Instituto Nacional de Astrofísica,
Óptica y Electrónica

TECHNICAL REPORT No. 697

Optics Department

Basic Discrete Mathematics
A Learning Guide

Gonzalo Urcid Serrano, Ph.D.

May 2025

Tonantzintla, Puebla.

Luis Enrique Erro No.1 Sta María Tonantzintla
C.P. 72840, Puebla, México

Preface

This technical report gives, in presentation style, a selection of topics that conforms a course in *Basic Discrete Mathematics* and its main purpose is to be used as a learning guide for students or self-educated persons interested in this area of mathematics. The use of colors in text, text backgrounds, and graphical elements is an essential characteristic of this kind of style format, helping the reader to distinguish and localize key words, fundamental ideas, or relevant suggestions.

The report can also serve as supporting material or as a didactic tool for known textbooks treating the same subject. A representative list of bibliographical references is provided in the next page. Thus, lecturers, instructors, or teaching assistants may also take advantage of the way topics are exposed herein.

As prerequisites for a better understanding of the kind of mathematics given here, the reader must have a background on general *Algebra*, elementary *Analytical Geometry*, and basic *Calculus*. Also, some *Computer Programming* knowledge and practice coding algorithms is required.

Gonzalo Urcid
Tonantzintla, May 22nd, 2025

Bibliographical References

(1975) Tremblay J.P. & Manohar R. *Discrete Mathematical Structures with Applications to Computer Science*, McGraw-Hill.
(1976) Prather R.E. *Discrete Mathematical Structures for Computer Science*. Houghton Mifflin.
(1985) Liu C.L. *Elements of Discrete Mathematics 2nd Ed.*, McGraw-Hill.
(1991) Abellanas M. & Lodares D. *Matemática Discreta*. Macrobit & Ra-Ma.
(1994) Graham R.L., Knuth D.E. & Patashnik O. *Concrete Mathematics: A Foundation for Computer Science 2nd Ed.* Addison-Wesley.
(1995) Ferrando J.C. & Gregori V. *Matemática Discreta 2da Ed.* Reverté.
(1997) Knuth D.E. *The Art of Computer Programming: Fundamental Algorithms 3rd Ed*. Addison-Wesley.
(2000) Rosen K.H., Michaels J.G., Gross J.L., Grossman J.W. & Shier D.R. *Handbook of Discrete and Combinatorial Mathematics*, CRC Press.
(2004) Grimaldi R.P. *Discrete and Combinatorial Mathematics: An Applied Introduction 5th Ed.* Pearson Addison-Wesley.
(2008) Matousek J. & Nesetril J. *Invitation to Discrete Mathematics 2nd Ed*. Oxford University Press.
(2014) Kolman B., Busby R. & Ross, S. *Discrete Mathematicsl Structures 6th Ed*. Pearson.
(2015) García-Merayo F. *Matemática Discreta 3ra Ed.* Paraninfo.
(2017) Espinosa-Armenta R. *Matemáticas Discretas 2da Ed.* Alfaomega.
(2017) Hunter D.J. *Essentials of Discrete Mathematics 3rd Ed.* Jones & Bartlett Learning.
(2018) Johnsonbaugh R. *Discrete Mathematics 8th Ed.* Pearson Education.
(2019) Rosen K.H. *Discrete Mathematics and Its Applications 8th Ed.* McGraw-Hill.
(2020) Epp S.S. *Discrete Mathematics with Applications 5th Ed.* Cengage.
(2020) Kumar B.V.S. & Dutta H. *Discrete Mathematical Structures: A Succinct Foundation.* CRC Press-Taylor & Francis Group.
(2022) Lipschutz S. & Lipson M. *Schaum's Outline of Discrete Mathematics 4th Ed.* McGraw-Hill.
(2024) Doerr A. & Levasseur K. *Applied Discrete Structures 3rd Ed*. CCA-SA 3.0, discretemath.org

Table of Contents

Table of Contents

Introduction a

Goal: To learn how to think mathematically.

1) Mathematical reasoning.
   Read, understand, and construct mathematical arguments.

2) Combinatorial analysis.
   Ability to count objects using basic and advanced techniques.

3) Discrete structures.
   Represent and relate discrete objects mathematically.

4) Algorithmic thinking.
   Specify and solve a problem by means of an algorithm
   in pseudocode language.

What is discrete mathematics?

- The study of discrete objects using mathematics,
- *Calculus* is based on the concept of *continuity*, but discrete math deals with separated, disconnected or discontinuous objects.
- Discrete objects are finite in nature and can be represented by natural or integer numbers.
- In a technological sense, discrete means digital.

Why do we need to study discrete mathematics?

- To develop our mathematical maturity, and our ability to understand and create mathematical proofs.

- As a gateway to more advanced courses in computer science such as data structures, algorithm analysis, database theory, formal languages, and computer security to name a few.

- To solve problems in applied sciences or engineering for example, in the industrial, chemical or biological areas.

Logic: Part I

- Propositions
- Operators
- Truth tables
- Implications
- Truth table construction

- The rules of logic give precise meaning to mathematical statements.
- Used to distinguish between valid and invalid math arguments.
- Also used to design computer circuits, construction of programs, and verification of correctness (in programs).

A proposition is a statement that is either *true* or *false*, but not both. Is is the basic building block on which logic is founded.

<u>Notation</u>

Truth values	**T** (true)	**F** (false)
Propositions	p, q, r, s, … (lower case)	

Propositions examples

Examples	Proposition?
$x + y = y + x$ for every pair of real numbers *x,y.*	Yes, **T**
Answer this question.	No, ??
Yesterday was our first class.	Yes, **F**
p $= "x + y = y + z \; if \; x = z."$	Yes, ***p*** is true
q = "Can you give me a prime number?"	No, ***q*** is a question
Miami is the capital of Florida.	Yes, **F**

A compound proposition is constructed by combining one or more <u>propositions</u> using <u>logical operators</u> and <u>connectives</u>.

Basic logic operators or connectives

Symbol	Meaning	Name
$\sim p$	it is not the case that ***p***	negation
$p \wedge q$	***p*** and ***q***	conjunction
$p \vee q$	*p* or (inclusive) ***q***	disjunction (could be both)
$p \oplus q$	***p*** or (exclusive) ***q***	exclusive or (not both)
$p \rightarrow q$	if ***p*** then ***q***	implication

hypothesis, premise

conclusion, consequence

NOTE: negation is a unary operation, the others are binary operations.

Operators examples

Let be, for example,

p = you have the flu.
q = you miss the final examination.
r = you pass the course.

$q \vee r \vee p$ = you miss the final exam, or pass the course, or have the flu.

$q \rightarrow \sim r$ = if you miss the final examination then you will not pass the course.

$r \oplus \sim r$ = you have the flu or you do not have the flu.

$\sim p \wedge \sim q$ = you have no flu and you did not miss the final examination.

How do we find the truth value of a compound proposition?

Use the logical values of each operator as defined by their truth tables.

Truth tables

given	find truth value of
p	$\sim p$
p,q	$p \wedge q$
p,q	$p \vee q$

p	$\sim p$
T	F
F	T

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

given	find truth value of
p, q	$p \oplus q$
p, q	$p \to q$

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

A brief exercise: find the truth tables for the compound propositions

a) $\boxed{\sim q \to \sim p}$ b) $\boxed{(p \to q) \wedge (q \to p)}$

Other ways of expressing an implication: $p \rightarrow q$

- "if ***p***, then ***q***"
  If you log on to the server, then you have a valid password.

- "***p*** is sufficient for ***q***"
  Logging on to the server is sufficient for a having a valid password.

- "***p*** implies ***q***"
  To log on to the server implies to have a valid password.

- "***q*** is necessary for ***p***"
  A valid password is necessary to log on to the server.

$p \rightarrow q \Leftrightarrow \sim q \rightarrow \sim p$

Contrapositive

$q \rightarrow p \Leftrightarrow \sim p \rightarrow \sim q$

Converse

We can consider compound propositions as functions of several logical variables having values in the set {**T**,**F**}.

rows in table

$n=1$ variable, $f(p)=f(p_1)=\sim p_1$ 2

$n=2$ variables, $g(p,q)=g(p_1,p_2)=\sim(p_1\vee p_2)$ 4

$n=3$ variables, $h(p,q,r)=h(p_1,p_2,p_3)=(p_1\to p_2)\to p_3$ 8

$\vdots$

n variables, $f(p_1,p_2,\ldots,p_n)$ 2^n

Each logical variable can assume the value **T** or **F**, therefore, the number of rows in the table for ***f*** is,

$$\overbrace{2\cdot 2\cdot\cdots\cdot 2}^{n}=2^n$$

Truth table examples[a]

a) $\sim q \rightarrow \sim p$

p	q	$\sim q$	$\sim p$	$\sim q \rightarrow \sim p$		$p \rightarrow q$
T	T	F	F	T	$\Leftrightarrow$	T
T	F	T	F	F		F
F	T	F	T	T		T
F	F	T	T	T		T

b) $(p \rightarrow q) \wedge (q \rightarrow p)$

p	q	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \wedge (q \rightarrow p)$	$p \leftrightarrow q$		$\sim (p \oplus q)$
T	T	T	T	T	T	$\Leftrightarrow$	T
T	F	F	T	F	F		F
F	T	T	F	F	F		F
F	F	T	T	T	T		T

This conjunction is called biconditional.

Truth table examples[b]

p	q	$\sim p$	$p \rightarrow q$	$\sim p \rightarrow q$	$(p \rightarrow q) \wedge (\sim p \rightarrow q)$
T	T	F	T	T	T
T	F	F	F	T	F
F	T	T	T	T	T
F	F	T	T	F	F

4 rows

p	q	r	$\sim q$	$\sim q \vee r$	$p \rightarrow (\sim q \vee r)$
T	T	T	F	T	T
T	T	F	F	F	F
T	F	T	T	T	T
T	F	F	T	T	T
F	T	T	F	T	T
F	T	F	F	F	T
F	F	T	T	T	T
F	F	F	T	T	T

8 rows

Logic: Part II

- Bits and bit string operations
- Logical equivalences
- Basic logical identities
- Algebraic manipulations
- Generalized operators

Logic-II

Bits & bit string operations

Language logic

- truth value
- logic variable
- proposition
- logical operations
- sequence of logical values

$$\{T, F\}$$

$$\{\sim, \wedge, \vee, \oplus\}$$

$$s = (T, T, F, F, T, F)$$

Computer logic

- bit (binary digit)
- boolean variable
- boolean expression
- bit operations
- bit string

$$\{0,1\}$$

$$\{\mathrm{NOT}, \mathrm{AND}, \mathrm{OR}, \mathrm{XOR}\}$$

$$s = 110010$$ ← length 6

Bitwise operations are performed on each corresponding bit for <u>two bit strings of the same length</u>.

$$\left.\begin{array}{l} x = 110010 \\ y = 011011 \end{array}\right\} \Rightarrow \left\{\begin{array}{l} x \wedge y = 010010 \\ x \oplus y = 101001 \\ x \vee y = 111011 \end{array}\right.$$

Equivalences[a]

- a proposition that is always true is called a tautology,
- a proposition that is always false is called a contradiction,
- otherwise it is called a contingency.

p	$\neg p$	$p \wedge \sim p$	$p \rightarrow \sim p$	$\sim p \rightarrow p$	$p \vee \sim p$
T	F	F	F	T	T
F	T	F	T	F	T

We also use algebraic notation, for example,

$$p \wedge \sim p = F \qquad p \vee \sim p = T$$

We say that ***p*** is logically equivalent to ***q*** if and only if the biconditional between ***p*** and ***q*** is a tautology. In symbols,

$$p \Leftrightarrow q \text{ when } p \leftrightarrow q = T$$

How do we verify a logical equivalence?

1) *by showing that **p** and **q** have the same final column in their respective truth tables, or*

2) *by reducing **p** to **q** using the rules of logic in algebraic notation.*

Completeness: a proposition ***p*** can be build using only the set of primitive logical connectives, that is to say, from {NOT, AND, OR}.

Duality: a proposition ***p*** has a dual proposition ***p***** obtained by exchanging AND's with OR's, and T's with F's.

Logic-II

Identities[a]

De Morgan's laws: $\boxed{\sim(p\wedge q)\Leftrightarrow\sim p\vee\sim q}$

p	q	$\sim p$	$\sim q$	$\sim(p\wedge q)$		$\sim p\vee\sim q$	$\sim(p\wedge q)\leftrightarrow\sim p\vee\sim q$
T	T	F	F	F	T	F	T
T	F	F	T	T	F	T	T
F	T	T	F	T	F	T	T
F	F	T	T	T	F	T	T

same last column

biconditional is a tautology

Using the principle of duality, another equivalence is immediately established, this is the second logic law of De Morgan,

$$\boxed{\sim(p\vee q)\Leftrightarrow\sim p\wedge\sim q}$$

Logical equivalence	Law name
$p \wedge T \Leftrightarrow p$	identity
$p \wedge F \Leftrightarrow F$	domination
$p \wedge p \Leftrightarrow p$	idempotent
$\sim(\sim p) \Leftrightarrow p$	double negation
$p \wedge q \Leftrightarrow q \wedge p$	commutative
$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$	associative
$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	distributive
$\sim(p \wedge q) \Leftrightarrow \sim p \vee \sim q$	De Morgan

All these fundamental logical identities or equivalences, and their duals are proved using truth tables.

Example of algebraic reduction using the <u>fundamental equivalences</u>:

The following implication is a tautology $\boxed{[\sim p \wedge (p \vee q)] \rightarrow q}$

$\Leftrightarrow [(\sim p \wedge p) \vee (\sim p \wedge q)] \rightarrow q$	distributive law
$\Leftrightarrow [F \vee (\sim p \wedge q)] \rightarrow q$	<u>contradiction</u>
$\Leftrightarrow (\sim p \wedge q) \rightarrow q$	identity law
$\Leftrightarrow \sim (\sim p \wedge q) \vee q$	<u>equivalence</u> of implication operator
$\Leftrightarrow (\sim (\sim p) \vee \sim q) \vee q$	De Morgan's law
$\Leftrightarrow p \vee (\sim q \vee q)$	double negation and associative law
$\Leftrightarrow p \vee T \Leftrightarrow T$	<u>tautology</u> and domination law

Generalized operations

The associative law allows to take out parentheses from an expression containing only conjunctions *xor* disjunctions. So, we can write, safely,

$$p_1 \wedge p_2 \wedge p_3 \qquad \text{instead of} \qquad (p_1 \wedge p_2) \wedge p_3$$

The generalized conjunction and disjunction are defined as:

$\wedge_{i=1}^{n} p_i = p_1 \wedge \cdots \wedge p_n$ — true when each $\boldsymbol{p}_i$ is true,

$\vee_{i=1}^{n} p_i = p_1 \vee \cdots \vee p_n$ — true when at least one $\boldsymbol{p}_i$ is true.

Example, the generalized De Morgan's laws are written as:

$$\sim \wedge_{i=1}^{n} p_i = \vee_{i=1}^{n} (\sim p_i) \quad \text{operator exchange}$$

$$\sim \vee_{i=1}^{n} p_i = \wedge_{i=1}^{n} (\sim p_i)$$

transfer
negation out-in

Logic: Part III

- Predicates
- Quantifiers
- Examples, one variable
- Examples, two variables
- Binding variables
- Quantifier negation

Predicates[a]

Reminder: *compound proposition* = *logical function* = *boolean expression,* they depend only on logical variables assuming values in the set {**T**,**F**}={**1**,**0**}.

$(x \leq 4) \wedge (x > -\infty)$

m is an odd number

Word ***x*** contains letter "a".

$P(x),\ x \in R$

$Q(m),\ m \in Z$

$S(x),\ x \in W$

these statements have other kinds of variables whose values belong to a certain set such as real or integer numbers, or words.

The set ***U*** from which ***x*** takes its values is called the universe of discourse or the set under discussion.

A propositional function is a predicate ***P*** about an object ***x*** or a property that ***x*** can have.

$P(x),\ x \in U$

one argument

$Q(x, y),\ x, y \in U$

two arguments

Is a propositional function *P*(*x*) a proposition?

NO if ***x*** remains without a value, YES if ***x*** is substituted by a specific value.

$$P(x) = (x \le 4) \wedge (x > -\infty)$$

Q(m) = *m* is an odd number

S(x) = word *x* contains letter "a"

$$\begin{cases} P(3) = T \\ P(5) = F \end{cases}$$

$$\begin{cases} Q(1098) = F \\ Q(2001) = T \end{cases}$$

$$\begin{cases} S(applied) = T \\ S(discrete) = F \end{cases}$$

1

$$P(x) \xrightarrow{\text{propositional function}} \boxed{\text{give a value } x = x_0 \in U} \xrightarrow{\text{proposition}} P(x_0) = \begin{cases} T \\ F \end{cases}$$

2

$P(x)$ $\xrightarrow{\text{propositional function}}$ quantification <u>all</u>, <u>some</u>, or <u>only one</u> ***x*** $\xrightarrow{\text{proposition}}$ $Q[P(x)] = \begin{cases} T \\ F \end{cases}$

quantifier

universal quantifier	$\forall x\, P(x)$	<u>true</u> if *P(x)* is true for all values of *x* in *U*.
existential quantifier	$\exists x\, P(x)$	<u>true</u> if *P(x)* is true for some values of *x* in *U*.
we use the symbol	$\exists! x\, P(x)$	<u>true</u> if *P(x)* is true for only one value of *x* in *U*.

For a <u>finite number of elements</u>, $\{x_1, x_2, \ldots, x_n\} = U$

$$\forall x\, P(x) \Leftrightarrow \wedge_{i=1}^{n} P(x_i)$$

$$\exists x\, P(x) \Leftrightarrow \vee_{i=1}^{n} P(x_i)$$

$$\exists! x\, P(x) \Leftrightarrow \oplus_{i=1}^{n} P(x_i)$$

in fact, you can take the <u>quantifier operators</u> as a natural extension of “and”, “or”, “xor” applied to a *finite* or *infinite* number of objects.

- Every computer science student needs a course in discrete mathematics.

U = set of all computer science students
$P(x)$ = x needs a course in discrete mathematics

$$\forall x \in U, P(x)$$

- There is a student in this class who owns a personal computer.

U = set of all students in this class
$Q(x)$ = x owns a personal computer

$$\exists x \in U, Q(x)$$

- Truth value of $\forall n \in Z, (n^2 \geq 0)$

T, since a square is always ≥ 0

- Truth value of $\exists n \in Z, (n^2 = 2)$

F, since the only solution to the equation is not an integer.

- Truth value of $\exists! x P(x) \rightarrow \exists x P(x)$

T, since “only one *x*” can be taken as “at least one *x*” or “some *x*”.

- Every student in this class has taken at least one computer science course.

U_1 = set of all students in this class
U_2 = set of all courses in computer science
$P(x,y)$ = x has taken y

$$\forall x \in U_1 \, \exists y \in U_2, P(x, y)$$

- There is a student in this class who has been in every room of at least one building on campus.

U_1 = set of all students in this class
U_2 = set of all buildings on campus
U_3 = set of all rooms
$P(z,y)$ = z is in y, $Q(x,z)$ = x has been in z

$$\exists x \in U_1 \, \exists y \in U_2 \, \forall z \in U_3,$$
$$(P(z, y) \to Q(x, z))$$

- Truth value of $\forall n \in Z \, \exists m \in Z, (n + m = 0)$ — **T**, since $m = -n$

- Truth value of $\exists n \in Z \, \exists m \in Z, (n^2 + m^2 = 6)$ — **F**, try low values for m,n.

$$m, n = 0, \pm 1, \pm 2, \pm 3$$

Implicit quantifiers

- Remember the definition of a limit in calculus? $\lim_{x \to a} f(x) = L$

For every real number $\varepsilon > 0$ there exists a real number $\delta > 0$ such that $|f(x) - L| < \varepsilon$ whenever $0 < |x - a| < \delta$

hidden or implicit universal quantifier

$$\forall \varepsilon > 0 \; \exists \delta > 0 \; \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \varepsilon)$$

- Write out the quantification $\exists ! x P(x)$ using the other quantifiers, and the logical operators.

$$\exists ! x P(x) \Leftrightarrow \exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \rightarrow x = y)$$

only one *x* → some *x*   we assure that *x* = *y* (unique)

Could be another *y*?

Binding variables

Given a propositional function *P(x),* if a quantifier is applied to *P(x)* or a specific value of ***x*** is given, we say that variable ***x*** is bound, otherwise it is free.

$\forall x P(x, y)$	$\forall x \exists y Q(x, y)$	$\forall x \forall y P(x, y) \vee R(z)$	$\exists y Q(x_0, y)$
x is bound *y* is free	*x,y* are bound	*x,y* are bound *z* is free	*y* is bound x_0 is a value (bound).

Binding variables is the general process that gives us a proposition from a propositional function.

3

$$P(x_1, x_2, \ldots, x_n) = p \quad \text{if and only if} \quad \forall i, x_i \textit{ is bound}$$

Quantifier negation

It is not the case that for all *x*, *P*(*x*) = T

$\Updownarrow$

There is an *x* that makes *P*(*x*) = F

operator exchange

$$\boxed{\sim \forall x P(x) \Leftrightarrow \exists x \sim P(x)}$$

negation transfer

It is not the case that for some *x*, *P*(*x*) = T

$\Updownarrow$

For all *x*, *P*(*x*) = F

$$\boxed{\sim \exists x P(x) \Leftrightarrow \forall x \sim P(x)}$$

$$\boxed{\sim \exists y \exists x P(x,y)} \Leftrightarrow \forall y \sim \exists x P(x,y) \Leftrightarrow \boxed{\forall y \forall x \sim P(x,y)}$$

$$\boxed{\sim \forall x (\exists y \forall z P(x,y,z) \wedge \exists z \forall y P(x,y,z))} \Leftrightarrow \exists x \sim (\exists y \forall z P(\bullet) \wedge \exists z \forall y P(\bullet))$$

$$\Leftrightarrow \exists x (\sim \exists y \forall z P(\bullet) \vee \sim \exists z \forall y P(\bullet))$$

$$\Leftrightarrow \exists x (\forall y \sim \forall z P(\bullet) \vee \forall z \sim \forall y P(\bullet))$$

$$\boxed{\Leftrightarrow \exists x (\forall y \exists z \sim P(\bullet) \vee \forall z \exists y \sim P(\bullet))}$$

- show that $\exists x P(x) \wedge \exists x Q(x)$ is *not logically equivalent* to $\exists x (P(x) \wedge Q(x))$

P(*x*) = *x* is an even number
Q(*x*) = *x* is an odd number

$P(8) = T \rightarrow \exists x P(x)$

$Q(7) = T \rightarrow \exists x Q(x)$

then $\boxed{(\exists x P(x) \wedge \exists x Q(x)) = T}$

$\boxed{\updownarrow = F}$

but there is no integer number that is both even and odd at the same time, therefore, $\boxed{\exists x (P(x) \wedge Q(x)) = F}$

- truth value of $\forall n \forall m \exists p, (p = \dfrac{m+n}{2})$ where ***U*** is the set of integers.

false because, for example,

$$n = k \wedge m = k + 1 \rightarrow p = \frac{2k+1}{2} = k + \tfrac{1}{2} \notin Z$$

Sets

- Sets
- Basic operations
- Special operations
- Hasse diagrams
- Algebraic identities
- Generalized operations

A set is a finite or infinite collection of objects called elements. We usually consider that elements are of the same kind.

Notation: $S = \{e_1, e_2, \ldots, e_n, \ldots\} \vee S = \{x \in U \mid P(x) = T\}$

by listing the elements or by using a predicate

$$x \in U \wedge P(x) = T \rightarrow x \in S$$
$$x \in U \wedge P(x) = F \rightarrow x \notin S$$

membership relation between an element ***x*** and a set ***S***.

Basic notions

the set with all elements

the set without elements

✓ universal set, empty set $U = \{x \mid x = x\} \qquad \varnothing = \{x \mid x \neq x\} = \{\quad\}$

✓ subset $A \subseteq B \Leftrightarrow \forall x (x \in A \rightarrow x \in B)$ each ***x*** in ***A*** is also in ***B***.

✓ equal sets $A = B \Leftrightarrow A \subseteq B \wedge B \subseteq A$ ***A, B*** have the same elements.

✓ sets (that we will use)

- sets of numbers: natural, integers, real, and complex,
- sets of functions: polynomials, exponentials, and logarithms,
- sets of discrete objects: strings, edges, nodes, etc.

✓ family of sets

a set whose elements are **SETS**!

Sets

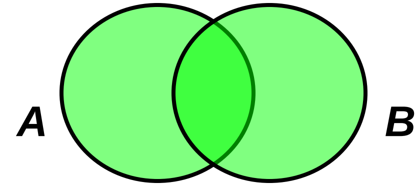
Basic operations

Basic set operators

Definition

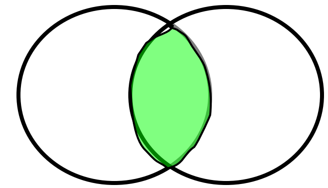
✓ union

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



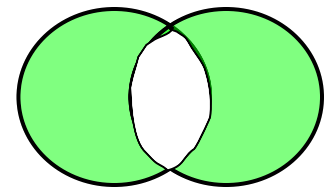
✓ intersection

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



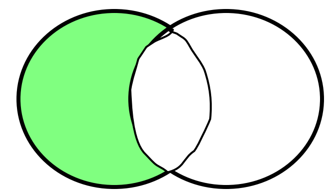
✓ symmetric difference

$$A \oplus B = \{x \mid x \in A \oplus x \in B\}$$



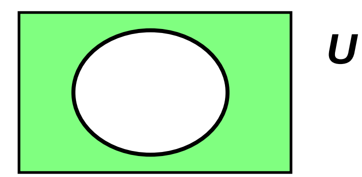
✓ difference

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

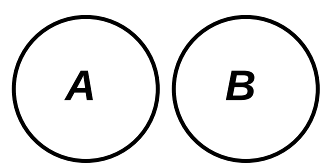


✓ complementation

$$A^c = U - A$$



Two sets are disjoint if $A \cap B = \varnothing$



Venn diagrams

The following operations provides us with additional tools for working with sets.

Additional set operators

	Definition	Examples
✓ cardinality	$\boxed{\lvert A \rvert = card(A)}$ number of elements in ***A***	$A = \{x \in N \mid x < 10\} \to \lvert A \rvert = 10$ $card(N) = card(Z) = \infty$
✓ power set	$\boxed{P(A) = \{S \mid S \subseteq A\}}$ is the family of all subsets ***S*** of ***A*** (including itself and the empty set)	$P(\{0,1\}) = \{\varnothing, \{0\}, \{1\}, \{0,1\}\}$ $P(\varnothing) = \{\varnothing\}$ $\lvert A \rvert = n, n \in N \to \lvert P(A) \rvert = 2^n$
✓ Cartesian product	$\boxed{A \times B = \{(a,b) \mid a \in A \wedge b \in B\}}$ the set of all ordered pairs formed from the sets ***A*** and ***B***	$\{a,b\} \neq (a,b)$ $A = B = \{T, F\} \to A \times B =$ $\{(T,T), (T,F), (F,T), (F,F)\}$

Determine whether each of the following statements is true or false:

$x \in \{x\}$	T	$\{x\} \subseteq \{x\}$	T	$\{x\} \in \{x\}$	F
$\{x\} \in \{\{x\}\}$	T	$\varnothing \subseteq \{x\}$	T	$\varnothing \in \{x\}$	F

Suppose that ***A***, ***B***, and ***C*** are sets such that ***A*** is included in ***B***, and ***B*** is part of ***C***. Show that ***A*** is a subset of ***C***, this is known as inclusion transitivity.

By definition:

$$\boxed{A \subseteq B \Leftrightarrow \forall x(x \in A \to x \in B)}$$

$$B \subseteq C \Leftrightarrow \forall x(x \in B \to x \in C)$$

Take an arbitrary value of ***x***, call it x_0 then

$$x_0 \in A \to x_0 \in B = p \to q$$

$$x_0 \in B \to x_0 \in C = q \to r$$

From logic, we know that:

$$\boxed{(p \to q) \wedge (q \to r) \Leftrightarrow (p \to r)}$$

So,

$$x_0 \in A \to x_0 \in C = p \to r$$

and again, by definition:

$$\boxed{\forall x(x \in A \to x \in C) \Leftrightarrow A \subseteq C}$$

Show that $A \oplus B = (A - B) \cup (B - A)$. We have to prove that both sets are equal.

By definition: $\boxed{A \oplus B = \{x \mid x \in A \oplus x \in B\}}$ now pick an arbitrary ***x***, say $\boldsymbol{x}_0$, then

$$x_0 \in A \oplus B \Leftrightarrow (x_0 \in A \vee x_0 \in B) \wedge (x_0 \notin (A \cap B))$$ belongs to ***A*** or ***B*** but not both.

$$\Leftrightarrow (x_0 \in A \wedge x_0 \notin (A \cap B)) \vee (x_0 \in B \wedge x_0 \notin (A \cap B))$$ distributive law

Also, we have that $\boxed{x_0 \notin A \cap B \Leftrightarrow x_0 \in (A \cap B)^c \Leftrightarrow x_0 \notin A \vee x_0 \notin B}$ (De Morgan)

$$(x_0 \in A \wedge x_0 \notin (A \cap B)) \Leftrightarrow F \vee (x_0 \in A \wedge x_0 \notin B)$$

$$(x_0 \in B \wedge x_0 \notin (A \cap B)) \Leftrightarrow F \vee (x_0 \in B \wedge x_0 \notin A)$$ therefore,

$$x_0 \in A \oplus B \Leftrightarrow (x_0 \in A \wedge x_0 \notin B) \vee (x_0 \in B \wedge x_0 \notin A)$$ finally, at the set level,

$$\boxed{A \oplus B = \{x \mid (x \in A \wedge x \notin B) \vee (x \in B \wedge x \notin A)\}}$$

$$= (A - B) \cup (B - A)$$

from which the result follows by the definition of set difference.

Sets Examples[c]

For sets ***A***, ***B*** show that $(A \cap B) \cup (A \cap B^c) = A$

$$(A \cap B) \cup (A \cap B^c) = [(A \cap B) \cup A] \cap [(A \cap B) \cup B^c]$$

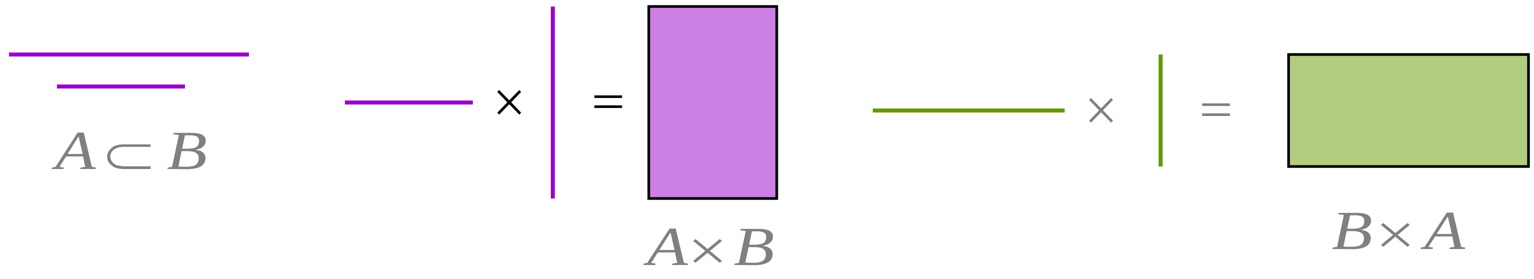
$$= [(A \cup A) \cap (B \cup A)] \cap [(A \cup B^c) \cap (B \cup B^c)]$$

ONLY intersections, $= A \cap \boxed{(B \cup A)} \cap \boxed{(A \cup B^c)} \cap \boxed{U} = A$ since ***A*** is a subset of the other three terms.

The Cartesian product is not commutative for ***A***, ***B*** nonempty sets, unless ***A*** = ***B***.

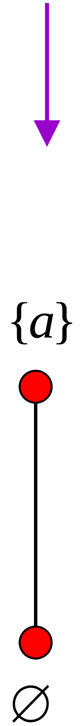
$A = B \rightarrow A \times A = A \times A$	If both sets are equal there is nothing to prove.
$A \neq B \rightarrow A \times B \neq B \times A$	Since ***A***, ***B*** are not equal we can make some choices:

For example,

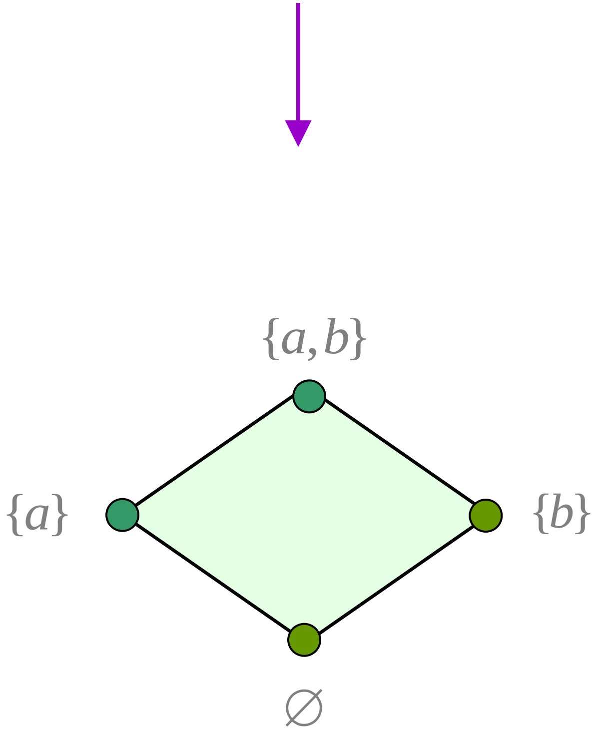


Sets — Hasse diagrams

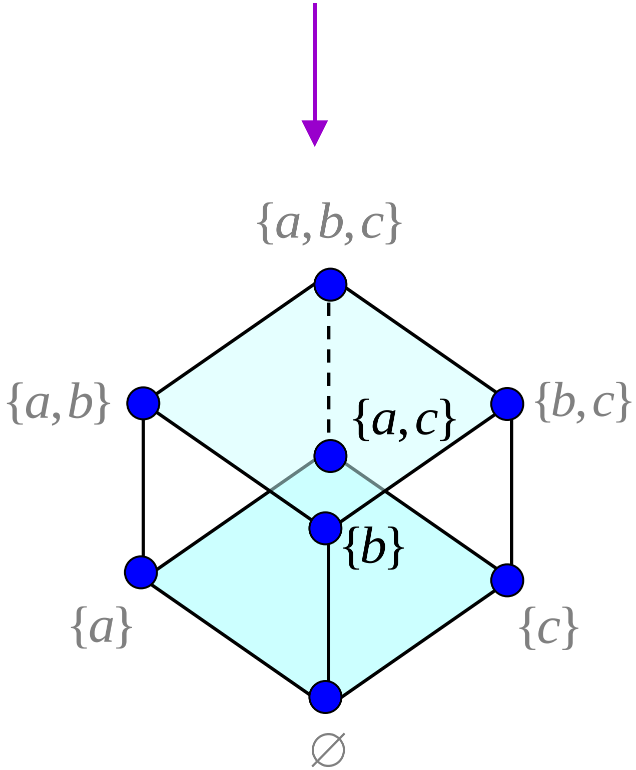
$U = \{a\}$



$U = \{a, b\}$



$U = \{a, b, c\}$



These graphical representations of the power set of ***U*** are known as Hasse diagrams.

Algebraic identities

The laws of sets are the same as the laws of logic, they share the same <u>algebraic structure</u>. Completeness and duality can also be applied to sets.

<u>Symbol exchange</u>

T, F	$U, \varnothing$
p, q, r	A, B, C
$\wedge$	$\cap$
$\vee$	$\cup$
$\sim p$	A^c

$\Rightarrow$

$A \cap U = A$	identity
$A \cap \varnothing = \varnothing$	domination
$A \cap A = A$	idempotent
$(A^c)^c = A$	double comp.

$A \cap B = B \cap A$	commutative
$(A \cap B) \cap C = A \cap (B \cap C)$	associative
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributive
$(A \cap B)^c = A^c \cup B^c$	De Morgan

They have the same form, so *we just have to remember one group of identities*.

Generalized operations

Suppose we have a finite family of sets $F = \{S_1, S_2, \ldots, S_n\}$, then we define the generalized set operations as follows:

- union

$$\bigcup F = \bigcup_{i=1}^{n} S_i = S_1 \cup S_2 \cup \cdots \cup S_n = \{x \mid \exists i, x \in S_i\}$$

- intersection

$$\bigcap F = \bigcap_{i=1}^{n} S_i = S_1 \cap S_2 \cap \cdots \cap S_n = \{x \mid \forall i, x \in S_i\}$$

- Cartesian product

$$\prod F = \prod_{i=1}^{n} S_i = S_1 \times \cdots \times S_n$$
$$= \{(x_1, \ldots, x_i, \ldots, x_n) \mid \forall i, x_i \in S_i\}$$

This is called an ordered *n*-tuple

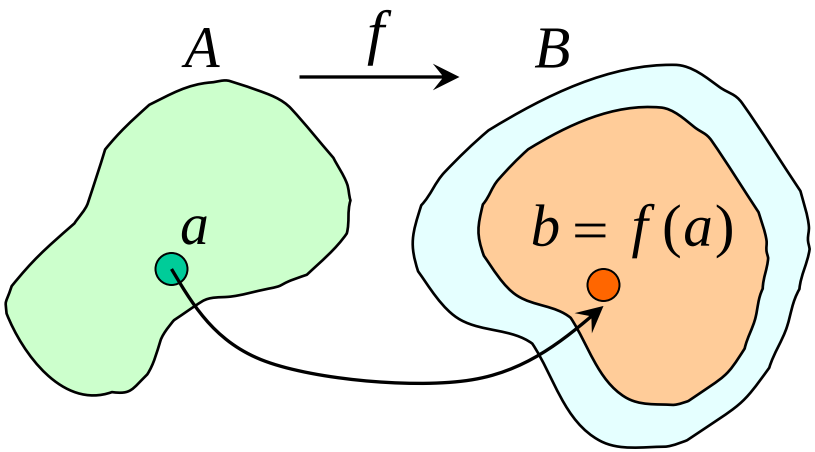
Functions: Part I

- Functions
- Classification
- Inverse and composition
- Discrete functions

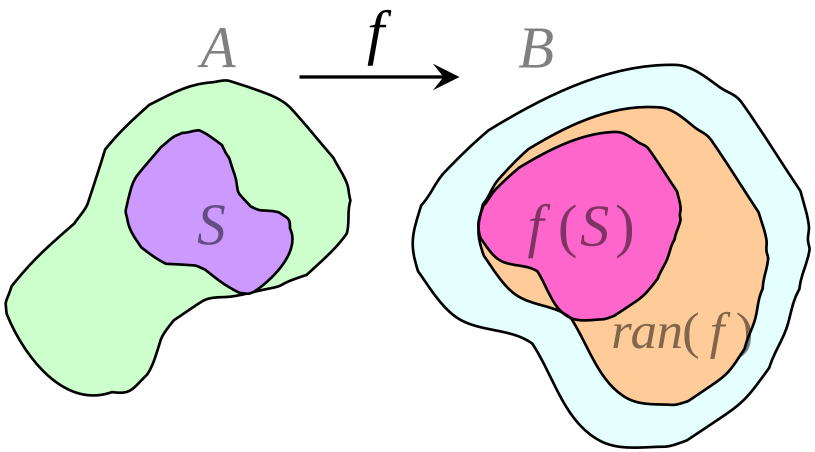
Functions-I

Concepts[a]

Concept	**Symbol**
• function	$f : A \to B$ $a \mapsto b = f(a)$
• domain	$dom(f) = A$
• codomain	$cod(f) = B$
• image of ***a***	$b\,;\; f(a) = b$
• pre-image	$a\,;\; f(a) = b$
• range	$ran(f) \subseteq B$
• image of ***S***	$f(S)\,;\, S \subseteq A$



A function ***f*** assigns only one element of ***B*** to each element of ***A***.



