the data on the fifth and sixth line of the following:

$$\begin{aligned}
514 &= 1 \cdot 514 + 0 \cdot 24 \\
24 &= 0 \cdot 514 + 1 \cdot 24 \\
10 &= 1 \cdot 514 + (-21) \cdot 24 \\
4 &= (-2) \cdot 514 + 43 \cdot 24 \\
2 &= 5 \cdot 514 + (-107) \cdot 24 \\
0 &= (-12) \cdot 514 + 257 \cdot 24
\end{aligned}$$

Of course (as before) at this point, you see that the $r' = 0$. Therefore

$$\gcd(514, 24) = 2$$

and furthermore from $c = 5, d = -107$, we get

$$5 \cdot 514 + (-107) \cdot 24 = \gcd(514, 24)$$

Here's a Python implementation with some test code:

```
ALGORITHM: EEA (Extended Euclidean Algorithm)
INPUTS:    a, b
OUTPUTS:   r, c, d where r = gcd(a, b) = c * a + d * b

a0, b0 = a, b
d0, d = 0, 1
c0, c = 1, 0
q = a0 // b0
r = a0 - q * b0

while r > 0:
    d, d0 = d0 - q * d, d
    c, c0 = c0 - q * c, c

    a0, b0 = b0, r
    q = a0 // b0
    r = a0 - q * b0

r = b0
return r, c, d
```

You can pound real hard at the Extended Euclidean Algorithm with this:

```
for a in range(1, 1000):
    for b in range(1, 1000):
        gcd, x, y = EEA(a, b)
        if gcd != a * x + b * y:
            print("ERROR! ...", a, b)
```

Note that even if you have to go through 100 lines of Euclidean computations, the number of variables stay the same: besides the inputs, there are 7 variables (i.e., `q, c, d, c', d', r, r'`)

By the way, this is somewhat similar to what we call *tail recursion* (CISS445) an extremely important technique in functional programming. All LISP hackers and people working in high performance computing and compilers swear by it. You don't see recursion in the above code, but you can replace the while-loop with recursion and if you have a compiler/interpreter that can perform true tail recursion, then it would run exactly like the above algorithm.

Why is this algorithm correct? And why does it terminate? Here's the earlier list of computations:

$$\begin{aligned}
r_2 &= (1) \cdot r_0 + (-q_1) \cdot r_1 \\
r_3 &= (1) \cdot r_1 + (-q_2) \cdot r_2 \\
r_4 &= (1) \cdot r_2 + (-q_3) \cdot r_3 \\
r_5 &= (1) \cdot r_3 + (-q_4) \cdot r_4 \\
r_6 &= (1) \cdot r_4 + (-q_5) \cdot r_5 \\
&\ldots
\end{aligned}$$

where $r_0 = a$ and $r_1 = b$. First of all the r_0, r_1, r_2 comes from the Euclidean property:

$$r_0 = r_1 q_1 + r_2, \;\; 0 \leq r_2 < r_1$$

Likewise

$$r_1 = r_2 q_2 + r_3, \;\; 0 \leq r_3 < r_2$$

Etc. Hence

$$0 \leq \ldots < r_3 < r_2 < r_1$$

Clearly at some point some r_k must be 0 and furthermore this is the smallest k such taht $r_k = 0$.

Furthermore

$$\gcd(r_0, r_1) = \gcd(r_1, r_2) = \gcd(r_2, r_3) = \ldots = \gcd(r_{k-1}, r_k)$$

Assuming $r_k = 0$, then

$$\gcd(r_0, r_1) = ... = \gcd(r_{k-1}, r_k) = \gcd(r_{k-1}, 0) = r_{k-1}$$

Also, from the earlier computation, from

$$\begin{aligned} r_3 &= c_1 \cdot r_0 + d_1 \cdot r_1 \\ r_4 &= c_2 \cdot r_0 + d_2 \cdot r_1 \end{aligned}$$

we get

$$\begin{aligned} r_4 &= c_2 \cdot r_0 + d_2 \cdot r_1 \\ r_5 &= (c_1 - q_4 c_2) \cdot r_0 + (d_1 - q_4 d_2) \cdot r_1 \end{aligned}$$

More generally, each r_i is expressed as a linear combination of $r_0 = a$ and $r_1 = b$.

Exercise 1.5.4. Leetcode 365 https://leetcode.com/problems/water-and-jug-problem/description/ (Also, Die Hard 3 problem https://www.math.tamu.edu/~dallen/hollywood/diehard/diehard.htm.) You are given two jugs with capacities `jug1Capacity` and `jug2Capacity` liters. There is an infinite amount of water supply available. Determine whether it is possible to measure exactly `targetCapacity` liter using these two jugs.

debug: exercises/nt-54/question.tex

If `targetCapacity` liters of water are measurable, you must have `targetCapacity` liters of water contained within one or both buckets by the end.

Operations allowed:

1. Fill any of the jugs with water.
2. Empty any of the jugs.
3. Pour water from one jug into another till the other jug is completely full, or the first jug itself is empty.

(Go to solution, page 55) □

You'll see that there are times when you're only interested in the value of x and not y (or y and not x – the above is symmetric about x and y). Do you notice x comes from c? If you analyze the above algorithm, you see immediately that c is computed from c' and c' is computed from c, c', q, q is computed from

r, r', r is computed from r', and finally (phew!) r' is computed from r, q, r'. Therefore if you're interested in c, you don't need to compute d or d'. So you can change the EEA to this:

```
ALGORTHM: EEA2 (sort of EEA ... without the d, d0)
INPUTS: a >=0, b >=0
OUTPUTS: r, c where r = gcd(a, b) = c*a + d*b for some d

    a0, b0 = a, b
    c0, c = 1, 0
    q = a0 // b0
    r = a0 - q * b0

    while r > 0:
        c, c0 = c0 - q * c, c

        a0, b0 = b0, r
        q = a0 // b0
        r = a0 - q * b0

    r = b0
    return r, c
```

Let's test this with 514 and 24 again.

STEP 1: $c = 1, c' = 0, r = 514, r' = 24$
STEP 2: $q = 21, c = c' = 0, c' = c - qc' = 1 - 21(0) = 1, r = 24, r' = r - qr' = 514 - 21(24) = 10$.
STEP 3: $q = 2, c = c' = 1, c' = c - qc' = 0 - 2(1) = -2, r = 10, r' = r - qr' = 24 - 2(10) = 4$
STEP 4: $q = 2, c = c' = -2, c' = c - qc' = 1 - 2(-2) = 5, r = 4, r' = r - qr' = 10 - 2(4) = 2$
STEP 5: $q = 2, c = c' = 5, c' = c - qc' = -2 - 2(5) = -12, r = 2, r' = r - qr' = 4 - 2(2) = 0$

At this point $r' = 0$. Therefore the gcd is again 2 and $x = 5$. Therefore

$$5 \cdot 514 + y \cdot 24 = 2$$

for some y.

Later you'll see why we compute only x.

Exercise 1.5.5. Compute the following gcd and the Bézout's coefficients of the given numbers by following the Extended Euclidean Algorithm.

debug: exercises/nt-55/question.tex

1. $\gcd(0, 10)$
2. $\gcd(10, 0)$
3. $\gcd(10, 1)$
4. $\gcd(10, 10)$
5. $\gcd(107, 5)$
6. $\gcd(107, 26)$
7. $\gcd(84, 333)$
8. $\gcd(F_{10}, F_{11})$ where F_n is the n–th Fibonacci number. (Recall: $F_0 = 0, F_1 = 1, F_{n+2} = F_{n+1} + F_n$.)
9. $\gcd(ab, b)$
10. $\gcd(a, a + 1)$
11. $\gcd(ab + a, b)$ where $0 < a < b$. Go as far as you can.
12. $\gcd(a(a + 1) + a, (a + 1))$ where $0 < a < b$. Go as far as you can.

(Go to solution, page 56) □

Exercise 1.5.6. Prove that if $a \mid c$, $b \mid c$, and $\gcd(a, b) = 1$, then $ab \mid c$. (Go to solution, page 57) □

debug: exercises/nt-56/question.tex

Exercise 1.5.7. Prove that if $a \mid c$, $b \mid c$, then

debug: exercises/nt-57/question.tex

$$\frac{ab}{\gcd(a, b)} \mid c$$

(Go to solution, page 58) □

Exercise 1.5.8. Leetcode 920.
https://leetcode.com/problems/number-of-music-playlists
Your music player contains n different songs. You want to listen to goal songs (not necessarily different) during your trip. To avoid boredom, you will create a playlist so that:

debug: exercises/nt-58/question.tex

- Every song is played at least once.
- A song can only be played again only if k other songs have been played.

Given `n`, `goal`, and `k`, return the number of possible playlists that you can create. Since the answer can be very large, return it modulo $109 + 7$. (Go

to solution, page 59) □

Solutions

Solution to Exercise 1.5.1.

debug: exercises/nt-51/answer.tex

Solution not provided.

Solution to Exercise 1.5.2.

debug: exercises/nt-52/answer.tex

Solution not provided.

Solution to Exercise 1.5.3.

debug: exercises/nt-53/answer.tex

Solution not provided.

Solution to Exercise 1.5.4.

debug: exercises/nt-54/answer.tex

Solution not provided.

Solution to Exercise 1.5.5.

debug: exercises/nt-55/answer.tex

Solution not provided.

Solution to Exercise 1.5.6.

debug: exercises/nt-56/answer.tex

Solution not provided.

Solution to Exercise 1.5.7.

debug: exercises/nt-57/answer.tex

Solution not provided.

Solution to Exercise 1.5.8.

debug: exercises/nt-58/answer.tex

Solution not provided.

1.6 Euclidean algorithm – GCD debug: gcd.tex

Now let me use the Euclidean property to compute the gcd of two integers.

Let's use the division algorithm on 20 and 6.

$$20 = 6 \cdot 3 + 2, \quad 0 \leq 2 < 6$$

Suppose I want to compute $\gcd(20, 6)$. Of course the example is small enough that we know that it is 2. But notice something about this:

$$20 = 6 \cdot 3 + 2, \quad 0 \leq 2 < 6$$

If d is a divisor of 20 and 6, then it must also divide 2. Therefore $\gcd(20, 6)$ must divide 2. The converse might not be true. In general, we have this crucial bridge between Euclidean property and common divisors:

Exercise 1.6.1. Let $A + B + C = 0$ where $A, B, C \in \mathbb{Z}$ where $(A, B) \neq (0, 0)$ and $(B, C) \neq (0, 0)$. Prove that $\gcd(A, B) = \gcd(B, C)$.

