Introduction to BST & Problem Solving

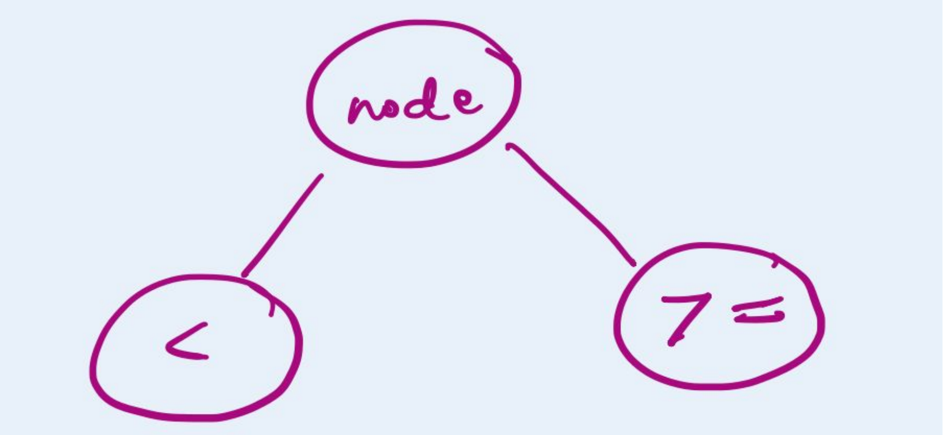
Relevel

by Unacademy

Binary Search Tree

Similar to Binary Tree, Binary Search Tree can also have at most 2 children with a special property that the nodes in the left subtree are less than the Root and the nodes in the right subtree that are greater than or equal to the root node.

Remember, the left subtree and right subtree are also Binary Search Trees.



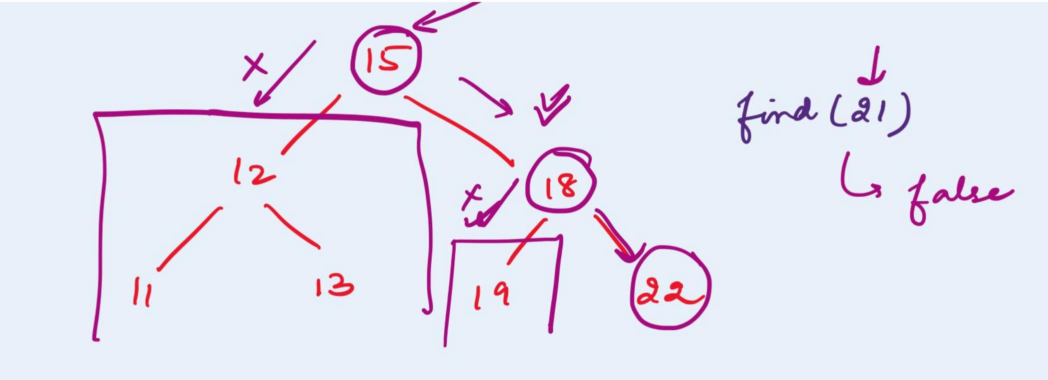
Binary Search Tree

This property makes the searching of elements inside a BST very fast.

Let's take an example.