Functions-I Concepts[b]

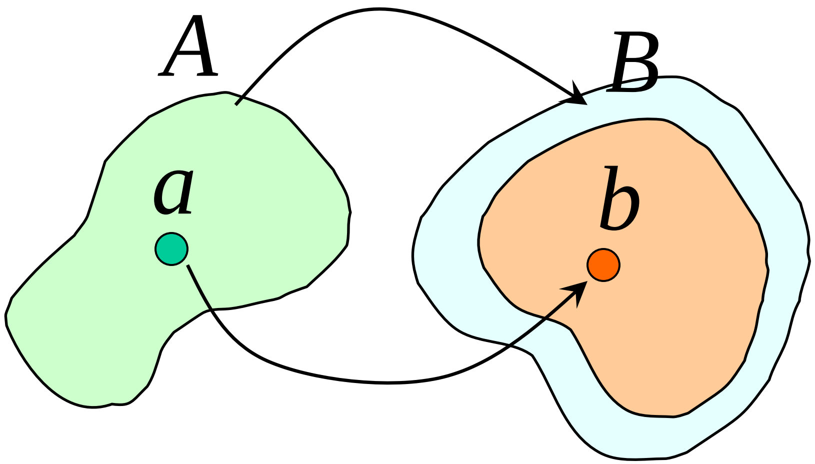
set level $A \to B$

Common verbal expressions are:

f is a function from ***A*** to ***B***,

f maps ***A*** to ***B***,

f is a transformation from ***A*** to ***B***.



element level $a \mapsto b$

We define the image of a subset ***S*** of ***A*** as:

$$\boxed{f(S) = \{f(a) \mid a \in S\}}$$

According to this definition the range of a function is $ran(f) = f(A)$

and the following chain of inclusions are valid,

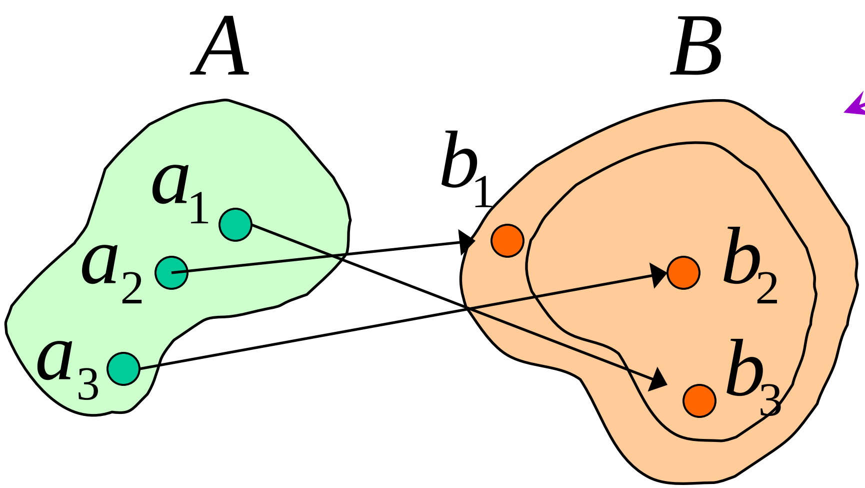
$$\boxed{f(S) \subseteq f(A) \subseteq B}$$

note that,

$$\boxed{f(\{a\}) = f(a) = b \in B}$$

Functions are classified as:

- injections (one-to-one) $\forall a_1, a_2 \in A, a_1 \neq a_2 \rightarrow f(a_1) \neq f(a_2)$
- surjections (onto) $\forall b \in B \exists a \in A, b = f(a)$
- bijections (both one-to-one and onto) $(\bullet) \wedge (\bullet)$



$a_1 \mapsto b_3$

$a_2 \mapsto b_1$

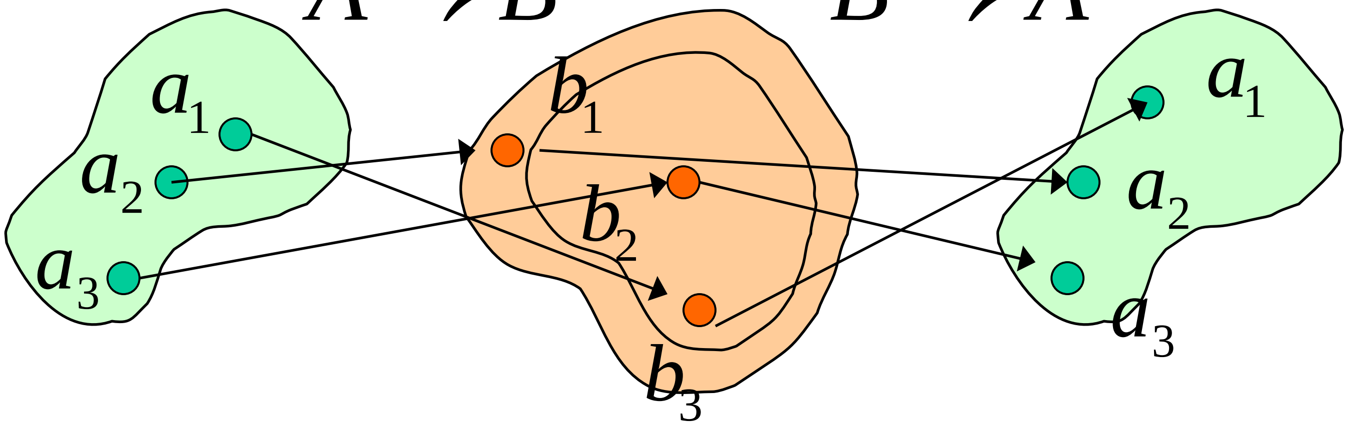
$a_3 \mapsto b_2$

Note that for a <u>surjection</u> or a <u>bijection</u>,

$$\boxed{f(A) = B}$$

Given a bijection ***f*** from ***A*** to ***B*** then it is possible to go back from ***B*** to ***A*** by means of the inverse function of ***f***.

$$A \xrightarrow{f} B \qquad B \xrightarrow{f^{-1}} A$$



$$[b = f(a)] \wedge [a = f^{-1}(b)]$$

The identity function from a set ***A*** to itself is the most elementary bijection:

$$id_A : A \to A;\ id_A(x) = x$$

Composition

Functions can be chained one after another by means of the composition operation that can be regarded as the most important algebraic operation between functions.

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[dr, "h"'] & & B \arrow[dl, "g"] \\ & C & \end{tikzcd}$$

$$h = g \circ f$$

$$h(x) = (g \circ f)(x) = g(f(x))$$

If ***g*** is the inverse function of ***f*** then ***h*** is the identity on ***A***.

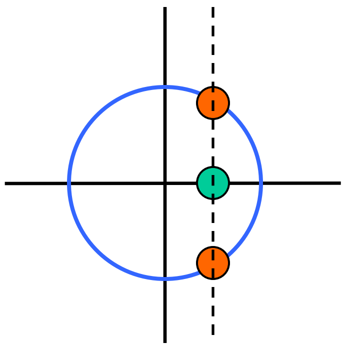
$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[dr, "id_A"'] & & B \arrow[dl, "f^{-1}"] \\ & A & \end{tikzcd}$$

Another possible diagram is the following, beginning with set ***B***,

$$\begin{tikzcd} B \arrow[rr, "f^{-1}"] \arrow[dr, "id_B"'] & & A \arrow[dl, "f"] \\ & B & \end{tikzcd}$$

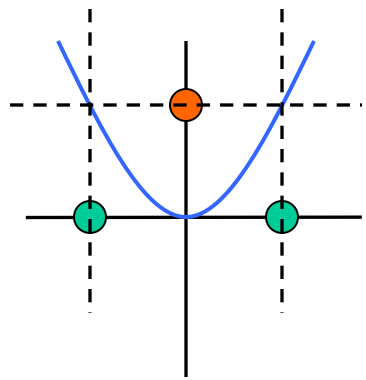
Functions-I

Examples[a]



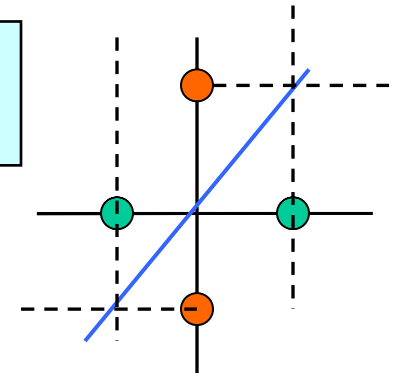
$\mathrm{circ}(x, y) \subseteq R^2$

It is not a function

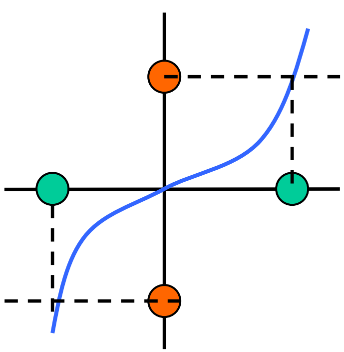


$x \mapsto x^2; \sim \mathrm{inj} \wedge \sim \mathrm{sur}$

$A = B = R$

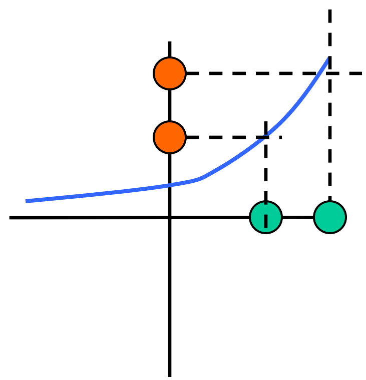


$x \mapsto x; \mathrm{bij}$



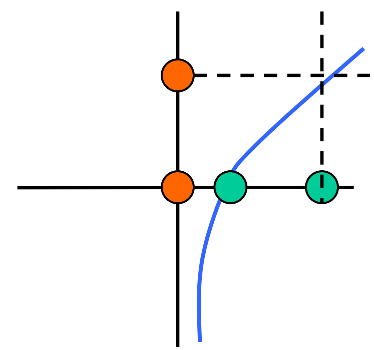
$x \mapsto x^3; \mathrm{bij}$

$A = B = R$



$x \mapsto \exp(x); \mathrm{bij}$

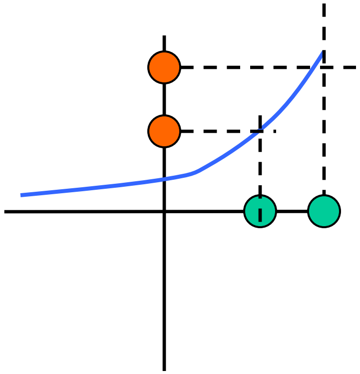
$A = R, B = R^+$



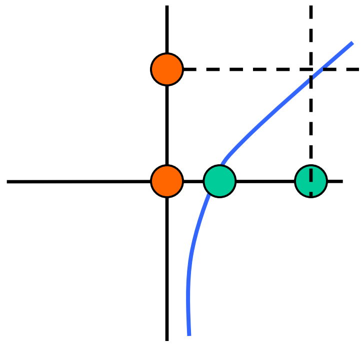
$x \mapsto \ln(x); \mathrm{bij}$

$A = R^+, B = R$

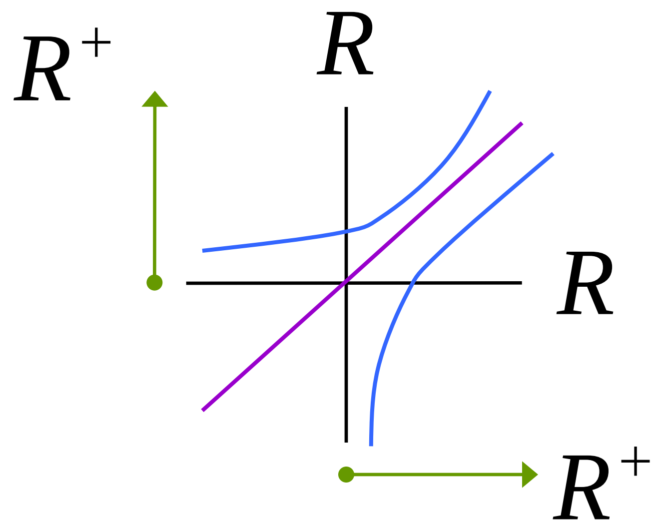
Examples[b]



$\exp: R \to R^{+}$



$\ln: R^{+} \to R$



$$\begin{tikzcd} R \arrow[rr, "\exp"] \arrow[dr, "id_R"'] & & R^{+} \arrow[dl, "\ln"] \\ & R & \end{tikzcd}$$

$$\begin{tikzcd} R^{+} \arrow[rr, "\ln"] \arrow[dr, "id_{R^{+}}"'] & & R \arrow[dl, "\exp"] \\ & R^{+} & \end{tikzcd}$$

In computer science the next functions are useful tools for several purposes, all of them have in common that the function values are discrete objects.

Discrete function	Definition	Value or image
cardinal	$\lvert\,\rvert : P(A) \to N$	$A \mapsto \lvert A \rvert = n$
characteristic	$c_A : U \to \{0,1\}$	$c_A(x) = \begin{cases} 1 ; x \in A \\ 0 ; x \notin A \end{cases}$
binary string	$\beta : P(U) \to B^n$	$A \subseteq U \mapsto s = s_1 s_2 \cdots s_n$ $\forall i, s_i = c_A(x_i)$
floor	$\lfloor\ \rfloor : R \to Z$	$\lfloor x \rfloor = m ; m \le x ; n \le x \to n \le m$
ceiling	$\lceil\ \rceil : R \to Z$	$\lceil x \rceil = m ; m \ge x ; n \ge x \to n \ge m$

- Find the value of $\left\lceil \left\lfloor \frac{1}{2} \right\rfloor + \left\lceil \frac{1}{2} \right\rceil + \frac{1}{2} \right\rceil = \lceil 0+1+0.5 \rceil = \lceil 1.5 \rceil = \boxed{2}$

- Let $f(x) = \lfloor x^2/3 \rfloor$ find $f(S) = f(\{-2,-1,0,1,2,3\})$

$$\left\{ \left\lfloor \frac{(-2)^2}{3} \right\rfloor, \left\lfloor \frac{(-1)^2}{3} \right\rfloor, \left\lfloor \frac{0^2}{3} \right\rfloor, \left\lfloor \frac{1^2}{3} \right\rfloor, \left\lfloor \frac{2^2}{3} \right\rfloor, \left\lfloor \frac{3^2}{3} \right\rfloor \right\} = \{1,0,0,0,1,3\} = \boxed{\{0,1,3\}}$$

- Show that for all ***x***, $\boxed{f_{A\oplus B}(x) = f_A(x) + f_B(x) - 2 f_A(x) f_B(x)}$

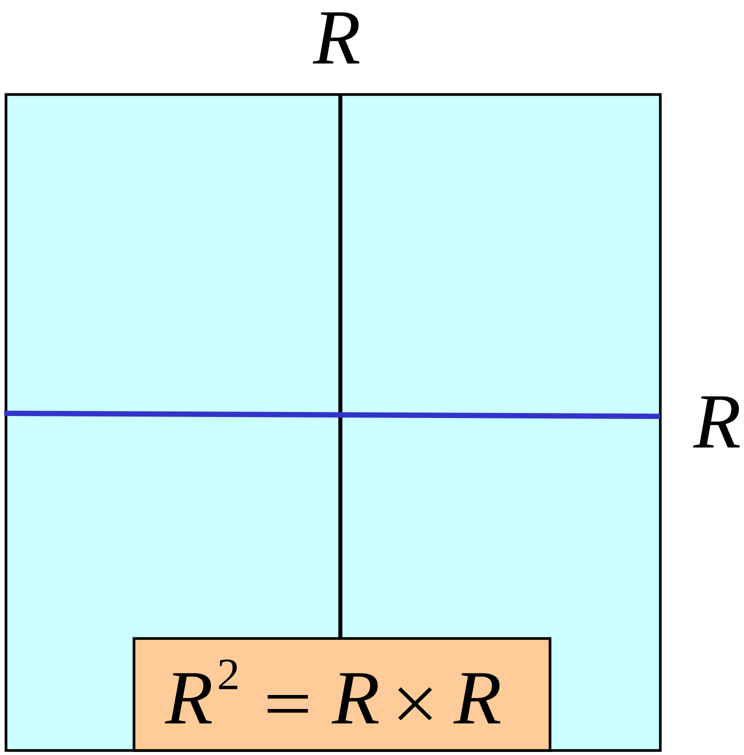
$$x \in A \oplus B \rightarrow f_{A\oplus B}(x) = 1 = \begin{cases} 1+0-2(1\cdot 0)\,;\, x \in A \wedge x \notin B \\ 0+1-2(0\cdot 1)\,;\, x \notin A \wedge x \in B \end{cases}$$

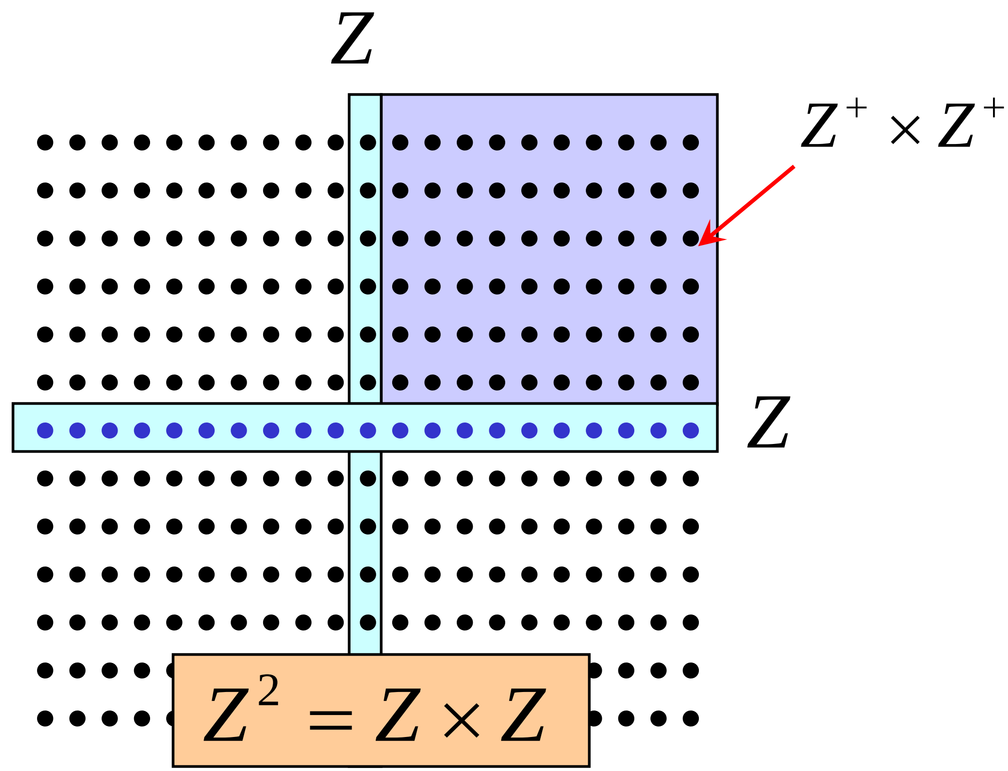
$$x \notin A \oplus B \rightarrow f_{A\oplus B}(x) = 0 = 1+1-2(1\cdot 1)\,;\, x \in A \cap B$$

Functions: Part II

- Discrete grids
- Floor and ceiling
- Sequences
- Summations
- Basic formulas

Functions-II — Discrete grids





This is a rectangular region representing the ***xy*** plane:

This is a rectangular set of points representing the discrete grid over the integers.

In discrete mathematics we are interested in functions defined over the ***xy*** plane, the discrete grid or one of their subsets.

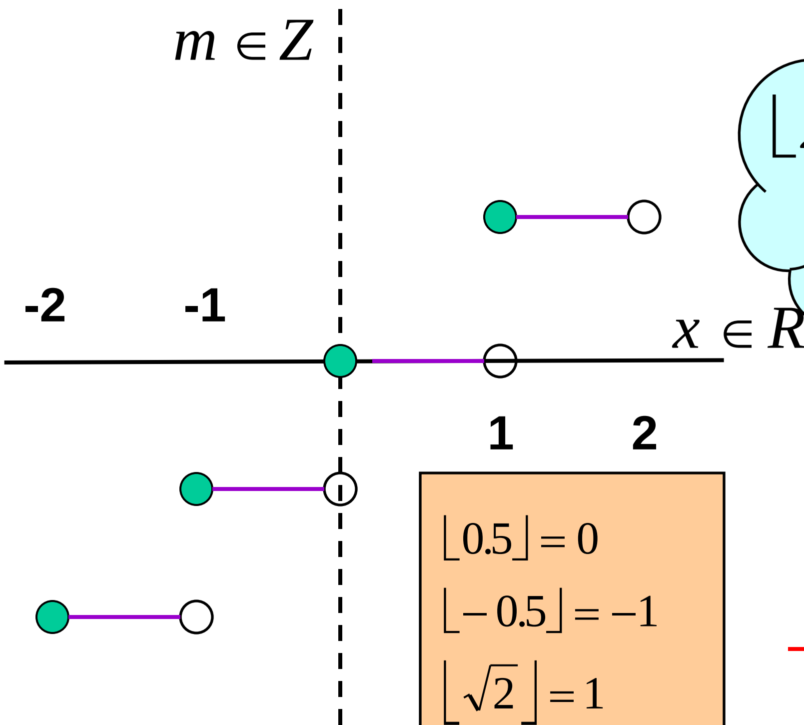
$R \in P(R^2)$ $\quad$ $Z \in P(Z^2)$ $\quad$ $Z^+ \times Z^+ \in P(Z^2)$

Floor & ceiling

floor $\lfloor\ \rfloor : R \rightarrow Z$

ceiling $\lceil\ \rceil : R \rightarrow Z$

$$\lfloor x \rfloor + 1 = \lceil x \rceil ?$$



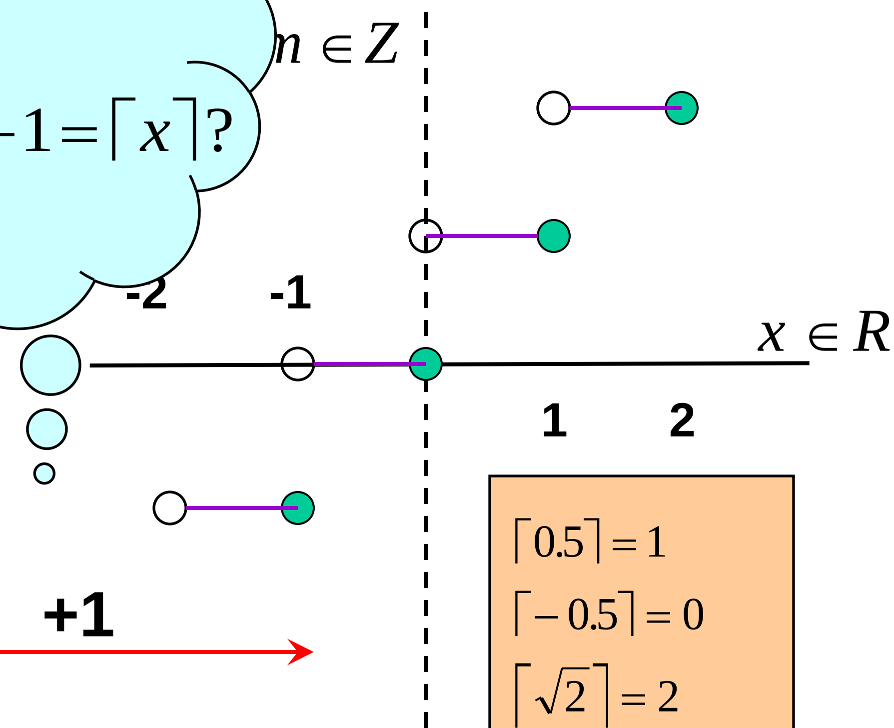
$\lfloor 0.5 \rfloor = 0$

$\lfloor -0.5 \rfloor = -1$

$\lfloor \sqrt{2} \rfloor = 1$

$\lfloor -1.001 \rfloor = -2$

+1



$\lceil 0.5 \rceil = 1$

$\lceil -0.5 \rceil = 0$

$\lceil \sqrt{2} \rceil = 2$

$\lceil -1.001 \rceil = -1$

$$\lfloor x \rfloor = m ; \boxed{m \le x} ; n \le x \rightarrow n \le m$$

$$\lceil x \rceil = m ; \boxed{m \ge x} ; n \ge x \rightarrow n \ge m$$

Functions-II — Sequences definition

A sequence is defined as a map that assigns a numerical value to an integer variable as follows:

$$s: A \to B\,;\; A \subseteq Z^{+} \cup \{0\},\, B \subseteq R$$

$$n \mapsto s(n) = s_n$$

general term of index ***n***

$$ran(s) = s(A) \neq \{s_n\}$$

the range of ***s*** is *not* the sequence of terms

$$A = N\,;\; a_n = 1 + (-1)^n \longrightarrow \{a_n\}_{n=0}^{\infty} = \{2,0,2,0,2,0,\ldots,\}$$

$$A = N\,;\; b_k = 2^k \longrightarrow \{b_k\}_{k=0}^{\infty} = \{1,2,4,8,16,32,\ldots,\}$$

$$A = N\,;\; c_m = m! \longrightarrow \{c_m\}_{m=0}^{\infty} = \{1,1,2,6,24,120,720,\ldots,\}$$

$$A = Z^{+}\,;\; a_j = \frac{1}{j} \longrightarrow \{a_j\}_{j=1}^{\infty} = \{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \ldots,\}$$

Sequences examples

Two important general sequences:

- arithmetic progression

first element   difference between terms

$$\boxed{s_n = a + nd\ ;\ d \neq 0}$$

$$s_{n+1} = a + (n+1)d = a + nd + d$$

$$s_{n+1} = s_n + d \ \rightarrow \boxed{d = s_{n+1} - s_n}$$

- geometric progression

first element   ratio between terms

$$\boxed{g_n = ar^n\ ;\ a \neq 0 \wedge r \neq 1}$$

$$g_{n+1} = ar^{n+1} = ar^n r$$

$$g_{n+1} = g_n r \ \rightarrow \boxed{r = g_{n+1} / g_n}$$

Summation definition

Summation notation is used to represent the sum of a finite number of terms in a given sequence.

upper limit

general term

$$\{a_j\}_{j=m}^{n} \rightarrow \boxed{\sum_{j=m}^{n} a_j = a_m + a_{m+1} + \cdots + a_n}$$

lower limit

index of summation

The choice of letter for the index of summation is *arbitrary*,

$$\sum_{j=m}^{n} a_j = \boxed{\sum_{i=m}^{n} a_i = \sum_{k=m}^{n} a_k}$$

Sometimes it is useful to sum a finite set of values obtained from a function, in that case, we use the following notation:

value of function at ***x***

$$\boxed{\sum_{x \in S} f(x)}$$

S is the indexing set

Summation basic formulas

$$\sum_{j=0}^{n} ca_j = c\sum_{j=0}^{n} a_j \;;\; c \in R$$

A constant value can be pulled out from the summation symbol.

$$\sum_{j=0}^{n} (a_j + b_j) = \sum_{j=0}^{n} a_j + \sum_{j=0}^{n} b_j$$

We can split the summation symbol for the sum of two finite sequences.

$$\sum_{j=m}^{n} b_{j+p} = \sum_{k=m+p}^{n+p} b_k \;;\; k = j + p \longleftarrow$$

An index can be shifted by ***p***, so the limits of summation change also.

$$\boxed{\sum_{j=1}^{n} c = nc}$$

$$\boxed{\sum_{k=0}^{n} c = (n+1)c}$$

The sum of an arithmetic progression:

$$\boxed{\sum_{j=0}^{n} s_j = \sum_{j=0}^{n}(a+jd)} = \sum_{j=0}^{n} a + \sum_{j=0}^{n} jd$$

$$= (n+1)a + d\sum_{j=0}^{n} j = (n+1)a + d\sum_{j=1}^{n} j$$

$$= (n+1)a + d\left[\frac{n(n+1)}{2}\right]$$

$$\boxed{\sum_{j=0}^{n}(a+jd) = (n+1)\left[a + \frac{d}{2}n\right]}$$

$$\boxed{\sum_{j=1}^{n} j = S = 1 + 2 + \cdots + n}$$

$$\sum_{j=n}^{1} j = S = n + (n-1) + \cdots + 1$$

$$2S = \underbrace{(n+1) + (n+1) + \cdots + (n+1)}_{n\ times}$$

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

Summation examples[b]

- List the first 10 terms of the sequence whose ***n***-th term is the sum of the first ***n*** positive integers.

$$\{s_n\}_{n=1}^{10} = \{\sum_{k=1}^{n} k\}_{n=1}^{10} = \left\{\frac{n(n+1)}{2}\right\}_{n=1}^{10} = \{\tfrac{1\cdot 2}{2}, \tfrac{2\cdot 3}{2}, \tfrac{3\cdot 4}{2}, \tfrac{4\cdot 5}{2}, \tfrac{5\cdot 6}{2}, \tfrac{6\cdot 7}{2}, \tfrac{7\cdot 8}{2}, \tfrac{8\cdot 9}{2}, \tfrac{9\cdot 10}{2}, \tfrac{10\cdot 11}{2}\}$$

$$= \boxed{\{1,3,6,10,15,21,28,36,45,55\}}$$

- Find the following sum:

$$\sum_{k=99}^{200} k^3 = \sum_{k=1}^{200} k^3 - \sum_{k=1}^{98} k^3 = \left[\frac{n(n+1)}{2}\right]_{n=200}^2 - \left[\frac{n(n+1)}{2}\right]_{n=98}^2$$

$$= \left[\frac{200(200+1)}{2}\right]^2 - \left[\frac{98(98+1)}{2}\right]^2 = (100\cdot 201)^2 - (49\cdot 99)^2 = \boxed{380{,}477{,}799}$$

- What is the value of the following product:

$$\prod_{i=1}^{100} (-1)^i = \prod_{i \text{ even}} (-1)^i \prod_{i \text{ odd}} (-1)^i = \prod_{i \text{ odd}} (-1)^i = (-1)^1(-1)^3 \underbrace{\cdots}_{50 \text{ times}} (-1)^{99} = \boxed{1}$$

Other examples[a]

- If $a_n = \left\lfloor \sqrt{2n} + \frac{1}{2} \right\rfloor$ find $\{a_n\}_{n=1}^{21}$

$a_1 = \left\lfloor \sqrt{2.1} + 1/2 \right\rfloor$, $a_2 = \left\lfloor \sqrt{2.2} + 1/2 \right\rfloor$, $a_3 = \left\lfloor \sqrt{2.3} + 1/2 \right\rfloor$ ....

$a_1 = 1$, $a_2 = 2$, $a_3 = 2$,…

The sequence is {1,2,2,3,3,3,4,4,4,4,5,5,5,5,5,6,6,6,6,6,6}

- Evaluate $\sum_{i=0}^{2}\sum_{j=0}^{3} i^2 j^3$

$$\sum_{i=0}^{2}\sum_{j=0}^{3} i^2 j^3 = (0^2 . 0^3) + (0^2.1^3) + (0^2.2^3) + (0^2.3^3) + (1^2.0^3) + (1^2.1^3)$$
$$(1^2.2^3) + (1^2.3^3) + (2^2.0^3) + (2^2.1^3) + (2^2.2^3) + (2^2.3^3)$$
$$= 0 + 0 + 0 + 0 + 0 + 1 + 8 + 27 + 0 + 4 + 32 + 108$$
$$= 180$$

- Show that $\sum_{j=1}^{n}(a_j - a_{j-1}) = a_n - a_0$ (telescoping)

$$\sum_{j=1}^{n}(a_j - a_{j-1}) = a_n - a_0 = (a_1 - a_0) + (a_2 - a_1) + (a_3 - a_2) + \dots + (a_{n-1} - a_{n-2}) + (a_n - a_{n-1})$$

The first value of each term when added to the second value of the next term equals 0. Hence, the final sum equals $a_n - a_0$.

- Using the fact that $\frac{1}{k(k+1)} = \frac{1}{k} - ?$ compute $\sum_{k=1}^{n}\frac{1}{k(k+1)}$

$$\sum_{k=1}^{n}\frac{1}{k(k+1)} = \sum_{k=1}^{n}\left(\frac{1}{k} - \frac{1}{k+1}\right)$$ since $$\boxed{\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1}}$$

$$= \left(\frac{1}{1} - \frac{1}{2}\right) + \left(\frac{1}{2} - \frac{1}{3}\right) + \dots + \left(\frac{1}{n} - \frac{1}{n+1}\right)$$

$$= 1 - \frac{1}{n+1}$$

Functions: Part III

- Cardinality
- Growth of functions
- Big-O concept
- Algebraic operations
- Big-Ω and big-Θ

Functions-III Cardinality

<u>Reminder</u>: the cardinal of a set ***A*** = number of elements in ***A***; it can be interpreted as the mapping from *P*(***A***) to the set of natural numbers ***N*** = {0,1,2,…}.

aleph 0

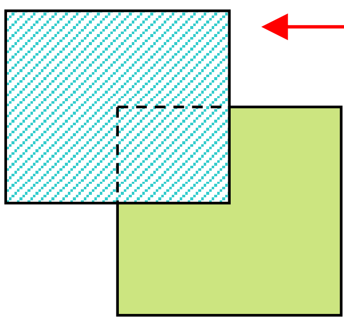
$$\boxed{|\,|: P(A) \to N}$$

$|\varnothing| = 0$ — the <u>null set</u> has no elements.

$|\{b_1, \ldots, b_m\}| = m$ — a <u>finite set</u> with ***m*** elements.

$|N| = \infty = \aleph_0$ — an <u>infinite set</u>; first transfinite number

- the principle of inclusion-exclusion; for finite sets ***A*** and ***B***: $\boxed{|A \cup B| = |A| + |B| - |A \cap B|}$



$|A|$ includes the elements of $A \cap B$

$|B|$ *includes again* the elements of $A \cap B$ so, we have to exclude them.

- note that if ***A*** and ***B*** are disjoint then: $\boxed{|A \cup B| = |A| + |B|}$

- two sets ***A*** and ***B*** have the same cardinality if a bijection can be established between them.

Countable sets

A countable set ***X*** is defined as a set that is finite or has the same cardinality of ***N***; otherwise we say that ***X*** is an uncountable set.

Set	**Bijection ?**	**Countable?**
$X = \{0,2,4,6,\ldots\}$	$n \mapsto 2n$	Yes
$Y = \{1,3,7,9,\ldots\}$	$n \mapsto 2n+1$	Yes
$[0,1] = \{x \in R \mid 0 \le x \le 1\}$	no bijection exists	No
$\{s_n\} = \{s_0, s_1, \ldots\}$	$n \mapsto s_n$	Yes

In other words, if the elements of set ***X*** can be listed in sequence, the set is countable. The real numbers are not countable because for a given ***x*** *we do not know what the next number is* (it is not ***x*** + 1 nor ***x*** + 0.001, etc.).

Consequently, ***R*** is a **"bigger set"** than ***N***, its cardinal is greater than aleph 0.

$$\boxed{|N| = \aleph_0 < \aleph_1 = |R|}$$ (second transfinite number)

Functions-III — Growth motivation

Suppose that <u>problem</u> ***P*** admits a computational solution; consider also that ***P*** is solved by means of three <u>algorithms</u> $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, and $\boldsymbol{A}_3$.

Common question: which of the algorithms is better?

In discrete mathematics, “better” means to establish a quantitative relation for algorithm $\boldsymbol{A}_i$ as <u>function of a parameter</u> ***n***. The <u>function value</u> $f(n)$ is usually interpreted as the amount of time required to solve problem ***P***.

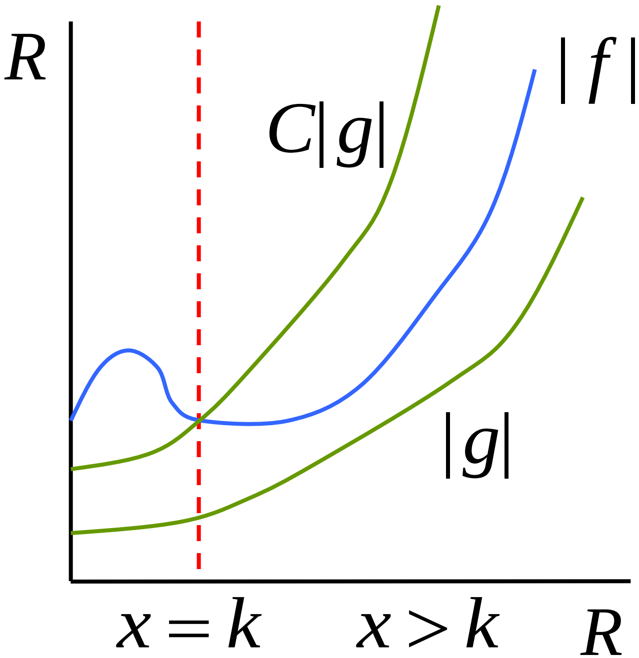
$$P \begin{cases} (A_1, f(n)) \\ (A_2, g(n)) \\ (A_3, h(n)) \end{cases}$$

A comparison must be established between functions ***f***, ***g***, and ***h*** in order to decide which algorithm is the best.

