#2: Binary Tre & Divide Conquer & DFS & BFS

Problem and Answer

Outline

1. Binary Tree DFS Traversal

- preorder / inorder / postorder

- Divide & Conquer

- Introduce DFS Template

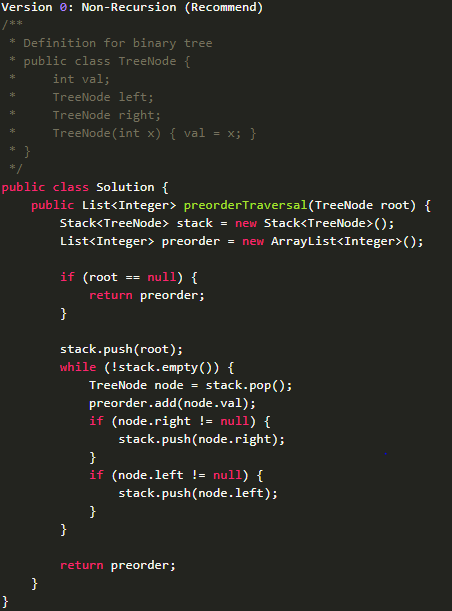
2. Binary Tree BFS Traversal

- Introduce BFS template

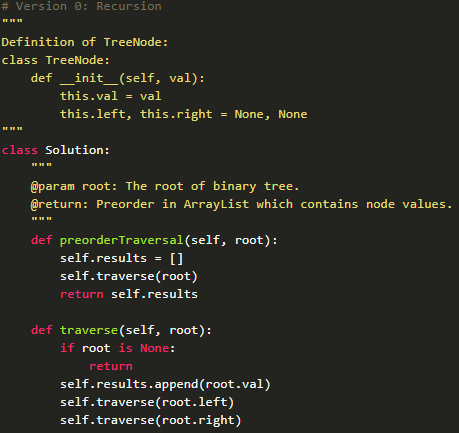
3. Binary Search Tree

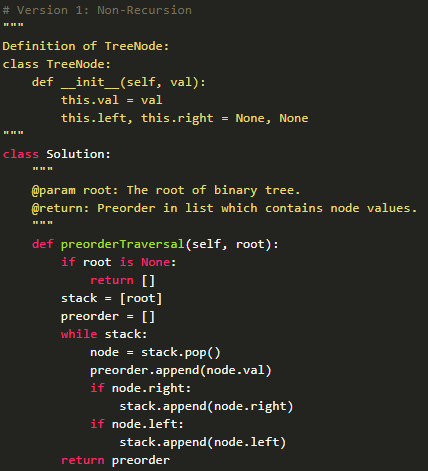
Binary Tree Preorder Traversal

Given a binary tree, return the preorder traversal of its nodes' values.



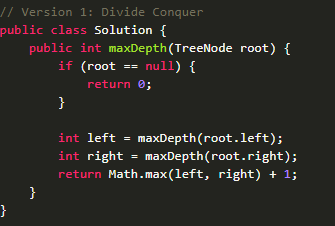


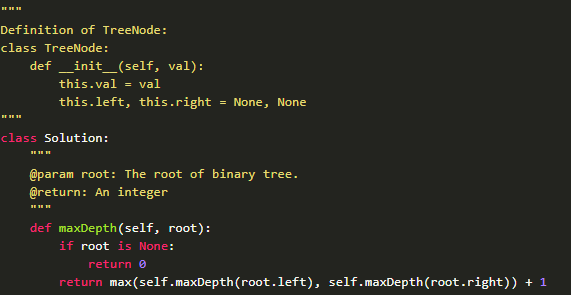




Maximum Depth of Binary Tree

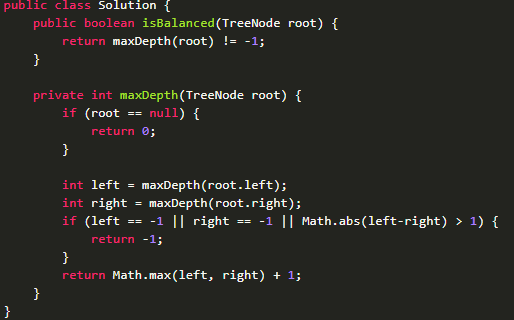
Given a binary tree, find its maximum depth. The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

