GeeksforGeeks

GATE फर्रै CSE

DISCRETE MATHEMATICS

SHORT NOTES

GRAPH THEORY

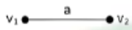
GRAPH

- Let **G(V, E)** be the graph:
- ○ **V** = set of vertices
- ○ **E** = edge set

Each edge must be associated with an unordered pair of vertices.

- ○ |V| = number of vertices = **order of graph**
- ○ |E| = number of edges = **size of graph**

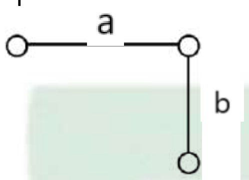
- **Adjacent Vertices :** Two vertices are adjacent if they have a **common edge**.



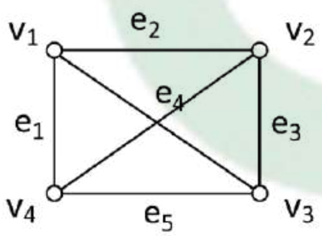
Example: If, then $\mathbf{v_1}$ and $\mathbf{v_2}$ are adjacent vertices.

- **Adjacent Edges :** Two edges are adjacent if they share a **common vertex.**

Example:



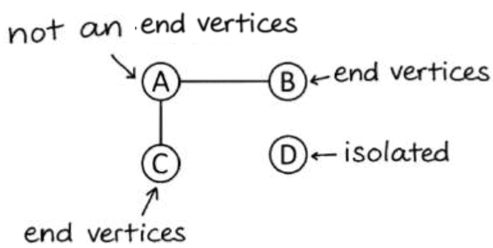
a, b →adjacent edges.



$e_1, e_2 \rightarrow$ adjacent edges.
$v_1, v_4 \rightarrow$ adjacent vertices

- **End Vertices :** Unordered pair of vertices at the end of an **edge**.

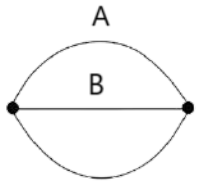
Example: b & c are end vertices



Types of Edges

1. **Parallel Edge :**Two edges connecting the same pair of vertices:

Example:

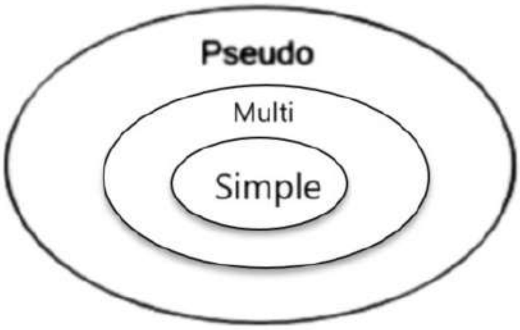


here, A & B are parallel edges A B

2. **Self Loop:** An edge that connects a vertex to itself.
   Example: Q - loop back to Q
   Q

Graph Type	Self Loop	Multi-edge(Parallel edges)
Simple Graph	×	×
Multi Graph	×	✓
Pseudo Graph	✓	✓

- **Graph Types** :



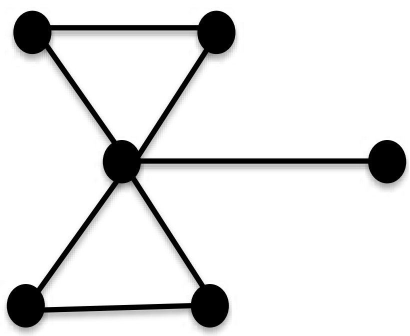
- **Degree (or Valency):** The degree of a vertex is the **number of edges incident** to it.
  Note: A **loop** is counted **twice** in the degree of a vertex.

- **Theorem 1:** Maximum degree of a vertex in a simple graph with n vertices is ≤ n−1.
  $\sum d(v_i) = 2|E|$
  where, |E| = number of edges
  $|V_i|$ = vertices

  Hand Shaking Lemma: Sum of all degrees = 2 * sum of all edges.
  Theorem 2: vertices with odd degree is always even.
- $\sum_{\text{odd}} d(v) = \text{even}$

  Note: The number of vertices with odd degree is even.
- **Theorem 3:** Max degrees in a given graph G is denoted as Δ(G) & Min degree is denoted as δ(G).
  $\delta(G) \le 2e/n \le \Delta(G) \le n-1$
  where, Δ(G) = Maximum degree in graph G
  δ(G) = Minimum degree in graph G
  e = Number of edges
  n = Number of vertices
- **Theorem 4:** Maximum degree in a simple graph:
  $\Delta(G) \le (n-1)$
  where n = number of vertices.

- **Theorem 5:** Maximum number of edges possible in a simple graph:
  $e \le [n(n-1)]/2$
- **Maximum number** of simple graphs: $2^{[n(n-1)]/2}$
- **Number of graphs** with e edges: $^{n(n-1)/2}C_e$
- A **degree sequence** is a list of the degrees of all vertices in a graph, arranged in **non-increasing (↓)** or **non-decreasing (↑)** order.
- **Graphical Sequence**: A **graphical sequence** is a **degree sequence** that corresponds to a **simple graph**.
- **eq**: 5, 2, 2, 2, 1.



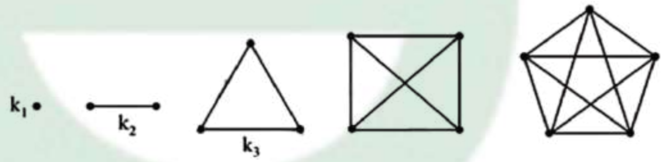
Note: If at any point a **negative value** appears **✕ Not a Graphical Sequence → No Simple Graph** exists.

- **Havel–Hakimi Algorithm:** It is used to **determine** whether a **degree sequence of positive integers is graphical**.

- **Some Important Terms:**
- **Distance:** Length or shortest path between two vertices.
- **Eccentricity(e(v)):** The farthest distance between a vertex **v** and any other vertex in the graph.
- **Radius:** The **smallest** eccentricity among all vertices in the graph.
- **Diameter:** The **largest** eccentricity among all vertices in the graph.
- **Length of Path:** The number of edges involved in the path between two vertices.
- **Inequality Relation:** rad(G)≤dia(G)≤2·rad(G)

TYPES OF SIMPLE GRAPH

1. Complete Graph (K_n : n ≥ 1):



$\Sigma d(v_1) = n * (n-1)$
- **Degree Sequence** : Degree of every vertex is n – 1.
- **No. of edges** = [n*(n-1)] / 2
- **Degree of vertices** = δ(G) = 2e/n = Δ(G) = n−1

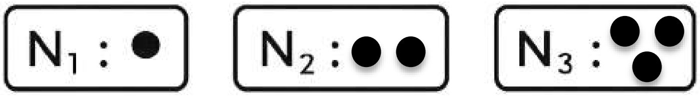
2. **Regular Graph(K- Regular)**: Each vertices has some degree and that is of K.

    $\delta(G) = 2e/n = \Delta(G)$

    Note - Every K_n is regular (True) and vice - versa not True.

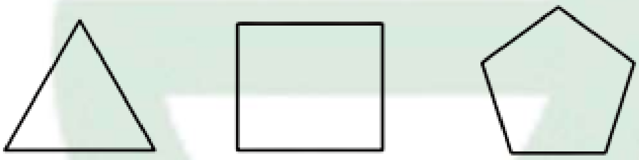
    Null Graph → 0 Regular graph

3. **Null Graph:** Null Graph having **No edges**.
    Example - (N_n : n ≥ 1)



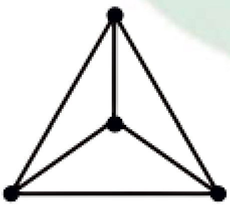
4. **Cycle Graph (C_n : n ≥ 3):** A **cycle graph** is a graph in which:

- Each vertex has **degree 2**.
- Every C_n is a **regular graph**
- **Number of edges** = number of vertices = n.
- **Degree Sequence :** 2, 2, 2, 2, 2...2
- |n| = |e| & **$\delta(G) = 2e/n = \Delta(G) = 2$**

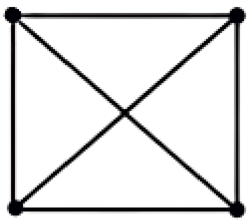


C_3 C_4 C_5

5. **Wheel Graph (W_n : n ≥ 4):**



W_4



W_5

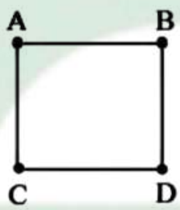
- A Graph(G) is Wheel Graph (W_n) then **e = 2*(n-1)** → True {vice - versa not True}
- **Degree Sequence:** {n-1, 3, 3, 3,....3}(with n-1 threes)

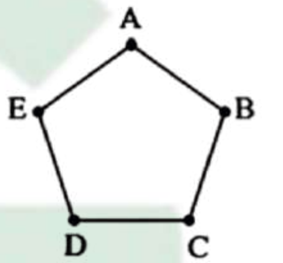
- **Degree** (middle vertices) = n-1

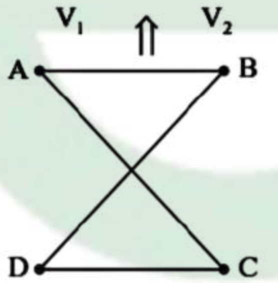
6. **Bipartite Graph(L(G)):** A graph is **bipartite if** its vertex set V can be partitioned into two disjoint sets V1 and V2, such that **every edge** connects a vertex in V1 to a vertex in V2.
    Adjacent **edges should not be** in the **same partitions**.