Functions-III Growth big-O

Function f is big-O of g if and only if $\exists C,k>0\,\forall x>k\,;|f(x)|\leq C|g(x)|$

test function f → $\boxed{f \approx O(g)}$ ← reference function g



$$f(x)=ax^2+bx+c\,;a,b,c\in R$$

$$x>1\rightarrow|f(x)|=|ax^2+bx+c|$$

$$\leq|a|x^2+|b|x+|c|$$

$$=x^2(|a|+\frac{|b|}{x}+\frac{|c|}{x^2})$$

$$\leq x^2(|a|+|b|+|c|)=C|x^2|$$

So in this example, $C=|a|+|b|+|c|\wedge k=1$

$$\boxed{ax^2+bx+c\approx O(x^2)}$$

- f grows almost as fast as g
- f behaves almost as g
- g is a simple upper bound of f

Functions-III Growth big-*O* example

- consider $f(n) = 1 + \cdots + n$

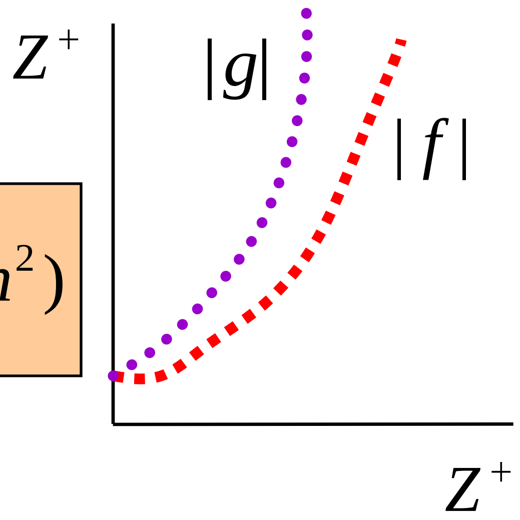
$$1 + \cdots + n \le n + \cdots + n$$

$$= n^2 = g(n)$$

Thus, taking $C = 1, k = 0$

$$\forall n > k, |f(n)| \le C|g(n)|$$

$$\boxed{\sum_{i=1}^{n} i \approx O(n^2)}$$



- consider $f(n) = n!$

$$n! = 1.2\ldots n \le n.n\ldots n = n^n = g(n) \longrightarrow \boxed{n! \approx O(n^n)}$$

- from this result we obtain another example,

$$f(n) = \log n! \le n \log n = g(n) \longrightarrow \boxed{\log n! \approx O(n \log n)}$$

Functions-III Growth operations

$$\boxed{f_1, f_2 \approx O(g) \rightarrow f_1 + f_2 \approx O(g)}$$

By definition, $f_1 \approx O(g) \leftrightarrow \exists C_1, k_1 \forall x > k_1, |f_1(x)| \leq C_1 |g(x)|$

$$f_2 \approx O(g) \leftrightarrow \exists C_2, k_2 \forall x > k_2, |f_2(x)| \leq C_2 |g(x)|$$

We have to prove that, $\exists C, k \,\forall x > k, |(f_1 + f_2)(x)| \leq C|g(x)|$ ←

$$C = ?$$
$$k = ?$$

$$|(f_1 + f_2)(x)| = |f_1(x) + f_2(x)|$$ triangle inequality,

$$\leq |f_1(x)| + |f_2(x)| \leq C_1 |g(x)| + C_2 |g(x)|$$

$$\leq (C_1 + C_2)|g(x)| \; ; (x > k_1) \wedge (x > k_2)$$ →

$$C = C_1 + C_2$$
$$k = \max(k_1, k_2)$$

Similarly, $$\boxed{f_1, f_2 \approx O(g) \rightarrow f_1 \cdot f_2 \approx O(g)}$$

Functions-III Growth operations example

$$f_1 \approx O(g_1) \wedge f_2 \approx O(g_2) \rightarrow f_1 + f_2 \approx O(\max\{|g_1|,|g_2|\})$$

$$f_1 \approx O(g_1) \wedge f_2 \approx O(g_2) \rightarrow f_1 \cdot f_2 \approx O(g_1 \cdot g_2)$$

Example: $(n! + 2^n)(n^3 + \log(n^2 + 1))$

$(n! + 2^n) \approx O(n!)$ since $2^n \leq n!\ \forall n > 3$

For the second factor, $(n^2 + 1) \approx O(n^2) \rightarrow \log(n^2) = 2\log n$ therefore,

$(n^3 + \log(n^2 + 1)) \approx O(n^3)$ since $2\log n \leq n^3\ \forall n > 0$

Using the second law for the product of two functions, the estimate is:

$$(n! + 2^n)(n^3 + \log(n^2 + 1)) \approx O(n!n^3)$$

Functions-III Growth big-Ω & big-Θ

Function ***f*** is big-Ω of ***g***

$$\boxed{f \approx \Omega(g) \Leftrightarrow g \approx O(f)}$$

Since ***g*** is a lower bound for ***f***, then ***f*** is an upper bound of ***g***.

As an example, consider again,

$$f(n) = 1 + \cdots + n \ge \lceil n/2 \rceil + (\lceil n/2 \rceil + 1) + \cdots + n$$

$$\ge \lceil n/2 \rceil + \lceil n/2 \rceil + \cdots + \lceil n/2 \rceil$$

$$\ge (n - \lceil n/2 \rceil + 1) \lceil n/2 \rceil \ge (n/2)(n/2) = n^2/4 = g(n)$$

Function ***f*** is big-Θ of ***g***

$$\boxed{f \approx \Theta(g) \Leftrightarrow [f \approx O(g)] \wedge [f \approx \Omega(g)]}$$

upper estimate

$$\left.\begin{array}{l}(1 + \cdots + n) \approx O(n^2) \\ (1 + \cdots + n) \approx \Omega(n^2)\end{array}\right\} \rightarrow \boxed{\sum_{i=1}^{n} i \approx \Theta(n^2)}$$

$$\tfrac{1}{4} n^2 \le \boxed{\tfrac{1}{2} n(n+1)} \le n^2$$

lower estimate

exact formula for the sum of the first ***n*** integers.

- Show that $x^3 \approx O(x^4)$ but $x^4 \text{ is not } O(x^3)$

Since $x^3 \leq x^4$ for all $x > 1$, we know that x^3 is $O(x^4)$. On the other hand, if $x^4 \leq Cx^3$, then (dividing by x^3) $x \leq C$. Since this latter condition cannot hold for all large x, no matter what the value of the constant C could be, we conclude that x^4 is not $O(x^3)$.

- Give a proof for: $\sum_{i=1}^{n} i^k \approx O(n^{k+1})\,;\, k \in Z^+$

$$\sum_{i=1}^{n} i^k = 1^k + 2^k + 3^k + \cdots + n^k$$

$$\leq n^k + n^k + n^k + \cdots + n^k$$

$$= n \cdot n^k = n^{k+1}$$

Hence, the given expression is $O(n^{k+1})$.

- Show that $f(x) \approx O(\log_b x) \rightarrow f(x) \approx O(\log_a x)\,;\, a,b > 1$

Assume that the corresponding base of each logarithm is greater than 1, then using the definition of big-*O*, we can write

$$\exists C,k > 0\ \forall x > k, |\ f(x)| \leq C\,|\log_b x|$$

from general algebra, the relation between the two logarithmic functions, is given by,

$$\log_a x = \log_b x / \log_b a \quad \Rightarrow \log_b x = \log_b a \cdot \log_a x = C_l \log_a x$$

therefore, $\exists C^*, k > 0\ \forall x > k, |\ f(x)| \leq C^*\,|\log_a x|$

Where *k* is the same though $C^* = C \cdot C_l = C \cdot \log_b a$

- Show that if ***f(x)*** and ***g(x)*** are real functions of ***x***, then ***f*** is big-*O* of ***g*** if and only if ***g*** is big-Ω of ***f***.

$$f \approx O(g) \Leftrightarrow \exists C,k \ \forall x > k, |f(x)| \le C|g(x)|$$

we can change the sense of the inequality, so

$$|g(x)| \ge C^{-1}|f(x)| = \boxed{C^*|f(x)|}$$

therefore,

$$\exists C',k \ \forall x > k, |g(x)| \ge C^*|f(x)| \Leftrightarrow g \approx \Omega(f)$$

- Explain what it means for a function to be Θ(1); remember that,

$$f \approx \Theta(g) \Leftrightarrow C_1|g| \le |f| \le C_2|g|$$

hence, if $\boxed{g(x) = 1}$

$$f \approx \Theta(1) \Leftrightarrow C_1 \le |f| \le C_2, \forall x > k$$

for values of ***x*** greater than ***k***, the function ***f*** will be bounded by the horizontal lines ***y*** = ***C***$_1$ and ***y*** = ***C***$_2$.

- Show that $(x^2 + xy + x\log y)^3 \approx O(x^6y^3)$

$$(x^2y^0 + xy \le x^2y) \wedge (x\log y \le xy) \rightarrow x^2y + xy \le x^2y$$

thus, $\boxed{(x^2y)^3 = x^6y^3, \forall x > 1, y > 1}$

Algorithms

- Algorithms and pseudocode
- Computational complexity
- Terminology
- Time estimation

An algorithm is a finite set of precise instructions for performing a computation or for solving a problem, generally, by means of a computing device.

An algorithm must have the following characteristics:

- Input (from a domain set) and output (the range set according to the input).
- Definiteness, each step must be *defined precisely.*
- Correctness, the output values *should be meaningful for a given input*.
- Finiteness, the output is reached after a *finite number of steps* (for any input).
- Effectiveness, each step must be performed in a *finite amount of time*.
- Generality, it must be applicable to a *class of related problems* and not only for particular inputs.

A pseudocode language is used to describe an algorithm in a generic way, independently from a specific *machine architecture* or *programming context*.

procedure $swap(x, y \in R)$

$z := x$

$x := y$

$y := z$

procedure $insert(x, a_1, \ldots, a_n \in Z)$

$\{a_1 \leq \cdots \leq a_n\}$

$a_{n+1} := 0$

$i := 1$

while $x > a_i$

$\quad i := i + 1$

$\quad$ **for** $j := 0$ **to** $n - i$

$\quad\quad a_{n-j+1} := a_{n-j}$

$a_i := x$

procedure $smallest(a_1, \ldots, a_n \in Z)$

$small := a_1$

for $i := 2$ **to** n

$\quad$ **if** $small > a_i$ **then** $small := a_i$

Algorithm: linear search

procedure $ls(x, a_1, \ldots, a_n \in Z)$

$i := 1$

while $(i \leq n \wedge x \neq a_i)$

  $i := i + 1$

if $i \leq n$

  then $location := i$

  else $location := 0$

$\{10,12,14,15\}; x = 14 \rightarrow n = 4$

i	$(i \leq n \wedge x \neq a_i)$	i	$i \leq n$	$location$
1	$(1 \leq 4 \wedge 14 \neq 10)$	2		
	$(2 \leq 4 \wedge 14 \neq 12)$	3		
	$(3 \leq 4 \wedge 14 = 14)$	3	$3 \leq 4$	3

$\{10,12,14,15\}; x = 11 \rightarrow n = 4$

i	$(i \leq n \wedge x \neq a_i)$	i	$i \leq n$	$location$
1	$(1 \leq 4 \wedge 11 \neq 10)$	2		
	$(2 \leq 4 \wedge 11 \neq 12)$	3		
	$(3 \leq 4 \wedge 11 \neq 14)$	4		
	$(4 \leq 4 \wedge 11 \neq 15)$	5	$5 \leq 4$	0

Algorithm: binary search

procedure $bs(x, a_1, \ldots, a_n \in Z)$

$\{a_1 \le \cdots \le a_n\}$

$i := 1$ {left end point}

$j := n$ {right end point}

while $i < j$

$m := \lfloor (i + j)/2 \rfloor$

if $x > a_m$

then $i := m + 1$

else $j := m$

if $x = a_i$

then $location := i$

else $location := 0$

$$\boxed{a_n = 7 + 3n\,;\, n = 1, \ldots, 16}$$

$$\boxed{\{10, 13, 16, \ldots, 55\}\,;\, x = 52 \to n = 16}$$

$i < j$	m	$x > a_m$	i	j
$1 < 16$	8	$52 > 31$	9	16
$9 < 16$	12	$52 > 43$	12	16
$12 < 16$	14	$52 > 49$	14	16
$14 < 16$	15	$52 > 52$	14	15
$14 < 15$	14	$52 > 49$	15	15

$$52 = a_{15} = 7 + 3(15) \to location = \boxed{15}$$

Complexity basic ideas

The computational complexity of an algorithm is defined as a quantitative measure of its performance when producing an output for a given input of size ***n***.

Computational complexity

- Time complexity: number of operations required for a given ***n***.
- Space complexity: memory required for a given ***n***.

Procedure	**# of comparisons / time complexity**
• smallest	$2n-1 \approx O(n)$
• linear search	$2n+2 \approx \boxed{O(n)}$
• binary search	$2\log n+2 \approx \boxed{O(\log n)}$

Algorithm: binary search

procedure $bs(x, a_1, \ldots, a_n \in Z)$

$\{a_1 \leq \cdots \leq a_n\}$

$i := 1$ {left end point}

$j := n$ {right end point}

while $i < j$

$m := \lfloor (i + j)/2 \rfloor$

if $x > a_m$

then $i := m + 1$

else $j := m$

if $x = a_i$

then $location := i$

else $location := 0$

Assume that the number of elements in the list is a power of **2** (remember the example):

$$\boxed{n = 2^k \rightarrow k = \log n}$$

During execution of the **while** loop <u>two comparisons</u> are made, one for testing the exit, the other for testing ***x***.

Each step within the loop <u>reduces the search interval by half</u>; this is done ***k*** times.

<u>One comparison</u> is realized when exiting the **while** loop and <u>another one</u> for testing if ***x*** was found in the list.

The total is: $\boxed{2 \cdot k + 2 = 2\log n + 2}$

Algorithms — Complexity terminology

Description	Complexity	Type of Problem	
constant	$O(1)$	tractable	SOLVABLE
logarithmic	$O(\log n)$		
linear	$O(n)$		
linear-log	$O(n \log n)$		
polynomial	$O(n^b)$		
exponential	$O(b^n)\,;\, b > 1$	untractable	
factorial	$O(n!)$		

There is no algorithm that can tell if given another program with its input, the program will halt or not (Alan Turing famous Halting Problem).

UNSOLVABLE

Just to have an idea of the amount of time needed for solving a problem with a certain time complexity if one bit operation takes 1 nanosecond.

1 nanosecond = 1 ns = 10^{-9} seconds

Input size / Bit operations used

n	$\log n$	n	$n \log n$	n^2	2^n
10	3	10	30	100	$1\ \mu s$
10^2	7	100	700	$10\ \mu s$	$4 \cdot 10^{13}$ yr
10^6	20	$100\ \mu s$	20 ms	17 min	$> 10^{100}$ yr

- Describe an algorithm that uses only assignments statements that replaces the triple ***(x,y,z)*** with ***(y,z,x).*** What is the minimum number of assignment statements needed?

$$w := x\,;\; x := y\,;\; y := z\,;\; z := w$$

w is a buffer variable, **4** assignments.

- Describe an algorithm that determines whether a function from a finite set to another finite set is one-to-one (an injection).

$$Dom(f) = A = \{a_1, \ldots, a_n\}$$

$$b := 1$$

$$i \in 1, \ldots, n$$

$$\quad j \in 1, \ldots, n$$

$$(i < j) \longrightarrow i \neq j \to f(a_i) = f(a_j)\,?$$

$$\qquad T \to b := 0 \wedge \text{exit}$$

$$b = 1?$$

$$\quad T \to f \text{ injective}$$

$$\quad F \to f \text{ not injective}$$

- How much time does an algorithm take to solve a problem of size ***n*** if this algorithm uses $\mathbf{2}\boldsymbol{n}^{\mathbf{2}} + \mathbf{2}^{\mathbf{n}}$ bit operations, each requiring **1** ns, with the following values of ***n***?

$$n = 10 \to 2(10)^2 + 2^{10} = 200 + 1024 = 1224$$

$$\Rightarrow 1224 \times 1 \text{ ns} = \boxed{1.224\ \mu s}$$

$$n = 20 \to 2(20)^2 + 2^{20} = 800 + 1024^2 = 1049376$$

$$\Rightarrow 1049376 \times 1 \text{ ns} = \boxed{1.049376 \text{ ms}}$$

- Devise an algorithm to compute x^n, where x is a real number and n is an integer. (*Hint*: First give a procedure for computing x^n when n is nonnegative by succesive multiplication by x, starting with 1. Then extend this procedure, and use the fact that $x^{-n} = 1 / x^n$ to compute x^n when n is negative.

$$\textbf{procedure } power(x \in R, n \in Z)$$
$$m := \text{abs}(n)$$
$$p := 1$$
$$\textbf{for } i := 1 \textbf{ to } m$$
$$\quad p := p \cdot x$$
$$\textbf{if } n < 0$$
$$\quad p := 1/p$$
$$\{p = x^n\}$$

# of arithmetical operations

best	worst	average (any)
$n > 0$	$n < 0$	$n \in Z$
n	$n+1$	$\frac{n+(n+1)}{2} = n + \frac{1}{2}$

best, worst, average $\approx O(n)$

- Describe an algorithm for finding the smallest integer in a finite sequence of natural numbers.

$$\textbf{procedure } smallest(a_1, \ldots, a_n \in N)$$
$$small := a_1$$
$$\textbf{for } i := 2 \textbf{ to } n$$
$$\quad \textbf{if } small > a_i \textbf{ then } small := a_i$$
$$\{small = \min(a_1, \ldots, a_n)\}$$

$$\textbf{procedure } smallest(a_1, \ldots, a_n \in N)$$
$$small := a_1$$
$$i := 2$$
$$\textbf{while } i \le n$$
$$\quad \textbf{if } small > a_i \textbf{ then } small := a_i$$
$$\quad i := i + 1$$

In this example there is no distinction between the best, worst, and average analysis since all elements in the sequence must be scanned.

\# of comparisons within the loop:

$$2[(n-2)+1] = 2(n-1)$$

\# of comparisons outside the loop:

$$1 \text{ when } i > n$$

$$\boxed{\text{total} = 2(n-1)+1 = 2n-1 \approx O(n)}$$

- Devise an algorithm that finds all terms of a finite sequence of integers that are greater than the sum of all previous terms of the sequence.

$\textbf{procedure } \mathit{findterms}(a_1, \ldots, a_n \in Z)$

{use boolean vector b_i

for accumulating terms}

$\textbf{for } i := 1 \textbf{ to } n$

  $b_i := 0$

  $s := 0$

  $\textbf{for } j := 1 \textbf{ to } i-1$

    $s := s + a_j$

  $\textbf{if } a_i > s \textbf{ then } b_i := 1$

{b is a binary vector showing

positions where $a_i >$ spt}

Since there are two nested loops, after counting the number of additions and comparisons it results that,

$\text{time complexity} \approx O(n^2)$

A better algorithm:

$\textbf{procedure } \mathit{findterms}(a_1, \ldots, a_n \in Z)$

$s := 0$

$\textbf{for } i := 1 \textbf{ to } n$

  $\textbf{if } a_i > s \textbf{ then } b_i := a_i$

    $\textbf{else } b_i := 0$

  $s := s + a_i$

Here, $\text{time complexity} \approx O(n)$

- Analyze the *average-case performance* of the linear search algorithm, if exactly half the time element ***x*** is not in the list and if ***x*** in the list is equally likely to be in any position.

Algorithm: linear search

$$\begin{aligned}
&\textbf{procedure } ls(x, a_1, \ldots, a_n \in Z)\\
&i := 1\\
&\textbf{while } (i \le n \wedge x \ne a_i)\\
&\quad i := i + 1\\
&\textbf{if } i \le n\\
&\quad \textbf{then } location := i\\
&\quad \textbf{else } location := 0
\end{aligned}$$

Case 1: when ***x*** is not in the list

$$2n\,(\text{within}) \;+1\,(\text{exit}) \;+1\,(\text{check location})$$

$$2n+2$$

Case 2: when ***x*** is in the list

$$x = a_i \rightarrow 2i+1\ (\text{comparisons})$$

$$\Rightarrow \sum_{i=1}^{n}(2i+1) = 2\sum_{i=1}^{n} i + \sum_{i=1}^{n} 1 = 2\cdot\frac{n(n+1)}{2} + n$$

$$= n(n+2) \Rightarrow \text{avg} = n+2$$

Total average # of ops. $= \dfrac{\text{case1}+\text{case2}}{2} = \dfrac{(2n+2)+(n+2)}{2} = \boxed{\dfrac{3n+4}{2} \approx O(n)}$

Number theory

- Integer division
- Prime numbers
- The division algorithm
- Modular arithmetic
- Random numbers
- The Euclidean algorithm
- Base-*b* representation
- Binary integer operations

Division between <u>real numbers</u>: "***a*** is divided by ***b***"

$$\boxed{/ : R \times R_0 \to R\,;\,(a,b) \mapsto a/b\,;\, b \neq 0, R_0 = R - \{0\}}$$

Division between <u>integer numbers</u>: "***a*** divides ***b***"

$$\boxed{| : Z_0 \times Z \to Z\,;\,(a,b) \mapsto a|b\,;\, a \neq 0, Z_0 = Z - \{0\}}$$

$$\boxed{a|b \Leftrightarrow \exists c \in Z, b = ca}$$

we say that ***a*** is a factor of ***b*** or ***b*** is a multiple of ***a***

Given two positive integers ***n*** > ***d***, the number of integers divisible by ***d*** not exceeding ***n*** is $\lfloor n/d \rfloor$

<u>Proof</u>: ***d*** is a divisor of all numbers of the form $kd, k \in Z_0^+$ therefore,

$$0 < kd \le n \to 0 < k \le n/d \quad \text{or} \quad k = \lfloor n/d \rfloor$$

A basic set of properties for divisibility: let ***a, b, c*** be integer numbers,

$$a|b \wedge a|c \rightarrow a|(b+c)$$

$$a|b \Leftrightarrow \exists k_1 \in Z, b = k_1 a$$

$$a|c \Leftrightarrow \exists k_2 \in Z, c = k_2 a$$

$$\boxed{\Rightarrow b+c = (k_1+k_2)a = ka \Rightarrow a|(b+c)}$$

$$a|b \rightarrow \forall c \in Z, a|bc$$

$$a|b \Leftrightarrow \exists k_1 \in Z, b = k_1 a$$

$$\boxed{\Rightarrow bc = (ck_1)a = ka \Rightarrow a|bc}$$

$$a|b \wedge b|c \rightarrow a|c$$

$$a|b \Leftrightarrow \exists k_1 \in Z, b = k_1 a$$

$$b|c \Leftrightarrow \exists k_2 \in Z, c = k_2 b$$

$$\boxed{\Rightarrow c = k_2 b = k_2(k_1 a) = ka \Rightarrow a|c}$$

A prime number (in the sense of primitive or primary) is a *positive integer* ***p*** whose only divisors are itself, ***p*** and **1**. If a number ***n*** is not prime it is called composite.

$$\boxed{P = \{2,3,5,7,11,13,\ldots,2^{3021377}-1,\ldots\} \subset Z^+}$$

There is no formula that generates all prime numbers; supercomputers are used to search for huge prime numbers, the prime shown in the list has **909,256** digits, it is of the form,

$$\boxed{2^p - 1,\ p \in P}$$

The fundamental theorem of arithmetic:

$$\boxed{\begin{aligned}\forall n \in Z^+, n &= \prod_{i=1}^{m} p_i^{e_i} \\ &= p_1^{e_1} \cdot p_2^{e_2} \cdots p_m^{e_m}\end{aligned}}$$

$$60 = 2\cdot 2\cdot 3\cdot 5 = \boxed{2^2\cdot 3\cdot 5}$$

$$210 = \boxed{2\cdot 3\cdot 5\cdot 7}$$

$$12{,}115{,}103 = \boxed{7^3\cdot 11\cdot 13^2\cdot 19}$$

$$n \notin P \to n = ab, a,b > 1 \Rightarrow \boxed{a \leq \sqrt{n} \vee b \leq \sqrt{n}}$$ otherwise,

$$\neg(a \leq \sqrt{n} \vee b \leq \sqrt{n}) \Leftrightarrow a > \sqrt{n} \wedge b > \sqrt{n} \quad \Rightarrow \underline{ab = \sqrt{n}\sqrt{n} > n}$$

Thus, ***a*** or ***b*** is a factor of ***n*** less than its square root, and it can be prime or by the fundamental theorem of arithmetic it has a prime factor. In either case,

$$\boxed{\exists p \in P, p|n \wedge p \leq \sqrt{n}}$$

A consequence of this result: if there are *no primes **p** dividing **n*** but are less than the square root of ***n***, then ***n*** is a <u>prime number</u>.

$$n = 131 \to \left\lfloor \sqrt{131} \right\rfloor = 11$$

since, **2,3,5,7,11≤11** do not divide **131**, then **131** is a *prime number*.

$$\boxed{a \in Z \wedge d \in Z^+ \rightarrow \exists! q, r\, ; (0 \le r < d) \wedge (a = dq + r)}$$

dividend divisor remainder quotient

$$67 = 7 \cdot 9 + 4\, ; 0 \le 4 < 7\, ; \qquad (a, d, r, q) = (67, 7, 4, 9)$$

$$-67 = 7 \cdot (-10) + 3\, ; 0 \le 3 < 7\, ; (a, d, r, q) = (-67, 7, 3, -10)$$

$$-67 \ne 7 \cdot (-9) + (-4)\, ; -4 < 0!$$

As computer integer operators,

$$\boxed{\begin{array}{c} q = a \operatorname{div} d \\ \\ r = a \bmod d \end{array}}$$

$\longrightarrow$ $67 \operatorname{div} 7 = 9$

$\longrightarrow$ $-67 \bmod 7 = 3$

Greatest common divisor of two integer numbers ***a***,***b***:

$$\boxed{D=\{d:d|a\wedge d|b\}\rightarrow \gcd(a,b)=\max D}$$

$$D=\{1,2,3,4,6,12\}\rightarrow \gcd(24,36)=12$$

Least common multiple of two integer numbers ***a***,***b***:

$$\boxed{M=\{m:a|m\wedge b|m\}\rightarrow \operatorname{lcm}(a,b)=\min M}$$

$$M=\{72,144,216,\ldots\}\rightarrow \operatorname{lcm}(24,36)=72$$

$$\boxed{a=\prod_{i=1}^{m}p_i^{a_i}\ ;\ b=\prod_{i=1}^{m}p_i^{b_i}\rightarrow\begin{cases}\gcd(a,b)=\prod_{i=1}^{m}p_i^{\min(a_i,b_i)}\\ \operatorname{lcm}(a,b)=\prod_{i=1}^{m}p_i^{\max(a_i,b_i)}\end{cases}}$$

$$24=2^3\cdot 3 \qquad \gcd(24,36)=2^{\min(3,2)}\cdot 3^{\min(1,2)}=2^2\cdot 3=12$$

$$36=2^2\cdot 3^2 \qquad \operatorname{lcm}(24,36)=2^{\max(3,2)}\cdot 3^{\max(1,2)}=2^3\cdot 3^2=72$$

Two integer numbers ***a,b*** are congruent modulo the positive integer ***m*** if ***m*** divides (***a***-***b***) or if ***a*** and ***b*** have the same remainder when divided by ***m***.

$$\boxed{a \equiv b(\operatorname{mod} m) \Leftrightarrow m|(a-b) \vee [a \operatorname{mod} m = b \operatorname{mod} m]}$$

$$13 \equiv 1(\operatorname{mod} 12) \Leftrightarrow 12|(13-1) \vee [13 \operatorname{mod} 12 = 1 = 1 \operatorname{mod} 12]$$

$$-7 \equiv -3(\operatorname{mod} 4) \Leftrightarrow 4|(-7+3) \vee [-7 \operatorname{mod} 4 = 1 = -3 \operatorname{mod} 4]$$

Since ***m*** divides ***(a-b)***, by definition,

$$\boxed{a \equiv b(\operatorname{mod} m) \Leftrightarrow a = b + km,\ k \in Z}$$

$$m|(a-b) \rightarrow m|(b+km-b) \rightarrow m|km = k$$

Modular arithmetic consists of the *usual sum and multiplication operations with respect to a fix modulus* ***m***. <u>Cyclic events or devices</u> can be described by a modular algebra; a common example is the clock algebra with ***m = 12*** or ***24***.

$$\boxed{a \equiv b \pmod m \wedge c \equiv d \pmod m \rightarrow a + c = b + d \pmod m}$$

$$a \equiv b \pmod m \Leftrightarrow m \mid (a - b) \Leftrightarrow a = b + k_1 m$$

$$c \equiv d \pmod m \Leftrightarrow m \mid (c - d) \Leftrightarrow c = d + k_2 m$$

$$\Rightarrow a + c = b + d + (k_1 + k_2) m$$

$$\Rightarrow (a + c) = (b + d) + km$$

$$\boxed{a \equiv b \pmod m \wedge c \equiv d \pmod m \rightarrow ac = bd \pmod m}$$

$$\Rightarrow ac = (b + k_1 m)(d + k_2 m) = bd + (k_2 b + k_1 d + k_1 k_2 m) m$$

$$\Rightarrow (ac) = (bd) + km$$

The following message without spaces

LNHWNQGHCQFIGMSHYDP

was encrypted using the following affine transformation,

$$f(p) = (7p + 3) \bmod 26$$

If someone in the class finds the original message using the corresponding decryption function then there will be no quiz on day ?, but if no one decrypts the message *we will do what the message says*.

Original message was: QUIZ UNTIL NEXT FRIDAY. Several students decrypted the message, but *where is the inverse function*?

$$f^{-1}(p) = ??$$

The following message without spaces

YDUHY JHDX

was encrypted using the following affine transformation

$$f(p) = (7p + 3) \bmod 26$$

The following solution has been established by David Miao. As you can see, it is a *nice inverse transformation* of the original function. Keep it, perhaps You will need to send a secret message.

$$f^{-1}(p) = \frac{1}{7}\left(26 \cdot \left[(11p + 2) \bmod 7\right] + p - 3\right)$$

Pseudorandom numbers: the linear congruential method for generating a sequence of this kind of numbers is given by the expression,

$$x_{n+1} = (a x_n + c) \bmod m; n \in N$$

- $\boldsymbol{x_0}$ is the <u>seed</u> of the generator
- $\boldsymbol{m}$ is the <u>modulus</u>
- $\boldsymbol{a}$ is the <u>multiplier</u>
- $\boldsymbol{c}$ is the <u>increment</u>

$$\forall n, 0 \le x_n < m; 2 \le a < m; 0 \le c < m$$

For example, consider the choice: $\boldsymbol{x_0 = 3}$, $\boldsymbol{m = 7}$, $\boldsymbol{a = 4}$ and $\boldsymbol{c = 1}$,

$$x_1 = (4x_0 + 1) \bmod 7 = 13 \bmod 7 = 6$$
$$x_2 = (4x_1 + 1) \bmod 7 = 25 \bmod 7 = 4$$
$$x_3 = (4x_2 + 1) \bmod 7 = 17 \bmod 7 = 3$$
$$x_4 = (4x_3 + 1) \bmod 7 = 13 \bmod 7 = 6$$

$$\{x_n\}_{n \in N} = \{3,6,4,3,6,4,\ldots\}$$

A useful generator is given by:

$$x_{n+1} = 7^5 x_n \bmod (2^{31} - 1)$$

The length of its cycle is $\boldsymbol{2^{31} - 2}$ (before repetition begins).

- In each of the following cases, what are the quotient and remainder?

$$-111/11 \rightarrow -111 = 11\cdot(-11) + 10\ ;\ 0 \le 10 < 11$$

$$-1/3 \rightarrow -1 = 3\cdot(-1) + 2\ ;\ 0 \le 2 < 3$$

- Find the prime factorization of **10!**

$$10! = 1.2.3.4.5.6.7.8.9.10 = 2.3.2^2.5.(2.3).7.2^3.3^2.(2.5) = \boxed{2^8 \cdot 3^4 \cdot 5^2 \cdot 7}$$

- Which memory locations are assigned by the hashing function ***h***(***k***) = ***k* mod 101** to the records of students with the following SSN?

$$h(104578690) = 104578690 \bmod 101 = 58$$

$$h(432222187) = 432222187 \bmod 101 = 60$$

With hashing functions it is possible to find a record very quickly.

- Decrypt the following message encrypted using the Caesar cipher.

Julius Caesar decryption method is given by $f^{-1}(p) = (p-3) \bmod 26$ therefore,

$$\text{WHVW WRGDB} \rightarrow \boxed{\text{TEST TODAY}}$$

Euclidean algorithm[a]

From the division algorithm we know that $\boxed{a = bq + r \,;\, 0 \le r < b}$

Consider the following,

$$d \mid a \wedge d \mid b \rightarrow d \mid a \wedge \boxed{d \mid b(-q)} \rightarrow \boxed{d \mid a - bq = r} \qquad D_{a,b} \subseteq D_{b,r}$$

$$d \mid b \wedge d \mid r \rightarrow \boxed{d \mid bq} \wedge d \mid r \rightarrow \boxed{d \mid bq + r = a} \qquad D_{b,r} \subseteq D_{a,b}$$

$$(D_{a,b} \subseteq D_{b,r}) \wedge (D_{b,r} \subseteq D_{a,b}) \Leftrightarrow D_{a,b} = D_{b,r}$$

Since both sets of common divisors are the same, they have the same maximum, so

$$\boxed{\gcd(a,b) = \max D_{a,b} = \max D_{b,r} = \gcd(b,r)}$$

$$18 = 12 \cdot 1 + 6$$

$$D_{18,12} = \{1,2,3,6\}$$

$$D_{12,6} = \{1,2,3,6\}$$

$$\gcd(18,12) = 6 = \gcd(12,6)$$

Using this idea we will find the **gcd(r_0, r_1)** after several divisions as follows:

$$r_0 = r_1 q_1 + r_2 \;;\; 0 \le r_2 < r_1 \qquad \gcd(r_0, r_1) = \gcd(r_1, r_2)$$

$$r_1 = r_2 q_2 + r_3 \;;\; 0 \le r_3 < r_2 \qquad \gcd(r_1, r_2) = \gcd(r_2, r_3)$$

$$r_2 = r_3 q_3 + r_4 \;;\; 0 \le r_4 < r_3 \qquad \gcd(r_2, r_3) = \gcd(r_3, r_4)$$

$$\vdots$$

$$r_{n-2} = r_{n-1} q_{n-1} + r_n \;;\; 0 \le r_n < r_{n-1} \qquad \gcd(r_{n-2}, r_{n-1}) = \gcd(r_{n-1}, r_n)$$

$$\boxed{r_{n-1} = r_n q_n} \qquad \boxed{\gcd(r_{n-1}, r_n) = \gcd(r_n, 0) = r_n}$$

last nonzero remainder

$$\boxed{277 = 123} \cdot 2 + 31$$

$$123 = 31 \cdot 3 + 30$$

$$31 = 30 \cdot 1 + 1$$

$$\boxed{54321 = 12345} \cdot 4 + 4941$$

$$12345 = 4941 \cdot 2 + 2463$$

$$4941 = 2463 \cdot 2 + 15$$

$$2463 = 15 \cdot 164 + 3$$

Number theory

Euclidean algorithm[c]

Algorithm: greatest common divisor

$$\text{procedure } \gcd(a, b \in Z^+)$$
$$\{a \geq b\}$$
$$x := a$$
$$y := b$$
$$\text{while } y \neq 0$$
$$\quad r := x \bmod y$$
$$\quad x := y$$
$$\quad y := r$$
$$\{\gcd(a,b) = x\}$$

$$\boxed{\gcd(277,123) = 1}$$

x	y	$y \neq 0$	r	x	y
277	123	$123 \neq 0$	31	123	31
		$31 \neq 0$	30	31	30
		$30 \neq 0$	1	30	1
		$1 \neq 0$	0	1	0
		$0 = 0$		1	

Two additional definitions:

- ***a*** and ***b*** are relatively prime if $\boxed{\gcd(a,b) = 1,}$
- a sequence {a_n} from ***n*** = 1 to ***m*** is pairwise relatively prime, if and only if,

$$\boxed{\gcd(a_i, a_j) = 1, \forall i \neq j}$$

Base-b representation[a]

Integers *are usually represented* using the decimal notation, but computers use other representations such as binary, octal or hexadecimal.

