Syllabus
COEN 279 Computer Algorithm
Department of Computer Engineering
Santa Clara University

Dr. Tunghwa Wang                         Fall Quarter 2016
Phone: (408) 886-8460      Email address: twang1@cse.scu.edu
Course website:                       http://www.cse.scu.edu/~twang1
Office Hours:           Tuesday & Thursday 9:00-9:30AM

Course Description
Techniques of design and analysis of algorithms: proof of correctness; running
times of recursive algorithms; design strategies: brute-force, divide and
conquer, dynamic programming, branch-and-bound, backtracking, and
greedy technique; max flow/matching. Intractability: lower bounds; P, NP,
and NP-completeness. (4 units). Also listed as AMTH 377.

Prerequisites
Basic programming experience and data structures (COEN 12): Most chapters
for data structures will be skipped assuming such knowledge already learnt.

Required Textbooks
   E. Leiserson, Ronald L. Rivest, and Clifford Stein ISBN: 978-0-262-03384-
   8, The MIT Press 2009

References
1. Donald E. Knuth 2011 "The Art of Computer Programming", Vol. 1,
   and Searching, Vol. 4A Combinatorial Algorithms, Addison-Wesley
2. S. Sahni 1998 "Data Structures, Algorithms, and Applications in C++“
   McGraw-Hill
   Addison-Wesley
4. Narasimha Karumanchi 2016 "Data Structure and Algorithmic Thinking
   with Python“ CareerMonk Publications
5. Jens Clausen 1999 “Branch and Bound Algorithms – Principles and
   Examples”, University of Copenhagen PDF
   Design for Branch and Bound”, in H. J. Greenberg “Tutorials on Emerging

Methodologies and Applications in Operations Research” Kluwer Academic
Press PDF
7. David G. Sullivan 2012 “Recursion and Recursive Backtracking”
8. Richard E. Korf 1999 "Artificial intelligence search algorithms". In Atallah, Mikhail
10. Matteo Frigo, Steven G. Johnson 2005 “The Design and Implementation of
    FFTW3" Proceedings of the IEEE 93(2):216-231
    Multiplication” in Proceedings of the 21st Annual International Conference
    on Supercomputing Pages 284-292 June
12. Michael A. Bender, Erik D. Demaine, Martin Farach-Colton 2000 “Cache-
    oblivious B-trees” in Proceedings of the 41st Annual Symposium on Foundations
    of Computer Science page 399-409

Course Objectives
1. To learn and master advanced knowledge, design, analysis, and
   implementation in computing algorithms and underlying data structures.
2. To read and understand research publications in the technical area of
   computing algorithms, beyond that of the traditional textbook level.
3. To conduct group project and to equip for scholarly research in computing
   algorithms to solve intended application domain problem.
4. To develop capability to analyze, experiment, and then propose the best
   algorithm for any given application domain problem.

Expected Learning Outcomes
1. Demonstrate the knowledge of the basic algorithm design techniques and
   the impact on underlying data structure and vice versa.
2. Demonstrate the skill of algorithm analysis: proof of correctness, running
   time analysis, amortized analysis, time-space trade-off, and intractability:
   lower bounds, P, NP, and NP-completeness.
3. Demonstrate the knowledge of designing efficient algorithms and
   analyzing its computational complexity and time-space trade-off.
4. Demonstrate the ability to realize the best implementation of the designed
   algorithms for the given application domain.
5. Demonstrate the ability to read current research papers and implement
   example research group project in any given application domain.
**Grading Policy**

Course grade is determined based on the total score (maximum 1000 points + up to 200 optional bonus points for extra work) from:

1. Mid-term and final exams of 200 points each (close book with one A4 note, no sitting together, no wireless connection.) Makeup exams (must have a very good reason) are much difficult than normal exams.

2. Three programming assignments of 200 points each (late penalty: 30 points/day.) No makeup is allowed. You can call Design Center at 408/554-4909 for setup account or IT support, and ssh linux.scudc.scu.edu to work remotely.

3. A group (2-3 people in a team) programming term project of 300 points (late penalty: 60 points/day.) No makeup is allowed.

