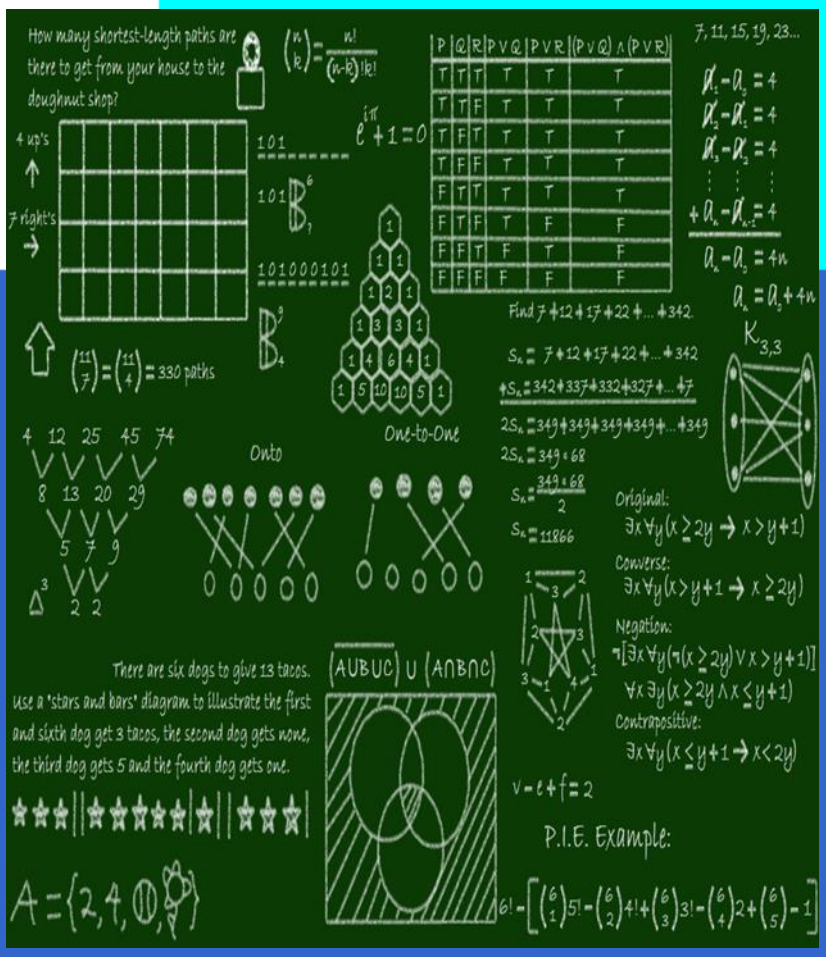
DISCRETE MATHEMATICS

LEARNING JOURNA

INSTITUTE OF INFORMATION SYSTEMS AND TECHNOLOGY



SUBMITTED TO:
Mr Wilmer M. Pascual

SUBMITTED BY:
Name: MA. Khate beryl S .Saguid
Course: BSIT
Year & Section: 2 F

LEARNING JOURNAL RUBRICS

(to be filled by the teacher)

Category	Exceeds Expectations		Meets Expectations		Approaching Expectations		Expectations Not Meet		% Factor	Earned Points
Ideas	Topic is clear and well written	10	Topic is clear	7	Topic is somewhat clear	4	Topic is not clear	1	50	
Organization	Topic sentence and detail sentence are complex and show variety	10	Topic sentences and details sentences may be simple but show variety	7	Topic sentence may or may not be present	4	No Topic sentence	1	30	
Literacy and Reader Response	Reflects a strong understanding of the text	10	Reflects a developing understanding of the text	7	Reflects a limited but growing understanding of the text	4	Does not reflect or fails to develop an understanding of the text	1	20	
									TOTAL POINTS EARNED	/ 10

MODULE 1:
Introduction to Discrete Mathematics

Introduction

Discrete Mathematics is the part of mathematics devoted to the study of discrete . Mathematics is a study of numbers but you also study function and lines and triangles and parallelepiped s and vectors and .. Or perhaps you want to say that mathematics is a collection of tools that allow you to solve problems. And have numbers that involve like functions,lines, triangles and etc. Math fundamentally deals with stuff that is individually separate and distinct. Is a branch of mathematics dealing with objects that can assume only distinct separated values. A mathematical language for computer science and as such it's important has increased dramatically in recent dicades.

