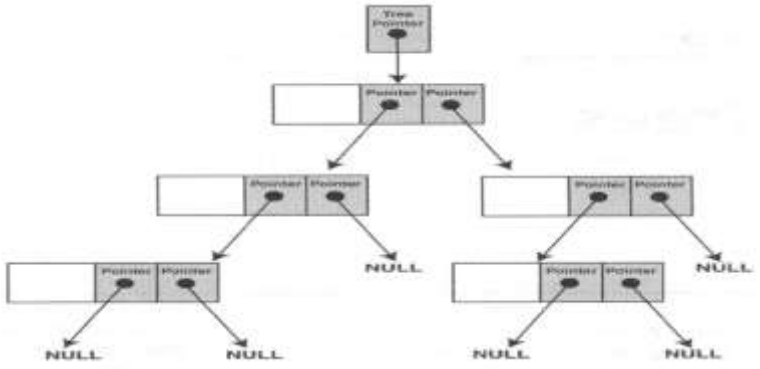
Tree ADT

A tree is a data structure consisting of data nodes connected to each other with pointers. Each node in a tree may be connected to two or more other nodes, rather than the single node allowed in a linked list. The maximum number of nodes to which any single node may be connected is called the **order** of the tree.

binary tree

The simplest tree is of order 2, is called a binary tree. Each node in this TREE contains one or more data fields, and two pointers; one to the left child and the other to the right child. The topmost node in the tree is called the root node. A node with no children is called a leaf (a node both of whose pointers are null). **Definition**: A binary tree is either:

1. an empty tree; or
2. consists of a node, called a root, and two children, left and right, each of which are themselves binary trees.



binary search trees (BST)

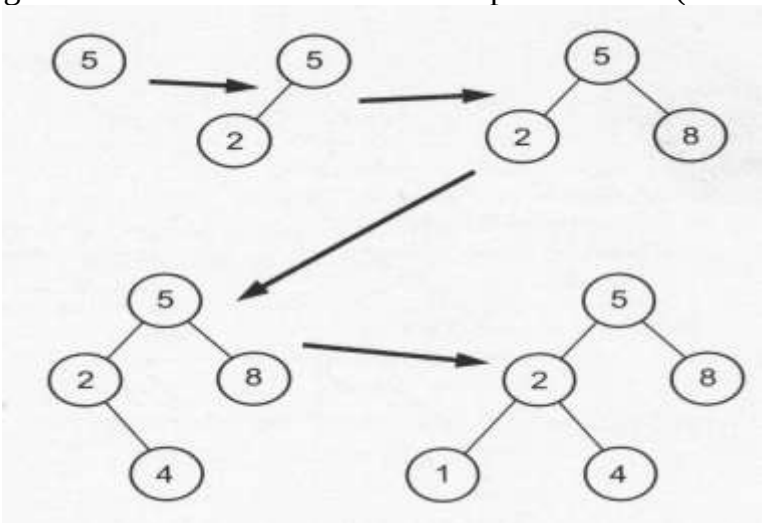
binary trees used for sorting and searching data are called, binary search trees.

® Trees (binary and otherwise) basic types of operations

- *Inserting a new node.*
- *Deleting a node.*
- *Listing or visiting the nodes of the tree* (traversal)

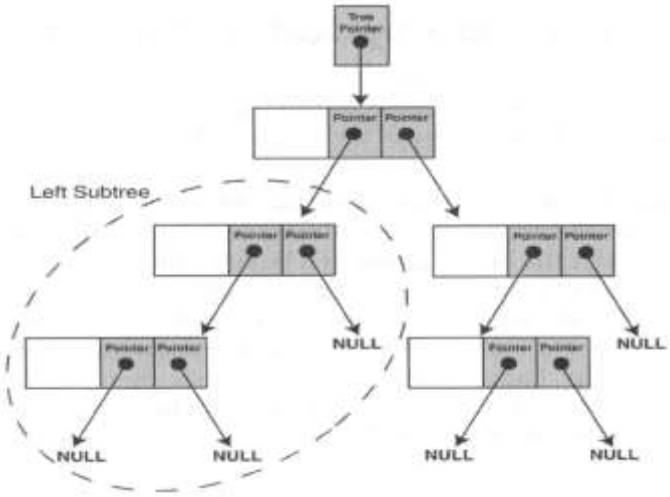
How to sort list of integers [5, 2, 8, 4, and 1] into ascending order using BST

