

Assignment # 3

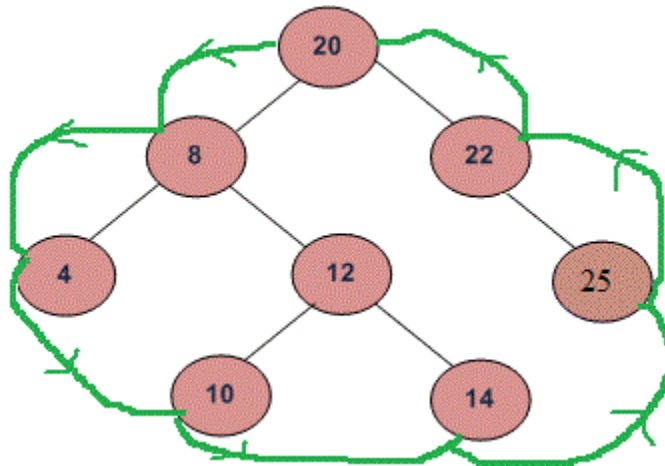
Data Structures and Algorithms

BS CS F22 and Special Section

Submission deadline: Friday 30th August, 2024.

Write C/C++ code for the solution of following problems.

- 1) A lazy-deletion binary search tree (BST) is a binary search tree where the deleted element is not physically deleted, rather it is just marked for deletion. We simply tag as 'deleted' while the nodes themselves remain in the tree. Whenever requested a cleanup() member function may be called to delete all erased nodes physically at once. All traversal operations must ignore the elements with 'deleted' tag.
- 2) Given two Binary Search Trees consisting of unique positive elements, we have to check whether the two BSTs contain same set or elements or not.
Note: The structure of the two given BSTs can be different.
- 3) Given a Binary Search Tree (BST), find the *kth* largest element. Repeat this for *kth* smallest element.
- 4) Given a binary tree, print boundary nodes of the binary tree Anti-Clockwise starting from the root. For example, boundary traversal of the following tree is "20 8 4 10 14 25 22"



- 5) Check a binary tree if it is a sum tree or not. In a sum tree, value at each non-leaf node is equal to the sum of all elements present in its left and right sub-tree. The value of a leaf node can be anything.
- 6) Given a team of N players. How many minimum games (comparisons) are required to find second best player? Do it by using tournament-tree (a form of heap).
- 7) Determine if a given Binary tree is a BST or not.
- 8) Sort all the values present in a binary tree using heap sort.
- 9) Given a Binary Search Tree (BST) and a range [min, max], remove all keys which are inside the given range. The modified tree should also be BST.

- 10) Given a Binary Search Tree (BST) and a range [min, max], remove all keys which are outside the given range. The modified tree should also be BST.
- 11) Given a binary tree, check if it is a complete binary Tree or not.
- 12) Given a binary tree, print its nodes level by level in reverse order i.e. all nodes present at last level should be printed first and then up-to first level i.e. root.
- 13) Calculate height of binary tree – Iterative
- 14) Calculate height of binary tree – Recursive