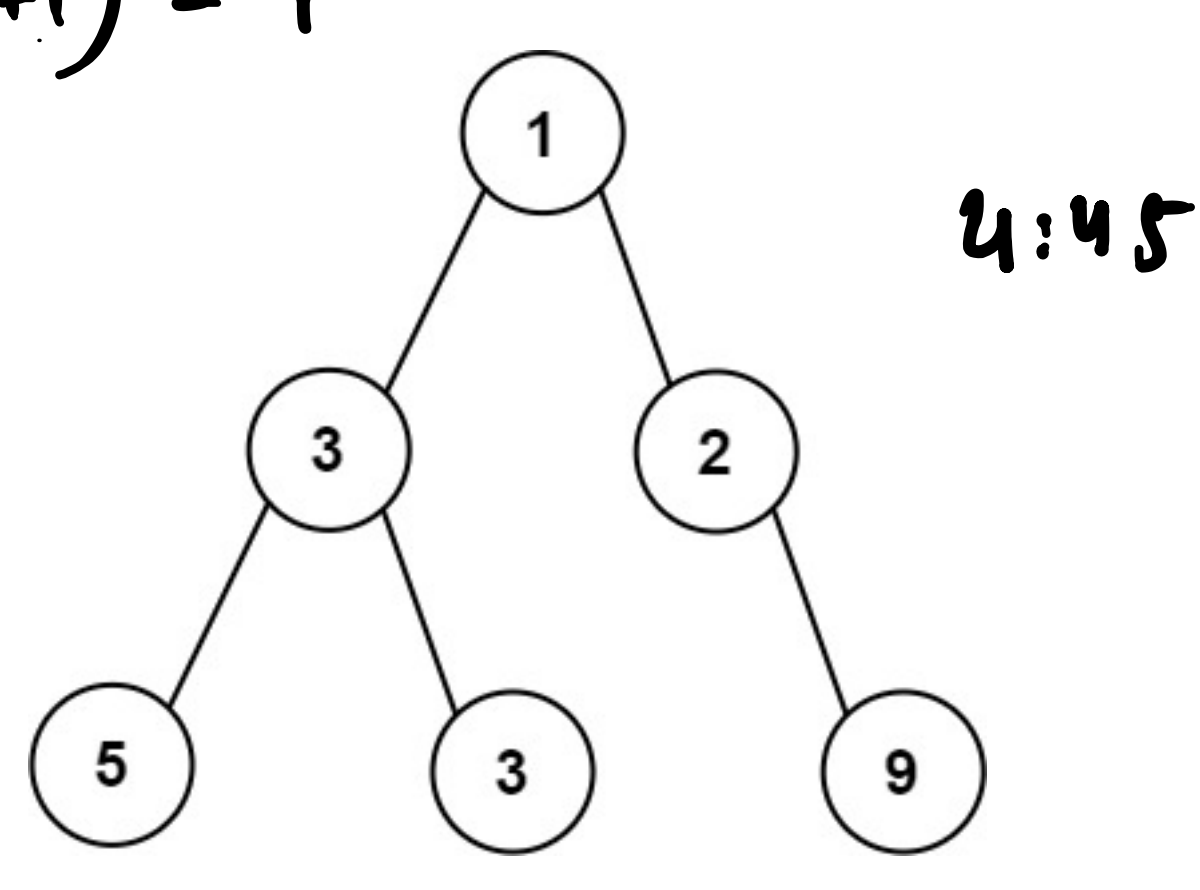
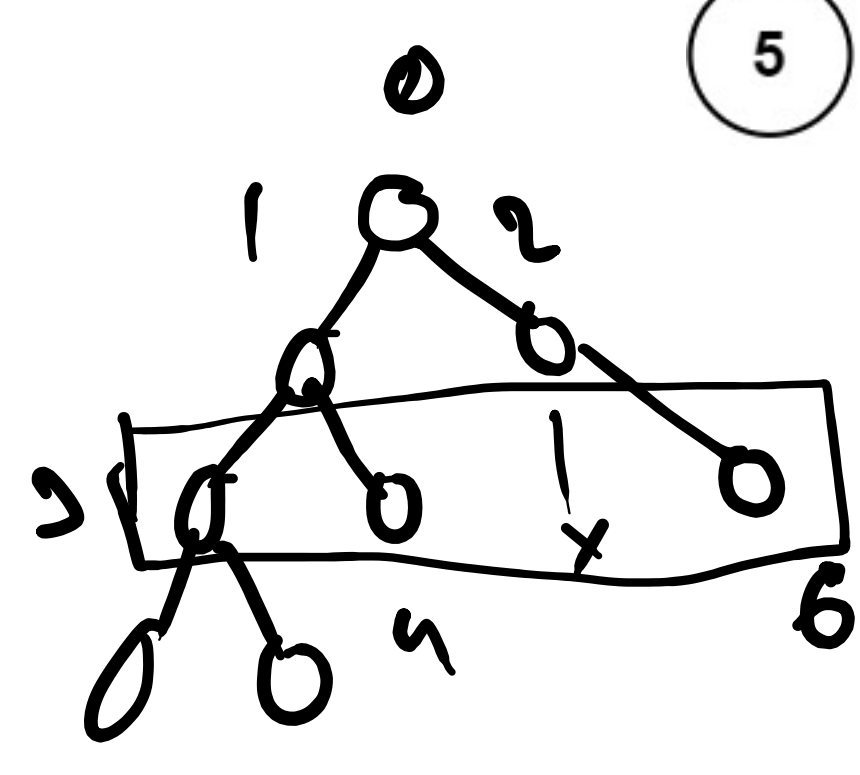