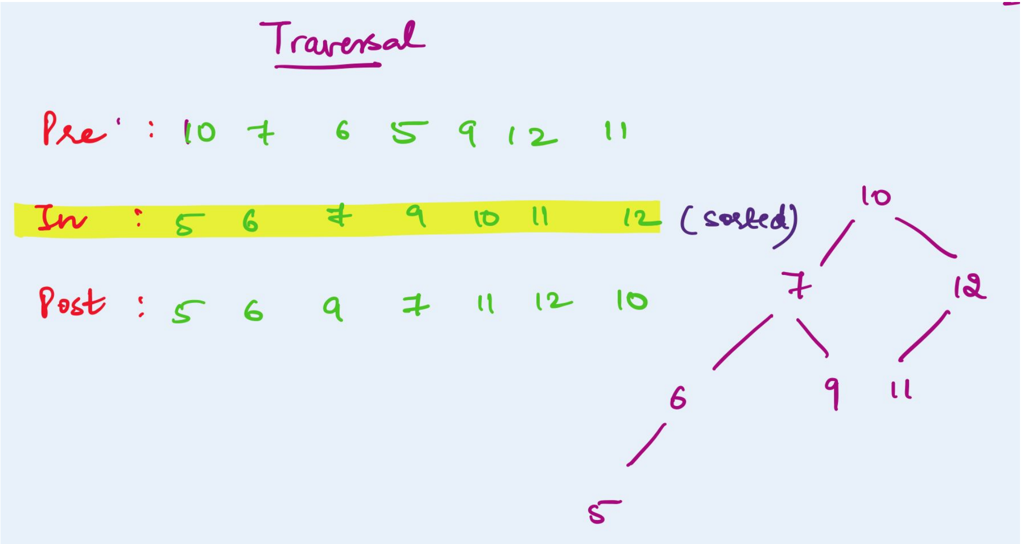
Suppose, in the binary tree given below, you need to find an element x = 21.

Now start comparing the root of the binary tree with x. Since root node (15) < 21, we can be sure that the 21 if present will be present in the right subtree, hence we can discard the left subtree. This way, we can keep on moving recursively. At every level, we either pick the left subtree or right subtree and keep moving forward.



BST Traversal

Now, let’s talk about iterating a BST, which is very similar to BT. One thing to notice here is that the inorder traversal of a BST is sorted in nature.
This property comes very handy in solving a lot of questions.



Traversal

Pre : 10 7 6 5 9 12 11

In : 5 6 7 9 10 11 12 (sorted)

Post : 5 6 9 7 11 12 10

BST Traversal

Operations in BST

We usually perform three types of operation in BST :

Traversal (Similar to BT, discussed above)

