CSE100 Practice Midterm
Fall 2015: October 29th, 2015

<table>
<thead>
<tr>
<th>Problem</th>
<th>Topic</th>
<th>Points Possible</th>
<th>Points Earned</th>
<th>Grader</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C++ Concepts</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Trees and data structure comparison</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>AVL and Treaps</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>RSTs and SkipList</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Red-Black Tree</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Run Time Analysis</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>90</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This exam is closed book, closed notes. **Write your name on every page, including reference and scratch paper.** Scratch paper must be turned in at the end of the exam.

You have 80 minutes to complete this exam. Work to maximize points. If you don’t know the answer to a problem, move on and come back later. Most importantly, stay calm and don’t panic. You can do this.

Name:__________________________________________
ID:___________________________________________    Proctor Initial:_________________________________

Name of student to your LEFT: ________________________
Name of student to your RIGHT: _______________________

(Write “N/A” if seat immediately to your left or right is not occupied, or a wall or aisle, etc.)

**DO NOT OPEN THIS EXAM UNTIL YOU ARE INSTRUCTED TO DO SO.**

**GOOD LUCK!**
1) C++ Concepts [15 points]
   a. (1 pt each correct; 0 pt if left blank or incorrect) True or False: [5 pts.]
      i. ______ In C++, an iterator for the std::set container will iterate over the elements in the set sorted order.
      ii. ______ A C++ reference can only be used as an alias with objects stored on the stack memory.
      iii. ______ Iterators can be "dereferenced" with the * operator just as pointers are.
      iv. ______ When the increment operator ++ is used on an iterator object, it modifies the container-object that the iterator is pointing to.
      v. ______ Dereferencing an uninitialized pointer may cause a segmentation fault.

   b. Consider following declarations. After some computation, the current state of memory is shown. Null pointers are shown as 0; non-null pointers are shown with an arrow indicating what object they point to. [10 pts]

      ```
      class FuManchu {
      public:
        FuManchu *mandarin;
        FuManchu *toki;
      }
      FuManchu *saoFeng;
      FuManchu *worf;
      ```

      Current state of memory:

      ```
      Starting with the current state of memory shown above, consider the C++ code shown below. In the space to the right, draw the state of memory after this code executes? [3 points]
      ```

      ```
      saoFeng->mandarin = 0;
      saoFeng->toki = worf;
      worf = worf->toki;
      ```
ii. True or False? Following statement would result in a segmentation fault starting from the state of memory shown below [2 pts]

\[
\text{saoFeng->mandarin->toki->toki = worf->toki->mandarin->toki->toki;}
\]

iii. Write C++ code (fill in the blanks) to change from memory state A to memory state B: [5 pts]

Memory State A: 

Memory State B:

\[
\begin{align*}
\text{saoFeng:} & \quad \text{mandarin:} \\
\text{toki:} & \quad 0 \\
\text{worf:} & \quad \\
\text{mandarin:} & \quad 0 \\
\text{toki:} & \quad 0 \\
\text{saoFeng:} & \quad \\
\text{toki:} & \quad 0 \\
\end{align*}
\]

\[
\begin{align*}
\text{saoFeng:} & \quad \text{mandarin:} \\
\text{toki:} & \quad 0 \\
\text{worf:} & \quad \\
\text{mandarin:} & \quad 0 \\
\text{toki:} & \quad 0 \\
\text{saoFeng:} & \quad \\
\text{toki:} & \quad 0 \\
\end{align*}
\]

\[
\begin{align*}
\text{__________} & \quad \text{mandarin = 0;} \\
\text{__________} & \quad \text{toki = 0;} \\
\text{worf->mandarin} & \quad \text{__________;} \\
\text{__________} & \quad \text{= worf;} \\
\text{worf = worf ->__________;} \\
\end{align*}
\]
2) Trees and data structure comparison [15 points]
   a) (1 pt each correct; 0 pt if left blank or incorrect) True or False: [5 pts.]
      i) _____ When the BST successor function is invoked on a given node, it always returns
         the right child of the node.
      ii) _____ In a BST, the minimum element is always the leftmost node in the tree.
      iii) _____ A Treap is a binary tree.
      iv) _____ Red-Black trees are generally slightly taller than AVL trees.
      v) _____ A red-black tree is a binary search tree.
   b) Answer the following questions regarding the binary search tree (BST) shown below. [6 pts]
      i. What is the minimum element? [1 pt]
      ii. What is the in-order successor of A? [1 pt]
      iii. What is the in-order successor of D? [1 pt]
      iv. C <= B: True or False? [1 pt]
      v. B >= F: True or False? [1 pt]
      vi. If the height of node D is zero, what is the height of the node A? [1 pt]
c) Draw the binary search tree after inserting following sequence of keys in order: [4 pt]
4, 5, 1, 2, 7, 8
3) AVL and Treaps [15 points]
   a) Build a Treap using following keys and priorities [4 pts].

