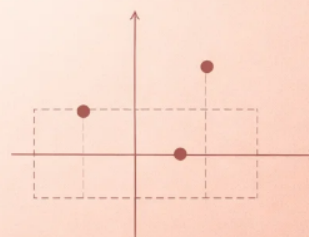


Mathematics – II

(255MT-198)



$$\lim_{x \rightarrow 2} f(x)$$



Unit 3 - Resources & Questions

Compiled by : Subhayu



Syllabus Covered So Far

Unit 1

Chapter 1 : Set Theory

Introduction to set theory, Types and classes of sets, Finite and infinite sets, Set operations (union, intersection, difference, complement), Algebra of sets, Principle of duality, Venn diagrams and their applications, Power sets, Minsets and Maxsets, Cartesian product of sets, Principle of inclusion and exclusion

Chapter 2 : Relations and Functions

Introduction to relations, Binary relations, Types of relations, Equivalence relations and partitions, Partial order relations, Introduction to functions, Types of functions, Composition of functions, Inverse functions, Mathematical induction

Chapter 3 : Propositional Calculus

Introduction to propositions, Logical connectives: AND ($\wedge$), OR ($\vee$), NOT ($\neg$), Compound statements, Truth values of compound propositions, Truth tables, Tautologies, Contradictions

For Unit-1 Notes: [Click here](#)

Unit 2

Chapter 4 : Recursion and recurrence relations

Recurrence relations with constant coefficients, homogeneous and particular solutions, solutions of recurrence relation using generating functions.

Chapter 5 : Algebraic structures

Introduction to algebraic structures, semi group, monoid, group, congruence relations, subgroups, cosets, normal subgroups, cyclic groups, homomorphism, isomorphism and automorphism.

For Unit-2 Notes: [Click here](#)



Syllabus for Unit 3

Mathematics-II (25SMT-198)

Chapter 1 : Combinatorial Mathematics – Techniques of counting, rule of sum, rule of product Permutations and combinations, pigeonhole principle.

Chapter 2 : Graph Theory – Introduction to graphs, directed and undirected graphs, sub-graphs, multigraphs, weighted graphs, homomorphic and isomorphic graphs, paths and circuits, shortest path in weighted graphs by dijkstra algorithm, Eulerian paths and circuits, Hamiltonian paths and circuits, planar graphs, Euler's formula, chromatic numbers(Coloring Problem)

Chapter 3 : Trees – Introduction to trees, difference between graphs and a tree rooted trees, path length in trees, spanning trees, cut-sets, minimum spanning trees by kruskal's algorithm, binary trees and its traversal.

Chapter 1

Combinatorial Mathematics

1.1 Introduction to Combinatorics

Combinatorics is a branch of mathematics that deals with counting, arrangement, and selection of objects. It provides systematic techniques to answer questions like:

- How many ways can something be done?
- In how many ways can objects be arranged?
- How many selections are possible under given constraints?

1.1.1 Why is Combinatorics Important?

Combinatorics plays a crucial role in:

- Probability theory
- Computer science (algorithms, data structures)
- Cryptography
- Artificial Intelligence and Machine Learning

1.1.2 Basic Terminology

- **Arrangement:** Ordering of objects.
- **Selection:** Choosing objects without considering order.
- **Permutation:** Arrangement where order matters.
- **Combination:** Selection where order does not matter.



1.2 Techniques of Counting

Counting is the foundation of combinatorics. Instead of listing all possibilities, we use rules to compute them efficiently.

1.2.1 Rule of Sum (Addition Principle)

If a task can be performed in m ways and another independent task in n ways, and both cannot occur together, then total number of ways:

$$m + n$$

Key Idea

Use the Rule of Sum when choices are **mutually exclusive**.

Example

A student can choose either:

- 3 science subjects, or
- 2 arts subjects

Total choices:

$$3 + 2 = 5$$

1.2.2 Rule of Product (Multiplication Principle)

If a task consists of two independent stages:

- First stage can be done in m ways
- Second stage in n ways

Total number of ways:

$$m \times n$$

Key Idea

Use the Rule of Product when tasks occur in **sequence**.

**Example**

A password consists of:

- 2 letters (26 choices each)
- 1 digit (10 choices)

Total passwords:

$$26 \times 26 \times 10 = 6760$$

1.3 Permutations

1.3.1 Definition

A permutation is an arrangement of objects where **order matters**.

