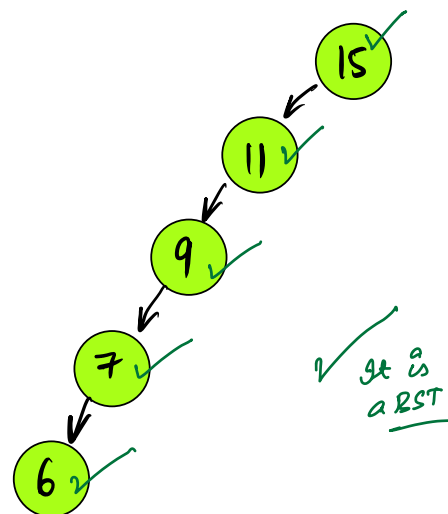
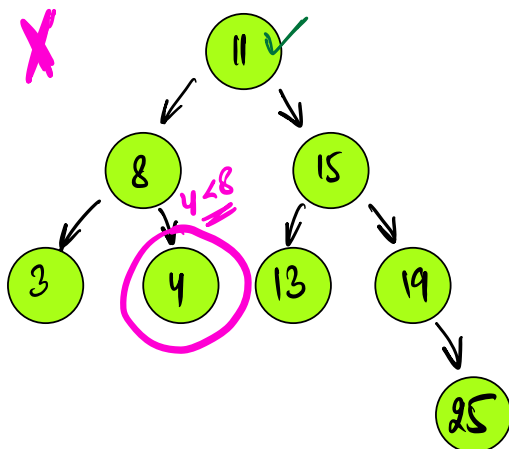
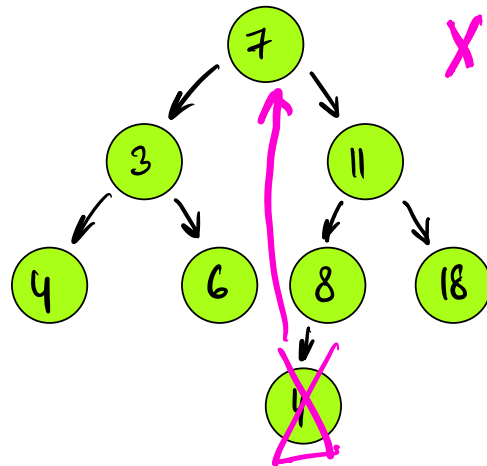
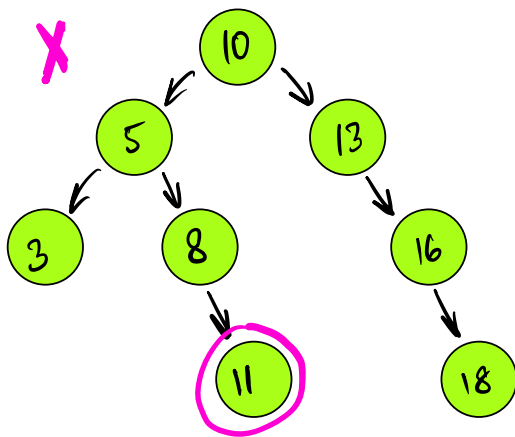
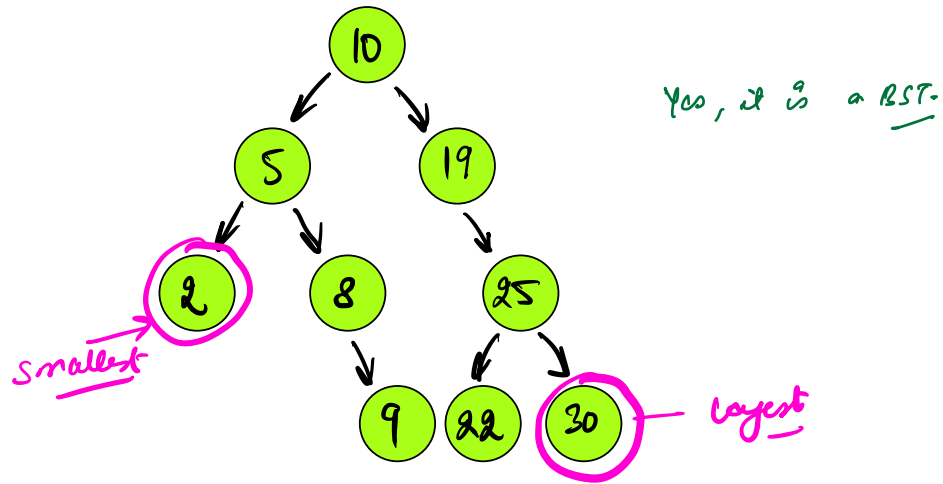


