Pre-Code WriteUp
CSI 5301

Assignment 6.1: BBST Select
Due: Fri June 17, 2022 11:59pm
Submitted by: Pawan Bhandari

System Implementation Details

- Save the current height of the node as a property in the Node class so that we can calculate balance factor to maintain a balanced BST.
- Given 'N' numbers are inserted one by one to the BBST so that the height of BST is logN
- Once the BBST is formed for N, perform a In-order traversal as push the value from each node as we traverse through. This results in a vector of 'N' elements which is sorted.
- SELECT subroutine will then return the element at (i-1) in this sorted vector. This value is the i^{th} smallest value. This subroutine will take O(n). If we stop the traversal once we populate the i^{th} element (once we have traversed i nodes by performing in-order traversal) we can stop the traversal as that will be the i^{th} order statistic and run time in this case will be O(i).

Algorithm

1) Read the std input lines and push numbers from first line to a vector<int> and int variable ith

2) Insert the numbers as nodes creating a Balanced Binary Search Tree

   i. Recursively insert the number as a leaf node and adjust nodes as in normal BST insertion maintaining the BST invariant

   ii. Update the height of this parent node

   iii. Calculate the balance factor of this parent node and check if this node is imbalanced now.

      a) Balance factor is the difference between the height of left and right child nodes

      b) For a BST to be BBST, magnitude of balance factor can't be greater than 1

      c) Handle imbalanced scenarios by rotating the sub-tree at this parent Node

3) Perform a in-order traversal and push the node values to another vector<int> sortedNumbers

4) Print the integer at (ith-1) position on sortedNumbers vector

References:

1) https://www.geeksforgeeks.org/avl-tree-set-1-insertion/
2) https://www.youtube.com/watch?v=1QSYxIKXXP4
3) https://github.com/williamfiset/Algorithms/blob/master/src/main/java/com/williamfiset/algorithms/datastructures/balancedtree/AVLTreeRecursive.java