1.3.2 Permutation Formula

Number of ways to arrange r objects from n distinct objects:

$${}^n P_r = \frac{n!}{(n-r)!}$$

1.3.3 Special Case: All Objects

$${}^n P_n = n!$$

Important

Factorial:

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 1$$

Example 1

In how many ways can 3 students be arranged out of 5?

$${}_5 P_3 = \frac{5!}{2!} = \frac{120}{2} = 60$$

1.3.4 Permutation with Repetition Allowed

If repetition is allowed:

$$n^r$$

**Example 2**

How many 3-digit numbers can be formed using digits 0–9?

$$10^3 = 1000$$

1.4 Combinations

1.4.1 Definition

A combination is a selection of objects where **order does not matter**.

1.4.2 Combination Formula

$${}^nC_r = \frac{n!}{r!(n-r)!}$$

1.4.3 Relation Between Permutation and Combination

$${}^nP_r = {}^nC_r \times r!$$

Example 1

Choose 2 students from 5:

$${}^5C_2 = \frac{5!}{2!3!} = \frac{120}{2 \times 6} = 10$$

Example 2 (Understanding Difference)

Arrange 2 students from 5:

$${}^5P_2 = 20$$

Select 2 students from 5:

$${}^5C_2 = 10$$

Clarification

Permutation = Arrangement (Order matters)

Combination = Selection (Order does NOT matter)



1.5 Pigeonhole Principle

1.5.1 Basic Idea

If more objects are placed into fewer boxes, then at least one box must contain more than one object.

1.5.2 Mathematical Statement

If n objects are placed into k boxes, then at least one box contains:

$$\left\lceil \frac{n}{k} \right\rceil$$

objects.

1.5.3 Simple Version

If $n > k$, then at least one box has more than one object.

1.5.4 Example 1

If 10 socks are placed into 9 drawers, then at least one drawer has:

$$\left\lceil \frac{10}{9} \right\rceil = 2$$

socks.

1.5.5 Example 2

In a class of 13 students, prove that at least two have birthdays in the same month.

Solution:

- Number of students = 13
- Number of months = 12

Since $13 > 12$, by pigeonhole principle, at least two students share the same birth month.

1.5.6 Real-Life Interpretation

- At least two people share the same birthday in a group of 367 people.
- In hashing (data structures), collisions are unavoidable.



Key Insight

The pigeonhole principle is not about exact counting, but about **guaranteeing existence**.

1.6 Summary

- Rule of Sum: Add when choices are exclusive.
- Rule of Product: Multiply when tasks are sequential.
- Permutations: Order matters.
- Combinations: Order does not matter.
- Pigeonhole Principle: Ensures minimum occurrence.

These concepts form the backbone of combinatorics and are widely used in problem-solving across mathematics and computer science.

Chapter 2

Graph Theory

2.1 Introduction to Graphs

Graph Theory is a branch of mathematics used to study relationships between objects. It is widely used in computer science, networking, social media analysis, and optimization problems.

2.1.1 Definition of a Graph

A graph G is defined as:

$$G = (V, E)$$

where:

- V = set of vertices (nodes)
- E = set of edges (connections between vertices)

2.1.2 Example

Let:

$$V = \{A, B, C\}, \quad E = \{(A, B), (B, C)\}$$

This represents a graph where:

- A is connected to B
- B is connected to C

Key Idea

Graphs model **relationships** — vertices represent entities and edges represent connections.



2.2 Types of Graphs

2.2.1 Undirected Graphs

In an undirected graph, edges have no direction.

Definition

An edge (u, v) is the same as (v, u) .

Example

If A is connected to B , then B is also connected to A .

2.2.2 Directed Graphs (Digraphs)

In a directed graph, edges have direction.

Definition

An edge (u, v) means a connection from u to v , but not necessarily from v to u .

Example

Instagram following system:

- If A follows B , it does not imply B follows A .

Clarification

Undirected Graph: Mutual relationship

Directed Graph: One-way relationship

2.3 Subgraphs

2.3.1 Definition

A subgraph is a graph formed from a subset of vertices and edges of another graph.

2.3.2 Example

If:

$$G = (V, E), \quad V = \{A, B, C, D\}$$



A subgraph can be:

$$V' = \{A, B\}, \quad E' = \{(A, B)\}$$

2.3.3 Important Note

- All edges in the subgraph must exist in the original graph.

Key Insight

A subgraph is simply a **smaller portion** of a larger graph.

2.4 Multigraphs

2.4.1 Definition

A multigraph is a graph that allows:

- Multiple edges between the same pair of vertices

2.4.2 Example

Between cities:

- Two cities may be connected by multiple roads, flights, or railways.

2.4.3 Key Point

- Unlike simple graphs, repeated edges are allowed.

2.5 Weighted Graphs

2.5.1 Definition

A weighted graph assigns a value (weight) to each edge.

