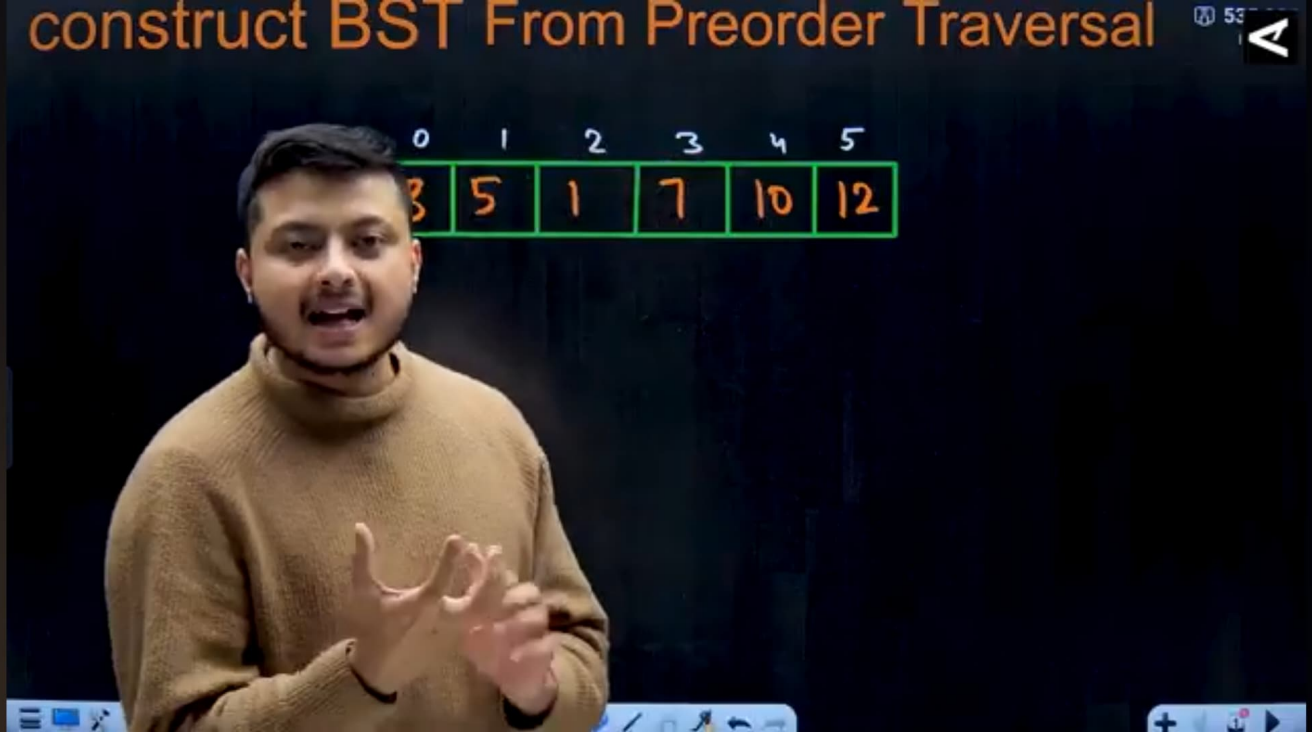

construct BST From Preorder Traversal
0 1 2 3 4 5
5 1 7 10 12

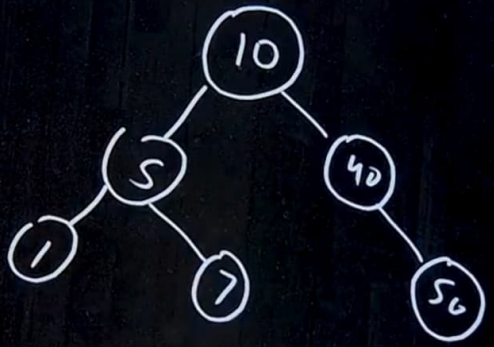
```
Node* BST(vector<int> preorder, int &index
          int lower, int upper)
{
    if(index == preorder.size() || preorder[index] < lower
                                || preorder[index] > upper)
        return NULL

    Node *root = new Node(preorder[index++]);
    root->left = BST(preorder, index, lower, root->data);
    root->right = BST(preorder, index, root->data, upper);

    return root;
}
```

