CS3112 Course Syllabus  
Summer 2019

Session 1  
Lecture  Tue/Thu 12:30 PM – 1:45 PM at ET A309  
Lab Tue/Thu 2:00 PM – 3:15 PM at ET A309

Instructor  
Jungsoo (Sue) Lim  
Email: jlim34@calstatela.edu  
Office hours:  
1. Tue/Thu: 10:00 AM – 11:30 AM at E&T A310  
2. Tue/Thu: 3:30 PM – 4:30 PM at E&T A310  
3. By appointment

Course name  
Analysis of Algorithms

Credits  
3 units

Contact hours  
6 hours/week

Text book  
Required: Introduction to Algorithms 3rd addition  
by: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein  

Course Information  
Required: Introduction to Algorithms 3rd addition  
a) Abstract data types and their use in constructing algorithms for manipulating  
lists, trees, and graphs; analysis of algorithms for searching, sorting, and data  
structure manipulation.

b) Prerequisites: Math 208, Math 248, CS203.

c) This course is required in the BS program.

Course Goals  
The Student Learning Outcomes that are addressed by the course are:  
SLO #1. Students will be able to apply concepts and techniques from computing  
and mathematics to both theoretical and practical problems.

SLO #3. Students will have a strong foundation in the design, analysis, and  
application of many types of algorithms.

Other outcomes of instruction:  
- Analyze the correctness and computational complexity of computer algorithms.
- Design (specify and implement) efficient advanced Data Structures.
- Know advanced design techniques and their nontrivial application to classic  
problems of searching, sorting, graph optimization and combinatorial optimization.
Brief list of topics to be covered

- Mathematical Foundations (Summation Formulas, Logarithms, Induction, Lower and Upper bounds, Asymptotic Notation, Recurrence Relations, Master Theorem, Loop Invariants).
- Advanced Data Structures (Binary Search Trees, Balanced Trees, Heaps, Indirect Heaps, Priority Queues, Dictionaries, Hash Tables, Union-Find).
- Graph Algorithms and Searching and Sorting Algorithms.
- Design Techniques (Divide and Conquer, Greedy and Dynamic Programming).

Out of class Assignments

Each week students will have an assignment. For these assignments, students may be required to implement algorithms learned in class, solve rigorous problems, design algorithms based on the algorithms/data structures learned in class.

Quizzes

Each class will have in-class quiz.

Grading Policy

Attendance: 10%
Quizzes: 10%
Midterm 1: 20%
Midterm 2: 20%
Final: 40%

A-, A  90 – 100
B-, B, B+  80 – 89
C-, C, C+  65 – 79
D, F  <65

Academic Integrity

Cheating will not be tolerated.
Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of “F” for the course and are reported to the proper authorities.

ADA Statement

Reasonable accommodation will be provided to any student who is registered
With the Office of Students with Disabilities and requests needed accommodation.
## CS3112 Course Weekly Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Lecture topics</th>
<th>Exercise problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Role of Algorithm in Computing (Ch 1)</td>
<td>2.1-1 – 2.1-3</td>
</tr>
<tr>
<td></td>
<td>• What are algorithms?</td>
<td>2.2-1 – 2.2-4</td>
</tr>
<tr>
<td></td>
<td>• What is the role of algorithms in computer science?</td>
<td>2.3-1 – 2.3-6</td>
</tr>
<tr>
<td></td>
<td>Getting Started (Ch 2)</td>
<td>2-1</td>
</tr>
<tr>
<td></td>
<td>• Loop invariants – verify the correctness of algorithm</td>
<td>2-2</td>
</tr>
<tr>
<td></td>
<td>• Analyzing algorithms – best case, average case, and worst-case</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Order of growth rates</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Designing algorithms – divide-and-conquer approach</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Growth of Functions (Ch 3)</td>
<td>3.1-1 – 3.1-8</td>
</tr>
<tr>
<td></td>
<td>• Review – standard notations and common functions</td>
<td>3.2-1 – 3.2-3</td>
</tr>
<tr>
<td></td>
<td>• Asymptotic notation, functions, and running times – ( \Theta ) (theta) ( \Omega ) (big oh) ( \Omega ) (big – omega)</td>
<td>3-1 – 3-4</td>
</tr>
<tr>
<td></td>
<td>• Asymptotic notation in equations and inequalities</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Divide-and-Conquer and Master Theorem (Ch 4)</td>
<td>4.1-1 – 4.1-4</td>
</tr>
<tr>
<td></td>
<td>• Review of recurrences</td>
<td>4.2-1 – 4.2-7</td>
</tr>
<tr>
<td></td>
<td>• Maximum sub-array problem – solution using divide-and-conquer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Analysis of divide-and-conquer algorithm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Strassen’s algorithm for matrix multiplication</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Divide-and-Conquer and Master Theorem (Ch 4) - continued</td>
<td>4.3-1 – 4.3-3</td>
</tr>
<tr>
<td></td>
<td>• Substitution method for solving recurrences</td>
<td>4.3-6 – 4.3-9</td>
</tr>
<tr>
<td></td>
<td>• Recursion-tree method for solving recurrences</td>
<td>4.4-1 – 4.4-8</td>
</tr>
<tr>
<td></td>
<td>• Master method for solving recurrences</td>
<td>4.5-1 – 4.5-4</td>
</tr>
<tr>
<td></td>
<td>Midterm 1 (Ch 1 – Ch 4)</td>
<td>4-1, 4-3</td>
</tr>
<tr>
<td>4</td>
<td>Probabilistic Analysis and Randomized Algorithms (Ch 5)</td>
<td>5.1-1</td>
</tr>
<tr>
<td></td>
<td>• Review Counting and Probability – counting, probability distribution, expected value, random variable, geometric series, and binomial distribution.</td>
<td>5.2-1 – 5.2-4</td>
</tr>
<tr>
<td></td>
<td>• Hiring problem Overview</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Probabilistic Analysis and Randomized Algorithms (Ch 5) – continued</td>
<td>5.3-1 – 5.3-4</td>
</tr>
<tr>
<td></td>
<td>• Hiring problem – worst case analysis</td>
<td>5.3-6 – 5.3-7</td>
</tr>
<tr>
<td></td>
<td>• Hiring problem – probabilistic analysis</td>
<td>5.4-1 – 5.4-2</td>
</tr>
<tr>
<td>5</td>
<td>Heapsort (Ch 6)</td>
<td>6.1-1 – 6.1-7</td>
</tr>
<tr>
<td>---</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td>Heapsort algorithm</td>
<td>6.2-1 – 6.2-6</td>
</tr>
<tr>
<td></td>
<td>Heapsort analysis</td>
<td>6.3-1 – 6.3-3</td>
</tr>
<tr>
<td></td>
<td>Priority queues</td>
<td>6.4-1 – 6.4-4</td>
</tr>
<tr>
<td></td>
<td>Time complexity analysis</td>
<td>6.5-1 – 6.5-5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>Quicksort (Ch 7)</th>
<th>7.1-1 – 7.1-4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quicksort algorithm</td>
<td>7.2-1 – 7.2-5</td>
</tr>
<tr>
<td></td>
<td>Randomized version of quicksort</td>
<td>7.3-1 – 7.3-2</td>
</tr>
<tr>
<td></td>
<td>Quicksort analysis</td>
<td>7.4-1 – 7.4-4</td>
</tr>
<tr>
<td></td>
<td>Time complexity analysis</td>
<td>7.1 – 7.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7</th>
<th>Sorting in Linear Time (Ch 8)</th>
<th>8.1-1 – 8.1-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower bounds for comparison sorting</td>
<td>8.2-1 – 8.2-4</td>
</tr>
<tr>
<td></td>
<td>Counting sort</td>
<td>8.3-1 – 8.3-2</td>
</tr>
<tr>
<td></td>
<td>Radix sort</td>
<td>8.4-1 – 8.4-2</td>
</tr>
<tr>
<td></td>
<td>Bucket sort</td>
<td>8.2 – 8.4</td>
</tr>
</tbody>
</table>

