31/05/23

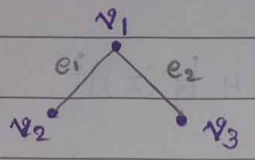
MODULE - 2

INTRODUCTION TO GRAPH THEORY

* **Graphs :**

A graph is a pair (V, E) where V is a non empty set and E is a set of unordered pairs of elements taken from the set V

eg:-



$\Rightarrow V = \{v_1, v_2, v_3\}$

$E = \{e_1, e_2\}$

$= \{v_1v_2, v_1v_3\}$

For a graph (V, E), the elements of V are called vertices and the elements of E are called undirected edges.

The set V is called the Vertex Set & the set E is called the Edge Set.

The graph (V, E) is also denoted by

$G = (V, E)$ (or) $G = G(V, E)$

Note:

* The Vertex set in a graph / digraph has to be non empty.

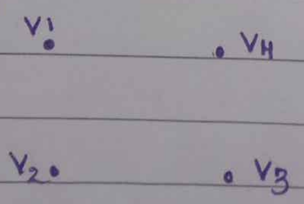
Thus a graph / digraph must contain atleast 1 vertex.

But the edge set can be empty

→ **NULL Graph :**

A graph / digraph containing no edges is called a NULL graph.

eg:-



→ **Trivial Graph :**

A NULL graph with only 1 vertex is called a trivial graph.

→ **Finite / Infinite Graph :**

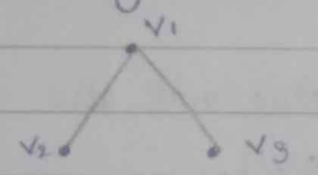
A graph/digraph with only a finite number of vertices as well as finite number of edges is called a finite graph / digraph otherwise it is called as infinite graph.

Order and Size :

The no. of vertices in a (finite) graph is called the order of the graph.

The no. of edges in a graph is called the size of the graph.

- The order of the graph is denoted by $|V|$ (cardinality of V) and size of the graph is denoted by $|E|$ (cardinality of E)



order of the graph $= |V| = 3$

size of the graph $= |E| = 2$

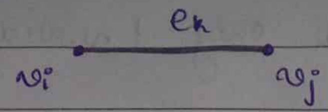
- A graph of order n and size m is called a (n, m) graph
- A null graph with n vertices is a $(n, 0)$ graph.

End vertices, loop, multiple edges :

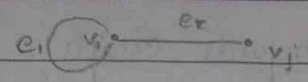
- If v_i and v_j denote 2 vertices of a graph and if e_k denotes an edge joining v_i and v_j then

v_i and v_j are called the End vertices of e_k

eg:-



- if an edge has same end vertices, then we call it as loop.



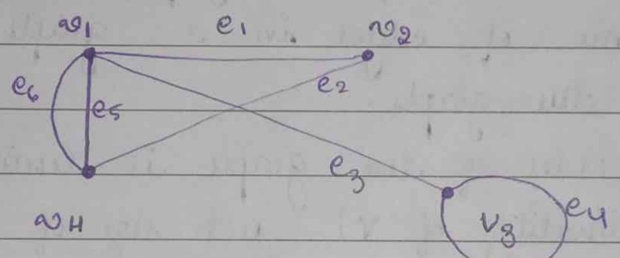
Parallel edges:

Edges which has same end vertices, we call it as parallel edges

Multiple edges:

If in a graph, there are 2 or more edges with the same end vertices are called as multiple edge.

eg:-



In the above figure:

e_4 - is a loop

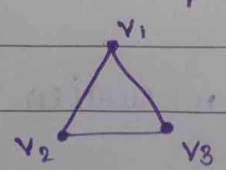
e_5, e_6 are parallel edge and also

e_5, e_6 are multiple edges.

Simple Graph / Multigraph / General graph:

- A graph which does not contain a loop and multiple edges is called simple graph

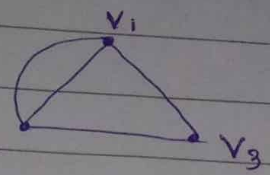
eg:



- A graph which does not contain a loop is called a loop free graph

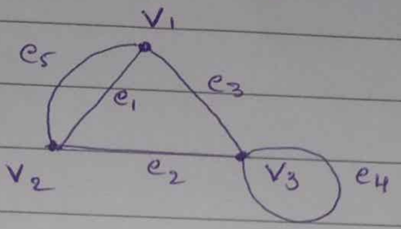
- A graph which contains multiple edges but no loops is called a <u>Multigraph</u>.

eg:-



- A graph which contains multiple edges or loops is called a <u>General Graph</u>

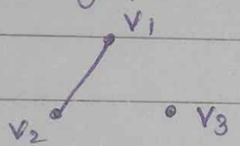
eg:-



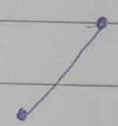
<u>Note</u>:

- If names are assigned to vertices of a graph, then the graph is called as <u>labelled graph</u> otherwise <u>unlabelled graph</u>.

eg:-



labelled graph

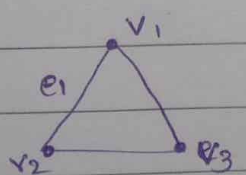


unlabelled

<u>Incidence</u>:

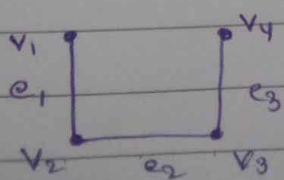
When a vertex v of a graph G is an end vertex of an edge e of the graph G, we say that the edge e is incident on the vertex v.

eg:-



e1 is incident on v2

e1 is incident on v1



in this we can say that e1 is incident on v2 ∵ v2 is one of its end vertex.

6/06/23

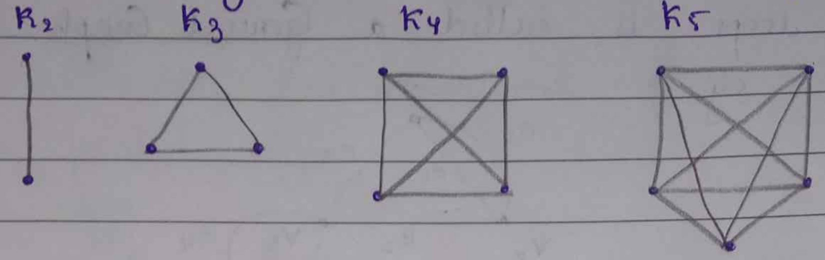
Complete Graph :

A simple graph of order ≥ 2 in which there is an edge between every pair of vertices is called a complete graph.

It is also called as full graph.

It is denoted by K_n where n is no. of vertices.

eg:



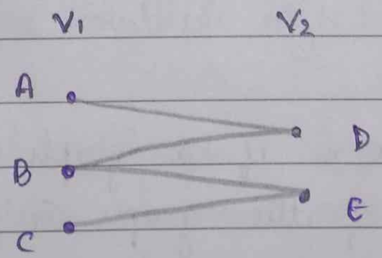
A complete graph with 5 vertices, that is, K_5 is called as Kuratowski's first graph.

imp

* Bipartite graph :

Suppose a simple graph G is such that its vertex set V is the union of 2 of its mutually disjoint non empty subsets V_1 and V_2 which are such that each edge in G joins a vertex in V_1 and the edge in V_2

eg:

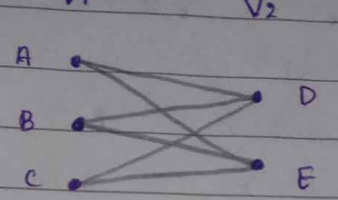


Bipartite graph is denoted by $\boxed{G(V_1, V_2 ; E)}$

Complete Bipartite graph :

A Bipartite graph $G(V_1, V_2 ; E)$ is called a complete bipartite graph, if there is an edge between every vertex of V_1 and every vertex in V_2.

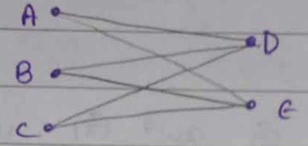
eg:- →



Note:

The complete Bipartite graph is denoted by $K_{r,s}$ where r is order of V_1 and s is order of V_2.

eg:-



$K_{3,2}$

- The complete bipartite graph $K_{3,3}$ is known as Kuratowski's second graph.

Problem:

1. Draw a diagram of the graph in each of the following cases.

i.) $V = \{A, B, C, D\}$, $E = \{AB, AC, AD, CD\}$

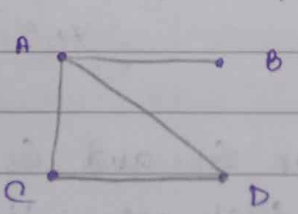
ii.) $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{v_1v_2, v_1v_3, v_2v_3, v_4v_5\}$

iii.) $V = \{P, Q, R, S, T\}$, $E = \{PS, QR, QS\}$

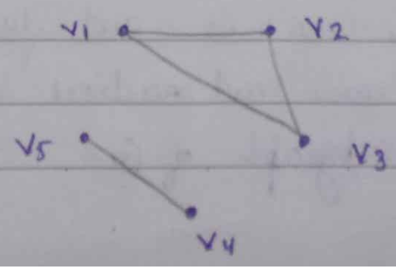
iv.) $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

$E = \{v_1v_4, v_1v_6, v_4v_6, v_3v_2, v_3v_5, v_2v_5\}$

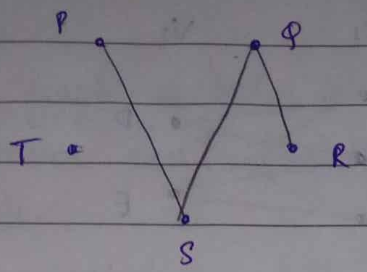
Ans: i.)



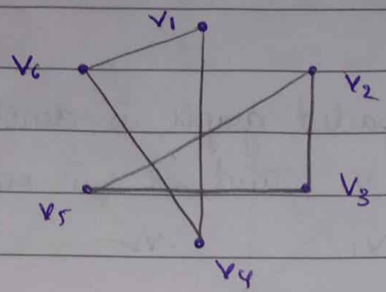
ii.)



iii.)



iv.)



7/06/23

Subgraphs:

Given 2 graphs G and G1, we say that G1 is a subgraph of G if the following conditions hold,

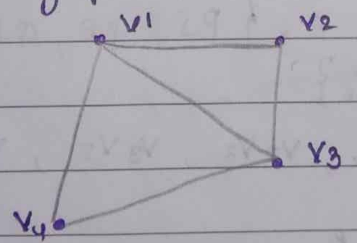
i.) All the vertices and all the edges of G1 are in G

ii.) Each edge of G1 has the same end vertices in G as in G1.

