Tree Data Structure

- Nonlinear hierarchical D.S.
- Consits of nodes connected by edges.

→ Why Tree D.S. ?

In linear D.S. like Stack, Queue, Arrays and Linked List time complenity increases with increase in data size.

→ Tree Terminologies :-

- Node - It consist of a key or value and pointer to its child node.
  - Last node of each path are Leaf nodes.
  - First node is the root node (Topmost node of tree
- Edge - Link between two nodes
- Height of Node - (Longest path from the node to a Leaf node)
  - Height of Tree - Root node to Leaf node.
- Depth of Node - No. of edges from root to the node.

→ Types of tree :-

i) Binary tree

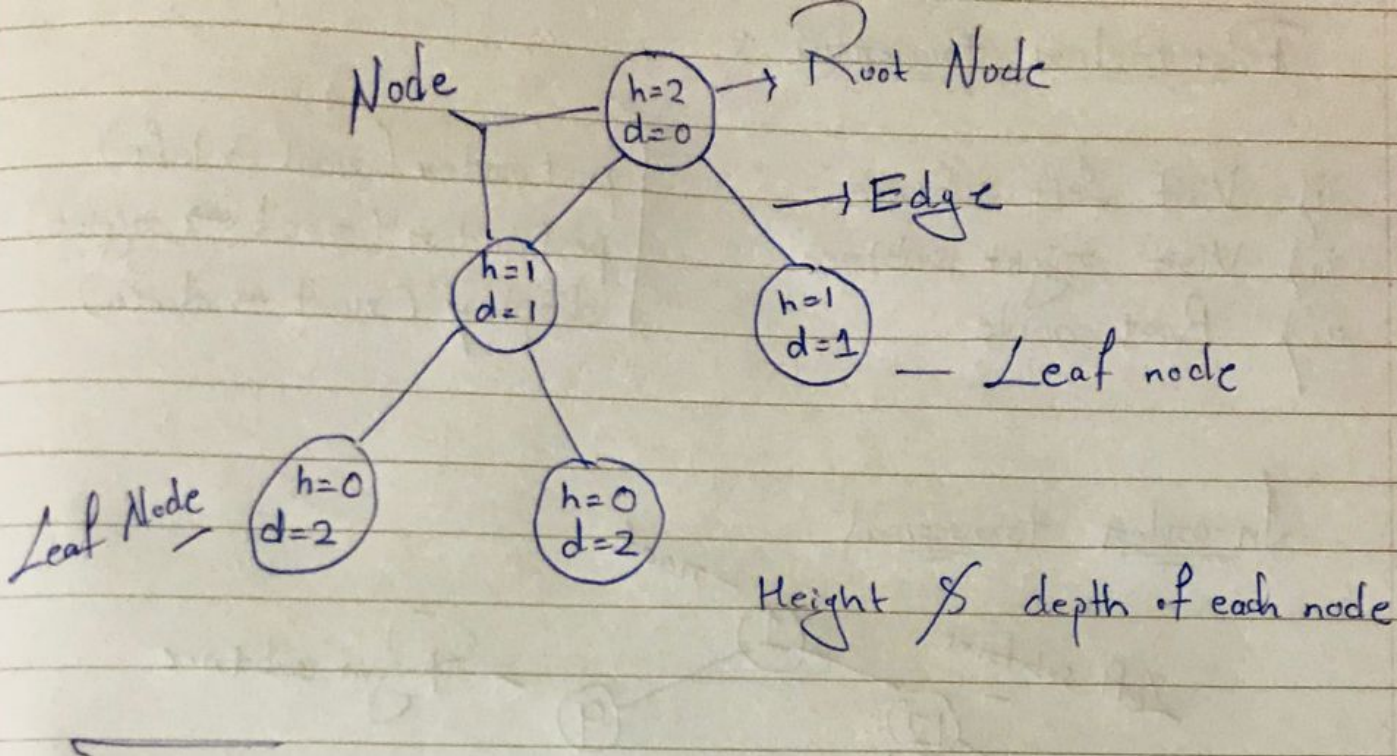
ii) Binary Search tree

iii) AVL

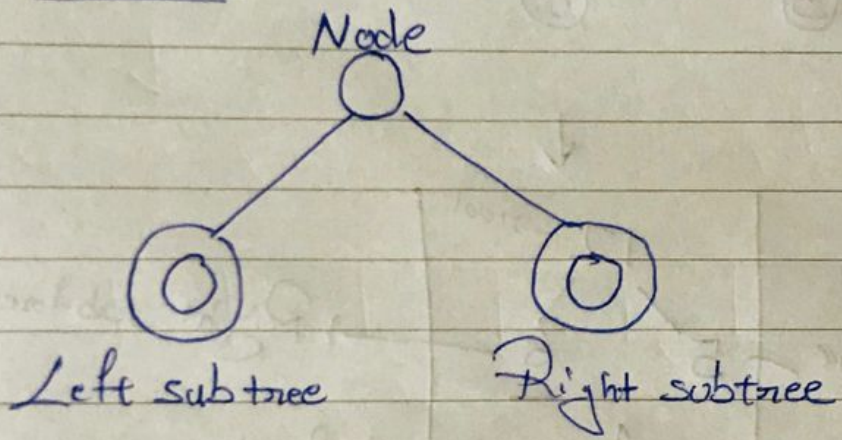
iv) B Tree

v) B+ Tree

vi) Red-Black Tree



Tree traversal :-



Inorder traversal

i) Visist all the node in Left subtree.
ii) Then the root
iii) Visit all the node in the right subtree.

```
inorder (root → left
display (root → data
inorder (root → righ
```

