CS5800 Assignment 3

Assigned: 2/8/2016 Due: 2/22/2016

In this assignment, you can assume node in BST has the following definition, the same as we used in class.

```
class Node {
  Key key;
  Value value;
  Node left;
  Node right;
  Node parent;
  int size;
}
```

Each problem is 10 points.

Problem 1

Draw the sequence of BSTs that results when you insert the keys E, A, S, Y, Q, U, E, S, T, I, O, N, in that order into an initially empty tree. And draw the sequence of BSTs that results when you delete the keys from the tree one by one in the order they were inserted.

Problem 2

Suppose that we have an estimate ahead of time of how often search keys are to be accessed in a BST, and the freedom to insert them in any order that we desire. Should the keys be inserted into the tree in increasing order, decreasing order of likely frequency of access, or some other order? Explain your answer.

Problem 3

For the set of keys {1, 4, 5, 10, 16, 17, 21}, draw BSTs of height 2, 3, 4, 5, and 6.

Problem 4

Design a method `random_key()` that returns a random key from BST in time proportional to the tree height in the worst case. Please write pseudocode, explain, and state running time.

Problem 5

We can sort a given set of n numbers by first building a BST containing these numbers (using insertion operations on each element one by one), and then printing the numbers by an inorder traversal. What are the worst case and best case running times for this sorting algorithm? Give concrete examples on when that happens.

Problem 6

Given the preorder traversal of a BST, design an algorithm to reconstruct the BST. Please write pseudocode, explain, and state running time.

Problem 7

Write a method `isBST()` that takes a Node as argument and returns true if the argument node is the root of a BST, false otherwise. Please write pseudocode, explain, and state running time.

Problem 8

Write the pseudocode for the following procedures in BST, and walk through the code with an example (at least 5 nodes in the tree).

- Find minimum key.
- Find kth smallest key.
- Delete a node.

Problem 9

Insert the keys 3, 13, 20, 21, 32, 39, 43, 45 in that order into an initially empty table of $m = 5$, using separate chaining. Use the hash function $h(k) = 11 \times k \ \% \ m$. Show what hash table looks like after each insertion.

Problem 10

Insert the keys 3, 13, 20, 21, 32, 39, 43, 45 in that order into an initially empty table of $m = 5$, using linear probing. Use the hash function $h(k) = 11 \times k \ \% \ m$. Show what hash table looks like after each insertion.