2.5.2 Example

- Distance between cities
- Cost of travel
- Time taken



2.5.3 Representation

$$E = \{(u, v, w)\}$$

where w is the weight.

Example

$(A, B, 5) \Rightarrow$ Distance between A and B is 5 units

Important

Weighted graphs are heavily used in algorithms like **Dijkstra's Algorithm**.

2.6 Homomorphic Graphs

2.6.1 Definition

A graph homomorphism is a mapping between two graphs that preserves adjacency.

If there exists a function:

$$f : V(G) \rightarrow V(H)$$

such that:

$$(u, v) \in E(G) \Rightarrow (f(u), f(v)) \in E(H)$$

2.6.2 Intuition

- Structure is preserved, but size may change.

2.6.3 Example

Collapsing multiple nodes into a single node while maintaining connections.

Key Idea

Homomorphism simplifies graphs while preserving their **connectivity structure**.

2.7 Isomorphic Graphs

2.7.1 Definition

Two graphs are isomorphic if they are structurally identical, even if their vertex labels differ.



2.7.2 Mathematical Definition

A bijection:

$$f : V(G) \rightarrow V(H)$$

such that:

$$(u, v) \in E(G) \Leftrightarrow (f(u), f(v)) \in E(H)$$

2.7.3 Key Properties

- Same number of vertices
- Same number of edges
- Same degree sequence

Example

Graph with edges:

$$(A, B), (B, C)$$

is isomorphic to:

$$(1, 2), (2, 3)$$

Clarification

Isomorphic graphs may **look different** but are **structurally identical**.

2.8 Paths and Circuits

2.8.1 Path

Definition

A path is a sequence of vertices where each adjacent pair is connected by an edge.

Example

$$A \rightarrow B \rightarrow C$$

2.8.2 Length of Path

Number of edges in the path.



2.8.3 Simple Path

A path with no repeated vertices.

2.8.4 Circuit (Cycle)

Definition

A circuit is a path that starts and ends at the same vertex.

Example

$$A \rightarrow B \rightarrow C \rightarrow A$$

2.8.5 Simple Circuit

A circuit with no repeated vertices except the starting/ending vertex.

Key Difference

Path: Start and end are different

Circuit: Start and end are same

2.8.6 Solved Example

Determine whether the following sequence is a path or circuit:

$$A \rightarrow B \rightarrow C \rightarrow A$$

Solution:

- Starts at A and ends at A
- No repetition except start/end

Hence, it is a **simple circuit**.

2.9 Summary

- Graphs represent relationships using vertices and edges.
- Directed and undirected graphs differ based on edge direction.
- Subgraphs are smaller parts of a graph.
- Multigraphs allow multiple edges.



- Weighted graphs assign values to edges.
- Homomorphic graphs preserve structure under mapping.
- Isomorphic graphs are structurally identical.
- Paths and circuits define traversal in graphs.

These concepts form the foundation for advanced topics like graph algorithms, network analysis, and optimization.

2.10 Shortest Path in Weighted Graphs (Dijkstra's Algorithm)

2.10.1 Introduction

In many real-life problems, we want to find the **shortest path** between two vertices in a weighted graph. For example:

- Shortest route between two cities (Google Maps)
- Minimum cost in networks
- Optimal path in AI search problems

2.10.2 Key Idea of Dijkstra's Algorithm

Dijkstra's Algorithm finds the shortest path from a **source vertex** to all other vertices in a graph with **non-negative weights**.

Important Condition

Dijkstra's Algorithm works only when all edge weights are **non-negative**.

2.10.3 Algorithm Steps

1. Assign distance 0 to the source vertex and ∞ to all others.
2. Mark all vertices as unvisited.
3. Select the unvisited vertex with the smallest distance.
4. Update distances of its neighbors.
5. Mark it as visited.
6. Repeat until all vertices are visited.



2.10.4 Solved Example

Consider a graph:

- $A \rightarrow B = 4$
- $A \rightarrow C = 2$
- $C \rightarrow B = 1$

Step 1: Start from A

$$d(A) = 0, \quad d(B) = \infty, \quad d(C) = \infty$$

Step 2: Update neighbors:

$$d(B) = 4, \quad d(C) = 2$$

Step 3: Choose C (smallest distance = 2)

Update:

$$d(B) = \min(4, 2 + 1) = 3$$

Final shortest distances:

$$A \rightarrow B = 3, \quad A \rightarrow C = 2$$

2.11 Eulerian Paths and Circuits

2.11.1 Definition

- **Eulerian Path:** A path that visits every edge exactly once.
- **Eulerian Circuit:** A circuit that visits every edge exactly once and starts/ends at the same vertex.

