COS 397 Spring 2010 Midterm Exam 2 *6 questions; 8 pgs.; 50 pts.; 50 min.* *2010-04-06*

Name: ____

1. (1 pt.) Unless otherwise noted, please **use the definitions and notational conventions described in class** (including class exercises and homework assignments) for all questions. You may refer to books, papers, and notes during the exam, but material may not be shared. You may use a computer and the Internet, *but only as a library for reference information*, and only if you do not disturb others. Ask for clarifications if needed.

   Write your name in the space provided near the top of this page.

2. (14 pts.) Depict the following graphs; depict the intermediate results as well.

   (a) $K_2 \oplus K_4$.

   (b) $(K_1 \oplus K_2 \times K_2)$—$(P_1 \oplus P_1 \,\Box\, P_1 \oplus P_1)$.

   (c) W_5—W_5.

[additional space for answering the earlier question]

3. (15 pts.)

   (a) Depict the *grid graphs* $G_{m,n}$ for all $0 \leq m \leq n \leq 4$.

   (b) Provide a *graph calculator* program that generates $G_{m,n}$ for fixed parameters m and n, or explain clearly why no such program exists. (The required program, or graph-calculator input, when consumed by the graph calculator described in the homework and in class, must print $G_{m,n}$ as output. The parameters m and n may be hard-coded by this program.) Try to make your program as concise as possible.

[additional space for answering the earlier question]

4. (10 pts.) Depict the *derivative* of each of the graphs of Question 1 (final graphs only, not intermediate results).

5. (10 pts.) Provide Java, C, or *detailed* pseudocode for an algorithm that computes the median of n objects in $O(n)$ *expected* time. The algorithm must be comparison-based (i.e., no ordering information about objects is available except that provided by a *compare(p,q)* function that returns -1, 0, or 1 if p is (respectively) less than, equal to, or greater than q). Provide a clear justification for the claimed running time.

6. (10 pts.) $\star$ **(extra credit)** Repeat Question 5 for an algorithm with $O(n)$ *worst case* time complexity. Hint: This question is significantly harder than the rest; do not attempt it at the cost of ignoring the others.

[additional space for answering the earlier question]