Illinois Institute of Technology
Department of Computer Science

Third Examination

CS 430 Introduction to Algorithms
Spring, 2017

Wednesday, April 26, 2017
10am–11:15am, 002 Herman Hall
11:25am–12:40pm, 104 Rettaliata Engineering Center

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. **Fibonacci Heaps**

   We decide that we do not want to bother with the mark bits in Fibonacci heaps, so we change the cascading-cut rule to cut a node from its parent as soon as it loses its *first* child. We want to prove that we still have $D(n) = O(\log n)$.

   (a) Prove an analogue of Lemma 19.1, namely

   New Lemma 19.1: Let x be any node in a (modified) Fibonacci heap with $\textsc{degree}(x) = k$, and children $y_1, y_2, \ldots, y_k$ (in the order in which they were linked to x, from earliest to latest). Then $\textsc{degree}(y_i) \geq i - 1$, for $1 \leq i \leq k$.

   (b) Prove an analogue of Lemma 19.4, namely

   New Lemma 19.4: Let x be any node in a (modified) Fibonacci heap. Then $\textsc{size}(x) \geq 2^{\textsc{degree}(x)}$.

   (c) Prove that the maximum degree $D(n)$ of an node in an n-node (modified) Fibonacci heap is $\lfloor \lg n \rfloor$.

2. **Topological Sort**

   Page 609 of CLRS3 gives the classification of edges as they are examined in depth-first search. Page 613 of CLRS3 gives an application of depth-first search which does topological sorting. Explain carefully the roles of each of the four edge types in that application.

3. **Shortest Path Verification**

   Your roommate has written a program to implement Dijkstra's shortest path algorithm (page 658 in CLRS3). Design and analyze a *linear time* algorithm to check your roommate's algorithm's results. That is, given a graph $G = (V, E)$ and your roommate's values of $v.d$ and $v.\pi$ for every vertex $v \in V$, your algorithm must verify their correctness or find a value that is wrong.

4. **SUBSET-SUM Reduction**

   In the proof of Theorem 34.15 (pages 1097–1100 in CLRS3), the target value used, for example, is 1114444 in Figure 34.19. Explain why a target values 1113333, 1112222, 1111111 are not adequate to prove the theorem.

5. **Polynomial-Time Reductions**

   Consider the two closely related problems:

   - HAM-PATH: Given an undirected graph G, determine whether G contains a simple *path* that visits every vertex exactly once.
   - HAM-CYCLE: Given an undirected graph G, determine whether G contains a simple *cycle* that visits every vertex exactly once.

   Describe polynomial-time reductions from HAM-PATH to HAM-CYCLE and from HAM-CYCLE to HAM-PATH.

   (*Hint*: You can add/delete edges/vertices to G and use, say, an algorithm for HAM-CYCLE a polynomial number of times to determine HAM-PATH.)