Construct BST From post order

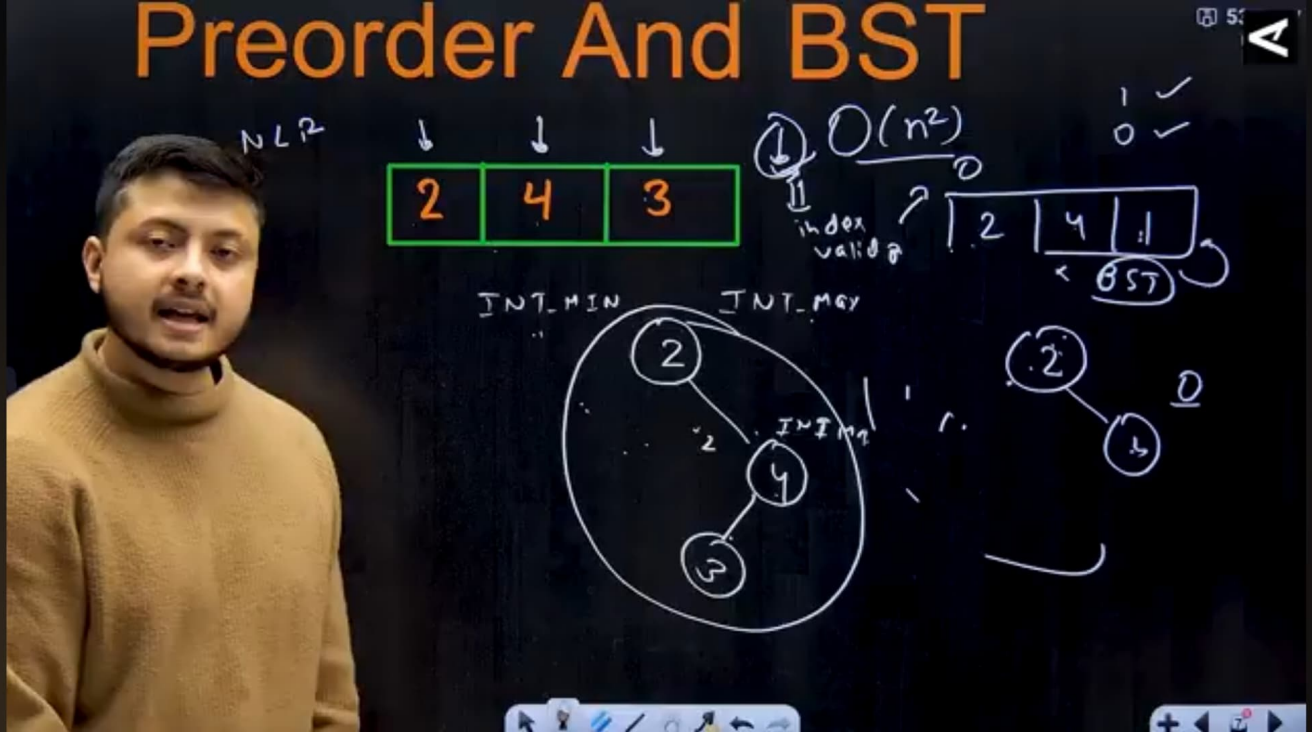
0	1	2	3	4	5
1	7	5	50	40	10

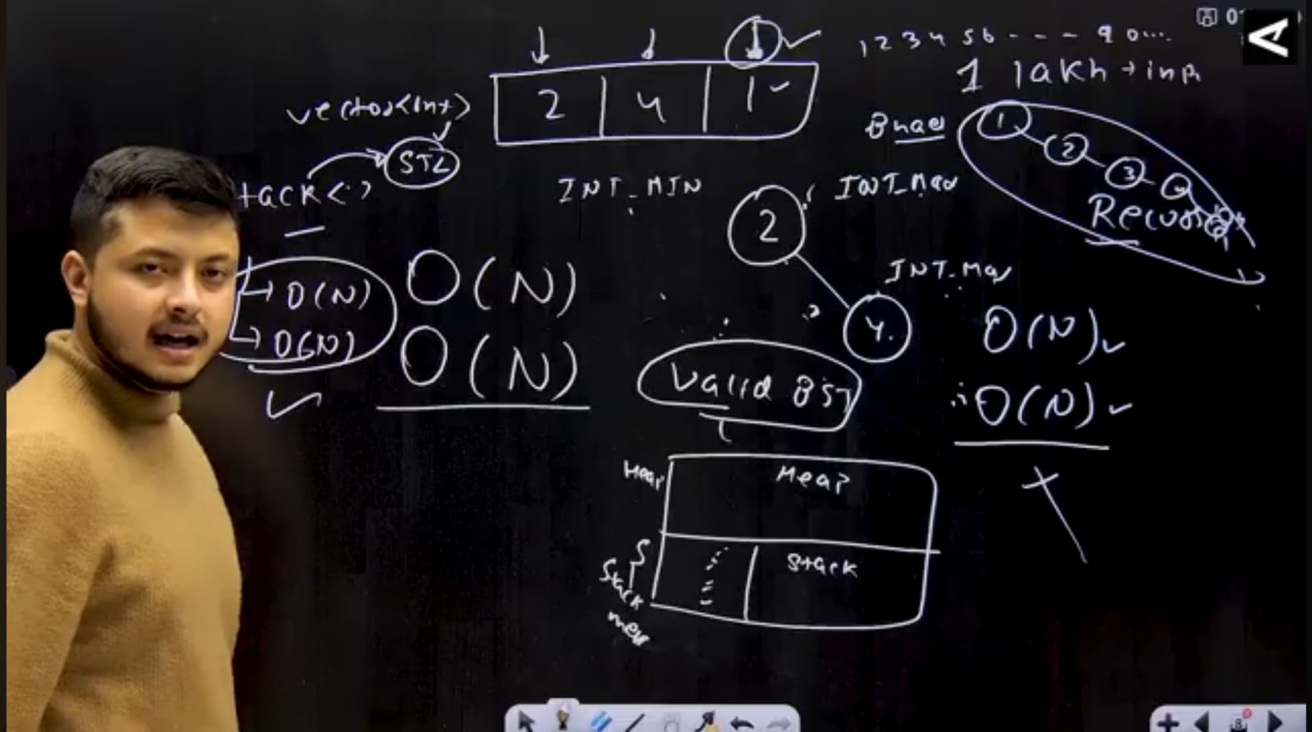
L R N

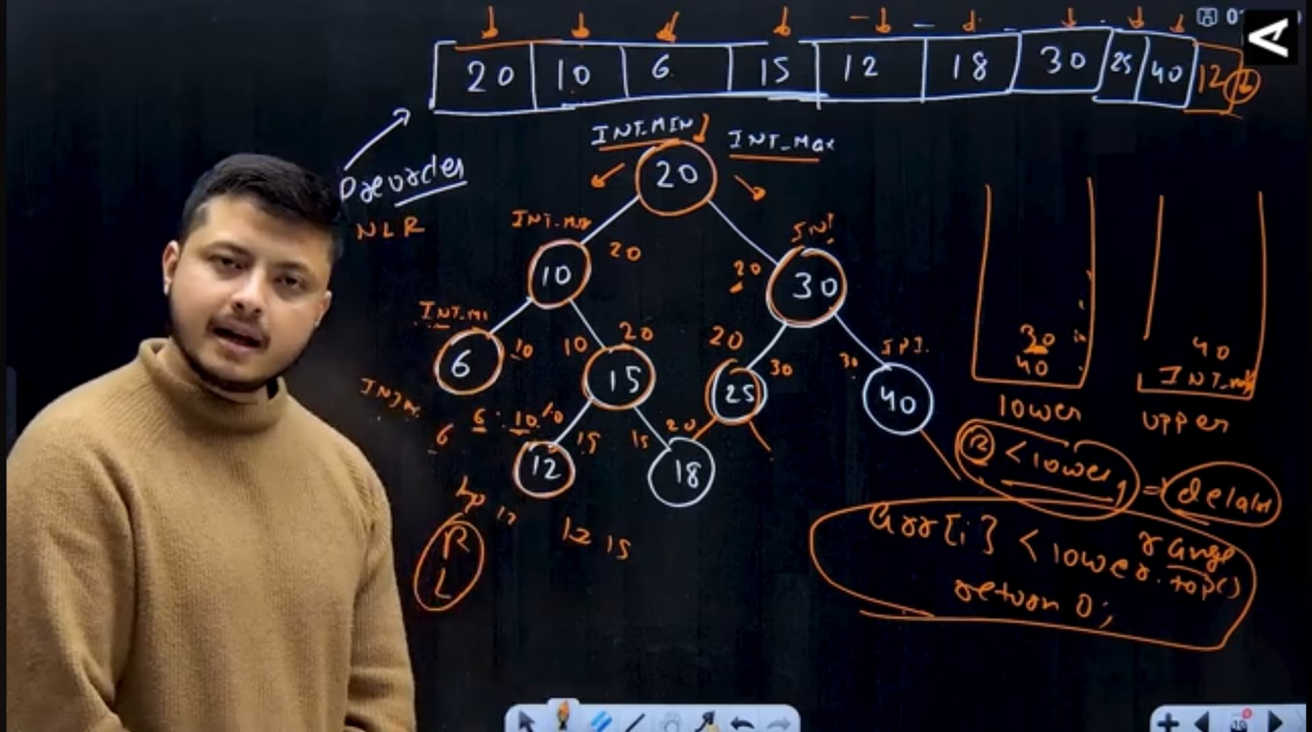
Post



Preorder And BST




vector<int>
STL
2
4
1
1 2 3 4 5 6 - - - 9 0 ....
1 lakh → inp
Brute
1
2
3
Recursion
INT_MIN
INT_MAX
2
4
INT_MAX
stack<>
O(N)
O(N)
O(N)
O(N)
O(N)
Valid BST
Heap
Heap
Stack
Stack
Stack mein


20 10 6 15 12 18 30 25 40 12
Preorder
NLR
INT_MIN
INT_MAX
20
10
30
6
15
25
40
12
18
lower
upper
delete
return 0;

for (int i=0; i<n ; i++)

① If (arr[i] < lower.top())
    return 0;

② while (arr[i] > upper.top())
    upper, low pop

③
left = lower
right = upper
lower, upper, pop
lower push : arr[i]
upper.push : right
lower.push : left
upper.push : curr[i]