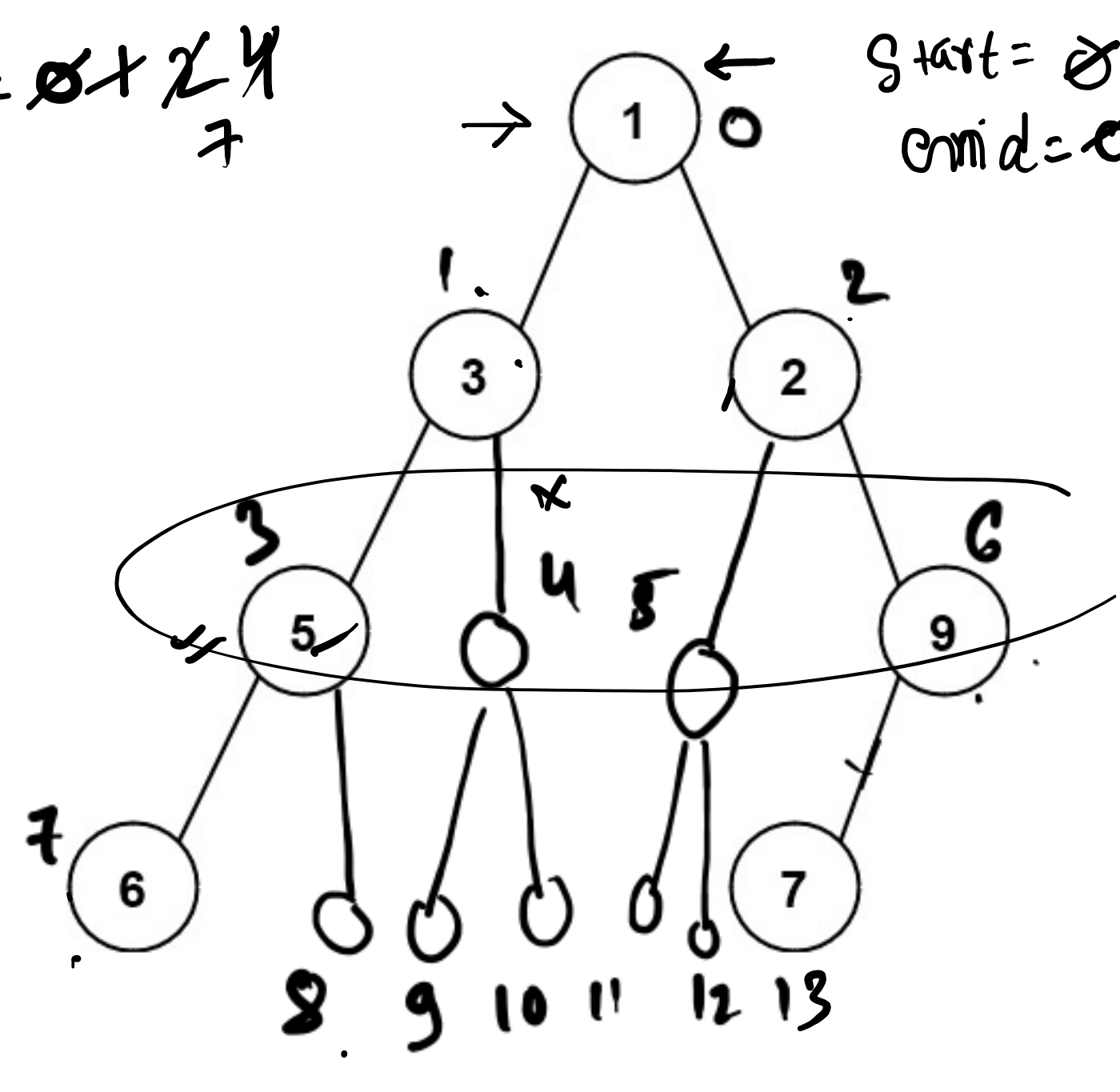
$(13 - 7 + 1) = 7$ ✓



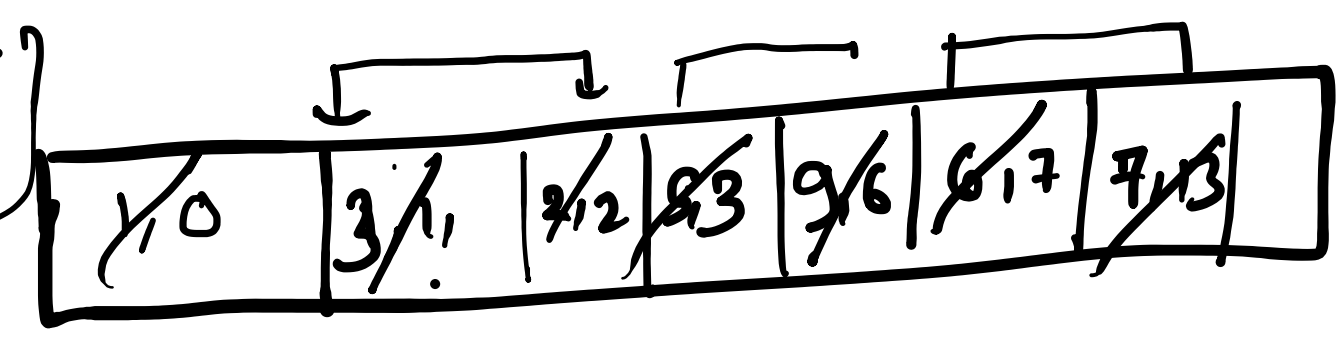
$6 - 3 + 1 = 4$



