

Nov23_PSP_3Apr

sudhakar venkatachalam

Vijay V A

Manjunatha I

Mayur Hadawale

Harshil Dabhoya

Sai Sharath

Yash Malviya

Rajeev

Kevin Theodore E

Shaurya Srivastava

Suraj Devraye

kameswarreddy Yeddula

Vigneshwaran K

ALLEN GEOSHAN M

manikandan m

Robin Dhiman

Pranadarth S

MD JASHIMUDDIN

Sarat Patel

Nitendra Rajput

SIJU SAMSON

Chandu

Pradeep Kumar Chandra

Mohammed Arshad

Pushkar Deshpande

Batch PSP

55.1 → 58%

Upcoming Contest

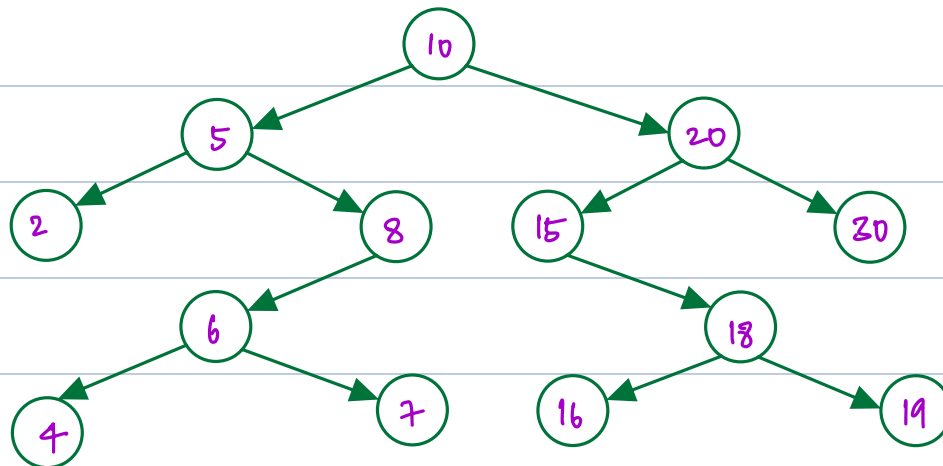
Two Pointer

Linked List

Stacks & Queues

pass → Dhruv will personally
call

Quick Recap

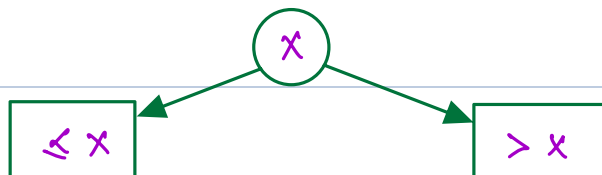


Binary Tree

- ① Hierarchical data structure
- ② Atmost two children (left and a right)

Binary search tree

\forall nodes x .



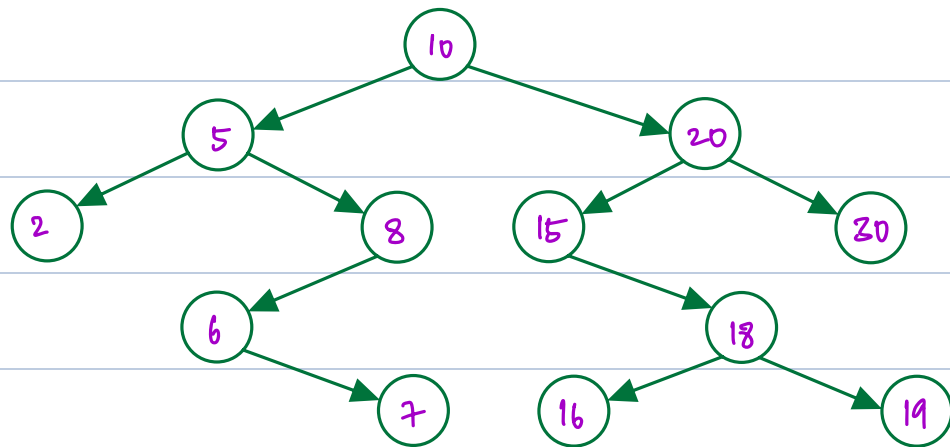
LST has all

data $\leq x$

RST has data

$> x$

Q1 → Given a Binary Search Tree and a positive Integer k , find k th smallest element in BST



