Homework 4: Binary Search Trees

Note: ^ means “raise to the power”.

1. (2 points) Give the recursive version of TREE-INSERT routine in BST.(Pseudocode or C/C++/Java/Python code is acceptable)

2.(1 point) What are the worst case and average case complexities of searching in a binary search tree?
a. O(n), O(n)
*b. O(n), O(lg n)
c. O(lg n), O(n)
d. O(lg ), O(lg n)

3. (2 points) If we want to find the predecessor of a node ‘x’ and the node ‘x’ does not have a left subtree, then
*a. The predecessor is one of the node’s ancestors
b. This means the node does not have a predecessor
c. We start looking in the node’s right subtree
d. None of the above
**Explanation:** By definition of predecessor of a node

4. (1 point) The predecessor of a node ‘x’ is the node with the greatest key smaller than ‘x’.
*True
**Explanation:** By definition of predecessor of a node

5.(4 points) Write recursive version (pseudocode) of TREE-MINIMUM and TREE-SUCCESSOR.
**Answer:**

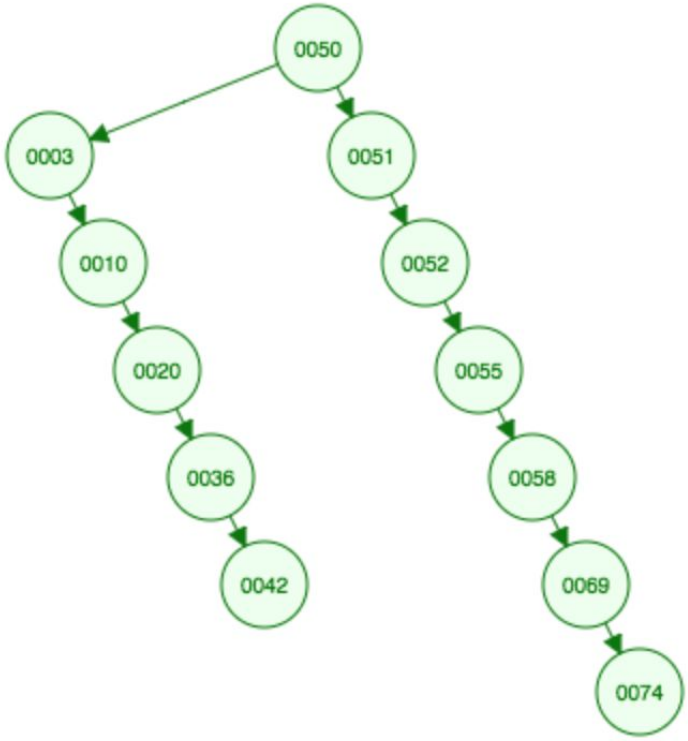
**TREE-MINIMUM:**

```
while x.left != NIL
        x = x.left
return x
```

**TREE-SUCCESSOR:**

```
if x.right != NIL
        return TREE-MINIMUM(x:right)
y = x.p
while y != NIL and x == y.right
        x = y
        y = y.p
return y
```

6. (2 points)The leaves of the BST formed by list/array elements 3,10,20,36, 42, 51,52,55, 58, 69, 74 and 50 as the root element will be?
*a. 42, 74
b. 3,20,42
c. 3,10,20,36, 42
d. 55, 58, 69, 74
**Explanation:** This is what the tree looks like and 42 and 74 will be the leave nodes.




7. (2 point) An algorithm swaps the left and right child of every root node in the Binary Search Tree converting it into a new Binary Tree.
*True
**Explanation:** It still remains a binary tree even though it is not a BST

8. (4 points) Write an algorithm to find the second smallest element in a Binary Search Tree.

The simplest algorithm according to CLRS would be returning the parent of the minimum.

**Answer:** you can use an inorder traversal to convert BST to a sorted array, and the second element of the array should be the second smallest element in a Binary Search Tree. Or you

can do it in a recursive way.
http://www.geeksforgeeks.org/find-k-th-smallest-element-in-bst-order-statistics-in-bst/

9. (2 points) The height of a BST is given as h. Consider the height of the tree as the number of edges in the longest path from root to the leaf. The maximum number of nodes possible in the tree is?
a. $2^{h-1}$ -1
*b. $2^{h+1}$ -1
c. $2^{h}$ +1
d. $2^{h-1}$ +1

10. (2 point) Given a binary search tree, which traversal type would print the values in the nodes in sorted order?
a. Post-order
*b. Inorder
c. Preorder
d. None of the above

11. (4 points) Convert the following expression into Postfix, Prefix and Infix notation and show the steps (A+B) *(C+D) demonstrate a few steps.

12. (2 points) The average time complexity for finding the SUCCESSOR and PREDECESSOR of a node in BST is:
a. O(1)
b. O(n)
c. O(n lg n)
*d. O(lg n)

13.(1 point) The time complexity to find max element in a binary search tree, which was created by inserting elements from a descending array, would be O(n).
*False
**Explanation:** Root element is the largest element if we create a binary search tree by inserting elements from a descending array, time complexity would be O(1).