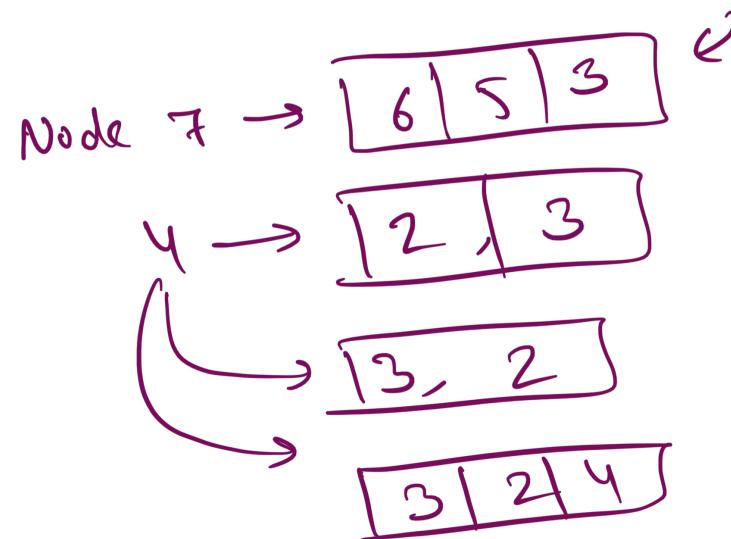
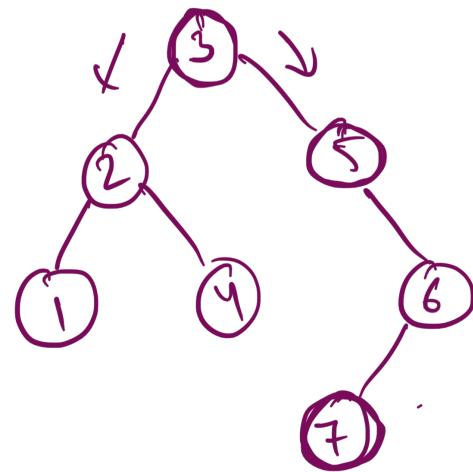


