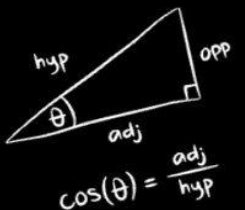
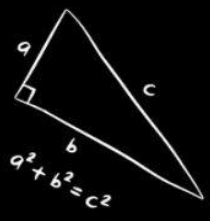
The national higher school of mathematics



$f(x)$

Math Path

YOUR GUIDE TO GO THROUGH



MATH PATH

FIRST SEMESTER ENGLISH PROJECT

THIS PROJECT IS MADE BY :

- **Harrat Amani**
- **Mida Maram**
- **Adjabi yassamine**
- **Kifouche Dounia Zed**
- **Tarmoul Lina**
- **Nadja Rahma**
- **Nouacer ibtihal**
- **Kara insaf**
- **Hadil zaaimia**
- **Medjadji Nour**

Content :

- Statistics and probability
- Analysis
- Algebra
- Physics
- Discrete mathematics
- Algorithm and data structure

Statistics and Probability :

Introduction :

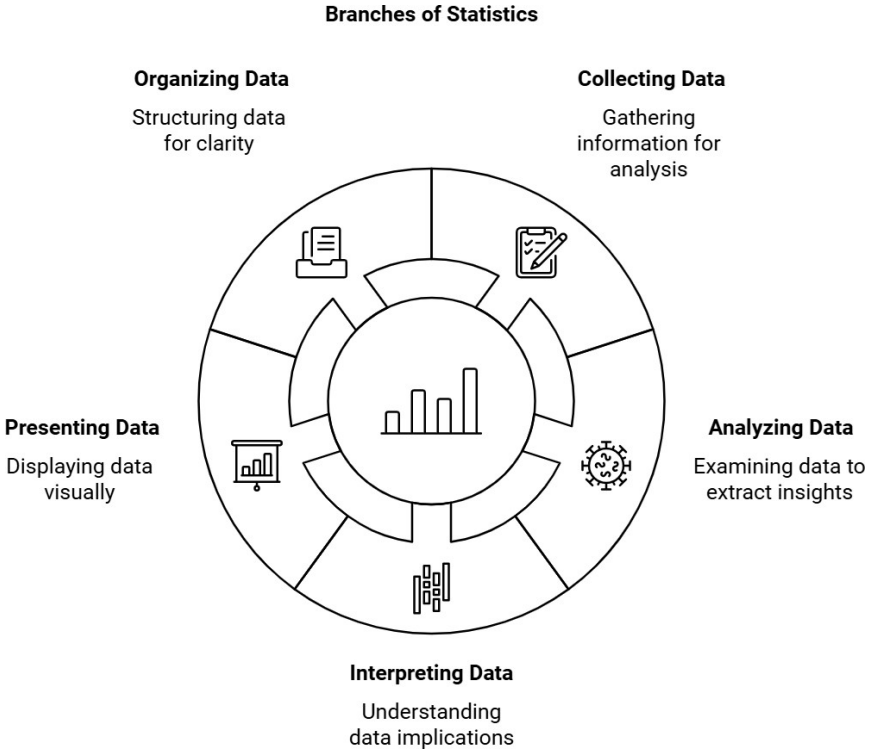
Statistics and probability are interconnected fields in mathematics such that probability quantifies the likelihood of events, calculated as the ratio of favorable outcomes to total possible outcomes, ranging from 0 (impossible) to 1 (certain) .

Statistics involves collecting, analyzing, interpreting, and presenting data, using methods such as descriptive statistics (mean, median) and inferential statistics (hypothesis testing) to draw conclusions about populations based on samples.

Both disciplines are essential for data analysis in various fields, including business and science.

Statistics :

The branch of mathematics that deals with collecting, analyzing, interpreting, presenting, and organizing data. It helps in making decisions based on data analysis and involves methods like averages, probability, and regression analysis.



Population :

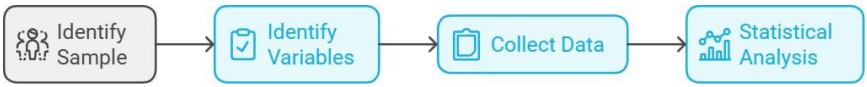
All the members of a group about which you want to draw a conclusion.

Sample :

Sample is the part of the population selected for analysis.

Variable :

A characteristic of an item or an individual that will be analyzed using statistics.



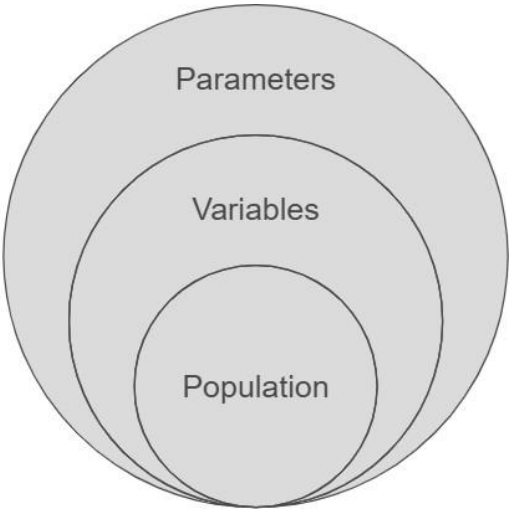
Parameter :

The parameter is a numerical measure that describes a variable (characteristic) of a population.

Numerical measures describing variables

Characteristics analyzed within the population

The entire group being studied

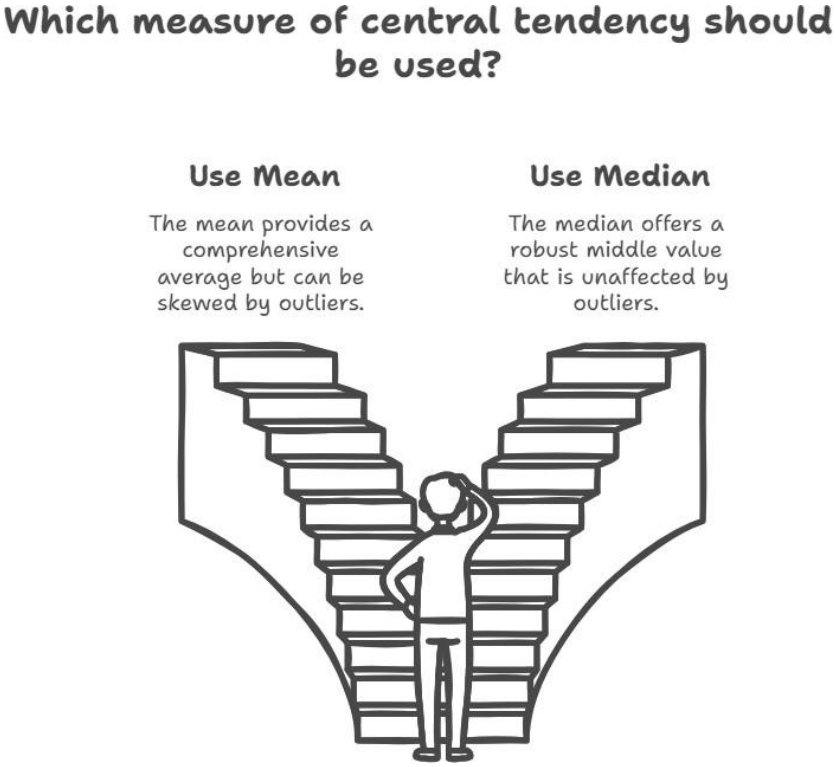


Mean :

The mean is the average of a set of numbers. It's found by adding all the numbers together and dividing by how many numbers there are.

Median :

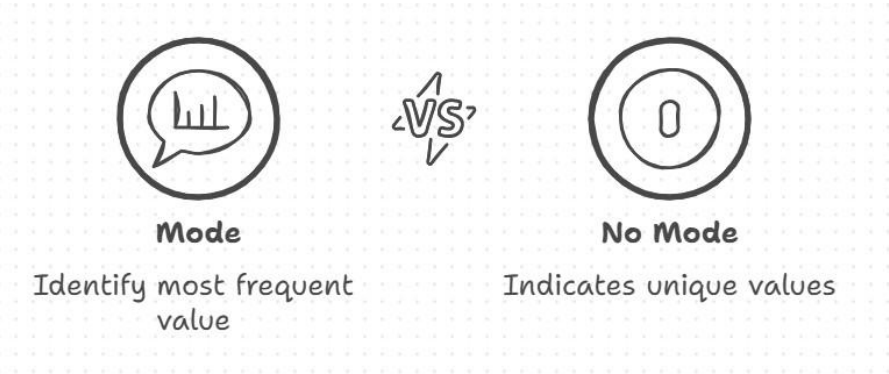
The middle number in a sorted list of numbers.



Mode :

The mode is the value that appears most frequently in a set of numbers.

If no number repeats, the set has no mode. If there is more than one value that repeats the most, the set is multimodal.

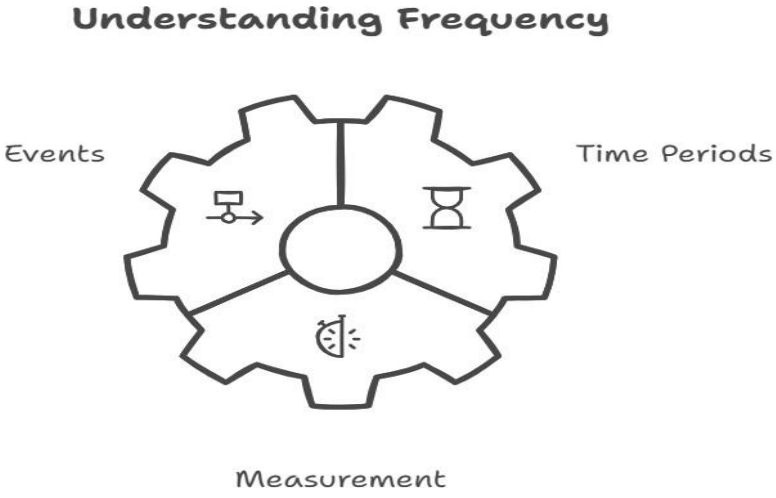


Survey :

The survey is one of the most common methods of collecting data. The person is interviewed based on a questionnaire. Surveys can be carried out via telephone or in writing (and with greater frequency, via the internet). Respondents give their own answers in all cases . it is an important source of statical data.

Frequency :

Is the number of times an event or occurrence happens within a specific period.



Range :

The range is a method of measuring dispersions. It is the difference between the largest and the smallest value in set of data. R=L – S .

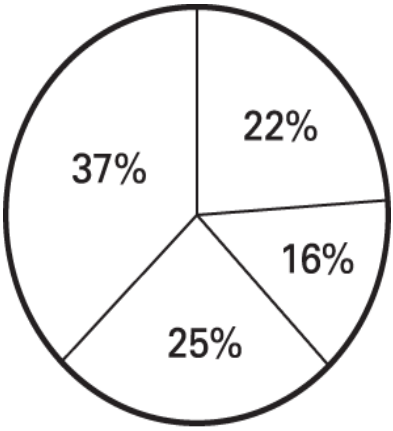
Diagram :

A diagram is a graphical representation of data, information, or a concept, often used to illustrate patterns, relationships, or structures. Examples include bar charts, line graphs, pie charts, and flowcharts.

Pie chart :

A pie chart is a circular graph in which numerical data are represented by sectors of a circle.

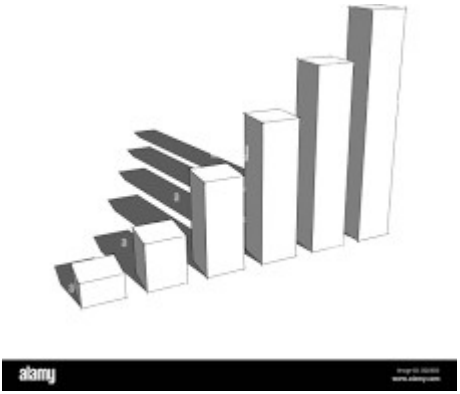
The angles of the sectors are proportional to the frequencies of the items they represent.



pie chart

Bar chart :

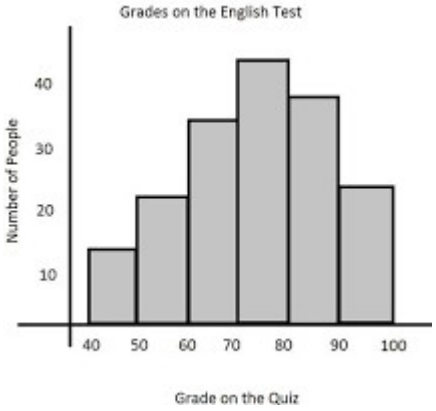
A graph that uses bars to represent data, with the length of each bar showing the value of the data.



Histogram :

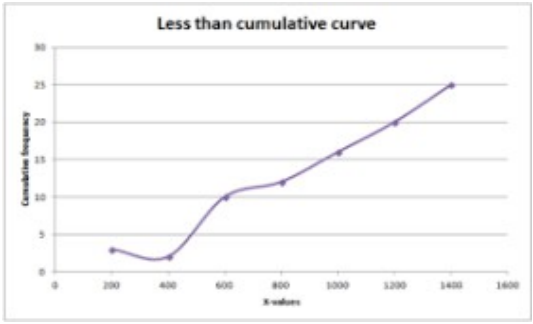
A histogram is a graph that shows the distribution of data by displaying bars.

Each bar represents the frequency of data within a specific range or interval.



Ogive Graph / Cumulative Frequency Polygon :

An ogive / sometimes called a cumulative frequency polygon, is a type of fre- quency polygon that shows cumulative frequencies. In other words, the cumulative percents are added on the
graph from left to right.

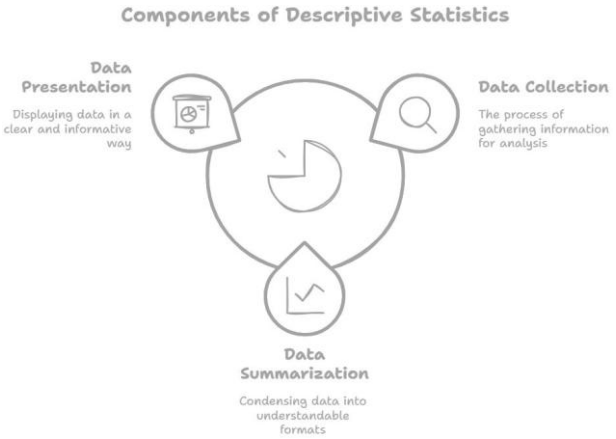


Cumulative frequency :

Cumulative frequency is the running total of frequencies in a data set. It shows the number of data points less than or equal to a certain value.

