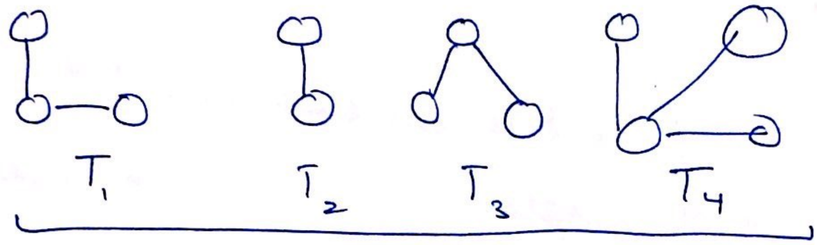
Trees. ch-9. (Rosen) [only 9.1 & 9.4]
`Introduction'

Trees:- A tree is a connected undirected graph with no simple circuits.

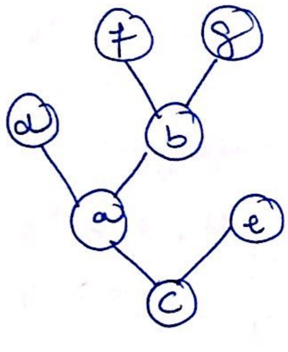
forests:- Each of the connected component is a tree.



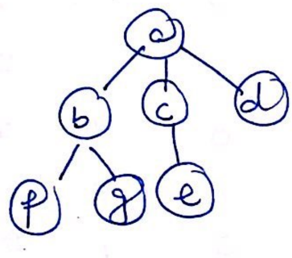
forest.

rooted tree:- is a tree in which one vertex has been designated as the root & every edge is directed away from the root.

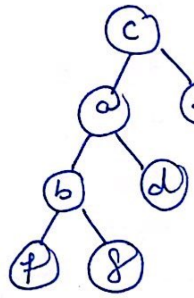
eg:- Tree (Graph)



rooted Tree with root 'a'



rooted tree with root 'c'



Th. An undirected Graph is a tree iff there is a unique path b/w any two of its vertices.

Proof:- A tree itself is a Graph. In a connected graph, there is atleast one path b/w every two vertices. If there are two or more than two paths, then their would be a circuit, which is not allowed in Tree. (acc. to the def^n of Trees).

Trees.

Def^n:- A tree is an undirected graph G which is connected & has no simple circuit.

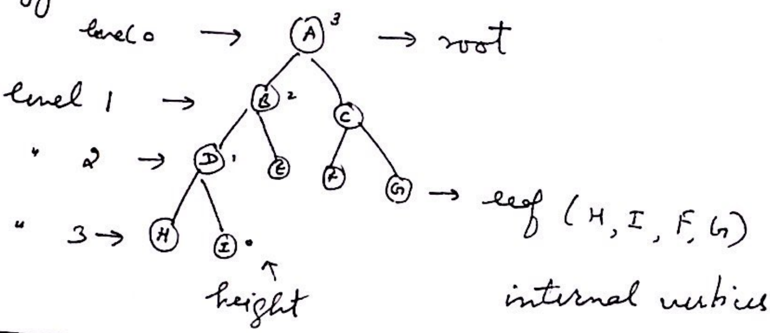
Properties

1). There is a unique path b/w every 2 vertices in a tree.

2). The no. of vertices is one more than the no. of edges in a tree. ie. $[E = V - 1]$

3). A tree with 2 or more vertices has atleast 2 leaves.

tree terminology



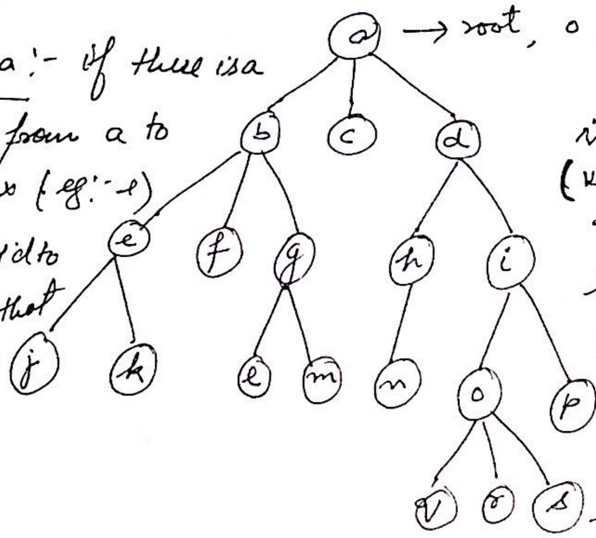
internal vertices = B, C D.

siblings = vertices having same Parent
(H,I) (D,E) (F,G) (B,C)

$\boxed{\text{largest level No = height of tree}}$

eg:- Find root (a), internal vertices & leaves of the tree. Identify Parent of h (d), siblings of o (p), ancestors of m (b,a). What are level-3 vertices. (j, k, ... o, p)

descendent of a:- if there is a directed path from a to any other vertex (eg:- e)
Also, 'a' is said to be ancestor of that vertex (eg:- e).



→ root, 0 incoming edge.

internal vertices → a, b, d, e, g, h, i, o
(vertex of degree larger than 1)

leaf (terminal node) ⇒ c, f, j, k, n, p, l, m, q, r, s.
(vertex of degree 1)

→ leaves, 0 outgoing edge.

forest:- disjoint trees is called a forest.

Th ⇒ $\boxed{E = V-1}$

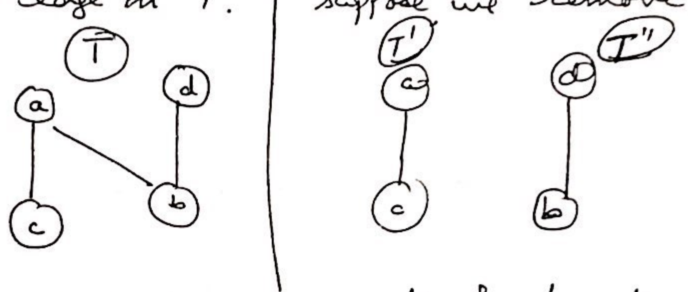
Proof. The Prop. can be proved by no. of vertices in the tree.

A tree with one vertex $\xrightarrow{\text{contain}}$ no edges ⇒ Ⓐ. —(1)

A tree with two vertex $\xrightarrow{\text{contain}}$ 1 edge ⇒ Ⓐ—Ⓑ —(2)

Let there is a tree T with V vertices & E edges.

Let (a, b) be an edge in T. | suppose we remove (a, b) from T.



The remaining edges form forest of two trees.

Now,

Let c be a vertex, such that path b/w a & c in T does not include the edge (a, b). Then Path b/w b & c in T must include the edge (a, b) because else there is a circuit in T.

Thus after the removal of edge (a, b) there is a path b/w a & c but no path b/w b & c. This tree is suppose T'.

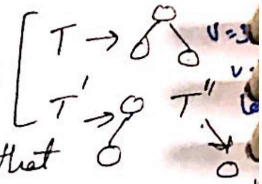
Similarly, let d be a vertex such that the path b/w a & d in T includes the {a, b}. Then the Path b/w b & d does not include the edge {a, b}. Thus, after the removal of edge {a, b} there is a path b/w b & d but no path b/w a & d.

∴ Removal of edge (a, b) divides T into two disjoint trees T' & T''.

T' contains a & all other vertices whose path to a in T do not contain the edge {a, b}.

T'' contains b & all other vertices whose path to b in T do not contain the edge $\{a, b\}$.

