Unit :- 3

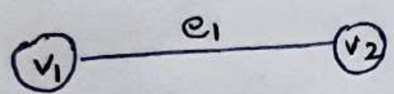
Graph Theory

Terminology Graph Representation, Graph Isomorphism, Connectedness, Various graph properties; Euler and Hamiltanion Graph, Shortest path Algorithms. Trees: Terminology, Tree Traversals, prefix codes, Spanning trees; Minimum Spanning Trees.

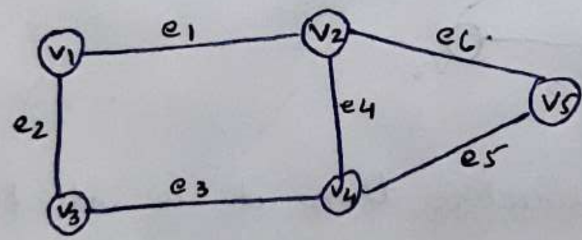
※ Graph :-

A graph G is mathematical structure consisting of 2 sets V and E, where V is non-Empty set of vertices and E is a non-Empty set of Edges.

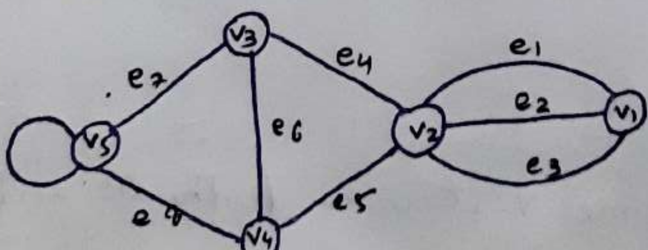
Ex :-



$V = \{v_1, v_2\}$
$E = \{e_1\}$



$V = \{v_1, v_2, v_3, v_4, v_5\}$
$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$



$V = \{v_1, v_2, v_3, v_4, v_5\}$
$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$.

✱ Basic Terminology :-

(1) **Trivial Graph :-**
A graph consisting only one vertex and no edge.

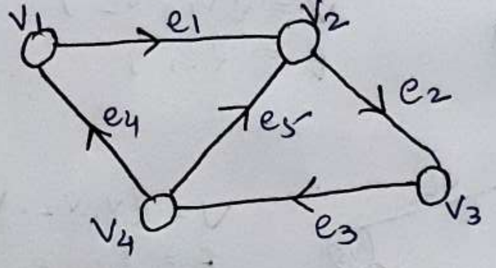
Ex :- •

(2) **Null Graph :-**
A graph consisting n vertices and no edges

Ex :- • • • •

(3) **Directed Graph :-**
A graph consist the direction of Edges then this is called Directed Graph.

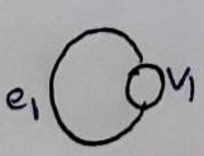
Ex :-

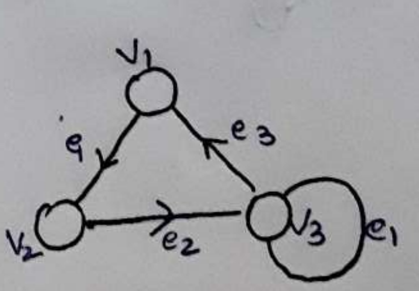


(4) **Un-directed Graph :-**
A graph which is not directed then it is called Undirected Graph.

(5) **Self Loop in a Graph :-**
If Edge having the same vertex as both its end vertices is called Self Loop.

Ex :-



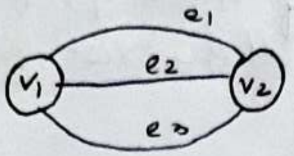


(6) **Proper Edge :-**
An Edge which is not self loop is called Proper Edge.

(7) Multi Edge :-

A collection of 2 or more edges having identically end point.

Ex:-

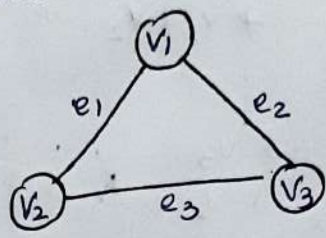


$\Rightarrow e_1, e_2, e_3$ are multi-Edge.

(8) Simple Graph :-

A graph does not contain any self loop and multi-Edges.

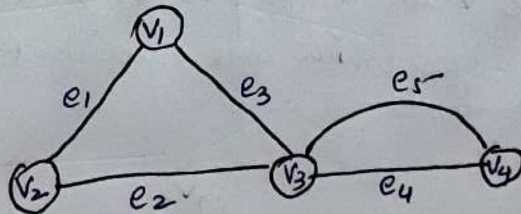
Ex:-



(9) Multi-Graph :-

A graph does not contain any self loop but contain multi-Edge is called Multi-Graph.

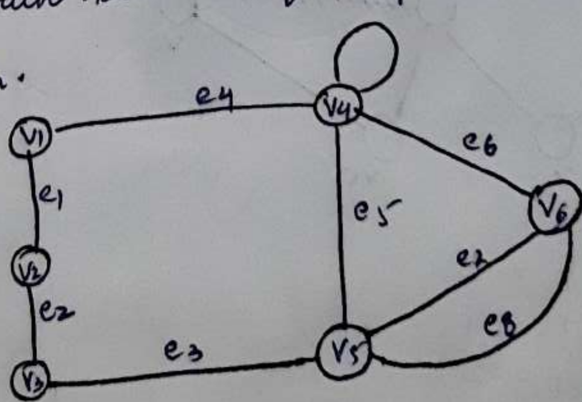
Ex:-



(10) Pseudo Graph :-

A graph contain both self loop and multiedge is called Pseudo Graph.

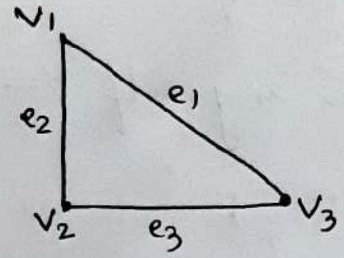
Ex:-



(11) Incidence and Adjacency :-

Let e_k be an edge joining two vertices V_i and v_j then e_k is said to be Incident of V_i and V_j. Two vertices are said to be Adjacent if there exist an edges joining this vertices.

Ex:-



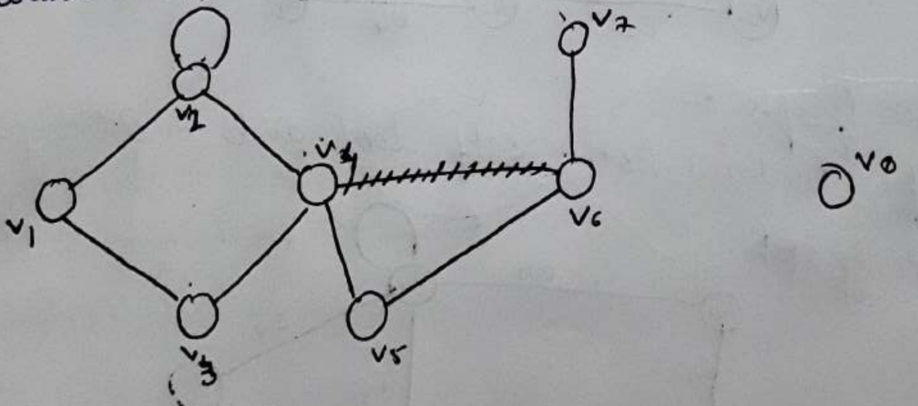
* v_1 and V_3 are adjacency of e_1
* V_2 & V_3 are adjacency. of e_3
* v_1 & V_2 are adjacency of e_2.

* e_1 is Incidente of v_1 and V_3.
* e_2 is Incidence of V_1 and V_2.
* e_3 is Incidence of V_2 and V_3.

(12) Degree of Vertex :-

The degree of vertex V in a graph G written as $d(V)$ is equal to No. of Edges which are incident on V with self loop counted twice.

Ex:-



here,
$d(V_1) = 2$
$d(V_2) = 4$
$d(V_3) = 2$
$d(V_4) = 3$
$d(V_5) = 2$
$d(V_6) = 2$
$d(V_7) = 1$
$d(V_8) = 0$

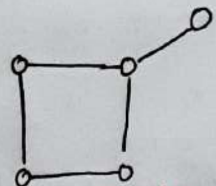
(13) Isolated Vertex and Pendant Vertex :-

A vertex having 0 is called Isolated vertex and a vertex having degree 1 is called Pendant vertex.

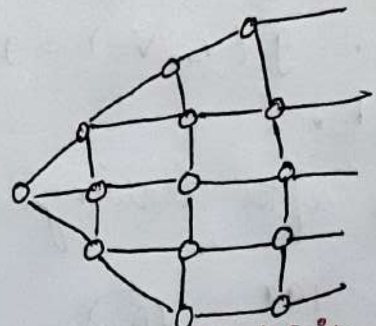
(14) Finite and Infinite Graph :-

A graph with a finite No. of vertices as well as Edges is called Finite Graph; otherwise it is infinite Graph.

Ex :-

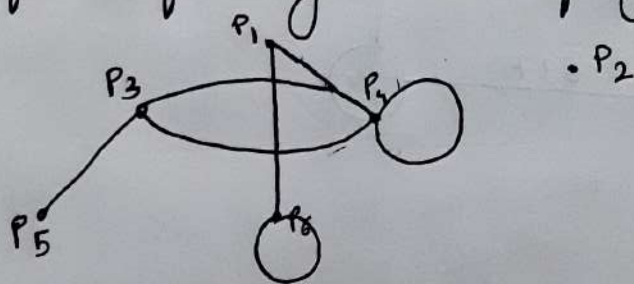


finite Graph



Infinite Graph.