Binary Search Trees \rightarrow BT's

\nexists nodes \rightarrow $\begin{matrix} \text{all elements} \\ \text{in LST} \end{matrix} < \text{node} < \begin{matrix} \text{all elements} \\ \text{in RST} \end{matrix}$

equal \rightarrow either left or right

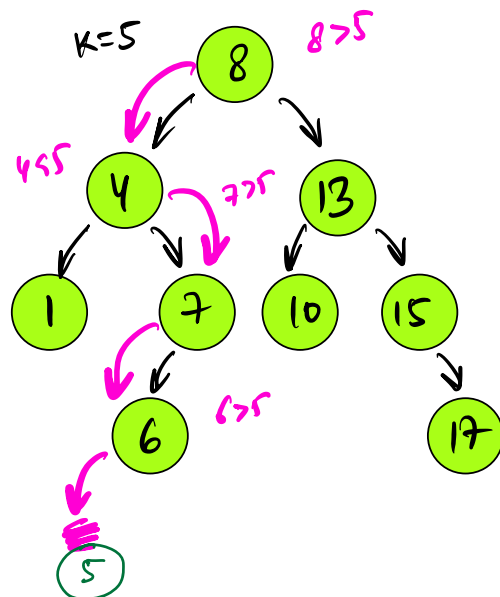
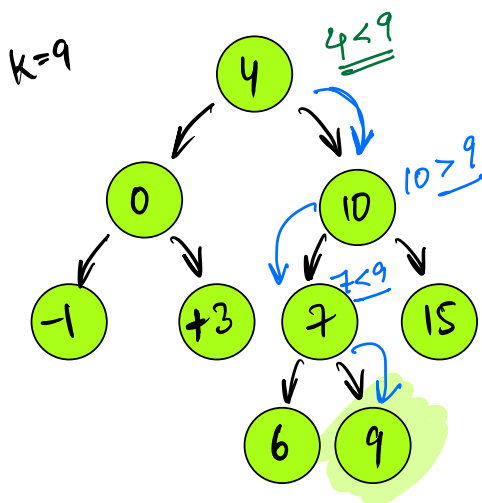




Inorder Traversal of BST is Sorted

$Left < Root < Right$

Search K in BST



```

bool search (Node root, int k)
{
    Node temp = root;

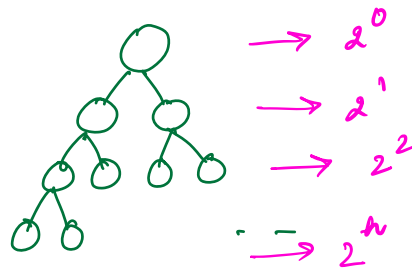
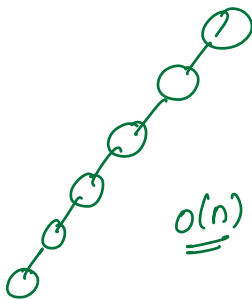
    while (temp != null)
    {
        if (temp->data == k) return true;
        else if (temp->data < k) temp = temp->right;
        else temp = temp->left;
    }

    return false;
}