4. Bonus assignments will be assigned at each week with 20 points each. Due before next lecture begin by email to me with title “coen279 bN” (where N can be 2, 3, ..., 10) and cc to the grader. (You automatically get 20 points for “coen279 b1” when you are on the roster.) The solution for bonus assignments will be posted on my protected web page. Please read solutions of bonus assignments before asking questions. No late work accepted for bonus assignments.

**Table 1: Grade-score table**

<table>
<thead>
<tr>
<th>Score</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>950-900</td>
<td>A</td>
</tr>
<tr>
<td>850-800</td>
<td>A-</td>
</tr>
<tr>
<td>750-700</td>
<td>B+</td>
</tr>
<tr>
<td>650-600</td>
<td>B</td>
</tr>
<tr>
<td>599-0</td>
<td>C</td>
</tr>
<tr>
<td>-</td>
<td>F</td>
</tr>
</tbody>
</table>

**Course Schedule**

(Tuesday & Thursday 7:10pm-9:00pm)

<table>
<thead>
<tr>
<th>#</th>
<th>Week</th>
<th>Readings</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>09/20</td>
<td>Computational complexity, time-space trade-off, randomized algorithms</td>
<td>Ch. 2, 3, &amp; 5</td>
</tr>
<tr>
<td></td>
<td>09/22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>09/27</td>
<td>Divide-and-conquer</td>
<td>Ch. 4, 6, 7, &amp; 30</td>
</tr>
<tr>
<td></td>
<td>09/29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10/04</td>
<td>Tree algorithms</td>
<td>Ch. 12 &amp; 13</td>
</tr>
<tr>
<td></td>
<td>10/06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10/11</td>
<td>Dynamic programming</td>
<td>Ch. 15</td>
</tr>
<tr>
<td></td>
<td>10/13</td>
<td></td>
<td>program #1 due 10/13</td>
</tr>
<tr>
<td>5</td>
<td>10/18</td>
<td>Greedy algorithms</td>
<td>Ch. 16</td>
</tr>
<tr>
<td></td>
<td>10/20</td>
<td></td>
<td>mid-term exam 10/20</td>
</tr>
<tr>
<td>6</td>
<td>10/25</td>
<td>Greedy algorithms, Amortized</td>
<td>Ch. 16 &amp; 17</td>
</tr>
<tr>
<td></td>
<td>10/27</td>
<td>analysis</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>11/01</td>
<td>Graph algorithms</td>
<td>Ch. 22</td>
</tr>
<tr>
<td></td>
<td>11/03</td>
<td></td>
<td>program #2 due 11/01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>last day to withdraw</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11/04</td>
</tr>
<tr>
<td>8</td>
<td>11/08</td>
<td>Graph algorithms</td>
<td>Ch. 23, 26</td>
</tr>
<tr>
<td></td>
<td>11/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>11/15</td>
<td>P, NP, and NP-complete</td>
<td>Ch. 34</td>
</tr>
<tr>
<td></td>
<td>11/17</td>
<td></td>
<td>program #3 due 11/17</td>
</tr>
<tr>
<td>10</td>
<td>11/29</td>
<td>Approximation algorithms</td>
<td>Ch. 35</td>
</tr>
<tr>
<td></td>
<td>12/01</td>
<td></td>
<td>final exam 12/01</td>
</tr>
<tr>
<td>11</td>
<td>12/06</td>
<td>Review and summary</td>
<td></td>
</tr>
</tbody>
</table>

**Reminder**

- No cheating, and no register complaint without talking to me first.
- No incomplete. No sit-in or audit the class except formally registered.
- Read files under /home/twang1/tips for help.
- Handouts, assignments, and solutions will be posted on the web. You should check the class web site at least once a week (and don't forget to refresh the webpage to get the latest versions). You are responsible for printing and bring the handout to the class if you prefer printed pages.
- Office hours: Tuesday & Thursday 9:00AM-9:30AM.

**Honor Code**

All students taking course in the school of engineering agree, individually and collectively, they will neither give nor receive unpermitted aid in examinations or other course work that is to be used by the instructor as a basis of grading.

**Disability Accommodation Policy**:

To request academic accommodations for a disability, students must contact Disability Resources located in The Drahmann Center in Benson, room 214, (408) 554-4111; TTY (408) 554-5445. Students must provide documentation of a disability to Disability Resources prior to receiving accommodations.