Data Structures

BST Homework 1

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Problem #1: LeetCode 700 - Search in a Binary Search Tree

- You are given the root of a binary search tree (BST) and an integer val.
- Find **the node** in the BST that the node's value equals val and return the subtree rooted with that node.
- If such a node does not exist, return None.
- Utilize the same logic as our recursive search from the lecture, but come up with an **iterative** version of the code

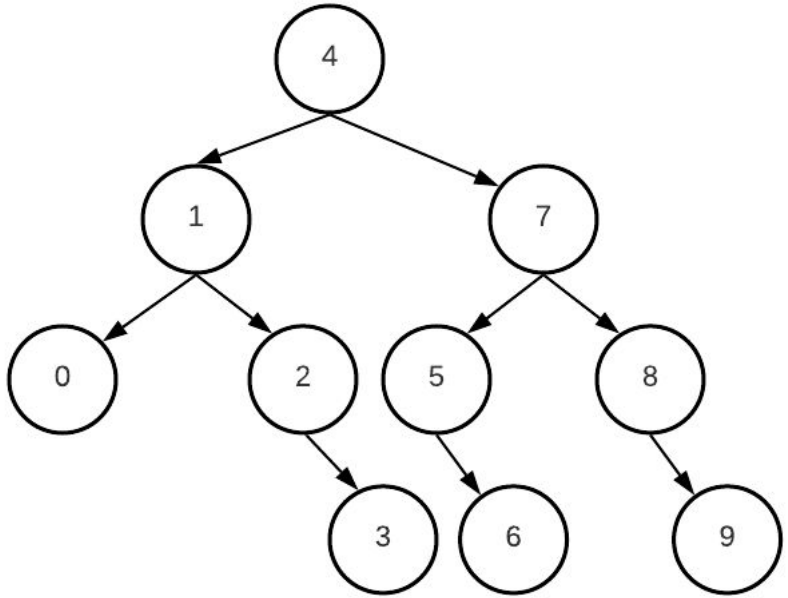
Problem #2: LeetCode 98 - Validate Binary Search Tree

- Given the root of a binary tree, determine if it is a valid binary search tree (BST).
  - Consider this: A binary tree with **duplicate** value is not BST
- Describe 2 ***fundamentally different*** approaches to check that
  - Important! Don't simply code recursive and iterative versions of code following identical logic.
  - I want two distinct implementations

Problem #3: LeetCode 08 - Convert Sorted Array to BST

- *Background: Sometimes we have a very unbalanced BST, and we want to convert it to a Balanced BST. We use the inorder traversal to rebuild it*
- Given an integer array nums where the elements are sorted in ascending order, convert it to a height-balanced binary search tree.
- A height-balanced binary tree is a binary tree in which the depth of the two subtrees of every node **never differs by more than one**.

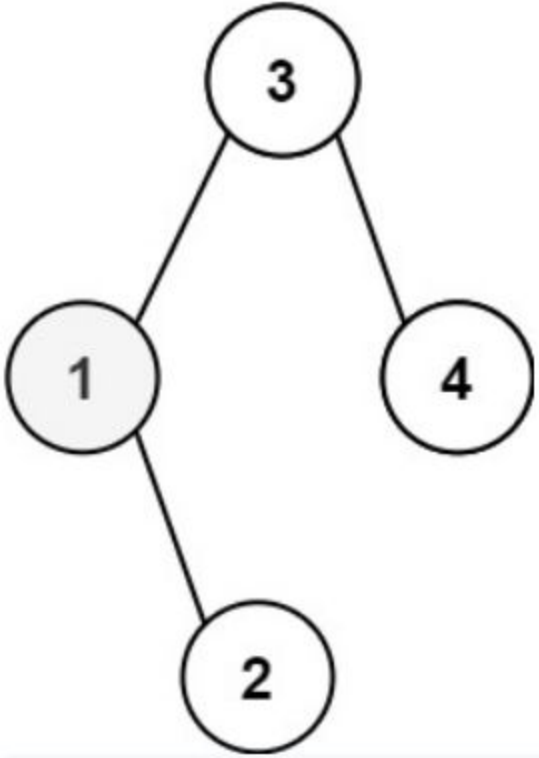
- Example: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- On the right side, there is only one way to make it a balanced BST



