Laboratory 9: Cover Sheet

Name __________Henry Huffman__________ Date _____10/20/14_____

Section _________1001________________________

Place a check mark in the *Assigned* column next to the exercises your instructor has assigned to you. Attach this cover sheet to the front of the packet of materials you submit following the laboratory.

Activities	**Assigned:** Check or list exercise numbers	**Completed**
Implementation Testing	✓	
Programming Exercise 1		
Programming Exercise 2		
Programming Exercise 3		
Analysis Exercise 1	X	
Analysis Exercise 2	X	
	Total	

Laboratory 9: Implementation Testing

Name ____ Date ______________________

Section _______________________________________

Check with your instructor whether you are to complete this exercise prior to your lab period or during lab.

<table>
<tr><th colspan="4">Test Plan 9-1 (Binary Search Tree ADT operations)</th></tr>
<tr><th>Test case</th><th>Commands</th><th>Expected result</th><th>Checked</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

Laboratory 9: Programming Exercise 1

Name ____ Date ______________________

Section _______________________________________

Test Plan 9-2 (accounts database indexing program)

Test case	Expected result	Checked

Laboratory 9: Programming Exercise 2

Name ______________________________________ Date _____________________

Section _____________________________________

<table>
<tr><th colspan="4">Test Plan 9-3 (getCount operation)</th></tr>
<tr><th>Test case</th><th>Commands</th><th>Expected result</th><th>Checked</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><th colspan="4">Test Plan 9-4 (getHeight operation)</th></tr>
<tr><th>Test case</th><th>Commands</th><th>Expected result</th><th>Checked</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

Laboratory 9: Programming Exercise 3

Name ______________________________________ Date _____________________

Section _____________________________________

Test Plan 9-5 (writeLessThan operation)

Test case	Commands	Expected result	Checked

Laboratory 9: Analysis Exercise 1

Name ______Henry Huffman ___________________ Date ___ 10/20/14___________

Section _______1001______________________________

What are the heights of the shortest and tallest binary search trees that can be constructed from a set of *N* distinct keys? Give examples that illustrate your answer.

The height, H, is <= N, with N being the number of distinct keys. This is true because the height counts the number of nodes in the tallest branch, which does not necessarily include all of the nodes in the tree. The following examples illustrate this.

```
       8                                    8                                      8
   7      9                             7     9                                 7   9
  6          10                        4                                             10
4  5                                                                              11   12

H = 4                                  H=3                                     H = 4
N = 7                                  N = 4                                   N = 6
```

Laboratory 9: Analysis Exercise 2

Name _____Henry Huffman___________________ Date ______10/20/14_______

Section ________1001___________________________

Given the shortest possible binary search tree containing *N* distinct keys, develop worst-case, order-of-magnitude estimates of the execution time of the following Binary Search Tree ADT operations. Briefly explain your reasoning behind each of your estimates.

retrieve O(N)

Explanation:

This function is linear based upon the traversal it uses. In order to retrieve the specified data from a node, The retrieve function may have to travel to each node. Because there are N number of nodes, I concluded that the function must represent the linear function N.

insert O(log(N))

Explanation:

Insert is log(N) because it only travels to certain branches, thus eliminating numerous possible search cases and decreasing the total runtime.

remove O(log(N))

Explanation:

I also found the remove function to represent log(N). This is true because it only searches for the item of removal on specified branches;therefore, it too will decrease the number of nodes to search and decrease runtime.

writeKeys O(N)

Explanation:

I found writeKeys to be a function that represents a linear function, N. This is because this function must travel to each node before it outputs the data. Since there are N distinct nodes, this function can be represented as N.