CS 330 - Fall '19 - Introduction to Algorithms – Syllabus

Official Course Description

Examines the basic principles of algorithm design and analysis; graph algorithms; greedy algorithms; dynamic programming; network flows; polynomial- time reductions; NP-hard and NP-complete problems; approximation algorithms; randomized algorithms This course fulfills a single unit in each of the following BU Hub areas: Quantitative Reasoning II, Critical Thinking.

Piazza (Q&A, discussion, as well as distribution of lecture notes and homework):
https://piazza.com/bu/fall2019/cascs330/home

Gradescope (homework and programming exercise submission):
https://www.gradescope.com/courses/56239 (Entry code: M3DY2Y)

Google calendar with labs and lectures:
https://calendar.google.com/calendar/b/1?cid=YnUuZWR1XzA5NnEwM3M0ZnY4c2ZpM3MzNmExZmRtMzI4QGdyb3VwLmNhbGVuZGFyLmdvb2dsZS5jb20

Prerequisites

The class assumes **working knowledge of CS 112 and CS 131** (or MA 293). CS majors typically complete their Group B coursework (any two of CS 132, CS 235 and CS 237) before taking CS 330, and that is recommended. If you don't have the prerequisites, please talk to an instructor before deciding to continue with this class.

Instructors and Teaching Fellows

Name	**Office Hours**	**Email@bu.edu** *
Prof. Adam Smith	Tue 12:45-2:30pm and Thu 1:45-3:15pm	ads22
Prof. Dora Erdos	Tues 3-4PM and Wed 9:30-11 am in MCS 288	edori
Gavin Brown (TF)	Mon 2:30-4:30pm and Wednesday 1:30-3:30pm in EMA 302 (or 303 if 302 is very busy)	grbrown
Xin Lu (TF)	Tuesday 5-7pm and Friday 3:30-5:30pm in EMA 302 (or 303 if 302 is very busy).	xl

***** Messaging via Piazza is preferable to email (and will get a faster response).

Textbook

Algorithm Design, by Kleinberg and Tardos. ISBN 0-321-29535-8.

Useful additional resources:

- Cormen, Leiserson, Rivest, and Stein. *Introduction to Algorithms*, 3rd ed. MIT Press.
- J. Erickson. *Algorithms,* 2019. Available from http://algorithms.wtf/
  See also the extensive exercises on the website.
- Mathematics for Computer Science by Eric Lehman, Tom Leighton, and Albert Meyer. (Useful background on discrete mathematics.)

Course Structure and Communication

The class will be co-taught by Professors Smith and Erdos. On any given lecture date, one of the two instructors will deliver the lecture for both the A1 and B1 sections. The TFs will lead the discussion sessions. The objective is to reinforce the concepts covered in the lectures through problem-solving, and to provide clarification and guidance on the homework assignments. The purpose of the office hours of the Instructors and Teaching Fellows is to answer specific questions or clarify specific issues.

We will be using Piazza for all discussions outside of class. The system is highly catered to getting you answers to your questions fast and efficiently from classmates, the TFs, and instructors. Please do not email questions to the teaching staff -- post your questions on Piazza instead. We also encourage you to post answers to other students' questions there (but obviously, not answers to problems on the problem sets!). *Your fastest route to get an answer to most questions is via Piazza.* Office hours are not to be used to fill you in on a class you skipped or to re-explain entire topics. Office hours are scheduled at times to provide the most help to students who start the homework before the last minute.

Lectures

Lecture A1: Tues/Thurs 9:30 - 10:45 AM, CAS B18
Lecture B1: Tues/Thurs 11:00AM - 12:15, CAS B18

We expect students to come to class, and to come on time. While the class is large, class participation and questions will be encouraged. Also, while our textbook will be very helpful, it is an imperfect substitute for in-class learning, which is the fastest (and easiest) way to learn the material. If you miss a class, please get the notes and work through the material with a fellow student.

Discussion Labs

Lab A2 (Mon 9:05 - 9:55)	MCS B29
Lab A3 (Mon 10:10 - 11:00)	MCS B31
Lab B2 (Mon 11:15 - 12:05)	MCS B33
Lab A4 (Mon 12:20 - 1:10)	MCS B33
Lab B3 (Mon 1:25 - 2:15)	MCS B29

Lab B4 (Mon 4:40 - 5:30) MCS B29

Labs will be an invaluable part of the course involving interactive problem-solving sessions, tips on homework questions, and supplemental material not covered in lecture. We will post labs on Piazza in advance -- please read before coming to lab. Lab solutions will be posted Monday evening, after all labs conclude.

Coursework and Grading

The course grade will break down as follows:

35% weekly homework assignments, due Wednesdays starting September 11.
5% programming exercises
25% in-class midterm exam (in-class, planned for **Thursday, October 24**).
35% comprehensive final, in the normal exam slot for classes in our respective time blocks.
Up to 5% bonus for participation in lecture, lab, and on Piazza.

Last day to drop without a "W": October 7, 2019. With a "W": November 8, 2019. Incompletes for this class will not be granted.

Exams: There will be one eighty minute in-class midterm held during the middle of the semester before spring break, on **Thursday, October 24, 2019**. The cumulative final will be held during the normal two-hour final exam slot. Please make your travel plans accordingly.