Que :- which of the following is True for given Graph?



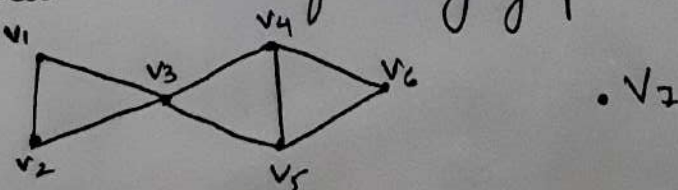
✓(a) $deg(P_6) = 3$

(b) $deg(P_6) = 2$

✓(c) $deg(P_1) = 2$

(d) $deg(P_2) = 1$

Que :- No. of isolated vertex in following graph is :-



(a) 0

✓(b) 1

(c) 2

(d) 3

Que: If $V = \{v_1, v_2, v_3, v_4, v_5\}$

$E = \{(v_1, v_4), (v_2, v_3), (v_3, v_4), (v_4, v_5)\}$

$f : V \rightarrow$ {indian cities}

$g : E \rightarrow N,$

$f(v_1) =$ Delhi , $g(v_1, v_4) = 199$

$f(v_2) =$ Raipur , $g(v_2, v_3) = 697$

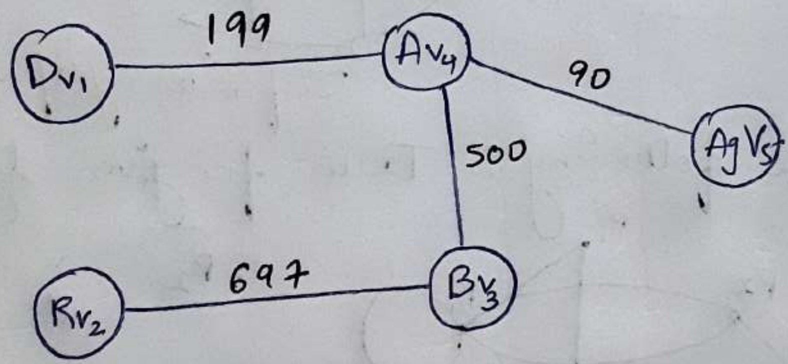
$f(v_3) =$ Bhopal, $g(v_3, v_4) = 500$

$f(v_4) =$ Agra, $g(v_4, v_5) = 90$

$f(v_5) =$ Aligarh.

Then to label after drawing the diagrams of the graph.

Ans:-

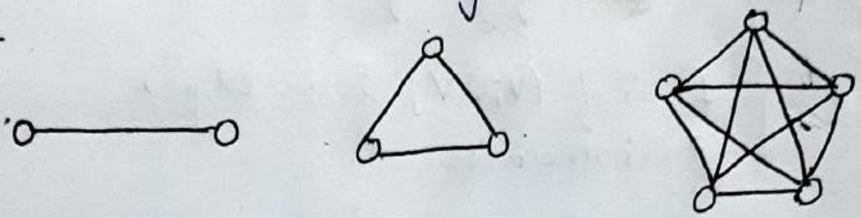


※ Some Important Graph :-

(1) Complete Graph :-

A simple connected graph is said to be complete if each vertex is connected to every other vertex.

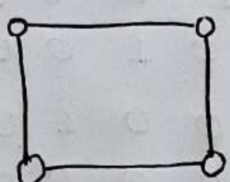
Ex :-



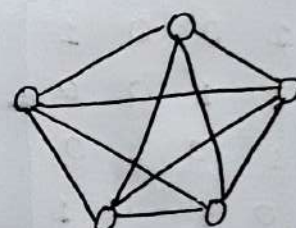
(2) Regular Graph :-

A graph G is said to be regular if every vertex has the same degree. If degree of each vertex of graph G is K, then it is called K-regular graph.

Ex :-



2- Regular Graph

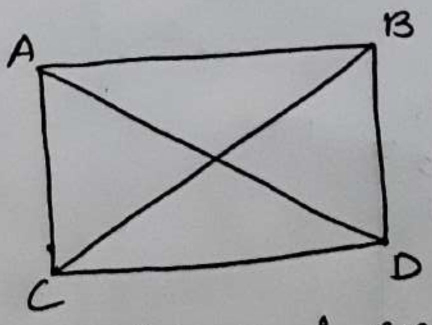


4- Regular Graph

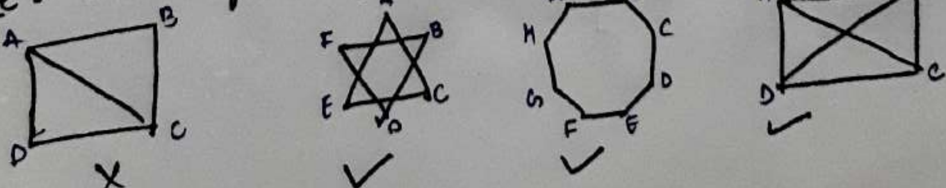
※ Connected Graph :-

A undirected graph is said to be connected if there is a path b/w every 2 vertices.

Ex :-



Que :- which of the following is regular graph ?



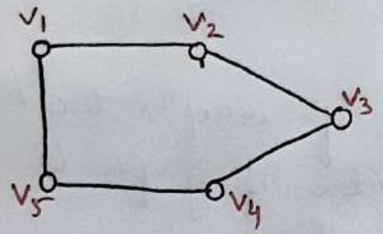
Matrix Representation of Graph :-

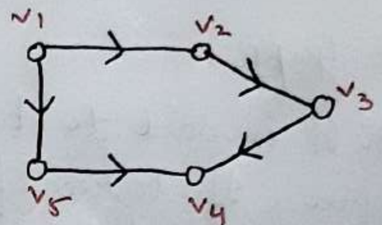
(1) Adjacency Matrix :-

Let a_{ij} denote the no. of edges (v_i, v_j), then $A = [a_{ij}]_{m\times m}$ is called Adjacency Matrix of G if

$$a_{ij} = \begin{cases} 1 & ; \text{if } (v_i, v_j \text{ is an edge}) \\ 0 & ; \text{otherwise} \end{cases}$$

(1) Ex :-



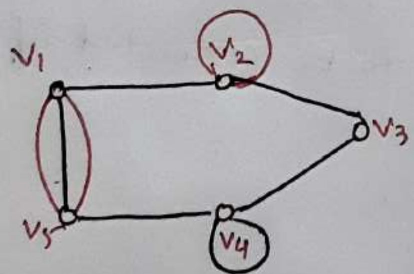


Sol :-

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

(2) Ex :-



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 3 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 3 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

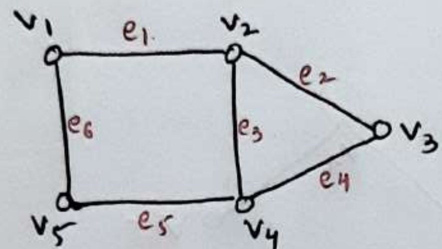
12) **Incidence Matrix :-**

Let G be a graph with m vertices $v_1, v_2, v_3 \ldots\ldots v_m$. and n edges $e_1, e_2, e_3, e_4, e_5 \ldots\ldots e_n$.

Let a matrix $M = [m_{ij}]_{m \times n}$ defined by.

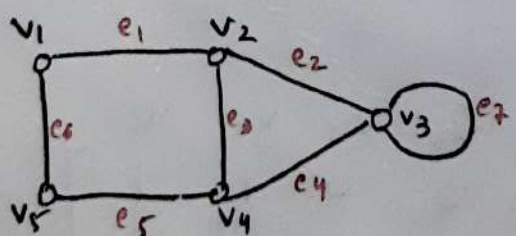
$$m_{ij} = \begin{cases} 1 ; & \text{if the vertex } v_i \text{ is incident on the } e_j. \\ 0 ; & \text{if } v_i \text{ is not incident on } e_j. \\ 2 ; & \text{if } v_i \text{ is an end of loop } e_j \text{ (self loop).} \end{cases}$$

Ex :-



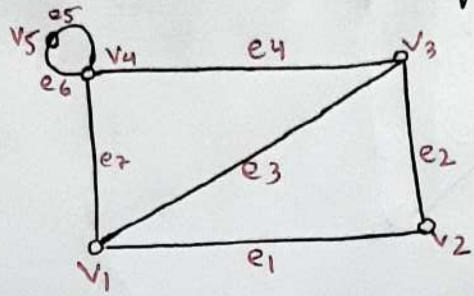
$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Ex :-



$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Ex:- Draw Incidence matrix of the given graph.



