Illinois Institute of Technology
Department of Computer Science

First Examination

CS 430 Introduction to Algorithms
Fall, 2010

11:25am–12:40pm, Wednesday, September 22, 2010
121 Life Sciences 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. **Time Bounds**

   Suppose you are given five algorithms with the following exact running times in terms of the input size n:

   (a) n^2
   (b) n^3
   (c) $100n$
   (d) $n\lceil \lg n \rceil$
   (e) 2^n

   How much *faster* do these algorithms become when you

   (a) Halve the size of the input?
   (b) Decrease the size of the input by 1?

2. **Insertion Sort**

   In analyzing INSERTION-SORT (page 18 in CLRS), the critical value analyzed in the notes of August 30 was $M = \sum_{j=1}^{n} d_j$, the number of "inversions" in the input (d_i is the number of elements to the left of $A[i]$ and larger than it). Give an $O(n \log n)$ *worst-case* algorithm to compute M from the input.

   (*Hint*: Modify MERGE-SORT on page 34 of CLRS.)

3. **Sorting Nuts and Bolts**

You are given n nuts and n bolts such that each nut fits exactly one bolt. Your only means of comparing these nuts and bolts is with

TEST(x, y): x is a nut and y is a bolt; returns either "nut is too big," "nut is too small," or "nut fits perfectly."

(a) Devise and analyze an $O(n^2)$ worst-case algorithm for matching the nuts and bolts.

(b) Devise and analyze an $O(n \log n)$ expected-time algorithm for the same problem.

4. **Heaps**

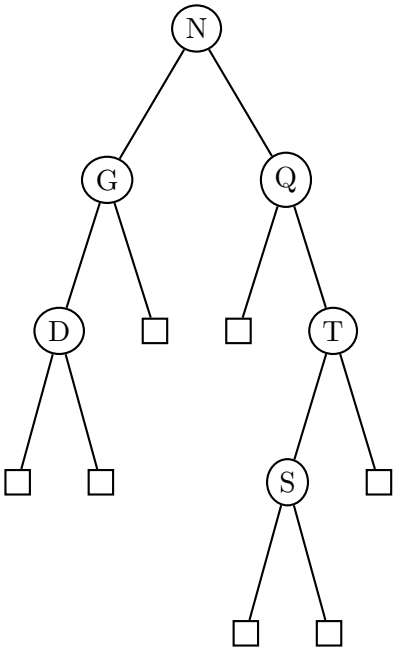
A *min-max heap* is a data structure that permits the retrieval of *both* the minimum *and* the maximum elements in a set in $O(1)$ time; the min-max heap can be constructed in $O(n)$ time, while INSERT, DELETEMAX and DELETEMIN take $O(\log n)$ time. In a min-max heap, the key of a node at *odd* level is no larger than the keys of any of its descendants, whereas the key of a node at *even* level is at least as large as the keys of any of its descendants; consider the root as level 1. Thus, the minimum element in the min-max heap appears at the root, whereas the maximum element appears as one of its two children.

Assuming that the min-max heap is stored in an array A in the same manner as an ordinary heap, give an $O(\log n)$ implementation of insertion into a min-max heap.

(*Hint*: The basic idea behind insertion is the same as in an ordinary heap.)

5. **Binary Search Trees**

Given the following binary search tree, in which internal nodes are shown as circles and external nodes (leaves; nil pointers) are shown as small boxes:



(a) Calculate the external path length.

(b) Is there a binary search tree on the same set of letters, {D, G, N, Q, S, T}, that has lower external path length? Either give such a tree or prove it does not exist.

(c) Can the nodes be colored red and black so as to form a proper red-black tree? If so, show how to color them; if not, prove it.