Attendance: We will not take formal attendance in this course. However, while our textbook will be very helpful, it is an imperfect substitute for in-class learning, which is the fastest (and easiest) way to learn the material. Some material covered in lecture and lab may not be in our textbooks. You are in all cases responsible to be up to date on the material. We ask that you please arrive in class on time, since it is disruptive to have students flowing in throughout the class period. While the class is large, class participation and questions are very much encouraged.

Homework problems: Homework problem sets, assigned weekly, allow you to practice (a) solving problems using the ideas from class, often in a new way, (b) communicating your ideas using technical language (precise descriptions, pseudocode, formal claims, proofs). Most problems require written solutions, but some of the problems will contain small-scale implementations and simulations.

The homework is probably the most useful learning tool in the course—take it seriously, allow yourself time to do it, and have fun! Alumni often describe this course's homework as critical to their success in job interviews. Limited collaboration is permitted; see below.

Homework Submission: Assignments will typically be due Wednesdays by 11:59PM, electronically via Gradescope. Solutions should be typeset (preferred) or neatly hand-written and scanned.

Programming homework and “self-study” exercises: In addition to occasional programming problems on the homework (which will count towards your homework grade), there will be regular programming exercises that are mainly intended as a self-study tool. You will primarily be implementing the algorithms covered in class (or related to them). Finding an efficient implementation is usually the key to understanding the algorithms. You will submit your code via Gradescope. Altogether, these exercises are worth 5% of the course grade, and it is enough to complete 80% to receive full score.

Late Policy for homework assignments**:** During the course, you will have **two** chances to electronically submit assignments on Gradescope up to 48 hours late with no penalty, but Friday 11:59PM is a hard deadline. Any assignment arriving between Wednesday 11:59PM and Friday 11:59PM is considered late. Please do not send emails to the staff about late submissions (not necessary) or requesting additional time.

For **self-study programming exercises** (**not** homework problems), the late policy is more lenient: late submissions are worth up to 80% of the normal grade. These may be submitted as late as the day of the final.

Regrade Policy: If, after reviewing your solution, you still believe a portion of your homework was graded in error, you may request a regrade, **via Gradescope**, *NOT* through email. One of the staff will consider your request and adjust your grade if appropriate. Note that when we regrade a problem, your score may go up or down.

Workload: Be forewarned -- the workload in this course will be heavy. There is a problem set (almost) every week. As you likely already know, assignments requiring substantial creativity can take more time than you expect, so plan to finish a day early.

Collaboration Policy

Collaboration on homework problems, with the exception of the self-study programming assignments, is permitted and even encouraged! If you choose to collaborate on some problems, you are allowed to discuss each problem with at most 5 other students currently enrolled in the class. Before working with others on a problem, you should think about it yourself for at least 45 minutes.

You must write up each problem solution by yourself (using your own words) without assistance, even if you collaborate with others to solve the problem. You must also identify your collaborators. If you did not work with anyone, you should write ``Collaborators: none.'' It is a violation of this policy to submit a problem solution that you cannot orally explain to an instructor or TF. You may get help on Piazza, from the TFs and instructors for the class for specific problems. (Don't expect them to do it for you, however!)

Finding answers to problems on the Web or from other outside sources (these include anyone not enrolled in the class) is strictly forbidden.

No collaboration whatsoever is permitted on exams. The collaboration policy for programming problems will be specified in the assignments.

Collaboration strategies: If you do collaborate, use it as an opportunity to practice group work skills: give everyone a chance to speak, listen carefully, acknowledge good suggestions. If you have a tendency to be shy, speak up! If you have the tendency to dominate conversations, make sure to give others the floor.

Academic Conduct:

Academic standards and the code of academic conduct are taken very seriously by our University, by the College of Arts and Sciences, and by the Department of Computer Science. Course participants must adhere to the CAS Academic Conduct Code -- please take the time to review this document if you are unfamiliar with its contents.

If in doubt, our department has an extensive compilation of **examples** with regard to Academic Conduct and permissible collaboration.

Violations of this policy will be dealt with according to University regulations.

Tentative schedule:

This schedule is subject to change. We will keep an up-to-date schedule on Piazza.

Week	Topic	Readings from *Algorithm Design*
1	Course syllabus. Stable matching. Asymptotic order of growth.	ch.1.1. , ch. 2.1-2.4
2	Asymptotic growth continued, recap of data structures. Basic graph defs and algorithms (e.g. connectivity, traversal, DAGs)	ch. 2.5, ch. 3.1-3.3
3	Greedy algorithms (e.g. scheduling, shortest paths, MST, basic clustering, Huffman codes)	ch.4.
4	Greedy algorithms continued.	
5	Divide-and-Conquer (closest pair of points, fast integer and matrix multiplication)	ch. 5.4-5.5

6	Dynamic programming (Weighted Interval Scheduling, Subset sum, Knapsack, Sequence alignment, shortest paths)	ch. 6.
7	Dynamic programming continued.	
8	Recap. Midterm	
9	Network Flow (Ford-Fulkerson, MFMC, extensions), flow applications (max matching, disjoint paths, etc.)	ch. 7.
10	Network flow continued.	
11	Polynomial time reductions. NP and NP-Complete	ch. 8.1.
12	NP continued. Thanksgiving	
13	Local search, randomized algorithms	ch. 12-13.