Documentation for AVL Tree class in Java:

This class implements the AVL tree. The time complexity for search, insertion and deletion is O(log n).

AVL Node class

I have used AVL Node class for creating the nodes.

This class has

1. ht variable to store the height of the tree below it.
2. Data – stores the data.
3. lchild – left child.
4. rchild – right child.
   2 constructors one with default height = 0 and another with height as per the choice of user.

AVL Tree class

Constructor methods:

1. public AVLTree()- it just initializes the tree with root node as null.
2. Public AVLTree(ArrayLiat<Integer>arr) this constructor creates a tree with all the elements of arrayList.

Private Variables:

1. private AVLNode root
2. private int max(int x, int y)
3. private int getht(AVLNode x)

   This method returns the height of the subtree below the AVLNode x.

4. private AVLNode lrRotate(AVLNode root)
5. private AVLNode llRotate(AVLNode root)
6. private AVLNode rrRotate(AVLNode root)
7. private AVLNode rlRotate(AVLNode root)

   These methods rotates the subtree and returns the new root.
8. Private AVLNode[] getGrandParentParentandChild(int x)- this method returns an array with the node x, it's parent and grandparent.
9. Private AVLNode erase(int x, AVLNode root) – this method is internally used by delete method to delete an element in the tree.
10. Private void doAppropriateRotations(AVLNode del_node_par, AVLNode del_node_gpar) – this method does rotations at the parent node of the deleted node.

Public variables:

1. public ArrayList<Integer> getPreOrderTraversal() – this method returns an arrayList of data in preordertraversal format.
2. public ArrayList<Integer> getPreOrderTraversalht() – this method returns an arraylist of height of subtrees under every node in preOrdertraversal format.
3. Public void insert(int x) – this method creates an AVLNode with data=x and inserts into the tree.
4. Public void delete(int x) – this method searches for the node with data = x, deletes the node and updates the root node.
5. Public Boolean search(int x) – this methid searches for the node with data = x and returns true if found else false.

##This code and documentation were written by sai dharma. For any suggestions please mail dharmasai0@gmail.com.