Written Assignment: Homework 7

1. Draw the binary search tree (filled with integers) whose postorder traversal is

   6  8  18  21  14  10

2. Let T be a binary tree (not a search tree). Denote the nodes with letters, e.g. A,B,C ...
   You are given below both the inorder and preorder traversal of the tree. Draw the tree.

   inorder:  F E D B A C
   preorder: B E F D C A

3. Given two binary trees, T1 and T2, Write the recursive method to test if T1 and T2 have
   the same shape. (In other words, check that each node in the trees has the same number
   of children in the same place).

4. What is the minimum number of nodes in an AVL tree of height 15? (This is problem
   4.18b in the text).

5. Consider the following AVL tree. (You should first verify that it is in fact a height-
   balanced tree).

   (a) Draw the result of inserting 11 into the tree, after rebalancing, no intermediate steps
       need to be drawn.
   (b) What kind of rotation did you use to rebalance in (a)?
6. Build a trie with the following words (these are the suffixes for the word MISSISSIPPI). The suffixes are: I PI PPI SIPPI SSIPPI ISSIPPI ISSISSIPPI ISSISSISSIPPI MISSISSIPPI MISSISSIPPI

7. Problem 5.1 on hashing from the text: Given input [4371, 1323, 6173, 4199, 4344, 9679, 1989] and a hash function $h(x) = x \mod 10$, show the resulting:
   
   (a) Separate chaining hash table
   (b) Hash table using linear probing
   (c) Hash table using quadratic probing
   (d) Hash table with second hash function $b_2(x) = 7 - (x \mod 7)$ (We didn’t cover a second hash function in class but it’s the obvious extension of linear and quadratic probing: $h_i(x) = h(x) + i \cdot h_2(x)$. It’s in the book.)

8. **Honors or extra credit**: Problem 4.49 in the text: Suppose we want to add the operation `findKth(k)` to return the kth smallest item in a binary search tree. Assume all elements in the tree are distinct. How can you modify what is stored in the tree to support this operation in $O(\log n)$ average time.

9. **Honors or extra credit**: Write an inorder traversal algorithm for binary search trees without using recursion but instead making explicit use of a stack.