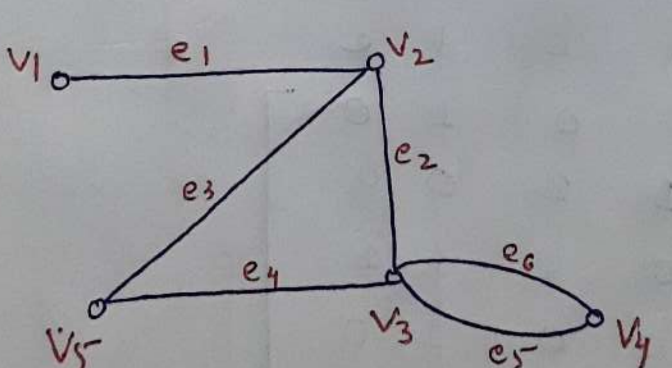
Sol:-

$$
\begin{array}{c}
 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}
$$

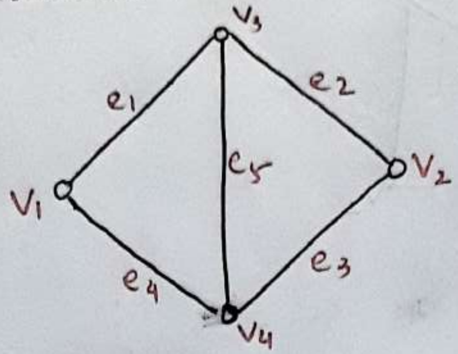
Que:- Draw the graph whose Incidence matrix A is given by

$$
\begin{array}{c}
 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5
\end{array}
\begin{array}{c}
\begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

Sol:-



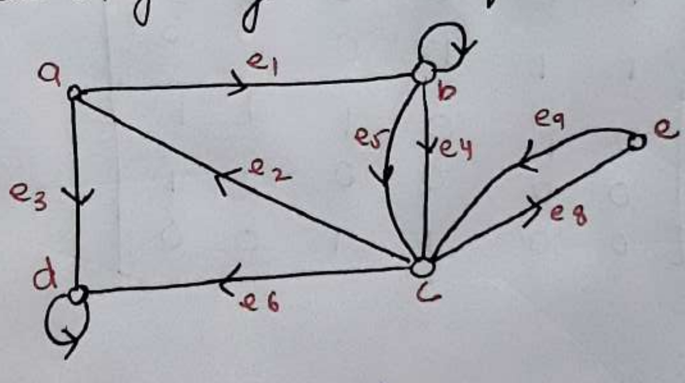
Ex :- Consider the undirected graph G as shown below:
Find incidence matrix.



Sol:- Incidence Matrix :-

$$\begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ \cancel{V_5} \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ & & & & \end{bmatrix} \end{array}$$

Ex :- Find the Adjacency Matrix of the given graph.



Sol:-

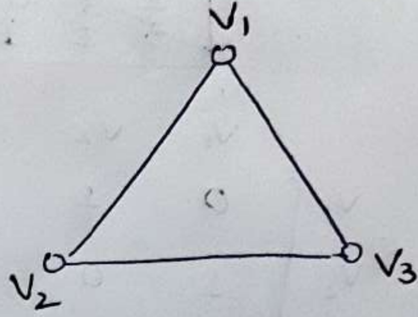
$$\begin{array}{c|ccccc} & a & b & c & d & e \\ \hline a & 0 & 1 & 0 & 1 & 0 \\ b & 0 & 1 & 2 & 0 & 0 \\ c & 1 & 0 & 0 & 1 & 1 \\ d & 0 & 0 & 0 & 1 & 0 \\ e & 0 & 0 & 1 & 0 & 0 \end{array}$$

Que :- Draw graph with the following Adjacency Matrix

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

Sol :-

$$\begin{array}{c c} & \begin{array}{ccc} V_1 & V_2 & V_3 \end{array} \\ \begin{array}{c} V_1 \\ V_2 \\ V_3 \end{array} & \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{array}$$

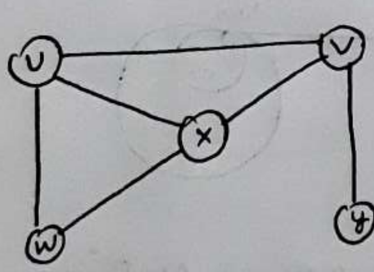


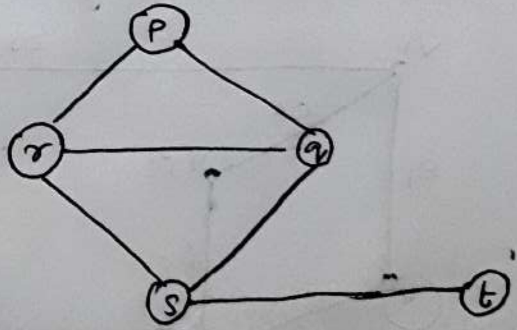
\# Isomorphic Graphs :-

Two graphs are Isomorphic (to each other). if there is a one to one correspondence b/w their vertices and b/w their Edges such that their Incidence relationship is preserved. Thus two graph are Isomorphic . if ;

1) They have same no. of Edges.
2) They have same no. of Vertices.
3) They have Equal no. of vertices with a given Degrees.

Ex :-





1) No. of vertices are same ;

$$\{u, v, w, x, y = p, q, r, s, t\}$$

2) No. of Edges are same ;

$$\{6 = 6\}$$

3) Degree of vertices.

u = 3	p = 2
v = 3	q = 3
w = 2	r = 3
x = 3	s = 3
y = 1	t = 1

* degree of 1 = 1 vertex {y = t}
degree of 3 = 3 vertex {u, v, w = q, r, s}
degree of 2 = 1 vertex {w = p}

4) vertex 1-1 correspondence and Edge 1-1 correspondence

* (1 degree) $y \to t$
(2 degree) $w \to p$
(3 degree) $v \to s$
$u \to q$
$x \to r$

{check y and t :-
correspondence of Y is V | correspondence of t is S
then, $\boxed{V = S}$
degree (V) = 3 and degree (S) = 3 }

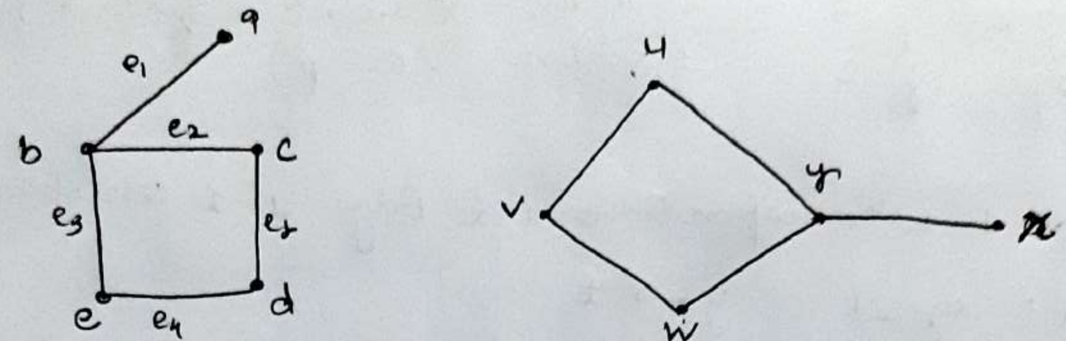
* Edges 1-1 Correspondence:

1. $\begin{bmatrix} y \to t \\ w \to p \end{bmatrix}$
2. $\begin{bmatrix} w \to p \\ V \to S \end{bmatrix}$ 2
3. $\begin{bmatrix} V \to S \\ u \to q \end{bmatrix}$ 3
4. $\begin{bmatrix} u \to q \\ x \to r \end{bmatrix}$ 4

{check, y - w : no edge , t - p : no edge.
w - V : no edge , p - s : no edge.
V - u = 1 Edge , s - q = 1 edge
u - x = 1 edge , q - r = 1 edge

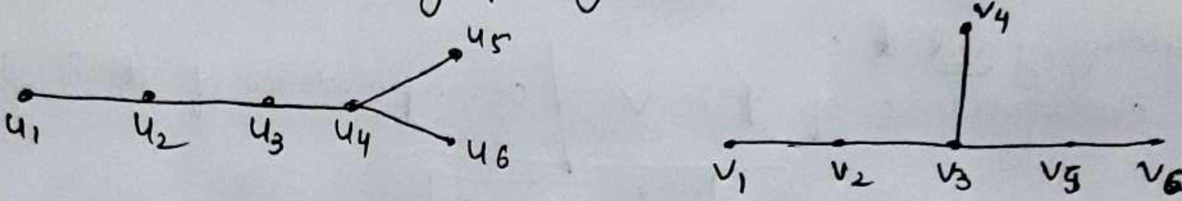
Hence, these Both graph are Isomorphic graph
*

Ex:- Show that the 2 graphs are Isomorphic :-



(Do it yourself)

Ex:- Show that the graph given below are not Isomorphic



Sol:- 1) No. of vertices are same;

$\{u_1, u_2, u_3, u_4, u_5, u_6 = v_1, v_2, v_3, v_4, v_5, v_6\}$

2) No. of Edges are Same;

$\{5 = 5\}$

3) degree of vertices.

