SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : 23ITT102_Programming in C and Data Structure

I YEAR/ IISEMESTER

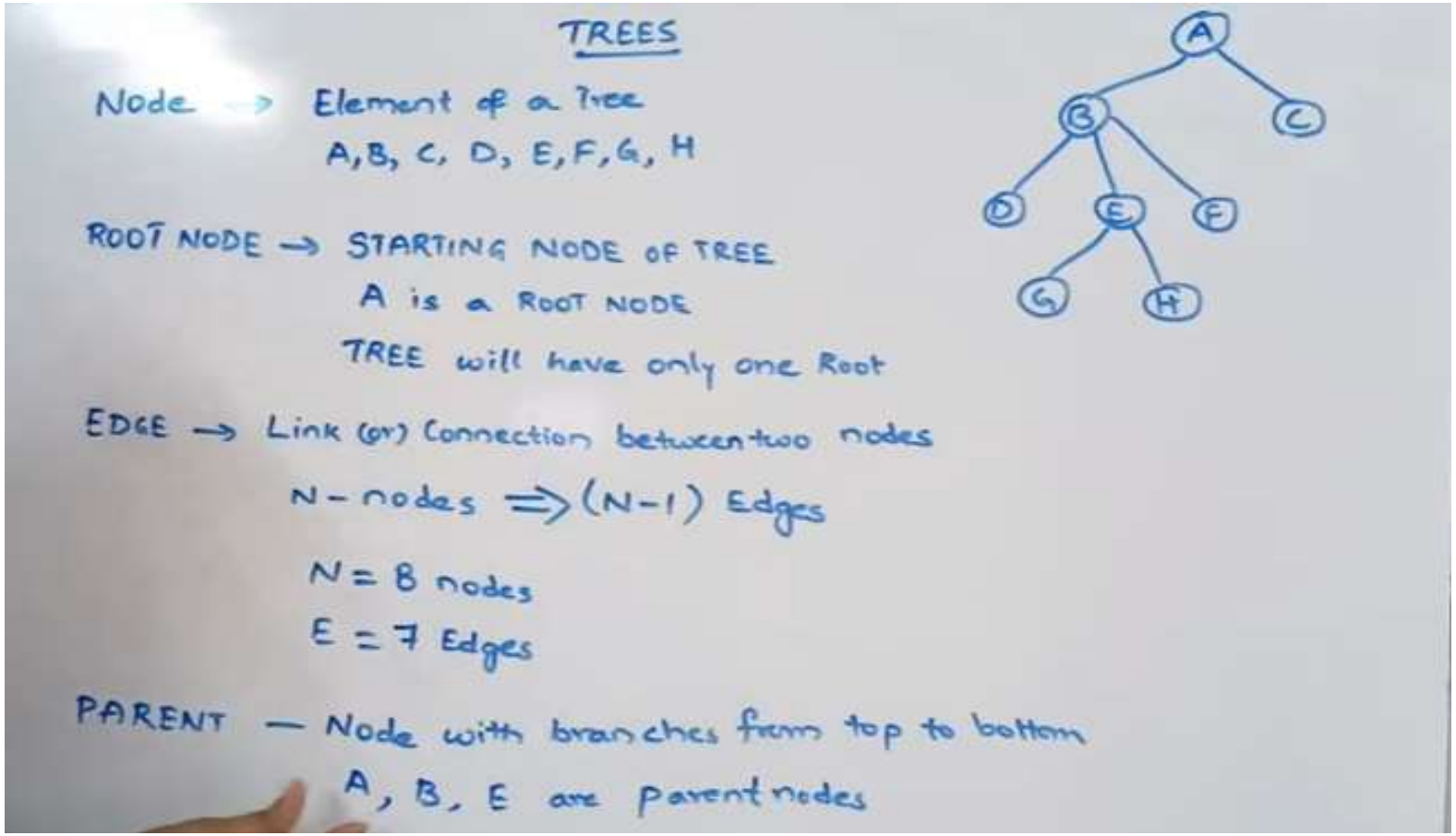
UNIT V

Topic: Tree ADT &*Binary Tree*

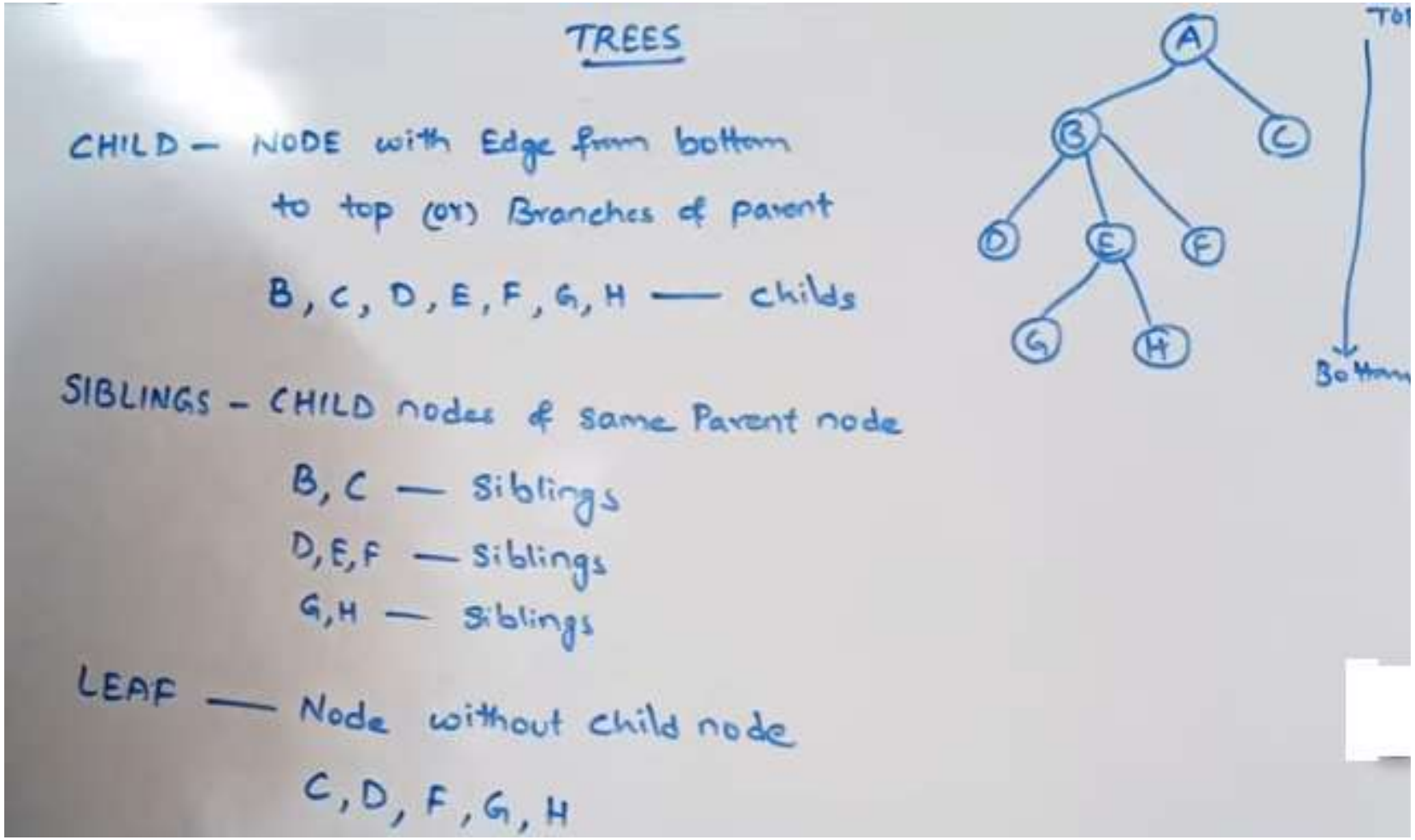
Tree Terminologies

A tree data structure is defined as a collection of objects or entities known as nodes that are linked together to represent or simulate hierarchy.