In general, we can consider a representation respect to a positive integer ***b*** > 1. The number ***b*** is called the base, and the corresponding representation of a positive number ***n*** in base ***b*** is called the base-***b*** expansion of ***n***.

$$n = \sum_{m=1}^{k} a_m b^m = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_0$$

$$k \in Z^+,\ a_k \neq 0 \wedge \forall m, a_m < b$$

$$n = (a_k a_{k-1} \cdots a_1 a_0)_b$$

leftmost digit

rightmost digit

- ***b*** = 2, binary
- ***b*** = 8, octal
- ***b*** =16, hexadecimal

$$B = \{0,1\}$$

$$O = \{0,1,\ldots,7\}$$

$$H = \{0,\ldots,9,A,\ldots,F\}$$

Base-b representation[b]

$$\boxed{(\bullet)_b \to (*)_{10}}$$

$$\boxed{(1011)_2} \to \underline{1}\cdot 2^3 + \underline{0}\cdot 2^2 + \underline{1}\cdot 2^1 + \underline{1}\cdot 2^0 = 8+2+1 = \boxed{(11)_{10}}$$

$$\boxed{(707)_8} \to \underline{7}\cdot 8^2 + \underline{0}\cdot 8^1 + \underline{7}\cdot 8^0 = 448+7 = \boxed{(455)_{10}}$$

$$\boxed{(3AF)_{16}} \to \underline{3}\cdot 16^2 + \underline{10}\cdot 16^1 + \underline{15}\cdot 16^0 = 768+160+15 = \boxed{(943)_{10}}$$

$$\boxed{(\bullet)_{10} \to (*)_b}$$

Apply the division algorithm until the <u>quotient is zero</u>, the first to the last remainder correspond to the digits in the expansion from right to left.

$$1023 = 16\cdot 63 + \underline{15} \to 63 = 16\cdot 3 + \underline{15} \to 3 = 16\cdot 0 + \underline{3}$$

$$\boxed{(1023)_{10} = (3FF)_{16}}$$

Number theory

Base-*b* representation[c]

Algorithm: base-*b* expansion

$$\textbf{procedure } bbe(n, b \in Z^{+})$$
$$q := n$$
$$k := 0$$
$$\textbf{while } q \neq 0$$
$$\quad a_k := q \bmod b$$
$$\quad q := \lfloor q / b \rfloor$$
$$\quad k := k + 1$$
$$\{\text{vector } a \text{ is the } b\text{-expansion}\}$$

$$(1023)_{10} = (3FF)_{16}$$

q	k	$q \neq 0$	a_k	q	k
1023	0	$1023 \neq 0$	15	63	1
		$63 \neq 0$	15	3	2
		$3 \neq 0$	3	0	3

The balanced ternary expansion is:

$$n = \sum_{m=1}^{k} e_m 3^m \; ; T = \{-1, 0, 1\}$$

$$(5)_{10} = 1 \cdot 3^2 + (-1) \cdot 3^1 + (-1) \cdot 3^0 = (1\bar{1}\bar{1})_{b3}$$

$$(13)_{10} = 1 \cdot 3^2 + 1 \cdot 3^1 + 1 \cdot 3^0 = (111)_{b3}$$

$$(79)_{10} = 1 \cdot 3^4 + (-1) \cdot 3^1 + 1 \cdot 3^0 = (100\bar{1}1)_{b3}$$

Number theory — Multiplication[a]

Two basic operations in binary arithmetic are addition and multiplication. Here is an example,

$$\boxed{(1110)_2 \times (1010)_2} \Rightarrow \begin{array}{r} 1110 = a \\ 1010 = b \\ \hline 0000 \\ 1110 \\ 0000 \\ 1110 \\ \hline \boxed{10001100} \end{array}$$

$$\begin{array}{l} c = 0 \leftarrow 0 \\ c = a \leftarrow 1 \\ c = 0 \leftarrow 2 \\ c = a \leftarrow 3 \end{array}$$

Two binary numbers ***a***, ***b*** of length ***n* = 4**,

The list of partial products ***c***; if ***b*** has a *zero bit* then ***c* = 0**, if ***b*** has a *one bit* then ***c*** = ***a*** but shifted to the left according to the bit position in ***b***.

The addition of all partial products gives the result containing at most **2*n* = 8** bits.

$$ab = a\sum_{j=0}^{n-1} b_j 2^j = \sum_{j=0}^{n-1} a(b_j 2^j) = \sum_{j=0}^{n-1} c_j$$

$$a(b_j 2^j) \rightarrow \mathrm{lshf}(ab_j, j)$$

$$ab_j = if\,(b_j = 0, 0, a)$$

Algorithm: binary multiplication

procedure $binmult(a, b \in Z^+)$

$\{a = (a_{n-1} \cdots a_0), b = (a_{n-1} \cdots a_0)\}$

for $j := 0$ **to** $n-1$

  if $b_j = 1$

    then $c_j := \text{lshf}(a, j)$

    else $c_j := 0$

{partial products are in c}

$p := 0$

for $j := 0$ **to** $n-1$

  $p := binadd(p, c_j)$

$\{p = ab\}$

To calculate the partial products, the number of shifts is given by:

$$0 + 1 + 2 + \cdots + (n-1) = \boxed{\sum_{i=0}^{n-1} i \approx O(n^2)}$$

In the final for loop, procedure *binadd* takes ***O***(***n***) operations to add two partial products. Therefore, the number of additions for ***p***, is:

$$\left.\begin{aligned} &\max\{j\} \approx O(n) \\ &binadd(p, c_j) \approx O(n) \end{aligned}\right\} \rightarrow \boxed{p \approx O(n^2)}$$

The total number of operations needed for this algorithm is then of the same order,

$$\boxed{O(n^2) + O(n^2) \approx O(n^2)}$$

Matrices

- Matrices
- Operations
- Boolean matrices

Matrices — Definitions

A matrix is a *rectangular array of numbers* with ***m*** rows and ***n*** columns. Upper case letters denote matrices of size ***m*** x ***n***, and each element of a matrix is a number.

Expanded notation

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

2nd row

2nd column

Compact notation

$$\boxed{A = [a_{ij}]; \begin{cases} i = 1, \ldots, m \\ j = 1, \ldots, n \end{cases}}$$

element or entry $\boldsymbol{a}_{ij}$ is located in the intersection of *row **i*** and *column **j***.

- The transpose of a matrix is obtained by interchanging rows and columns:

$$\boxed{A^T = [a_{ji}]; \begin{cases} j = 1, \ldots, n \\ i = 1, \ldots, m \end{cases}}$$

its size is ***n*** x ***m***

- A square matrix is obtained by taking ***m*** = ***n*** (same number of rows and columns).

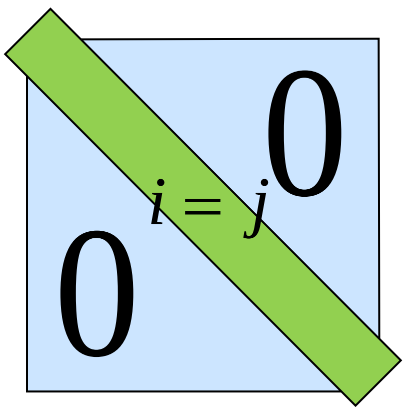
Matrices Types

A square matrix has n^2 elements, if $i = j$ the set $\{a_{ii}\}$ is the main diagonal.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

A symmetric matrix is a matrix equal to its transpose.

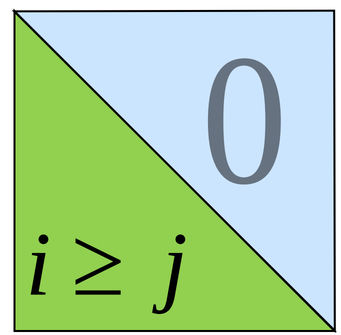
$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = A^T$$



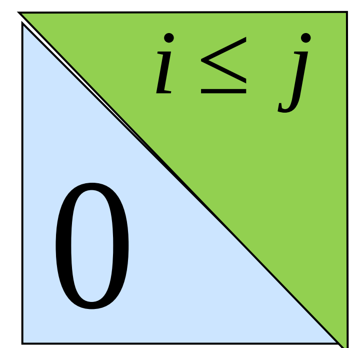
diagonal

- The identity matrix I is defined as a *diagonal matrix* where:

$$\delta_{ij} = \begin{cases} 1 & , i = j \\ 0 & , i \neq j \end{cases}$$



lower triangular



upper triangular

Arithmetic operations:

- addition and difference
  $\boldsymbol{A},\boldsymbol{B}$ of size $\boldsymbol{m}$ x $\boldsymbol{n}$

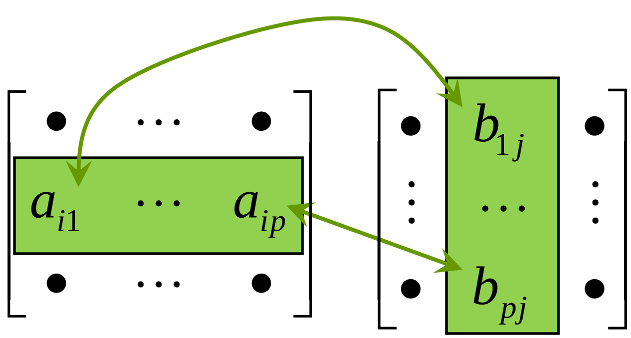
$$A \pm B = C \Leftrightarrow \forall i, j\,; [c_{ij}] = [a_{ij} \pm b_{ij}]$$

- multiplication
  $\boldsymbol{A}$ of size $\boldsymbol{m}$ x $\boldsymbol{p}$, $\boldsymbol{B}$ of size $\boldsymbol{p}$ x $\boldsymbol{n}$

$$A \cdot B = C \Leftrightarrow \forall i, j\,; [c_{ij}] = \sum_{k=1}^{p} a_{ik} b_{kj}$$

- inversion of $\boldsymbol{A}$
  only for square matrices $\boldsymbol{n}$ x $\boldsymbol{n}$

$$\exists C,\ A \cdot C = I = C \cdot A \Rightarrow C = A^{-1}$$



$$[c_{ij}] = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{ip} b_{pj}$$

$$m \times p \longleftrightarrow p \times n$$

the # of columns in $\boldsymbol{A}$ must be equal to the # of rows in $\boldsymbol{B}$.

Powers of square matrices are defined as:

$$A^0 = I, \quad A^k = \underbrace{A \cdot A \cdots A}_{k \text{ times}}$$

Algorithm: matrix multiplication

$$
\begin{aligned}
&\textbf{procedure}\ \ matmult\,(A,B)\\
&\{A \text{ is } m\times p, B \text{ is } p\times n\}\\
&\textbf{for } i=1 \textbf{ to } m\\
&\quad \textbf{for } j=1 \textbf{ to } n\\
&\qquad c_{ij} := 0\\
&\qquad \textbf{for } k=1 \textbf{ to } p\\
&\qquad\quad c_{ij} := c_{ij} + a_{ik}\cdot b_{kj}\\
&\{C = AB, C \text{ is } m\times n\}
\end{aligned}
$$

Number of operations in the innermost loop:

$a_{ik}\cdot b_{kj}\,;\, k=1,\dots,p \rightarrow p$ multiplications

$c_{ij}+(\bullet)\,;\, k=1,\dots,p \rightarrow p-1$ additions

$2p-1$ operations per element

Number of entries in matrix ***C*** is $m\cdot n$

Total is: $\boxed{m\cdot n\cdot(2p-1)}$ (only multiplications, $m\cdot n\cdot p$)

for square matrices, $m\cdot n\cdot(2p-1)\approx \boxed{O(n^3)}$

- matrix multiplication is not commutative but it is associative.

$$\boxed{AB\neq BA,\quad (AB)C = A(BC)}$$

$(A, m\times p)$

$(B, p\times q)$

$(C, q\times n)$

$(AB)C \rightarrow mqp + mnq = mq(p+n)$

$A(BC) \rightarrow mnp + pnq = np(m+q)$

The order for multiplying ***A, B, C*** can be selected by calculating,

$$\boxed{\min(mq(p+n), np(m+q))}$$

- Find the product ***AB***, where

$$A=\begin{bmatrix}1 & 0 & 1\\ 0 & -1 & -1\\ -1 & 1 & 0\end{bmatrix};\ B=\begin{bmatrix}0 & 1 & -1\\ 1 & -1 & 0\\ -1 & 0 & 1\end{bmatrix}$$

Some explicit calculations are:

$$\boxed{c_{11}=1\cdot 0+0\cdot 1+1\cdot(-1)=-1}$$

$$c_{12}=1\cdot 1+0\cdot(-1)+1\cdot 0=1$$

$$c_{13}=1\cdot(-1)+0\cdot 0+1\cdot 1=0$$

$$\Rightarrow AB=\boxed{\begin{bmatrix}-1 & 1 & 0\\ 0 & 1 & -1\\ 1 & -2 & 1\end{bmatrix}}$$

- Let ***A*** and ***B*** be two ***n x n*** matrices. Show that,

$$\boxed{(AB)^t = B^t A^t}$$

$$[c_{ij}]=\sum_{k=1}^{n} a_{ik}b_{kj} \Rightarrow [c_{ji}]=\sum_{k=1}^{n} a_{ki}b_{jk} = \sum_{k=1}^{n} b_{jk}a_{ki}$$

- Let ***A*** be a matrix. Show that the matrix ***AA***$^{\mathbf{t}}$ is symmetric.

$$\boxed{(AA^t)^t}=(A^t)^t A^t = \boxed{AA^t}$$

by definition, the given matrix equals its transpose.

Binary or boolean matrices have entries in the set ***B*** = {**0**,**1**} and their operations correspond to the usual logic or bit calculations. They are also called zero-one matrices.

- join

$$A \vee B = [a_{ij} \vee b_{ij}] = [a_{ij} \text{ or } b_{ij}]$$

- meet

$$A \wedge B = [a_{ij} \wedge b_{ij}] = [a_{ij} \text{ and } b_{ij}]$$

- boolean product

$$A \otimes B = C = [c_{ij}] = \bigvee_{k=1}^{p} (a_{ik} \wedge b_{kj})$$

- powers

$$A^0 = I \; ; \; A^k = \underbrace{A \otimes A \otimes \cdots \otimes A}_{k \text{ times}}$$

For square binary matrices ***A*** and ***B***, $\boxed{A \otimes B \approx O(n^3)}$ bit operations.

- Find the Boolean product of ***A*** and ***B***, where

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}; B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$$

Since ***A*** is 3 x 4 and ***B*** is 4 x 2 the result ***C*** has size 3 x 2,

$$C = AB = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{bmatrix} = \boxed{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}}$$

$$c_{11} = (1 \wedge 1) \vee (0 \wedge 0) \vee (0 \wedge 1) \vee (1 \wedge 1) = 1$$

$$\boxed{c_{12} = (1 \wedge 0) \vee (0 \wedge 1) \vee (0 \wedge 1) \vee (1 \wedge 0) = 0}$$

- Let ***A*** be an ***n*** x ***n*** zero-one matrix. Let ***I*** be the ***n*** x ***n*** identity matrix.
  Show that, $\boxed{A \otimes I = A = I \otimes A}$

$$[c_{ij}] = \bigvee_{k=1}^{n} (a_{ik} \wedge \delta_{kj}) = \boxed{(a_{ij} \wedge \delta_{jj})} \vee \bigvee_{k \neq j}^{n} (a_{ik} \wedge \delta_{kj}) = \boxed{(a_{ij} \wedge 1)} = [a_{ij}]$$

$$[c_{ij}] = \bigvee_{k=1}^{n} (\delta_{ik} \wedge a_{kj}) = \boxed{(\delta_{ii} \wedge a_{ij})} \vee \bigvee_{k \neq i}^{n} (\delta_{ik} \wedge a_{kj}) = \boxed{(1 \wedge a_{ij})} = [a_{ij}]$$

Mathematical reasoning: Part I

- Rules of inference
- Fallacies
- Methods of proof
- Mathematical propositions

- When is a mathematical argument correct?
- What methods can be used to construct mathematical arguments?
- How are mathematical propositions classified?

A mathematical argument has the form:

premises or hypotheses ← $$(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \rightarrow q$$ → conclusion or thesis

A mathematical argument is valid if and only if the implication is a tautology:

$$[(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \rightarrow q] = T \; ; \; p_1, p_2, \cdots, p_n \therefore q$$

The rules of inference are then universal valid arguments that constitute the fundamental patterns or forms that we use for higher thinking and we can consider them as our basic ***human built-in operators***.

Reasoning-I

Rules of inference basic forms

$p \therefore p \vee q$

addition

$p \wedge q \therefore p$

simplification

$p, q \therefore p \wedge q$

conjunction

p	q	$p \vee q$	$p \rightarrow p \vee q$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	T

Modus ponens or law of detachment

$p, p \rightarrow q \therefore q$

$\sim q, p \rightarrow q \therefore \sim p$

Modus tollens

$\sim p, p \vee q \therefore q$

Disjunctive syllogism

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

Hypothetical syllogism

p	q	$p \rightarrow q$	$p \wedge (p \rightarrow q)$	$(p \wedge (p \rightarrow q)) \rightarrow q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

- Construct an argument using rules of inference to show that the hypotheses "If it does not rain or if it is not foggy, then the sailing race will be held and the life-saving demonstration will go on", "If the sailing race is held, then the trophy will be awarded," and "The trophy was not awarded" imply the conclusion "It rained."

$$\underbrace{(\sim r \vee \sim f \rightarrow s \wedge l),\ s \rightarrow t,\ \sim t}_{\text{hypotheses}}\ \therefore\ r$$

conclusion

by **modus tollens**, $s \rightarrow t, \sim t \therefore \sim s$

by **simplification**, $s \wedge l \therefore s$

hence by **hypothetical syllogism** $\sim r \vee \sim f \rightarrow s$

So, we can apply again **modus tollens** to conclude that $\sim(\sim r \vee \sim f)$

Finally we use **De Morgan's law** and **simplification** again, $r \wedge f \therefore r$

Quantification rules of inference

$$\forall x\, P(x) \therefore P(c), c \in U$$

universal instantiation
c is a particular element of ***U***

$$P(c), c \in U \therefore \forall x\, P(x)$$

universal generalization
c is an arbitrary element of ***U***

$$\exists x\, P(x) \therefore P(c), c \in U$$

existential instantiation
c is a specific element of ***U***,
we have to find it if possible.

$$P(c), c \in U \therefore \exists x\, P(x)$$

existential generalization
c is a particular element of ***U***,
we already know its value.

The <u>rules of inference</u> for propositional logic and quantified statements are used extensively in mathematical arguments.

Quantification inference examples

- What rules of inference are used in the following famous argument?
  "All men are mortal. Socrates is a man. Therefore, Socrates is mortal."

$$\forall x(H(x) \rightarrow M(x)), H(\text{Socrates}) \therefore M(\text{Socrates})$$

We apply universal instantiation with $x = \text{Socrates}$ so we have

$$H(\text{Socrates}) \rightarrow M(\text{Socrates}), H(\text{Socrates}) \therefore \boxed{M(\text{Socrates})}$$

Note that modus ponens has been used to obtain the conclusion.

- Ryan, a student in this class, knows how to write programs in JAVA. Everyone who knows hot to write programs in JAVA can get a high-paying job. Therefore, someone in this class can get a high paying job."

$$C(\text{Ryan}) \wedge J(\text{Ryan}), \forall x(J(x) \rightarrow H(x)) \therefore \exists x(C(x) \wedge H(x))$$

We apply universal instantiation with $x = \text{Ryan}$ then $J(R) \rightarrow H(R)$
also from $J(R) \rightarrow H(R), J(R)$ we get $H(R)$; finally, combining this result with the first hypothesis the conlusion is obtained from existential generalization.

Fallacies

A fallacy resembles a rule of inference but is based on contingencies rather than tautologies, so it corresponds to an invalid argument.

$p \rightarrow q, q \therefore ?\, p$

affirming the conclusion

$p \rightarrow q, \sim p \therefore ? \sim q$

denying the hypothesis

$p \therefore ?\, p$

begging the question or circular reasoning

$\boxed{p \rightarrow p = T}$

but then *we are not proving anything* !

p	q	$p \rightarrow q$	$(p \rightarrow q) \wedge q$	$(p \wedge (p \rightarrow q)) \rightarrow p$
T	T	T	T	T
T	F	F	F	T
F	T	T	T	F
F	F	T	F	T

$\boxed{p = F\,;\, q = T}$

$$\Rightarrow (p \rightarrow q) \wedge \sim p = (F \rightarrow T) \wedge T = T$$

$$\Rightarrow T \rightarrow \sim q = T \rightarrow F = \boxed{F}$$

- The following argument is an incorrect proof of the theorem "If n^2 is not divisible by 3, then n is not divisible by 3." The reason it is incorrect is that circular reasoning has been used. Where has the error in reasoning been made?

If n^2 is not divisible by 3, then n^2 does not equal $3k$ for some integer k. Hence, n does not equal $3l$ for some integer l. Therefore, n is not divisible by 3.

The initial theorem is of the form: $p \to q$ but the argument given looks like,

$$p \to \boxed{r \to r} \to q$$

To complete the problem we will try to give an indirect proof using the **contrapositive**,

$$\sim q \to \sim p \Leftrightarrow 3 | n \to 3 | n^2$$

Applying the definition of **integer division**,

$$3|n \Leftrightarrow \exists k \in Z, n = 3k \to n^2 = \boxed{3(3k^2)} = 3l, l \in Z$$

Reasoning-I — Methods of proof

- Direct
- Indirect $p \to q \Leftrightarrow \sim q \to \sim p$

$$p \to q$$

$$\sim q \to \sim p$$

premise conclusion

- Vacuous $p \to q\, ;\, p = F$ ($F \to q = T$)
- Trivial $p \to q\, ;\, q = T$ ($p \to T = T$)

- Contradiction

$$p\,;\sim p \to (r \wedge \sim r) = T \Rightarrow \sim p = F$$

We negate the thesis ***p*** and derive a contradiction, therefore our *assumption that the thesis was not true is false*, hence ***p*** is true.

- By cases

$$\left(\bigvee_{k=1}^{n} p_k\right) \to q \Leftrightarrow \bigwedge_{k=1}^{n} (p_k \to q)$$

- Multiple equivalences

$$[p_1 \leftrightarrow p_2 \leftrightarrow \cdots \leftrightarrow p_n] \Leftrightarrow \bigwedge_{k=1}^{n} (p_k \to p_{(k+1) \bmod n})$$

In both methods we use the logical equivalence given by the conjunction of several implications, so we prove each of this implications one by one.

- Prove the proposition ***P***(***1***), where ***P***(***n***) is the proposition "If ***n*** is a positive integer, then $n^2 \geq n$." What kind of proof did you use?

$$1 \in Z^+ \rightarrow 1^2 \geq 1$$

This is a **trivial proof** since the conclusion is true (evident) without using the premise.

- Prove that at least one of the real numbers $a_1, a_2, \ldots, a_n$ is greater than or equal to the average of these numbers. What kind of proof did you use?

$$\exists a_i, a_i \geq \alpha = \frac{1}{n}\sum_{i=1}^{n} a_i$$ by **contradiction**, $$\forall a_i, a_i < \alpha$$ however,

$$a_i < \alpha \rightarrow a_1 + a_2 + \cdots + a_n < \underbrace{\alpha + \alpha + \cdots + \alpha}_{n \text{ times}} = n\alpha$$ or,

$$\frac{1}{n}\sum_{i=1}^{n} a_i < \alpha \rightarrow \alpha < \alpha$$

and this is the contradiction we were looking for.

- Prove that if ***x*** and ***y*** are real numbers, then **max**(***x***,***y***)+**min**(***x***,***y***)=***x***+***y***. (*Hint*: Use a proof by cases, with the two cases corresponding to ***x*** ≥ ***y*** and ***x*** < ***y***.)

Case x ≥ y $[\max(x,y)=x]\wedge[\min(x,y)=y]\rightarrow \boxed{x+y=x+y}$

Case x < y $[\max(x,y)=y]\wedge[\min(x,y)=x]\rightarrow \boxed{y+x=x+y}$

- Use a proof by cases to show that **min**(***a***,**min**(***b***,***c***))=**min**(**min**(***a***,***b***),***c***) whenever, ***a***,***b***, and ***c*** are real numbers.

Case *a* < *b* < *c* $\min(a,\min(b,c))=\min(a,b)=a=\min(a,c)=\min(\min(a,b),c)$

Case *b* < *c* < *a* $\min(a,\min(b,c))=\min(a,b)=b=\min(b,c)=\min(\min(a,b),c)$

Case *c* < *a* < *b* $\min(a,\min(b,c))=\min(a,c)=c=\min(a,c)=\min(\min(a,b),c)$

$\min(a,\min(b,c))=\min(\min(a,b),c)=\boxed{\min(a,b,c)}$ associative property of **min**

- Prove that n^4 - 1 is divisible by **5** when ***n*** is not divisible by **5**. Use a **proof by cases**, being four different cases - one for each of the non-zero remainders that an integer not divisible by **5** can have when you divide it by **5**.

$$5 \nmid n \to 5 \mid n^4 - 1$$

From the binomial expansion, $(x+y)^4 = x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$

$$\boxed{\text{case } n = 5k+1} \to n^4 = (5k+1)^4 = 5(\bullet) + 1^4 = 5l_1 + 1 \to n^4 - 1 = 5l_1$$

$$\boxed{\text{case } n = 5k+2} \to n^4 = (5k+2)^4 = 5(\bullet) + 2^4 = 5(\bullet) + 16$$

$$= 5(\bullet) + 15 + 1 = 5l_2 + 1 \to n^4 - 1 = 5l_2$$

$$\boxed{\text{case } n = 5k+3} \to n^4 = (5k+3)^4 = 5(\bullet) + 3^4 = 5(\bullet) + 81$$

$$= 5(\bullet) + 80 + 1 = 5l_3 + 1 \to n^4 - 1 = 5l_3$$

$$\boxed{\text{case } n = 5k+4} \to n^4 = (5k+4)^4 = 5(\bullet) + 4^4 = 5(\bullet) + 256$$

$$= 5(\bullet) + 255 + 1 = 5l_4 + 1 \to n^4 - 1 = 5l_4$$

p = "The square root of 2 is not a rational number." $= \boxed{\sqrt{2} \notin Q}$ by contradiction,

$$\sqrt{2} \in Q \to \sqrt{2} = \frac{m}{n}$$

$$Q = \{\tfrac{m}{n} \mid m, n \in Z \wedge \gcd(m,n) = 1 \wedge n \neq 0\}$$

definition

$$\to m^2 = 2n^2 \to m^2 \text{ is even}$$ 1

Intermediate step: $\boxed{m^2 \text{ even} \to m \text{ even} \iff m \text{ odd} \to m^2 \text{ odd}}$ 2

$$m = 2k+1 \to m^2 = (2k+1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1 = 2l + 1$$

We go back to step 1 knowing that $\boxed{m = 2j \text{ for some } j \in Z^+}$ 3

$\to 2n^2 = 4j^2 \to n^2 = 2j^2$ so n^2 is even and apply step 2 again, thus

From step 3 and step 4 we conclude that: $\boxed{n = 2i \text{ for some } i \in Z^+}$ 4

$\boxed{(2 \mid m) \wedge (2 \mid n)}$ this is the contradiction we were looking for.

- Prove that the square root of 5 is irrational. Proof by contradiction,

$$\sqrt{5} \in Q \to \sqrt{5} = \frac{m}{n} \to (m^2 = 5n^2) \vee (5|m^2) \qquad \text{A}$$

Auxiliary step: if **5** divides ***m*²** then **5** divides ***m***. We show this using an undirect proof, i.e., if **5** does not divide ***m*** then **5** does not divide the square of ***m***. Consider the other possible remainders of ***m*** when divided by **5**, and treat each case as follows:

$$\left.\begin{aligned}
&\text{a) } m = 5k+1 \to m^2 = 25k^2 + 10k + 1 = 5l_1 + 1\\
&\text{b) } m = 5k+2 \to m^2 = 25k^2 + 20k + 4 = 5l_2 + 4\\
&\text{c) } m = 5k+3 \to m^2 = 25k^2 + 30k + 9 = 5l_3 + 4\\
&\text{d) } m = 5k+4 \to m^2 = 25k^2 + 40k + 16 = 5l_4 + 1
\end{aligned}\right\} \to 5 \nmid m^2$$

Since ***m*** = 5***p***; substitution in **A** gives: $25p^2 = 5n^2 \to (n^2 = 5p^2) \vee (5|n^2)$ **B**

Therefore, applying to **B** the same result established in the auxiliary step, ***n*** = 5***q***; The *contradiction* is that ***m*** and ***n*** have a common factor equal to **5**.

- existence $\boxed{\exists x\, P(x)}$

Constructive proof: we *find* or *build* a specific element ***c*** such that $P(c) = T$

Non-constructive proof: we only show that under certain assumptions there is an object ***c*** that makes ***P*** true *somehow*.

- counterexample $\boxed{\forall x\, P(x) = F \Leftrightarrow \exists x \neg P(x) = T}$

To show that a universal quantification is false we need to find just one element ***c*** in ***U*** such that the *negation of **P** is true*, in that case ***c*** is a counterexample.

- Prove or disprove each of the following statements about the floor and ceiling functions.

  - $\forall x \in R, \lfloor \lceil x \rceil \rfloor = \lceil x \rceil$ consider an arbitrary real number ***c*** then

$$\lceil c \rceil = m \geq c, m \in Z \rightarrow \lfloor m \rfloor = m \rightarrow \lfloor \lceil c \rceil \rfloor = \lceil c \rceil$$

Therefore, by **universal generalization** we see that the quantified predicate is true.

  - $\forall x, y \in R, \lfloor x + y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor$ take, ***x*** = ***y*** = **0.5**, then

$$\left\lfloor \tfrac{1}{2} + \tfrac{1}{2} \right\rfloor = 1 \neq 0 = \left\lfloor \tfrac{1}{2} \right\rfloor + \left\lfloor \tfrac{1}{2} \right\rfloor$$

this is a **counterexample**, so the double quantified predicate is false.

  - $\forall x \in R, \left\lfloor \sqrt{\lceil x \rceil} \right\rfloor = \left\lfloor \sqrt{x} \right\rfloor$ take, ***x*** = **1/4**, then

$$\left\lfloor \sqrt{\left\lceil \tfrac{1}{4} \right\rceil} \right\rfloor = 1 \neq 0 = \left\lfloor \sqrt{\tfrac{1}{4}} \right\rfloor$$

this is also a **counterexample**, so the quantified predicate is false.

- Prove or disprove that $n^2 + n + 1$ is prime whenever n is a positive integer.

The proposition is of the form: $\forall n \in Z^+, n^2 + n + 1 \in P$

and in this specific case is **false**, so we must prove that, $\exists n \in Z^+, n^2 + n + 1 \notin P$

Thus, it is enough to give a **counterexample**, a value of n for which $n^2 + n + 1$ is not a prime number. We test the following values,

$$n = 1 \to 1^2 + 1 + 1 = 3 \in P,$$

$$n = 2 \to 2^2 + 2 + 1 = 7 \in P,$$

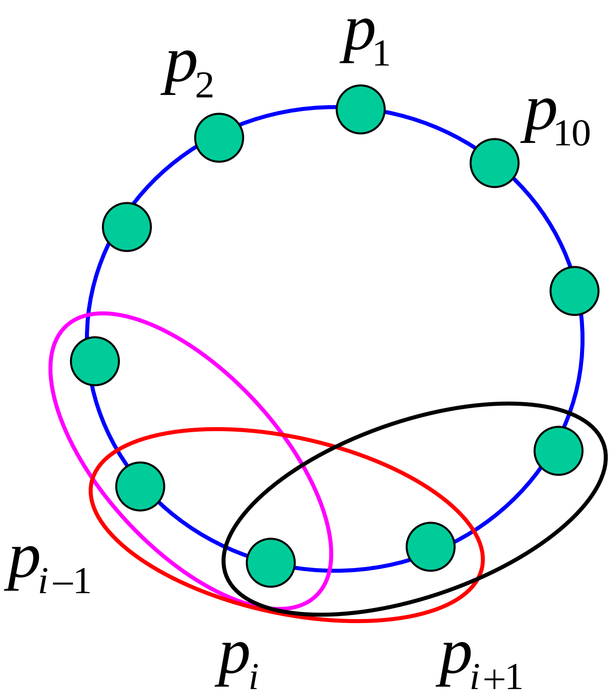
$$n = 3 \to 3^2 + 3 + 1 = 13 \in P,$$

$$\boxed{n = 4 \to 4^2 + 4 + 1 = 21 \notin P.}$$

Reasoning-I Methods of proof examples[f]

- Show that if the first 10 positive integers are placed around a circle, in any order, there exist 3 integers in consecutive locations that have a sum greater than or equal to 17.

p$_i$ means the *i-th position* of any given integer from 1 to 10.



$$\boxed{\forall i,\ p_i \in G_{i-1}, G_i, G_{i+1}}$$

$$\{p_1, p_2, p_3\} = G_1$$
$$\{p_2, p_3, p_4\} = G_2$$
$$\{p_3, p_4, p_5\} = G_3$$
$$\{p_4, p_5, p_6\} = G_4$$
$$\{p_5, p_6, p_7\} = G_5$$
$$\{p_6, p_7, p_8\} = G_6$$
$$\{p_7, p_8, p_9\} = G_7$$
$$\{p_8, p_9, p_{10}\} = G_8$$
$$\{p_9, p_{10}, p_1\} = G_9$$
$$\{p_{10}, p_1, p_2\} = G_{10}$$

Define the sequence of #s:

$$a_i = \sum_{x \in G_i} x\ ;\ i = 1, \ldots, 10$$

then its average is given by,

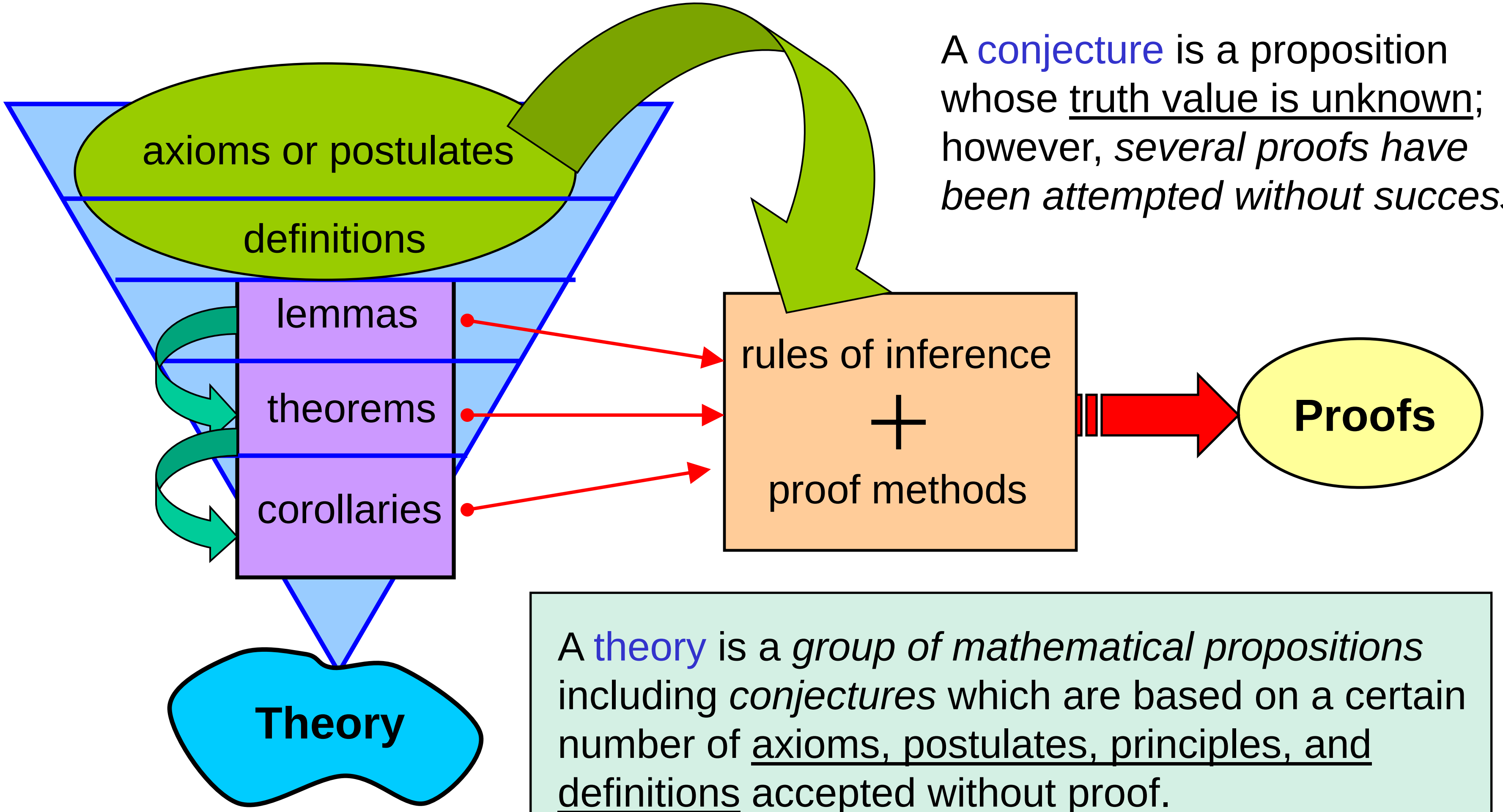
$$\alpha = \frac{3}{10}\sum_{i=1}^{10} i = \frac{3 \cdot 55}{10} = 16.5$$

From examples[a,2] there is a number ***a***$_j$ greater than 16.5 Since ***a***$_j$ is an integer then it is greater than or equal to **17**.

This is a non-constructive proof for the existence of an object.

Mathematical propositions

It is a usual practice to classify mathematical propositions by type, in order to organize the results of a *theory*.



A conjecture is a proposition whose truth value is unknown; however, *several proofs have been attempted without success*.

A theory is a *group of mathematical propositions* including *conjectures* which are based on a certain number of axioms, postulates, principles, and definitions accepted without proof.

Mathematical reasoning: Part II

- Well ordering
- Mathematical induction

The well ordering (w.o.) principle: every non-empty subset of the natural numbers has a least element.

$$\forall S \neq \varnothing \subseteq N, \exists m \in N; m = \min(S)$$

This principle is simple and intuitive; it works for finite and infinite sets. Examples,

$$\min\{2n+1 | n \in N\} = 1 \qquad \min\{x | x \text{ is a prime}\} = 2$$

$$\min\{a_n \mid n \in N \wedge a_i < a_{i+1}\ \forall i\} = a_0 \qquad \min\{n \equiv 0 (\text{mod } 7) | n > 0\} = 7$$

The principle of mathematical induction (m.i.): assume that ***S*** is a subset of ***N*** such that,

$$[(0 \in S) \wedge \forall k (k \in S \rightarrow k+1 \in S)] \rightarrow S = N$$

This principle is simple but not easy to grasp, however it is a property of the natural numbers when treated from an axiomatic point of view. It is the foundation of inductive reasoning in mathematics.

