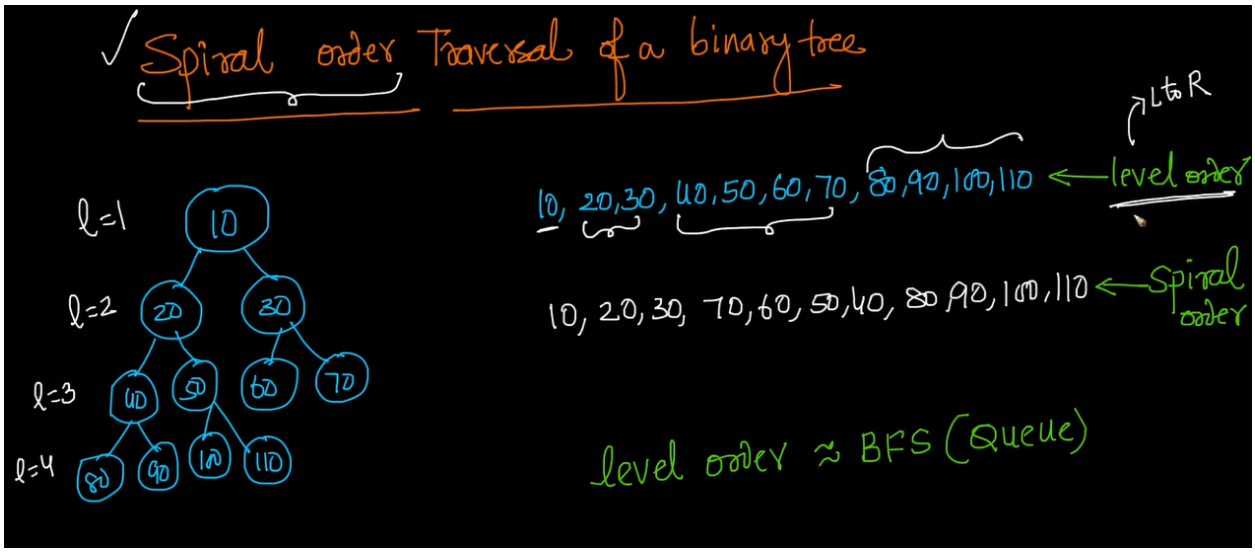
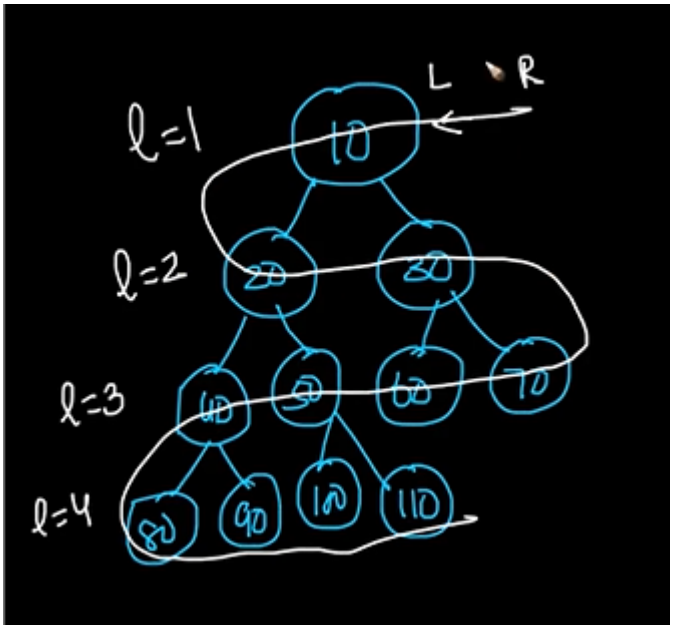
103. Binary Tree Zigzag Level Order Traversal