SETS

CHARACTERISTICS OF SETS

CHARACTERISTICS OF SETS

a. Traditionally named using capital letters.
b. Unordered elements and can be distinct or not distinct.
c. Can be a finite, infinite, empty or universal

Kinds of Sets:
a. Finite Set determines elements of a set particularly.
Ex. The set of all positive numbers less than 10.
b. Infinite Set determines elements of a set endlessly.
Ex. The set of all numbers less than 10.
c. Empty Set no elements within a set. Symbol “Ø”.
d. Universal Set Its elements were belong to its subsets and the compliments of that subsets are also its elements. Sometimes called the universe.

Set Notation/ Set builder

{y | y is an integer, 0< y <5}

Sets can also use symbols showing memberships. Memberships symbols used in sets to denote its occurrence.

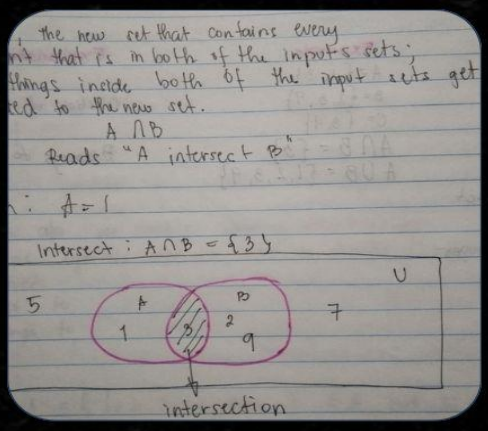
∈- to indicate that an element belongs to a set . Reads “belongs to” or “is an element of” or “is a member of”.

∉ - to indicate that an element does not belong to a set. Reads “does not

Venn Diagram

A Venn diagram, also called logic diagram, is a diagram that can be used to expresst he logical relationships between various finite sets. It can be used to determinew hether a statement about sets is true or false.

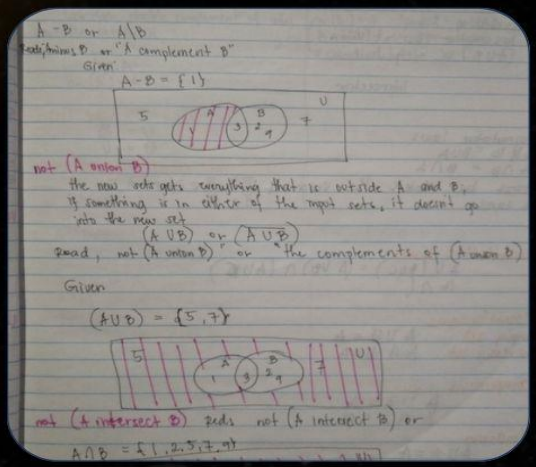
Example 1:
For Instance A∪B:
Reads, “A union B”.
Means, the new set that contains every element from either of set A and set B; if a thing is in either one of these sets, its in the new set.

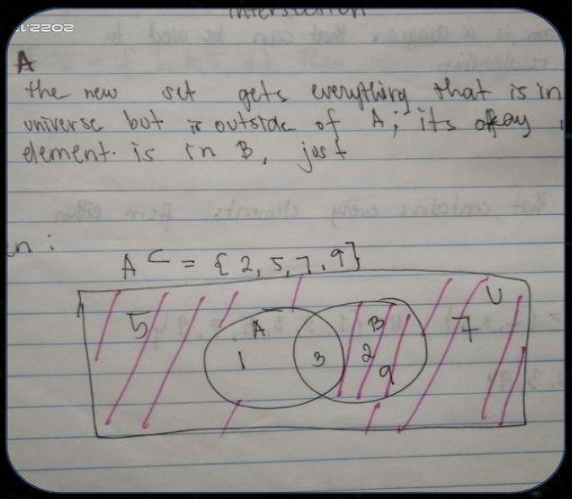


For instance A ∩ B:
Reads, “ A intersect B”
Means, the new set that contains every element that is in both of the inputs sets; only things inside both of the input sets get added to the new set.

Example 3:
For instance AC or ¬A or Ā :
Reads, “Complement A” or “Not A”.
Means, the new set gets everything that is in the universe but is outside of A; it’s okay if the element is in B, just so long as it is not also in A.