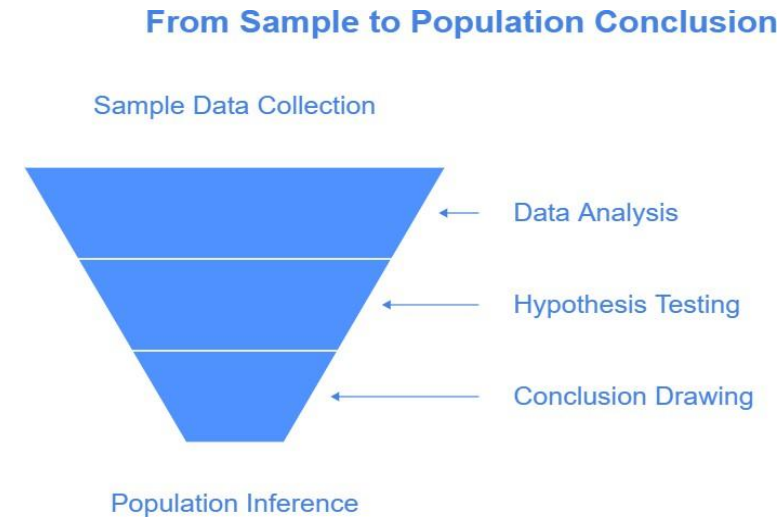
Descriptive Statistics :

Is the branch of statistics that focuses on collecting, summarizing, and presenting a set of data.



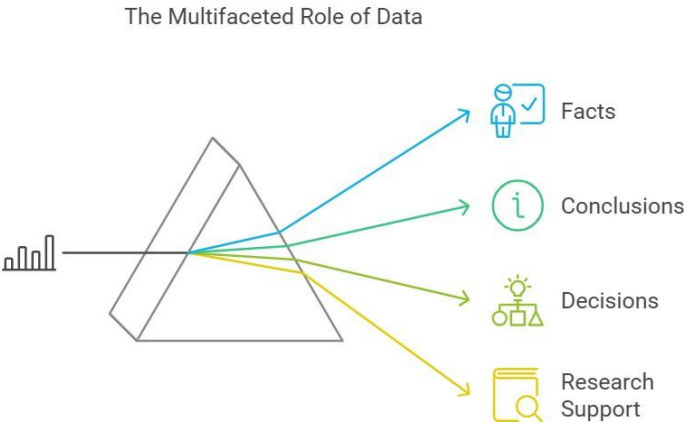
Inferential Statistics :

The branch of statistics that analyzes sample data to reach conclusions about a population .



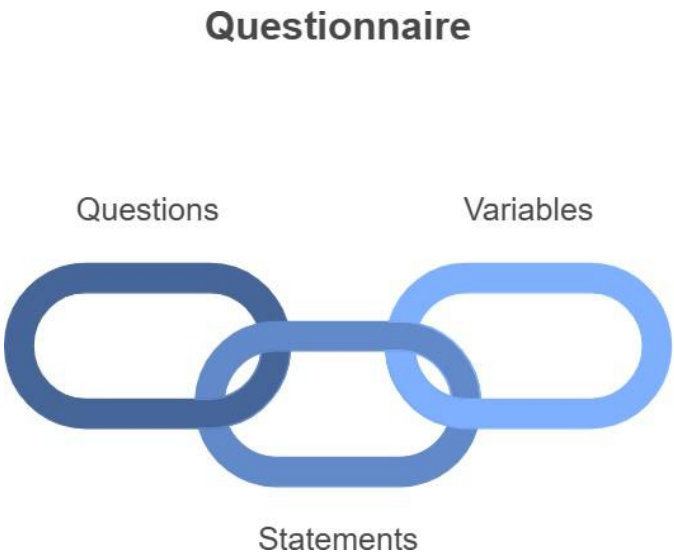
DATA :

Facts or information collected for analysis or reference.



Questionnaire :

A set of questions or statement is assembled to get information on a variable (or a set of variable). The entire package of questions or statement is called a questionnaire.

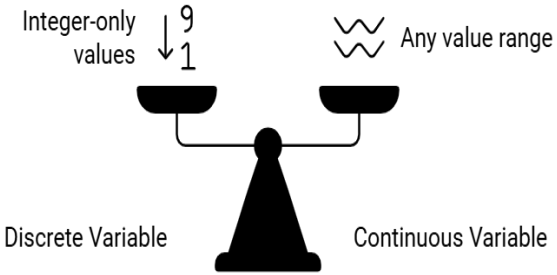


Discrete variable :

Is one whose value change by steps. Its value may be obtained by counting. It normally takes integer values.

continuous variable :

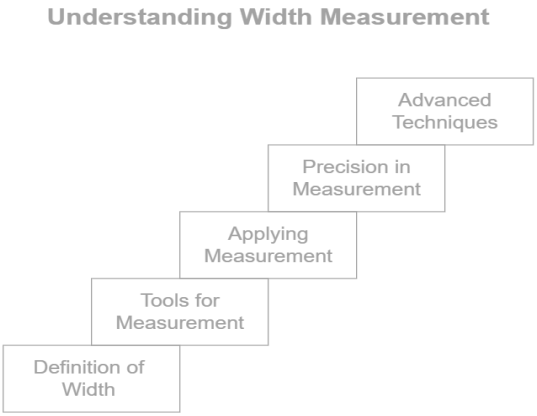
is a variable that can take any value within a range, including decimals and fractions.(e.g., height, weight, time, temperature)



Comparing Data Type Flexibility

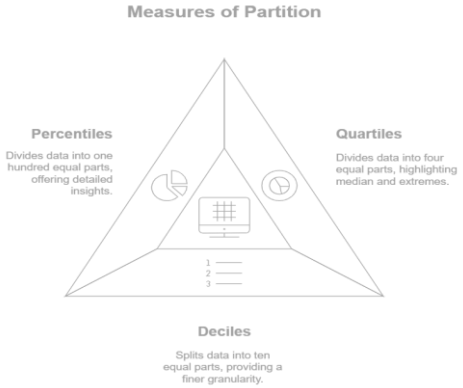
Width :

The measurement of something from side to side ; how wide it is.



Measures of partition :

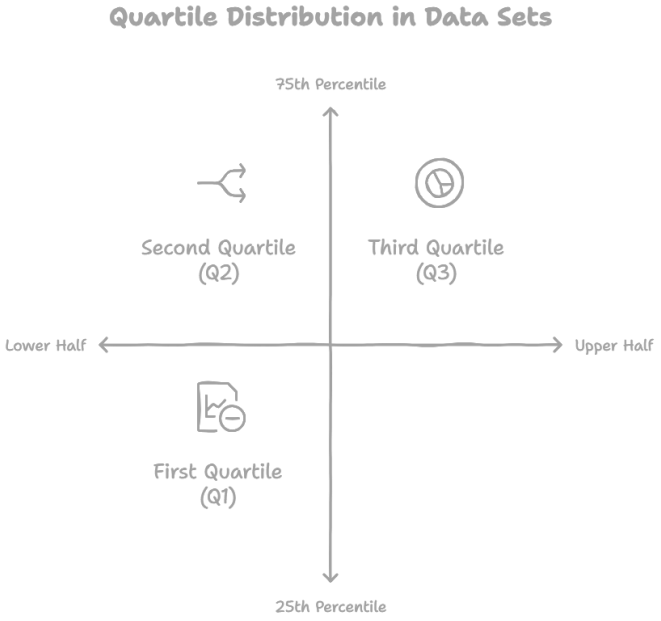
divides a set of data (in an array) into diferent equal parts. Such data must have been arranged in order of magnitude. Some of the partition values are: the quartile, deciles and percentile.



Quartiles :

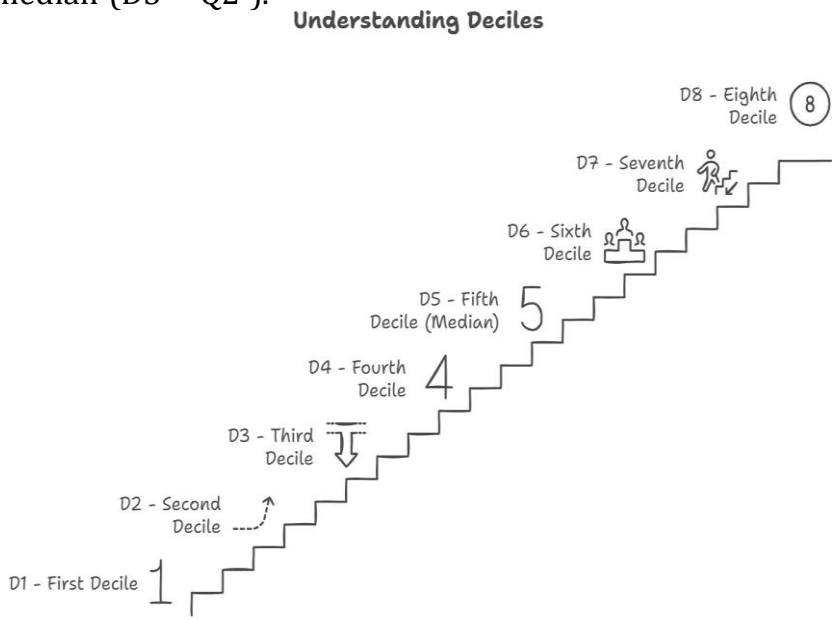
Quartile are values that divide a data set into four equal parts. There are three quartiles:

1. First Quartile (Q1): The median of the lower half of the data (25th percentile).
2. Second Quartile (Q2): The median of the entire data set (50th percentile).
3. Third Quartile (Q3): The median of the upper half of the data (75th percentile).



DECILES :

The values of the variable that divide the frequency of the distribution into ten equal parts are known as deciles and are denoted by D1; D2; …..; D9: the fifth deciles is the median (D5 = Q2).

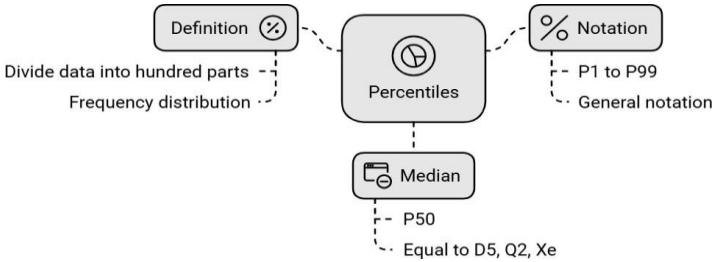


PERCENTILE :

The values of the variable that divide the frequency of the distribution into hundred equal parts are known as percentiles and are generally denoted by P1; ….; P99:

The fiftieth

percentile is the median (P50 = D5 = Q2).



The box-plot :

A box plot shows the distribution of data using a box for the interquartile range (Q1 to Q3), a line for the median, and whiskers for the min and max values. Outliers are shown as dots outside the whiskers.

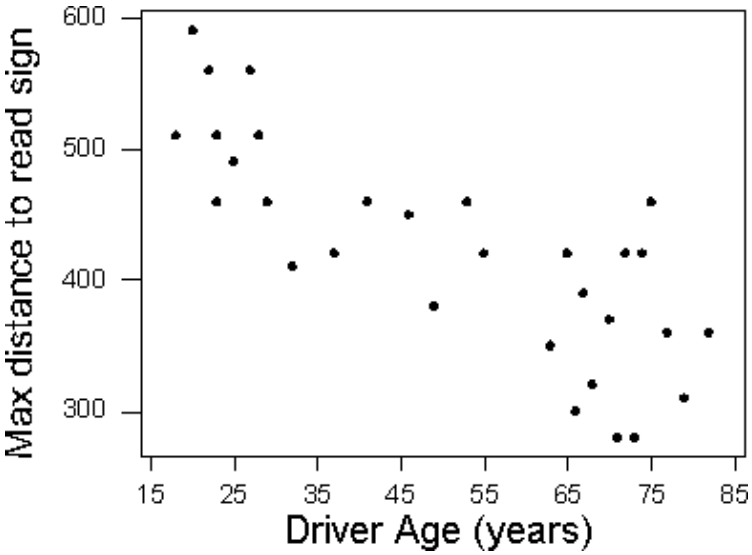
Dispersion :

Dispersion shows how spread out the data is. Common measures include:

Range: Difference between the max and min values.

Variance: Average squared difference from the mean.

Standard deviation: Square root of variance.



Covariance :

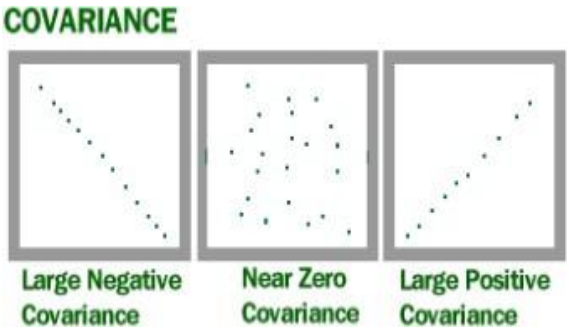
Covariance measures how two variables change together:

Positive covariance: Both variables increase or decrease together.

Negative covariance: One increases while the other decreases.

Zero covariance: No relationship.

It's calculated as the average of the product of the differences from their means.



Empirical Rule :

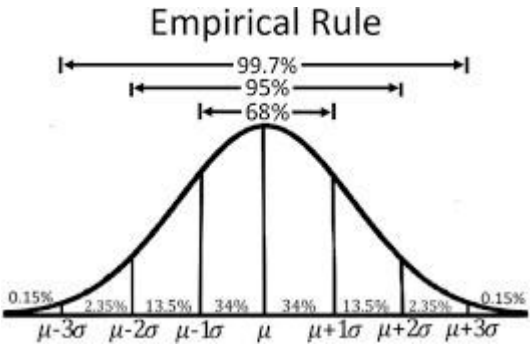
The Empirical Rule (68-95-99.7 Rule) applies to data that follows a normal distribution (bell -shaped curve). It states:

68% of the data falls within 1 standard deviation of the mean.

95% falls within 2 standard deviations.

99.7% falls within 3 standard deviations.

It helps to understand how data is spread around the mean in a normal distribution.



The standard deviation :

The standard deviation, usually denoted by the Greek alphabet (small signal is for the population) is defined as the "positive square root of the arithmetic

mean of the squares

of the deviation of the given observation from their arithmetic mean".

