Course Information (DRAFT)

Welcome to CS526, introductory graduate Algorithms!
As a prerequisite, you should have already taken an undergraduate course in algorithms, a standard part of a Computer Science major. Such a course typically uses a standard textbook (such as those by Cormen or Sedgewick), it it covers basic data structures, graph algorithms, and a few further topics such as network-flow and NP-hard problems. If you do not have such a background, then you should consider taking CS523 (graduate Data Structures and Algorithms I) instead. Or if you have already had a similar graduate course, you may possibly place out of CS526 (talk to me about this).

This course is mostly presented at the whiteboard (or projector), I’ll release scans from each lecture after the meeting. We review several familiar algorithmic paradigms, but with (we hope) new examples. We take a theoretical approach: we will prefer prefer general worst-case arguments, rather than experimental results. We are interested in both correctness and efficiency. As part of the course, I’ll ask you to make and present such arguments.

This course is organized into two distinct parts, as follows:

First Part (about 2/3): The first part is lectures, about eight weeks, ending with a “midterm” exam. We will review the core algorithmic curriculum, including some examples that are typically not in the undergraduate course. I want to cover (roughly) the following topics:

Topic 1: Divide and Conquer: FFT, the Master Theorem, separators.
Topic 2: Dynamic programming: MIS, treewidth, Baker’s PTAS.
Topic 3: Greed: from MST to matroids, set cover, Karger min-cut.
Topic 4: Optimization Reductions: max-flow, LP, min-cut, SDP.

If time permits, I may try to add some new material. During this part of the course we’ll have (probably) four written homework assignments, submitted using Canvas. The exam is tentatively on Tuesday, November 5.

Second Part (last 1/3): The second part is more of a seminar, lasting about 3.5 weeks. In small groups, students will take turns presenting advanced topics, with two presentations per meeting (so about 12-14 presentations total). We’ll negotiate these topics during the first part of the course, you might want to propose something related to your interests.

Our goal is to see more recent research, and also to get some practice with algorithmic reading and presentations. Besides the live presentation, each group will also prepare two documents: their slides (or video), and a topic writeup. There will be no homework or exam on this part of the course.

If I need to make any changes to the above schedule, I will announce it in class and on Canvas.

Book and Rough Syllabus: For the first part of the course, our main text is Algorithms by Dasgupta, Papadimitriou, and Vazirani (2006). This is a slim algorithms text, not so good for undergraduates, but better suited for our review purposes. We may also use the CLRS textbook, and some external sources. For the second part of the course you will be using external materials: conference papers, lecture notes, monographs, etc.
Meetings: We meet 1:00pm-2:15pm Tuesdays and Thursdays, in room N304. We will have one exam, as described above. During the second part of the course, you will be part of a group that will make a topic presentation, and you are expected to attend the other presentations. (I may take attendance during the seminar part.)

Staff: Your instructor (writing this) is Michelangelo Grigni. Contact me by e-mail at mgrigni@emory.edu or by phone at 7-7922. My office is room W426. My office hours will be posted on the web, and I am also available by appointment. I like talking to students, please come. We may have a TA, to be announced.

Grading: During the first part of the course there will be four written homeworks and a midterm exam. There may also be some programming challenges (probably on “hackerrank”). In the second part of the course, I’ll grade your presentation (and support documents), and I’ll also keep track of participation (attendance and interaction). Each graded item gets a mark in the range 0 to 100, curved so that the median is at least 85 (B). To get your final course average, I plan to take a weighted average of the marks, as follows:

- each homework gets a weight of one,
- programming challenges (all together) get a weight of one,
- the midterm exam gets a weight of two,
- your group presentation (including preparation and documents) gets a weight of two,
- class participation (including any quizzes, attendance, etc.) gets a weight of one.

If I need to make any change to the above weights, I will announce it in class and on Canvas.

Online Support: Our course page is https://cs.emory.edu/~cs526001/ It should have handouts, and materials for each meeting (at least brief notes and blackboard images). It will also link to the Emory Canvas service, which we will use for announcements, homework, and discussions.

Policies: If you miss an exam without prior arrangement, you need to arrange a medical excuse. Your work for this class is governed by the Laney graduate school Honor Code. See our course page (above) for links to the honor code, and my grading policy.