1. The first stage involves inserting the integers into a binary search tree
2. The second stage is how to list the node in some specific order (inorder traversal)

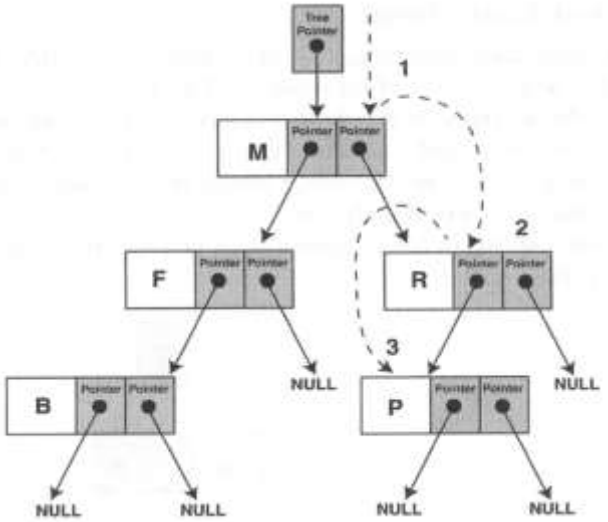


The completed list after stage two is 1, 2, 4, 5, 8, which is the correctly sorted list.

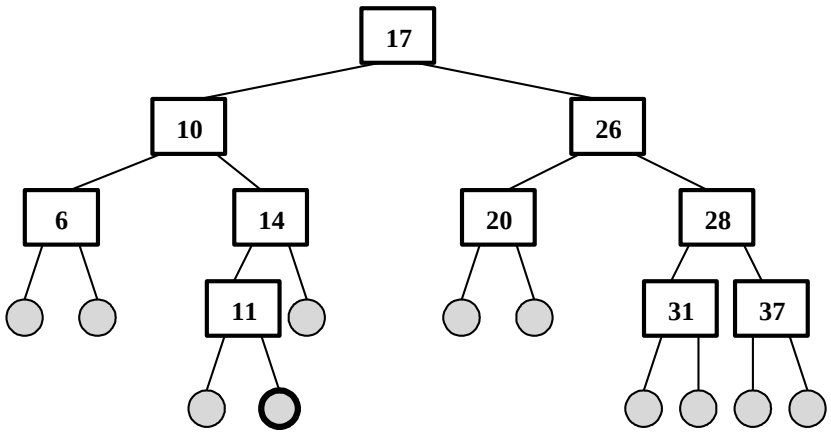
Subtree: is an entire branch of the tree, from one particular node down



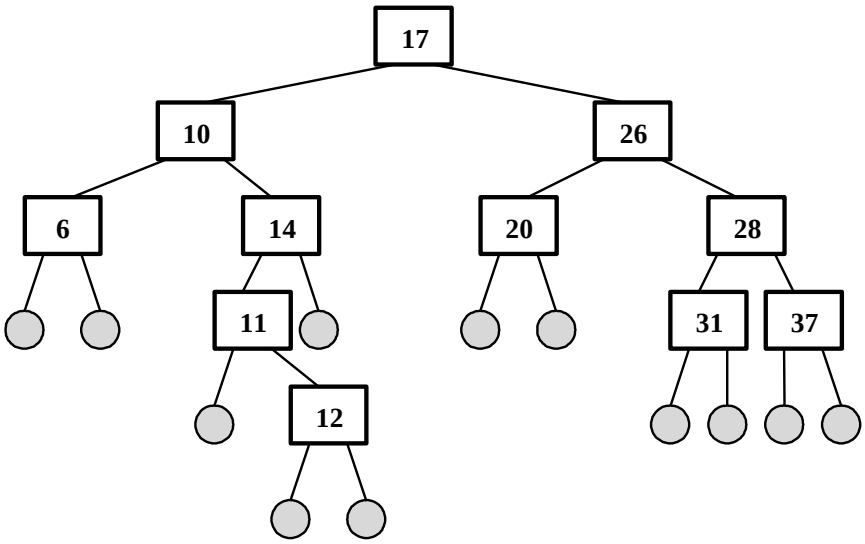
How to search for P



How to insert 12 in the tree below



The Binary Search Tree after 12 inserted



Traversal

There are 3 common methods for traversing a binary tree and processing the value of each node:

