Practice Test

October 13, 2016

1. The classes MyBinaryTreeNode and MyBinaryTree are given below. Objects of type MyBinaryTree are linked binary trees.

```
public class MyBinaryTreeNode
{
   Object element;
   BinaryTreeNode leftChild;     // left subtree
   BinaryTreeNode rightChild;    // right subtree
 }

public class MyBinaryTree
{
   BinaryTreeNode root;                // root node
   // code you write will come here
}
```

   You are to write a public method maxHeightDifference(). The invocation x.maxHeightDifference() returns 0 if the binary tree x is empty; otherwise, it returns the maximum difference in the heights of the left and right subtrees of any node in the tree.

   (a) Write Java code for the public method maxHeightDifference. You may define, implement and create additional methods and variables as needed. You may use Java's methods Math.max and Math.abs. Math.max(a,b) returns the larger of a and b, and Math.abs(a) returns the absolute value of a. You may not create any new nodes, new instances of MyBinaryTree, or invoke any methods (other than Math.max and Math.abs) for which you have not provided code. (Hint: use recursion.)

   (b) What is the time complexity of your code as a function of the number of nodes in the binary tree?

2. Write a method reverseFirstK that accepts an integer k and a queue of integers as parameters and reverses the order of the first k elements of the queue, leaving the other elements in the same relative order. For example, suppose a variable q stores the following elements:

   (a) front [10, 20, 30, 40, 50, 60, 70, 80, 90] back

   The call of reverseFirstK(4, q); should change the queue to store the following elements in this order:

   (a) front [**40, 30, 20, 10**, 50, 60, 70, 80, 90] back.

If k is 0 or negative, no change should be made to the queue. If the queue passed is null or does not contain at least k elements, your method should throw an IllegalArgumentException.

3. Solve the following problems:

   (a) Consider the hash function Hash(X) = X mod 10 and the ordered input sequence of keys 51, 23, 73, 99, 44, 79, 89, 38. Draw the result of inserting these keys in that order into a hash table of size 10 (cells indexed by 0, 1, . . . , 9) for each of the following collision resolution strategies:

      i. separate chaining;

      ii. open addressing with linear probing

   (b) Suppose you have a binary tree whose data fields are single characters. When the data fields of the nodes are output in inorder, the output is ABCDEFGHIJ, and when they are output in preorder, the output is BAHCEDGFJI. Draw the binary tree showing the data in each node and the pointers between nodes. Show the steps used to arrive at the result.