Since T' & T'' have atmost $v-1$ vertices,



it follows from induction hypothesis ① & ② that

$$e' = v' - 1 \quad — ③$$

$$e'' = v'' - 1, \quad — ④$$

[e' = no. of edges in T'
e'' = " " " in T''
v' = no of vertices in T'
v'' = " " " " T'']

Thus, Add ③ & ④

$$\underbrace{e' + e''} = v' + v'' - 2 \quad — ⑤$$

since,

$$e = e' + e'' + 1 \rightarrow \text{removed edge.}$$

∴ ⑤ can be written as.

$$\underbrace{e' + e'' + 1} = v' + v'' - 1$$

$$\boxed{e = v - 1}$$

Hence Proved.

Th 2. There is a unique path b/w every two vertices in a tree.

Sol^n ~~A tree is~~ In a connected graph, there is atleast one path b/w every two vertices. However, if there were two or more paths b/w a pair of vertices, there would be a circuit in a tree. (which is not allowed in Tree).

Th. A tree with 2 or more vertices has atleast two leaves

Solⁿ we know that,

$$\sum_{v \in V} deg(v) = 2e. \quad — (1)$$

$$\& \; E = V-1 \quad — (2).$$

$$\therefore \sum_{v \in V} deg(v) = 2V-2.$$

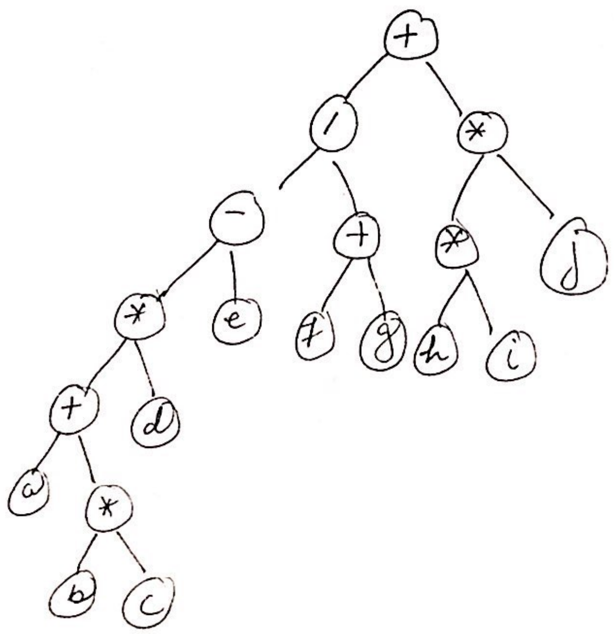
or $2 = 2.2 - 2$ $[V = 2]$

we know that, a tree with more than one vertex cannot have any isolated vertex (degree 0 vertex), there must be atleast 2 vertices of degree 1 each.

Since both the vertices have degree 1, the given two vertices are leaves.

Binary tree representation

$$(((a+b*c)*d-e)/(f+g))+h*i*j$$



Kruskal' Algorithm

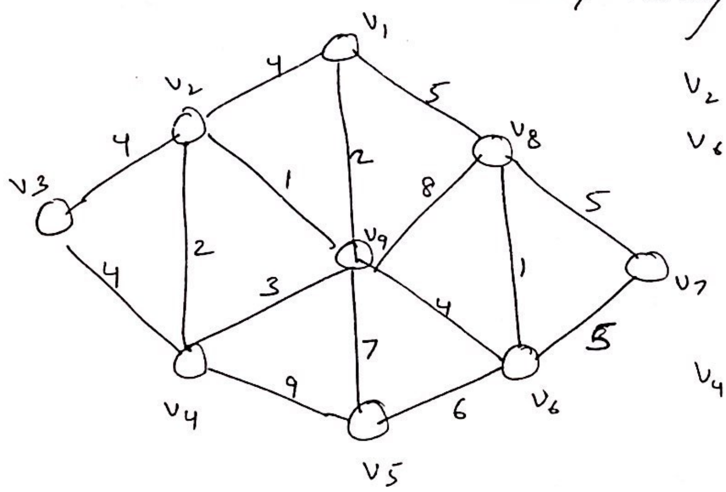
1). List all the edges of Graph G in the increasing order of weights.

2). Select the smallest edge (having minimum weight) from the list and add it to the spanning tree (which is initially empty), if the inclusion of edge does not make a circuit. If the selected edge makes a circuit, remove it from the list.

3). Repeat steps ① & ② untill the tree contain V-1 edges or the list is empty.

4). Now, if the tree contain less than V-1 edges & the list happened to be empty then no spanning tree is possible, else it gives the min. spanning tree.

eg:-



$v_2 - v_9 \rightarrow 1$

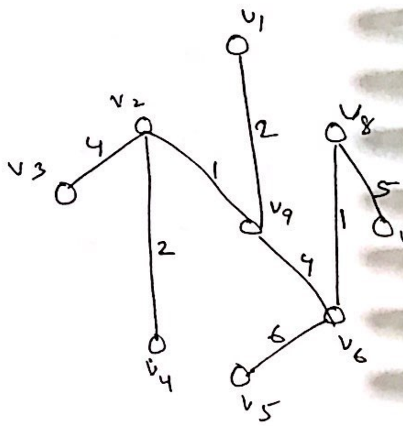
$v_6 - v_8 \rightarrow 1$

.

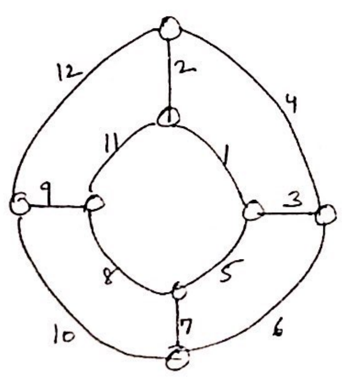
|

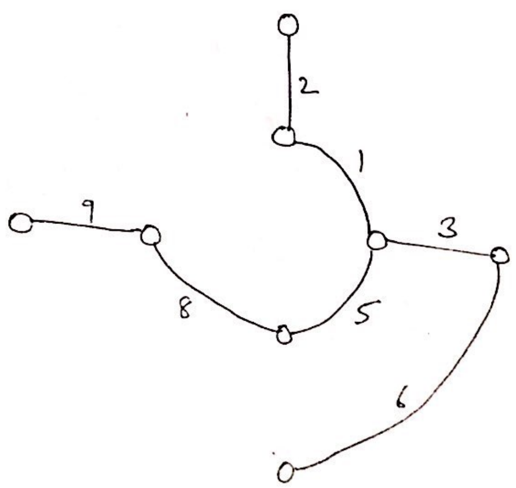
|

$v_4 - v_5 \rightarrow 9$



eg:-





Prim's Algorithm.

14

1) Draw n isolated vertices & label them as $v_1, v_2, \ldots v_n$ where n is the no. of vertices.

2). Represent the given weights of the edges of Graph G in an $n \times n$ matrix.

3). Assign the weights of nonexistent edges as very high (∞).

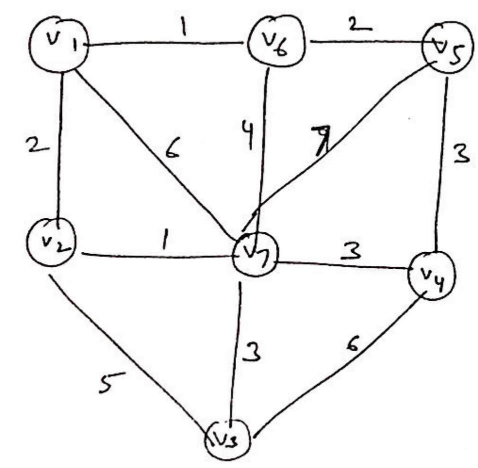
4). start from vertex v_1 & connect it to the nearest neighbour, say v_k. If more than one smallest entry is there, then arbitrarily select one of them.