Problem Statement

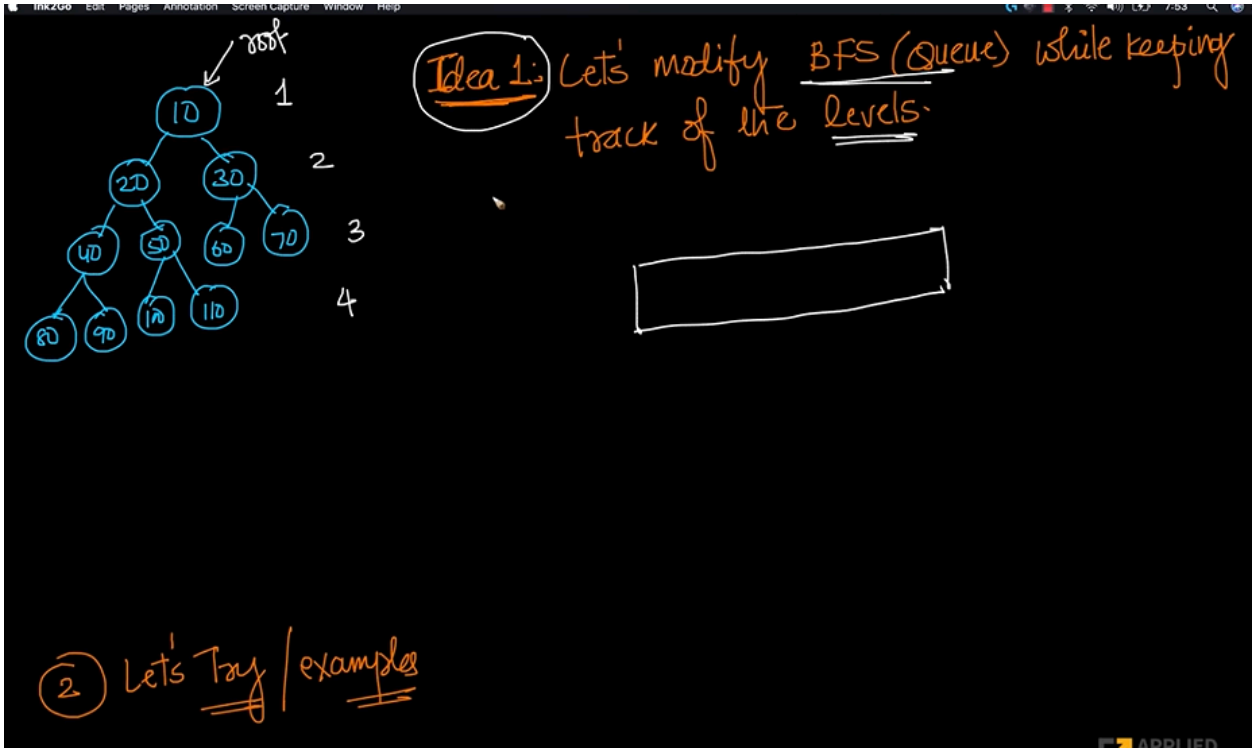
1. Print the tree in the spiral level order traversal



