Illinois Institute of Technology
Department of Computer Science

Third Examination

CS 430 Introduction to Algorithms
Spring, 2012

8am–10am, Tuesday, May 1, 2012
104 Stuart Building

Print your name and student ID, *neatly* in the space provided below; print your name at the upper right corner of *every* page. Please print legibly.

Name:	
Student ID:	

This is an *open book* exam. You are permitted to use the textbook, any class handouts, anything posted on the web page, any of your own assignments, and anything in your own handwriting. Foreign students may use a dictionary. *Nothing else is permitted*: No calculators, laptops, cell phones, Ipods, Ipads, etc.!

Do all five problems in this booklet. *All problems are equally weighted, so do not spend too much time on any one question.*

Show your work! You will not get partial credit if the grader cannot figure out how you arrived at your answer.

Question	Points	Score	Grader
1	20		
2	20		
3	20		
4	20		
5	20		
Total	100		

1. **Two-Coloring Graphs**

   (a) Design an algorithm based on DFS (CLRS, page 604) to 2-color an undirected graph. That is, you must assign the color red or green to each vertex of the graph so that no two adjacent vertices are the same color. If the input graph cannot be so colored, your algorithm must indicate that.

   (b) Analyse the time required by your algorithm.

   (c) Explain why your algorithm will always succeed in coloring a tree (that is, an undirected, acyclic graph).

2. **Spanning Tree**

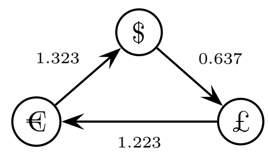
The "bottleneck" edge of a spanning tree is the edge of largest weight edge in the spanning tree. A *minimum bottleneck spanning tree* (MBST) is the spanning tree with the smallest bottleneck edge.

(a) Give an example of a graph in which an MBST is *not* an MST.

(b) Prove that any MST is also an MBST.
(*Hint*: Assume you have an MST which is not a MBST; what happens if you delete the costliest edge in the MST and replace it with a carefully chosen edge from the MBST?)

3. **Bellman-Ford Algorithm**

Foreign currency traders try to "arbitrage," that is to follow a sequence of currency exchanges that results in more money for you at the end than you had at the beginning. Assume trading among n different currency denominations. You can represent these exchanges with a directed graph in which the vertices correspond to currencies and the edges are weighted by the exchange rate between the respective currencies. Thus the edge connecting the US dollar to the British pound might have a weight of 0.637 because you can get 0.637 British pounds for one US dollar:



Assume that any two currencies are exchangeable (that is, there are directed edges from every vertex to every other vertex in the graph).

For example, with the above exchange rates, you could exchange \$100.00 for £63.70, which can in turn be changed to €77.91 and thence to \$103.07, making a profit of \$3.07.

(a) Explain how the Bellman-Ford algorithm can be applied to tell us whether such a sequence of money-making exchanges exists.

(*Hint*: Use a logarithmic transformation of the edge weights; remember, logarithms turn multiplcation into addition.)

(b) How would you modify your answer in part (a) if you were charged a 10% commission on each exchange?

4. **Longest Cycle in a Graph**

   The *longest cycle problem* is to determine the simple cycle (that is, cycle with no repeated vertices) in an undirected, unweighted graph of maximum length.

   (a) Formulate the longest cycle problem as a decision problem.

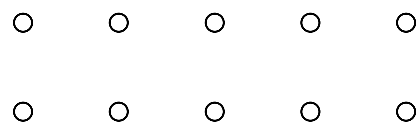
   (b) Prove that the decision problem is in NP.

   (c) Prove that the decision problem is NP-hard.

   (d) Prove that the decision problem is NP-complete.

5. **Approximation Algorithms**

In lecture on April 25 we discussed approximating the traveling salesman problem on a graph G that satisfies the triangle inequality by a traversal around a minimum spanning tree of G (CLRS, section 35.2.1; see Figure 35.2 on page 1113). A graph with $n = 2k$ vertices for which this method comes close to its worst behavior can be constructed as ladder: two parallel lines of k vertices each, adjacent vertices separated by one unit of distance. The parallel lines of vertices are separated by one unit of distance. Here, for example, are the vertices of the graph for $k = 5$:



The graph has all possible edges; define the distance between any pair of points as the "taxicab distance": two points (x_1, y_1) and (x_2, y_2) have distance $|x_1 - x_2| + |y_1 - y_2|$, the sum of the linear distances between them in each dimension.

(a) Prove that this "taxicab" distance satisfies the triangle inequality.

(b) Prove that the optimal TSP tour of this graph has cost $2k = n$.
(*Hint*: Remember, you need to prove that the *optimal* TSP has that cost.)

(c) Prove that the cost of the TSP tour discovered by the approximation algorithm can be as bad as $2n - 2$.
(*Hint*: Begin with a comb-like minimum spanning tree.)