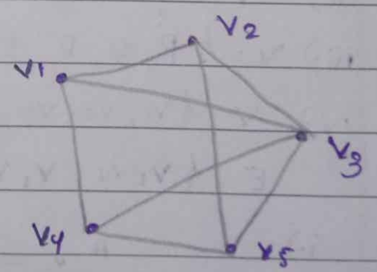
Note:

A subgraph is a graph which is a part of another graph.

eg:-



G_1



G

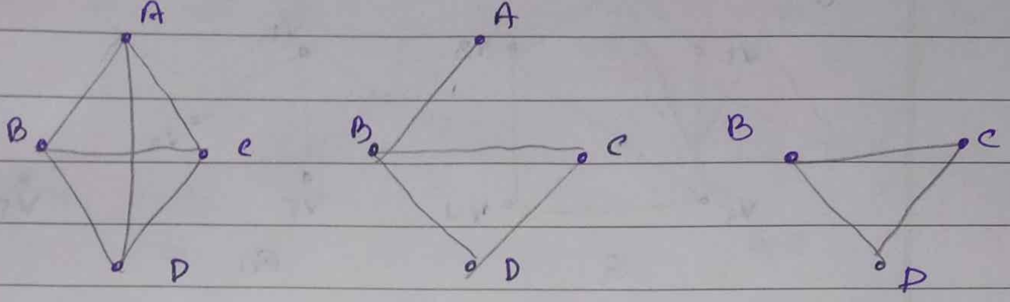
Consider 2 graphs G1 and G as shown in figure. we observe that all vertices and edges of the graph G1 are in G and that every edge in G1 has the same end vertices in G as in G1

∴ G1 is a subgraph of G.

→ **Spanning Subgraph:**

Given a graph $G = (V, E)$, if there is a subgraph $G_1 = (V_1, E_1)$ of G such that $V_1 = V$, then G_1 is called a spanning subgraph of G.

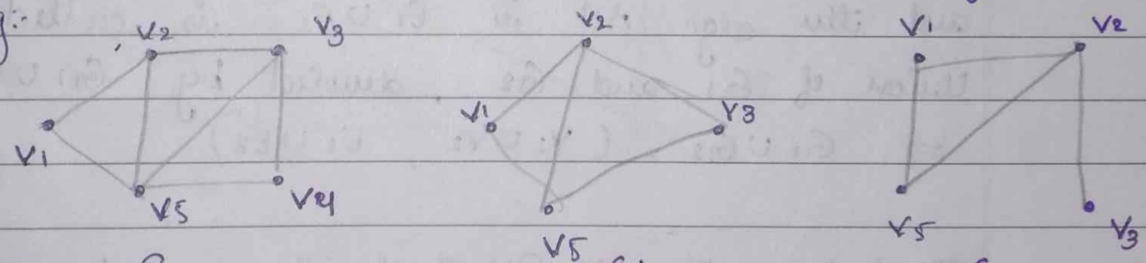
A subgraph G_1 of graph G is a spanning subgraph of G whenever G_1 contains all vertices of G.



→ **Induced Subgraph:**

Given a graph $G = (V, E)$, suppose there is a subgraph $G_1 = (V_1, E_1)$ of G such that every edge $\{A, B\}$ of G, where $A, B \in V_1$ is an edge of G_1 also, then G_1 is called an induced subgraph of G and is denoted by $\langle V_1 \rangle$

eg:-



G   G1   G2

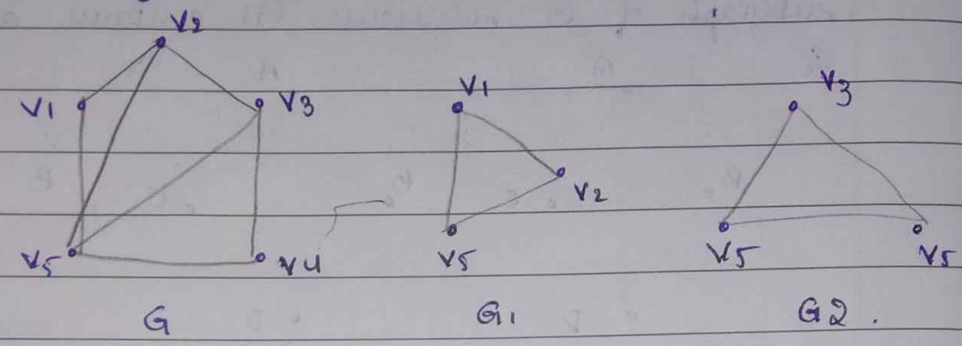
For the graph G, graph G_1 is an induced subgraph - induced by the set of vertices $V_1 = \{V_1, V_2, V_3, V_5\}$ whereas graph G_2 is not an induced subgraph.

Edge - disjoint and Vertex - disjoint Subgraphs:

Let G be a graph and G_1 and G_2 be 2 subgraphs of G. Then

1.) G_1 and G_2 are said to be edge-disjoint if they do not have any edge in common.
2.) G_1 and G_2 are said to be vertex-disjoint if they do not have any common edge and any common vertex.

eg:-



For the graph G, graphs G_1 and G_2 are edge-disjoint subgraphs.

9/06/23

Operations on Graphs:

Consider 2 graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$

Then the graph whose vertex set is $V_1 \cup V_2$ and the edge set is $E_1 \cup E_2$ is called the union of G_1 and G_2, denoted by $G_1 \cup G_2$

$\Rightarrow G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$

Similarly, if $V_1 \cap V_2 \neq \emptyset$, then graph whose vertex set is $E_1 \cap E_2$ is called intersection of G_1 and G_2, is denoted by $G_1 \cap G_2$.

$\therefore G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$ where $V_1 \cap V_2 \neq \emptyset$.

Suppose we consider the graph whose vertex set is $V_1 \cup V_2$ and edge set is $E_1 \Delta E_2$ where it Δ (Ring Sum) it is the symmetric difference of E_1 and E_2.

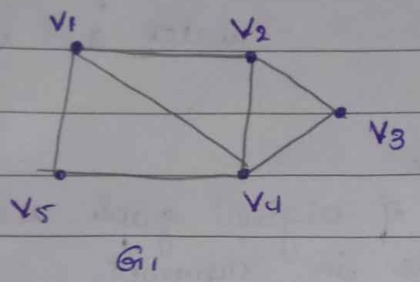
$E_1 \Delta E_2 = (E_1 \cup E_2) - (E_1 \cap E_2)$

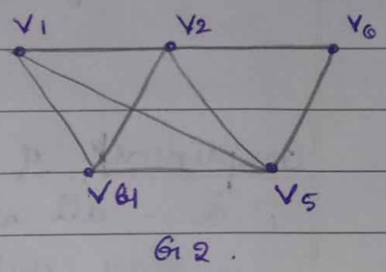
- the graph is called called ring sum of G_1 & G_2 & its denoted by $G_1 \Delta G_2$

$G_1 \Delta G_2 = (V_1 \cup V_2, E_1 \Delta E_2)$

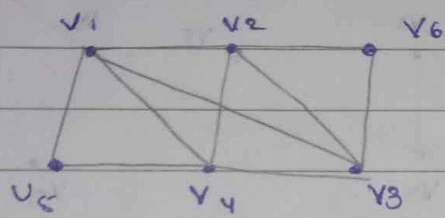
where $E_1 \Delta E_2 = (E_1 \cup E_2) - (E_1 \cap E_2)$.

eg:-

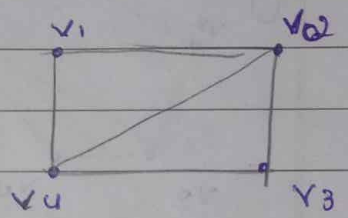




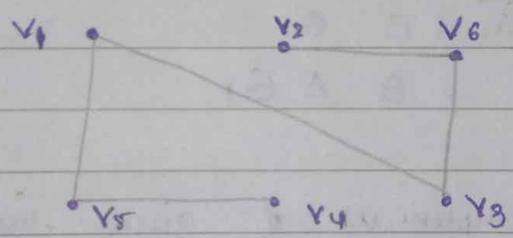
$G_1 \cup G_2$.



$G_1 \cap G_2$.



$G_1 \Delta G_2$



<u>Decomposition</u> :

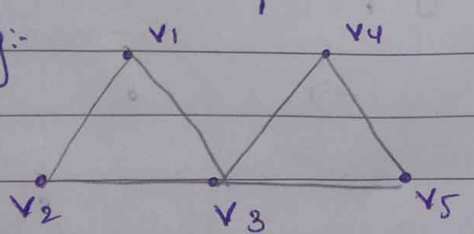
when $G_1 \cap G_2$ = NULL graph.

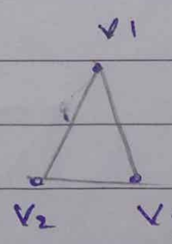
G_1 and G are edge disjoint

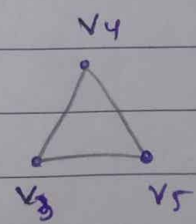
& $G_1 \cup G_2 = G$

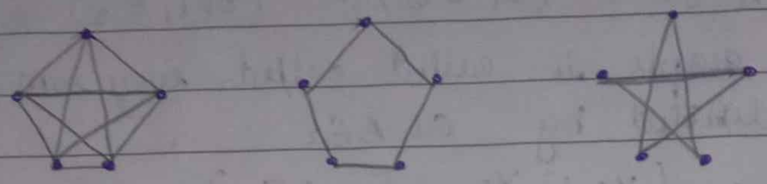
-> its decomposition.

eg:-







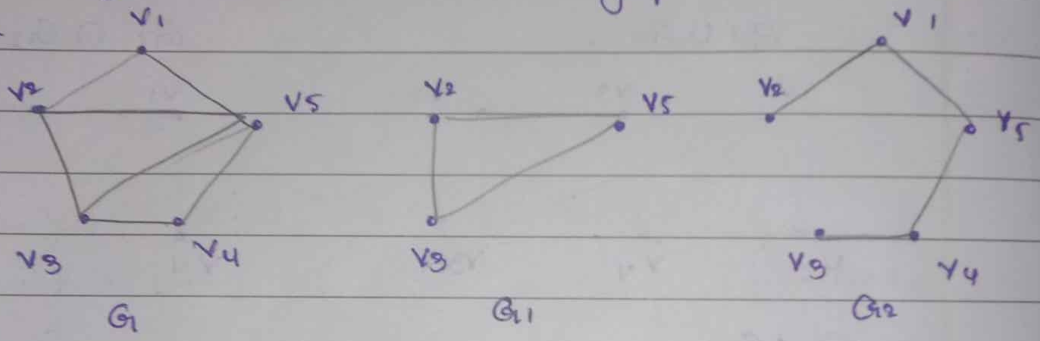


i.) when you delete edge → only edge (deleted).
ii.) when you delete vertex → corresponding edges of vertex and vertex (deleted)

Compliment of graph:

i) $\bar{G_1}$ - all vertices of original graph and all the edges not there in subgraph