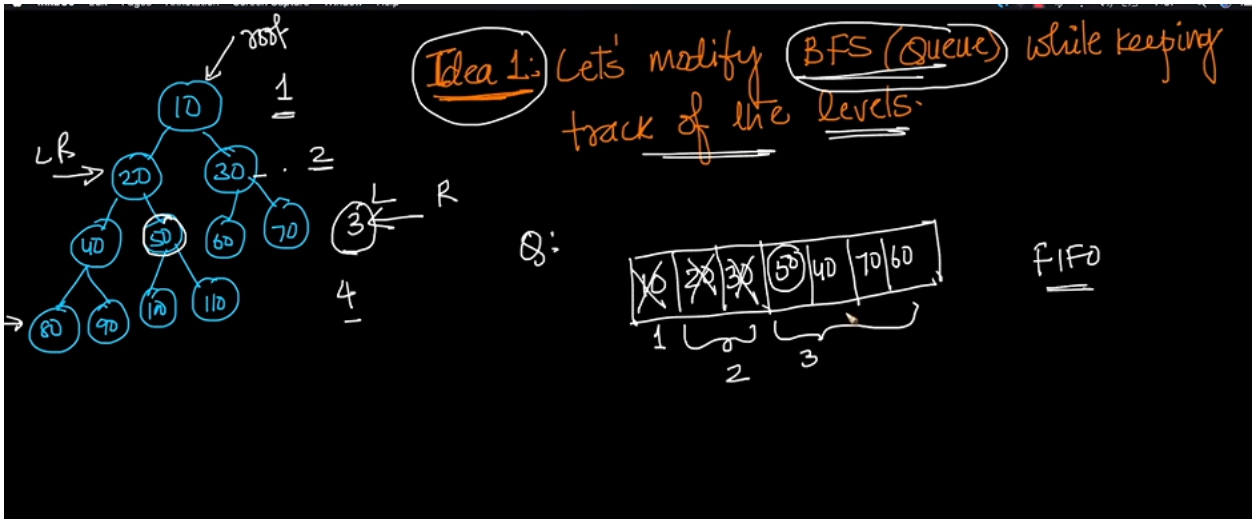
Approach:



1. So we have to traverse in this way
2. Here for every problem solving …we'll map a solution of a similar problem which we have solved earlier..
3. Here we have solved level order traversal..which used BFS(queue) to solve
4. So here..is there any way we can tweak the BFS and solve this problem?

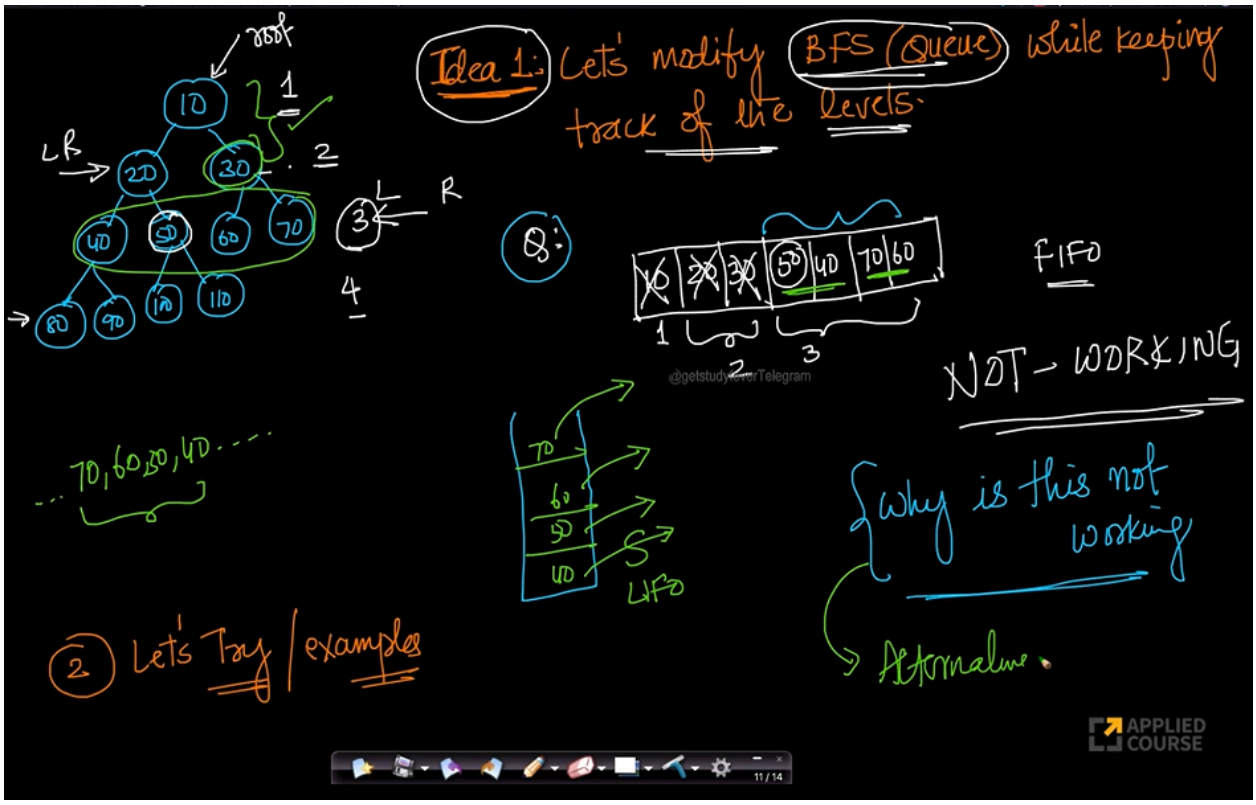
5. 