5). consider v_1 & v_k as one subgraph & connect the subgraph to its closest neighbours (i.e. to a vertex other than v_1 & v_k which has the smallest entry in rows 1 and k) suppose this new vertex be v_i.

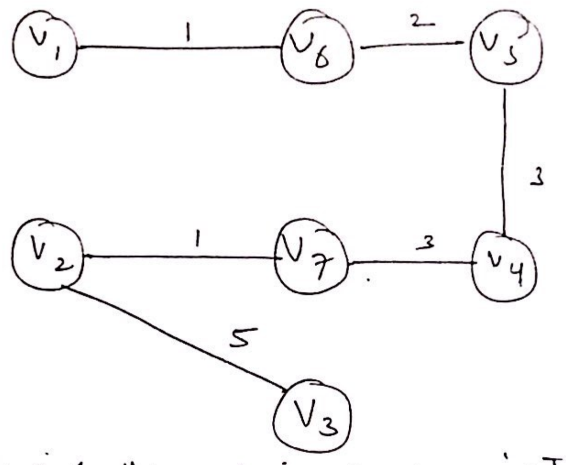
6) consider the tree with vertices $v_1, v_k,$ & v_i as one subgraph & continue the Process untill all the n vertices have been connected by $n-1$ edges.

7). Exit.

eg:-

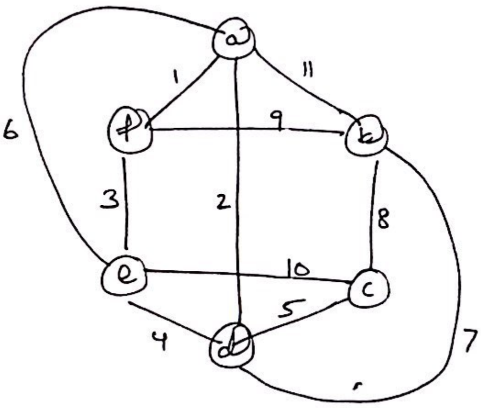


$$
\begin{array}{c}
 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} \\
\begin{bmatrix}
- & 2 & - & - & - & 1 & 6 \\
2 & - & 5 & - & - & - & 1 \\
- & 5 & - & 6 & - & - & 3 \\
- & - & 6 & - & 3 & - & 3 \\
- & - & - & 3 & - & 2 & 7 \\
1 & - & - & - & 2 & - & 4 \\
6 & 1 & 3 & 3 & 7 & 4 & -
\end{bmatrix}
\end{array}
$$

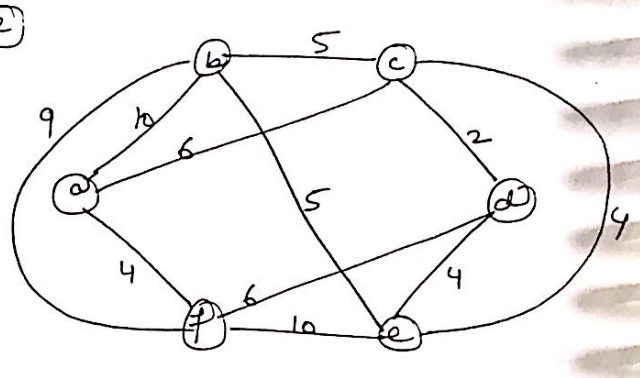


egs. Determine the minimum spanning tree !-

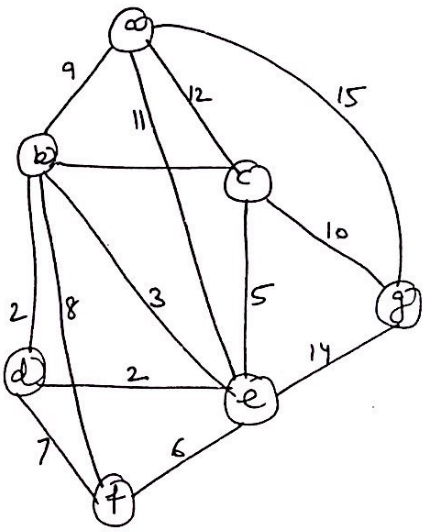
(1)



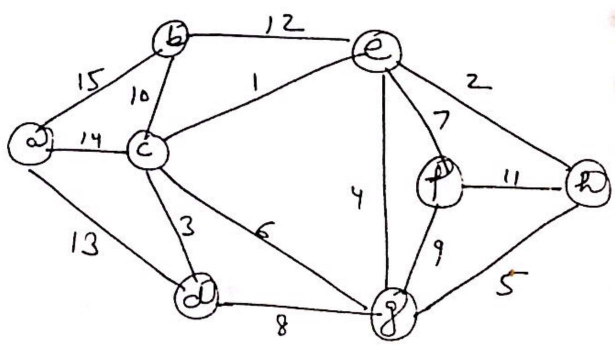
(2)



(3)

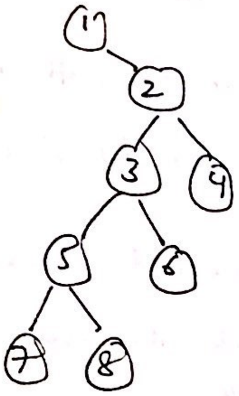


(4)



egs. Traverse the foll.

(1)

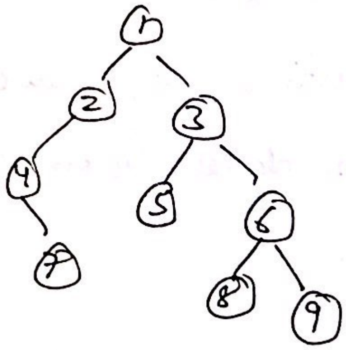


Inorder :- 1 7 5 8 3 6 2 4

Preorder :- 1 2 3 5 7 8 6 4

Post order :- 7 8 5 6 3 4 2 1

(2)



Pre ~~Inorder~~ :- 1 2 4 7 3 5 6 8 9

In ~~Pre~~order :- 4 7 2 1 5 3 8 6 9

Postorder :- 7 4 2 5 8 9 6 3 1.

Draw the Tree :-

Preorder :- G B Q A C K F P D E R H

Inorder :- Q B K C F A G P E D H R.

A Tree is an n-tree if every vertex in it has atmost n-offspring.

If every vertex has exactly n-offspring then T is a regular n-tree or full n-tree.

If $n=2$ then n-tree T is a called a Binary tree.

subtree of a T :- subtree that has son of a T as root.

Almost Complete Binary Tree :- The tree T is said to be complete if all its levels, except possibly the last, have the maximum no. of possible nodes, and if all the nodes at the last level appear as far left as possible.

Depth of a complete tree T_n with n node is $D_n = \lfloor \log_2 n + 1 \rfloor$

Q:- A tree has 2n vertices of deg-1, 3n vertices of deg-2, n vertices of deg-3. Determine the no. of vertices & edges in the tree.

Ans. Total no. of vertices $= 2n + 3n + n = 6n$.

In a tree :- $|E| = |V| - 1$

$\therefore E = 6n - 1. \quad — (1).$

In an undirected graph,

$$2|E| = \sum_{v \in V} deg(v) \quad — (3).$$

Here $\sum_{v \in V} deg(v) = 2n.1 + 3n.2 + n.3 = 11n. \quad — (2).$

Put (1) & (2) in (3).