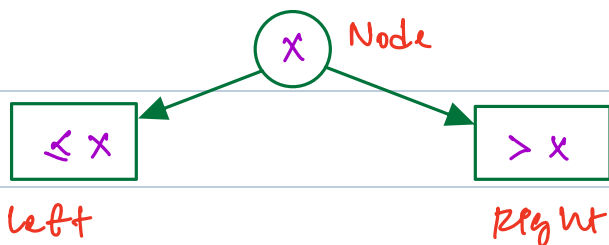
example

$K = 4 \rightarrow 7$

$K = 6 \rightarrow 10$

Observation

\forall nodes x ,



Inorder traversal is sorted

$left \leq Node < right$

Solution

Do Inorder traversal and store k th smallest element

pseudo code

int counter = 0; int ans = -1;

void Pnorder (root, k) {

 if (root == null) return;

 Pnorder (root.left, k)

 counter++;

 if (counter == k) ans = root.data;

 Pnorder (root.right, k);

}

T.C = $O(N)$

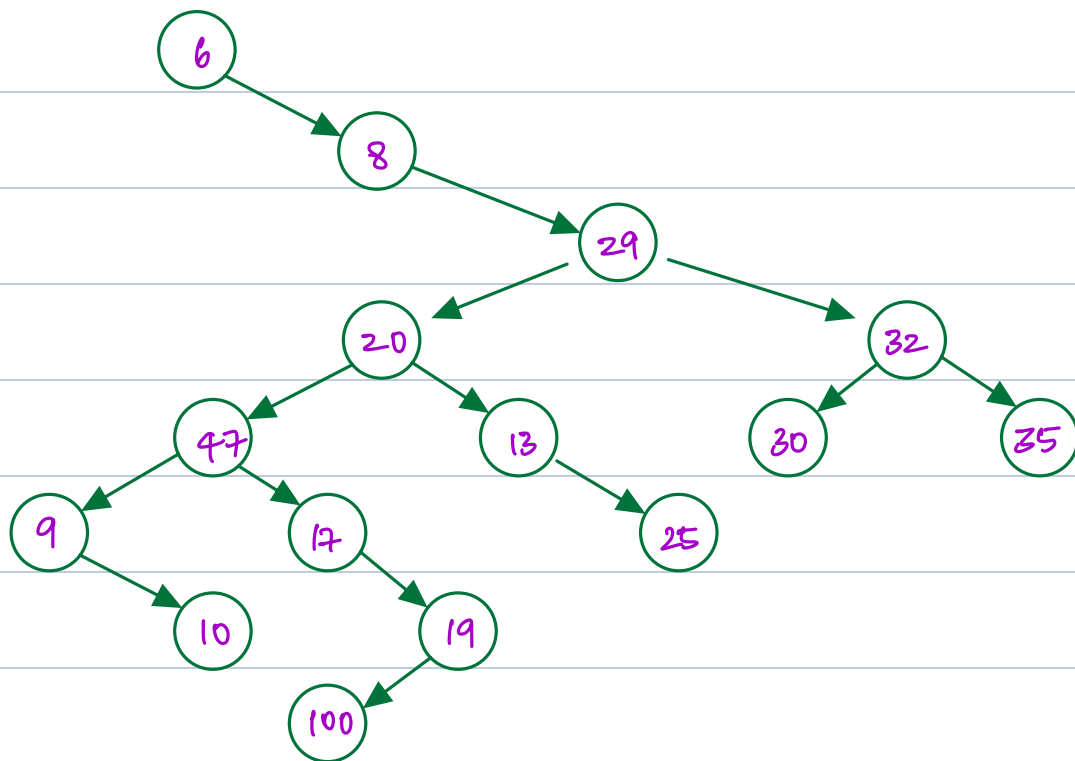
S.C = $O(1)$

Q2 → Morris Inorder Traversal

Without using any space, output the Pnorder traversal

Recursion → we cannot use, stack space

Iteration → we cannot use stacks.