Well ordering implies induction

Theorem: $\text{w.o.} \leftrightarrow \text{m.i.} \quad \Leftrightarrow \boxed{(\text{w.o.} \rightarrow \text{m.i.})} \wedge (\text{m.i.} \rightarrow \text{w.o.})$

Proof: by contradiction, the hypotheses are,

$$\text{w.o.}, \; 0 \in S, \; k \in S \rightarrow k+1 \in S$$

The conclusion is:

$$S = N$$

$$S \neq N \rightarrow (D = N - S \neq \emptyset) \wedge (D \subset N)$$

The set difference is key to find a contradiction !

$$m = \min(D) \rightarrow m \neq 0 \rightarrow m > 0$$

$$m-1 \notin D \rightarrow m-1 \in S \rightarrow m \in S$$

$$(m \in D \wedge m \in S) = F$$

The <u>second part</u> is left as an exercise. $\boxed{\text{m.i.} \rightarrow \text{w.o.}}$

Well ordering examples

Recall that a set is well-ordered if every nonempty subset of this set has a least element. Determine whether each of the following sets is well-ordered.

a) the set of integers,
b) the set of integers greater than **-100**,
c) the set of positive rationals.
d) the set of positive rationals with denominator less than **100**.

a) ***Z*** is not a well-ordered set because $Z^{-} \neq \emptyset \subset Z \text{ but } \min(Z^{-}) = -\infty \notin Z$

b) this set is well-ordered since $A = \{x \in Z | x > -100\} \rightarrow \min(A) = -99$
Thus any nonempty subset ***S*** of ***A*** has a least element greater than or equal to **-99**.

c) the set of positive rationals defined as, $Q^{+} = \{r \in Q | r > 0\}$ is not well-ordered, e.g.

$$S = \{\frac{1}{n} \in Q | n > 0\} \rightarrow \min(S) \text{ does not exist.} \qquad \text{Note: } \lim_{n \to \infty} \frac{1}{n} = 0 \text{ but } 0 \notin Q^{+}$$

d) this last set is well ordered since $B = \{\frac{p}{q} > 0 | q < 100\} \rightarrow \min(B) = \frac{1}{99}$
Therefore, any nonempty subset ***S*** of ***B*** has a least element greater than or equal to **1/99**.

Mathematical induction is used as a proof technique when predicates are related to the domain of the natural numbers or one of its subsets; also stated in the following equivalent form:

Induction hypothesis

$$P(0) \wedge [\boxed{P(k)} \rightarrow P(k+1)] \therefore \forall n\, P(n)$$

basis step inductive step conclusion

In order to give a proof that ***P***(***n***) is true for all integers ***n*** we verify two steps,

1. **Basis step**: show that ***P***(**0**) is true; this part is *almost trivial*, substitute ***n*** = **0** in ***P***(***n***) and check if *the corresponding proposition is true*. Also, the basis step can begin with a specific value greater than **0**.

2. **Inductive step**: this is the *difficult one*; assume ***P***(***k***) is true for an *arbitrary integer* ***k*** and prove that ***P***(***k***+1) is also true. Avoid to substitute directly ***k*** with ***k***+1; this is circular reasoning because ***k***+1 = ***m*** is also an arbitrary integer and then your are assuming what you want to prove.

$$\boxed{\forall n > 0, \sum_{j=1}^{n} (2j-1) = n^2}$$

Basis step:

$$n = 1 \rightarrow \sum_{j=1}^{1} (2j-1) = (2 \cdot 1 - 1) = 1 = 1^2$$

Inductive step:

$$n = k \rightarrow \sum_{j=1}^{k} (2j-1) = k^2$$

Induction hypothesis

$$\sum_{j=1}^{k+1} (2j-1) = \boxed{\sum_{j=1}^{k} (2j-1) + (2 \cdot (k+1) - 1)} = k^2 + (2k+1) = (k+1)^2$$

$$P(k+1): \sum_{j=1}^{k+1} (2j-1) = (k+1)^2$$

$$\boxed{\forall n \in N, |A| = n \rightarrow |P(A)| = 2^n}$$

Basis step: $n = 0 \rightarrow (A = \varnothing \wedge |A| = 0) \rightarrow |P(\varnothing)| = |\{\varnothing\}| = 1 = 2^0$

Inductive step: $|A| = k \rightarrow |P(A)| = 2^k$ **Induction hypothesis**

$$|A| = k \leftrightarrow A = \{a_1, \ldots, a_k\} \rightarrow |A \cup \{x\}| = k+1;\ x \neq a_i$$

$$P(A \cup \{x\}) = F_1 \cup F_2 = P(A) \cup \boxed{\{S \cup \{x\} | S \in P(A)\}}$$

$$F_1 \cap F_2 = \varnothing \rightarrow |P(A \cup \{x\})| = |F_1| + |F_2| = 2^k + 2^k = 2 \cdot 2^k$$

$$|A^*| = k+1 \rightarrow |P(A^*)| = 2^{k+1}$$

$$\boxed{\forall n \geq 4, n^2 \leq 2^n}$$

Basis step: $n = 4 \rightarrow 4^2 = 16 \leq 16 = 2^4$

Inductive step: $k^2 \leq 2^k \ ; k > 4$ **Induction hypothesis**

$$\boxed{k > 4 \rightarrow 2k + 1 \leq 2^k}$$

$$\forall a, b, c, d \in Z^+, (a < b) \wedge (c < d) \rightarrow a + c < b + d$$

$$k^2 + (2k+1) \leq 2^k + 2^k$$

$$P(k+1) : (k+1)^2 \leq 2^{k+1}$$

Induction examples[d]

- Use **mathematical induction** to show that $\mathbf{2^n} > \boldsymbol{n}^{\mathbf{2}} + \boldsymbol{n}$ whenever $\boldsymbol{n}$ is an integer greater than **4.**

Basis step, take $\boldsymbol{n} = \mathbf{5} > \mathbf{4}$, then $2^5 = 32 > 30 = 25 + 5 = 5^2 + 5$

Inductive step, assume the inequality is true for $\boldsymbol{n} = \boldsymbol{k}$, i.e.,

$$\overbrace{2^k > k^2 + k}^{\text{induction hypothesis}} = k(k+1) > 2(k+1) \text{ since } k \geq 6$$

by transitivity of > , and adding both inequalities, $\boxed{2^k > 2(k+1)}$

$$\boxed{2^{k+1}} = 2^k + 2^k > k^2 + k + 2(k+1) = k^2 + k + 2k + 2$$

$$> (k^2 + 2k + 1) + (k+1) = \boxed{(k+1)^2 + (k+1)}$$

this is the right side for $\boldsymbol{n} = \boldsymbol{k} + \mathbf{1}$.

- For all positive integers ***n***, show, by **mathematical induction**, that:

$$\frac{1}{1\cdot 3}+\frac{1}{3\cdot 5}+\cdots+\frac{1}{(2n-1)(2n+1)}=\frac{n}{2n+1}$$

Basis step, take ***n*** = 1, then $\boxed{\frac{1}{1\cdot 3}}=\frac{1}{3}=\boxed{\frac{(1)}{2(1)+1}}$

Just to see if the formula works, take, for example ***n*** = 2, then both sides are equal,

$$\boxed{\frac{1}{1\cdot 3}+\frac{1}{3\cdot 5}}=\frac{1}{3}+\frac{1}{15}=\frac{6}{15}=\frac{2}{5}=\boxed{\frac{(2)}{2(2)+1}}$$

Inductive step: assume that the formula is true for ***n*** = ***k***, and show its validity for ***n*** = ***k*** +1.

$$\sum_{j=1}^{k}\frac{1}{(2j-1)(2j+1)}=\frac{k}{2k+1}$$ **induction hypothesis**

$$\sum_{j=1}^{k+1}\frac{1}{(2j-1)(2j+1)}=\sum_{j=1}^{k}\frac{1}{(2j-1)(2j+1)}+\frac{1}{[2(k+1)-1][2(k+1)+1]}$$

$$=\frac{k}{2k+1}+\frac{1}{[2(k+1)-1][2(k+1)+1]}$$

$$=\frac{k}{2k+1}+\frac{1}{(2k+1)(2k+3)}=\frac{1}{2k+1}\left[k+\frac{1}{2k+3}\right]$$

$$=\frac{1}{2k+1}\left[\frac{2k^2+\boxed{3k}+1}{2k+3}\right]=\frac{1}{2k+1}\left[\frac{2k^2+\boxed{2k+k}+1}{2k+3}\right]$$

$$=\frac{1}{2k+1}\left[\frac{2k\boxed{(k+1)}+\boxed{(k+1)}}{2k+3}\right]=\cancel{\frac{1}{2k+1}}\cdot\frac{\cancel{(2k+1)}(k+1)}{2k+3}$$

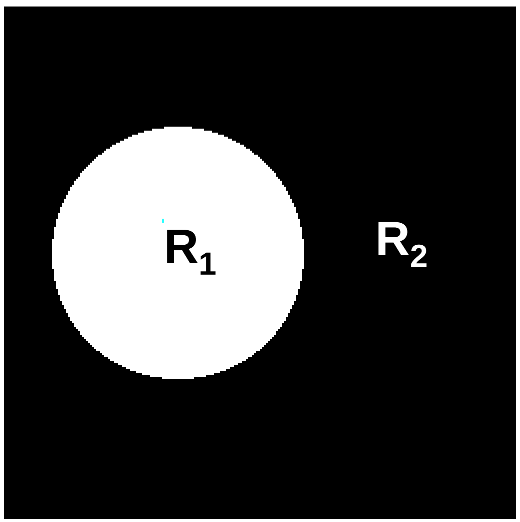
$$=\frac{k+1}{2k+3}=\frac{\boxed{k+1}}{2\boxed{(k+1)}+1}$$

and this is the right side for ***n*** = ***k*** +1.

- Show that ***n*** circles divide the plane into $\boldsymbol{n^2 - n + 2}$ regions if every two circles *intersect in exactly two points* and *no three circles contain a common point*.

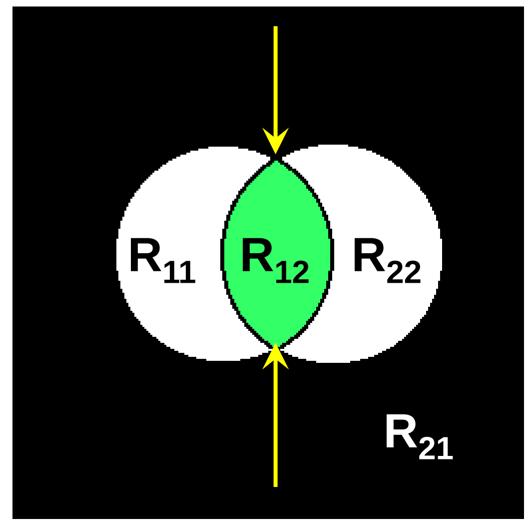
To answer this problem we combine induction with geometrical reasoning.

$n = 1$ (one circle)



$1^2 - 1 + 2 = 2\,;\{R_1, R_2\}$

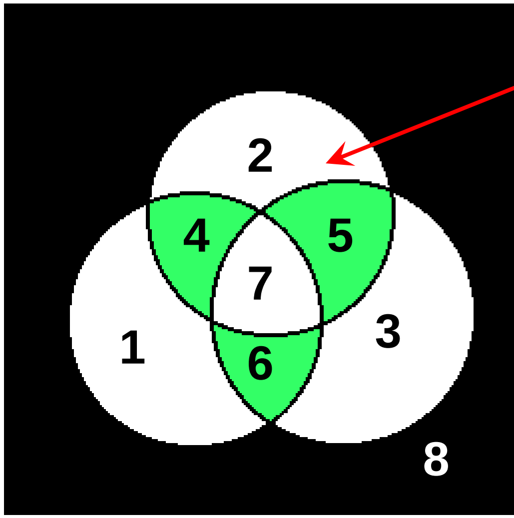
$n = 2$ (two circles)



$2^2 - 2 + 2 = 4\,;\{R_{11}, R_{12}, R_{21}, R_{22}\}$

Introducing a 2nd circle splits each existing region, so for $\boldsymbol{n = 2}$ we have two new additional regions or $\boldsymbol{2k = 2.1 = 2}$ where $\boldsymbol{k = 1}$ is the previous # of circles.

$n = 3$ (three circles)

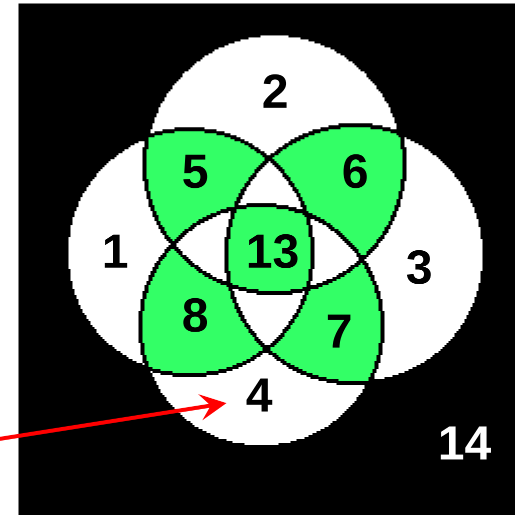


$3^2 - 3 + 2 = 8$

Introducing the 3rd circle splits again previous regions; it generates **2,4,5,7**.

Introducing the 4th circle splits again previous regions; it generates **4,7,8,9,11,13**.

$n = 4$ (four circles)



$4^2 - 4 + 2 = 14$

Induction hypothesis: ***k*** circles under the stated conditions divide the plane into $\boldsymbol{k^2 - k + 2}$ regions. If we introduce the $\boldsymbol{k+1}$ circle it will generate $\boldsymbol{2k}$ additional regions. Therefore,

$$k + 1 \text{ circles divide the plane into } (k^2 - k + 2) + 2k = k^2 + 2k + 1 - k + 1$$

$$= (k+1)^2 - k + 1 - 1 + 1 = \boxed{(k+1)^2 - (k+1) + 2}$$

The principle or method of mathematical induction has a second form that uses the *same basis step* but *modifies the inductive step* as follows:

Induction hypothesis

$$P(0) \wedge [\boxed{P(0) \wedge \ldots \wedge P(k)} \rightarrow P(k+1)] \therefore \forall n\, P(n)$$

basis step inductive step conclusion

- The **induction step** now assumes the truth of all values less than or equal to ***k***,

$$P(m) = T\,;\, m = 0, \ldots, k$$

- Note that if we use *modus ponens* we have only that ***P***(***k***) = **T**, so 1st form of mathematical induction results. *Both forms of the principle are equivalent*.

- This 2nd form is helpful when we need several previous instances of ***P***(***k***) to be true in order to show the truth of ***P***(***k***+1).

Prove the next proposition using the 2nd form of mathematical induction.

$$\forall n \geq 3,\ f_n > \alpha^{n-2}\ ;\ \alpha = \frac{1+\sqrt{5}}{2}$$

"golden ratio or divine proportion"

$$\frac{l}{h} = \alpha$$

- in the basis step it is shown that, $(f_3 = 2 > \alpha) \wedge (f_4 = 3 > \alpha^2)$

- besides using the modified inductive step (based on 2nd form of m.i.), we have that (looks like a trick but it is not),

$$\alpha^2 - \alpha - 1 = 0 \rightarrow \alpha^2 = \alpha + 1$$

- this "trick" is justified since a quadratic equation is associated with the recursive definition of $\boldsymbol{f}_n$, that is to say,

$$f_n - f_{n-1} - f_{n-2} = 0 \ \Rightarrow\ x^2 - x - 1 = 0$$

The number of divisions used by the Euclidean algorithm to find **gcd**(***a***,***b***) is less than or equal to five times the number of decimal digits in ***b*** when ***a*** ≥ ***b***.

- after ***n*** divisions it is shown that, $b \geq f_{n+1}$
- from the previous slide, if ***n*** > **2** then $f_{n+1} > \alpha^{n-1} \rightarrow b > \alpha^{n-1}$
- taking common logarithms (base 10) to both sides of the last inequality,

$$\log_{10} b > (n-1)\log_{10}\alpha \approx 0.208(n-1) > (n-1)/5$$

- if the number ***b*** has ***k*** decimal digits then

$$b < 10^k \rightarrow (n-1) < 5k \rightarrow n \leq 5k$$

- the number of decimal digits in ***b*** is calculated as

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

# of divisions:

$$n_d \leq 5 \cdot (\log_{10} b + 1) \approx O(\log b)$$

Mathematical reasoning: Part III

- Recursive definitions
- Recursive sets
- Recursive algorithms
- Iterative algorithms

Certain objects such as functions, sequences, and sets can be defined in two different ways:

- direct or explicit, the object is defined by a specific expression *that does not depend on the object itself*,
- recursive or implicit, the object is defined by an expression *that includes the same object*.

$$f(n) = \boxed{\prod_{k=1}^{n} k} = \left(\prod_{k=1}^{n-1} k\right) \cdot n = f(n-1)\cdot n \to n! = n(n-1)!$$

factorial function

$$s(n) = \boxed{\sum_{k=1}^{n} k} = \left(\sum_{k=1}^{n-1} k\right) + n = s(n-1) + n \to n? = n + (n-1)?$$

termial function

$$p(n) = \boxed{a^n} = a^{n-1} \cdot a = p(n-1)\cdot a$$

power function, ***a*** > 1

In the context of integer numbers it is usual to exchange notations between functions and sequences (recall that a sequence is a type of function).

$$f(n) = f_n \, ; n \in S \subseteq N$$

Direct definition

$$f_n = \varphi(n) \, ; \varphi \neq f$$

The general term of the sequence is given by an expression that depends only on the index ***n*** and does not require initial values.

Recursive definition

$$f_n = \psi(f_{n-1}, f_{n-2}, \ldots, f_{n-k}) \, ; \psi \neq f$$
$\{f_0, \ldots, f_{k-1}\}$ are initial values.

The general term of the sequence is given by an expression that depends on previous values of the same sequence and requires the knowledge of the first ***k*** terms.

Recursive functions examples[a]

$$\boxed{a_n = 4n-2} \rightarrow a_{n-1} = 4(n-1)-2 = 4n-2-4$$

we take the difference, so

$$a_n - a_{n-1} = 4 \rightarrow \boxed{a_n = a_{n-1} + 4\,;\, a_0 = -2}$$

$$\boxed{a_n = 1+(-1)^n} \rightarrow a_{n-1} = 1+(-1)^{n-1} = 1-(-1)^n$$

adding both expressions,

$$a_n + a_{n-1} = 2 \rightarrow \boxed{a_n = 2 - a_{n-1}\,;\, a_0 = 2}$$

$$\boxed{f_n = f_{n-1} + f_{n-2}\,;\, f_0 = 1,\, f_1 = 1}$$

note that two initial values are needed to compute the next terms of the sequence.

$$f_2 = f_1 + f_0 = 1+1 = 2$$

$$f_3 = f_2 + f_1 = 2+1 = 3$$

$$f_4 = f_3 + f_2 = 3+2 = 5$$

$$f_5 = f_4 + f_3 = 5+3 = 8$$

$$\longrightarrow \boxed{f_n = \{1,1,2,3,5,8,13,21,\ldots\}}$$

known as the Fibonacci numbers.

- The **McCarthy 91** function is defined using the rule,

$$M(n) = \begin{cases} n-10 & ; n > 100 \\ M(M(n+11)) & ; n \le 100 \end{cases}$$

for all positive integers ***n***. By successively using the defining rule for ***M***(***n***), find a) ***M***(**102**), b) ***M***(**101**), c) ***M***(**99**), and d) ***M***(**97**).

$$\text{a) } M(102) = 102 - 10 = 92 \text{ since } 102 > 100$$

$$\text{b) } M(101) = 101 - 10 = \boxed{91} \text{ since } 101 > 100$$

$$\text{c) } M(99) = M(M(99+11)) \text{ since } 99 \le 100$$

$$= M(M(110)) = M(110-10) = M(100)$$

$$= M(M(100+11) \text{ since } 100 \le 100$$

$$= M(M(111)) = M(111-10) = M(101) = \boxed{91}$$

$$\text{d) } M(97) = M(M(97+11)) = M(M(108)) = M(98)$$

$$M(98) = M(M(98+11)) = M(M(109)) = M(99) = \boxed{91}$$

Thus, from b) to d)

$$\boxed{\forall n \le 101,\ M(n) = 91}$$

As in the case of functions and sequences, a set ***S*** of objects can also be defined recursively by performing two steps.

1.- Initial element, in this step a specific element ***x*** (or elements) is defined to belong to ***S***,

2.- Generation, in this step the rest of the elements in ***S*** is generated by means of a *rule* or a *procedure* to combine previous elements (initial element).

$$\begin{array}{l} x \in S \\ y \in S \to r(y) \in S \end{array}$$

$$\begin{array}{l} x, y \in S \\ z, w \in S \to p(w,z) \in S \end{array}$$

In fact, **mathematical induction** can be taken as a *recursive definition of the natural numbers* if ***P*** is the identity predicate, ***P***(***k***) = ***k***.

$$\left.\begin{array}{l} 0 \in S \\ k \in S \to k+1 \in S \end{array}\right\} \Rightarrow S = N$$

A recursive definition for the set of positive integers powers of **3**:

$$\left.\begin{array}{l} 3 \in S \\ k \in S \rightarrow 3 \cdot k \in S \end{array}\right\} \Rightarrow \boxed{S = \{3^n | n \in Z^+\}}$$

The set ***C*** of well-formed formulae (**wff**) for compound propositions is defined recursively as follows:

$$\left.\begin{array}{l} T, F, p, q \in C \\ p, q \in C \rightarrow \sim p, p \vee q, p \wedge q, p \rightarrow q, p \leftrightarrow q \in C \end{array}\right\} \Rightarrow \boxed{\text{wff}}$$

The set of strings **Σ*** over the finite alphabet Σ:

$$\left.\begin{array}{l} \lambda \in \Sigma^* \text{ (empty string)} \\ (w \in \Sigma^*) \wedge (x \in \Sigma) \rightarrow wx \in \Sigma^* \end{array}\right\}$$

$$\boxed{\Sigma = \{0,1\} \rightarrow \Sigma^* = \{0,1,00,01,111,\ldots\}}$$

The length of a string can be defined as:

$$\left.\begin{array}{l} l(\lambda) = 0 \\ (w \in \Sigma^*) \wedge (x \in \Sigma) \rightarrow l(wx) = l(w) + 1 \end{array}\right\}$$

Reasoning-III Recursive sets examples[b]

- Let ***S*** be the set of strings defined recursively by **abc** ∈ ***S***, **bac** ∈ ***S***, **acb** ∈ ***S***, and **abcx** ∈ ***S***; also, **abxc** ∈ ***S***, **axbc** ∈ ***S***, **xabc** ∈ ***S*** if **x** ∈ ***S***.
  a) Find all elements of ***S*** of length eight or less,
  b) Show that every element of ***S*** has a length divisible by three.

$l(w)=3$ $\quad$ $l(w)=6$ $\quad$ $l(w)=6$

$$\left.\begin{matrix} abc \\ bac \\ acb \end{matrix}\right\} \boxed{abcx} \quad \left\langle \begin{matrix} \boxed{abc \mid abc} \\ abc \mid bac \\ abc \mid acb \end{matrix} \right\rangle \quad \left\langle \begin{matrix} abxc \to \\ axbc \to \\ xabc \to \end{matrix} \left| \begin{matrix} ab|abc|c,\ ab|bac|c,\ \boxed{ab|acb|c} \\ a|abc|bc,\ \boxed{a|bac|bc},\ a|acb|bc \\ \boxed{abc|abc},\ bac|abc,\ acb|abc \end{matrix} \right. \right\rangle$$

3 strings $\quad$ 3 strings $\quad$ 9 strings $\quad$ **Total = 15 - 2 = 13**

The proof of b) is by induction on the length of the string. The basis step is part a), for the inductive step, assume that the length of a string **x** in ***S*** is a multiple of **3**. Then, the following *generated strings* have a length that is a multiple of **3**, i.e.,

$$l(abcx)=l(abxc)=l(axbc)=l(xabc)=3+l(x)=3+3k=3m.$$

A recursive algorithm is an algorithm that calls itself using a set of initial values.

Recursive algorithm: ***n***-th *Fibonacci number*

procedure $fib(n \in N)$

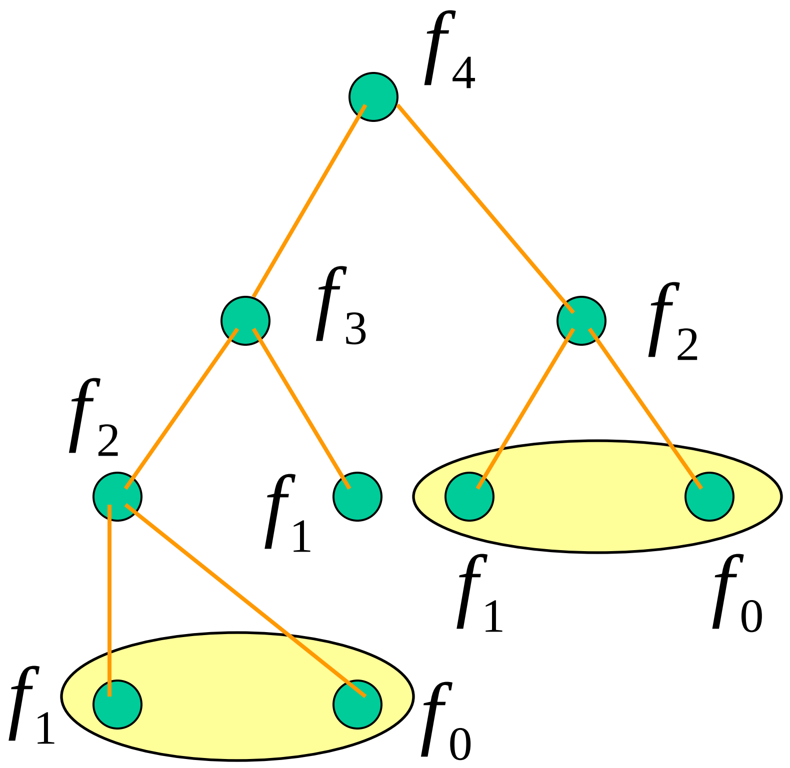
if $n = 0$

then $fib(0) := 0$

else if $n = 1$

then $fib(1) := 1$

else $fib(n) := fib(n-1) + fib(n-2)$



Features of recursion:

- works from top to bottom reducing input size,
- needs additional memory to store partial calls,
- inefficient in terms of time complexity,
- easy to understand and compact pseudocode,
- a recursive definition can be translated in a recursive algorithm.

# of additions:

$$f_{n+1} - 1$$

Recursive algorithms examples

Recursive algorithm: greatest common divisor

$$\textbf{procedure} \ \gcd(a, b \in N)$$
$$\{\text{assume } b < a\}$$
$$\textbf{if } b = 0$$
$$\quad \textbf{then} \ \gcd(a,b) := a$$
$$\quad \textbf{else} \ \gcd(a,b) := \gcd(b, a \bmod b)$$

Input size reduction:

$$a \bmod b < b$$

Recursive algorithm: binary search

$$\textbf{procedure} \ bs(x, i, j)$$
$$m := \lfloor (i + j)/2 \rfloor$$
$$\textbf{case} \ x = a_m$$
$$\quad location := m$$
$$\textbf{case} \ (x < a_m) \wedge (i < m)$$
$$\quad bs(x, i, m-1)$$
$$\textbf{case} \ (x > a_m) \wedge (j > m)$$
$$\quad bs(x, m+1, j)$$
$$\textbf{otherwise}$$
$$\quad location := 0$$

Input size reduction: $m < j$

An iterative algorithm is an algorithm that does not call itself. However, it can be based on a recursive definition using its initial values to find the next value.

Iterative algorithm: ***n***-th *Fibonacci number*

$$
\begin{array}{l}
\textbf{procedure } fib(n \in N) \\
\textbf{if } n = 0 \\
\quad \textbf{then } y := 0 \\
\quad \textbf{else } x := 0 \\
\qquad y := 1 \\
\qquad \textbf{for } i := 1 \textbf{ to } n-1 \\
\qquad\quad z := x + y \\
\qquad\quad x := y \\
\qquad\quad y := z \\
\{y = f_n\}
\end{array}
$$

Features of iteration:

- works from bottom to top,
- needs few memory to store values,
- efficient in terms of time complexity,
- longer pseudocode,
- a recursive definition can be used to design it.

\# of additions:

$$\boxed{n-1;\, n \geq 1}$$

It is clear that, $\boxed{\forall n > 3, n-1 \leq f_{n+1} - 1}$

so the *iterative version performs better than the recursive version for large values of* ***n***.

Reasoning-III Recursive vs iterative: about memory

procedure $fact(n \in N)$

if $n = 0$

then $fact(n) := 1$

else $fact(n) := n * fact(n-1)$

n = 4

Frames	**Locations**
4! = 4 . 3!	1+2
3! = 3 . 2!	1+2
2! = 2 . 1!	1+2
1! = 1	1+1
4	**11**

Faster in stack oriented machines.

Recursive vs. iterative factorial

procedure $fact(n \in N)$

$x := 1$

for $i := 1$ **to** n

$x := i * x$

$\{x = n!\}$

n = 4

Vars	**Locations**
x = 1, i.x	1
i = 1, n	1
2	**2**

Faster in register oriented machines.

Counting: Part I

- Sum and product rules
- Inclusion-exclusion
- Tree diagrams
- Pigeonhole principle

The basic principles for counting are based on the corresponding laws between cardinals for families of sets (both finite).

The sum rule: suppose that the tasks $\boldsymbol{T}_1,\boldsymbol{T}_2,\ldots,\boldsymbol{T}_m$ can be done in $\boldsymbol{n}_1,\boldsymbol{n}_2,\ldots,\boldsymbol{n}_m$ ways respectively, and *no two of these tasks can be done at the same time*. Then the number of ways to do task $\boldsymbol{T}_1$ or $\boldsymbol{T}_2$ or … or $\boldsymbol{T}_m$ is $\boldsymbol{n}_1 + \boldsymbol{n}_2 + \ldots + \boldsymbol{n}_m$.

Set interpretation

Let $\boldsymbol{T}_i$ be the task of choosing an element from set $\boldsymbol{A}_i$, then there are $|\boldsymbol{A}_i| = \boldsymbol{n}_i$ ways to do $\boldsymbol{T}_i$. For $\boldsymbol{m}$ sets the expression is:

$$\left|\bigcup_{i=1}^{m} A_i\right| = \sum_{i=1}^{m} |A_i|;\ \forall i \neq j, A_i \cap A_j = \varnothing$$

$$\forall i, |A_i| = n \rightarrow \sum_{i=1}^{m} |A_i| = n \cdot m$$

Pseudocode interpretation

Let $\boldsymbol{T}_i$ be the task of traversing the independent $\boldsymbol{i}$-th loop, then there are $\boldsymbol{n}_i$ ways to do $\boldsymbol{T}_i$. For $\boldsymbol{m}$ loops the pseudocode is:

$k := 0$

for $i_1 := 1$ **to** n_1 ; $k := k + 1$

for $i_2 := 1$ **to** n_2 ; $k := k + 1$

$\vdots$

for $i_m := 1$ **to** n_m ; $k := k + 1$

Sum rule examples

There are **18** mathematics majors and **325** computer science majors at a college. How many ways are there to pick one representative who is either a mathematics major or a computer science major.

T_1 is the task of selecting a mathematics major,
T_2 is the task of selecting a computer science major, therefore

# of ways to do T_1 or T_2 is **18** + **325** = **343**.

How many ways are there to choose a symbol from capital or lowercase letters, or a decimal digit?

T_1 is the task of selecting a capital letter,
T_2 is the task of selecting a lowercase letter,
T_3 is the task of selecting a decimal digit, therefore,

# of ways to do T_1 or T_2 or T_3 is **26** + **26** + **10** = **62**.

In most applications, the sum rule is used together with the product rule to count objects satisfying certain conditions.

The product rule: suppose that a procedure is carried out by performing $T_1, T_2, \ldots, T_m$ tasks. If task T_i can be done in n_i ways *after* tasks $T_1, T_2, \ldots, T_{i-1}$ have been done, then there are $n_1\, n_2 \ldots n_m$ ways to carry out the procedure.

Set interpretation

Let T_i be the task of choosing an element from set A_i, then there are $|A_i| = n_i$ ways to do T_i.
For m sets the expression is:

$$|\bigtimes_{i=1}^{m} A_i| = \prod_{i=1}^{m} |A_i|;\ \text{cartesian product}$$

$$\forall i, |A_i| = n \rightarrow \prod_{i=1}^{m} |A_i| = n^m$$

number of ways of selecting an ordered sequence $(a_1, a_2, \ldots, a_m)$

Pseudocode interpretation

Let T_i be the task of traversing the nested i-th loop, then there are n_i ways to do T_i.
For m loops the pseudocode is:

$k := 0$
for $i_1 := 1$ **to** n_1
  for $i_2 := 1$ **to** n_2
    $\vdots$
      for $i_m := 1$ **to** n_m
        $k := k + 1$

How many different bit strings are there of length **10**?

$\boldsymbol{T}_{\mathrm{i}}$ is the task of selecting a bit value for the ***i***-th position in the string, for each ***i*** the number of ways is **2** (0 or 1), therefore the number of strings is,

$$\underbrace{2\cdot 2\cdots 2}_{10\,\text{positions}} = 2^{10} = 1024 \quad \text{or} \quad \boxed{n^m\ ;\ n=2, m=10}$$

How many injections are there from a set ***A*** with ***m*** elements to a set ***B*** with ***n*** elements if ***m*** ≤ ***n***?

$$\begin{array}{llll} \{b_1, b_2, \ldots, b_n\} & \{b_2, \ldots, b_n\} & \{b_3, \ldots, b_n\} & \{b_m, \ldots, b_n\} \\ \Uparrow & \Uparrow & \Uparrow & \cdots\Uparrow \\ a_1 \to \text{one of } n & a_2 \to \text{one of } n-1 & a_3 \to \text{one of } n-2 & a_m \to \text{one of } n-(m-1) \end{array}$$

So, the number of one-to-one functions is, $\boxed{n(n-1)(n-2)\cdots(n-m+1)}$

How many different license plates (d.l.p.) are available if each plate contains a sequence of *three letters* followed by *three digits*?

The format of a license plate is **LLLDDD**, thus there are

$$26^3 \cdot 10^3 = \boxed{17{,}576{,}000 \ \text{ d.l.p.}}$$

Each user on a computer system has a password, which is **6** to **8** characters long, where each character is an uppercase letter or digit. Each password must *contain at least one digit*. How many possible passwords are there?

- For a password 6 characters long (combining sum and product rules):

$$\left.\begin{array}{ll} \text{CCCCCC} & \to (26+10)^6 = 36^6 \\ \text{LLLLLL} & \to 26^6 \end{array}\right\rangle \to P_6 = 36^6 - 26^6 = \boxed{1{,}867{,}866{,}560}$$

- For a password 7 characters long:

$$\left.\begin{array}{ll} \text{CCCCCCC} & \to (26+10)^7 = 36^7 \\ \text{LLLLLLL} & \to 26^7 \end{array}\right\rangle \to P_7 = 36^7 - 26^7 = \boxed{70{,}332{,}353{,}920}$$

- For a password 8 characters long:

$$\left.\begin{array}{ll} \text{CCCCCCCC} & \to (26+10)^8 = 36^8 \\ \text{LLLLLLLL} & \to 26^8 \end{array}\right\rangle \to P_8 = 36^8 - 26^8 = \boxed{2{,}612{,}282{,}842{,}880}$$

Thus, the # of passwords is $P = P_6 + P_7 + P_8 = 2{,}684{,}483{,}063{,}360 \approx \boxed{2.7 \times 10^{12}}$

- How many positive integers with *exactly* **4 digits** between **1000** and **9999** *inclusive*,

a) are divisible by **9**? $999 < 9k \le 9999 \to 111 < k \le 1111 \to 1111 - 111 = \boxed{1000}$

b) are **even**? $9999 - 1000 + 1 = 9000 / 2 = \boxed{4500}$

c) have **distinct** digits? $dddd \to (10-1)(9)(8)(7) = 9^2 \cdot 8 \cdot 7 = \boxed{4536}$

d) are *not* divisible by **3**? $999 < 3k \le 9999 \to 333 < k \le 3333 \to 3333 - 333 = 3000$

$\to 9000 - 3000 = \boxed{6000}$

e) are divisible by **5 or 7**? $1000 \le 5k < 10000 \to 200 < k \le 2000 \to 2000 - 200 = 1800$

$$\left.\begin{array}{l} \lfloor 1000/7 \rfloor = 142 \to 7 \cdot 142 = 994 \\ \lfloor 10000/7 \rfloor = 1428 \to 7 \cdot 1428 = 9996 \end{array}\right\} \to 1428 - 142 = 1286$$

$$\left.\begin{array}{l} \lfloor 1000/35 \rfloor = 28 \to 35 \cdot 28 = 980 \\ \lfloor 10000/35 \rfloor = 285 \to 35 \cdot 285 = 9975 \end{array}\right\} \to 285 - 28 = 257$$

$\to 1800 + 1286 - 257 = \boxed{2829}$

f) are *not* divisible by either **5 or 7**? $\rightarrow 9000 - 2829 = \boxed{6171}$

g) are divisible by **5 but not by 7**? $\rightarrow 1800 - 257 = \boxed{1543}$

h) are divisible by **5 and 7**? $\rightarrow \boxed{257}$

- How many *different functions* are there from a set with **10** elements to sets with the following number of elements?

In general, for $f : A \rightarrow B$ when $|A| = n \wedge |B| = m, \boxed{\#(f) = m^n}$

a) **2** $\rightarrow m = 2, n = 10 \rightarrow \#(f) = 2^{10} = \boxed{1024}$

b) **3** $\rightarrow m = 3, n = 10 \rightarrow \#(f) = 3^{10} = \boxed{59049}$

c) **4** $\rightarrow m = 4, n = 10 \rightarrow \#(f) = 4^{10} = 2^{20} = 1024^2 = \boxed{1048576}$

d) **5** $\rightarrow m = 5, n = 10 \rightarrow \#(f) = 5^{10} = \boxed{9765625}$

The principle of inclusion-exclusion is applied to situations in which two or more tasks can be *realized at the same time* and we need to count the number or ways to do one of these tasks.

For 2 tasks (in terms of sets): $\boxed{|A \cup B| = |A| + |B| - |A \cap B|;\ A \cap B \neq \varnothing}$

How many bit strings of length **8** either start with a 1 bit or end with the two bits 00?

