A tree can be defined as a non-linear data structure that stores data in the form of nodes and nodes are connected to each other with the help of edges. There are different ways of traversing a tree depending upon the order in which the trees nodes are visited and the types of data structure used for traversing the tree. There are various data structure involved in traversing a tree as traversing a tree involved all nodes in some manner. Types of traversals are inorder, pender, Elpost order traversal.

· inorder traversal (left, root, right)

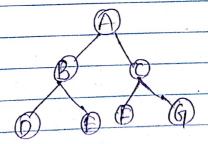
· preorder traversal (root, left, right)

· post order traversal (left, right, root)

Example:

Inorder traversal:-

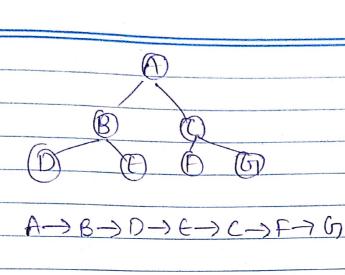
14 a binary tree is traversed in order the output will produce sorbal key values in ascending order.



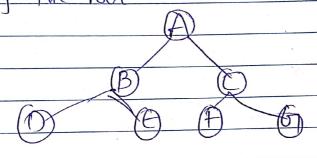
D->B->E->A->F->C->G

Preorder traversal:

In this traversal method, the root node is visited first, then the left subtree and finally the right subtree.



post-order trainers al-The root node is visited last, hence the same first we trainerse the left subtree than the right subtree & finally the root.



## D-1 (-) B-) F-> (1-) C-)A

Corder

Inorder:

class treenode:

det\_\_init\_\_ (self, val):

self-val=val

self-left=None

def Provder traversal (voot):

answer=[]

Enorder traversal until (root answer)

return answer

det Enordertraversaluntil (root, answer):

it voot is None: return inorder traversaluntil (root left, onsoer) answer append (root value) inordertraileral un fil (voot-right, answer) YOUT = Tree node(1) voot left Treende(1) voot- vight= Tree node (3) root-left left = Treenode (4) root-left-vight=Tree node (5) print (inorder traversal (root)) Dre-order: def preorder traversal (root): answerz [] preorder trailersal until (root, answer) return answer det preorder traversal until (root, answer): it voot Is None! return answer append (voot val) preordentraversaluntil (root. left, answer) preorder traversal until (rout. right, answer) return root=Treenode(1) voot-left = Tree node(2) root-right = Treenode (3) voot-left. left = Tree notely) voot-left-vight = Tree node (5) print preorder traversal (root)

Post Order:	
def postordertraversal (root):	and the same of th
answey=[]	1000
postordertraversal antil (voot, answer)	Control of the Contro
return answer	Cont.
det postordertraversal (1001; answer):	
ît root is None:	
return	
postordertraversaluntil (root-left, answa)	The same of the sa
pustorder traversal until (root right, answer)	Ligano de la companya
answer-append (root-val)	-009
vetuvn	
root=Treenode(1)	
root-lett=Treenode(2)	
root-right=Tree node (3)	
root-left-left=Tree node(u)	
root-left-right=Treenode(s)	No.
print-postordertraversal(root)	
	Street Contract Contr
	and the second s
	and the second s
	and the second s
	the particular and specimen and construction to the second state of the second state o
	And the second second of the second
	And the second s
	and the second
	·

Output-Inorder -4 2 5 13 preorder - 1 2 4 53
postorder - 4 5 2 31