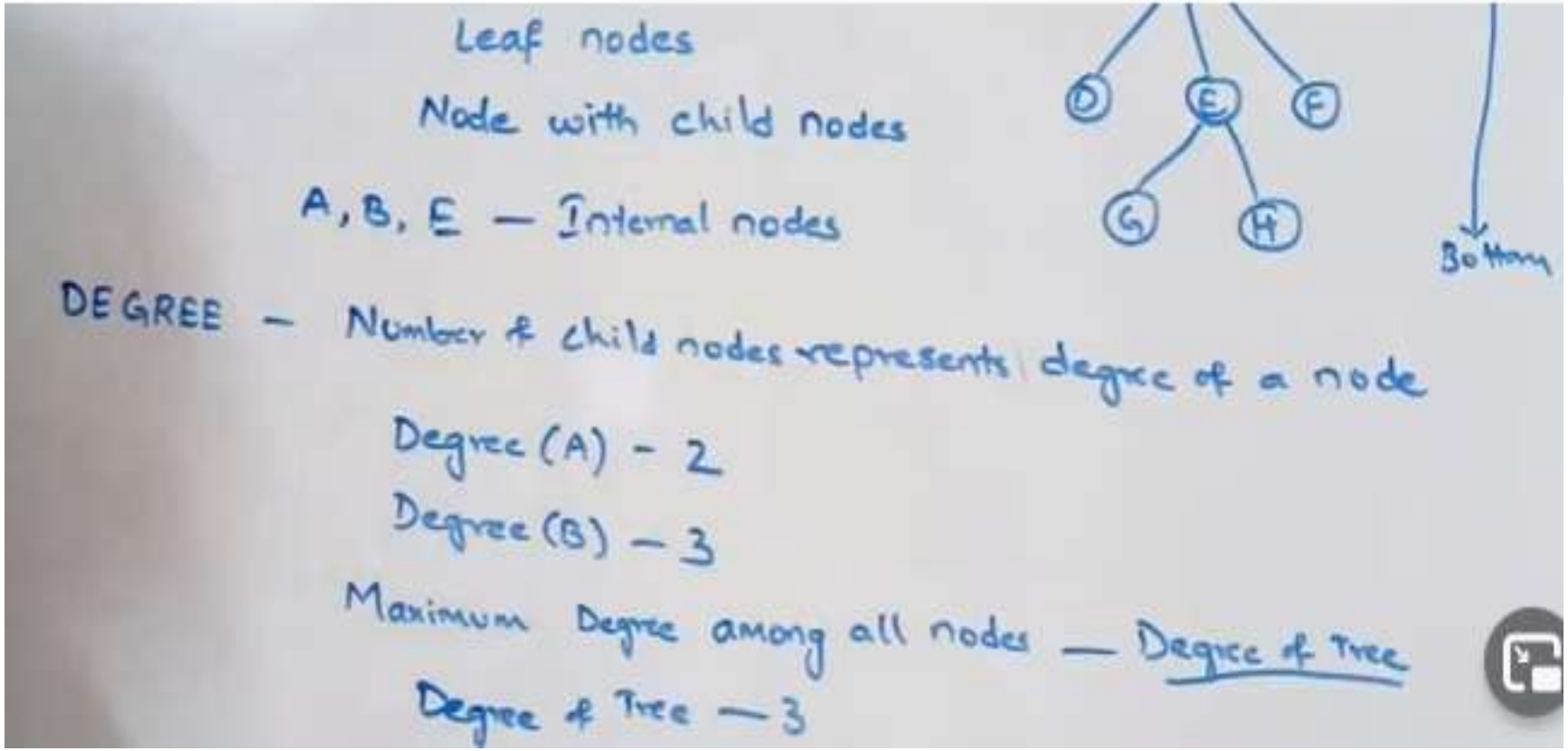
A tree data structure is a non-linear data structure because it does not store in a sequential manner. It is a hierarchical structure as elements in a Tree are arranged in multiple levels.



