

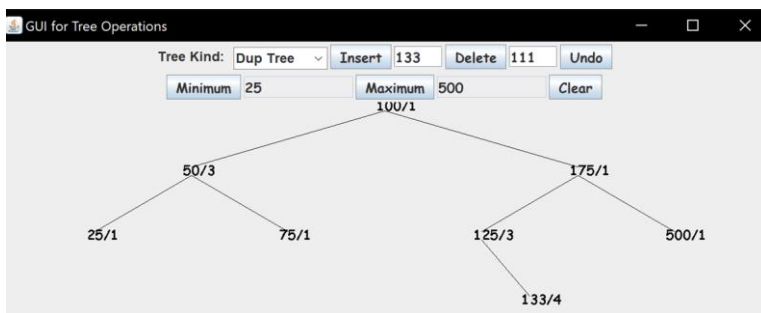
## Assignment 2 Part 2: Tree Memento

(to be done by the same team as in Part 1)

Due Date: Friday, October 16, 2020 (11:59 pm)

Lecture 9 contains a demo of a Graphical User Interface (GUI) for trees and duptrees. A screen-shot of the GUI is shown below. Using this GUI we can perform the four familiar operations on trees and duptrees: `insert`, `delete`, `min`, and `max`.

Most of the code for Part 2 is given at [Resources](#) → [Assignments](#) → [TreeGUI.java](#). You are to fill out the missing parts in the assignment. The focus here is on the `Undo` operation using which you should be able to undo all changes made to the tree/dup tree, restoring earlier trees/dup trees.



More specifically, you are to code the following parts:

- (i) The `clone()` operation in class `AbsTree`;
- (ii) The entire class `TreeMemento`;
- (iii) The `actionPerformed` methods for `deleteButton` and `UndoButton` in class `TreeGUI`. The `insertButton` code is provided to you for reference.

You are permitted to add extra fields and code in class `TreeGUI` in order to support the above operations.

**What can be undone:** Only those inserts and deletes that actually change the state of a tree/dup tree can be undone. Inserts and deletes that do not change the state of the tree/delete are not eligible for being undone. Hence, unlike A1 Part 1, the `insert` and `delete` methods here return a `boolean` indicating whether the state of the tree/dup tree has changed – this convention is followed in the state updating methods of Java's collection classes. Also, unlike A1 Part 1, it is permissible to delete the last value from a tree/dup tree.

Exception messages to be printed: "No more undo operations are possible," "Cannot delete from an empty tree," "Cannot delete non-existent value n," "Number format error – please re-enter value" "Cannot take min of an empty tree," and "Cannot take max of an empty tree."

**Demos and Testing.** Two demos, [A2\\_Part2\\_Tree\\_demo.mp4](#) and [A2\\_Part2\\_DupTree\\_demo.mp4](#), will be posted to clarify the expected behavior of your program for trees and dup trees.

**What to Submit.** Prepare a top-level directory named `A2_Part2_UBITId1_UBITId2` if the assignment is done by a team of two students; otherwise, name it as `A2_Part2_UBITId` if the assignment is done solo. (Order the `UBITId`s in alphabetic order, in the former case.) In this directory, place your revised `TreeGUI.java`. Compress the directory and submit the compressed file using the `submit_cse522` command. Only one submission per team is required.

**End of Assignment 2 Part 2**