# Assignment 4

## AVL Trees

This assignment helps to reinforce your understanding of AVL trees (and thus binary search trees).

Implement method `restructure` in file AVLTree.java. Submit only file AVLTree.java.

You may use either the cut/link restructuring algorithm or the trinode restructuring algorithm (in the textbook). The latter is recommended as it makes program tracing and debugging easier.

Use the main program testProgram.java to test your code. Here is the expected output.

Download all the files in this directory/folder to your computer, including AVLTree.java and testProgram.java. To compile all the files, simply issue the command:

```
javac testProgram.java
```

To run the main program, issue the command:

```
java testProgram
```

### Notes

- Do not modify the given interface, class and method definitions and implementations.
- Do not add packages to the submited Java file.
- The input data given in the above main program are only examples for your testing. We will use different data sets to grade your implementations.
- Assume that all inputs are valid and distinct (no duplicate keys). Each tree node entry stores a key (e.g., your student ID) and an object (e.g., your personal information). We use integer keys in this assignment, and are not concerned about the values of the associated objects/records.
- Do not add any `System.out.print()` or `System.out.println()` to your code. We will redirect the output to a text file and parse the text when marking your program.
- The Dictionary ADT is equivalent to the Map ADT discussed in the lecture.

---

*Posted 28 February 2020*
*Last updated 28 February 2020*