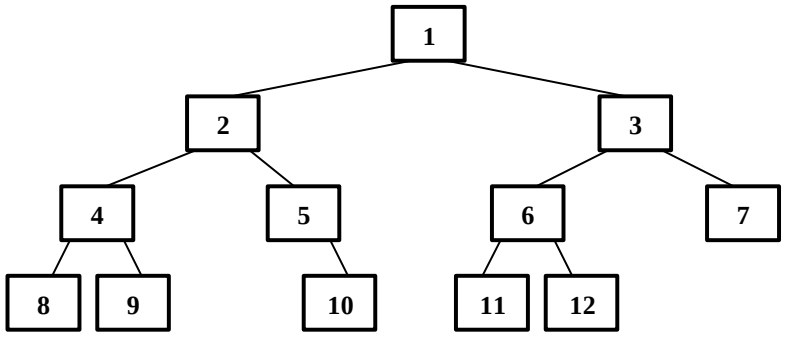
- ***Inorder traversal:***
  1. The node's left subtree is traversed.
  2. The node's data is processed.
  3. The node's right subtree is traversed

• ***Preorder traversal:***

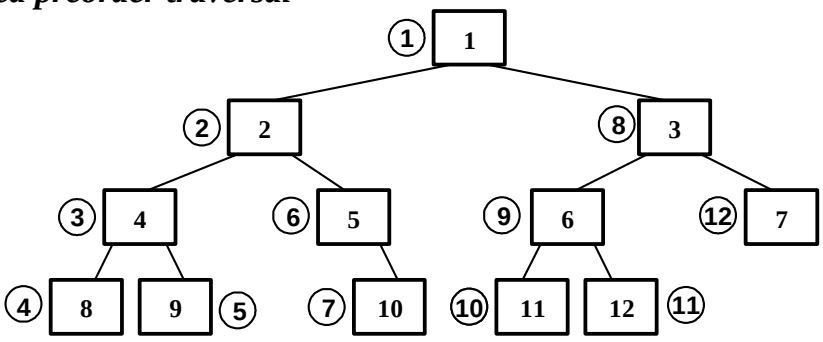
1. The node's data is processed
2. The node's left subtree is traversed.
3. The node's right subtree is traversed.

• ***Postorder traversal:***

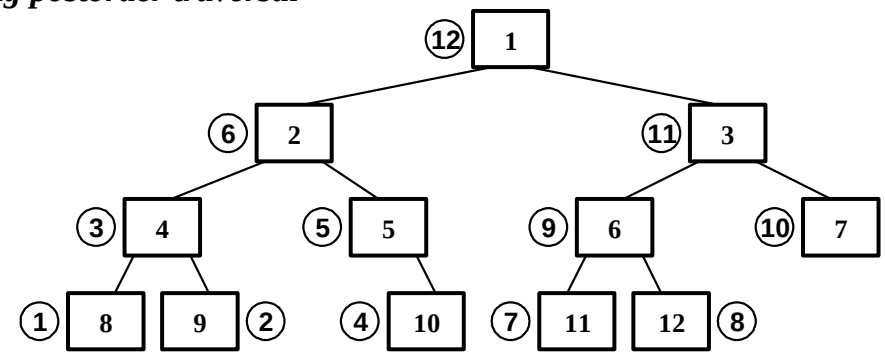
1. The node's left subtree is traversed.
2. The node's right subtree is traversed.
3. The node's data is processed.



