Sets

We discuss an informal (naive) set theory as needed in Computer Science.

It was introduced by G. Cantor in the second half of the nineteenth century.

Most students have seen sets before. This is intended to give:

- a review of basic notation,
- an introduction to some less common set operations,
- the relationship between set theory and logic.

Set ... a collection of objects.
Objects in the collection are called **members** of the set.

To specify a set, you can:

- list all elements of the set between $\{\ \}$.
  Example: a set of binary digits: $\{0, 1\}$

- list all elements of the set between $\{\ \}$,indicate the continuation of a pattern by ...
  Example: $\{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$ is the set of all integers.

- state the property all elements in the set satisfy.: **set builder** notation $\{x \mid P(x)\}$.
  Example: $\{x \mid x \text{ is a student in this class}\}$

Definitions

$x \in A \iff x$ is a member of A.

$x \notin A \iff \neg(x \in A)$

$A \subseteq B \iff [\forall x \ (x \in A) \rightarrow (x \in B)]$

$A = B \iff (A \subseteq B) \wedge (B \subseteq A)$

$A \subset B \iff (A \subseteq B) \wedge \neg(A = B)$

$[\exists x \ x \in \emptyset] \iff F$

$[\forall x \ x \in U] \iff T$

Cardinality

A set with exactly k distinct elements for some natural number k is called a **finite** set and k is its **cardinality** (or **cardinal number**). We say $|A| = k$.

If A is not finite then A is **infinite**.

<u>Example 1:</u> Let $A = \{i \in Z \mid 1 \leq i \leq 26\}$.
Then $|A| = 26$.

<u>Example 2:</u> Let $B = \{i \in Z \mid i \text{ is a prime number }\}$.
Then B is infinite.

$P(A)$, the **power set** of A is the set of all subsets of A.

$P(\{1,2,3\}) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$

The **cartesian** product of sets A and B is the set of **ordered pairs** of A and B,
$A \times B = \{(a,b) \mid a \in A \text{ and } b \in B\}$

If $A = \{1,2\}$, $B = \{x,y,z\}$ then

$A \times B = \{(1,x),(1,y),(1,z),(2,x),(2,y),(2,z)\}$

$(1,x) \in A \times B$ $\qquad$ $(x,1) \notin A \times B$

Set operations

Definitions: Let A and B be sets.

The **union** of A and B is the set consisting of all elements in A and all elements in B.

$$A \cup B = \{x \mid (x \in A) \vee (x \in B)\}$$

The **intersection** of A and B is the set consisting of all elements in both A and B.

$$A \cap B = \{x \mid (x \in A) \wedge (x \in B)\}$$

The **difference** of A and B is the set consisting of all elements in A but not in B.

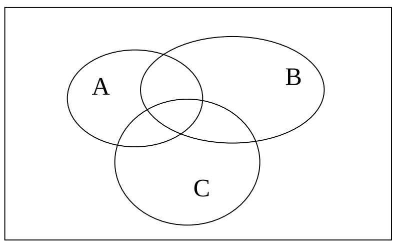
$$A - B = \{x \mid (x \in A) \wedge (x \notin B)\}$$

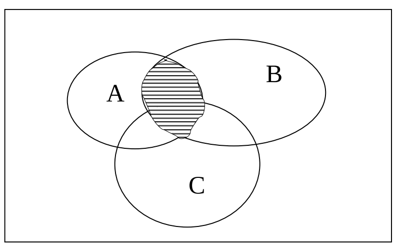
Let U be the universal set. The **complement** of A is the set consisting of all elements in U but not in A

$$\overline{A} = U - A$$

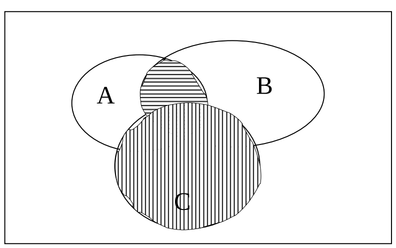
Venn diagram

A graphical representation of sets, that helps us to <u>visualize</u> the results of set operations.





A B



(A B) C

Partitions

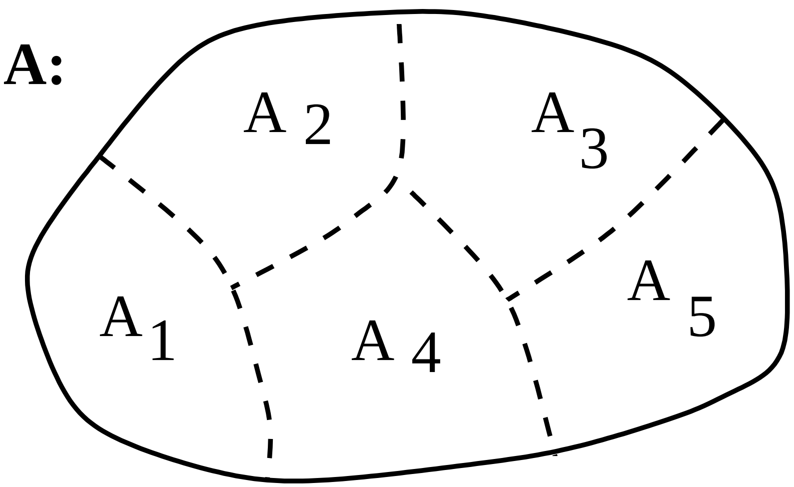
Sets A, B are called **disjoint** if $A \cap B = \emptyset$.
Sets $A_1, A_2, \ldots A_n$ are called **mutually disjoint** or **pairwise disjoint** if for all $i, j \in \{1, 2, \ldots n\}$, $A_i \cap A_j = \emptyset$ whenever $i \neq j$.

A collection of non-empty sets $\{A_1, A_2, \ldots A_n\}$ is a **partition** of a set A if:

1. $A = A_1 \cup A_2 \cup \ldots \cup A_n$

2. $A_1, A_2, \ldots, A_n$ are mutually disjoint.

A:



Properties of subsets

1. Inclusion of intersection.
   - $A \cap B \subseteq A$
   - $A \cap B \subseteq B$
2. Inclusion in union.
   - $A \subseteq A \cup B$
   - $B \subseteq A \cup B$
3. Transitive property of subsets
   $(A \subseteq B) \wedge (B \subseteq C) \Rightarrow (A \subseteq C)$

Set identity	Name
$A \cup \varnothing = A$ $A \cap \mathsf{U} = A$	identity
$A \cup \mathsf{U} = \mathsf{U}$ $A \cap \varnothing = \varnothing$	domination
$A \cup A = A$ $A \cap A = A$	idempotent
$\overline{(\overline{A})} = A$	complement
$A \cup B = B \cup A$ $A \cap B = B \cap A$	commutative
$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$	associative
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	distributive
$\overline{(A \cap B)} = \overline{A} \cup \overline{B}$ $\overline{(A \cup B)} = \overline{A} \cap \overline{B}$	de Morgan

Other useful identities

$$A \cup \overline{A} = U$$

$$A \cap \overline{A} = \emptyset$$

$$A - B = A \cap \overline{B}$$

Showing set identities

- Using set builder notation and logical equivalences (element proof).
- Using set identities (algebraic proof).
- Using membership tables.

