COP 5536 Fall 2022  
Programming Project

AVL Tree

**Instructions to Run:**

Navigate to the project directory and run the following set of commands.

|  |
| --- |
| * make * java avltree <input\_file> |

The program will create a tree and perform the operations listed (Initialize/Search/Insert/Delete) in the input file. The output for the search file will be written to the filename called **output\_file.txt.**

**Program Structure:**

The whole program is divided into 7 classes.

1. **TreeNode:**

This class represents the structure of the individual node of the AVL tree.

It encapsulates the following attributes and their getters setters:

1. **data:** Key associated with the node
2. **height:** Height of the subtree.
3. **left:** Pointer to the left child
4. **right:** Pointer to the right child.
5. **OpCode**

This is the Enum class that represents the operations to be done on the tree i.e programmatically in the form of enumeration. It provides enumerations for the following actions to be done on a tree.

1. ***Initialize***
2. ***Insert***
3. ***Delete***
4. ***Search***
5. **Operation:**

This class is the programmatic representation of the individual record that is being read from the input file provided.This class consists of the following attributes and their getters setters:

1. ***opcode*:** Enum for the operations provided and is of type code.
2. ***val1*:** First Integer value associated with the operation.
3. ***val2*:**  Second Integer value associated with the operation. Only Applicable for Search Range Operation
4. **InputReader**

This class contains functions that help with reading operations specified in the supplied input file. It contains two functions:

1. ***private static Operation readRecordFromLine(String line):***

This is a helper function that takes a string as a parameter and parses it with the regular expression **^([A-Za-z]+)\(([0-9,]\*)\)$** to obtain pattern groups: the first group contains the operation name (Initialize/Insert/Delete/Search) and the second group contains the data associated with the operation.

For e.g: The text “Insert(28)” will be divided into two groups first being “Insert” and the second being “28”. These groups will be then used to construct an Operations Object and that object will be returned from this function.

1. ***public List<Operation> getOperationsFromFile(String filename):***

This function reads lines from the filename passed as the function parameter.

And then calls the function “**readRecordFromLine(String line)”**  on every line to construct an Object of class Operation. This object is added to a list. And the list of such objects is returned from the function.

1. **ResultWriter**

This class contains the functionality to read the list operations supplied in the input file.

It contains the following functions:

1. ***public void initializeFile():***

Creates a FileWriter Object that in turn creates a new file named **output\_file.txt**

1. ***public void writeToFile(String txt):***

Takes a string as a parameter and writes that string as a newline to the file created earlier.

1. ***public void close(String txt):***

Closes the file object created by the initializeFile() function.

1. **AVLTreeUtil:**

This class maintains and takes care of the operations on the Actual Tree. It contains the following functions:

1. **insert(int data):**
2. **insert(TreeNode treeNode, int data):**
3. **delete(int data):**
4. **delete(TreeNode treeNode, int data):**
5. **getHeight(TreeNode treeNode):**
6. **getBalanceFactor(TreeNode treeNode):**
7. **findReplaceableNode(TreeNode treeNode):**
8. **rotateLeft(TreeNode treeNode):**
9. **rotateRight(TreeNode treeNode):**
10. **search(int val)**
11. **search(TreeNode treeNode, int val):**
12. **search(TreeNode treeNode int val1, int val2):**
13. **search(TreeNode treeNode, int val1, int val2, List<String> result):**
14. **avltree:**

This is the main class that acts as a driver/wrapper for calling functions of classes mentioned above. It instantiates objects of other classes to drive the operations.

1. **initializeTree():**

Initalizes the new instance of AVLTreeUtil Class.

1. **writeResult(String result):**

Wrapper call for the ResultWriter’s writeToFile(String text) method of AVLTreeUtil class .

1. **closeWriter():**

Wrapper function close() method of AVLTreeUtil class.

1. **insert(int val):**

Wrapper function for insert(int val) method of AVLTreeUtil class instance.

1. **delete(int val):**

Wrapper function for delete(int val) method of AVLTreeUtil class instance.

1. **search(int val):**

Wrapper function for search(int val) method of AVLTreeUtil class instance.

1. **search(int val1, int val2):**

Wrapper function for search(int val1,int val2) method of AVLTreeUtil class instance.

1. **readOperations(String filename):**

Wrapper function for readOperations(String filename) method of InputReader class instance. The InputReader’s readOperations call returns a List of Objects of class Operations, which is then assigned to local attribute named operations.

1. **performOperations():**

Iterates over objects from list and perform operations on the basis of the opcode of the operations and the data associated with each operation.

**Program Structure:**

**Diagram, schematic

Description automatically generated**