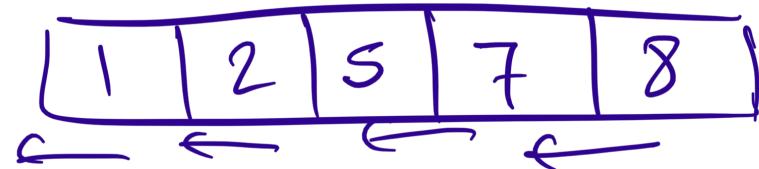
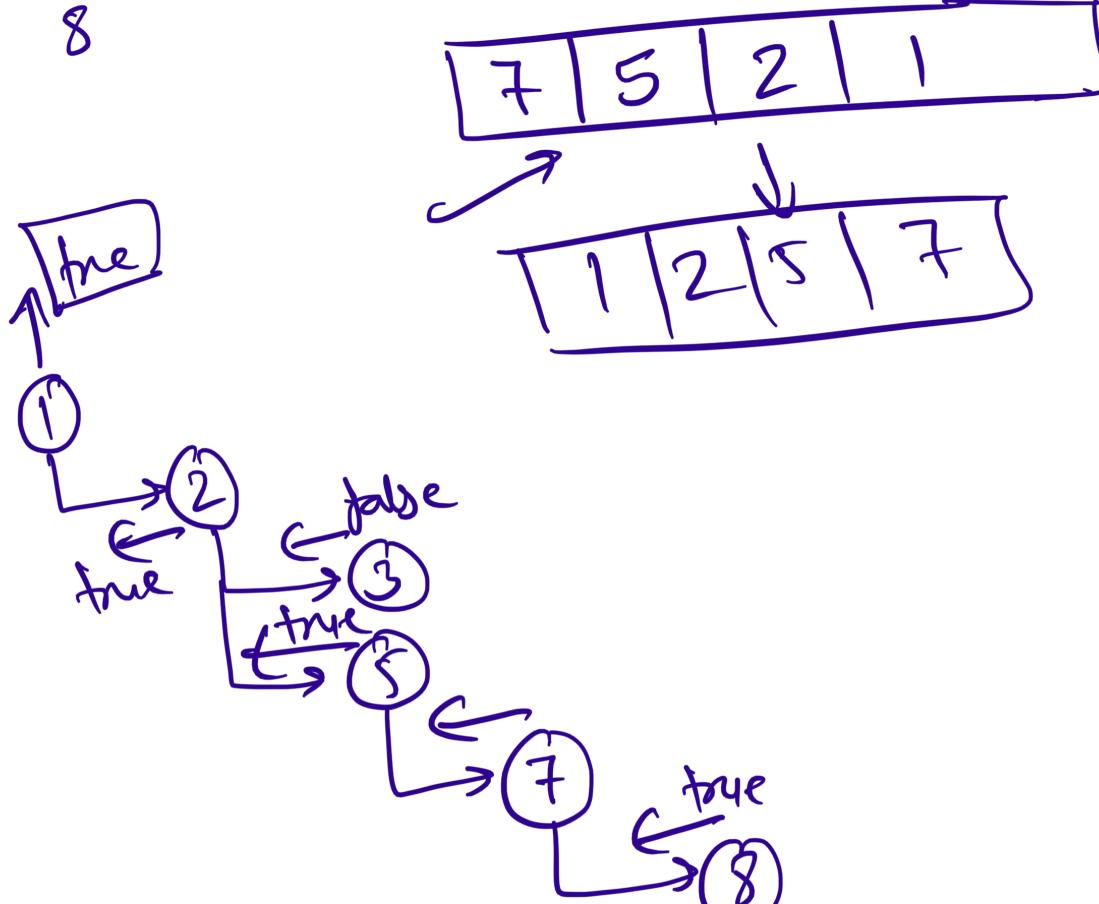
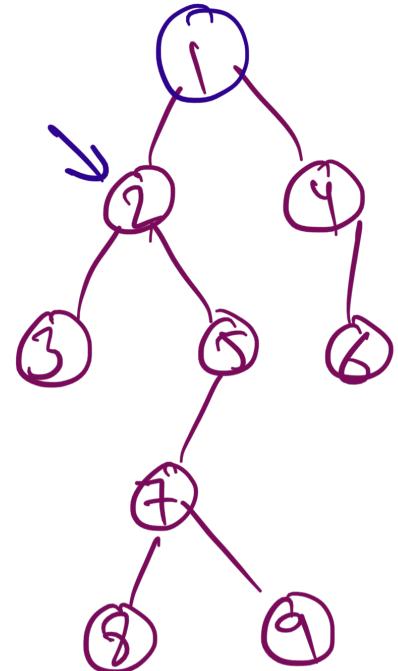
# Binary Tree - III



## Print all the ancestors of a Given Node in a Binary Tree

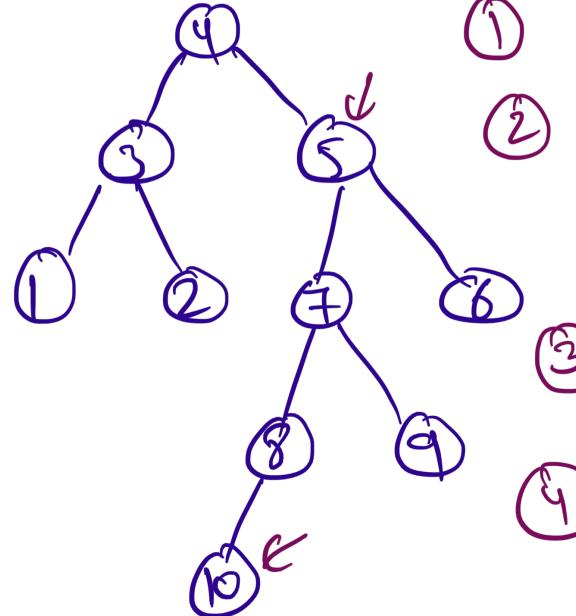


```
isPresent ( Node root, int key) {  
    if ( root.data == key ) return true;  
    if ( isPresent (root.left, key) ||  
        isPresent (root.right, key) ) {  
        ans.add (root.data);  
        return true;  
    }  
    return false;  
}
```



## Lowest Common Ancestor of a Binary Tree

$(10, 6) \rightarrow 5$   
 $(2, 8) \rightarrow 4$   
 $(5, 10) \rightarrow 5$   
 $(10, 9) \rightarrow 7$



- ① root  $(4, 10) \rightarrow 4$
- ② if one from left and one from right  $\rightarrow$  root  
 $(1, 9) \rightarrow 4$
- if both from same side  
 $(1, 2) \rightarrow 3$
- if both not present both sides, return null;

