Fall 2018, CMPSC 465: Final Exam (practice).
Closed book and closed notes, two 'cheat sheets' allowed, no calculators allowed.
Please don't use cell phones during the exam.
Answer questions in the space provided.
The exam is for 75 points.

NAME: ____________________ SECTION: __________

1. A contiguous subsequence of a list $S$ is a subsequence made up of consecutive elements of $S$. Give a linear-time dynamic programming algorithm for the task of determining a contiguous subsequence of maximum sum. Assume that a subsequence of length zero has sum zero.
(**10 points**)

2. Show the longest common subsequence (LCS) table for the following two strings:
   X: EXAMINE and Y: ELIMINATE.
   What is the length of the LCS?
   (**7 points**)

3. Suppose you are given an instance of the fractional knapsack problem in which all items have the same weight. Show how you can solve the problem in this case in $\mathrm{O}(n)$ time.
(**7 points**)

4. Prove that the following statement is correct or give a counter-example if it is incorrect: If $e$ is part of some MST of $G$, then it must be a lightest edge across some cut of $G$. Assume that the graph $G$ is undirected and connected.
**(7 points)**

5. Using appropriate notation, define the traveling salesman problem. Explain the distinction between the Minimum Spanning Tree (MST) problem and the traveling salesman problem. (**7 points**)

6. Show that, if $c$ is a positive real number, then $g(n) = 1 + c + c^2 + \cdots + c^n$ is $\Theta(1)$ if $c < 1$, $\Theta(n)$ if $c = 1$, and $\Theta(c^n)$ if $c > 1$.
(**6 points**)

7. Given a sorted array of distinct integers $A[1,\ldots,n]$, you want to find out whether there is an index $i$ for which $A[i] = i$. Give a divide-and-conquer algorithm that runs in time $O(\log n)$. (**10 points**)

8. Suppose that we have numbers between 1 and 1000 in a binary search tree, and we want to search for the number 363. Which of the following sequences could not be the sequence of nodes examined?

   (a) 2, 252, 401, 398, 330, 344, 397, 363.

   (b) 924, 220, 911, 244, 898, 258, 362, 363.

   (c) 925, 202, 911, 240, 912, 245, 363.

   (**7 points**)

9. Given a directed acyclic graph $G$, give a linear time algorithm to determine if $G$ contains a directed path that touches every vertex exactly once.
(**7 points**)

10. You are given a strongly connected directed graph $G = (V, E)$ with positive edge weights, along with a particular node $v_0 \in V$. Give an efficient algorithm for finding shortest paths between all pairs of nodes, with the one restriction that these paths must all pass through $v_0$.
**(7 points)**