$2(6n-1) = 11n \quad \Rightarrow 12n - 2 = 11n$

$\therefore n = 2. \quad \Rightarrow 1.n = 2 \quad \Rightarrow n = 2.$

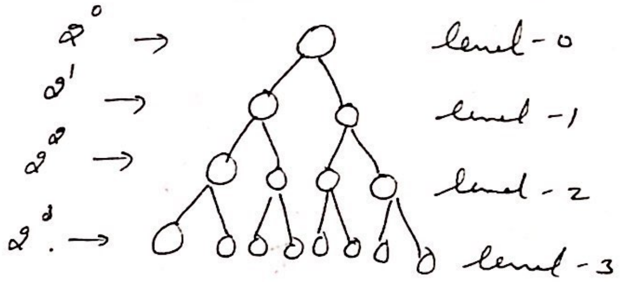
$\therefore E = 11$ edges & 12 vertices $\quad \therefore$ vertices $= 6n = 6(2) = 12$.

$E = V - 1 = 12 - 1 = 11$

Scanned with CamScanner

Balanced Tree :- A rooted tree is said to be balanced if all leaves are at level h or $h-1$.

Maximum no. of vertices in a binary tree of height h is $2^{h+1}-1$.



height = 3.

Total no. of leaves in a Binary tree of height h is atmost 2^h.

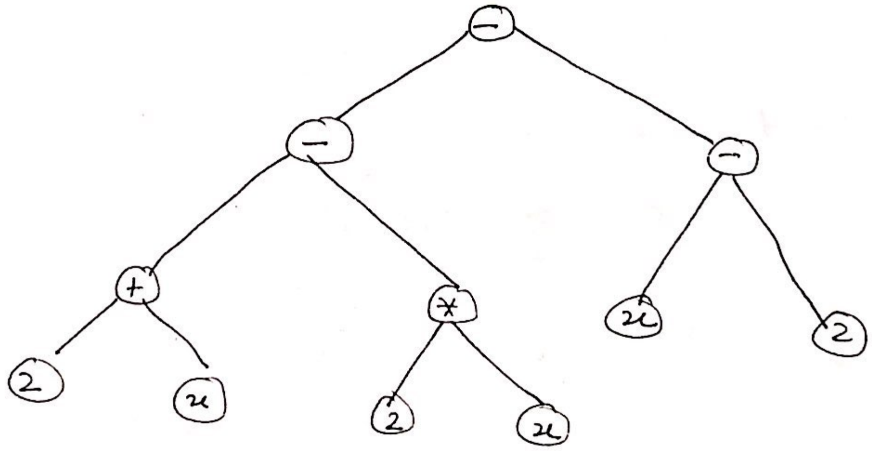
Rest of the nodes = $2^h - 1$.

$\therefore$ Total no. of vertices = $2^h + 2^h - 1 = 2^{h+1} - 1$.

Path length :- a vertex in a rooted tree is defined to be the no. of edges in the path from the root to the vertex.

Binary tree Representation

$$((2+x) - (2*x)) - (x-2)$$



Prefix code (Huffman code).

Defⁿ. :- These are the codes that do not represent the prefix of some other codeword.

Two types: 1) fixed length code.

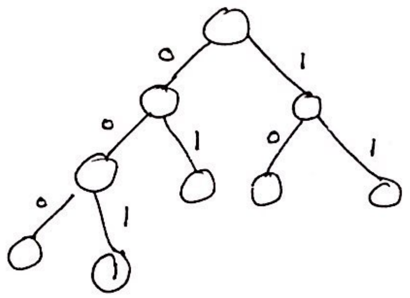
2) variable length code.

eg:- which of the foll. set is a Prefix code :-

ⓐ $M_c = \{1, 00, 01, \underline{000}, \underline{0001}\}$:- Not a prefix code.

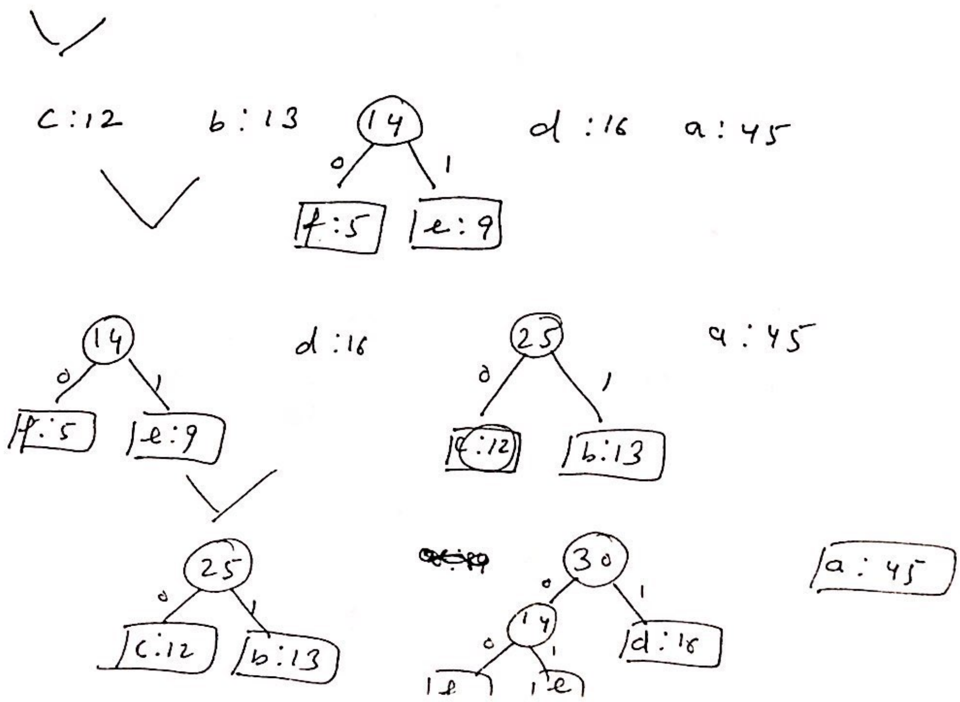
ⓑ $M_c = \{000, 001, 01, 10, 11\}$:- Prefix code.

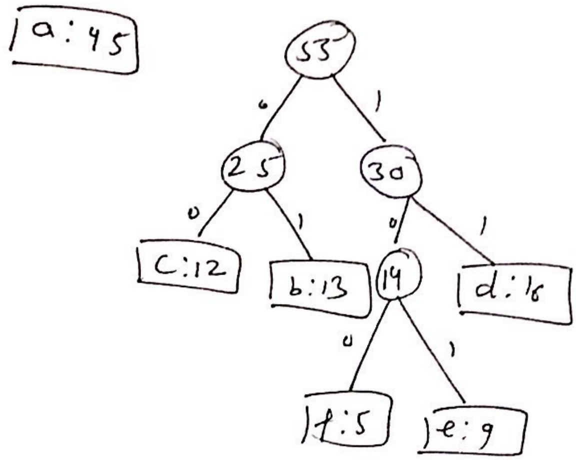
Binary tree for ⓑ,

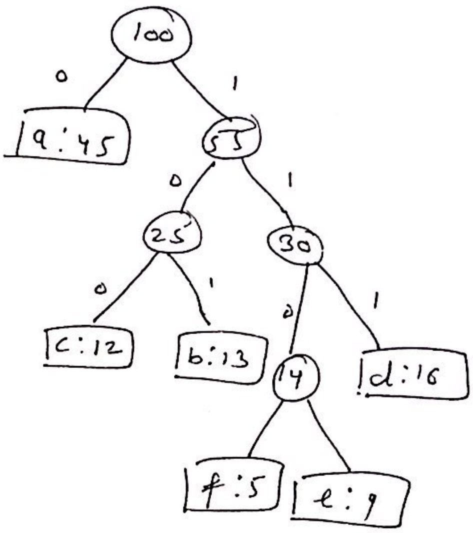


eg:- Design Huffman code for the set of letters.

