<table>
<thead>
<tr>
<th>Dept Number</th>
<th>CS 455</th>
<th>Course Title</th>
<th>Advanced Algorithm Design and Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester Hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Qiang Cheng</td>
</tr>
<tr>
<td>Catalog Description</td>
<td>An in-depth treatment of the design, analysis and complexity of algorithms with an emphasis on problem analysis and design techniques.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Textbooks**


**References**

- *Algorithm Design*, Tardos, Eva and Jon Kleinberg.

**Course Learning Outcomes**

- Deeper understanding of algorithm design.
- To learn the design techniques for efficient algorithms.
- To learn the methods for analyzing the complexity of the algorithms.
- To design algorithms with an emphasis on proving the correctness and proving the optimality in terms of time efficiency.
- To learn the basic concepts of NP-completeness and approximation algorithms.

**Assessment of the Contribution to Student Outcomes**

<table>
<thead>
<tr>
<th>Outcome</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessed</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Prerequisites by Topic**

CS 330 with a grade of C or better or graduate standing.
## Major Topics Covered in the Course

1. Mathematical preliminaries: principles and examples of algorithm analysis, recurrence relationships, worst case analysis {4 classes}
2. Asymptotically tight bounds: lower/upper bounds for finding minimum and sorting, lower bound analysis, growth rate of various functions {4 classes}
3. Divide-and-conquer: merge sort, quick sort, median selection, polynomial algorithms, and matrix algorithms, shortest distance, fast Fourier transform (FFT) {8 classes}
4. Greedy algorithms: elements of the greedy strategy, minimum spanning tree, shortest path, proof of optimality {5 classes}
5. Advanced graph algorithms: bi-connected components, strongly connected components, flow algorithms {5 classes}
6. Dynamic programming: optimal secondary structure prediction, optimal search trees, approximate string matching, Floyd's algorithm {6 classes}
7. NP-completeness and approximation algorithms {4 classes}
8. PRAM algorithms {4 classes}