Recursion Concepts & Qns ...

Motivation (भाषण) ...

"The EXPERT in anything today was once a BEGINNER..."

video 12

#codestorywithMIK

Facebook
Instagram → codestorywithMIK

(Twitter) → CSwithMIK

Company :- amazon

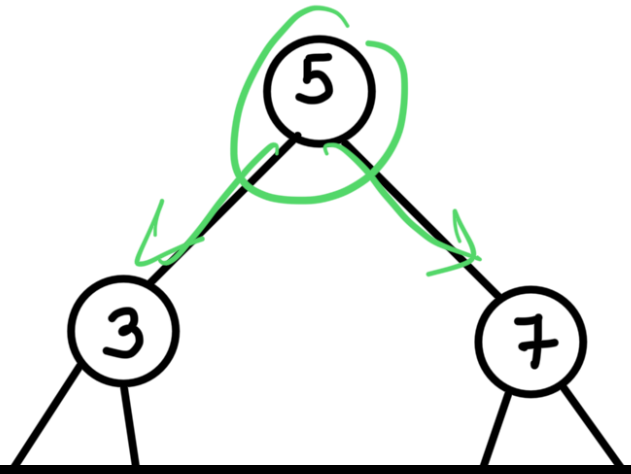
Flatten BST to sorted list

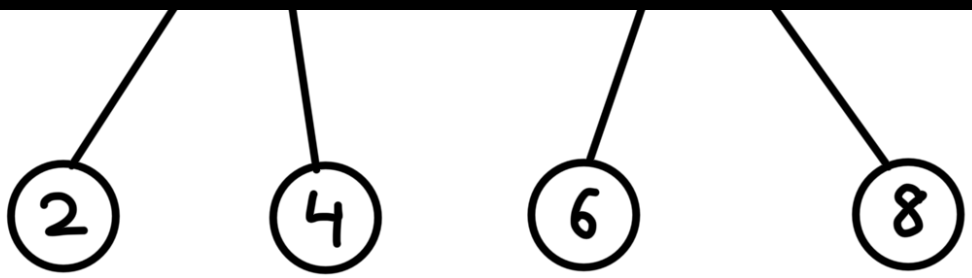
Medium Accuracy: 69.03% Submissions: 13K+ Points: 4

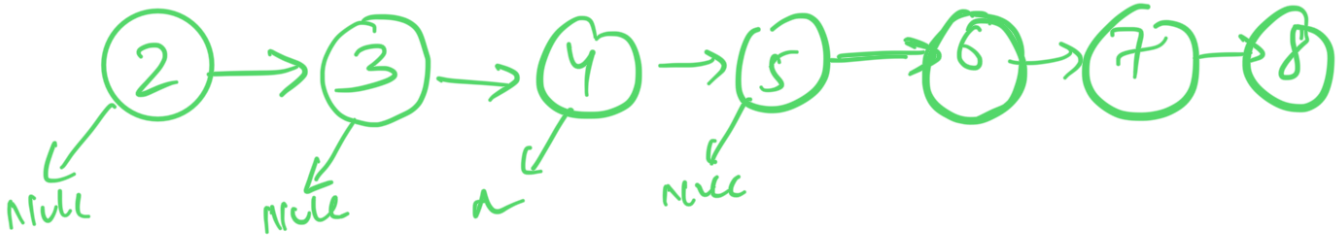
You are given a **Binary Search Tree (BST)** with **n** nodes, each node has a distinct value assigned to it. The goal is to flatten the tree such that, the **left child** of each element points to nothing **(NULL),** and the **right child** points to the next element in the sorted list of elements of the **BST** (look at the examples for clarity). You must accomplish this **without using any extra storage**, except for recursive calls, which are allowed.

S.C. (Auxi) = O(1)

Note: If your **BST** does have a **left child**, then the system will print a **-1** and will skip it, resulting in an **incorrect solution**.