Example 4:
For instance A – B or A \ B:
Reads, “A minus B” or “A complement B”
Means, the new set gets everything that is in A except for anything in its overlap with B; if it’s in A and not in B, then it goes into the new set; nothing from the overlap in the diagram (being in the intersection of the input sets) goes into the new set.





Functions

Types of numbers

N= the natural numbers
Z = the integers
Q = the rationals
R = the real numbers

Definition of function

If we travel for a certain amount of time at a constant rate, we know that

Distance = rate x time

Surjective, Injective, Bijective Function

When the sort of the things DOES NOT happen (when everything in the codomain is in the Range), the function is onto (or the function maps the domain onto the codomain).
Thus, this is a sample of surjection. Notice both properties of Injective and Surjective are determined by what happens to elements of codomain: they could be repeated as images or they could be missed. So that, Surjective functions do not missed elements , but might or might not havee repeats. Injective functions do not have repeats but might or might not missed elements.The Bijective functions are those that do not repeats and not missed elements.

Composition of Functions

Two functions f : X →Y and g: Y →Z can be composed to give a composition g ◦ f. This is a
function from X to Z defined by (g ◦ f)(x) = g(f(x))

Relations

A relation as we define is to be the set order pairs.

Directed Graphs

A directed graph, or digraph,consists of a set of vertices V (dots or circle) and a set of (arrows).
A simple directed graph has no parallel edges.
A bipatite directed graph is where all edges go form a vertices in A to vertices in B.
A directed graph that contains no looping sequence of edges (no cycle of edges) leading back to it's starting point is called a directed acyclic graph.

Properties of Relation

There are 6 Properties of Relation there are:
1. Reflexive- A relation R on a set A is said to be reflexive relation if every element of A is related to itself. Thus, R is reflexive if and only if (x,x) □ R for all x □ A. A relation R on a set A is not reflexive if there is an element x □A such that (x,x) □ R.
Example:
A= {1,2,3,4}
2. Symmetric A relation R on a set is said to be symmetric relation if the elements of A is related in back to back fashion. Thus, R is symmetric if (x,y) □ R then (y,x) □ R. So in relation R, R-1 =R.
Example:
A= {a,b,c,d} defined by, R= {(a,a),(b,c),(c,d),(d,d)}
3. Antisymmetric - A relation R on a set A is said to be antisymmetric relation such that the elements of A is related to each other if (x,y) □ R, then x= y, Thus, the relation between vertices is only at one edge.
Example:
A= {1,2,3} , defined by, R= {(1,1), (1,2), (3,2),(3,3)}
4. Transitive - A relation R on a set A is said to be transitive relation such that the elements of A is related to each other if (x,y) □ R, then (x,z) □ R for all x,y,z □ A.
Example:
A={1,2,3}, defined by, R = {(1,2),(1,3), (1,4),(2,3),(2,4),(3,4),(4,3)}
5. Inverse - Let R be a relation from X to Y. The inverse of R, denoted R-1, is the relation from Y to X defined by R-1 = {(y,x)|(x,y) □ R}.
Example:
X={2,3,4} to Y={3,4,5,6,7}
6. Cimposition - Let R1 be a relation from X to Y and R2 be a relation from Y to Z. The composition of R1 and R2 , denoted by R1 ◦ R2 , is the relation from X to Z defined by R1 ◦ R2 = {(x,y) |(x,y) □ R1 and (y,z) □ R2 for some y □ Y }.
The composition of relation from X to Y and R1 = {(1,2),(1,6),(2,4),(3,4),(3,6),(3,8)} and R2 = {(2,u),(4,s), (4,t), (6,t), (8,u)}.

MODULE 2: Mathematical Reasoning

LOGIC

Logic is a formal study of mathematics.
The study of reasoning and proofs itself.
Logic is the basis of all mathematical reasoning, and of all automated reasoning.

Proposition

A proposition statement or simply proposition is any statement that is answerable or can be interpreted by Truth or false.

Connectives

Connectives are functions of a truth value to another truth value. Each defined by their corresponding Truth Table.

A proposition and a logic has only two values, that's True (1) or False (0). It can be noted using small letters from the alphabet.