An element proof

Example: Show that $A \cup (B - A) = A \cup B$.

$$A \cup (B - A) = \{x \mid (x \in A) \vee (x \in B - A)\}$$
$$= \{x \mid (x \in A) \vee ((x \in B) \wedge (x \notin A))\}$$
$$= \{x \mid ((x \in A) \vee (x \in B)) \wedge ((x \in A) \vee (x \notin A))\}$$
$$= \{x \mid ((x \in A) \vee (x \in B)) \wedge T\}$$
$$= \{x \mid (x \in A) \vee (x \in B)\}$$
$$= \{x \mid x \in A \cup B\}$$
$$= A \cup B$$

An algebraic proof

$A \cup (B - A)$

$= A \cup (B \cap \overline{A})$ shown earlier

$= (A \cup B) \cap (A \cup \overline{A})$ distributive law

$= (A \cup B) \cap U$ shown earlier

$= A \cup B$ identity laws

Computer Representation of Sets

Let $U = \{a_1, a_2, a_3, \ldots, a_k\}$ be a finite set with k distinct elements.

We fix an order of elements for the computer representation.

a_i .... ith element of the set

Represent any subset $T \subseteq U$ by a binary string of length k.

bit at position $i = 1$ iff $a_i \in T$,

bit at position $i = 0$ iff $a_i \notin T$.

$0010010...010 = \{a_3, a_6, a_{k-1}\}$

1111...11 represents U,

0000...00 represents $\varnothing$.

Example:
$U = \{a, e, i, o, u\}$ ... $k = |U| = 5$

11111 ... $\{a, e, i, o, u\}$
01101 ... $\{e, i, u\}$
10010 ... $\{a, o\}$
00000 ... $\varnothing$

We store in the computer:

- the list of elements in U (to convert between a binary string representation and the usual representation).
- a binary string of length k for each set.

Each set operation corresponds to a **logical operation** on the corresponding bit strings representing the sets.

$\cap$ .... bitwise and ... $\wedge$
$\cup$ .... bitwise or .... $\vee$
$\overline{X}$ .... bitwise negation ... $\neg$

Logical operations on computer words are basic operations in any computer, thus they are efficient.

Everybody should know:

the definition of a set;

the meaning of $\in$, $=$, $\subseteq$, $\subset$, $\varnothing$, U, cardinality.

Set operations:
the power set, $\times$, $\cap$, $\cup$, complement,
set difference.

Basic set equivalences.

How to prove set properties using
set equivalences and
using a translation into logic.

Functions

Motivation: Functions are basic components of program design.

Definition: For sets A and B, a **function** f from A to B is an assignment of exactly one element of B to each element of A.

$$f : A \rightarrow B$$

$f(x)$ denotes the element assigned by f to x.

The symbol $\rightarrow$ is being used here to express *from* A *to* B and is not a conditional operator.

(This is an example of operator overloading, which is the use of a symbol for several purposes. The correct meaning can be deduced from the context.)

Three items are needed to specify a function:
domain, codomain, and action.

Example:

function f

Domain $A = \{\text{Montreal}, \text{Toronto}, \text{Ottawa}, \text{Boston}, \text{Buffalo}\}$

Co-domain $B = \{1, 2, 3, 4, 5\}$

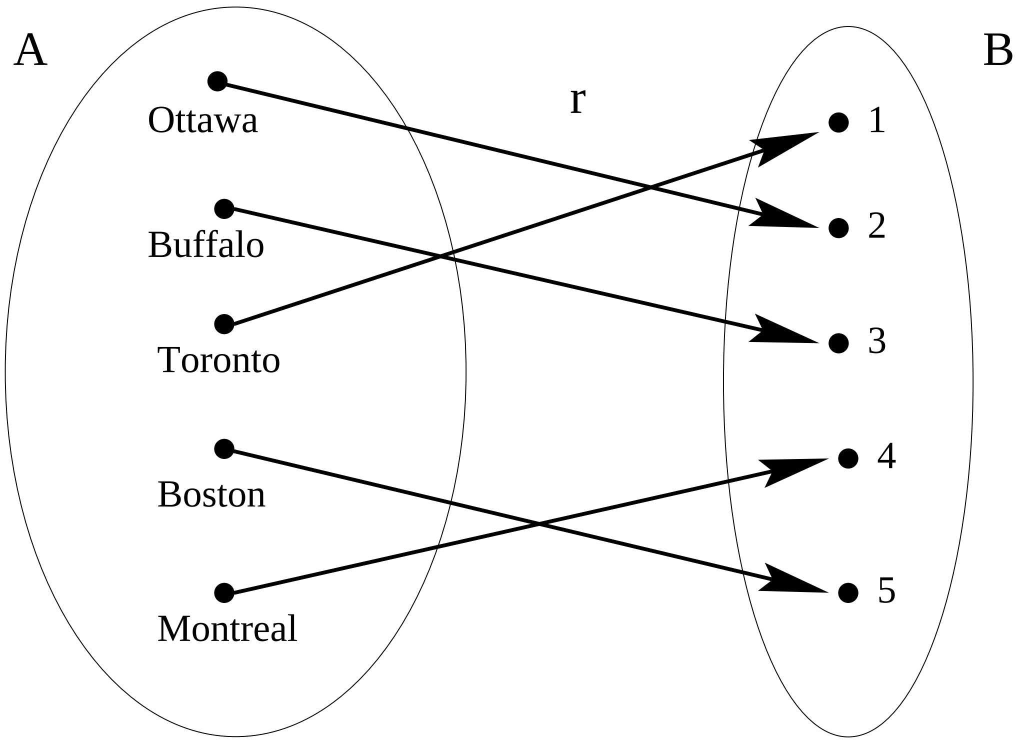
$$f : A \to B$$

f is the following assignment:

Boston . . . 5
Buffalo . . . 3
Montreal . . . 4
Ottawa . . . 2
Toronto . . . 1

We write $f(\text{Montreal}) = 4$, $f(\text{Boston}) = 5$, etc.

The arrow diagram of a function



Definitions

$f : A \to B$

A is the **domain** of f,
B is the **codomain** of f.