Exercise 1.6.2. Let $A, B, C, x \in \mathbb{Z}$ with $(A, B) \neq (0, 0)$ and $(B, C) \neq (0, 0)$. If $A + Bx + C = 0$, then $\gcd(A, B) = \gcd(B, C)$.

Lemma 1.6.1. (**GCD Lemma**) *Let $a, b, q, r \in \mathbb{Z}$ such that not both a, b are zero and not both b, r are zero. If* GCD Lemma

$$a = bq + r$$

then

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

Proof. TODO □

In particular, given $a, b \in \mathbb{Z}$ where $a > b > 0$. By the Euclidean property of $\mathbb{Z}$, there exist $q, r \in \mathbb{Z}$ such that

$$a = bq + r, \ 0 \leq r < b$$

Hence

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

Note that in the above, I only require $a = bq + r$. For instance for to $\gcd(120, 15)$, I can use $120 = 1 \cdot 15 + (120 - 15)$, i.e., $a = 120, b = 15, q = 1, r = 120 - 15$. Then $\gcd(120, 15) = \gcd(15, 120 - 15) = \gcd(15, 105)$.

However if I use the division algorithm, then r is "small":

$$0 \leq r < b$$

So if you want to compute $\gcd(a, b)$, make sure $a \geq b$ (otherwise swap them). Then $\gcd(a, b) = \gcd(b, r)$ and you would have $a \geq b > r$. So instead of computing $\gcd(a, b)$, you are better off computing $\gcd(b, r)$.

But like I said, we do not need the q and r to be the quotient and remainder. For instance suppose I want to compute the GCD of 514 and 24.

$$514 = 24 \cdot 1 + (514 - 24)$$

Then

$$\gcd(514, 24) = \gcd(24, 514 - 24)$$

which gives us

$$\gcd(514, 24) = \gcd(24, 490)$$

Note that $\gcd(0, n) = n$ for any positive integer n.

Of course this gives rise to the following algorithm

```
ALGORITHM: GCD
INPUTS:    a, b
OTPUT:     gcd(a, b)

if b > a:
    swap a, b

if b == 0:
    return a
else:
    return GCD(a - b, b)
```

This only subtracts one copy of b from a. Suppose we can compute

$$a = bq + r, \quad 0 \leq r < b$$

Then

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

Of course r is the remainder when a is divided by b. Using this we rewrite the above code to get the **Euclidean Algorithm**:

Euclidean Algorithm

```
ALGORITHM: GCD (Euclidean algorithm)
INPUTS:    a, b
OUTPUT:    gcd(a, b)

if b > a:
    # To make sure that for gcd(a,b), a >= b
    swap a, b

if b == 0:
    return a
else:
    return GCD(b, a % b)
```

Note that if `a` $<$ `b`, then

```
GCD(a, b) = GCD(b, a % b) = GCD(b, a)
```

Therefore the swap is not necessary:

```
ALGORITHM: GCD (Euclidean algorithm)
INPUTS:    a, b
OUTPUT:    gcd(a, b)

if b == 0:
    return a
else:
    return GCD(b, a % b)
```

In this case, I'm assuming that `a % b` is available. As an example:

$$\begin{aligned}\gcd(514, 24) &= \gcd(24, 514\%24) = \gcd(24, 10)\\ &= \gcd(10, 24\%10) = \gcd(10, 4)\\ &= \gcd(10, 10\%4) = \gcd(10, 2)\\ &= \gcd(2, 10\%2) = \gcd(2, 0)\\ &= 2\end{aligned}$$

The above can also be done in a loop:

```
ALGORITHM: GCD (Euclidean algorithm)
INPUTS:    a, b
OUTPUT:    gcd(a, b)
```

```
while 1:
    if b == 0:
        return a
    else:
        a, b = b, a % b
```

Exercise 1.6.3. Compute the following using the Euclidean Algorithm explicitly.

debug: exercises/nt-06/question.tex

(a) $\gcd(10, 1)$
(b) $\gcd(10, 10)$
(c) $\gcd(107, 5)$
(d) $\gcd(107, 26)$
(e) $\gcd(84, 333)$

(Go to solution, page 67) □

Exercise 1.6.4. Compute the following. You should go as far as you can. In other words, either you can a fixed integer (such as 1) or derive $\gcd(\alpha, \beta)$ where α, β are as simple as possible. For instance, to simplify $\gcd(3 + 2a, a)$, since $3 + 2a = 2 \cdot a + 3$, by the GCD Lemma, we have

debug: exercises/nt-07/question.tex

$$\gcd(3 + 2a, a) = \gcd(a, 3)$$

In the following $a, b, x, n \in \mathbb{Z}$ are positive integers.

(a) $\gcd(ab, b)$
(b) $\gcd(a, a + 1)$
(c) $\gcd(ab + a, b)$ where $0 < a < b$
(d) $\gcd(a(a + 1) + a, (a + 1))$ where $0 < a < b$
(e) $\gcd(1 + x + \cdots + x^n, x)$
(f) $\gcd(F_{10}, F_{11})$ where F_n is the n–th Fibonacci number. (Recall: $F_0 = 1, F_1 = 1, F_{n+2} = F_{n+1} + F_n$ for $n \geq 0$.)

(Go to solution, page 68) □

Despite the fact that the Euclidean algorithm is one of the fastest algorithm to compute the GCD of two numbers and has been discovered by Euclid a long time ago (BC 300), the actual runtime was not known until Lamé proved in 1844 that the number of steps to compute $\gcd(a, b)$ using the Euclidean algorithm is ≤ 5 times the number of digits (in base 10 notation) of $\min(a, b)$.

For instance for the example above of $\gcd(514, 24)$, the number of digits of $\min(514, 24)$ is 2. Lamé theorem says that the number of steps made by the Euclidean algorithm in the computation of $\gcd(514, 24)$ is at most $5 \times 2 = 10$. The actual number of steps in the earlier computation

$$\begin{aligned}\gcd(514, 24) &= \gcd(24, 514\%24) = \gcd(24, 10)\\ &= \gcd(10, 24\%10) = \gcd(10, 4)\\ &= \gcd(10, 10\%4) = \gcd(10, 2)\\ &= \gcd(2, 10\%2) = \gcd(2, 0)\\ &= 2\end{aligned}$$

is 4 (not counting the base case step), i.e.,

$$\gcd(514, 24) = \gcd(24, 10) = \gcd(10, 4) = \gcd(10, 2) = \gcd(2, 0)$$

Lamé's work is generally considered the beginning of computational complexity theory, which is the study of resources needed (time or space) to execute an algorithm. Another fascinating fact about Lamé's theorem is that historically the proof below is the first ever "use" of the Fibonacci sequence.

Theorem 1.6.1. (Lamé 1844) *Let $a > b > 0$ be integers. If the GCD computation of a, b using Euclidean algorithm results in n steps:*

$$\gcd(a_{n+1}, b_{n+1}) = \gcd(a_n, b_n) = \cdots = \gcd(a_1, b_1), \;\; b_1 = 0$$

where $(a_{n+1}, b_{n+1}) = (a, b)$, and $a_i > b_i$, then

(a) *$a \geq F_{n+2}$ and $b \geq F_{n+1}$, where F_n are the Fibonacci numbers ($F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, F_5 = 5$, etc. Note that the index starts with 1.)*
(b) *n is at most 5 times the number of digits in b.*

Proof. TODO □

Proposition 1.6.1. *The number of digits of a positive integer b is*

$$\lfloor \log_{10} b + 1 \rfloor$$

Proof. TODO □

On analyzing the proof, the above is in fact true for any base > 1. In other words the number of base–B symbols to represent b is

$$\lfloor \log_B b + 1 \rfloor$$

where $B > 1$. For instance the number of bits needed to represent b is

$$\lfloor \log_2 b + 1 \rfloor$$

For instance, $b = 9_{10} = 1001_2$ which has 4 bits and

$$\lfloor \log_2 9 + 1 \rfloor = \lfloor 3.1699... + 1 \rfloor = \lfloor 4.1699... \rfloor = 4$$

Proposition 1.6.2. *Let positive integer b be written in base B where $B > 1$ is an integer. Then the number of base-B symbols used to represent b is*

$$\lfloor \log_B b + 1 \rfloor$$

Proof. TODO

Exercise 1.6.5. Leetcode 650. debug: exercises/nt-08/question.tex
https://leetcode.com/problems/2-keys-keyboard/
There is only one character `'A'` on the screen of a notepad. You can perform one of two operations on this notepad for each step:

- Copy All: You can copy all the characters present on the screen (a partial copy is not allowed).
- Paste: You can paste the characters which are copied last time.

Given an integer `n`, return the minimum number of operations to get the character `'A'` exactly `n` times on the screen. (Go to solution, page 69) □

Exercise 1.6.6. Leetcode 2447. debug: exercises/nt-09/question.tex
https://leetcode.com/problems/number-of-subarrays-with-gcd-equal-to-k
Given an integer array `nums` and an integer `k`, return the number of subarrays of `nums` where the greatest common divisor of the subarray's elements is `k`. A subarray is a contiguous non-empty sequence of elements within an array. The greatest common divisor of an array is the largest integer that evenly divides all the array elements. (Go to solution, page 70) □

Exercise 1.6.7. Leetcode 1998

debug: exercises/nt-10/question.tex

https://leetcode.com/problems/gcd-sort-of-an-array/

You are given an integer array `nums`, and you can perform the following operation any number of times on `nums`:

- Swap the positions of two elements `nums[i]` and `nums[j]` if `gcd(nums[i], nums[j]) > 1` where `gcd(nums[i], nums[j])` is the greatest common divisor of `nums[i]` and `nums[j]`.

Return true if it is possible to sort `nums` in non-decreasing order using the above swap method, or false otherwise. (Go to solution, page 71) ☐

Solutions

Solution to Exercise 1.6.3.

debug: exercises/nt-06/answer.tex

Solution not provided.

Solution to Exercise 1.6.4.

debug: exercises/nt-07/answer.tex

Solution not provided.

