Final Exam Review Topics

The Final is 3 - 4:50 PM on Thursday (5/5). Approximately, 30% of the test will be comprehensive (chapters 1 - 4, Appendix B) and 70% will be from chapters 5 - 9.

The test will be closed-book, except for three 8.5”x11” sheets of paper containing notes (you may use both front and back of each sheet). You might want to read all of the test questions before answering any questions. This way you will be able to manage your time better.

Comprehensive Part (chapters 1-4, Appendix B):
1) Algorithm Design: You should understand and be able to apply divide-and-conquer, dynamic programming, and the greedy approach to “new”, simple problems.
2) Algorithm Analysis: You should be able to analyze recursive and non-recursive algorithms to determine their theta notation.

1) understand the general concept of backtracking and best-first search with branch-and-bound pruning problem-solving techniques
2) understand backtracking and best-first search with branch-and-bound pruning algorithms for problems discussed in class or on homework assignments (see list below)
3) be able to apply backtracking and best-first search with branch-and-bound pruning problem-solving techniques to "new" (i.e., not in the text or seen in class) problems

<table>
<thead>
<tr>
<th>Problem-Solving Technique</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backtracking</td>
<td>Coin-change problem</td>
</tr>
<tr>
<td></td>
<td>0-1 Knapsack Problem</td>
</tr>
<tr>
<td></td>
<td>Sum-of-Subsets Problem</td>
</tr>
<tr>
<td></td>
<td>Traveling-Salesperson</td>
</tr>
<tr>
<td>Branch-and-Bound</td>
<td>0-1 Knapsack Problem</td>
</tr>
<tr>
<td></td>
<td>Traveling-Salesperson</td>
</tr>
</tbody>
</table>

Chapter 7: Computational Complexity of Sorting (Mainly sections: 7.1, 7.8, 7.9)
An understanding of the computational complexity argument for sorting using comparison of elements. The results of the analysis.
Understanding of Radix sort (don't waste your time memorizing the code, but understand how it works and its analysis)

Chapter 8: Computational Complexity of Searching (Mainly sections: 8.1, 8.4)
An understanding of the worst-case, computational complexity argument for searching using comparison of elements. The results of the analysis.
Understanding of the general concept of hashing.

Chapter 9. Introduction of the Theory of NP (Mainly sections: 9.1, 9.2, 9.3, 9.4.1, 9.4.2, 9.5.1)
Categories of Problems: P, intractable, and NP
Meaning of NP -- decision problems, polynomial-time nondeterministic algorithm
Meaning and usefulness of NP-Complete concept -- polynomial-time many-one reducible
Existence of NP-Complete problems -- CNF-Satisfiability and Cook's Thm. (Theory 9.2)
Using algorithm reduction/transformation to show a problem NP-Complete
Meaning of NP-hard problems
Handling NP-hard problems -- Backtracking & Branch-and-bound, poly-time algorithm for a restricted subclass of the problem, approximation algorithms
Approximation algorithm for TSP problems satisfying the triangle-inequality