

---

# COS 226 – Introduction to Data Structures - Homework #3

Due: 5:00PM 11/3/2023

---

**Homework Submission:** Homework must be submitted via Brightspace as PDF files. This includes your code when appropriate. Please use a high quality scanner if possible, as found at the library or your departmental copy room. If you must use your phone, please don't just take photos, at least use an app like CamScanner that provides some correction for shading and projective transformations. If you have never used Brightspace before, please familiarize yourself with it well before the homework deadline. They offer tutorials for students, and the best way to learn is to upload a draft of your solutions well before the deadline.

## 1. AVL Tree Insertion and Rotation Visualization (15 points)

Show the resulting AVL tree after adding the following elements. You may assume the tree is empty at the start. If a rotation is necessary, depict the tree prior to that rotation (circle the tree once it has been rebalanced):

96,45,35,85,52,49,5,98,92,43

Instructions:

- (a) Draw the AVL tree at each stage of insertion.
- (b) Clearly indicate any rotations that are required to maintain the AVL property.
- (c) Circle the AVL tree once it has been rebalanced.

## 2. AVL Tree Implementation (25 points)

Implement an AVL tree with the following methods, you may use the AVL tree implementation from class. Create the following methods which take in a AVL tree of any length:

- (a) **Maximum Value Lookup:** Implement a method to find the maximum key in the AVL tree.
- (b) **Pre-order Traversal:** Implement a method to print the keys in the AVL tree in pre-order.
- (c) **Post-order Traversal:** Implement a method to print the keys in the AVL tree in post-order.
- (d) **Level-order Traversal (Breadth-first Traversal):** Implement a method to print the keys in the AVL tree in level-order.
- (e) **In-order Traversal:** Implement a method to print the keys in the AVL tree in in-order.

Note: In Pre-order Traversal, the order of visitation is: Root -> Left -> Right. In Post-order Traversal, it's: Left -> Right -> Root.

### **Sanity Check**

Insert the following elements into an empty AVL tree: 30, 20, 40, 10, 25, 35, 50

Expected outputs:

Pre-order Traversal: 30, 20, 10, 25, 40, 35, 50

In-order traversal: 10, 20, 25, 30, 35, 40, 50

Post-order Traversal: 10, 25, 20, 35, 50, 40, 30

Level-order Traversal: 30, 20, 40, 10, 25, 35, 50

Maximum Value: 50