If $f(a) = b$
b is the **image** of a under f, or the **value** of f at a.
a is the **preimage** of b under f.
The **range** of f is $\{f(a) \mid a \in A\}$.

Example:

$f(\text{Toronto}) = 1$
$f(\text{Ottawa}) = 2$
$f(\text{Buffalo}) = 3$
$f(\text{Montreal}) = 4$
$f(\text{Boston}) = 5$

Therefore, by function f:

4 is the image of Montreal,
1 is the image of Toronto, etc.

Montreal is the preimage of 4,
Toronto is the preimage of 1, etc.

To specify a function $f : A \to B$:

- for each element in A, specify an element in B. (This is possible when the domain is finite.)
  Example: the function f just studied

- give an expression that specifies the assignment for all values in the domain.
  Example 1:

  $f :\ N \to N$
  $$f(n) = 2n + 1$$

  Example 2:
  $g :\ N \to N$
  $$g(n) = \begin{cases} 2n + 1 \text{ if } n \text{ is odd,} \\ n/2 \text{ if } n \text{ is even.} \end{cases}$$

Let f_1, f_2 be functions whose codomain is R.

$$(f_1 + f_2)(x) = f_1(x) + f_2(x)$$
$$(f_1\, f_2)(x) = f_1(x) f_2(x)$$

<u>Example</u>:

$f : N \to N$ $\qquad f(n) = 2n + 1$

$g : N \to N$ $\qquad g(n) = 3n + 2$

$(f + g)(3) = f(3) + g(3) = 7 + 11 = 18$

$(f\, g)(3) = f(3) \cdot g(3) = 7 \cdot 11 = 77$

Let $f : A \to B$ and S be a subset of A.

$f(S) = \{f(s) \mid s \in S\}$

<u>Example</u>: $\qquad g(\{1, 3, 4, 11\}) = \{5, 11, 14, 35\}$

Definition: A function $f : A \to B$ is **one-to-one**, or **injective** if for any two distinct elements $x, y \in A$ we have $f(x) \neq f(y)$.

f is one-to one $\iff \forall x \, \forall y \, [x \neq y \to f(x) \neq f(y)]$
where A is the universe of discourse for x, y.

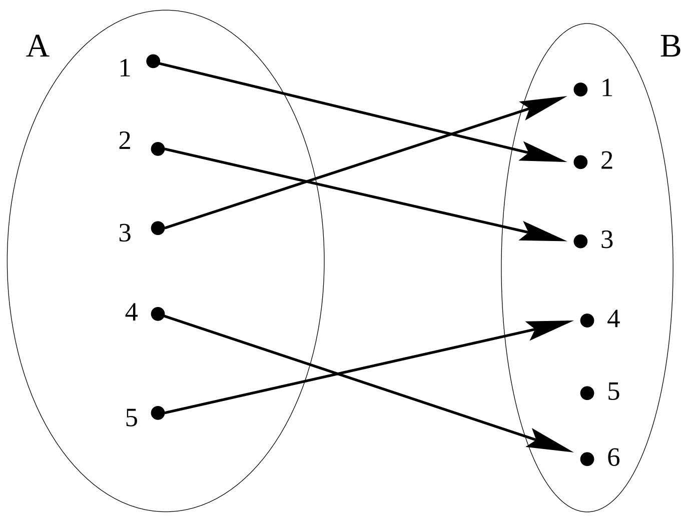
Example 1: $f_1 : N \to N$ $\qquad f_1(n) = 2n + 1$

If $x \neq y$ then $2x + 1 \neq 2y + 1$. Thus, f_1 is one-to-one.

Example 2: $f_2 : Z \to N$ $\qquad f_2(x) = x^2$

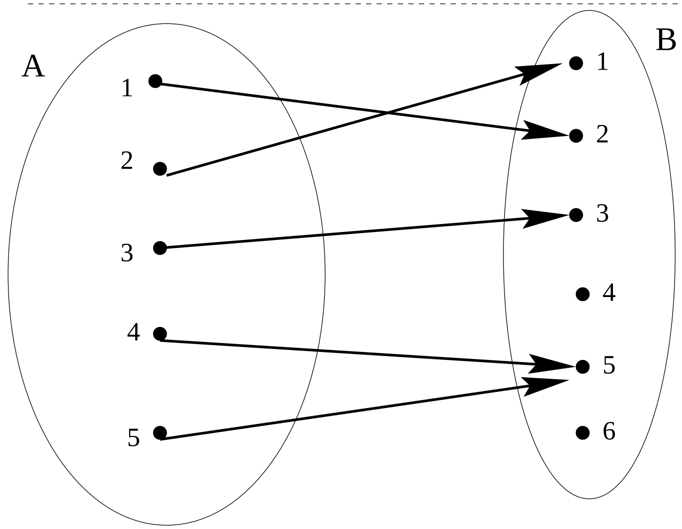
$f_2(-1) = 1 = f_2(1)$. Thus, f_2 is not one-to-one.

Example 3: For a function $R \to R$, if f is *strictly increasing* or *strictly decreasing* then f is one-to-one.



one-to-one

function



not a
one-to-one
function

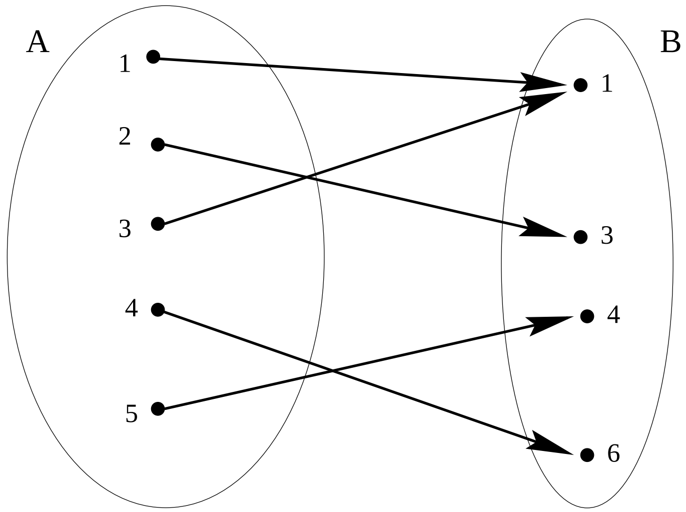
Definition: A function $f : A \to B$ is **onto**, or **surjective** if for any element $y \in B$ there exists an element $x \in A$ so that $f(x) = y$.

f is onto $\iff \forall y \, \exists x \, [f(x) = y]$ where the u. of discourse for y and x are B and A respectively.