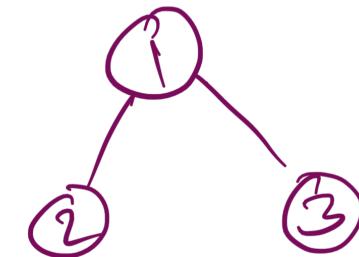
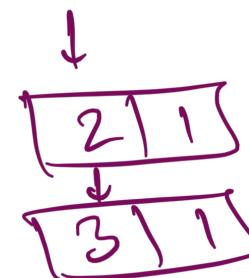
5  
10 → [10 | 8, | 7, | 5, | 4]

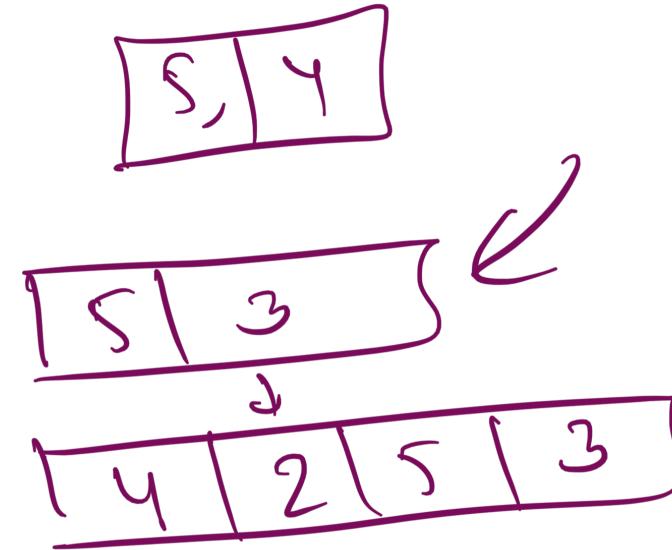
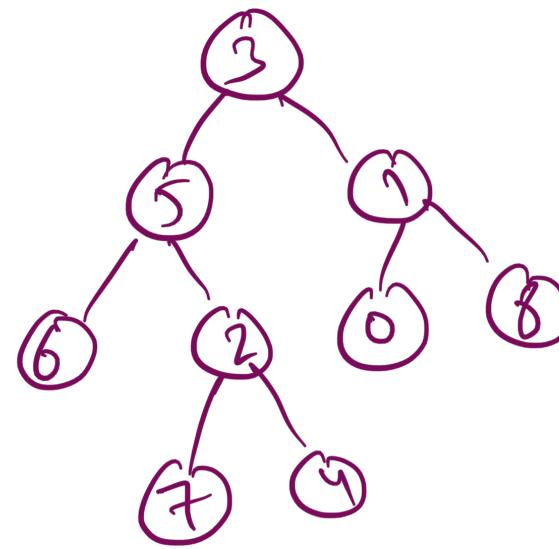
6 → [6 | 5, | 4]