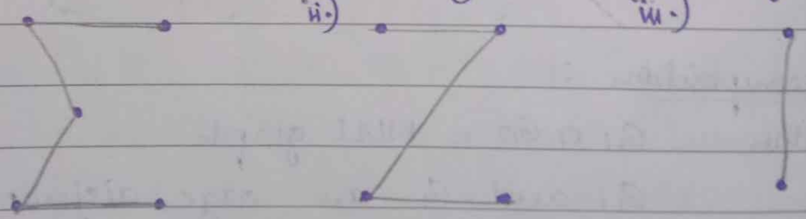
eg:-



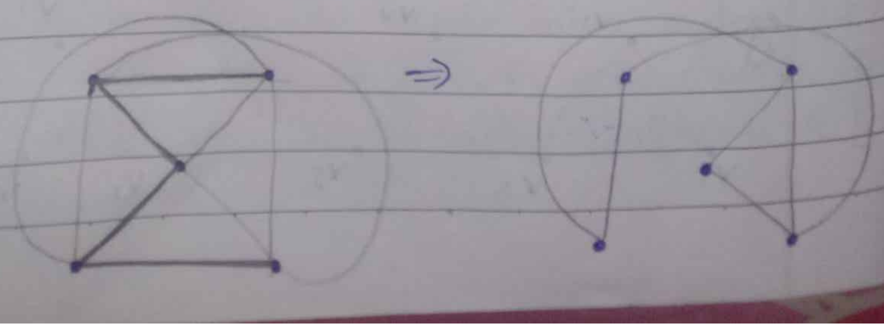
$$\bar{G_1} = G - G_1$$
$$= G \Delta G_1$$

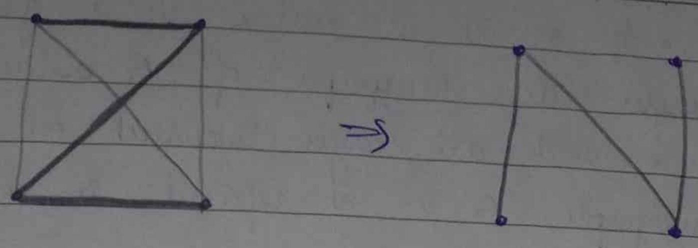
Q. Find the complement of given simple graph.

i.) ii.) iii.)

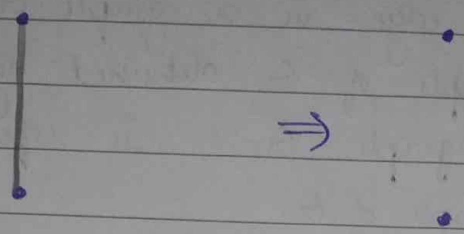


Ans: Consider the complete graph.



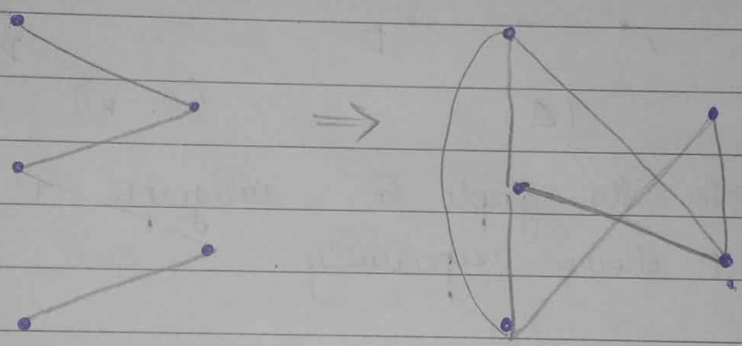


iii.)

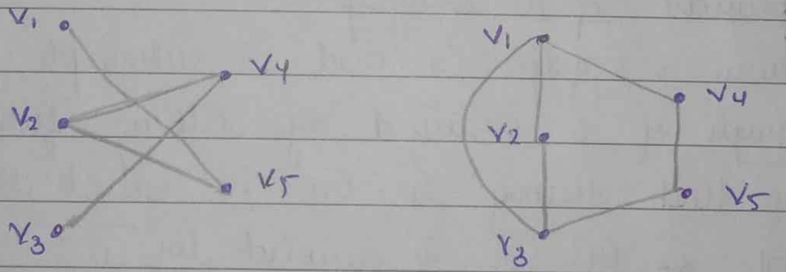


Q. Show that compliment of a Bipartite graph need not be a bipartite.

Ans



(or)



First figure shows a bipartite graph which is of order 5. The complement of this graph is the second figure, and it is not a bipartite graph.

Theory:

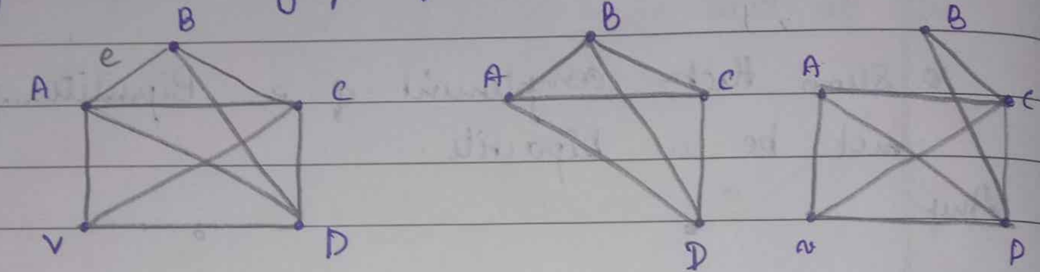
→ Decomposition:

We say that a graph G is decomposed into 2 subgraphs G1 and G2 if $G_1 \cup G_2 = G$ and $G_1 \cap G_2 =$ NULL graph.

Deletion: • If v is a vertex in a graph G, then $G-v$ denotes the subgraph of G obtained by deleting v and all edges incident on v, from G. This subgraph, $G-v$ is referred to a **vertex-deleted subgraph** of G.

• If 'e' is an edge in a graph G, then $G-e$ denotes the subgraph of G obtained by deleting e from G. This subgraph $G-e$ is refered to as **edge-deleted subgraph** of G.

eg:-



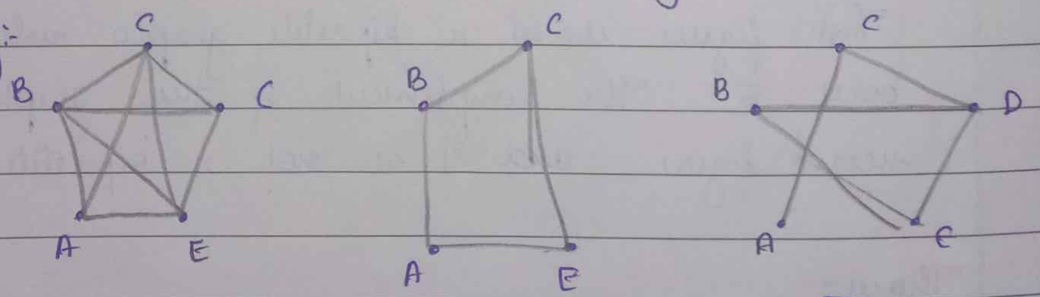
(G) $(G-v)$ $(G-e)$

For the graph G, subgraph $G-v$ and $G-e$ are shown respectively.

Complement of a Subgraph:

Given a graph G and a subgraph $G1$ of G, the subgraph of G obtained by deleting from G all the edges that belong to $G1$ is called the complement of $G1$ in G. It is denoted by $\overline{G_1}$.

eg:-



G G_1 $\overline{G_1}$

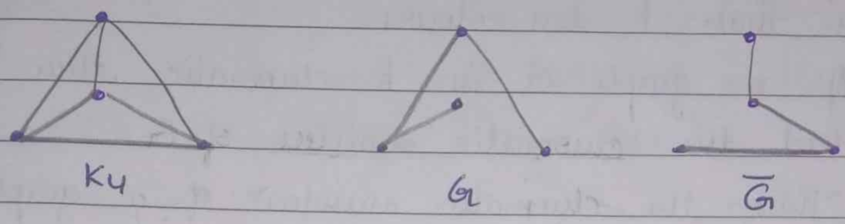
Consider graph G. Let $G1$ be subgraph of G, and then complement of $G1$ in G, namely $\overline{G_1}$ is shown.

Complement of a simple graph:

The complement $\overline{G}$ of a simple graph G with n vertices is that graph which is obtained by deleting those edges in K_n which belong to G.

Thus $\overline{G} = K_n - G = K_n \Delta G$.

eg:- Consider a complete graph K_4 and simple graph G of order 4. The complement, $\overline{G}$ of G is as shown.



Graph Colouring:

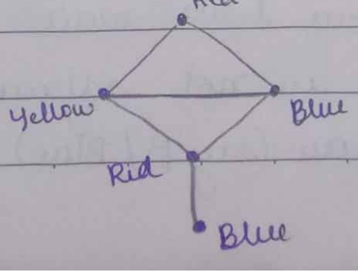
A graph which can be represented by atleast one plane drawing in which the edges meet only at the vertices is called a planar graph.

A graph which cannot be represented by a plane drawing in which the edges meet only at the vertices is called a non-planar graph.

Given a planar or non-planar graph G, if we assign colours to its vertices in such a way that no 2 adjacent vertices have the same colour, then we say that the graph G is properly coloured.

In other words, proper colouring of a graph means assigning colours to its vertices such that adjacent vertices have different colours.

eg:-



Chromatic Number :-

A graph G is said to be k-colourable if we can properly colour it with k (number of) colours. A graph G which is k-colourable but not (k-1) colourable is called a k-chromatic graph.

A k-chromatic graph is a graph that can be properly coloured with k-colours but not with less than k ~~ter~~ colours

If a graph G is k-chromatic, then k is called the chromatic number of G

Thus, the chromatic number of a graph is the minimum number of colours with which the graph can be properly coloured.

The chromatic number of a graph G is usually denoted by $\chi(G)$.

points :- if G_1 is a subgraph of G, then

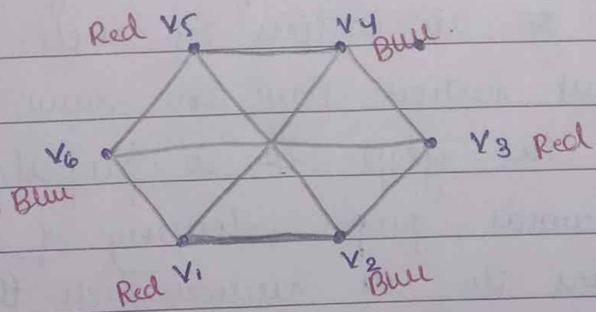
→ $\chi(G) \geq \chi(G_1)$

→ $\chi(K_n) = n$ (no. of vertices)

→ $\chi(G) \geq \chi(K_n) = n$.

Q. Find the chromatic number of each of these,

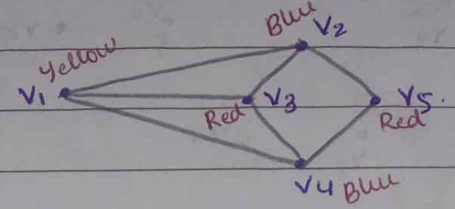
i.)