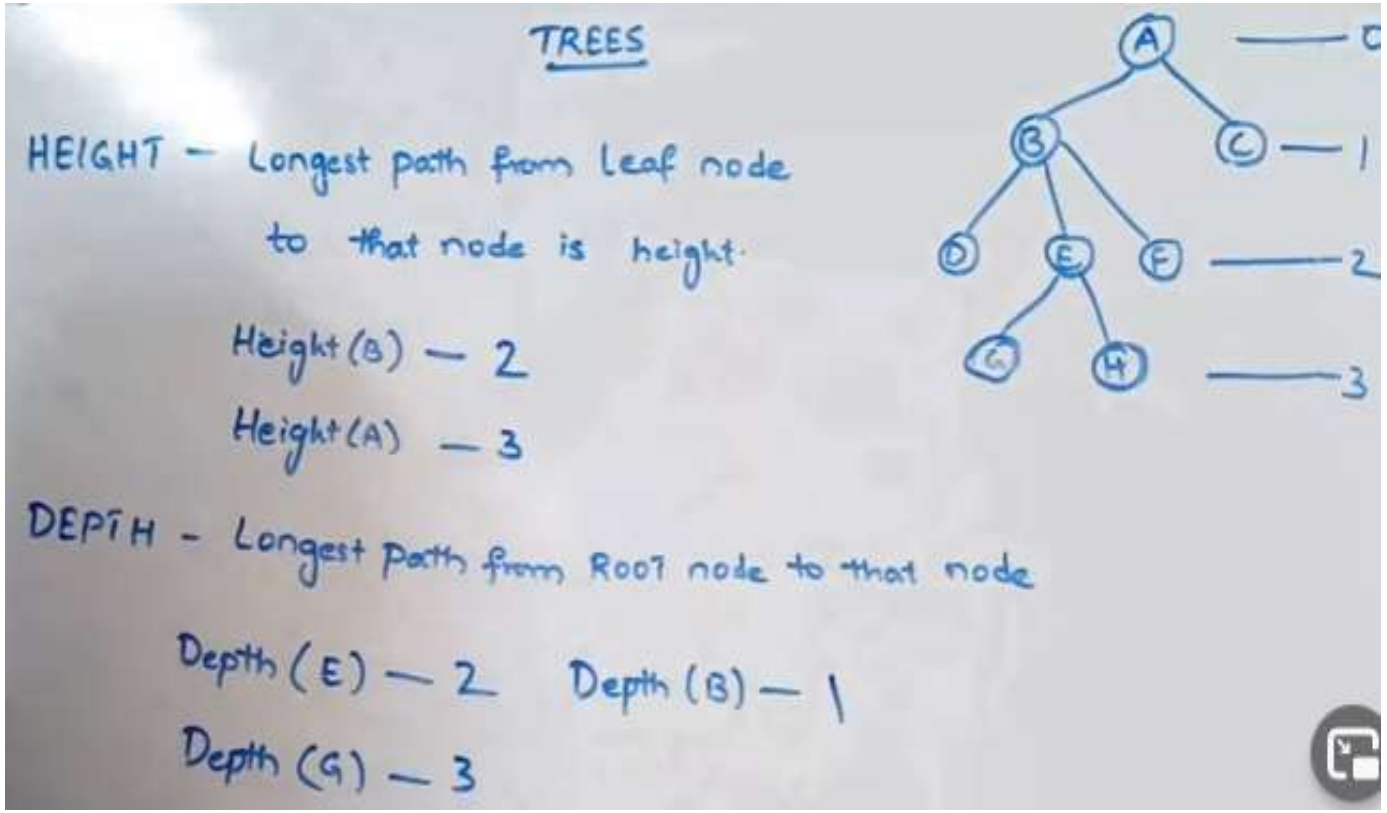
Tree Terminologies



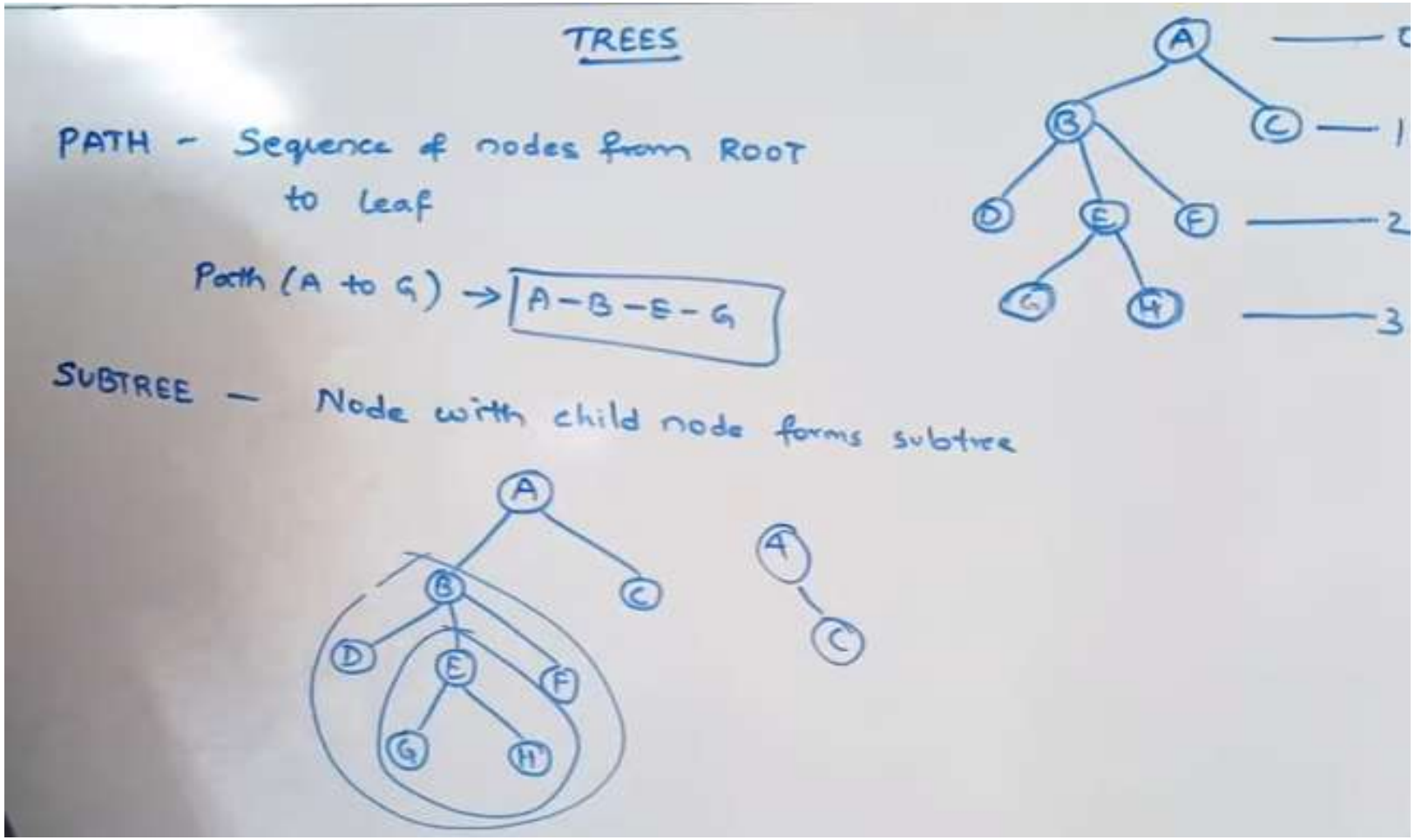
Tree Terminologies



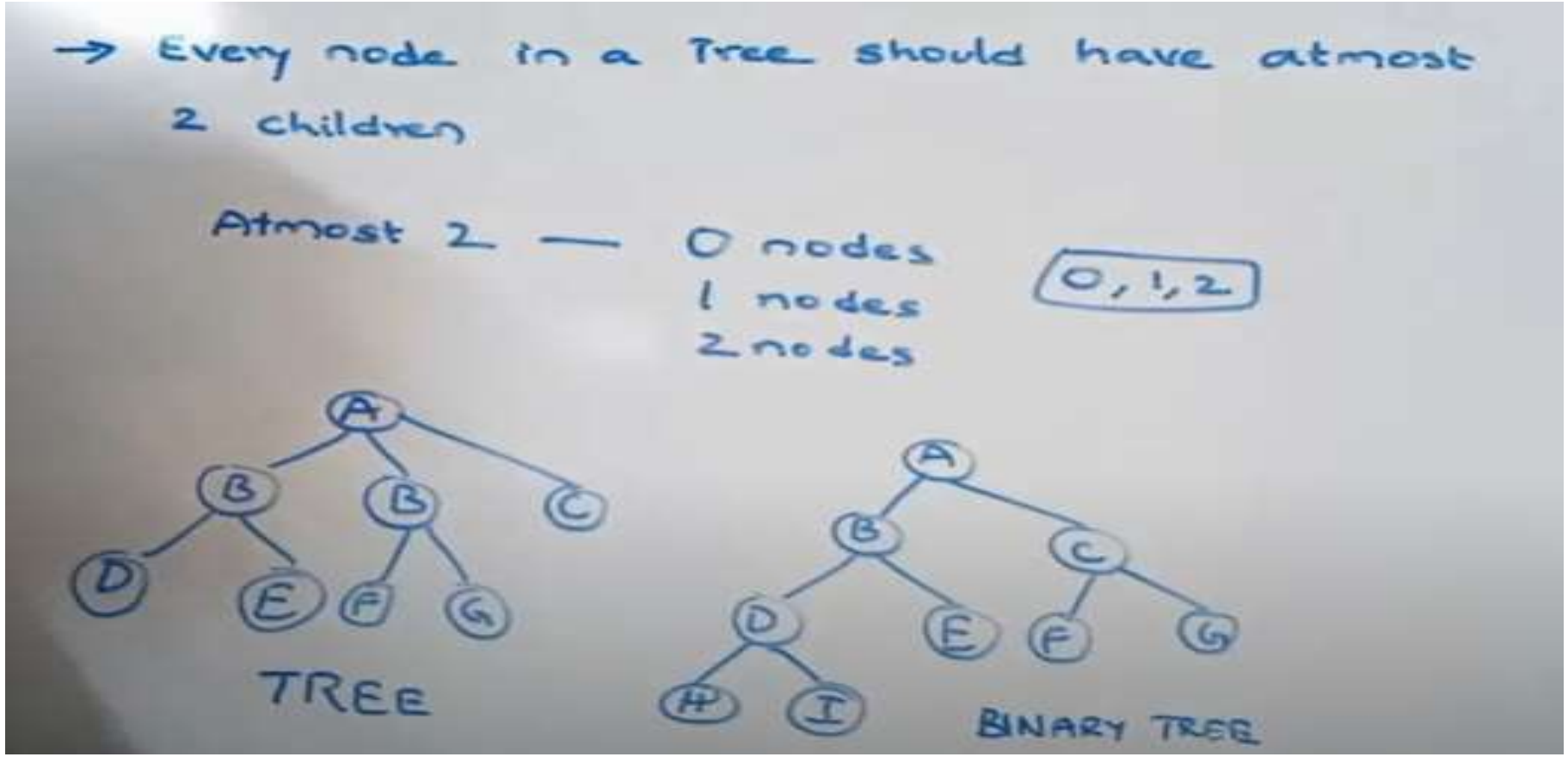
Tree Terminologies



Tree Terminologies



Binary Tree

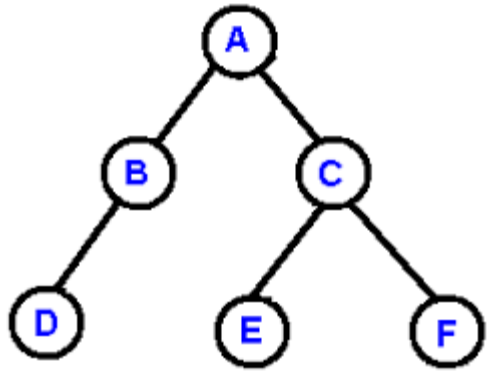

→ Every node in a Tree should have atmost 2 children
Atmost 2 — 0 nodes
1 nodes
2 nodes
0, 1, 2
A
B
B
C
D
E
F
G
TREE
A
B
C
D
E
F
G
H
I
BINARY TREE