Preorder traversal

i) Visit root
ii) All Left subtree
iii) All right subtree

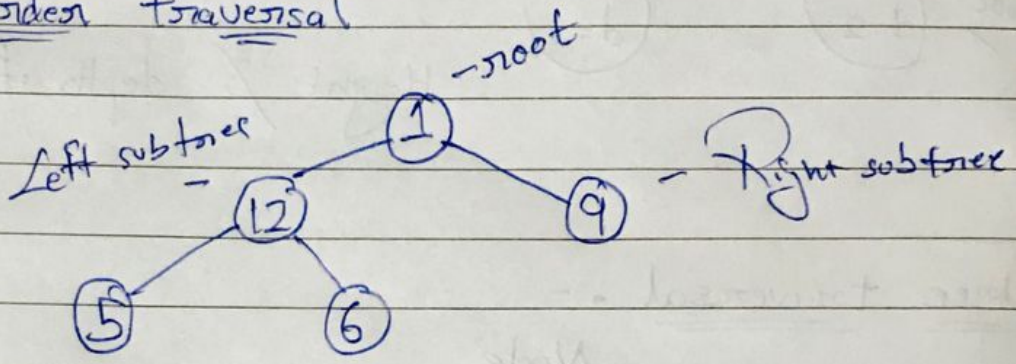
```
display (root → data)
Preorder (root → left)
preorder (root → right)
```

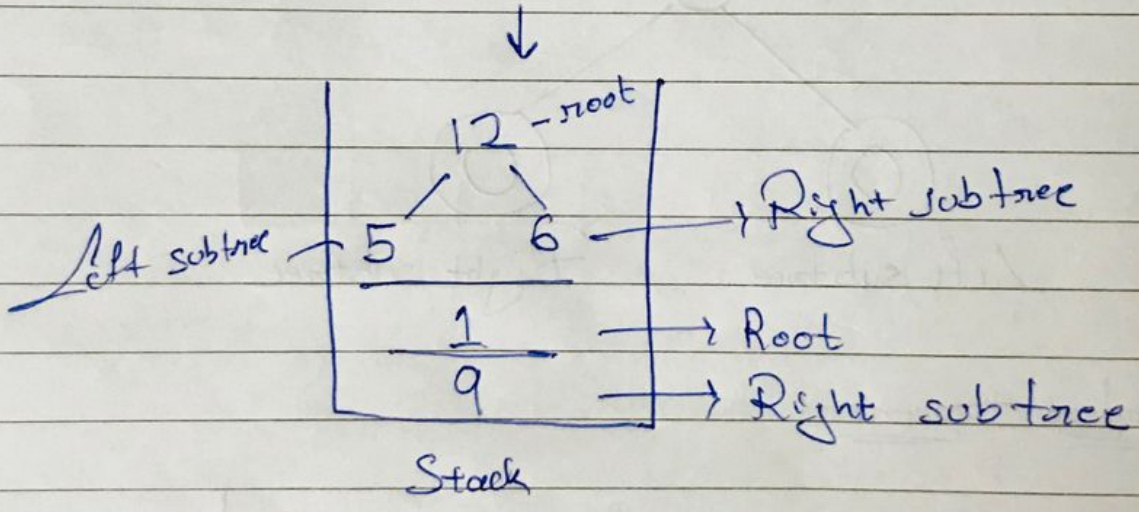
Postorder traversal :

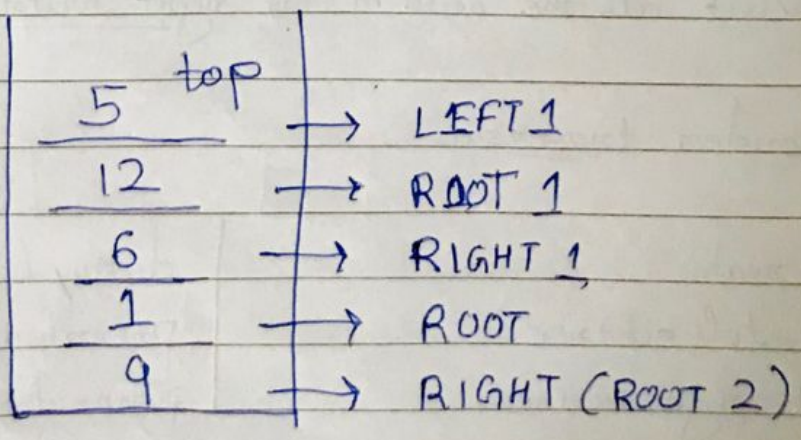
i) Visit Left subtree
ii) Visit right subtree
iii) Root node

postorder (root → left)
postorder (root → right)
display (root → data)

