Algorithms CMSC-37000 Second Quiz. January 23, 2007
Instructor: László Babai

Name: ________________________________

Show all your work. **Do not use book, notes, or scrap paper.** Write your answers in the space provided. When describing an algorithm in pseudocode, **explain the meaning of your variables** (in English). This quiz contributes 6% to your course grade.

1. (5+7 points) We have n coins, one of which is fake. We want to tell, using three measurements on a balance, which coin is fake and whether it is heavier or lighter. Prove that this is impossible (a) if $n = 14$ (b) if $n = 13$. (A measurement on a balance has 3 possible outcomes: L, E, R: left-heavy, equal, right-heavy. Any number of coins can be placed in the trays of the balance.)

2. (7 points) Disprove the following statement: If a_n, b_n, c_n are sequences of positive reals such that $a_n \sim b_n + c_n$ then $a_n - b_n \sim c_n$. (Give a counterexample.)

3. (9 points) Given a (directed) graph by an array of adjacency lists, decide in linear time ($O(n+m)$) whether or not it is strongly connected. (A graph is strongly connected if for every pair v, w of vertices there exists a directed path from v to w.) You may refer to BFS and other algorithms discussed in detail in class, without reproducing their pseudocodes.

4. (9 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. Write your algorithm in **pseudocode**. State the name of the algorithm used. Do not assume that n is a power of 2; be careful about rounding.

5. (4+6 points) A divide-and-conquer algorithm reduces an instance of size n to 3 instances of size $n/4$. The cost of the reduction is $O(n)$. Let $T(n)$ denote the cost of the algorithm. (a) State the recurrent inequality for $T(n)$ that follows from such a reduction. (b) Use the method of reverse inequalities to prove that $T(n) = O(n)$. (Assume $n = 4^k$ and ignore rounding.)

6. (13 points) Given a linked list L of n integers between 1 and $3n$, we wish to create a linked list M consisting of the exact same integers but omitting duplicates. The original numbers should appear in M in the order of their first appearance in L. For instance, if $L = (5, 8, 5, 3, 8)$ then $M = (5, 8, 3)$. Solve this in $O(n)$ steps. Describe your algorithm as a numbered sequence of instructions (stated in English).