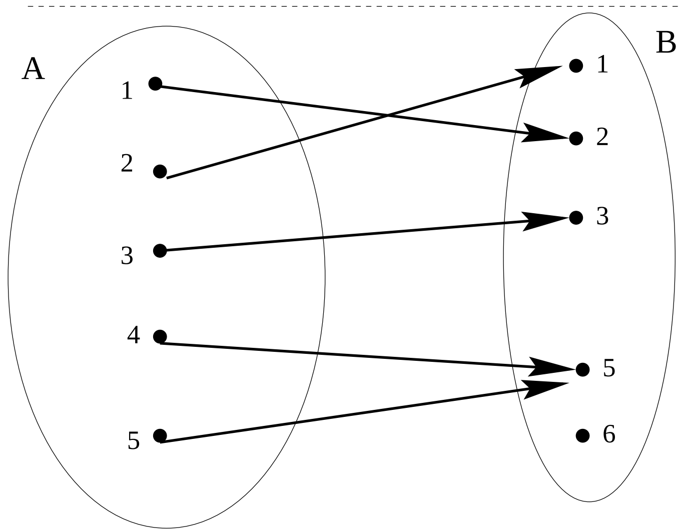
Example 1: $f_1 : N \to N$ $\qquad f_1(n) = 2n + 1$
If y is even then no element of N is mapped to y.
Thus, f_1 is not onto.

Example 2: $f_2 : N \to N$ $\qquad f_2(x) = x^2$ For $y = 2$, there is no integer x such that $x^2 = 2$.
Thus, f_2 is not onto.

Example 3: $f_3 : R \to R$ $\qquad f_3(n) = 2n + 1$
For any y, the function f_3 maps $\frac{y-1}{2}$ to y.
Thus, f_3 is onto.

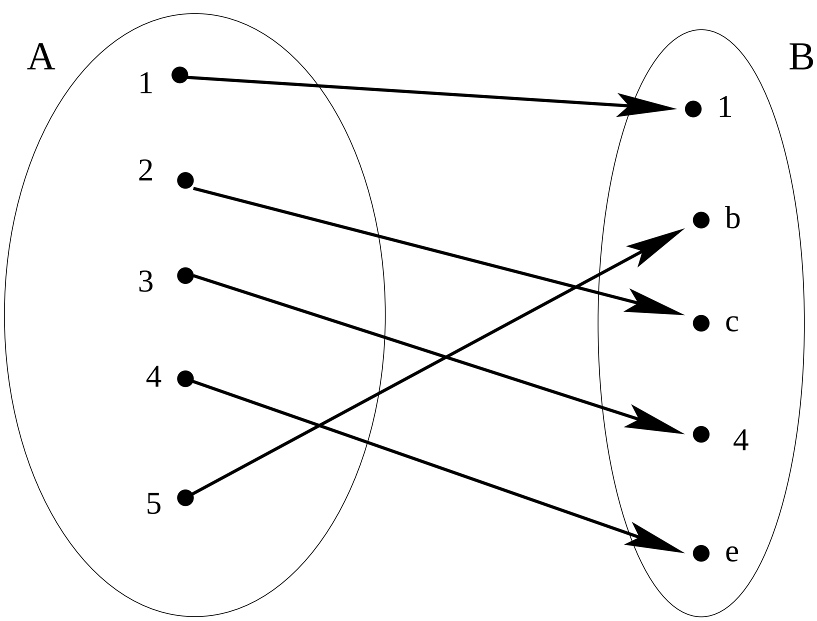


onto function



not an onto function

Definition: A function f is a **one-to-one correspondence** or a **bijection** if it is both one-to-one and onto.



one−to−one and onto
therefore, a one−to−one correspondence

Examples:

$f_3 : R \to R \qquad\qquad f_3(n) = 2n + 1$

f_3 is a one-to-one correspondence.

The ASCII mapping of computer characters to
the set $\{0, 1, 2, \ldots 255\}$
is a one-to-one correspondence.

The **identity function** i_A on a set A:

$$\forall x \in A \ [i_A(x) = x]$$

i_A is a bijection.

Graph of a function

A function $f : A \to B$ assigns to each element of A an element of B.

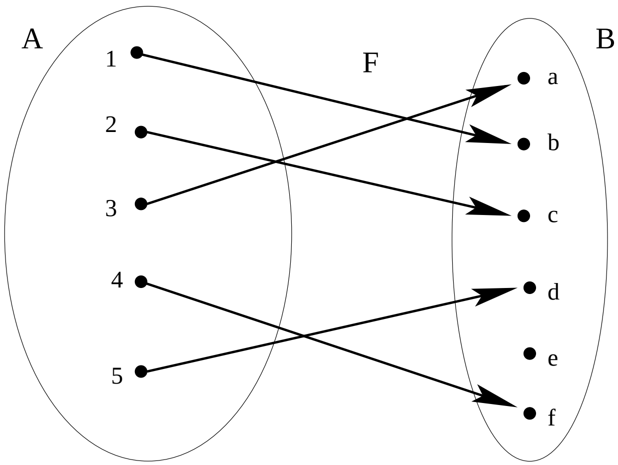
f may be interpreted as a set of pairs $\{(a, f(a)) \mid a \in A\}$

So f may be interpreted as a subset of $A \times B$.

Definition: The **graph of** f is
a display of pairs in $\{(a, f(a)) \mid a \in A\}$ in a plane representation of $A \times B$:

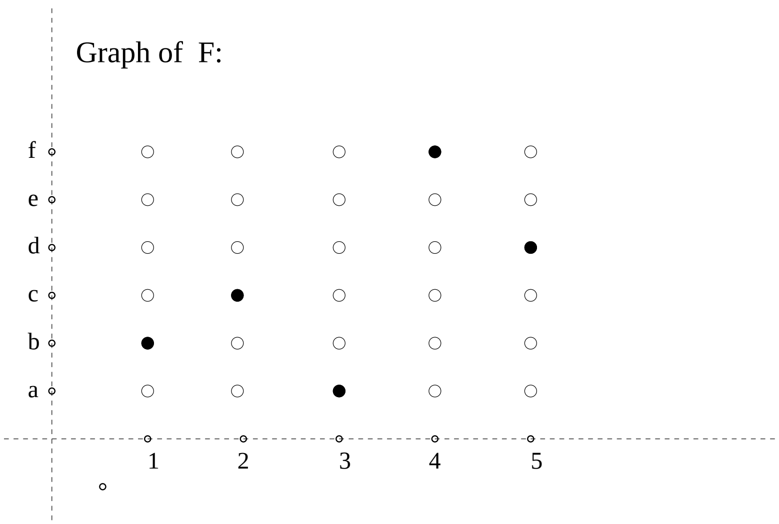
Put all points of A on a horizontal line,
put all points of B on a vertical line.