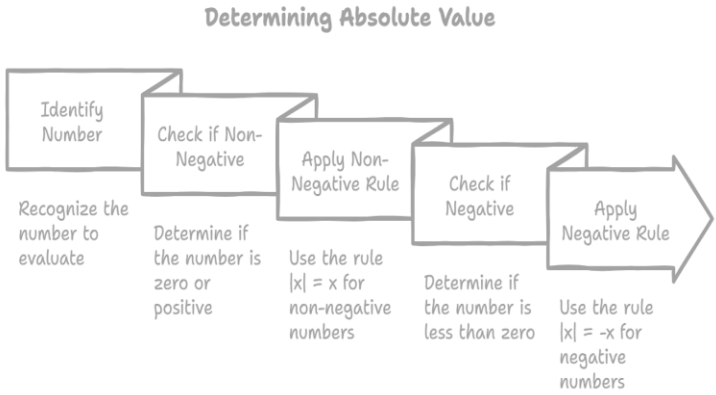
Analysis :

INTRODUCTION :

The term analysis denotes the activities that help us to breaking down complex information, data or systems into smaller parts to get more understandable components so we gain insights, identify patterns, and make informed decisions. which is a necessary and important step in problem-solving and making decisions across various fields.

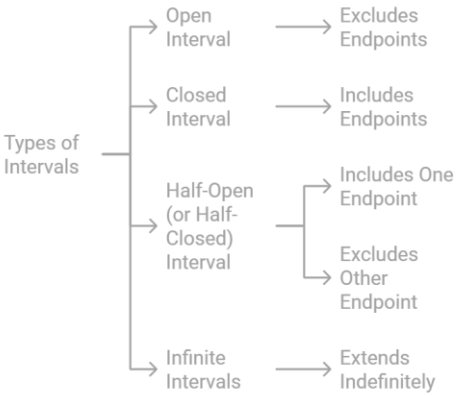
Absolute value :

The absolute value of a number or integer is the actual distance of the integer from zero, in a number line. Therefore, the absolute value is always a positive value and not a negative number. We can define the absolute values like the following: if $a \leq 0$ $|a| = -a$; if $a \geq 0$,$|a| = a$



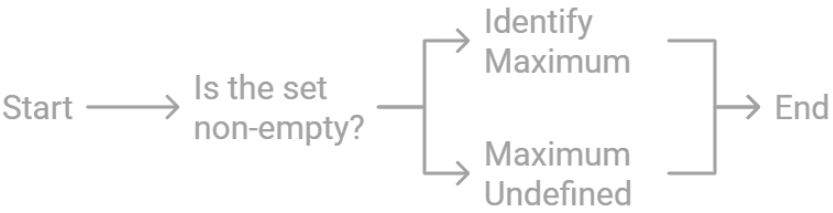
Intervals:

An interval is a range of numbers between two given numbers and includes all of the real numbers between those two numbers.



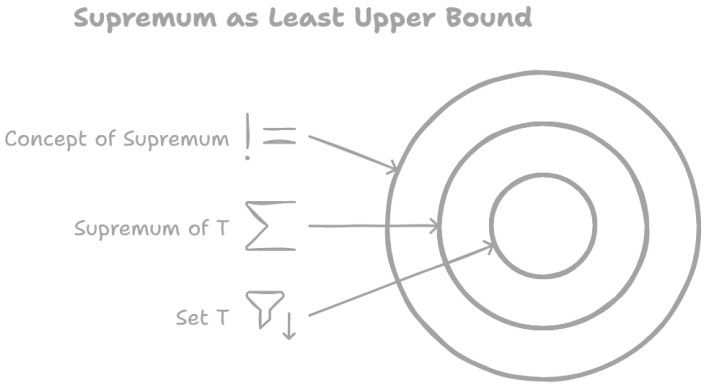
The maximum:

The maximum is the largest number in the set.



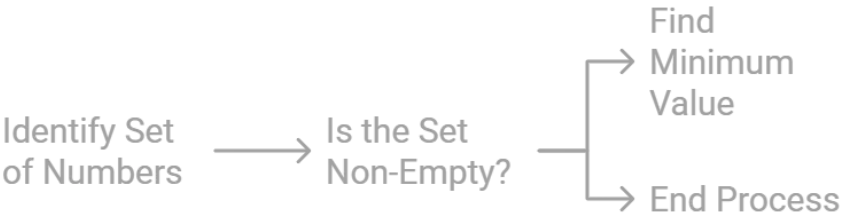
The supremum:

The supremum is the least upper bound number in the set.



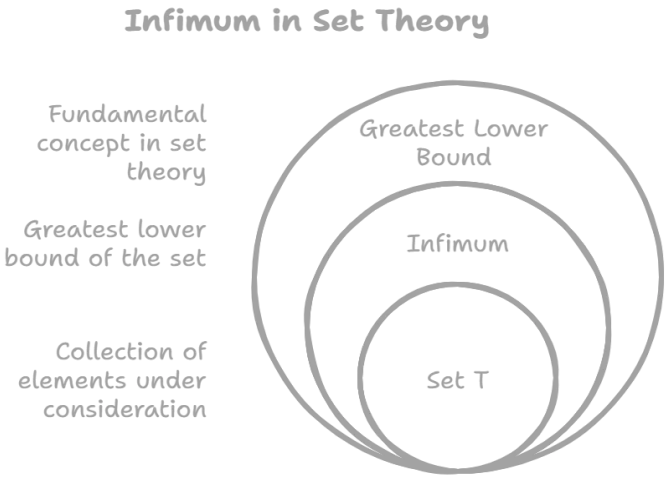
The minimun:

The minimun is the smallest number in the set.



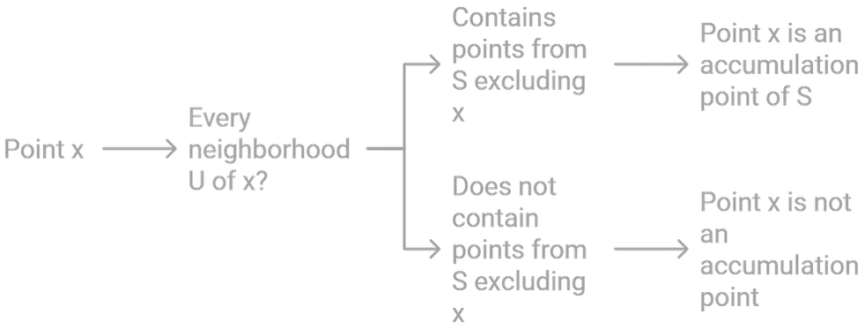
The infimum:

The infimum is the greatest lower bound in the set .



Accumulation Point:

The point x is called an accumulation point of the set E if for any $\varepsilon > 0$ the neighborhood $(x - \varepsilon, x + \varepsilon)$ contains infinitely many elements of E .



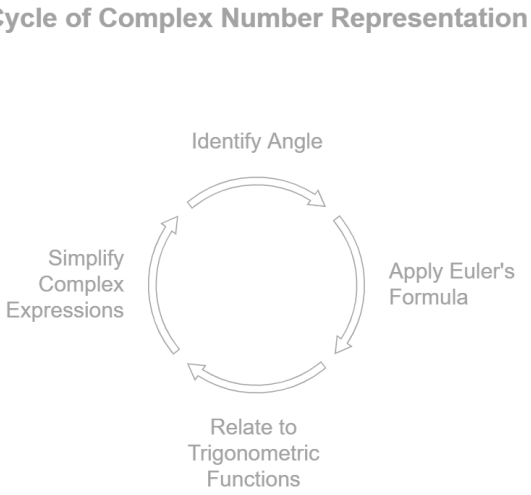
Isolated Point:

The point x is called an isolated point of E if $(x - \varepsilon, x + \varepsilon) \cap E / \{x\}$ for some $\varepsilon > 0$

Euler's formula:

Euler's law states that 'For any real number x, **e^ix = cos x + i sin x**. This complex exponential function is sometimes denoted cis x

("cosine plus i sine"). The formula is still valid if x is a complex number.



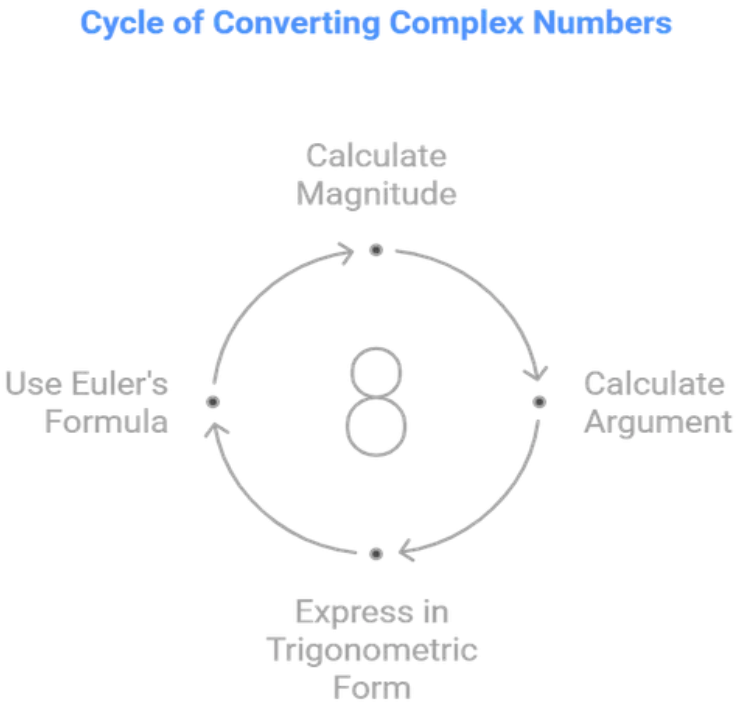
Trigonometric Form:

The trigonometric form of a complex number z = a + b i is $z = r\ \cos\ (\theta) + r\, i \sin$

Exemple :

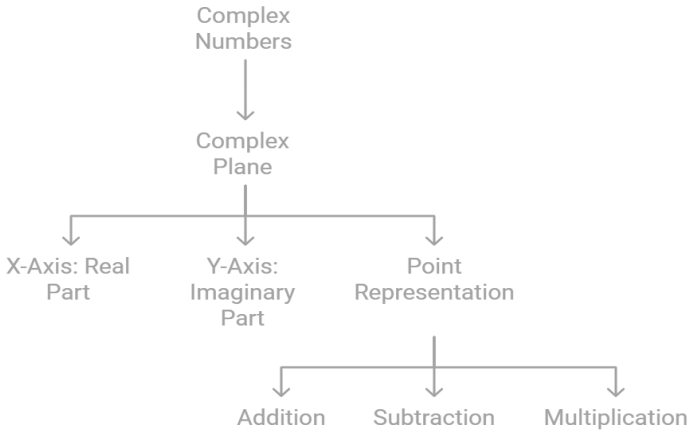
$Z = 4(\cos x + i \sin x)$

$X = 2\pi/3$



Complex Plan:

A complex plane is a coordinate grid on which the x-axis is a real number line, and the y-axis is an imaginary number line. A complex plane is also called an Argand plane or a Gauss plane.



Complex numbers :

The numbers that are expressed in the form of a+ib where, a,b are real numbers and 'i' is an imaginary number called “iota”. The value of i = ($\sqrt{-1}$). For example, 2+3i is a complex number, where 2 is a real number (Re) and 3i is an imaginary number (Im).

The rational numbers :

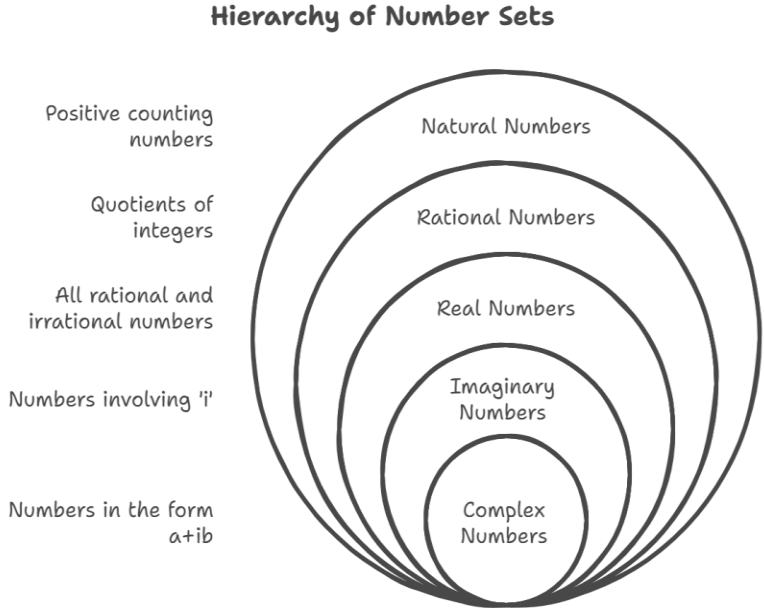
Written $\mathbb{Q}$, is the set of all quotients of integers. Therefore, $\mathbb{Q}$ contains all elements of the for a/b where a and b are integers and b is nonzero.

Integers :

The number zero (0), and positive natural numbers (1, 2, 3, . . .), and negative numbers (−1, −2, −3, . . .). The set of all integers is often denoted by the
boldface Z .

Natural numbers :

All positive integers from 1 to infinity. They are also called counting numbers as they are used to count objects.



Prime numbers :

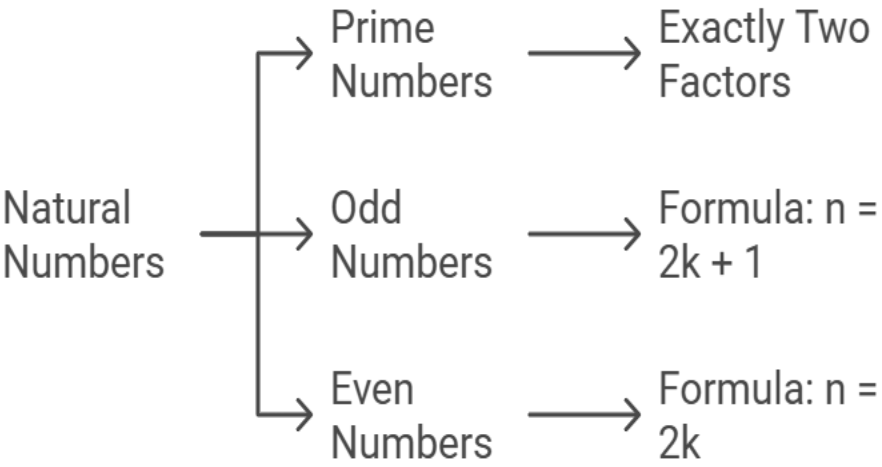
Natural numbers that are divisible by only 1 and the number itself. In other words, they are positive integers greater than 1 with exactly two factors, 1 and the number itself.

the odd numbers :

An integer of the form n=2k+1, and not divisible by 2.

the even numbers :

An integer of the form n=2K, and divisible by 2.