Inorder traversal

6	8	9	10	47	17	100	19	20	13	25	29	30	32	35
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Observation

- ① If $\text{node}.\text{left} == \text{null}$, print data and go right
- ② If $\text{node}.\text{left} \neq \text{null}$

Find the right most node on LST of node and point its right reference to node

while finding right most node If we come across curr node then remove the reference

a) Print curr. data

b) traverse to curr. right.

pseudo code

Node Morris Inorder (root) {

 If (root == null) return Null;

 Node curr = root;

 while (curr != null) {

 if (curr.left == null) {

 print (curr.data);

 curr = curr.right;

 }

 else { // Find right most node in LST

 temp = curr.left;

 while (temp.right != null &&

 temp.right != curr)

 temp = temp.right;

 // create link

 if (temp.right == null) {

 temp.right = curr

 curr = curr.left

 }

// delete the link

else {

temp.right = null;

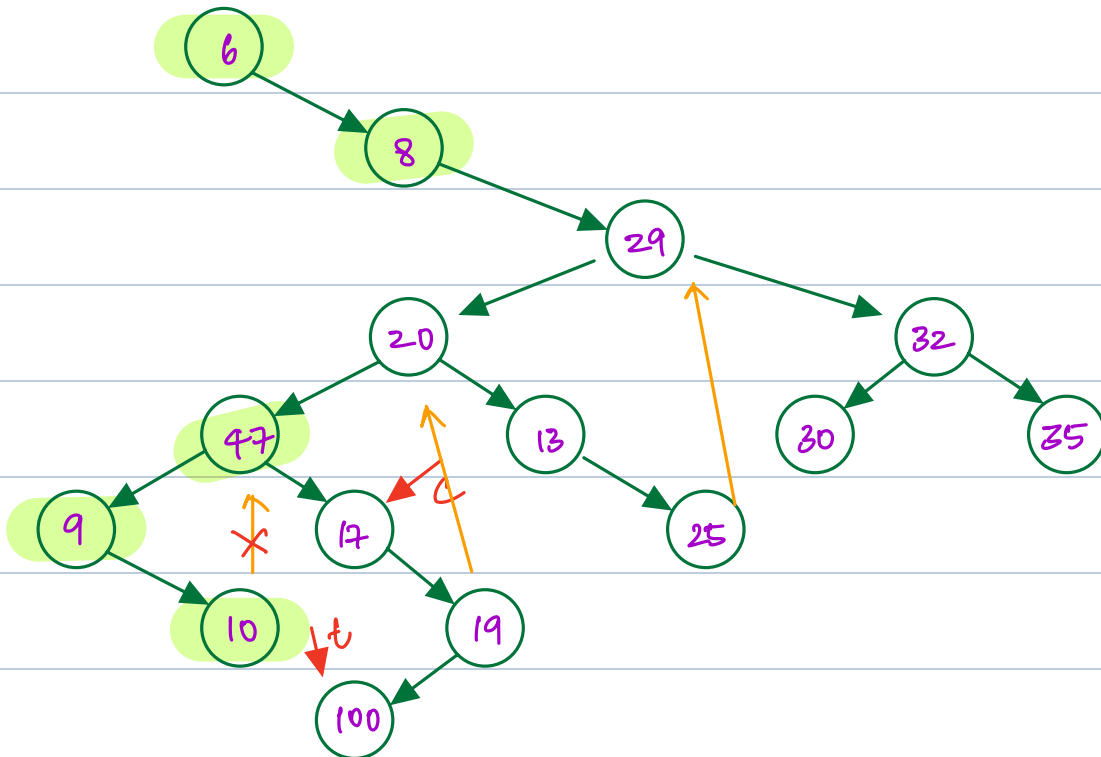
print (curr.data);

curr = curr.right;