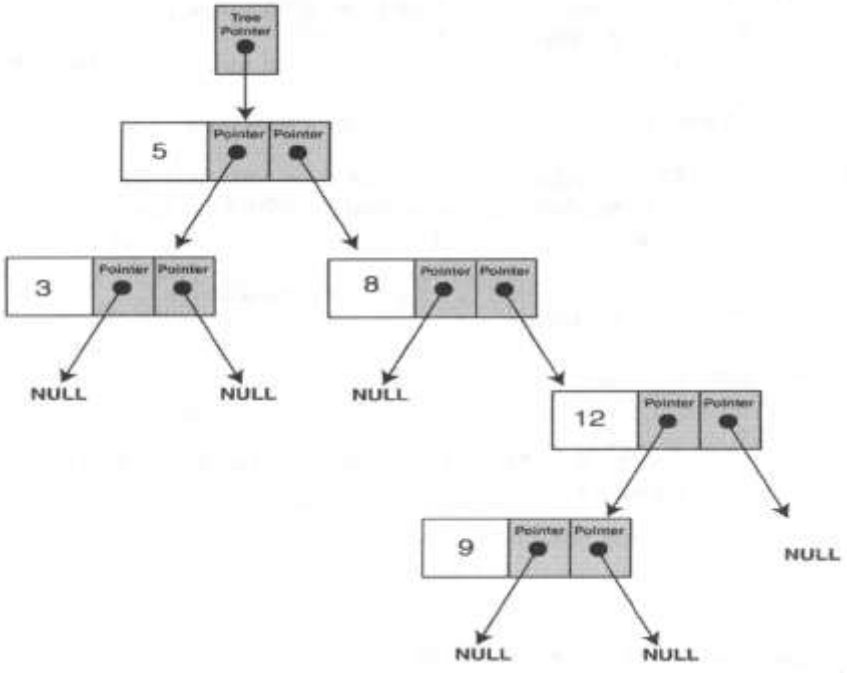
Tree after completed preorder traversal



Tree visited using postorder traversal




```
void main(void)
{
        BinaryTree   tree;
        tree.insertNode(5);
        tree.insertNode(8);
        tree.insertNode(3);
        tree.insertNode(12);
        tree.insertNode(9);
}
```



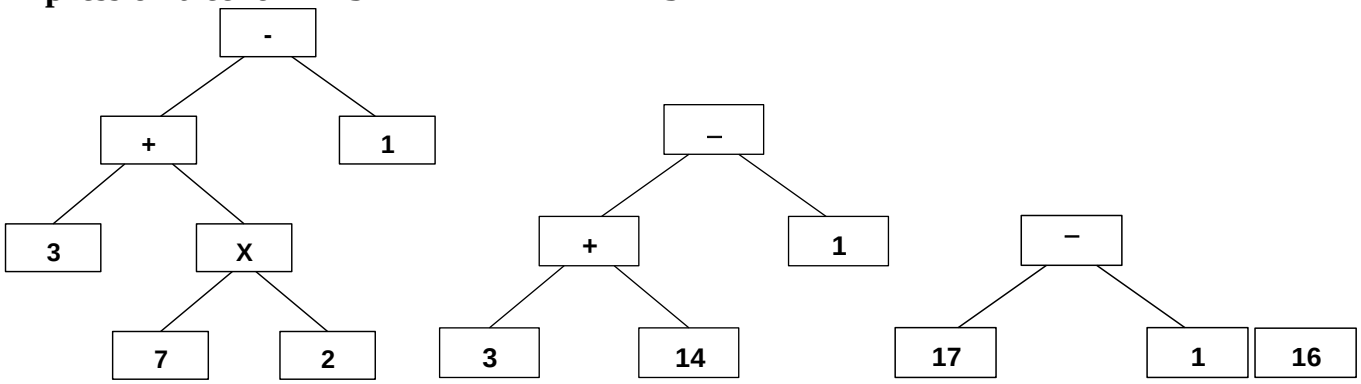
- **Inorder traversal:**

  3
  5
  8
  9
  12

- **Preorder traversal:**
  5
  3
  8
  12
  9

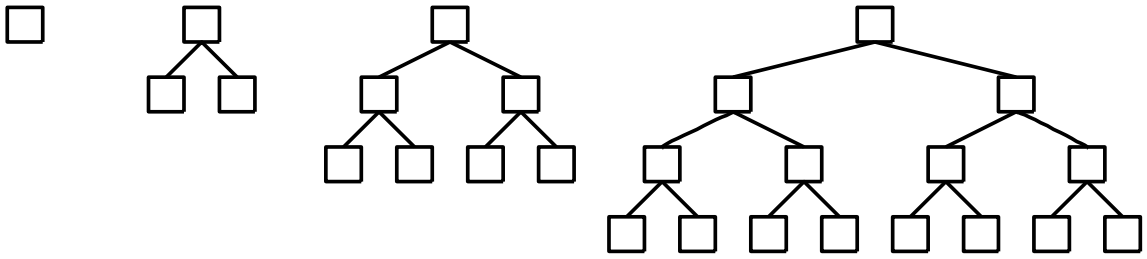
- **Postorder traversal:**
  3
  9
  12
  8
  5

