Algorithms CMSC-37000 Midterm Exam. February 18, 2010

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. (5+5+14 points) **(AVL trees)** (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 height $\lesssim c \log n$. Determine the smallest value of c for which this is true.

2. (10 + 10B points) **(Fibonacci)** (a) Prove: the n-th Fibonacci number F_n cannot be computed in polynomial time. (The input is the number n written in binary.) (b) (BONUS) Prove that $(F_n \bmod m)$ can be computed in polynomial time. (Both n and m are given in binary.)

3. (24 points) **(Cutting up a segment)** We are given an $n \times n$ array of real "costs" $C[i,j]$ $(1 \le i,j \le n)$. We want to divide the set $\{1,\dots,n\}$ into segments of minimum total cost, i. e., we want to find values $1 = i_0 < i_1 < \cdots < i_k = n$ so as to minimize the sum $\sum_{j=0}^{k-1} C[i_j, i_{j+1}]$. ($k \ge 1$, the number of segments, is not given in advance.) Compute this minimum sum in $O(n^2)$ steps (arithmetic and comparison of reals, bookkeeping). Describe your algorithm in pseudocode. Name the method used. Half the credit goes for the "brain" of the algorithm.

4. (28 points) **(Range-sum)** Design a dynamic data structure that stores a list of reals ("keys") and upon receiving a pair (a,b) of reals, $a < b$, returns the sum of all keys x in the interval $a \le x < b$. In additon to these "RANGE-SUM" 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. The cost is the number of arithmetic operations and comparisons of reals plus bookkeeping (pointer operations). *Instructions.* Modify a data structure studied in class. Clearly state the name of the data structure and the additional information that one needs to store. Describe how to update the information in $O(\log n)$ after an INSERT without rebalancing.

5. (8+6+10 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. These two algorithms are very similar.

   (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. State how many times each data structure operation is invoked.

   (c) Describe the UPDATE(x, y) operation in pseudocode for each algorithm (here (x, y) is an edge).

6. (8 points) **(An important recurrece)** Name two significant and unrelated computational tasks, discussed in class in full detail, each of which led to the recurrent inequality $T(n) \leq 2T(n/2) + O(n)$.

7. (2 + 12 points) A **divide-and-conquer** algorithm reduces an instance of size n to two instances of size $n/2$. The cost of the reduction is $O(\sqrt{n})$. (a) State the resulting recurrence for the function $R(n)$, the cost of the algorithm on the worst input of size n. (b) Prove: $R(n) = O(n)$. (You may assume that n is a power of 2.)

8. (4+14+8 points) **(Heapify)** (a) Describe the array implementation of a heap. (b) Describe in elegant pseudocode how to heapify an array of n data. Your input is an array $A[1 \dots n]$; the output should be a heap, implemented as an array $H[1 \dots n]$. Do not assume any "subroutines;" if you need one, describe it in pseudocode. Your algorithm should run in $O(n)$. (c) Prove that your algorithm makes $O(n)$ comparisons.

9. (6+14 points) **(Union-Find)** (a) Describe what kind of data is being maintained by a UNION-FIND data structure and what are the queries served. (b) Prove that under the hierarchical implementation with the "deepest wins" rule, the depth of each tree is $\leq \log n$ (base-2 log) where n is the number of cities.

10. (BONUS PROBLEM, 8 points) **(Wrong pivot)** Given n data (real numbers), Mr. Starbuck wishes to select a "pivot" as a first step toward finding the median. Ahab proposes 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, etc., until we end up with a single item. Queequeg warns Ahab that this final item may not be among the middle 98% of the data. Prove that Queequeg is right. (For one more bonus point, name the boat on which this conversation might have occurred.)