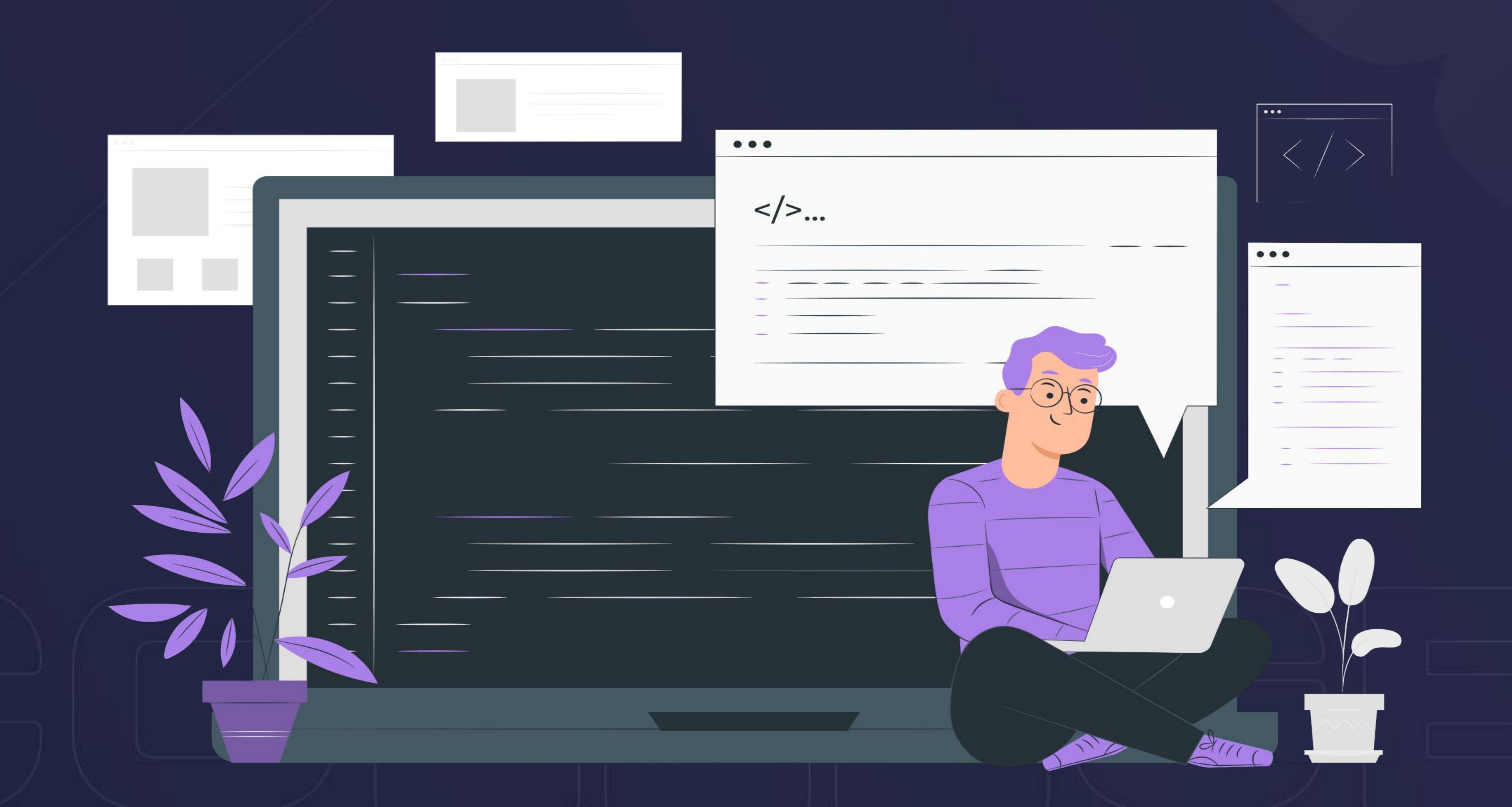


# Lecture 55 Binary Search Trees





#### Recap

- Trees
- Interview Problems on Trees



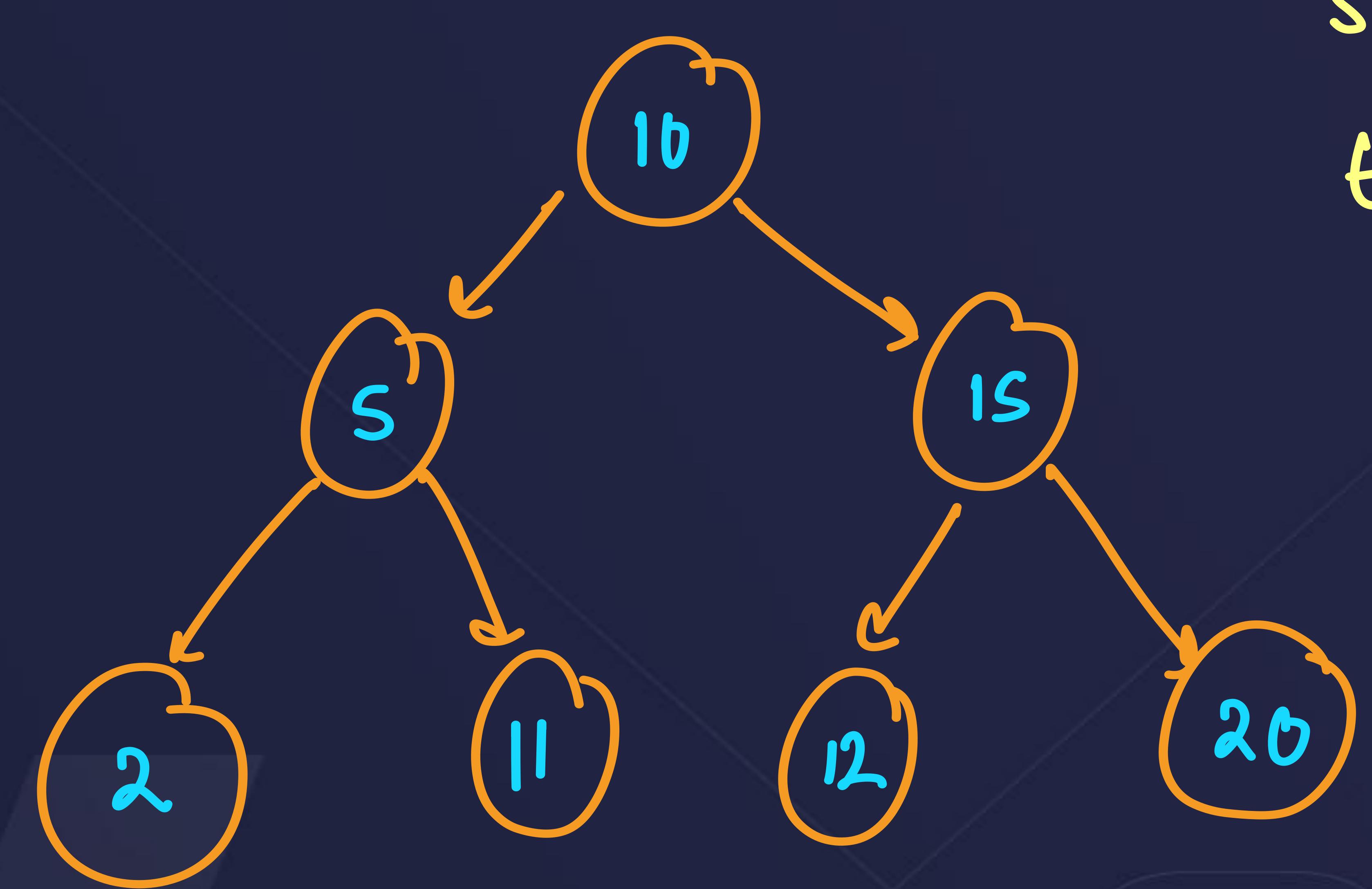
#### Today's Checklist

- Why Binary Search Tree?
- What is Binary Search Tree?
- Advantages
- Disadvantages
- Applications
- Insertion
- Traversal Inorder, Preorder, Postorder
- Searching
- Practice problems on BST



What and Why? -> Every node to the left of anode Should be smaller & Every node to the right should be greater. applies for all nodel

