Boston University, CAS CS 330 Spring 2024
Introduction to Analysis of Algorithms
Syllabus

Course Description

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

This course fulfills a single HUB unit in each of the areas **Quantitative reasoning II.** and **Critical Thinking**.

Prerequisites

Students must complete **CS 112 and CS 131** (or MA 293) prior to taking CS330. In addition CS majors need to complete **at least one** of their Group B coursework (any of CS 132/MA242, CS235/MA294 and CS237/MA581), other majors have to have completed at least one of these courses or their equivalents. If you don't have the prerequisites, you **must** talk to an instructor before continuing with this class.

Sign up ASAP:

Piazza (Q&A, discussion, as well as distribution of lecture notes, homework, additional material, course logistics info):
https://piazza.com/bu/spring2024/cascs330

Gradescope (Homework submission) :
https://www.gradescope.com/courses/698856 entry code: GPJ8BY

TopHat: A platform for in-class interaction and questions. You have to purchase a membership, there are semester and year-long options. If you use TopHat in multiple courses, you only have to pay the fee once. *If you do not have an account and the fee presents a financial hardship, please let the instructor know via a private Piazza post*.

https://app.tophat.com/e/689783 code: 689783

Instructors and Teaching Fellows (reachable on Piazza) Office hour info is posted on Piazza

Name	`@ bu.edu`
Prof. Dora Erdos	`edori`
TF Patrick Lutz	`plutz`
TF Themis Nikas	`tnikas`
TA Osama Dabbousi	`dabbousi`
TA Nada Abdelwahab	`nawahab`
TA Mohammed Aldahmani	`dahmani`
TA Ross Mikulskis	`rkulskis`

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.)

Instructor:
Dora Erdos: email edori@bu.edu , office CDS 910.

Communication related to class material should be primarily conducted through Piazza, for personal questions you can use the “private” post option.

Structure:
The class consists of two 75 minute lectures (Tues, Thur) and one 50 minute discussion (Mon) each week. Attendance in lectures is mandatory and will be tracked through TopHat participation (see later section for details). The two sections of the course, A1 and B1, will be treated as one class. The content of the two lectures is identical[1], assignments, course staff and all resources will be shared, students can mix-and-match A and B lecture and discussion sections.

The TFs and TAs 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.

We will hold multiple office hours throughout the week, the purpose of these is to answer specific questions about the material and help with assignments. Students are welcome to attend office hours with any of the instructors. The exact schedule will be communicated on Piazza.

Communication:

We will be using **Piazza for all discussion** pertaining to the class. You should post your questions about the material, lectures, homework or course logistics here. Piazza is highly preferred over sending the course staff emails. Most often your question and the answer will be just as useful to your fellow students as yourself. We encourage you to respond to questions. The course staff will be monitoring Piazza and chime in as needed. You are encouraged to ask questions about the homework assignments, e.g. clarifications, related material, but **do not post solutions** to the problems. If you think your question is too specific or would reveal too much, then please ask in a private post. See the pinned "Ethics and Etiquette" post on Piazza for further guidance.

Our **Piazza** page:
https://piazza.com/bu/spring2024/cascs330

Lectures
A1 section : Tue, Thur 11 :00 - 12 :15 pm, CGS 511
B1 section : Tue, Thur 12:30 - 1 :45 pm, CGS 511

Discussion Labs: Labs will be an invaluable part of the course involving interactive problem-solving sessions, tips on homework questions, and supplemental material not covered

[1] If on a given day you can't attend your own lecture section then - as long as there are seats available - you are welcome to attend the other lecture. No need to inform the instructor. For your **midterm exam** you have to attend the section **you are registered for**.

in lecture. We will post lab notes on Piazza in advance -- please read before coming to lab. Solutions will be posted after all labs conclude.

Topics We will mostly follow the order and content in the textbook. Topics are subject to change. A tentative weekly schedule can be found at the end of this document.

- Asymptotics, data structures, how to describe an algorithm (pseudocode)
- Graphs – data structures, graph traversals, connectivity, DAGs
- Greedy algorithms – scheduling, shortest paths, minimum spanning trees
- Divide-and-conquer – variations on sorting and searching, integer multiplication, closest pair of points
- Dynamic programming – interval scheduling, subsetSum, knapsack, sequence alignment, Bellman-Ford
- Network flow – Ford-Fulkerson, MFMC theorem, applications
- Polynomial time reductions, NP, NP-Completeness

Course atmosphere, diversity and inclusion: We intend to provide a positive and inclusive atmosphere in class and on the associated virtual platforms. Students from a wide range of backgrounds and with a diverse set of perspectives are welcome. We ask that students treat each other with thoughtfulness and respect, and do their part to make all their peers feel welcome. Your suggestions are encouraged and appreciated. Please let us know ways to improve the effectiveness of the course for you personally or for other students or student groups.