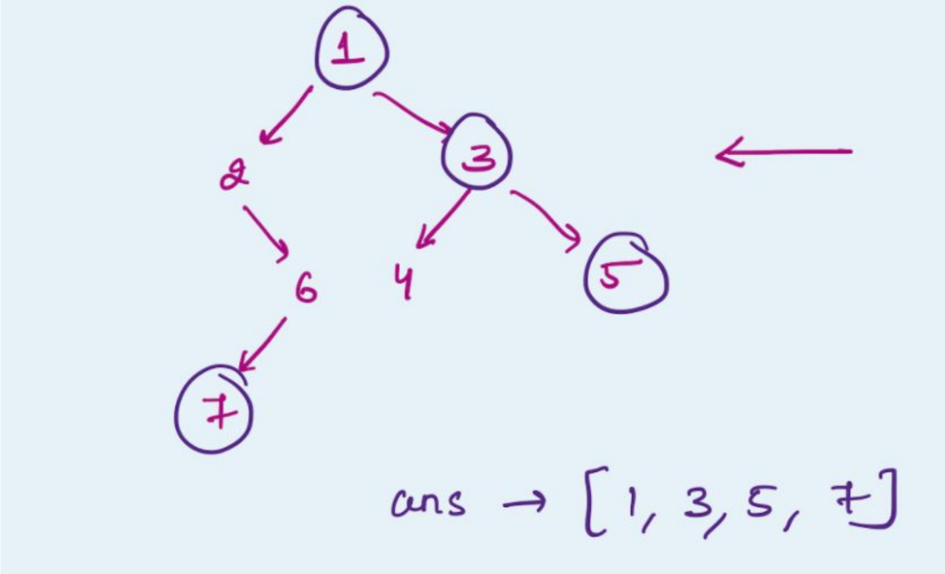
Insertion

Deletion

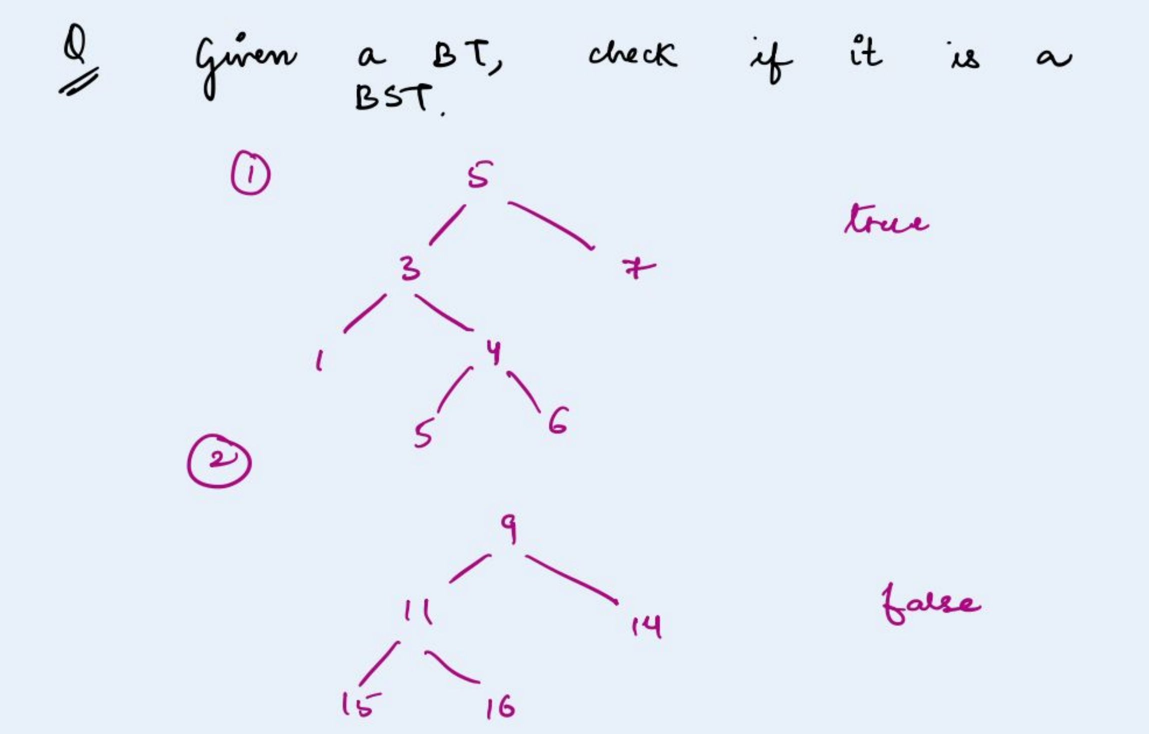
Problem Solving

Problem 1:

Given a BT, print the right view of BST.



Problem 2:



Q Given a BT, check if it is a BST.

① true

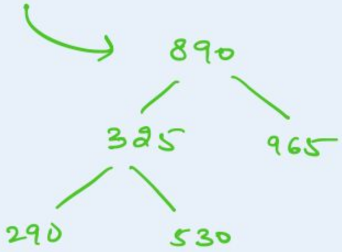
② false

Problem 3:

Q Given the preorder traversal of a BST, find the leaf nodes of BST.

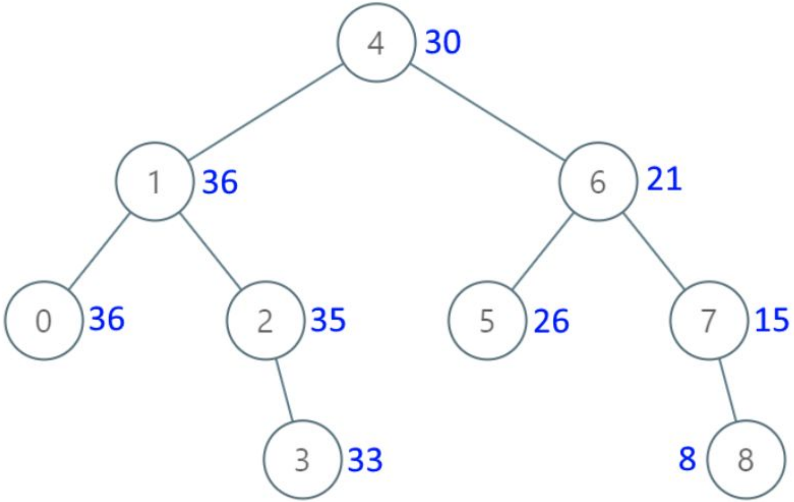
pre → [890, 325, 290, 530, 965]

output → [290, 530, 965]



