Homework Assignment 9
CS 330 Discrete Structures
Summer Semester, 2016

Name:

Due: 8:50am June 27, 2016

1. (5pts) How many edges does a tree with 14,381,132 vertices have? Do NOT need to explain your answer.

   A tree of |V| vertices have |V| − 1 edges, therefore the answer is 14,381,131.

2. (15pts) Given the following tree

   (a) What is the sequence of letters when the tree is traversed with preorder?
       abcdefgh
   (b) What is the sequence of letters when the tree is traversed with inorder?
       cbdaegfh
   (c) What is the sequence of letters when the tree is traversed with postorder?
       cdbghfca
3. (15pts) Construct a complete binary (every node other than leaves have two children and all leaves are at the same level) tree whose traversal is a,b,c,d,e,f,g when
   (a) it is traversed with preorder traversal

   ![Preorder Traversal Diagram]

   (b) it is traversed with inorder traversal

   ![Inorder Traversal Diagram]

   (c) it is traversed with postorder traversal

   ![Postorder Traversal Diagram]

4. (20pts)
   (a) Form a Binary Search Tree (BST) by inserting the numbers below in the order they appear in the sequence.

   3, 7, 5, 1, 4, 9, 2

   ![BST Diagram]
(b) What is the new BST after 7 is deleted from the above BST?

5. (10pts) Use Prim’s and Kruskal’s algorithms to find the MST in the following graph. Show the edge choices step by step as in Lecture note 9.

(a) Prim’s algorithm:

<table>
<thead>
<tr>
<th>Choice #</th>
<th>Chosen edge</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(a, b)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>(a, e)</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>(e, d)</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>(c, d)</td>
<td>1</td>
</tr>
</tbody>
</table>

Total: 6

(b) Kruskal’s algorithm:

<table>
<thead>
<tr>
<th>Choice #</th>
<th>Chosen edge</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(a, b)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>(c, d)</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>(a, e)</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>(e, d)</td>
<td>2</td>
</tr>
</tbody>
</table>

Total: 6

6. (20pts) Prove that:

The minimum spanning tree in a connected graph is unique if all weights are distinct.

(Hint: Assume there are more than one MST in the graph. Then, two distinct MSTs $T_1$, $T_2$ may share some edges, and the other edges either only exist in $T_1$ or only in $T_2$. Among those edges only in $T_1$ or only in $T_2$, we find the edge with smallest weight, denoted as $e_x$. Without loss of generality, let’s say $e_x \in T_1$. Try add $e_x$ to $T_2$ and see what happens.)
(Hint 2: simply saying the Prim’s or Kruskal’s algorithm will generate only one result does not prove this problem. Those algorithms generate ONE minimum spanning tree. They do not enumerate all possible minimum spanning trees of a given graph.)

In fact, the hint did almost everything except the last step in the proof. Following the hint, after $e_x$ is added to $T_2$, a new cycle must have been formed since before the insertion $T_2$ was a tree. In the cycle containing $e_x$, at least one edge must be in $T_2$ but not in $T_1$ (if all other edges in the cycle are in $T_1$, the same cycle must exist in $T_1$ as well since $e_x \in T_1$), say $e_y$. Because $e_x$ is the one with smallest weight among all edges only in $T_1$ or only in $T_2$, $w(e_x) < w(e_y)$ ($w(e)$ denotes the weight of the edge). Then, if we remove $e_y$ from the cycle, $T_2$ becomes another spanning tree with smaller total weight, which contradicts the fact that $T_2$ as a minimum spanning tree. This completes the proof by contradiction.

7. (10pts) In the previous proof, what happens if not all weights are distinct? What part of the proof does not hold any more?

If not all weights are distinct, it is not guaranteed that $w(e_x) < w(e_y)$, and removing $e_y$ may not generate a spanning tree with smaller weight, therefore the above proof does not hold.