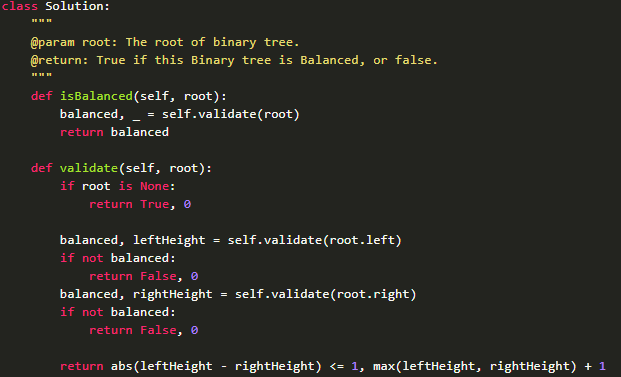




Balanced Binary Tree

Given a binary tree, determine if it is height-balanced. A height-balanced binary tree is defined as a binary tree in which the depth of the two subtrees of every node never differ by more than 1.

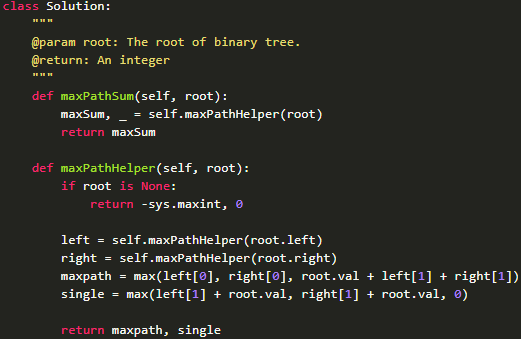




Binary Tree Maximum Path Sum

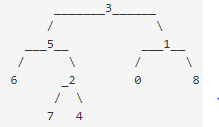
Given a binary tree, find the maximum path sum. The path may start and end at any node in the tree.



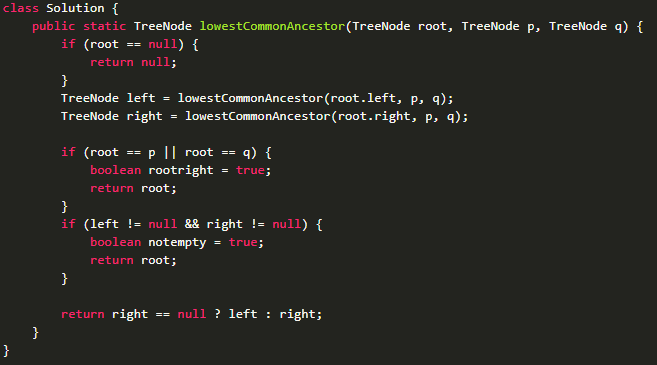


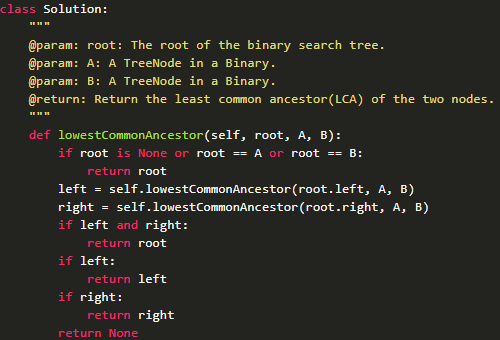
Lowest Common Ancestor of a Binary Tree

Given the root and two nodes in a Binary Tree. Find the lowest common ancestor(LCA) of the two nodes. The lowest common ancestor is the node with largest depth which is the ancestor of both nodes. Given the following binary tree: root = [3,5,1,6,2,0,8,null,null,7,4]