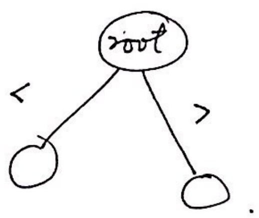
f:5 e:9 c:12 b:13 d:16 a:45.





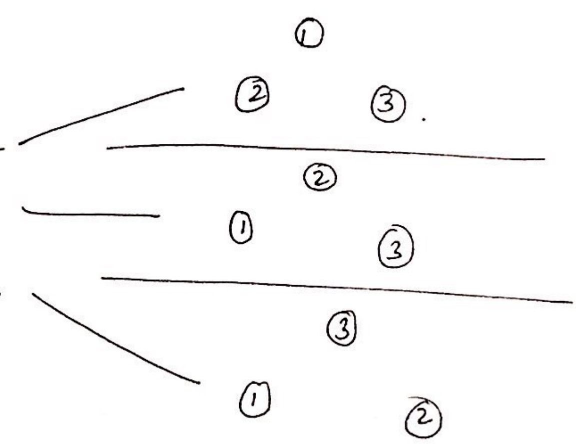


Binary Search Tree

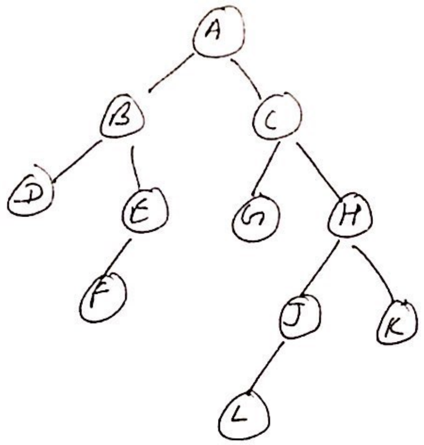


Traversal

1). Breadth first search.
2). Depth first search.
- Preorder
- Inorder
- Postorder.



eg.



Inorder :- D B F E A G C L J H K.

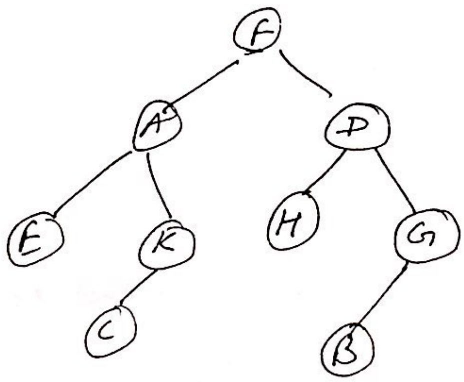
Postorder :- D F E B G L J K H C A.

Preorder :- A B D E F C G H J L K.

eg.

Inorder :- E A C K F H D B G

Preorder :- F A E K C D H G B.

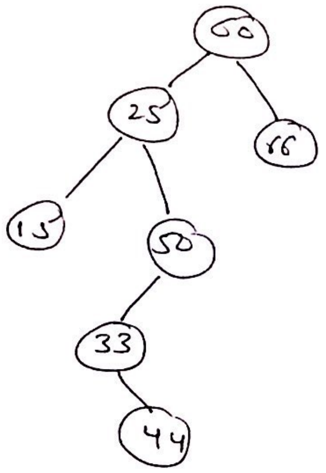


Deletion in BST.

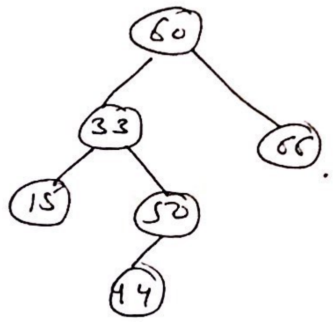
Case I :- No child. → simply delete & free the memory space.

Case II :- one child → delete the node & shift child to the node deleted. free memory space of child.

Case - III :- two child ⇒ find inorder successor of the node to be deleted. Replace it with the node & shift the child eg:-



Delete (25)



Prefix, Infix & Postfix notations :-

Precedent :-

( )

^

* /

\+ −

from left to right.

① ((9 / 3) * ((2 * 4) + (7−6)))

Prefix. (9/3) * ((* 2 4) + (− 7 6))

= 9 / 3 * (+ * 2 4 − 7 6)

= (/ 9 3) * (+ * 2 4 − 7 6)

= * / 9 3 + * 2 4 − 7 6 (If this exp. is given evalua

Sol^n. start from ~~right~~ to left.

= * / 9 3 + * 2 4 1

= * / 9 3 + 8 1

= * / 9 3 9

= * 3 9

= 27.

eg $((9/3) * ((2*4) + (7-6)))$

Post fix :-

$= ((9/3) * ((2\,4\,*) + (7\,6\,-))$

$= (9/3) * (2\,4\,*\ 7\,6\,-\ +)$

$= (9\,3\,/) * (2\,4\,*\ 7\,6\,-\ +)$

$= 9\,3\,/\ 2\,4\,*\ 7\,6\,-\ +\ *$

Convert or evaluate the above exp :-

1. move from left to right.

$= 3\ 2\ 4\ *\ 7\ 6\ -\ +\ *$

$= 3\ 8\ 7\ 6\ -\ +\ *$

$= 3\ 8\ 1\ +\ *$

$= 3\ 9\ *$

$= \underline{27}.$

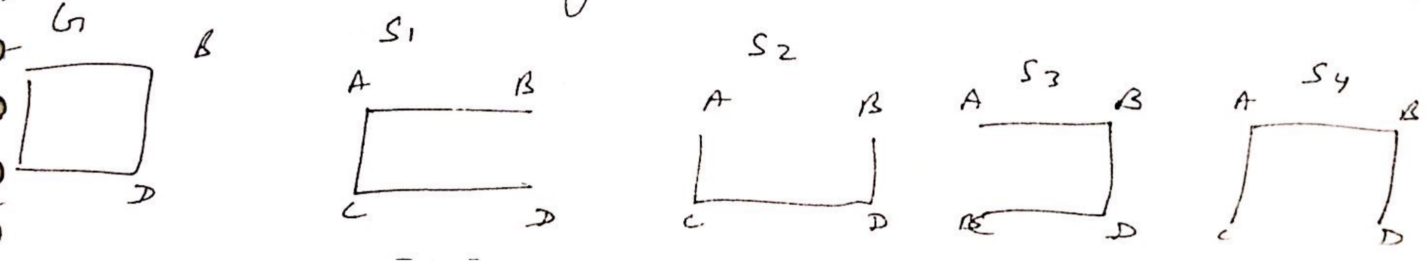
Spanning Trees :-

Def^n :- Let $G = (V, E)$ be a connected graph having $T(V_1, E_1)$ a tree as its subgraph, if $V = V_1$ then T is a spanning tree of the connected graph G.

Def^n :- A spanning tree of simple graph G is a subgraph of G that is a tree containing every vertex of G.

Prop :- $E - V + 1$ edges should be removed from a graph to obtain a spanning tree.

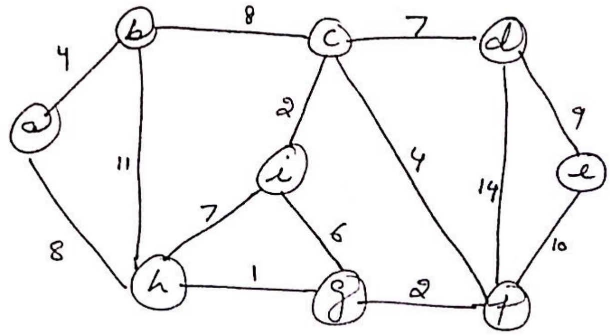
Eg.