Ans:- In the above graph V_1, V_3, V_5 are not adjacent vertices so they can have same colour (say α/Red) then also V_2, V_4, V_6 are not adjacent vertices, can have same colour (say β/Blue) different from α

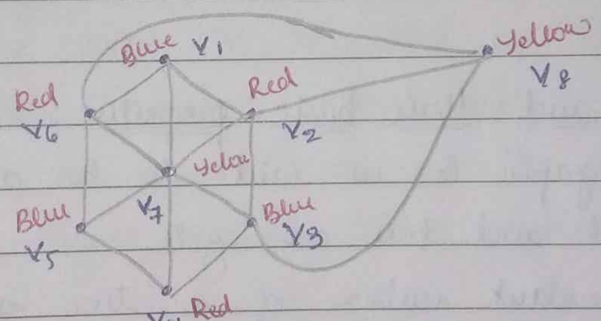
Hence we have used 2 colours to colour the graph
∴ Its 2 chromatic.

ii.)



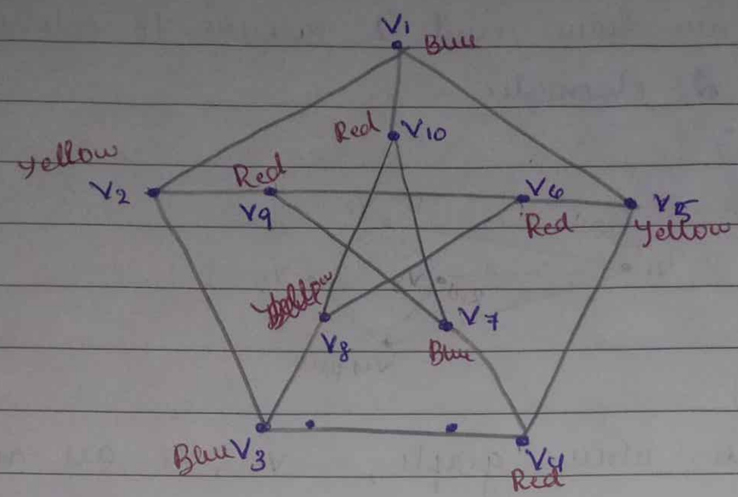
Ans: In the above graph, V_5, V_3 are not adjacent vertices, so they can the same colour (say Red), and V_2, V_4 are also not adjacent vertices so it can have same colour (say Blue), but V_1 is adjacent to V_2, V_3, V_4, so it will have a different colour (say yellow).
Hence we have used 3 colours to colour the graph
∴ Its 3 chromatic.

iii.)



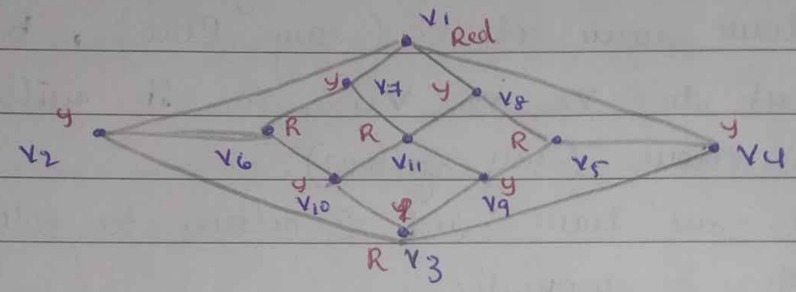
Ans: In the above graph, V_1, V_3, V_5 are not adjacent so they can have same colour (say Blue) and V_2, V_4, V_6 are not adjacent vertices so it can have same colour (say Red), ~~but~~ then V_7, V_8 are not adjacent vertices (so they can have a colour yellow.
Hence we have used 3 colours to colour the graph
∴ Its 3 chromatic.

iv.)



It is 3 chromatic.

v.)



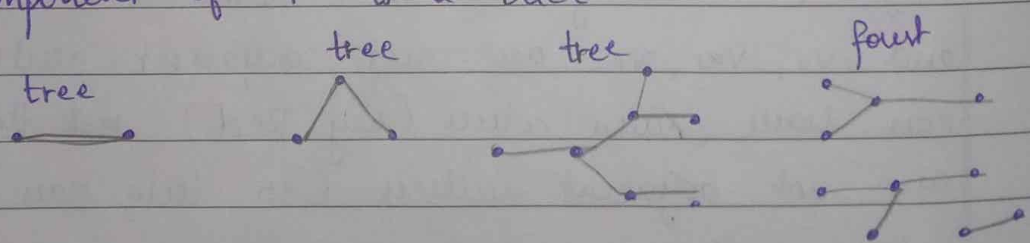
It is 2 chromatic.

Trees and their basic properties:

A graph G is said to be a tree if it is connected and has no cycle.

A pendant vertex of a tree is called a leaf.

A disconnect graph is said to be forest if each component of it is a tree.



- Theorem 1: In a Tree, there is one and only one path between every pair of vertices.
- Theorem 2: If a graph G there is one and only one path between every pair of vertices, then G is a tree.

- **Theorem 3**: A tree with n vertices has n-1 edges
- **Theorem 4**: A connected graph G is a tree if and only if adding an edge between any 2 vertices in G create exactly one cycle in it.
- **Theorem 5**: Any connected graph G with n vertices and (n-1) edges is a tree.

Minimally connected graph:

A connected graph is said to be minimally connected if the removal of any edge from it disjoints the graph.

Rooted Trees:

A directed tree is a directed graph whose underlying graph is a tree.

A directed tree T is called a Rooted Tree if

i.) T contains a unique vertex called the root, whose in-degree is equal to 0

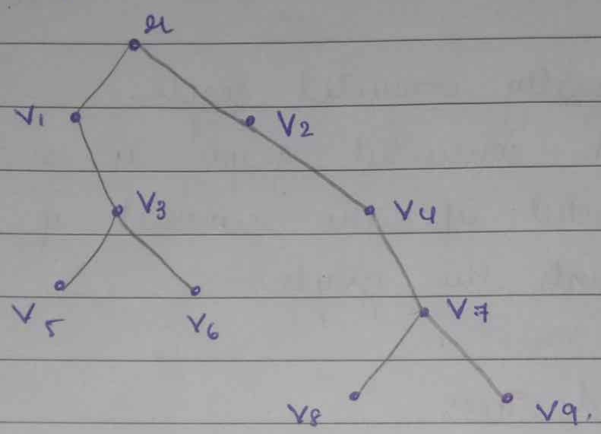
ii.) the in-degree of all other vertices of T are equal to 1.

- Root of a tree denoted by r.
- A vertex v (other than root r) of a rooted tree is said to be at k^{th} level or has level k, if the path from r to v is of length k.
- If v_1 and v_2 are 2 vertices such that v_1 has a lower level number, then we say that v_1 is an ancestor of v_2 or that v_2 is a descendant of v_1
- If v_1 and v_2 are 2 vertices such that v_1 has a lower level number and there is an edge from

v_1 to v_2, then v_1 is called the parent of v_2 or v_2 is called the child of v

- 2 vertices with a common parent are referred to as siblings.
- In a rooted tree a vertex whose out-degree is 0 is called a leaf and a vertex which is not a leaf is called an internal vertex.

eg:-



i.) v_1 and v_2 are at first level
v_3, v_4 are at second level
v_5, v_6, v_7 are at third level
v_8, v_9 are at fourth level.

ii.) v_1 is ancestor of v_3, v_5, v_6 or v_3, v_5, v_6 are the descendants of v_1; v_2 is the ancestor of v_4, v_7, v_8, v_9.

iii.) v_1 is parent of v_3 (or v_3 is child of v_1)

iv.) v_5 and v_6 are siblings.

v.) v_5, v_6, v_8, v_9 are leaves.

m-ary Tree:

A rooted tree T is called an m-ary tree if every internal vertex of T is of out-degree $\leq m$; that is if every internal vertex of T has at most m-children.

A rooted tree T is called a complete m-ary tree if every internal vertex of T is of out-degree m; that is every internal vertex of T ~~is of out-degree~~ has exactly m-children.

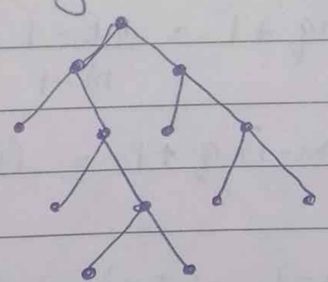
Binary-Tree:

An m-ary tree for which $m=2$ is called a binary tree. A rooted tree T is called a binary tree if every vertex of T is of out-degree ≤ 2 that is if every vertex has at most 2 children.

A complete m-ary tree for which $m=2$ is called a complete binary tree.

In other words, a rooted tree T is called a complete binary tree if every internal vertex of T is of out degree 2; that is if every internal vertex of T is of out-degree 2; that is if every internal vertex has exactly 2 children.

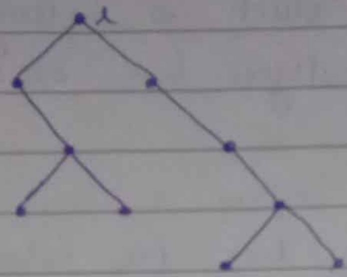
eg:- complete binary tree:



Balanced Tree:

If T is a rooted tree and h is the largest level number achieved by a leaf of T, then T is said to have height h. A rooted tree of height h is said to be balanced if the level number of every leaf is h or h-1.

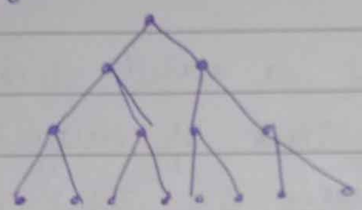
eg:- for balanced tree :



<u>Full binary Tree</u> :

Let T be a complete binary tree of height h. Then T is called a full binary tree if all the leaves in T are at level h.

eg: full binary tree :



<u>Note</u> : Let T be a complete m-ary tree of order n, with p leaves and q internal vertices then

ⓐ $n = mq + 1 = \frac{mp - 1}{m - 1}$

ⓑ $p = (m-1)q + 1 = \frac{(m-1)n + 1}{m}$

ⓒ $q = \frac{n-1}{m} = \frac{p-1}{m-1}$

Q. i) Find the no. of internal vertices in a complete 5-Ary tree with 817 leaves.

ii) Find the no of leaves in a complete 6-ary tree of order 733

Ans i) Given $m = 5$, $p = 817$, $q = ?$

$$q = \frac{n-1}{m} = \frac{p-1}{m-1}$$

$$= \frac{817-1}{5-1} = \frac{816}{4}$$

$$q = 204$$

There are 204 internal vertices

ii.) $m = 6$, order $= n = 733$ $p = ?$

$$p = (m-1)q + 1 = \frac{(m-1)n+1}{m}$$

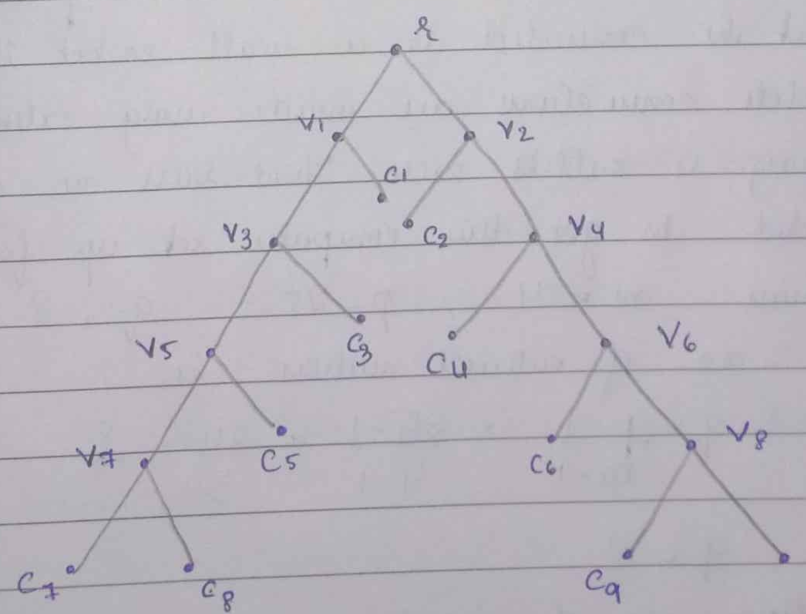
$$= \frac{(6-1)(733)+1}{6} = \frac{3665+1}{6} = \frac{3666}{6}$$

$$= \underline{\underline{611}}$$

$\therefore$ There are 611 leaves.

