Algorithms CMSC-37000 First Quiz. January 24, 2008
Instructor: László Babai

Name: ______________________________________

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

1. (6 points) Given a sorted array $A[1..n]$ of n real numbers $A[1] \le A[2] \le \cdots \le A[n]$ 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 $y \le z$?") 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.

2. (2+6 points) (a) State the Knapsack Problem. Conciseness and clarity are paramount. (b) Assume there are n items; the weights are positive integers; and the weight limit is B. Solve the problem in $O(nB)$ steps. Describe the algorithm in pseudocode. **Define your variables.** A clear mathematical definition of your variables will account for half the credit.

3. (6+4B points) Let $\{a_n\}$ be a sequence of positive integers. Prove: (a) If $a_n! \gtrsim n$ then $a_n \gtrsim \ln n / \ln\ln n$. (b) BONUS: Prove that the converse is false.

4. (2+6+2 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 $n = 3^k$ and ignore rounding.) (c) Prove that your β is best possible.

5. BONUS (4B + 5B) (Scheduling to minimize maximum lateness) We are given n "jobs" to be performed by a single processor. Job i is characterized by its time length $t_i > 0$ and deadline d_i. A *schedule* is a sequence of start times $s_1, \ldots, s_n$ such that job i starts at time s_i and ends at time $s_i + t_i$; these intervals must not overlap. The *lateness* of job i is $L_i = \max\{0, s_i + t_i - d_i\}$. The *maximum lateness* of a given schedule is $L = \max\{L_1, \ldots, L_n\}$. All jobs are released at time zero, so we can start at time zero and proceed with the jobs in any order. We wish to minimize L. (a) Given the real numbers $t_1, \ldots t_n > 0$ and $d_1, \ldots, d_n$, find an optimal schedule by an efficient algorithm. The algorithm is allowed to sort data but otherwise must have linear ($O(n)$) cost. Arithmetic operations and comparison of real numbers cost one unit. (b) Indicate why your algorithm produces an optimal schedule.