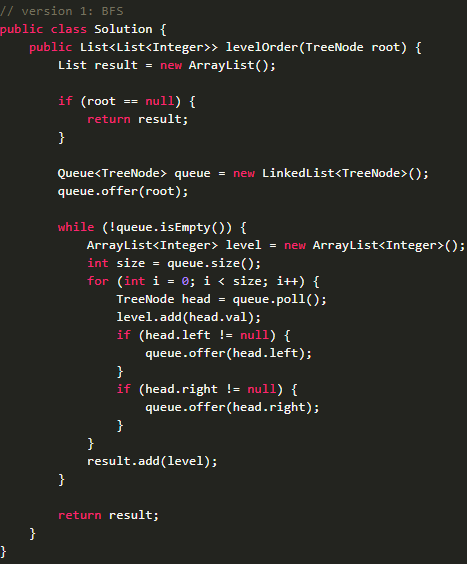
The LCA of of nodes 5 and 1 is 3. The LCA of nodes 5 and 4 is 5.

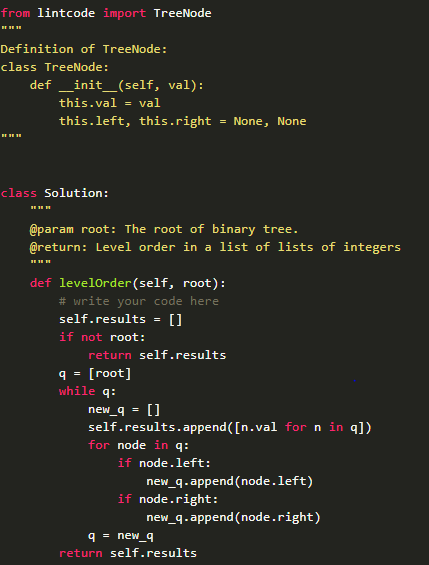




Binary Tree Level Order Traversal

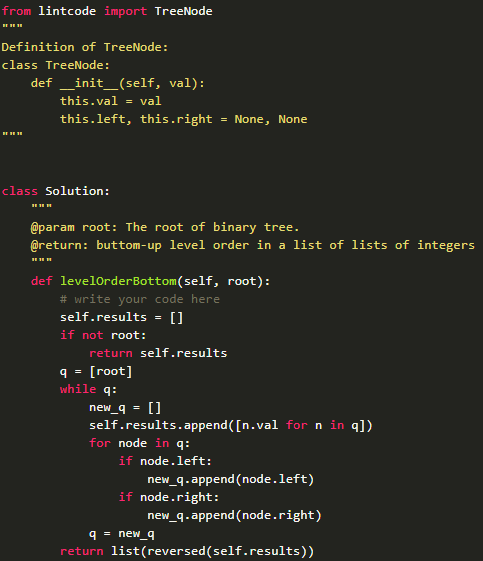
Given a binary tree, return level order traversal of its nodes' values. (ie, from left to right, level by level).





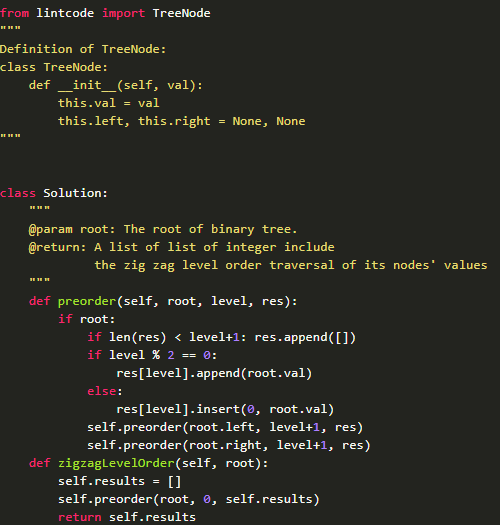
Binary Tree Level Order Traversal II

Given a binary tree, return the bottom-up level order traversal of its nodes' values. (ie, from left to right, level by level from leaf to root).



Binary Tree Zigzag Level Order Traversal

Given a binary tree, return the zigzag level order traversal of its nodes' values. (ie, from left to right, then right to left for the next level and alternate between).