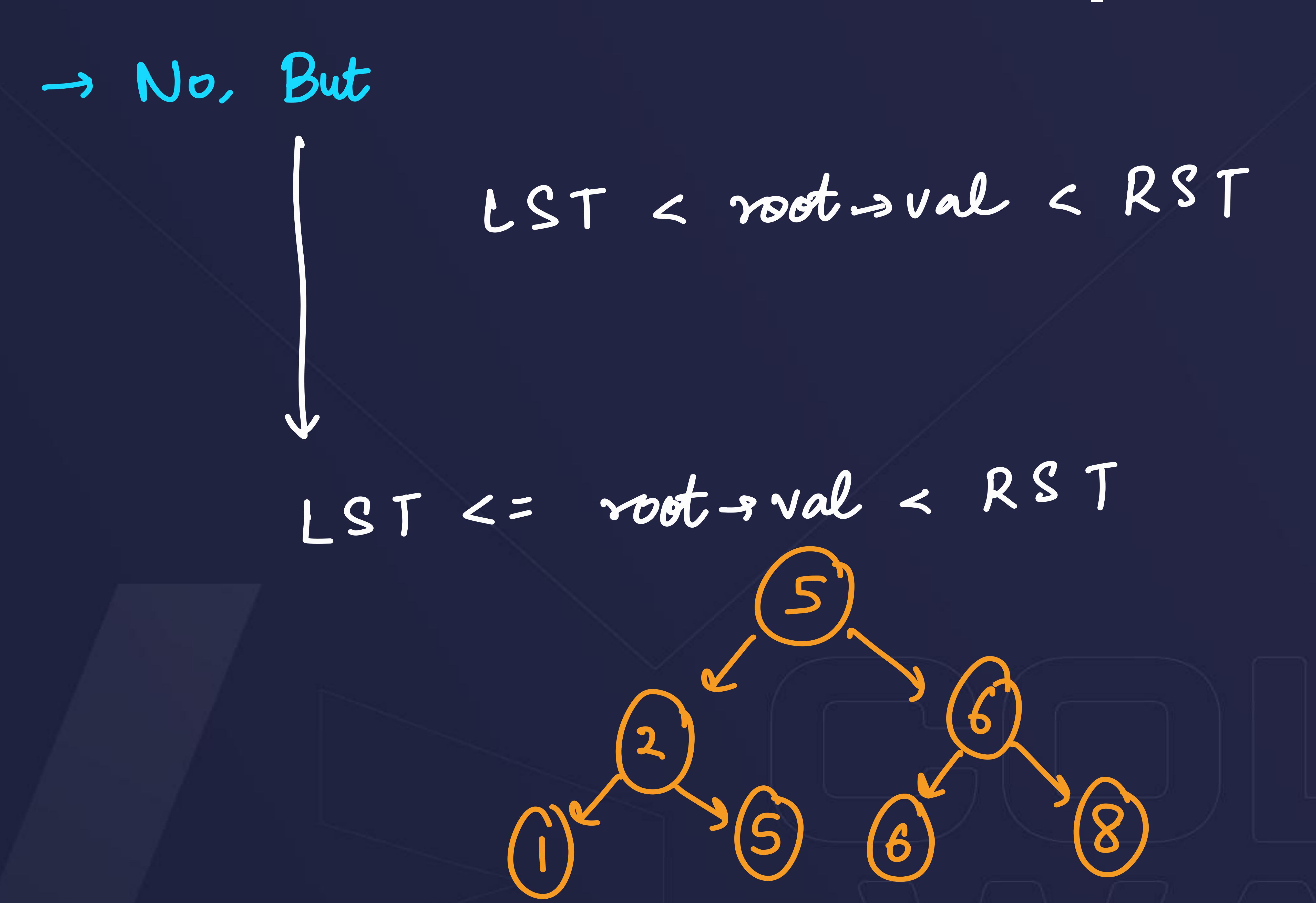
the entire LST



in valid BST



#### Ques: Can a BST contain duplicate elements?





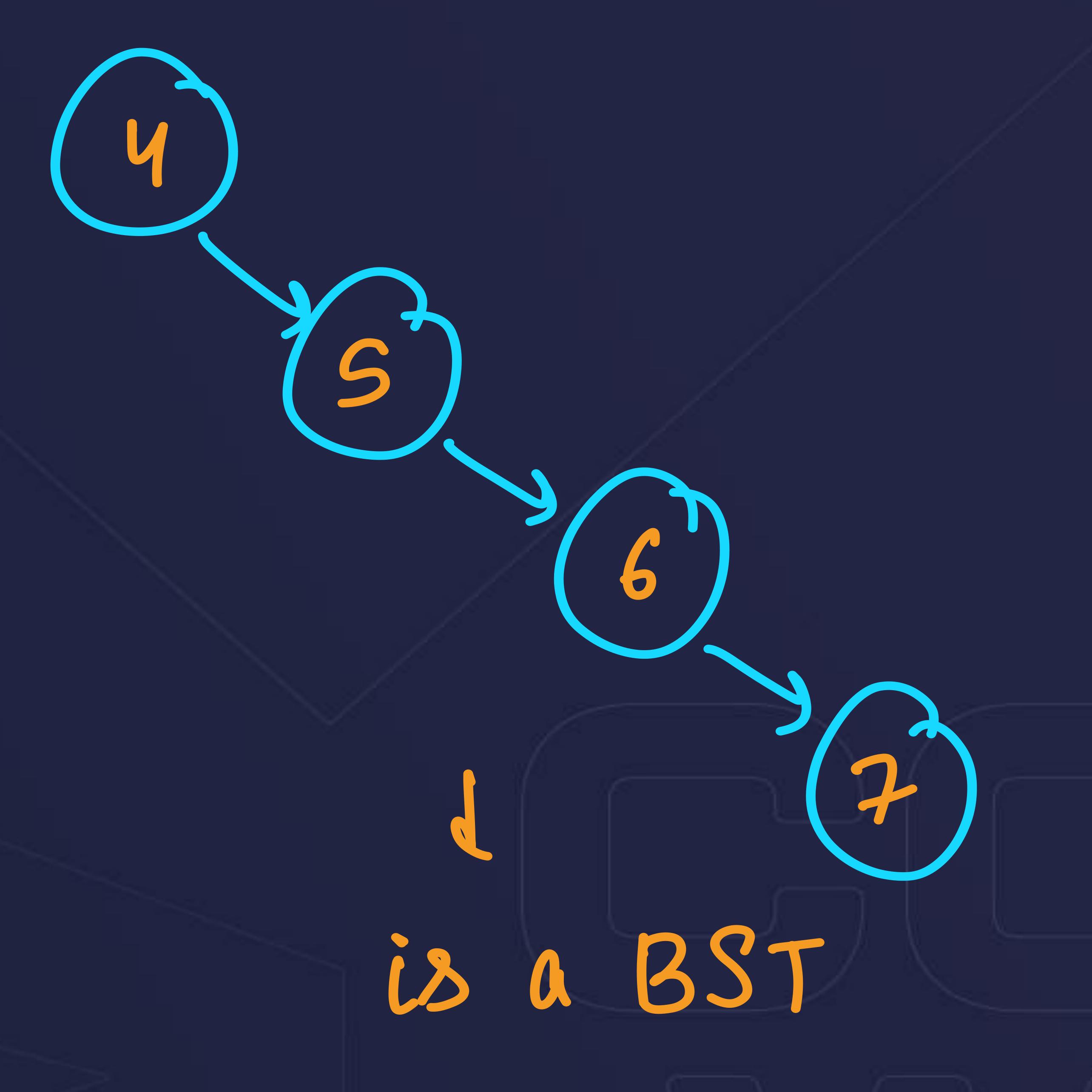
#### Advantages

- Efficient Searching
- Efficient insertion and deletion
- Usage in implementation of other data structures like sets, maps, priority queues etc.



#### Disadvantages

- Lack of support for range queries
- Not that efficient in case of Unbalanced Trees





#### Applications

- Phonebook
- Dictionary
- Stock market analysis



## Concept Builder:

Q, You have 3 nodes with diff-values, for 2x > 1,2,3.
How many unique Binary Trees can be formed?

