

## CSE 204: Offline 4

### Binary Search Tree

A Binary Search Tree (BST) is a binary tree with the following properties:

- The left subtree of a node contains only nodes with keys less than the node's key.
- The right subtree of a node contains only nodes with keys greater than the node's key.
- The left and right subtree each must also be a binary search tree.

In this assignment, you will implement a BST of integers. You need to code in C++ and use object oriented programming style. Your implementation must permit the following operations on the BST:

1. Search: Take an integer as input and return a Boolean indicating whether the integer is present in the BST.
2. Insert: Take an integer as input. If the integer is not present in the BST, insert the integer into the BST. If the integer is present in the BST, do nothing. (Consequently there will be no duplicate integers in the BST.)
3. Delete: Take an integer as input. If the integer is present in the BST, delete the integer from the BST. If the integer is not present in the BST, do nothing.
4. Inorder traversal: print the inorder traversal of the BST.
5. Preorder traversal: print the preorder traversal of the BST.

For the first three operations (search, insert, and delete), the time complexity in average case and worst case must be  $O(\log n)$  and  $O(n)$  respectively. For the last two operations (inorder and preorder traversal), the time complexity must be  $O(n)$ . Here,  $n$  is the number of nodes in the BST.

Please bring your implementation of the BST in the next sessional class and be prepared to sit for an online on BST.