CS 385
Data Structures & Algorithms II
Fall 2012

Course Details
Instructor: Vivek Pathak
Web: http://www.cs.stevens.edu/~vpathak/
Email: vpathak+385@cs.stevens.edu
Office: Lieb 307
Office Hours: Thursday 2PM-3PM, or by appointment.
Telephone: (201) 216-5048

Classes: 11 – 11.50 AM, Babbio 110 (MRF)

Prerequisites
- CS 284 or CS 181
- Ability to program in Java.
- Knowledge of recursion and intermediate data structures like priority queues, heaps, and binary tree.

Objectives
This is a course on design and analysis of algorithms. It covers asymptotic complexity analysis, space-time tradeoffs, standard algorithm design techniques, and classic algorithms that serve as examples of design techniques. This course also develops C++ language programming skills by implementing graph algorithms, numerical algorithms, and complex search and sort algorithms. At the end of this course, students shall be able to:
- Calculate the asymptotic running time of standard algorithms.
- Explain the meaning of big Oh, Theta, and Omega notations and use them to reason about the performance of diverse algorithms.
- Use the Master Theorem to prove asymptotic assumptions.
- Implement standard algorithms using graphs and weighted graphs in C/C++. (e.g. DFS, BFS, MST)
- Compare and analyze basic and advanced sorting algorithms.
- Implement advanced search trees such as B tree, AVL tree, and 2-3 tree.
- Implement numerical algorithms such as Gauss elimination, binary exponentiation, and simplex method.
Texts


2. *Accelerated C++*
   Andrew Koenig and Barbara E. Moo
   ISBN 020170353X

Grading

Grading will be assigned according to the following scheme:

- Assignments : 4 with total weightage 40%.
- Midterm : 20%.
- Final : 30%.
- Participation, Contribution, and Attendance : 10%.

Outline

Tentative outline:

<table>
<thead>
<tr>
<th>WEEK</th>
<th>TOPICS</th>
<th>READING</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Introduction. Important problem types. Review of fundamental data structures.</td>
<td>1.1-1.4 Levitin</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Analysis framework; $O$, $\Theta$, $\Omega$ notations. Analysis of non recursive algorithms.</td>
<td>2.1-2.3 Levitin</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Analysis of recursive algorithms. Master theorem.</td>
<td>2.4-2.5 Levitin, App. B Levitin</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Brute force algorithms. Exhaustive search.</td>
<td>3.1-3.4 Levitin</td>
<td>Assignment #1</td>
</tr>
<tr>
<td>5.</td>
<td>Decrease and conquer algorithms. DFS, BFS, topological sort.</td>
<td>3.5-4.5 Levitin</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Divide and conquer algorithms.</td>
<td>5.1-5.5 Levitin</td>
<td>Assignment #2</td>
</tr>
<tr>
<td>7.</td>
<td>Tranform approach</td>
<td>6.1-6.6 Levitin</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Space-time tradeoffs: string matching, hashing, B-trees.</td>
<td>7.1-7.4 Levitin</td>
<td>Midterm</td>
</tr>
<tr>
<td></td>
<td>Topic</td>
<td>Pages</td>
<td>Assignment</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------</td>
<td>---------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>10.</td>
<td>Greedy technique. Prim’s algorithm, Kruskal’s algorithm, Dijkstra’s algorithm, and Huffman trees.</td>
<td>9.1-9.4 Levitin</td>
<td>Assignment #3</td>
</tr>
<tr>
<td>11.</td>
<td>Iterative algorithms. Simplex Method, Maximum flow through Ford Fulkerson method.</td>
<td>10.1-10.2 Levitin</td>
<td></td>
</tr>
</tbody>
</table>
| 12. | Lower bound arguments  
Decision Trees  
P, NP, and NP complete problems. | 11.1-11.3 Levitin    | Assignment #4 |
| 13. | Numerical algorithms. Backtracking, Branch-and-bound.                | 11.4-12.2 Levitin    |              |
| 14. | Review                                                               |                     |              |