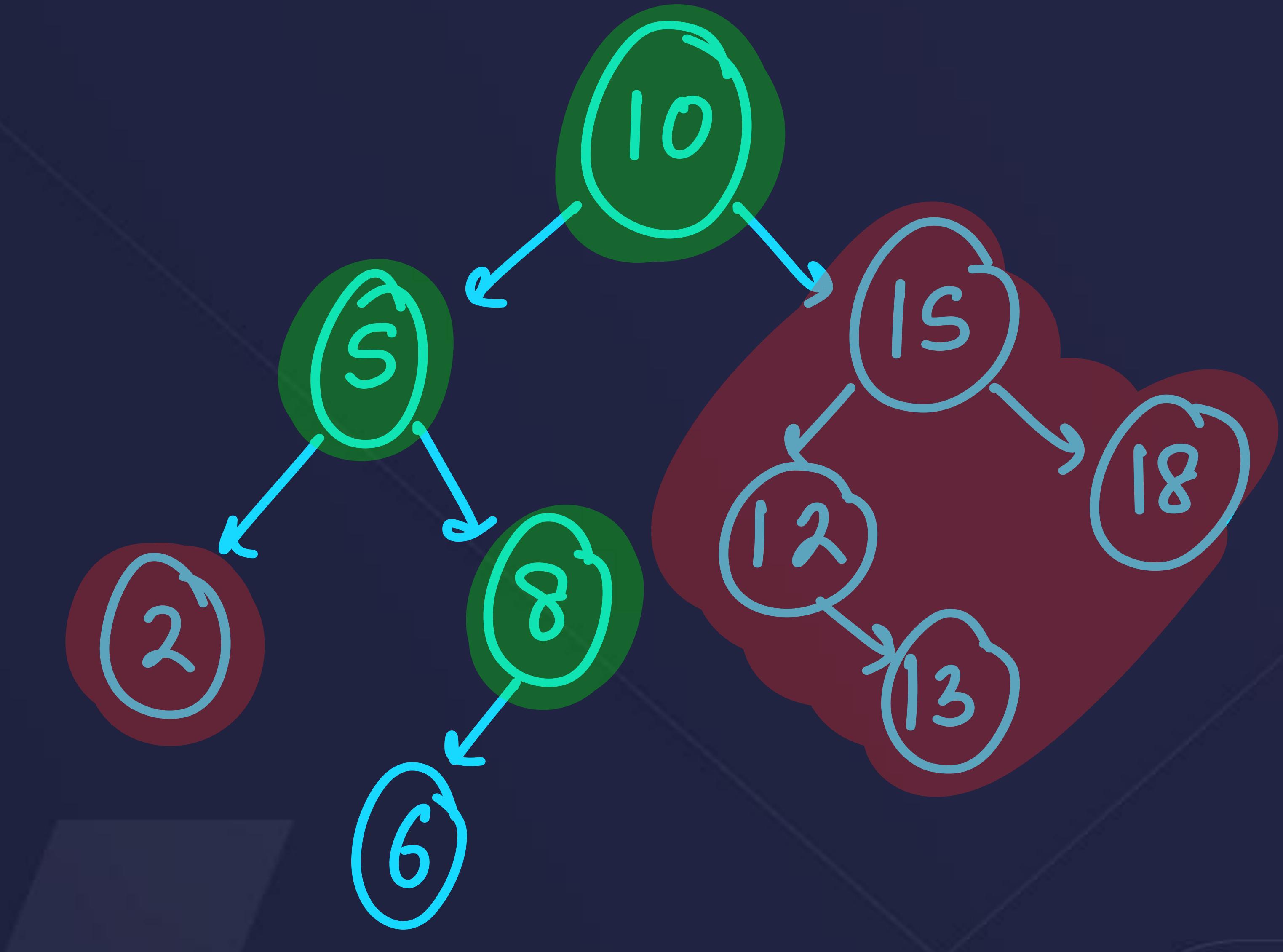








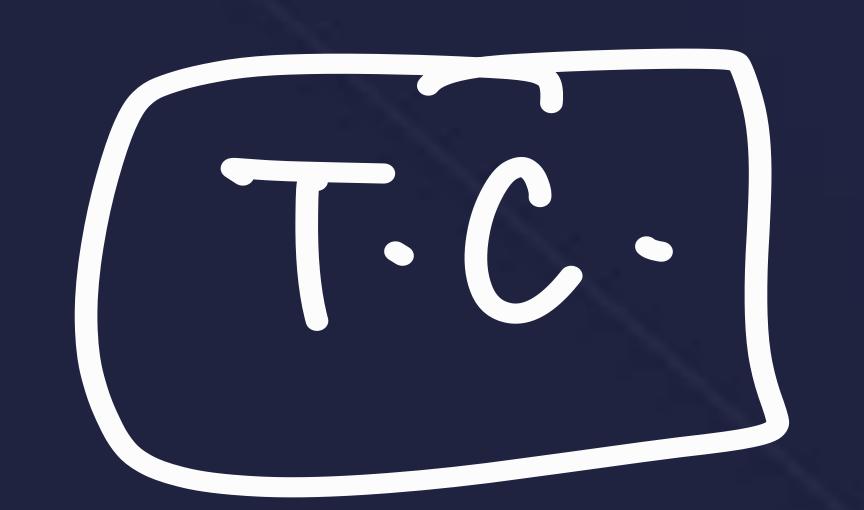
#### Search in a BST



#### [LeetCode 700]



#### Search in a BST



- Best (ase = O(logn)/O(h)

