Algorithms CMSC-37000 First Quiz. January 14, 2010

Instructor: László Babai

Name: ______________________________

Show all your work. **Do not use book, notes, or scrap paper.** Write your answers in the space provided and **continue on the reverse** if necessary. When describing an algorithm in pseudocode, **explain the meaning of your variables** (in English). WARNING: The bonus problems are underrated. Do the ordinary problems first. – This quiz contributes 6% to your course grade.

1. (6+10 points) A divide-and-conquer algorithm reduces an instance of size n to four instances of size $n/3$ each. The cost of the reduction is $O(n)$. Let $T(n)$ denote the cost of the algorithm on the worst instance of size n. (a) State the recurrent inequality for $T(n)$ that follows from such a reduction. (b) Use the method of reverse inequalities to prove that $T(n) = O(n^\beta)$; determine the best possible β achievable based on the information given. (Assume $n = 3^k$.)

2. (6 points) Give a simple description of all sequences $\{a_n\}$ that are asymptotically equal to the $0, 0, \dots$ ("all-zero") sequence. Do not prove.

3. (12 points) Describe in elegant pseudocode the DECREASE-KEY(x, *newkey*) operation in a heap. (x is the name of a node; *newkey* is the value with which we replace key(x). We assume *newkey* $<$ key(x).)

4. (12 points) Given a sorted array $A[1..n]$ of n real numbers $A[1] \leq A[2] \leq \cdots \leq A[n]$ and a real number x, decide whether or not x is in the array. Use the minimum possible number of comparisons. (Comparisons are binary; they answer questions of the form "is $y \leq z$?") Write your algorithm in **pseudocode**. State the name of the algorithm. Do **not** assume that n is a power of 2; be careful about **rounding**. Do not analyze the algorithm.

5. (14 points) Let a_n, b_n be sequences of reals. Recall that we say that $a_n \gtrsim b_n$ if $a_n \sim \max\{a_n, b_n\}$. Assume $a_n \gtrsim b_n$. Prove that it does **not** follow that $2^{a_n} = \Omega(2^{b_n})$.

6. (BONUS PROBLEM, 8 points) (How helpful is a heap?) Andrea designed an algorithm that takes as input n data (real numbers) arranged in a heap, performs at most $t(n)$ comparisons on the data, and returns the sorted list of the data. Prove that $t(n) \gtrsim n \log_2 n$. Warning: this is NOT a question about Heapsort! Andrea's algorithm has random access to the data; it can ignore the heap or rely on comparisons implied by the heap as needed. The only type of operations Andrea's algorithm is permitted to do on the data are comparisons. Bookkeeping is free. (This includes copying, comparing, and doing arithmetic with addresses, recording the outcomes of previous comparisons, etc.)