ans = 0 + 2 + 4 + 7

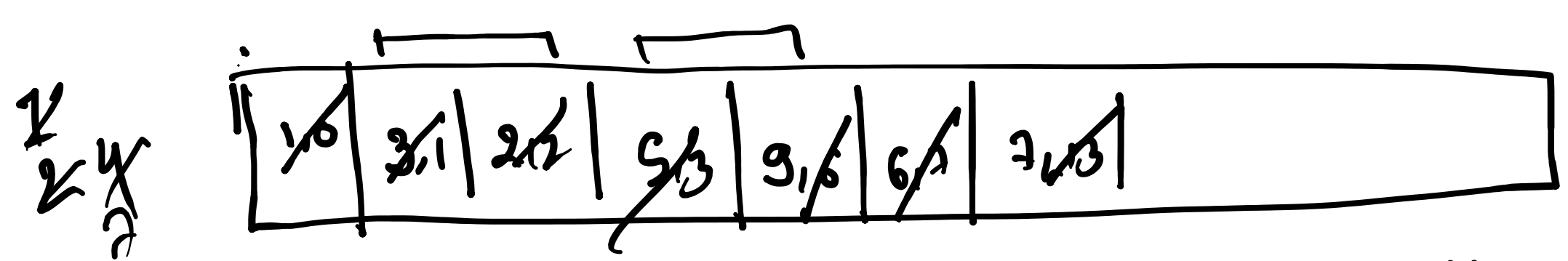


start = 0 + 1 2 7
end = 0 2 8 13



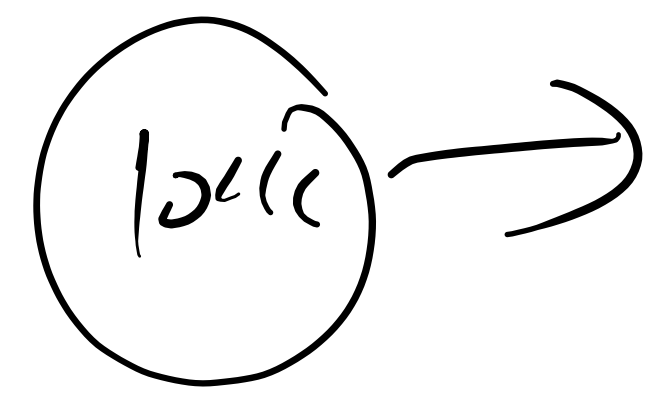