}

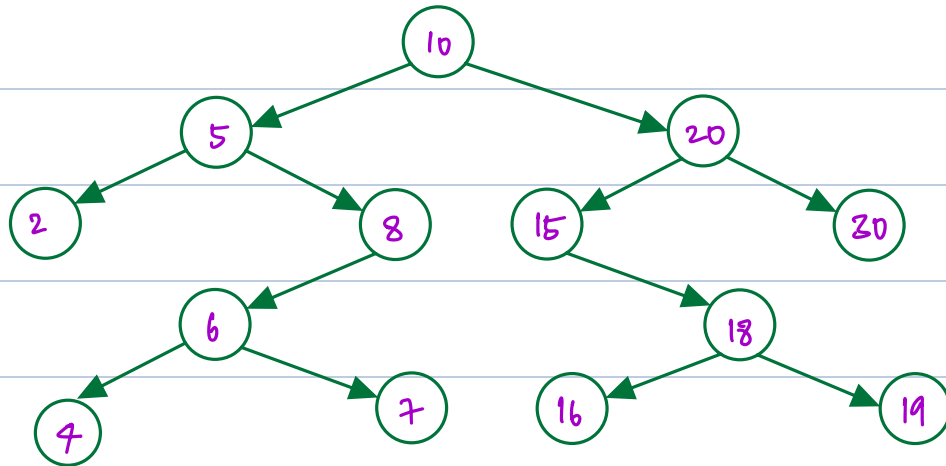
T.C = $O(n)$

S.C = $O(1)$



Break 10:20 pm → 10:26 pm

Q → Given a value, find path from root to that node (Binary Tree)



example

$K = 7$

$10 \rightarrow 5 \rightarrow 8 \rightarrow 6 \rightarrow 7$

$K = 18$

$10 \rightarrow 20 \rightarrow 15 \rightarrow 18$

Observation

what traversal can we use?

level order traversal ✗

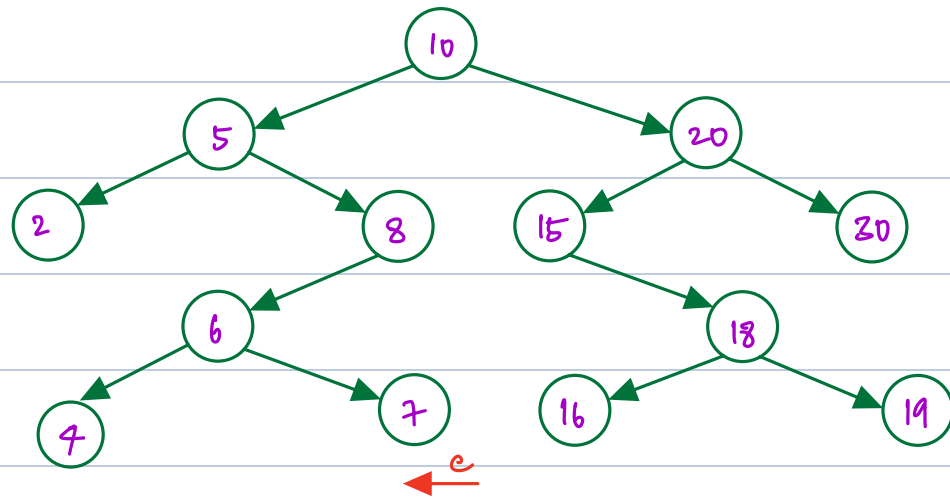
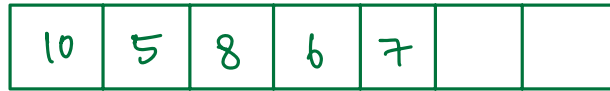
preorder → inserting data

inorder

postorder → removing data

Dry Run

Path →



$k = 7$

pseudo code

path = []

boolean travel (root, k) &

if (root == null) return false

path.append (root)

if (root.data == k) return true

boolean f-left = travel (root.left, k)

if (f-left) return true;

boolean f-right = travel (root.right, k)

if (f-right) return true

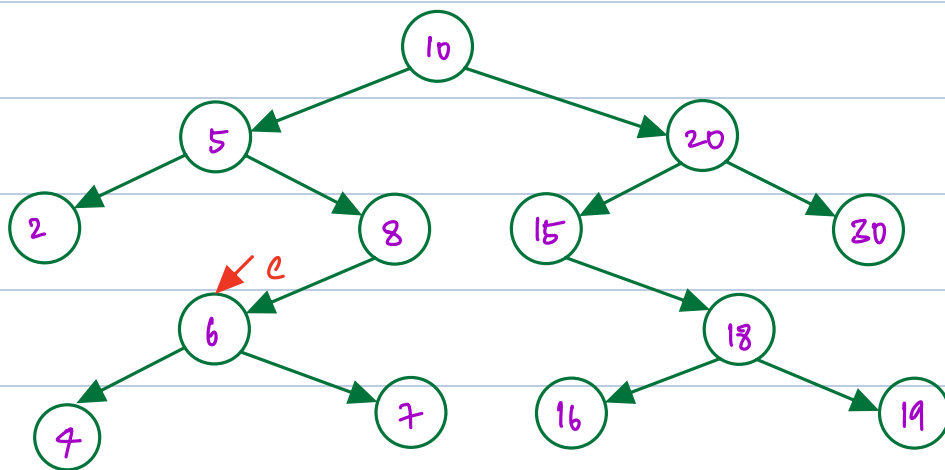
path.removeLast();

return false;