  (balanced tree)
- worst case = 0(n)/0(n)
- · Avg. Case = O(h)

#### LeetCode 700

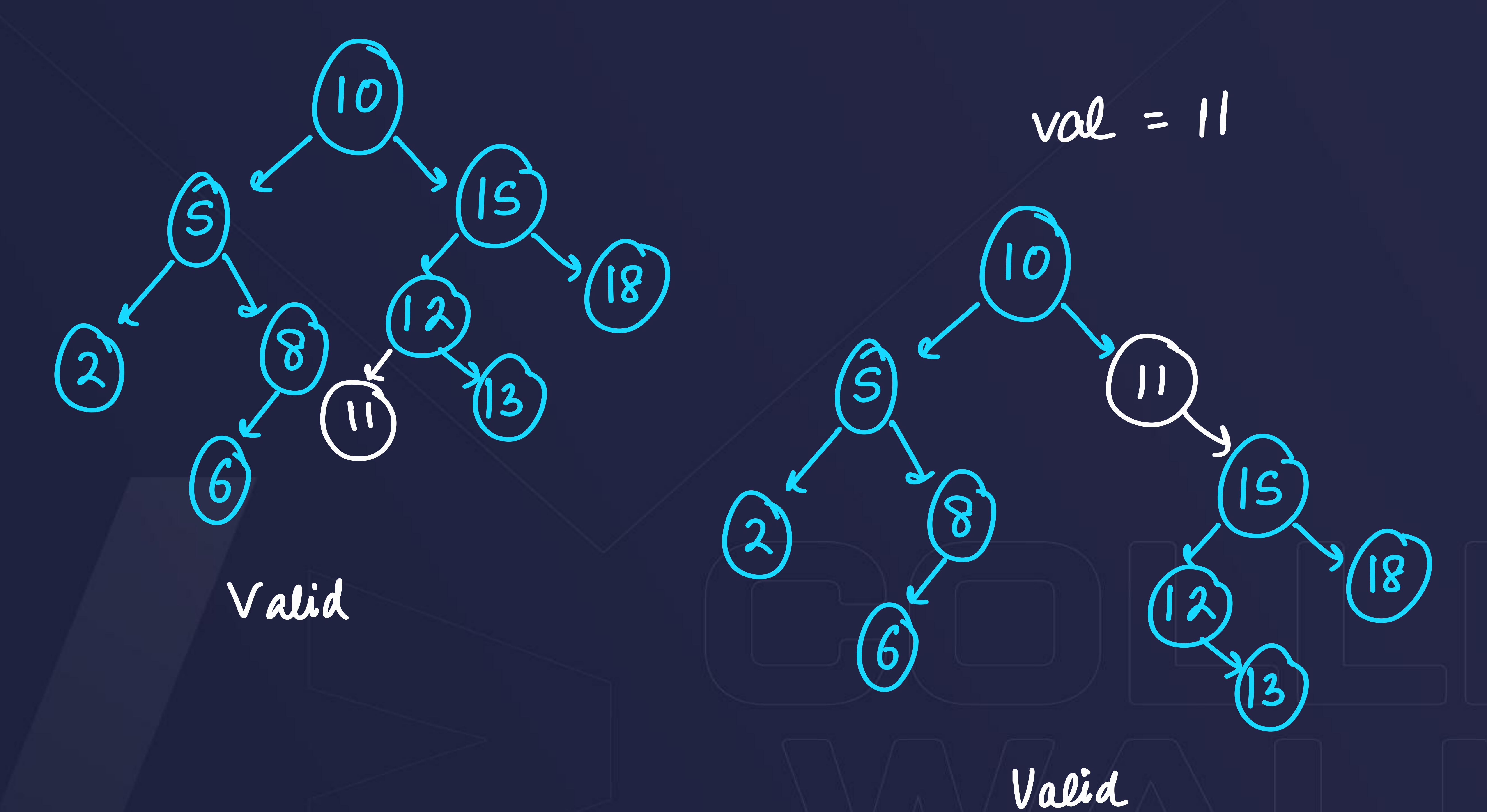
S-C.

Same as T.C.



