Algorithms CMSC-37000 Midterm Exam. February 15, 2007

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). This midterm contributes 15% to your course grade. **TAKE THIS PROBLEM SHEET HOME** and work on it.

1. (4+4+4) For each of the following functions, decide whether or not they are polynomially bounded. Prove your answers. If your answer is "yes," state all correct exponents c in the $O(n^c)$ upper bound. (a) $n^2 \log n$; (b) $(\log n)^{\log\log n}$; (c) $5^{\log n}$.

2. (5+5+15 points) Let x, y, z be Boolean variables. The Boolean function PARITY(x, y, z) is defined to be $x + y + z$ mod 2. For instance, PARITY(1,1,1)=1 and PARITY(1,0,1)=0. Describe the PARITY function by (a) a DNF formula and (b) a CNF formula. (Recall: a DNF formula is an OR or ANDs; a CNF formula is an AND of ORs.) (c) Generalize PARITY to n Boolean valiables. Prove that the Booelan function PARITY$(x_1, \dots, x_n)$ can be represented by a Boolean formula of size $O(n)$. (We define the "size" of a Boolean formula as the number of symbols in it: AND, OR, negation, parenthesis, variable. Each variable counts as a single symbol, so when writing x_n, we don't charge $\log n$ for writing down the number n.) Hint: divide-and-conquer. Write a clear recurrence for the size and evaluate it.

3. (10+6+6)

   (a) Describe the "update" subroutines for Dijsktra's algorithm and for Jarník's (a.k.a. Prim's) algorithm.

   (b) Give an accurate definition of the problems solved by each of these algorithms (input, including the assumptions on the input, and an exact description of the output; make a clear distinction between directed and undirected graphs).

   (c) Name the three abstract data structure operations required to implement each of these algorithms.

4. (6 points) Name two significant computational tasks, discussed in class in full detail, each of which led to the recursive inequality $T(n) \leq 2T(n/2) + O(n)$.

5. (8+8 points) Consider the implementation, discussed in class, for the UNION-FIND data structure, which builds a directed tree in every "country," directed toward the "capital." We discussed two UNION strategies: (a) bigger wins; (b) deeper wins. Prove that under each of these strategies, the height of each tree remains $\leq \log n$ where n is the number of "cities."

6. (a) (10 points) Define the concept of a "loop invariant." Be as formal as reasonable. Make sure you give a clear definition of what kind of statement can be a candidate loop invariant. Include the definition of the domain and range of the predicates and transformations (functions) involved.

   (b) (5+5+5 points) Decide which of the following statements are loop-invariants for Dijkstra's algorithm. Refrain from giving artificial interpretations of these English statemenst; each has one natural interpretation. Reason your answers. (b1) All black vertices are accessible. (b2) All accessible vertices are black. (b3) All accessible vertices eventually become black.

7. (5+5+20 points) (a) Define the topology of AVL trees (no data, just nodes and links). (b) Draw the smallest AVL-tree of height 4. State the number of nodes. (c) Prove that an AVL tree with n nodes has heights $\lesssim c \log n$. Determine the smallest value of c for which this is true.

8. (14 points) Consider the Knapsack problem with integer weights (including the weight limit) and integer values. Does the dynamic programming solution studied in class solve this problem in polynomial time (as a function of the number of input bits)? Prove your answer.

9. (BONUS PROBLEM, 15 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.