Illinois Institute of Technology
Department of Computer Science

Third Examination

CS 430 Introduction to Algorithms
Spring, 2009

10:30am–12:30pm, Monday, May 11, 2009
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, 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. **BFS and DFS**

   We are given an undirected, connected graph $G = (V, E)$ and a specific vertex $u \in V$. Suppose that we compute a depth-first forest F rooted at u and a breadth-first tree T rooted at u and obtain the identical structure—that is, $F = T$. Prove that $G = T$.

2. **Shortest Paths**

   (a) Suppose we are given a weighted, undirected graph $G = (V, E)$ and a vertex $s \in V$. Assume that all edges weights are positive. Let P be a minimum weight path from s to t. Now we double each edge weight, replacing $w(e)$ by $2w(e)$, thereby creating a new shortest path problem with the same graph but different weights. Is P necessarily still a minimum-weight s–t path? If true give a proof; if false give a counterexample.

   (b) Answer part (a) if instead of doubling the edge weights we *square* them, replacing $w(e)$ by $w(e)^2$.

3. **Spanning Trees**

   Do parts (a), (b), and (d) of problem 23-1 on page 575 of CLRS; note that the parts of the problem can be solved independently. For extra credit, do part (c).

3. **Spanning Trees, continued.**

4. **NP-Hardness**

Given an $n \times m$ array P of pixels (a picture) and $u \times v$ pattern of pixels T, we say that the picture P contains the pattern T if there are two ascending sequences of indices, $r_1 < r_2 < \cdots < r_s$ and $c_1 < c_2 < \cdots < c_t$ such that $P[r_i, c_j] = T[i, j]$, for all $1 \leq i \leq s$, $1 \leq j \leq t$.

(a) Prove that the problem of determining whether a picture contains a pattern is in the class NP.

The *Balanced Complete Bipartite Subgraph* (BCBS) problem, known to be NP-complete, is: Given a bipartite graph (page 1083 of CLRS) $G = (V, E)$ and a positive integer $k \leq |V|$, do there exist two disjoint subsets V_1, V_2 of V such that $|V_1| = |V_2| = k$ and such that for every $v \in V_1$ and every $u \in V_2$, there is an edge $(u, v) \in E$.

(b) Prove that the problem of determining whether a picture contains a pattern is NP-hard by showing that if you could solve it in polynomial time, you could solve BCBS in polynomial time.

(*Hint*: Given a BCBS problem, consider the adjacency matrix of a the graph as a picture and let the pattern be a $k \times k$ array of 1s.)

4. **NP-Hardness, continued.**

5. **Approximation Algorithms**

   Problem 35.2-3 on page 1033 of CLRS introduces the *closest-point heuristic* for finding a traveling salesman tour.

   (a) Describe how to implement the algorithm and analyze the cost for an n-city problem.

   Consider the n-city problem in which the cost of going from city i to city j is

   $$C_{ij} = C_{ji} = \min(|j - i|, |n - j + i|).$$

   (Think of the cities as being around a circle with the distance from city i to city j being the shortest path of edges around the circle.)

   (b) Show these intercity distances satisfy the triangle inequality.

   (c) What is an optimal tour (and its cost) on these n cities? Prove it is optimal.

   (d) Show that the closest-point heuristic can yield a tour costing almost twice the cost of the optimal tour.

5. **Approximation Algorithms, continued.**