Task 1 (set ***A***), string format starting with **1**, **1**BBBBBBB $\rightarrow 2^7 = \boxed{128 \text{ strings}}$

Task 2 (set ***B***), string format ending with **00**, BBBBBB**00** $\rightarrow 2^6 = \boxed{64 \text{ strings}}$

Common task (***A*** and ***B***), string format, **1**BBBBB**00** $\rightarrow 2^5 = \boxed{32 \text{ strings}}$

Hence, applying inclusion-exclusion, $|A \cup B| = 128 + 64 - 32 = \boxed{160}$

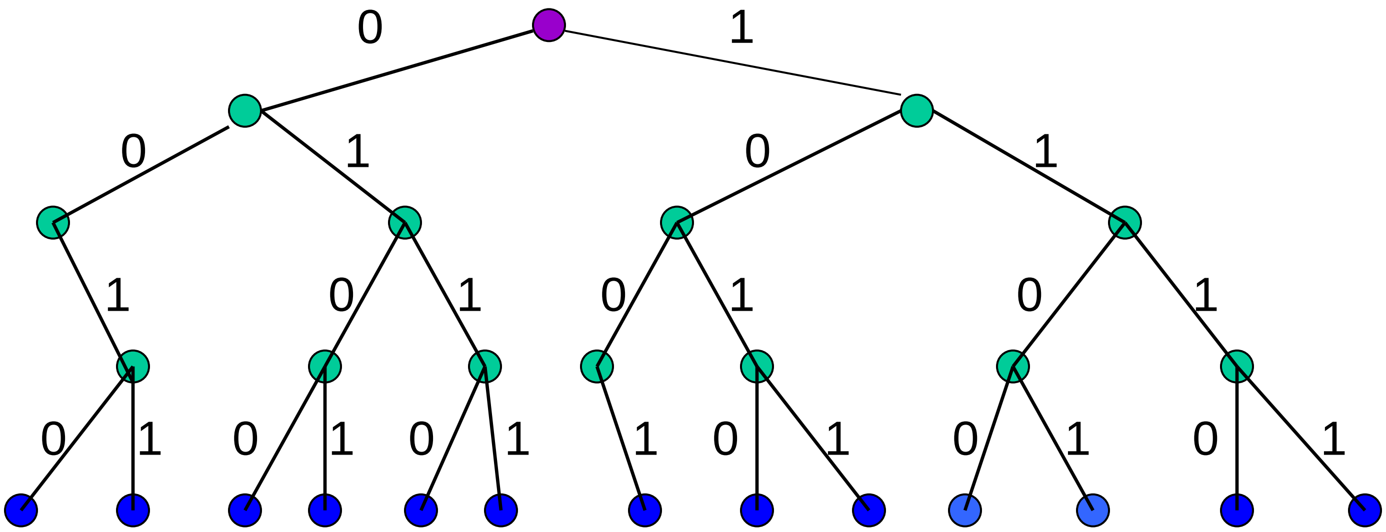
For 3 tasks (in terms of sets):

$$\boxed{\begin{aligned} |A \cup B \cup C| = |A| + |B| + |C| & \\ -|A \cap B| - |A \cap C| - |B \cap C| & \\ +|A \cap B \cap C| & \end{aligned}}$$

A tree is a graphical object that has a *root*, a number of *branches* leaving the root, and possible *additional branches* leaving the endpoints of other branches.

In the context of counting, we use a branch to represent each possible choice, and the *leaves* of the tree represent the outcomes (endpoints of the tree).
Trees are useful for modeling problems with small values.

Use a tree diagram to find the number of bit strings of length 4 with no 3 consecutive 0's.



The number of strings is the number of leaves at the end of the tree = 13.

Pigeonhole principle

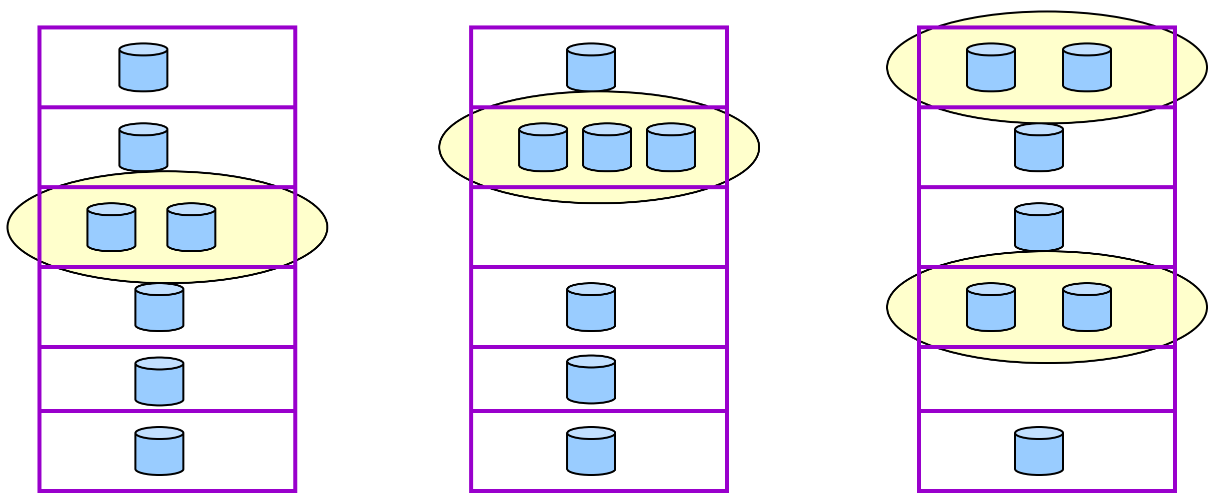
The pigeonhole principle in its simplest form states that if there are more pigeons than pigeonholes, then there must be at least one pigeonhole with at least two pigeons in it.

The formal mathematical statement is known as the Dirichlet drawer principle.

If ***k*** + **1** or more objects are placed into ***k*** boxes, then there is at least one box containing two or more of the objects.

$k = 6$ boxes

$k + 1 = 7$ objects



2 objects in box 3, drawer is full.

3 objects in box 2, nothing in box 3.

2 objects in boxes 1,4; no objects in box 5.

Pigeonhole examples[a]

To apply the pigeonhole principle, it is very important to identify which are the objects and which are the boxes.

- Show that if there are **30** students in a class, then at least **2** have last names that begin with the same letter.

The students correspond to the "objects", the letters to the "boxes", then there are **26** boxes and **30** objects where **30** > **26**, thus by the pigeonhole principle there is a box with at least **2** objects, i.e., **2** names begin with the same letter.

- Show that if $\boldsymbol{f}$ is a function from $\boldsymbol{S}$ to $\boldsymbol{T}$ where $\boldsymbol{S}$ and $\boldsymbol{T}$ are finite sets with $|\boldsymbol{S}|>|\boldsymbol{T}|$, then there are elements $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ in $\boldsymbol{S}$ such that $\boldsymbol{f}(\boldsymbol{s}_1) = \boldsymbol{f}(\boldsymbol{s}_2)$, or in other words, $\boldsymbol{f}$ is not one-to-one.

The elements of $\boldsymbol{S}$ are the "objects", their images the "boxes"; since $|\boldsymbol{S}|>|\boldsymbol{T}|$ (more objects than boxes) by the pigeonhole principle. there is an element of $\boldsymbol{T}$ that is the image of at least two elements of $\boldsymbol{S}$, hence $\boldsymbol{f}$ is not one-to-one.

Show that among any ***n*** + **1** positive integers not exceeding **2*n*** there must be an integer that divides one of the other integers.

First, represent the ***n*** + **1** integers as follows:

$$\{a_j\}_{j=1}^{n+1} \to a_j = 2^{k_j} q_j \,;\, (k_j \ge 0) \wedge (q_j \text{ is odd})$$

by construction, $q_j \le 2n \,;\; j = 1,2,\ldots,n+1$ (objects)

also, there are ***n*** odd numbers (boxes) less than **2*n*** (the other ***n*** are even).

From the pigeonhole principle, two of the integers q_j must be equal, i.e.,

$$\exists i \exists j, (i \ne j \wedge q_i = q_j = q) \to (a_i = 2^{k_i} q) \wedge (a_j = 2^{k_j} q)$$

Then,

$$k_i < k_j \to 2^{k_i} \mid 2^{k_j} \to a_i \mid a_j$$

$$k_i > k_j \to 2^{k_j} \mid 2^{k_i} \to a_j \mid a_i$$

Theorem: Every sequence of ***n*²** **+ 1** distinct real numbers contains a subsequence of length ***n*** **+ 1** that is either *strictly increasing* or *strictly decreasing*.

Let $\{a_k\}_{k=1}^{n^2+1} = \{a_1, a_2, \ldots, a_{n^2}, a_{n^2+1}\}$ where $\forall k \neq l,\ a_k \neq a_l$

To each term of the sequence associate an ordered pair as follows

$$\boxed{a_k \leftarrow (i_k, d_k)} \quad \begin{cases} i_k & \text{length of longest increasing subsequence} \\ d_k & \text{length of longest decreasing subsequence} \end{cases} \quad \{a_k, a_{k+1}, \ldots\}$$

Note that the total number of ordered pairs is ***n*²** **+ 1** (objects)

By **contradiction**, there is *no strictly increasing sequence* and there is *no strictly decreasing sequence* of length ***n*** **+ 1**.

$$\rightarrow i_k \leq n < n+1 \text{ and } d_k \leq n < n+1\,;\ k = 1, \ldots, n^2+1$$

Now, by the product rule, the number of length pairs is at most ***n*²** (boxes)

Consequently, applying the pigeonhole principle, there are at least two ordered pairs in the same box, i.e.,

$$\exists s \neq t,\ (i_s, d_s) = (i_t, d_t)$$ Not possible!

Since, by assumption, the terms of the sequence are distinct, then

$$a_s \neq a_t \rightarrow (a_s < a_t) \vee (a_s > a_t)$$

By cases,

$$(a_s < a_t) \rightarrow a_s < a_t < \ldots < a_{t+i_t-1} \rightarrow i_s = 1 + i_t \rightarrow \boxed{i_s > i_t}$$

$$(a_s > a_t) \rightarrow a_s > a_t > \ldots > a_{t+d_t-1} \rightarrow d_s = 1 + d_t \rightarrow \boxed{d_s > d_t}$$

Generalized pigeonhole principle

There can be a number of objects greater than a multiple of the number of boxes. So the generalized pigeonhole principle (g.p.p.) states the following:

If ***n*** objects are placed into ***k*** boxes, then there is at least one box containing at least $\lceil n/k \rceil$ objects.

- There are **38** different time periods during which classes at a university can be scheduled. If there are **677** different classes, how many different rooms will be needed?

  The classes are the "objects", the time periods are the "boxes"; note that **677** is greater than some multiple of **38**, hence using the g.p.p. the number of rooms needed is $\lceil 677/38 \rceil = 18$

- Show that there are at least **4** people in California (population: 25 million) with the same three initials who were born on the same day of the year.

The number of ways of selecting the **3** initials is $\mathbf{26}^3$, also there are **366** possible birthdays (counting leap years). Hence the number of "boxes" is $\mathbf{26}^3$ x **366** = **6432816**, therefore applying the g.p.p. the answer is computed as

$$\lceil 25000000 \;/\; 6432816 \rceil = 4$$

- Show that if **7** integers are selected from the first **10** positive integers, there must be at least two pairs of these integers with the sum 11.

We must first define from the context of the problem which are the "objects" and which are the "boxes". The pairs whose sum is 11 are the boxes, i.e.,

$$\boxed{\{1,10\},\{2,9\},\{3,8\},\{4,7\},\{5,6\}}$$

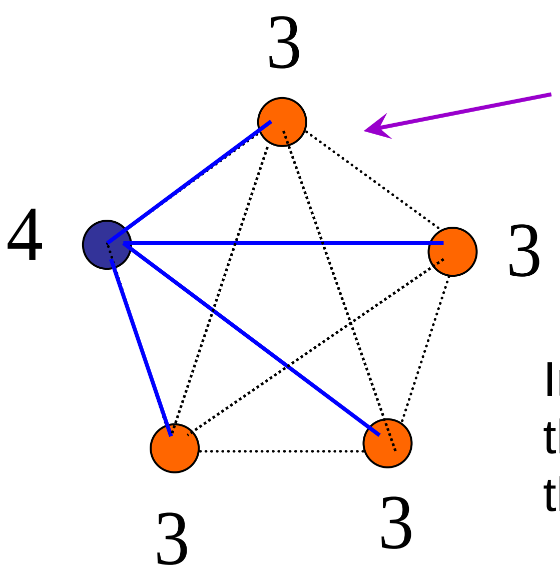
Note that all **10** numbers are in these **5** boxes; the 7 integers are the objects and **7 > 5**, hence by the g.p.p. there are at least $\lceil 7/5 \rceil = 2$ two integers in a box. Now we have 4 boxes and 5 integers, again **5 > 4**, then by the same principle there is another box containing two integers.

- How many ordered pairs of integers $(\boldsymbol{a},\boldsymbol{b})$ are needed to guarantee that there are two pairs $(\boldsymbol{a}_1,\boldsymbol{b}_1)$ and $(\boldsymbol{a}_2,\boldsymbol{b}_2)$ such that $\boldsymbol{a}_1 \textbf{ mod } \mathbf{5} = \boldsymbol{a}_2 \textbf{ mod } \mathbf{5}$ and $\boldsymbol{b}_1 \textbf{ mod } \mathbf{5} = \boldsymbol{b}_2 \textbf{ mod } \mathbf{5}$.

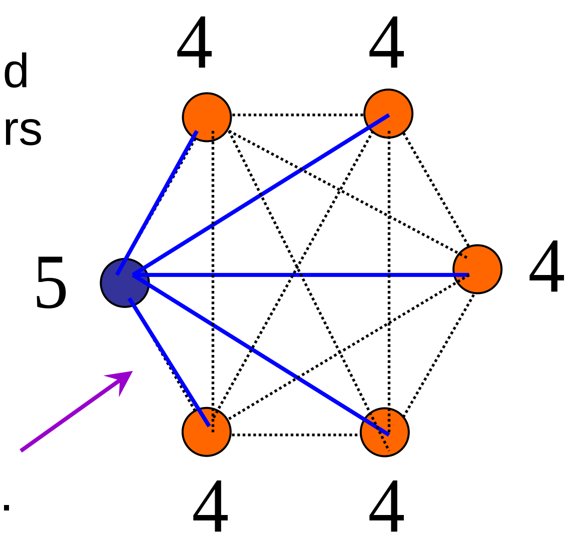
In each of the last equalities we have **5** possible remainders, **{0,1,2,3,4}**, since there are **2** equalities the number of pairs of remainders is **25**, therefore,

$$\lceil N/25 \rceil = 2 \rightarrow N \geq 26 \rightarrow N\,(\text{minimum}) = \boxed{26 \text{ pairs}}$$

- Show that in a group of **5** people (*where any two people are either friends or enemies*), there are not necessarily **3 mutual friends or 3 mutual enemies**. (**Ramsey theory**)



In this graphical model, there are **4** pairs/person and the rest are non-related pairs to the same person.



In this graphical model, there are **5** pairs/person and the rest are non-related pairs.

"**objects**" = pairs of **F⊕E**
"**boxes**" = mutual friends or mutual enemies

by the g.p.p. $\lceil 4/2 \rceil = \boxed{2}$

So, not necessarily 3 F's or 3 E's.

by the g.p.p. $\lceil 5/2 \rceil = \boxed{3}$

So, 3 F's or 3 E's.

Counting: Part II

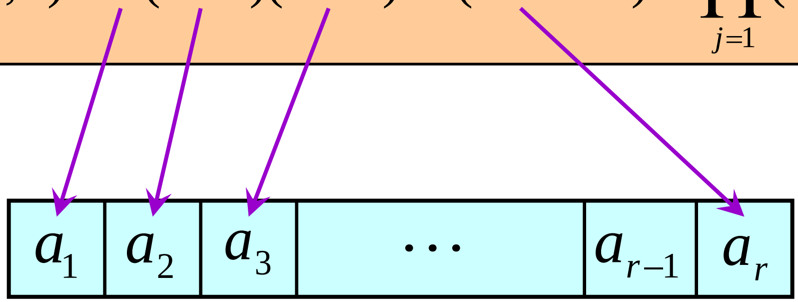
- Permutations
- Combinations
- Identities
- Binomial expansion

Permutations

A permutation of a set of distinct objects is an ordered arrangement of these objects. An ordered arrangement of ***r*** elements of a set is called an ***r**-permutation*.

The number of ***r***-permutations of a set with ***n*** distinct elements is

$$P(n,r)=n(n-1)(n-2)\cdots(n-r+1)=\prod_{j=1}^{r}(n-j+1)\,;1\le r\le n$$



Select ***r*** elements from a set with ***n*** elements; the first element, $\boldsymbol{a}_1$ can be selected in ***n*** ways, $\boldsymbol{a}_2$ can be selected in (**n** - **1**) ways until the last element $\boldsymbol{a}_r$, that can be selected among the remaining ***n*** - (***r*** - 1) elements. By the product rule the total number of permutations of size ***r*** is ***P***(***n***,***r***).

$$P(n,r)=\prod_{j=1}^{r}(n-j+1)\cdot\prod_{j=r+1}^{n}(n-j+1)/\prod_{j=r+1}^{n}(n-j+1)=\boxed{\frac{n!}{(n-r)!}}$$

Permutations examples[a]

- Specific cases of the formula ***P***(***n***,***r***),

$$P(n,1)=\frac{n!}{(n-1)!}=\frac{n(n-1)!}{(n-1)!}=n$$

$$P(n,n)=\frac{n!}{(n-n)!}=\frac{n!}{0!}=n!\,;\ 0!=1$$

- List all permutations of the set ***A*** = {***a***,***b***,***c***}.

Perm(***A***) = {(*a*,*b*,*c*),(*b*,*c*,*a*),(*c*,*a*,*b*),(*a*,*c*,*b*),(*b*,*a*,*c*),(*c*,*b*,*a*)}

$$P(3,3)=3!=\boxed{6}$$

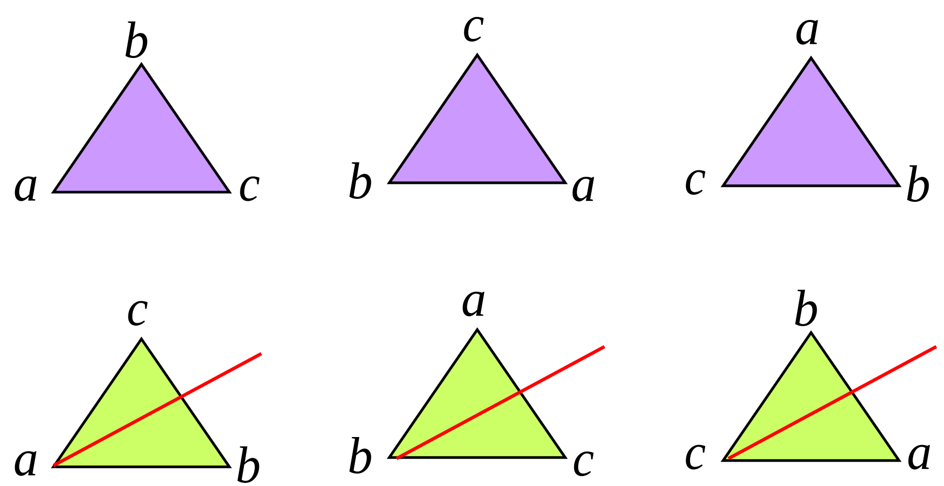
cyclic permutation = 120° rotation

$$(a,b,c)\rightarrow(b,c,a)\rightarrow(c,a,b)$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

$$(a,c,b) \qquad (b,a,c) \qquad (c,b,a)$$

transposition = flip about a vertex



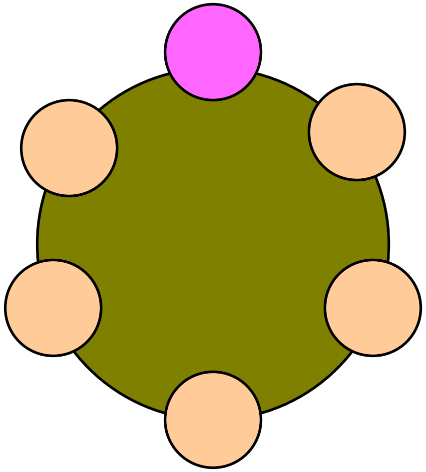
- Evaluate ***P***(**8**,**5**).

$$P(8,5)=8!/(8-5)!=8!/3!=8.7.6.5.4.3!/3!=8.7.6.5.4=\boxed{6720}$$

Permutations examples[b]

- How many ways are there to seat **6** people around a *circular table,* where seatings are considered to be the same if they can be obtained from each other by rotating the table?

fixed person



remaining persons

In general, for ***n*** people, one of them is the anchor, i.e., is fixed (note that we can choose any person), the rest can be arranged in (***n*** - **1**)! or ***P***(***n*** - **1**,***n*** - **1**) ways.

This kind of ordered arrangement (in a circle) is called circular permutation of size ***n***. For the present problem,

$$\rightarrow n = 6 \rightarrow P(5,5) = 5! = \boxed{120}$$

A combination of a set of distinct objects is an unordered selection of these objects. An ***r**-combination* is simply a subset with ***r*** elements.

The number of ***r***-combinations of a set with ***n*** distinct elements is

$$C(n,r) = \frac{n!}{r!(n-r)!}\ ;\ 0 \le r \le n$$

$$C(n,r) = \binom{n}{r}$$

binomial coefficient

Combinatorial proof: an ***r***-combination is just a set whose elements are selected from a set with ***n*** elements, its number is ***C***(***n***,***r***); on the other hand, there are ***P***(***r***,***r***) possible permutations taken ***r*** at a time, so by the product rule,

$$P(n,r) = C(n,r)P(r,r) = C(n,r)r!$$

The following result is helpful in computing ***C***(***n***,***r***):

$$C(n,n-r) = \frac{n!}{(n-r)!(n-(n-r))!} = C(n,r)$$

or

$$\binom{n}{r} = \binom{n}{n-r}$$

Combinations examples[a]

- Specific cases of the formula ***C***(***n***,***r***),

$$C(n,0)=\binom{n}{0}=\frac{n!}{0!(n-0)!}=1$$

$$\binom{n}{n}=\binom{n}{0}=1$$

$$C(n,1)=\frac{n!}{1!(n-1)!}=n$$
$$=C(n,n-1)$$

- Let ***S*** = {**1**,**2**,**3**,**4**,**5**}. List all the **3**-combinations of ***S***.

First compute ***C***(**5**,**3**) = **5**! / **3**!(**5**-**3**)! = **5.4.3**! / **3**! **2**! =**10**, this is the number of subsets of ***S*** that contain **3** elements from **5**. The list of **3**-combinations is

Comb$_3$(***S***) = {{**1**,2,3},{**1**,2,4},{**1**,2,5},{**1**,3,4},{**1**,3,5},{**1**,4,5},{**2**,3,4},{**2**,3,5},{**2**,4,5},{**3**,4,5}}

- Find the value of ***C***(**12**,**6**) and ***C***(**5**,**1**).

$$C(12,6)=\frac{12!}{6!(12-6)!}=\frac{12!}{6!6!}=\frac{12\cdots 7\cdot 6!}{6!6!}$$
$$=\frac{12\cdot 11\cdot 10\cdot 9\cdot 8\cdot 7}{2\cdot 3\cdot 4\cdot 5\cdot 6}=11\cdot 3\cdot 4\cdot 7=\boxed{924}$$

$$C(5,1)=C(5,5-1)=C(5,4)$$
$$=\frac{5!}{4!(5-4)!}=\frac{5.4!}{4!}=\boxed{5}$$

Combinations examples[b]

- A group contains ***n*** men and ***n*** women. How many ways are there to arrange these people in a row if the men and women alternate?

$$M = \{m_1, \ldots, m_n\}$$
$$W = \{w_1, \ldots, w_n\}$$

There are **2** ways of organizing these people in a row.

$$(m_1, w_1, \ldots m_n, w_n)$$
$$(w_1, m_1, \ldots w_n, m_n)$$

The number of permutations in each is, $n \cdot n \cdot (n-1) \cdot (n-1) \cdots 1 \cdot 1 = (n!)^2$

Therefore, the total number of arrangements is $\boxed{2(n!)^2}$

- Suppose that a department contains **10** men and **15** women. How many ways are there to form a committee with **6** members if it must have the *same number of men and women?*

The committee must have **3** men and **3** women; we can select the men from ***C***(**10**,**3**) possible combinations, and the women from ***C***(**15**,**3**) combinations. By the product rule, the total number of ways to form such a committee is,

$$\binom{10}{3} \cdot \binom{15}{3} = \frac{10!}{3!7!} \cdot \frac{15!}{3!12!} = \frac{10.9.8}{6} \cdot \frac{15.14.13}{6} = \boxed{54600}$$

- How many ways are there to select **12** countries in the United Nations to serve on a council if **3** are selected from a block of **45**, **4** are selected from a block of **57**, and *the others* are selected from the remaining **69** countries?

$$\begin{array}{r} 3 \text{ from block } 1,\ n_1 = 45 \\ 4 \text{ from block } 2,\ n_2 = 57 \end{array} \rightarrow 3 + 4 = 7 \text{ countries from two blocks}$$

$$12 - 7 = 5 \text{ from block } 3,\ n_3 = 69$$

Therefore, applying the product rule and the number of combinations in each block, the answer is given by,

$$\binom{n_1}{r_1} \cdot \binom{n_2}{r_2} \cdot \binom{n_3}{12 - (r_1 + r_2)} = \boxed{\binom{45}{3} \cdot \binom{57}{4} \cdot \binom{69}{5}}$$

$$= \frac{45!}{3!42!} \cdot \frac{57!}{4!53!} \cdot \frac{69!}{5!64!} = \frac{45 \cdot 44 \cdot 43}{6} \cdot \frac{57 \cdot 56 \cdot 55 \cdot 54}{24} \cdot \frac{69 \cdot 68 \cdot 67 \cdot 66 \cdot 65}{120}$$

$$= 14190 \cdot 395010 \cdot 11238513 \approx \boxed{6.3 \times 10^{16}}$$

Pascal's identity: Let ***n*** and ***k*** be positive integers with ***n*** ≥ ***k***. Then

$$\boxed{C(n+1,k)=C(n,k-1)+C(n,k)}$$

Combinatorial proof: assume a finite set ***S*** with ***n***+1 elements. Let ***x*** belong to ***S*** so ***S**** = ***S*** - {*x*} has ***n*** elements. First, we have ***C***(***n***+1,***k***) subsets of size ***k*** from ***S***.

Second, a subset of ***S*** of size ***k*** either contains ***x*** together with ***k*** - **1** elements of ***S****, or does not contain ***x*** and has ***k*** elements from ***S****. There are ***C***(***n***,***k***-1) subsets of ***S*** that contain ***x*** and there are ***C***(***n***,***k***) subsets of ***S*** that do not contain ***x***. The result follows from the sum rule because both families of subsets are disjoint.

Pascal's triangle

$$\binom{0}{0}$$
$$\binom{1}{0}\quad\binom{1}{1}$$
$$\binom{2}{0}\quad\binom{2}{1}\quad\binom{2}{2}$$
$$\binom{3}{0}\quad\binom{3}{1}\quad\binom{3}{2}\quad\binom{3}{3}$$

$$C(4,2)=C(3,1)+C(3,2)$$

$$C(5,3)=C(4,2)+C(4,3)$$

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

We have shown by mathematical induction that the power set ***P***(***A***) of a finite set ***A*** with ***n*** elements has $\mathbf{2^n}$ subsets. A new identity with binomial coefficients is established using a combinatorial proof.

The power set ***P***(***A***) is a *union of families* $\boldsymbol{F_k}$ each one containing subsets of size ***k*** taken from the ***n*** elements of ***A***, i.e.,

$$P(A) = \bigcup_{k=0}^{n} F_k \; ; F_i \cap F_j = \varnothing \text{ for } i \neq j$$

Therefore,

$$|P(A)| = |\bigcup_{k=0}^{n} F_k| = \sum_{k=0}^{n} |F_k| = \boxed{\sum_{k=0}^{n} C(n,k) = 2^n}$$

Vandermonde's identity: let ***m***, ***n***, and ***r*** be nonnegative integers with ***r*** not exceeding either ***m*** or ***n***. Then

$$\boxed{C(m+n,r) = \sum_{k=0}^{r} C(m,r-k)C(n,k)}$$

The **binomial theorem**: let ***x***, ***y*** be variables, let ***n*** be a positive integer. Then,

$$(x+y)^n = \sum_{j=0}^{n} C(n,j)x^{n-j}y^j = \sum_{j=0}^{n} \binom{n}{j} x^{n-j}y^j$$

Combinatorial proof: when the product is realized it is clear that all possible terms $x^{n-j}y^j$ occur in the expansion for **j** = 0,1,2, ... ,***n***; note that, (***n*** - ***j***) + ***j*** = ***n***.

The number of times that a term $x^{n-j}y^j$ appears for a fixed ***j*** is ***C***(***n***,***j***) if we count y^j or ***C***(***n***,***n*** - ***j***) if we count x^{n-j} (in the *exclusive sense*). Since ***C***(***n***,***j***) = ***C***(***n***,***n*** - ***j***) by Pascal's identity, the result follows from the generalized sum rule.

Corollaries:

- let **x** = **y** = **1**, then $(1+1)^n = 2^n = \sum_{j=0}^{n} C(n,j)$
- let **x** = **1** and **y** = -**1**, then $(1-1)^n = 0 = \sum_{j=0}^{n} (-1)^j C(n,j)$

Combinations examples[d]

- What is the coefficient of x^8y^9 in the expansion $(3x + 2y)^{17}$?

In this case, $x^8 y^9 = x^{n-j} y^j$ thus we have that $n = 17$, $j = 9$, and $n - j = 17 - 9 = 8$. The coefficient is given by $C(17,9)$ modified by the new $x^* = 3x$ and the new $y^* = 2y$, i.e.,

$$C(n,j)(3)^{n-j}(2)^j = C(17,9)\cdot 3^8 \cdot 2^9 = 24310\cdot 6561\cdot 512 = \boxed{81{,}662{,}929{,}920}$$

- Show that if n is a positive integer, then $C(2n,2) = 2C(n,2) + n^2$.

We use Vandermonde's identity with $m = n$ and $r = 2$, i.e.,

$$C(m+n,r) = \sum_{k=0}^{r} C(m,r-k)C(n,k) \rightarrow C(2n,2) = \sum_{k=0}^{2} C(n,2-k)C(n,k)$$

$$\rightarrow C(2n,2) = C(n,2)C(n,0) + C(n,1)C(n,1) + C(n,0)C(n,2)$$

$$\rightarrow C(2n,2) = C(n,2)\cdot 1 + n\cdot n + 1\cdot C(n,2) = \boxed{2C(n,2) + n^2}$$

Advance Counting: Part I

- Recurrence relations
- Applications
- Types of recurrence relations
- Solving recurrence relations

A recurrence relation for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, $a_0, a_1, \ldots, a_{n-1}$ for all integers $n \geq n_0$ and $n_0 \in \mathbf{Z}^+$.

A sequence is called a solution of a recurrence relation if its terms satisfy the recurrence relation. The initial conditions specify the terms that precede the first term where the recurrence relation takes place.

- Show that the sequence $a_n = 2(-4)^n + 3$ is a solution of the recurrence relation $a_n = -3a_{n-1} + 4a_{n-2}$.

We compute a_{n-1} and a_{n-2} from the expression for a_n and substitute them in the recurrence relation or equation; thus,

$$\left.\begin{aligned} a_{n-1} &= 2(-4)^{n-1} + 3 \\ a_{n-2} &= 2(-4)^{n-2} + 3 \end{aligned}\right\} \rightarrow -3[2(-4)^{n-1} + 3] + 4[2(-4)^{n-2} + 3]$$

$$= -6(-4)^{n-1} - 9 - 2(-4)^{n-1} + 12$$

$$= -8(-4)^{n-1} + 3 = \boxed{2(-4)^n + 3 = a_n}$$

Recurrence application[a]

Recurrence relations are helpful for modeling a variety of situations that involve the terms of a sequence which are related in a quantitative way.

- A person deposits **$ 1000** in an account that yields **9%** *interest compounded yearly*. **a)** Setup a recurrence relation for the amount in the account at the end of ***n*** years. **b)** Find an explicit formula for the amount in the account at the end of ***n*** years. **c)** How much money will the account contain after **10** years?

a) Let A_n denote the amount of money in the account at the end of n years, so A_0 is the initial deposit, also **9/100** = **0.09**. Therefore,

$$\boxed{A_n = A_{n-1} + 0.09 A_{n-1} = 1.09 A_{n-1}}$$

b) We unfold the previous relation until the initial term is reached, i.e.,

$$A_n = 1.09 A_{n-1} = (1.09)^2 A_{n-2} = \cdots = (1.09)^k A_{n-k} = \cdots = \boxed{(1.09)^n A_0}$$

c) substitution of n = **10** in the last formula gives the amount of money after **10** years,

$$A_{10} = (1.09)^{10} A_0 = (1.09)^{10} \cdot 1000 = \boxed{\$\ 2,367.36}$$

A popular puzzle of the late nineteenth century is known as the **Tower of Hanoi**.

- Let $\boldsymbol{H_n}$ denote the *number of moves* needed to solve the puzzle with $\boldsymbol{n}$ disks.

- transfer the top $\boldsymbol{n}$ - **1** disks in peg 1 to peg 3 using $\boldsymbol{H_{n-1}}$ moves,
- then, using one move, put the largest disk of peg 1 in peg 2,
- finally, transfer again using $\boldsymbol{H_{n-1}}$ moves the $\boldsymbol{n}$ - **1** disks in peg 3 to peg 2.

Therefore,

$$H_{n-1}+1+H_{n-1}=\boxed{H_n=2H_{n-1}+1;\,H_1=1}$$

To find an explicit formula, iterate until the initial value is reached, i.e.,

$$H_n=2H_{n-1}+1=2(2H_{n-2}+1)+1=2^2H_{n-2}+2+1$$

$$=2^2(2H_{n-3}+1)+2+1=2^3H_{n-3}+2^2+2+1=\cdots$$

$$=2^kH_{n-k}+\sum_{j=0}^{k-1}2^j=\cdots=2^{n-1}H_1+\sum_{j=0}^{n-2}2^j=\sum_{j=0}^{n-1}2^j=\boxed{2^n-1}$$

- **a)** Find a recurrence relation for the number of bit strings of length ***n*** that contain **3** consecutive **0s**. **b)** What are the initial conditions? **c)** How many bit strings of length seven contain **3** consecutive **0s**?

a) Let s_n be the number of bit strings of length ***n*** containing **000**. Consider the following exhaustive possibilities for building all these strings,

- the same type of strings but of length ***n*** - **1** beginning with a **1**,
- the same type of strings but of length ***n*** - **2** beginning with a **01**,
- the same type of strings but of length ***n*** - **3** beginning with a **001**,
- strings beginning with **000** and a string of length ***n*** - **3**.

Therefore,
$$\boxed{s_n = s_{n-1} + s_{n-2} + s_{n-3} + 2^{n-3}}$$

b) There are no strings of length **0**, **1**, and **2** with **000**, so $s_0 = s_1 = s_2 = 0$

Note that $s_3 = s_2 + s_1 + s_0 + 2^{3-3} = 1$ which is "**000**".

c) we find the value of s_7 as follows,

$$s_7 = s_6 + s_5 + s_4 + 2^{7-3} = s_6 + s_5 + s_4 + 16$$

$$s_7 = (s_5 + s_4 + s_3 + 2^{6-3}) + s_5 + s_4 + 16$$

$$= 2s_5 + 2s_4 + 25 = 2(s_4 + s_3 + s_2 + 2^{5-3}) + 2s_4 + 25$$

$$= 4s_4 + 35 = 4(s_3 + s_2 + s_1 + 2^{4-3}) + 31 = 4s_3 + 43$$

$$= \boxed{47}$$

There are **47** bit strings of length **7** with **3** consecutive **0s**.

There is no single method to solve a recurrence relation for the general term a_n . However, the methods shown here apply to a certain class of recurrences that can be solved in a sistematic way.

A linear homogeneous recurrence (**LHR**) relation of degree k with constant coefficients (**CC**) has the following form

$$\boxed{a_n = c_1 a_{n-1} + \cdots + c_k a_{n-k} = \sum_{j=1}^{k} c_j a_{n-j} \; ; \forall j, c_j \in R \wedge c_k \neq 0}$$

The sequence $\{a_n\}$ satisfying this type of recurrence relation is unique once the k initial conditions are given

$$\boxed{a_0 = C_0, a_1 = C_1, \ldots, a_{k-1} = C_{k-1} \; ; \forall j, C_j \in R}$$

- $P_n = (1.11) P_{n-1}$; LHR, $k = 1$
- $f_n = f_{n-1} + f_{n-2}$; LHR, $k = 2$
- $a_n = a_{n-5}$; LHR, $k = 5$
- $a_n = a_{n-1} + a_{n-2}^2$; not linear
- $H_n = 2H_{n-1} + 1$; not homogeneous
- $B_n = nB_{n-1}$; n is variable

In order to solve a **LHR** of degree ***k*** with **CC** we *assume* that solutions are a power of some real number ***r***, i.e., $a_n = r^n$ with ***r*** a constant. Then,

$$r^n = \sum_{j=1}^{k} c_j r^{n-j} = c_1 r^{n-1} + c_2 r^{n-2} + \cdots + c_k r^{n-k}$$

$$\Rightarrow r^n - (c_1 r^{n-1} + c_2 r^{n-2} + \cdots + c_k r^{n-k}) = 0$$

$$\Rightarrow r^{n-k}(r^k - c_1 r^{k-1} - c_2 r^{k-2} - \cdots - c_k) = 0$$

$$\Rightarrow \boxed{\mathrm{pol}_k(r) = r^k - c_1 r^{k-1} - c_2 r^{k-2} - \cdots - c_k = 0}$$

Therefore, the sequence $\{a_n\}$ with $a_n = r^n$ is a solution if and only if ***r*** is a solution of the polynomial known as the characteristic equation of the recurrence relation. In that case, ***r*** is a characteristic root.

The general form of the term a_n of the sequence that solves a **LHR** of degree ***k*** with **CC** will depend on the number and nature of the characteristic roots.

