CSE 522
Homework 1: Binary Seach Trees

Assigned: Sept 7, 2017
Due: Sept 18, 2017 (11:59 pm)

In the class lectures, we discussed two object-oriented definitions of the binary search tree (BST), called class `Tree` and class `AbsTree` respectively. In this homework, you are to provide a delete operation for both versions of the binary search tree.

Part 1: Define delete in class Tree

Write a `delete(int n)` method in class `Tree`. Essentially, it should remove the value `n` if present in the tree while ensuring that the remaining values in the tree maintain the binary search tree property. A good explanation of `delete` is given at:

http://www.algolist.net/Data_structures/Binary_search_tree/Removal

There are three main cases to `delete` depending upon whether the value to be deleted is:
(i) at a leaf node, or
(ii) at a non-leaf node with only one non-null subtree, or
(iii) at a non-leaf node with both non-null subtrees.

The file `BST_Part1.java` posted at `Resources` → `Homeworks` provides the outline of code for `delete` in addition to the code for the methods `insert` and `main`. A screen-cast `HW1.mp4` has been posted clarifying the three cases of delete.

Complete class `Tree` with the code for the `delete` operation. Essentially, you need to define in class `Tree` three `private' void procedures called `case1`, `case2`, and `case3` for the three main cases, respectively; the method `delete` should call these procedures appropriately. Before defining these three procedures, it would be helpful to define in class `Tree` three additional procedures, `min()`, `max()`, and `find(n)`, which return, respectively, the `Tree` node with the minimum value, the `Tree` node with the maximum value, and the `Tree` node with value `n`.

Run your program under JIVE and save the object and sequence diagrams in files called `obj.png` and `seq.png`, respectively, at the point when all `insert` and `delete` operations have been performed by `main`, but `main` has not yet exited.

Note: Print out error messages on the Console when the value to be deleted is either not present in the tree or is present at the root of the tree with both subtrees null.

What to Submit: Prepare a top-level directory named *HW1_Part1_UBITId* where *UBITId* is your UBIT id. In this directory, place `BST_Part1.java`, `obj.png` and `seq.png`. Compress the directory and submit the compressed file using the Linux command `submit_cse522`. Details regarding online submission will be posted in due course at

`Resources` → `Homeworks` → `Online_Submission_CSE522.pdf`.

Part 2: Define delete in class AbsTree

The file BST_HW1_Part2.java posted at Resources → Homeworks provides code for the classes AbsTree, Tree, and DupTree. Define a delete method in class AbsTree in such a manner that it works for ordinary trees as well as for duptrees. The second half of the screen-cast HW1.mp4 (from 2:50) shows how the delete operation works for duptrees.

It is important that the code for delete not be duplicated in classes Tree and DupTree. Rather, the code for the delete method in class AbsTree should capture what is common to trees and duptrees, and the differences should be defined in terms of one or more *protected abstract methods* that are implemented in the classes Tree and DupTree.

For duptrees, the delete(n) operation, assuming that value n is present in the duptree with a count > 1, should decrement the count field but not delete the node. If n is present in the duptree and the count == 1, the node should be deleted from the duptree.

Run your program under JIVE and save the object and sequence diagrams in files called obj2.png and seq2.png, respectively, at the point when all insert and delete operations have been performed by main, but main has not yet exited.

Note: Print out error messages on the Console when the value to be deleted is either not present in the tree/duptree or is present at the root of the tree (or root of the duptree with count == 1) and both subtrees are null.

What to Submit: Prepare a top-level directory named *HW1_Part2_UBITId* where *UBITId* is your UBIT id. In this directory, place BST_Part2.java, obj2.png and seq2.png. Compress the directory and submit the compressed file using the Linux command submit_cse522. Details regarding online submission will be posted in due course at

Resources → Homeworks → Online_Submission_CSE522.pdf.

End of Homework #1