NYU | TANDON SCHOOL OF ENGINEERING

Prof. Boris Aronov	*CS-GY 2413: Design & Analysis of Algorithms II — Fall 2020*
Office (but mostly via Zoom)	370 Jay Street, 8th floor, room 857 +1-646-997-3092
Classroom and lecture	370 Jay Street, room 202 (and via Zoom) MonWed 12:30–1:50pm
Office hours	TBA (via Zoom)

Course Syllabus

Catalog description: This course covers fundamental principles of the design and analysis of algorithms. Topics include asymptotic notation, recurrences, randomized algorithms, sorting and selection, balanced binary search trees, augmented data structures, advanced data structures, algorithms on strings, graph algorithms, geometric algorithms, greedy algorithms, dynamic programming and NP completeness. —

Prerequisites (from the catalog): Prerequisites for Brooklyn Engineering Students: (CS-UY 2134 or CS-UY 1134) and (CS-UY 2124 or CS-UY 1124) (C- or better) and MA-UY 2314 — Prerequisites for Abu Dhabi Students: (ENGR-UH 3510 or CS-UH 1050) (C- or better) and CS-UH 1002 — Prerequisite for Shanghai Students: CSCI-SHU 210 (C- or better) and CSCI-SHU 2314.

Prerequisites in more detail: Though the prerequisite material will be briefly reviewed in class, familiarity with basic data structures, with the concept of a graph, with asymptotic notation, with basic probability theory will be assumed. Class summaries (or pointers to the material) will be provided, but what is important is these will be used as everyday tools in the course, so you need to get up to speed/review the material ASAP. The summaries are not expected to replace having taken a suitable course in the subject or having equivalent knowledge. There is an emphasis in the course on *analyzing* things: Proving that algorithms are correct. Analyzing running times and other resources used by an algorithm. Proving lower bounds on algorithms and problems. The tools are often mathematical and using them requires, in addition to some specific technical knowledge, some mathematical maturity.

Valuable resource: Greg Aloupis is teaching an online class on the same subject in the same time slot this semester. He has a vast collection of resources (http://research.engineering.nyu.edu/~greg/algorithms/2413/resources.html) devoted to the class: Videos for lectures, sometimes in several different versions, pointers to materials of various kinds, reviews of prerequisite material, etc etc.

Text (recommended): Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms*, 3rd Edition, MIT Press, 2009; ISBN 9780262033848; paperback international version ISBN 9780262533058; e-version ISBN 9780262259460. It is usually referred to as *CLRS* or "the fat algorithms book." There is a way to access the book online via NYU libraries (Proquest Ebook Central seems to have it, Skillsoft/Skillport seem to have it; Knovel will *remove* (??) it soon.)

Clarification: The book is huge. It covers several courses worth of material. We will not cover anywhere near all of it. Some people like textbooks. Some people hate them. Some parts of the book are very nice. Some parts I cannot make much sense of. It was written by a committee. The textbook is not strictly required. However, sometimes I might find it useful to refer to a section of the textbook for an explanation or illustration, or for a specific set of rules a data structure should follow, and I expect you to be able to look it up. As mentioned above, I believe there is a way for NYU students to access this book already without buying/renting them.
Other reasonable algorithms textbooks (they cover a lot less material, though): J. Kleinberg and É. Tardos, *Algorithm Design*, Addison Wesley, 2006, ISBN 0-321-29535-8.
S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, *Algorithms*, 2006, ISBN 978-0073523408.
There are many many many algorithms textbooks out there. Some are reasonable. Many are not reasonable at all. Ask me or see if you can look up some reviews before you rely on an arbitrary algorithms book.

Assignments and exams: There will be regular (approximately weekly to biweekly) homework assignments, **three** midterm exams, and one final exam. You are encouraged to experiment with writing code for the algorithms, but no required programming assignments will be given. Missing an exam requires a good reason, with documentation, or will result in a grade of ZERO for the exam. Make up arrangements for missed exams will be considered *only* with adequate documentation.

Participation: Participation is a *required* part of the course. You will not get a grade if you do not participate. (There's no specific number attached to participation in your grade.) There are several different ways to participate in the course, which also depend on how you are taking the class: in person or remotely, synchronously or asynchronously. You can ask and answer Piazza questions, ask questions in class, participate in a homework solution discussion and so forth.

Grading scale:
Homeworks ZERO (0%). Homeworks are there for you to ensure you are following the material in the course and to get some practice. See a separate document describing the homeworks. There will be *four* exams in total. Each exam will be worth 1/4 of the course grade. There are many different ways to learn more or learn better, but **there will not be any extra-credit activities**.

