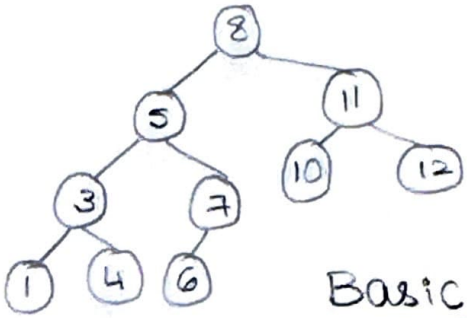
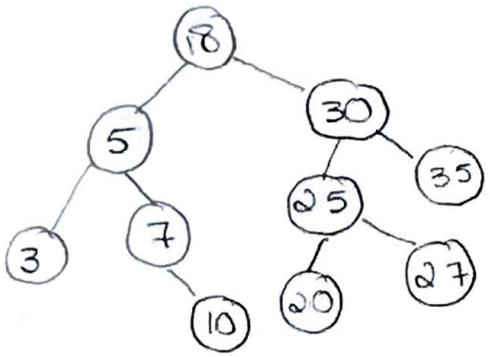
Bottom Up Approach



Basic Rule of Binary Search Tree

Left child's Max Node ↳ Right Most Node $<$ root $<$ Right Childs Min Node ↳ Left Most Node

Eg:



For Node – 18

Left Childs Max Node = 10

LeftChilds Min Node = 3

Right Childs Max Node = 35

Right Childs Min Node = 20

* As per Rule :

Lyt Childs Max Node $<$ root $<$ Right childs Min Node

$10 < 18 < 20$ → True, Hence the Tree is a Valid BST as per this Node

* Wbu Leaf Node

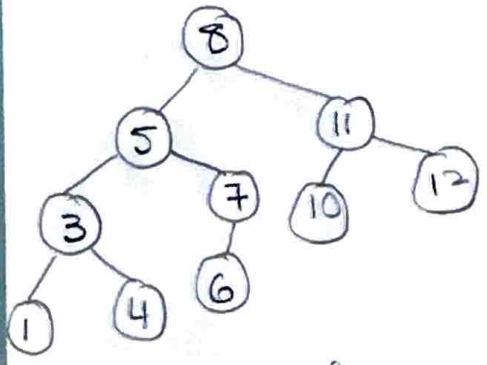
For Leaf Node 10

we will asume

Left Childs Max Node = $-\infty$

Right Childs Min Node = ∞

SO $-\infty < 10 < \infty$



Bottom Up Approach — Inorder

* We Initialize a class, that keep track of size (No of Nodes), Min Value of that tree, Max Value of the Tree and if the Current tree is a BST

Class Node:

Self.size = No of Nodes in the tree

Self.min = Min Value in the tree

Self.max = Max Value in the tree

Self.isBST = Is this a Valid BST or Not

For Leaf Node 1 :

Left Node :

min 0
max ∞
-∞
True

Right Node :

0
∞
-∞
True

Note: Why is min = ∞ and max = -∞

Ans: As per BST rule, we Compare

Left Node Max Value < root < Right Node Min Value

$-\infty < 0 < \infty$

In Order to satisfy the condition, we initialize it this way.

This also helps In keep track of min & max value for leaf Nodes. ie for leaf node, min & max node should be the node itself

So for Leaf Node → 1

$$\text{Min Val} = \min(1, \text{LeftNode.Min Val})$$

$$= \min(1, \infty) = 1$$

$$\text{Max Val} = \max(1, \text{RightNode, max Val})$$

$$= \max(1, -\infty) = 1$$

⇒ How do we count the no of nodes?

* If the given tree is a valid BST

ie LeftChilds Max Value < root < RightChilds Min Value

We just add 1 to Leftchilds_nodes & no of nodes from right child.