Q The computer lab of a school has 10 computers that are to be connected to a wall socket that has 2 outlets. Connections are made using extension chords that have 2 outlets each. Find the least no. of chords needed to get those computers set up for use.

Ans:



(complete 2-ary tree)

$\Rightarrow$ $m = 2$ $p = \text{leaves} = 10$

(8 extensions - from tree)

$$q = \frac{n-1}{m} = \frac{p-1}{m-1} = \frac{10-1}{2-1} = 9$$

$q = 9$ (internal vertices)

$q - 1$ (root) $= 9 - 1 = \underline{8 \text{ chords}}$.

The wall socket may be regarded as the root of the complete Binary Tree / complete 2 ary Tree with the computer as its leaves and its internal vertices other than the root as extension chords.

Thus, here $m = 2$ and no. of leaves (no. of computers) $p = 10$

$\therefore$ The no. of internal vertices is

$$q = \frac{p-1}{m-1} = \frac{10-1}{2-1} = \frac{9}{1} = \underline{9}$$

Hence the no. of extension chord needed is

$q - 1 = 9 - 1 = \underline{8}$

Q. A classroom contains 25 micro computers that must be connected to a wall socket that has 4 outlets connections are made using extension chords having 4 outlets each. Find least no. of chords needed to get this computer set up for the class.

Ans: Given $m = 4$, $p = 25$ $q = ?$

$\therefore$ no. of internal vertices is

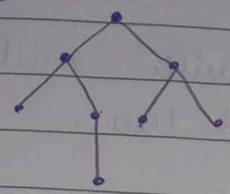
$$q = \frac{p-1}{m-1} = \frac{25-1}{4-1} = \frac{24}{3} = \underline{8}$$

$q = 8$

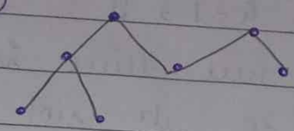
Hence no. of chords $\Rightarrow q - 1 = 8 - 1 = \underline{7}$

Q. Which of the following is rooted tree.

(a)

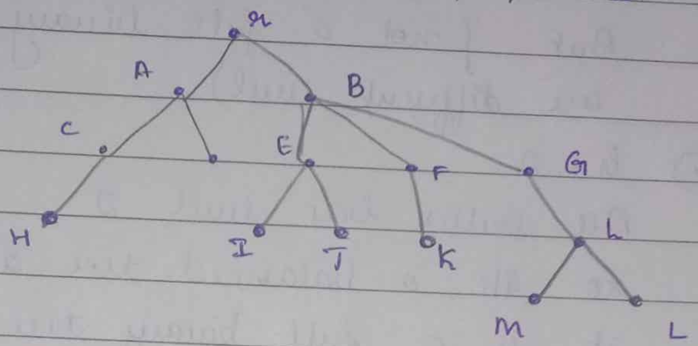


(b)



↑ this is rooted tree

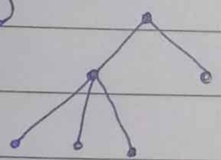
Q. Find levels of vertices of A, H, F, M in



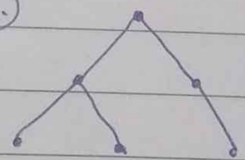
Ans: Level of A = 1, H = 3, F = 2, M = 4

Q. Which of the following is binary tree / complete binary tree

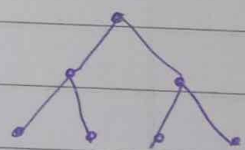
(a)



(b)



(c)



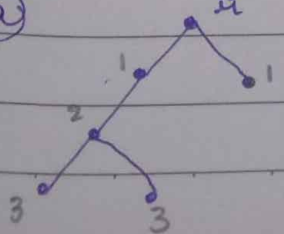
Ans: (a) It's not a binary tree

(b) It is a binary tree but not complete binary tree.

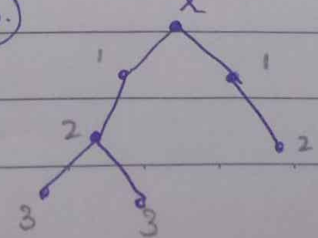
(c) It is a complete binary tree.

Q. Which of the following is balanced tree or Full binary tree.

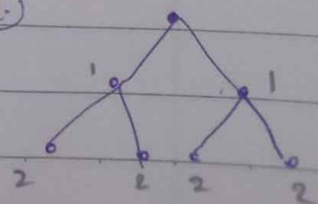
(a)



(b)



(c)



Ans (a) $h = 3$

$h - 1 = 2$

here there is a leaf with level 1, so its not a balanced tree.

(b.) $h = 3$

$h - 2 = 2$

All leaf has level either 2 or 3, So its balanced tree.

But not a full binary tree (as there are different level)

(c.) $h = 2$

All leaves has level 2, so its a balanced tree as well as it is a full binary tree.

Ans (a) h = 3
h - 1 = 2
here there is a leaf with level 1,
so its not a balanced tree.

(b) h = 3
h - 2 = 2
All leaf has level either 2 or 3,
So its balanced Tree.
But f not a full binary tree (as there are different level)

(c) h = 2
All leaves has level 2,
so its a balanced tree as well as it is a full binary tree.

20/6/23

Sorting:

i. **Merge Sort:**

The splitting and merging process is done by the use of balanced complete tree. This method of sorting a list is known as Merge Sort.

→ eg:- using merge sort method, sort the list
→ 7, 3, 8, 4, 5, 10, 6, 2, 9

- First we recursively split the given list and all subsequent lists in half or as close as possible to half, until each sublist contain a single element.
- Next we merge the sublist in non decreasing order until the items in the original list have been sorted.

7, 3, 8, 4, 5, 10, 6, 2, 9

{7, 3, 8, 4, 5}, {10, 6, 2, 9}

{7, 3, 8}, {4, 5} {10, 6}, {2, 9}

{7, 3}, {8} {4}, {5} {10}, {6} {2}, {9}.

{7}, {3} 8 4 5 10 6 2 9

7 3

7 3 8 4 5 10 6 2 9

{3, 7} {4, 5} {6, 10} {2, 9}

{3, 7, 8} {2, 6, 9, 10}

{3, 4, 5, 7, 8}

{2, 3, 4, 5, 6, 7, 8, 9, 10}

Thus the sorted list is as written above.

Q. Apply merge sort to the list.

(i) −1, 7, 4, 11, 5, −8, 15, −3, −2, 6, 10, 3

(ii) 6, 2, 7, 3, 4, 9, 5, 1, 8.

(iii) −9, 6, 5, −3, 4, 2, −7, −5, 10, −11, 0, 1.

Ans: (i) -1, 7, 4, 11, 5, -8, 15, -3, -2, 6, 10, 3

{-1, 7, 4, 11, 5, -8}, {15, -3, -2, 6, 10, 3}

{-1, 7, 4}, {11, 5, -8} {15, -3, -2}, {6, 10, 3}

{-1, 7}, {4} {11, 5}, {-8} {15, -3}, {-2} {6, 10}, {3}

{-1} {7} 4 {11}, {5} -8 {15}, {-3} -2 {6} {10} 3

-1 7 11 5 15 -3 6 10

-1 7 4 11 5 -8 15 -3 -2 6 10 3

{-1, 7} {5, 11} {-3, 15} {6, 10}

{-1, 4, 7} {-8, 5, 11} {-3, -2, 15} {3, 6, 10}

{-8, -1, 4, 5, 7, 11} {-3, -2, 3, 6, 10, 15}

{-8, -3, -2, -1, 3, 4, 5, 6, 7, 10, 11, 15}

(ii) ~~6~~ 6, 2, 7, 3, 4, 9, 5, 1, 8

{6, 2, 7, 3, 4} {~~4~~, 9, 5, 1, 8}

{6, 2, 7}, {3, 4} {9, 5}, {1, 8}

{6, 2}, {7} {3}, {4} {9}, {5} {1}, {8}

{6}, {2} 7 3 4 9 5 1 8

6 2

6 2 7 3 4 9 5 1 8

{2, 6} {3, 4} {5, 9} {1, 8}

{2, 6, 7} {1, 5, 8, 9}

{2, 3, 4, 6, 7}

{1, 2, 3, 4, 5, 6, 7, 8, 9}

(iii) -9, 6, 5, -3, 4, 2, -7, -5, 10, -11, 0, 1.

{-9, 6, 5, -3, 4, 2}, {-7, -5, 10, -11, 0, 1}

{-9, 6, 5}, {-3, 4, 2} {-7, -5, 10}, {-11, 0, 1}

{-9, 6}, {5} {-3, 4}, {2} {-7, -5}, {10} {-11, 0}, {1}

{-9}, {6} 5 {-3}{4} 2 {-7}{-5} 10 {-11}, {0} 1

-9 6 -3 4 -7 -5 -11 0

-9 6 5 -3 4 2 -7 -5 10 -11 0 1

{-9, 6} {-3, 4} {-7, -5} {-11, 0}

{-9, 5, 6} {-3, 2, 4} {-7, -5, 10} {-11, 0, 1}

{-9, -3, 2, 4, 5, 6} {-11, -7, -5, 0, 1, 10}

{-11, -9, -7, -5, -3, 0, 1, 2, 4, 5, 6, 10}

Prefix codes and weighted trees :

A sequence is a set whose elements are listed in order as the first element, second, third element and so on.

The number of elements contained in a sequence is called its length

eg: anvitha - 7

A sequence consisting only 0 and 1 is called a binary sequence or a binary string

eg: 01, 001, 101, 11001, 1000100.

are binary sequences with length : 2, 3, 3, 5, 7 respectively.