Solution to Exercise 1.6.5.

debug: exercises/nt-08/answer.tex

Solution not provided.

Solution to Exercise 1.6.6.

debug: exercises/nt-09/answer.tex

Solution not provided.

Solution to Exercise 1.6.7.

debug: exercises/nt-10/answer.tex

Solution not provided.

1.7 Primes debug: prime.tex

Definition 1.7.1. A prime p is a positive integer greater than 1 that is divisible by only 1 and itself. In other words $p \in \mathbb{N}$ is a **prime** if $p > 1$ and if $d \in \mathbb{Z}, d > 0, d \mid p$, then $d = 1$ or $d = p$. prime

Examples of primes are $2, 3, 5, 7, 11, 13, 17, 19, \ldots$.

Integers at least zero can be divided into the following types:

- 0 – the zero element
- 1 – the unit element (i.e. the only invertible element ≥ 0)
- primes – $2, 3, 5, 7, 11, \ldots$
- **composites** – integers > 1 which are not primes

composites

(Instead of primes of $\mathbb{N} \cup \{0\}$, it's also possible to define primes of $\mathbb{Z}$. A prime of $\mathbb{Z}$ is an integer not $-1, 0, 1$ such that if $d \mid p$, then $d = \pm 1$ or $d = \pm p$. In that case primes of $\mathbb{Z}$ are $\pm 2, \pm 3, \pm 5, \pm 7, \pm 11, \ldots$.)

For the next few theorems, you may assume various properties of integers. There is no circular arguments since since these properties does not depend on the next few theorems.

Proposition 1.7.1. (Euclid) *There are infinitely many primes.*

Proof. TODO □

Exercise 1.7.1. Prove that there are infinitely many composites. (Go to solution, page 80) □

debug: exercises/nt-11/question.tex

Exercise 1.7.2. Prove that there are infinitely many primes of the form $4k+3$. (Go to solution, page 81) □

debug: exercises/nt-12/question.tex

Exercise 1.7.3. Prove that there are infinitely many primes of the form $4k+1$. (Go to solution, page 82) □

debug: exercises/nt-13/question.tex

The above two exercises are special cases of the following theorem:

Theorem 1.7.1. (Dirichlet 1837) *If* $\gcd(a, b) = 1$*, then there are infinitely many primes of the form* $an + b$*.*

The proof of Dirichlet's theorem above uses group theory and complex analysis and is beyond the scope of this book.

Proposition 1.7.2. (Euclid) *There are arbitrarily long consecutives integers of composites.*

Proof. BEGINTODO
$n! + 2$, $n! + 3$, $n! + 4$, ..., $n! + n$ are all compositics for $n \geq 2$. This list is therefore a list of consecutive composites of length $n - 1$. ENDTODO

□

The follow lemma is extremely important and is used for instance in the fundamental theorem of arithmetic to prove the uniqueness of prime factorization in $\mathbb{N}$ (or $\mathbb{Z}$). To break tradition, we will call this a theorem (instead of a lemma):

Theorem 1.7.2. (**Euclid's lemma**) *If* p *is a prime and* $p \mid ab$*, then either* $p \mid a$ *or* $p \mid b$*.*

Euclid's lemma

Proof. TODO □

The above generalizes easily (by induction) to the following:

Corollary 1.7.1. *If* p *is a prime and* $p \mid a_1 a_2 \cdots a_n$*, then* p *divides at least one of the* $a_1, \ldots, a_n$*.*

Proof. TODO □

Theorem 1.7.3. (Euclid's **Fundamental Theorem of Arithmetic**) *Every positive integer* > 1 *can be written as a unique product of primes up to permutation of the prime factors. This means*

Fundamental Theorem of Arithmetic

(a) *If* $n > 1$ *is an integer, then* n *can be written as a product of primes.*

(b) *If n is written as two products of primes:*

$$n = p_1 p_2 \cdots p_k = q_1 q_2 \cdots q_\ell$$

where p_i and q_j are primes arranged in ascending order, i.e.,

$$\begin{aligned} p_1 \leq p_2 \leq \cdots \leq p_k \\ q_1 \leq q_2 \leq \cdots \leq q_\ell \end{aligned}$$

then $k = \ell$ and

$$p_1 = q_1, \quad p_2 = q_2, \quad \cdots, \quad p_k = q_k,$$

Proof. TODO □

The statement of the Fundamental Theorem of Arithmetic can include the case of $n = 1$ if we accept that the product of an empty tuple of integers is 1:

$$\prod_{p \in ()} p = 1$$

Furthermore the statement can be slightly modified to include negative integers: Every nonzero integers can be written uniquely in the form

$$u \prod_{i=1}^{g} p_i^{e_i}$$

where $u = \pm 1$ (a unit), p_i are distinct primes, $e_i > 0$ are integers.

Proposition 1.7.3. *Let $a = \prod_{p \in P} p^{a_p}$, $b = \prod_{p \in P} p^{b_p}$ and $c = \prod_{p \in P} p^{c_p}$ where P is a finite set of primes. Then*

(a) $c = ab \implies c_p = a_p + b_p$.
(b) $a \mid b \implies a_p \leq b_p$ *for all* $p \in P$.
(c) $c = \gcd(a, b) \implies c_p = \min(a_p, b_p)$.
(d) $c = \operatorname{lcm}(a, b) \implies c_p = \max(a_p, b_p)$.
(e) $\gcd(a, b) \cdot \operatorname{lcm}(a, b) = ab$

Here, $\operatorname{lcm}(ab)$ is the **lowest common multiple** of a, b, i.e., it is the smallest positive integer m such that $a \mid m, b \mid m$.

lowest common multiple

The above assumes the easily proven facts that

$$\prod_{p \in P} p^{a_p} \prod_{p \in P} p^{b_p} = \prod_{p \in P} p^{a_p + b_p}$$

(by commutativity of $\cdot$) and

$$\prod_{p \in P} p^{a_p} = \prod_{p \in P} p^{b_p} \implies a_p = b_p \text{ for all } p \in P$$

by uniqueness of prime factorization from the Fundamental Theorem of Arithmetic. Here, P is a set of distinct primes.

Proposition 1.7.4. *If $n > 1$ is not a prime, then there is a prime factor p such that $p \leq \sqrt{n}$.*

Proof. TODO □

Therefore a very simple primality test algorithm for n is the following:

```
ALGORITHM: BRUTE-FORCE-PRIMALITY-TEST
INPUT: n
OUTPUT: true if n is prime. If n < 2, false is returned.

if n < 2: return false

d = 2
while d <= sqrt(n):
    if n % d == 0:
        return false
    d = d + 1
return true
```

In terms of n, the runtime is $O(\sqrt{n})$. However in terms of the bits of b of n, by Proposition 1.5.2,

$$b = \lfloor \log_2(n+1) \rfloor = \log_2(n+1) - \alpha$$

where $0 \leq \alpha < 1$. Hence

$$n = 2^b 2^\alpha - 1$$

Hence in terms of the number of bits of n, the runtime is

$$O(\sqrt{n}) = O(\sqrt{2^b 2^\alpha - 1}) = O(2^{b/2})$$

It is common to denote the number of bits of the input by n. Hence the runtime in number of bits n is $O(2^{n/2})$, i.e., it has **exponential runtime with linear exponent**.

exponential runtime with linear exponent

$O(\sqrt{n})$ is said to be the **pseudo-polynomial runtime** of the algorithm to indicate that the n is the numeric input and not a correct measure of the complexity of the input, which should be in number of bits of the input.

pseudo-polynomial runtime

Exercise 1.7.4. Let $P = \{p_1, ..., p_n\}$ be a set of distint primes. Consider the expression $N(P) = \prod_{p \in P} p + 1$ in Euclid's proof of infinitude of primes. This is sometimes called Euclid's construction. How often is this a prime? (Go to solution, page 83) □

debug: exercises/nt-15/question.tex

Exercise 1.7.5. Let $P = \{p_1, ..., p_n\}$ be a set of distint primes. Carry out the Euclid's construction on all possible subsets of P. If a Euclid construction is a prime, put that prime into P. If it does not, put the smallest prime factor into P. Repeat. For instance if you start with $P = \{\}$, you'll get 2 and the new P is $\{2\}$. Next you'll get $P = \{2, 3\}$. The Euclid constructions you get from P are $2, 3, 4, 7$. So the next P is $\{2, 3, 7\}$. At the next stage you get $2, 3, 4, 8, 7, 15, 22, 43$. This means that the next P is $\{2, 3, 7, 43\}$. Etc. Does your P always grow? Notice that your P's so far does not capture 5. When, if at all, will 5 appear? (Go to solution, page 84) □

debug: exercises/nt-16/question.tex

Exercise 1.7.6. In the above, if an Euclid construction does not give you a prime, you take the smallest prime factor. What if you pick the largest prime factor? (Go to solution, page 85) □

debug: exercises/nt-17/question.tex

The following are some DIY exercises for self-study on famous unsolved problems in number theory. You might want to write programs to check on the conjectures.