In-order traversal







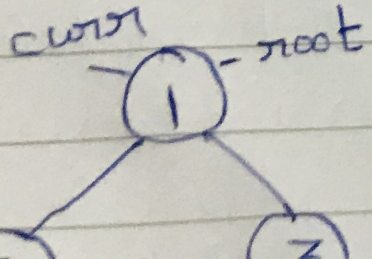
Final Stack

Inorder Tree traversal without Recursion & Stack!
(Morris Traversal)

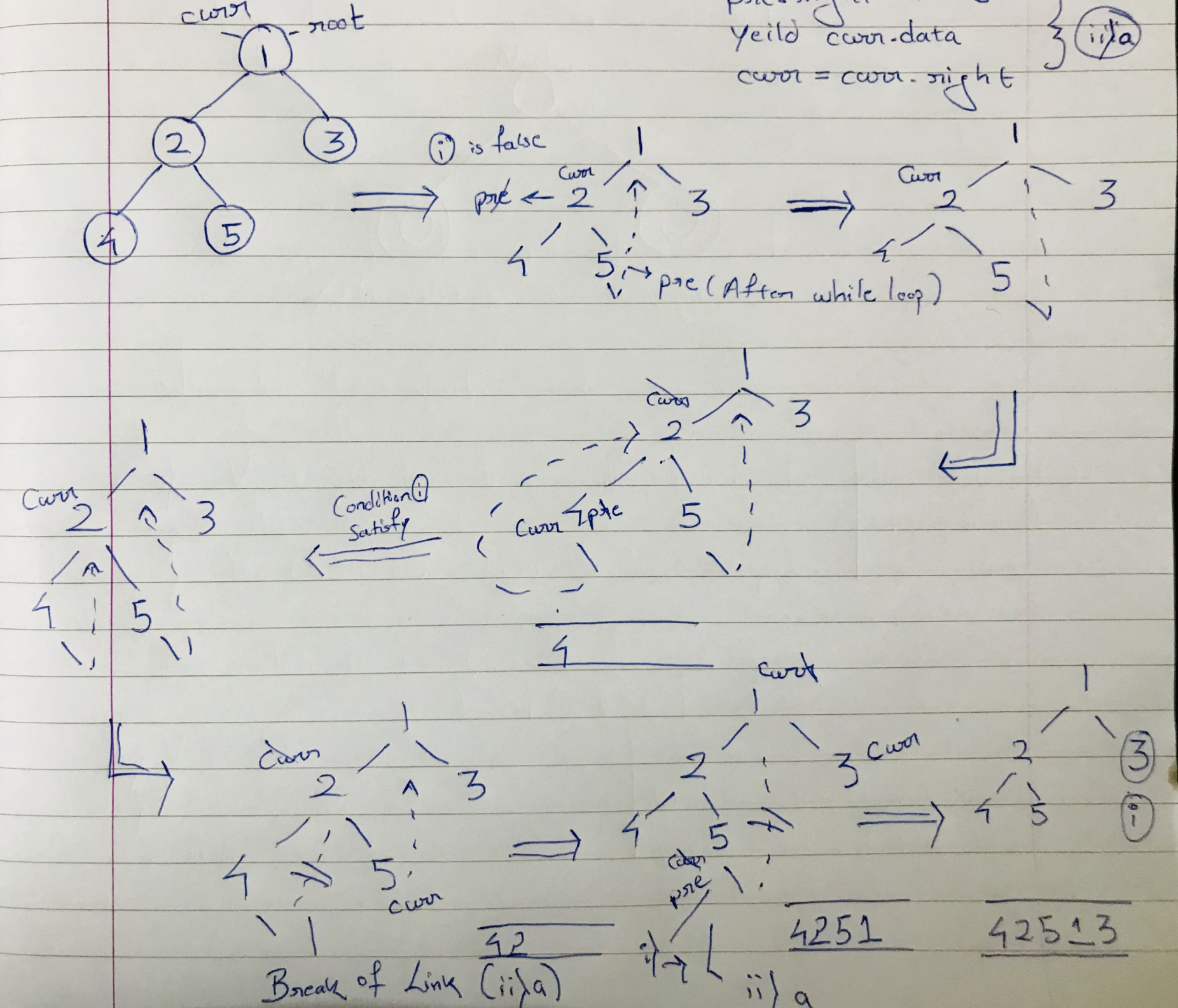
Algo

```
curr = root
while curr is not None:
    if curr.left is None:
        yeild curr.data
        curr = curr.right
    else:
        pre = curr.left
        while pre.right != None and
              pre.right != curr:
            pre = pre.right
        if pre.right is None:
            pre.right = curr
            curr = curr.left
        else:
            pre.right = None
            yeild curr.data
            curr = curr.right
```

(i) (ii) (iii)a



(i) = false


curr
root
1
2
3
4
5
yeild curr.data
curr = curr.right
(ii)a
(i) is false
curr
pre
After while loop
Condition (i) Satisfy
Break of Link (ii)a
4251
42513