ECE242 Data Structures and Algorithms
Fall 2008

2nd Midterm Examination
(120 Minutes, closed book)

Name: ____________________________
Student ID: ________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(10)</td>
</tr>
<tr>
<td>2</td>
<td>(20)</td>
</tr>
<tr>
<td>3</td>
<td>(25)</td>
</tr>
<tr>
<td>4</td>
<td>(10)</td>
</tr>
<tr>
<td>5</td>
<td>(15)</td>
</tr>
<tr>
<td>6</td>
<td>(20)</td>
</tr>
</tbody>
</table>

NOTE: Any questions on writing code must be answered in Java using data structures topics covered in lectures 1 - 26.
1. [10pt] Consider the below sequence of integers:

5  7  1  4  8  6  3  2  10  9

Show the steps involved when a merge sort algorithm is run on the above sequence of integers to sort them in ascending order (smallest value on the left). Note: You do not need to write any Java code, however, you need to show each step in the sorting process to receive full credit.
2. [20pt] 

   a) [17pt] Consider two unsorted arrays, ArrayA and ArrayB that store Na and Nb non-repeating integers, respectively. Note that some of the integers in ArrayA could be present in ArrayB as well. Write a Java method mergeSortAB that merges ArrayA and ArrayB into a single sorted list, ArrayC, such that ArrayC contains integers that are in ArrayA or ArrayB but not in both. In other words, the ArrayC will not contain any duplicate elements. Note: You may create any additional methods you like.

   Hint: The mergeSortAB method is exactly the same as the mergeSort method, except with an added feature that eliminates duplicates while merging.
b) [3pt] What is the running time of the new merge sort algorithm created in (a)?

3. [25pt] An unsorted list of 8 elements is given in the following array:

\[ A[0…7] = \begin{array}{cccccccc}
15 & 3 & 9 & 31 & 11 & 17 & 7 & 23 \\
\end{array} \]

*Note: You do not need to write any Java code in this question, however, you need to show each step in the sorting process to receive full credit.*

a) [10pt] Apply the insertion sort algorithm to sort the above array.
b) [10pt] Apply recursive quick sort algorithm to sort the above array. *Assume that the pivot is always the last array element for quick sort.*
c) [5pt] What are the big-O running times of insertion sort and quick sort in terms of input data size $n$? Provide a brief explanation for full credit.

4. [10pts] Demonstrate the insertion of the keys 5, 28, 19, 15, 20, 33, 12, 17, 10 into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be $h(k) = k \mod 9$.

Note: You do not need to write any Java code, however, you need to show each step in the insertion process to receive full credit.
5. [15pt] You are given a binary search tree (BST) consisting of $N$ nodes that store integers as elements. Assume that you have access to the BSTNode and the BST class with the following methods.

**Code for BSTNode**

```java
private class BSTNode
{
    private int id;
    private BSTNode left;
    private BSTNode right;

    public BSTNode(int key, BSTNode l, BSTNode r) {
        id = key;
        left = l;
        right = r;
    }
}
```

**Methods for BST**

```java
public class BST implements BSTInterface
{
    public void insert(int id);
    public void remove(int id);
    public void search(int id);
    public void printall();
    public int height();
    public void show();
}
```
a) [6pt] Write a method \texttt{findMin} for the class BST above that returns the element with the minimum value in the BST.
b) [6pt] Write a method `findMax` for the class BST above that returns the element with the maximum value in the BST.
c) [3pt] Analyze the running times of \texttt{findMin} and \texttt{findMax} methods for the best case and the worst case. Explain your answer with a picture.

6. [20pt]

a) [3pt] If a perfect (complete) binary tree has \( n \) leaves and all levels are fully populated, how many nodes does the tree have in terms of \( n \)?
b) [3pt] What is the maximum number of nodes in a binary tree of height $h$ in terms of $h$?

c) [10pt] Consider the following binary search tree (BST)

![Figure1: A binary search tree]

i) [3pt] Perform pre-order, inorder and post order traversal on the BST shown in Figure 1 and write the output sequence of each traversal.
ii) [3pt] In the BST shown in Figure 1, insert a new node with value 60. Show the resulting tree and all intermediate steps.
iii) [4pt] In the BST shown in Figure 1, remove the node 57 and show the resulting tree using one of the two methods used in lecture. Show all steps.
d) [4pt] Bob claims, “the order in which a fixed set of elements are inserted into a binary search tree does not matter; the same tree results every time”. Is Bob correct? If yes, explain why. If not explain why not. *You may use a simple example to prove your answer.*