Elements of $A \times B$ correspond to points in the plane. We display just those points which belong to f.



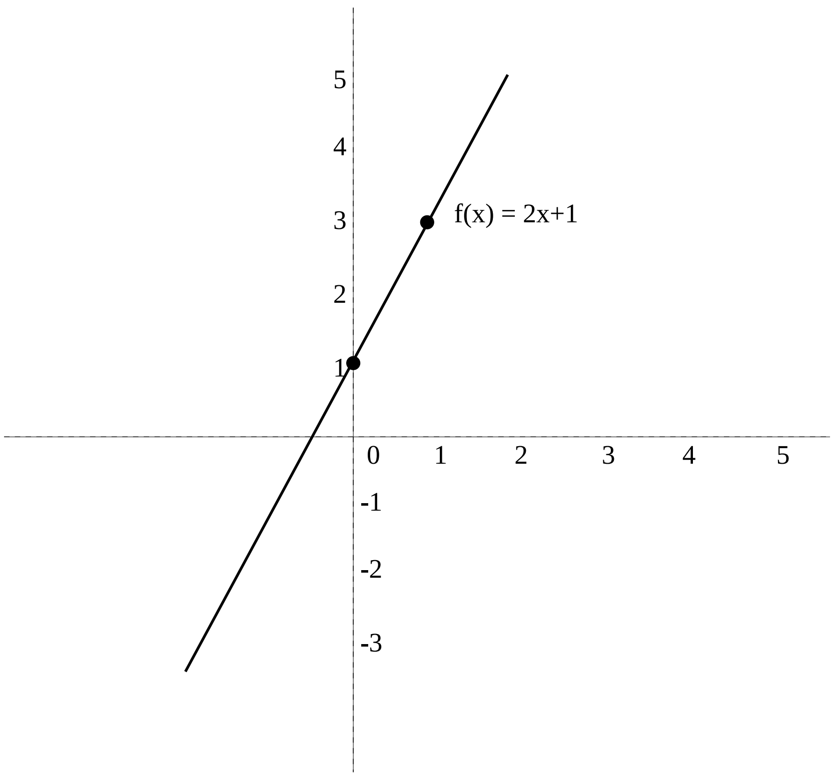
Function F contains pairs {(1,b),(2,c), (3,a),(4,f),(5,d)}

Graph of F:



Functions used in calculus are from R to R.

$f:\ R \rightarrow R,\ f(x) = 2x + 1$



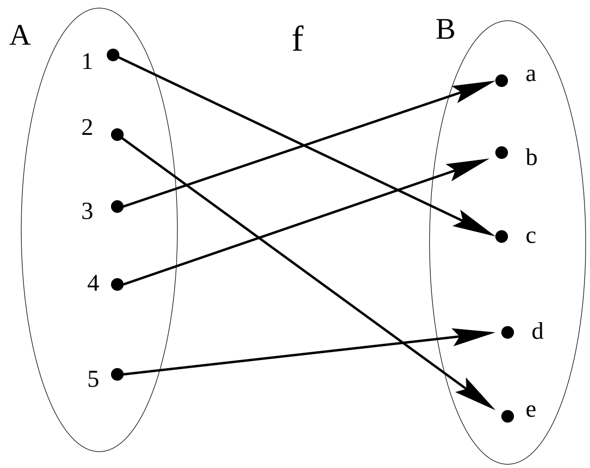
Inverse functions

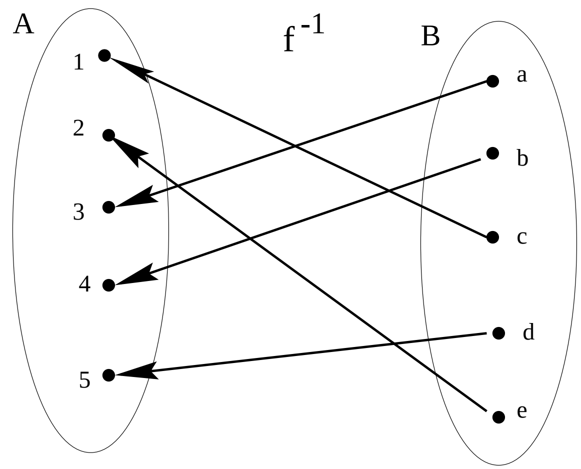
Definition: Let f be a one-to-one correspondence from A to B. The **inverse function** of f is the function that assigns to each element $b \in B$ the element $a \in A$ such that $f(a) = b$.

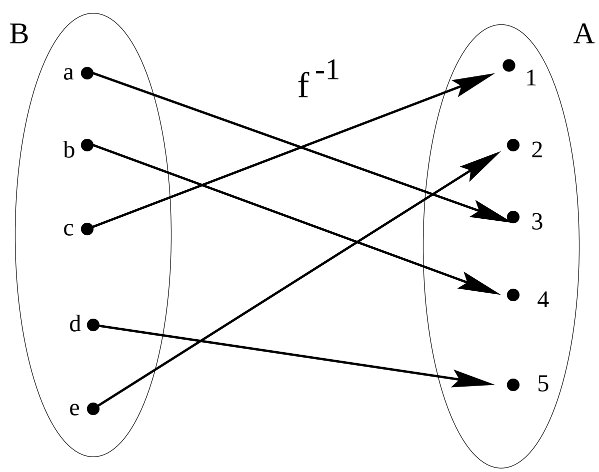
The inverse of f is denoted f^{-1}

$$f^{-1}(b) = a \iff f(a) = b$$

A function f is called **invertible** iff it is one-to-one and onto.


A
f
B
1
2
3
4
5
a
b
c
d
e


A
f -1
B
1
2
3
4
5
a
b
c
d
e


B
f -1
A
a
b
c
d
e
1
2
3
4
5

Example 1:

$f_1 : R \to R$ $\qquad f_1(n) = 2n + 1$

f_1 is a one-to-one correspondence.

$f_1^{-1}(y) = \frac{y-1}{2}$

Example 2:

$f_2 : N \to N$ $\qquad f_2(x) = x^2$

f_2 is not onto and therefore f_2 is not invertible.
Note this property is domain dependent.
For a function from R to R, invertibility can often be seen from its graph.

Example 3: Define $f_3 : Z \times Z \to Z \times Z$ by
$f_3(m, n) = (m + n, m - n)$.
Show that f_3 is one-to-one but not onto.