```

T.C

$O(\text{height})$



②

$$H \approx \underline{\underline{\log N}}$$

$$N = 2^0 + 2^1 + 2^2 + \dots + 2^h$$

$$N = \frac{1(2^{h+1} - 1)}{2 - 1}$$

$$N = 2^{h+1} - 1$$

$$N + 1 = 2^{h+1}$$

$$\boxed{\log_2 N \approx h}$$

insertion

Node temp = root; prev = NULL;

```
while (temp != null) → prev = temp;
{
    if (temp.data == k) return true;
    else if (temp.data < k) temp = temp.right;
    else temp = temp.left;
}

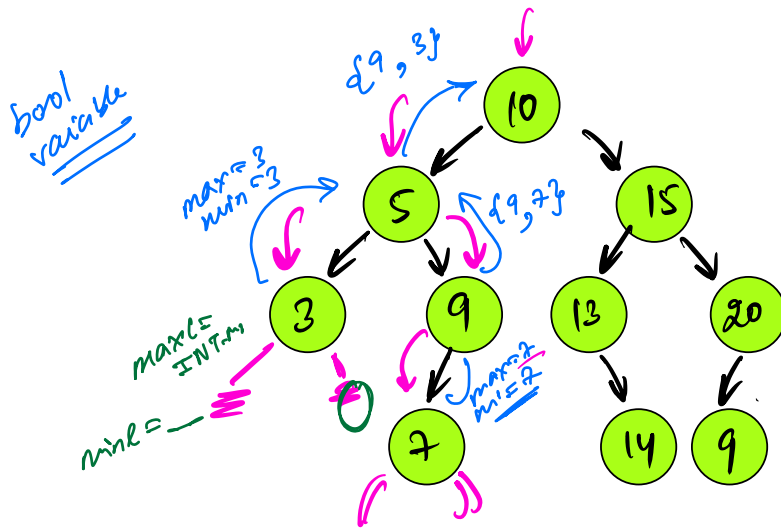
if (prev == NULL) { new node will be your tree }
    if (k < prev.data)
        prev.left = new Node(k);
    else
        prev.right = new Node(k);
```

- check if a given tree is BST?

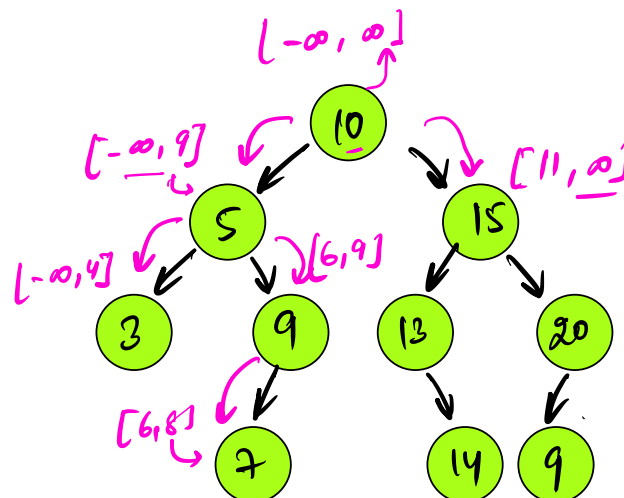
A1: # in-order traversal - sorted

A2:

$$\max(LST) < \text{node} < \min(RST)$$



A3:



preorder

•

bool isBST(root, l, r)

if (root == null) return true;

if (l <= root->data && root->data <= r)

bool x = isBST(root->left, l, root->data-1);

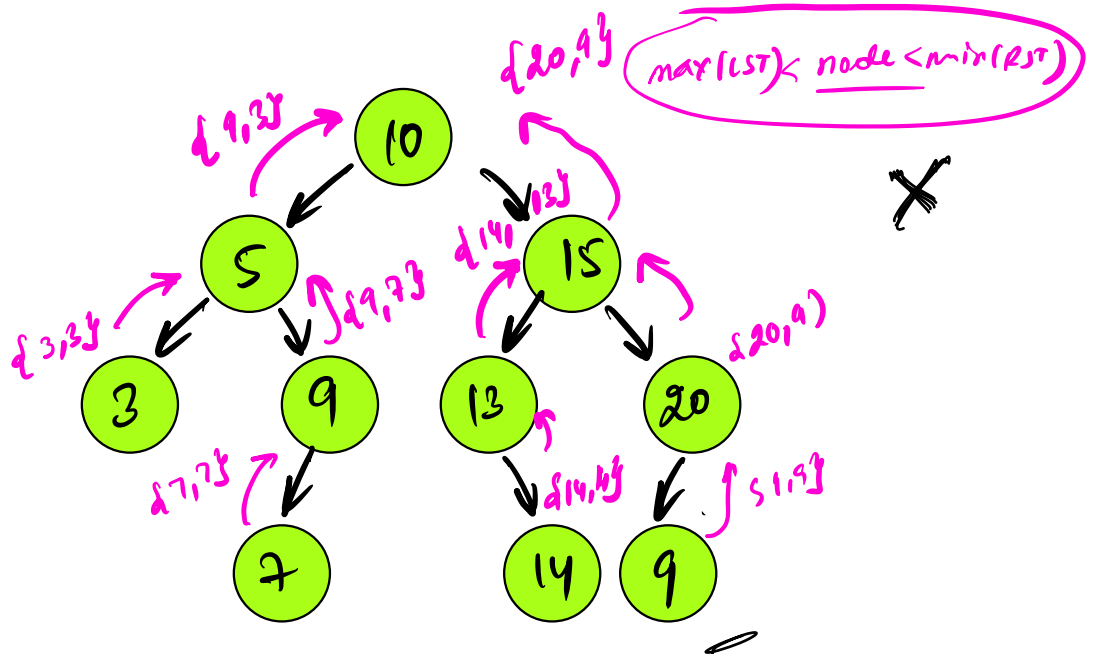
bool y = isBST(root->right, root->data+1, r);

return x && y;