Algebra :

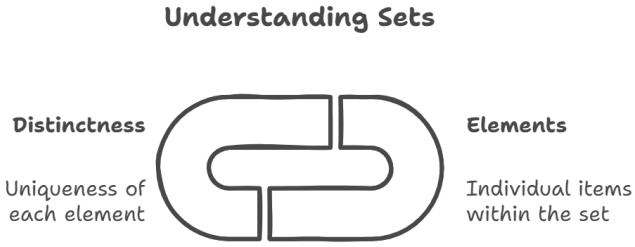
Introduction :

Algebra is a branch of mathematics that uses symbols or often letters to represent numbers and operations. It involves the study of mathematical relationships, equations, and the rules for manipulating these symbols for problem-solving. It also provides a foundation for more advanced areas of mathematics and solving real-world problems as science, engineering, and everyday's problem-solving.

Set :

Is a collection of distinct objects or elements (numbers , letters ...) for example

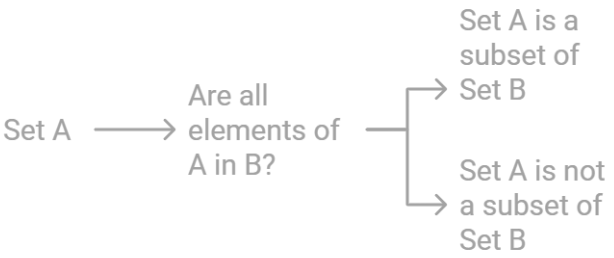
{1,2,7,9}



Subset :

A set A is said to be a subset of a set B if all elements of A are included in the set B and we write $A \subseteq B$

Example : A ={1,2 } ; B = {0,1,2,7,9} .



Union :

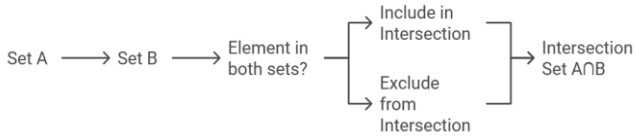
The union of two sets A and B is the set that contains all the elements which are either in A or B and is denoted by $A \cup B$

Example : let A ={9} and B ={1,3} the union of A and B is $A \cup B$ ={1,9,3}

Intersection :

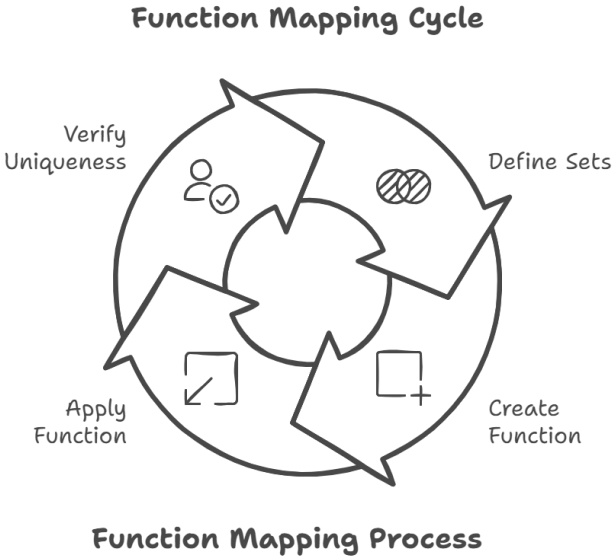
The intersection of two sets A and B is the set that contains all the elements which belong to both A and B and we denote it by $A \cap B$

Example : let A={4,6,8,9} and B={1,3,6,9} the intersection of A and B is $A \cap B$ ={6,9}.



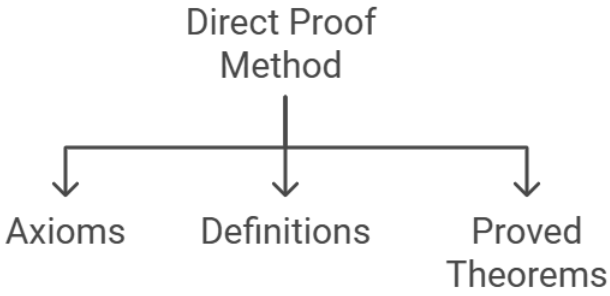
Map :

let A and B two sets , a map or a function f from A into B is a process by which for every element of A associate a unique element of B for example : Let f:Z→Z be the map defined by f(x)=2x



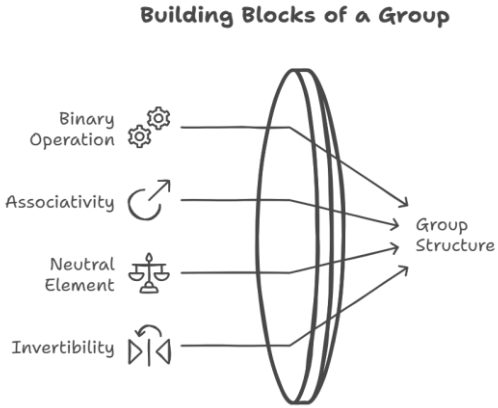
Direct proof :

is a serval method of proof used to establish the truth of mathematical statements , in this method we use axioms , definitions and proved theorem directly .



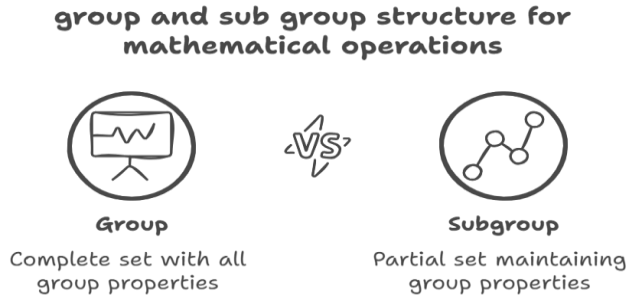
Group :

a group is an non empty set G along with a binary operation (multiplication, addition ...) which is associative , admits a neutral element and every element of the set G is invertible with respect to the binary operation .



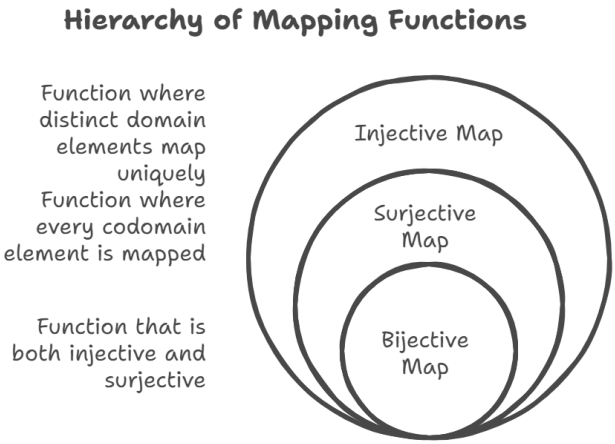
Subgroup :

a subgroup is a subset of a group that forms annother group under the same operation as the original group G



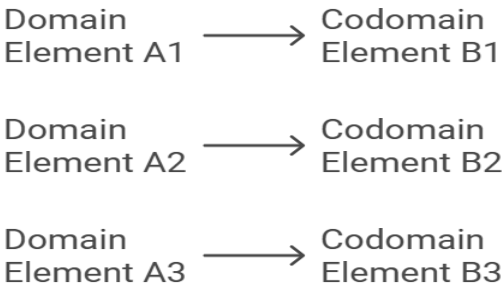
Injective map :

we can say that a map is injective or "one to one" function if distinct element in the domain has a distinct images in the codomain



Surjective map :

we say that a map is surjective or "onto" map if any element in the codomain is mapped to some element of the domain

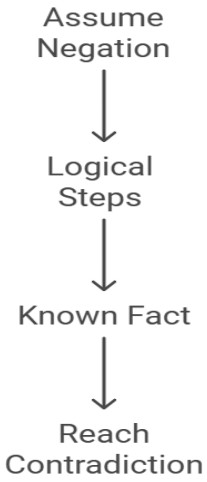


Bijective map :

We say that a map is bijective if it is surjective and injective at the same time .

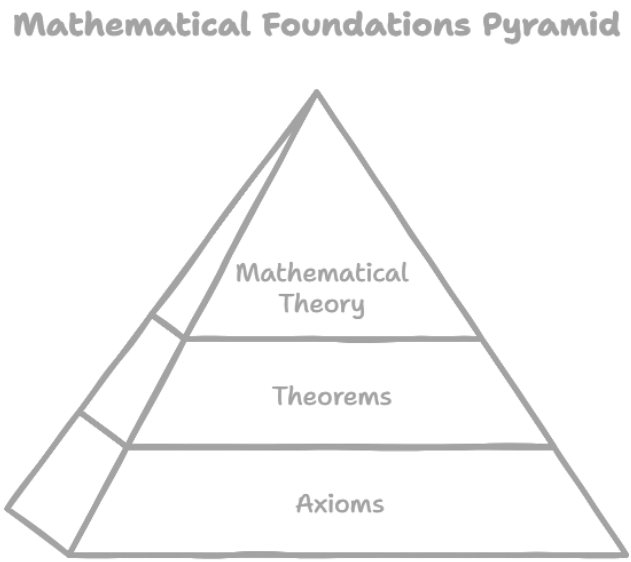
Proof by contradiction :

This method of proof is works by supposing the negation of a statement to be proven , then showing that this assumption result in a contradiction with a fact .



Axiom:

Is mathematical statement supposed to be true without proof, and it used as the starting point for deducing other statements.
Eg. The whole is greater than the part.

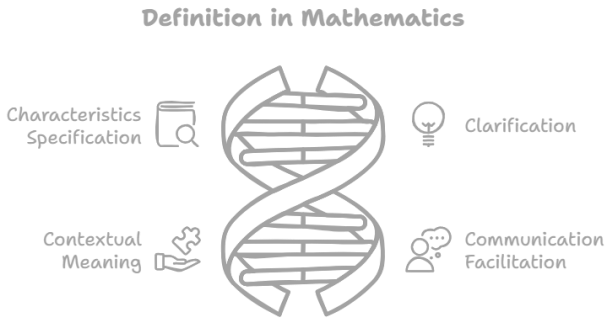


Theorem :

A statement that can be proved to be true based on axioms and definitions Eg. pythagorean Theorem .

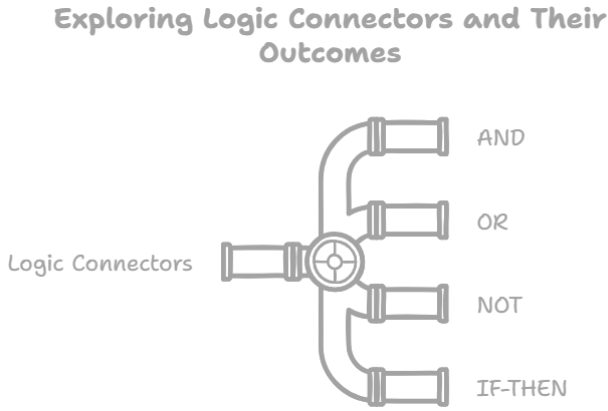
Definition:

Is a statement that defines mathematical objects. Eg. Prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.



Logic connectors:

Are tools used to create from given propositions new propositions whose truth values depend on the truth values of the old one.

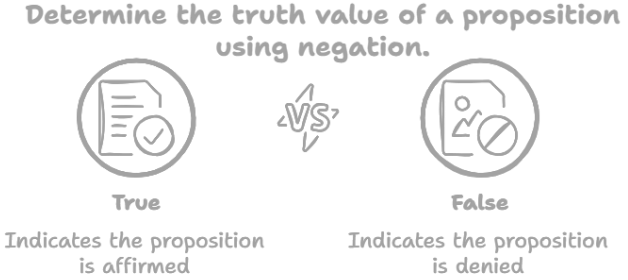


The negation:

Is logic operator based on :the negation of proposition p is true when P is false and false when p is true.

Eg. The proposition: for $\forall x \in R$, $x^2=0$

The negation: $\exists x \in R$, $x^2 \neq 0$



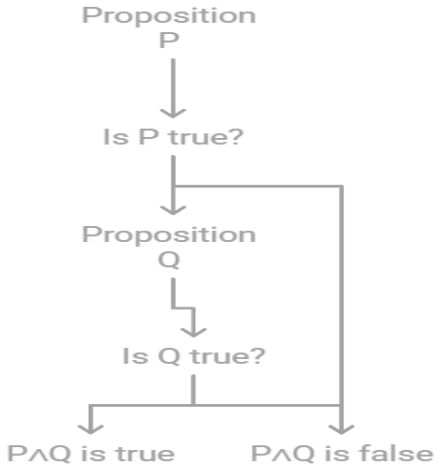
The conjunction:

Is a logic operator, based on that the conjunction of the proposition p and Q. is the proposition

Denoted $P \wedge Q$ which is true when both P،Q are true, and false in all other scales.

Eg.If p represents "It is raining" and q

represents "It is cold," then p∧q means "It is raining and it is cold.

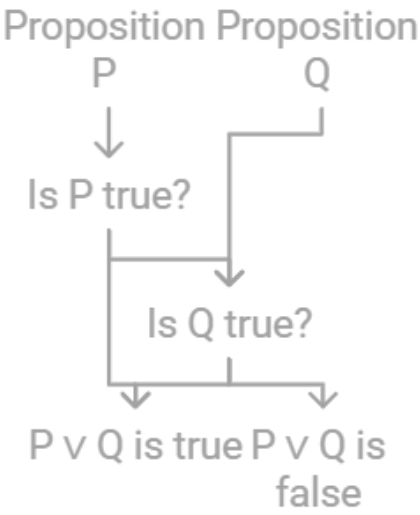


The disjunction:

Is logic operator based on: the disjunction of the proposition

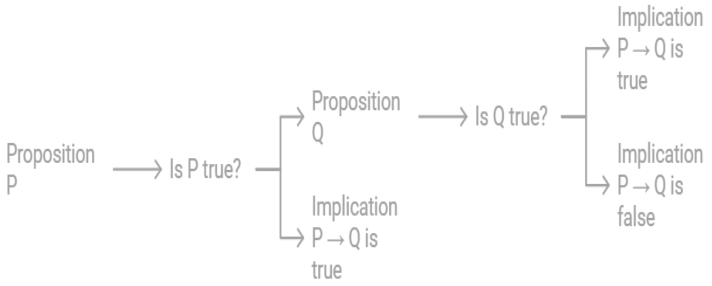
P and Q, is the proposition denoted pVQ which is true if at least one of the two statements p,Q is true, and false when both statements are false.

