CS171 Midterm 2 Practice Exam

Name:_______________

You are to honor the Emory Honor Code. This is a closed-book and closed-notes exam. You have 50 minutes to complete this exam. Read each problem carefully, and review your answers. Good luck!

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Earned</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Below are some potentially helpful formulas.

\[
1 + 2 + 3 + \ldots + (N - 1) + N = \frac{N \times (N + 1)}{2}
\]

\[
1 + x + x^2 + \ldots + x^{N-1} + x^N = \frac{x^{N+1} - 1}{x - 1}
\]
Problem 1. Give brief answers to the following questions.

1. What are the general rules for selecting which data structure to use among arrays, linked list, and binary search trees?

2. If the items are inserted into a binary search tree in random order, the **worst-case** height can be O(______) but the **expected** height is O( _____ ).

3. Which graph representation, adjacency list or adjacency matrix, is preferred for large, sparse graphs?

4. Which data structure, queue or stack, is used for depth-first search (DFS) algorithm for a graph?

5. What is hashing?
Problem 2

1. Given an array of integers: 6, 8, 3, 2, 5, 9, 1, 7, show the key intermediate steps of how the array gets sorted in ascending order. Indicate the pivot value at each step.

2. Write down the cost function for quicksort algorithm in the worst case in recurrence relations, and solve it.
Problem 3. Consider the partial implementation for the Binary Search Tree below.

```java
public class BST {
    private Node root; // root of BST

    private class Node {
        private int key; // sorted by key
        private Node left, right; // left and right subtrees
        private int N; // number of nodes in subtree

        public Node(int key, int N) {
            this.key = key;
            this.N = N;
        }
    }

    private int size(Node x) {
        if (x == null) return 0;
        else return x.N;
    }

    private Node put(Node x, int key) {
        if (x == null) return new Node(key, 1);
        if (key < x.key) x.left = put(x.left, key);
        else if (key > x.key) x.right = put(x.right, key);
        x.N = 1 + size(x.left) + size(x.right);
        return x;
    }
}
```

1. Draw the binary search tree that the `main` below would generate.

```java
public static void main(String[] args) {
    Node root = null;
    put(root, 4);
    put(root, 2);
    put(root, 3);
    put(root, 6);
    put(root, 5);
    put(root, 9);
}
```
2. Show the sequence of keys visited by in-order and pre-order traversal for the tree above.

3. Write a method which computes the rank of a given key in a given subtree, i.e. the number of keys in the subtree less than the given key. Note: x is not necessarily in the tree.

   private int rank(int key, Node x)
Problem 4

1. Briefly describe or write the pseudo code for the algorithms of inserting and removing minimum from a minimum binary heap tree.

2. Given the following heap, show the resulting tree after each of the following actions: insert 4, remove minimum, remove minimum
Problem 5.

1. Briefly describe or write the pseudo code for the Breadth First Search (BFS) and Depth First Search (DFS) algorithms for a graph.

2. Given the following graph, show the results of BFS search and DFS search.