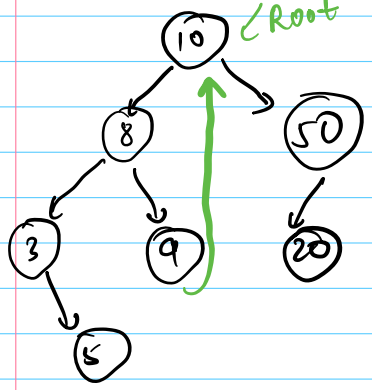
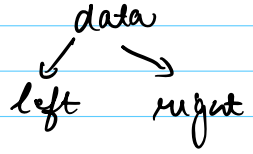
4/8/2023

TREES - 4 - LCA

Morris Inorder traversal



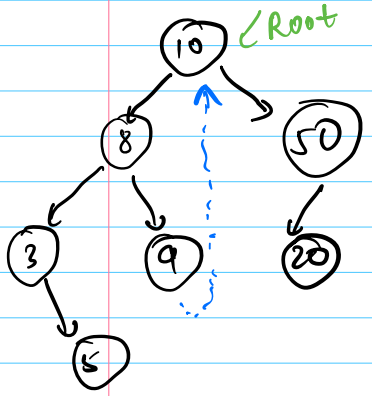
TC: O(N)
SC: O(H) → O(1)



3 5 8 9 10 20 50

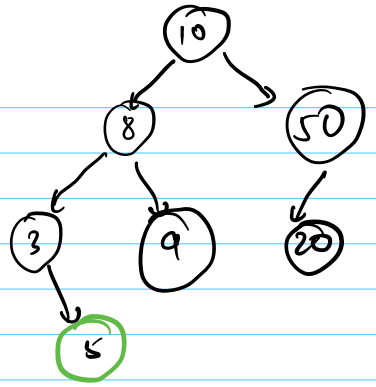
inorder of left subtree → 9 is the rightmost element

[rightmost node of left subtree].right = root




```
Node rightmost(root){
    temp = root.left;

    while(temp.right != NULL
    && temp.right != root){
        temp = temp.right;
    }

    return temp;
}
```

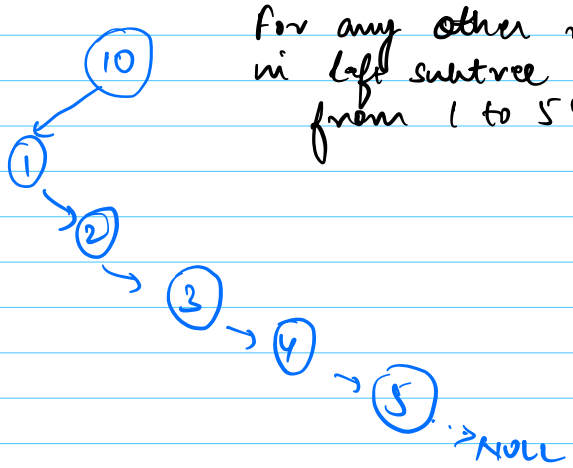
```
curr = root
while(curr != null){
    if(curr.left == null){
        print(curr.data);
        curr = curr.right;
    }
    else{
        R = rightmost(curr);
        if(R.right == null){
            R.right = curr;
            curr = curr.left;
        }
        else{
            print(curr.data);
            curr = curr.right;
            R.right = null;
        }
    }
}
```



3 5 8 9 10 20 50

↓ NULL call

SC: O(1)

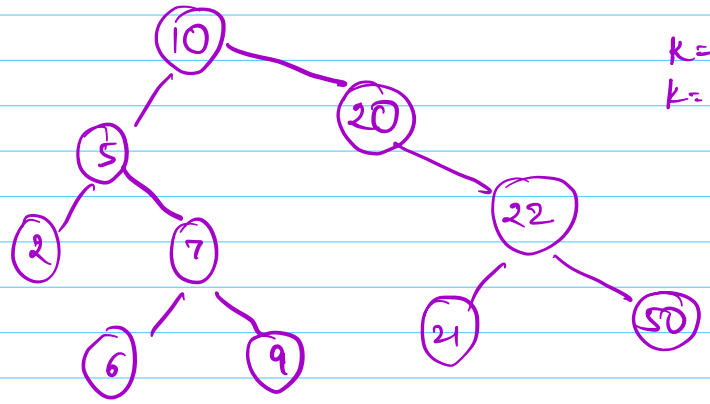


For any other nodes rightmost node in left subtree will we travel from 1 to 5? No

All nodes are atmax visited once while finding rightmost node in left subtree → TC: O(N)

2*f(N) ⇒ O(2N) ≃ O(N)

Q Find K^{th} smallest element in BST.