Eg.If p represents "It is sunny" and q represents "It is warm," then p∨q means "It is sunny or it is warm (or both)."



Implication :

The implication of the propositions P and Q denoted by $P \rightarrow Q$; is the proposition which is false when P is true and Q is false, and true in all other cases,Eg.$n \geq 2 \rightarrow n^2 \geq 4$



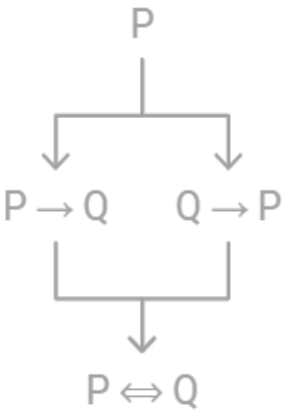
equivalence :

Also called biconditional , the equivalence from P to Q, denoted $P \iff Q$, is defined to be the proposition $(P \rightarrowtail Q) \wedge (Q \longrightarrow P)$.

Eg :

$x^2=9 \iff x=\pm 3$

This means "$x^2=4$ if and only if $x=\pm 3$."



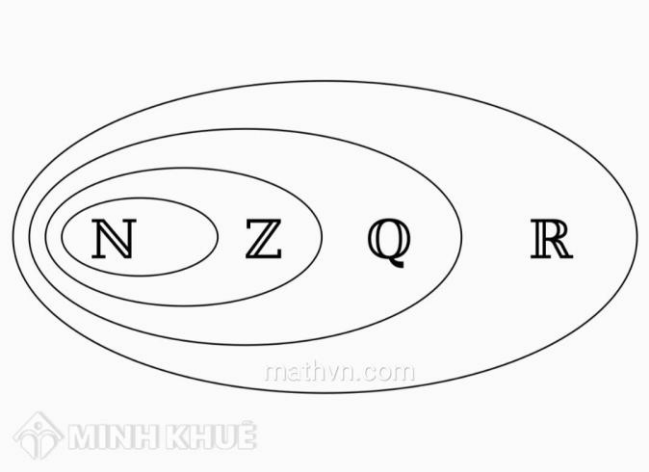
inclusion:

Means that one mathematical object is contained within another.

Eg. Let A ,B be two sets

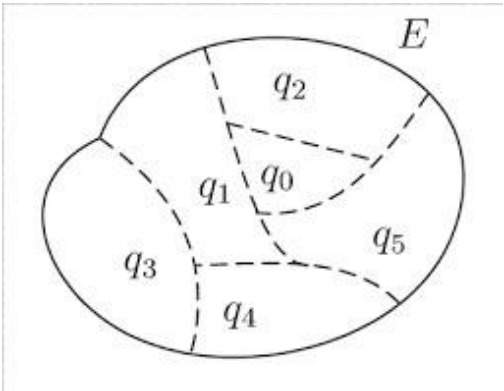
A:{1,2,3}B:{1,2,3,4,5}IN then A⊂B

It means that A is included in B .



partition:

A partition of a set based on the dividing of this set into disjoint subsets such that every element of the original set belongs to exactly one subset and the union of all the subsets equals the original set .



Restriction:

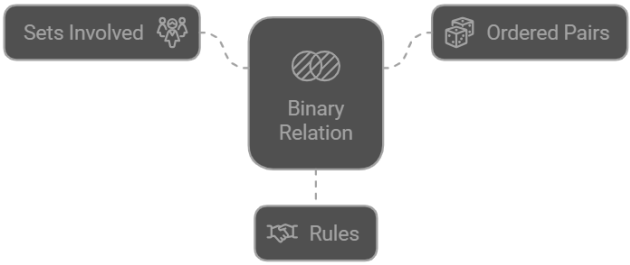
The restriction of a function means limiting its domain to a smaller subsets.

Eg. Function: f(x)=sin(x) where f:R*R

Restriction: If we restrict f(x)=sin(x) to the domain C=[0,π] the restricted function is

f|c(x)=sin(x) for x∈[0,π].

This means the sin function is now only considered on the interval from 0 to π.



Extension :

The extension of a function means defining it on a larger domain.

Physics :

Introduction :

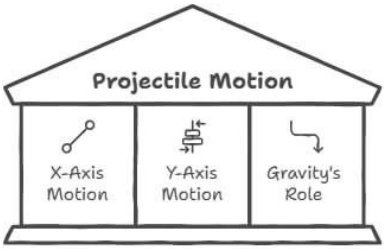
Physics is the application of scientific method to discover the laws

Governing phenomena subject to cause-and-effect relationships

"Knowledge of the present leads to predict the future".

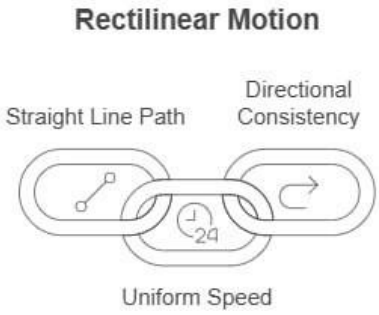
The projectile motion :

Projectile motion is the motion of an object when we throw it or projectile it we study it along X axis and Y axis and it's subjected only to its gravitational force .



The rectilinear motion :

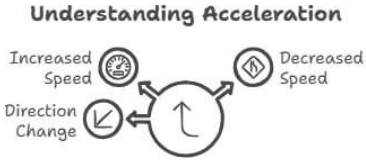
Rectilinear motion is the motion of an object in a straight line like when car moves in the road .



The acceleration :

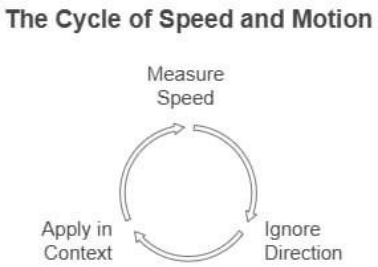
An object is said to be accelerated if there is a change in its velocity. The change in the velocity of an object could be an increase or decrease in speed.

or a change in the direction of motion.



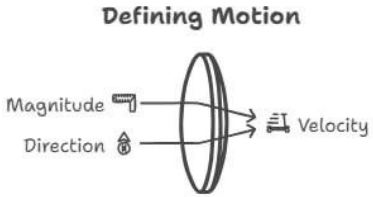
The speed :

Is a scalar quantity It represents how fast an object is moving, despite of its direction Saying Ariel the CAT runs at 9 km/h is a speed.



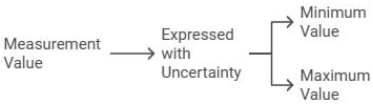
The velocity :

Velocity is basically speeding in a specific direction. It is a vector quantity, which means we need both magnitude and direction to define velocity .



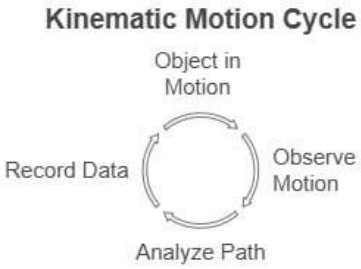
Uncertainty :

Refers to the doubt or variability in a value For example, if the value of a resistance is written: R =(3.4;0.1)Ω then the uncertainty is 0.1



Kinematics :

Studies the motion of objects without considering the forces that cause the motion for example : A person skydiving or An apple that falls from a tree .



The scalar :

Is quantity that is completely described by its magnitude. Examples of scalars are volume, speed, energy ,mass , have both magnitude and direction .

VECTOR :

It is a quantity that has both magnitude and direction .

For example, displacement, velocity, and acceleration are vector quantities .

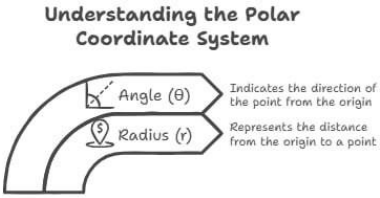
circular motion :

The motion of an object moves such that its distance from a fixed point is remains the same for example the motion of rolle coaster .

The polar system of coordinate :

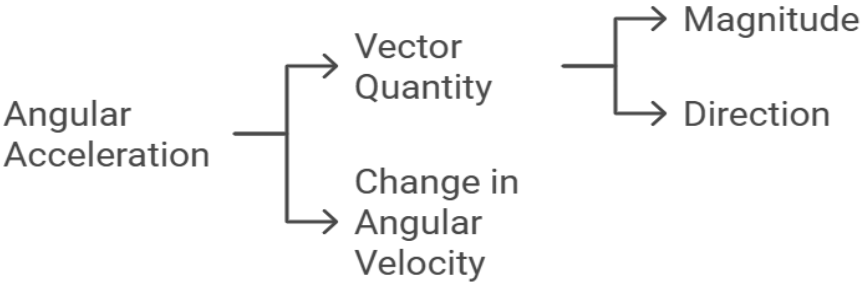
Is a two dimensional coordinate system This coordinates are expressed as (r,θ)

.



The angular acceleration :

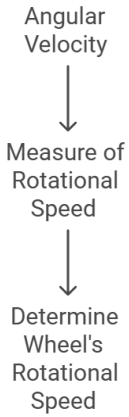
It is a vector quantity that has both magnitude and direction and is also known as rotational acceleration , it tells you how quickly an object's angular velocity changes over time .



The angular velocity :

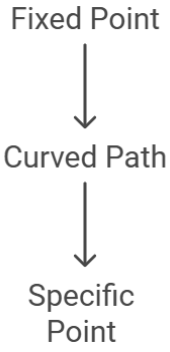
It tells you how fast something is rotating in a circular motion and it can helps us to

determine how fast a wheel can rotate in a given period .



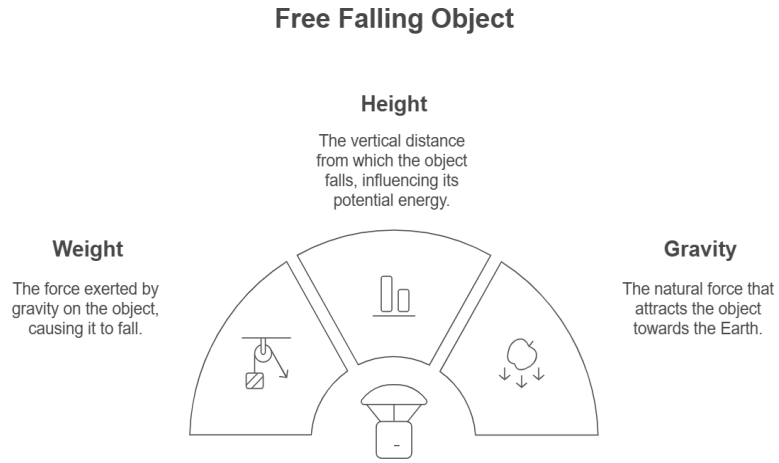
the curvilinear abscissa :

It refers to the length of the curved path from a fixed point to a specific point on that path .



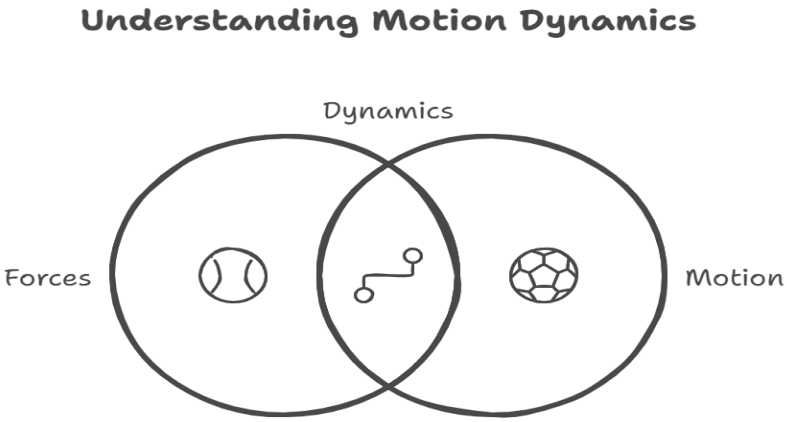
free falling object :

It is the falling of an object from a hight where it is subject to only its weight for example :a ball falling from above .



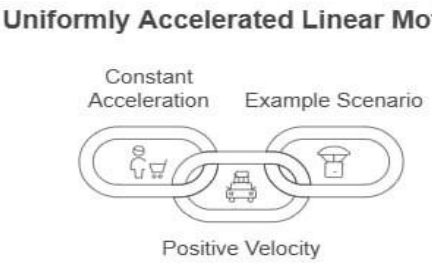
Dynamics :

It is the branch of classical physics which takes into account the forces for example :If you throw the ball with a certain force (taking into account air resistance), you can calculate the net force acting on the ball, its mass, and thus determine its acceleration and how it affects the ball's motion.



Uniformaly acceleratered linear motion:

It is a type of linear motion when we take the product between acceleration and velocity the value is positive for example :. a ball fall from the top of a ladder .



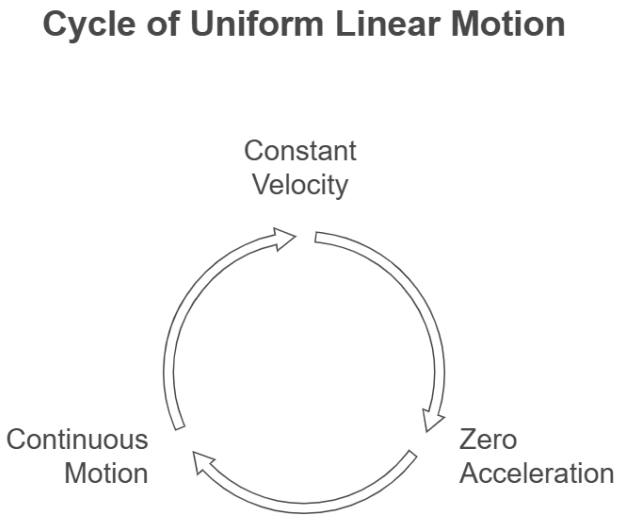
Uniformaly decelerated linear motion :

It is a type of linear motion when we take the product between acceleration and velocity the value is negative for example :when we throw a ball straight up in

the air when it rises .