Minimum spanning trees

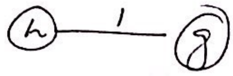
1) Kruskal Algo
2). Prim's Algo.

I. Select the edges with least weights & not forming cycles

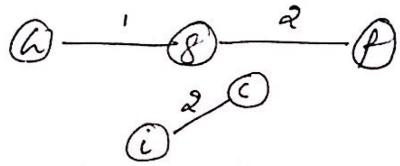


steps.

I



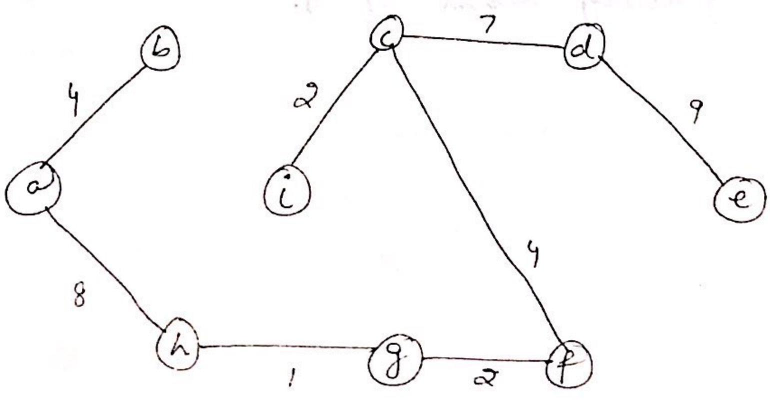
II



III

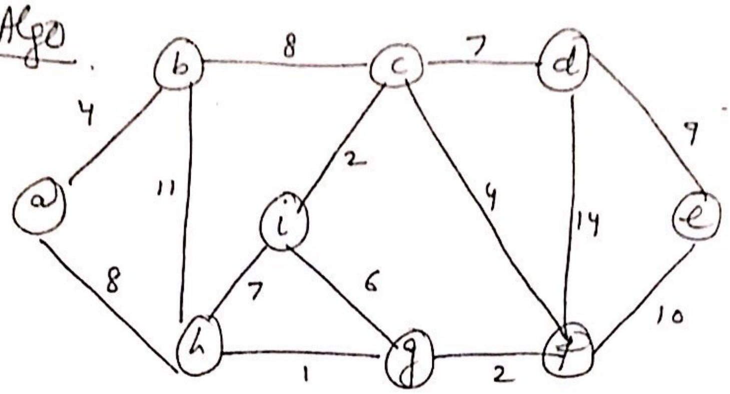
⋮

IV

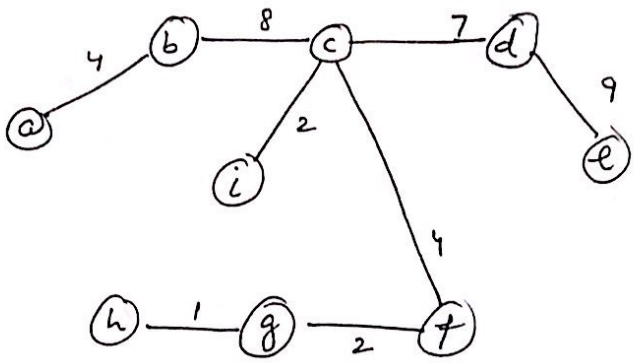


Prim's Algo

e.g.



start with 'a'



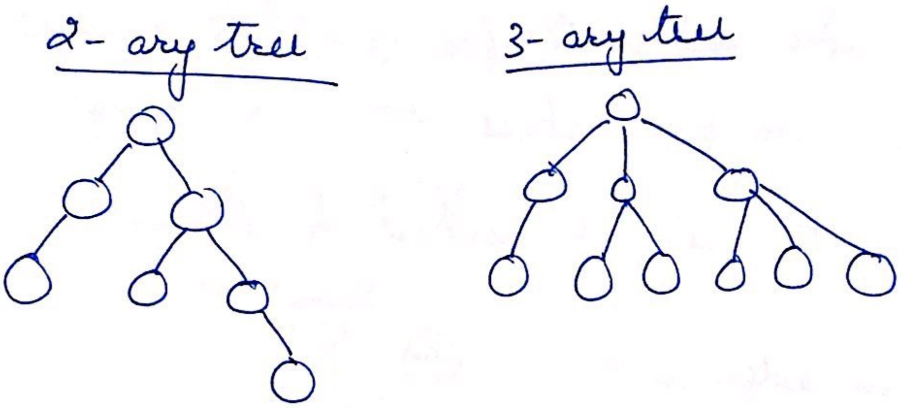
(a, b) = 4 ✓
(a, h) = 8 ×
(b, c) = 8 ✓
(b, h) = 11
(c, d) = 7 ✓
(c, i) = 2 ✓
(c, f) = 4 ✓
(i, h) = 7 ×
(i, g) = 6 ×
(f, e) = 10
(f, g) = 2 ✓
(f, d) = 14
(g, h) = 1 ✓

minimum weight = 37.

Th:- A simple graph is connected iff it has a spanning tree.

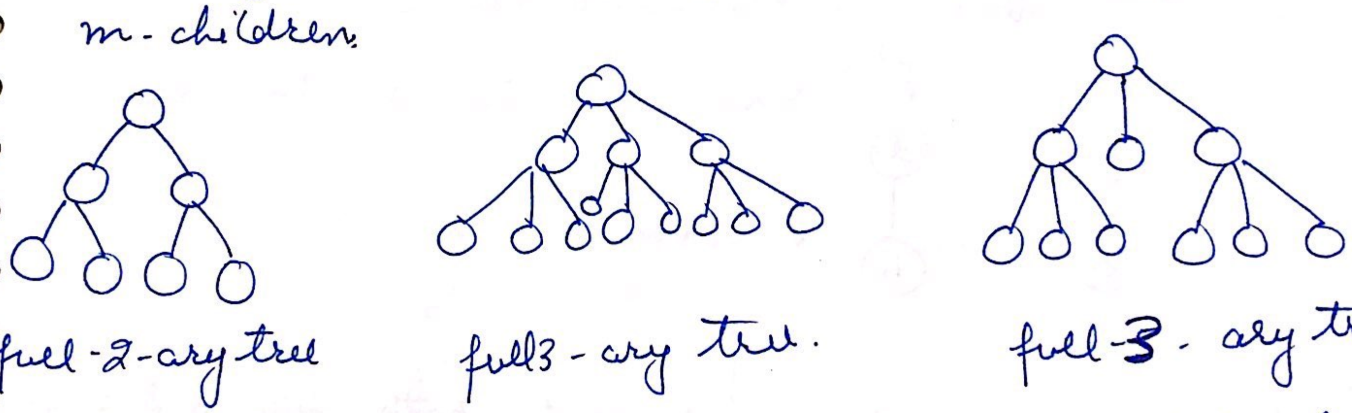
Proof:- Suppose that simple graph G has a spanning tree T. T contains every vertex of G. Also there is a path in T b/w any two of its vertices. Because T is a subgraph of G, there is a path in G b/w any two of its vertices.
Hence G is connected.

<u>m-ary tree</u> :- A rooted tree is called m-ary tree if every internal vertex has no more than m-children.



2-ary tree is called <u>Binary tree</u>.

<u>full m-ary tree</u> :- If every internal vertex has exactly m-children.



<u>Internal vertices</u> :- nodes except leaves are Internal vertices.