Exercise 1.7.7. Are there infinitely many primes p such that $p + 2$ is also a prime? If p and $p+2$ are both primes, then they are called **twin primes**. The **twin prime conjecture** states that there are infinitely many twin primes. (See `https://en.wikipedia.org/wiki/Twin_prime`.) Write a program that prints $p, p+2$ if both are primes. Print the time elapsed between the discovery of pairs of twin primes. (Go to solution, page 86) □

debug: exercises/nt-18/question.tex
twin primes

twin prime conjecture

Exercise 1.7.8. Can even positive integer be written as the sum of two primes? When the two primes are > 2, then the sum of these two primes is even. The **Goldbach conjecture** states that every positive integer can be written as the sum of two primes. (See https://en.wikipedia.org/wiki/Goldbach%27s_conjecture.) Write a program that prints $n \geq 2$ as a sum of two primes as n iterates from 2 to a huge positive integer N entered by the user. (Go to solution, page 87) □

debug: exercises/nt-19/question.tex

Goldbach conjecture

Exercise 1.7.9. A positive integer is a **perfect** number if it is the sum of its positive divisors strictly less than itself. For instance 6 is perfect since $6 = 1 + 2 + 3$. 28 is also perfect since $28 = 1 + 2 + 4 + 7 + 14$. Euclid knew that if p is prime and $2^p - 1$ is a prime, then $2^{p-1}(2^p - 1)$ is an even perfect number. If p is a prime and $2^p - 1$ is a also a prime, then $2^p - 1$ is called a **Mersenne prime**. Almost 2000 years after Euclid, Euler proved the converse, that every even perfect number must be of the form $2^{p-1}(2^p - 1)$ where $2^p - 1$ is a Mersenne prime. There are two famous unsolved problems in number theory on perfect numbers:

debug: exercises/nt-20/question.tex

1. Are there odd perfect numbers?
2. Are there infinitely many perfect numbers?

(See https://en.wikipedia.org/wiki/Perfect_number.) There is an ongoing search for primes and Mersenne primes using computers. At this point (2023), the top 8 largest known primes are all Mersenne primes. (See https://en.wikipedia.org/wiki/Great_Internet_Mersenne_Prime_Search.) As of Feb 2023, the largest known prime is $2^{82,589,933} - 1$. (See https://en.wikipedia.org/wiki/Largest_known_prime_number.) Write a program that prints perfect numbers, printing the time between pairs of perfect numbers discovered. (Go to solution, page 88) □

Exercise 1.7.10. Leetcode 204
https://leetcode.com/problems/count-primes/
Given an integer n, return the number of prime numbers that are strictly less than n. (Go to solution, page 89) □

debug: exercises/nt-21/question.tex

Exercise 1.7.11. Leetcode 507
https://leetcode.com/problems/perfect-number/
A perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself. A divisor of an integer `x` is an integer

debug: exercises/nt-22/question.tex

that can divide x evenly. Given an integer n, return true if n is a perfect number, otherwise return false. (Go to solution, page 90) □

Exercise 1.7.12. Leetcode 866
https://leetcode.com/problems/prime-palindrome/
Given an integer n, return the smallest prime palindrome greater than or equal to n. An integer is prime if it has exactly two divisors: 1 and itself. Note that 1 is not a prime number. For example, 2, 3, 5, 7, 11, and 13 are all primes. An integer is a palindrome if it reads the same from left to right as it does from right to left. For example, 101 and 12321 are palindromes. The test cases are generated so that the answer always exists and is in the range $[2, 2 * 108]$. (Go to solution, page 91) □

debug: exercises/nt-23/question.tex

Exercise 1.7.13. Leetcode 1175
https://leetcode.com/problems/prime-arrangements/ Return the number of permutations of 1 to n so that prime numbers are at prime indices (1-indexed.) (Recall that an integer is prime if and only if it is greater than 1, and cannot be written as a product of two positive integers both smaller than it.) Since the answer may be large, return the answer modulo $10^9 + 7$. (Go to solution, page 92) □

debug: exercises/nt-24/question.tex

Exercise 1.7.14. Leetcode 1362
https://leetcode.com/problems/closest-divisors/
Given an integer num, find the closest two integers in absolute difference whose product equals num + 1 or num + 2. Return the two integers in any order. (Go to solution, page 93) □

debug: exercises/nt-25/question.tex

Exercise 1.7.15. Leetcode 1390
https://leetcode.com/problems/four-divisors/
Given an integer array nums, return the sum of divisors of the integers in that array that have exactly four divisors. If there is no such integer in the array, return 0. (Go to solution, page 94) □

debug: exercises/nt-26/question.tex

Exercise 1.7.16. Leetcode 2523
https://leetcode.com/problems/closest-prime-numbers-in-range/
Given two positive integers left and right, find the two integers num1 and num2 such that:

debug: exercises/nt-27/question.tex

- `left <= nums1 < nums2 <= right`
- `nums1` and `nums2` are both prime numbers.
- `nums2 - nums1` is the minimum amongst all other pairs satisfying the above conditions.

Return the positive integer array `ans = [nums1, nums2]`. If there are multiple pairs satisfying these conditions, return the one with the minimum nums1 value or `[-1, -1]` if such numbers do not exist. A number greater than 1 is called prime if it is only divisible by 1 and itself. (Go to solution, page 95)
□

Exercise 1.7.17. Leetcode 2521 debug: exercises/nt-28/question.tex
`https://leetcode.com/problems/distinct-prime-factors-of-product-of-array/`
Given an array of positive integers `nums`, return the number of distinct prime factors in the product of the elements of `nums`. (Go to solution, page 96) □

Exercise 1.7.18. Leetcode 1071 debug: exercises/nt-29/question.tex
`https://leetcode.com/problems/greatest-common-divisor-of-strings/`
For two strings s and t, we say "t divides s" if and only if $s = t + \cdots + t$ (i.e., t is concatenated with itself one or more times). Given two strings `str1` and `str2`, return the largest string `x` such that `x` divides both `str1` and `str2`. (Go to solution, page 97) □

Exercise 1.7.19. Verify by hand or by using a program that if

$$P(x) = x^2 - x + 41$$

then $P(0), P(1), ..., P(40)$ are all primes. This "prime generating" polynomial was first discovered by Euler. Find positive integer n such that $x^2 - x + n$ are primes for $0 \leq x < n$. Such an n is called an Euler lucky number. 41 is an example of an Euler lucky number.

Exercise 1.7.20. Prove that if $P(x)$ is a non-constant polynomial, then it is impossible for all $P(0)$, $P(1)$, $P(2)$, ... to be prime. In fact there are infinitely many n such that $P(n)$ are all composites.

Solutions

Solution to Exercise 1.7.1.

debug: exercises/nt-11/answer.tex

Solution not provided.

Solution to Exercise 1.7.2.

debug: exercises/nt-12/answer.tex

Solution not provided.

Solution to Exercise 1.7.3.

debug: exercises/nt-13/answer.tex

Solution not provided.

Solution to Exercise 1.7.4.

debug: exercises/nt-15/answer.tex

Solution not provided.

Solution to Exercise 1.7.5.

debug: exercises/nt-16/answer.tex

Solution not provided.

Solution to Exercise 1.7.6.

debug: exercises/nt-17/answer.tex

Solution not provided.

Solution to Exercise 1.7.7.

debug: exercises/nt-18/answer.tex

Solution not provided.

Solution to Exercise 1.7.8.

debug: exercises/nt-19/answer.tex

Solution not provided.

Solution to Exercise 1.7.9.

debug: exercises/nt-20/answer.tex

Solution not provided.

Solution to Exercise 1.7.10.

debug: exercises/nt-21/answer.tex

Solution not provided.

Solution to Exercise 1.7.11.

debug: exercises/nt-22/answer.tex

Solution not provided.

Solution to Exercise 1.7.12.

debug: exercises/nt-23/answer.tex

Solution not provided.

Solution to Exercise 1.7.13.

debug: exercises/nt-24/answer.tex

Solution not provided.

Solution to Exercise 1.7.14.

debug: exercises/nt-25/answer.tex

Solution not provided.

Solution to Exercise 1.7.15.

debug: exercises/nt-26/answer.tex

Solution not provided.

Solution to Exercise 1.7.16.

debug: exercises/nt-27/answer.tex

Solution not provided.

Solution to Exercise 1.7.17.

debug: exercises/nt-28/answer.tex

Solution not provided.

Solution to Exercise 1.7.18.

debug: exercises/nt-29/answer.tex

Solution not provided.

1.8 Multiplicative inverse in $\mathbb{Z}/N$ debug: multiplicative-inverse-mod-N.tex

Quick review: In algebra classes, lots of time is spent on solving equations. You usually start with something like: "Find the roots of $x+b$". This means finding a value for x such that

$$x + a = 0$$

Of course this is dead easy. As long as you have the concept of $-a$ (additive inverse) you just add $-a$ to both sides and voila:

$$\begin{aligned}(x+a)+(-a) &= 0+(-a)\\ x+(a+(-a)) &= 0+(-a)\\ x+0 &= 0+(-a)\\ x &= 0+(-a)\\ x &= -a\end{aligned}$$

Next up, you usually see this: "Find the roots of $ax+b$" which is the same as finding a value for x such that

$$ax + b = 0$$

Assuming you are working in $\mathbb{R}$, you would do something like this:

$$\begin{aligned}(ax+b)+(-b) &= 0+(-b)\\ ax+(b+(-b)) &= 0+(-b)\\ ax+0 &= 0+(-b)\\ ax &= 0+(-b)\\ ax &= -b\end{aligned}$$

and at this point you would do this:

$$\begin{aligned}ax &= -b\\ a^{-1}(ax) &= a^{-1}(-b)\\ (a^{-1}a)x &= a^{-1}(-b)\\ 1x &= a^{-1}(-b)\\ x &= a^{-1}(-b)\end{aligned}$$

and vóila (again) and you're done.

In the case of real numbers (or even just fractions), if you're given a number a

(which is not zero), you always have another number called a^{-1} (multiplicative inverse) such that

$$a \cdot a^{-1} = 1 = a^{-1} \cdot a$$

The reason why we like this is exactly because it allows us to solve the above equation.

Given a number a, the number $-a$ is called an **additive inverse** if it behaves like this: additive inverse

$$a + (-a) = 0 = (-a) + a$$

The number a^{-1} is called a **multiplicative inverse** of a if it does this: multiplicative inverse

$$a \cdot a^{-1} = 1 = a^{-1} a$$

If a has a multiplicative inverse, we also say that a is invertible.

Definition 1.8.1. We also say that a is a **unit** if a has a multiplicative inverse. unit

Definition 1.8.2. The set of multiplicatively invertible elements of $\mathbb{Z}/N$ is denoted by $(\mathbb{Z}/N)^\times$ or $U(\mathbb{Z}/N)$. Likewise the multiplicatively invertible elemnts of $\mathbb{Z}$ is denoted by $\mathbb{Z}^\times$ or $U(\mathbb{Z})$. Likewise we have $\mathbb{Q}^\times$, $\mathbb{R}^\times$, and $\mathbb{C}^\times$.