Class communication: Piazza (course site: https://piazza.com/nyu/fall2020/csuy2413/home) will be used almost exclusively for all announcements and discussion. I will probably use Gradescope for homeworks and exams. The instructions will be posted separately.
Everyone is encouraged to ask questions on Piazza. And everyone is encouraged to answer them. Learning many different ways to correctly answer a question is a great way to understand things. Learning why a particular approach is wrong is also very helpful.
I will post the plan for what will be covered in an upcoming lecture (hopefully before the actual lecture) and what was covered in previous lectures, regularly, on Piazza.

Changes: All the information in this syllabus is subject to change. Changes will be announced in class and on Piazza.

Important dates:

First lecture	Wednesday, September 2
No class	Monday, September 7 (Labor Day)
Last day to add/drop	Monday, September 15
Midterms	Sept 28, Oct 21, Nov 16 (*tentative*)
Last day to withdraw with a grade of W	Tuesday, November 30 (*changed!*)
Final exam	TBA during exam week December 15–21

Tentative course outline: This is a rough list of topics in the course, not necessarily in the order they will be covered, time permitting.

1. Course logistics.
2. Introduction. What's an algorithm? How do you evaluate algorithms? Speed/running time. Space. Other metrics. Correctness. Termination. Lower bounds. Best case/worst case/average case.
3. Review of asymptotic notation: big-Oh and friends. Only a quick review; it is your responsibility to recall all the important concepts and refresh your memory.
4. Sorting (will include: slow boring algorithms, faster algorithms that mostly illustrate various analysis techniques). Possibly will include linear-time sorting and sorting lower bounds.
5. A review of probability, indicator random variables; again, just a review.
6. Randomized algorithms and their analysis [not a separate topic, will pop up here and there].
7. Quicksort. Quickselect. And friends.
8. Searching: hashing and search trees, with relatives.
9. Augmented data structures [tentative].
10. Amortization [tentative: Greg loves this topic; I am not sure I can fit it in].
11. Greedy algorithms [might not treat this as a separate topic].
12. Dynamic programming.
13. Graphs: review of what they are, then BFS, DFS, strongly connected components (time permitting).
14. Fundamental graph algorithms: Single-source shortest paths, minimum spanning trees.
15. P vs. NP

A running log of what was covered in previous classes and the plan for the upcoming lecture will be maintained on Piazza.

Homework policy A policy on homeworks will be posted as a separate document.

Exam policy A policy on exams will be posted as a separate document.

Cheating The precise rules for what is and is not allowed on an exam will be posted separately. Generally, cheating on an exam will result in an F in the course and a report to the department and the school.

Moses Center statement of disability: If you are a student with a disability who is requesting accommodations, please contact New York University's Moses Center for Students with Disabilities (CSD) at 212-998-4980 or mosescsd@nyu.edu. You must be registered with CSD to receive accommodations. *Please do this at the start of the semester.*
Information about CSD can be found at http://www.nyu.edu/csd. It is located at 726 Broadway on the 2nd floor.

NYU School of Engineering policies and procedures on academic misconduct: *Introduction*: The School of Engineering encourages academic excellence in an environment that promotes honesty, integrity, and fairness, and students at the School of Engineering are expected to exhibit those qualities in their academic work. It is through the process of submitting their own work and receiving honest feedback on that work that students may progress academically. Any act of academic dishonesty is seen as an attack upon the School and will not be tolerated. Furthermore, those who breach the School's rules on academic integrity will be sanctioned under this Policy. Students are responsible for familiarizing themselves with the School's Policy on Academic Misconduct.
Definition: Academic dishonesty may include misrepresentation, deception, dishonesty, or any act of falsification committed by a student to influence a grade or other academic evaluation. Academic dishonesty also includes intentionally damaging the academic work of others or assisting other students in acts of dishonesty. Common examples of academically dishonest behavior include, but are not limited to, the following:
1. Cheating: intentionally using or attempting to use unauthorized notes, books, electronic media, or electronic communications in an exam; talking with fellow students or looking at another person's work during an exam; submitting work prepared in advance for an in-class examination; having someone take an exam for you or taking an exam for someone else; violating other rules governing the administration of examinations.
2. Fabrication: including but not limited to, falsifying experimental data and/or citations.
3. Plagiarism: intentionally or knowingly representing the words or ideas of another as one's own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information.
4. Unauthorized collaboration: working together on work that was meant to be done individually.
5. Duplicating work: presenting for grading the same work for more than one project or in more than one class, unless express and prior permission has been received from the course instructor(s) or research adviser involved.
6. Forgery: altering any academic document, including, but not limited to, academic records, admissions materials, or medical excuses.
See https://engineering.nyu.edu/campus-and-community/student-life/office-student-affairs/policies/student-code-conduct for the Code of Conduct.