* If the tree is not a valid BST then?

we return the maximum no of nodes from Leftchild or the rightchild

So for Node 0 :

$$0 \begin{bmatrix} 1+0+0 \\ \min(1, \infty) \\ \max(1, -\infty) \\ \text{True, True} \end{bmatrix} \Rightarrow \quad 0 : \begin{bmatrix} 1 \\ 1 \\ 1 \\ \text{True} \end{bmatrix}$$

For leaf Node 4

LeftNode Max = $-\infty$

RightNode Min = ∞

so $-\infty < 4 < \infty$ = True

Min Node for TreeNode 4 = Min (4, LeftNode Min Value)

= Min (4, ∞) = 4

Max Node for TreeNode 4 = Max (4, RightNode Max Value)

= Max (4, $-\infty$) = 4

& Since Left and right are both Valid BST

4 will look like $\begin{bmatrix} 1+0+0 \\ 4 \\ 4 \\ \text{True} \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 4 \\ 4 \\ \text{True} \end{bmatrix}$

⇒ For Node 3

Left SubTree = 1 = $\begin{bmatrix} 1 \\ 1 \\ 1 \\ \text{True} \end{bmatrix}$ Right SubTree = 4 $\begin{bmatrix} 1 \\ 4 \\ 4 \\ \text{True} \end{bmatrix}$

For Node 3

Smallest Node from Left SubTree = 1

Largest Node from LeftSubTree = 1

Smallest Node from right SubTree = 4
Largest Node from right SubTree = 4

So:

Min Node for Tree 3 = min (3, LeftTree min Val)
= min (3, 1) = 1

Max Node for Tree 3 = max (3, right Tree max Value)
= max (3, 4) = 4

Since Tree 3 is a valid BST
ie Left child max Value < 3 < right child Min Value
1 < 3 < 4

no of Node = 1 + no of nodes on Left subtree + no of nodes on right subtree
= 1 + 1 + 1 = 3

so Node 3

$$3 = \begin{bmatrix} 3 \\ 1 \\ 4 \\ \text{True} \end{bmatrix}$$

=> For Node 6

Since this is a Leaf Node 6 = $\begin{bmatrix} 1 \\ 6 \\ 6 \\ \text{True} \end{bmatrix}$

⇒ For Node 7

Left SubTree or Left Child Tree = $6\begin{bmatrix} 1 \\ 6 \\ 6 \\ \text{True} \end{bmatrix}$

Right Child Tree = $\begin{bmatrix} 0 \\ \infty \\ -\infty \\ \text{True} \end{bmatrix}$

↑ This since Node 7 has No right child

Left Tree:
- Smallest Node = 6
- Largest Node = 6

Right Tree
- Smallest Node = ∞
- Largest Node = $-\infty$

So:

Min Node for Tree 7 = min (7, Left Tree Min Node)
= min (7, 6) = 6

Max Node for Tree 7 = max (7, Right Tree Max Value)
= max (7, $-\infty$) = 7

This makes Sense:

The min Value for this Tree is Left childs Left Most Node = 6

The Max Value is Right childs Right most Node = 7 itself since there is No right child

So No of nodes for 7 will, be

Since this is a Valid BST

Lft child Max Value (6) < 7 < right childs Min Val ∞

$1 + 1 + 0 = 2$

$$7 \begin{bmatrix} 2 \\ 6 \\ 7 \\ \text{True} \end{bmatrix}$$

⇒ Node 5

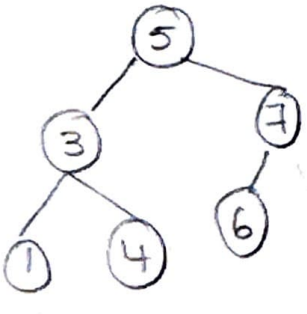
$$\text{LeftChild} = 3 \begin{bmatrix} 3 \\ 1 \\ 4 \\ \text{True} \end{bmatrix} \qquad \text{Right child} = 7 \begin{bmatrix} 2 \\ 6 \\ 7 \\ \text{True} \end{bmatrix}$$