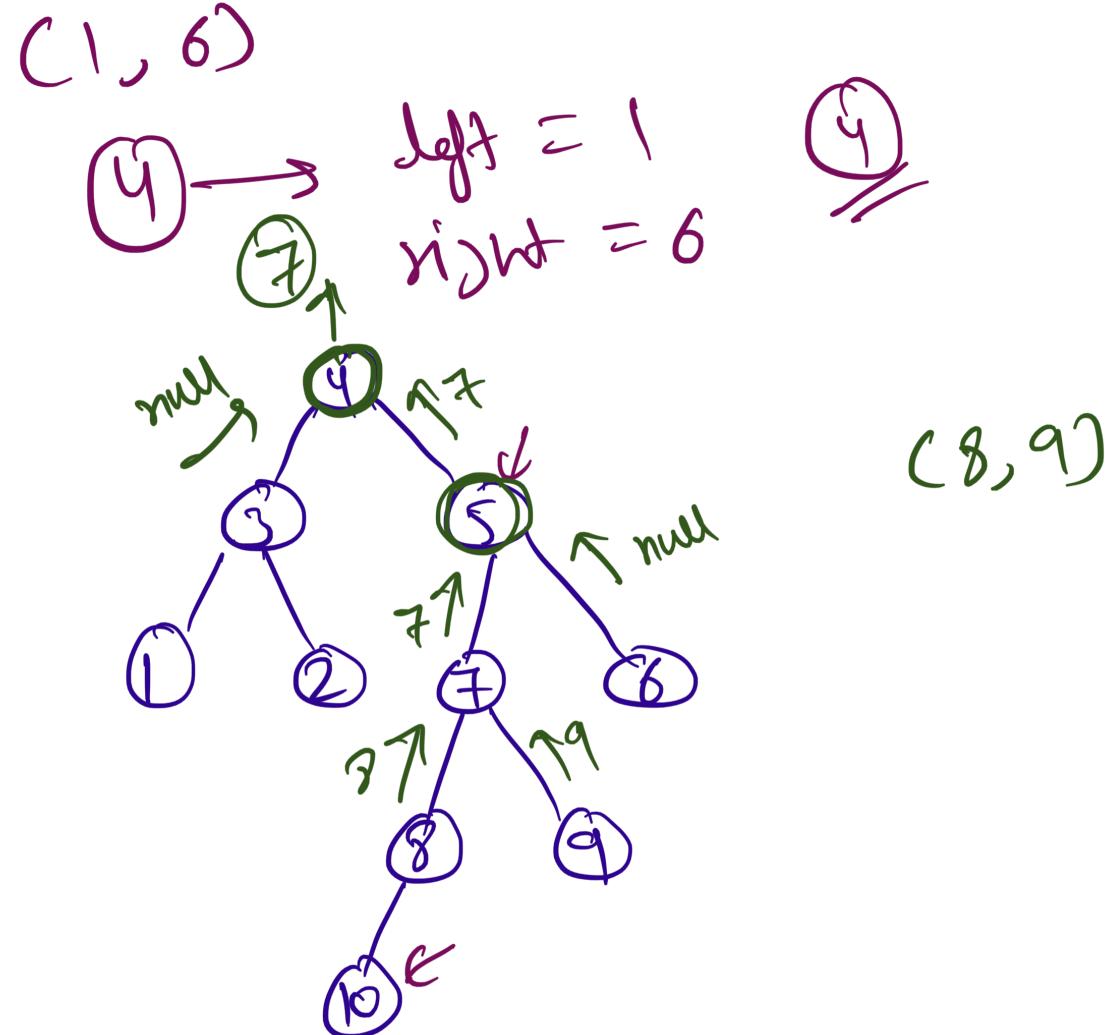
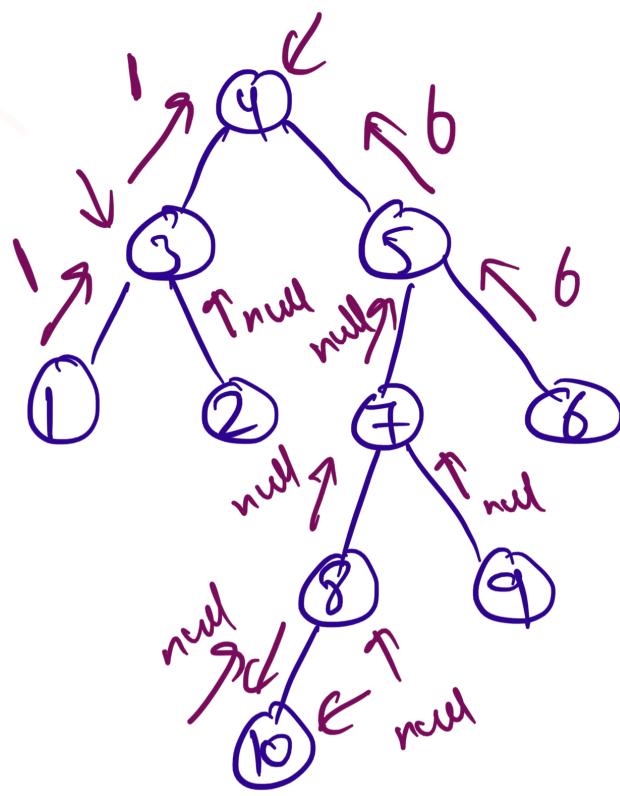
without space. O(1) space.

