Course Information (DRAFT)

Welcome to CS523, graduate Data Structures and Algorithms I. This course introduces practical algorithms and data structures, for students entering graduate computer science from other fields of study. If you have already taken an undergraduate Computer Science algorithms course (such as our CS323), then you probably do not need to take this course, and you could consider taking CS526 instead. Please talk to me about this if you are unsure.

As a prerequisite, you should already know introductory computer programming (in a procedural language such as Java or python), and you should have some exposure to basic data structures such as arrays and lists. In this course you will do a mix of written work and programming work. In the written work, you will be arguing about the correctness and efficiency of various data structures and algorithms. In the programming work, you’ll solve some computational challenges, using ideas from the class.

Textbook: Our main text is Introduction to Algorithms by Cormen, Leiserson, Rivest, and Stein (2009, 3rd edition). It is widely used as an undergraduate textbook, and it contains a great deal of material, much more than we have time to cover. We’ll be considering its core topics (parts I, II, III), graph algorithms (much of part VI), and other topics as time permits.

Topics: This is my first time using this textbook at this level, so I don’t have a detailed syllabus, but I expect to cover at least this core sequence of topics:

- Basics, merge sort, big-Oh notation (Chapters 2 and 3).
- More divide and conquer (Chapter 4).
- The priority queue, heapsort (Chapter 6).
- Quicksort and radixsort (in Chapters 7, 8).
- Linked structures (Chapter 10).
- Hashing, variants (Chapter 11).
- Binary search trees, balanced trees (Chapters 12 and 13).
- Dynamic programming examples, like LCS (in Chapter 14).
- Greedy examples, like Huffman (in Chapter 15).
- Basic graph algorithms, DFS and BFS (Chapter 22).
- MST algorithms (Chapter 23).
- Shortest path algorithms (Chapter 24).
- Maximum flow (Chapter 26).

I hope to add other book topics (like amortized analysis and approximation) and outside topics (like persistence and perfect hashing) as time permits.
Languages: I teach the undergraduate course (CS323) using Java, since our undergraduates learn that as a prerequisite. However in this course I cannot assume that you all have the same programming background, so I will try to make this course more language agnostic. In particular, our textbook uses “pseudocode”. I’ll reproduce some textbook examples as working code, typically Java or python (my favorites). For homework, I’ll attempt to make the programming challenges language agnostic, so you can use the programming language of your choice.

Meetings: We meet 4:00pm-5:15pm Mondays and Wednesdays in W301 (28 meetings). We will have two in-class midterm exams, about 50 minutes each, tentatively on Monday October 1 and on Wednesday November 7. We will also have a final exam at the time scheduled by the registrar, 11:30am Thursday December 13.

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. I am also available by appointment. I encourage office visits, please come by and introduce yourself.

Grading: There will be a series of programming challenges (about five), altogether counting for about 15% of your grade; I’ll help you to complete these if necessary. There will also be a series of written homework sets (again about five), altogether counting for 30% of your grade. The written problems may be difficult, but they will be graded lightly, so that the median mark is around a 90 (A-). The written work should be regarded as review for the exams. The two in-class exams will each count for 15% of your grade, and the final exam will count for 20%. The exams will be curved so the median mark is a B+ (roughly 87 of 100). The last 5% of your grade is for “participation”, which may include writing up notes for one lecture.

Online Support: Our course page is [http://mathcs.emory.edu/~cs323001/](http://mathcs.emory.edu/~cs323001/) It should have handouts, and materials for each meeting (at least brief notes and blackboard images). We will use the Emory Canvas service for announcements, discussion, written homework submission, and grading. We may also a 3rd-party online judge (such as hackerrank.com) to submit programming challenges.

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 graduate school Honor Code. Programming work is also covered by the Math/CS SPCA (Statement of Policy on Computer Assignments). In particular, you should take care to protect the confidentiality of your course-related work.