$u_1 = 1$	$v_1 = 1$
$u_2 = 2$	$v_2 = 2$
$u_3 = 2$	$v_3 = 3$
$u_4 = 3$	$v_4 = 1$
$u_5 = 1$	$v_5 = 2$
$u_6 = 1$	$v_6 = 1$

1-1 correspondence :- {vertices}

* degree of 1 = 3 vertices : $u_1 \leftrightarrow v_1$, $u_5 \leftrightarrow v_4$, $u_6 \leftrightarrow v_6$

degree of 2 = 2 vertices : $u_2 \leftrightarrow v_2$, $u_3 \leftrightarrow v_5$

degree of 3 = 1 vertex : $u_4 \leftrightarrow v_3$

1 - 1 correspondence of Edges :-

$u_1 - v_1$

$u_5 - v_4$

$u_6 - v_6$

$u_2 - v_2$

$u_3 - v_5$

$u_4 - v_3$

$u_1 - u_5 \rightarrow$ no edge , $v_1 - v_4$ no edge

$u_5 - u_6 \rightarrow$ no edge , $v_4 - v_6$ no edge

$u_6 - u_2 \rightarrow$ no edge , $v_6 - v_2$ no edge

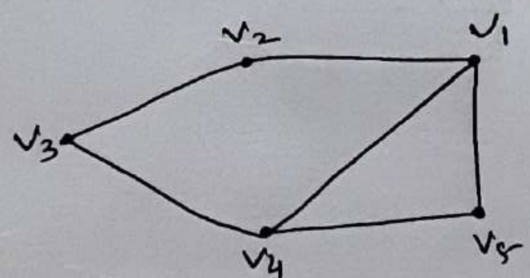
$u_2 - u_3 \rightarrow$ 1 edge , $v_2 - v_5$ no edge

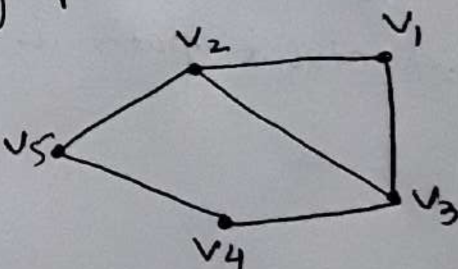
$u_3 - u_4 \rightarrow$ 1 edge , $v_5 - v_3$ no edge

Hence, there is no correspondence b/w $u_2 - u_3$ and $u_3 - u_4$

Hence, the given graph is not Isomorphic

Que :- Show that the given graph is Isomorphic :





* Various Graph properties :-

1) **walk** :- A walk is a finite alternating Sequence $v_1 e_1 v_2 e_2 \dots v_n e_n$. of vertices and edges, beginning and ending with same or different vertices.

* **Length of walk** :- The no. of Edges is called length of the walk {which are covered in walk}

* **closed and open walk** :- A walk is said to be closed if its origin and terminals vertex ($v_0 = v_n$) is Equal otherwise it is ~~closed~~ open walk.

2) **Trail** :- Any walk having different edges is called Trail. {not Repeated Edge}

3) **Circuit** :- A closed trail is called Circuit.

4) **Path** :- A walk is called path if all vertices are not repeated.

5) **Cycle** :- A closed path is called Cycle.

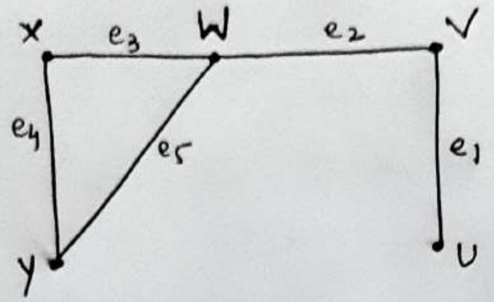
{Jab vertices sari cover ho jaenge and circuit close bhi hoga tab Hamiltanion graph hota hai!!! OR
Jab sari ~~vertices~~ Edges cover ho jaengi to Euler graph but Edges repeat nahi honi chahiye }.

If we are moving on graph, by covering all the edges of the graph and returning to the initial vertex, such a walk is called a Euler line and graph is called an Euler Graph.

Eulerian Path :-

A path in a graph is said to be an Eulerian path if it traverses each edge in the graph once and only once.

Ex :-

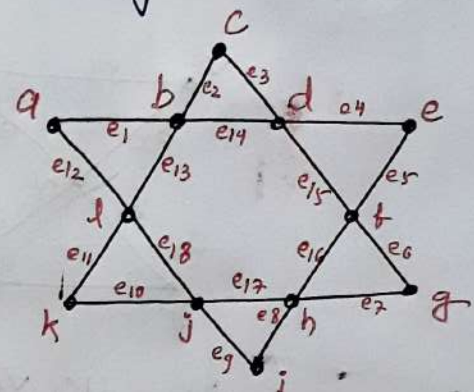


The Eulerian path of this graph is :- $Ue_1 Ve_2 We_3 Xe_4 Ye_5 W$

Eulerian Circuit :-

A circuit in a graph is ~~call~~ said to be an Eulerian Circuit if it traverses each edge in the graph one & only once, except the starting vertex.

Ex :-

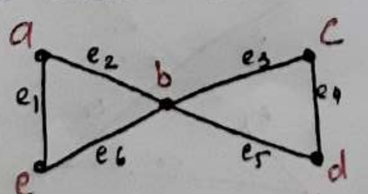


The Eulerian circuit of this graph is :-

le_{11}, ke_{10}, je_9, ie_8, he_7, ge_6, fe_5, ee_4, de_3, ce_2, be_1, ae_{12}, le_{18}, e_{17}, e_{16}, e_{15}, e_{14}, e_{13} l.

Eulerian Graph :-

A connected graph which contain an Eulerian circuit is called Eulerian graph.

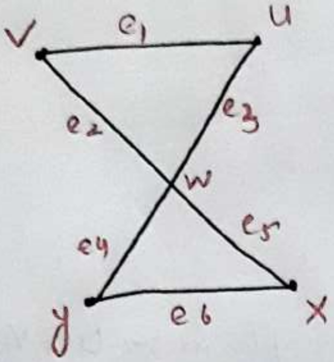


The Eulerian graph of this graph is be_3, ce_4, de_5, be_2, ae_1, ee_6, b.

※ Hamiltonian Path :-

A path which contain every vertex of a graph G exactly once is called Hamiltonian graph.

Ex:-

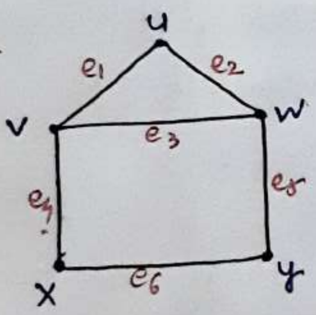


then $ue_1, ve_2, we_5\, xe_6\, y$.

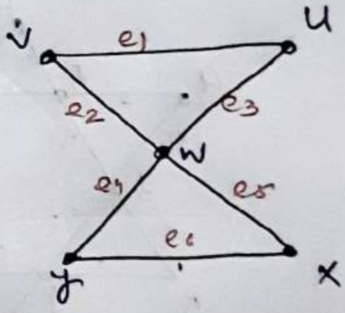
※ Hamiltonian Circuit :-

A circuit that passes through each of the vertices in a group G exactly one except the starting vertex and End vertex is called Hamiltonian Circuit.

Ex:-



then, $ue_1, ve_4, xe_6, ye_5, we_2\, u$



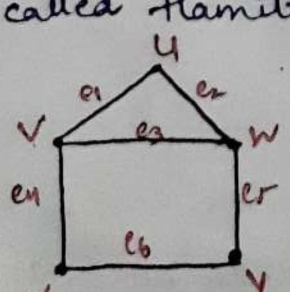
then, $ue_1\, ve_2\, we_5\, xe_6\, ye_4$ $we_3\, u$.

is not Hamiltonian circuit because w vertex Repeat.

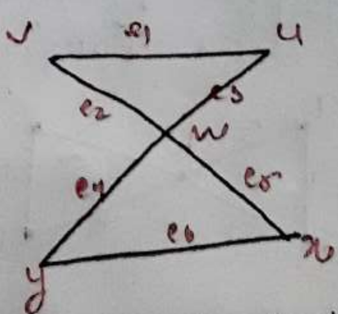
Hamiltonian Graph :-

A connected graph which contain Hamiltonian Circuit is called Hamiltonian Graph.

Ex:-



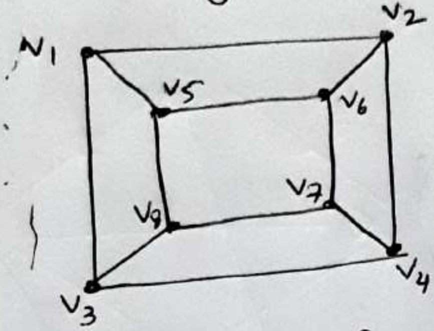
u - v - x - y - w - u



v - u - w - y - x - w - v

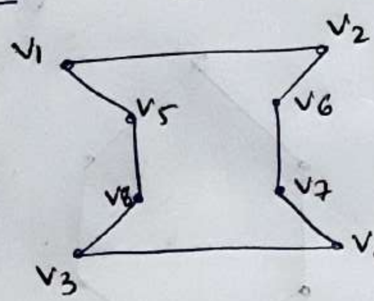
⇒ w is repeating ∴ it is not Hamiltonian graph.

Que: Discuss the diagram shown in below:-

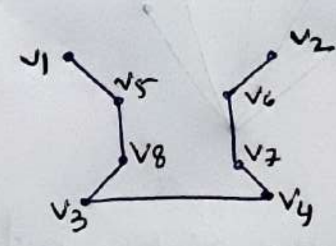


(i) Draw Hamiltonian Circuit and Hamiltonian path.
(ii) It it Hamiltonian graph?
(iii) Is it Euler graph?

Sol:- (i) Hamiltonian Circuit :- Graph which covers all vertices.



Hamiltonian Path :-



(ii) Hamiltonian graph = ?
* Graph which covers all vertices.
* connected
* Circuit.

It satisfies all the properties for Hamiltonian graph.

(iii) Euler graph :- Graph which cover all Edges in given graph is known Euler graph.
But in this graph Edges get repeated So, it is not Euler graph.

Since the given grap

Fig. 3.33 is a Hamiltonian graph.