2.11.2 Conditions for Eulerian Graphs

Eulerian Circuit Exists If:

- All vertices have **even degree**
- Graph is connected

**Eulerian Path Exists If:**

- Exactly **two vertices have odd degree**

Quick Trick

Count the degrees of vertices:

- 0 odd $\rightarrow$ Eulerian Circuit
- 2 odd $\rightarrow$ Eulerian Path
- Otherwise $\rightarrow$ Not Eulerian

2.11.3 Example

If a graph has vertex degrees:

2, 2, 4, 4

All are even $\rightarrow$ Eulerian Circuit exists.

2.12 Hamiltonian Paths and Circuits**2.12.1 Definition**

- **Hamiltonian Path:** Visits every vertex exactly once.
- **Hamiltonian Circuit:** Visits every vertex exactly once and returns to the start.

2.12.2 Key Difference from Eulerian

- Eulerian $\rightarrow$ focuses on **edges**
- Hamiltonian $\rightarrow$ focuses on **vertices**

Clarification

Eulerian = Cover all edges

Hamiltonian = Cover all vertices

2.12.3 Important Note

Unlike Eulerian graphs, there is **no simple formula** to determine Hamiltonian paths. It is usually solved using:

- Trial and error



- Backtracking

2.12.4 Example

For a square graph (4 vertices in a cycle):

$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$$

This is a Hamiltonian Circuit.

2.13 Planar Graphs

2.13.1 Definition

A graph is planar if it can be drawn on a plane without any edges crossing.

2.13.2 Example

- Simple triangle $\rightarrow$ planar
- Square with diagonals $\rightarrow$ non-planar (edges cross)

2.13.3 Famous Non-Planar Graphs

- K_5 (complete graph with 5 vertices)
- $K_{3,3}$ (bipartite graph)

Important

A graph is non-planar if it contains K_5 or $K_{3,3}$ as a subgraph.

2.14 Euler's Formula

2.14.1 Statement

For any connected planar graph:

$$V - E + F = 2$$

where:

- V = number of vertices
- E = number of edges



- F = number of faces (regions)

2.14.2 Example

For a cube graph:

$$V = 8, \quad E = 12, \quad F = 6$$

$$8 - 12 + 6 = 2$$

2.14.3 Use of Euler's Formula

- Check if a graph is planar
- Analyze graph structure

Key Insight

Euler's Formula connects vertices, edges, and faces — a powerful tool for planar graphs.

2.15 Chromatic Numbers (Graph Coloring)

2.15.1 Introduction

Graph coloring involves assigning colors to vertices such that:

- No two adjacent vertices have the same color

2.15.2 Chromatic Number

The minimum number of colors required to color a graph is called its **chromatic number**, denoted by $\chi(G)$.

2.15.3 Examples

- A line graph $\rightarrow \chi(G) = 2$
- Triangle graph $\rightarrow \chi(G) = 3$

2.15.4 Real-Life Applications

- Map coloring (countries sharing borders)
- Timetable scheduling
- Register allocation in compilers



2.15.5 Solved Example

Consider a triangle graph:

- Each vertex connected to the other two

Solution:

- First vertex $\rightarrow$ Color 1
- Second vertex $\rightarrow$ Color 2
- Third vertex $\rightarrow$ cannot use 1 or 2 $\rightarrow$ Color 3

Thus,

$$\chi(G) = 3$$

Key Idea

Chromatic number represents the **minimum resources needed** under conflict constraints.

2.16 Summary of Advanced Concepts

- Dijkstra's Algorithm finds shortest paths in weighted graphs.
- Eulerian paths/circuits depend on vertex degrees.
- Hamiltonian paths/circuits depend on visiting all vertices.
- Planar graphs can be drawn without edge crossings.
- Euler's formula relates vertices, edges, and faces.
- Chromatic number solves coloring problems efficiently.

Chapter 3

Trees

3.1 Introduction to Trees

Trees are a special type of graph that play a crucial role in computer science, especially in data structures, databases, and hierarchical modeling.

3.1.1 Definition of a Tree

A tree is a connected graph with no cycles.

3.1.2 Formal Definition

A graph $G = (V, E)$ is called a tree if:

- It is **connected**
- It contains **no cycles**

3.1.3 Key Properties of Trees

- A tree with n vertices has exactly $n - 1$ edges.
- There is exactly one path between any two vertices.
- Adding any edge creates a cycle.
- Removing any edge disconnects the tree.

Key Idea

A tree is the **simplest connected graph** — no loops, no redundancy.



