while (node.left != null) {
}
→ skeleton of step 1

Flatten binary tree : Step 0:

traverse until left != null

Traverse the whole tree until null

↓

while (node != null)

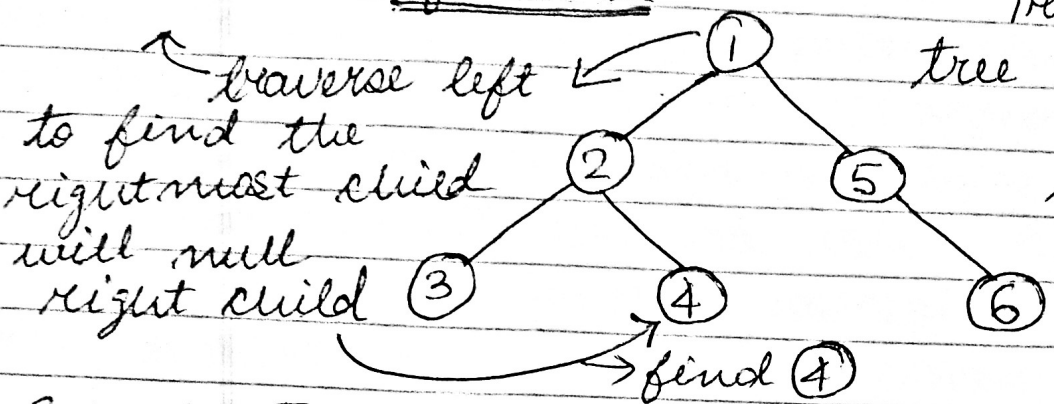
{

step 1:

step 2:

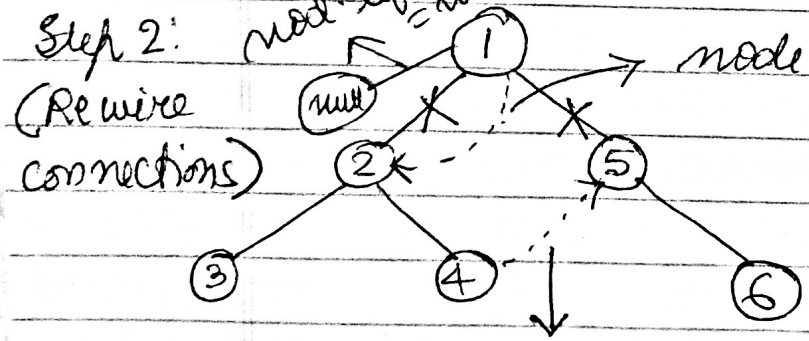
step 3:

}



Step 1: Find the rightmost node, which has no right child. This because we will append the original node's right child (@ 1 → original node) to the rightmost node's right child which has no more right child (:4 in this example).

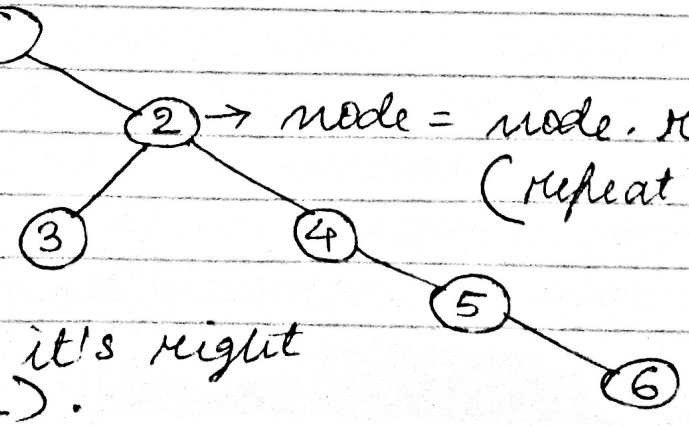
Step 2: (Rewire connections)



node.right = node.left

rightmost.right = node.right

Step 3: move to original node's right child (original node is 1. Now it's right child is 2).



→ node = node.right

(repeat above steps).