#### Insert into a BST

#### [LeetCode 701]





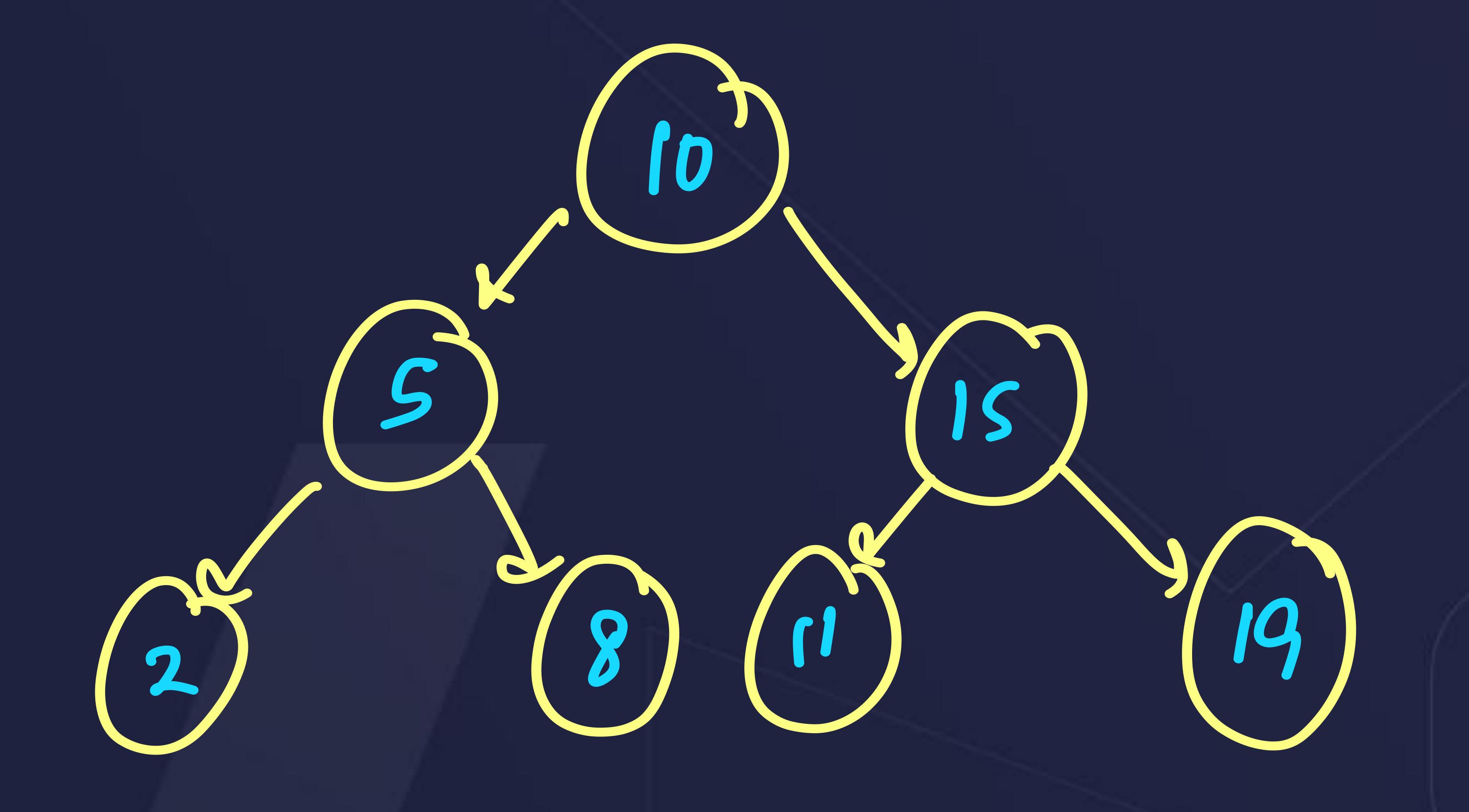
### Traversal (Same as BT)

inorder of BST is SORTED

Preorder, Gnorder, Post Order

XXIMP

X Inorder: (Lebt Root Right)



2 5 8 10 11 15 19 L Sorted

Pre: 10 5 2 8 15 11 19
Post: 2 8 5 11 19 15 10



#### 

Consider a binary search tree with n nodes.

What is the maximum possible height of the tree?

```
A. O(n) n

B. O(togn) lugn

C. O(triogn) nlugn

D. O(sgrt(n)) Sgrt(n)
```



#### MCQ-2

Consider a binary search tree with n nodes. What is the minimum number of comparisons required to search for a value in the worst-case scenario?

- A. O(1)
- B. O(log n)
- C. O(nlogn)
- D. O(n)



#### LCA of a BST

#### LeetCode 235



if (root = val > p=val ble root = val > q=val) return root;
if (root = val > p=val ble root = val > q=val



#### LCA of a BST

#### LeetCode 235





#### Validate BST

#### LeetCode 98

You are given a binary tree (root).

You have to tell if the BT is a BST or not

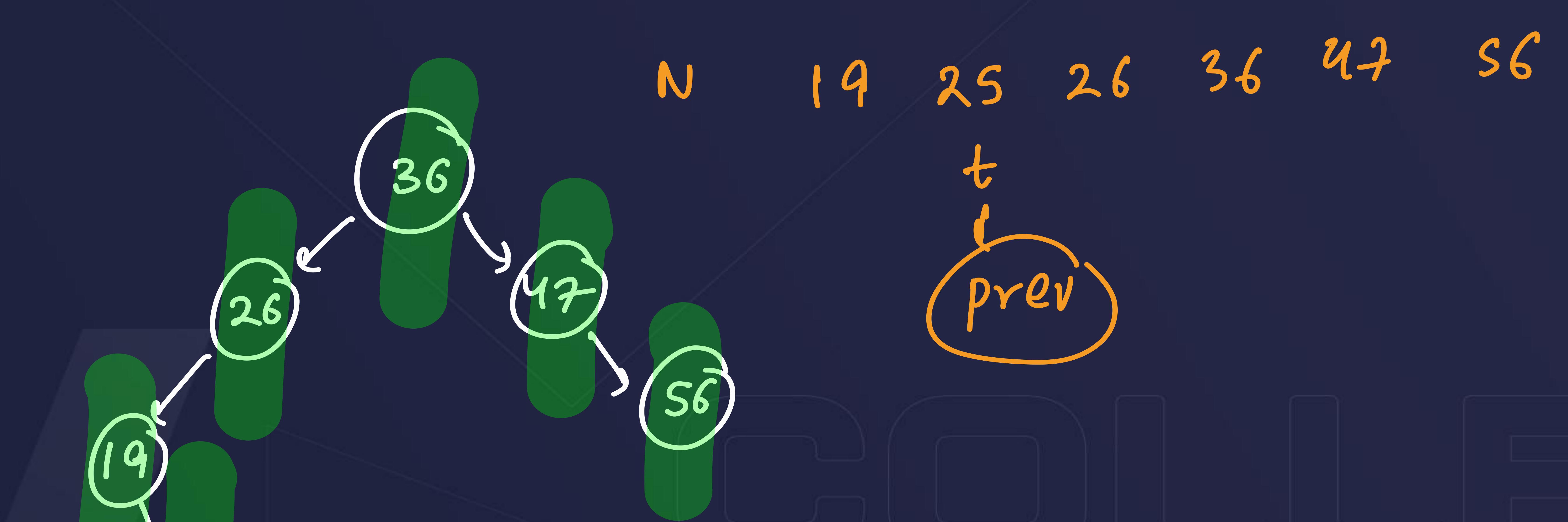
M-I: Use max of tree (Brute Force)