3.1.4 Real-Life Examples

- Family hierarchy
- File directory structure
- Organization charts

3.2 Difference Between Graphs and Trees

3.2.1 General Graph vs Tree

Property	Graph	Tree
Connectivity	May or may not be connected	Always connected
Cycles	May contain cycles	No cycles allowed
Edges	Any number	Exactly $n - 1$ edges
Path	May have multiple paths	Exactly one path
Structure	General	Hierarchical

Clarification

Every tree is a graph, but not every graph is a tree.

3.2.2 Example

- A triangle graph $\rightarrow$ Not a tree (contains a cycle)
- A chain of nodes $\rightarrow$ Tree (connected, no cycle)

3.3 Rooted Trees

3.3.1 Definition

A rooted tree is a tree in which one vertex is designated as the **root**.

3.3.2 Hierarchy in Rooted Trees

- Root: Top-most node
- Parent: Node connected above
- Child: Node connected below
- Leaf: Node with no children



- Internal node: Node with at least one child

3.3.3 Example

Consider a tree:

- A is root
- B and C are children of A
- D is child of B

3.3.4 Levels in a Tree

- Root is at level 0
- Its children are at level 1
- And so on...

Key Insight

Rooted trees introduce a **direction and hierarchy** into trees.

3.3.5 Height of a Tree

The height of a tree is the length of the longest path from the root to a leaf.

Example

If longest path is:

$$A \rightarrow B \rightarrow D$$

Height = 2

3.4 Path Length in Trees

3.4.1 Definition of Path

A path is a sequence of vertices connected by edges.

3.4.2 Path Length

The path length is the number of edges in the path.



3.4.3 Example

$$A \rightarrow B \rightarrow C$$

Path length = 2

3.4.4 Depth of a Node

The depth of a node is the number of edges from the root to that node.

Example

- Root $A \rightarrow \text{depth} = 0$
- Node $B \rightarrow \text{depth} = 1$
- Node $D \rightarrow \text{depth} = 2$

3.4.5 Internal Path Length

The sum of path lengths from root to all internal nodes.

3.4.6 External Path Length

The sum of path lengths from root to all leaf nodes.

Important

Path length is always counted in terms of **edges**, not vertices.

3.4.7 Solved Example

Consider a tree:

- A (root)
- B, C children of A
- D child of B

Depths:

- $A = 0$
- $B = 1, C = 1$
- $D = 2$



Internal Path Length:

$$0 + 1 = 1$$

External Path Length:

$$1 + 2 = 3$$

3.5 Summary

- Trees are connected graphs without cycles.
- Trees always have $n - 1$ edges for n vertices.
- Trees differ from graphs in structure and properties.
- Rooted trees introduce hierarchy (parent, child, levels).
- Path length measures distance using edges.

Trees form the foundation for advanced data structures like binary trees, heaps, and search trees, making them essential for efficient computation.

3.6 Spanning Trees

3.6.1 Definition

A spanning tree of a connected graph is a subgraph that:

- Includes all the vertices of the graph
- Is a tree (i.e., connected and acyclic)

3.6.2 Key Properties

- A spanning tree has exactly $n - 1$ edges (for n vertices)
- It contains no cycles
- A graph can have multiple spanning trees

Key Idea

A spanning tree connects all vertices using the **minimum number of edges** without forming cycles.



3.6.3 Example

If a graph has vertices $\{A, B, C\}$ and edges $\{AB, BC, AC\}$:

- Possible spanning trees:
 - $\{AB, BC\}$
 - $\{AB, AC\}$
 - $\{AC, BC\}$

3.7 Cut-Sets

3.7.1 Definition

A cut-set is a set of edges whose removal disconnects the graph.

3.7.2 Understanding the Concept

- Removing these edges breaks the graph into two or more components
- It helps in analyzing network reliability

3.7.3 Example

If removing edge (A, B) disconnects the graph, then:

$$\{(A, B)\}$$

is a cut-set.

Key Insight

Cut-sets represent **critical connections** in a graph.

3.8 Minimum Spanning Trees (Kruskal's Algorithm)

3.8.1 Definition

A Minimum Spanning Tree (MST) is a spanning tree with the **minimum total edge weight**.



3.8.2 Why MST?

- Minimizes cost in networks (roads, cables)
- Used in clustering and optimization

3.8.3 Kruskal's Algorithm

Idea

Select edges in increasing order of weight while avoiding cycles.

Steps

1. Sort all edges in increasing order of weight
2. Pick the smallest edge
3. Add it if it does not form a cycle
4. Repeat until $n - 1$ edges are selected

Important

Cycle detection is essential in Kruskal's Algorithm.