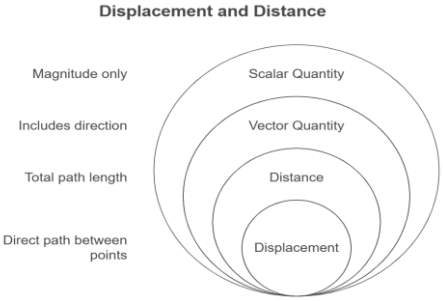
Uniformaly linear motion :

It is a type of linear motion when we take the product between acceleration and velocity the value is zero for example : the motion of the earth around the sun .



Displacement :

Is a vector quantity as it depends upon both magnitude and direction and it's the direct length between any two points when measured along the minimum path between them .



Distance :

Distance is a scalar quantity as it only depends upon the magnitude and not the direction and it is complete length of the path between any two points .

Discrete Mathematics :

Introduction :

Discrete mathematics is a field of mathematics that deals with discrete (non- continious) and countable objects. Which focuses on the study of structures that are distinct and separate such as numbers, logical statement and algebraic structures . Following , you will find the most important terms used in this module .

FACTORIAL :

The product of all positive integers less than or equal to a given positive integer denoted by that integer and an exclamation point.

Thus, factorial seven is written 7! meaning 1 × 2 × 3 × 4 × 5 × 6 × 7.

By definition ; 0! = 1

MULTIPLICATION PRINCIPLE :

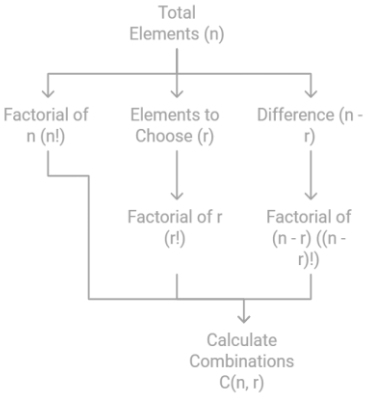
The fundamental counting principle (fcp) is a “big idea” in mathematics. It says that if an event E can occur in a ways and event F can occur in b ways, then event E followed by event F can occur in a × b ways. It is easy to understand and easy to apply. For instance, if Ng has 3 sweatshirts and 2 pairs of jeans, he has the possibility of 3 × 2, or 6, different outfits.

ADDITION PRINCIPLE :

The addition principle, or in other words rule of sum, is a fundamental concept in combinatorics which states that if there are A ways to perform one action and B ways to perform another, and these actions cannot occur simultaneously, then there are A+B ways to choose one action.

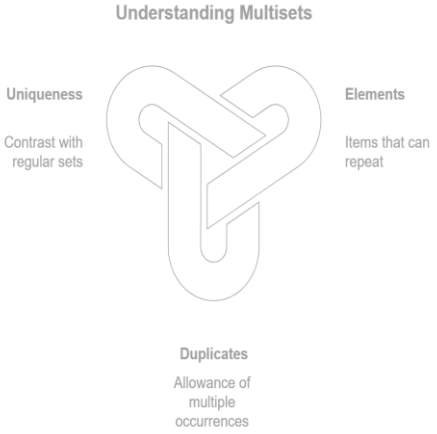
COMBINATION :

A combination is a mathematical technique that determines the number of possible arrangements in a collection of items where the order of the selection does not matter. In combinations, you can select the items in any order.



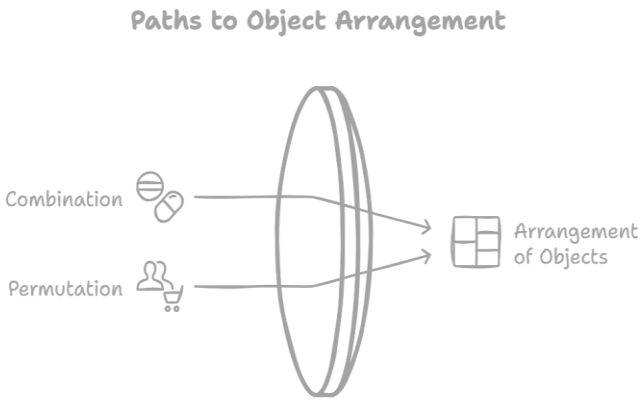
MULTISET :

A multiset is an unordered collection of elements, in which the multiplicity of an element may be one or more than one or zero.



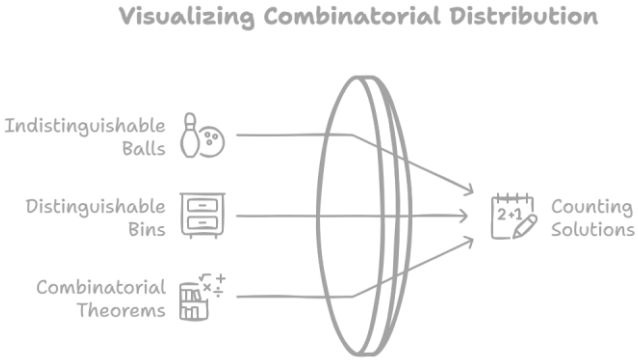
ARRANGEMENT :

An arrangement of objects is simply a grouping of them. The number of "arrangements" of items is given either by a combination (order is ignored) or permutation (order is significant).
for example arranging 5 people in 5 seats .



STARS AND BARS :

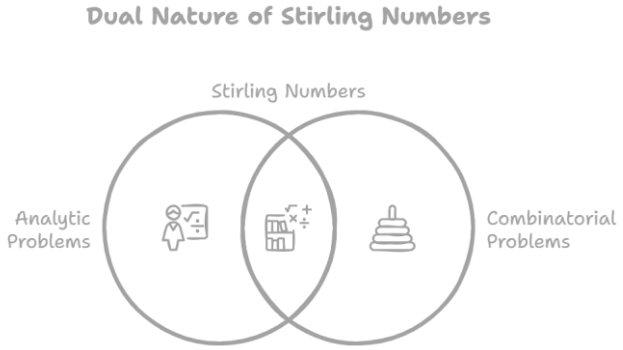
Stars and bars (also called "sticks and stones","balls and bars",and "dots and dividers") is a graphical aid for deriving certain combinatoric theorems. It can be used to solve many simple counting problem , such as how many ways there are to put n indistinguishable balls into k distinguishable bins.



STIRLING NUMBERS :

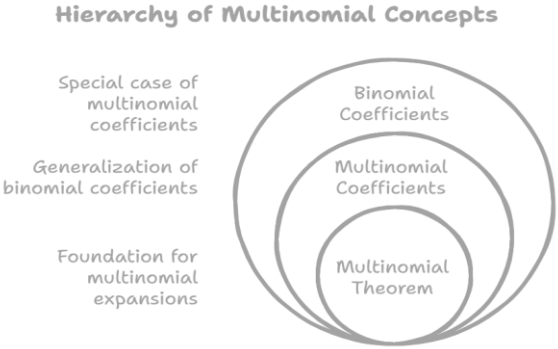
Stirling numbers arise in a variety of analytic and combinatorial problems. They are named after James Stirling, who introduced them in a purely algebraic

setting in his book *Methodus differentialis* (1730). They were rediscovered and given a combinatorial meaning by Masanobu Saka in 1782 .



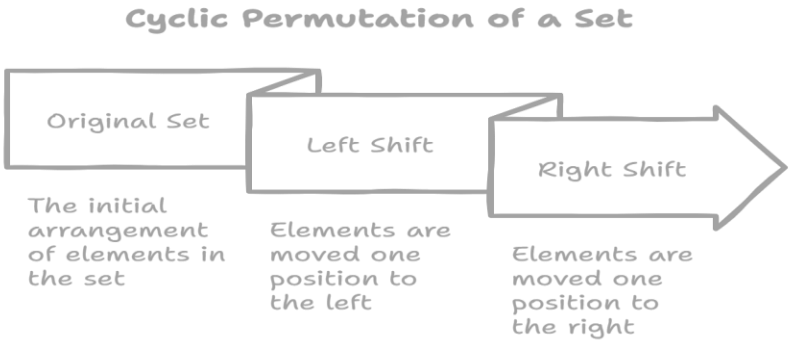
MULTINOMIAL COEFFICIENTS :

Multinomial coefficients are generalizations of binomial coefficients with a similar combinatorial interpretation. They are the coefficients of terms in the expansion of a power of a multinomial, in the multinomial theorem .



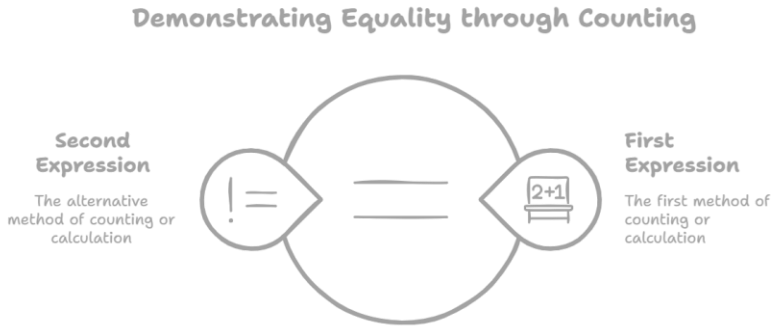
CYCLIC PERMUTATION :

A permutation which shifts all elements of a set by a fixed offset, with the elements shifted off the end inserted back at the beginning.



COUNTING ARGUMENT :

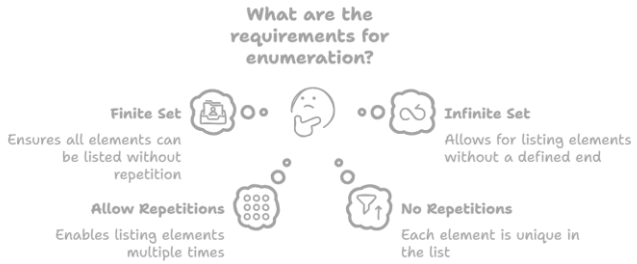
a proof that two expressions are equal by showing that they are two different ways of counting the same thing. It is also known as double counting .



ENUMERATION :

An **enumeration** is a complete, ordered listing of all the items in a collection.

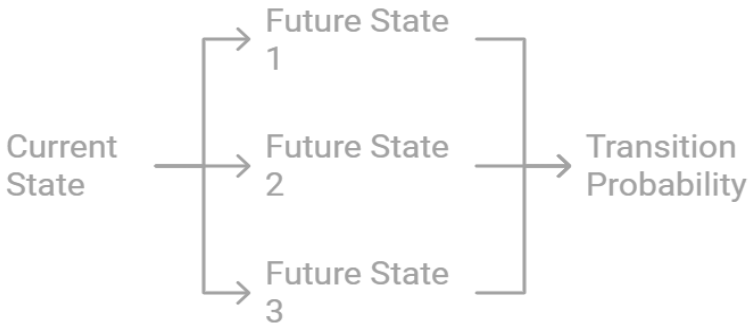
The term is commonly used in mathematics to refer to a listing of all of the elements of a set . The precise requirements for an enumeration (for example, whether the set must be finite , or whether the list is allowed to contain repetitions) depend on the discipline of study and the context of a given problem.



MARCOV CHAINS :

A **Markov chain** is a mathematical system that experiences transitions from one state to another according to certain probabilistic rules. The defining characteristic of a Markov chain is that no matter *how* the process arrived at its present state, the

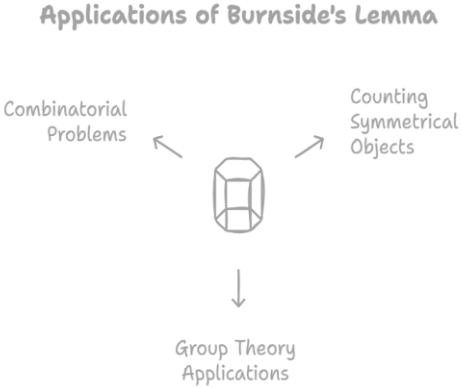
possible future states are fixed. In other words, the
probability of transitioning to any particular state is dependent solely on the current state and time elapsed. The **state space**, or set of all possible states, can be anything: letters, numbers, weather conditions, baseball scores, or
stock performances.



ORBIT COUNTING THEOREM :

Burnside's Lemma is also sometimes known as orbit counting theorem. It is one of the results of group theory . It is used to count distinct objects with

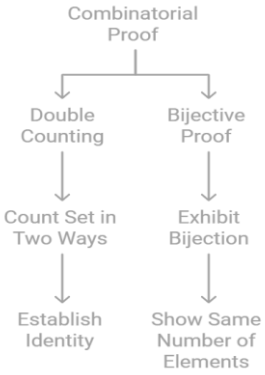
respect to symmetry. It basically gives us the formula to count the total number of combinations, where two objects that are symmetrical to each other with
respect to rotation or reflection are counted as a single representative.



COMBINATORIAL PROOF :

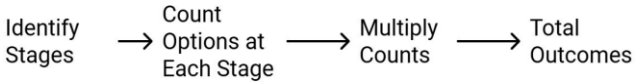
the term **combinatorial proof** is often used to mean either of two types of mathematical proofs : double counting and bijective proof .

"combinatorial proof" may also be used more to refer to any kind of elementary proof in combinatorics.



Counting principle :

The counting principle, also known as the fundamental counting principle, is a basic rule in probability and combinatorics that helps in determining and calculating the total number of possible outcomes efficiently, which is the product between the possible of all outcomes. For instance, if you have 3 shirts and 4 pants, the total outfit combinations are $3 \times 4 = 12$. It applies to independent choices, where the outcome of one does not affect the other.



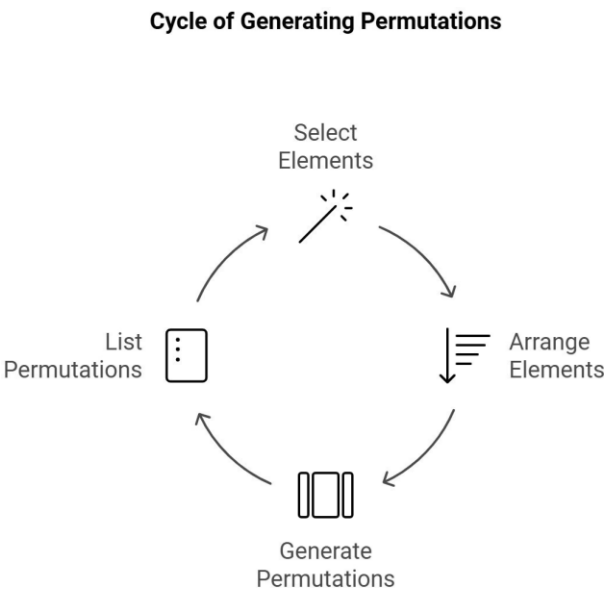
Permutation:

Is an arrangement of a number of objects in a particular order that has applications in several fields, including mathematics, computer science, and combinatorics. Where order matters. Consider S as a set of n elements. An arrangement of k elements of S refers to any ordered selection of k elements from S. For example, arranging three elements of S={A,B,C,D,E} can yield combinations like ABC, ACB, and BDA. Such that the number of the permutation is can be written by the formula:

$$P(n,r) = \frac{n!}{(n-r)!},$$

where:

- $n!$: The factorial of n, calculated as $n \times (n-1) \times (n-2) \times \cdots \times 1$,
- $(n-r)!$: The factorial of $n-r$,
- n: Total number of objects,
- r: Number of objects chosen.