Proof. (i) First we show that f_3 is one-to-one.
Suppose $f_3(m_1, n_1) = f_3(m_2, n_2)$.
Then $(m_1 + n_1, m_1 - n_1) = (m_2 + n_2, m_2 - n_2)$.
That is, $m_1 + n_1 = m_2 + n_2$
and $m_1 - n_1 = m_2 - n_2$.

Add the equations to find that $m_1 = m_2$ and subtract one from the other to find that $n_1 = n_2$.

Thus, $(m_1, n_1) = (m_2, n_2)$ and f_3 is one-to-one.

(ii) Next we show that f_3 is not onto.
Take arbitrary $(a, b) \in Z \times Z$. Does it have a pre-image?

Let $f_3(m, n) = (a, b)$.
Then $m + n = a$ and $m - n = b$.

Solving the simultaneous equations, we get:
$m = (a + b)/2$ and $n = (a - b)/2$.

But if $a = 1$ and $b = 2$, then m and n are not integers. Thus, f_3 is not onto.

If f_3 is same as above, but defined as:
$f_3 : R \times R \rightarrow R \times R$
then it is a bijection.

Problem: Give an example of a bijection between Z and N.

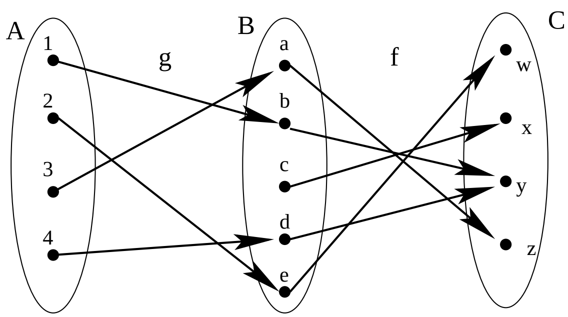
0	-1	1	-2	2	-3	3	-4	4
0	1	2	3	4	5	6	7	8

$$f(n) = \begin{cases} 2n & \text{if } n \geq 0 \\ -(2n+1) & \text{if } n < 0 \end{cases}$$

Exercise: What is f^{-1}?

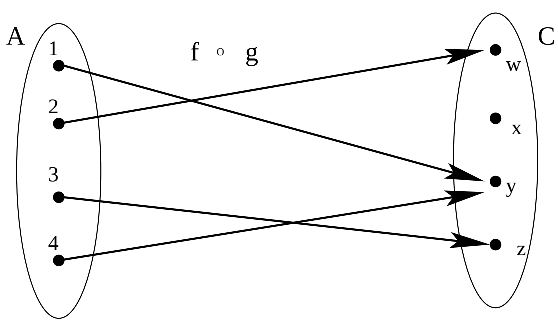
Composition of functions

$g : A \rightarrow B$ $\qquad$ $f : B \rightarrow C$



g(2) = e $\qquad$ f(e) =w

Thus, $\quad$ f(g(2)) = f(e) = w



Composition of functions

<u>Definition:</u> Let $g : A \to B$ and $f : B \to C$.

The **composition** of the functions f and g, denoted

$$f \circ g$$

is defined by

$$(f \circ g)(a) = f(g(a))$$
$$f \circ g : A \to C$$

<u>Important:</u> To get $f \circ g$ we need

codomain of g = domain of f

Generally, $f \circ g \neq g \circ f$
and sometimes one or both may not exist at all.

Example 1:

g : set of items $\rightarrow N$
It assigns to each item its bar code.

$f : N \rightarrow R$
It assigns to each bar code a price.

$f \circ g$
A function that assigns to each item a price.

$g \circ f$ does not exist, since

codomain of $f \neq$ domain of g

Example 2: $g : Z \to Z$
$g(x) = 2x + 3$

$f : Z \to Z$
$f(x) = (x+1)^2$

---

codomain of g = domain of $f \implies f \circ g$ exists.
$(f \circ g)(x) = f(g(x)) = f(2x+3) =$
$(2x+3+1)^2 = 4x^2 + 16x + 16$

---

codomain of f = domain of $g \implies g \circ f$ exists.
$(g \circ f)(x) = g(f(x)) = g((x+1)^2) =$
$g(x^2 + 2x + 1) = 2(x^2 + 2x + 1) + 3 = 2x^2 + 4x + 5$

Inverse and composition

Let $f : A \rightarrow B$ be an invertible function. Then $f^{-1} : B \rightarrow A$.

$(f^{-1} \circ f) : A \rightarrow A$ and $(f \circ f^{-1}) : B \rightarrow B$

$f(a) = b \iff f^{-1}(b) = a$

$(f^{-1} \circ f)(a) = f^{-1}(f(a)) = f^{-1}(b) = a$
$(f \circ f^{-1})(b) = f(f^{-1}(b)) = f(a) = b$

So, $(f^{-1} \circ f) = i_A$, the identity function on A,
$(f \circ f^{-1}) = i_B$, the identity function on B.

If f and $f \circ g$ are one-to-one,
does it follow that g is one-to-one?

If g and $f \circ g$ are one-to-one,
does it follow that f is one-to-one?

If g and $f \circ g$ are onto,
does it follow that f is onto?

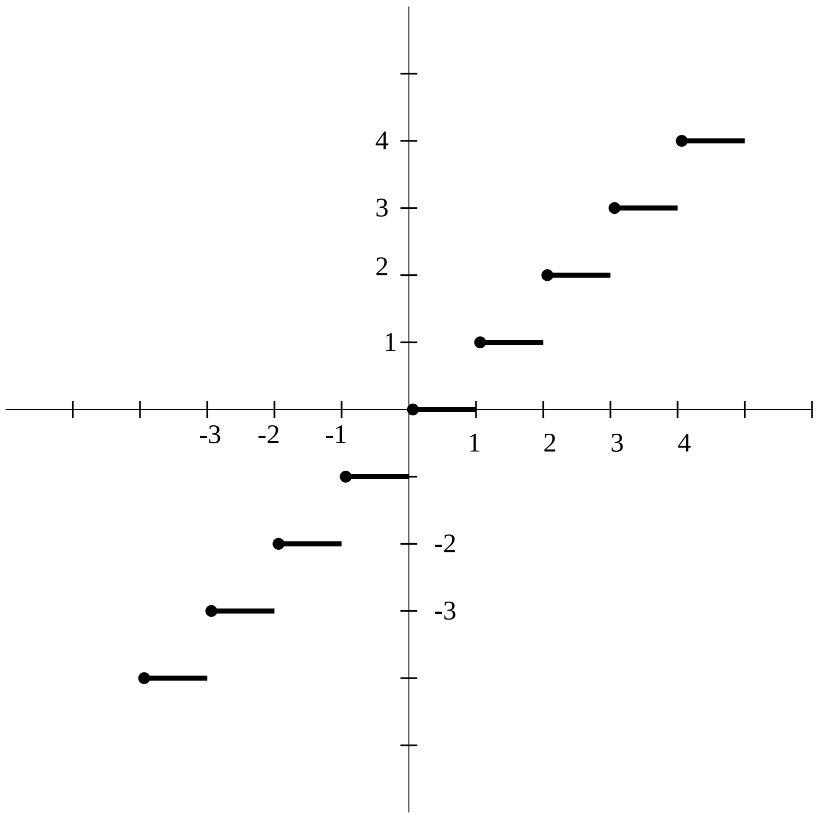
If f and $f \circ g$ are onto,
does it follow that g is onto?

