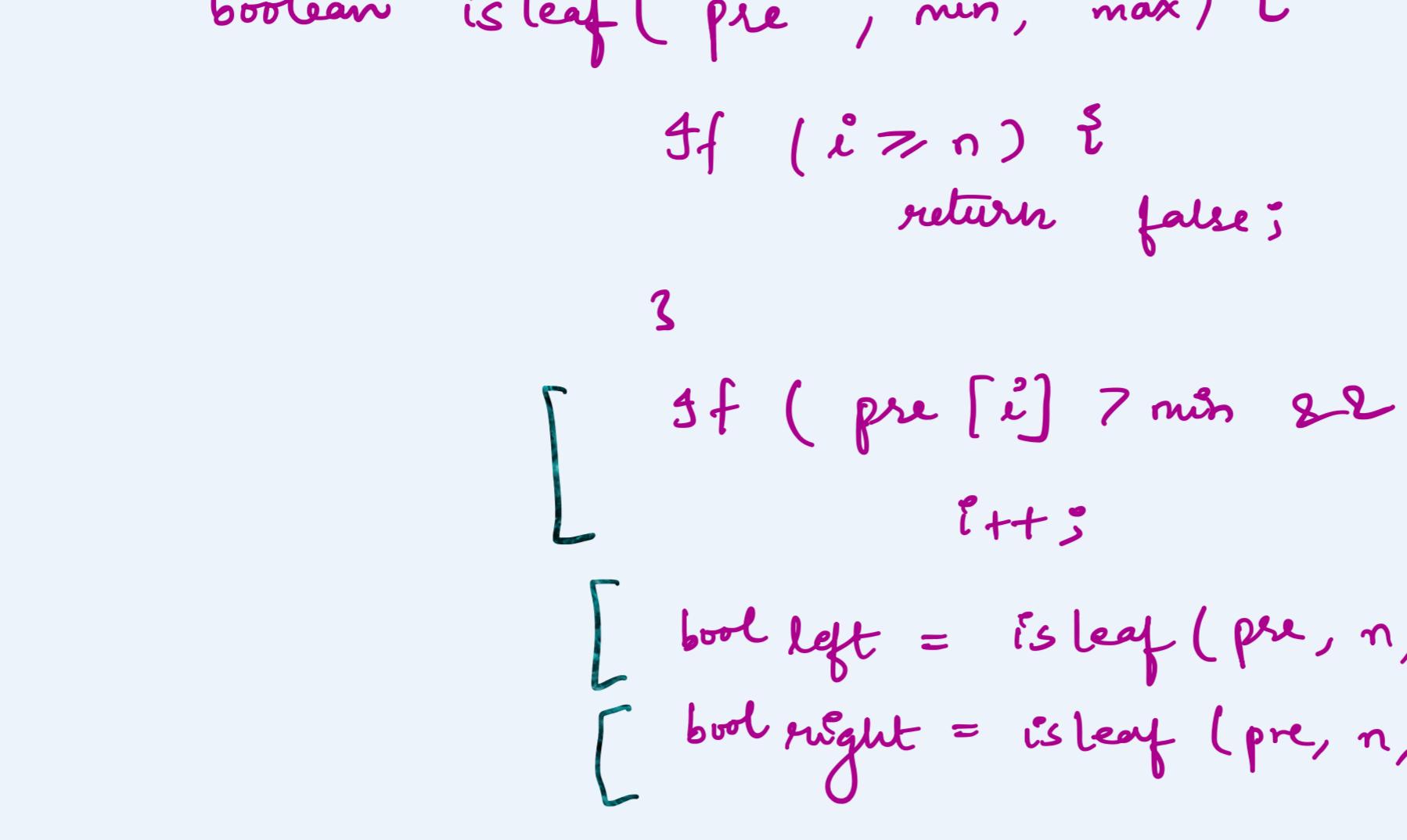