Note:

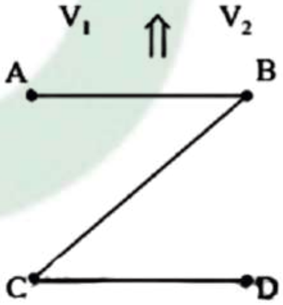
- A graph containing an **odd-length cycle** is **not bipartite**.
- A complete graph K_n is **never bipartite**.(n ≥ 3)
- **Maximum edges** occur when each partitions contain n/2 vertices (almost).
- Bipartite graphs do not contain **odd length cycle.**







∴ C_4 is a bipartite graph

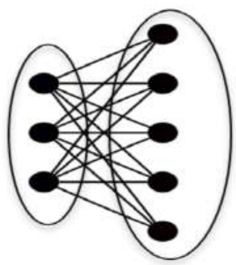


Now adding E in either of the sets violates the property of bipartite graph
∴ C_5 is not a bipartite graph

- **Theorem 6:** Maximum number of edges in a bipartite graph: $e \le \lfloor n^2/4 \rfloor$

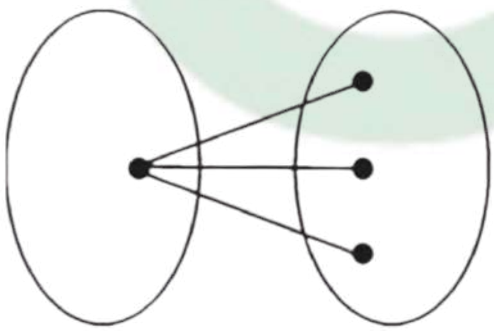
7. Complete bipartite graph ($K_{m,n}$) :

- $|V_1|=m$, $|V_2|=n$
- Each vertex in V_1 is connected to **every** vertex in V_2
- **Total number of vertices** = m+n
- **Total number of edges** = m·n
- **Maximum degree $\Delta(K_{m,n})$** = max(m,n)
- **Minimum degree $\delta(K_{m,n})$** = min(m,n)



8. Star Graph ($K_{1,n-1}$):

- $|V_1|=1$, $|V_2|=n-1$
- **Total vertices** = n
- **Total edges** = n−1
- **Maximum degree $\Delta(K1,n-1)$** = n−1
- **Minimum degree $\delta(K1,n-1)$** = 1
- There is **1 central vertex**.
- It is connected to **n−1** outer (leaf) vertices.



9. Complement Graph ($\overline{G}$):

- $\overline{G}$: Edge is **present**
- ($\overline{G}$): Edge is **absent in G**, but **present** in G.
- $V(G)=V(\overline{G})$
- $G+\overline{G} = K_n$
- $e(G)+e(\overline{G})=[n(n-1)]/2$

10. Line Graph(L(G)):

- Edges of graph G will be **vertices** in graph L(G).
- **Degree** of line graph of a complete graph K_n: $d(L(K_n))=2(n-2)$

11. Isomorphic Graph:

- A graph having the same incident property, same number of edges, vertices, and degree as another graph.(Unique Mapping)

12. Self-Complement Graph ($G \equiv (\overline{G})$): Isomorphic to its own complement.

- $e(G)+e(\overline{G})=e(K_n)$
- $e = \frac{n(n-1)}{4}$

13. Hypercube (n≥1): n ⟶ Bit Signal or n-cube

- **Total vertices** = 2^n
- **Degree** of each vertex = n
- $e(G_n)=n\cdot 2^{n-1}$
- $e\ \overline{G_n}=[2^n(2^n-1)]/2-n\cdot 2^{n-1}$
- Two vertices are **adjacent** if they differ by 1bit.

Name	Repeated Vertex (Vertices)	Repeated Edge(s)	Open	Closed
Walk (open)	Yes	Yes	Yes	
Walk (closed)	Yes	Yes		Yes
Trail	Yes	No	Yes	
Circuit	Yes	No		Yes
Path	No	No	Yes	
Cycle	No	No		Yes

- **Graph**

Connected (K=1)
Path is available between all pair of vertices.

Disconnected (K>=2)

Path is unavailabe between at least ⊥ pair of vertices

K→ No. of Component

- Every disconnected graph contains a **connected subgraph. (K)**

Disconnected Connected

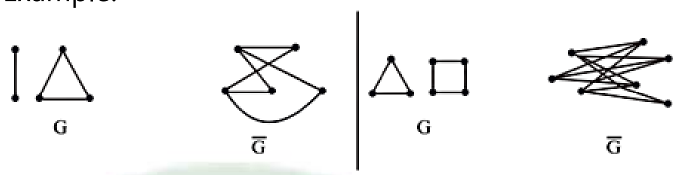
- If G is connected, then G¬ is disconnected

- **Connected Graph (K = 1)**
- ○ n−1 ≤ e ≤ [n(n−1)]/2
- ○ **Tree:**
- ■ Minimally connected graph
- ■ Acyclic
- ■ Unique path is available between all pairs of vertices

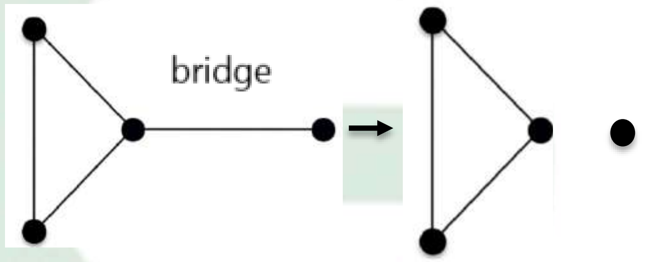
- **Disconnected Graph (K > 2)**
- ○ N-k ≤ e ≤ [(n−k)(n−k+1)]/2
- ○ **Max Edges : K=2**
- ○ If δ(G)≥(n−1)/2 ⇒ Graph is **Connected** (Vice versa → Not true).

- ○ If G is connected then G¬ may be connected or disconnected

- **Theorem 6:** If G is disconnected, then G¬ is connected.
Example:



- Connected → Disconnected
- **Bridge / Cut Edge :** Used to **make graph disconnected.**
- ○ Involves **removal of a single edge**



- ○ If cut-edge exists, it **does not belong to a cycle**
- ■ **Cut Edge Set / Cut Set → Set of edges**
- **Cut Vertex / Articulation Point :** Used to **make graph disconnected.**
  - ○ Removal of a **single vertex** increases number of components

- ○ If **no cut vertex** → Graph is **Biconnected.**
- ■ **Cut Vertex Set** → Set of vertices

There can be multiple cut sets or cut vertex sets of different sizes.

- **Edge-Connectivity (λ(G))**
- ○ Removal of Minimum number of **edges**.
- ○ **λ(G) ≤ δ(G)**
- **Vertex-Connectivity (K(G))**
- ○ Removal of Minimum number of **vertices**.
- ○ **K(G) ≤ λ(G)**

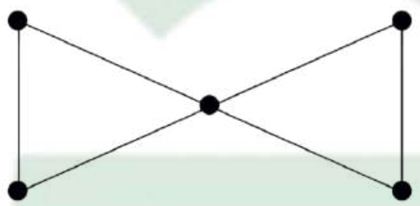
Important Inequality Chain:

$K(G) \le \lambda(G) \le \delta(G) \le 2e/n \le \Delta(G) \le n-1$

- **For Complete Bipartite Graph ($K_{m,n}$):**
  - $\lambda(K_{m,n}) = \min(m,n)$
  - $K(K_{m,n}) = \min(m,n)$
- If a **connected graph** has a **cut point**, it also has a **cut edge**, and vice versa.
- A **complete graph** does **not** have any cut vertex.

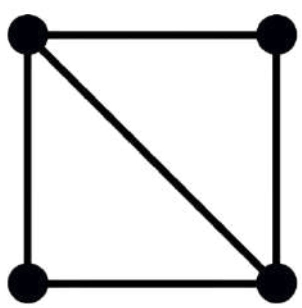
Euler Graph: Containing Euler Circuit / Euler Cycle

- Closed trail where all **edges** should be covered exactly **once**.
- **Euler Line / Path:** Open trail (start vertex ≠ end vertex) + cover **all edges once**.
- Graph containing Euler line → **Semi-Eulerian**
- **Euler Graph → Always even degree vertices**



Hamiltonian Graph : Containing Hamiltonian Cycle

- Closed path (cycle) where all **vertices** should be covered **once**.
- **Hamiltonian Path: All vertices covered once + open path (start vertex ≠ end vertex).**
- For K_n, No. of different **Hamiltonian cycles** = $[(n-1)!] / 2$
- For ($K_{m,n}$: m = n) : No. of different **Hamiltonian cycles = $n!(n-1)! / 2$**



$\delta(G) \ge n/2$ → Hamiltonian Graph

Coloring

- Paint
- Colour different to all adjacent vertices.
- Use **minimum** number of colours.
- **Theorem :** Sum of size of minimum vertex cover and size of maximum independent set is equal to number of vertices .
- **Proper Coloring :**
  - Paint
  - Adjacent vertices should not have the same colour.
- **Chromatic Number of Graph ($X(G)$)** → Min colours for proper colouring
  - **Null graph**: $X(G) = 1$
  - **Tree**: $X(T) = 2$, $n \ge 1$
  - **Cycle(C_n):**
    - $X(C_n) = 2$ if n is even
    - $X(C_n) = 3$ if n is odd
  - **Wheel Graph(W_n):**
    - $X(W_n) = 3$ if n is odd
    - $X(W_n) = 4$ if n is even
  - **Any Bipartite Graph($K_{m,n}$):**
    - $X(G) = 2$ (and vice versa)
  - $X(K_{m,n}) = 2$
  - $X(Q_n) = 2$
- **Every even-length cycle** is always **2-chromatic**

A. Chromatic Partitions -

- Independent Set: Set of non-adjacent vertices
- Maximal Independent Set (MIS): An independent set where no more vertex can be added.