**Midterm 2 (Ch 5 - 8)**

<table>
<thead>
<tr>
<th>8</th>
<th>Hash Tables (Ch 11)</th>
<th>11.1-1 – 11.1-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hash tables Overview</td>
<td>11.2-1 – 11.2-4</td>
</tr>
<tr>
<td></td>
<td>Collision resolution: Open addressing (linear probing, quadratic probing, and double hashing), separate chaining</td>
<td>11.4-1 – 11.4-3</td>
</tr>
<tr>
<td></td>
<td>Hash functions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Universal hashing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Perfect hashing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time complexity analysis</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9</th>
<th>Binary Search Trees (Ch 12)</th>
<th>12.1-1 – 12.1-5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overview of binary search tree</td>
<td>12.2-1 – 12.2-6</td>
</tr>
<tr>
<td></td>
<td>Querying a BST – search, find min/max, successor and predecessor</td>
<td>12.3-1 – 12.3-3</td>
</tr>
<tr>
<td></td>
<td>Insertion/deletion</td>
<td>13.1-1 – 13.1-4</td>
</tr>
<tr>
<td></td>
<td>Red-Black Trees (Ch13)</td>
<td>13.2-1 – 13.2-3</td>
</tr>
<tr>
<td></td>
<td>Red-Black Tree Overview</td>
<td>13.3-1 – 13.3-4</td>
</tr>
<tr>
<td></td>
<td>Properties of red-black trees</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rotations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Insertion algorithm</td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>Topic</td>
<td>Sections</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>9</td>
<td><strong>B-Tree (Ch 18)</strong></td>
<td>18.1-1 – 18.1-4</td>
</tr>
<tr>
<td></td>
<td>- Definition of B-Tree</td>
<td>18.2.1 – 18.2-3</td>
</tr>
<tr>
<td></td>
<td>- Basic operation on B-trees</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>Elementary Graph Algorithms (Ch 22)</strong></td>
<td>22.1-1 22.1-6</td>
</tr>
<tr>
<td></td>
<td>- Representations of graphs</td>
<td>22.2.1 – 22.2.6</td>
</tr>
<tr>
<td></td>
<td>- Breadth-first search</td>
<td>22.3.1 – 22.3-7</td>
</tr>
<tr>
<td></td>
<td>- Depth-first search</td>
<td>22.4-1 – 22.4-4</td>
</tr>
<tr>
<td></td>
<td>- Topological sort</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Strongly connected components</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>Minimum Spanning Trees (Ch 23)</strong></td>
<td>23.1-1 – 23.1-3</td>
</tr>
<tr>
<td></td>
<td>- Growing a minimum spanning tree</td>
<td>23.2.1 – 23.2-5</td>
</tr>
<tr>
<td></td>
<td>- Kruskal’s algorithm</td>
<td>23-1</td>
</tr>
<tr>
<td></td>
<td>- Prim’s algorithm</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Review for final exam</td>
<td></td>
</tr>
</tbody>
</table>

- Time complexity analysis