3.8.4 Solved Example

Consider edges:

$$AB = 1, \quad BC = 2, \quad AC = 3$$

Step 1: Sort edges:

$$1, 2, 3$$

Step 2: Select AB (weight 1)

Step 3: Select BC (weight 2)

Step 4: Skip AC (forms cycle)

MST edges:

$$\{AB, BC\}$$

Total weight:

$$1 + 2 = 3$$



3.9 Binary Trees

3.9.1 Definition

A binary tree is a rooted tree in which each node has at most **two children**:

- Left child
- Right child

3.9.2 Types of Binary Trees

- **Full Binary Tree**: Every node has 0 or 2 children
- **Complete Binary Tree**: All levels filled except possibly last
- **Perfect Binary Tree**: All levels completely filled

Key Idea

Binary trees are fundamental in data structures like **Binary Search Trees (BST)** and heaps.

3.10 Traversal of Binary Trees

Traversal means visiting all nodes of a tree in a specific order.

3.10.1 Types of Traversal

1. Inorder Traversal (Left, Root, Right)

- Visit left subtree
- Visit root
- Visit right subtree

2. Preorder Traversal (Root, Left, Right)

- Visit root
- Visit left subtree
- Visit right subtree



3. Postorder Traversal (Left, Right, Root)

- Visit left subtree
- Visit right subtree
- Visit root

Quick Memory Trick

Inorder $\rightarrow$ LNR

Preorder $\rightarrow$ NLR

Postorder $\rightarrow$ LRN

3.10.2 Solved Example

Consider binary tree:

- Root: A
- Left child: B
- Right child: C

Inorder:

B, A, C

Preorder:

A, B, C

Postorder:

B, C, A

3.11 Summary

- Spanning trees connect all vertices without cycles.
- Cut-sets identify critical edges in a graph.
- Minimum spanning trees minimize total weight.
- Kruskal's algorithm builds MST using greedy approach.
- Binary trees restrict nodes to two children.
- Tree traversal defines systematic visiting of nodes.



Final Examination – Sample Paper

Total Marks : 60

Time Allotted : 3 hours

Instructions:

- The question paper consists of 3 sections. It is compulsory for students to attempt all questions of Section A and Section B.
- Section A has 5 questions of 2 marks each, Section B has 4 questions of 5 marks each, and Section C has 4 questions of 10 marks each (out of which any 3 are to be attempted).
- Students to attempt ANY ONE question from question no. 12 & question no. 13 of Section C.

Q.No.	Statement
Section A (5×2 = 10 marks)	
1	In a class of 60 students, 35 study Mathematics, 30 study Physics, and 15 study both. Find the number of students who study at least one of the two subjects.
2	Find the general solution of the recurrence relation: $a_n - 4a_{n-1} + 4a_{n-2} = 0$ and discuss the nature of its roots.
3	Verify Euler's formula for a planar graph with $V = 6$ and $E = 9$. Find F .
4	In how many ways can 5 different books be arranged on a shelf such that 2 specific books are always together?
5	Given a binary tree with 7 nodes, determine its possible height range and justify.
Section B (4×5 = 20 marks)	
6	Given functions $f(x)$ and $g(x)$, analyze whether: $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$ holds. Provide justification with an example.



Q.No.	Statement
7	Prove that every cyclic group is abelian. Extend the argument to determine whether the converse holds.
8	Apply Dijkstra's Algorithm to the weighted graph with edges: AB=4, AC=1, CB=2, BD=1, CD=5 Find shortest path from A to all vertices.
9	A committee of 4 is to be formed from 6 men and 5 women. In how many ways can it be formed if at least 2 women must be included?
Section C (3×10 = 30 marks)	
10	Define relations and explain equivalence relations with a suitable example. Using mathematical induction, prove that: $1^2 + 2^2 + 3^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$
11	Solve the following recurrence relation: $a_n - 2a_{n-1} = 3, \quad a_0 = 2$ and interpret the nature of its solution.
Optional Question for Section C	
12	What do you mean by Eulerian circuits? State their characteristic properties which differ them from other circuits or graphs. Determine whether a graph with vertices of degrees (2, 2, 2, 2) has an Eulerian circuit. Justify.
OR	
13	Perform inorder, preorder, and postorder traversals of the binary tree and explain the significance of each traversal. <pre> 7 /\ 5 6 /\ /\ 1 2 3 4 </pre>

Answer Key

Section A

Question 1

Given:

- $|M| = 35, |P| = 30, |M \cap P| = 15$

Using Inclusion-Exclusion Principle:

$$\begin{aligned}|M \cup P| &= |M| + |P| - |M \cap P| \\ &= 35 + 30 - 15 = 50\end{aligned}$$

Answer: 50 students study at least one subject.

