CSE 101 Exam 1

Spring 2016

Instructions: Do not open until the exam starts. The exam will run for 45 minutes. The problems are roughly sorted in increasing order of difficulty. Answer all questions completely (though pay attention to exactly what the question is asking for). You are free to make use of any result in the textbook or proved in class. You may use up to 6 1-sided pages of notes, and may not use the textbook nor any electronic aids. Write your solutions in the space provided, the pages at the end of this handout, or on the scratch paper provided (be sure to label it with your name). If you have solutions written anywhere other than the provided space be sure to indicate where they are to be found.

If the problem asks for an algorithm, giving a correct algorithm with worse runtime efficiency than what is asked for will be awarded partial credit.

Name:

ID Number:

Discussion Section:

Seat:

<table>
<thead>
<tr>
<th>Problem</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Question 1** (Pre and Post Orderings, 30 points). *Give the pre- and post- orders obtained for all vertices when DFS is run on the directed graph below. Assume that whenever the algorithm has a choice of which vertex to explore next, it always picks the alphabetically first one.*
Question 2 (Closest Fire Station, 35 points). The layout of Tindertown is given by an undirected, unweighted graph $G = (V, E)$ (all of the roads in the city are of exactly the same length). A subset $F \subseteq V$ of these intersections have fire stations. The town council wants to better understand fire response times, and thus wants to know for every vertex $v \in V$ what the distance to the closest fire station is. Provide an algorithm that given $G$ and $F$ computes for every vertex $v \in V$ the distance to the closest element of $F$. For full credit, your algorithm should run in time $O(|V| + |E|)$ or better.
Question 3 (Air Travel, 35 points). The nation of Graphania has cities given by a set $V$. These cities are connected by two kinds of (undirected) edges (forming a set $E$), roads and air routes. Give an algorithm that given $V$ and $E$, with edges labelled as either roads or air routes, determines whether or not it is the case that every pair of cities have a path between them with at most one air route (and any number of roads). For example, your algorithm should say ‘yes’ if the graph is connected even without the air routes, and should say ‘no’ if there is any pair of cities that requires at least 2 air routes to get between. For full credit, your algorithm should run in time $O(|V| + |E|)$ or better.