

CS251 Data Structures – Fall 2009

Homework 5 – 70 points

Due: Nov. 19

1. (10 points)
 - (a) Display all binary trees which have exactly 4 nodes.
 - (b) Let $T(n)$ = number of binary trees with exactly n nodes. Develop a recurrence relation for $T(n)$. Show that it gives the appropriate answer for $n = 4$. Determine $T(10)$. (NOTE: I'm not looking for a closed-form solution to the recurrence relation; just use it to generate $T(10)$.)

2. (10 points)
 - (a) Write a recursive method `int numFound(Object x)` which returns the number of times that the item `x` occurs in a binary tree. This method should be a non-static method in the `BinaryNode` class.
 - (b) Write a static version of the same function

3. (10 points) Assume we are given the in-order and post-order traversals of a tree. Write a Java program which reads in these two traversals and outputs the corresponding pre-order traversal for the tree. You may assume that each node is labeled by a single unique uppercase letter, that the traversals are specified on the command line (in-order first), and that no spaces are used in the traversal. For example, if you entitle your program `PrePrint` and run with the following input

```
java PrePrint DBFEAC DFEBCA
```

then the output should be

```
ABDEFC
```

You should make use of the `String` methods `char charAt(int)`, `String substring(int, int)` and `int indexOf(char)`. This latter method returns the index of the first occurrence of a character in string.

4. (15 points) Assume we insert the following values into an initially empty binary search tree: 6, 2, 5, 4, 11, 12, 9, 8, 1, 7, 3, 10
 - (a) Show the resulting tree after the 4, 8 and 10 have been added.
 - (b) Give the pre-, in- and post-order traversals of the final tree.
 - (c) What is the height of the final tree?
 - (d) Show the two possible trees that can result from deleting the root value from the tree.

(over)

5. (20 points)
- (a) Assume we insert the following values into an initially empty AVL search tree: 6, 2, 5, 4, 11, 12, 9, 8, 1, 7, 3, 10. Show the tree prior to and after any rotation which must be applied.
 - (b) Build a red-black tree using the insertion sequence in part (a). Show what the tree looks like after the insertion of the every other value starting at 2. You should use the top-down insertion technique.
 - (c) Build a splay tree using the insertion sequence in part (a). Show what the tree looks like after the insertion of the every other value starting at 2.
 - (d) The *internal path length* of a tree is the sum of the depths of all the nodes in the tree. Compare the internal path length of the final trees from Problem 4 and from parts (a), (b), and (c) of this problem.
6. (5 points) Assume we have a perfectly balanced splay tree which contains the values 1 through 7. Shown what the tree might have looked like prior to the last insert.