K=2 ans = 5
K=8 ans = 21

2	5	6	7	9	10	20	21	22	50
1	2	3	4	5	6	7	8	9	10

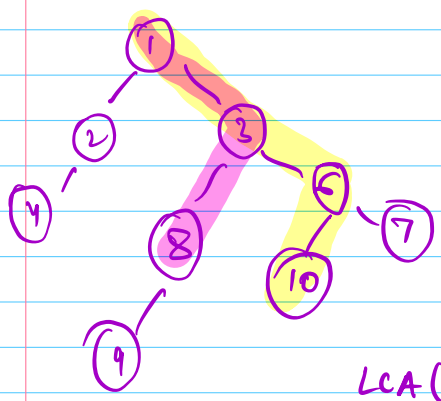
Idea → K^{th} node in inorder traversal of BST.

TC: O(N) , SC: O(H) / O(1)

HW → K^{th} largest element in BST.

Lowest Common Ancestor (LCA)

↳ All nodes in the path from root to current node

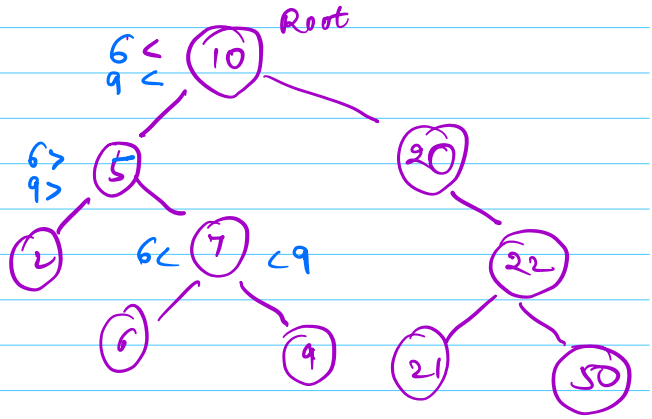


ancestor(10) = | 1 | 3 | 6 | 10 |
ancestor(8) = | 1 | 3 | 8 |

LCA(10, 8) = 3

LCA(9, 7) = 3
LCA(4, 7) = 1

Q Find LCA in BST



LCA(6, 9) = 7
LCA(20, 21) = 20

```
// LCA(x, y)
temp = root
while (temp != null) {
    if (temp.data > x && temp.data > y)
        temp = temp.left;
    else if (temp.data < x && temp.data < y)
        temp = temp.right;
    else
        return temp.data;
}
```

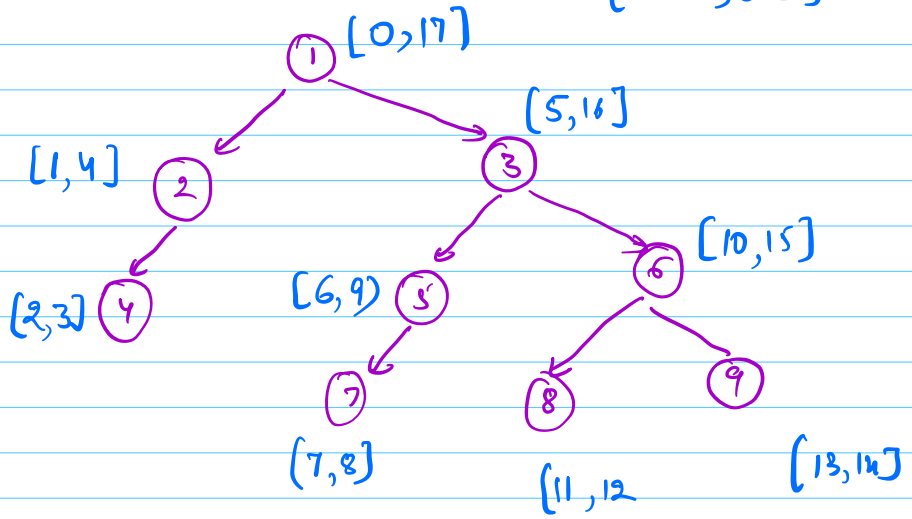
TC: O(H)
SC: O(1)

