# Difference between Binary Tree and Binary Search Tree

**Binary Tree Data Structure**

A tree whose elements have at most 2 children is called a binary tree. Since each element in a binary tree can have only 2 children, we typically name them the left and right child.



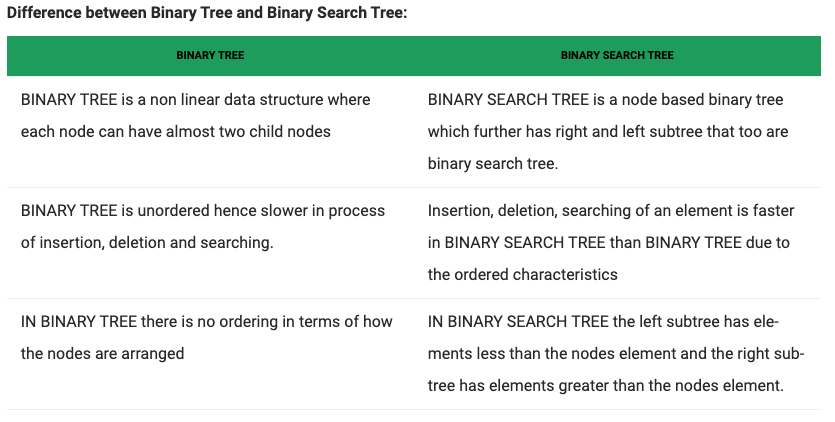
**Binary Search Tree Data Structure**

Binary Search Tree is a node-based binary tree data structure which has the following properties:

* The left subtree of a node contains only nodes with keys lesser than the node’s key.
* The right subtree of a node contains only nodes with keys greater than the node’s key.
* The left and right subtree each must also be a binary search tree.
* There must be no duplicate nodes.



<https://www.youtube.com/watch?v=5a9NUeRsBPo>



Traverse a Binary Tree or Binary Search Tree using DF or BF

\*Note Binary Tree isn’t a linear data structure like array or list or linkedlist. When moving through a linear data structure there is only one next node. With a non-linear data structure there are two possible two directions (left or right node)

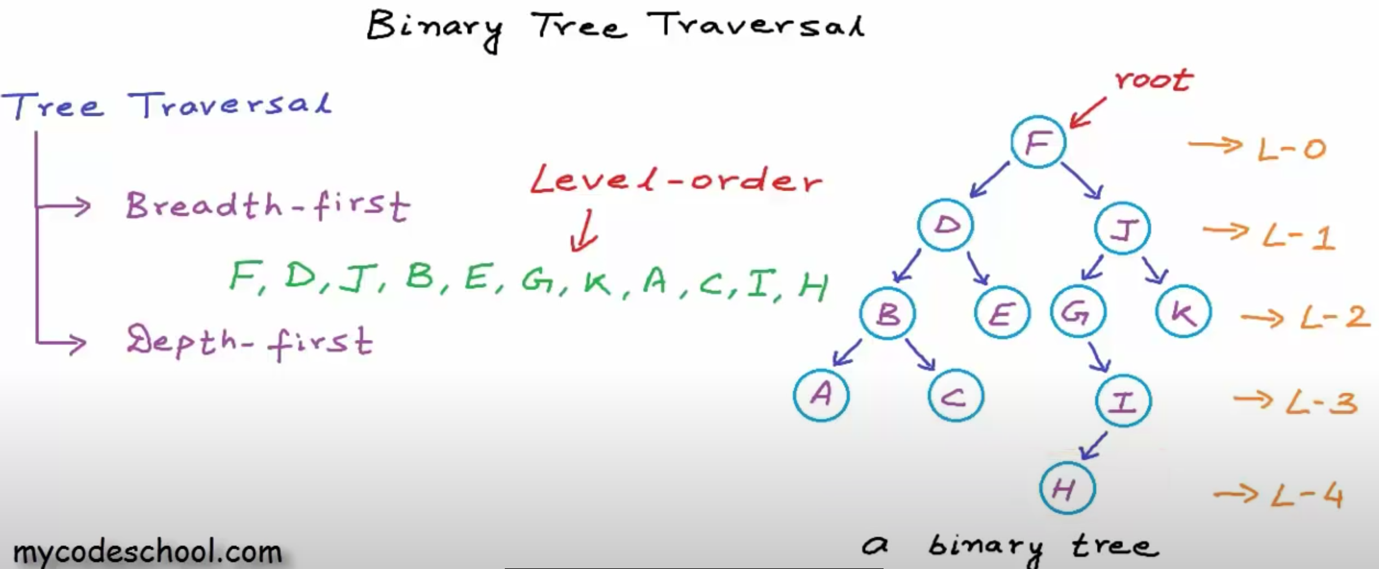
Tree Travesral – process of visiting (read/process data in node) each node in the tree exactly once in some order