Binomial theorem:

The binomial theorem provides a formula for expanding expressions of the formula $(a+b)^r$ into a sum involving terms of the form "$a^h \times b^{r-h}$ ", where h ranges from 0 to n, which is a peculiar but very useful theorem.

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k,$$

where:

- $\binom{n}{k}$ (read as "n choose k") is the **binomial coefficient**, defined as:

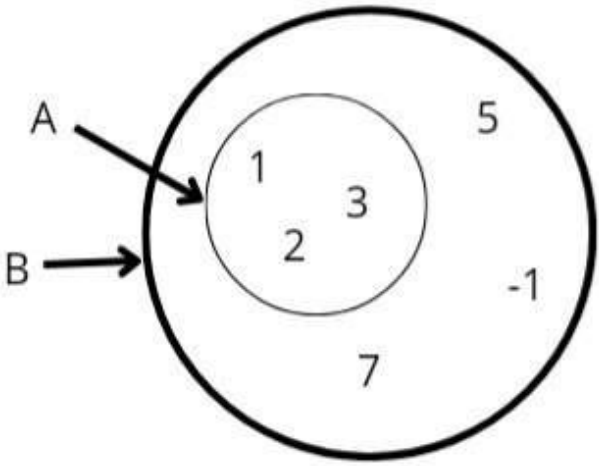
$$\binom{n}{k} = \frac{n!}{k!(n-k)!},$$

- k is the term index, ranging from 0 to n.

Subset :

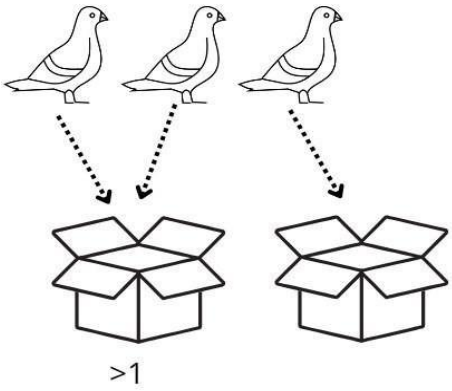
In discrete mathematics, a subset is a set whose elements are all contained within another set.

Which is denoted as A⊆B. It can be used in combinatorics to count and organize groups, logic, or also in set theory. For example, consider A and B two sets such that A={1,2,3} and {-1,1,2,3,5,7}. We say that A⊆B



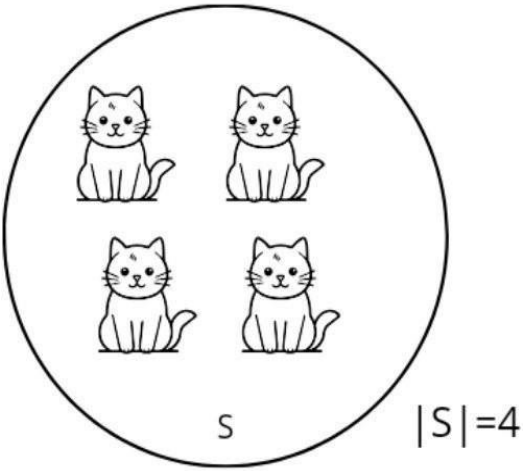
Pigeonhole principle:

Is a simple but at the same time powerful concept in combinatorics and mathematics. It states that if n items are placed in m containers such that n ≥ m, then at least one container must hold more than one item.



Cardinality :

It is a fundamental concept in set theory that a natural number refers to the number of elements in a set such that it can be finite or infinite, with infinite sets categorized. Denoted as $|A|$. For example, in a set B={-4,1,3,4}, the cardinality of B is $|B|=4$



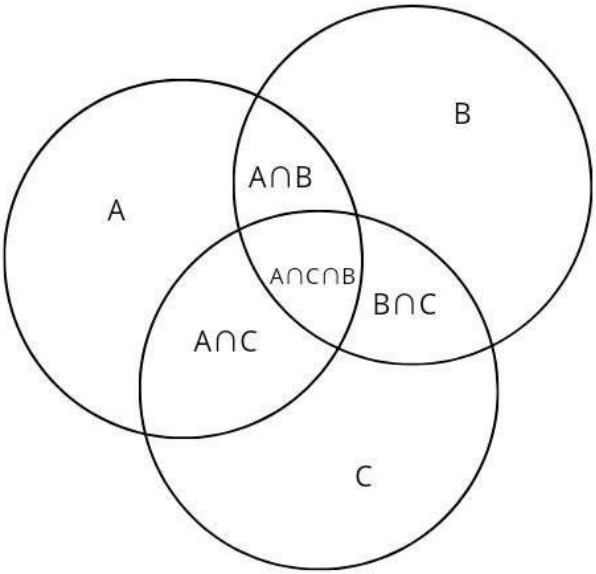
Inclusion-Exclusion principle :

Is a combinatorial tool that helps to calculate the size of the union of multiple sets. Which manages effectively over counting. Which is the sum of the cardinalities of all individual sets minus the sum of the sizes of pairwise

intersections, then adding and subtracting the sizes of higher-order intersections. As an example, if we take A={1,2,3,7} and B={-1,1,2,3,4} then

$A \cup B = \{-1,1,2,3,4,7\}$ and $A \cap B = \{1,2,3\}$ so

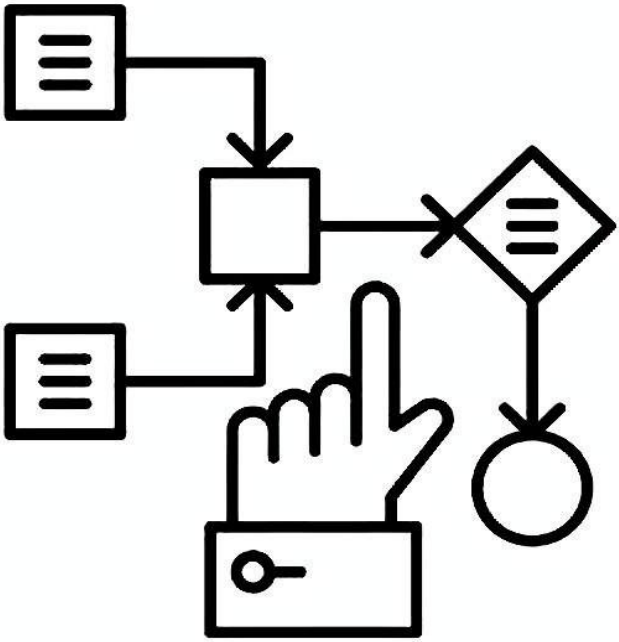
$|A|=4, |B|=5, |A \cap B|=3,$
$|A \cup B|=6, \ |A|+|B|-|A \cap B| = 4+5-3=6=|A \cup B|.$



Algorithm and data structure :

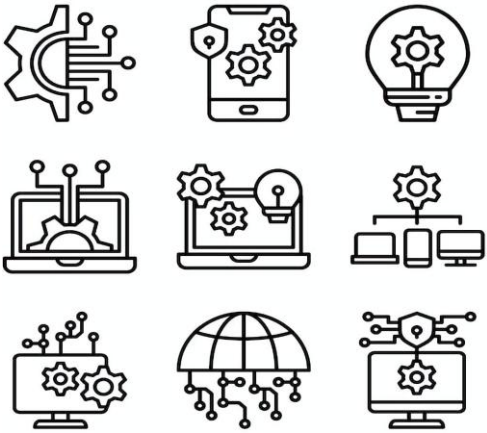
Introduction to the topic :

In the world of technology we're living in today, data structures and algorithms serve as the foundation for nearly all digital applications we rely on. A lot of people don't know it, but Algorithms are a part of our everyday life, from organizing our emails to determining the quickest path on a map and specially when using apps or websites. So, what are they exactly?



Algorithm :

An algorithm is a procedure used for solving problems or performing computations. They act as an exact list of instructions and words that conduct specified actions step by step in either hardware- or software-based routines to get the desired outcome. Algorithm is a necessary thing for computer science since it's the source of learning how information is performed in programming.



Data structure :

Data structure is a specialized format to organize algorithm's information and tools. It gives better analyses, control and access to the data we're dealing with and this solves many complexed problems and difficulties. They are a means of handling information, rendering the data for easy use. it has many and many types

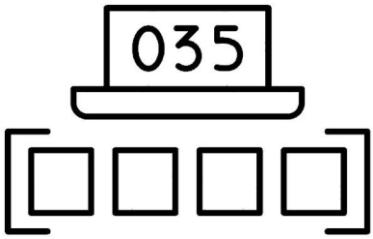
Types of data structure :

They're mainly categorized into two types, with further subcategories in each:

Linear data structure :

In linear data structures, the elements are arranged in a sequential manner, and each element is related to its previous and next element.

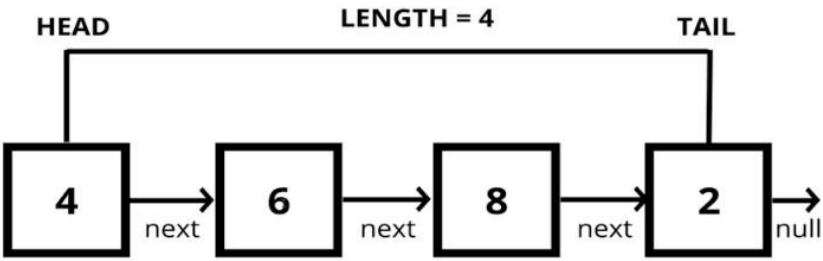
Examples



1.1. Arrays :

Arrays are data structures that hold a collection of elements, usually of the same data type, in contiguous blocks of memory. Each element within an array can be accessed by referring to its index or position, which is one

of the efficient ways of accessing and manipulating data. ex : [1,2,3,4,5]

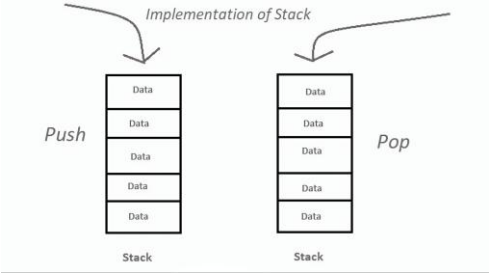


12. Linked list :

Linked lists can best be described as a linear data structure where elements are linked using pointers. Each node contains some data and a pointer— meaning a reference to the succeeding node in the sequence of nodes. Unlike in linked lists, arrays do not require a contiguous memory block and thus provide dynamic memory allocation. And mainly, it has 2 types :

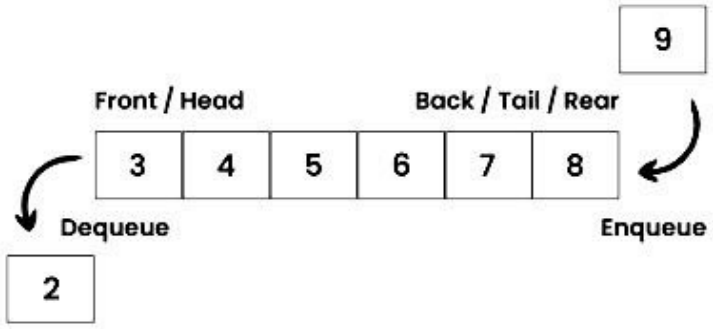
◉ Singly Linked List (one-way connection).

◉ Doubly Linked List (two-way connection).



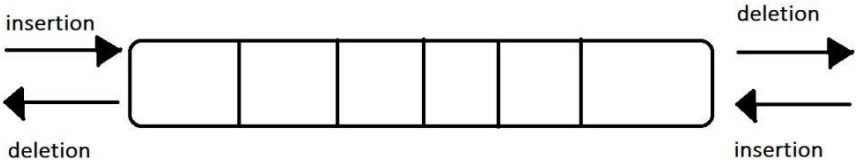
13. Stacks :

A stack is one of the primitive data structures that operates on a policy called Last-In-First-Out or LIFO. Whatever comes last will be the first to get removed. Think of it as a stack of plates; the plate that you put on top last is the first one that comes off.



14. Queue :

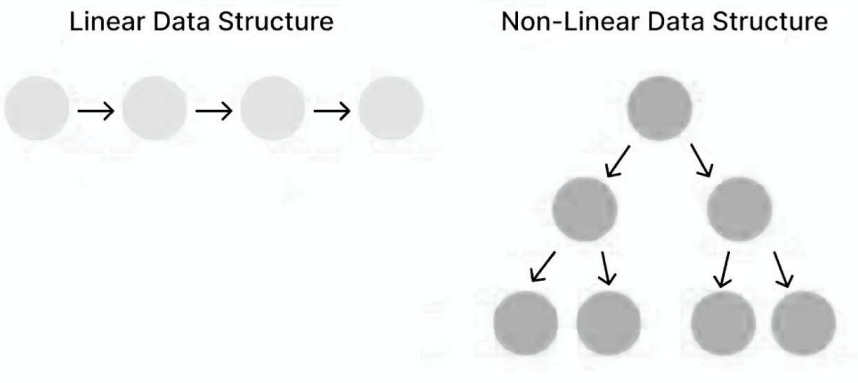
A queue is the reverse of Stack, it's a linear data structure that follows First-In- First-Out (FIFO) principle which means the first element added is the first to be removed. Elements are added at the rear and removed from the front as in a line at a ticket counter. The basic operations include enqueue-an element added to the rear, dequeue-removing the front element, and peek-a look at the front element without removing it. The variants of queues are simple queues, circular queues, which efficiently use memory by wrapping the rear to the front when full, and priority queues where elements are processed based on their priority rather than their order of arrival. Queues find rather extensive applications in task scheduling, BFS traversal, and handling data streams.



