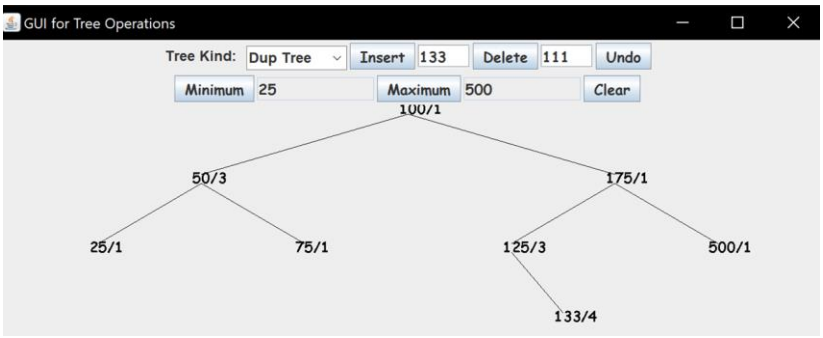
Assignment 2 Part 2 (please see small correction in *red*)

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

Due Date for Part 2: Friday, October 11, 2019 (11:59 pm)

Due Date for Part 1 (as before): Wednesday, October 9, 2019 (11:59 pm)

In Lecture 10 we saw 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`. The focus of Part 2 is on the `Undo` operation using which you should be able to undo all changes made to the tree/duptree, restoring earlier trees/duptrees. You are to code the following pieces:

(i) The `clone`() operation in class `AbsTree`;
(ii) The entire class `TreeMemento`;
(iii) The `actionPerformed` method for `UndoButton` in `TreeGUI`();

The definition of the `actionPerformed` method for `UndoButton` is similar to those for `insertButton` and `deleteButton` (given in the file). You may need to add some extra fields in class `TreeGUI` as well as some extra code for the other buttons (`clearButton`, `insertButton`, and `deleteButton`) in order to support the `Undo` operation. Test your program by interleaving insert's, delete's, and undo's.

Note: The `insert` and `delete` methods here differ from those of A1 Part 1 in that they return a `boolean` indicating whether the state of the tree/duptree has changed. The code for the `deleteButton` should recognize when a tree/duptree will become empty and should take suitable actions.

Exceptions to be raised: *"**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."*

***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 firle using the `submit_cse522` command. Only one submission per team is required.

End of Assignment 2 Part 2