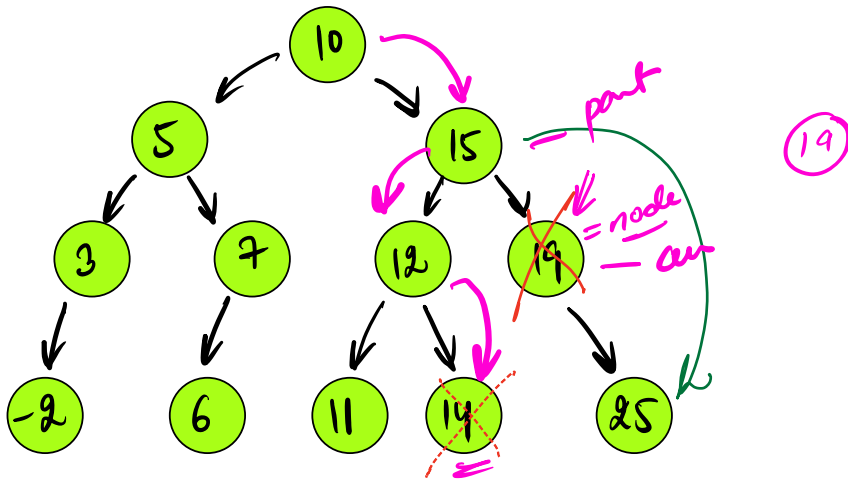
return false;

