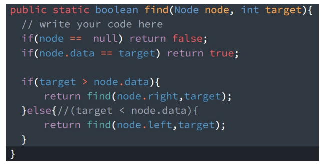
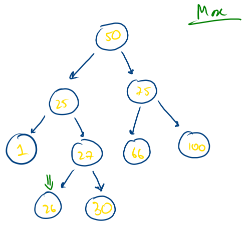


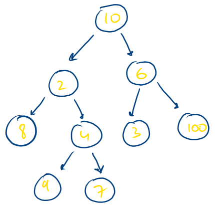
```
public static boolean find(Node node, int target){
  // write your code here
  if(node ==  null) return false;
  if(node.data == target) return true;

  if(target > node.data){
     return find(node.right,target);
  }else{//(target < node.data){
     return find(node.left,target);
  }
}
```

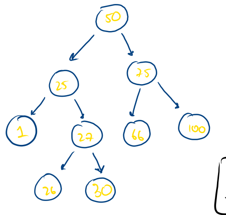
26

Max



Binary Search trees





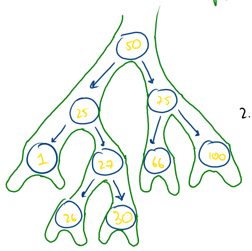
I For every Node

all the nodes in left subtree should have < values.

all the nodes in right subtree should have > values.

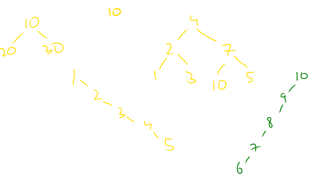
II Inorder of BST is sorted

if(root == null) return)



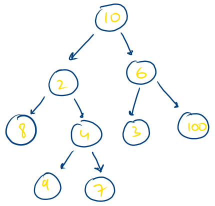
2. In : 1, 25, 26, 27, 30, 50, 66, 75, 100

Pre
L
In
R
Post



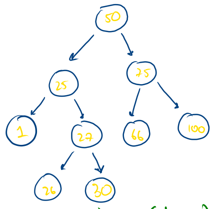
Normal BT

100?

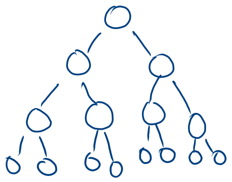


TC: O(n)

Cool BST 👍



TC: O(H) = O(log n)



15

$2^4 - 1 = 15$

$\approx 2^4$

$\log_2(2^{20}) = 4$

$4 \Rightarrow 2^4 \Rightarrow 16$

$16 \rightarrow 16$

O(n)

log(n)

1024

1

O(1)

$n = 1024 \Rightarrow 2^{10}$

$n = 2^{32}$

$\log_2(1024) = 10$

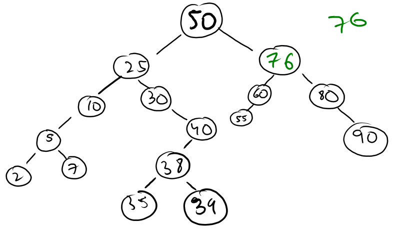
$\log_n(2^{32}) = 32$

$O(1) \approx \log(n)$

Validate BST

rMin

40 ⇒ Max from left or
55 ⇒ Min from right



76

Case 1
leaf Node
* return null

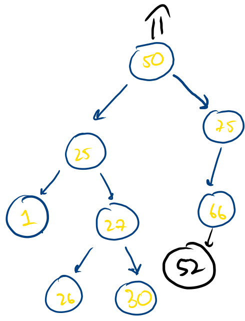
Case 2
Only left child
* Retun leftchild

Case 3
Only right child
* Return right chid

Case 4
Both children

```
rMin = Min(node.right)
node.val = rMin.val
delete(node.right, rMin.val)
return node,
```