Key Point

Always subtract intersection to avoid double counting.

Question 2

Given recurrence:

$$a_n - 4a_{n-1} + 4a_{n-2} = 0$$

Characteristic Equation:

$$r^2 - 4r + 4 = 0 \Rightarrow (r - 2)^2 = 0$$

Roots:

- Repeated root: $r = 2$

General Solution:

$$a_n = (A + Bn)2^n$$

Nature of Roots:



- Real and equal roots

Key Idea

Repeated roots $\Rightarrow$ multiply one term by n .

Question 3

Euler's Formula:

$$V - E + F = 2$$

Substitute:

$$6 - 9 + F = 2$$

$$F = 2 - 6 + 9 = 5$$

Answer: Number of faces = 5

Key Point

Euler's formula is valid only for **connected planar graphs**.

Question 4

Idea: Treat 2 specific books as one unit

- Total units = 4 (combined pair + 3 other books)
- Arrange these: $4! = 24$
- Internal arrangement of pair: $2! = 2$

Total ways:

$$4! \times 2! = 24 \times 2 = 48$$

Answer: 48 ways

Key Trick

Group constrained elements into a single block.

Question 5

Given: Number of nodes = 7

Height depends on structure:



Case	Height
Skewed tree (like linked list)	6
Balanced tree	$\lfloor \log_2 7 \rfloor = 2$

Possible Height Range:

$$2 \leq h \leq 6$$

Justification:

- Minimum height when tree is balanced
- Maximum height when tree is completely skewed

Key Insight

Height varies based on structure, not just number of nodes.

Section B

Question 6

Statement:

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

Result: True (under invertibility)

Justification:

- Let $y = f(g(x))$
- Apply inverse step-by-step:
 - First undo f using f^{-1}
 - Then undo g using g^{-1}
- Hence:

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

Example:

- $f(x) = 2x, g(x) = x + 3$
- $f \circ g = 2(x + 3) = 2x + 6$
- $(f \circ g)^{-1}(x) = \frac{x-6}{2}$



RHS:

- $f^{-1}(x) = x/2$
- $g^{-1}(x) = x - 3$
- $g^{-1}(f^{-1}(x)) = \frac{x}{2} - 3 = \frac{x-6}{2}$

Conclusion: Both sides equal.

Key Point

Order reverses when taking inverse of composition.

Question 7

To Prove: Cyclic group $\Rightarrow$ Abelian

Proof:

- Let $G = \langle a \rangle$
- Any elements: $x = a^m, y = a^n$
- Then:

$$xy = a^m a^n = a^{m+n}$$

$$yx = a^n a^m = a^{n+m}$$

- Since $m + n = n + m$:

$$xy = yx$$

Hence: G is abelian.

Converse:

- Not every abelian group is cyclic

Counterexample:

- $(\mathbb{Z}_2 \times \mathbb{Z}_2)$ is abelian but not cyclic

Property	Result
Cyclic $\Rightarrow$ Abelian	True
Abelian $\Rightarrow$ Cyclic	False



Question 8

Edges:

$$AB = 4, AC = 1, CB = 2, BD = 1, CD = 5$$

Step-wise (Dijkstra from A):

- Initialize:

$$d(A) = 0, d(B) = \infty, d(C) = \infty, d(D) = \infty$$

- From A:

$$d(B) = 4, d(C) = 1$$

- Choose C (smallest = 1):

$$d(B) = \min(4, 1 + 2) = 3, d(D) = \min(\infty, 1 + 5) = 6$$

- Choose B (3):

$$d(D) = \min(6, 3 + 1) = 4$$

Final distances:

$$A = 0, C = 1, B = 3, D = 4$$

Shortest Paths:

- $A \rightarrow C$
- $A \rightarrow C \rightarrow B$
- $A \rightarrow C \rightarrow B \rightarrow D$

Key Idea

Always pick the vertex with minimum tentative distance.

Question 9

Given:

- Men = 6, Women = 5
- Committee size = 4
- At least 2 women

Cases:



- Case 1: 2 women, 2 men

$${}^5C_2 \times {}^6C_2 = 10 \times 15 = 150$$

- Case 2: 3 women, 1 man

$${}^5C_3 \times {}^6C_1 = 10 \times 6 = 60$$

- Case 3: 4 women

$${}^5C_4 = 5$$

Total ways:

$$150 + 60 + 5 = 215$$

Answer: 215 ways

Key Strategy

Break into cases based on constraints.