}

A2%



Deletion



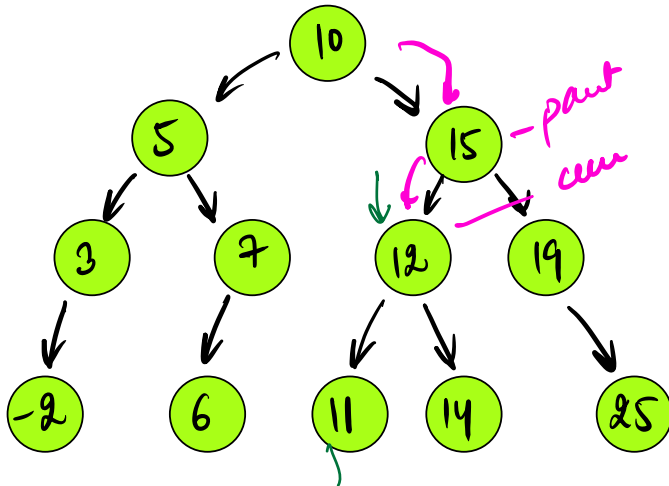
Case I:- Node is leaf

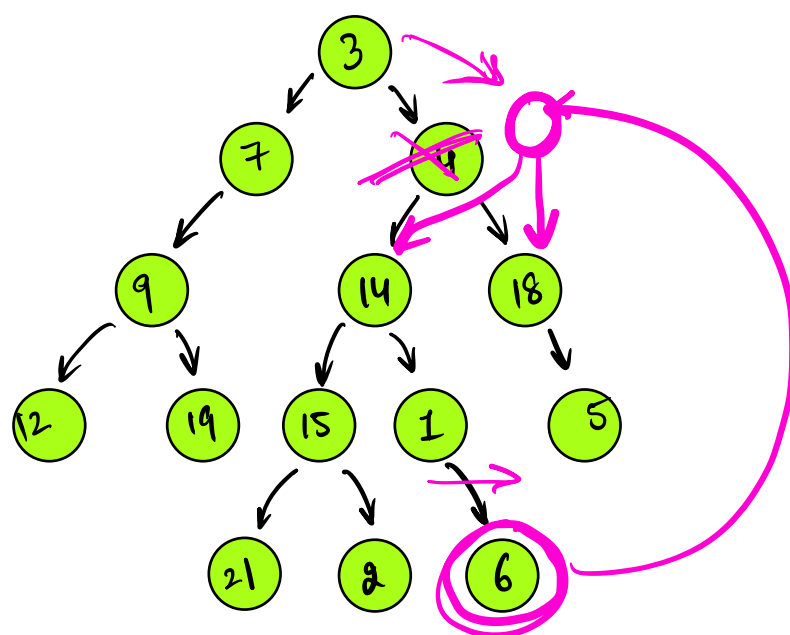
parent-left/right = null

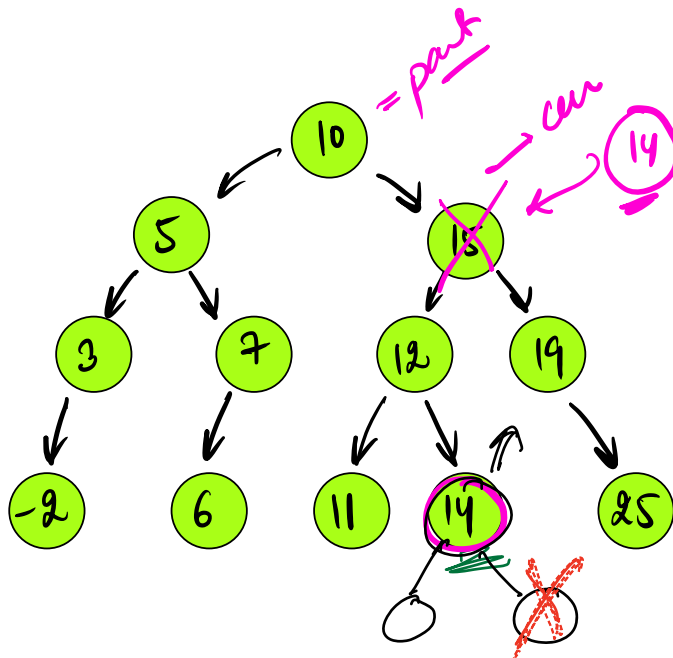
cor II: node with \pm child

parent.left/right = cur.left/right

Case III: Node with 2 children