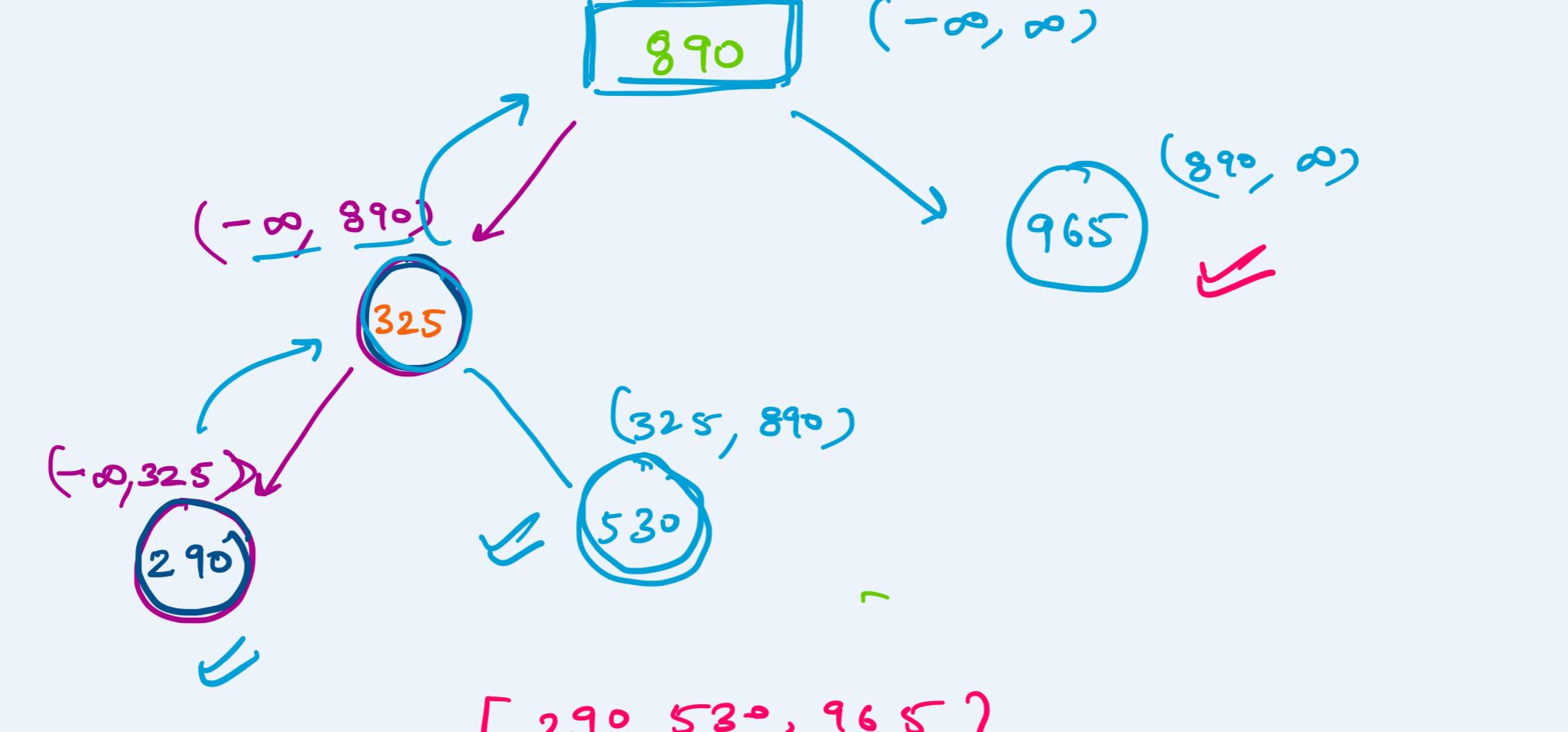