0

T.C = $O(n)$

S.C = $O(H)$

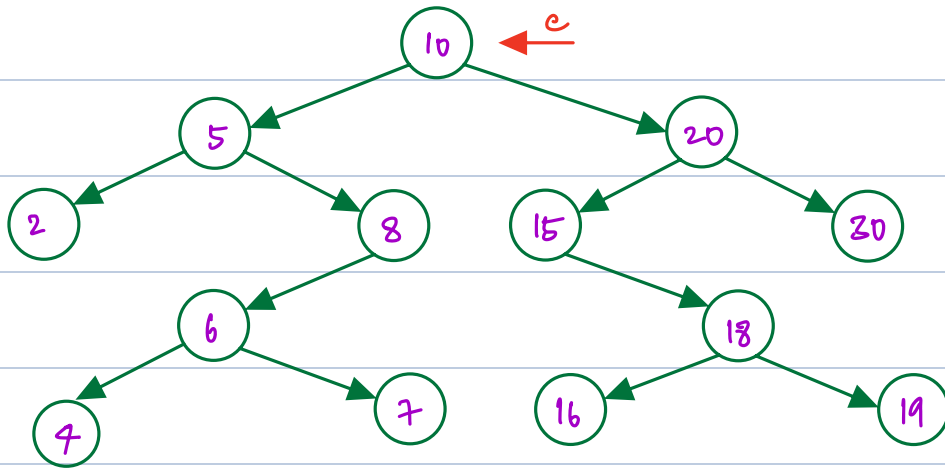


Q → Find the lowest Common Ancestor of two nodes

Ancestors

all nodes from root node to current node

is ancestor of current node



ancestors of 4 : [10, 5, 8, 6, 4]

ancestors of 2 : [10, 5, 2]

pseudo code

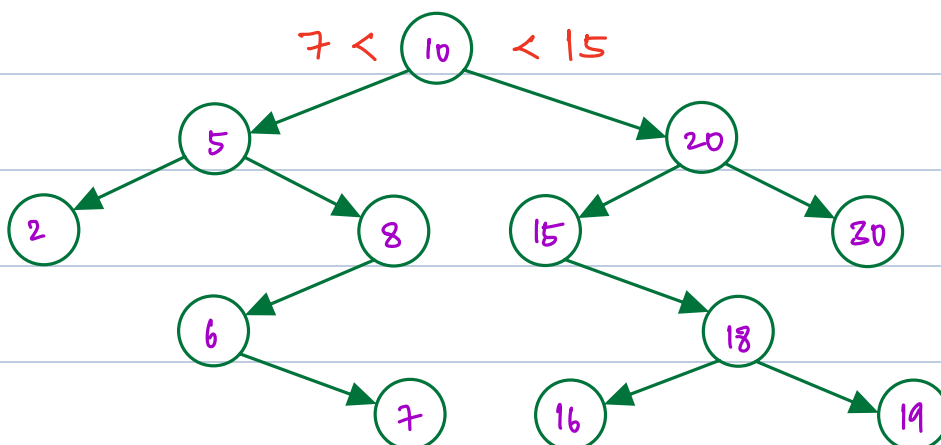
curr = root;

path - a = travel (curr, node-a.data);

curr = root;

path - b = travel (curr, node-b.data);

Q → Find LCA of Binary Search Trees



$$LCA(7, 15) = 10$$

$$LCA(2, 7) = 5$$

$$LCA(16, 30) = 20$$

pseudo code

curr = root; LCA = -1;

while (curr != null) {

 if (curr.data > x && curr.data > y)

 curr = curr.left;

 else if (curr.data < x && curr.data < y)

 curr = curr.right;