Binary Tree

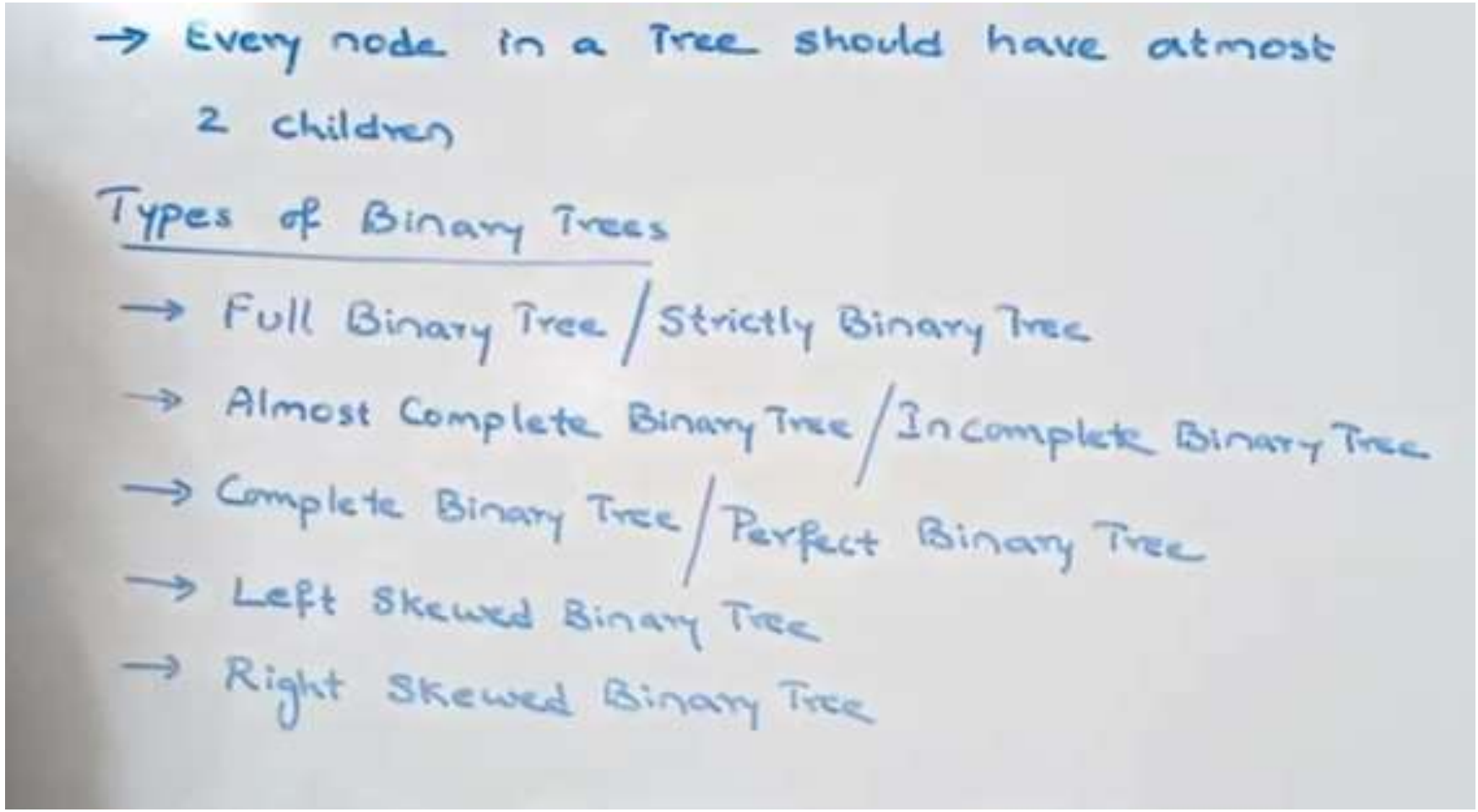
- Tree in which no node can have more than two children

Node declaration

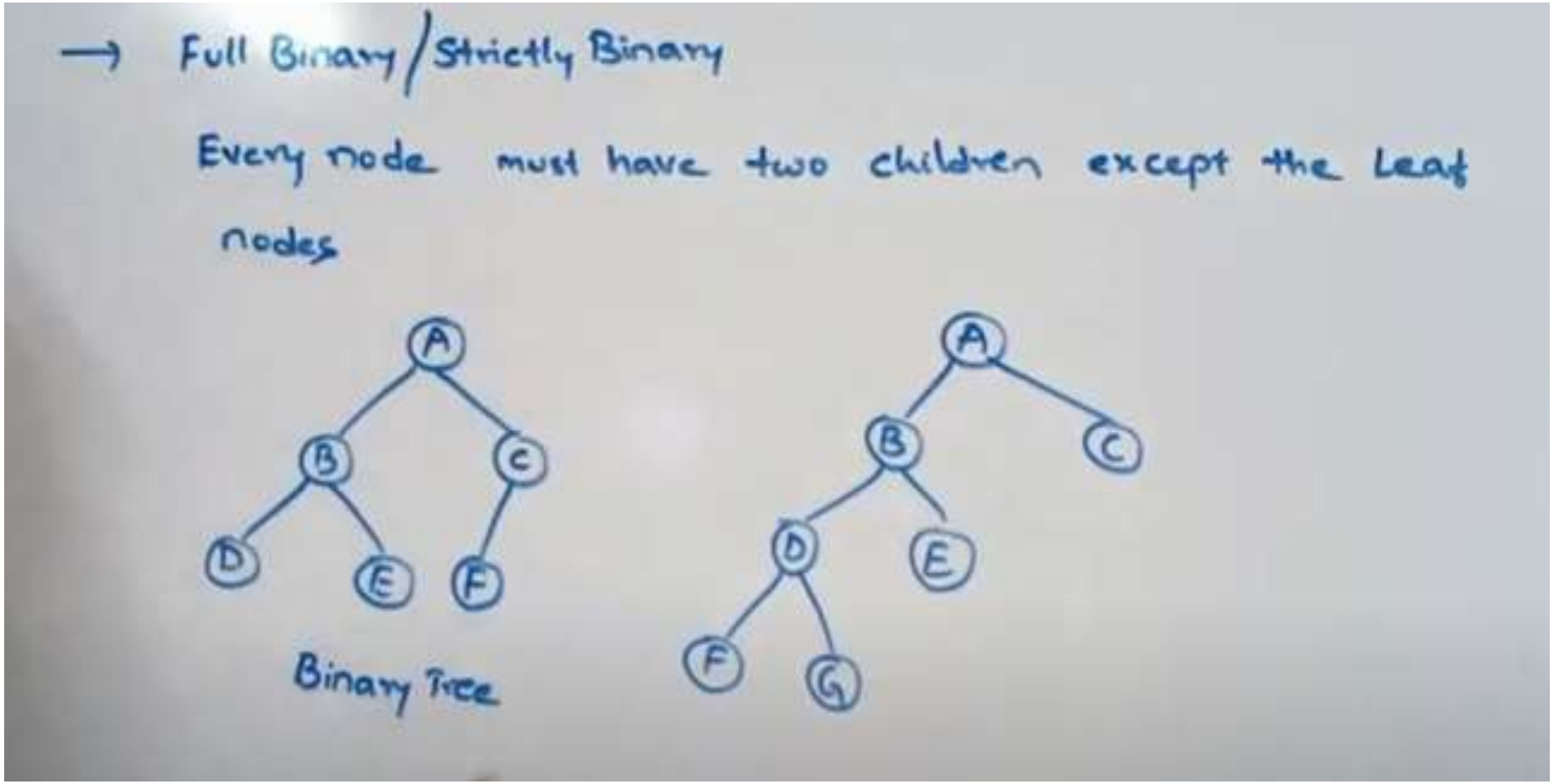
```
Struct TreeNode
{
Int Element;
Struct TreeNode *Left;
Struct TreeNode *Right;
}
```