Expression tree for $3 + 7 \times 2 - 1$ **= - + 3 × 7 2 1**



??? Create an expression tree for the following expression: 6 * 4 / 2 + 7 - 5 * 3

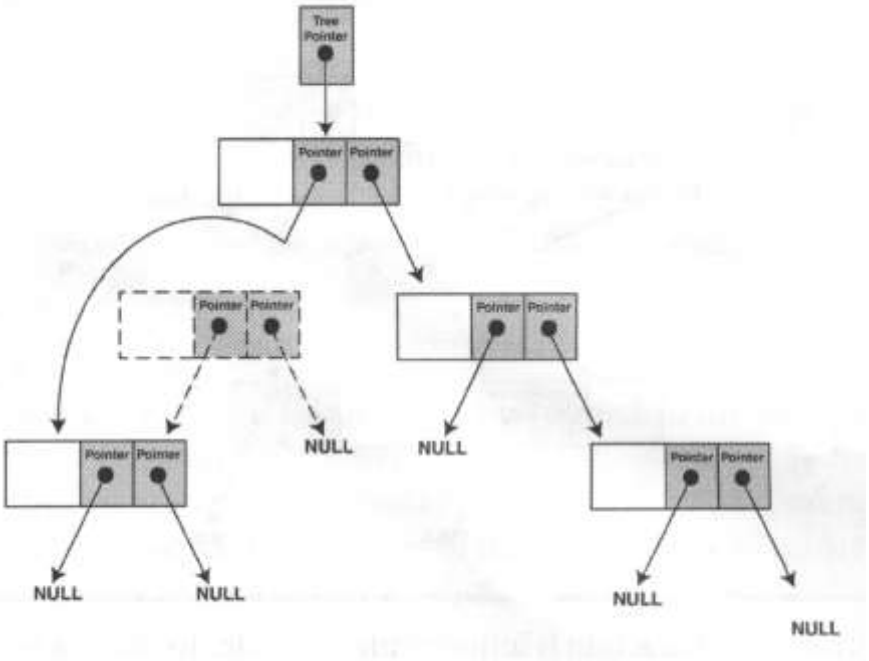
Shallowest trees for n=1, n=3, n=7, and n=15 → for n nodes, number of levels = $\log_2 (n+1)$



Deleting a node

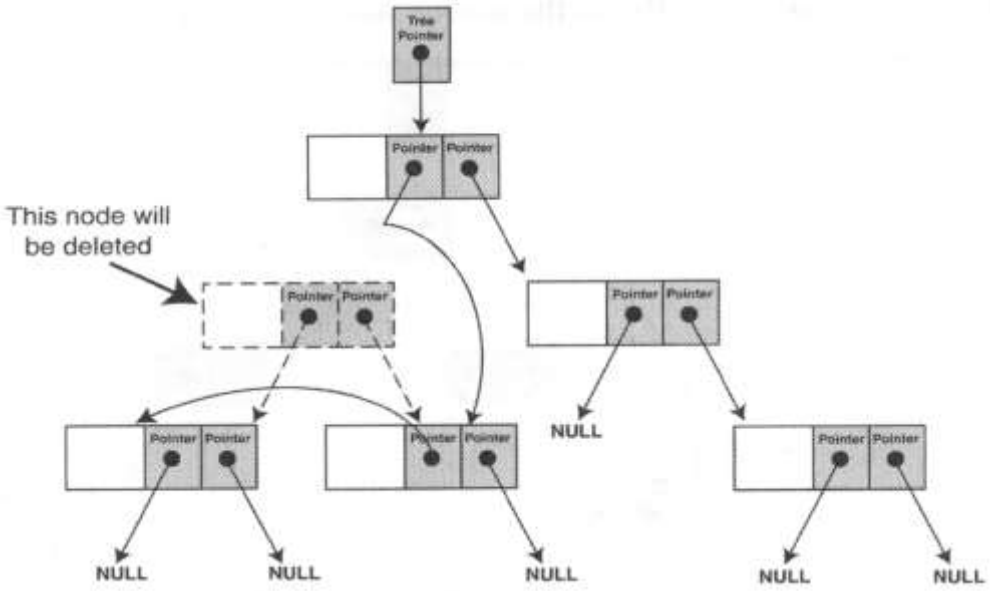
Deleting a leaf node, we find its parent and set the child pointer that links to it to NULL, and then free the node's memory. There are 2 possible situations to face when deleting a non-leaf node:

