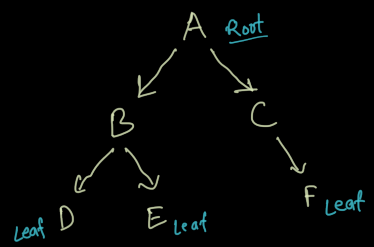
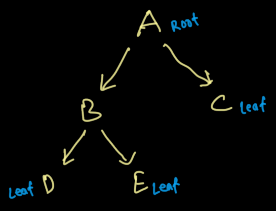
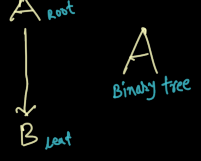
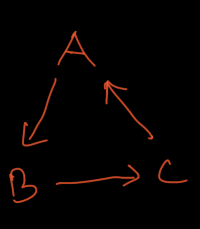
Binary Tree

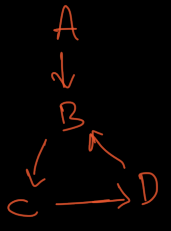


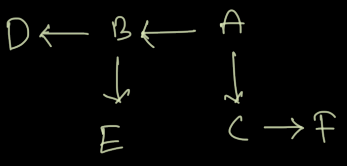
→ At most 2 children (can have less child).
→ Exactly 1 root.
→ Exactly 1 path b/w root and any node.
→ Empty tree can be considered as binary tree.





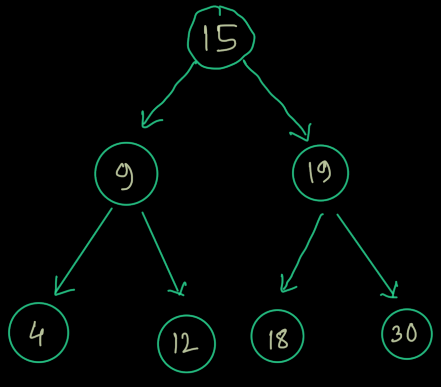






⇒ Binary search tree.
↳ Left sub keys are less than Root key.
↳ Right sub keys are greater than Root key.
↳ Left & Right subtree must be BST (Binary search tree).
↳ NO Duplicate.

Binary Search Tree (BST)

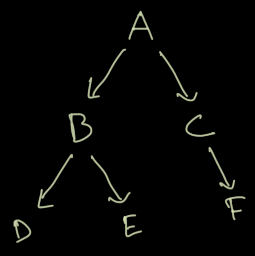


BST Node
↳ key
↳ Left
↳ Right

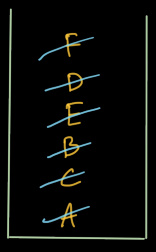
BST Tree
↳ Root

⇒ Deletion of node in BST
1. Leaf node (no children)
2. 1 child
3. 2 Node
↳ In order Successor
(smallest key in right subtree)

Depth First Traversal



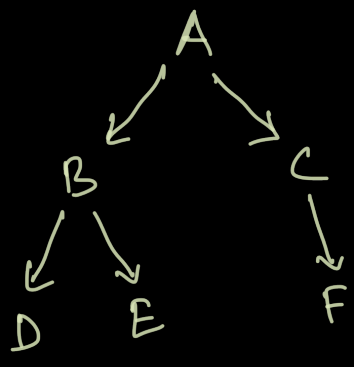
Values = A, B, D, E, C, F
↳ Go depth first
↳ stack



Stack == ~~Yes~~ NO

Values = A, B, D, E, C, F

Breadth first Traversal



Values = A, B, C, D, E, F

→

~~F E D C B A~~

Values: A, B, C, D, E, F

Q == yes