COP 5536 Project 1 Report

Sanjay Nair

Spring 2016

Compiler User: Java

Files and Function Prototypes

bbst.java Contains the top level functions specified in the project description. Instantiates a Tree internally and interfaces it through its functions. Some functions provide input and output processing to the console for the user.

- public static void parseAndExcute(String input)
- public static void increase(int theID, int m)
- public static void reduce(int theID, int m)
- public static void count(int theID)
- public static void inRange(int id1, int id2)
- public static void next(int theID)
- public static void previous(int theID)

Node.java Node structure contained in the Tree. Contains key, value, and color information as well as pointers to children and parent.

- public Node(int id)
- public Node(int id, int value)
- public Node(int id, int value, Color color)
- public Node left()
- public Node right()
- public Node parent()
- public Node grandparent()
- public boolean isLeft(Node n)
- public boolean isRight(Node n)
- public void flipColor()
- public int increaseValue(int amount)
- public int decreaseValue(int amount)
- public int degree()
- public int redDegree()
- (getters and setters for left, right, parent, key, value, and color fields)

Tree.java Implementation of a Red-Black Tree that holds integer keys and values. This is the structure that the top level program maintains to hold all the inputted keys and values.

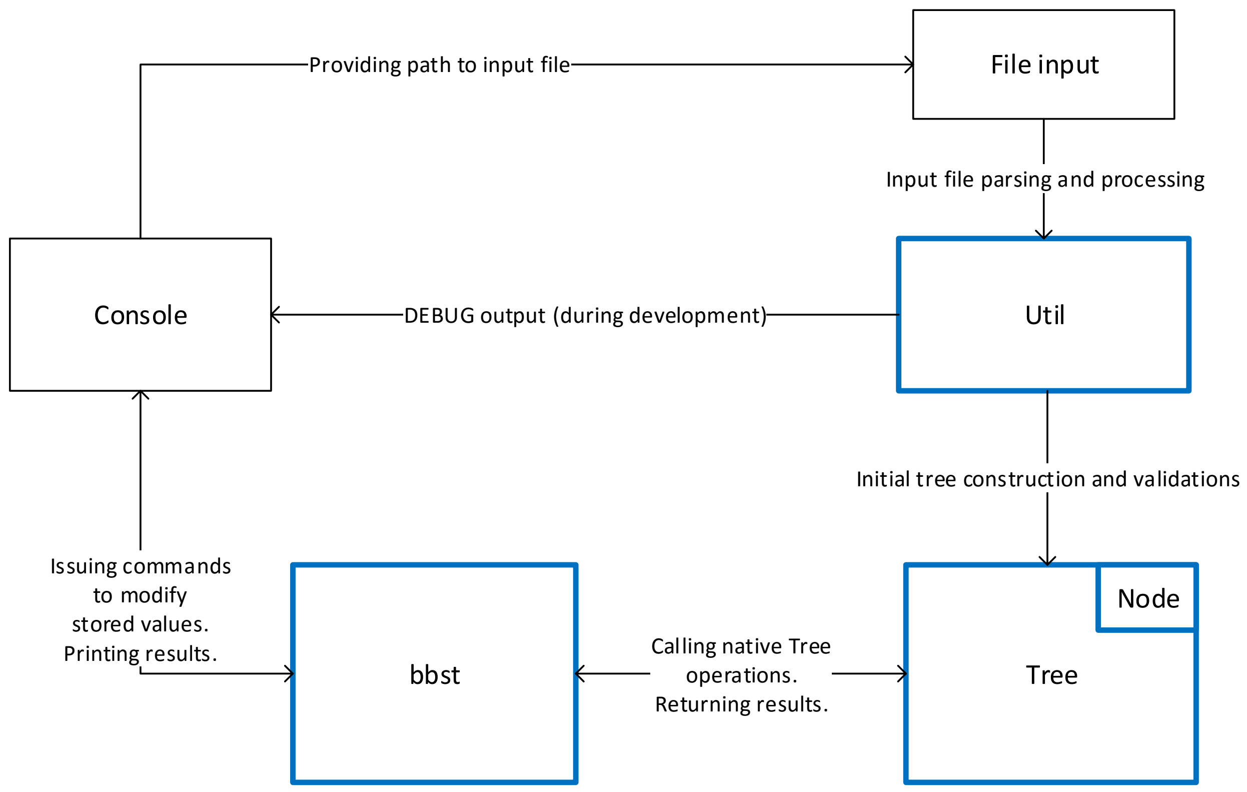
- public Tree()
- public Tree(Node root)

- public void insert(int val)
- public void insert(int id, int value)
- private void fixRBInsert(Node reference)
- public Node getNode(int id)
- public int getValue(int id)
- public Node getNextNode(int id)
- public Node getPreviousNode(int id)
- public boolean delete(int id)
- private void fixRbDelete(Node y, Node py, boolean isRight)
- private Pair normalDelete(Node n)
- private Node getMinNode(Node root)
- private Node getMaxNode(Node root)
- public int increase(int theID, int m)
- public int decrease(int theID, int m)
- public Node leftRotate(Node z, boolean flip)
- public Node rightRotate(Node z, boolean flip)
- public void leftRightRotate(Node z, boolean flip)
- public void rightLeftRotate(Node z, boolean flip)
- public void customRotateRight(Node py, Node v)
- public void customRotateLeft(Node py, Node v)
- public int inRange(int id1, int id2)
- private int inRangeHelper(Node root, int id1, int id2)

Util.java Contains utility functions for actions such as verifying the properties of the Tree, building a Tree from an input file, and printing out a given tree for debugging purposes.

- public static Tree readInputFile(String filename)
- public static void printTree(Node root)
- public static boolean checkValidBST(Node root)
- public static boolean checkValidRbBST(Node root)
- public static boolean verifyProperty1(Node n)
- public static boolean verifyProperty2(Node root)
- public static boolean verifyProperty4(Node n)
- public static void verifyProperty5(Node root)
- private static int verifyProperty5Helper(Node n, int blackCount, int pathBlackCount)
- public static Node buildBBSTFromSortedArray(int[] keys, int[] values, int start, int end)
- public static void colorDeepestLeafs(Node root)
- public static List<Node> findDeepestNodes(Node root)
- private static void findDeepestNodesHelper(Node root, int level, Object[] levelNodes)
- public static void createHugeInputFile()

Program Structure


File input
Providing path to input file
Input file parsing and processing
Console
DEBUG output (during development)
Util
Initial tree construction and validations
Issuing commands to modify stored values. Printing results.
bbst
Calling native Tree operations. Returning results.
Node
Tree