Syllabus: Introduction to Analysis of Algorithms

CSE 102-02 (44895) Syllabus: Introduction to Analysis of Algorithms Feb. 10, 2020

Instructor: Allen Van Gelder, 355 Engr. II, 459-4611, avg@cs.ucsc.edu
Office Hours: Mon., Wed. 4:15–5:15, and appt./drop in
Teaching Asst.: Zekun Zhao, {zzhao99}@ucsc.edu

Errata linked in instructor's home directory.

Optional: (Excerpts may be used in lectures and posted in Handouts directory)
Kleinberg and Tardos Algorithm Design 2006
Cormen, Leiserson, Rivest, Stein, Introduction to Algorithms 2nd or 3rd ed.

Sedgewick, Algorithms in C.
Aho, Hopcroft, Ullman, Data Structures and Algorithms.
Roberts, Eric S., The Art and Science of C.

Optional and reference texts are on reserve at the library. Look before you leap.

Prerequisites: The official prerequisite for this class is: CSE 101. Working knowledge of (using old numbers) CMPE 16(or 16H), Math 19B(or 20B) and AMS 10 (or Math 21) is also expected, as these are pre-reqs for CSE 101.

Transfer students must have credit for CSE 101 approved by the School of Engr office, BE225.

Course Work: This course will have weekly problem sets. Evaluation will be based on a review quiz, problem sets, and 3 in-class exams. Some changes to the evaluation scheme might be announced early in the quarter. Students should plan to be present during the scheduled final-exam time unless it is announced otherwise.

Review Quiz 4%
Problem Sets 15%
Midterms and Quizes (in class) 81% (35%, 35%, 11%)

Students may be requested to show student-ids at exams and quizzes.

Policy on Academic Dishonesty: Any instance of academic dishonesty (cheating) is grounds for failing the course, regardless of how the student performed on other parts of the course. In general, academic dishonesty is the submission of any class work as though it were your own work, when in fact it is not. All assignments are to be done individually.

Plagiarism: Plagiarism constitutes academic dishonesty (see above), and will not be tolerated. Copying another person’s program, written assignment, exam answer, or any other work is plagiarism. Copying information from the internet and presenting it as your own work is plagiarism. Plagiarism is also considered unprofessional conduct by all computer professional societies, such as ACM and IEEE.

Principle of Disclosure: If in doubt, acknowledge and give credit for anything that you did not create yourself, and then you cannot be accused of plagiarism or dishonesty. If you disclose what you did (copying, got help, or whatever) in the work you turn in, then you clearly are not trying to conceal the facts. You might get a zero if you were not supposed to do that, but it’s an “honest zero”. If in doubt about any issue, ask the instructor.

File copying prohibited: In no case should you copy another student’s class-related computer files, nor should you permit another student to copy your class-related files. This is includes copying from the Internet or posting your code to a general-access site on the Internet. Files in the class locker, on the class “web” page, supplements to the main text, and so on, may be and often should be copied. Files elsewhere on the Internet in general may not be copied, because that will not help you achieve the learning goals of any assignment.

30 Minute Rule: After studying together in a group, do not take away notes on specific homework or program assignments, and wait at least 30 minutes before writing up or typing in your solutions. This helps to ensure that what you write reflects your understanding and not just your memory of what someone else told you. The same applies if the TA in section works on a specific homework problem. More often the TA will work on similar problems, not the exact homework.
**Syllabus: Introduction to Analysis of Algorithms**

**Online Information:** Bookmark the class URL: [https://classes.soe.ucsc.edu/cse102/Winter20-02](https://classes.soe.ucsc.edu/cse102/Winter20-02)

Its **Handouts** subdirectory has class-related documents, including this syllabus. **You are responsible to be aware of the class mailing list,** especially near deadlines, for announcements, questions and answers, mainly about assignments: cse102-w20-02@soe.ucsc.edu, cse102-w20-02-announce@soe.ucsc.edu.

With web browsers and other software, be sure to disable “threads,” so you see all the official messages.

**Reading:** I recommend that you read through the appropriate sections before lecture, and note areas where you have questions; be sure they get answered in class. Later, go back and read carefully, to ensure that you fully understand the material. The main text is intended to be understandable on its own. Optional texts should be on reserve at the library.

**Written Assignments:** will be turned in at class on paper **within the first 5 minutes,** or else directly to a TA outside class. Do not email any work to the instructor; it will wind up in spam.

**Tentative Schedule**

Topics marked **Review** were at least partially covered in CMPS 101 and/or math pre-requisites, and I will mainly cover newer material in class. You are expected to be already familiar with them, and/or read them on your own to brush up.

“B3rd” refers to the Baase and Van Gelder text, 3rd Ed. “K&T” refers to the Kleinberg and Tardos text. “CLRS” refers to the Cormen, Leiserson, Rivest, Stein text.

<table>
<thead>
<tr>
<th>Week</th>
<th>Chapter</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 J/06</td>
<td>B3rd 1.3, 1.5.1, K&amp;T 2.1, 2.2, 2.4, B3rd 6.7, K&amp;T 2.5</td>
<td>REVIEW QUIZ AND ATTENDANCE MON., Review: Math, Asymptotic Notation, Priority Queue ADT with Decrease Key, Pairing Forests.</td>
</tr>
<tr>
<td>2 J/13</td>
<td>B3rd 7, K&amp;T 3.1—3.6</td>
<td>Review: Depth First Search and Topological Sort, Strongly Connected Components, other DFS Applications.</td>
</tr>
<tr>
<td>3 J/20(H)</td>
<td>B3rd 10, K&amp;T 6.1—6.5</td>
<td>Dynamic Programming as Depth First Search. hw01 DUE FRI.</td>
</tr>
<tr>
<td>4 J/27</td>
<td>B3rd 8.1—4, 6.7, K&amp;T 4.1—4.5</td>
<td>More Dynamic Programming, Greedy Algorithms, Prim’s Minimum Spanning Tree Algorithm, Dijkstra’s Shortest Path Algorithm, hw02 DUE FRI.</td>
</tr>
<tr>
<td>5 F/03</td>
<td>B3rd 6.6, K&amp;T 4.6</td>
<td>Union-Find ADT, Kruskal’s Algorithm, review, MIDTERM 1 (FRI 02/07).</td>
</tr>
<tr>
<td>7 F/17(H)</td>
<td>CLRS 26.1—2, K&amp;T 7.1—3, K&amp;T 8.1—4, B3rd 13.1—2</td>
<td>Bipartite Matching, hw04 DUE WED. Computational Intractability, hw05 DUE FRI.</td>
</tr>
<tr>
<td>8 F/24</td>
<td>K&amp;T 8.1—4, B3rd 13.1—2</td>
<td>The Class NP of Decision Problems review, MIDTERM 2 (FRI 02/28).</td>
</tr>
<tr>
<td>9 M/02</td>
<td>K&amp;T 8.1—4, B3rd 13.1—2</td>
<td>NP-Completeness, Reductions, Standard NP-complete Problems</td>
</tr>
<tr>
<td>10 F/09</td>
<td>B3rd 13.3, K&amp;T 10.4—5</td>
<td>Intractability vs. Dynamic Programming review, QUIZ 3 (FRI 03/13).</td>
</tr>
<tr>
<td>11 M/16</td>
<td></td>
<td>Finals Week.</td>
</tr>
</tbody>
</table>