21/06/23

Prefix code :

Let P be a set of binary sequences that represents a set of symbols.

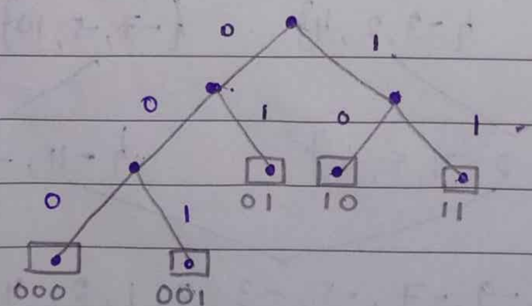
Then P is called a Prefix code, if no sequence in P is the prefix of any order sequence in P.

Note: Prefix codes can be represented by binary tree.

Consider a prefix code

P = {000, 001, 01, 10, 11}.

In this code, longest sequence has length 3, then in full binary tree will have a height 3.



/ /

<u>Note</u> : Consider a code,

P : a : 1 e : 0 n : 10 r : 01 t : 101

- in this code, we can represent 'eat' as 01101 when we decode it, we get words 'eat', 'rt', 'eana', 'rna', etc, since its not a prefix code.

Consider a code,

P : a : 10 e : 0 n : 1101 r : 111 t : 1100

which is a prefix code.

By using this, we can represent 'eat' as 0 10 1100,

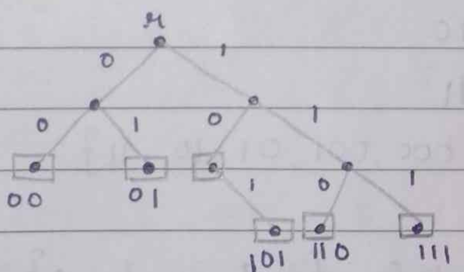
When we decode it, we will get the same word 'eat'.

<u>Problems</u> :

Q. Construct the binary tree that represent the prefix code.

P = {00, 01, 101, 110, 111}

Ans:



Q. Consider the prefix code a : 111, b : 0, c : 1100, d : 1101, e : 10, using this code decode the following sequences.

i.) 1001 1111 01

ii.) 1011110011000 11 01

iii.) 11 01111110010

Ans i.) 10|0|1 11|1101|

e b a d → ebad

ii.) 10|111|1 0|0|1 1 000|0|1101|

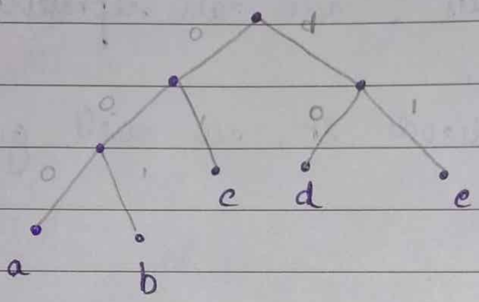
e a e b c b d

→ eaebcbd

iii.) 11 0 1|1 11|11 00|10

d a c e → dace.

Q. Obtain the prefix code represented by the following binary tree.



Ans $a = 000$

$b = 001$

$c = 01$

$d = 10$

$e = 11$

$P = \{000, 001, 01, 10, 11\}$

Q. A code for $\{a, b, c, d, e\}$ is given by,
$a: 00$, $b: 01$, $c: 101$, $d: x10$, $e: yz1$
where $x, y, z \in \{0, 1\}$,
Determine x, y, z, so that given code is a prefix code.

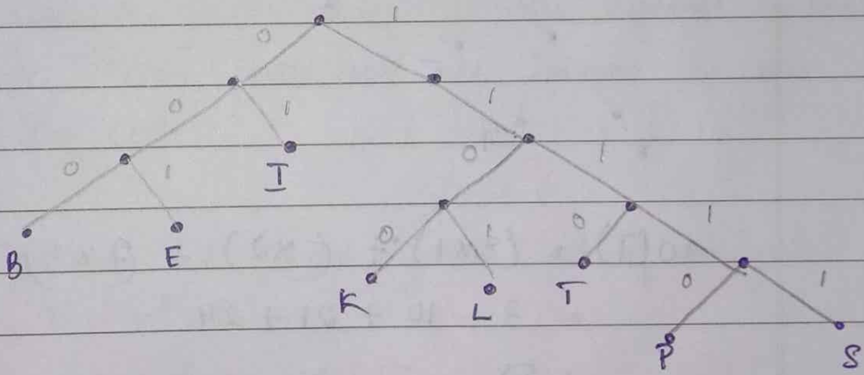
Ans $x = 1$

$y = 1$

$z = 1$

If we put $x = 0$, the code will not become prefix code, since 1 sequence b is prefix of d
$\therefore$ x can have value 1.
Suppose we put $y = 0$, $z = 0$, or $y = 0, z = 1$ or $y = 1, z = 0$, the code will not become prefix code.
Hence y and z can have value 1, 1 respectively.

Q. Find the prefix code for the letters B, E, I, K, L, T, P, S if the coding scheme is as shown in the figure. Hence,
i.) Find the code for the word ~~and~~ PIPE and BEST.
ii.) Decode the strings a.) 0000 1110 0001
b.) 111 111 11 0110 1011110



Ans B = 000, E = 001, I = 01, K = 1100, L = 1101,
T = 1110, P = 11110, S = 11111
i.) PIPE – 11110 01 11110 001
BEST – 000 001 11111 1110
ii.) (a) 000|0 1|110 0|001|
B I K E = BIKE
(b) 11111|111 0|11 01|01|1110|
S T L I T = STLIT.

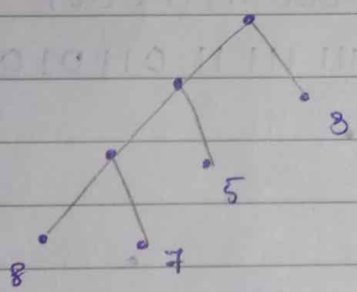
Weighted tree :

Consider a set of n positive integers, $w_1, w_2, w_3, \ldots, w_n$, where $w_1 \leq w_2 \leq w_3 \leq \ldots \leq w_n$.

Suppose we assign these integers to the n leaves of a complete binary tree $T = (V, E)$ in any one to one manner. The resulting tree is called a complete, weighted, binary tree with $w_1, w_2, \ldots, w_n$ as weights.

If $l(w_i)$ is the level number of the leaf of T to which the weight w_i is assigned then $\boxed{w(T) = \sum_{i=1}^{n} w_i \times l(w_i)}$ is called the weight of the tree.

eg:-



$$w(T) = (3\times1) + (5\times2) + (7\times3) + (8\times3)$$
$$= 3 + 10 + 21 + 24$$
$$= \underline{58}$$

Optimal Tree :

Given a set of weight. Suppose we consider, set of all complete binary tree, to whose leaves these weights are assigned. A tree in this set which carries the minimum weight is called an optimal tree for the weights.

Note: Optimal Tree is also called as **Huffman's Tree**

24/06/23

~~Optimal Prefix code~~ :

The process of finding optimal Tree is called as Huffman's process.

Optimal Prefix code:

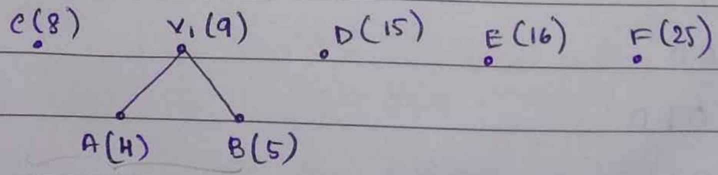
Huffman Tree (Optimal tree) can be used to obtain a prefix code for the symbols representing its leaves. For this purpose we first label symbols 0 and 1 to its edges by the labelling procedure indicated earlier. Then all the vertices and the leaves of the tree can be identified by binary sequences.

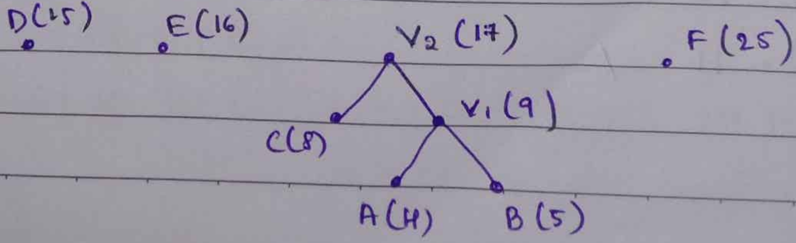
The binary sequences through which the leaves are identified gives a prefix code for the symbol representing these leaves. This prefix code is known as an Optimal Prefix Code.

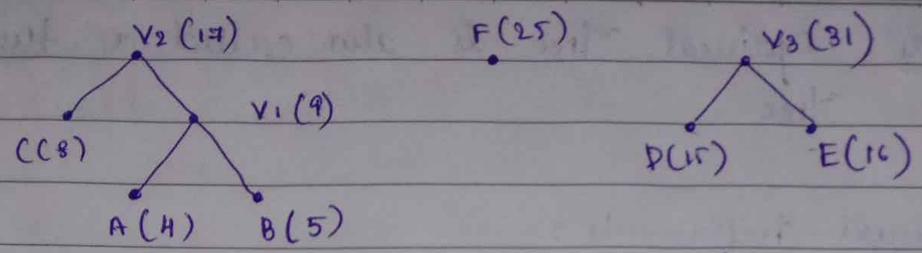
Q. Construct Optimal Tree for a given set of weights.
{ 4, 15, 25, 5, 8, 16}

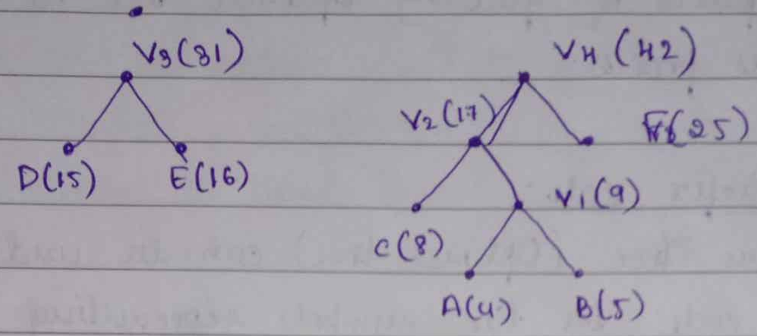
Ans: Arrange in ascending order:

4	5	8	15	16	25
A	B	C	D	E	F

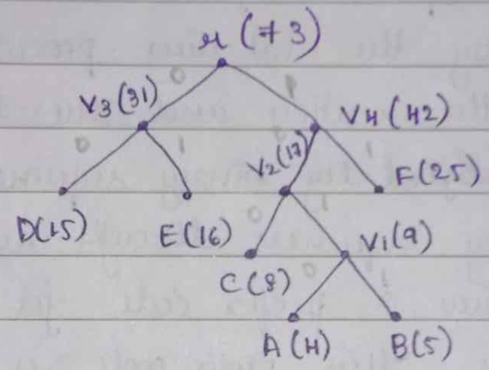








⇒



$w(T) = (4 \times 4) + (5 \times 4) + (8 \times 3) + (15 \times 2) + (16 \times 2)$
$+ (25 \times 2)$
$= 16 + 20 + 24 + 30 + 32 + 50$
$= \underline{172}$