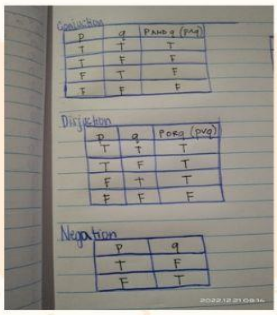
Example
p = The Information Technology Department is under IIST
q = There are available e-books online for this subject

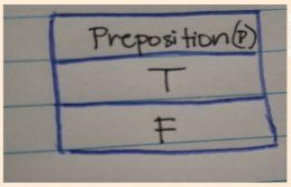
Logical Connectives

1. Conjunction
Logical Connective AND or a conjunction is represented by . AND outputs are True if and only if (iff) pboth or all of the inputs (propositions) are true.
2. Disjunction
Logical Connective OR or a disjunction is represented by ∨. OR outputs are True if at least one of the proposition is True.
3. Negation
Logical condition NOR or the negation is denoted by ¬p read as Not p.
It inverts the Truth value of a certain preposition.

Truth Table

Truth tables can be used to represent much more than just the actions of primitive logical operators. Truth tables can represent the arbitrary input/output behaviour of Boolean (logical) formula or circuits.





MODULE 3

Counting Principles

Basic Counting Principles

The counting principle, also called the counting rule, is a way to figure out the number of outcomes in a probability problem. Such that it enables us to count the following, without having to list all of the items:

The number of ways
The number of samples
The number of outcomes

Before learning some of the basic principles of counting, let see some of the notation well needs.

Number of Outcomes in an Event

As an example, we may have an event E defined as

E=”days of the week”

So, write the number of outcomes of an event E as n(E). So, in this example, we will get, n(E) = 7, since there are 7 days in a week.

There are two rules in Counting Principle: a. Addition Rule

We are getting sum. Let E1 and E2 be mutually exclusive events (or the events has no common outcomes). Let event E describe the situation where either event E1 or event E2 will occur.

b. Multiplication Rule

Now consider the case when two events E1 and E2 are to be performed and the events E1 and E2 are independent events (one does not affect the other’s outcome).

Example1:

We have 2 t-shirts and with each t-shirt we could have 4 pairs of jeans.

Altogether there are,
2 x 4 = 8 possible combinations,
We could write,
E1 = “choose t-shirt”
E2 = “choose jeans”
Suppose that the evet E1 can result in any one of n(E1) possible outcomes; and
for each outcome of the event E1, there are n(E2) possible outcomes of event
E2

That is, if event E is the event that both E1 and E2 must occur, then n(E) = n(E1) x n(E2)

Permutations and Combination

Permutation (Ordered Arrangements)

An arrangement (or ordering) of a set of objects is called a permutation.

In a permutation, the order that we arrange the objects in is important.

There are 4 theories of Permutation, namely, a.

Arranging n Objects

In general, n distinct objects can be arranged in n! ways.

Like the example 1.

b. Number of Permutations

The number of permutations of n distinct objects taken r at a time, denoted by Pr n , where repetitions are not allowed, is given by

PR

$$n = n(n-1)(n-2) \ldots (n-r=1) = n!$$

$$(n-r)!$$

c. Permutations of Different Kind of Operations

The number of different permuations of n objects of which n1 are of one kind, n2 are of a second kind,.. nk are of a k-th kind is n! n1! x n2! x n3! x ... x nk!

d. Arranging Objects in a Circle

There are (n -1)! Ways to arrange n distinct objects in a circle (where the clockwise and anti-clockwise arrangements are regarded as distinct.) Example5:

In how many ways can 5 people be arranged in a circle?

Solution:

(5-1)! =4! =24 ways

Combination (Unordered Selections)

A combination of n objects taken r at a time is a selection which does not take into account the arrangement of the objects. That is, the order is not important.

Example:

Consider the selection of a set of 4 different letters from the English alphabet.

Probability

The statistician is basically concerned with the drawing conclusions (or inference) from experiments involving uncertainties. For these conclusions and inferences to be reasonably accurate, an understanding of probability theory is essential.

Experiment is any process of observation or procedure that; 1. Can be repeated (theoretically) an infinite number of times;

and

2. Has a well -defined set of possible outcomes.

Sample Space is the set of all possible outcomes of an experiment.

Event is a subset of the sample space of an experiment.

Definition of Probability

When an experiment is performed, we set up a sample space of all possible outcomes. In a sample of N equally likely outcomes we assign a chance (or weight) of 1 N to each outcome.

$$P(E) = \frac{n(E)}{n(S)}$$

Where,

n(E) is the number of outcomes favourable to E, and n(S) is the total number of equally likely outcomes in the sample space S of the experiment.

