Information on the Final Exam

Date: Monday, June 7, 2010  
Time: 7:10 -10:00 p.m. (evening)  
Location: Bldg. 21, room 133  
Type: “closed book”

This is a 3 hour comprehensive exam on all topics covered in our course. You are expected to know the material exactly the way it was covered in class. Note that:

-- Any subject covered in class (in lectures, labs, and projects) can be included in the test.
-- There won’t be anything in the test that was not covered in a lecture, lab, or project.

Exam Topics: The material covered in the test consists of three portions:

1) All topics listed in the Midterm1 study sheet (the handout “Information on Midterm1”)
2) All topics listed in the Midterm2 study sheet (the handout “Information on Midterm2”)
3) All topics covered in last few weeks of the quarter that have not been tested yet, namely:
   (i) Analysis of Recursive Routines
   (ii) Advanced Sorting Algorithms
   (iii) Graphs and Graph algorithms

Below find details on the last portion of Exam Topics listed in item 3) above. For these topics I mention the chapter and the section in the textbook where you can find the related material. I recommend you to take as a guide the lecture handouts I provided for you since in the book there are certain things that we covered slightly differently.

(i) Analysis of Recursive Routines (use the handout “Running time of recursive routines”).
   Be able to solve the given recurrence equation/relation using either the telescoping method or the method of substitution as discussed in our class.

(ii) Advanced Sorting Algorithms (Chapter 7, sections 7.6 and 7.7 (exclude 7.7.6)):
   These are mergesort and quicksort

For mergesort
- Be able to implement (write Java code) different pieces of the algorithm: you should know the whole implementation.
- Be able to give the running time estimate of the algorithm.
- Be able to give the recurrence equation for the time function of this algorithm, as well as solve this equation to obtain the time function.

For quicksort
- Be able to implement (write Java code) different pieces of the algorithm: you should know the whole implementation.
- For a given list be able to show the content of the list at the end of setPivotToEnd method.
- For a given list be able to show the content of the list at the end of splitList method.
- Be able to give the running time estimate of the algorithm.
- Be able to identify best, worst, average cases and give running time estimates for each case.
- Be able to give recurrence equations for the time function of this algorithm in the best and worst cases; for each case be able to solve the equation to obtain the time function.

//Continued on the back
(iii) **Graphs and Graph Algorithms** (Chapter 9, sections 9.1, 9.2, and 9.3 (9.3.1 & 9.3.2 only)):

1. Know the main concepts and terminology of graphs; be able to answer questions on concepts discussed and used in our algorithms.

2. Know both representations of a graph (i) via adjacency matrix and (ii) via adjacency lists.

3. Be able to implement (write Java code) the Graph ADT for **both** representations: for each representation, be able to define instance variables, give the implementation for constructors and basic methods: `addEdge`, `deleteEdge`; additionally, be able to implement other operations appropriate for Graph ADT (e.g. `edgeCount`, `indegree`, `outdegree`, etc).

4. Know the Topological Sort algorithm:
   - be able to show the work of this algorithm on a given example.
   - know the implementation details of the algorithm
   - know the running time estimate of the efficient implementation of this algorithm.

5. Know the Shortest Path Algorithm for unweighted graphs:
   - be able to illustrate the work of this algorithm on a given example.
   - know the implementation details of the algorithm
   - know the running time estimate of the efficient implementation of this algorithm.

---

**Lecture handouts related to the material covered in Final Exam:**

1. All lecture handouts listed in “Information on Midterm #1” handout.

2. All lecture handouts listed in “Information on Midterm #2” handout.

3. The following lecture handouts on the material not covered in Midterms 1 and 2:
   - Running Time of Recursive Routines
   - Advanced Sorting Algorithms: Mergesort
   - Advance Sorting Algorithms (continued 1): Quicksort
   - Sorting: Conclusion
   - Graphs
   - Topological Sort
   - Topological Sort Algorithm (example)
   - Shortest Path Algorithm for Unweighted DiGraphs
   - Shortest Path Algorithm (example)