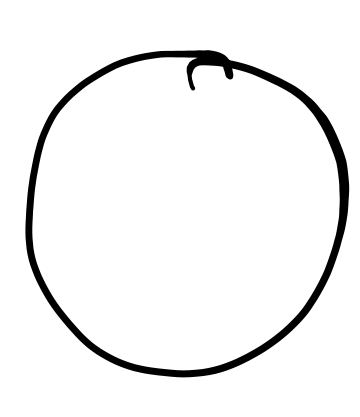
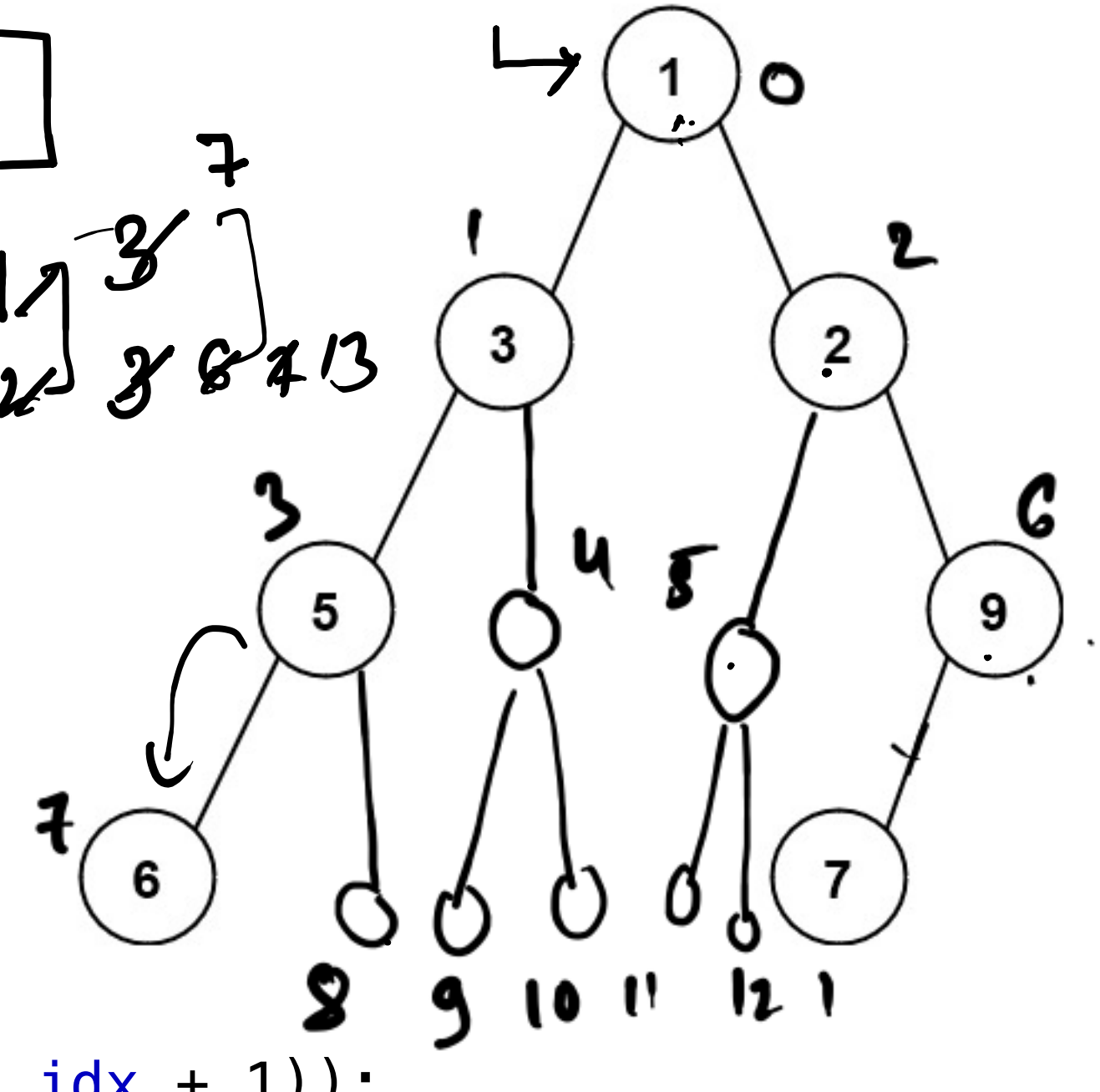
0 x 2 + 1 = 1
0 x 2 + 2 = 2
1 x 2 + 1 = 3
2 x 2 + 2 = 6