Grading

The course grade will break down as follows:

5% class participation (based on TopHat)
30% weekly homework assignments (due Wednesdays, first hw due **Wed, Jan 31**).
30% in-class midterm exam (in-class, planned for **Thu, March 7**).
35% comprehensive final (during finals week).

Students **must pass both exams** with at least **40% of the grade** to pass the course. This will be strictly enforced.

Last day to **drop without a "W": February 22, 2024** . **With a "W": March 29, 2024**. Incompletes for this class will be granted based on CAS Policy as well as CS departmental policy (mostly only for last minute emergencies).

Exams: Both exams will consist of problem solving and short questions about the material. The midterm will be during class time and takes 75 minutes. The final is during the University-assigned final exam slot. The content of the final is cumulative.

No collaboration whatsoever is permitted on exams, any violations will be reported to the College.

Accommodations: All are welcome in the course. If you require particular accommodations for exams or coursework, please contact the instructor (and forward any relevant documentation from Disability and Access Services) in a timely manner. If you are facing unusual circumstances during the semester, please reach out to us early on so that we can find a good arrangement.

Attendance and participation: Your participation grade depends on answering TopHat questions, which requires your presence in class. 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. Class participation and questions are very much encouraged. *Please ask as many questions in class, labs and on Piazza as you need.* Chances are that your question and answer will be as helpful to your classmates as to you.

TopHat is a web-based platform for interactive questions during class. Our goal in using it is to make lectures more interactive, get you thinking actively about the material, and get some feedback on what you are learning. TopHat questions are generally multiple choice. Most of the points (80%) are for participation. The remaining 20% is for correctness. You will get the full 5% of the course grade if you get at least 80% of the possible TopHat points for the semester.

Course TopHat page:
https://app.tophat.com/e/689783 code: 689783

Homework:

Homework problems: There will be weekly homework assignments, 10 total for the semester. Problems will be posted **Thursdays** and due the following **Wednesday** evening. Homework problem sets will consist of written problems. They will allow you to practice (a) solving problems using the ideas from class, often in a new way, (b) to assess and analyze the correctness of your solution and (c) communicate your ideas using technical language (precise descriptions, pseudocode, formal claims, proofs). Be aware that this latter is just as important as the two former and we will place high emphasis on it.

The two lowest grades on your homework assignments will be **dropped**.

Content: The homework is designed to aid you in understanding of the material. The problems will usually ask you to apply an algorithm we study in some clever way or design a slight

modification of it. Lab assignments will often be helpful to solving these problems. Solving problems is probably the most useful learning tool in this 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 on homework is permitted; see below.

Homework Submission: Assignments will be due **Wednesdays by 11:59PM**, electronically via Gradescope. Solutions to written problems should be *typeset*. Hand-written solutions will not be accepted. We'll provide you with resources on how to get started if you're not familiar yet with typesetting mathematical-algorithmic text.

link to Gradescope:
https://www.gradescope.com/courses/698856 entry code: GPJ8BY

Late Policy: Late assignments will **not** be accepted as we intend to post solutions the next morning. You can use your dropped grades to cover for up to two late assignments. Also, be mindful that sometimes it's ok to submit partial results if you weren't able to fully finish your assignment, don't miss the deadline because of last minute work.

Regrade Policy: If, after reviewing the solutions and your answer, 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. Regrade requests can be submitted up to one week (7 days) after grades for that assignment have been posted.

Workload: CS 330 is a substantial amount of work. There is a problem set every week as well as two exams to study for. As you likely already know, assignments requiring substantial creativity can take more time than you expect, so plan to finish a day early.

Personal difficulties: Unfortunately, it may happen that you find yourself under circumstances that are affecting your ability to perform well in this class. We are here to support you and find the best way to help you in this course. Please reach out, so that we can help.

Collaboration, Citation, and Academic Honesty

Citation policy: You can reference anything from the textbook, lecture and discussion notes, information given by the course staff without having to cite it. However, if you make use of any other information, you have to include **proper citation**. If you omit to do this you are committing plagiarism.
You are allowed and encouraged to further your knowledge by finding related material online or in books, but you have to cite it. Your citation may be a url, the title and chapter of a book, or a paper reference.

Searching explicitly for answers to problems on the Web or from persons not enrolled in the class this current semester is strictly forbidden. (This includes students who took the course in previous semesters, posted solutions, Chegg, GitHub, etc.) Using AI tools, e.g. ChatGPT, for solving problems is forbidden.

