

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: Iterative Search

- `bool search_iterative(int target)`
- The same logic as our search but iterative

Problem #2: Is BST?

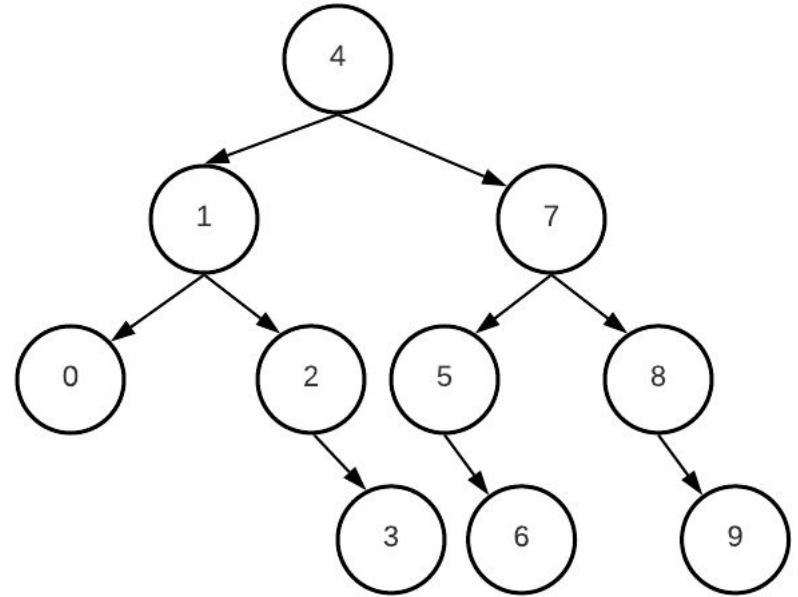
- `bool is_bst()`
- The function returns true if the current binary tree is BST
- Describe 2 ***fundamentally different*** approaches to check that
 - E.g. don't do it recursive then iterative. This is same idea with different implementation

Problem #3: Array to Balanced BST

- *Background:* Sometimes we have very imbalance BST and we want to convert to Balanced BST. We get its inorder traversal and rebuild it
- Given a **sorted** array create external function to create balanced BST
- `BinarySearchTree* build_balanced_bst_tree(vector<int> &values)`

Problem #3: Array to Balanced BST

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

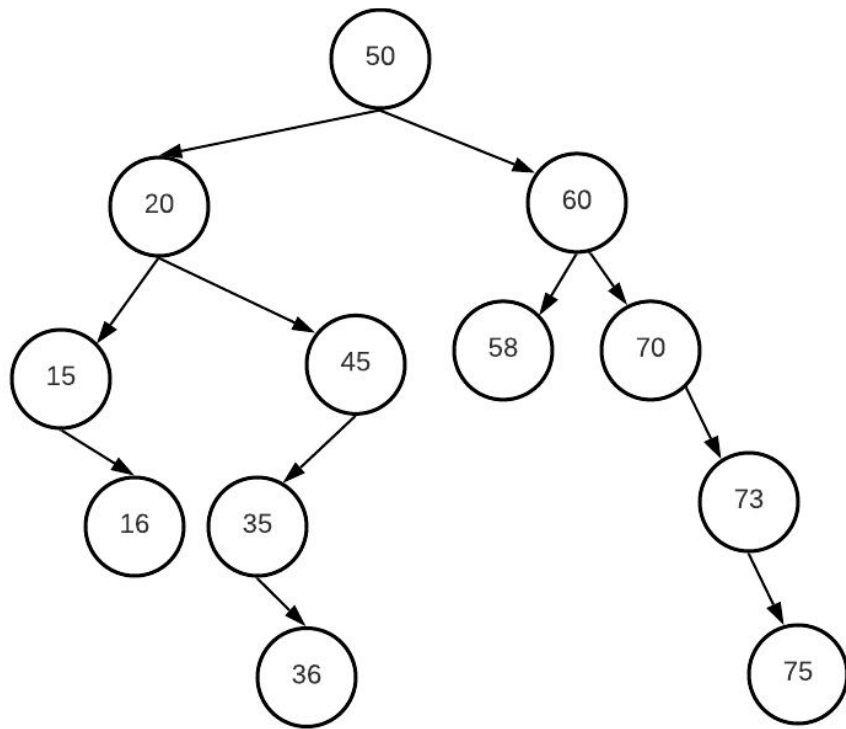


Problem #4: Kth smallest number

- Given a BST and integer K (1-based), find the kth smallest number
 - A trivial way: compute inorder traversal, output is `inorder[k-1]`
- Implement: `int kth_smallest(int &k)`
 - It returns integer for the kth value, or -1234 if it doesn't exist
- E.g. if we inserted in the BST tree: 20 10 80 15
 - The `kth(1) = 10`, `kth(2) = 20`, `kth(6) = -1234`

Problem #5: Lowest Common Ancestor (LCA)

- Given 2 values that exists in a BST, find their LCA
- $LCA(x, y)$: the farthest node from the root that is an ancestor for both x and y .
 - Root is common ancestor for any pair, but now lowest
 - $LCA(16, 45) = 35$
 - $LCA(45, 36) = 45$
 - $LCA(15, 70) = 50$
 - $LCA(58, 70) = 60$
 - $LCA(36, 75) = 50$
 - $LCA(70, 75) = 70$



“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”