

Assignment Two

In this assignment, you will implement a class named `ExtendedAVLTree`. `ExtendedAVLTree` extends the `AVLTree` class to include the following methods:

- `Public static <K, V> AVLTree<K, V> clone(AVLTree<K,V> tree)`
This class method creates an identical copy of the AVL tree specified by the parameter and returns a reference to the new AVL tree. **(3 marks)**

The time complexity of this method must be $O(n)$, where n is the size of the input avl tree. Put your running time analysis as comments after the code.

- `public static <K, V> AVLTree<K, V> merge(AVLTree<K,V> tree1, AVLTree<K,V> tree2)`
This class method merges two AVL trees, `tree1` and `tree2`, into a new tree.

The time complexity of your merge method must be $O(m+n)$, where m and n are the numbers of nodes of the two input AVL trees. **Hint:** 1. Create a sorted array list containing all the entries in both avl trees in $O(m+n)$ time. 2. Construct an avl tree based on the sorted array list in $O(m+n)$ time. Put your running time analysis as comments after the code. **(4 marks)**

- `public static <K, V> void print(AVLTree<K, V> tree)`
This class method creates a new window and prints the AVL tree specified by the parameter on the new window. Each internal node is displayed by a circle containing its key and each external node is displayed by a rectangle. You need to choose a proper size for all the circles and a proper size for all the rectangles and make sure that this method never prints a tree with crossing edges. **(3 marks)**

For simplicity, we assume that `K` is `int` and `V` is `String`.

All the related classes are in the package `net-datastructures-4-0`. Please download `net-datastructures-4-0`, install it on your own computer and create the new class `ExtendedAVLTree` in the same package.

You need to read the code of all the related classes in order to understand how the `AVLTree` class works.

How to submit?

Follow this link: <https://cgi.cse.unsw.edu.au/~give/Student/give.php>. Do the following:

1. Use your z-pass to log in.
2. Select current session, COMP9024 and assn2.
3. Submit `ExtendedAVLTree.java` containing all the code, excluding the code in `datastructures-4-0`.

Marking

The full mark of this assignment is 10. Marking will be based on the correctness, time efficiency and readability of your code.

Deadline

23:59:59 30 April, 2016.

No late submissions will be accepted.

References

1. <http://docs.oracle.com/javase/tutorial/2d/index.html>.
2. http://www.deitel.com/articles/java_tutorials/20050923/IntroductionToJava2D_Page6.html.
3. <http://docs.oracle.com/javase/tutorial/uiswing/components/frame.html>.