CSC/CPE 349 Design and Analysis of Algorithms

1. CSC/CPE 349 Design and Analysis of Algorithms

2. credit units  4   contact hours  6

3. Course Coordinator: Clint Staley


5. a. Course Description:  Intermediate and advanced algorithms and their analysis. Mathematical, geometrical, and graph algorithms. NP-complete problems. Additional topics will be chosen from pattern matching, file compression, cryptology, dynamic and linear programming, and exhaustive search. Course may be offered in classroom-based or online format. 3 lectures, 1 laboratory. Crosslisted as CPE/CSC 349.
   b. Prerequisite: CSC/CPE 103, with a grade of C- or better, and MATH 142; and STAT 312 or STAT 321 or STAT 350.
   c. Required/Elective/Selective Elective for CPE, CSC, EE, SE

<table>
<thead>
<tr>
<th></th>
<th>CSC</th>
<th>CPE</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Elective</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selective Elective</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

6. a. Course Goals/Outcomes
The student will be able to:
   • General
     o Understand the role of algorithms in computer science.
     o Become familiar with some classical algorithms of computer science.
     o Understand the meaning and significance of the problem classes P, NP, NP-complete and NP-hard and know examples of problems in each class.
     o Become familiar with some common parallel algorithms for several parallel machine models.
   • Design
     o Understand that there is no algorithm for designing algorithms, but there are heuristics.
     o Become skillful in designing correct algorithms with iteration.
Become skillful in designing correct algorithms by the heuristics of divide and conquer and recursion.
Become skillful in designing correct algorithms by the heuristics of memorized recursion and dynamic planning.
Recognize circumstances in which a greedy strategy can be correctly used.
Recognize the design strategies in (6) - (9) in given algorithms.
Recognize situations in which the design strategies in (6) - (9) may be successful.

- Analysis
  - Prove algorithms are correct using standard methods / recognize when they are not correct.
  - Analyze algorithms performance and express it in terms of BIG THETA, BIG O and BIG OMEGA notation.

b. How Student Outcomes addressed
   (“B” = Basic level, “I” = Intermediate level, “A” = Advanced level)

<table>
<thead>
<tr>
<th></th>
<th>3a</th>
<th>3b</th>
<th>3c</th>
<th>3d</th>
<th>3e</th>
<th>3f</th>
<th>3g</th>
<th>3h</th>
<th>3i</th>
<th>3j</th>
<th>3k</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>SE/</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td></td>
<td>I</td>
</tr>
<tr>
<td>CPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Major Topics Covered: (number of lecture hours each)
- Week 1-2 Intro to algorithms, role in Computer Science iterative programming heuristic (i.e. loops)
  - correctness and performance, review of O, Q, W notation (6)
- Week 3-4 Recursion and divide and conquer heuristic, correctness and performance, writing and solving recurrence relations for performance (6)
- Week 5-6 Memorized recursion and dynamic planning heuristic (3)
  - correctness and performance (3)
- Week 7 Greedy heuristic (1)
  - sufficient conditions for it to work(exchanges / matroids) (1)
  - correctness and performance (1)
- Week 8-9 The problem classes P, NP, NP-Complete (3)
- Week 9-10 Intro to parallel algorithms (3)