Example 2. *Show that the graph in Fig. 3.36 is Hamiltonian. Draw a Hamiltonian circuit and a Hamiltonian path from this graph.*

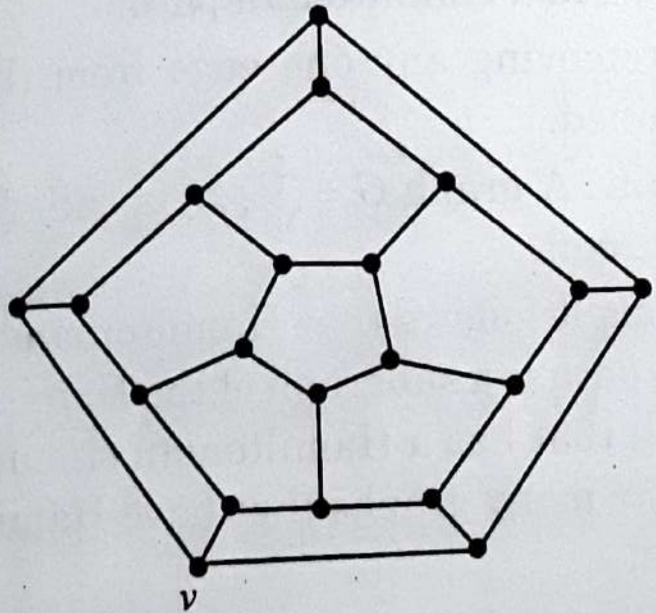


Fig. 3.36. Hamiltonian Graph

gree.

Example 4. *Draw a graph with six vertices containing an Eulerian circuit but Hamiltonian circuit.*

Solution. The required graph is as shown in Fig. 3.41.

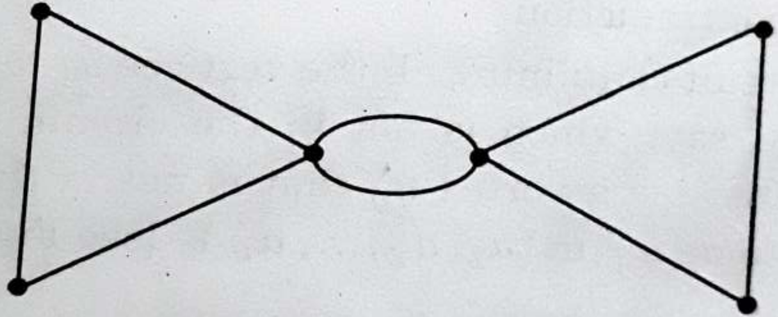
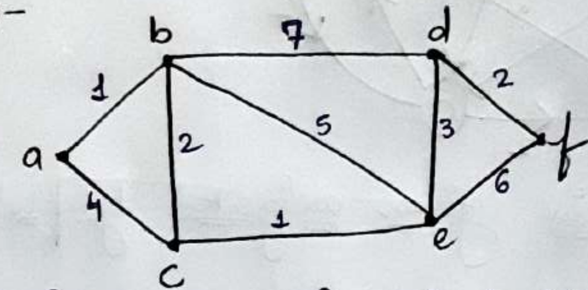


Fig. 3.41

※ Weighted Graph and Shortest Path :-

A ~~weig~~ graph is called weighted graph if a non-negative integer W(e) associated to each edge, and this W(e) is a weight of corresponding edge.

Que: Find the shortest path b/w a and f for the following graph :-



Sol:- Shortest Path in Weighted Graph :-

* Dijkstra's Algorithm :-

a	b	c	d	e	f	
$\boxed{0}$	∞	∞	∞	∞	∞	a
$\boxed{0}$	$\boxed{1}$	4	∞	∞	∞	a → b
$\boxed{0}$	$\boxed{1}$	$\boxed{3}$ 1+2	8 1+7	6 1+5	∞	a → b → c
$\boxed{0}$	$\boxed{1}$	$\boxed{3}$	8	$\boxed{4}$	∞	a → b → c → e
$\boxed{0}$	$\boxed{1}$	$\boxed{3}$	$\boxed{7}$	$\boxed{4}$	10 4+6	a → b → c → e → d
$\boxed{0}$	$\boxed{1}$	$\boxed{3}$	$\boxed{7}$	$\boxed{4}$	$\boxed{9}$	a → b → c → e → d → f

$= 0 + 1 + 2 + 1 + 3 + 2$

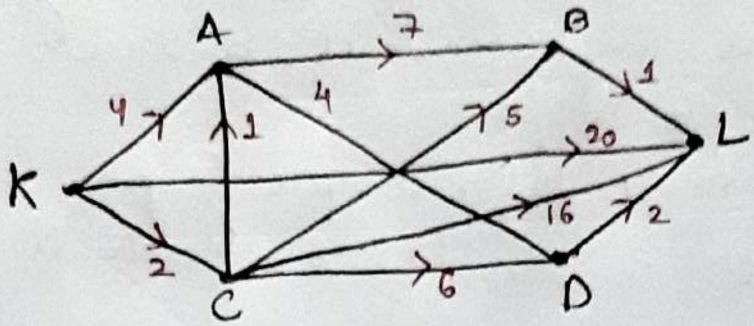
$= \underline{9}$

Hence, the path is;

{a, b, c, e, d, f}

*

Que: Find minimum Distance b/w 2 vertices K and L of graph.



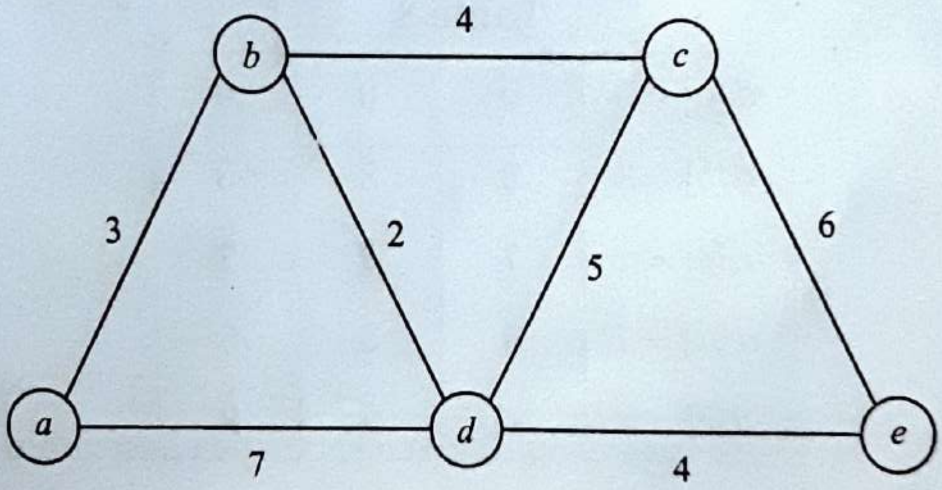
Sol:- Shortest Path in given weighted graph :-

K	A	B	C	D	L	
$\boxed{0}$	4	∞	$\boxed{2}$	∞	20	K → C = 2
$\boxed{0}$	$\boxed{3}$ (2+1)	7 (2+5)	$\boxed{2}$	8 (6+2)	18 (2+16)	K → C → A = 3
$\boxed{0}$	$\boxed{3}$	$\boxed{7}$ (min)	$\boxed{2}$	7 (4+3)	18	K → C → A → B = 7
$\boxed{0}$	$\boxed{3}$	$\boxed{7}$	$\boxed{2}$	$\boxed{7}$	8	K → C → A → B → D = 7
$\boxed{0}$	$\boxed{3}$	$\boxed{7}$	$\boxed{2}$	$\boxed{7}$	$\boxed{8}$	K → C → A → B → D → L

minimum Distance = 8

Path = K → C → B → L

Example 2. *Solve the following using Dijkstra's algorithm assuming source node as a.*



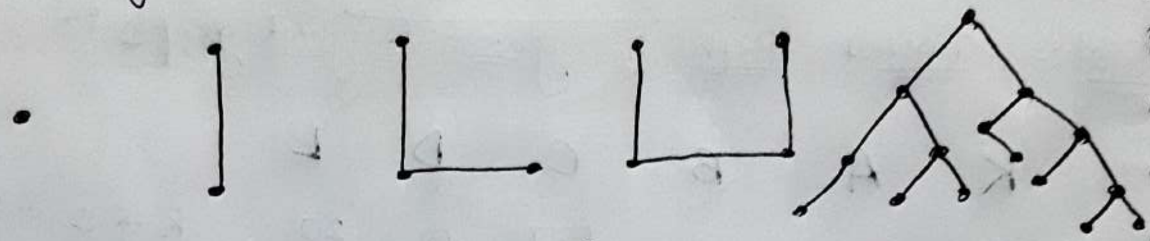
TREES :-

A tree is a Simple Connected graph without any circuit.
∵ tree cannot have a Simple circuit, a tree cannot contain multiple edges or loops.

∴ ~~Tree~~ Any tree must be a simple graph.

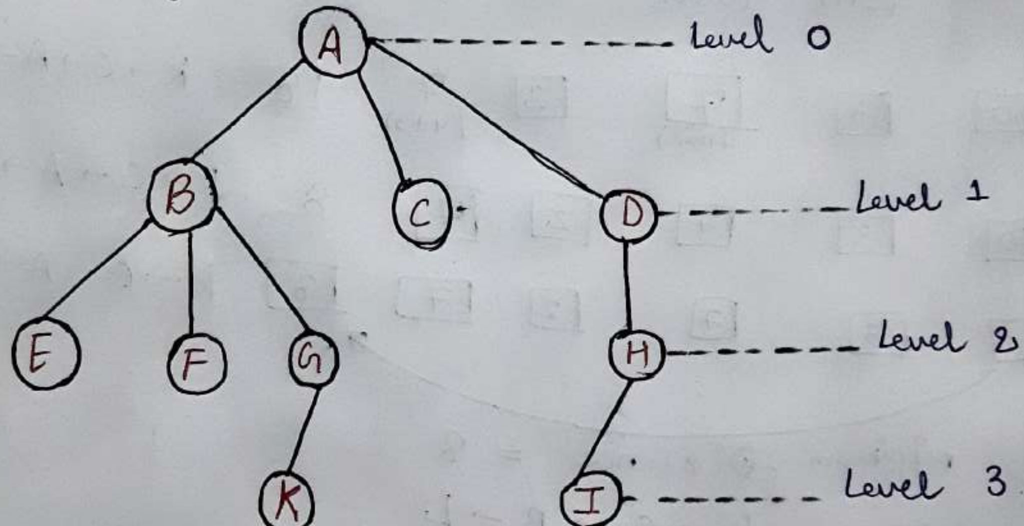
A tree having no vertex can be viewed as a Null trees.