(Property not about size)

- **Independence No. ($\beta(G)$): Number of vertices in largest MIS**

  $\beta(G) \ge n / X(G)$

B. Dominance Set :

- Set of vertices $D \subseteq V$, such that either a vertex or its adjacent vertex belongs to D.
- **Minimal Dominating Set (MDS):** No more vertex can be removed.
- **Domination Number ($\alpha(G)$): Number of vertices present in smallest MDS.**

Hierarchy:

- Dominance No. → Smallest → Minimal
- Index / No. → Largest → Maximal

Independence Edge Set (Matching Set) → Set of non-adjacent edges.

- **Maximal Matching Set** : No more edges can be added.
- **Matching Number :** No. of edges in largest MMS.
  - $M(C_n)=M(W_n)=M(K_n)=\lfloor n/2 \rfloor$
  - $M(K_{m,n})=\min(m,n)$

Perfect Matching: Every vertex in the graph is connected to exactly one edge.(1-factor or Complete Matching)

- $K_{n,n} = n!$
- $K_{2n} = (2n)! / 2^n * n!$

Perfect Matching exists → Only if **Number of vertices is even**.

Planarity : Drawing graph on a plane without intersecting its edges.

- K_5 → Non-planar (**Kuratowski's 1st graph**) - 1st graph (non-planar with min vertices)
- $K_{3,3}$ → Non-planar (**Kuratowski's 2nd graph**) → 2nd non planar graph with min no. of edges
- **Embedding:** Drawing a graph in the plane. (K_5-e) and $(K_{3,3}-e)$ → **Planar**

Drawing Planar Graph Introduces Faces / Regions

- **Types**:
  - Bounded / Finite / Closed
  - Unbounded / Open / Infinite
- **Euler's Formula:**
  - $n-e+f=2$ (for k=1)
  - $n-e+f=k+1$ (for $k\geq2$)

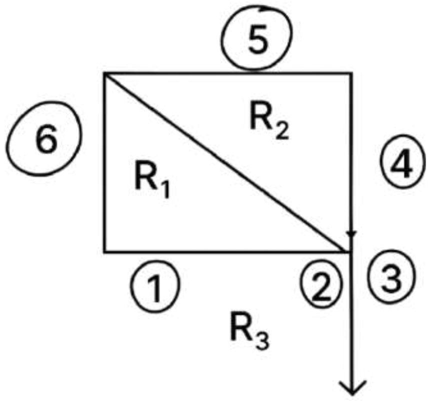
where, n → number of vertices k = components
e → number of edges
f → number of faces

- **Kuratowski's Theorem :** A graph is nonplanar if and only if it contains s subgraph that is homomorphic to either K_5 or $K_{3,3}$.

Degree of Region Degree of Region = Number of edges involved

→ Every **edge** introduces 2 degrees
deg(R1)+deg(R2)+deg(R3)=12 → 2*e⇒**∑deg(Ri)=2*e**

Example:



- deg(R1) = 3
- deg(R2) = 3
- deg(R3) = 6

In planar graphs, deg (Region)≥3 (Region is made by at least 3 edges)
If we have **3 regions** and at least 3 edges:
d(R1)≥3
d(R2)≥3
d(R3)≥3
So, ∑d(Ri)=3×3⇒**2e= 3f**
2e=3(2+e−n)
or, 3n−6≥e
E≤3n−6
Let's ,5 regions:
d(Ri)≥5*3
here, 5 is f

- If graph is **planar**, then **e≤3n−6**(Vice-versa not true)
  - $\delta(G)\leq 2e/n \leq \Delta(G) \leq n-1$
  - $\delta(G)\leq 2e/n$
  - $\delta(G)\leq(6n-12)/n$
  - **$\delta(G) \leq 5$**

If G is planar → $\delta(G)\leq5$
Or $\delta(G)\leq5$ then G may be planar

Adjacency Matrix (m, m): Primary matrix A of size m×m

- 1if there is a **connection** and otherwise 0.
- $X_{ij}=1$ indicates a **self-loop.**

Degree of a Vertex : Number of 1s in its corresponding **rows or column.**
(No self-loop and no multiple edges)

Note :

- In A^2, the **principal diagonal elements** represent the **degree** of the corresponding vertex.
- **$A^k = [x_{ij}]$,** x_{ij} = number of different paths from vertex i to vertex j of length k.
- **Trace Property :** $\frac{1}{6}\,Tr(A^3)$=Number of triangles in G

For adjacency matrix, $\lambda_1,\lambda_2,\ldots,\lambda_n \to$ Eigen Value

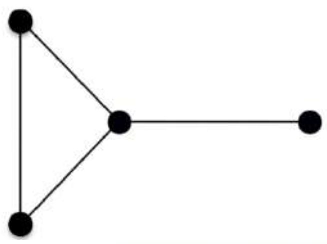
a) $\lambda_1+\lambda_2+\cdots+\lambda_n=0$

b) $\lambda_1^2+\lambda_2^2+\cdots+\lambda_n^2=2|E(x)|$

c) $\lambda_1^3+\lambda_2^3+\cdots+\lambda_n^3=6\cdot Tr(x)$

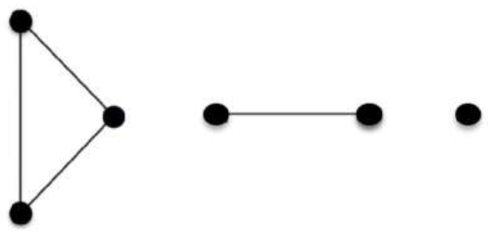
Note: In an **undirected finite graph**, at least **2 vertices** must have the **same degree**.

Graph Notation: $G=(V,E)$



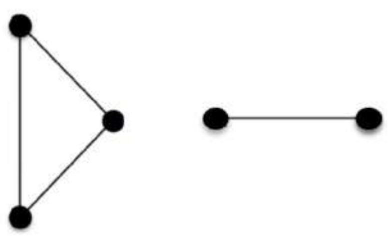
Subgraph: $G_1=(V_1,E_1)$

where: $V1 \subseteq V$, $E1 \subseteq E$



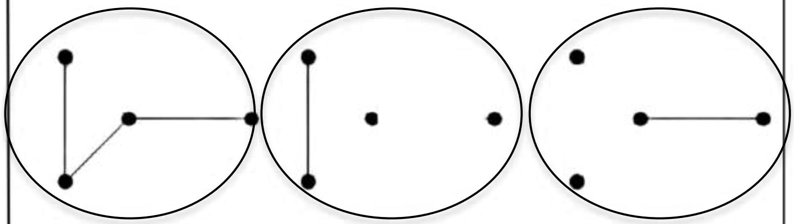
Clique: A **complete subgraph**

- **Clique Number** ($\omega(G)$) = **largest clique**

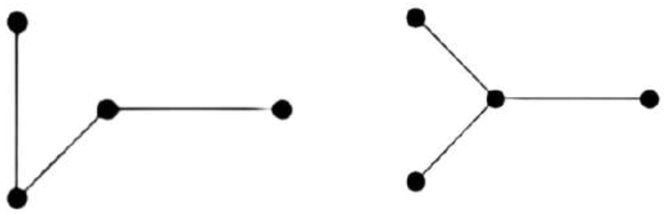


Spanning Subgraph: A subgraph $G_2=(V_2,E_2)$

where: $V_2=V$



Spanning tree: A **minimally connected spanning subgraph** (Contains **no cycles**).