5 → 5, 4 ↴

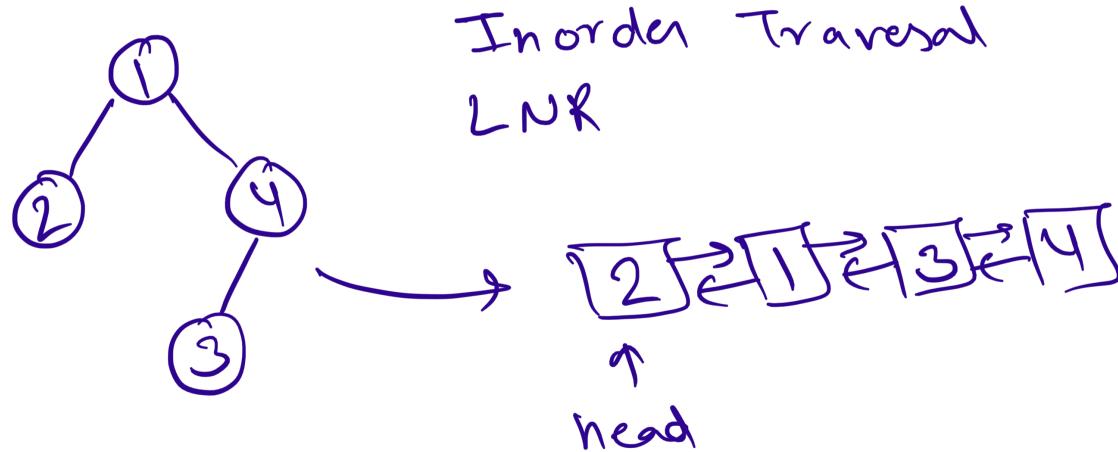
10 → 10, 8, 7, 5, 4 ↴







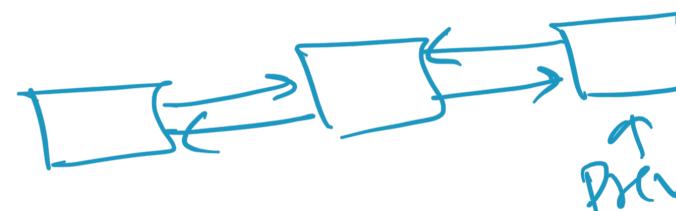
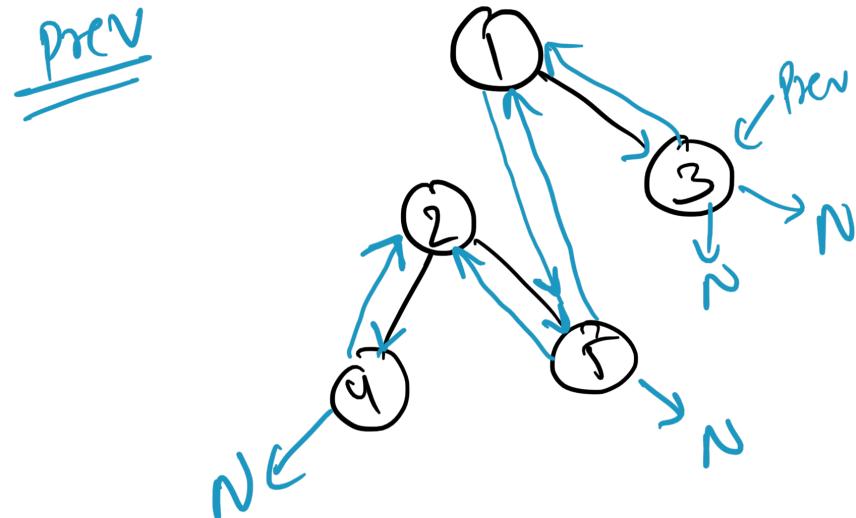
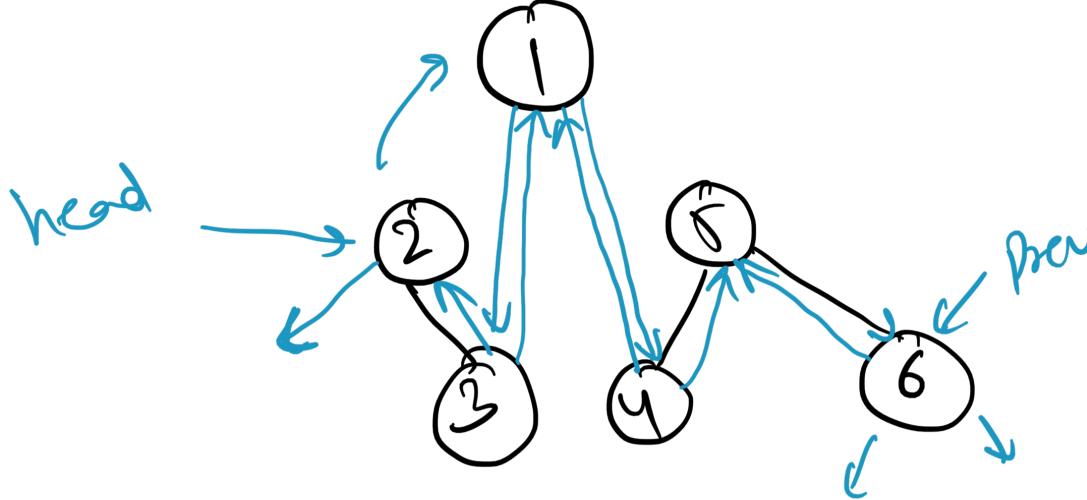
## Convert a Binary Tree to a Doubly Linked List



```
Node {  
    int data  
    Node left, right;  
}
```