class Pair {
 TreeNode node;
 int idx;
}



```
public int widthOfBinaryTree(TreeNode root) {  
    Queue<Pair> q = new LinkedList<>();  
    int width = 0;  
    q.add(new Pair(root, 0));  
    while (!q.isEmpty()) {  
        Pair start = q.peek();  
        int s = q.size();  
        Pair end = null;  
        for (int i = 0; i < s; i++) {  
            end = q.poll();  
            if (end.node.left != null) {  
                q.add(new Pair(end.node.left, 2 * end.idx + 1));  
            }  
            if (end.node.right != null) {  
                q.add(new Pair(end.node.right, 2 * end.idx + 2));  
            }  
        }  
        width = Math.max(width, end.idx - start.idx + 1);  
    }  
    return width;  
}
```

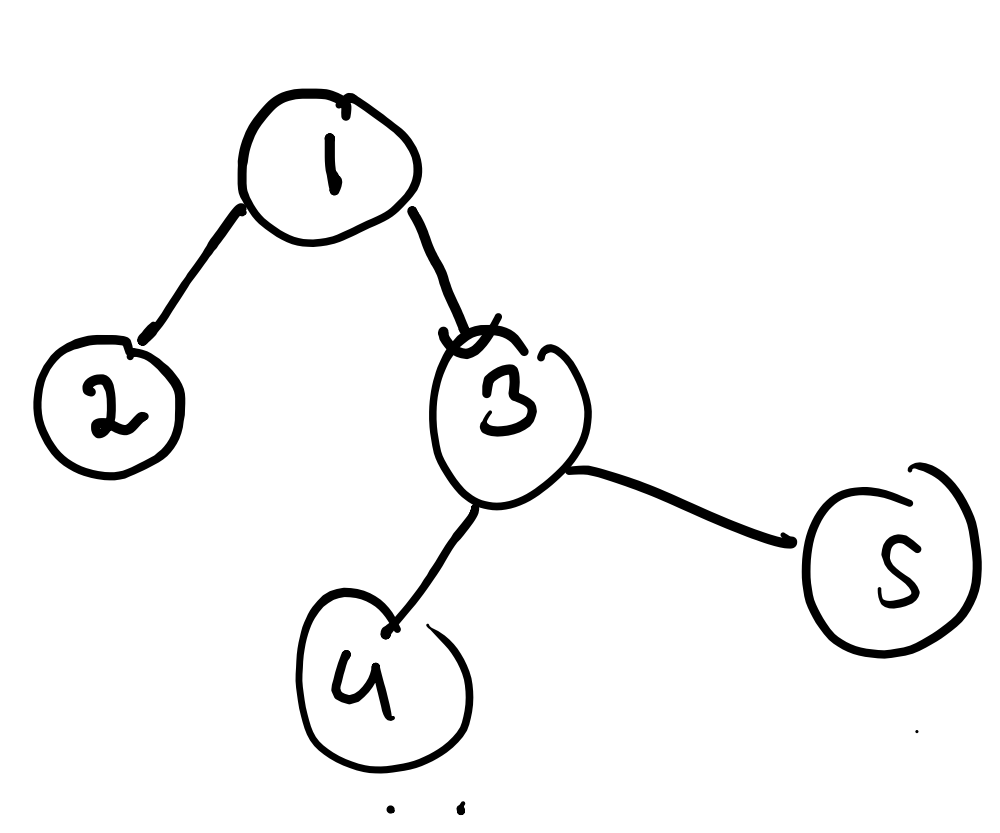
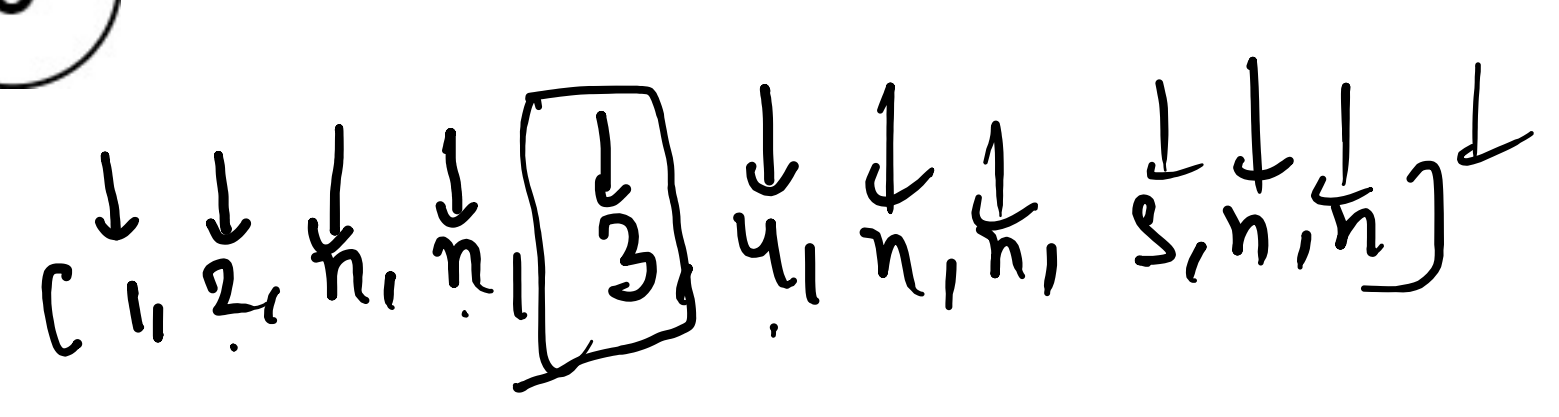
start = 0 1 2 7
end = 0 2 8 13