6. Here we'll modify our BFS(deque) in such a way that… it keeps the track of the levels…and for odd level we insert right child first and then left child



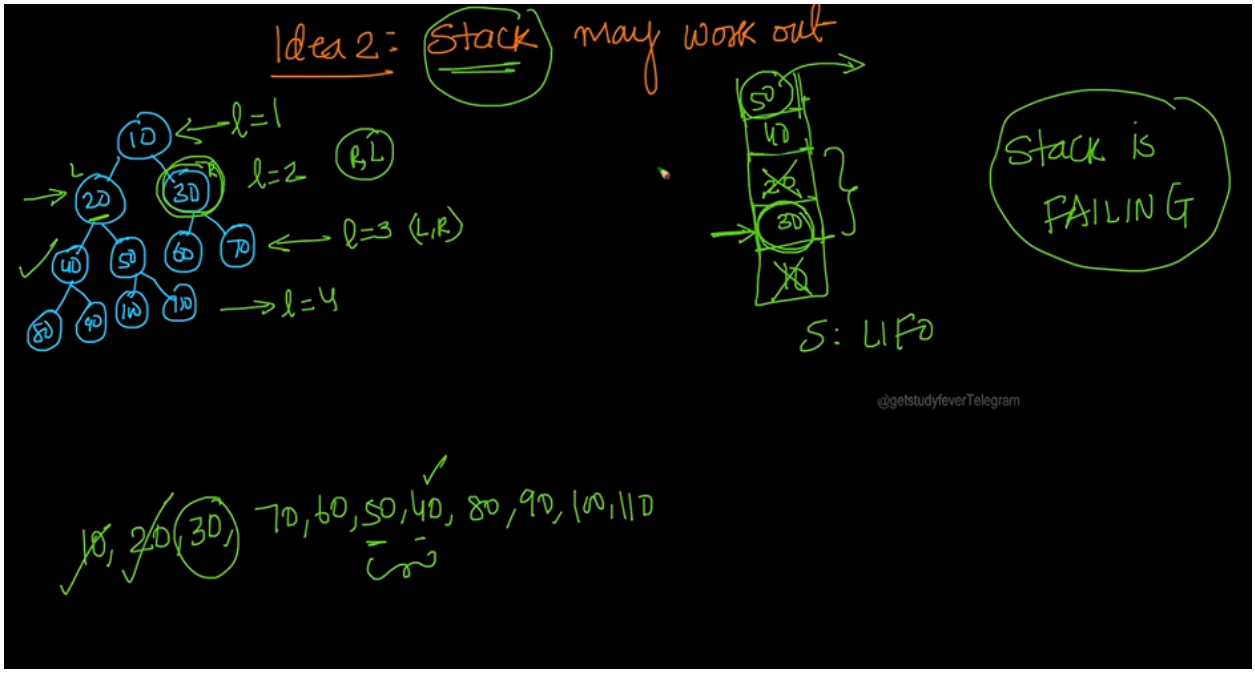
7. **But this method is not working..we are not getting the correct order.**
8. So from here…we'll move to other approach

9. What if used a stack to store the elements? Here we have added level 3 elements into the stack..and now we can retrieve them in the correct order