Almost all values in $\mathbb{R}$ are invertible; the only exception being 0. Therefore $\mathbb{R}^\times = \mathbb{R} - \{0\}$. On the other hand, note that most numbers in $\mathbb{Z}$ do not have multiplicative inverses. In fact the only numbers in $\mathbb{Z}$ that have inverses are 1 and -1. So the only units in $\mathbb{Z}$ are $1, -1$:

$$U(\mathbb{Z}) = \{-1, 1\}$$

Proposition 1.8.1. *The only units of $\mathbb{Z}$ are 1, -1. In particular, $1^{-1} = 1$ and $(-1)^{-1} = -1$.*

Proof. This was already proven earlier. The following is a different proof that uses the fundamental theorem of arithmetic. TODO □

Recall from the section on congruences, we have a relation $\equiv \pmod{N}$ that relates values of $\mathbb{Z}$. For instance when $N = 26$,

$$3 \equiv 29 \pmod{26}$$

and

$$3 \equiv -26 \pmod{26}$$

You can think of $\equiv \pmod{26}$ as a kind of equality on $\mathbb{Z}$ so that, in mod 26, 3 and 29 are the same and 3 and -26 are the same. In this context, write $\mathbb{Z}/N$ for the set

$$\mathbb{Z}/N = \{0, 1, 2, ..., N-1\}$$

We call the set $\mathbb{Z}/N$ "$\mathbb{Z}$ mod N". Note that this set $\mathbb{Z}/N$ also contains N, but N is "the same as" 0, i.e., N is just another way to write 0 in the sense that

$$N \equiv 0 \pmod{N}$$

Note that $\mathbb{Z}/N$ also has $+$, $\cdot$ and has 0 and 1.

In fact $(\mathbb{Z}/N, +, \cdot, 0, 1)$ satisfies the algebraic properties of $+$, properties of $\cdot$, and distributivity. If $N > 1$, $(\mathbb{Z}/N, +, \cdot, 0, 1)$ also satisfies the nontriviality axiom, i.e., $0 \not\equiv 1 \pmod{N}$. However $(\mathbb{Z}/N, +, \cdot, 0, 1)$ does not satisfy the integrality axiom. Furthermore there's no concept of order $<$ on $\mathbb{Z}/N$ and hence there's no WOP or induction on $\mathbb{Z}/N$.

The above will be made precise in the section on congruence classes.

Later we will define the concept of commutative rings which are exactly sets, each with its own $+$ and $\cdot$ operations, satisfying the properties of $+$ and $\cdot$ and distributivity of $\mathbb{Z}$. $\mathbb{Z}$ and $\mathbb{Z}/N$ are both commutative rings.

The ability for a number in some commutative ring to have a multiplicative inverse (in that ring) is special.

While $\mathbb{Z}$ has only two units, $\mathbb{Z}/N$ ($N > 1$) might contain lots of units. For instance look at $\mathbb{Z}/10$. Is 3 invertible mod 10? In other words is there some x such that

$$3x \equiv 1 \pmod{10}$$

$x \equiv 7 \pmod{10}$ satisfies the above. In other words in mod 10, 7 is the multiplicative inverse of 3.

(You can talk about $\mathbb{Z}/N$ for $N = 1$, but it's not very interesting nor useful. $\mathbb{Z}/1$ has only one value, i.e., 0 which behaves like the additive neutral element and the multiplicative neutral element.)

Here's the obvious brute algorithm to find the multiplicative inverse of an integer mod N:

```
ALGORITHM: brute-force-inverse
INPUT: a, N
OUTPUT: x such that a * x % N is 1

for x = 1, 2, 3, ..., N - 1:
    if a * x % N == 1:
        return x
return None
```

Note that in the above algorithm we need not test 0.

Exercise 1.8.1.

debug: exercises/nt-59/question.tex

(a) Find all the units and their multiplicative inverses in $\mathbb{Z}/10$.
(b) Find all the units and their multiplicative inverses in $\mathbb{Z}/5$.
(c) Find all the units and their multiplicative inverses in $\mathbb{Z}/6$.

Do you see a pattern? □ (Go to solution, page 105) □

It turns out that in $\mathbb{Z}/N$, you can easily find all the elements with multiplicative inverses: $a \in \mathbb{Z}/N$ has a multiplicative inverse if $\gcd(a, N) = 1$.

Proposition 1.8.2. *Let a, N be integers where $N > 0$. Then a is (multiplicatively) invertible in $\mathbb{Z}/N$ iff* $\gcd(a, N) = 1$.

If m, n are two integers such that $\gcd(m, n) = 1$, we say that m and n are **coprime**.

coprime

Proof. TODO □

The above immediately implies that

$$\begin{aligned}(\mathbb{Z}/N)^\times &= \{a \mid 0 \leq a \leq N-1,\ \ a \text{ is invertible mod } N\} \\ &= \{a \mid 0 \leq a \leq N-1,\ \ \gcd(a, N) = 1\}\end{aligned}$$

(Of course $\gcd(0, N) = N > 0$, so we can throw away 0 in the above if $N > 1$.) So the key is the computation of x, y such that

$$1 = \gcd(a, N) = ax + Ny$$

which is achieved by the Extended Euclidean Algorithm. But wait a minute. I just need the x. So I'll just modify the Extended Euclidean Algorithm slightly.

Here's the (earlier) EEA algorithm, slightly modified, together with a function to compute the multiplicative inverse of $a \pmod{N}$:

```
ALGORITHM: EEA2 (sort of EEA ... without the d, d0)
INPUTS: a, b
OUTPUTS:  r, c where r = gcd(a, b) = c*a + d*b for some d

a0, b0 = a, b
c0, c = 1, 0
q = a0 // b0
r = a0 - q * b0

while r > 0:
    c, c0 = c0 - q * c, c

    a0, b0 = b0, r
    q = a0 // b0
    r = a0 - q * b0

r = b0
return r, c


ALGORITHM: mod-inverse
INPUTS: a, N
OUTPUT: x such that  (a * x) % N is 1

g, x = EEA2(a, N)
if g == 1:
    return x % N
else:
    return None
```

Example 1.8.1. Does 135 have an inverse mod 1673? If it does, find it using the Extended Euclidean Algorithm.

Solution.

1. c0, c, q, r: 1, 0, 0, 135
2. c0, c, q, r: 0, 1, 12, 53
3. c0, c, q, r: 1, -12, 2, 29
4. c0, c, q, r: -12, 25, 1, 24
5. c0, c, q, r: 25, -37, 1, 5

6. c0, c, q, r: -37, 62, 4 4
7. c0, c, q, r: 62, -285, 1, 1
8. c0, c, q, r: -285, 347, 4, 0
9. r, c: 1, 347

Therefore $135^{-1} \pmod{1673}$ is 347. And we check:

$$135 \cdot 347 = 34845 = 1 + 28 \cdot 1673 \equiv 1 \pmod{1673}$$

Exercise 1.8.2. Compute the $\gcd(16, 123)$. If it's 1, find x such that $16x \equiv 1 \pmod{123}$. Use the above version of Extended Euclidean Algorithm and compute by hand. When you're done, write a program implementing the above algorithm and check that it gives you the same result. Solve the equation

debug: exercises/nt-60/question.tex

$$16x + 5 \equiv 0 \pmod{123}$$

i.e. find an integer x such that $0 \leq x < 123$ satisfying the above congruence. (Go to solution, page 106) □

Exercise 1.8.3. In your `Zmod.py`, complete the following methods:

debug: exercises/nt-61/question.tex

- multiplicative inverse mod N (i.e., `inv`)
- invertibility mod N (i.e., `is_invertible`)
- division (i.e., `__div__`)

(Go to solution, page 107) □

Exercise 1.8.4. Leetcode 1622.
`https://leetcode.com/problems/fancy-sequence/description/`
Write an API that generates fancy sequences using the `append`, `addAll`, and `multAll` operations. Implement the `Fancy` class:

debug: exercises/nt-62/question.tex

- `Fancy()` Initializes the object with an empty sequence.
- `void append(val)` Appends an integer `val` to the end of the sequence.
- `void addAll(inc)` Increments all existing values in the sequence by an integer `inc`.
- `void multAll(m)` Multiplies all existing values in the sequence by an integer `m`.
- `int getIndex(idx)` Gets the current value at index `idx` (0-indexed) of the sequence modulo $109 + 7$. If the index is greater or equal than the

length of the sequence, return -1.

(Go to solution, page 108) $\square$

Solutions

Solution to Exercise 1.8.1.

debug: exercises/nt-59/answer.tex

Solution not provided.

Solution to Exercise 1.8.2.

debug: exercises/nt-60/answer.tex

Solution not provided.

Solution to Exercise 1.8.3.

debug: exercises/nt-61/answer.tex

Solution not provided.

Solution to Exercise 1.8.4.

debug: exercises/nt-62/answer.tex

Solution not provided.

1.9 Euler totient function debug: euler-totient-function.tex

Definition 1.9.1. Let N be a positive integer. $\phi(N)$ is the number of positive integers from 0 to $N-1$ which are coprime to N, i.e.

$$\phi(N) = |\{a \,|\, 0 \leq a \leq N-1,\ \ \gcd(a, N) = 1\}|$$

Note that you can also view ϕ in this way:

$$\begin{aligned}\phi(N) &= |\{a \mid 0 \leq a \leq N-1,\ \ \gcd(a, N) = 1\}| \\ &= |\{a \mid 0 \leq a \leq N-1,\ \ a \text{ is invertible mod } N\}| \\ &= |(\mathbb{Z}/N)^\times|\end{aligned}$$

Recall that $\{a \,|\, 0 \leq a \leq N-1,\ \ \gcd(a, N) = 1\}$ is the set of units of $\mathbb{Z}/N$ which is denoted by $(\mathbb{Z}/N)^\times$ or $U(\mathbb{Z}/N)$. So the above definition is the same as

$$\phi(N) = |U(\mathbb{Z}/N)|$$

Note that by definition $\phi(1) = 1$. Here are some important properties of ϕ.

Proposition 1.9.1. *Let $n > 0$ be a positive integer.*

(a) *Then*

$$\phi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right)$$

where "$\prod_{p|n}$" is "product over all primes p dividing n".