1, 2, null, null, 3, 4, null, null, 5, null, null

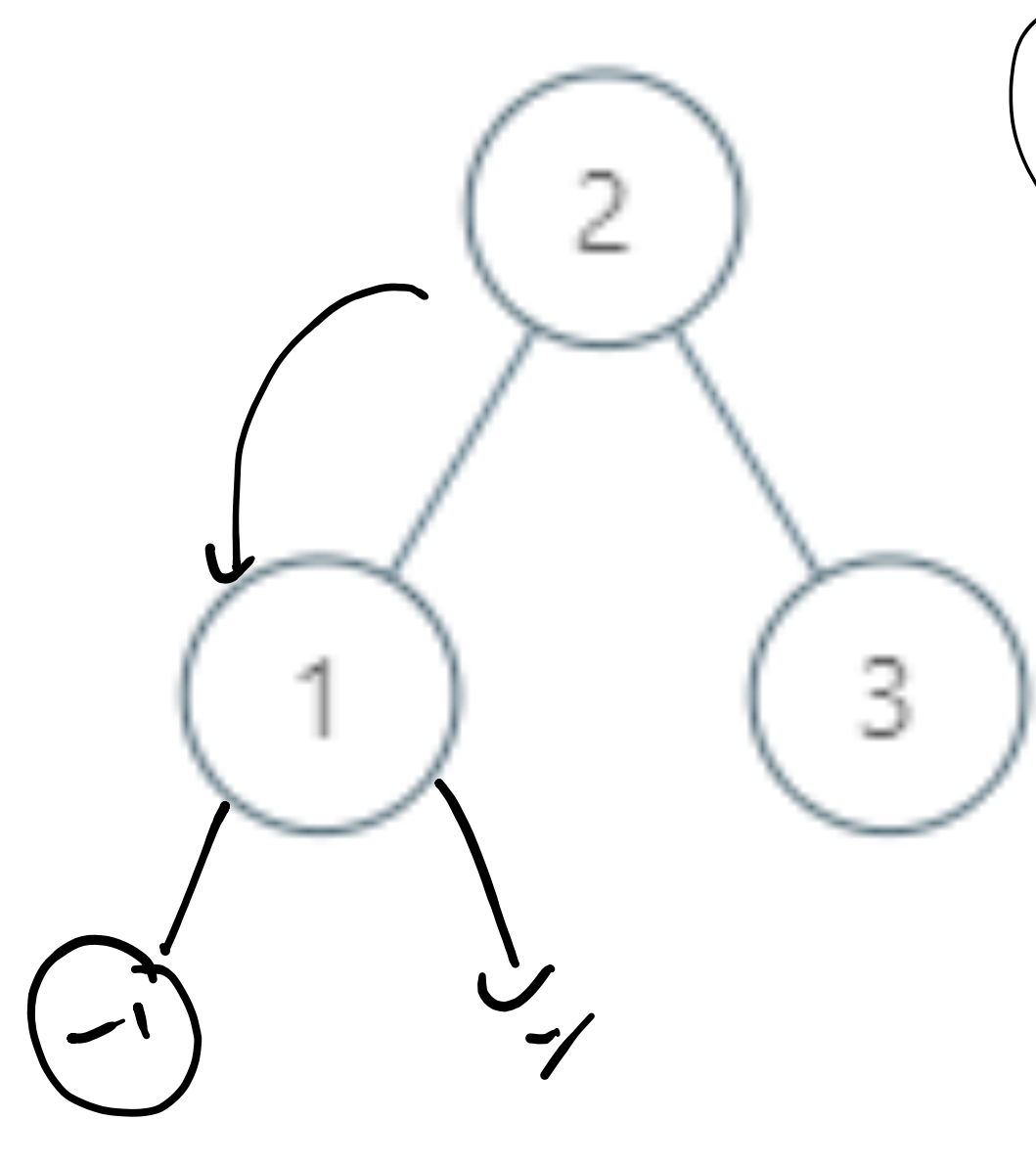
[1, 2, "null", null, 3, 4, ..., null]