52

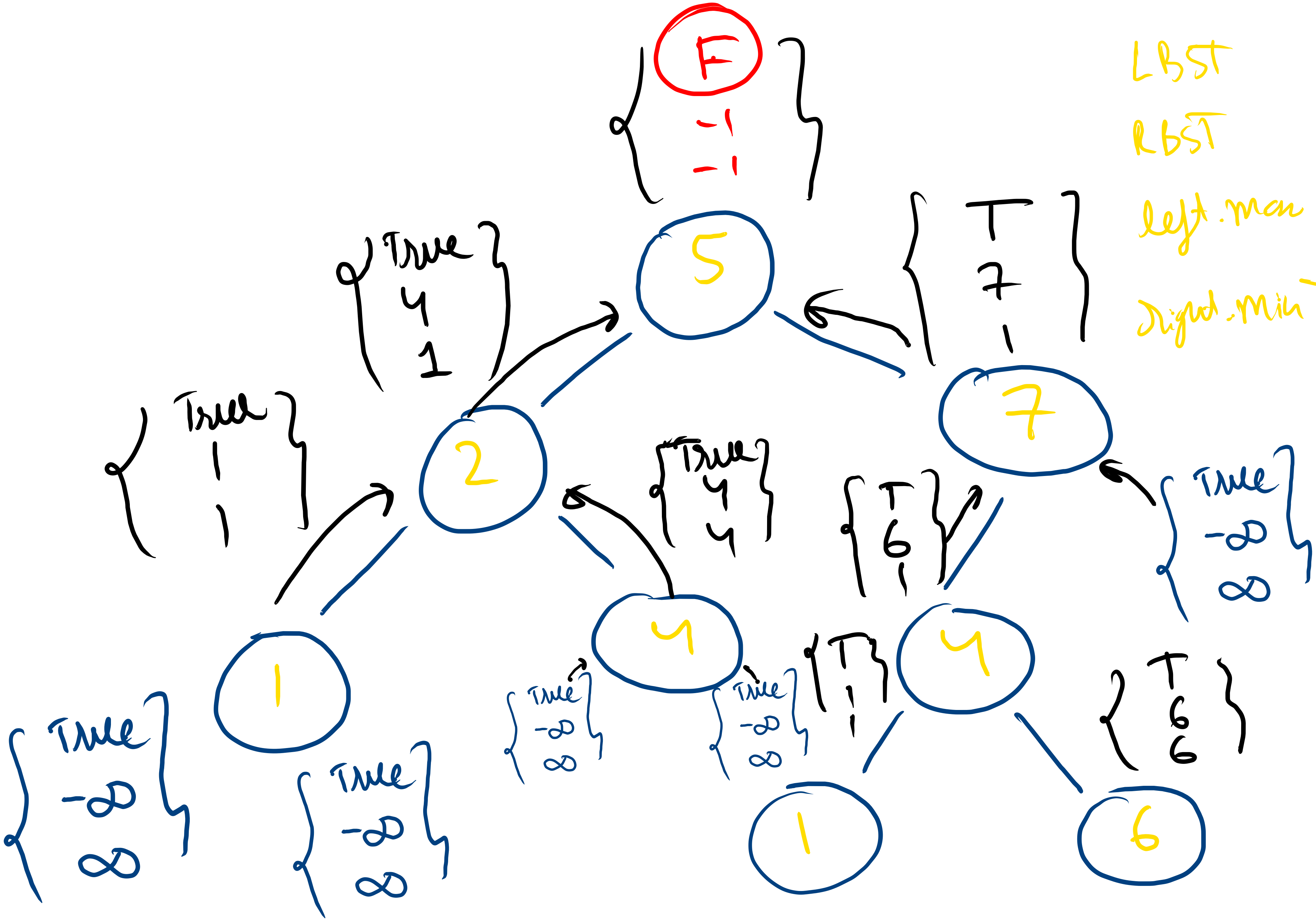


```
func(node, val) {
  if(node==null) {
    Node nn = new Node(val)
    return nn;
  }

  if(val > node.data) {
    updated right = fun(node.right, val)
    node.right = updated right
  } else {
    updatedleft = func(node.left, val)
    node.left = updated left
  }
  return node
```

Validate BST

Validate BST

isBST
Max
Min



LBST
RBST
left.max < node.val
right.min > node.val