 else

 return curr.data

}

$$T.C = O(H)$$

$$S.C = O(1)$$

Additional Problem

In Time / Out Time

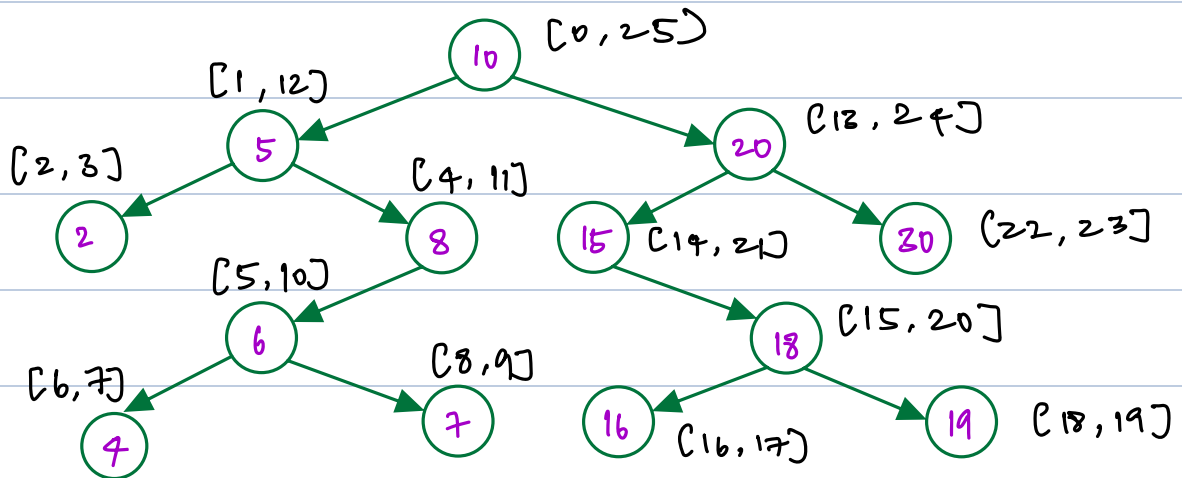


start subtree



end subtree

Binary tree



pseudo code

t = 0;

void travel (root) {

if (root == null) return

in [root] = t;

t += 1;

travel (root.left);

travel (root.right);

out [root] = t;

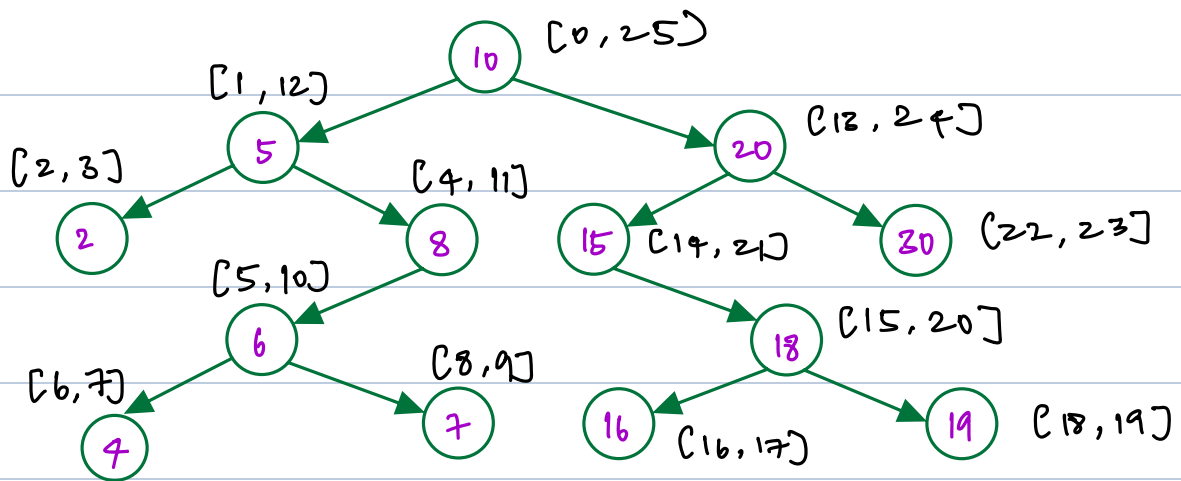
t += 1;

}

in / out

HashMap

< node : int >



LCA (7, 2)

example (5) as ancestor

$in(5) < in(2) \& in(7)$

$1 < 2 \& 7$

$out(5) > out(2) \& out(7)$

$12 > 3 \& 9$

LCA (a, b)

curr = root;

while (curr != null) {

 if (curr.left is ancestor of a & b) {

 curr = curr.left;

 }

 else if (curr.right is ancestor of a & b) {

 curr = curr.right;

 }

else return curr;