Algorithms CMSC-37000 Midterm Exam. February 21, 2008

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 18% to your course grade. **TAKE THIS PROBLEM SHEET HOME** and work on it.

1. (15 points) **(Edit-distance)** We transform a word (string of characters) into another word using the following "edit" operations: delete, insert, replace. For instance, here is how to turn "NAIVE" into "FANATIC:"

   NAIVE - NAIVC - NAIC - NATIC - FNATIC - FANATIC.

   The sequence of operations was REP,DEL,INS,INS,INS. The *edit-distance* of two words is the minimum number of edit operations needed to turn one word into the other. (If the above sequence of operations is optimal, then the edit-distance of NAIVE and FANATIC is 5.)

   Describe an algorithm which finds the edit-distance of two given words in $O(km)$ steps where k and m are the respective lengths of the two input words.

   Describe your algorithm in pseudocode. It should be very simple, no more than a few lines. Name the algorithmic technique used. **Define the meaning of your variables.** Half the credit goes for the clear definition (the "brain" of your algorithm).

2. ($3 + 10$ points) **(Strassen's matrix multiplication)** Strassen reduced the multiplication of two $n \times n$ matrices to multiplication of 7 pairs of $n/2 \times n/2$ matrices. The cost of the reduction is $O(n^2)$ (it involves computing certain linear combinations of matrices of these dimensions and bookkeeping). (a) State the recurrence for the complexity $T(n)$ of this algorithm (number of arithmetic operations with real numbers). (b) Prove that $T(n) = O(n^\alpha)$ where $\alpha = \log 7 \approx 2.81$.

3. (16 points) **(RANGE-SELECTION)** Design a dynamic data structure that stores a list of reals ("keys") and upon receiving a pair (a, k) where a is real and k is a positive integer, returns the k-th smallest among all keys $\geq a$ at cost $O(\log n)$ where n is the current number

of data. In additon to these "RANGE-SELECTION" requests, the data structure should also serve INSERT and DELETE requests. All requests should cost $O(\log n)$ where n is the current number of data. *Hint.* Modify a data structure studied in class. Clearly state the name of the data structure, the additional information that one needs to store, and describe how to update the information in $O(\log n)$.

4. (9 points) **(Trial division)** We determine whether or not a given integer $x \geq 3$ is prime by the following algorithm:

   $k := 2$
   while k does not divide x and $k \leq \sqrt{x} - 1$
       $k := k + 1$
   endwhile
   if k divides x then **return** "COMPOSITE"
       else return "PRIME"
   endif
   end

   Prove that this algorithm is not polynomial time.

5. (8+5 points) **(Min-cost)** Recall that Dijsktra's algorithm solves the "min-cost path" problem and Jarník's (a.k.a. Prim's) algorithm solves the min-cost spanning tree problem.

   (a) 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).

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

6. (16 points) **(Recognizing a DAG)** Given a digraph, decide in linear time whether or not it is a DAG. Write your algorithm in elegant pseudocode. Make sure you define all your variables.

7. (8 points) **(An important recurrece)** Name two significant and completely different computational tasks, discussed in class in full detail, each of which led to the recurrent inequality $T(n) \leq 2T(n/2) + O(n)$.

WARNING: The bonus problems are underrated.

8. (BONUS PROBLEM, 6 points) **(Evenly splitting the coins)** Suppose we have $2n$ coins, some of which are fake. All the fake coins have equal weight, and they are lighter than the true coins (which also have equal weight). Using $O(\log n)$ measurements on a balance, divide the coins into two sets of n coins of nearly equal weight. "Nearly equal" means if the number of fake coins is even, they must be evenly split between the two parts; if their number is odd, one part must have exactly one more fake coin than the other part. You don't have to write pseudocode but clarity of the explanation matters.

9. (BONUS PROBLEM, 6 points) **(Wrong pivot)** Given n data (real numbers), Mr. Starbuck wishes to select a "pivot" as a first step toward finding the median. Ahab suggests the following algorithm: divide the data into groups of 3, select the median of each, and repeat the process with the $n/3$ data obtained. Queequeg warns Ahab that his output may not be among the middle 98% of the data. Prove that Queequeg is right. (For one more bonus point, name the continent on which this dialog may have occurred.)

10. (BONUS PROBLEM, 8 points) **(Hashing)** Consider the following hashing scheme. Let the universe U consist of all n-digit positive integers. Select at random a prime number $p < n^4$. Let $H(x) = (x \bmod p)$. (So $0 \le H(x) \le p - 1$.) Given n data (elements from the universe), prove that the expected number of collisions among them approaches zero as $n \to \infty$. (The data are chosen by an adversary who knows our method and wants to maximize the chance of collision but has no information about the prime number chosen (other than that $p < n^4$). (Note that in this problem, n-digit numbers are hashed down to $O(\log n)$-digit numbers.)