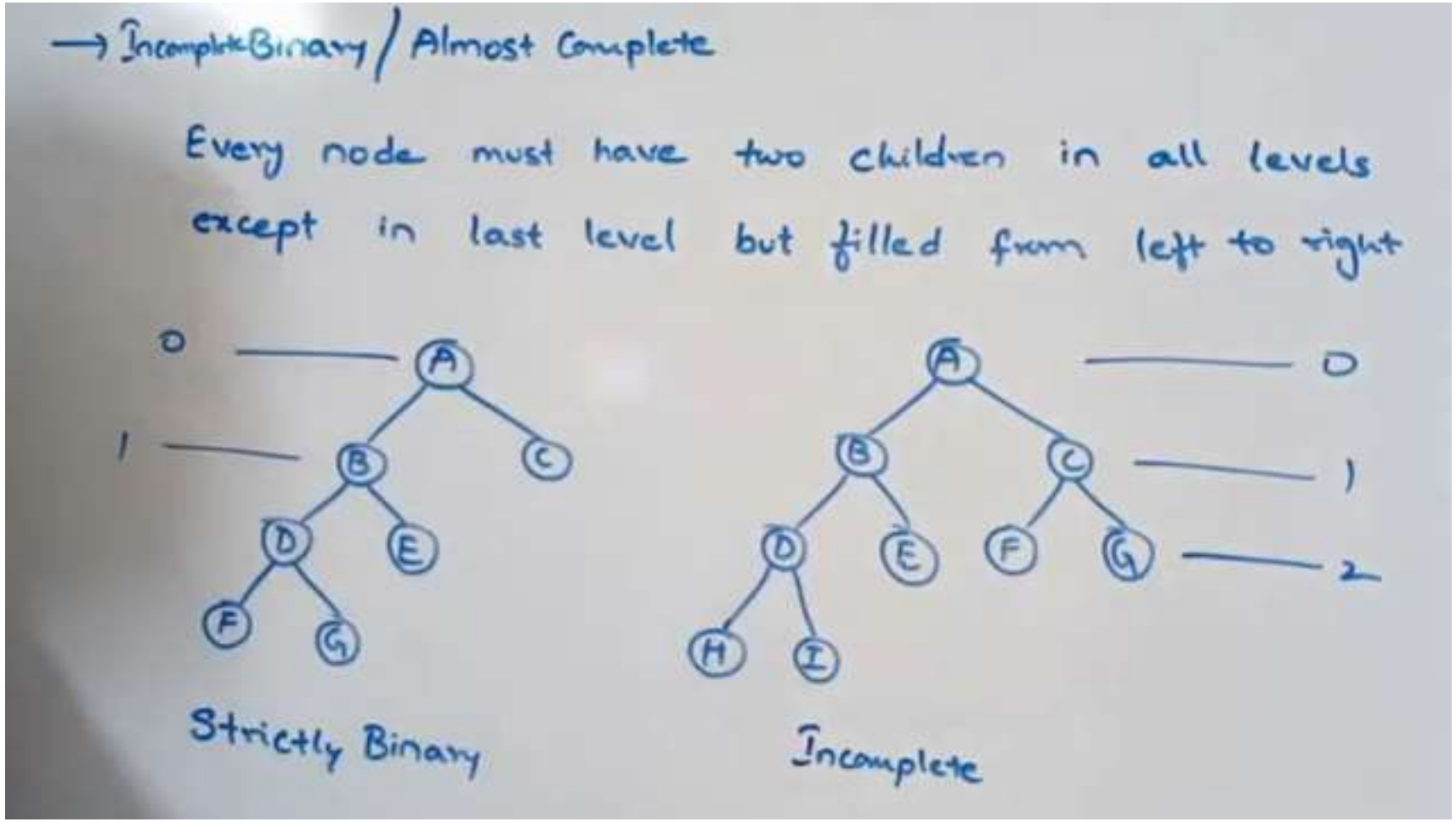
Binary Tree


→ Every node in a Tree should have atmost 2 children
Types of Binary Trees
→ Full Binary Tree / Strictly Binary Tree
→ Almost Complete Binary Tree / Incomplete Binary Tree
→ Complete Binary Tree / Perfect Binary Tree
→ Left Skewed Binary Tree
→ Right Skewed Binary Tree