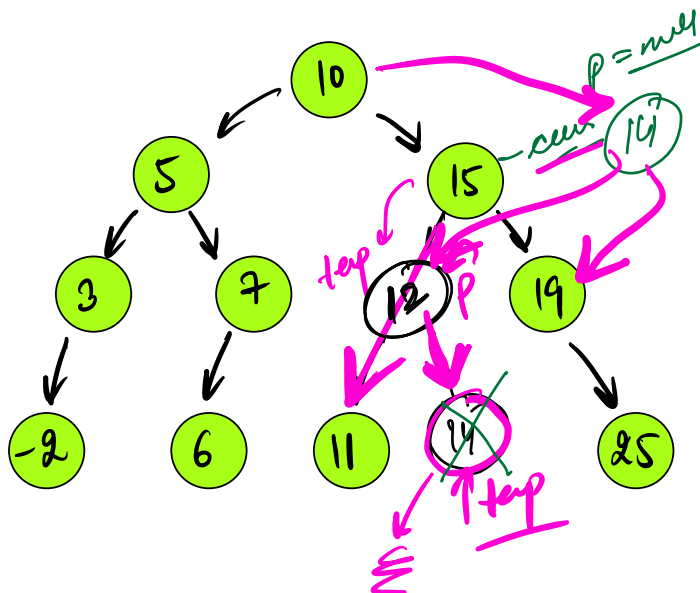




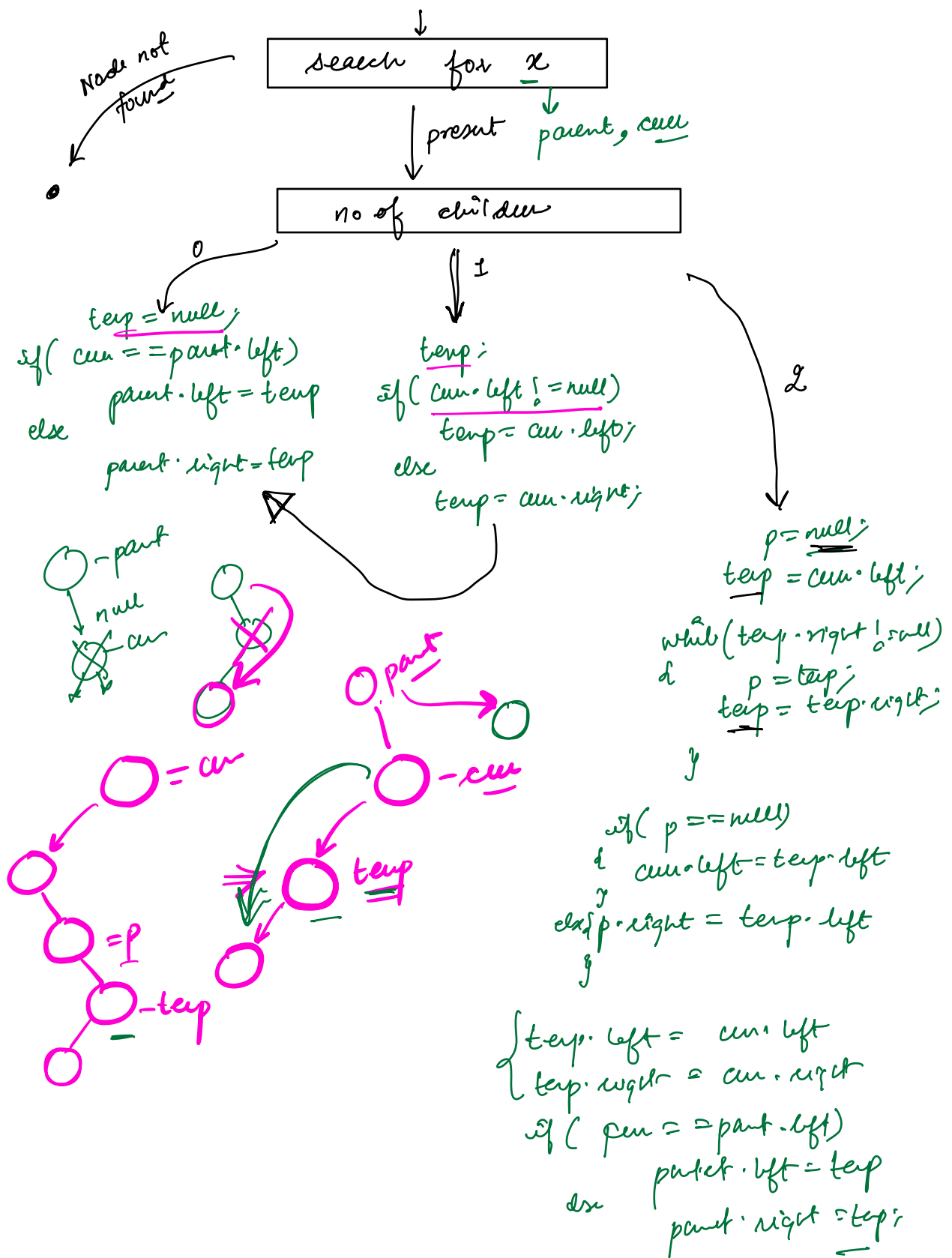


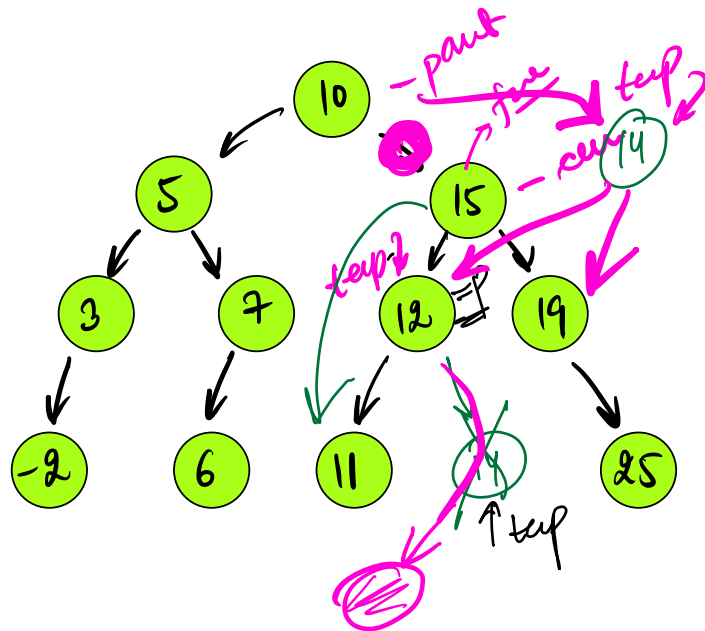
\uparrow
max(LST)
min(RST)

there is not
going to be
anything on right
of max(LST)



max(LST)





$cur \cdot left = temp \cdot left$
 $p \cdot right = temp \cdot left$