#### Validate BST

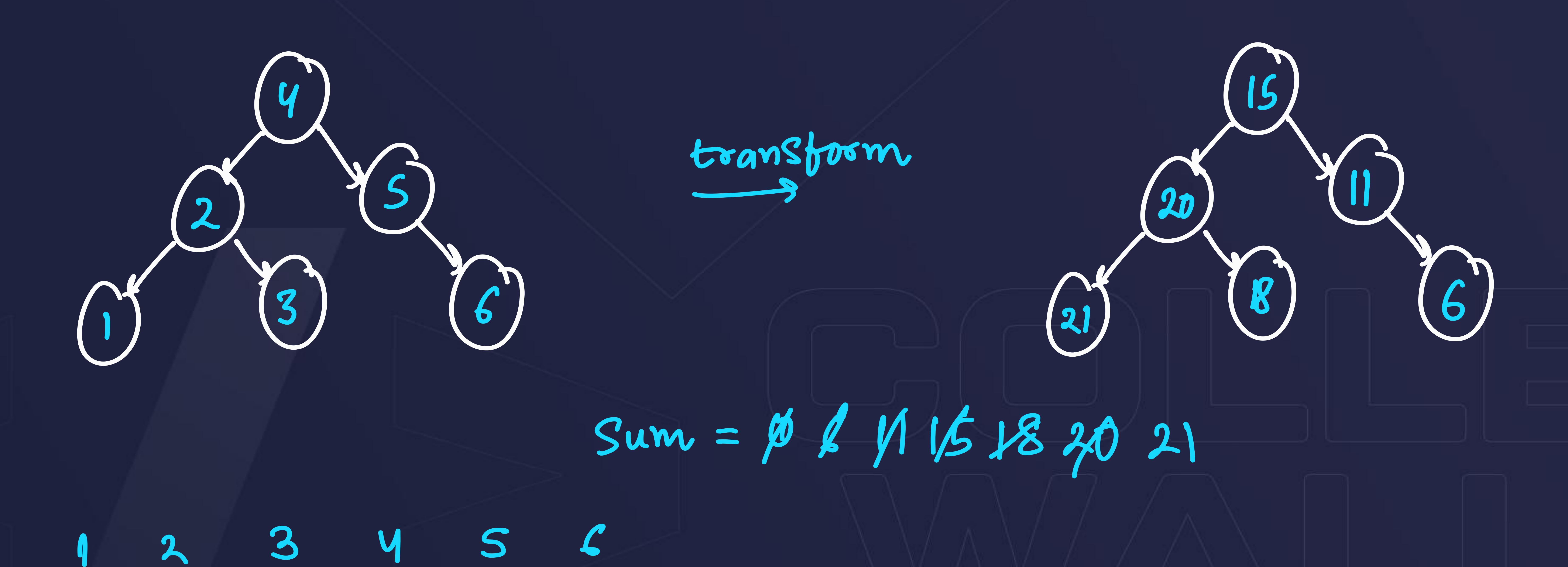
LeetCode 98

Method-2: Hint: Inorder Traversal



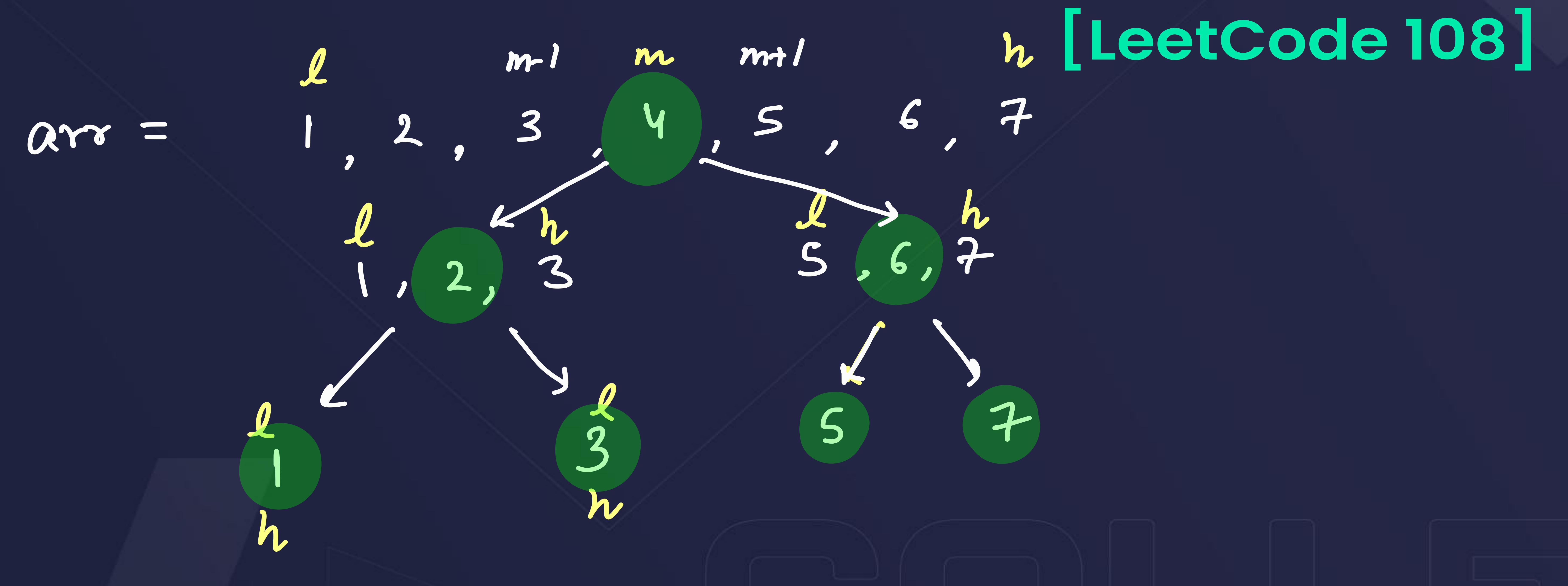


Ques: Given a BST, transform it into a greater sum tree where each node contains the sum of all nodes greater than that node. \* riginal [Leetcode 1038]



