Algorithms CMSC-37000 Third Quiz. February 1, 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. (6 points) We have n keys (real numbers) arranged in a heap. The heap is implemented as an array to which we have random access. We wish to sort the keys. Prove that this still requires $\gtrsim n \log n$ comparisons (i. e., the information already encoded in the heap does not help). (NOTE: this is NOT about Heapsort; your lower bound should stand against every possible algorithm.)

2. (6 points) Recall that Dijstra's algorithm solves the single-source min-cost paths problem and Jarník's (a.k.a. Prim's) algorithm solves the min spanning tree problem. List the 4 differences in what kind of input each algorithm accepts. (Lose 2 points per mistake.)

3. (9 points) Let us divide a list L of $n = 5k$ distinct real numbers into k groups of 5 numbers each. Assume k is odd. Let us put the median of each group of 5 on a list M, so M is a list of k numbers. Let x denote the median of M. Let r be the rank of x on the list L. Prove that x is in the middle 40% of L, i. e., $3n/10 < r < 7n/10$.

4. (4+3+(3+3)+8 points) (a) List the 3 types of data operations a data structure needs to serve in order to support an implementation of Dijkstra's algorithm. (Lose 2 points per mistake.) (b) State the number of times each operation is invoked for a graph with n vertices and m edges. (Assume $n \geq 2$, $m \geq 1$.) (c) State the total cost in (c1) the naive array implementation of the priority queue (c2) in the heap implementation. Give the simplest expression in each case. (d) Prove that for almost all graphs, the naive array implementation is more efficient. ("Almost all graphs" refers to the edges being chosen by independent coin flips and $n \to \infty$. "More efficient" means cost of the first method is little-oh of the cost of the second.)

5. (8 points) Define the concept of a "loop invariant." Name the loop relative to which you state your definition. Define the categories used in your definition (configuration, predicate over a set, transformation of a set). Watch your quantifiers. Make sure you specify the type of each variable used (example: "f is a legal 3-coloring of the graph G"). Other than in these specifications, your definition should symbols only, no English words (except for logical connectives).

6. (10 points, lose 4 per mistake) For each statement, decide whether or not it is a loop-invariant for BFS: (a) "Vertex #2 is black." (b) "Vertex #2 is white." (c) "Vertex #2 cannot change from black to white." Reason your "NO" answers.