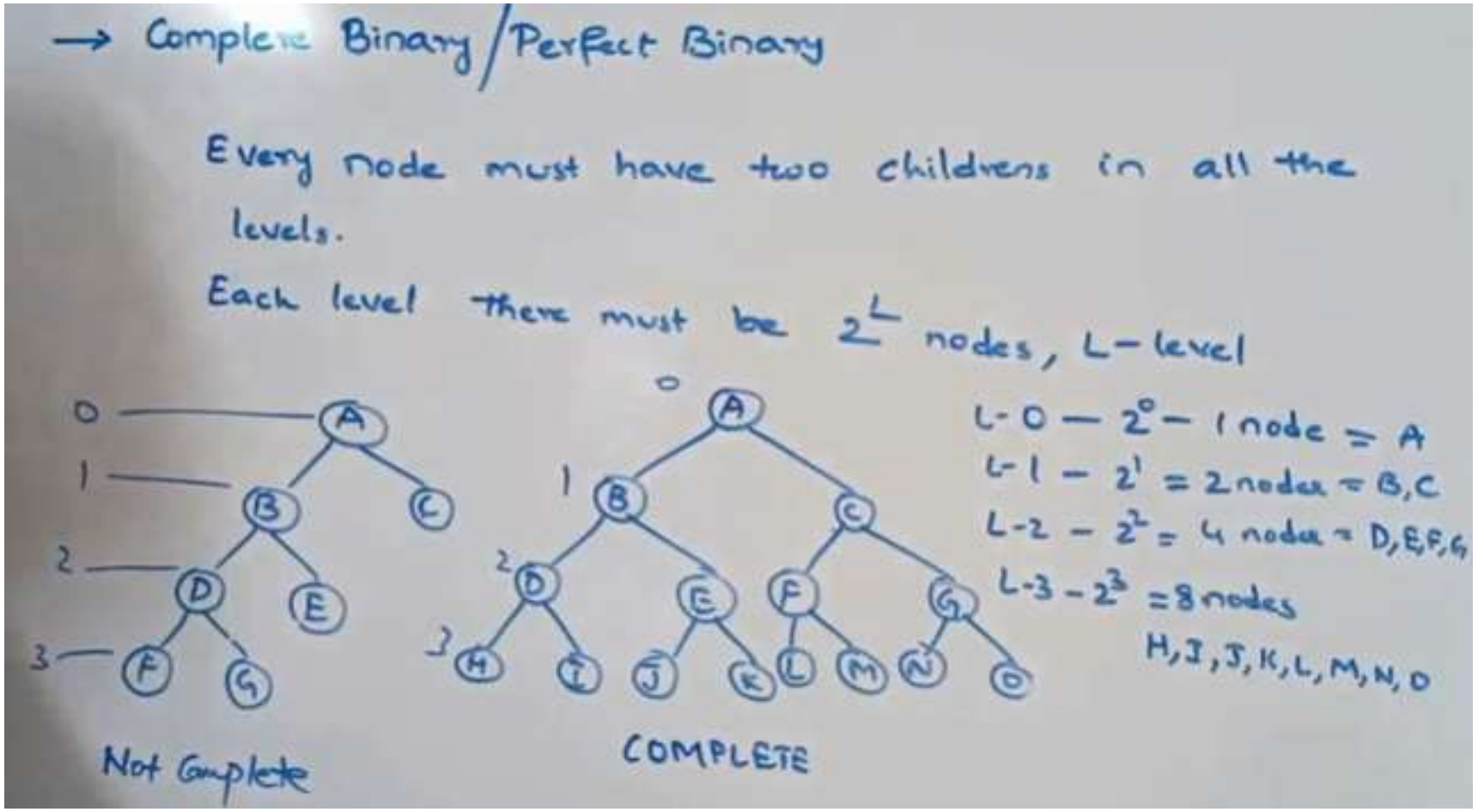
Types of Binary Tree



Types of Binary Tree



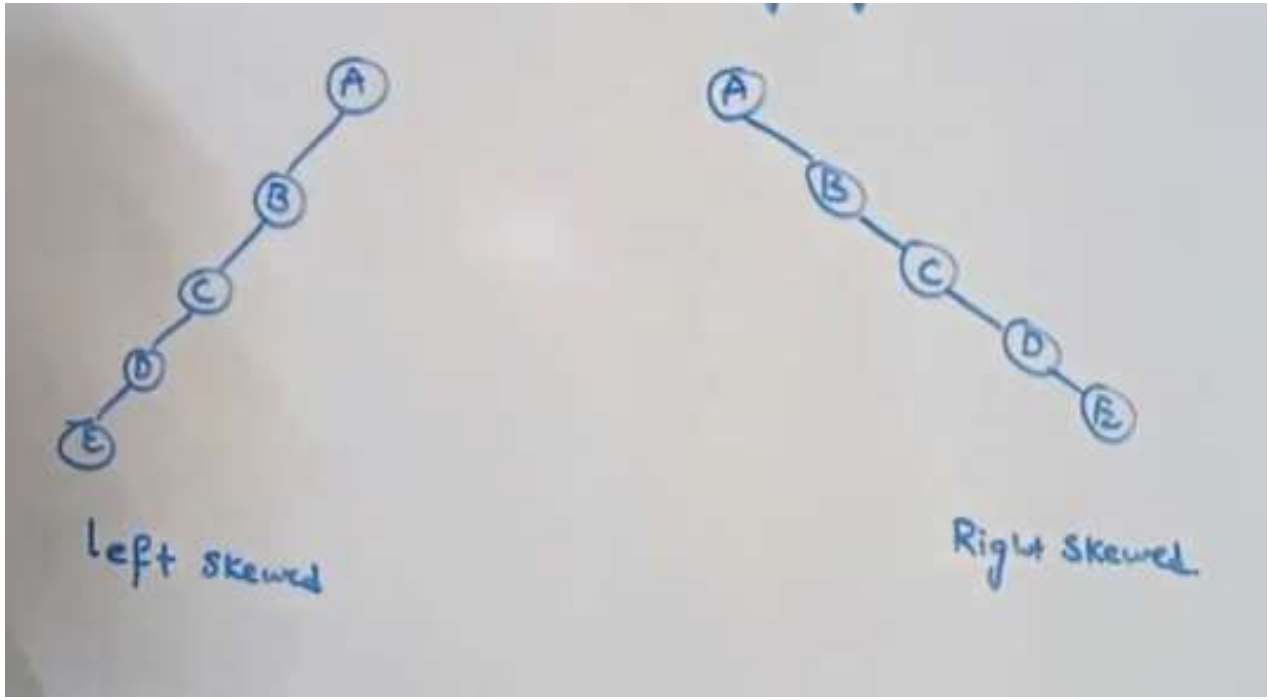
Types of Binary Tree



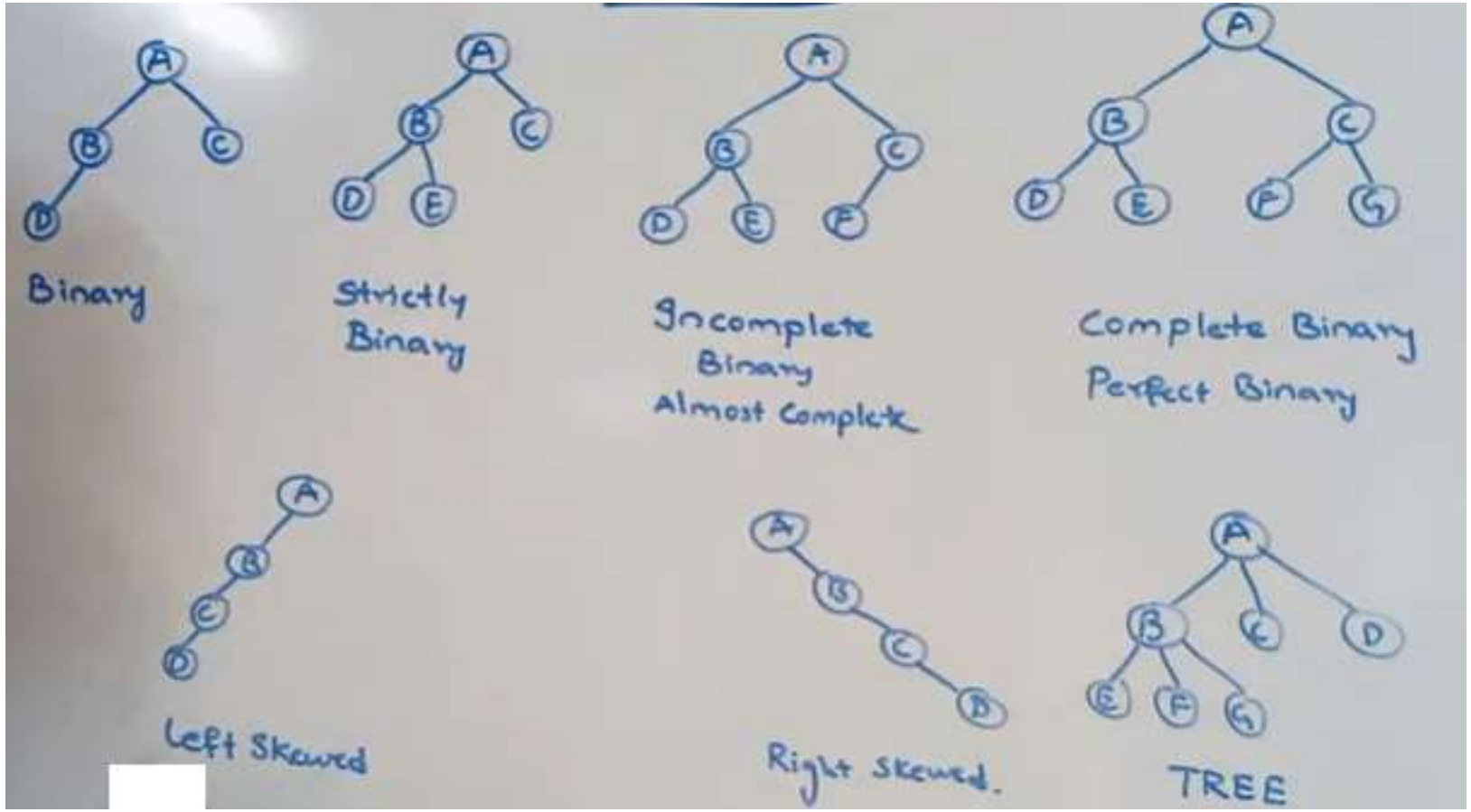
Types of Binary Tree

Left Skewed Tree: Every node should have only left Children

Right Skewed Tree : Every node should have only Right Children

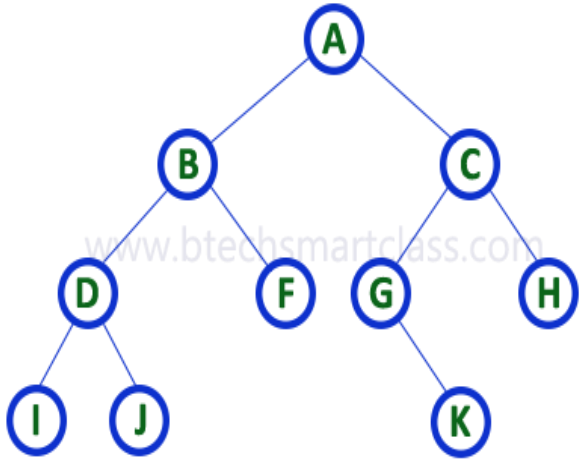


Types of Binary Tree



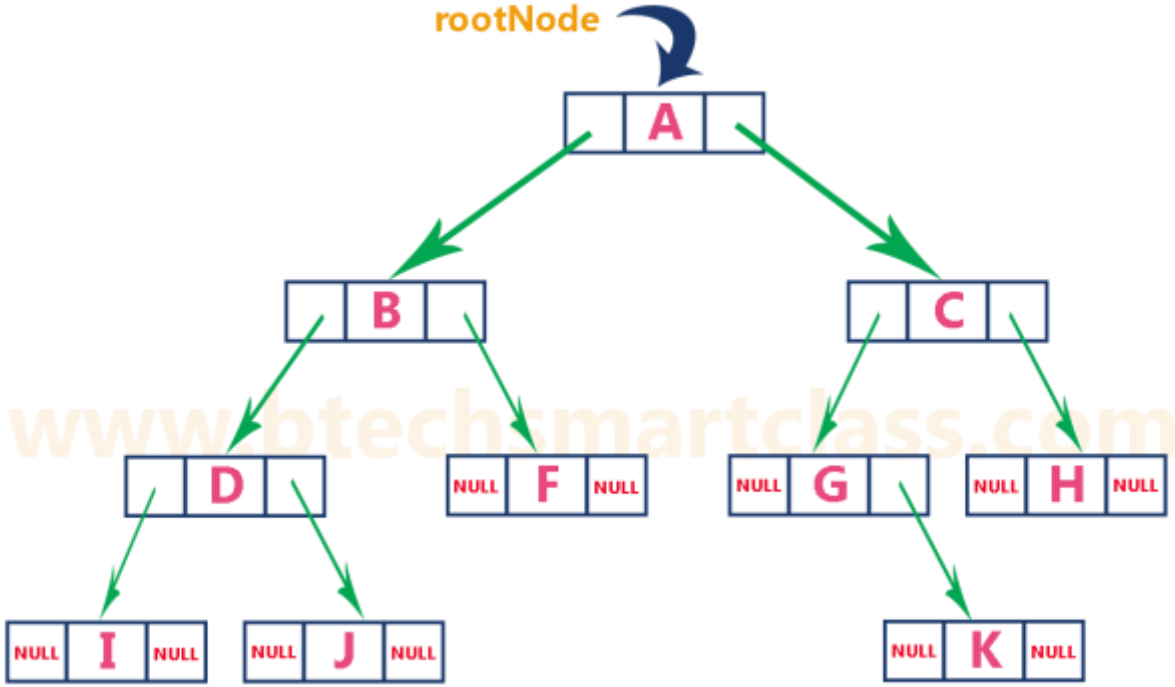
Representation of Binary Tree -Linear Representation