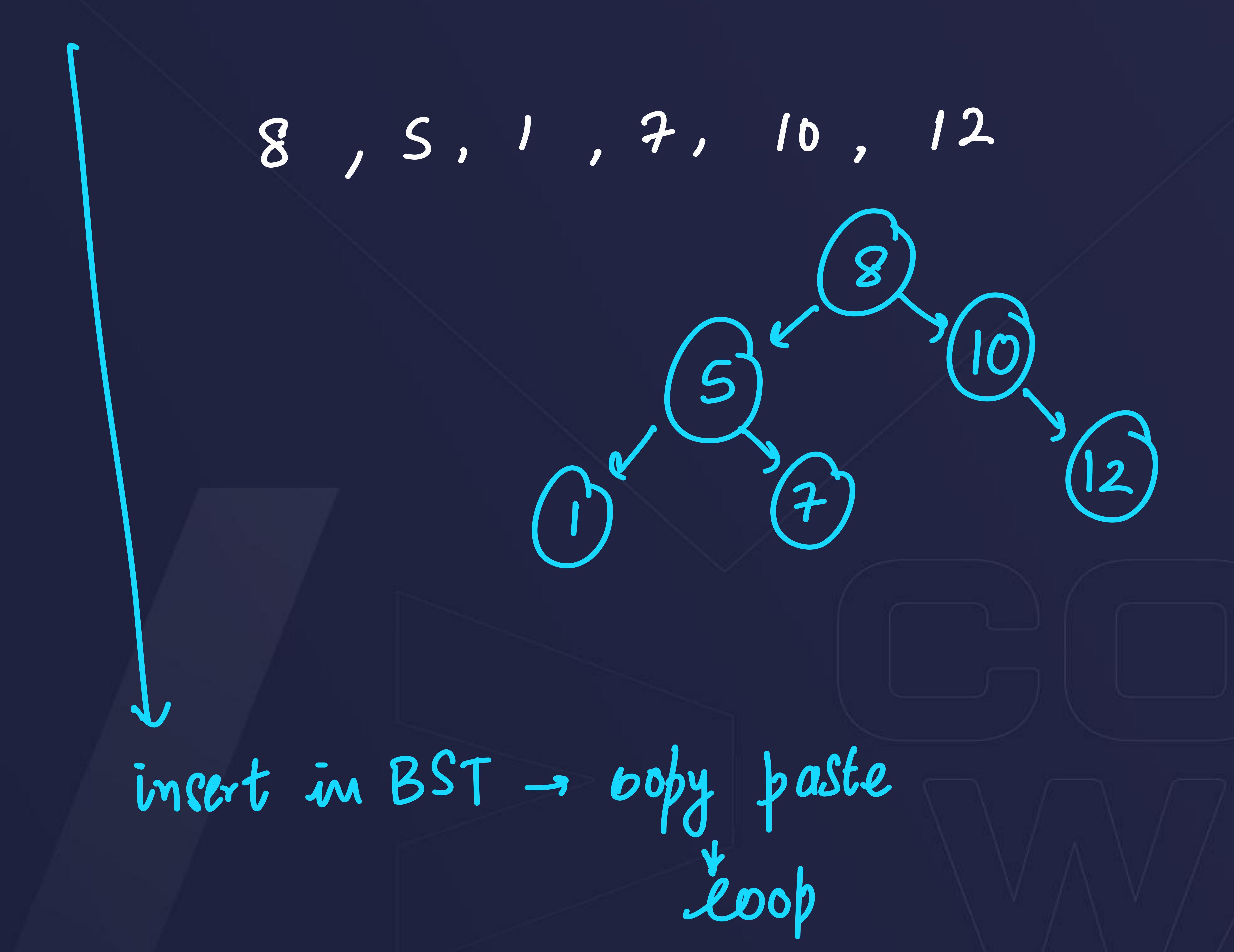


### Converted Sorted Array to Balanced BST





# Ques: Given the preorder traversal of a binary search tree, construct the BST. [Leetcode 1008]





#### Next Lecture

Deletion of Nodes, more questions on BST



## THANKYOU