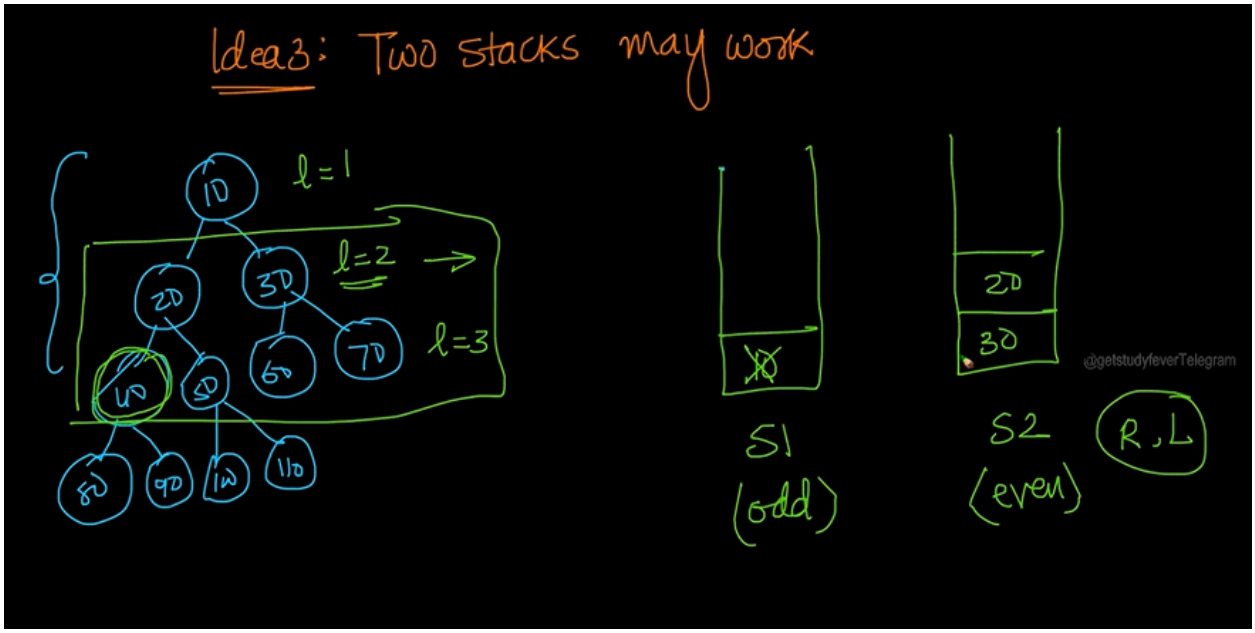


As the Queue soln didn't worked..we have to think for an alternative and stack is the alternative