3.12 Section C Answers

Question 10

Relation:

- A relation R from set A to B is a subset of $A \times B$
- i.e., $R \subseteq A \times B$

Equivalence Relation: A relation R on set A is called an equivalence relation if it is:

- **Reflexive:** aRa
- **Symmetric:** $aRb \Rightarrow bRa$
- **Transitive:** $aRb, bRc \Rightarrow aRc$

Example: Let R on $\mathbb{Z}$ be defined as:

$$aRb \iff a \equiv b \pmod{3}$$

- Reflexive: $a - a = 0$ divisible by 3
- Symmetric: if $a - b$ divisible by 3, so is $b - a$
- Transitive: follows from divisibility



Hence: R is an equivalence relation.

Key Insight

Equivalence relations partition a set into disjoint equivalence classes.

Mathematical Induction Proof:

Statement:

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

Step 1: Base Case ($n = 1$)

$$LHS = 1, \quad RHS = \frac{1 \cdot 2 \cdot 3}{6} = 1$$

True.

Step 2: Inductive Hypothesis Assume true for $n = k$:

$$1^2 + 2^2 + \dots + k^2 = \frac{k(k+1)(2k+1)}{6}$$

Step 3: For $n = k + 1$

$$LHS = \frac{k(k+1)(2k+1)}{6} + (k+1)^2$$

Simplify:

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

$$= \frac{(k+1)[k(2k+1) + 6(k+1)]}{6}$$

$$= \frac{(k+1)(2k^2 + 7k + 6)}{6}$$

$$= \frac{(k+1)(k+2)(2k+3)}{6}$$

This equals RHS for $n = k + 1$

Conclusion: Proven by induction.

Question 11

Given:

$$a_n - 2a_{n-1} = 3, \quad a_0 = 2$$



Step 1: Homogeneous Solution

$$a_n - 2a_{n-1} = 0 \Rightarrow r = 2$$

$$a_n^{(h)} = C \cdot 2^n$$

Step 2: Particular Solution Assume constant: $a_n = A$

$$A - 2A = 3 \Rightarrow -A = 3 \Rightarrow A = -3$$

General Solution:

$$a_n = C2^n - 3$$

Using $a_0 = 2$:

$$2 = C - 3 \Rightarrow C = 5$$

Final Answer:

$$a_n = 5 \cdot 2^n - 3$$

Nature of Solution:

- Exponential growth due to 2^n
- Shifted by constant -3

Key Idea

Solution = Homogeneous part + Particular part.

Question 12 (Optional)

Eulerian Circuit:

- A circuit that visits every edge exactly once and returns to the starting vertex

Characteristics:

Property	Condition
Connected graph	Required
Degree of vertices	All even
Edge usage	Each edge exactly once

Difference from other paths:



- Eulerian: focuses on edges
- Hamiltonian: focuses on vertices

Given degrees: $(2, 2, 2, 2)$

- All vertices have even degree
- Graph assumed connected

Conclusion:

- Eulerian circuit exists

Quick Rule

All vertices even $\Rightarrow$ Eulerian circuit exists.

Question 13 (Alternative)

Tree Structure Interpretation:

- Root: 7
- Left subtree: $5 \rightarrow (1, 2)$
- Right subtree: $6 \rightarrow (3, 4)$

Traversals:

- **Inorder (LNR):**

1, 5, 2, 7, 3, 6, 4

- **Preorder (NLR):**

7, 5, 1, 2, 6, 3, 4

- **Postorder (LRN):**

1, 2, 5, 3, 4, 6, 7

Significance:

Traversal	Use
Inorder	Sorted output (BST)
Preorder	Tree copying / prefix expression
Postorder	Deletion / postfix evaluation

Closing Note

I sincerely hope that these notes and the preparatory questions included alongside them prove helpful in strengthening your understanding and boosting your confidence for the examination you are preparing for.

As you revise, try not to rush through the content—pause at examples, work out the numericals on your own, and revisit sections that feel slightly uncomfortable. That discomfort usually signals learning in progress. Use these notes as a guide and a companion, not as something to be memorized line by line.

Wishing you the very best for your exams—may your preparation feel clear, structured, and calm when it matters most.

For future updates, refinements, and additional resources, keep checking **Noteify**:

<https://subhayu18.github.io/Noteify/>

If you have any doubts, suggestions, or simply wish to connect, feel free to reach out to me on LinkedIn:

<https://www.linkedin.com/in/subhayu-mukherjee-146421294/>

You can also use the direct contact options provided on the Noteify website itself.

Regards,
Subhayu