1. [15 points] Start with an empty Binary Search Tree.
   a. Draw a picture of the tree that will result from adding the following values to your tree, in the order they are given: 20 30 10 24 18 4 12
   b. How will your answer change if the BST is replaced by an AVL tree?
2. [5 points] What is the Balance Condition for AVL trees?

3. [15 points] Now add data 19 and 11 (in that order) to your AVL tree from (1 b). Draw the resulting AVL tree.
4. [15 points] Here is an array of 8 integer values:

\[
\begin{array}{c|c|c|c|c|c|c|c}
2 & 9 & 4 & 1 & 7 & 6 & 8 & 3
\end{array}
\]

We can think of this as a tree. Give the array that will result from turning this array into a heap. You may find it easier to think in terms of trees, but put your final answer back into an array.
5. [20 points] I want to have a map where the keys are names of movies and the values are lists of the actors in the movie. I could implement this as a hashmap with separate chaining (linked lists) or a treemap with an AVL tree. I will put n movies into this structure.

a) What are Big-Oh estimates of the worst-case and average case insertion time for a new movie for both the hashmap and treemap versions?

<table>
<thead>
<tr>
<th></th>
<th>Average Case</th>
<th>Worst Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hashmap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Treemap</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b) Estimate the worst-case and average-case lookup times for a movie that is in the map, for both the hashmap and treemap.

<table>
<thead>
<tr>
<th></th>
<th>Average Case</th>
<th>Worst Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hashmap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Treemap</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

c) Suppose I want to output the movies in alphabetical order; which structure makes that easier?

d) Suppose I want to output the actors in alphabetical order; which structure makes that easier?
6. [15 points] Suppose we have a directed graph with positive weights and want to find the LONGEST path through the graph from a given source to each other node.
   a) When will there not be a longest path?
   
   b) Give an algorithm to find the weight of the longest (i.e., most weight) path from the source to each other node. (I’m not looking for pseudo-code; just an English description of the algorithm and whatever data structures are needed to implement it.)

Note for the 2014 class – we haven’t gotten to this yet. We will before the exam, but this won’t be covered on the exam.
7. [15 points] Here is a game. Start by building a structure, which can take a while. After you build the structure I will give you pairs of words w1 and w2. Your job is to find the shortest sequence of changes that turns w1 into w2, changing only one letter at a time, and always using actual English words. For example, we can turn cat into dog with 4 changes: “cat” “bat” “bag”, “bog” “dog”. Here is how bob becomes a spy: “bob” “bomb” “comb” “come” “comet” “covet” “covert” You should count the following as legal changes:
   a. Replacing any one instance of one letter with a different letter.
   b. Removing any one letter
   c. Adding any one letter

Give an algorithm for playing this game.
[This page is for scratch work]

Please write and sign the Honor Pledge at the end of your solutions.