Ex:-



Trees with 1, 2, 3, 4, n vertices.

Basic Terminologies of Trees :-



* Root of the tree :-

A root node is the origin of the tree. whole tree structure accessed from the Root Node. In a tree, root node is unique (only one). Root Node does not have any parent Node.

'A' is Root Node of the given tree.

* **Degree of Node :-**

The degree of Node is the number of child Node are connected to it.

d(A) = 3 d(D) = 1 d(G) = 1

d(B) = 3 d(E) = 0 d(H) = 1

d(C) = 0 d(F) = 0 d(I) = 0

d(K) = 0

* **Path of tree :-**

A path is nothing but, a sequence of edge b/w that Nodes. There are multiple path are exists in a single tree.

Ex :- A → B → E A → C

A → B → G → K A → B → F

A → D → H → I

* **Parent Node :-**

Parent Node of any node is a Immediate preceder (predecessor) of that Node.

* **Child Node :-**

Child Node is immediate succesor of any Node.

Ex :- A is parent Node for B, C, D and

B, C, D are child Nodes for A.

* **Sibilings :-**

Sibilings are Nodes with same parents Node.

Ex :- B, C, D are sibilings of each other.

* **Pendent vertex (Node) in a tree :-**

A vertex of degree 1 is called pendent vertex (Node) of tree

Ex :- D, G, H.

* Leaf Nodes :-

A Node with no child Node.
In other words, a node whose degree is zero is called as Leaf Node.

Ex :- E, F, K, I are Leaf Nodes

* Internal Nodes :-

Internal Node is a Node between Root Node and Leaf Nodes.

Ex :- B, D, G, H are Internal Nodes.

* Level of tree :-

The level of the tree is represented by No. of stages from Root Node to leaf Node.
On each stage, there are some Node available. generally root Node is at level 0.

* Depth of Node :-

The depth of Node is the No. of Edges from the Node to Root Node of the tree.

Depth of G = 2
Depth of I = 3

* Height of tree :-

The height of the tree is the No. of Edges on the longest downward path b/w the root Node and a leaf Node.

$A \rightarrow B \rightarrow G \rightarrow K = 3$
$A \rightarrow D \rightarrow H \rightarrow I = 3$

height of tree = 3.

* Degree of tree :-

Maximum No. of Degree of a ~~tree~~ Node in given tree is known as Degree of tree.

The degree of given tree = 3.

* Properties of tree :-

1) A graph G is a tree if and only if there exist a unique path b/w every distinct pair of vertices of G.

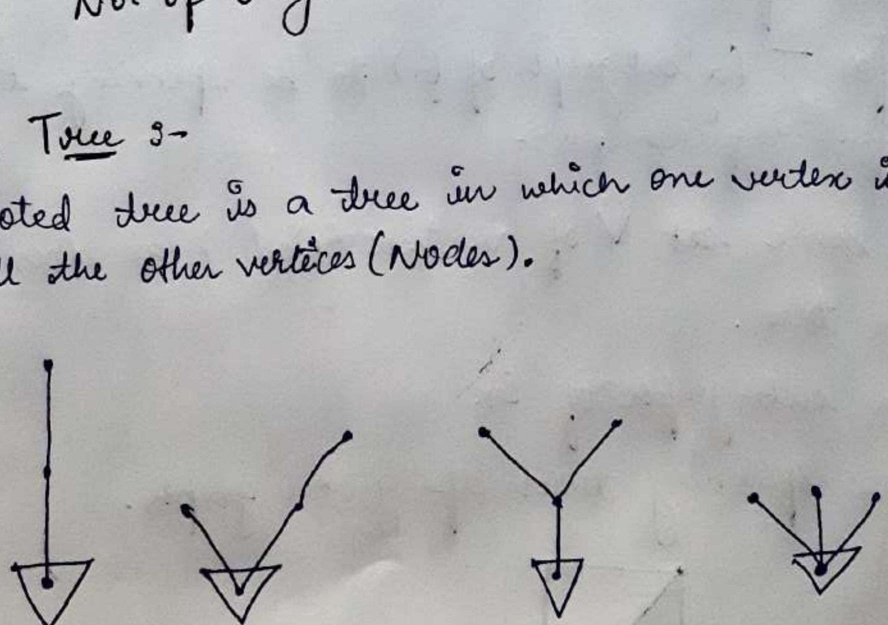
2) A graph G of n vertices is a tree if and only if G is connected and has exactly $(n-1)$ edges.

Ex :- 1) If we have to go to k then we have only and only 1 path $A \rightarrow B \rightarrow G \rightarrow K$.

2) No. of ~~ver~~ Nodes $= n = 10$
No. of Edges $= (n-1) = 9$.

* Rooted Tree :-

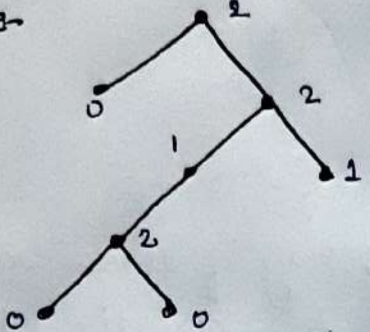
A Rooted tree is a tree in which one vertex is distinguished from all the other vertices (Nodes).



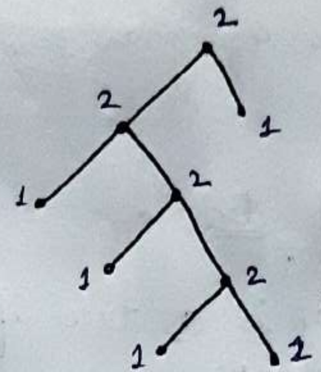
※ Binary tree :-

Binary tree is a tree in which every Node or vertex has atmost 2 childs.

Ex :-



Binary tree.



full Binary tree

※ Spanning Tree :-

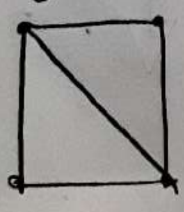
Let $G = (V, E)$ be a connected graph. A spanning tree (also called as Skeleton) of G is a subgraph $T = (V, E')$ of G, which is tree, where $E' \subseteq E$.

In other words, a Spanning tree T of a connected graph G is a subgraph of G which is a tree and includes all the vertices of G.

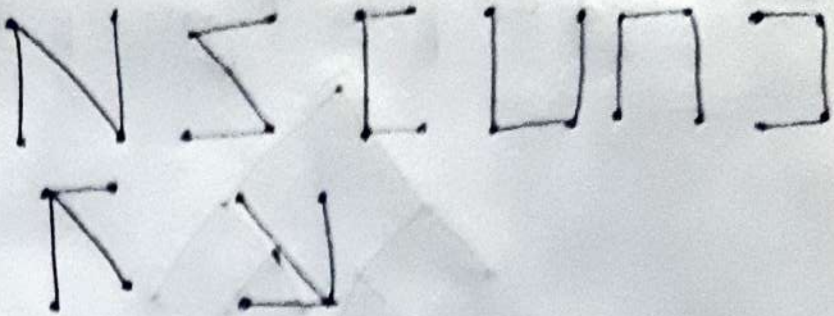
Thus for T (a subgraph of G) to be spanning tree, we have few requirements :-

1) T has same set V of vertices (Nodes) as does G has.
2) T is a tree.
3) T is a sub-graph of G.

Ex :- Draw Spanning trees of given graph.

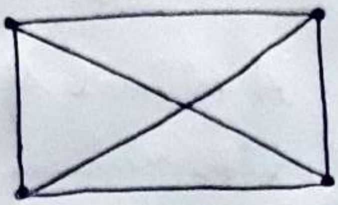


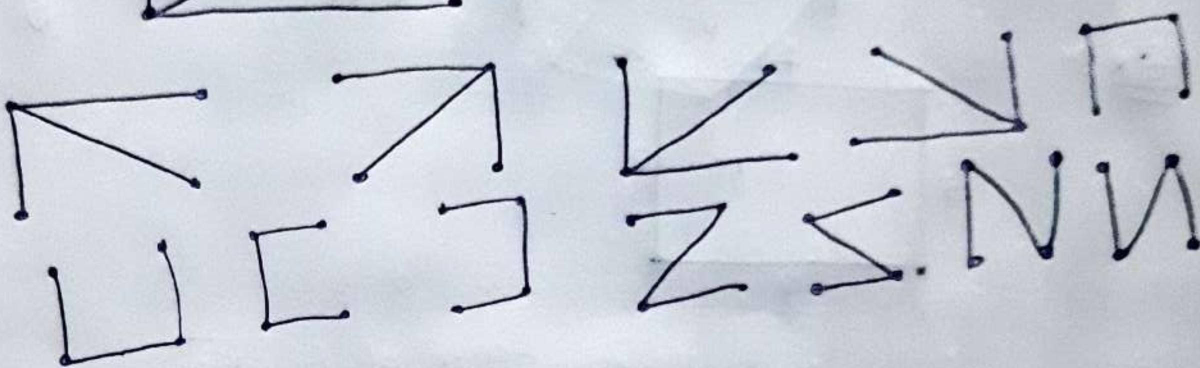
Sol:-



Que :- Draw a complete graph of 4 vertices and find Spanning tree in this graph.

