CSE 250 Spring 2010
Homework 5
Due Date: April 6, Wed, by 2:05pm
Total Points: 34

1. (3 pts) Consider the array:

   1, -9, 89, 25, 0, 5, 99, -4

   Show the progress and the array after each pass, when `quick_sort` is performed on this array. (Assume the first element of the array is used as the pivot element).

2. (3 + 3 + 1 = 7 points) As discussed in class, a `dictionary` is an abstract data type that supports the following operations:

   - `insert(key)`;
   - `find(key)`;
   - `delete(key)`;

   Here we assume that the `insert(key)` function simply inserts a new entry into the dictionary, without checking if there is already an entry with the `key` value in the `dictionary`.

   `Dictionary` can be implemented by using a sorted array, or an un-sorted array. The runtime of functions for the two implementations are summarized in the following table:

	insert(key)	find(key)	delete(key)
sorted array	$O(n)$	$O(\log n)$	$O(n)$
un-sorted array	$O(1)$	$O(n)$	$O(n)$

   (a) We can also use an *un-sorted linked list* to implement a `dictionary`. Briefly describe how to implement the three functions for this implementation. (Just description, not code). State the runtime of each function. (Your implementation should be as efficient as possible).

   (b) Consider the following program segment:

```
Dictionary<int> D;

for (int i = 0; i < n; i++) {
     D.insert(i);
     for (int j = 0; j < i; j++)
           D.find(j)
    }
D.delete(2);
D.delete(n-2);
```

   Determine the numbers of times each of the three functions of dictionary are called in the segment. (Either exact number, or in O notation).

   (c) For this program segment, which implementation (sorted array, or un-sorted array) should be used for the dictionary `D`? Explain why.

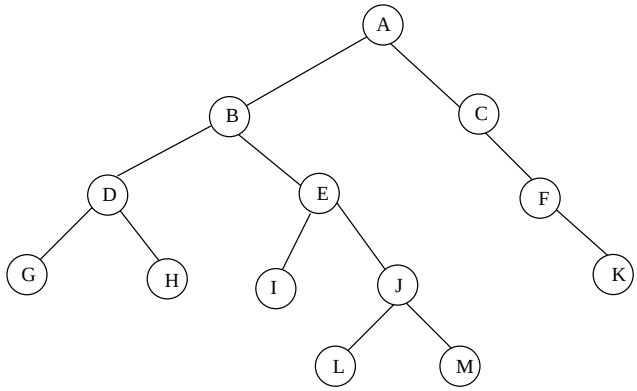

Figure 1: A binary tree.

3. (3 points) Consider the tree T shown in Figure 1. Show the listing of nodes in T by using:

   (a) In-order traversal.

   (b) Pre-order traversal.

   (c) Post-order traversal.

4. (1+2+2=5 points). Draw an expression tree corresponding to each of the following.

   (a) Inorder Traversal is `a / b - 3 * x / y`

   (b) Preorder Traversal is `* + a - x y / c d`

   (c) Postorder Traversal is `a b c + x y - z * / -`

5. (4+2 = 6 points) (a) Items with the following key values 5, 2, 9, 6, 3, 1, 4, 7 are inserted into an initially empty BST (binary search tree). Show the resulting tree after each insert operation.

   (b) Show the result of deleting the root.

6. (2+2+2= 6 points) Write efficient functions that take a pointer to the root of a binary tree T, and computes:

   (a) The number of nodes in T.

   (b) The number of leaves in T.

   (c) The height of T. (The height of T is the length of the longest path from the root of T to any leaf of T.)

7. (4 points) Write a function that generate a perfectly balanced binary search tree of height h with keys 1, 2, $\ldots 2^{h+1} - 1$. For example, when $h = 2$, the balanced binary search tree looks like:

```
      4
     / \
    /   \
   2     6
  / \   / \
 1   3 5   7
```