Propositional Logic

- **Fact** → Propositional statement
  - ✓ Yes
  - ✗ No
- **Types:**
  - ○ Simple
  - ○ Compound
- **Connectives –**
  - **1) Conjunction (∧) :** AND → Result is False if any operand is **false.**
  - **2) Disjunction (∨) :** OR → True if at least one operand is **true**
  - **3) Single Implication / Conditional (→)**
- **p → q:**
  - ○ If p then q / If p, q / p implies q / q unless p / q whenever p.
  - ○ p ⟹ q
    T → F is the only condition that gives False
    p → q ≡ ¬p ∨ q
- **Variants:** p → q
  - ○ **Inverse:** ¬p → ¬q
  - ○ **Converse:** q → p
  - ○ **Contrapositive:** p → q ≡ ¬q → ¬p

4) Double Implication / Biconditional (↔) : "if and only if"/ "iff"

- p ↔ q is true if both p and q are either T or F (T ↔ T = T || F ↔ F = T)

Properties: These 3 are equivalent if and only if

- A ≡ B
- A ↔ B = tautology
- A, B are having same truth behavior

Truth Table Outcomes:

- If all are true → **Tautology / Valid**
- If all are false → **Contradiction**
- If at least one operand is true → **Satisfiable**
- If mix of true and false → **Contingency**

Types of Logical Equivalence:

Type-1: Try to make the statement to false → if derived → Truth/valid

Type 2: Laws in Propositional Logic

De Morgan's Law

- ¬(p ∧ q) ≡ (¬p ∨¬q)
- ¬(p ∨ q) ≡ (¬p ∧¬q)

Some other laws :

- **Commutative Laws**
  - ○ p ∨ q ≡ q ∨ p
  - ○ p ∧ q ≡ q ∧ p
- **Associative Laws**
  - ○ p ∨ (q ∨ r) ≡ (p ∨ q) ∨ r
  - ○ p ∧ (q ∧ r) ≡ (p ∧ q) ∧ r
- **Distributive Laws**
  - ○ p ∨ (q ∧ r) ≡ (p ∨ q) ∧ (p ∨ r)
  - ○ p ∧ (q ∨ r) ≡ (p ∧ q) ∨ (p ∧ r)
- **Identity Laws**
  - ○ p ∨ F ≡ p
  - ○ p ∧ T ≡ p
- **Domination Laws**
  - ○ p ∨ T ≡ T
  - ○ p ∧ F ≡ F
- **Idempotent Laws**
  - ○ p ∨ p ≡ p
  - ○ p ∧ p ≡ p
- **Inverse Laws**
  - ○ p ∨ ¬p ≡ T
  - ○ p ∧ ¬p ≡ F
- **Absorption Laws**
  - ○ p ∨ (p ∧ q) ≡ p
  - ○ p ∧ (p ∨ q) ≡ p

Truth Tables for propositional statements

Truth Tables for propositional statements

p	q	¬p	¬q	p ∨ q	p ∧ q	p → q	q → p	p → q	¬p → q	p ∧ z
T	T	F	F	T	T	T	T	T	T	F
T	F	F	T	T	F	F	T	F	T	T
F	T	T	F	T	F	T	F	F	T	F
F	F	T	T	F	F	T	T	T	F	F

- **Some conversions.**
- p → q ≅ ¬p ∨ q ≅ ¬q → ¬p
- q→p≅¬q∨p≅¬p→¬(¬q)
- p ↔ q ≅ (p → q) ∧ (q ⟶ p) ≅ ¬p ↔ ¬q
- ¬p ⟶ q ≅ ¬(¬p) ∨ q ≅ ¬q ⟶ ¬(¬p)

Type 3 (Inference Rule) [Premises → Conclusion]

Rule of Inference	Tautology	Name
$p \to q$ p ———— $\therefore q$	$(p \land (p \to q)) \to q$	Modus Ponens (Rule of Detachment)
$p \to q$ $\neg q$ ———— $\therefore \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus Tollens
$p \to q$ $q \to r$ ———— $\therefore p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical Syllogism (Law of the Syllogism)
$p \lor q$ $\neg p$ ———— $\therefore q$	$(\neg p \land (p \lor q)) \to q$	Disjunctive Syllogism
P ——— $\therefore (p \lor q)$	$p \to (p \lor q)$	Addition
$p \land q$ / ———— / $\therefore p$ or $p \land q$ / ——— / $\therefore q$	$(p \land q) \to p$ or $(p \land q) \to q$	Simplification
$p \lor q$ $\neg p \lor r$ ———— $\therefore q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution
$(p \land q) \to r$ —————— $\therefore p \to (q \to r)$	$[(p \land q) \to r] \to [p \to (q \to r)]$	Exportation

Predicate Logic :

- Predicates are expressions in logic that represent properties, conditions, or relationships involving one or more variables.
  Example : $P(x) \land Q(x)$
- **Open Statement** → Can't define truth value directly; it depends on input value(s).
- **Domain / Universe / Domain of discourse** - Set of allowed possible values for open statements.
- **Subject + predicate → property**
- **Predicate Variable:** $P(x) : x^2 \leq 100$ → where x → Predicate Variable.
  - P: "2 is an even number" → where P → Propositional logic.
- **Quantifiers** : Tool used to define the truth value in terms of **quantity**.
- **Types:**
  - **Universal Quantifier**: ∀ (for all)
  - **Existential Quantifier**: ∃ (there exists)

Example :
Universal: $\forall x\, P(x) \to$ Every element in domain is true → AND operator
Existential: $\exists x\, P(x) \to$ At least one value is true → OR operator

Negation of Quantifiers:

- $\sim(\forall x\, P(x)) = \exists x(\sim P(x))$
- $\sim(\exists x\, P(x)) = \forall x(\sim P(x))$

Implication (with quantifier): $\forall x\, P(x) \to \exists x\, P(x)$

Type 4: Box Method

- **∃x→ use '∨'**
- **∀x→ use '∧'**

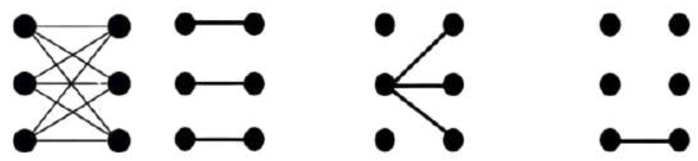
1. $\forall x[P(x) \land Q(x)] \equiv \forall x\, P(x) \land \forall x\, Q(x)$
2. $\forall x P(x) \lor \forall x Q(x) \to \forall x\, [P(x) \lor Q(x)]$
3. $\exists x[P(x) \lor Q(x)] \equiv \exists x\, P(x) \lor \exists x\, Q(x)$
4. $\exists x[P(x) \land Q(x)] \to \exists x\, P(x) \land \exists x\, Q(x)$
5. $\forall x[P(x) \to Q(x)] \to \forall x\, P(x) \to \forall x\, Q(x)$
6. $\forall x[P(x) \leftrightarrow Q(x)] \to \forall x\, P(x) \leftrightarrow \forall x\, Q(x)$
7. $\forall x[P(x) \to A] \to \forall x\, P(x) \to A$

Predicate Logic
Example:

"All mothers are female."
All of x . x is mother . x is female.
∀x   mT(x)
F(x)

Type 5 :

∀x∀y **∀x∃y** **∃y∀x** **∃x ∃y**



(Minimum condition to be true is indicated)

$$\begin{tikzcd}
\forall x\forall y \arrow[rr, leftrightarrow] \arrow[d] & & \forall y\forall x \arrow[d] \\
\exists y\forall x \arrow[d] & & \exists x\forall y \arrow[d] \\
\forall x\exists y \arrow[d] & & \forall y\exists x \arrow[d] \\
\exists y\exists x \arrow[rr, leftrightarrow] & & \exists x\exists y
\end{tikzcd}$$

Rule of Inference	Name
$\dfrac{\forall x P(x)}{\therefore P(c)}$	Universal instantiation
$(P(c)$ for an arbitary $c)$ / $(\therefore \forall x P(x))$	Universal generalization
$(\exists x P(x))$ / $(\therefore$ for some element $c)$	Existential instantiation
$(P(c)$ for some elemet $c)$/ $(\therefore \exists x P(x))$	Existential generalization

- **Quantifiers with Inference Rules :**

Universal Quantifier Rules:

1. ∀x P(x)→P(a)
2. P(a)→∀x P(x) → **Generalization**

Existential Quantifier Rules:

3. ∃x P(x)→P(a) → **Specification**
4. P(a)→∃x P(x)

SET THEORY

SET

- **Set:** A well-defined collection of distinct objects.
  Example: A = {1, 2, 3}
- **Forms:**
- ○ **Roster: A = {1, 2, 3, 4}**
- ○ **Set Builder:** A = {x | x ∈ N, x ≤ 4}
- **Important Types:**
- ○ **Singleton Set:** One element
- ○ **Empty/Null Set:** No elements (∅)
- ○ **Cardinality:** No. of elements (|A|)
- **Key Properties:**
- ○ **Order & repetition don't matter**: {1,2} = {2,1} = {1,2,2}
- ○ a ∈ A → a is an element of set A
- ○ **Equal Sets**: Same elements ⇒ A = B
- ○ **Subset**: A ⊆ B (all elements of A in B)
- ○ **Proper Subset**: A ⊂ B and A ≠ B
- ○ **Power Set**: P(A) = all subsets of A, $|P(A)| = 2^n$
- **Equal Sets:** Two sets are equal if they have exactly the same elements, regardless of order.
- ○ **Example:** A = {1, 2, 3}, B = {3, 2, 1} ⇒ A = B.
- **Subset(A ⊆ B):** Every element of A is also in B.
- ○ **Example:** A = {1, 2}, B = {1, 2, 3} ⇒ A ⊆ B.
- **Proper Subset (A ⊂ B):** A ⊆ B and A ≠ B.
- ○ **Example:** A = {1, 2}, B = {1, 2, 3} ⇒ A ⊂ B.
- **Improper Subset:** Every set is an improper subset of itself.
- ○ **A ⊆ A but A ⊄ A**
- **Power Set P(A):** Set of all subsets of A.
- ○ **Example:** A = {1, 2} ⇒ P(A) = {∅, {1}, {2}, {1, 2}}
- **Power Set Size:** $|P(A)| = 2^n$ (n = no. of elements in A)
- **Cardinality (|A|):** Number of elements in a set.
- ○ **Example:** A = {5, 6, 7} ⇒ |A| = 3.
- **Cartesian Product:** A × B = Set of all ordered pairs (a, b).
- ○ If |A| = m, |B| = n ⇒ |A × B| = m × n