Let us consider a **LHR** with **CC** and ***k*** = **2**. Thus,

$$\boxed{a_n = c_1 a_{n-1} + c_2 a_{n-2}\ ; c_2 \neq 0} \quad \Rightarrow \boxed{\mathrm{pol}_2(r) = r^2 - c_1 r^1 - c_2 = 0}$$

Suppose that the roots r_1 and r_2 of this quadratic equation are real and distinct, then, the solution, where the values of the constants α_1 and α_2 are determined by the initial conditions $a_0 = C_0$ and $a_1 = C_1$, is given by

$$\boxed{a_n = \alpha_1 r_1^n + \alpha_2 r_2^n} \qquad \boxed{\alpha_1 = \frac{C_1 - C_0 r_2}{r_1 - r_2} \qquad \alpha_2 = \frac{C_0 r_1 - C_1}{r_1 - r_2}}$$

- Find an explicit formula for the **Fibonacci numbers.**

$$f_n = (1)\, f_{n-1} + (1)\, f_{n-2} \Rightarrow \mathrm{pol}_2(r) = r^2 - r - 1 = 0 \quad \Rightarrow r_{1,2} = \begin{cases} (1+\sqrt{5})/2 = \varphi \\ (1-\sqrt{5})/2 = \tilde{\varphi} \end{cases}$$

$$f_0 = 1 \wedge f_1 = 1 \Rightarrow \alpha_1 = 1/\sqrt{5} \wedge \alpha_2 = -1/\sqrt{5}$$

Consequently, $$\boxed{f_n = \tfrac{1}{\sqrt{5}}(\varphi^n - \tilde{\varphi}^n)}$$

Let us consider a **LHR** with **CC** and ***k*** = **2**. Thus,

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}\ ;\ c_2 \neq 0 \qquad \Rightarrow \mathrm{pol}_2(r) = r^2 - c_1 r^1 - c_2 = 0$$

Suppose that the root r_0 of this quadratic equation has multiplicity = 2, then, the solution, where the values of the constants α_1 and α_2 are determined by the initial conditions $a_0 = C_0$ and $a_1 = C_1$, is given by

$$a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$$

- Find the solution to the recurrence relation (initial conditions are $a_0 = 1$, $a_1 = 6$)

$$a_n = 6a_{n-1} - 9a_{n-2} \Rightarrow \mathrm{pol}_2(r) = r^2 - 6r + 9 = 0 \Rightarrow r_0 = 3$$

$$a_0 = 1 \wedge a_1 = 6 \Rightarrow \alpha_1 = 1 \wedge \alpha_2 = 1$$

Hence, $$a_n = 3^n + n3^n = 3^n(n+1)$$

The previous examples presented simple **LHR**s of degree ***k*** = **2**. The general case is more complicated since the characteristic roots can be *real* or *complex*, and *each one with its own multiplicity*.

We present a generalization for the case of ***k*** different real roots $\boldsymbol{r_j}$, i.e.,

$$a_n = \sum_{j=1}^{k} c_j a_{n-j} \Rightarrow \mathrm{pol}_k(r) = r^k - \sum_{j=1}^{k} c_j r^{k-j} = 0 \quad \Rightarrow \mathrm{pol}_k(r) = \prod_{j=1}^{k} (r - r_j)$$

The solution has the form (the α_j can be found by applying the initial conditions):

$$\boxed{a_n = \sum_{j=1}^{k} \alpha_j r_j^n}$$

- Find the solution to the **LHR** of degree ***k*** **= 3**, $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$ with initial conditions $\boldsymbol{a_0 = 2}$, $\boldsymbol{a_1 = 5}$, $\boldsymbol{a_2 = 15}$.

$$\Rightarrow r^3 - 6r^2 + 11r - 6 = (r-1)(r-2)(r-3) = 0 \quad \Rightarrow r_1 = 1, r_2 = 2, r_3 = 3$$

$$a_0 = 2, a_1 = 5, a_2 = 15 \Rightarrow \alpha_1 = 1, \alpha_2 = -1, \alpha_3 = 2$$

The unique solution is $\boxed{a_n = 1 - 2^n + 2 \cdot 3^n}$

Advance Counting: Part II

- Divide and conquer relations
- Computational complexity

Divide-and-conquer recurrence relations are used in the *analysis of recursive algorithms*. Remember that a recursive algorithm solves a problem by decreasing the size of the input until the smallest input is computed.

The fact that a recursive algorithm divides a problem into subproblems of smaller size is expressed by saying that it is a divide-and-conquer procedure.

- The binary search algorithm.
  - the input is a sequence of size ***n***,
  - it reduces the input to ***n*** / **2** when ***n*** is even,
  - **2** comparisons are needed for this reduction.

***f*(*n*)** = # of comparisons

$$\Rightarrow f(n) = f(n/2) + 2$$

- A fast matrix multiplication algorithm.
  - the inputs are 2 matrices of size ***n*** x ***n***,
  - it reduces the inputs to ***n*/2** x ***n*/2** for ***n*** even
  - it uses **7** multiplications of 2 matrices,
  - it uses **15** additions of 2 matrices.

***f*(*n*)** = # of operations

$$\Rightarrow f(n) = 7f(n/2) + 15n^2/4$$

Structure of a general divide-and-conquer recurrence (**DCR**) relation:

- an algorithm splits a problem of size ***n*** into ***a*** subproblems,
- each subproblem is of size ***n*** / ***b*** (where ***b*** divides ***n***),
- a total of ***g***(***n***) extra operations are required for the split,
- if ***f***(***n***) represents the total number of operations, then

$$\boxed{f(n) = af(n/b) + g(n)} \qquad (1)$$

Our purpose is to provide an estimate of the rate of growth or time complexity for functions that satisfy divide-and-conquer recurrence relations.

By assumption, ***b*** is a divisor of ***n***, so we can take $n = b^k$ for some $k \in Z^+$

$$\Rightarrow f(n) = af(n/b) + g(n) = a[af(n/b^2) + g(n/b)] + g(n)$$

$$= a^2 f(n/b^2) + ag(n/b) + g(n)$$

$$= a^3 f(n/b^3) + a^2 g(n/b^2) + ag(n/b) + g(n)$$

$$= a^k f(n/b^k) + \sum_{j=0}^{k-1} a^j g(n/b^j)$$

$$\Rightarrow \boxed{f(n) = a^k f(1) + \sum_{j=0}^{k-1} a^j g(n/b^j)} \quad (2)$$

Equation 2 is used to establish the time complexity of the function ***f***(***n***).

Let ***f***(***n***) be an *increasing function* of ***n*** that satisfies the **DCR** given by,

$$\boxed{f(n) = af(n/b) + c\,;\, a \geq 1, b|n, b > 1, c \in R^+}$$

$$\boxed{\Rightarrow f(n) = \begin{cases} O(n^{\log_b a}) \text{ if } a > 1 \\ O(\log n) \text{ if } a = 1 \end{cases}}$$

When $n = b^k$ for some $k \in Z^+$, the *explicit formulas* for $f(n)$ are:

$$a = 1 \to f(n) = f(1) + c\log_b n \qquad (3)$$

$$a > 1 \to f(n) = \left(f(1) + \frac{c}{a-1} \right) n^{\log_b a} - \frac{c}{a-1} \qquad (4)$$

- Find $f(n)$ when $n = 3^k$, where f satisfies $f(n) = 2f(n/3) + 4$ with $f(1) = 1$.

  Apply Eq. 4 with $b = 3$, $a = 2$, and $c = 4$; hence,

$$f(n) = \left(1 + \tfrac{4}{2-1}\right) n^{\log_3 2} - \tfrac{4}{2-1} = \boxed{5n^{\log_3 2} - 4}$$

- Estimate the time complexity of $f(n)$ if it is an increasing function of n.

  Since $a > 1$, $b > 1$, $b \mid 3^k$, and $c > 0$, we have

$$\boxed{f(n) = O(n^{\log_3 2})}$$

Let ***f***(***n***) be an *increasing function* of ***n*** that satisfies the **DCR** given by,

$$f(n) = af(n/b) + cn^d \ ; a \ge 1, b|n, b > 1, c, d \in R^+$$

$$\Rightarrow \boxed{f(n) = \begin{cases} O(n^{\log_b a}) & \text{if } a > b^d \\ O(n^d \log n) & \text{if } a = b^d \\ O(n^d) & \text{if } a < b^d \end{cases}}$$

- Estimate the time complexity of ***f***(***n***) of the fast matrix multiplication algorithm.

Recall that $f(n) = 7f(n/2) + 15n^2/4$

Since ***a*** = **7**, ***b*** = **2**, ***d*** = **2**, ***a*** > ***b***^{d}, and ***c*** = **15/4**, we have

$$\boxed{f(n) = O(n^{\log_2 7}) \approx O(n^{2.8})} \quad \text{vs } O(n^3)$$

direct matrix multiplication

n	C_d	C_f
10	10^3	631
100	10^6	398108

Relations

- Binary relations
- Types of relations
- Operations with relations
- Representations
- Partition of a set
- Equivalence relations

Let ***A*** and ***B*** be sets. A binary relation from ***A*** to ***B*** is a subset of ***A*** X ***B***.

$$R \subseteq A \times B \quad ; \quad aRb \leftrightarrow (a,b) \in R \quad ; \quad a\not{R}b \leftrightarrow (a,b) \notin R$$

Binary relations represent a correspondence between the elements of two sets.

Let ***A*** be a set. A binary relation on ***A*** is a relation from ***A*** to ***A***.

$$R \subseteq A \times A = A^2$$

- every function or mapping from ***A*** to ***B*** *is a binary relation,*

$$f : A \to B \Leftrightarrow f \subseteq A \times B ; b = f(a) \Leftrightarrow a\ f\ b$$
$$(a,b) \in f \wedge (a,c) \in f \to b = c$$

- not every relation from ***A*** to ***B*** *is a function* from ***A*** to ***B***.

$$(a,b) \in R \wedge (a,c) \in R \wedge b \neq c$$

The following two examples are taken from number theory,

- the divisibility relation: $\boxed{D \subseteq Z^+ \times Z^+ ;\, mDn \leftrightarrow m|n}$

$$(3,3),(2,6),(5,100) \in D \text{ but } (2,1),(3,7),(7,3) \notin D$$

$$(1,n) \in D \text{ but } (n,1) \notin D \text{ for } n > 1$$

$$(m,n) \in D \text{ for } n = m$$

- the congruence relation modulo ***m*** (positive integer):

$$\boxed{C_m \subseteq Z \times Z ;\, xC_m y \leftrightarrow m|(x-y) \leftrightarrow x \equiv y (\operatorname{mod} m)}$$

$$(5,5),(12,0),(0,12) \in C_{12} \text{ but } (8,2),(2,8),(1,11) \notin C_{12}$$

$$(x,y) \in C_{12} \rightarrow (y,x) \in C_{12}$$

$$(x,y) \in C_{12} \text{ for } y = x$$

Consider a relation on ***A***, i.e., ***R*** is a subset of ***A*** X ***A***, then

$$R \text{ is reflexive } \Leftrightarrow \forall x \in A, (x,x) \in R$$
$$R \text{ is symmetric } \Leftrightarrow (x,y) \in R \rightarrow (y,x) \in R$$
$$R \text{ is antisymmetric } \Leftrightarrow (x,y) \wedge (y,x) \in R \rightarrow x = y$$
$$R \text{ is transitive } \Leftrightarrow (x,y) \wedge (y,z) \in R \rightarrow (x,z) \in R$$

- the divisibility relation on ***Z***$^+$ is reflexive, antisymmetric, and transitive.

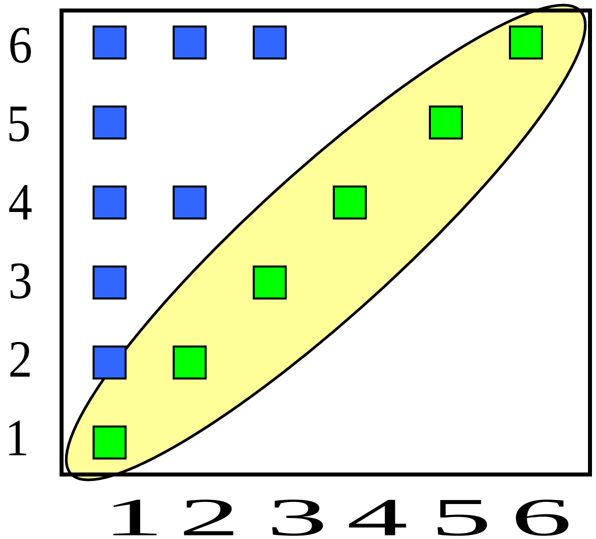
Antisymmetric, $$(m|n) \wedge (n|m) \rightarrow (n = k_1 m) \wedge (m = k_2 n)$$
$$\rightarrow n = k_1(k_2 n) = (k_1 k_2) n$$
$$\rightarrow k_1 k_2 = 1 \rightarrow k_1 = k_2 = 1$$

- the congruence relation on ***Z*** is reflexive, symmetric, and transitive.

Symmetric, $$x \equiv y (\operatorname{mod} m) \leftrightarrow m|(x-y)$$
$$\leftrightarrow m|(y-x) \leftrightarrow y \equiv x (\operatorname{mod} m)$$

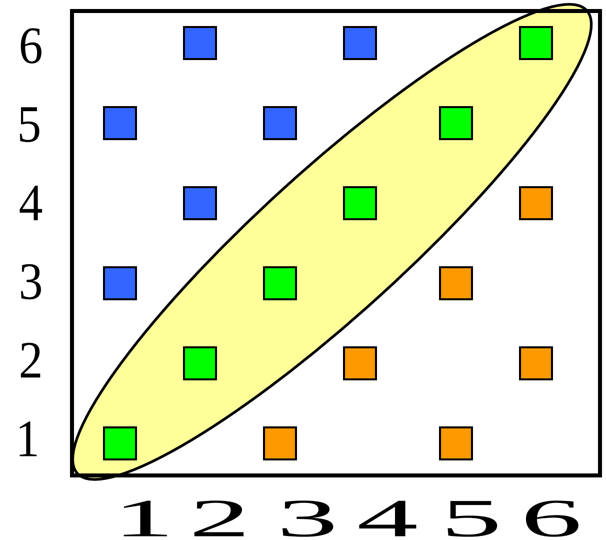
Types examples

Consider a *finite subset* of the positive integer numbers, say $\boldsymbol{A} = \{\mathbf{1},\mathbf{2},\mathbf{3},\mathbf{4},\mathbf{5},\mathbf{6}\}$.

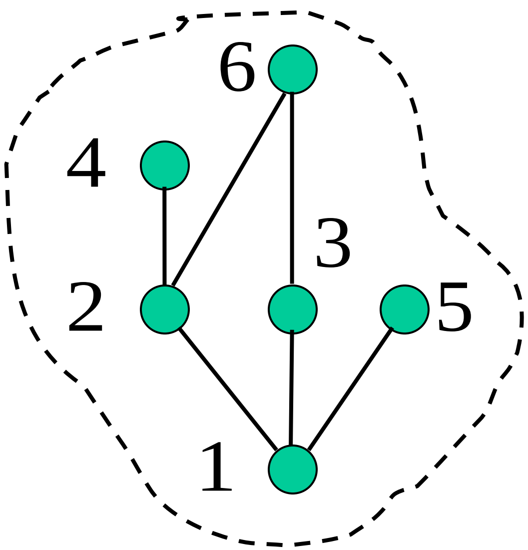


$$A \times A = A^2$$

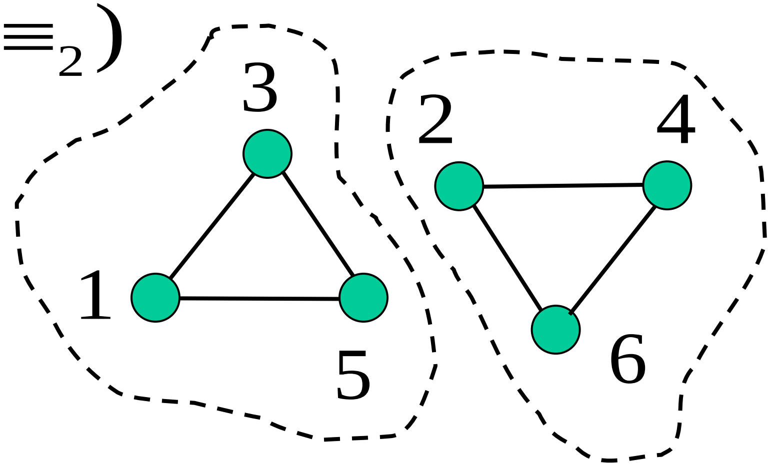
Cartesian support



$(A, |)$



$(A, \equiv_2)$



Since binary relations ***R*** are subsets of ***A*** X ***B*** or subsets of ***A*** X ***A***, the normal operations of set theory can be applied to relations.

Suppose that both sets are finite, i.e., |***A***| = ***n*** and |***B***| = ***m***, where ***m***,***n*** are positive integers. How many relations are there *from* ***A*** *to* ***B***?, *on* ***A***?

• the number of relations from ***A*** to ***B*** is

$$|P(A\times B)| = 2^{|A\times B|} = 2^{|A|\cdot|B|} = \boxed{2^{nm}}$$

• for the number of relations on ***A*** take ***m*** = ***n*** in the previous formula, then

$$\boxed{|P(A^2)| = 2^{n^2}}$$

Composition of relations:

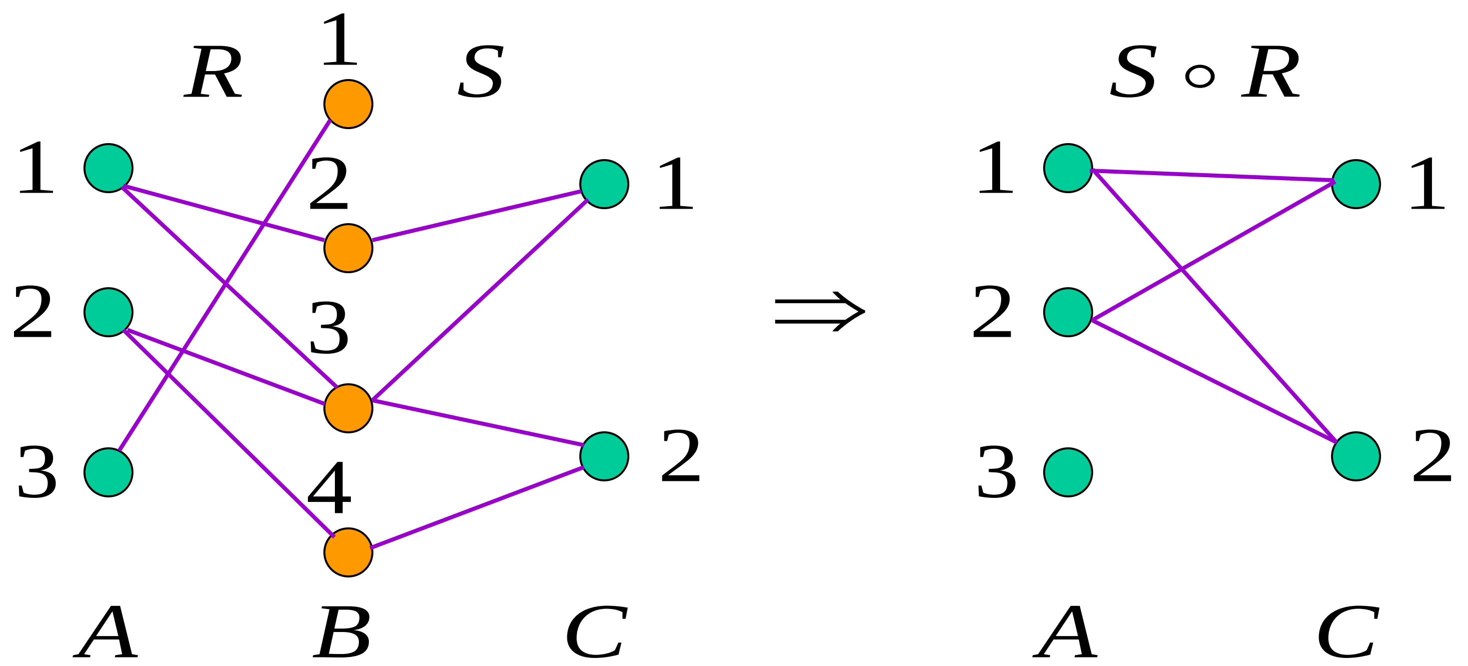
$$\boxed{R\subseteq A\times B\,;\, S\subseteq B\times C \to S\circ R = \{(a,c) | \exists b\in B, aRb \wedge bSc\}}$$

Powers of a relation on ***A***:

$$\boxed{R^1 = R\,;\, R^{n+1} = R^n\circ R \quad \forall n\in Z^+}$$

Composition example[a]

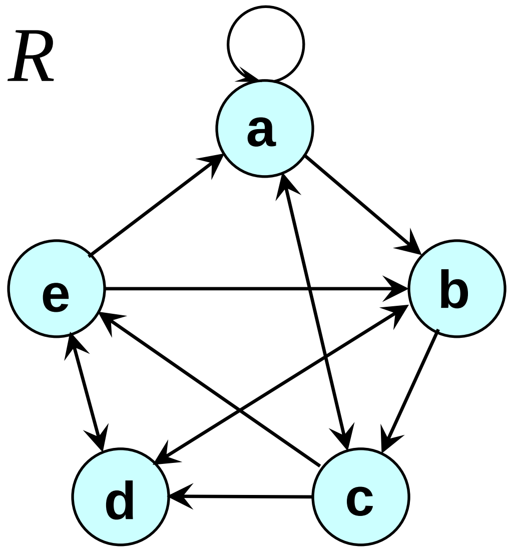
Let R = {(1,2),(1,3),(2,3),(2,4),(3,1)} and S = {(2,1),(3,1),(3,2),(4,2)}.
Find the composition of R with S.



$$\boxed{S \circ R = \{(1,1),(1,2),(2,1),(2,2)\}}$$

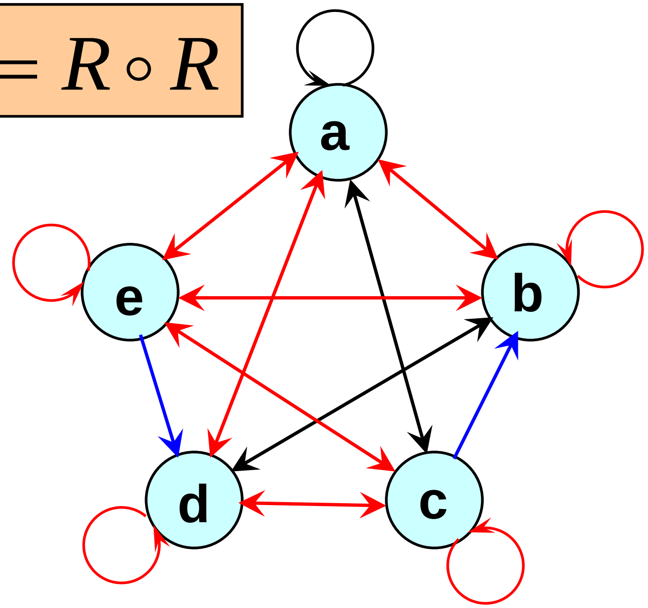
Composition example[b]

Let $\boldsymbol{R}$ be the relation on the set $\boldsymbol{A}$ = {**a**,**b**,**c**,**d**,**e**} containing the ordered pairs (**a**,**a**),(**a**,**b**),(**a**,**c**),(**b**,**c**),(**b**,**d**),(**c**,**a**),(**c**,**d**),(**c**,**e**),(**d**,**b**),(**d**,**e**),(**e**,**a**),(**e**,**b**),(**e**,**d**).
Find the following powers $\boldsymbol{R}^2$, $\boldsymbol{R}^3$, $\boldsymbol{R}^4$ and $\boldsymbol{R}^5$.



$\Rightarrow$

$$R^2 = R \circ R$$



$$R^3 = R^2 \circ R = A \times A$$

Thus, $\boldsymbol{R}^3$ is the total relation on $\boldsymbol{A}$ containing all pairs.

$$R^5 = R^4 = R^3$$

In this case, increasing powers *do not add any new pairs*.

Relations — Representations

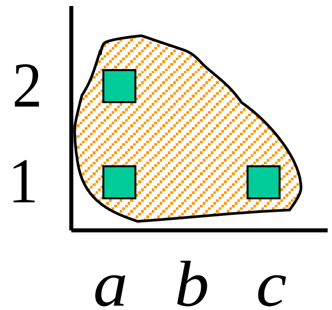
There are basically four ways to represent binary relations *from* ***A*** *to* ***B*** or *on* ***A***.

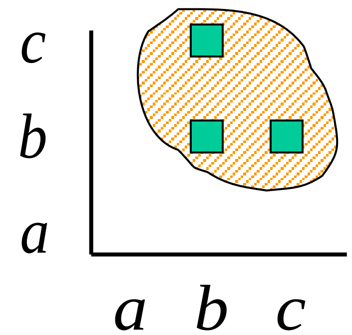
$$\boxed{R \subseteq A \times B} \qquad \boxed{R \subseteq A \times A = A^2}$$

- Set builder notation

$$R = \{(a,1),(a,2),(c,1)\} \qquad R = \{(b,b),(c,b),(b,c)\}$$

- Cartesian support



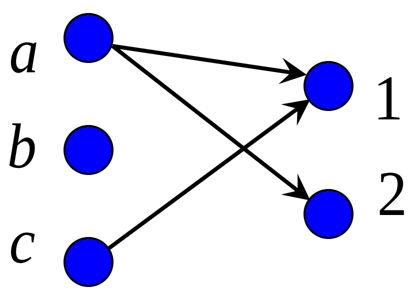


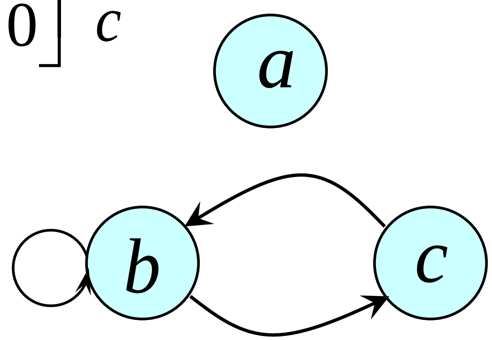
- Zero-one matrix

$$M_R = \begin{array}{cc} 1 & 2 \end{array}$$
$$M_R = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{array}{c} a \\ b \\ c \end{array}$$

$$\begin{array}{ccc} a & b & c \end{array}$$
$$M_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}\begin{array}{c} a \\ b \\ c \end{array}$$

- Directed graph





Relations — Matrix form

A binary relation from ***A*** to ***B*** has a corresponding ***m*** x ***n*** boolean matrix whenever ***A*** has ***m*** elements and ***B*** has ***n*** elements. The *relation matrix* is defined as

$$R \subseteq A \times B \Leftrightarrow M_R = [m_{ij}] = \begin{cases} 1 \text{ if } (a_i, b_j) \in R \\ 0 \text{ if } (a_i, b_j) \notin R \end{cases}$$

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & 0 \end{bmatrix} \qquad \begin{bmatrix} & 1 & 1 & 0 \\ 1 & & 1 & 0 \\ 1 & 1 & & 1 \\ 0 & 0 & 1 & \end{bmatrix} \qquad \begin{bmatrix} & 1 & 0 & 1 \\ 0 & & 0 & 0 \\ 1 & 0 & & 0 \\ 0 & 0 & 1 & \end{bmatrix}$$

$\forall i, m_{ii} = 1$ — ***R*** is reflexive

$\forall i, m_{ii} = 0$ — ***R*** is irreflexive

$m_{ij} = m_{ji}$ — ***R*** is symmetric

$m_{ij} = 1 \rightarrow m_{ji} = 0$ — ***R*** is antisymmetric

$$R^{-1} = \{(b,a) | (a,b) \in R\} \text{ (inverse)} \rightarrow M_{R^{-1}} = M_R^t \text{ (transpose)}$$

$$\overline{R} = \{(a,b) | (a,b) \notin R\} \text{ (complementary)} \rightarrow M_{\overline{R}} = M_{A^2} - M_R \text{ (complement)}$$

The operations between binary relations are performed using the *algebra of zero-one* or *boolean matrices*. Suppose that ***R*** and ***S*** are relations on a finite set ***A*** with ***n*** elements whose corresponding matrices are M_R and M_S, then

$$M_{R\cup S} = M_R \vee M_S \text{ (union - join)}$$

$$M_{R\cap S} = M_R \wedge M_S \text{ (intersection - meet)}$$

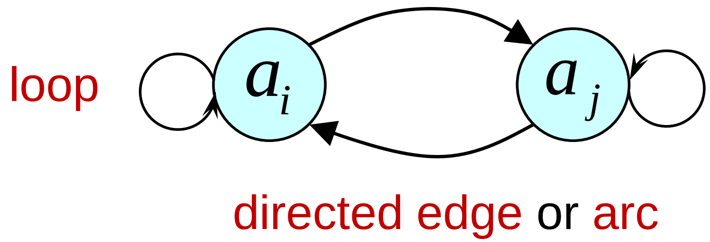
$$M_{R\oplus S} = M_R \oplus M_S \text{ (symmetric difference - xor)}$$

$$M_{S\circ R} = M_R \otimes M_S \text{ (composition - boolean product)}$$

$$M_{R^n} = M_R^{[n]} \text{ (power - power)}$$

In other words, the algebra of relations ***R*** on a set ***A*** is the same as the algebra of zero-one square matrices M_R. These equations are useful for *computational purposes*. So the basic data structure of a binary relation is a boolean square matrix.

A *binary relation on* ***A*** can be represented in a pictorial way by means of a directed graph or digraph ***G*** = (***V***,***E***) where the elements of ***A*** are the vertices or nodes in ***V*** and the ordered pairs belonging to ***R*** are the elements of ***E*** called edges or arcs.

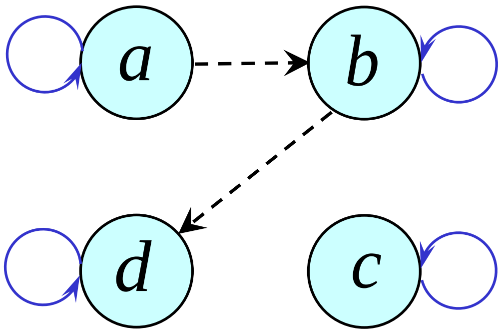


$$a_i R a_i \Leftrightarrow (a_i, a_i) \in R \rightarrow \text{ loop}$$

$$a_i R a_j \;;\, i \neq j \Leftrightarrow (a_i, a_j) \in R \rightarrow \text{ directed edge}$$

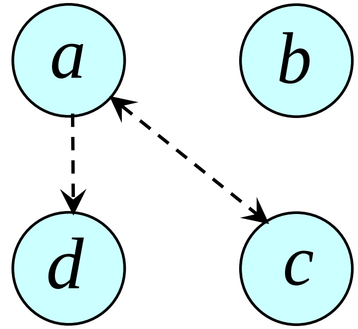
Representing a binary relation with a digraph is a visual aid for understanding the properties and types of relations. They serve as an introduction to the concepts of graph theory and are useful for modeling problems.

Relations Properties



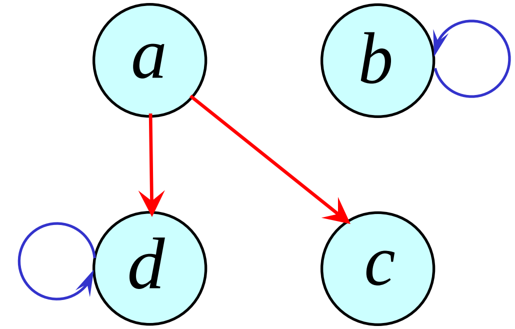
$$\forall i, \overline{a_i a_i} \in E$$

R is reflexive



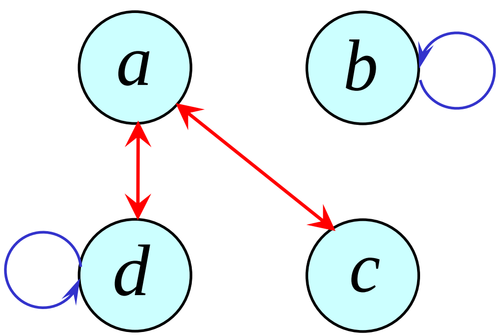
$$\forall i, \overline{a_i a_i} \notin E$$

R is irreflexive



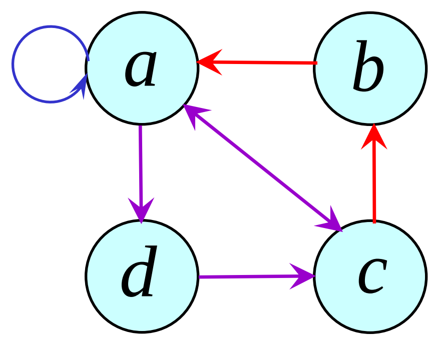
$$\overline{a_i a_j} \in E \to \overline{a_j a_i} \notin E$$

R is antisymmetric



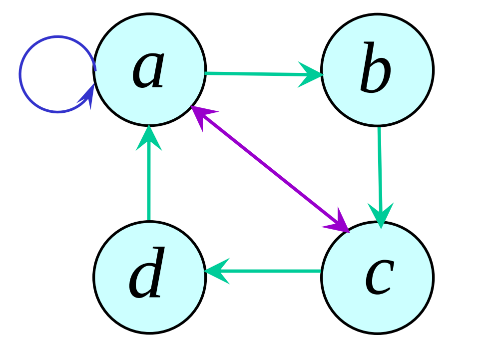
$$\overline{a_i a_j} \in E \to \overline{a_j a_i} \in E$$

R is symmetric



$$\overline{a_i a_k} \in E \wedge \overline{a_k a_j} \in E$$
$$\to \overline{a_i a_j} \in E$$

R is transitive



$$\overline{a_i a_j} \in E \to \overline{a_j a_i} \in E^{-1}$$

inverse of ***R*** (previous)

$\boxed{R}$

$a \to b$

$a \to b \to c$

$a \to b \to c \to d$

$\Rightarrow$

$\boxed{R^1}$ $a \to b$

$\boxed{R^2}$ $a \to c$

$\boxed{R^3}$ $a \to d$

A new edge is added if there is a *path of length* **1**, **2**, and **3** in the original relation.

In general, the ***n***-th power of a binary relation ***R*** on ***A*** will contain (***i***, ***j***) if there is a path of length ***n*** from ***i*** to ***j*** in ***R***; here, length means the number of arcs.

$$\boxed{R \text{ is transitive } \Leftrightarrow \forall n, R^n \subseteq R}$$

$$n = 1 \text{ (trivial)}, \; R^1 \subseteq R$$

$$\boxed{n = 2 \text{ (basis)}, \; R^2 = R \circ R \subseteq R}$$

$$(a,c) \in R^2 \leftrightarrow \exists b \in A, (a,b), (b,c) \in R$$

$$\text{since } R \text{ is transitive } \rightarrow (a,c) \in R$$

Proof of the direct implication:

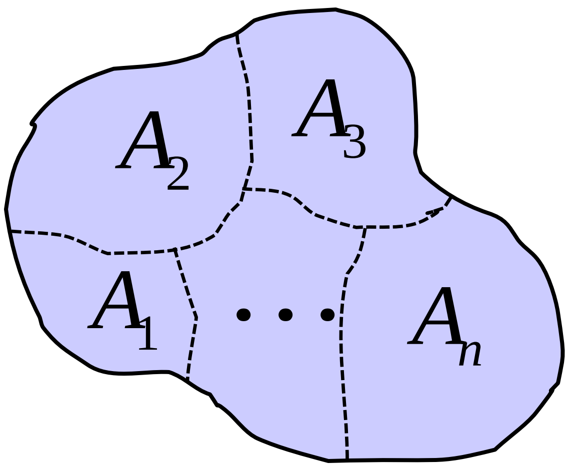
induction hypothesis $n = k, \; R^k \subseteq R$ then

$$\boxed{\begin{aligned} R^{k+1} &= R^k \circ R \subseteq R \circ R \\ &= R^2 \subseteq R \end{aligned}}$$

A family of sets ***P*** = {***A***$_1$,***A***$_2$, ... ,***A***$_n$} of a given set ***A*** is a partition if and only if the sets ***A***$_i$ *cover* ***A*** and *any two subsets* in ***P*** are *disjoint*, i.e.,

$$A = \bigcup_{i=1}^{n} A_i \; ; \forall i \neq j, \; A_i \cap A_j = \varnothing$$

A natural binary relation on ***A*** is defined as follows: $xR_P y \leftrightarrow \exists i;\, x, y \in A_i$



The sets ***A***$_i$ are called blocks, cells or classes.

The relation induced by ***P*** has the properties,

- $xR_P x \leftrightarrow \exists i,\, x \in A_i$ (reflexive)
- $xR_P y \rightarrow yR_P x$ since $\exists i;\, x, y \in A_i \rightarrow y, x \in A_i$ (symmetric)
- $xR_P y \wedge yR_P z \rightarrow xR_P z$ (transitive) $\exists i = j;\, x, y \in A_i \wedge y, z \in A_j \rightarrow x, z \in A_i$

An equivalence on ***A*** is a binary relation ***R*** on ***A*** that is reflexive, symmetric, and transitive. Every equivalence on ***A*** *induces a partition* of ***A*** known as the quotient set denoted by ***A*** / ***R***.

For each element ***a*** in ***A*** we define the equivalence class of ***a*** as the set:

$$[a]_R = [a] = cl(a) = \{b \in A | aRb\}$$

Therefore, the quotient set (or partition induced by ***R*** on ***A***) is defined as the family:

$$A/R = \{[a] | a \in A\}$$

It is not difficult to prove that,

$$aRb \leftrightarrow [a] = [b] \;\; ; \;\; a\not{R}b \leftrightarrow [a] \cap [b] = \emptyset \;\; ; \;\; A = \bigcup_{a \in A} [a]$$

Any element or *member* of a class [***a***] is called a representative.

$$\text{Equivalence on } A \Leftrightarrow \text{Partition of } A$$

Same idea, but expressed In different forms!

This is the classic example of an equivalence. It shows how this idea is used to build new objects from old ones in a clever way.

Let ***R*** be the binary relation of congruence modulo *m* between two integers ***x***,***y*** $\in$ ***Z***.

$$\begin{aligned}&\text{reflexive,} \quad x \equiv x (\text{mod}\, m)\\ &\text{symmetric,} \quad x \equiv y (\text{mod}\, m) \to y \equiv x (\text{mod}\, m)\\ &\text{transitive,} \quad x \equiv y (\text{mod}\, m) \wedge y \equiv z (\text{mod}\, m) \to x \equiv z (\text{mod}\, m)\end{aligned}$$

The equivalence classes and the quotient set are computed as follows:

$$[x] = \{ y \mid y \equiv x (\text{mod}\, m) \} = \{ y \mid y - x = km \} = \{ y \mid y = x + km \}$$

or

$$[x] = \{ x + km, k \in Z \} = \{ \ldots, x - 2m, x - m, x, x + m, x + 2m, \ldots \}$$

Each equivalence class has an *infinite number of elements* from ***Z*** and we take as representatives *the possible remainders* for *m*, i.e., 0,1, ... , *m* - 1. Therefore,

$$Z / R = \{ [r] \mid r = 0,1, \ldots, m-1 \} = \{ [0], [1], \ldots, [m-1] \} = Z_m$$

This quotient set is called the integers modulo *m* and it is a **finite set**!