Scanned with CamScanner

Th $E = V-1$.

OR

A tree with 'n' vertices has $n-1$ edges.

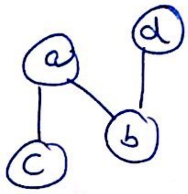
Proof:- The Prop. can be proved by no. of vertices in the tree.

Basis step A tree with one vertex $(n=1)$ $\xrightarrow{\text{contain}}$ 0 edges. - (1)

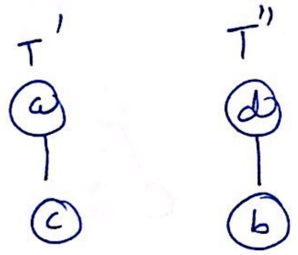
A tree with two vertices $\xrightarrow{\text{contain}}$ 1 edge. - (2)

Let there is a Tree T with V vertices & E edges.

Let (a, b) be an edge in T.



Suppose we remove (a, b) from T, so that remaining edges form forest of two trees.



Now,

Let c be a vertex such that path b/w a & c in T does not include the edge (a, b). Then path b/w b & c in T must include edge (a, b) because else there is a circuit in T. Thus after the removal of edge(a, b) there is a path b/w a & c but no path b/w b & c.

This tree is suppose T'.

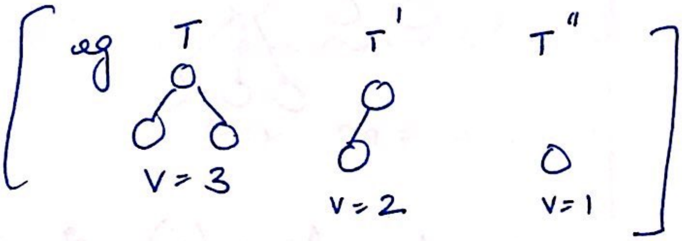
Similarly, let 'd' be a vertex such that the path b/w a & d in T includes the {a, b} edge. Then the path b/w b & d does not include the edge (a, b). Thus, after the removal of edge (a, b) there is a path b/w b & d but no path b/w a & d.

∴ Removal of edge (a, b) divides T into two disjoint trees T' & T''.

T' contains 'a' & all other vertices whose path to a in T do not contain the edge (a, b).

T'' contains 'b' & all other vertices whose path to b in T do not contain the edge (a, b).

Since T' & T'' have atmost v-1 vertices,



it follows from induction hypothesis, 1 & 2 that

$$e' = v' - 1 \quad — ③$$

$$e'' = v'' - 1 \quad — ④$$

Induction Hypothesis states that every tree with K vertices has K-1 edges.

Add (3) & (4)

$$e' + e'' = v' + v'' - 2 \quad — (5)$$

Since $e = e' + e'' + 1 \rightarrow$ (removed edge)

Eq (5) can be written as,

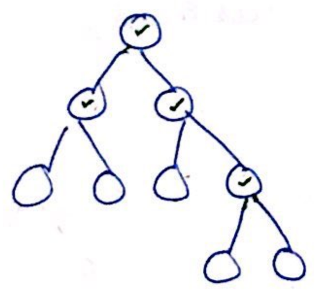
$$\underbrace{e' + e'' + 1} = \underbrace{v' + v''} - 1$$

$$\boxed{e = V - 1}$$

Hence Proved.

Th. A full m-ary tree with i internal vertices contains $n = mi + 1$ vertices.

~~Pf.~~ eg. full-2 ary tree



Internal vertices $(i) = 4$

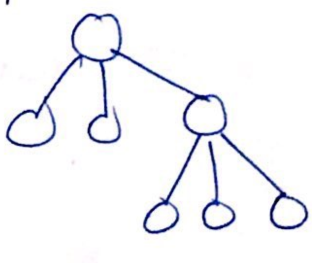
$m = 2$

vertices $(n) = mi + 1$

$= 2.4 + 1$

$= 9$

full 3-ary tree.



internal vertices $(i) = 2$

$m = 3.$

vertices $(n) = mi + 1$

$= 3.2 + 1$

$= 7.$

Proof :- Every vertex except the root, is the child of an internal vertex.

Each of the 'i' internal vertices has 'm' children, thus 'mi' vertices, other than the root.

$\therefore$ Total vertices in a tree are $\boxed{n = mi + 1}$.

Some Useful Results :-

~~Example~~ :- T is a full m-ary tree.

i = no. of internal vertices.

l = no. of leaves

n = no. of vertices.

(A) Given :- n vertices, & m.

then $i = (n-1)/m$, $l = [(m-1)n+1]/m$

(B) Given :- i internal vertices & m.

then $n = mi + 1$ & $l = (m-1)i + 1$

(C) Given :- l leaves & m.

then

$$n = (ml - 1)/(m-1) \quad \& \quad i = (l-1)/(m-1)$$

Proof (A). we know that $n = i + l$ — (1)

Also, $n = mi + 1$, solving for i, $\boxed{i = (n-1)/m}$ — (2)

Put (2) in (1), $n = (n-1)/m + l$

$$\boxed{l = [(m-1)n + 1]/m}$$

Proof :- Every vertex except the root, is the child of an internal vertex.

Each of the 'i' internal vertices has 'm' children, thus 'mi' vertices, other than the root.

$\therefore$ Total vertices in a tree are $\boxed{n = mi + 1}$.

Some Useful Results :-

~~Given~~ :- T is a full m-ary tree.

i = no. of internal vertices.

l = no. of leaves

n = no. of vertices.

A

Given :- n vertices, & m.

then $i = (n-1)/m$, $l = [(m-1)n + 1]/m$

B

Given :- i internal vertices & m.

then $n = mi + 1$ & $l = (m-1)i + 1$

C

Given :- l leaves & m.

then $n = (ml - 1)/(m-1)$ & $i = (l-1)/(m-1)$

Proof A. we know that $n = i + l$. — 1

Also, $n = mi + 1$, solving for i, $\boxed{i = (n-1)/m}$ — 2

Put 2 in 1, $n = (n-1)/m + l$

$\boxed{l = [(m-1)n + 1]/m}$

Scanned with CamScanner

Q. A tree has $2n$ vertices of deg-1.
$3n$ vertices of deg-2
n vertices of deg-3.
find no. of vertices & edges in Tree.

Sol^n. Total no. of vertices = $2n + 3n + n = 6n$.

Since Tree is given $E = V - 1$

$$\boxed{E = 6n - 1} \quad — (1)$$

In an undirected graph,

$$2e = \sum_{v \in V} \deg(v)$$

$$= 2n.1 + 3n.2 + n.3 = 11n$$

$$\therefore 2e = 11n \quad — (2)$$

Put (1) in (2),

$$2(6n-1) = 11n$$

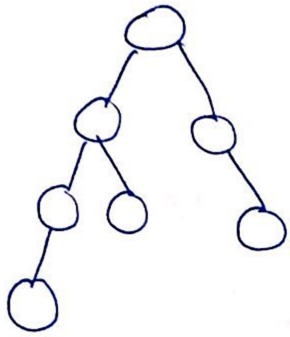
$$n = 2.$$

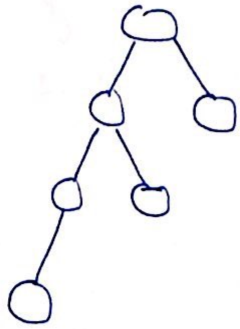
$\therefore$ no. of vertices $= 6n = 6.2 = 12.$
no. of edges $\Rightarrow E = V - 1 = 12 - 1 = 11$

Balanced Tree

A rooted m-ary tree of height 'h' is balanced if all leaves are at level 'h' or 'h-1'.

eg:

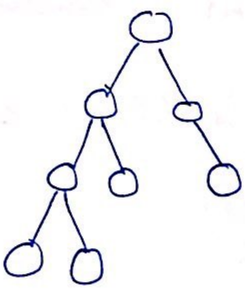




Balanced   unbalanced.

