CSCI 432/532: Advanced Algorithms – Fall 2020
(note: subject to in-semester revision)

Course information
Meeting time/location: MWF 12:00-12:50PM   http://wheelerlab.org/csci532

Instructor information
Instructor: Travis Wheeler
Office: Social Science 420
E-mail: travis.wheeler@umontana.edu
Phone: 406-243-6219
Office Hours: (http://wheelerlab.org/officehours)
Mon 2:00 – 3:30
Thu 2:00 – 3:30
Or by appointment (http://wheelerlab.org/schedule)

Course Objectives
The purpose of this course is to introduce you to advanced data structures and algorithms. With exposure to these concepts, you will gain comfort with interpretation and design of complex algorithms and their underlying concepts. You will be able to:
- Explain and apply concepts related to automata and computational theory
- Explain and apply advanced algorithm and data structure design strategies;
- Explain and apply dynamic programming algorithms, including a variety of acceleration strategies
- Explain and apply approximation algorithms;
- Explain and apply randomized algorithms;
- Explain and apply cache-efficient algorithms;
- and more…

Prerequisite:
CSCI 332 (Analysis of Algorithms)

Required textbook
None

General Structure of the course
(NOTE: all of this is subject to change; this will be a strange semester, and I may decide to change things up if they aren't working well)
The class is designed as a survey of advanced algorithmic concepts. After briefly covering automata and theory of computation from a classic text (which I'll provide), we'll dive into the literature to explore complexity and several (I think) important algorithmic strategies that may influence how you approach problems in your future careers. We'll cover one topic each week, and the coverage will be student-led. Here's what a typical week will look like:
- There's some assigned reading each week. This will usually be a paper, but the first couple weeks, it'll come from a text I'll share.
- The week's reading will be managed by a group of 2 or 3 students
  - On Monday, they will present the core content of the reading.

    - On Wednesday, they will do a deep dive on a small number of specific topics, with help from the rest of the class (see below).
    - On Friday, they will lead some in-class exercise/discussion/project, of their design.
    - These students will meet with me two times related to their reading:
      (1) on the Thursday or Friday preceding their paper's discussion, to ensure that their Monday presentation is ready to go. Note: this means they need to have a draft presentation ready by this time.
      (2) on Monday afternoon or some time Tuesday of their presentation week, to finalize plans for their Wednesday and Friday sessions
    - Graduate students will each be part of two groups during the semester
    - Undergraduates will be part of one group (i.e. will present only one reading); at most one undergrad per group.
- All students will have read the assigned reading before "showing up" to class on Monday.
- Before Monday's class, all students will submit a brief synopsis of the paper. This should include:
    - ~2-4 sentences describing the key ideas of the paper, and
    - at least one specific suggestion for a topic to discuss in the Wednesday class; these might come in the form of a question ("e.g. what's going on with the complicated thing in section X?") or a request ("e.g. let's talk in detail about difficult topic Y … or about how this might apply to another problem Z")
    - The presenting group will review these submissions with me, and during their Tuesday meeting, we will discuss:
        - grade assignments
        - plans for W/F sessions
- Reading material will be available on moodle

Grading

Graduate students		
Presentations:	40%	(note: score includes meetings with me)
Weekly writeups/exercises:	30%	
In-class participation:	30%	

Undergraduate students		
Presentations:	25%	(note: score includes meetings with me)
Weekly writeups/exercises:	40%	
In-class participation:	35%	

(note: that's right, no exam)

Impact of COVID-19

This is a new course. It would be a little bit loose in the best of circumstances. These are not the best of circumstances. I want you to be OK, to enjoy the course, and learn something useful from it. If we need to change the course, or if you need some individual flexibility to allow for some covid-induced trouble, reach out (early), and we can work it out. That said:

when you're responsible for the week's content, your classmates are depending on you. Please be particularly proactive about your needs and contributions during that week. Note: every student will be allowed to miss one weekly writeup with no required explanation.

Disabilities
Students with disabilities are encouraged to meet with me to discuss *any* accommodations they require.

Topics (expected sequence, approximately):

Part 1: Theory

Week 2: Aug 24-28: Sipser – Intro to Theory of Computation
Finite Automata: 1.1 (31-47)
Nondeterminism: 1.2 (47-58)

Week 3: Aug 31- Sep 4: Sipser – Intro to Theory of Computation
Regular Expressions: 1.3 (63-77)
Nonregular Languages: 1.4 (77-82)

Week 4: Sep 9-11: Sipser – Intro to Theory of Computation
* Note: no meeting Sept 7 *
Context-Free Grammars: 2.1 (101-111)
Pushdown Automata: 2.2 (111-125)

Week 5: Sep 14-18: Sipser – Intro to Theory of Computation
Turing Machines: Chapter 3 (165-187)

Week 6: Sep 21-25: Sipser – Intro to Theory of Computation
Time Complexity: Chapter 7 (275-311)

Week 7: Sep 28 - Oct 2: Dasgupta – Algorithms
NP-complete problems: Chapter 8

Part 2: Practice

Week 8: Oct 5-9: Approximation algorithms - Online Bin Packing
Lee, C. C., & Lee, D. T. A simple on-line bin-packing algorithm. Journal of the ACM, 32(3), 562-572. 1985

Week 9: Oct 12-16: Genetic algorithms - knapsack
Chu, P. C., & Beasley, J. E. A genetic algorithm for the multidimensional knapsack problem. Journal of heuristics, 4(1), 63-86. 1998.

Week 10: Oct 19-23: Dasgupta – Algorithms
Linear Programming: Chapter 7

Week 11: Oct 26-30: Dynamic programming, hidden Markov models – profile HMMs
Durbin et al. Biological Sequence Analysis, chapter 5, through 5.5

Week 12: Nov 2-6: String matching, indexed search – FM index
Ferragina, Paolo, and Giovanni Manzini. "Opportunistic data structures with applications." Proceedings 41st Annual Symposium on Foundations of Computer Science. IEEE, 2000.

Week 13: Nov 9-13: Branch and Bound
Wheeler, T. J.. Large-scale neighbor-joining with NINJA. In International Workshop on Algorithms in Bioinformatics. 2009.

Week 14*: Nov 16-25: Streaming Algorithms - Count-Min Sketch
(bleeds into finals week)
Cormode, G., & Muthukrishnan, S. An improved data stream summary: the count-min sketch and its applications. Journal of Algorithms, 55(1), 58-75. 2005.