In time Out time concept

start T = 0

traversal of sub tree starts — traversal of subtree completes

[start, end]



t = 0

Map < Node, Integer> In, Out;

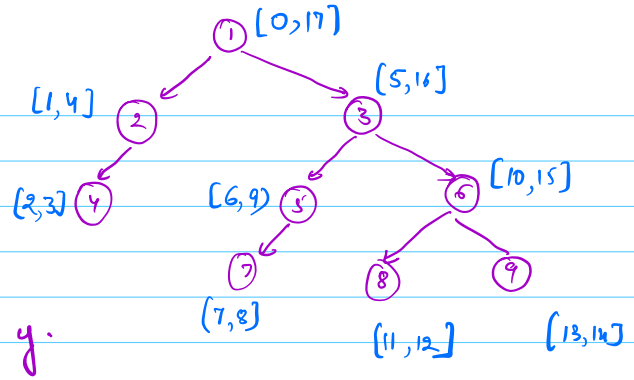
```
void trovel(root) {

    if (root == null) return;

    in(root) = t;                    Node
    t++;

    trovel(root.left);               left
    trovel(root.right);              Right

    Out(root) = t;                   Node
    t++;
}
```

Preorder

Postorder.

TC: O(N)

SC: O(N+H) ≈ O(N)



nodes x, y

in(x) ≤ in(y) →

subtree x starts before y.

out(x) ≥ out(y) ⇒ subtree of x ends after subtree of y.

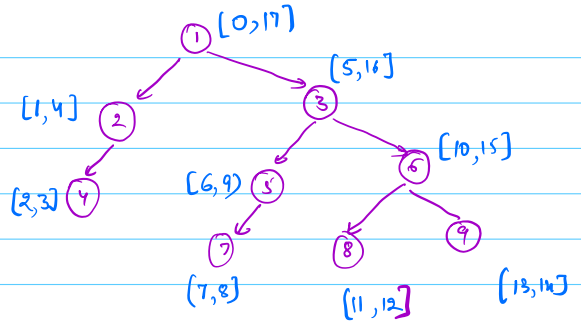
↳ x is an ancestor of y.

Q Find LCA(a,b), BT

```
while(root != null) {
  if (root.left is ancestor of a & b)
      root = root.left
  else if (root.right is ancestor of a & b)
      root = root.right;
  else
      root;
}
```

TC: O(H) → O(N)
SC: O(N)

```
boolean isAncestor(Node parent, Node child) {
  return in[parent] <= in[child] &&
         out[parent] >= out[child];
}
```



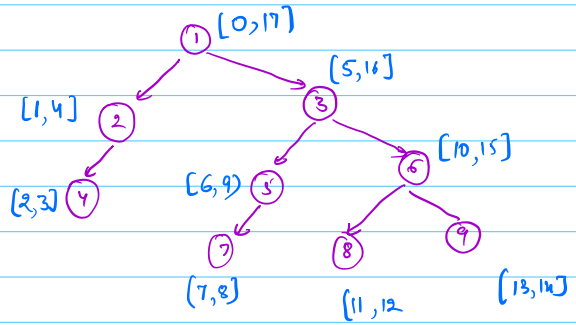
LCA(7,9)

in(2) <= in(7) && out(2) >= out(7) ⇒ 2 is not ancestor of 7
1 <= 7 && 4 >= 8 ✗

in(3) <= in(7) && out(3) >= out(7) ⇒ 3 is ancestor of 7
5 <= 7 && 16 >= 8

in(3) <= in(9) && out(3) >= out(9) ⇒ 3 is ancestor of 9
5 <= 13 && 16 >= 14

is 5 ancestor of 7 & 9 ✗
is 6 ancestor of 7 & 9 ✗
} return 3.



1