Floor function

The **floor function** $\lfloor x \rfloor$ is a function from R to Z

Its value is the largest integer $\leq x$

$\lfloor 3.6 \rfloor = 3$ $\lfloor 12.1 \rfloor = 12$
$\lfloor 15 \rfloor = 15$ $\lfloor -3.4 \rfloor = -4$

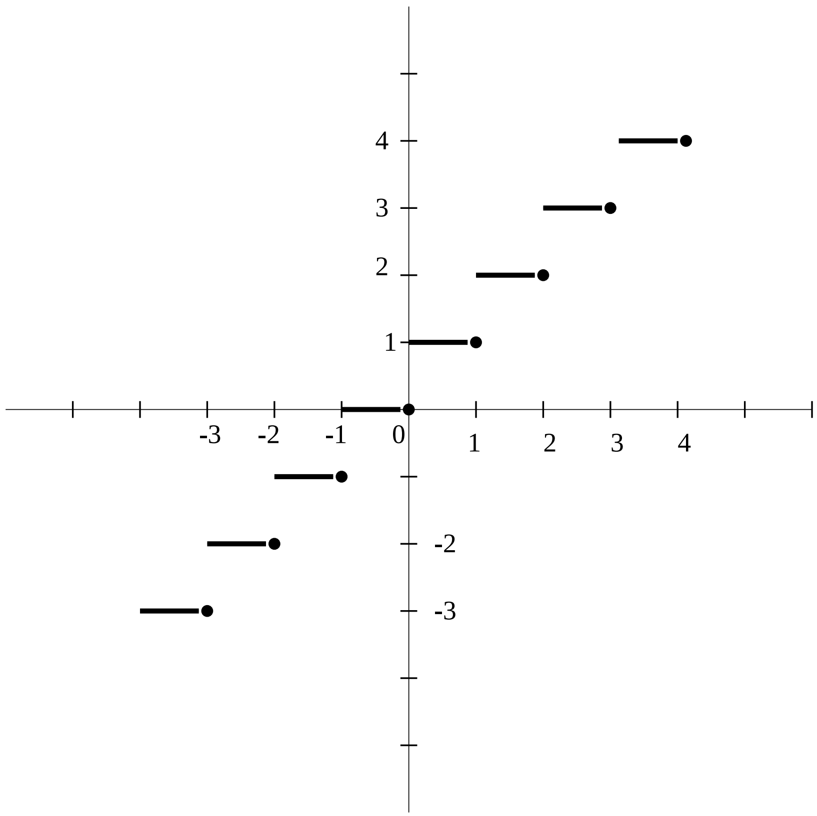

4
3
2
1
-3
-2
-1
1
2
3
4
-2
-3

Ceiling function

The **ceiling function** $\lceil x \rceil$ is
a function from R to Z

Its value is the smallest integer $\geq x$

$\lceil 3.6 \rceil = 4$	$\lceil 12.1 \rceil = 13$
$\lceil 15 \rceil = 15$	$\lceil -3.4 \rfloor = -3$


4
3
2
1
-3
-2
-1
0
1
2
3
4
-2
-3

Problem 1: Assume that in each word of computer memory we can store k bytes. Find a function $w : N \rightarrow N$ that specifies the number of words needed to store n bytes.

Solution: $w(n) = \lceil \frac{n}{k} \rceil$

Problem 2: A bank must round the calculations involving money to cents.
i.e., 5.33453 is rounded to 5.33 and
5.13618 is rounded to 5.14
Give a function $round : R \rightarrow R$ that rounds any real number to 2 decimal points.

Solution: $round(x) = (\lfloor (x * 100 + 0.5) \rfloor)/100$

Properties of floor and ceiling functions

For all real numbers x and integers m

1. $x - 1 \leq \lfloor x \rfloor \leq x \leq \lceil x \rceil \leq x + 1$

2. $\lceil -x \rceil = -\lfloor x \rfloor$

3. $\lfloor -x \rfloor = -\lceil x \rceil$

4. $\lfloor x + m \rfloor = \lfloor x \rfloor + m$

5. $\lceil x + m \rceil = \lceil x \rceil + m$

Everybody should know

- The definition of a function
- The definition of
  domain, codomain, range, image, preimage,
  one-to-one, onto function,
  one-to-one correspondence (bijection),
  inverse function, composition of functions.
- Given a function, determine its type.
- Given two functions, find their composition.
- Properties of floor and ceiling functions.

Integers and Division

We review basic elements of **number theory** and introduce some notions needed later.

Some elements of number theory are needed in:

Data structures,
Random number generation,
Encryption of data for secure data transmission,
Scheduling, etc.

For integers a and b with $a \neq 0$ we define

a **divides** b iff $\exists$ an integer c such that

$$b = ac$$

a divides b is written as $a \mid b$

$3 \mid 15$
$3 \nmid 16$
$4 \mid 16$
$16 \nmid 4$

$a \neq 0$ and $a \mid b$ is equivalent to each of:

a is a **factor** of b

b is a **multiple** of a

Theorem: Let a, b, and c be integers. Then

(1) if $a \mid b$ and $a \mid c$ then $a \mid (b + c)$.

(2) if $a \mid b$ then $a \mid bc$ for all integers c.

(3) if $a \mid b$ and $b \mid c$ then $a \mid c$.

Prime and composite numbers

A **prime** is a positive integer p that has only two distinct positive factors, 1 and p.

Examples: $2, 3, 5, 7, 11, 13, 29, 53, 997, 7951, \ldots$

We will often use the term **positive integer**.
A positive integer is greater than 0.
(0 is neither negative nor positive.)

A positive integer greater that 1 which is not a prime is called **composite**.

Examples: $6 = 2 \cdot 3$, $35 = 5 \cdot 7$, $57 = 3 \cdot 19$, etc.

Fundamental Theorem of Arithmetic

Every positive integer can be written uniquely as a product of primes, where the prime factors are written in order of their size.