Prefix codes :

D(15) = 00

E(16) = 01

C(8) = 100

A(4) = 1010

B(5) = 1011

F(25) = 11

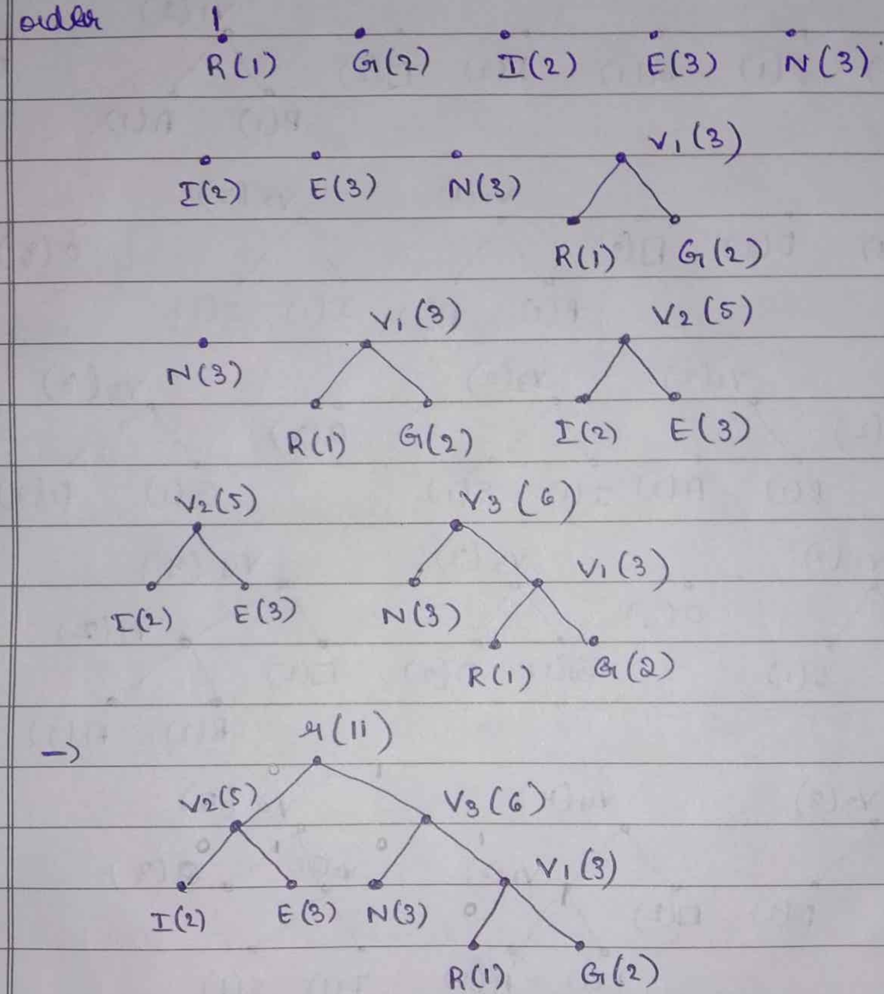
(same nos take left side)

Q. Construct an optimal prefix code for the letters of the word : ENGINEERING.
Hence deduce the code for this word.

Ans. weights → frequency of letters.

E	N	G	I	R
3	3	2	2	1

order



Prefix codes :
- I : 00
- E : 01
- N : 10
- R : 110
- G : 111

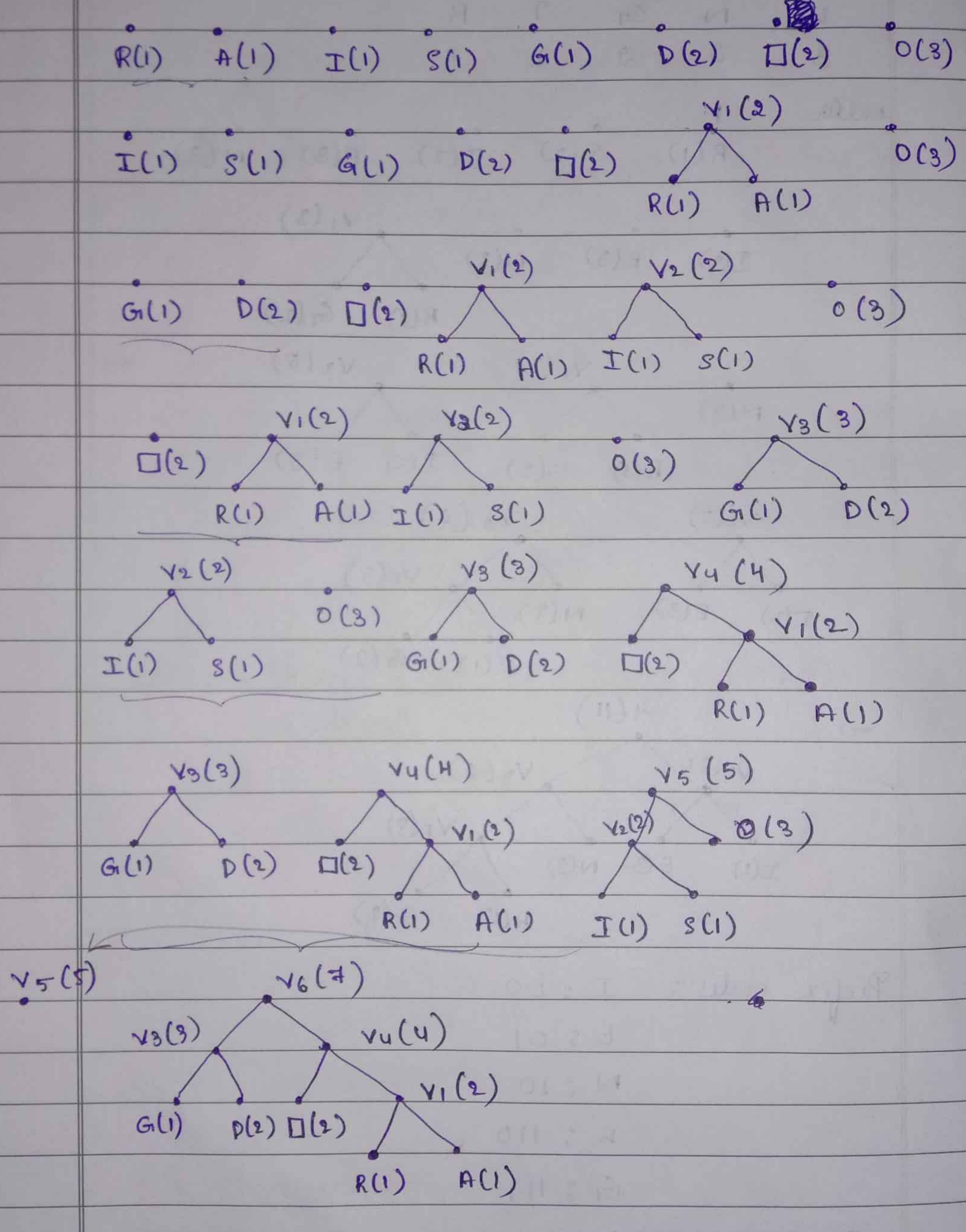
ENGINEERING

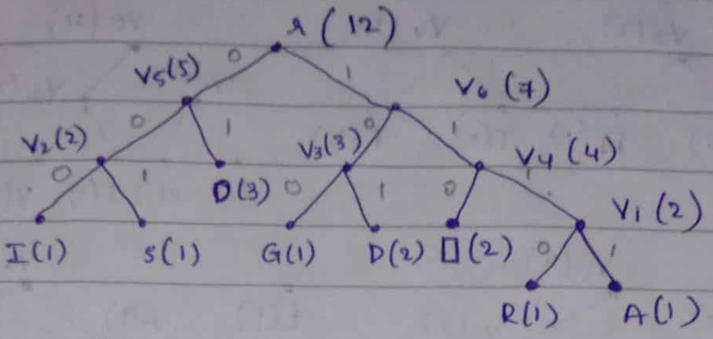
⇒ 01 10 111 00 10 01 01 110 00 10 111

Q. Obtain an optimal prefix code for the message ROAD IS GOOD. Indicate the code.

Ans

R	O	A	D	I	S	G	□ (space)
1	3	1	2	1	1	1	2





Prefix codes : I : 000   D : 101

S : 001   □ : 110

O : 01   R : 1110

G : 100   A : 1111

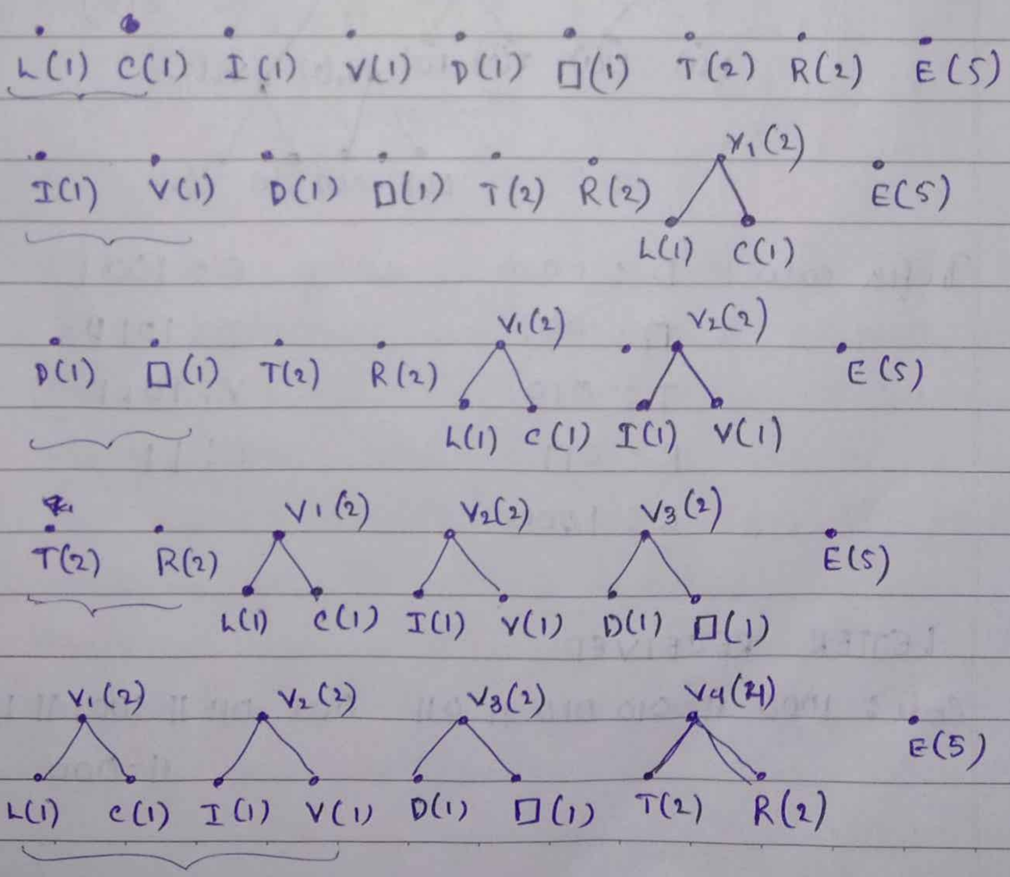
ROAD IS GOOD

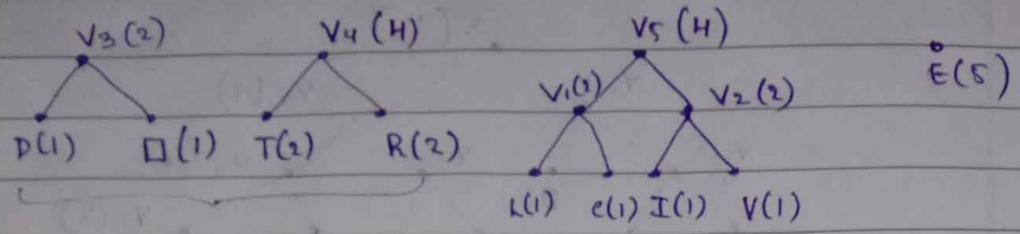
Code : 111001 1111 101 ~~000~~ 110 000 001 110 100 01 01 101