Collaboration Policy: Collaboration on homework is permitted and even encouraged. If you choose to collaborate on some problems, you are allowed to discuss each problem with **at most 3** other students currently enrolled in the class. Before working with others on a problem, you should think about it by 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. Identical worded answers, including identical pseudocode, will receive no grade. You must also identify your collaborators clearly on the first page of your assignment. 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 the instructors. You may get help on Piazza or in office hours from the instructors for the class for specific problems. You don't need to list them as collaborators .

No collaboration whatsoever is permitted on exams!

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. We strongly encourage you to find a small group of classmates that you regularly discuss and review material with. Feel free to post on Piazza to find a study mate.

Academic Conduct:
Academic standards and the code of academic conduct are taken very seriously by the University, the College of Arts and Sciences, and 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.

Any case of academic misconduct, including but not limited to plagiarism and submission of work that was not solved by you – be that a tutor, friend, web resource, AI, etc., will be reported to the College and will also carry a grading penalty.

HUB areas:

This course fulfills a single unit in each of Quantitative Reasoning II. and Critical Thinking.

Quantitative Reasoning II:

Learning Outcome 1:

Students will frame and solve complex problems using quantitative tools, **such as** analytical, statistical, or computational methods.

This is the *central* course in computer science in which problem solving methods and tools are considered in detail, after students have had 3-5 programming and theory classes.

Learning Outcome 2:
Students will apply quantitative tools in diverse settings to answer discipline-specific questions **or** to engage societal questions and debates.

Learning Outcome 3:
Students will formulate, and test an argument by marshaling and analyzing quantitative evidence.

Students will marshal arguments and evidence of an empirical nature to test the properties of algorithms.

Learning Outcome 4:
Students will communicate quantitative information symbolically, visually, numerically, **or** verbally.

The course gives ample practice in communicating symbolically.
Learning Outcome 5:
Students will recognize and articulate the capacity and limitations of quantitative methods and the risks of using them improperly.

This course considers the notion of limitations in great detail, including the fundamental limitation of NP-Hard problems!

Critical Thinking:

Learning Outcome 1:
Students will both gain critical thinking skills and be able to specify the components of critical thinking appropriate to a discipline or family of disciplines. These may include habits of distinguishing deductive from inductive modes of inference, methods of adjudicating disputes, recognizing common logical fallacies and cognitive biases, translating ordinary language into formal argument, distinguishing empirical claims about matters of fact from normative or evaluative judgments, and/or recognizing the ways in which emotional responses or cultural assumptions can affect reasoning processes.

This course is in many ways the intellectual "center" of the CS major, where a large variety of classic modes of thinking are explored, and their properties exploited to understand a variety of classic algorithms. Proof methods and empirical techniques are used to prevent cognitive bias and logical fallacies, and to rigorously test assumptions.

Learning Outcome 2:
Drawing on skills developed in class, students will be able to critically evaluate, analyze, and generate arguments, bodies of evidence, and/or claims, including their own.

Students apply the critical thinking techniques of this course to a large variety of algorithms, covering all the major categories. They write many different algorithms and evaluate them for correctness and time complexity.

Tentative schedule:

This schedule is for informational purposes only and is subject, and is likely, to change as we progress through the semester.

week	date	topic
1	1/18 Thu	course intro, pseudocode
2	1/23 Tue	asymptotics - # of comp steps to func
	1/25 Thu	adjacency list, nested hash table. graph traversal in general. - BFS
3	1/30 Tue	BFS proof, runtime
	2/1 Thu	Directed graphs. DFS - recursive/iterative implementation, runtime. (strong connectivity)
4	2/6 Tue	DAGs
	2/8 Thu	DAGs, (strong connectivity)
5	2/13 Tue	greedy int scheduling. IsCompatible implementation
	2/15 Thu	greedy int partitioning, PQ
6	2/20 Tue	Dijktsra
	2/22 Thu	Dijkstra+MST
7	2/27 Tue	MST
	2/29 Thu	DQ: mergesort, binary search, recurrences

8	3/5 Tue	review
	3/7 Thu	**midterm**
9	3/9-3/17	spring break
10	3/19 Tue	DP: WIS, Fibonacci
	3/21 Thu	SubsetSum, backtracking
11	3/26 Tue	knapsack, variants, backtracking
	3/28 Thu	Bellman-Ford
12	4/2 Tue	Sequence Align
	4/4 Thu	max flow, Ford-Fulkerson
13	4/9 Tue	flow apps
	4/11 Thu	MFMC
14	4/16 Tue	flow apps
	4/18 Thu	poly-time reductions
15	4/23 Tue	NP-decision problems
	4/25 Thu	NP-C
16	4/30 Tue	review