Q Given the preorder traversal of a BST.

Find the leaf nodes of the BST.



boolean isLeaf(pre, min, max) {

if (i > n) { return false; }

if (pre[i] >= min & pre[i] <= max) { i++; }

bool left = isLeaf(pre, n, min, pre[i-1]);  
 bool right = isLeaf(pre, n, pre[i-1], max);

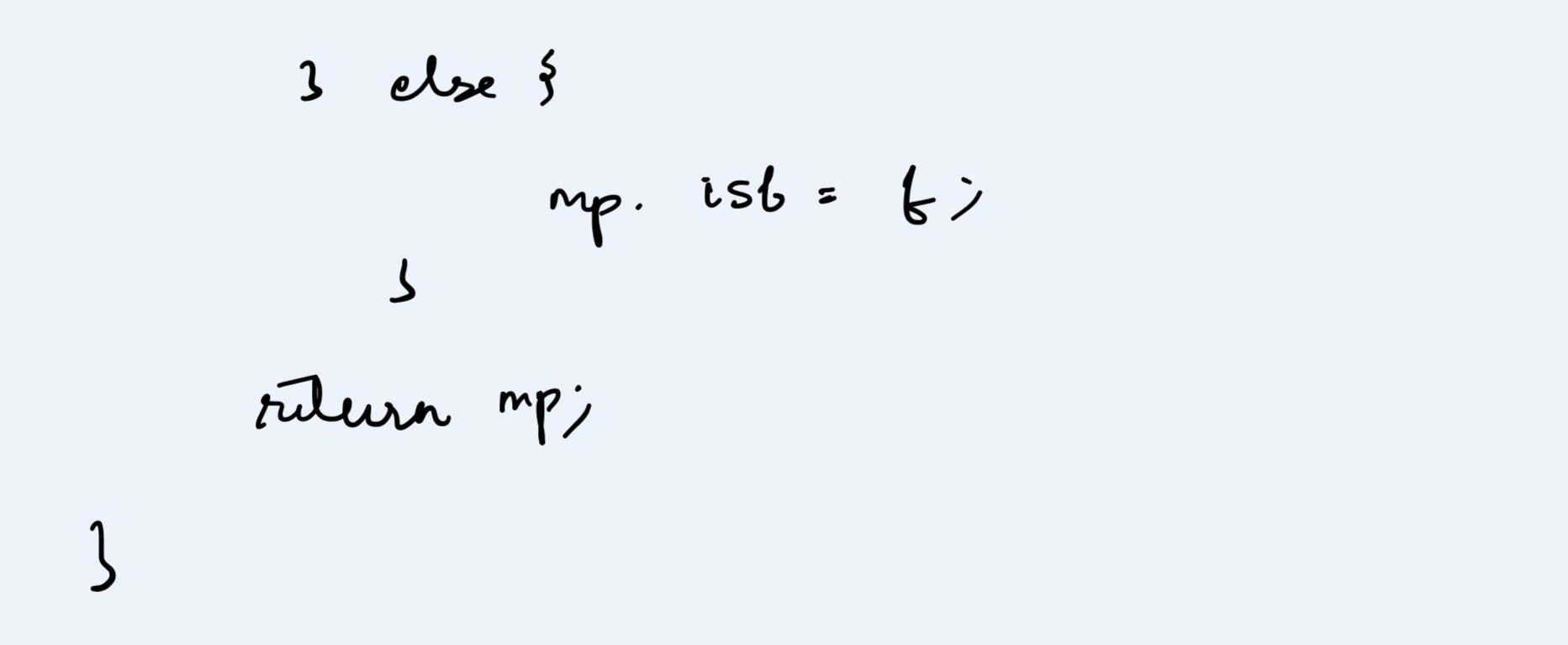
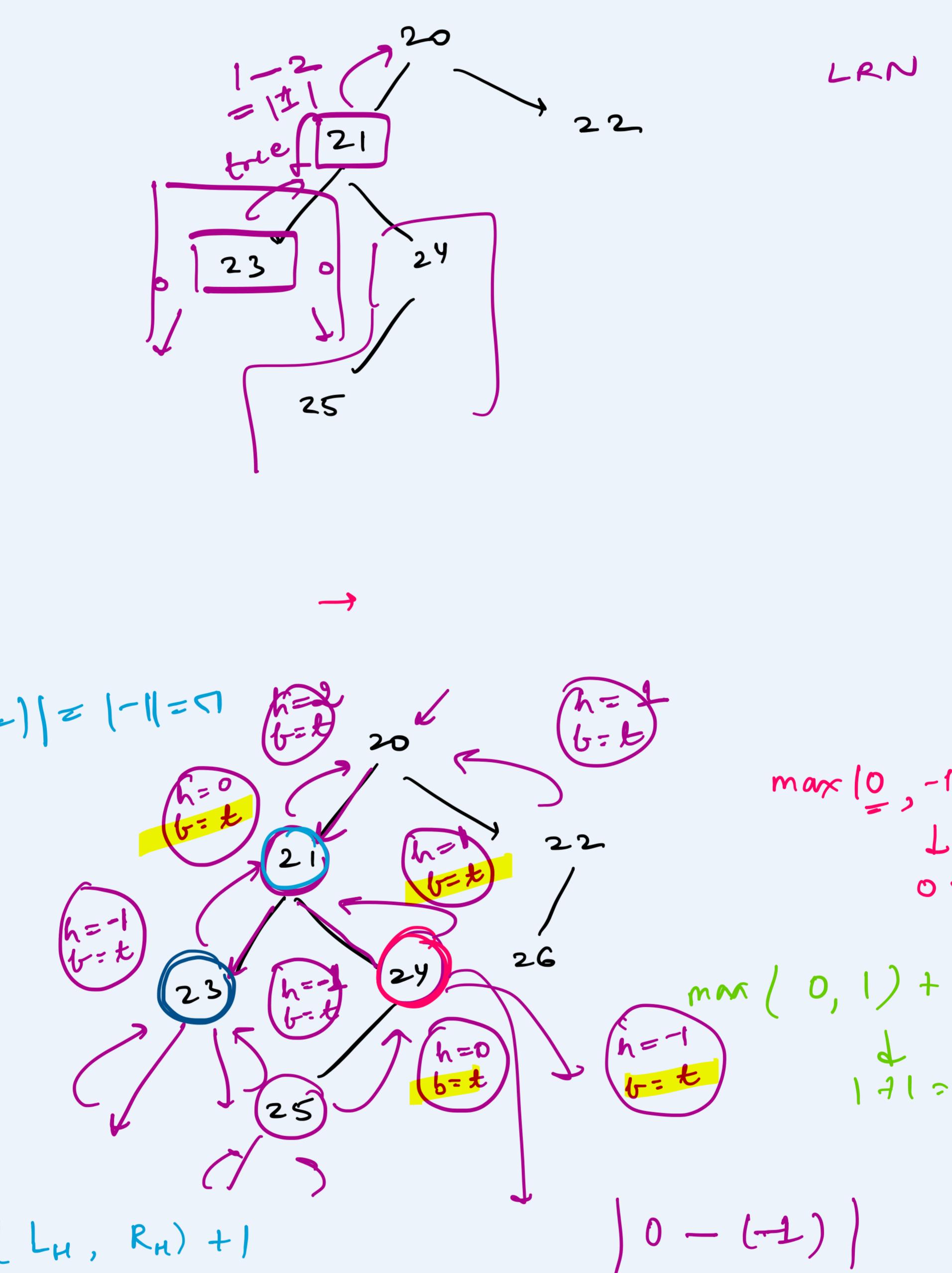
if (!left & !right) { print(pre[i-1]); }

return true;

return false;

TC → O(n)

Q Given the root of a BST, convert it into a greater tree such that every key of the original BST is changed to the original key + sum of all keys greater than original key.



TC → O(n)

if (root.right != null)

convert(root.right);

pre = root.val = (pre + root.val);

if (root.left != null) {

convert(root.left);

return root;

}

return mp;

}

Q Given a BT, check if it is balanced or not.

abs(HL - HR) ≤ 1