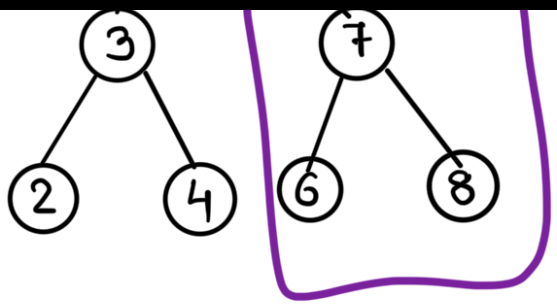


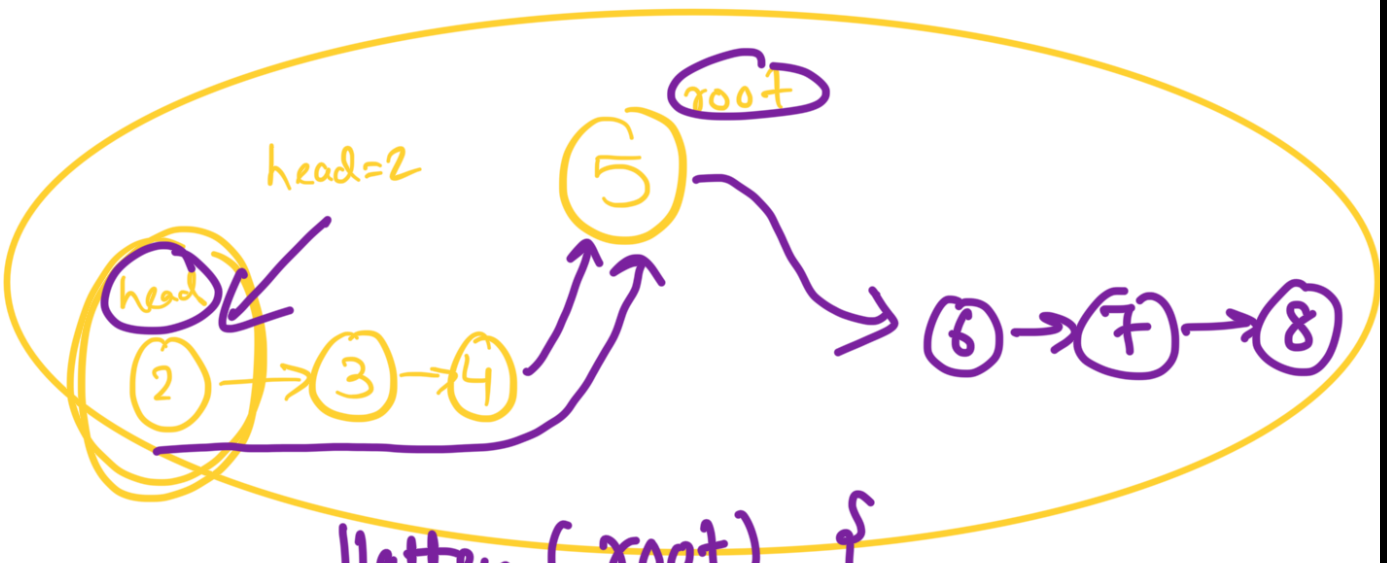


*Category "Flattening"

Best for visualising

"TRUST in Recursion" ...



