Time, place: Tuesday/Thursday 9:40am–11:05am
Dunn Hall 124

Instructor: Thomas Watson
Dunn Hall 315
Thomas.Watson@memphis.edu
http://www.cs.memphis.edu/~twwtson1/

Office hours: Tuesday/Thursday 11:20am–12:20pm

TAs: Jeff Atkinson (Jeff.Atkinson@memphis.edu)
Stephen Lee (smlee@memphis.edu)

Website: http://elearn.memphis.edu/

Description: In this course you will learn techniques for designing, analyzing, and implementing efficient algorithms for fundamental computational problems.

Topics include: stable matchings, Python programming, asymptotic runtime analysis, priority queues, graph traversal, greedy algorithms, shortest paths, minimum spanning trees, data compression, divide-and-conquer, dynamic programming, applications to computational biology.

Prerequisites: COMP 2700 (Discrete Structures). Since COMP 2150 (Object-Oriented Programming and Data Structures) is a corequisite of 2700, it is also effectively a prerequisite of 4030/6030.

Textbook: Strongly recommended:
Algorithm Design by Jon Kleinberg and Éva Tardos

Homeworks: There will be eleven homework assignments, each covering the material of about two lectures. See the calendar near the end of this document for the schedule. You may discuss homework problems with other students, but you must write up solutions entirely on your own and in your own words. You may not copy code from any source (person or website); changing variable names does not count as making the code “yours.” However, you may use the demo code the instructor writes during lecture as a starting point for solving homework problems.

You must submit each homework in the corresponding dropbox folder in the elearn website for the course. Collect all your solutions, including code as .py files, into a single .zip file for each homework. If you choose to handwrite solutions (rather than using software such as \LaTeX{}), you may turn in a scan or photo as long as the image quality is good enough that the TA will have no
problem reading it. If the TA finds it difficult to read one of your solutions, you will get 0 points for that problem.

Homework is due right before the beginning of lecture, and late homeworks cannot be accepted since model solutions will be distributed in class. Each 4030 student’s lowest two homework scores from the whole semester, and each 6030 student’s lowest one homework score, will be automatically dropped from the final grade calculation, so if extenuating circumstances prevent you from submitting a homework on time, one of these “freebies” will cover it by allowing you to take a 0 without harming your final grade.

Exams: The midterm exam is on March 14th in class (9:40am–11:05am, Dunn Hall 124) and will cover homeworks 1–6.

The final exam is on April 30th (10:30am–12:30pm, Dunn Hall 124) and will be cumulative but with an emphasis on homeworks 7–11.

For the midterm exam you may bring one double-sided sheet of notes, and for the final exam you may bring two double-sided sheets of notes (feel free to use your midterm sheet for one of them). Your sheets of notes may be typed. You may not use anything else during an exam; this means no calculators, textbooks, phones, earbuds, or anything else.

Grading:

<table>
<thead>
<tr>
<th></th>
<th>4030 students</th>
<th>6030 students</th>
</tr>
</thead>
<tbody>
<tr>
<td>5% each for 9 highest hw scores</td>
<td>4.5% each for 10 highest hw scores</td>
<td></td>
</tr>
<tr>
<td>25% for the midterm exam</td>
<td>25% for the midterm exam</td>
<td></td>
</tr>
<tr>
<td>30% for the final exam</td>
<td>30% for the final exam</td>
<td></td>
</tr>
</tbody>
</table>

We will calculate final letter grades in two different ways; then each student will receive the higher of the two letter grades. One way is a fixed grading scale, with the following cutoffs:

- A ≥ 90%   A– ≥ 85%   B+ ≥ 80%   B ≥ 75%
- B– ≥ 70%   C+ ≥ 65%   C ≥ 60%   C– ≥ 55%

The other way is a curve, with the following percentages of students receiving each grade:

- A: 15%   A–: 15%   B+: 15%   B: 15%
- B–: 10%   C+: 10%   C: 10%   C–: 10%

However, we will give an F to any student who clearly did not put effort into the course (or an A+ to any student with truly exceptional performance).

Cheating: Plagiarism or cheating behavior in any form is unethical and detrimental to proper education and will not be tolerated. All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student’s own work. The plagiarism is incurred when any part of anybody else’s work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other,
but appropriate references must be included for any materials consulted.

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor’s discretion) a failing grade in the course. The course instructor may also decide to forward the incident to the Office of Student Conduct for further disciplinary action. The following are not allowed in this course: posting homework problems to any website (including Chegg); copying code or other solutions from any person or website (including Chegg) even if you change the variable names; referring to solutions from previous semesters of this course.

**Calendar:**

- Jan 15: lecture 1
- Jan 17: lecture 2, hw 1 assigned
- Jan 22: lecture 3
- Jan 24: lecture 4, hw 1 due, hw 2 assigned
- Jan 29: lecture 5
- Jan 31: lecture 6, hw 2 due, hw 3 assigned
- Feb 05: lecture 7
- Feb 07: lecture 8, hw 3 due, hw 4 assigned
- Feb 12: lecture 9
- Feb 14: lecture 10, hw 4 due, hw 5 assigned
- Feb 19: lecture 11
- Feb 21: lecture 12, hw 5 due, hw 6 assigned
- Feb 26: lecture 13
- Feb 28: lecture 14, hw 6 due, hw 7 assigned
- Mar 05: Spring break—no class
- Mar 07: Spring break—no class
- Mar 12: review session
- Mar 14: midterm exam (in class)
- Mar 19: lecture 15
- Mar 21: lecture 16, hw 7 due, hw 8 assigned
- Mar 26: lecture 17
- Mar 28: lecture 18, hw 8 due, hw 9 assigned
- Apr 02: lecture 19
- Apr 04: lecture 20, hw 9 due, hw 10 assigned
- Apr 09: lecture 21
- Apr 11: lecture 22, hw 10 due, hw 11 assigned
- Apr 16: lecture 23
- Apr 18: lecture 24
- Apr 23: review session, hw 11 due
- Apr 25: study day
- Apr 30: final exam (10:30am–12:30pm, Dunn Hall 124)

**ABET outcomes:**

1. Analyze the complexity of functions using Big-$O$, $\Omega$, and $\Theta$.
2. Analyze and identify the running time of iterative functions.
3. Analyze and identify the running time of recursive functions.
4. Design and evaluate algorithms using the divide and conquer strategy on linear data structures.
5. Design and evaluate algorithms using the divide and conquer strategy on tree-like structures.
6. Design solutions to store repeated computation to improve the running time efficiency of algorithms.
7. Design solutions to enumerate all possible candidates to select the correct solution.
8. Create solutions by applying the decrease-and-conquer strategy.