15. Deque

A deque, short for double-ended queue, is a

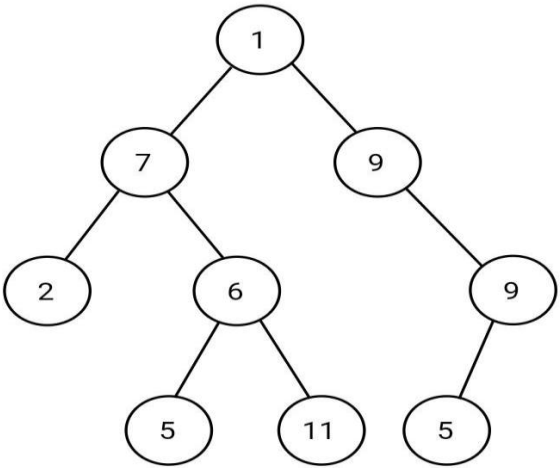
linear data structure that allows both the ends for insertion and deletion, which makes it more versatile than a regular queue. It supports the operations like enqueue front (adding an element at the front), enqueue rear (adding an element at the rear), dequeue front (removing the front element), and dequeue rear (removing the rear element). Variants include input-restricted deques, where insertion is allowed only at one end, and output-restricted deques, where deletion is allowed only at one end. Deques will be implemented for sliding window problems, palindromes, or undo/redo functionality. That provides some flexibility in maintaining data
streams.



Non-Linear data structure

Non-linear data structures are the arrangement of elements in data not in a sequence or linear order. Instead, these are structured hierarchically or as interconnected networks, allowing for more complex relationships between data points. This structure is best suited to represent scenarios like hierarchical relationships, networks, and graphs, where direct or indirect connections between elements are to be modeled .

Examples:



1. Tree:

A tree is a hierarchical structure in which each element, referred to as a node, has a parent -

except for the root node-and zero or more children. Trees find their application in many scenarios, including file systems, databases, and representation of hierarchical data.

Types of Trees:

Binary Tree: It is a tree where each node can have at most two children, which are usually referred to as the left and right child.

Binary Search Tree (BST): A type of binary tree that keeps an order such that for any node, the values in its left subtree are lesser, and the values in the right subtree are greater. This property makes BSTs efficient for searching and sorting.

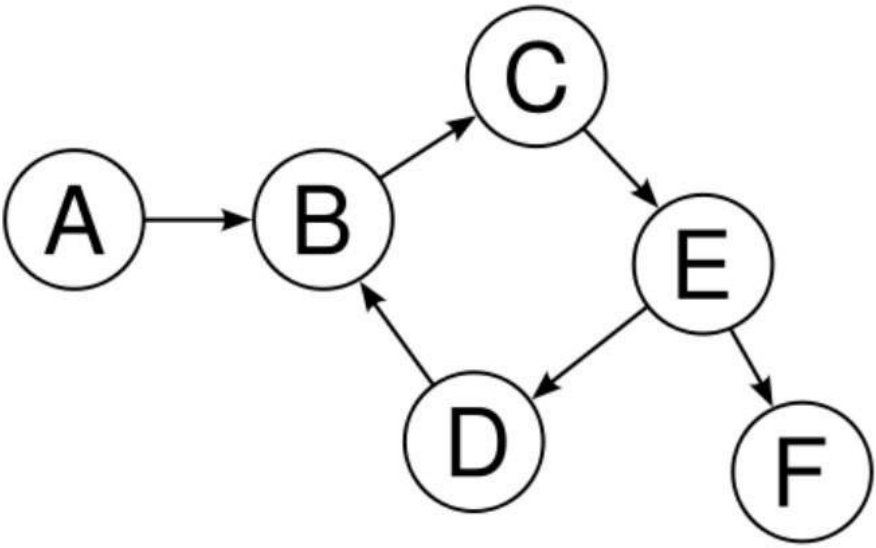
AVL Tree:
In AVL trees, the difference between the height (balance factor) of the left and the right subtrees cannot exceed one for any node. In this way, it promises better performance when inserting or deleting items.

Heap: A complete binary tree that is widely used to implement a priority queue. It comes in two kinds:

Min-Heap: Its root node contains the value of the minimum, with every parent node having lesser or equal value than that of its children.

Max-Heap: The root node has the maximum

value, and every parent node is greater than or equal to its children.



2. Graph:

A graph is made up of a collection of nodes, referred to as vertices, and a collection of connections between these nodes, called edges. Graphs can be used to model a wide range of real-world systems, including social networks, transportation systems, and dependency structures.

Types of Graphs:

Directed Graph (Digraph): In this type, edges have a specified direction, indicating one-way relationships between nodes (e.g., A * B).

Undirected Graph: Here, edges do not have a direction, implying mutual relationships between nodes (e.g., A — B).

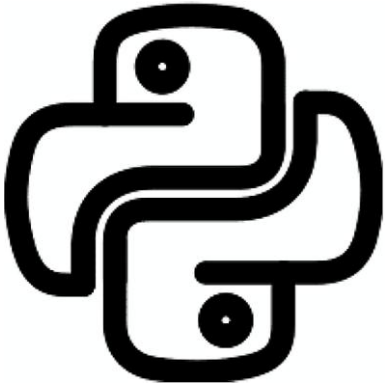
Weighted Graph: In this variation, edges are assigned weights or costs, often representing distances, times, or other quantifiable metrics. Weighted graphs

can be directed or undirected.

Algorithm languages :

Algorithms are central to solving computational problems, with their expression or implementation varying widely depending on the language. Just as a single recipe can be made differently by every chef, depending on their style and manner. This diversity arises from the differing goals, applications, and audiences for these languages. Some aim at simplicity and readability, and therefore are suitable for beginners or rapid development, while others emphasize performance, control, or specific tasks such as database management or

hardware-level programming. Understanding this divergence is fundamental in choosing the proper language in which to express an algorithm because the language greatly impacts efficiency, scalability, and clarity of the solution. This discussion will outline the diversity of algorithm languages and their special roles in shaping how we approach and solve problems in computer science and beyond.



Python :

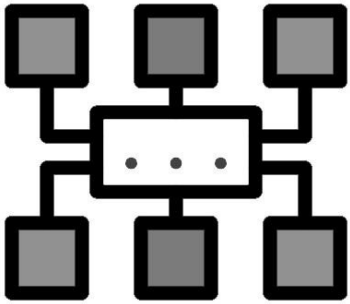
Python is a high-level, interpreted programming language that is simple, readable, and versatile. It was designed to be easy to read and write, with a clean syntax emphasizing readability of code. Python is a great language for

both beginners and professionals. It supports multiple paradigms, such as procedural, object-oriented, and functional programming. It is used in web development, data analysis, artificial intelligence, machine learning, and scientific computing. Python has an extensive standard library and a vast ecosystem of third-party modules that enhance its capabilities .

Java :

Java is a strong, high-level, object-oriented language that's designed to have few implementation dependencies; it's supposed to be "write once, run anywhere." This is realized

in the Java Virtual Machine (JVM), which enables the execution of Java programs on any machine supporting a compatible JVM.
Java is known for being reliable, scalable, and performing well in web applications, enterprise software, mobile applications, especially for Android, and backend systems. Its strong typing, vast libraries, and focus on security make it a favorite among developers around the world.



Peudocode :

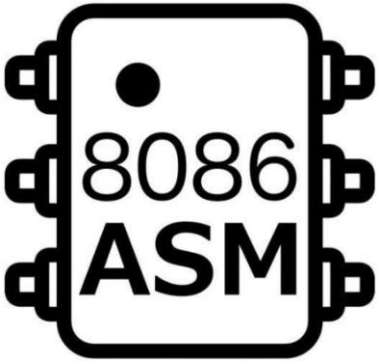
Pseudocode is a simplified, informal way of describing the logic of an algorithm without using the syntax of any programming language. In essence, pseudocode is human-readable and is intended to be as plain as possible, with simple constructs so that attention can be paid to the logic and steps of the solution rather

than the technical details. It's sort of a intermediary stepbetween the problem-solving process and the actual coding, and it is an excellent tool for planning, teaching, and communicating ideas across diverse teams.

SQL :

SQL is a special-purpose language designed for managing, querying, and manipulating relational databases. It allows the user to fetch information, update records, describe database structures, and control access to the database. With its declarative syntax, SQL

enables users to express their queries in a straightforward, human-readable manner. It finds extensive application in business analytics, web development, and data management and forms the backbone of modern data-driven systems.

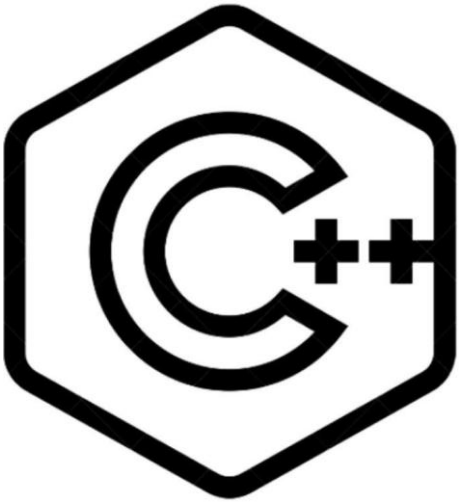


Assembly :

Assembly language is a low-level programming language that represents, using ssymboli codes, the machine code which a computer's hardware directly executes. It utilizes mnemonic codes for operations and symbolic addresses

that refer to memory locations to make writing and debugging easier compared to raw binary instructions. As it is architecture-specific and provides direct control over hardware, assembly is suitable for performance-critical

tasks, embedded systems, and fine-grained optimization.



C++ :

C++ is a high-performance, general-purpose language that combines the efficiency of low-level, hardware-level programming with the flexibility and abstraction of high-level languages. Known for its speed and control over hardware resources, C++ supports multiple paradigms, including procedural, object-oriented, and generic programming, providing developers with powerful tools to optimize performance and manage

complexity in large software projects.

Fundamental Programming Concepts :

Programming concepts are the building blocks in the development of algorithms and writing code to solve problems. These concepts are a kind of universal language that tells how programs should be written and executed, regardless of what programming language is being used. From the manipulation of data through variables and data types to the logic using control structures, these principles ensure that we can systematically analyze problems and design solutions. Whether you work with simple programs or complex systems, mastering these concepts allows for efficient and logical programming. and we'll be explaining them using pseudocode programming language.

1. Variables and Data Types :

Variables are temporary storalge areas that hold data, and data types define the type of data a variable can store, such as numbers, text, or boolean values.

```
DECLARE age AS INTEGER
SET age = 25
```

2. Control Structures :

Control structures provide the ways in which the flow of a program is controlled, including making decisions and repetition.

Conditional Statements: Make decisions based on conditions.

Loops: Repeat actions until a condition is met .

```
IF age > 18 THEN
    PRINT "You are an
adult."
ELSE
    PRINT "You are not an
adult."
ENDIF

FOR i FROM 1 TO 5 DO
    PRINT i
ENDFOR
```

3. Functions :

Functions are reusable blocks of code that perform specific tasks, taking inputs and returning outputs.

```
FUNCTION AddNumbers(a, b)
    RETURN a + b
ENDFUNCTION
```

4. Input and Output (I/O) :

Input gathers data from the user, and output displays results.

```
INPUT name
PRINT "Hello, " + name
```

5. Data Structures :
   Data structures organize and store data efficiently for access and modification. Examples include arrays, stacks, and queues.

```
DECLARE numbers AS ARRAY OF
INTEGER
SET numbers = [1, 2, 3, 4, 5]
```

6. Object-Oriented Programming (OOP) :

OOP models real-world entities using concepts like classes and objects.

```
CLASS Car
    DECLARE brand AS STRING
    FUNCTION
SetBrand(newBrand)
        SET brand = newBrand
    ENDFUNCTION
ENDCLASS
```

7. Error Handling :

Mechanisms to handle unexpected situations gracefully.

```
TRY
    INPUT number
    PRINT 100 / number
CATCH DivisionByZero
    PRINT "Cannot divide by
zero!"
ENDTRY
```

8. Algorithms :

Algorithms are a sequence of steps to accomplish a par .

```
FUNCTION FindMax(numbers)
    SET max = numbers[0]
    FOR each num IN numbers DO
        IF num > max THEN
            SET max = num
        ENDIF
    ENDFOR
    RETURN max
ENDFUNCTION
```

9. File Handling :

Reading and writing data to files for storage and retrieval.

```
OPEN file "data.txt" FOR READ
WHILE NOT EOF(file) DO
    READ line FROM file
    PRINT line
ENDWHILE
CLOSE file
```

10. Recursion :

Recursion is a function calling itself to solve smaller parts of a problem.

```
FUNCTION Factorial(n)
    IF n == 1 THEN
        RETURN 1
    ELSE
        RETURN n * Factorial(n
- 1)
    ENDIF
ENDFUNCTION
```

11. Operators :

Operators are symbols used to perform operations on variables and values, such as comparison, or logical operations like addition and multiplication.

```
DECLARE a AS INTEGER = 10
DECLARE b AS INTEGER = 5
PRINT "Addition: " + (a +
b)    // Arithmetic Operator
PRINT "Is a greater than b? "
+ (a > b)  // Comparison
Operator
```

12. Boolean Logic :

Boolean logic deals with contradicted values, used in conditions and decision- making where you have two choose between two cases.

```
DECLARE isSunny AS BOOLEAN =
TRUE
IF isSunny THEN
    PRINT "It is sunny today."
ELSE
    PRINT "It is not sunny
today."
ENDIF
```

13. Switch Case :

A switch case evaluates a variable and executes a matching case, improving readability. It's like a boolean logic but with more cases and choices .

```
DECLARE day AS INTEGER = 3
SWITCH day
    CASE 1:
        PRINT "Monday"
    CASE 2:
        PRINT "Tuesday"
    CASE 3:
        PRINT "Wednesday"
    DEFAULT:
        PRINT "Invalid day"
ENDSWITCH
```

14. Type Conversion :

Type conversion changes variables from one type to another, for example : changing a variable from string to integer.

```
DECLARE numberString AS STRING
= "123"
DECLARE number AS INTEGER =
CONVERT_TO_INTEGER(numberStrin
g)
PRINT "The integer value is: "
+ number
```

15. MOdular programming :

It is a method of dividing a program into small, independent units called “Modules”. Each unit implements a specific part of the required functionality and can be easily used and reused in different parts of the program. This method helps in improving the organization of the code, making it easier to maintain, and reducing errors when modifying or adding.

```
MODULE CalculateSquare
    FUNCTION Square(number)
        RETURN number * number
    ENDFUNCTION
ENDMODULE

CALL Square(4)  // Call the
function from the module
```

16. Print :

This is a basic command or function in programming that allows programmers to display messages or any information to the user. It is essential for debugging, interacting with the user, or providing feedback during program execution. The print statement or function is implemented differently across programming languages but serves the same purpose universally.

```
DECLARE name AS STRING
SET name = "Alice"
PRINT "Hello, " + name
```