INSTRUCTOR:
John Bonomo
163 Hoyt  Phone: 7287
Class Time:  MWF 10:30-11:30

TEXT:
Data Structures and Problem Solving using Java, Weiss (4th edition)

POINT VALUES:
Exams:  
   Midterm-1 (around Feb. 23) 10%
   Midterm-2 (around Mar. 28) 10%
   Final (Tuesday, May 8, 8:00) 15%

     35%

Homeworks     8-10 assignments  35%
Projects       2-3                  30%

OBJECTIVE:
This course is designed to complete your study of both data structures and
algorithm design techniques. As with CS251, it is hoped that this course will
stimulate an interest in the more theoretical and aesthetic side of computer
science.

GOALS:
– learn the basic algorithm types and see examples of each
– learn the mechanics and the uses of the graph data structure, as well as
   several variants of previously studied data structures
– learn various algorithmic solutions to string matching, geometric and nu-
   merical problems
– learn some advanced algorithm analysis methods

OUTCOMES:
– students will be able to differentiate between various types of algorithms
  and select the appropriate ones to solve a given problem.
– students will be able to apply mathematical techniques in the analysis of
  algorithms.
– students will be able to identify and distinguish between various implemen-
  tations of the graph data structure.
– students will be able to select the appropriate data structures and use
  them to solve various programming problems.

All outcomes will be measured by specific, identified questions on homeworks
and exams.
GENERAL POLICIES:
The general policies for all my courses (covering grade cutoffs, late penalties, office hours, extra credit, etc) can be found at:

http://www.westminster.edu/staff/bonomojp/GP.html

COURSE OUTLINE: (tentative)

<table>
<thead>
<tr>
<th>WEEK</th>
<th>AREA</th>
<th>TOPIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/15</td>
<td>Algorithm Design</td>
<td>Review big oh; review divide &amp; conquer; backtracking; greedy algorithms</td>
</tr>
<tr>
<td>1/22</td>
<td></td>
<td>Dynamic programming</td>
</tr>
<tr>
<td>1/29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2/5</td>
<td>Graphs</td>
<td>Intro, defns, implementations, topological sorting, DFS/BFS, shortest path (unweighted, Dijkstra’s)</td>
</tr>
<tr>
<td>2/12</td>
<td></td>
<td>Shortest path (Bellman-Ford), all pairs</td>
</tr>
<tr>
<td>2/19</td>
<td></td>
<td>MCST, biconnected components</td>
</tr>
<tr>
<td>2/26</td>
<td></td>
<td>Strongly connected components</td>
</tr>
<tr>
<td>3/12</td>
<td></td>
<td>Network flow, bipartite matching</td>
</tr>
<tr>
<td>3/19</td>
<td></td>
<td>Min cost matching and applications</td>
</tr>
<tr>
<td>3/26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/2</td>
<td></td>
<td>Euler, Hamilton and bitonic cycles</td>
</tr>
<tr>
<td>4/9</td>
<td>String Matching</td>
<td>—</td>
</tr>
<tr>
<td>4/16</td>
<td>Geometric Algorithms</td>
<td>Defs, polygon problems, closest pair problem, convex hull</td>
</tr>
<tr>
<td>4/23</td>
<td></td>
<td>Intersecting segment problem, intersecting polygons, min triangularization</td>
</tr>
<tr>
<td>4/30</td>
<td>We’ll see</td>
<td>???</td>
</tr>
</tbody>
</table>