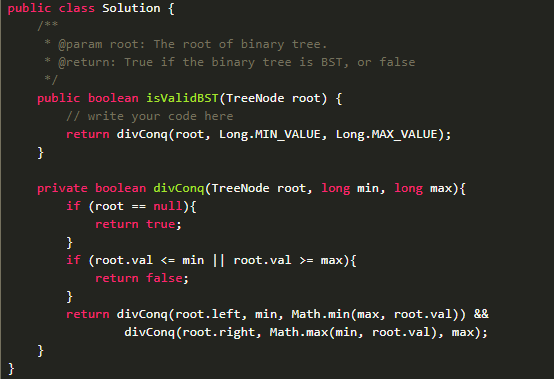
Validate Binary Search Tree

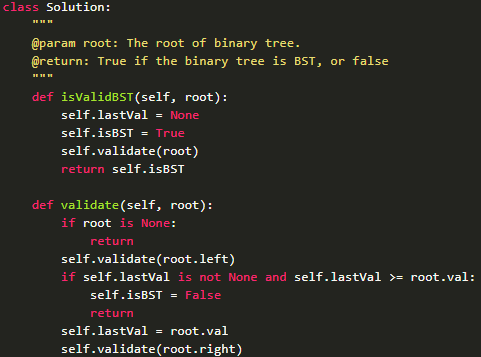
给定一个二叉树，判断它是否是合法的二叉查找树(BST)

一棵BST定义为：

节点的左子树中的值要严格小于该节点的值; 节点的右子树中的值要严格大于该节点的值

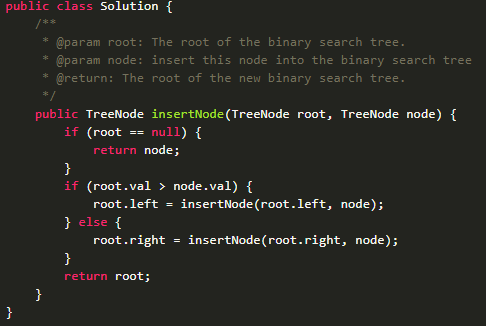
左右子树也必须是二叉查找树; 一个节点的树也是二叉查找树

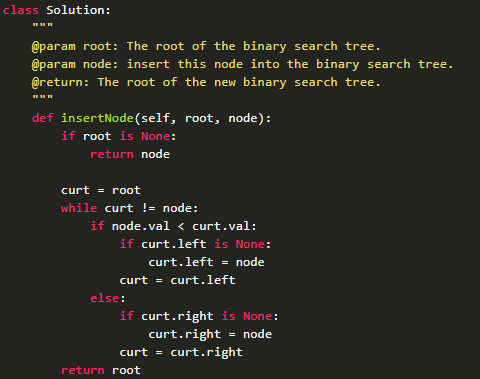




Insert a Node in Binary Search Tree

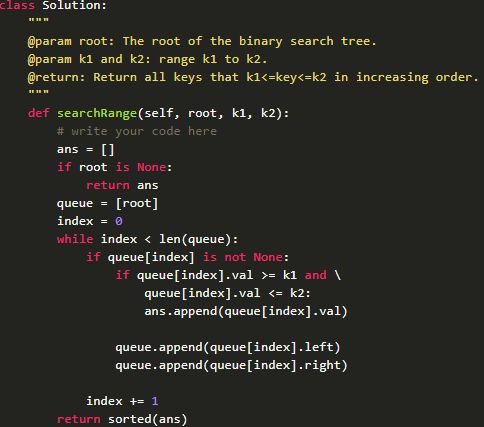
给定一棵二叉查找树和一个新的树节点，将节点插入到树中。你需要保证该树仍然是一棵二叉查找树。





Search Range in a Binary Search Tree

Given a binary search tree and a range [k1, k2], return all elements in the given range.



Implement iterator of Binary Search Tree

Design an iterator over a binary search tree with the following rules:

Elements are visited in ascending order (i.e. an in-order traversal)

next() and hasNext() queries run in O(1) time in average.