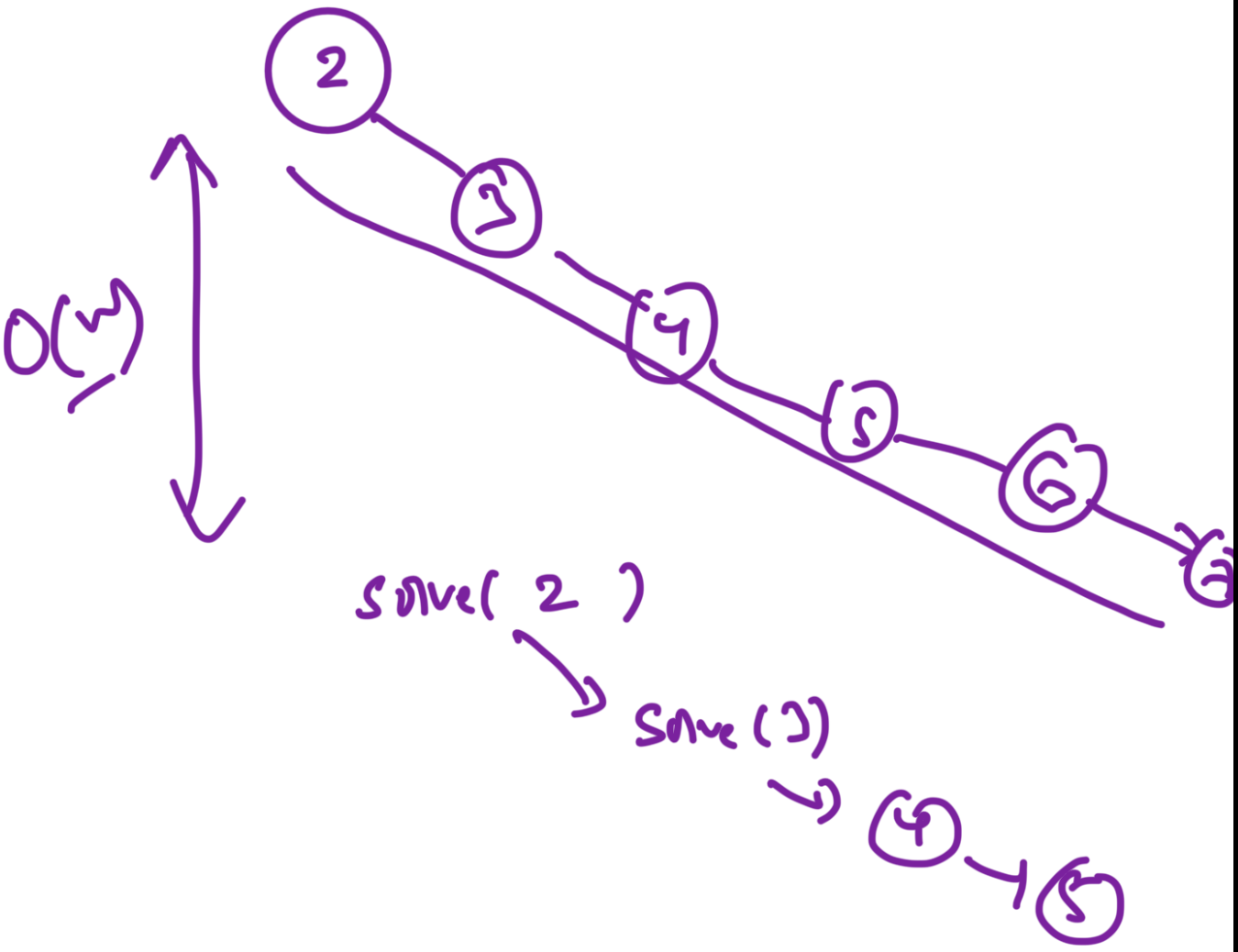


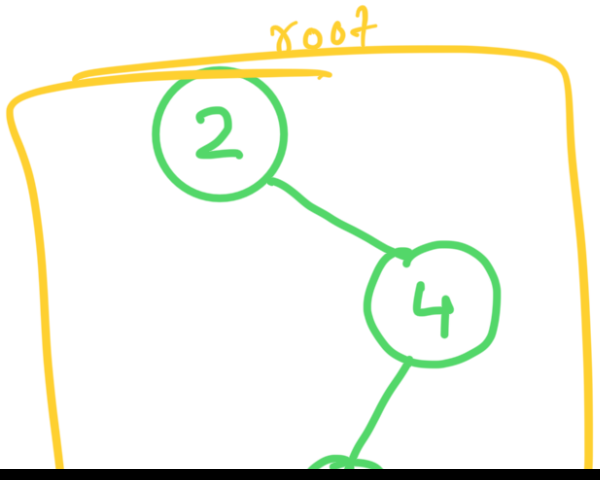

```
flatten (root) {
    if(root == NULL) return NULL;
    head = flatten (root->left) ; //Trust
    root->left = NULL ;
    temp = head ;
    while (temp && temp->next) {
        temp = temp->next ;
    }
    temp->next = root;
    root->right = flatten (root->right) ; //Trust
    return head;
}
```

T.C = O(n)

S.C = Auxil. → O(1)

Recu. stack space → O(n)





head

2 root

```
if (head=NULL) {
    head = root
} else {
```

H.W.

flatten

sorted linked list

Nested list → pushing.