MODULE 4 ; Number Theories

DIBISIBILITY

If a and b are integers with a ≠ 0, then a divides b, if there exists an integer q
such that b = aq. When a divides b we write a| b. We say that a is a factor or divisor of b and b is a multiple of a.

Example: b ÷ a = q

Where,

b is the dividend, a is the divisor; and q is the quotient 32 ÷ 8 = 4

DIVISIBILITY TEST

Divisibility by 2 – The units' digit must be even.

Divisibility by 4 – The number formed by its last two digits must be divisible by 4.

Divisibility by 5 – The units' digit must be 0 or 5.

Divisibility by 6 – It must be even and divisible by 3.

Divisibility by 7 – When the units' digit is doubled and subtracted from the number formed by the remaining digits, the resulting number must be divisible by 7.

Divisibility by 8 – The number formed by its last three digits must be divisible by 8.

Divisibility by 10 – Its last digit must be 0.

MODULE 4 ; Number Theories

Division Algorithm When an integer is divided by a positive integer, there is a quotient and a remainder. This is traditionally called the Division Algorithm.

Ex.

If a = 9 and b = 2, then q = 4 and r = 1.

- **Modulo Operations the modulo operation returns the remainder or signed remainder of a division, after one number is divided by another called the modulusof the operation.**

Ex.

“5 mod 2" would evaluate to 1, because 5 divided by 2 has a quotientof 2 and a remainder of 1"

Calculation of Modulo

1. Start by choosing the initial number
2. Choose the divisor
3. Divide one number by the other, rounding down
4. Multiply the divisor by the quotient.
5. Subtract this number from your initial number (dividend)
6. The number you obtain is the result of the modulooperation

MODULE 4 ; Number Theories

Prime Numbers, Prime Factorization, GCD and LCM

Prime Numbers

A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. Prime numbers are an important concept in number theory and have many interesting properties. The first few prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, and 31.

Prime numbers are important because they are the building blocks of all other numbers. Every integer greater than 1 can be written as a product of prime numbers in a unique way, known as the prime factorization of the number. For example, the prime factorization of 12 is 2 x 2 x 3, and the prime factorization of 15 is 3 x 5.

MODULE 4 ; Number Theories

Prime Factor

A prime factor of a positive integer is a prime number that is a factor of that integer. For example, the positive integer 12 has the prime factorization 2 x 2 x 3, so the prime factors of 12 are 2, 2, and 3.

The prime factorization of a positive integer is the unique way to express that integer as a product of prime numbers. Every positive integer can be expressed as a product of prime numbers in this way, and the prime factors of a positive integer are the prime numbers that appear in this factorization.

Detailed Example of Getting prime factor: Suppose we want to find the prime factors of the positive integer 15. To do this, we can use prime factorization as follows:

MODULE 4 ; Number Theories

Finding GCD

One method for finding the GCD of two numbers is to use prime factorization. This method involves expressing each number as a product of prime numbers and then identifying the common prime factors. The GCD is the product of the common prime factors.

For example, suppose we want to find the GCD of the numbers 24 and 16. We can use prime factorization as follows:

Find the prime factorization of 24, which is 2 x 2 x 2 x 3.
Find the prime factorization of 16, which is 2 x 2 x 2 x 2.
Identify the common prime factors, which are 2 and 2.
The GCD is the product of the common prime factors, which is 2 x 2 = 4

Therefore, the GCD of 24 and 16 is 4. This method can be useful because it provides a way to find the GCD of two numbers without having to perform any divisions. It is also a good way to learn about the fundamental theorem of arithmetic, which states that every positive integer can be expressed as a product of prime numbers in a unique way.

MODULE 4 ; Number Theories

Finding LCM

The least common multiple (LCM) of two numbers is the smallest positive integer that is a multiple of both numbers. It can be found using prime factorization, which involves expressing each number as a product of prime numbers and then multiplying the highest exponent of each prime number to obtain the LCM.

For example, suppose we want to find the LCM of the numbers 12 and 15. We can use prime factorization as follows:
Find the prime factorization of 12, which is 2 x 2 x 3.
Find the prime factorization of 15, which is 3 x 5. Multiply the highest exponent of each prime number to obtain the LCM. In this case, the highest exponent of 2 is 2 and the highest exponent of 3 is 1, so, the LCM is 2 x 2 x 3 = 12.