$40 = 2 \cdot 2 \cdot 2 \cdot 5 = 2^3 \cdot 5$

$42 = 2 \cdot 3 \cdot 7$

$780 = 2 \cdot 2 \cdot 3 \cdot 5 \cdot 13 = 2^2 \cdot 3 \cdot 5 \cdot 13$

$550 = 2 \cdot 5 \cdot 5 \cdot 11 = 2 \cdot 5^2 \cdot 11$

Theorem *If n is a composite number then n has a factor $\leq \sqrt{n}$.*

This is an important bound when trying to find a factorization of a number.

Note that factors come in pairs, $\{k, n/k\}$.

Example 1: $n = 311$

$\sqrt{311} \doteq 17.6$

Test division by 2, 3, 5, 7, 11, 13, 17.

If none of these divides 311, it is a prime, otherwise we have found a factor.

Example 2: $n = 253$

$\sqrt{253} \doteq 15.9$

Test division by 2, 3, 5, 7, 11, 13.

$253 = 11*23$

Factorization of very large numbers by computers is a difficult problem.

This fact is used by some encryption systems.
RSA encryption system, named after the inventors Rivest, Shamir, and Adelman.

Breaking a code would require factoring numbers with 250 to 500 digits that have only two prime factors, both large primes.

GCD and LCM

Definition: $GCD(a,b)$, called the **greatest common divisor** of a and b, is the largest factor of a and b.

$GCD(18,24) = 6$
$GCD(18,13) = 1$

When $GCD(a,b) = 1$, we say that a and b are relatively prime (or coprime).

Definition: $LCM(a,b)$ is the **least common multiple** of a and b. It is the smallest integer having a and b as factors.

$LCM(8,6) = 24$
$LCM(8,12) = 24$

GCD and LCM

The prime factorization of a and b can be used to find $GCD(a, b)$ or $LCM(a, b)$:

$$780 = 2 \cdot 2 \cdot 3 \cdot 5 \cdot 13 = 2^2 \cdot 3 \cdot 5 \cdot \quad 13$$
$$550 = 2 \cdot 5 \cdot 5 \cdot 11 \quad = 2 \cdot \quad 5^2 \cdot 11$$

$GCD(780, 550) = 2 \cdot 5 = 10$
take the factors common to both numbers.

$LCM(780, 550) = 2^2 \cdot 3 \cdot 5^2 \cdot 11 \cdot 13 = 42900$
take all factors in both numbers with highest exponent.

If $a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}$ and $b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n}$

$$gcd(a,b) = p_1^{min(a_1,b_1)} p_2^{min(a_2,b_2)} \cdots p_n^{min(a_n,b_n)}$$

$$lcm(a,b) = p_1^{max(a_1,b_1)} p_2^{max(a_2,b_2)} \cdots p_n^{max(a_n,b_n)}$$

Note that $min(a_i,b_i) + max(a_i,b_i) = a_i + b_i$, leading to

---

Theorem Let a and b be positive integers. Then

$$ab = gcd(a,b) \cdot lcm(a,b)$$

Example:
$GCD(780, 550) = 2 \cdot 5 = 10$
$780 \cdot 550 = 429000$
$LCM(780, 550) = 42900$

Co-prime integers

Definition: The integers a and b are said to be **co-prime** or **relatively prime** if $gcd(a, b) = 1$.

Example 1:
6 and 25 are co-prime, as $gcd(6, 25) = 1$.

Example 2:
6 and 27 are not co-prime, since $gcd(6, 27) = 3 \neq 1$.

Example 3:
Any two distinct prime numbers are relatively prime.

The Division Algorithm

Let a be an integer and d a positive integer. Then there exist unique integers q and r,
$0 \leq r < d$, such that

$$a = dq + r$$

a is called the **dividend**
d is called the **divisor**
r is called the **remainder**
q is called the **quotient**.

Modular Arithmetic

Let a be an integer and m be a positive integer.

$$a \bmod m$$

is defined as the remainder when a is divided by m.

$$0 \leq (a \bmod m) < m$$

$8 \bmod 7 = 1$
$12 \bmod 7 = 5$
$30 \bmod 7 = 2$
$-3 \bmod 7 = 4$ since $-3 = -1 \cdot 7 + 4$
$-22 \bmod 6 = 2$ since $-22 = -4 \cdot 6 + 2$

Example of the use of *mod*:

We have *processors* $1, 2, 3, 4, 5$
and *jobs* $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...$

Scheduling: Given a job number, select a processor on which to execute the job.

Round-robin scheduling:
jobs $1, 6, 11, 16, 21, ...$ are done on processor 2
jobs $2, 7, 12, 17, 22, ...$ are done on processor 3
jobs $3, 8, 13, 18, 23, ...$ are done on processor 4
jobs $4, 9, 14, 19, 24, ...$ are done on processor 5
jobs $5, 10, 15, 20, 25, ...$ are done on processor 1
job i is assigned to processor $(i \bmod 5) + 1$

Congruences

Definition: Let a and b be integers and m be a positive integer. We say that a is **congruent** to b **modulo** m if $m \mid (a - b)$.

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

Examples:

$5 \mid (14 - 9) \iff 14 \equiv 9 \pmod{5}$

$5 \mid (19 - 9) \iff 19 \equiv 9 \pmod{5}$

$5 \mid (32 - 12) \iff 32 \equiv 12 \pmod{5}$

$7 \mid (14 - 7) \iff 14 \equiv 7 \pmod{7}$

Theorem Let a and b be integers and m be a positive integer.
$a \equiv b \pmod{m} \iff (a \bmod m) = (b \bmod m)$

Theorem
Let a and b be integers and m be a positive integer.

$$a \equiv b \pmod{m} \text{ iff } a = b + km \text{ for some integer } k$$

<u>Problem:</u> Find all integers congruent to 7 modulo 6.
The answer is the infinite set $\{a : a = 7 + 6k,\ k \in Z\}$.

$7 \equiv 13 \pmod{6}$ $\quad$ $7 \equiv 19 \pmod{6}$
$7 \equiv 37 \pmod{6}$ $\quad$ $7 \equiv 1 \pmod{6}$
$7 \equiv -5 \pmod{6}$ $\quad$ $7 \equiv -11 \pmod{6}$

Theorem Let m be a positive integer.
If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$ then

$$a + c \equiv b + d \pmod{m}$$

$$a \cdot c \equiv b \cdot d \pmod{m}$$

Everybody should know

- Definition of $a \mid b$, factor, multiple, **prime** and **composite** numbers.
- The fundamental theorem of arithmetic and how to do **prime factorizations**.
- GCD and LCM.
- The Euclidean algorithm for computing the GCD.
- The division algorithm.
- The definition of $a \bmod m$ and the notion of **congruence modulo** m.