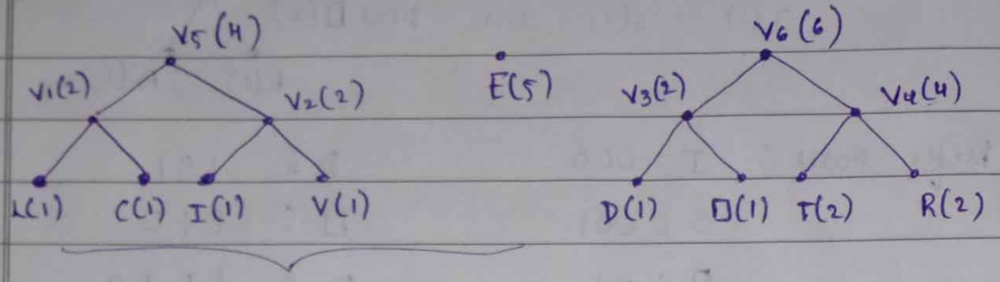
Q Obtain an optimal prefix code for the Message :
LETTER RECIEVED . Indicate the code.

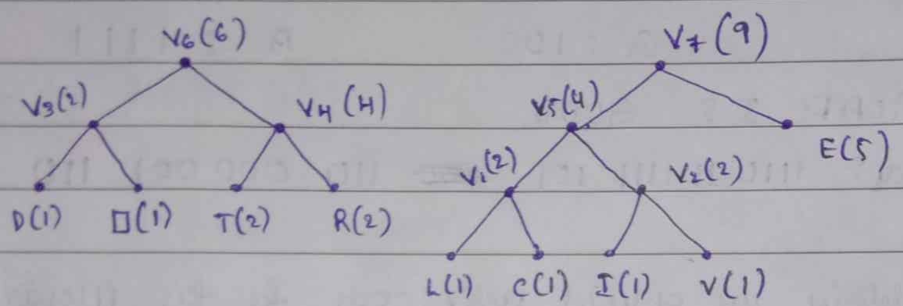
Ans

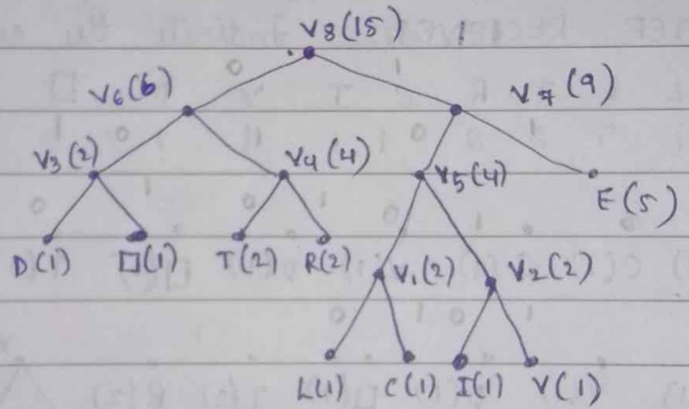
L	E	T	R	C	I	V	D	□
1	5	2	2	1	1	1	1	1











Prefix codes : D : 000 C : 1001
□ : 001 I : 1010
T : 010 V : 1011
R : 011 E : 11
L : 1000

LETTER RECEIVED

Code : 1000 11 010 010 11 011 001 011 11 1001 11 1010 1011
11·000

/ /

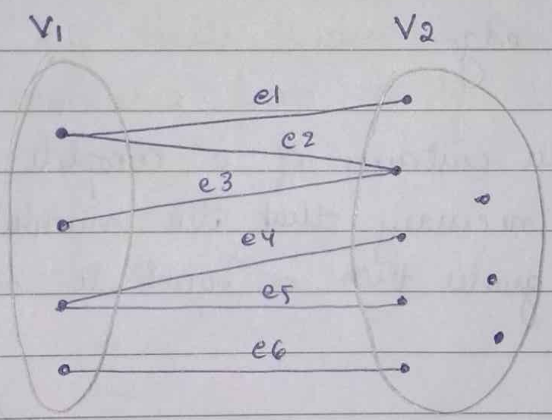
Matching :

Consider a bipartite graph $G(V_1, V_2 ; E)$ where V_1 and V_2 are as usual the 2 cartesians of the vertex set of the graph G, and E is the edge set of G. In this graph, every edge has one end vertex in V_1 and the other end vertex V_2.

A subset $M(E)$ is called a Matching in G if no 2 edges that belong to M share a common vertex in $V1$ or $V2$.

A matching M in G is called a complete matching from V_1 to V_2, if every vertex in V_1 is an end vertex of some edge that belongs to M

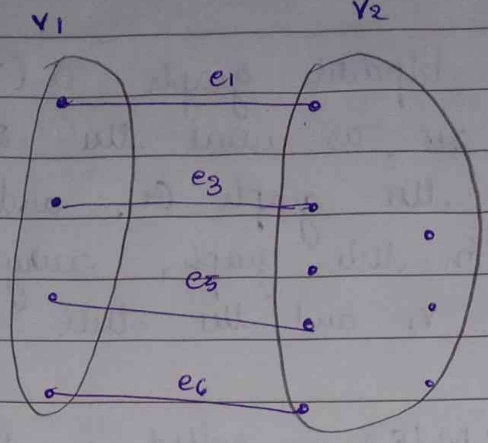
eg:-



In this graph, the set $M = \{e_1, e_3, e_5, e_6\}$ is a matching and it is a complete matching from $V1$ to $V2$ since it includes all the vertices of $V1$, but $M_2 = \{e_2, e_4, e_6\}$ is a matching but not a complete matching since it doesnot include all the vertices of $V1$

~~Comple~~

Complete matching is shown separately.



→ In a complete matching from V_1 and V_2, corresponding to every vertex in V_1, there must exist a unique vertex in V_2 such that there is an edge between the 2 vertices. But corresponding to a vertex in V_2 there may not be a vertex in V_1 to which it is joined by an edge.

→ For the existance of a complete matching V_1 to V_2 it is necessary that the number of vertices in V_2 is greater than or equal to number of vertices in V_1.

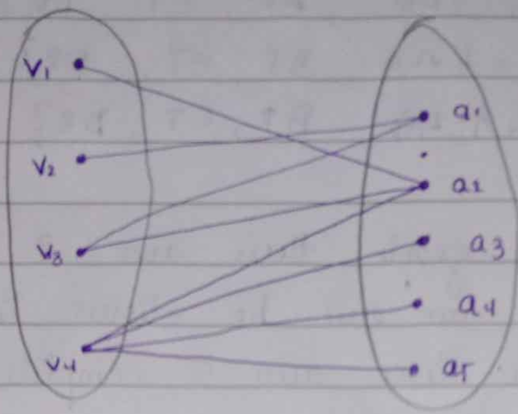
→ A complete matching from V_1 to V_2 does not exist if V_2 has lesser number of vertices than V_1.

imp.

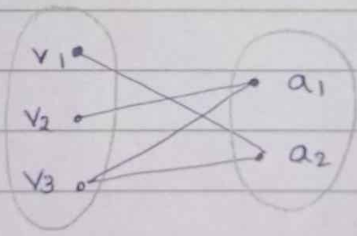
Hall's Theorem I:

In a bipartite graph G, $G(V_1, V_2; E)$ a complete matching from V_1 to V_2 exists if and only if, every subset of k vertices in V_1 is collectively adjacent to k or more vertices in V_2 for all possible values of k.

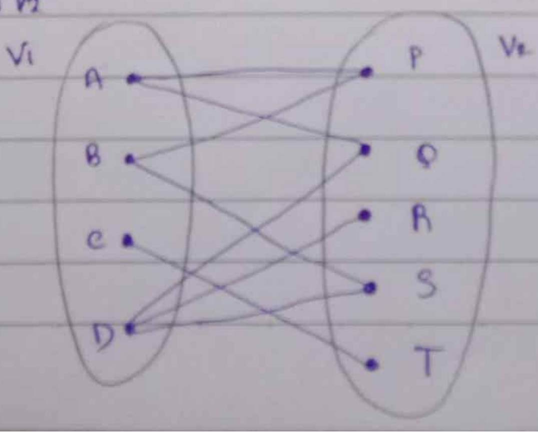
Q. Prove that the Bipartite graph shown in the figure does not have a complete matching from V_1 to V_2.



Ans: We observe that the 3 vertices v_1, v_2, v_3 in V_1 are together joined to 2 vertices a_1, a_2 in V_2. Thus, there is a subset of 3 vertices in V_1 which is collectively adjacent to 2 vertices in V_2. Hence by Hall's theorem, there does not exist a complete matching from V_1 to V_2.



Q. Consider the bipartite graph shown below. If 4 edges of this graph are chosen randomly, what is the probability that they form a complete matching from V_1 to V_2



Ans

$M1 = \{AP, BS, CT, DR\}$
$M2 = \{AP, BS, CT, DQ\}$
$M3 = \{AQ, BP, CT, DR\}$
$M4 = \{AQ, BS, CT, DR\}$
$M5 = \{AQ, BP, CT, DR\}$

In this graph, there are 8 edges out of which 4 edges can can be chosen in $^8C_4$ ways.
In this there are 5 complete matchings, i.e. M1, M2, M3, M4, M5.
Hence the probability that they form a complete matching is $P = \frac{5}{^8C_4} = \frac{5}{70} = \frac{1}{14}$