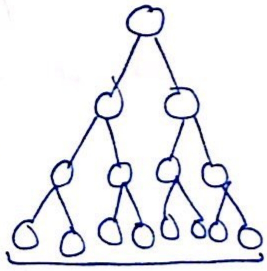
Th. There are atmost m^h leaves in an m-ary tree of height 'h'.

eg:



$h = 3, \ m = 2$

$\therefore$ atmost $m^h = 2^3 = 8$ leaves



8 leaves.

2-ary tree

The Proof uses mathematical induction on the height.

first consider m-ary trees of height 1.

$h = 1$
$m = 2$

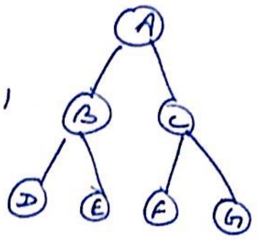
The tree consist of a root with no more than 'm' children, each of which is a leaf. Hence, there are no more than $m^1 = m$ leaves in an m-ary tree of height 1.

Now assume that the result is true for all m-ary trees of height less than 'h'.

Next, Let T be an m-ary tree of height 'h'.
The leaves of T are the leaves of the subtrees of T obtained by deleting the edges from the root to each of the vertices at level 1

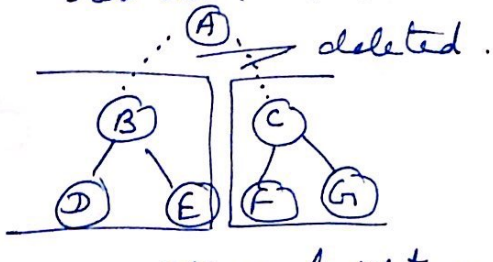
T

level-1



height = 2.

Subtree 1 & 2.



subtrees height = 1

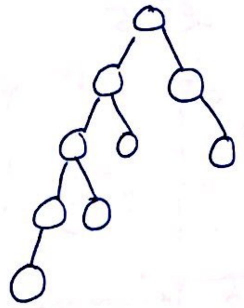
$\therefore$ Each of these subtrees has height less than or equal to $h-1$.
So each of these subtrees has atmost m^{h-1} leaves.
Also, there can be atmost 'm' such subtrees each with a maximum of m^{h-1} leaves;
Therefore, maximum leaves in a tree are,

$$m \cdot m^{h-1} = m^{h} \text{ leaves}$$

Hence Proved.

Th. If an m-ary tree of height 'h' has 'l' leaves, then $h \geq \lceil \log_m l \rceil$.

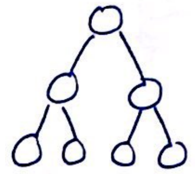
eg. 2-ary tree



$h = 4$, $m = 2$
$l = 4$.

$h \geq \lceil \log_2 4 \rceil$

$h \geq 2$



$h = 2$, $m = 2$, $l = 4$.

$h \geq \lceil \log_2 4 \rceil$

$h \geq 2$.

Proof. :- we know that,

$l \leq m^h$ [Atmost m^h leaves in an m-ary tree of height h.]

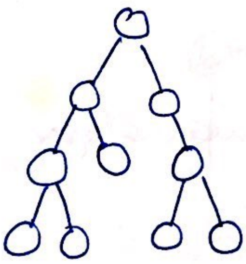
Taking log to the base m both sides,

$\log_m l \leq h \log_m m$

$\log_m l \leq h$

Because, h is an integer, we have $h \geq \lceil \log_m l \rceil$

eg.



$h = 3$
$l = 5$
$m = 2$

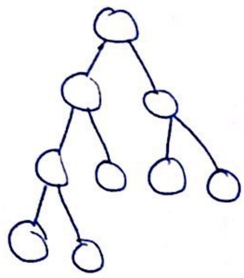
$h \geq \lceil \log_2 5 \rceil$

$h \geq \lceil 2.... \rceil$

$h \geq 3$

Th. If an m-ary tree is full and balanced, then $h = \lceil \log_m l \rceil$.

eg:



$h = 3, m = 2, l = 5$

$h = \lceil \log_2 5 \rceil$

$= \lceil 2. \ldots \rceil = 3$

$\boxed{h = 3}$

Proof

If tree is balanced, then each leaf is at level h or $h-1$.

Since the height is 'h', there is atleast one leaf at level 'h'.

Also, there must be more than m^{h-1} leaves.

ie more than $(2^{3-1} = 4)$ leaves ie $\boxed{m^{h-1} < l}$

Also, ~~Because~~, $l \leq m^h$, we have.

$$m^{h-1} < l \leq m^h$$

Taking log to the base m in this inequality, we have,

$$h-1 < \log_m l \leq h$$

Hence, $\boxed{h = \lceil \log_m l \rceil}$

(1) If G is a connected planar simple graph with e edges & v vertices, where $v \geq 3$, then prove that $e \leq 3v - 6$. (4)

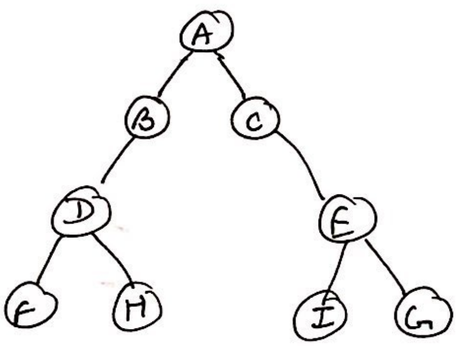
(2) What is chromatic no. of Graph C_n where $n \geq 3$. (2)

(3). If an m-ary tree of height 'h' has 'l' leaves, then prove, $h \geq \lceil \log_m l \rceil$ (3).

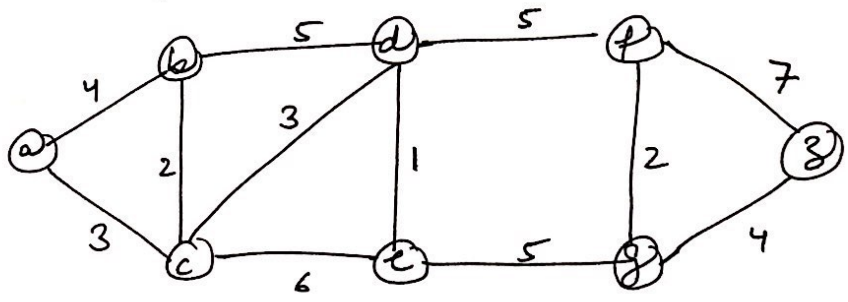
(4). Define spanning Trees. Construct a Binary tree representing $((x+y) \uparrow 2) + ((x-4) / 3)$. $2+2=$ (4)

(5) Write the foll. traversals – Preorder, Inorder, Postorder

$2+2+2=$ (6)



(6) Find shortest distance using dijkstra's Algo. (from a to z)



(6).

Scanned with CamScanner

Trees.

Th. A circuit & the complement of any spanning tree must have at least one edge in common. (Pg 275 liu).

Th. A cut-set & any spanning tree must have at least one edge in common. (Pg 275-liu)

Th. Every circuit has, an even no. of edges in common with every cut-set. (Pg 275-liu).

Th. There is a unique path b/w every two vertices in a tree. (Pg 12 Not)

Th. A tree with two or more vertices has atleast two leaves (Pg 13 Note