A) the node has one child, or
B) the node has two children.



(A) the node has one child

(B) Node has two child, attach the node's right subtree to the parent, and then find a position in the right subtree to attach the left subtree.



[١]Identify true or False and rewrite the correct version of False one.

A- A tree is data structure consisting of data nodes connected to with pointers **(true)**

B- The simple tree is of order 3 and is called binary tree. **(false → order 2)**

C- A binary tree node with no children is called a root.**(false → leaf)**

D- A binary tree is a linear linked list where each node may point to two other nodes.**(false → non-linear)**

E- A binary trees are suitable for algorithms that must search large amounts of information.

F- Inorder, Preorder and Postorder are common methods for traversing a binary tree and processing the value of each node.**(true)**

[٢]What are the basic types of operations for tree.

[٣]draw the structure of a binary tree, and identify:

* Root - node
* Right childs
* Left childs
* Leaf - node

each node contain data and two pointers.

[4] Write the algorithm for :-

* Inorder,
* Preorder and
* Postorder for traversing a binary tree

[٥]The following is the code for insertNode member function of IntBinaryTree, Identify the errors and rewrite the correct version.

void IntBinaryTree::deletNode(int num) TreeNode newNode, // Pointer to a new node
nodePtr; // Pointer to traverse the tree // Create a new node
newNode = new TreeNode;
newNode->value = newNode;

```
newNode->left = newNode->left+ 1 = NULL; if (!root)          // Is the tree empty?
leof = newNode;
else nodePtr = root;
while (nodePtr != root(
 }if (num < nodePtr->value(
 }if (nodePtr->right(
nodePtr = nodePtr->left;
else { nodePtr->left = newNode; break;} else if (num > nodePtr-1 ->value(
}if (nodePtr->right(
nodePtr = nodePtr->right;
else {nodePtr-> left = newNode; break;}
else {cout « "Duplicate found in tree.\n"; break;}
```

The correct version of the program

[6] This program demonstrates the insert function of previous question that builds a binary tree with 5 nodes. Draw the structure of the binary tree built by the program.

```
#include <iostream.h>
#include "IntBinaryTree.h"
void main(void)
{ IntBinaryTree tree; cout « "Inserting nodes.";
tree insertNode(5); tree insertNode(8); tree insertNode(3); tree.insertNode(12);
tree.insertNode(9); cout « "Done.\n"; }
```

[٧] Suppose the following values are inserted into a binary tree, in the order given:

2,17,9,10,22,24,30,18, 3,14, 20 ,

Draw a diagram of the resulting binary tree.

[٨]Draw a diagram of the resulting binary tree after the following operations-:

insert 5, 8, 3 , 12, 9,
remove 8, 12
insert 10.

A- Fill in the blank

1-The first node in a binary tree is called the_____
2- A node with no children is called a ___.
3- A _____ is an entire branch of the tree, from one particular node down.
4- The three common types of traversal with a binary tree are _____, _____, and _____

B- True or False

1. Each node in a binary tree must have at least two children.
2. When a node is inserted into a tree, it must be inserted as a leaf node.
3. Values stored in current node's left subtree are less than value stored in the current node.
4. The shape of a binary tree is determined by the order in which values are inserted.
5. In inorder traversal, node's data is processed first, then the left and right nodes are visited.

C- insert in a binary tree : 12, 7, 9, 10, 22, 24, 30, 18, 3, 14, 20