Illinois Institute of Technology
Department of Computer Science

Third Examination

CS 430 Introduction to Algorithms
Spring, 2018

Wednesday, April 25, 2018
10am–11:15am & 11:25am–12:40pm
111 Robert A. Pritzker Science 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 course 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.!

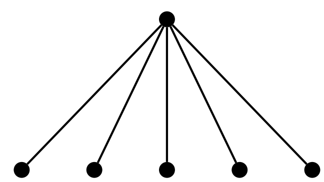
Do all five problems in this booklet. *All problems are equally weighted, so do not spend too much time on any one question.*

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**

   Consider the tree with a root and n children, each a singleton node; for example, when $n = 5$ we have the tree

   For what values of $n \geq 0$ can this shape of tree occur as a Fibonacci heap? In each case, either describe a sequence of heap commands that results in the shape or prove that the shape is impossible.

   (*Hint*: See Lemma 19.1 on page 523 of CLRS3.)

2. **DFS on a Directed Graph**

   DFS on a directed graph (page 609 in CLRS3) classifies edges as one of four types, tree edges, back edges, forward edges, and cross edges. The particular classification of any edge depends on the moment in the DFS when it is encountered, which depends on the order of vertices and edges in the adjacency structure, as well as the starting vertex of the DFS.

   (a) Can *any* edge be classified as a tree edge?

   (b) Can *any* edge be classified as a back edge?

   (c) Can *any* edge be classified as a forward edge?

   (d) Can *any* edge be classified as a cross edge?

   You must justify your yes/no answers.

   (*Hint*: See Figure 22.9 on page 616 of CLRS3.)

3. **Minimum Spanning Trees**

   Let G be a weighted undirected graph, with no two equal edge weights. Call an edge e *dangerous* if it is the longest edge on some cycle in G, and *useful* if it does not lie in any cycle in G.

   (a) Prove that the minimum spanning tree of G contains every useful edge.

   (b) Prove that the minimum spanning tree of G does not contain any dangerous edge.

4. **A proof that P = NP?**

   Given an arbitrary set S of n positive integers, the *partition* problem is to determine whether S can be split into two sets A and $\bar{A} = S - A$ such that $\sum_{x \in A} x = \sum_{x \in \bar{A}} x$. This problem is NP-complete (Exercise 34.5-5 on page 1101 in CLRS3). Suppose you discover an algorithm to solve this problem that runs in $O(n|S|)$ where $|S| = \sum_{x \in S} x$; would that imply that P = NP?

5. **NP-Completeness**

   An *independent set* in an undirected graph G is a subset of the vertices with no edges between them. The *independent set* problem is to determine the size of the largest independent set in a given graph.

   (a) Is this problem in the class NP?

   (b) Prove this problem is NP-hard.

   (*Hint*: Look at Figure 34.14 on page 1088 of CLRS3. Consider a variation in which each clause is a triangle connecting three vertices which correspond to the literals of that clause. What connections should there be between vertices in different triangles?)