Set Operations

- **Union (A ∪ B):** All elements in A or B.
- ○ A ∪ B = {x | x ∈ A or x ∈ B}
- ○ n(A ∪ B) = n(A) + n(B) – n(A ∩ B)
- **Intersection (A ∩ B):** Common elements. A ∩ B = {x | x ∈ A and x ∈ B}
- **Difference (A − B):** In A, not in B [Note: A − B ≠ B − A (not symmetric)]. A − B = {x | x ∈ A and x ∉ B}
- **Complement (A'):** Elements in U not in A. A' = {x ∈ U | x ∉ A}
- **Symmetric Difference (A ⊕ B):** In A or B, not both.A ⊕ B = (A − B) ∪ (B − A)
- **Multisets**:
- ○ Allows Repetition: Example: {1, 2, 2, 3, 3, 3}
- ○ Operations:
- ■ Union: $\max(m_1(x), m_2(x))$
- ■ Intersection: $\min(m_1(x), m_2(x))$
- ■ Addition: $m_1(x) + m_2(x)$

Relation :

A set of ordered pairs is a relation.

- If |A|=n, then **number of subsets** = 2^n
- If |A|=m, |B|=n, then **number of subsets** = 2^{mn}
- ○ Total number of **relations** = total number of **subsets** of A×B = 2^{mn}
- If |A| = m, |A x A| = m^2, total number of **relations** = 2^{m^2}
- **Inverse relation**: For a relation R the inverse of the relation is given by R^{-1}.
- ○ R^{-1} = { (y, x) (x, y) ∈ R }

• Properties of Relations

- **Reflexive**: ∀a∈A, (a,a) ∈ R {Diagonal of matrix must be present}
- **Symmetric**: ∀a ∀b, (a,b) ∈ R → (b,a)∈R
- **Irreflexive**: ∀a∈A, (a,a)∉R {Diagonal of matrix must be absent}
  Irreflexive is not opposite to Reflexive
- **Transitive**: ∀a,∀b,∀c∈A, {(a,b), (b,c)∈R → (a,c)∈R}
  (Use directed graph: if R = R^{-1} → transitive)
- **Anti-symmetric**: ∀a,∀b∈A, {(a,b)∈R∧(b,a)∈R → a=b}
- **Asymmetric**: ∀a,∀b∈A, {(a,b)∈R → (b,a)∉R} for all a ≠ b
- A relation is a partial order if (**Reflexive + Antisymmetric + Transitive**)
- A relation is an equivalence if (**Reflexive + symmetric +Transitive)**

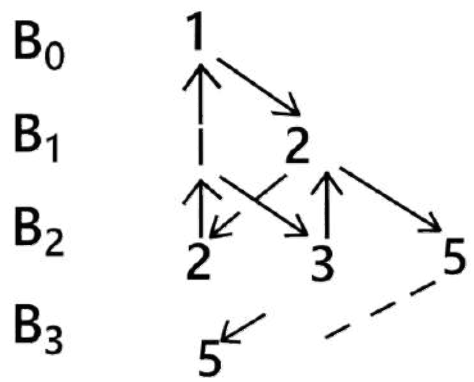
Type of Relation	No. of Relatives possible	Union	Intersection
Reflexive	2^{n^2-n}	✓	✓
Irreflexive	2^{n^2-n}	✓	✓
Symmetric	$2^n . 2^{n(n-1)/2}$	✓	✓
Anti-symmetric	$2^n . 3^{n(n-1)/2}$	×	✓
Asymmetric	$3^{n(n-1)/2}$	×	✓
Transitive	-	×	✓

- No. of relation : **Reflexive + Irreflexive** = 0
- No. of relation : **Irreflexive + Symmetric** = $2^{n(n-1)/2}$
- No. of relation : **Reflexive + Symmetric** = $2^{n(n-1)/2}$

Type of Relation	Flipping	Same Element
Symmetric	✓	✓
Asymmetric	×	×
Anti-Symmetric	×	✓

- If R is Reflexive & Symmetric then R is **not Transitive.**
- If R_1, R_2 is Anti-symmetric and Asymmetric
  - then $R_1 \cap R_2$ are also Anti-symmetric and Asymmetric.
  - but $R_1 \cup R_2$ are not Anti-symmetric and Asymmetric.
- **Equivalence Relations & Partial Order:** Instead of z × z, we also can take $z^2 \times z^2$ (ordered pairs).
- **Equivalence Relation:**
  An equivalence relation = Reflexive + Symmetric + Transitive
  - If R is an equivalence relation (EQR), then it creates a **partition** (on equal classes) on a set.
  Example:
  - a+b is even:
- **Odd + Odd = Even** → 1st Partition
- **Even + Even = Even** → 2nd Partition
- Odd + Even = Odd
  - R_2={(a,b)|a≡b (mod 4)}⇒4 partitions
- **Bell Number (B_n):** Total number of different partitions of a set of size n.

Total number of equivalence relations (EQR)



- **Partial Order Relation (POR):** Reflexive + Anti-symmetric + Transitive
  - Some elements are **comparable**
  - If **all elements are comparable**, then:
  - It becomes a **Total Order Relation (TOR) or Linear Order** or **Chain Order**
  - Let R → RAT → POR
- If all elements are R→ RAT→TOR (Comparable) then:
  - **YES** ⇒ Linear Order
  - **NO** ⇒ Only Partial Order
  Where, R → Relation
  RAT → **R**eflexive, **A**ntisymmetric, **T**ransitive
  TOR → **Total Order Relation**
- If the **Relation** is **RAT** (Reflexive, Antisymmetric, Transitive),then the structure becomes a **Partial**

Order Set *(Poset).*

- **Hasse Diagram :** A **Hasse diagram** is a simplified graphical representation of a **partially ordered set (poset)**.
- **Steps to Construct a Hasse Diagram:**

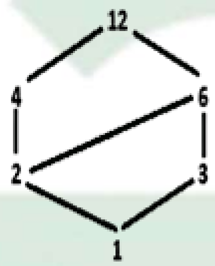
1. **Vertices**:
   - Create one vertex for each element in the set AA.
2. **Edges**:
   - Draw an edge between two elements x and y **only if** x covers y (or vice versa).
   - **Do not** include:
     - **Self-loops** (reflexive pairs).
     - **Edges implied by transitivity**.

- **Covering Relation (in Poset) :** Let (A,≤) be a **poset**.
  - An element y covers x if: x<y and no z∈ A such that x<z<y

- The set of all such pairs (x,y) forms the **covering relation.**

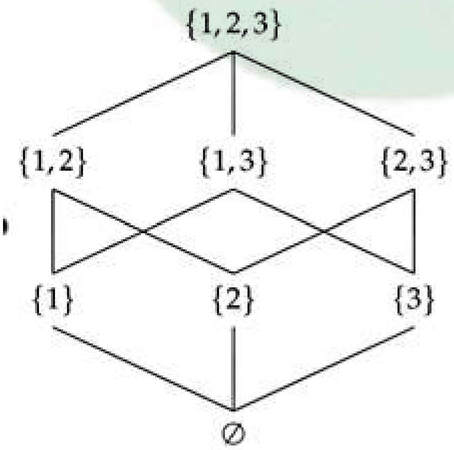
- **Key Points:**
- **Hasse Diagram** shows only covering relations.
- Reflexive + Transitive **Closure** of covering relation gives back the original poset.
- Hasse Diagram of a totally ordered set (**toset**) is a chain.

- **Hasse Diagram of Power Set[P(A), ⊆]**:Forms a **hypercube** of dimension n, where |A|=n.
- **(D_{12}, 1) - Hasse Diagram :** Remove self-loop and apply transitivity.
  For example, **(a→b)**
  where a → lower , b → higher



Hasse Diagram

- A = {1, 2, 3}, {(P(A)), ⊆ }



- Total edge = 12
  $e = n * 2^{n-1}$
  here n = 3,
  e = 12
- **Maximum (Maximal) Element:** An element is **maximal** if no other element is greater than it in the poset.
  It is **not necessarily** comparable to all other elements.
- **Greatest (Maximum) Element:** $a \le x$ for **all elements of A**, x∈A
  ⇒ x is **Greatest Element (G.E.)**
- If G.E. exists ⇒ it is **unique**
- **Minimal Element :** An element is **minimal** if no other element is smaller than it in the poset.
  Not **necessarily** the least.
- **Least (Minimum) Element:** x≤a for **all elements of A**, x,a∈A
  ⇒ x is **Least Element (L.E.)**
- If L.E. exists ⇒ it is **unique**
- **Key Points:**
- Every finite, non-empty lattice has at least one minimal and one maximal element.
- Greatest/Least elements, if they exist, are unique.
- If greatest/least elements exist, they are also the only maximal/minimal elements, respectively.