↑  
Binary Tree

```
Node {  
    int data  
    Node prev, next;  
}  
↑  
Linked List
```

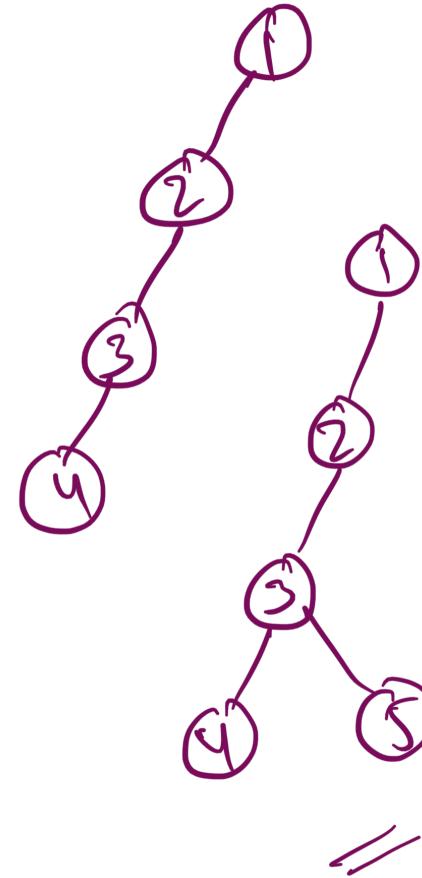
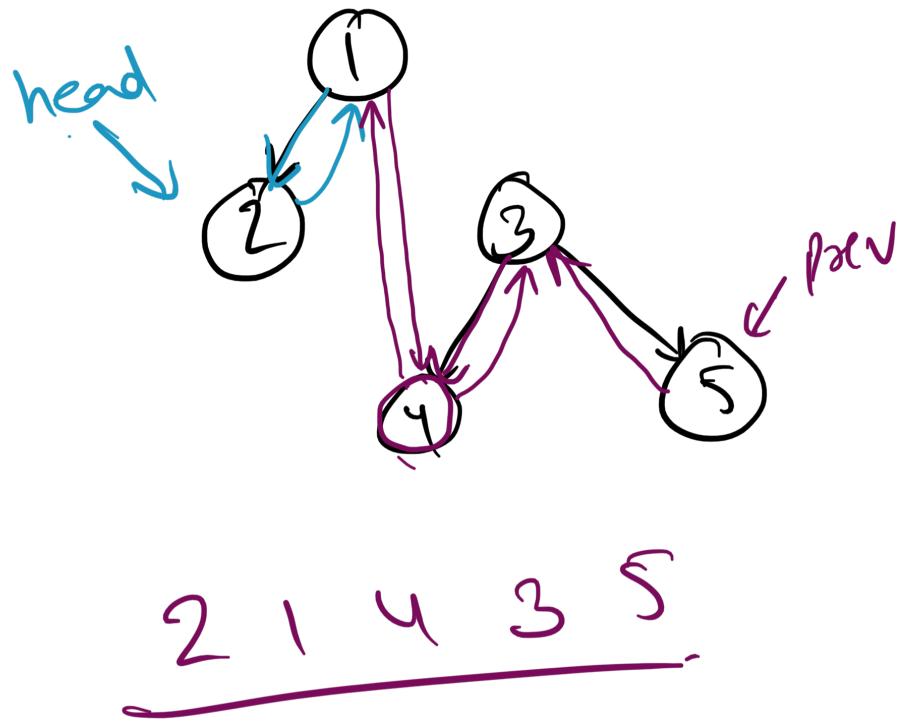


Inorder  
L N R

$4 \rightarrow 2 \rightarrow 5 \leftarrow 1 \leftarrow 3$

2 3 1 4 5 6 ↘

LNR



---

## Practice Problems

1. Burn a binary tree starting from a leaf node.
2. <https://www.geeksforgeeks.org/top-50-tree-coding-problems-for-interviews/>
3. <https://www.interviewbit.com/courses/programming/tree-data-structure>