CSC 415  Analysis of Algorithms  3 cr.

Catalog description:
Advanced concepts from data structures and general algorithms are discussed from both theoretical and applied viewpoints. Topics covered include: multi-lists, graph theory, searching and sorting algorithms, and general n-way tree structures. Techniques for analysis of algorithms for average and best/worst cases are presented. Laboratory work may involve programming in a high-level language. Three lecture hours per week.

Prerequisite: CSC 260, MAT 220, and at least one of the following: CSC 311, CSC 320, CSC 325, CSC 340, CSC 390.

Goals:
The aims of this course are:
CG01:  to present a coherent view of the algorithms encountered in earlier courses;
CG02:  to introduce a selection of more advanced algorithms;
CG03:  to present the most common tools and techniques for assessing the costs and complexity of these algorithms.

Objectives:
Upon completion of the course, the student will have demonstrated the ability to:
CO01:  perform best case, worst case, and average case analysis of selected algorithms for managing lists, trees, heaps, and hash tables;
CO02:  explain and analyze examples of greedy algorithms, such as Huffman codes;
CO03:  explain and use the main ideas of probabilistic and amortized analysis;
CO04:  explain the analysis and use of breadth-first and depth-first search, minimum spanning trees, and shortest-path algorithms.

<table>
<thead>
<tr>
<th></th>
<th>PO1</th>
<th>PO2</th>
<th>PO3</th>
<th>PO4</th>
<th>PO5</th>
<th>PO6</th>
<th>PO7</th>
<th>PO8</th>
<th>PO9</th>
<th>PO10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO01</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CO02</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CO03</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CO04</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Topics:
- brief review of course programming language (as appropriate)
- mathematical tools for algorithm analysis
  - limits and big-O notation
  - summation formulas
  - recursion relations
  - generating functions
- numeric algorithms (Fibonacci numbers, polynomial arithmetic, Fast Fourier Transform, etc.)
- analysis of searching and sorting algorithms
  - finding largest (or smallest), second largest, etc.
  - linear search, binary search
  - insertion sort, quicksort, merge sort, tree sort, heap sort
- multi-lists
- tree structures
  - general trees
  - n-way trees
  - balancing algorithms
  - traversal algorithms
  - expression trees
- review of basic hashing techniques
- advanced fast-access algorithms
- graphs, their representations, and traversal algorithms
- computational complexity, NP-completeness

DS3(3), DS4(3)  AL3(31)  AL6(3)
The primary emphasis of this course is on the understanding of a set of algorithms with wide-ranging applicability, together with their costs, advantages, and disadvantages. Each algorithm is presented and discussed in detail, including classic analysis to determine best, average, and worst-case performance. Implementation considerations will be touched on, along with proper solution development. Two or three algorithms may be assigned for implementation in programming projects, which will include analysis of time and space costs. (Possible programming assignments: creation and evaluation of expression trees; balancing algorithms; breadth-first vs. depth-first graph traversal; hashing.) There will be periodic written homework assignments on the major topics of the course.

The course grade will be determined using the following approximate weights: lab reports (if any) - 20%, written homework - 40%, final examination - 20%, other tests - 20%.

<table>
<thead>
<tr>
<th></th>
<th>Laboratory Reports</th>
<th>Written Homework</th>
<th>Tests</th>
<th>Final Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO01</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CO02</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CO03</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CO04</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Bibliography:

[Mathematical background]

Graham, Ronald L.; Knuth, Donald E.; Patashnik, Oren. Concrete Mathematics. A Foundation for Computer Science  Addison-Wesley, 1989

[Analysis of algorithms]