- Min Node for Tree 5 = Min (5, Left child Min)
  = Min (5, 1) = 1
- Max Node for Tree 5 = Max (5, Right child Max)
  = Max (5, 7) = 7
- This is a Valid BST

  Left Child Max Value $4 < 5 < 6$ Right Child Min Value

no of nodes = 1 + 3 + 2 = 6

= 1 + no of nodes from left child + no of nodes from right child



so $5 = \begin{bmatrix} 6 \\ 1 \\ 7 \\ \text{True} \end{bmatrix}$

=> Node 10 , Leaf Node

Left Side = $\begin{bmatrix} 0 \\ \infty \\ -\infty \\ \text{True} \end{bmatrix}$ Right child = $\begin{bmatrix} 0 \\ \infty \\ -\infty \\ \text{True} \end{bmatrix}$

Min Node = min (10, Leftchild Min) = (10, ∞) = 10

Max Node = max (10, right child Max) = (10, -∞) = 10

No of node = 1 + 0 + 0 = 0

So 10 = $\begin{bmatrix} 1 \\ 10 \\ 10 \\ \text{True} \end{bmatrix}$

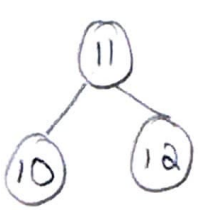
=> Node 12, Leaf Node 12 = $\begin{bmatrix} 1 \\ 12 \\ 12 \\ \text{True} \end{bmatrix}$

=> Node 11,

Left child = $\begin{bmatrix} 1 \\ 10 \\ 10 \\ \text{True} \end{bmatrix}$ Right Child = $\begin{bmatrix} 1 \\ 12 \\ 12 \\ \text{True} \end{bmatrix}$

Min Node = min (11, Leftchild Min) = min (11, 10) = 10

Max Node = max (11, right child Max) = max (11, 12) = 11

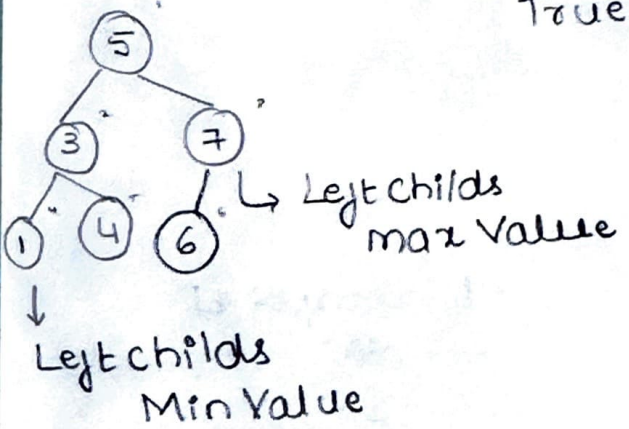


no of nodes = 1 + 1 + 1 = 3

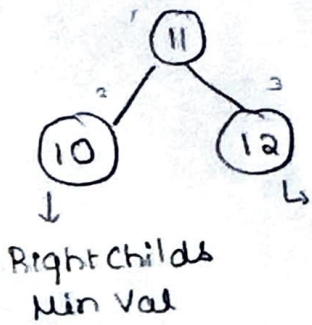
11 = $\begin{bmatrix} 3 \\ 10 \\ 12 \\ \text{True} \end{bmatrix}$

=> Node 8

Left child = 5 = $\begin{bmatrix} 6 \\ 1 \\ 6 \\ \text{True} \end{bmatrix}$



Right Child = 11 $\begin{bmatrix} 3 \\ 10 \\ 12 \\ \text{True} \end{bmatrix}$



Min Node = min (8, Left child Min Val) = min (8, 1) = 1

Max Node = max (8, Right child Max Val) = max (8, 12) = 12

no of nodes = 1 + 6 + 3 = 10

is Valid BST

Left Child Max Val < root < Right child Min Val

7 < 8 < 10 ∴ 8 = $\begin{bmatrix} 10 \\ 1 \\ 12 \\ \text{True} \end{bmatrix}$

Vizualize