Graphs-Part I

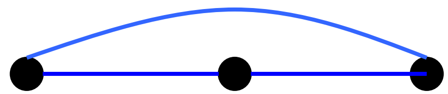
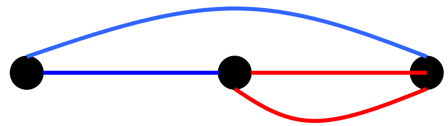
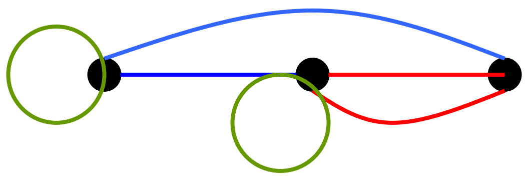
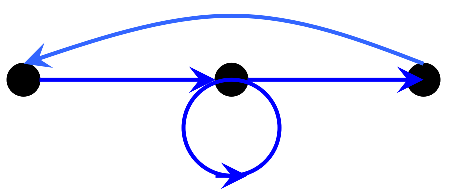
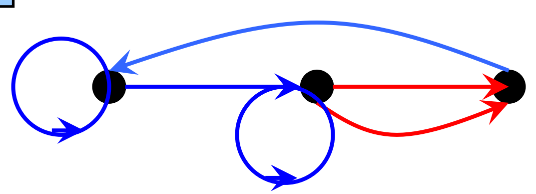
- Types of graphs
- Graphs as models
- Graphs examples
- Applications of graphs
- Operations with graphs

Graphs-I

Types

Undirected Graphs

Directed Graphs

	Name	Multiple edges?	Loops?
	simple graph	no	no
	multigraph	yes	no
	pseudograph	yes	yes
	directed graph	no	yes
	directed multigraph	yes	yes

Mathematical definitions for pseudographs and directed multigraphs:

$$\boxed{\mathrm{PG} = (V, E, f : E \to \{\{u,v\} \mid u,v \in V\})}$$

$$f(e_1) = f(e_2) = \{u,v\} \wedge u \neq v \text{ ; parallel edges } e_1, e_2$$

$$f(e) = \{u,v\} \wedge u = v \text{ ; loop } e$$

$$\boxed{\mathrm{MDG} = (V, E, f : E \to \{(u,v) \mid u,v \in V\})}$$

$$f(e_1) = f(e_2) = (u,v) \wedge u \neq v \text{ ; directed parallel edges } e_1, e_2$$

$$f(e) = (u,v) \wedge u = v \text{ ; directed loop } e$$

A graph is a model when the vertices (set ***V***) and edges (set ***E***) are assigned a *specific meaning according to the problem* being represented by the graph.

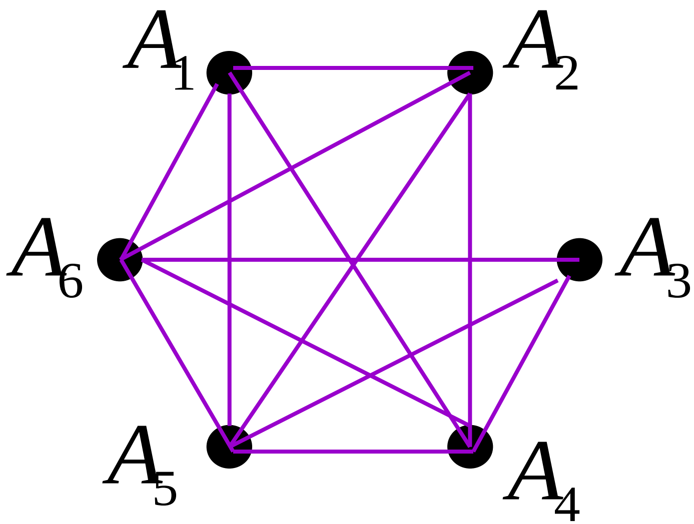
- niche overlap graph: interaction between animal species; ***V*** = {species}, {***u***,***v***} means species ***u*** competes with species ***v*** for food resources.
- precedence graph: execution of programs in concurrent mode; ***V*** = {instructions}, (***u***,***v***) means instruction ***v*** can be executed if instruction ***u*** has been executed.

➢ The intersection graph of a collection of sets A_1, A_2, ... , A_n is the graph that has a vertex for each of these sets and has an edge connecting the vertices representing two sets if these sets have a nonempty intersection. Construct the intersection graph for the following collection of sets.

$$A_1 = \{x | x < 0\},\ A_2 = \{x | -1 < x < 0\},\ A_3 = \{x | 0 < x < 1\},$$

$$A_4 = \{x | -1 < x < 1\},\ A_5 = \{x | x > -1\},\ A_6 = R$$

Note that: $A_1 \cap A_3 = A_2 \cap A_3 = \varnothing$ otherwise $A_i \cap A_j \neq \varnothing$ if $i \neq j$



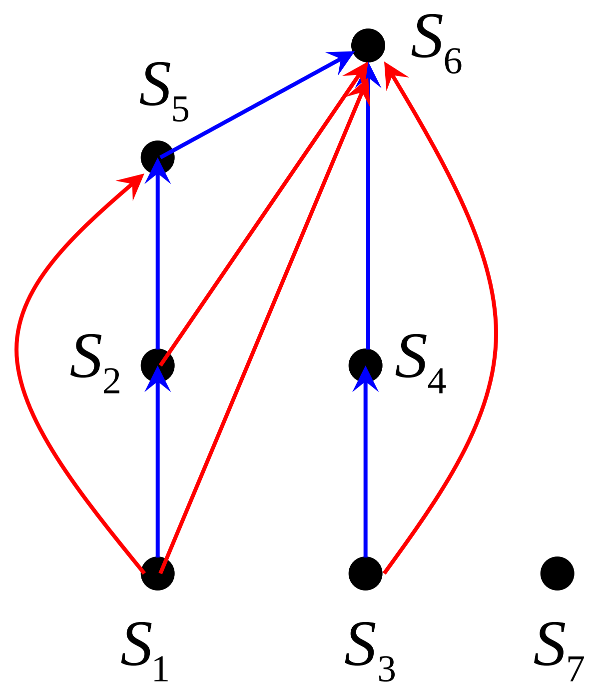
- Construct a precedence graph for the following program:

S_1: $x := 0$
S_2: $x := x + 1$
S_3: $y := 2$
S_4: $z := y$
S_5: $x := x + 2$
S_6: $y := x + z$
S_7: $z := 4$

The set of vertices ***V*** contains as elements the *statements* or *instructions* of the program, i.e.,

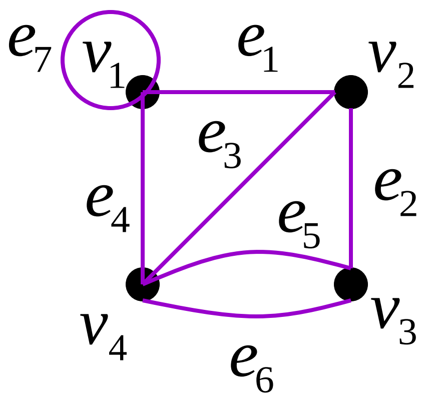
$$V = \{S_1, S_2, \ldots, S_7\}$$

There is an edge ***e*** = (***S***$_i$, ***S***$_j$) if ***S***$_j$ *can be executed after* ***S***$_i$ *has been executed*, therefore the directed graph is



General terminology for undirected graphs:

- two vertices ***u*** and ***v*** are called adjacent or neighbors if {***u***,***v***} is an edge,
- if ***e*** = {***u***,***v***} , the edge is called incident with the vertices ***u*** and ***v***,
- the edge ***e*** is also said to *connect* ***u*** and ***v***,
- the vertices ***u*** and ***v*** are called endpoints of the edge {***u***,***v***}.



v_1, v_2 adjacent ; v_1, v_3 not adjacent

e_3 incident with v_4, v_2

e_5, e_6 parallel edges, endpoints are v_4, v_3

- the degree of a vertex = number of edges incident with it,
- a loop at a vertex contributes twice to the degree of that vertex,
- the degree of a vertex ***v*** is denoted by **deg**(***v***).

$$\deg(v_1) = 4; \deg(v_2) = 3$$

$$\sum_{i=1}^{4} \deg(v_i) = 4+3+3+4 = 14 = 2 \cdot 7 = 2\,|E|$$

Euler's theorem (handshaking): Let ***G*** = (***V***,**E**) be an undirected graph with ***e*** edges, then (it applies also to *multigraphs* or *pseudographs*)

$$\sum_{i=1}^{n} \deg(v_i) = \boxed{\sum_{v \in V} \deg(v) = 2e} = 2\,|E|$$

An undirected graph has an *even number of vertices of odd degree*.

$$V_{\text{even}} = \{v \in V | \deg(v) \text{ is even}\}\,; V_{\text{odd}} = \{v \in V | \deg(v) \text{ is odd}\}$$

$$\sum_{v \in V} \deg(v) = \sum_{v \in V\,(\text{even})} \deg(v) + \sum_{v \in V\,(\text{odd})} \deg(v) = 2k_1 + p = 2e \quad \rightarrow p \text{ is even}$$

Since the *second summation is an even number* and is obtained by adding odd numbers there must be an even number of them.

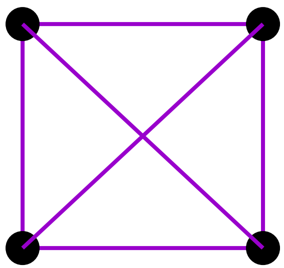
Particular kinds of vertices:

$$\boxed{v \text{ is isolated } \leftrightarrow \deg(v) = 0\,; v \text{ is pendant } \leftrightarrow \deg(v) = 1}$$

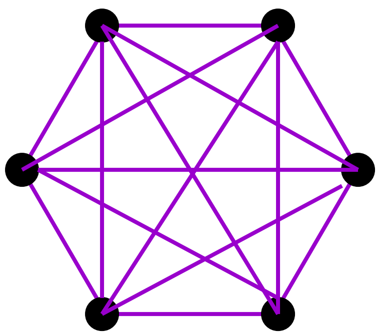
Special Types[a]

Complete graphs

K_4



K_6

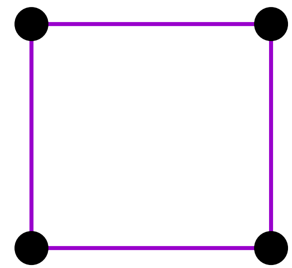


K_n ; $n \geq 1$

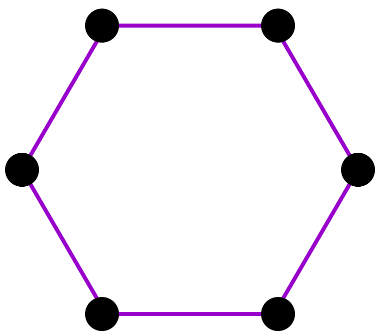
$\forall i \neq j, \{v_i, v_j\} \in E$

Cycles

C_4



C_6

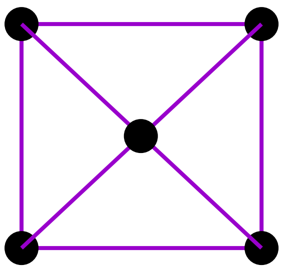


C_n ; $n \geq 3$

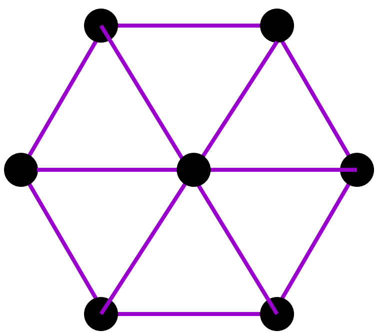
$\forall i, \{v_i, v_{i+1 \bmod n}\} \in E$

Wheels

W_4



W_6



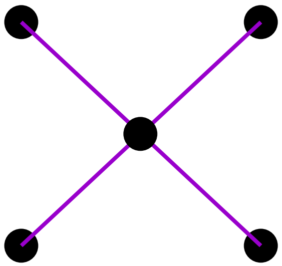
W_n ; $n \geq 3$ a cycle

plus $\forall i \neq 0, \{v_0, v_i\} \in E$

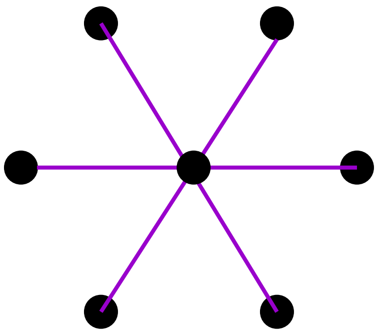
Graphs-I Special Types[b]

Stars

S_4

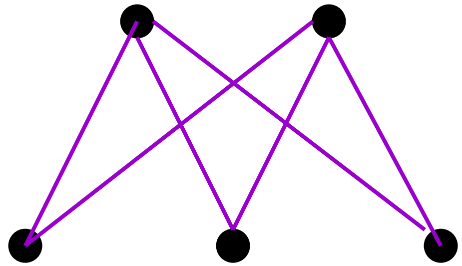


S_6

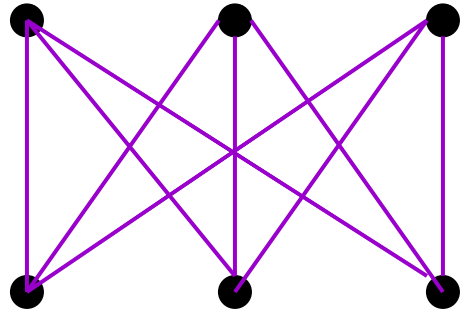


Complete bipartite graphs

$K_{2,3}$

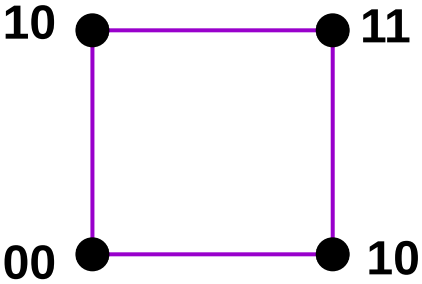


$K_{3,3}$

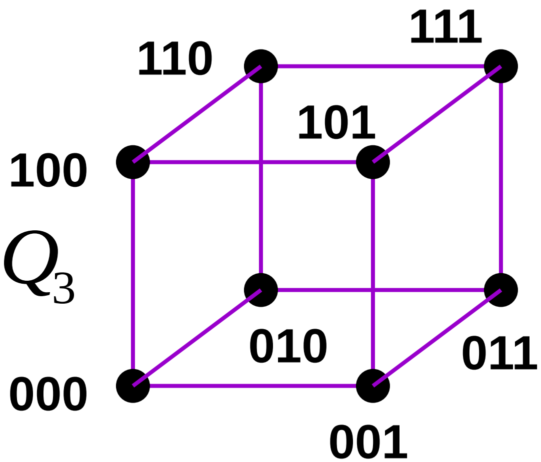


Cubes

Q_2



Q_3



$S_n = K_{1,n} \; , n \geq 1$

$K_{m,n} \; ; m,n \geq 1, \forall i \neq j$

$\{u_i, v_j\} \in E \leftrightarrow u_i \in U, v_j \in V$

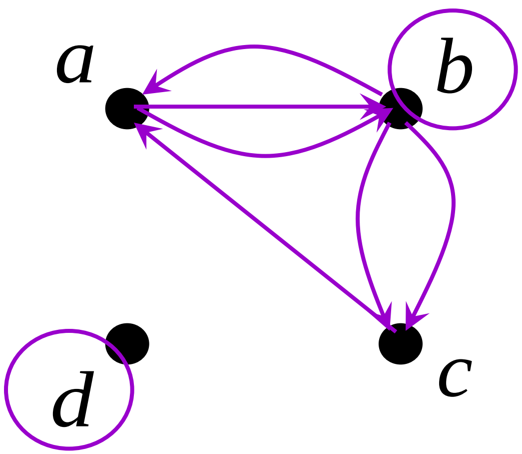
$U \cap V = \varnothing$

$Q_n ; n \geq 1, v \in B^n$

$\forall i \neq j, \{v_i, v_j\} \in E \leftrightarrow$

$H_d(v_i, v_j) = 1$

- Determine the sum of the in-degrees of the vertices and the **sum of the outdegrees** of the vertices directly. Show that they are *both equal to the number of edges* in the graph.



$$\sum_{v \in V} \deg^-(v) = \delta^-(a) + \delta^-(b) + \delta^-(c) + \delta^-(d)$$

$$= \boxed{2+3+2+1=8}$$

$$\sum_{v \in V} \deg^+(v) = \delta^+(a) + \delta^+(b) + \delta^+(c) + \delta^+(d)$$

$$= \boxed{2+4+1+1=8}$$

In general, for a directed multigraph the following relation always holds:

$$\boxed{\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|}$$

- How many vertices and how many edges do the following graphs have?

$$K_n \text{ (complete)} \rightarrow |V| = n\,;\, |E| = C(n,2) = \frac{n(n-1)}{2}$$
$$C_n \text{ (cycle)} \rightarrow |V| = n\,;\, |E| = n$$
$$W_n \text{ (wheel)} \rightarrow |V| = n+1\,;\, |E| = 2n$$
$$K_{m,n} \text{ (complete bipartite)} \rightarrow |V| = m+n\,;\, |E| = mn$$
$$Q_n \text{ (cube)} \rightarrow |V| = 2^n\,;\, |E| = n2^{n-1}$$

- Let ***G*** be a graph with ***v*** vertices and ***e*** edges. Let ***M*** be the maximum degree of the vertices of ***G***, and let ***m*** be the minimum degree of the vertices of ***G***. Show that a) **2*e* / *v*** ≥ ***m***, and b) **2*e* / *v*** ≤ ***M***.

$$\text{a)}\quad \forall v, m \leq \delta(v) \rightarrow v \cdot m \leq \sum_{v \in V} \delta(v) = 2e \rightarrow \boxed{2e / v \geq m}$$

$$\text{b)}\quad \forall v, M \geq \delta(v) \rightarrow v \cdot M \geq \sum_{v \in V} \delta(v) = 2e \rightarrow \boxed{2e / v \leq M}$$

An Application

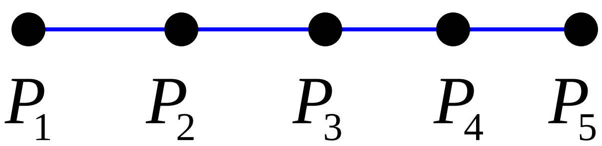
Graphs with a certain structure as those shown before can be used to model, for example, *computer networks* or arrangements of units for *parallel processing*.

Computer networks topology: **rings** (cycles), **stars** or **hybrid** (wheels).

Parallel processing machines: **complete graphs**, **grids** or **hypercubes** (cubes).

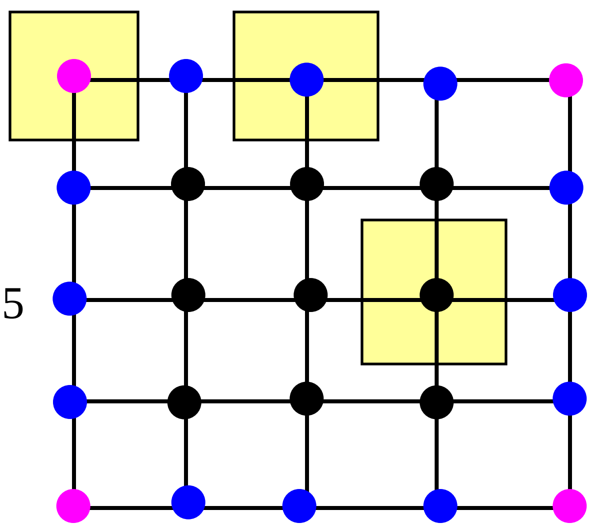
$K_n \,; n \le 5$ *Low values* of ***n*** ; otherwise, *number of connections* = ***C***(***n***,**2**).

L_5



linear array of **5** processors; only *two direct connections* between ***P*$_{\mathbf{i}}$** and processors ***P*$_{\mathbf{i-1}}$** , ***P*$_{\mathbf{i+1}}$** (except ***P*$_{\mathbf{1}}$**,***P*$_{\mathbf{5}}$**)

$M_{5,5}$



a mesh or grid (two dimensional array) of **5** x **5** = **25** processing units. Each interior processor (not on the boundary) has **4** *direct connections* with its neighbors.

Communication between some processors require a number of intermediate links:

$$\boxed{O(\sqrt{n}) = O(m)\,;\, m \times m \text{ mesh}}$$

Two basic operations on graphs are performed by *eliminating* or *adjoining* vertices or edges. The formal definitions are as follows:

- a subgraph of a graph $\boldsymbol{G} = (\boldsymbol{V},\boldsymbol{E})$ is a graph $\boldsymbol{H} = (\boldsymbol{W},\boldsymbol{F})$ where $\boldsymbol{W} \subseteq \boldsymbol{V}$ and $\boldsymbol{F} \subseteq \boldsymbol{E}$.

- the union of two simple graphs $\boldsymbol{G}_1 = (\boldsymbol{V}_1,\boldsymbol{E}_1)$ and $\boldsymbol{G}_2 = (\boldsymbol{V}_2,\boldsymbol{E}_2)$ is the simple graph $\boldsymbol{G} = \boldsymbol{G}_1 \cup \boldsymbol{G}_2$ where $\boldsymbol{V} = \boldsymbol{V}_1 \cup \boldsymbol{V}_2$ and $\boldsymbol{E} = \boldsymbol{E}_1 \cup \boldsymbol{E}_2$.

Examples:

- Every cycle $\boldsymbol{C}_n$ for $\boldsymbol{n} \geq \mathbf{3}$ is a subgraph of $\boldsymbol{K}_n$ (complete) and of $\boldsymbol{W}_n$ (wheel)

- Every star $\boldsymbol{S}_n$ for $\boldsymbol{n} \geq \mathbf{1}$ is a subgraph of $\boldsymbol{K}_{m,n}$ (complete bipartite) and of $\boldsymbol{W}_n$ (wheel)

- Every wheel $\boldsymbol{W}_n$ for $\boldsymbol{n} \geq \mathbf{3}$ is the union of $\boldsymbol{C}_n$ (cycle) and of $\boldsymbol{S}_n$ (star)

- The cube $\boldsymbol{Q}_n$ for $\boldsymbol{n} \geq \mathbf{1}$ is a subgraph of the cube $\boldsymbol{Q}_{n+1}$

Graphs-Part II

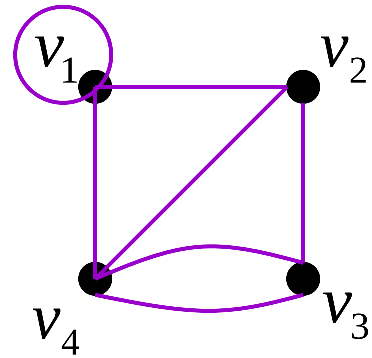
- Adjacency matrix
- Incidence matrix
- Graph isomorphism

Adjancecy Matrix

A usual representation of a graph (as in the case of binary relations) is a matrix. The ***n*** x ***n*** adjacency matrix $\boldsymbol{A_G}$ of a graph $\boldsymbol{G} = (\boldsymbol{V},\boldsymbol{E})$ with $|\boldsymbol{V}| = \boldsymbol{n}$ where $\boldsymbol{m}$ is a positive integer denoting the multiplicity of an edge is defined as follows:

Undirected graphs

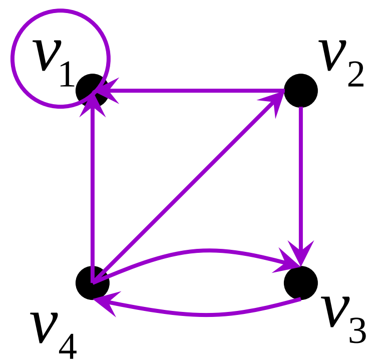
$$[a_{ij}] = \begin{cases} m\,;\{v_i, v_j\} \in E \\ 0 \quad \text{otherwise} \end{cases}$$



$$\begin{array}{c|c} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{bmatrix} \end{array}$$

Directed graphs

$$[a_{ij}] = \begin{cases} m\,;(v_i, v_j) \in E \\ 0 \quad \text{otherwise} \end{cases}$$



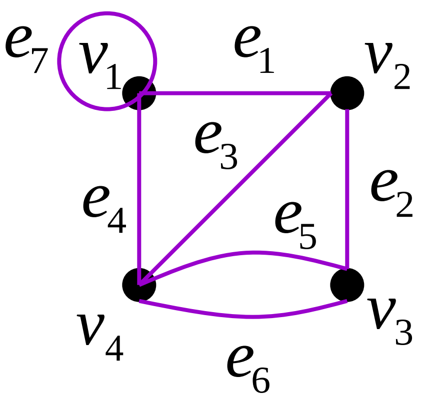
$$\begin{array}{c|c} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Incidence Matrix

Another matrix representation for an **undirected graph** ***G*** = (***V***,***E***) with ***n*** vertices and ***m*** edges is the ***n*** x ***m*** incidence matrix M_G defined as follows:

$$[m_{ij}] = \begin{cases} 1;\ \text{if } e_j \text{ is incident with } v_i \\ 0 \quad \text{otherwise} \end{cases}$$

Before displaying the matrix M_G the vertices and edges *must be labeled* in a certain order



$\Rightarrow$

parallel edges have identical columns

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

loops appear as a column having a single entry = **1**

An **isomorphism** ("isos" means *equal* and "morphe" means *form*) between to simple graphs ***G*** = (***V***,***E***) and ***H*** = (***U***,***F***) is a bijection from ***V*** to ***U*** that preserves edge adjacency. In that case, ***G*** and ***H*** are **isomorphic**.

$$G \approx H \leftrightarrow \exists\ f_{\mathrm{bij}}:V \rightarrow U\ ,\{v_i,v_j\} \in E \rightarrow \{f(v_i), f(v_j)\} \in F$$

Besides finding (possibly) a *one-to-one and onto function* between the vertices of both graphs, the following **invariants** are useful to test for graph isomorphism

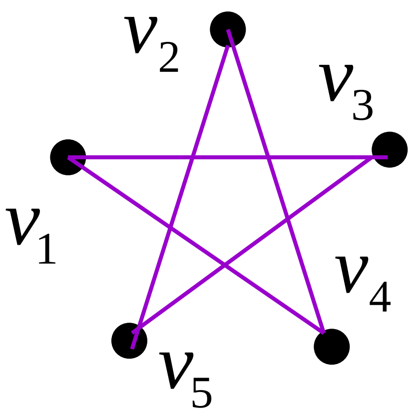
- both graphs must have the same number of vertices and edges,
- the degrees of the corresponding vertices must be the same,

Additional **criteria** to test for graph isomorphism are

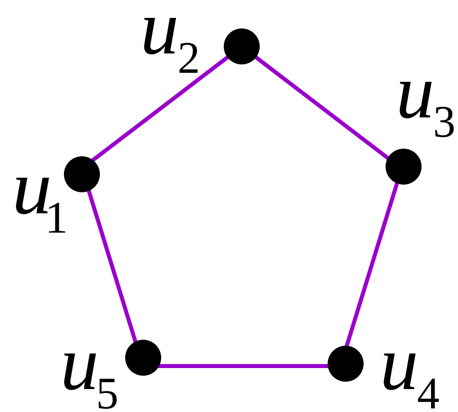
- subgraphs of both graphs made up of vertices with degree **3** and the edges connecting them must be isomorphic,
- the adjacency matrix of the second graph labeled by the bijection ***f*** must be equal to the adjacency matrix of the first graph.

Isomorphism example[a]

- Determine whether the given pair of graphs is isomorphic.



$$G = (V, E)$$



$$H = (U, F)$$

A bijection ***f*** between ***V*** and ***U*** is

$$v_1 \mapsto u_4$$
$$v_2 \mapsto u_2$$
$$v_3 \mapsto u_5$$
$$v_4 \mapsto u_3$$
$$v_5 \mapsto u_1$$

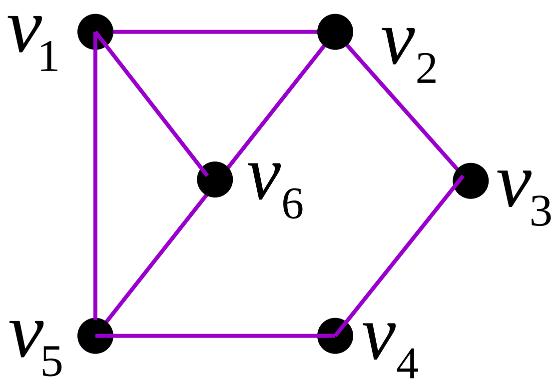
- same # of vertices
- same # of edges
- each vertex has the same degree.

So, we verify using the respective adjacency matrices:

$$\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \qquad \begin{array}{c} u_4 \\ u_2 \\ u_5 \\ u_3 \\ u_1 \end{array} \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

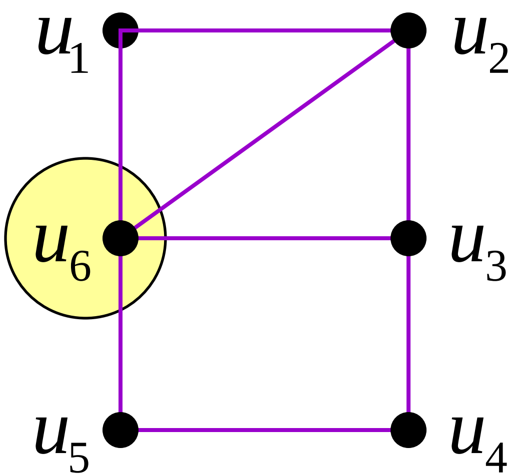
$$\Rightarrow \boxed{G \approx H}$$

- Determine whether the given pair of graphs is isomorphic.



$$G = (V, E)$$

- same # of vertices (**6**)
- same # of edges (**8**)



$$H = (U, F)$$

$$\deg(\{v_1, v_2, v_5, v_6\}) = 3$$
$$\deg(\{v_3, v_4\}) = 2$$

Consider the degree of each vertex:

$$\deg(\{u_1, u_4, u_5\}) = 2$$
$$\deg(\{u_2, u_3\}) = 3$$
$$\deg(u_6) = 4$$

Since vertex $\boldsymbol{u_6}$ of graph ***H*** has degree **4** there is no corresponding vertex in graph ***G*** with the same degree. Therefore, the graphs ***G*** and ***H*** are not isomorphic.

- Show that isomorphism of simple graphs is an equivalence relation.

Consider the set ***S*** of all *finite simple graphs*, let ***G***, ***H***, ***K*** $\in$ ***S***. We show that the relation of isomorphism between graphs is an equivalence on ***S*** if it is *reflexive*, *symmetric*, and *transitive*.

$\boxed{\forall G \in S, G \approx G}$ since $f_{\text{bij}} = id_G$

$\boxed{G \approx H \to H \approx G}$ since $f_{\text{bij}} \to g = f_{\text{bij}}^{-1}$ preserves adjacency between H and G.

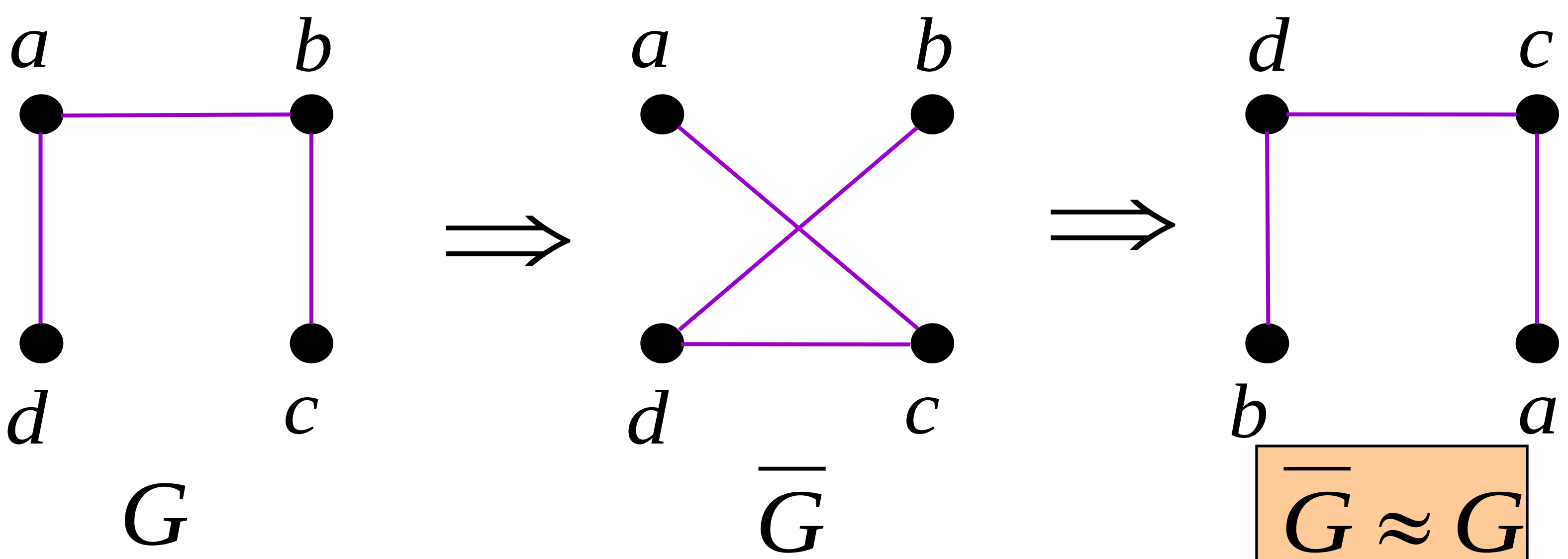
$\boxed{G \approx H \wedge H \approx K \to G \approx K}$ since $h = g \circ f$ is a bijection between G and K preserving adjacency.

The equivalence class [***G***] contains *all simple graphs isomorphic to* ***G***, and the quotient set ***S*** / $\approx$ is the corresponding partition of ***S***.

Isomorphism example[d]

The complementary graph $\overline{G}$ of a simple graph **G** has the same vertices as **G**. Two vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in **G**.

- A simple graph **G** is called **self-complementary** if **G** and $\overline{G}$ are isomorphic.
  Show that the following graph is self-complementary.

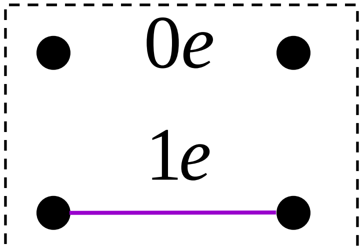


$$\boxed{\overline{G} \approx G}$$

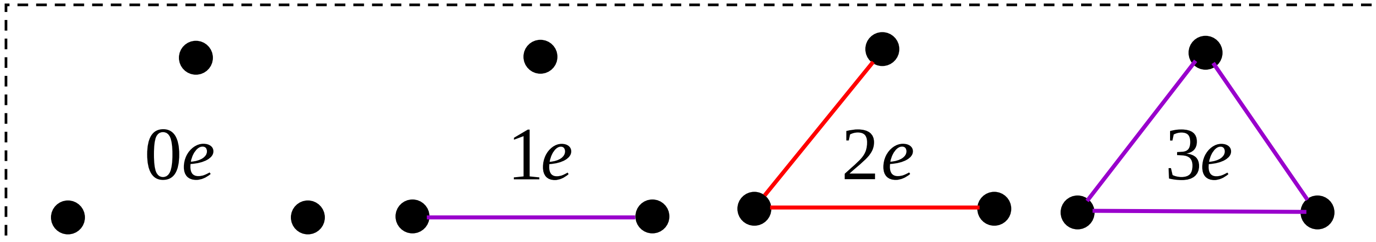
Isomorphism example[e]

- How many nonisomorphic simple graphs are there with ***n*** vertices, when ***n*** is a) **2**?, b) **3**?, and c) **4**?

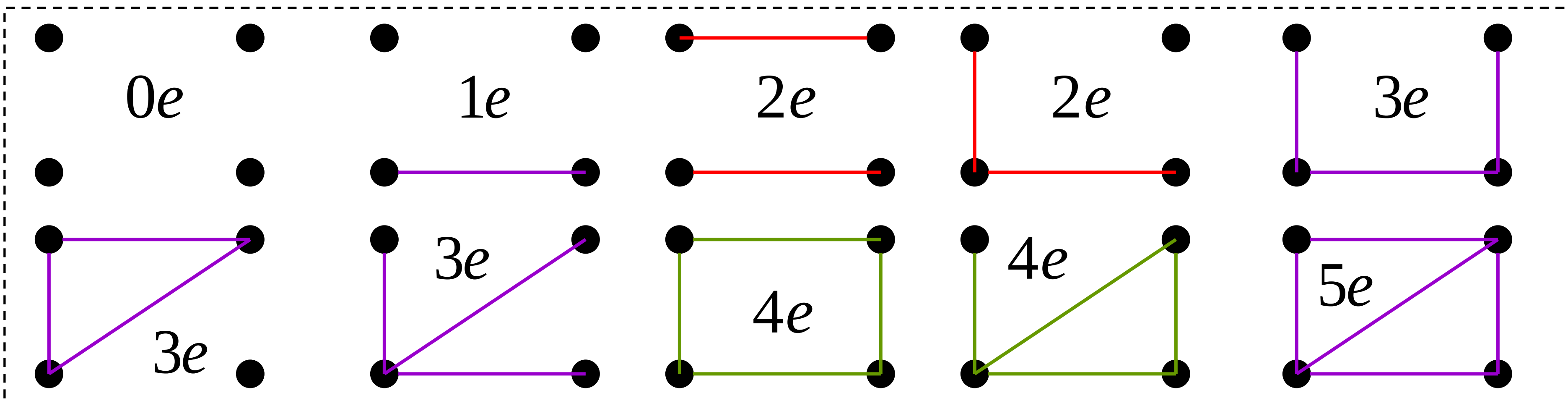
$n = 2$ $\boxed{\#G = 2}$



$n = 3$ $\boxed{\#G = 4}$



$n = 4$ $\boxed{\#G = 11}$



Isomorphism example[f]

- Determine whether the graphs without loops with the following *incidence matrices* are isomorphic.

a) $M_G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, M_H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

Since both incidence matrices are equal after a permutation of columns, graphs ***G*** and ***H*** are *isomorphic*. So, we write:

$$\boxed{G \approx K_3 \approx H}$$

b) $M_G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}, M_H = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

Again, since both incidence matrices are equal after a permutation of columns, graphs ***G*** and ***H*** are *isomorphic*. Both graphs are the same as the *unique graph that exists with* **4** *vertices and* **5** *edges* (see previous slide).