Round 1 type->

Question 1-> LCA of Binary tree .Forgot to how to implement to return lca pointer

Solution 1->

bool printAncestors(struct node \*root, int target)

{

/\* base cases \*/

if (root == NULL)

return false;

if (root->data == target)

return true;

/\* If target is present in either left or right subtree of this node,

then print this node \*/

if ( printAncestors(root->left, target) ||

printAncestors(root->right, target) )

{

cout << root->data << " ";

return true;

}

/\* Else return false \*/

return false;

}

Solution 2->

void printAncestors(struct Node \*root, int key)

{

if (root == NULL) return;

// Create a stack to hold ancestors

struct Stack\* stack = createStack(MAX\_SIZE);

// Traverse the complete tree in postorder way till we find the key

while (1)

{

// Traverse the left side. While traversing, push the nodes into

// the stack so that their right subtrees can be traversed later

while (root && root->data != key)

{

push(stack, root); // push current node

root = root->left; // move to next node

}

// If the node whose ancestors are to be printed is found,

// then break the while loop.

if (root && root->data == key)

break;

// Check if right sub-tree exists for the node at top

// If not then pop that node because we don't need this

// node any more.

if (peek(stack)->right == NULL)

{

root = pop(stack);

// If the popped node is right child of top, then remove the top

// as well. Left child of the top must have processed before.

// Consider the following tree for example and key = 3. If we

// remove the following loop, the program will go in an

// infinite loop after reaching 5.

// 1

// / \

// 2 3

// \

// 4

// \

// 5

while (!isEmpty(stack) && peek(stack)->right == root)

root = pop(stack);

}

// if stack is not empty then simply set the root as right child

// of top and start traversing right sub-tree.

root = isEmpty(stack)? NULL: peek(stack)->right;

}

// If stack is not empty, print contents of stack

// Here assumption is that the key is there in tree

while (!isEmpty(stack))

printf("%d ", pop(stack)->data);

}