- **Bounds in Posets**
- **Upper Bound:** For a poset [A,R] and B ⊆ A: B
- Upper Bound of B={x∈A|∀b∈B,b≤x}
- x is Upper Bound of B
- **Least Upper Bound (LUB / Join / Supremum):** Least element among all upper bounds of a,b∈A. Denoted by a∨b.
- **Conditions:**
  - a≤a∨b,b≤a∨ba
  - ∀x ∈ Upper Bound of {a,b}⇒a∨b≤x
- If no such least exists, lub does not exist.
- **Lower Bound:** For a poset [A,R] and B ⊆ A:
- Lower Bound of B={x∈A|∀b∈B,x≤b}
- x is Lower Bound of B
- **Greatest Lower Bound (GLB / Meet / Infimum):** Greatest element among all lower bounds of a,b∈A. Denoted by a∧b.
- **Conditions**:
  - a∧b≤a,a∧b ≤ b
  - ∀d∈Lower Bound of {a,b}⇒d≤a∧b
- If no such element exists, glb does not exist.

Poset	lub(a, b)	glb(a, b)
$[D_n,\|]$	LCM(a, b)	GCD(a, b)
[P(A),⊆]	x∪y	x∩y

- If **G.E. = L.E.** for all elements, it's possible; but lub/glb are related to subsets, not the whole set, and they are unique if they exist.
- **Examples in Common Posets**
- **Join Semi-Lattice:** A poset where **every pair of elements** has a **least upper bound** (lub).
- **Meet Semi-Lattice:** A poset where **every pair of elements** has a **greatest lower bound** (glb).
- **(A, R) → Poset** – For every pair, if **lub** and **glb** exist → **Lattice → (A, R)**

LATTICE

- A **lattice** is a poset in which every pair of elements has both glb & lub.
  i.e., Lattice is both join semi lattice & meet semi lattice.
  A lattice L is denoted as $(L, \wedge, \vee)$
- A lattice L is denoted as $(L, \wedge, \vee)$

○ It is not needed that every lattice has greatest and least element.

○ Every finite lattice has least and greatest elements.

- **Properties of Lattice:**

○ $a\wedge a=a$	Idempotent
○ $a\vee a=a$	Idempotent
○ $a\vee b=b\vee a$	Commutative
○ $a\wedge b=b\wedge a$	Commutative
○ $a\vee(b\vee c)=(a\vee b)\vee c$	Associative
○ $a\wedge(b\wedge c)=(a\wedge b)\wedge c$	Associative
○ $a\vee(a\wedge b)=a$	Absorption law
○ $a\wedge(a\vee b)=a$	Absorption law

○ $a\le b \Leftrightarrow a\vee b=b\wedge a=a$

○ If $a\le b$ and $c\le d$ then:

■ $a\vee c\le b\vee d$

■ $a\wedge c\le b\wedge d$

- **Sublattice:** If A is a lattice, B is called sublattice of A if:

○ B itself is a lattice.

○ lub of any two elements of B is same as lub of the two elements in A.

A B

In the above figure B is a subset of A and B is a lattice still B is not a sublattice of A.
Because:

- In A, glb(2,3) = 4
- In B, glb(2,3) = 5

⇒Not equal ⇒ Not sublattice

- A **sublattice** of a complemented lattice **need not** be complemented.
- A **sublattice** of a bounded lattice **is** a bounded lattice.

TYPES OF LATTICE:

1. **Bounded Lattice:** A lattice with **greatest** and **least** element is called a **bounded lattice.**
   Every **finite lattice** is always bounded lattice.

○ G.E. = 1

○ L.E. = 0

2. **Complement Lattice:** A complemented lattice is a **bounded lattice** in which every element has **at least one complement.**

○ If $a + b = 1$ and $a \cdot b = 0$ then, a & b are complements of each other.
or

○ An element b is said to be the complement of a **iff**:

■ **glb(a, b) = least element (0)**

■ **lub(a, b) = greatest element (1)**

3. **Distributive Lattice:** A lattice is called **distributive** if the following laws hold:

○ $a\vee(b\wedge c)=(a\vee b)\wedge(a\vee c)$

○ $a\wedge(b\vee c)=(a\wedge b)\vee(a\wedge c)$

$(D_{12}, 1)$ → Bounded but not **complemented** but **distributive**

○ If L is a **bounded distributive lattice**, then the **complement (if it exists) is unique**. However, the **converse is not true**.

4. **Boolean Algebra:** A **Boolean algebra** is a lattice that is both **distributive** and **complemented**.

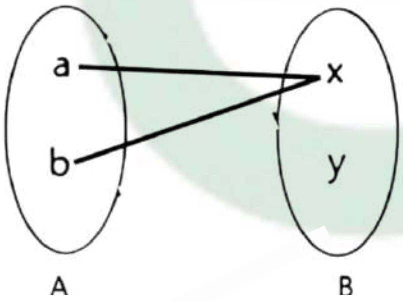
- In Boolean algebra, **every element has exactly one complement**.
  Any lattice containing diamond or pentagon → Not distributive (see notes)

Examples:

- $[D_n;1]$ is a **distributive lattice** for any n (since distributive properties hold for **LCM** and **GCD**).
- $[D_n;1]$ is a **Boolean lattice iff n** is a **square-free number**.
- $[P(S); \subseteq]$ is a **Boolean lattice**.
- Every Boolean lattice with 2^n elements is **isomorphic** to the lattice $(P(S);\subseteq)$, where S is a set with n elements.
- Such Boolean lattices form a **hypercube** Q_n.

FUNCTION/MAPPING/TRANSFORMATION/ASSIGNMENT

- A function f : A→B assigns **exactly** one element of B to each element of A.
  - A: Domain
- B: Co-domain



- x : image
  - a, b: preimage
- Range: Set of images of all the elements of A.
- Range ≠ Co-domain
  (Range need not to be equal to co-domain).

Note:

- **Not allowed:** mapping from a single element of X to **multiple** elements of Y.
- **Allowed**: multiple elements of X mapping to a **single** element of Y.
- **Total number of function** = $(n(B))^{n(A)}$

- If A and B are two sets with |A| = n,|B| = m then **number of functions possible** from A to B = m^n

Types of Functions:

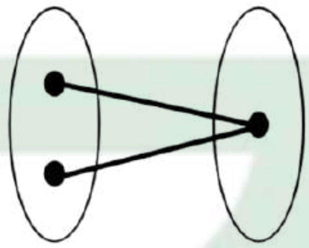
1. One-One (Injective)

Let f:A→B

If f(a) = f(b), then a = b
or
If a ≠ b, then f(a) ≠ f(b)

then it's one-one.

- It f : A → B is one–one then **|A| ≤ |B|.**
- If f|A| = m and f|B| = n.
  Then, **total number** of one-one functions (injective):$_mP_n$
- If m = n = k, then number of one : one function = k!
  - Note: Not one-one: Mapping two different elements of A to the same element in B



Not 1:1

2. Onto (Surjective): A function is **onto** if **Range = Co-domain.** (Right side must be fully covered)

- If f : A → B is onto then |A| ≥ |B|
  If |A|=m, |B|=n, then
  Number of onto function = $\sum_{i=0}^{n}$ $(-1)^i\ {}^nC_i(n-i)^m$

- If m = n = k, then number of one : one function = k!
- **Inclusion-Exclusion Principle** → Works for both **m→different, n→identical** cases
  S(m, n) = 1/(n!) $\sum_{i=0}^{n}$ $(-1)^i\ {}^nC_i(n-i)^m$ = onto/(n!)

3. One-One Correspondence:

A function $f:A \to B$ is called a bijective function if and only if:

- f is one-one (injective), and
- f is onto (surjective).

Therefore, **Bijection = Injection + Surjection**

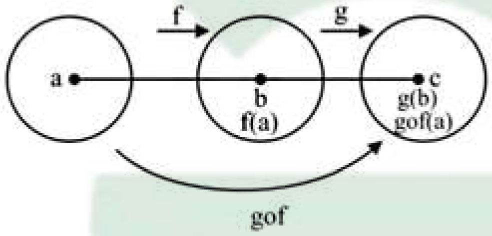
- If a function is bijective then the $n(A) = n(B)$
- If $|A|=|B|$, then **total number** of bijective functions $= n!$
- If $f : A \to A$ is a function and A is a finite set then A is one–one $\Leftrightarrow$ A is onto

4. Composition of Functions:

Let $f:A\to B$, $g:B\to C$, Then $g\circ f:A\to C$

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

- A is the **domain** of $g\circ f$
- C is the **codomain** of $g\circ f$



Properties:

i) If f and g are one:one → Then $g\circ f$ is one:one
ii) If f and g are onto → Then $g\circ f$ is onto
iii) If $g\circ f$ is one:one → Then f is one:one
iv) If $g\circ f$ is one:one → It does **not** mean g is one:one.
v) If $g\circ f$ is onto → Then g is onto
vi) If f & g are bijection → Then $g\circ f$ is bijection

Inverse of a Function : A function $f : A\to B$ is **invertible** if its inverse relation f^{-1} is also a function from $B\to A$.