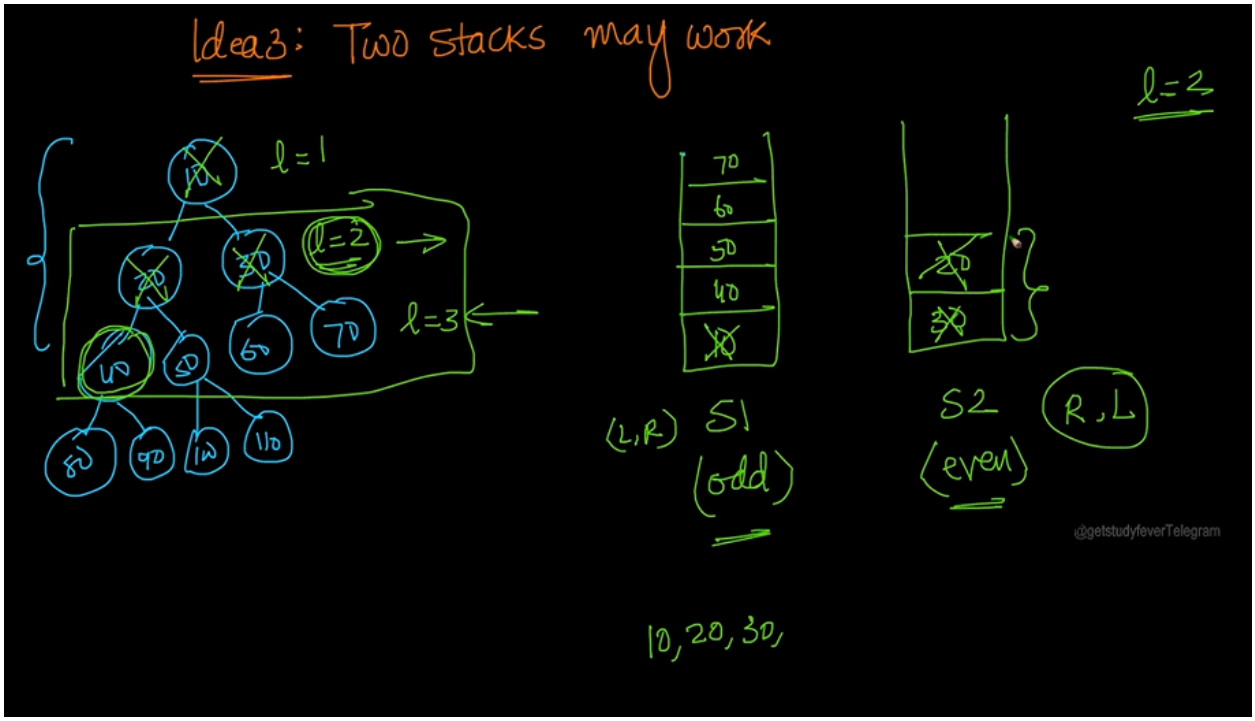
10. Idea 2 - Stack May work out



11. Here the problem is after popping 20 and adding its children…we have to pop() 30…but here stack does not allows us to pop() 30 as it is LIFO
12. **So our stack approach is also not working…lesson learned is levels are messing up**
13. But using two stacks may work…like..one stack stores elements of odd level and other stack stores elements of even level

