Algorithms CMSC-37000 Midterm Exam. February 19, 2009

Instructor: László Babai

Show all your work. **Do not use text, notes, or scrap paper.** When describing an algorithm in pseudocode, **explain the meaning of your variables** (in English). If you are not sure you understand a problem properly, **ask the proctor.** This midterm contributes 18% to your course grade.
TAKE THIS PROBLEM SHEET HOME and work on it after the test.

1. (20 points) **(Skip or zero)** Let $A[1,\dots,n]$ be an array of integers. Suppose $A[1] = -1$ and $A[n] = 1$. Find i such that either $A[i] = 0$ or $A[i] \geq A[i-1] + 2$ $(2 \leq i \leq n)$. Each access to the array (query "$A[i] = ?$") costs 1 unit. Minimize the (worst-case) cost. Show that the cost is $O(\log n)$; determine the constant implicit in the big-Oh notation. Describe your algorithm in pseudocode, in full detail. Do not assume n is a power of 2; be careful about rounding. Name the method used.

2. (20+2B points) **(AVL trees)** (a) Let $M(h)$ be the number of nodes of the smallest AVL-tree of hight h. Recall that $M(h) = F_{h+3} - 1$, where F_n is the n-th Fibonacci number. (You do not need to prove this.) Prove that an AVL tree with n nodes has height $\lesssim c \log n$. Determine the smallest value of c for which this is true. Prove. (b) (BONUS) State the names of the AVL authors.

3. (35+2B = 3+10+8+8+6+2B points) **(An Ancient Algorithm)** (a) Name a polynomial-time algorithm published about 2,300 years ago. State the problem solved. (b) Describe the algorithm in pseudocode. (c) State the key loop-invariant that can be used to prove correctness. (d) Prove that the algorithm runs in polynomial time. (e) This algorithm used in the RSA scheme. For what purpose, how? (f) (BONUS) What is the title of the work where the algorithm appeared about 2,300 years ago?

4. (15 points) Prove that, given the positive integer n (in binary), the n-th Fibonacci number cannot be computed in polynomial time.

5. (25 = 5+20 points) (a) Define the relation $a_n \gtrsim b_n$. (b) Prove: if $b_n \to \infty$ and $a_n \gtrsim b_n^2 \ln b_n$ then $b_n \lesssim \sqrt{2a_n / \ln a_n}$.

6. (30 = 3+9+12+6 points) **(DYNAMIC-SELECTION)** Design a dynamic data structure that stores a list of reals ("keys") and upon receiving a postive integer k, returns the k-th smallest key (or NIL if $k > n$ where n is the current number of keys). The data structure must also support INSERT and DELETE requests. Each request should cost $O(\log n)$. Modify a data structure studied in class. (a) State the name of the relevant data structure from class. (b) State the additional information that needs to be stored at each node. (c) Describe DYNAMIC-SELECTION(k) in pseudocode (given the data structure, how do we select the k-th smallest key?). (d) Describe how to maintain the additional information under simple INSERT (no rebalancing).

7. (15 points) **(Huffman code)** Consider the probability distribution $p(A) = 0.07$, $p(B) = 0.08$, $p(C) = 0.09$, $p(D) = 0.11$, $p(E) = 0.15$, $p(F) = 0.5$ over the alphabet $\{A, B, C, D, E, F\}$. Construct an optimal prefix encoding of the alphabet with respect to this distribution. Perform Huffman's algorithm step-by-step. **Show every step!** State the (0,1)-string corresponding to each letter. Do not prove optimality.

8. (20 = 5+15 points) Consider the recurrent inequality $T(n) \leq T(n/5) + T(7n/10) + O(n)$. (a) What algorithm led to this recurrence? State the problem solved (input, output). (b) Prove: $T(n) = O(n)$. (Ignore rounding.)

9. (BONUS PROBLEM, 20 points) Let G be a weighted DAG (directed acyclic graph) with nonnegative weights. Find the cost of a <u>max cost</u> path from a vertex s to a vertex t ("critical path"). Your algorithm should run in linear time. Describe your algorithm in pseudocode. You may refer to a problem discussed in tutorial. Indicate why your algorithm is correct (gives the right answer).