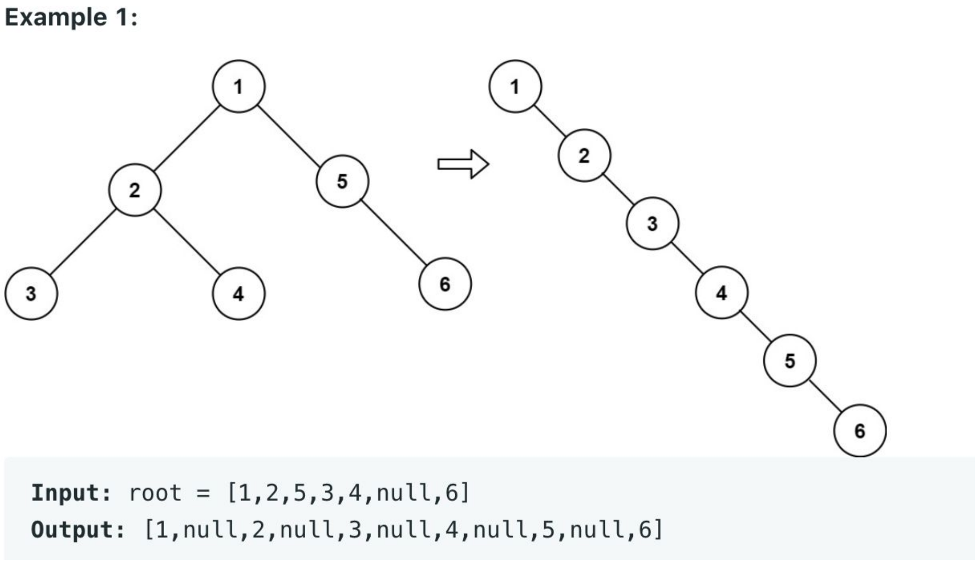
Problem 4:

Given the root of a Binary Search Tree (BST), convert it to a Greater Tree such that every key of the original BST is changed to the original key plus the sum of all keys greater than the original key in BST.



Homework Problems

1. Given a binary tree, flatten the tree into a linked list. The linked list should use the same nodes where the right child is pointing towards the next node in the list and the left child pointer is null.

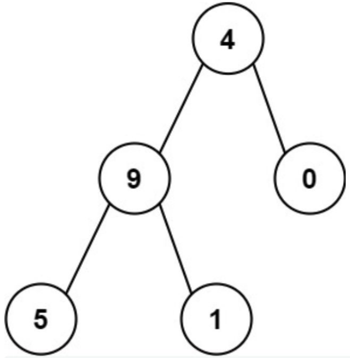


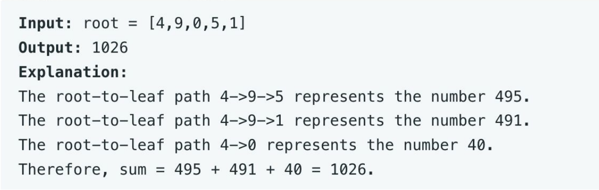
Example 1:

```
Input: root = [1,2,5,3,4,null,6]
Output: [1,null,2,null,3,null,4,null,5,null,6]
```

Homework Problems

2. You are given a binary tree containing only digits from 0 to 9.
   Each root-to-leaf path in the tree represents a number. For example, 1 -> 2 -> 3 represents the number 123. You need to find and return the total sum of all root-to-leaf numbers.





```
Input: root = [4,9,0,5,1]
Output: 1026
Explanation:
The root-to-leaf path 4->9->5 represents the number 495.
The root-to-leaf path 4->9->1 represents the number 491.
The root-to-leaf path 4->0 represents the number 40.
Therefore, sum = 495 + 491 + 40 = 1026.
```

Thank you!