A function is invertible if and only if it is a bijection.

- If a function is **not a bijection**, its inverse does **not exist** as a function.
- However, we can still find the **inverse image** of a subset of the codomain.

Let $f : A\to B$ be a function and $S\subseteq B$. Then,

$f^{-1}(S)=\{a\in A \mid f(a)\in S\}$

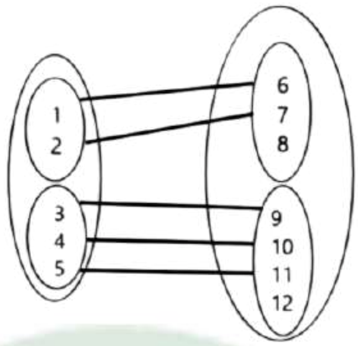
- If **$f:A\to B$ and $S,T\subseteq A$:**

○ $f(S\cup T)=f(S)\cup f(T)$
○ $f(S\cap T)\subseteq f(S)\cap f(T)$
○ $f(S\cap T)=f(S)\cap f(T) \Rightarrow$ **only if f is bijective**

- If **$f:A\to B$ and $S,T\subseteq A$:**

○ $f^{-1}(S\cup T)=f^{-1}(S)\cup f^{-1}(T)$
○ $f^{-1}(S\cap T)=f^{-1}(S)\cap f^{-1}(T)$
○ $f^{-1}(\bar{S})=\neg(f^{-1}(S))$

Inverse of Elements:



- $f^{-1}\{6, 7, 8\} = \{1, 2\}$
- $f^{-1}\{9, 10, 11, 12\} = \{3, 4, 5\}$

- **Identity Function:** A **mapping** $I_A:A\to A$ is called an **identity function** if:

○ $I_A=\{(x,x) \mid x\in A\}$

- **Constant Function:** A function $f:A\to B$ is called a **constant function** if:

○ $f(x)=c\ \forall x\in A$

That is, every element of domain A maps to the same single constant $c\in B$.

- **Partial Function:** A function $f:A\to B$ is called **partial** if it is defined only for **some elements** of A.
 The set where f is defined is called its **domain of definition**.

GROUP THEORY

- **Binary operation:** A binary operation * on a non-empty set A is defined if:
  $\forall a,b\in A,\ (a*b)\in A$
  i.e., the result of the operation on any two elements of A must also lie in A (Closure property).
- **Semigroup:** A set with a binary operation (*)\ is a **semigroup** if:
  (i) Closure
  (ii) Associativity
- **Monoid:** A semigroup(A, *) is a **monoid** if:
  (i) Closure
  (ii) Associativity
  (iii) Identity

- **Group:** A monoid(A, *) is a **group** if:
  (i) Closure
  (ii) Associativity
  (iii) Identity
  (iv) Inverse
- **Additional Notes:**
  - Every finite group has a well-defined order = number of elements.
  - If $|G| = 2$, then $a^{-1} = a$ for all $a \in G$.
  - $(0, 1, \ldots, m-1, \oplus_m)$ is a group
  - $S_n = \{k < n \mid \gcd(k, n) = 1\}$, then $(S_n, \oplus_n)$ is a group.
- **Abelian Group:** A group is **Abelian** if *,is commutative: $a*b = b*a$
- **Properties of Groups:**
  - Left Cancellation: $ab = ac \Rightarrow b = c$
  - Right Cancellation:$ba = ca \Rightarrow b = c$
  - $(ab)^{-1} = b^{-1} * a^{-1}$
  - If G is Abelian $\Leftrightarrow (ab)^{-1} = b^{-1} * a^{-1}$
  - Identity: $a^0 = e$
- **Subgroup:** H will be a **subgroup** of G if:
  - $H \subseteq G$
  - H should be a group.
  - Identity element must be in every subgroup.
  - Lagrange's Theorem: The order (number of elements) of any subgroup of a finite group must divide the order of the group. (converse holds only in Abelian groups)
  - If are subgroups, then is a subgroup, need not be
  - If and finite, closure alone is sufficient to prove subgroup
  - Every group of composite order has a non-trivial subgroup

  Theorem: If H is a subgroup of G, then |G|/|H|.
- **Cyclic Groups:**
  - G is cyclic if $\exists a \in G$ such that every element in G is a^n for some $n \in Z$.
  - a is called a generator.
  - a^{-1} is also a generator.
  - All subgroups of cyclic group are cyclic.
  - No. of generators of cyclic group of order n :$\Phi(n)$
  - Order of $a^m = n / (\gcd(n, m))$
  - Number of subgroups = Number of divisors of |G|.
  - Subgroup of order is generated by $a^{n/d}$.
- **Cyclic Subgroup:** For $a \in$ **G** , the smallest subgroup containing a: $\langle a \rangle = \{a^{n>} \mid n \in Z\}$
  - <a> is a cyclic subgroup with order equal to order of a
  - If H is subgroup and contains , then $\langle a \rangle \subseteq H$
  - Every $a \in$ **G** forms a cyclic subgroup $\langle a \rangle \Rightarrow$ used to define generating sets.
- **Homomorphism:** Let $(G,*)$ and $(G',\oplus)$ be two groups.
  - A function $f: G \to G'$ is a **homomorphism** if:
    - $f(a*b)=f(a)\oplus f(b),\ \forall\ a,b \in G$
- **Isomorphism:** If $f: G \to G'$ is a **Objective homomorphism**, then:
  - f is called an **isomorphism**
  - Denoted as $G \cong G'$

- **Key Properties**
  - **f**$(e_1)=e_2$, where e_1,e_2 are identity elements of G and G′
  - $f(a^{-1})=(f(a))^{-1},\ \forall\ a \in G$
- **Order of an Element:** Order of $a \in$ **G** : smallest $n>0$ such that $a^n = e$
  - $\text{order}(a) = \text{order}(a^{-1})$
  - If $a^{-1} = b$, then $a^{-n} = b^n$.
- **Notes:**
  - Every group of prime order is cyclic, and every element except identity is a generator.
  - Every cyclic group is abelian.
  - A group is cyclic $\Leftrightarrow$ it cannot be expressed as a union of two proper subgroups.
  - **Direct product:** If $(G, *),(H, \oplus)$, then$(G \times H, .)$ where $(g_1,h_1).(g_2, h_2) = (g_1*g_2, h_1 \oplus h_2)$.
  - Every group of order≤ 5 is abelian.
  - Unique group exists for orders 1, 2, and 3.
  - **Exponential:** By taking the exponential of any element, if it generates all the elements, then it is a **cyclic group**.
  - Subgroup of a cyclic group is also **cyclic**.

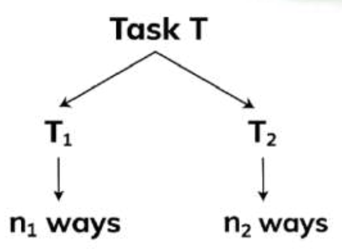
DISCRETE MATHEMATICS

GATE फर्रे

COMBINATORICS

- **Inclusion – Exclusion Principle**
  - $A1 \cup A2 \cup A3 \cup \cdots \cup A_n = \sum A_i - \sum A_i \cap A_j + \sum A_i \cap A_j \cap A_k - \cdots + (-1)^{r+1} \sum A_{i1} \cap A_{i2} \cap \cdots \cap A_{ir}$.
  - **Use cases**: Counting, no. divisible by given numbers (like 8, etc.)
- **Derangements(Dn):** No element appears in its original position.
  $D_n = n![1/(2!) - 1/(3!) + 1/(4!) + \ldots\ldots\ldots + (-1)^n 1/(n!)]$
- **Pigeonhole Concept:**
  If you have n holes and (n+1) pigeons,
  → then at least one hole contains more than one (actually ≥ 2) pigeon.
- **Extended Pigeonhole Principle :** If N objects are distributed into k boxes, then at least one box contains at least $\left\lceil \frac{N}{k} \right\rceil$ objects.

- **Euler's ϕ-Function (ϕ(n)):**
  - **ϕ(n)** = Total number of **relative prime numbers** which are **less than n**.
  - If **gcd(a, b) = 1**, then a and b are **relatively prime** /coprime.
    Example: $\phi(12) = \phi(3) \times \phi(4)$
    $= 2 \times 2 = \mathbf{4}$
  - $\phi(p) = \mathbf{p - 1}$
- **Formula for Euler's Totient Function:**
  - If $n = p_1 \times p_2 \times p_3 \ldots$ (prime factorized),
    Then: $\phi(n) = n(1-1/p_1)(1-1/p_2)(1-1/p_3)\ldots\phi(n)$
- **Product Rule / Sum Rule:**



- **Product Rule:**
  - Task **T** is divided into $\mathbf{T_1}$ and $\mathbf{T_2}$.
  - If $\mathbf{T_1}$ can be done in n_1 ways and for each of those $\mathbf{T_2}$ can be done in n_2 ways:
  - **Total** $= \mathbf{n_1 \times n_2}$
- **Sum Rule:**
  - If $\mathbf{T_1}$ and $\mathbf{T_2}$ are **independent** (don't occur simultaneously):
  - **Total** $= \mathbf{n_1 + n_2}$.