(b) *If m, n are coprime, i.e. $\gcd(m, n) = 1$, then $\phi(mn) = \phi(m)\phi(n)$.*

(c) *If p is a prime and $k > 0$, then $\phi(p^k) = p^{k-1}(p-1) = p^k - p^{k-1}$.*

Proof. TODO □

Let's compute $\phi(10)$. Note that $10 = 2 \cdot 5$ and $\gcd(2, 5) = 1$. Therefore using (b) of the above theorem we get

$$\phi(10) = \phi(2^1 \cdot 5^1) = \phi(2^1) \cdot \phi(5^1)$$

since $\gcd(2^1, 5^1) = 1$. Using (c) of the above theorem I get

$$\phi(10) = \phi(2^1) \cdot \phi(5^1) = (2^1 - 2^{1-1}) \cdot (5^1 - 5^{1-1}) = 1 \cdot 4 = 4$$

Of course you can also use (a) above to get

$$\phi(10) = 10 \cdot (1 - 1/2) \cdot (1 - 1/5) = 10 \cdot \frac{1}{2} \cdot \frac{4}{5} = 4$$

In both cases, we get 4. This means in $\mathbb{Z}/10$, there are 4 elements which are invertible. Running a through $\{0, 1, 2, ..., 9\}$, you'll see that invertible a's are $1, 3, 7, 9$ with inverses $1, 7, 3, 9$ respectively.

Exercise 1.9.1.

debug: exercises/nt-30/question.tex

(a) Compute $\phi(735)$.
(b) Compute $\phi(900)$.
(c) Compute $\phi(263891)$.

(Go to solution, page 117) □

Exercise 1.9.2.

debug: exercises/nt-31/question.tex

(a) Let p be a prime. What is $\phi(2p)$ in terms of p?
(b) How many solutions are there to $\phi(n) = 2n$?
(c) How many solutions are there to $\phi(n) = n/2$?

(Go to solution, page 118) □

Exercise 1.9.3. Easy: What is $\phi(pq)$ as an integer expression involving p and q? Can you write it as an expression involving the sum and product of p and q? (i.e., besides constants and operators, your expression contains only $p + q$ and pq).

debug: exercises/nt-32/question.tex

BEGIN COMMENT

END COMMENT

$(p - 1)(q - 1) = pq - (p + q) + 1$ (Go to solution, page 119) □

Exercise 1.9.4.

debug: exercises/nt-33/question.tex

(a) Solve $\phi(n) = 2$, i.e., find all positive integers n such that $\phi(n) = 2$. (Hint: Write down the prime factorization of $n = p_1^{e_1} \cdots p_g^{e_g}$ and use the equation $\phi(n) = 2$.)
(b) Solve $\phi(n) = 3$.
(b) Solve $\phi(n) = 6$.

(Go to solution, page 120) □

Exercise 1.9.5. Prove that

debug: exercises/nt-34/question.tex

$$\phi(mn) = \phi(m)\phi(n) \cdot \frac{g}{\phi(g)}$$

where $g = \gcd(m, n)$. (Note that the above does *not* assume m, n are coprime. What is the above if m, n are coprime?)

BEGIN COMMENT END COMMENT

(Go to solution, page 121) □

Exercise 1.9.6. * Plot a function of the graph $y = \phi(x)$ for integer values of x running through 1 to 10000. See any pattern? Note that if you want an approximation (for instance in the asymptotics) that's not too difficult. Note the following: There are two ways to compute $\phi(n)$:

debug: exercises/nt-35/question.tex

(a) $\phi(n) = |\{x \mid 0 \leq x < n-1, \ \gcd(x, n) = 1\}|$ which required Euclidean algorithm in a loop.
(b) $\phi(n) = \phi(p_1^{\alpha_1} \cdots p_g^{\alpha_g}) = (p_1^{\alpha_1} - p_1^{\alpha_1 - 1}) \cdots (p_1^{\alpha_g} - p_1^{\alpha_g - 1})$ which requires prime factorization.

The first method is slow: you need to loop your x and for each x you need to execute the EEA which has a loop. The second method is fast only if you can find the prime factorization of n. Is it possible to find $\phi(n)$ as a formula in n without finding the prime factorization of n? For instance the factorial function $n!$ has this interpolation: If

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} \, dt$$

then $\Gamma(n+1) = n!$ for positive integers n. Is there a fast interpolation of $\phi(n)$?

BEGIN COMMENT

https://proofwiki.org/wiki/Asymptotic_Growth_of_Euler_Phi_Function
(Go to solution, page 122) □

END COMMENT

Exercise 1.9.7. * Can you find an n such that $\phi(n)$ divides $n + 1$?

debug: exercises/nt-36/question.tex
BEGIN COMMENT END COMMENT

(Go to solution, page 123) □

Exercise 1.9.8. *

debug: exercises/nt-37/question.tex

(a) If p is a prime, then $\phi(p) = p - 1$. Prove that if $\phi(n) = n - 1$, then n is a prime.
(b) Can you find an $n > 1$ which is not a prime (i.e. composite) and such that $\phi(n)$ divides $n - 1$? If you can find one, let me know ASAP. Or if you can prove that such as n does not exist, let me know ASAP.

BEGIN COMMENT END COMMENT

(Go to solution, page 124) □

Exercise 1.9.9. *

debug: exercises/nt-38/question.tex

(a) Can you find some n such that

$$\phi(\phi(n)) = 1$$

(b) Write $\phi^2(n) = \phi(\phi(n))$. Can you find *all* n such that $\phi^2(n) = 1$.
(c) Write ϕ^k to the composition of k Euler ϕ. What about $\phi^3(n) = 1$? Can you find some n satisfying the above equation?
(d) What about $\phi^k(2^n) = 1$? What is the smallest k for $\phi^k(2^n) = 1$?
(e) What about $\phi^k(2^m \cdot 3^n) = 1$? What is the smallest k such that $\phi^k(2^m \cdot 3^n) = 1$?

(Go to solution, page 125) □

As an aside, note that ϕ as a function has domain of $\mathbb{N}$. In this case, we say that ϕ is an **arithmetic function**. Furthermore, ϕ satisfies the property that if $\gcd(m, n) = 1$, then $\phi(mn) = \phi(m)\phi(n)$. A function $\mathbb{N} \to \mathbb{C}$ satisfying this property is said to be **multiplicative** . The Euler ϕ function is one of many multiplicative arithmetic functions. Multiplicative functions are extremely important in number theory.

arithmetic function

multiplicative

The following theorem allows you to compute powers in mod p extremely fast:

Theorem 1.9.1. (**Fermat's Little Theorem**). *Let p is a prime number and a be a positive integer not divisible by p. Then*

Fermat's Little Theorem

$$a^{p-1} \equiv 1 \pmod p$$

Proof. TODO □

Exercise 1.9.10. Compute r where r is the smallest positive integer satisfying

debug: exercises/nt-39/question.tex

$$5^{642} \equiv r \pmod{641}$$

(Go to solution, page 126) □

Corollary 1.9.1. *Let p be a prime. Then $a^p \equiv a \pmod{p}$.*

Note that the corollary does not require $p \nmid a$. The proof of the corollary is easy.

Proof. TODO □

Exercise 1.9.11. What is the remainder of $3^{122436481}$ mod 13? (Go to solution, page 127) □

debug: exercises/nt-40/question.tex

Note that Fermat's Little Theorem can be used to compute powers very rapidly if you work in mod p where p is a prime. What if you need to work in mod N where N is not a prime? There is a generalization of Fermat's Little Theorem due to Euler. Note that since $p - 1 = \phi(p)$, Fermat's Little Theorem

$$a^{p-1} \equiv 1 \pmod{p}$$

can be stated as

$$a^{\phi(p)} \equiv 1 \pmod{p}$$

This statement actually holds if p is replaced by any positive integer.

Theorem 1.9.2. (Euler's Theorem). *Let a and n be positive integers such that* $\gcd(a, n) = 1$. *Then*

Euler's Theorem

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

Proof. TODO □

Make sure you step back and take in the whole proof and make your head one size larger. Look at the place where you used $\gcd(a, n) = 1$.

Note that for $\mathbb{Z}/N$, a computation of

$$a^k \pmod{N}$$

Euler's Theorem can lower the k if $\gcd(a, N)$. But after you have lowered the k, say to ℓ, you still need to compute $a^\ell \pmod N$. See the squaring algorithm in the RSA chapter.

Exercise 1.9.12. Compute r where r is the smallest positive integer satisfying

debug: exercises/nt-41/question.tex

$$5^{642} \equiv r \pmod{640}$$

(Go to solution, page 128) □

Exercise 1.9.13. What is the remainder of $3^{123456789}$ mod 100? (Go to solution, page 129) □

debug: exercises/nt-42/question.tex

Exercise 1.9.14. What is the hundreds digit of $3^{123456789}$? (Go to solution, page 130) □

debug: exercises/nt-43/question.tex

Exercise 1.9.15. In your `Zmod.py` complete the following:

debug: exercises/nt-44/question.tex

- Exponentiation (i.e., `__pow__`)

Note that $x^{-1000} \pmod N$ is $(x^{-1})^{1000} \pmod N$. You want to first check if you can use Euler's Theorem. If it is, use the theorem to lower the exponent to say ℓ. Then use the obvious loop to compute x^ℓ and apply $\pmod N$ as frequently as possible. After you are done with the above, you might want to improve your `__pow__` by *not* use Euler's theorem if the exponent is "small". You can determine for yourself what if "small". For instance you can choose to use Euler's Theorem only when the exponent is greater than 10. (Later in the RSA chapter, we will talk about the squaring method.) (Go to solution, page 131) □

Exercise 1.9.16. Leetcode 372.

debug: exercises/nt-45/question.tex

`https://leetcode.com/problems/super-pow/`
Your task is to calculate $a^b \pmod{1337}$ where a is a positive integer and b is an extremely large positive integer given in the form of an array. For instance for $a = 2, b = [1, 0]$, the output is 1024. (Go to solution, page 132) □

Exercise 1.9.17. Leetcode 1015. debug: exercises/nt-46/question.tex
https://leetcode.com/problems/smallest-integer-divisible-by-k/
Given a positive integer k, you need to find the length of the smallest positive integer n such that n is divisible by k, and n only contains the digit 1. Return the length of n. If there is no such n, return -1. Note: n may not fit in a 64-bit signed integer. (Go to solution, page 133) □

Exercise 1.9.18. Leetcode 1622. debug: exercises/nt-47/question.tex
https://leetcode.com/problems/fancy-sequence/
Write an API that generates fancy sequences using the append, addAll, and multAll operations. Implement the Fancy class:

- `Fancy()` Initializes the object with an empty sequence.
- `void append(val)` Appends an integer val to the end of the sequence.
- `void addAll(inc)` Increments all existing values in the sequence by an integer inc.
- `void multAll(m)` Multiplies all existing values in the sequence by an integer `m`.
- `int getIndex(idx)` Gets the current value at index `idx` (0-indexed) of the sequence modulo 109 + 7. If the index is greater or equal than the length of the sequence, return `-1`.