- For any element in position i, the left child is in position 2i,the right child is in position 2i+1 and the parent is in position (i/2)



A	B	C	D	F	G	H	I	J	-	-	-	K	-	-	-	-	-	-	-	-

Linked List Representation



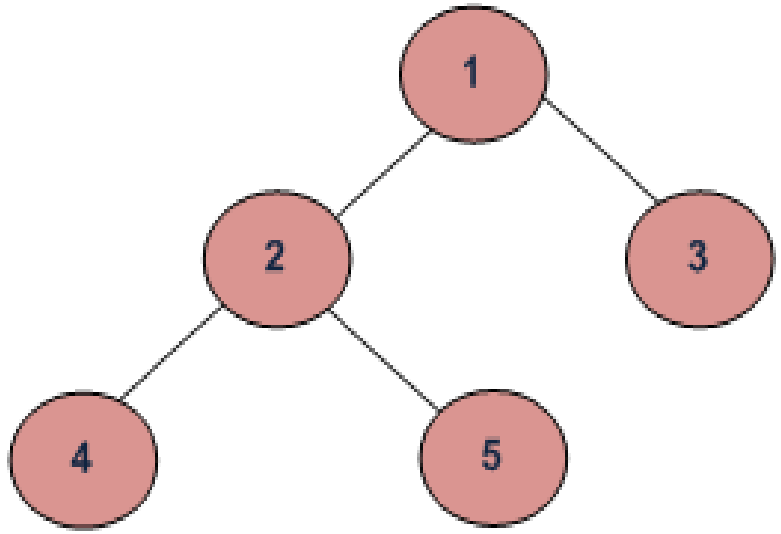
Tree Traversals

- Visiting each node only once
- Three types of tree traversals

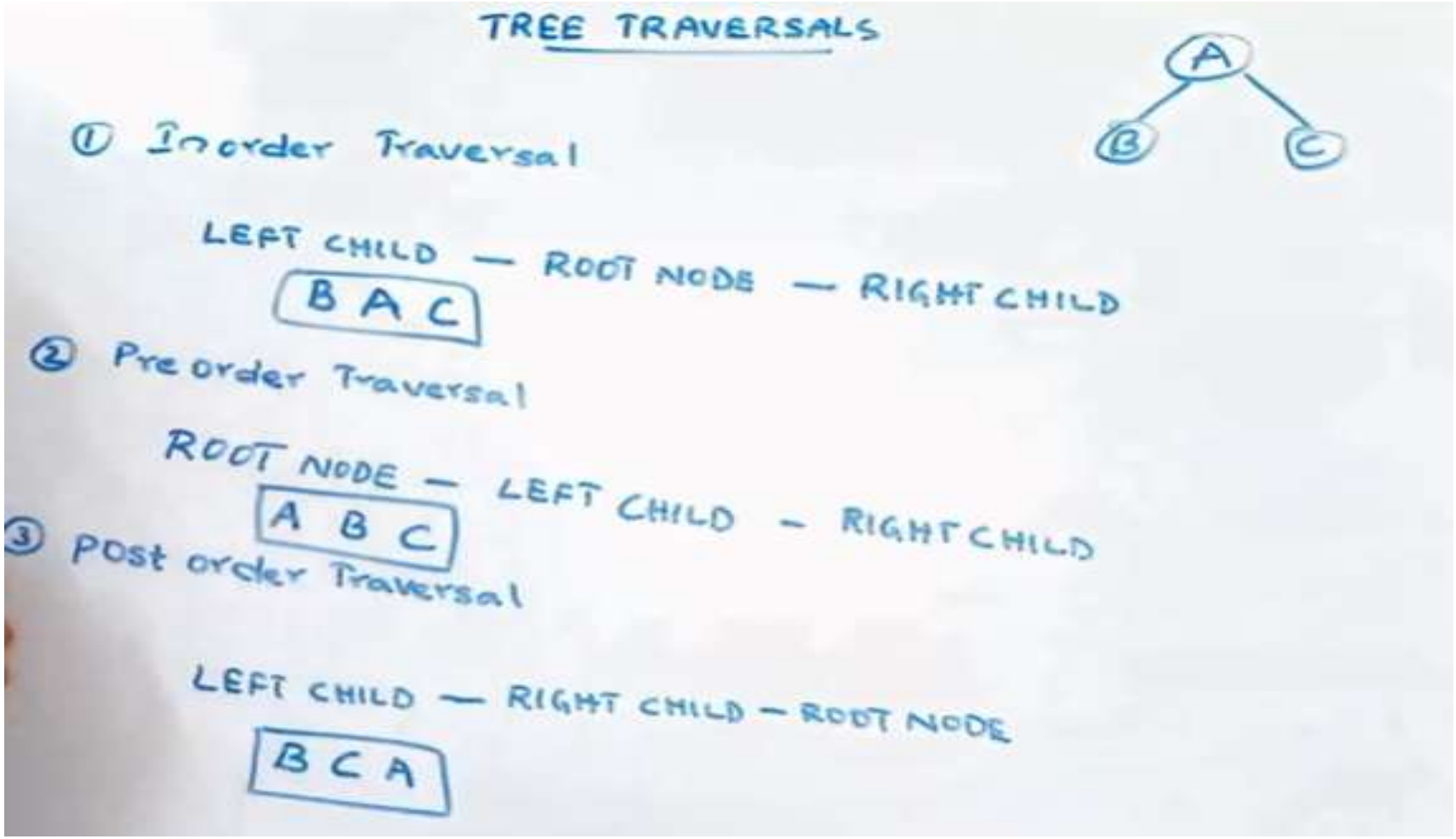
➢Inorder Traversal

➢Preorder Traversal

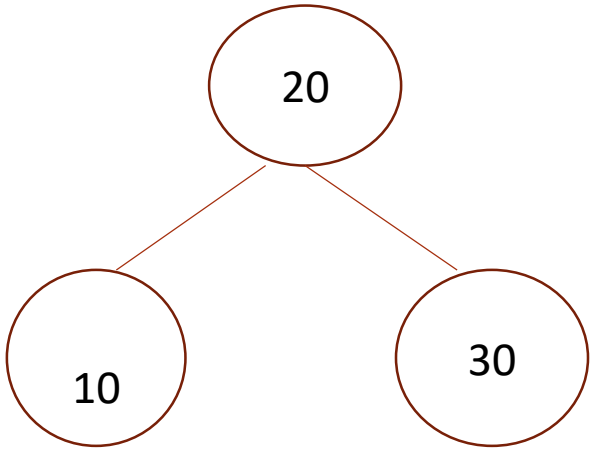
➢Postorder Traversal



Tree Traversals



Examples



- Inorder 10 20 30
- Preorder 20 10 30
- Postorder 10 30 20

Inorder Traversal

Traverse the left subtree in inorder

Visit the root

Traverse the right subtree in inorder

Routine

```
Void Inorder(Tree T)
{
If(T!=NULL)
{
Inorder(T->left);
printElement((T->Element);
Inorder(T->right);
}
}
```

Preorder Traversals

Visit the root

Traverse the left subtree in preorder

Traverse the right subtree in preorder

```
Void Preorder(Tree T)
{
If(T!=NULL)
{
printElement((T->Element);
Preorder(T->left);
Preorder(T->right);
}
}
```

