

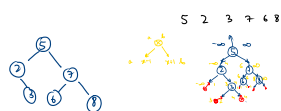
1-2-3
4-5-6
7-8-9



[1 2 1 2 3 4]

0

(2,1) (1,2) (2,3) (3,4) (4,5)



```

int isBST(int root) {
    if (root == null) return true;
    if (root == 0) return false;
    if (root < 1 || root > 100) return false;
    // Base case
    Node n = new Node(root);
    n.left = isBST(n.left);
    n.right = isBST(n.right);
    return n;
}

```



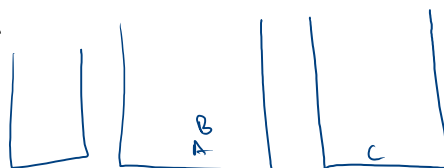
(0,0)

W(A)
W(B)
W(C)
U
R
U
Read()
U
U
Read

1 2 3 4 5

AB

BA



curr

write -> curr.push(curr.pop())
undo -> prev.push(curr.pop())
redo -> curr.push(prev.pop())

Read -> Temp stack
Push element of curr to temp
While moving element from temp to curr, print



(5-2) = 3
(5-2) = 3