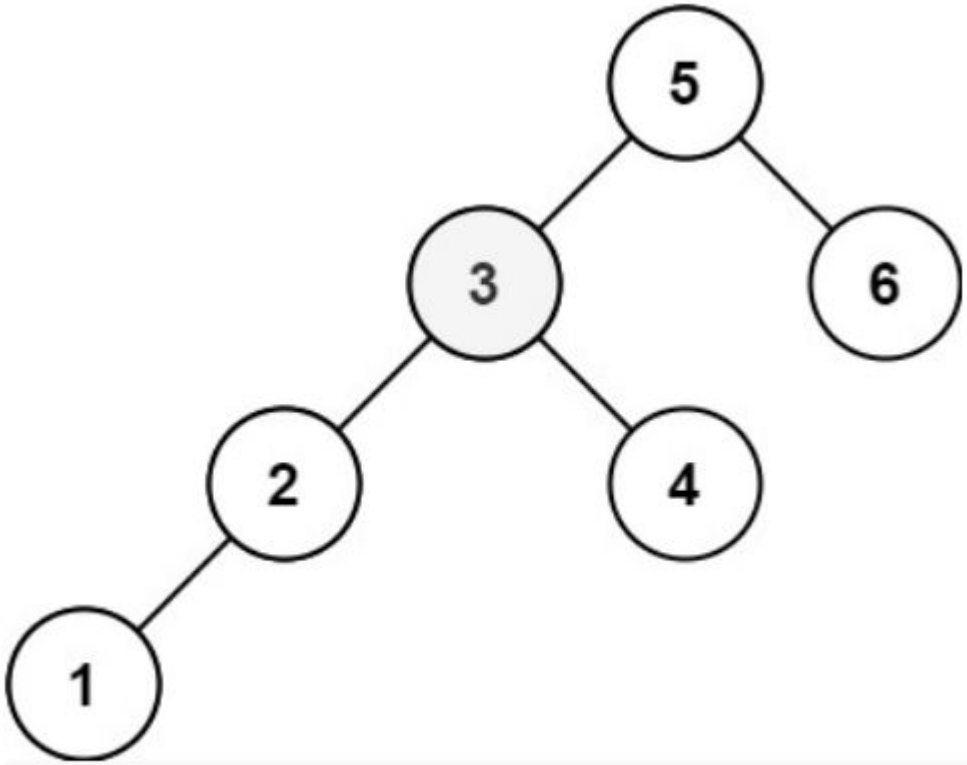
Problem #4: LeetCode 230 - Kth Smallest Element in a BST

Given the `root` of a binary search tree, and an integer `k`, return *the* k^{th} *smallest value (**1-indexed**) of all the values of the nodes in the tree.*

- A trivial way: compute inorder traversal, output is inorder[k-1]
  - Do something more efficient



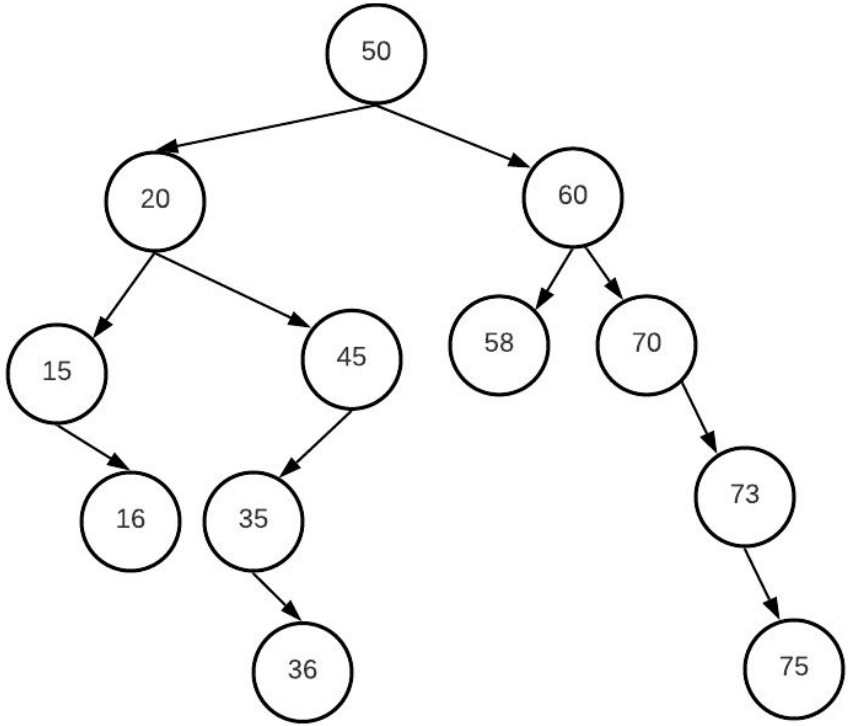
Input: root = [3,1,4,null,2], k = 1
Output: 1



Input: root = [5,3,6,2,4,null,null,1], k = 3
Output: 3

Problem #5: LeetCode 235 - Lowest Common Ancestor of a Binary Search Tree

- Given **2 nodes**, find their LCA
- LCA(x, y): the farthest node from the root that is an ancestor for both x and y.
  - The root is common ancestor for any pair,, but we want to find the farest from root
  - LCA(16, 45) = 35
  - LCA(45, 36) = 45
  - LCA(15, 70) = 50
  - LCA(58, 70) = 60
  - LCA(36, 75) = 50
  - LCA(70, 75) = 70



"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."