Sol:-

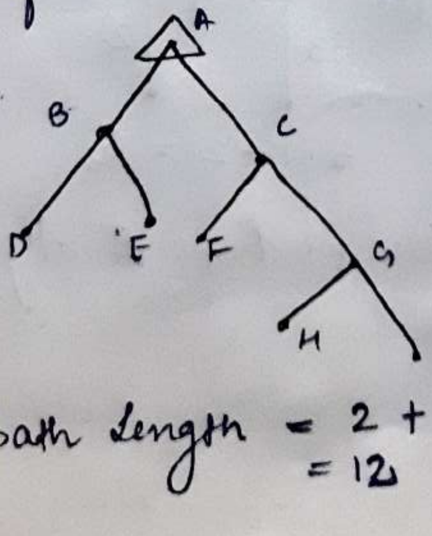




* Path Length of tree :-

A path length of a tree is defined as the sum of edges from the root of all pendent vertices / Leaf Nodes.

Ex:-



$D \rightarrow B \rightarrow A = 2$

$E \rightarrow B \rightarrow A = 2$

$F \rightarrow C \rightarrow A = 2$

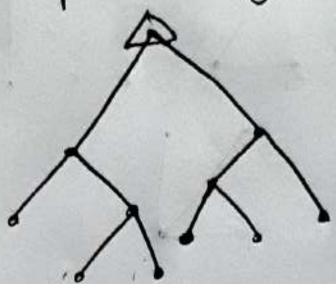
$H \rightarrow G \rightarrow C \rightarrow A = 3$

$I \rightarrow G \rightarrow C \rightarrow A = 3$

then path length $= 2 + 2 + 2 + 3 + 3$

$= 12$

Que :- Find the path Length of given tree :-

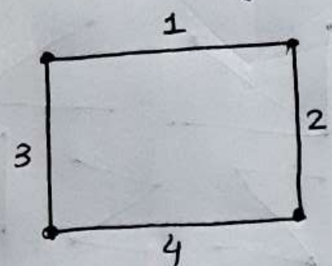


Sol:- path Length $= 2 + 3 + 3 + 3 + 3 + 2 = 16$

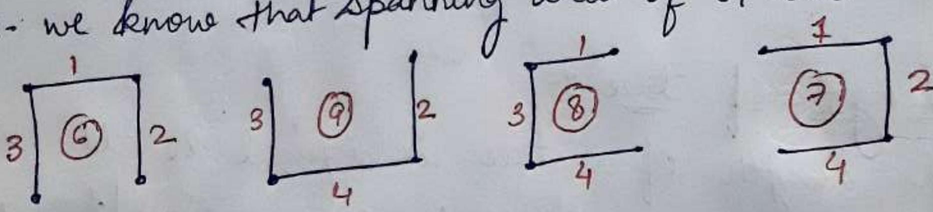
* Minimal Spanning tree :-

Let G be a weighted graph. A minimal Spanning tree of G is a Spanning tree of G with minimum weight.

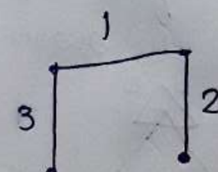
Ex :-



Sol:- we know that Spanning tree of G are



But minimal Spanning tree is



$3 + 1 + 2 = 6$

To find minimal Spanning tree we have 2 Algorithms:-

1) Krushkal's Algorithm.
2) Prism's Algorithm.

(1) KRUSHKAL'S ALGORITHM :-

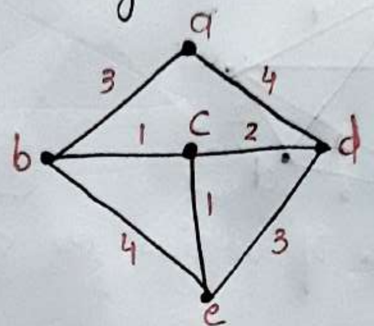
Working Rule :-

(i) choose an Edge of minimal weight.

(ii) At each step, choose the edge whose inclusion will not create a circuit.

(iii) If G has n vertices, stop after (n-1) edges.

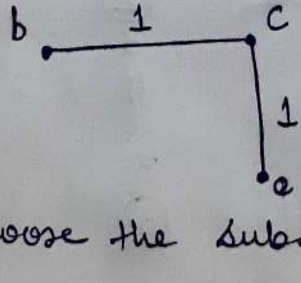
Que :- Find minimal Spanning tree of a given graph by Krushkal's Algorithm.



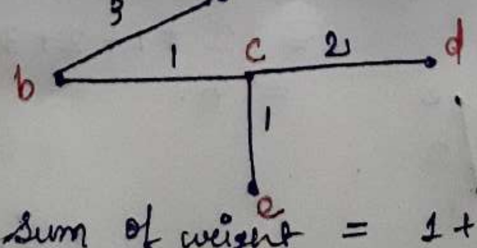
Sol :- In a graph G has 5 vertices ∴ the Spanning tree has 4 Edges.

Edges :-	bc	ce	cd	ab	de	be	ad
weight :-	1	1	2	3	3	4	4

Now choose an edge have minimum weight ; bc, ce



Now we choose the subsequent edges cd and ab we have



then, the sum of weight = 1 + 1 + 2 + 3 = 7.

(2) PRISM's ALGORITHM :-

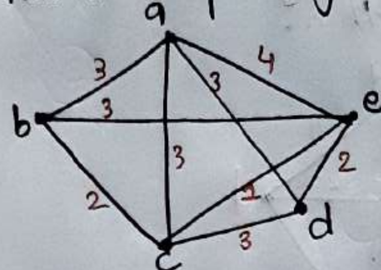
Working Rule :-

(i) Select any vertex and choose the Edge and Smallest weight from G.

(ii) At each stage, choose the edge of smallest weight joining a vertex already included to vertex not yet includ

(iii) Continue until all vertices are included.

Que :- Find the minimal spanning tree of given graph :-



Sol :- In a given graph No. of vertices are 5 and ∴ the span. tree has 4 Edges.

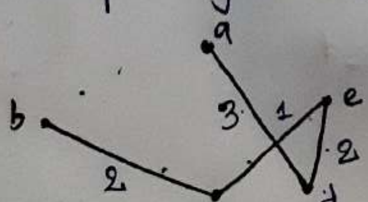
Now I^{st} we select arbitarily a vertex "b".

Step	vertex that may be added	Edges	weight
II	a c e	ba bc be	3 2 3
III	a e d	ca ce cd	3 1 3
IV	d b a	ed eb ea	2 3 4
V	c a	cd ad	3 3

Here minimal spanning tree will be → b → c → e → d → a

= 2 + 1 + 2 + 3

= 8.



are illustrated by the following examples :

Example 1. *Find a minimum spanning tree for the graph shown in Fig. 4.25.*

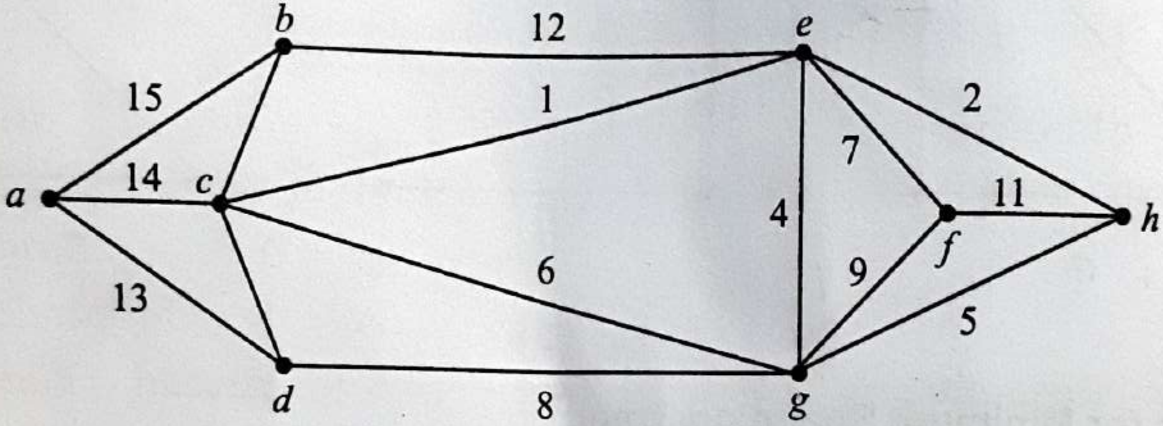


Fig. 4.25

Solution. We shall solve this problem by both the methods namely Kruskal's algorithm and Prim's algorithm.

Example 2. *Find the minimum spanning tree for the graph shown in Fig.* 4.27.