Therefore, the LCM of 12 and 15 is 12. This method can be useful because it provides a way to find the LCM of two numbers without having to perform any divisions or multiplications. It is also a good way to learn about the fundamental theorem of arithmetic, which states that every positive integer can be expressed as a product of prime numbers in a unique way.

MODULE 4 ; Number Theories

Cryptology is the study of methods for secure communication and the design of algorithms and protocols for encrypting and decrypting information. It is a interdisciplinary field that combines aspects of computer science, mathematics, and electrical engineering. Cryptology has a long history, dating back to ancient civilizations that used simple codes and ciphers to protect their communications. Over the centuries, cryptologists have developed increasingly sophisticated techniques for encrypting and decrypting information, and today cryptology plays a vital role in protecting sensitive data and communications in many different applications.

Cryptology has two main branches: cryptography, which deals with the design and analysis of algorithms and protocols for encrypting and decrypting information, and cryptanalysis, which involves the study of techniques for breaking codes and ciphers. Cryptologists use a wide range of mathematical and computational techniques to solve problems in these areas, and they often work closely with computer scientists and engineers to develop practical solutions to real-world problems.

MODULE 5: ALGORITHM

An algorithm is a set of steps or procedure for solving a specific problem or achieving a desired result. It is a systematic method for solving a problem in a finite number of steps.

Properties of Algorithm

Input: An algorithm needs input to solve a problem.
Output: An algorithm produces output as a solution to a problem.
Precision: Precision refers to how accurately an algorithm produces the expected result. A precise algorithm produces consistent, accurate results
Finiteness: An algorithm must be able to finish within a certain time and with limited resources. Correctness: An algorithm must be able to solve a problem correctly. Generality: An algorithm should be able to solve many different problems. Determinism: An algorithm must always produce the same result for a given input.

MODULE 5: ALGORITHM

Searching Algorithm

Searching algorithms are used to find a specific element or group of elements in a data structure. There are different types of searching algorithms, including linear search and binary search. Linear search looks at each element insequence, while binary search divides the data structure in half at each step.

Linear Search

Example:

Input: a sequence s = [4, 6, 2, 9, 1], and the value x = 2
Output: the location of 2 in the sequence, which is 2
To find the location of 2 in the sequence using linear search, we can use the following steps:

1. Set i to 0
2. While i is less than 5:

a. If the value at position i in the sequence is 2, return i
b. Increment i by 1
3. Return -1

At the first step, i is 0, and the value at position 0 in the sequence is 4, which is not 2. So, i is incremented to 1. At the second step, i is 1, and the value at position 1 in the sequence is 6, which is not 2. So, i is incremented to 2. At the third step, i is 2, and the value at position 2 in the sequence is 2, which is the value we are searching for. So, we return 2 as the output.
Therefore, the location of 2 in the sequence is 2.

MODULE 5: ALGORITHM

Analysis of Algorithm

The analysis of an algorithm is the study of its performance and characteristics. This helps us understand the strengths an limitations of the algorithm and identify potential improvement There are different ways to analyze an algorithm including time complexity, space complexity, and experimental analysis. By analyzing an algorithm, we can ensure that it is correct, efficien and effective.

Three types of Case Time

Best-case is the scenario in which an algorithm performs at its best. This means that the input is most favorable for the algorithm, and it is able to find the solution to the problem quickly and efficiently. For example, if an algorithm is designed to search for a specific value in a sorted array, the best-case scenario would be when the value is located at the beginning of the array, so the algorithm can find it in the first step.

Worst-case is the scenario in which an algorithm performs at its worst. This means that the input is most unfavorable for the algorithm, and it takes the most number of steps or the longest amount of time to find the solution to the problem. For example, if an algorithm is designed to search for a specific value in an array, the worst-case scenario would be when the value is not in the array, so the algorithm has to search through the entire array without finding it. This would take the maximum number of steps or the longest amount of time.

Average-case is the scenario in which an algorithm performs on average. This means that the input is randomly generated, so the algorithm takes an average number of steps or an average amount of time to find the solution to the problem. For example, if an algorithm is designed to search for a specific value in an array, the average-case scenario would be when the value is located somewhere in the middle of the array, so the algorithm has to search through half of the array to find it. This would take an average number of steps or an average amount

GRAPH THEORIES

Graph theories

branch of mathematics that studies the properties of graphs, which are structures that consist of a set of vertices (also called nodes) connected by edges. Graphs can be used to represent a wide range of real-world situations, such as social networks, transportation networks, and computer networks.