1, 2, 3



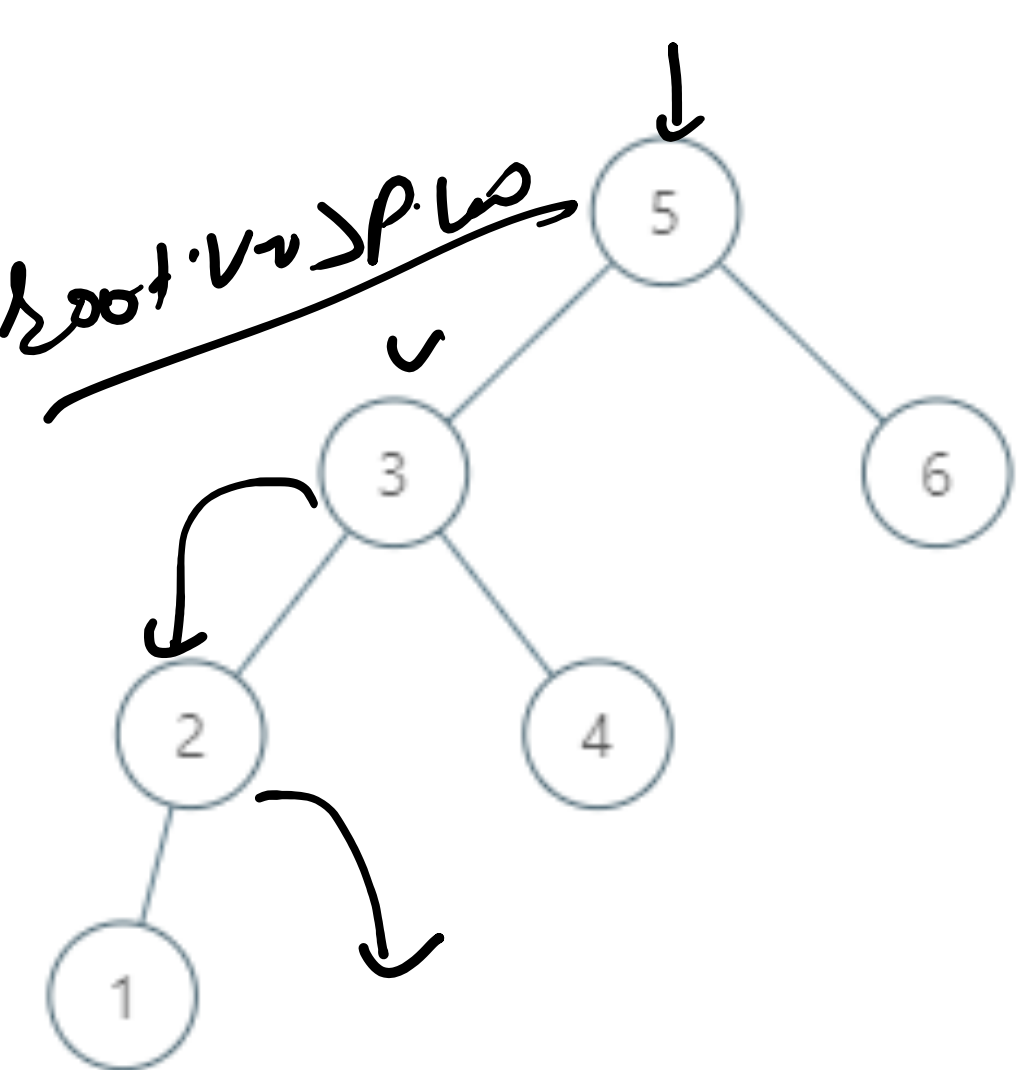
p.val < root.val

sum -> root



sum > root.val && root.val > p.val

sum = 5
3



1 2 3 4 5 6