(Go to solution, page 134) □

Exercise 1.9.19. Leetcode 1952 debug: exercises/nt-48/question.tex
https://leetcode.com/problems/three-divisors/
Given an integer `n`, return `true` if `n` has exactly three positive divisors. Otherwise, return `false`. An integer `m` is a divisor of `n` if there exists an integer `k` such that `n = k * m`. (Go to solution, page 135) □

Exercise 1.9.20. https://acm.timus.ru/problem.aspx?space=1&num=1673 debug: exercises/nt-49/question.tex
At the end of the previous semester the students of the Department of Mathematics and Mechanics of the Yekaterinozavodsk State University had to take an exam in network technologies. N professors discussed the curriculum and decided that there would be exactly N^2 labs, the first professor would hold labs with numbers $1, N+1, 2N+1, ..., N^2-N+1$, the second one — labs with numbers $2, N+2, 2N+2, ..., N^2-N+2$, etc. N-th professor would hold labs with numbers $N, 2N, 3N, ..., N^2$. The professors remembered that during the last years lazy students didn't attend labs and as a result got bad

marks at the exam. So they decided that a student would be admitted to the exam only if he would attend at least one lab of each professor. N roommates didn't know the number of labs and professors in this semester. These students had different diligence: the first student attended all labs, the second one — only labs which numbers were a multiple of two, the third one — only labs which numbers were a multiple of three, etc… At the end of the semester it turned out that only K of these students were admitted to the exam. Find the minimal N which makes that possible.

Input: An integer K ($1 \leq K \leq 2 \cdot 10^9$).

Output: Output the minimal possible N which satisfies the problem statement. If there is no N for which exactly K students would be admitted to the exam, output 0.

Example: Input:8, output:15. Input:3, output:0.

(Go to solution, page 136) □

Solutions

Solution to Exercise 1.9.1.

debug: exercises/nt-30/answer.tex

Solution not provided.

Solution to Exercise 1.9.2.

debug: exercises/nt-31/answer.tex

Solution not provided.

Solution to Exercise 1.9.3.

debug: exercises/nt-32/answer.tex

Solution not provided.

Solution to Exercise 1.9.4.

debug: exercises/nt-33/answer.tex

Solution not provided.

Solution to Exercise 1.9.5.

debug: exercises/nt-34/answer.tex

Solution not provided.

Solution to Exercise 1.9.6.

debug: exercises/nt-35/answer.tex

Solution not provided.

Solution to Exercise 1.9.7.

debug: exercises/nt-36/answer.tex

Solution not provided.

Solution to Exercise 1.9.8.

debug: exercises/nt-37/answer.tex

Solution not provided.

Solution to Exercise 1.9.9.

debug: exercises/nt-38/answer.tex

Solution not provided.

Solution to Exercise 1.9.10.

debug: exercises/nt-39/answer.tex

Solution not provided.

Solution to Exercise 1.9.11.

debug: exercises/nt-40/answer.tex

Solution not provided.

Solution to Exercise 1.9.12.

debug: exercises/nt-41/answer.tex

Solution not provided.

Solution to Exercise 1.9.13.

debug: exercises/nt-42/answer.tex

Solution not provided.

Solution to Exercise 1.9.14.

debug: exercises/nt-43/answer.tex

Solution not provided.

Solution to Exercise 1.9.15.

debug: exercises/nt-44/answer.tex

Solution not provided.

Solution to Exercise 1.9.16.

debug: exercises/nt-45/answer.tex

Solution not provided.

Solution to Exercise 1.9.17.

debug: exercises/nt-46/answer.tex

Solution not provided.

Solution to Exercise 1.9.18.

debug: exercises/nt-47/answer.tex

Solution not provided.

Solution to Exercise 1.9.19.

debug: exercises/nt-48/answer.tex

Solution not provided.

Solution to Exercise 1.9.20.

debug: exercises/nt-49/answer.tex

Solution not provided.

1.10 Relations debug: relations.tex

relation

Let X, Y be two set. A **relation** R on X, Y is a subset of $X \times Y$. For instance for $X = \{0, 1, 2\}$ and $Y = \{a, b\}$,

$$R = \{(0, b), (2, a), (2, b)\}$$

is a relation on $X \times Y$. Instead of writing

$$(2, a) \in R$$

and say "$(2, a)$ is an R", I can also write

$$2Ra$$

and say ""2 is related to a (for relation R)". Note that $(0, a) \not\in R$. In this case, I will also say "0 is not related to a" (through R) and will write

$$0 \not R\, a$$

It's convenient to draw a relation as a directed graph. For instance for the above relation

$$R = \{(0, b), (2, a), (2, b)\}$$

I can draw

$$\begin{tikzcd}
0 \arrow[dr] & a \\
1 & b \\
2 \arrow[uur] \arrow[ur] &
\end{tikzcd}$$

You can represent R as a 2D array (matrix):

$$\begin{array}{c|c|c|}
 & a & b \\
\hline
0 & 0 & 1 \\
\hline
1 & 0 & 0 \\
\hline
2 & 1 & 1 \\
\hline
\end{array}$$

This version of the R is more suitable for computations with a program, especially after the column names a,b are changed to 0, 1. A 0 in the matrix at row r, column c means "not related" while a 1 means "is related".

If $X = Y$, then instead of saying "R is a relation on X and X", I will just say "R is a relation on X".

For instance $\equiv \pmod N$ is a relation on $\mathbb{Z}$. This is (of course) defined by

$$a \equiv b \pmod N \text{ if } N \mid a - b$$

Definition 1.10.1. Let R be a relation on set X.

(a) R is **reflexive** if for all $x \in X$, we have xRx. reflexive
(b) R is **symmetric** if for all $x, y \in X$, if xRy, then yRx. symmetric
(c) R is **transitive** if for all $x, y, z \in X$, if xRy and yRz, then xRz. transitive

R is said to be **transitive** if R is reflexive, symmetric and transitive. transitive

The following are examples of relations.

Example 1.10.1. For a general set X you have the "$=$" and "$\neq$" relation. A function $f : X \to Y$ can be viewed as a relation since f gives us the set

$$\{(x, f(x)) \mid x \in X\}$$

This set is usually called the **function graph** of f. function graph

Example 1.10.2. For a set of sets (for instance a powerset), besides " $=$ " and "$\neq$", you also have the following relations:

(a) $\subseteq$, $\subsetneq$
(b) "intersects (nontrivally)". X intersects Y means $X \cap Y \neq \emptyset$.
(c) "disjoint from" where X is disjoint from Y means $X \cap Y = \emptyset$.
(d) "same cardinality" where $|X| = |Y|$ means there is a bijection $X \to Y$.

Example 1.10.3. Besides "$=$" and "$\neq$", here are some relations on $\mathbb{R}$:

(a) the "$<$" and the "$\leq$" relation.
(b) the "$>$" and the "$\geq$" relation.

(c) “less than 1 apart” is a relation, i.e., for $x, y \in \mathbb{R}$, define xRy if $|x-y| < 1$. Of course changing “1” in the above to another fixed real number also gives you a relation.
(d) “Has the same fractional part as” is a relation. For instance 1.25 is related to 7.25 since they both have fractional part of 0.25.
(e) “Has the same integer part as” is also a relation.

Example 1.10.4. On $\mathbb{Z}$ here are some relations:

(a) Divisibility relation $a \mid b$ if $ax = b$ for some x.
(b) For a fixed $N > 0$, congruence relation mod $N > 0$, i.e., $a \equiv b \pmod{N}$ if $N \mid a - b$.
(c) “has same number of distinct prime factors” is a relation. For instance 100 has as many prime factors (i.e., 2) as 35.
(d) Fix a prime p. Then “has the same highest power of p factor” is a relation. For instance fixing $p = 5$, $75 = 3 \cdot 5^2$ is related to $3502 \cdot 5^2 \cdot 7$. Sometimes the highest power of p dividing n is denoted by $v_p(n)$.

Example 1.10.5. On the set of strings,

(a) “has the same length” is a relation. For instance `cat` has the same length as `pie`.
(b) For bitstrings, “same number of 1s” is a relation.
(c) For bitstrings, “has the same parity for the number of 1s” is a relation. For instance 10010101000 and 00001101111 are related since both have an even number of 1s.

Example 1.10.6. From basic geometry you have

(a) The “is parallel to” is a relation on the set of lines in $\mathbb{R}^2$.
(b) The “is similar to” is a relation on the set of triangles in $\mathbb{R}^2$.
(c) The “is congruent to” is a relation on the set of triangles in $\mathbb{R}^2$.

Example 1.10.7. From graph theory, you have the following examples:

(a) “Reachability” is relation on the vertex set of a graph: vRv' if there is a path from v to v'.
(b) “is isomorphic to” is a relation on graphs.

(c) "is subgraph of" is a relation on graphs.
(d) "is homeomorphic to" is a relation on graphs.

Definition 1.10.2. Let R be a relation on X. Let $a \in X$. The **left relation class** of a is

left relation class

$$aR = aR\bullet = \{x \in X \mid aRx\}$$

The **right relation class** of a is

right relation class

$$Ra = \bullet Ra = \{x \in X \mid xRa\}$$

Let X/R denote the set of left relation classes of X:

$$X/R = \{aR \mid a \in X\}$$

and $R\backslash X$ denote the set of right relation classes of X:

$$R\backslash X = \{Ra \mid a \in X\}$$

Exercise 1.10.1. Find a relation on X such that there is some $a \in X$ such that $aR \neq Ra$. (Go to solution, page 144) □

debug: exercises/nt-63/question.tex

Proposition 1.10.1. *Let R is an equivalence relation on X. Let $a, a' \in X$.*

(a) $aR = Ra$
(b) $a \in aR$
(c) $aR = a'R$ *iff* aRa'
(d) $aR \cap a'R = \emptyset$ *iff* $a \not R a'$

Proof. (a) We have

$$\begin{aligned} x \in aR &\implies aRx \\ &\implies xRa && \text{because } R \text{ is symmetric} \\ &\implies x \in Ra \end{aligned}$$

Hence $aR \subseteq Ra$. Also,

$$\begin{aligned} x \in Ra &\implies xRa \\ &\implies aRx && \text{because } R \text{ is symmetric} \\ &\implies x \in aR \end{aligned}$$

Hence $Ra \subseteq Ra$. Therefore $aR = Ra$.