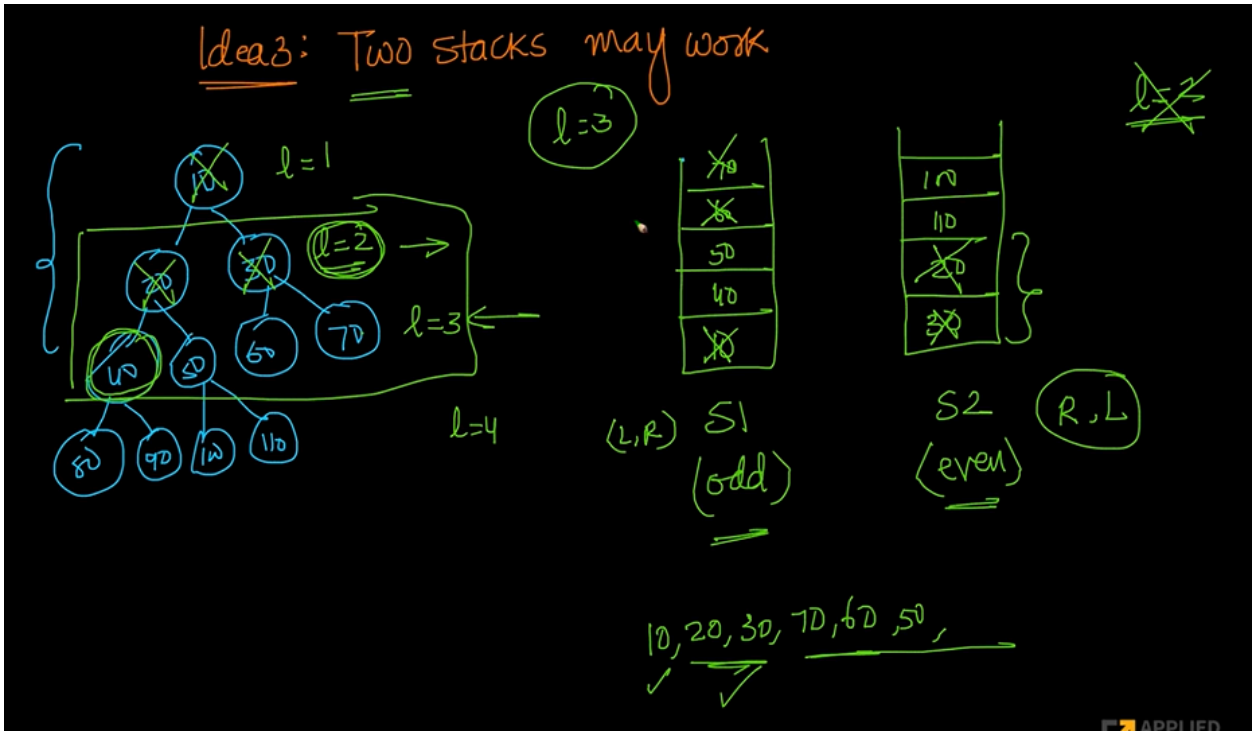


14. Now we'll pop from our S2

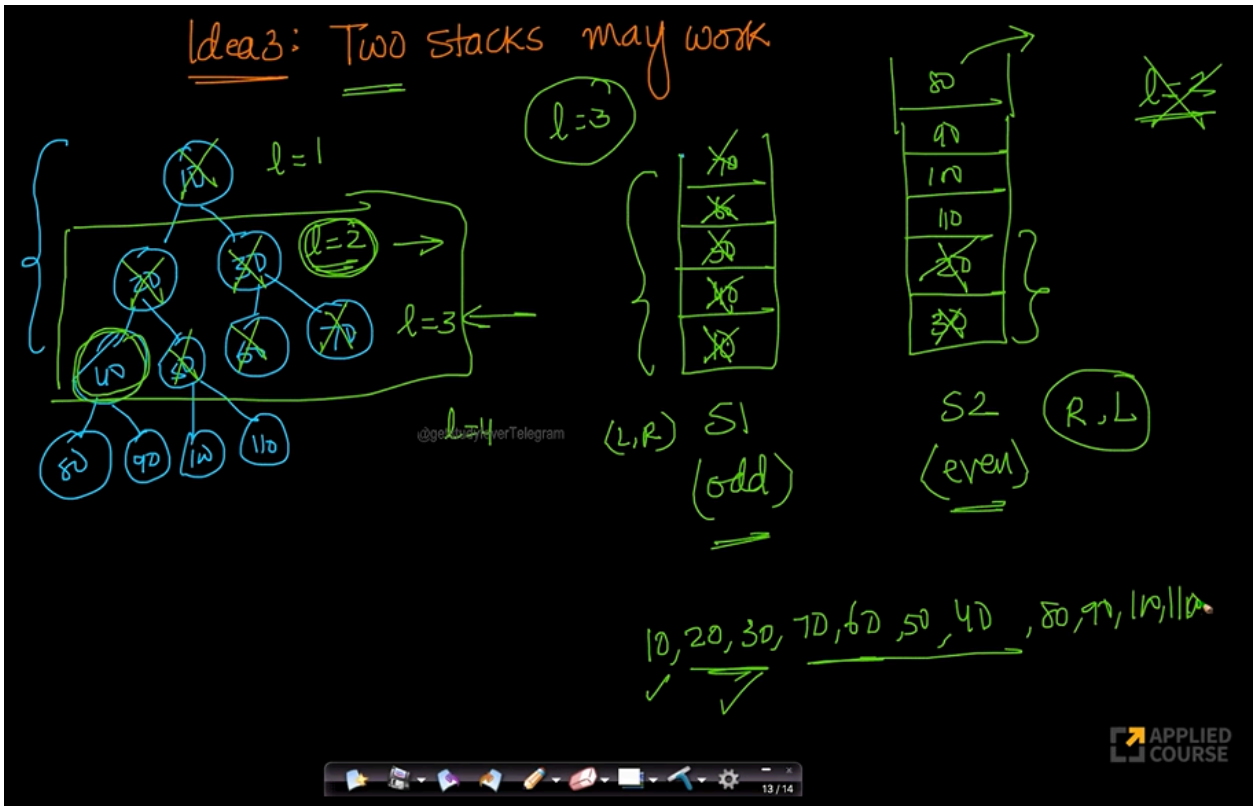


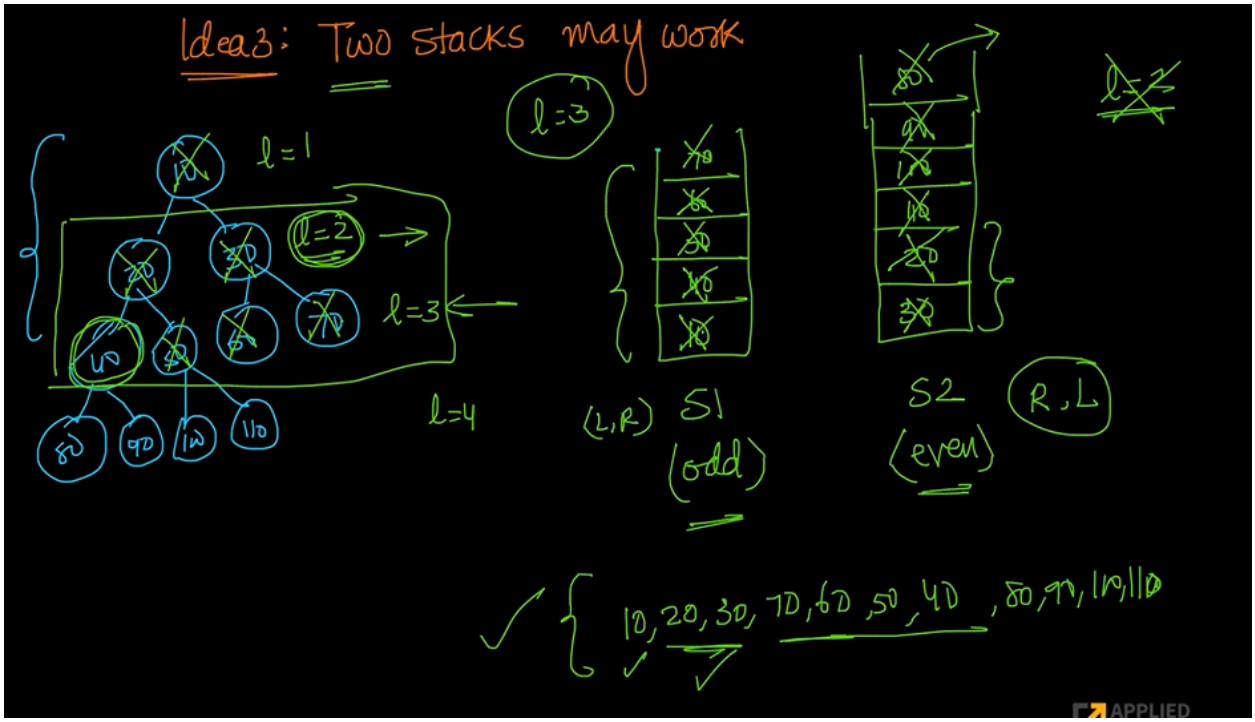
here we have to completely empty the S2 before moving to S1 …and vice versa

15. And after completing level 2 we'll move to level 3



16. Finally



17. 

Python code:

1.
```
def printSpiralRecursive(root):
    stack1=Stack()
    stack2=Stack()
    stack1.push(root)

    while (stack1.isEmpty() is not True) or (stack2.isEmpty() is not True):
        while(stack1.isEmpty() is not True):
            temp=stack1.pop()
            print(temp.info)
            if(temp.right is not None):
                stack2.push(temp.right)
            if(temp.left is not None):
                stack2.push(temp.left)
        while(stack2.isEmpty() is not True):
            temp=stack2.pop()
            print(temp.info)
            if(temp.left is not None):
                stack1.push(temp.left)
            if(temp.right is not None):
                stack1.push(temp.right)
```