**[Flatten Binary Tree to Linked List](https://leetcode.com/problems/flatten-binary-tree-to-linked-list/)**

**public** **class** FlattenBTToLinkedList {

**public** **static** TreeNode *prev* = **null**;

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

TreeNode root = **new** TreeNode(10);

root.left = **new** TreeNode(5);

root.left.left = **new** TreeNode(3);

root.left.right = **new** TreeNode(7);

root.right = **new** TreeNode(15);

root.right.right = **new** TreeNode(18);

*flatten*(root);

System.***out***.println(root.right.val);

}

**public** **static** **void** flatten(TreeNode root) {

**if**(root == **null**) {

**return**;

}

*dfs*(root);

}

**public** **static** **void** dfs(TreeNode root) {

**if**(root == **null**) {

**return**;

}

*dfs*(root.right);

*dfs*(root.left);

root.right = *prev*;

root.left = **null**;

*prev* = root;

}

}

Time complexity : O(n) n is Number of nodes in given tree

Space Complexity : O(1) , constant space