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: Hasmik Gharibyan

4. Textbook (and/or other required material): one of the following two textbooks:

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.

   b. Prerequisite: CSC 141 and MATH 142; or CSC 348 and MATH 142; or CPE/CSC 102 and CPE/CSC 103 and MATH 248; or CPE/CSC 202 and CPE/CSC 203 and MATH 248.

   c. Required/Elective/Selective Elective for CPE, CSC, EE, SE

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

6. a. Course Learning Objectives

   The student will be able to:
   • General
     o Explain the role of algorithms in computer science.
     o Restate some classical algorithms of computer science.
     o Explain the meaning and significance of the problem classes P, NP, NP-complete.
   • Analysis
     o Prove an algorithm’s correctness using standard methods
     o Analyze algorithm’s performance and express it’s running time via big-Oh, big-Theta, big-Omega notation.
   • Design
     o Design correct algorithms with iteration.
     o Design correct algorithms with recursion.
     o Design correct algorithms using the divide and conquer heuristic.
     o Design correct algorithms using the heuristics of dynamic programming and memoized recursion.
     o Design correct algorithms using the greedy heuristic.
     o Recognize the above-mentioned design strategies in given algorithms.
o Recognize conditions for the above-mentioned design strategies to be successful.

b. Level at which Student Outcomes are addressed
(“B” = Basic level, “I” = Intermediate level, “A” = Advanced level)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC</td>
<td>I</td>
<td>I</td>
<td></td>
<td></td>
<td>1</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>SE/CPE</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

7. Major Topics Covered: (number of lecture hours each)

- Introduction to algorithms, their properties, methods of representation, and their role in Computer Science. 1-2 hours
- Iterative programming; proof of correctness; performance analysis. 4 hours
- Recursion; proof of correctness; performance analysis, providing and solving recurrence relations. 3 hours
- Divide and conquer heuristic; solution of specific problems using this strategy. 3 hours
- Dynamic programming heuristics including “top-down memoized” and “bottom-up” approaches; conditions for these strategies to be efficient; solution of specific problems using this strategy. 5-6 hours
- Greedy heuristic; conditions for this strategy to work correctly; solution of specific problems using this strategy. 3 hours
- Classic graph algorithms; identifying design heuristics used in each of them. 4 hours
- The problem classes P, NP, NP-Complete. 3 hours