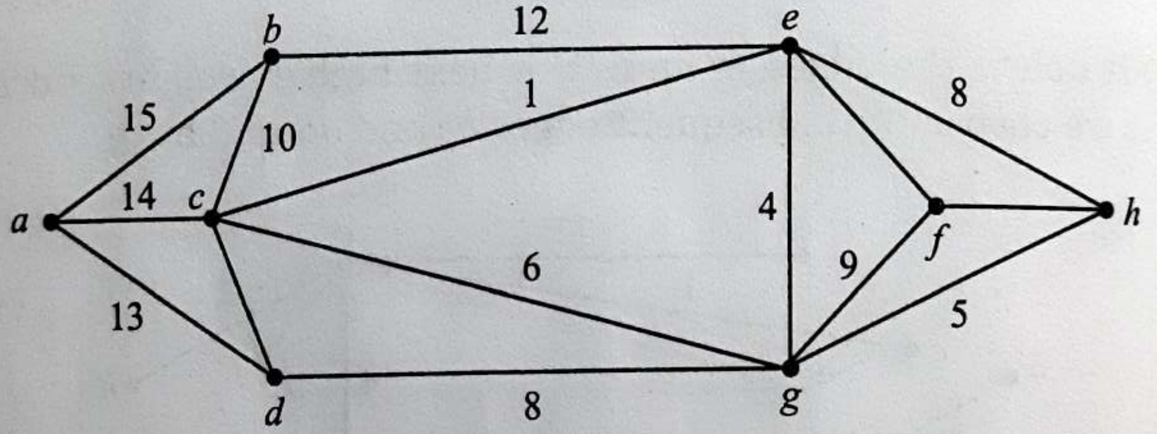


Fig. 4.27

Solution. The given graph G has 8 vertices, therefore any spanning tree will have 7 edges.

Example 6. *Find the minimum spanning tree for the graph given in Fig. 4.35.*

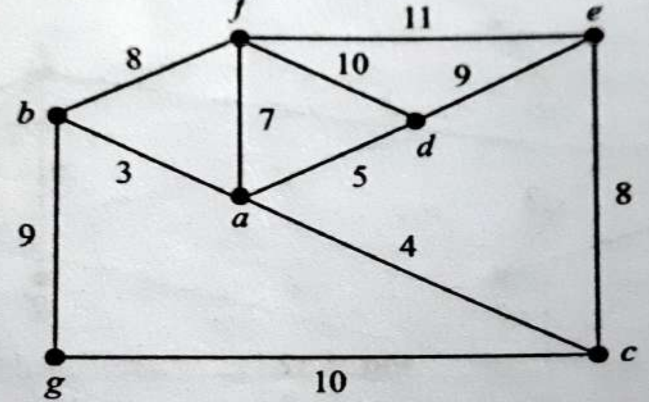


Fig. 4.35

* Tree's Transversal :-

A traversal of a tree is a process to traverse a tree in a Systematic way so, that each vertex is visited exactly once.

There are 3 methods of Tree Traversal :-

1) Pre order Traversal
2) Post order Traversal.
3) Inorder Traversal

* Pre-Order Traversal :-

The pre-order Traversal of a Binary tree is defined recursively as follows :-

(1) visit Root.
(2) Traverse the left subtree in preorder.
(3) Traverse the right sub-tree in pre-order.

(Root, left, Right).

* Post-order Traversal :-

The post order Traversal of a Binary tree is defined reccursively as follows :-

(i) Traverse the left sub-tree in post order
(ii) Traverse the right sub-tree in post order
(iii) visit Root.

(Left, Right, Root).

* In-Order Traversal :-

The in order Traversal of a Binary tree is defined reccursively as follows :-

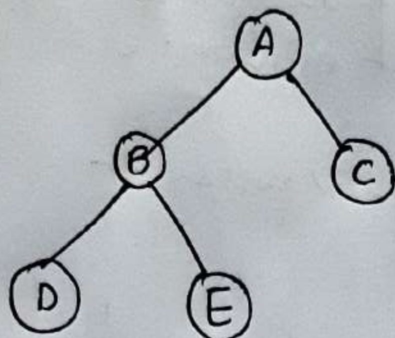
(i) Traverse the Left subtree in In-order.
(ii) Visit Root.
(iii) Traverse the Right sub-tree in In-order.

(Left . Root, Right).

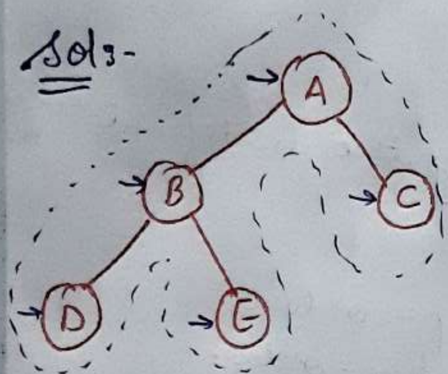
{NOTE :- In tree traversal we have to find path in 3 different ways.

Que: Find the tree traversal of given tree?

(1)



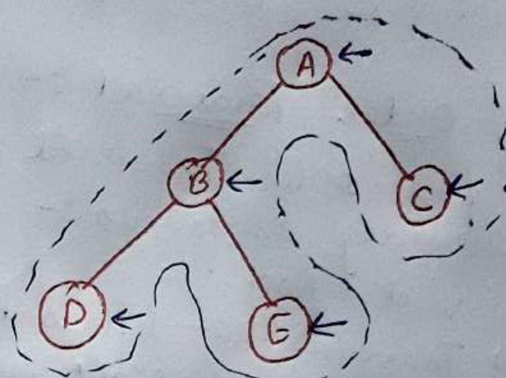
Sol:-



A → B → D → E → C ~~A~~

Pre-ordered Traversal.

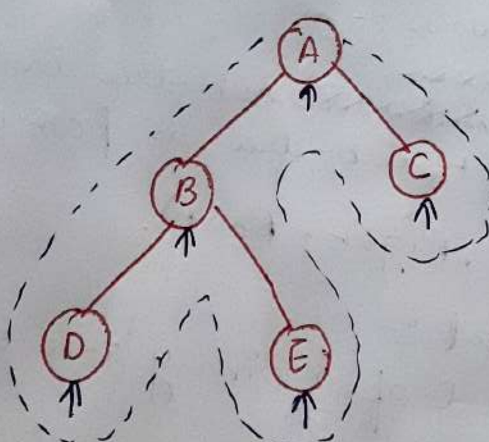
(ABDEC~~A~~)



D → E → B → C → A

Post-ordered Traversal

(DEBCA)

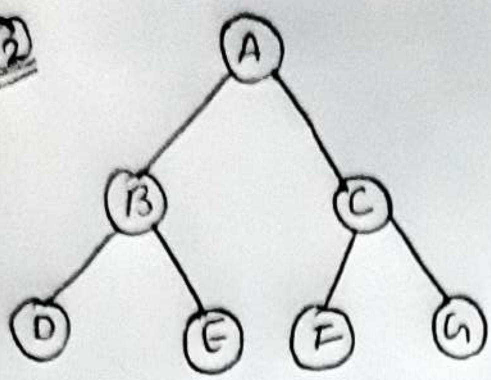


D → B → E → A → C

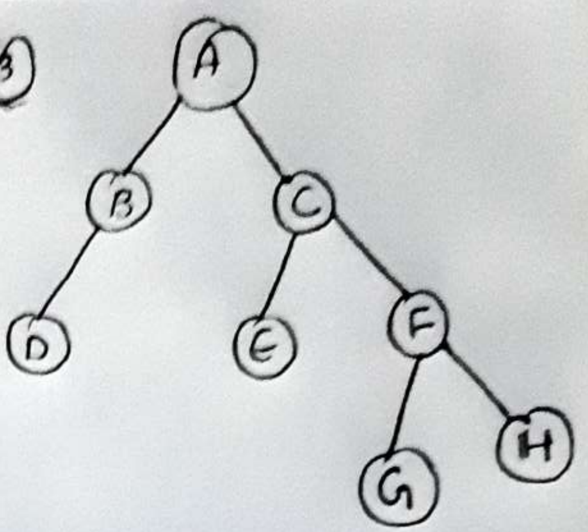
In-Order Traversal

(DBEAC)

(2)



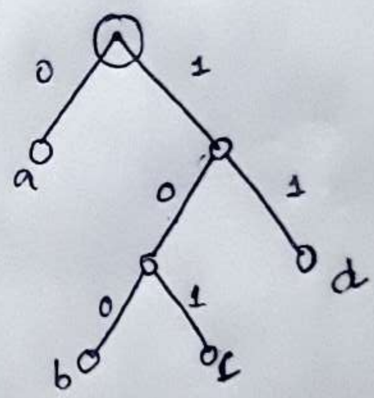
(3)



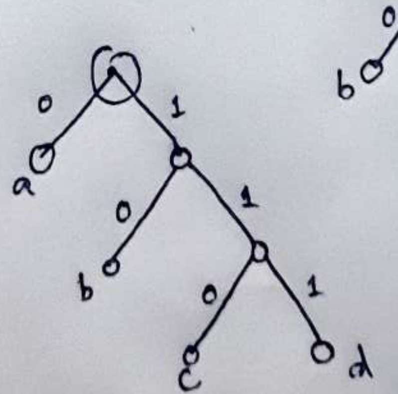
* Prefix Code :-

A prefix Code is a variable length code in which no codeword is prefix of another one. The codes are assigned in such a way that the code assigned in such a way that the code assigned to one character is not prefix of code assigned to any other character.

Ex:- a = 0.
b = 100
c = 101
d = 11



a = 0
b = 10
c = 110
d = 111



Que:- which of these codes are prefix Codes?

$A_1 = \{0, 10, 110, 1111\}$

$A_2 = \{1, 01, 001, 000\}$

$A_3 = \{1, 11, 101, 001, 0011\}$

}

$A_5 = \{0, 10, 101, 1010, 1100\}$