<table>
<thead>
<tr>
<th>Key</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>20</td>
</tr>
<tr>
<td>O</td>
<td>30</td>
</tr>
<tr>
<td>N</td>
<td>40</td>
</tr>
<tr>
<td>E</td>
<td>50</td>
</tr>
<tr>
<td>T</td>
<td>60</td>
</tr>
</tbody>
</table>
b) Insert (B, 40) in the following Treap. Draw the final treap. [3 pt]

![Treap Diagram]

---

c) Identify the nodes in the following binary search tree that violates the AVL tree property by shading over them. [3 pts]

![Binary Search Tree Diagram]
d) Insert B in the following AVL tree (perform appropriate AVL rotations to get a final AVL tree). [5 pts]
4) **RSTs and SkipLists. [15 points]**
   a) (1 pt each correct; 0 pt if left blank or incorrect) True or False: [5 pts.]
      i) _______ RST’s are treaps in which priorities are assigned randomly.
      ii) _______ In a skip list, probability of a new node being level 1 is same as level 2.
      iii) _______ In an RST the value of the keys that have been inserted determine which node is the root of the tree.
      iv) _______ Height of an RST with N nodes (N > 0) is always less than N/2.
      v) _______ The total number of forward pointers in a skip list with N nodes is always less than 2N.

   b) Highlight the forward pointers that must be checked or followed to find the node with key 21 by marking the symbol “//” over the appropriate arrows. [5 pts]

   ![Skip List Diagram]

   c) If we insert a node X with key 20 and level 4 in the above skip list then (Assume 0-based indexing of pointers within nodes, write the letters as answers from the above skip list): [5 pt]
      i) The node whose index 0 pointer will change is: _______
      ii) The node whose index 3 pointer will change is: _______
      iii) The node X index 1 pointer will point to: _______
      iv) The node X index 2 pointer will point to: _______
      v) The node X index 3 pointer will point to: _______
5) **Red-Black Tree. [15 points]**
   a) Consider the red-black tree shown below, where the nodes 10, 20 and 35 are black, and 12 is red. Insert 11 into the tree. Show the final tree. [5 pts]

   ![Red-Black Tree Diagram](image)

   b) What is the minimum number of nodes in a red-black tree with four black nodes? [2pt]

   c) In a red-black tree with N black nodes, what is the tight lower bound on the number of nodes in the longest path from the root to a node? (circle one answer below) [2pt]

   i) $2 \times \log_2(N+1)$
   ii) $\log_2(N+1)$
   iii) $N$
   iv) 1

   d) Design a “worst-case” red-black tree with 6 nodes, i.e., a red-black tree with the longest possible path from the root to a leaf. [6 pt]
6) Run Time Analysis [15 points]
   a) Choose only one option in the following multiple-choice questions. [5 pts.]
      i) The worst case run time for finding an element in a BST of height H is:
         A. O(H)
         B. O(log_2H)
         C. O(1)
         D. O(Hlog_2H)

      ii) Given N unique keys which of the following orders of key insertions into an initially empty
          BST would result in the maximum height of the BST
          A. Sorted order (smallest key to largest key)
          B. Random order
          C. The order of insertion does not affect the height of the BST
          D. Cannot comment unless the actual values of the N keys are known

      iii) What is the average case runtime of finding an element in an RST with N keys?
           A. O(1)
           B. O(log_2N)
           C. O(N)
           D. O(N*log_2N)

      iv) The worst case Big O run time of finding the minimum key is smallest in which of the
          following data structures? Compare run times in terms of the number of nodes.
          A. AVL Tree
          B. RST
          C. General BST
          D. SkipList

      v) The worst case Big O run time of inserting an element is smallest in which of the following
          data structures? Compare run times in terms of the number of nodes.
          A. Red-Black tree
          B. RST
          C. General BST
          D. SkipList

   b) What is the total depth of the following BST if the probability of searching for keys 4, 10 and 20
      is 0.4, 0.1 and 0.5 respectively [2 pts]
c) What is the expected total depth of BSTs that can be constructed with the keys (4, 11, 20, 21, 25), assuming that key value 11 is always inserted first and all orders of insertion for the remaining keys are equally likely. Use the following recursive relationship, wherever necessary:

\[ N \times D(N) = (N+1) \times D(N-1) + 2N - 1, \]

where \( D(N) \) is the expected total depth of all trees with \( N \) keys, under the assumption that all orders of key insertions are equally likely. [8 pts]