- **Permutation(Order Mattters):**
  - Without repetition: $P(n, r) = n! / (n - r)!$
  - With repetition: n^r
- **Combination(Order Doesn't Matter):**
  - Without repetition: $C(n, r) = n! / [r!(n - r)!]$
  - With repetition: $C(n + r - 1, r) = (n + r - 1)! / [r!(n - 1)!]$
- **Generating Function**

Binomial theorem: The **Binomial theorem** provides a formula for expanding expressions of the form:$(a+b)^n$

General Formula: $\mathbf{(x + y)^n = \sum_{k=0}^{n} {}^nC_k\, x^{(n-k)} y^k}$

where: $\mathbf{{}^nC_k = n!/[r!*(n-r)!]}$ **is binomial coefficient.**

n is a non-negative integer.

the expansion has **(n + 1)** terms.

- **Pascal's Triangle:** Each row gives the coefficients for $(a+b)^n$:

  n = 0: 1
  n = 1: 1 1
  n = 2: 1 2 1 $(a+b)^2$
  n = 3: 1 3 3 1 $(a+b)^3$
  n = 4: 1 4 6 4 1 $(a+b)^4$

- $\mathbf{{}^nC_r + {}^nC_{r+1} = {}^{n+1}C_{r+1}}$
- $\mathbf{{}^nC_k = n! / n! * (n-k)!}$
- $\mathbf{{}^{-n}C_k = (-1)^k * {}^{n+k-1}C_k}$

$<a_0, a_1, a_2, \ldots\ldots\ldots\ldots>$

$G(x) = a_0x^0 + a_1x^1 + a_2x^2 + \ldots\ldots\ldots\ldots$

$\mathbf{G(x) = \sum_{i=0}^{\infty} a_i * x^i}$ **or** $\mathbf{\sum_{n=0}^{\infty} a_n * x^n}$

Common Expansions

1. $(1 + x)^n = {}^nC_0\, x^0 + {}^nC_1x^1 + {}^nC_2x^2 + \ldots + {}^nC_nx^n$
2. $(1 + ax)^n = {}^nC_0\, (ax)^0 + {}^nC_1(ax)^1 + \ldots + {}^nC_n(ax)^n$
3. $(1 + x^m)^n = {}^nC_0x^0 + {}^nC_1x^m + {}^nC_2x^{2m} + \ldots + {}^nC_n\, x^{nm}$.

- **Special Infinite Series**

1. $1 / (1 - x) = 1 + x + x^2 + x^3 + \ldots = \Sigma\, x^i$
2. $1 / (1 - ax) = 1 + ax + a^2x^2 + a^3x^3 + \ldots$
3. $(1 - x^{n+1})/(1 - x) = 1 + x + x^2 + \ldots + x^n$
4. $1 / (1 + x)^n = \Sigma\, (-1)^{i} * {}^nC_i * x^i$

- **Recurrence Relation**
  - **Type-1:** $a_n = d * a_{n-1}$
    → This is a first-order linear recurrence.
    Initial condition: a_0
    Then, $\mathbf{a^n = d^n * a_0}$

○ **Type - 2: $a_n=a_{n-1}+a_{n-2}$** for $n\geq2$
Initial conditions: $a_1=1, a_2=1, a_0=0$;

- **Note:**

○ For **second-order** recurrence relations:
- **Roots**: R1,R2

○ If **R1≠R2** (distinct real roots):
$a_n=C_1R_1^n+C_2R_2^n$

○ If R1=R2=R(repeated root):
$a_n=C_1R^n+C_2nR^n$

○ For **third-order** recurrence relations:
- **Roots**: R1,R2,R3

○ If all distinct:
$a_n=C_1R_1^n+C_2R_2^n+C_3R_3^n$

○ If R1=R2≠R3:
$a_n=C_1R_1^n+C_2^nR_1^n+C_3R_3^n$

○ If R1=R2=R3=R:
$a_n=C_1R^n+C_2nR^n+C_3n^2R^n$

- **Classification** of Recurrence Relations:

○ **Homogeneous**: $f(n)=a_n-2a_{n-1}$
■ If f(n)=0, it's homogeneous.

○ **Non-Homogeneous:** $f(n)= a_n -2a_{n-1}$
■ If f(n)≠0, it's non-homogeneous.

○ **Characteristic Equation:** $a_n = a_n^{\#} + a_n^p$
where $a_n^{\#}$ is Homogeneous
a_n^p is Particular

- **Particular Solution (for Non-Homogeneous):**
Assume form based on f(n):

Type of f(n)	**Assume particular solution of the form**
Constant A	A
Linear An	An+B
Quadratic An^2	An^2+Bn+C
Exponential A^n	A^n

a) Consider the nonhomogeneous recurrence relation

$$a_{n+2} - 10a_{n+1} + 21a_n = f(n), n \geq 0$$

Here the homogeneous part of the solution is

$$a_n^{(h)} = c_1(3^n) + c_2(7^n),$$

$\mathbf{f(n)}$	$\mathbf{a_n^{(p)}}$
5	A_0
$3n^2 - 2$	$A_3n^2 + A_2n + A_1$
$7(11^n)$	$A_4(11^n)$
$31(r^n), r \neq 3,7$	$A_5(r^n)$
$6(3^n)$	A_6n3^n
$2(3^n) - 8(9^n)$	$A_7n3^n + A_8(9^n)$
$4(3^n) + 3(7^n)$	$A_9n3^n + A_{10}n7^n$

b) The homogeneous component of the solution for

$$a_n + 4a_{n-1} + 4a_{n-2} = f(n), n \geq 2,$$

is

$$a_n^{(h)} = c_1(-2)^n + c_2n(-2)^n,$$

where c_1, c_2 denote arbitrary constants. Consequently,

1. if $f(n) = 5(-2)^n$, then $a_n^{(p)} = An^2(-2)^n$;
2. if $f(n) = 7n(-2)^n$, then $a_n^{(p)} = n^2(-2)^n(A_1n + A_0)$; and
3. if $f(n) = -11n^2(-2)^n$, then $a_n^{(p)} = n^2(-2)^n(B_2n^2 + B_1n + B_0)$. (Here, the constants A, A_0, A_1, B_0, B_1, and B_2 are determined by substituting a_1 into the given nonhomogeneous recurrence relation.)

GeeksforGeeks

GATE CSE BATCH

KEY HIGHLIGHTS:

- 300+ HOURS OF RECORDED CONTENT
- 900+ HOURS OF LIVE CONTENT
- SKILL ASSESSMENT CONTESTS
- 6 MONTHS OF 24/7 ONE-ON-ONE AI DOUBT ASSISTANCE
- SUPPORTING NOTES/DOCUMENTATION AND DPPS FOR EVERY LECTURE

COURSE COVERAGE:

- ENGINEERING MATHEMATICS
- GENERAL APTITUDE
- DISCRETE MATHEMATICS
- DIGITAL LOGIC
- COMPUTER ORGANIZATION AND ARCHITECTURE
- C PROGRAMMING
- DATA STRUCTURES
- ALGORITHMS
- THEORY OF COMPUTATION
- COMPILER DESIGN
- OPERATING SYSTEM
- DATABASE MANAGEMENT SYSTEM
- COMPUTER NETWORKS

LEARNING BENEFIT:

- GUIDANCE FROM EXPERT MENTORS
- COMPREHENSIVE GATE SYLLABUS COVERAGE
- EXCLUSIVE ACCESS TO E-STUDY MATERIALS
- ONLINE DOUBT-SOLVING WITH AI
- QUIZZES, DPPS AND PREVIOUS YEAR QUESTIONS SOLUTIONS

ENROLL NOW

TO EXCEL IN GATE
AND ACHIEVE YOUR DREAM IIT OR PSU!

ENROLL NOW

STAR MENTOR CS/DA

KHALEEL SIR
ALGORITHM & OS
29 YEARS OF TEACHING EXPERIENCE

SATISH SIR
DISCRETE MATHEMATICS
BE in IT from MUMBAI UNIVERSITY

VIJAY SIR
DBMS & COA
M. TECH FROM NIT
14+ YEARS EXPERIENCE

SAKSHI MA'AM
ENGINEERING MATHEMATICS
IIT ROORKEE ALUMNUS

AVINASH SIR
APTITUDE
10+ YEARS OF TEACHING EXPERIENCE

CHANDAN SIR
DIGITAL LOGIC
GATE AIR 23 & 26 / EX-ISRO

MALLESHAM SIR
M.TECH FROM IIT BOMBAY
AIR – 114, 119, 210 in GATE
(CRACKED GATE 8 TIMES)
14+ YEARS EXPERIENCE

PARTH SIR
DA
IIIT BANGALORE ALUMNUS
FORMER ASSISTANT PROFESSOR

SHAILENDER SIR
C PROGRAMMING & DATA STRUCTURE
M.TECH in Computer Science
15+ YEARS EXPERIENCE

AJAY SIR
PH.D. IN COMPUTER SCIENCE
12+ YEARS EXPERIENCE