Post order Traversals

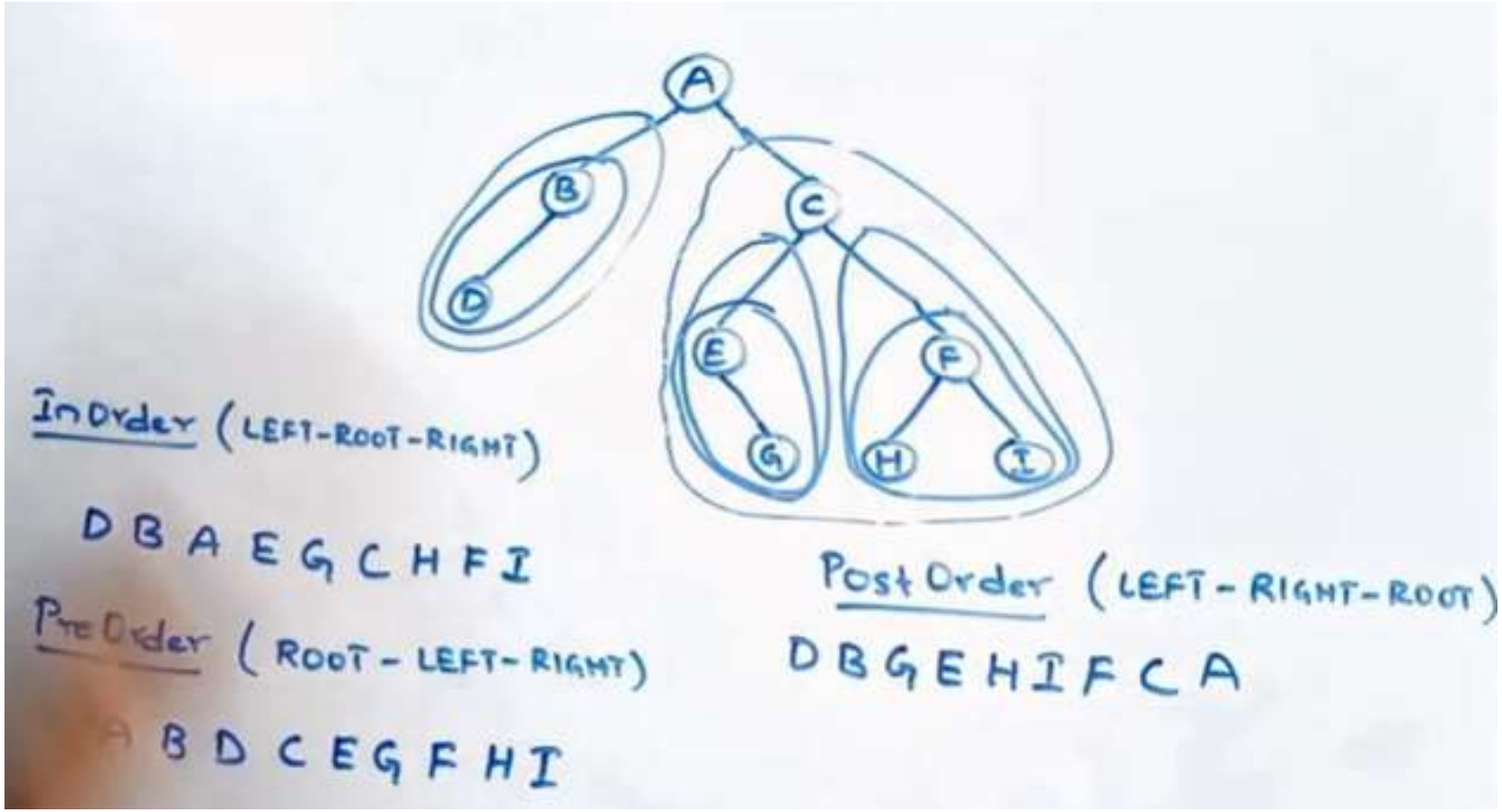
Traverse the left subtree in preorder

Traverse the right subtree in preorder

Visit the root

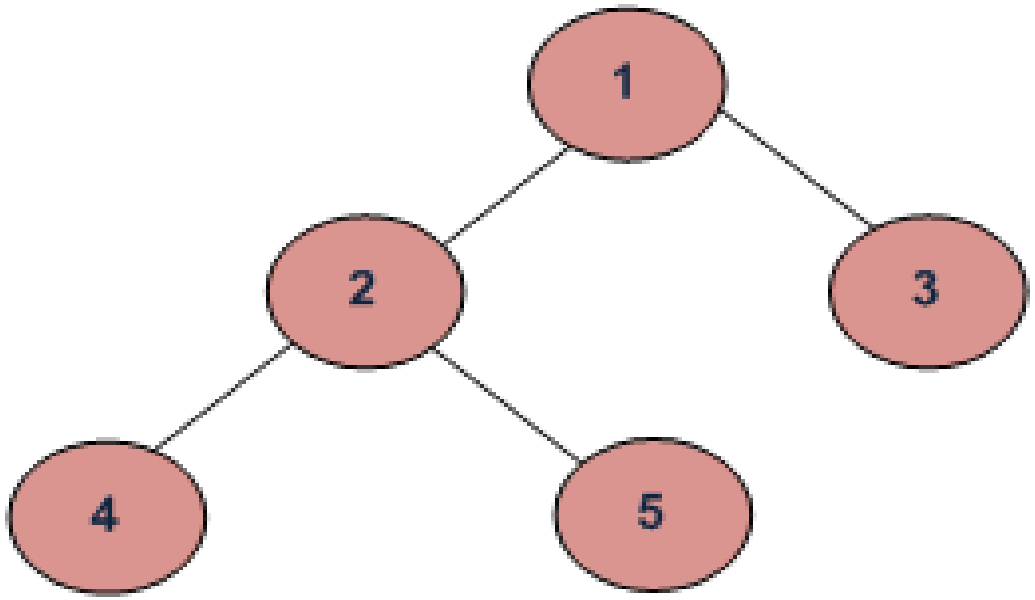
```
Void Postorder(Tree T)
{
If(T!=NULL)
{
Postorder(T->left);
Postorder(T->right);
printElement((T->Element);
}
}
```

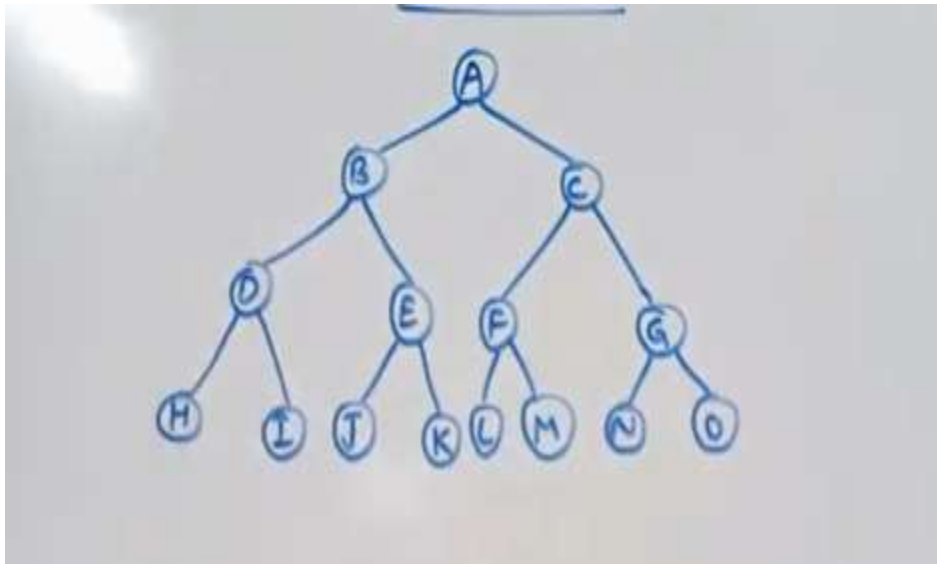
Assessment


A
B
C
D
E
F
G
H
I
InOrder (LEFT-ROOT-RIGHT)
D B A E G C H F I
PreOrder (ROOT - LEFT- RIGHT)
A B D C E G F H I
Post Order (LEFT - RIGHT- ROOT)
D B G E H I F C A

Assessment

Find out the Tree Traversals for the following Tree





References

1. M. A. Weiss, “Data Structures and Algorithm Analysis in C”, Pearson Education, 2nd Edition, 2002.
2. A. V. Aho, J. E. Hopcroft and J. D. Ullman, “Data Structures and Algorithms”, Pearson Education, 2nd Edition, 2007
3. Ashok Kamthane, " Data Structures Using C ", Pearson Education, 2nd Edition, 2012.
4. Sahni Horowitz, “Fundamentals of Data Structures in C”Universities Press; Second edition 2008

Thank You