(b) aRa since R is reflexive. Hence $a \in aR$ and $a \in Ra$.

(c) We have

$$\begin{aligned} aR = a'R &\implies a \in aR = a'R && \text{by (a)} \\ &\implies a \in a'R \\ &\implies a'Ra \end{aligned}$$

Now assume $a'Ra$. Since R is symmetric, we also have aRa'.

$$\begin{aligned} x \in Ra &\implies xRa \\ &\implies xRa' && \text{by } xRa, aRa' \text{ and transitivity} \\ &\implies x \in Ra' \end{aligned}$$

Hence $Ra \subseteq Ra'$. Also,

$$\begin{aligned} x \in Ra' &\implies xRa' \\ &\implies xRa && \text{by } xRa', a'Ra \text{ and transitivity} \\ &\implies x \in Ra \end{aligned}$$

Hence $Ra' \subseteq Ra$. Therefore $Ra = Ra'$. $\square$

From (b) above, a left equivalence class is a right equivalence class and vice versa. Therefore for equivalence relation, we only need the term **equivalence class** (no left or right). In this case aR and Ra is denoted by $[a]_R$ or simply $[a]$ if R is understood.

equivalence class

Proposition 1.10.2.

(a) *If R is an equivalence relation, then there is a partition of X by equivalence classes.*
(b) *If X has a partition $X = \dot{\bigcup}_{i \in I} X_i$, then X has an equivalence relation R such that X_i are equivalence classes, i.e., R defined by*

$$xRy \text{ if } x, y \text{ are in the same } X_i$$

Proposition 1.10.3. *Let $N > 0$. Then the relation $\equiv \pmod{N}$ on $\mathbb{Z}$ is an equivalance relation*

Let R be an equivalence relation on X. Define the function

$$f_R : X \to X/R$$
$$f(x) = [x]_R$$

Proposition 1.10.4.

(a) *If xRy, then $f_R(x) = f_R(y)$.*
(b) *f_R is a well-defined onto function.*

Let R, R' be two relations on X such that R' is **finer** in the sense that finer

$$xR'y \implies xRy$$

Here are simple examples of the concept of finer relation:

(a) "sibling" is finer than "have the same great-grandfather".
(b) "divisible by 100" is finer than "divisible by 5".
(c) "$\equiv \pmod{100}$" is finer than "$\equiv \pmod{5}$".

If R is an equivalence relation, we have equivalence classes (through R) for X. You can then define a relation R' on X/R as follows:

$$[x]_R R' [y]_R \text{ if } xR'y$$

Watch out: Note that R' is the given relation X and also R' denotes the new relation on X/R. Here are simple examples of the concept of finer relation:

(a) "sibling" is finer than "have the same great-grandfather". Let $[\text{John}]_{\text{sibling}} = \{\text{John, Sue, Tom}\}$, i.e., John, Sue, Tom are siblings. Also, let $[\text{Mary}]_{\text{sibling}} = \{\text{Bob, Mary}\}$, i.e., Bob, Mary are siblings. We can talk about whether [John] and [Mary] are related or not through the same great-grandfather relation. This is determined by whether John and Mary have the same great-grandfather. If John and Mary do have the same great-grandfather, then every $x \in [\text{John}]$ and $y \in [\text{Mary}]$ also have the same great-grandfather.
(b) "$\equiv \pmod{100}$" is finer than "$\equiv \pmod{5}$". Therefore for $x, y \in \mathbb{Z}$, I can define

$$[x]_5 \equiv [y]_5 \pmod{100} \text{ if } x \equiv y \pmod{100}$$

Proposition 1.10.5. *If R' is finer than R, then $R' \subseteq R$.*

If R is a relation on X, you can enlarge the relation by "taking closure". For instance the **reflexive closure** of R is the relation R, by you add (x, x) to R:

reflexive closure

$$R' = \{(x, x) \mid x \in X\} \cup R$$

Then R' is reflexive. It's the smallest subset of $X \times X$ that is reflexive and contains R.

The **symmetric closure** R' of R is the smallest relation of $X \times X$ that contains R and is symmetric:

symmetric closure

$$R' = R \cup \{(y, x) \mid (x, y) \in R \text{ for } x, y \in X\}$$

The **transitive closure** R' of R is the smallest relation of $X \times X$ that contains R and is transitive. Note that whereas reflexive closure and symmetric closures are easy to understand, transitive closure is a little bit more complicated. For instance the following R' is *not* the transitive closure of R:

transitive closure

$$R' = R \cup \{(x, z) \mid (x, y), (y, z) \in R \text{ for } x, y, z \in X\}$$

Exercise 1.10.2. Find the simplest R such that the above R' construction is not transitive.

debug: exercises/nt-64/question.tex

(Go to solution, page 145) □

The concept of "closure" is very common in math and CS.

Back to the transitive closure, if R is a relation between X, Y and R' is a relation between Y, Z, then we can define the **composition** of $R \circ R'$:

composition

$$R \circ R' = \{(x, z) \in X \times Z \mid (x, y) \in R, (y, z) \in R' \text{ for some } y \in Y\}$$

I'll write R^2 for $R \circ R$. In general $R^1 = R$, R^{n+1} will denote $R^n \circ R$.

Proposition 1.10.6. *R is transitive if $R^2 \subseteq R$.*

Proposition 1.10.7. *R is transitive if $\bigcup_{n=1}^{\infty} R^n$.*

Solutions

Solution to Exercise 1.10.1.

debug: exercises/nt-63/answer.tex

Solution not provided.

Solution to Exercise 1.10.2.

debug: exercises/nt-64/answer.tex

Solution not provided.

1.11 Congruence classes debug: congruence-classes.tex

I want to view modulo arithmetic in a different way. This section requires knowledge of equivalence relations and equivalence classes.

Recall that for $N > 1$, we think of $\mathbb{Z}/N$ as the set

$$\mathbb{Z}/N = \{0, 1, 2, ..., N-1\}$$

where we can add, multiply, etc. There are other symbols in this system. For instance I can write N. But in this system,

$$N \equiv 0 \pmod{N}$$

In other words, informally, in this system N is just another notation for 0. That's the point of the congrence notation $\equiv \pmod{N}$.

Recall the concept of relations on a set X. If a relation R is an equivalence relation on X, then X can be partitioned into subsets.

The $\equiv \pmod{N}$ relation on $\mathbb{Z}$ is an equivalence relation. For instance $\equiv \pmod{6}$ is an equivalence relation on $\mathbb{Z}$. Therefore this partitions $\mathbb{Z}$ into disjoint subsets. In fact the disjoint subsets are

$$\begin{aligned}
&\{-12, -6, 0, 6, 12, ...\} \\
&\{-11, -5, 1, 7, 13, ...\} \\
&\{-10, -4, 2, 8, 14, ...\} \\
&\{-9, -3, 3, 9, 15, ...\} \\
&\{-8, -2, 4, 10, 16, ...\} \\
&\{-7, -1, 5, 11, 17...\}
\end{aligned}$$

I'll give these subsets of $\mathbb{Z}$ names:

$$\begin{aligned}
[0]_6 &= \{-12, -6, 0, 6, 12, ...\} \\
[1]_6 &= \{-11, -5, 1, 7, 13, ...\} \\
[2]_6 &= \{-10, -4, 2, 8, 14, ...\} \\
[3]_6 &= \{-9, -3, 3, 9, 15, ...\} \\
[4]_6 &= \{-8, -2, 4, 10, 16, ...\} \\
[5]_6 &= \{-7, -1, 5, 11, 17...\}
\end{aligned}$$

These sets are called **congruences classes** of $\equiv \pmod{6}$. Note that when I

congruences classes

write

$$[0]_6$$

I mean the congruence class that contains 0. I could have chosen 6 and write

$$[6]_6$$

However

$$[0]_6 = [6]_6$$

Right?

You can think of $\mathbb{Z}/6$ as the set of these congruence classes:

$$\mathbb{Z}/6 = \{[0]_6, [1]_6, [2]_6, [3]_6, [4]_6, [5]_6\}$$

When write

$$2 + 4 \equiv 0 \pmod 6$$

we can rephrase that as

$$[2]_6 + [4]_6 = [0]_6$$

In other words we define $+_6$ and $\cdot_6$ on these congruence classes as

$$\begin{aligned} [a]_6 +_6 [b]_6 &= [a+b]_6 \\ [a]_6 \cdot_6 [b]_6 &= [a \cdot b]_6 \end{aligned}$$

The $+_6$ and $\cdot_6$ on the left are operations on the congruence classes of $\mathbb{Z}/6$. The $+$ and $\cdot$ on the right are the usual operations in $\mathbb{Z}$. I will usually write $+$ for $+_6$ and $\cdot$ for $\cdot_6$.

With these two operations, the set $Z6$ becomes

$$(\mathbb{Z}/6, +_6, \cdot_6, [0]_6, [1]_6)$$

which satisfies the same set of axioms of $(\mathbb{Z}, +, \cdot, 0, 1)$: CAIN for addition, CAN for multiplication, and distribution. But not the integrality, topology axioms. In other words $(\mathbb{Z}, +, \cdot, 0, 1)$ and $(\mathbb{Z}/6, +_6, \cdot_6, [0]_6, [1]_6)$ are both ring. In particular, since multiplication for both rings are commutative, both are called commutative rings. Every formula of the form

$$x \equiv y \pmod 6$$

can be re-expressed as

$$[x]_6 = [y]_6$$

The above is true when 6 is replaced by any $N > 0$.

Note that whereas $\mathbb{Z}$ is an infinite ring, $\mathbb{Z}/N$ is a finite ring of size N. Furthermore, there is a map

$$\mathbb{Z} \rightarrow \mathbb{Z}/N$$

given by

$$x \mapsto [x]_N$$

A ring is a **field** if every nonzero element has a multiplicative inverse. Therefore if $p > 1$, then $\mathbb{Z}/p$ is a field iff p is prime. field

Index

Bibliography