Tree Traversal

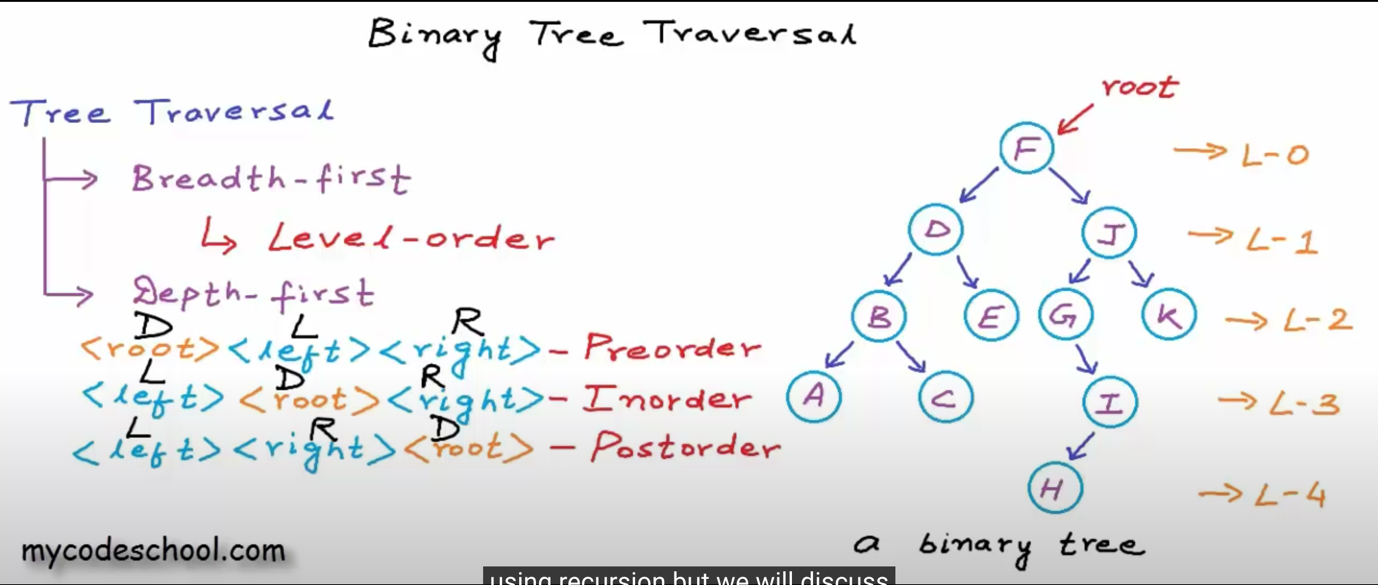
Breadth-first

Depth-first

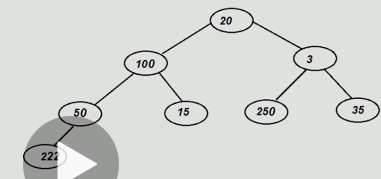
In bredth first approach we visit all of the nodes at the same level, before visiting the nodes at the next level. We go level by level from left to right.



In depth first traversal, the algorithm explores as far as possible along each branch before backtracking.



Algorithm for BFS Level Order Traversal



levelOrderTraversal(root)

create a Queue(Q)

enqueue(root)

while(Queue is not empty)

enqueue() the child of first element

deque() and print