Types of graphs

Null Graph

graph that contains no edges.

Empty Graph

graph that contains no vertices.

Directed Graph

also called a digraph, is a graph in which the edges have a direction associated with them.

Non -Directed Graph

also called an undirected graph, is a graph in which the edges do not have a direction associated with them.

Complete Graph

is a graph in which every pair of distinct vertices is connected by an edge. In other words, in a complete graph, every vertex is connected to every other vertex by an edge.

Cycle Graph

type of graph that has a single cycle, which is a path that starts and ends at the same vertex without passing through any other vertex more than once. A cycle graph is made up of a set of vertices and a single cycle that connects these vertices.

Acyclic Graph

graph that contains no cycles, which are paths that start and end at the same vertex without passing through any other vertex more than once. **Finite Graph**

type of graph that has a finite number of vertices and edges. **Infinite Graph**

type of graph that has an infinite number of vertices and edges. In other words, it is a graph that has an unbounded and potentially limitless number of objects and connections between those objects. **Bipartite Graph**

type of graph in which the vertices can be divided into two sets, such that all edges in the graph connect a vertex in one set to a vertex in the other set. **Complete Bipartite Graph**

a type of bipartite graph in which every vertex in one set is connected to every vertex in the other set. **Planar Graph**

type of graph that can be drawn on a plane without any of the edges crossing each other. **Multi Graph**

graph that allows multiple edges to connect the same pair of vertices. **Euler Graph**

type of graph that has a special property called Eulerianness.

TREES

A tree is a type of graph that consists of a set of vertices connected by edges. In a tree, there is a special node called the root, which is the starting point for the tree, and there are no cycles.

A tree has a hierarchical structure, with the root at the top and the other nodes organized into levels below it. The nodes at the bottom level are called leaf nodes, and the nodes in the levels above them are called parent nodes. A node can have one or more children, which are the nodes below it in the tree.

A tree has a unique path from the root to every other node.
The depth of a node is the number of edges on the path from the root to that node.
The height of a tree is the maximum depth of any node in the tree.
The degree of a node is the number of children it has.
A binary tree is a tree in which each node has at most two children.

TYPES OF TREES

Binary tree
tree data structure in which each node has at most two children. The two children of a binary tree node are referred to as the left child and the right child.

Rooted tree
tree data structure in which one node, called the root, is designated as the topmost node in the tree.
Properties of Rooted tree

Parent
the parent of a vertex is the vertex that is directly connected to it by an incoming edge.

Child
child of a vertex is a vertex that is directly connected to it by an outgoing edge.

Siblings
siblings are vertices that have the same parent.

Ancestors
ancestors of a vertex are the vertices that can be reached by following a path of incoming edges from the vertex to the root of the tree.

Descendants
descendants of a vertex are the vertices that can be reached by following a path of outgoing edges from the vertex to the leaves of the tree.

Properties of Rooted Trees

Parent

In a rooted tree, the parent of a vertex is the vertex that is directly connected to it by an incoming edge. In other words, the parent of a vertex is the vertex that is one step "above" it in the tree.

Child

In a rooted tree, a child of a vertex is a vertex that is directly connected to it by an outgoing edge. In other words, a child of a vertex is a vertex that is one step "below" it in the tree.

BOOLEAN ALGEBRA

is the algebra of variables that can assume two values:
True and False. Conventionally we associate these as follows:

True = 1 and False = 0.
This association will become important
when we consider the use of Boolean
components to synthesize arithmetic circuits,
such as a binary adder.

AND GATE
Boolean algebra is a branch of mathematics that deals with operations on logical values with binary variables.

The output of an AND gate attains state 1 if and only if all the inputs are in state 1.

OR GATE
The output of an OR gate attains state 1 if one or more inputs attain state 1.

NOT GATE
The output of a NOT gate attains state 1 if and only if the input does not attain state 1.

NAND GATE
This basic logic gate is the combination of AND and NOT gates.

NOR GATE
This gate is the combination of OR and NOT gate.

XNOR GATE
The output of a two-input XOR gate attains state 1 if one adds only input attains state 1.

XOR GATE
The output is in state 1 when both inputs are the same, that is, both 0 or both 1.