Problem 6-1 (Number of Predecessors) 8 points

Assume you are given a binary search tree $T$ on $n$ elements. Using a slight modification of the PostOrder-Tree-Walk procedure, argue (by writing the pseudocode!) that in time $\Theta(n)$ you can compute, for every node $v$, the number of nodes (call it $v.less$) in $v$’s sub-tree which are less than $v$.

**Hint:** In addition to $v.less$, also compute the total number of nodes in $v$’s subtree.

(a) (5 points) Start with the simpler case when all the elements are distinct.

(b) (3 points) Your solution for part (a) will probably not work if some of the elements of $T$ could be the same. Show how to extend the solution for part (a) to handle this case. (Feel free to right away solve this general case for all 5+3=8 points.)

Problem 6-2 (Reconstructiong Tree) 6 points

The Preorder-Tree-Walk of a binary search tree $T$ is: 6, 2, 1, 4, 3, 5, 7, 9, 8. Draw $T$.

Problem 6-3 (Commutation of BST?) 10 points

Prove or show a counterexample for the following statements:

(a) (5 points) For any binary search tree $T$ and any element $x \notin T$, if one applies in sequence $\text{INSERT}(x)$ followed by $\text{DELETE}(x)$, then the result will be always $T$.

(b) (5 points) For any binary search tree $T$ and any element $x \in T$, if one applies in sequence $\text{DELETE}(x)$ followed by $\text{INSERT}(x)$, then the result will be always $T$.

Problem 6-4 (Non-Commutation of 2-3 Trees) 10 points

Consider the following 2-3 tree $T$:

```
  9
 / \
5   9
 /  \
1 3 5 7 8 9
```

If the statement is true, you need to give a general argument for any $T$. If false, you choose a specific $T$ which illustrates the problem.
(a) (5 points) Show an element \( x \notin T \) such that applying in sequence \( \text{INSERT}(x) \) and \( \text{DELETE}(x) \) will result with a tree \( T' \) that is different from \( T \).

(b) (5 points) Show an element \( x \in T \) such that applying in sequence \( \text{DELETE}(x) \) and \( \text{INSERT}(x) \) will result with a tree \( T' \) that is different from \( T \).