The information contained on this page is designed to give students a representative example of material covered in the course. Any information related to course assignments, dates, or course materials is illustrative only.

NYU TANDON SCHOOL OF ENGINEERING

Course Syllabus

Computer Science and Engineering

Design & Analysis of Algorithms I

Course Information

Course Pre-requisites

Discrete math is important. Basic understanding of some data structures will matter too. See the Resources page and search for the word "prerequisite" to see when you will need to know background material.

You will probably find this course difficult if you're not comfortable with induction, recursion and proofs, or if big-O seems like a challenging concept. It is assumed that you know the basics about arrays, linked lists and trees. Knowledge of basic probability will be helpful for two or three lectures but I will recap what you need to know.
If you have not done well in discrete math or have not passed data structures, you should contact me.

Course Description

This is an introduction to the design and analysis of algorithms, which involves discussing a few basic data structures as well. Many topics could fit in such a course, and not all intro courses go over exactly the same material.

We will place all emphasis on theory instead of programming. This course is about figuring out how to solve a problem before you start coding.

To see what is taught in this course, please visit the Resources page.

Course Objectives

This course will provide students with the opportunity to:

This is an introduction to the design and analysis of algorithms, which involves discussing a few basic data structures as well. Many topics could fit in such a course, and not all intro courses go over exactly the same material.

Course Structure

This course is conducted entirely online, which means you do not have to be on campus to complete any portion of it. You will participate in the course using NYU Classes located at https://newclasses.nyu.ed
Your final grade will be computed as a combination of the components shown below.

- Exam 1: 30%
- Exam 2: 30%
- Exam 3: 40%
- Homework: 0%, but mandatory

  There will be roughly 20 homework "units".

  If you fail to submit 5 or more units without justification, your final grade will drop by a letter (e.g., B to C).

  If you fail to submit 8 or more units, you will not pass the course. Furthermore, submitting an answer that shows almost no effort or understanding will also count towards this rule.

- If you have a high homework average, your final grade may be increased a little (e.g., A- to A).
  If you have a low homework average, your final grade may decrease (e.g., B to B-).
  I will not be defining "high" and "low" quantitatively until the end of the semester.

Weekly Structure

Week 1: Discrete Math and Data Structures Review

- Review of basic relevant programming forms.

Week 2: Big-O Notation (aka Theta notation)

- Common functions
- Book: chapter 2, p.43-52.

Week 3: Recurrences

- The recursion tree and substitution method
- The Master method
- Examples of using recurrences; recursive algorithms. (parts b-e are not critical for exams)

Week 4: Sorting

- Insertion sort (also serves as an informal prelude to big-O)
- Mergesort (also serves as a quick intro to recurrences)
- Heapsort
- Quicksort
- Lower bound for comparison model
- Counting sort and radix sort
- Other

Weeks 5-6: Binary Search Trees

- Building a BST randomly, average depth, and relation to Quicksort.
- Red-black trees (an example of dynamic balanced BST)

Week 7: Augmenting Data Structures

- Intro, dynamic rank queries and selection
- Range counting (almost the same as dynamic rank queries)
- Interval trees

Week 8: Dynamic Programming

- Counting paths
- Longest common subsequence
- Longest increasing subsequence
- Rod cutting

Week 9: Amortization

- Multi-pop stack, and binary counter
- Book: Chapter 17
- New video for array doubling.

Week 10: Graph Basics

- Representation and types of graphs (and a little bit about planarity)
- Book: Chapter 22, p.589-592
- Breadth-first search (BFS) and depth-first search (DFS)

- Topological sort
- Strongly connected components

Week 11: Minimum Spanning Trees

- Intro and properties
- Kruskal's algorithm
- Prim's algorithm
- Book: Chapter 23, p.634-636.

Week 12: Single source Shortest Paths

- General properties, relaxing edges, and an important lemma.
- Bellman-Ford algorithm
- Algorithm for DAGs
- Dijkstra's algorithm

Week 13: All-pairs Shortest Paths

- Prerequisite knowledge: single-source shortest paths.
- Book: Chapter 25 (in particular Johnson's algorithm is in 25.3)

Week 14: Hashing

- Intro, chaining, simple uniform hashing
- Open addressing, linear and quadrating probing, double hashing, analysis with uniform hashing assumption.
- Recommended further reading: Universal hashing, Perfect hashing, Cuckoo hashing (and more about the cuckoo mafia)

Week 15: NP-Harding, Indicator Random Variables, Lossless compression

- Intro
- Examples of reductions
- Book: Chapter 5
- Finding local maxima in permutations.

Learning Time Rubric

Please modify the below table to represent the breakdown of learning time in each week of your course.

Learning Time Element	**Asynchronous* / Synchronous****	**Time on Task for Students (weekly)**	**Notes**
Reading Assignments / Recorded Lecture	Asynchronous	2.5 hours	Video format. Expect quizzes throughout the module or weekly chapter readings
Weekly Discussion Board	Asynchronous	1.5 hours	Students are expected to post initial response to weekly topic questions. See Interaction Policy.
Assessment (Labs and Programming assignments)	Asynchronous	2 hours	Students submit their assignment by [the end of the week]
Reading Assignment	Asynchronous	2 hours	Reading assigned textbook chapters and journal articles.

Live webinars	Synchronous	2 hours	Group discussion in class, live, overly weekly chapter

Course Communication

Interaction Policy

Please follow the interaction guidelines stated below for this course.

- First make sure the answer to your question isn't already on the course website (e.g., when is the exam, when are office hours, etc)
- We have a Piazza page.
  Piazza is mainly meant for asking questions that others would benefit from seeing. If you have a question that does not affect other students, or that may reveal answers to homework questions, please use email instead. Do not post questions like "I tried to solve homework X like this but I'm not sure it's correct". In general, do not use Piazza as a substitute for sending me private messages. Email me instead.
- Grading concerns: submit a regrade request in Gradescope. If it remains unresolved for a few days, contact me.
- Major illness or other issue? Feel free to contact me, but you are expected to contact Health Services and/or your dean.
- Office hours may vary and will be updated weekly on the main page.
- Click on a TA name below to see their photo.
- email addresses are @nyu.edu

Announcements

Announcements will be posted on NYU Classes on a regular basis. You can locate all class announcements under the *Announcements* tab of our class. Be sure to check the class announcements regularly as they will contain important information about class assignments and other class matters.

Email

You are encouraged to post your questions about the course in the Forums discussions on NYU Classes. This is an open forum in which you and your classmates are encouraged to answer each other's questions. But, if you need to contact me directly, please email me. All homework, labs or programming assigments related questions must be researched first on own time, then posted on forums, then discussed with TAs during weekly reviews, and then can be forwarded to me. Typically, you can expect a response within 48 hours.

Readings

The non-required text for the course is: Introduction to Algorithms, 3rd edition, by Cormen, Leiserson, Rivest and Stein.

This is commonly just referred to as ``CLRS". More info at [MIT press].

Note: my Resources page refers to this book so you can find relevant chapters if you wish. The course doesn't explicitly rely on the book though. Also, this book is massive. We will not cover everything in it. See "Topics" below.

You can access NYU's central library here: http://library.nyu.edu/

You can access NYU Tandon's Bern Dibner Library here: http://library.poly.edu/

RECOMMENDED READINGS are online journal articles provided in each lecture You can access NYU's central library here: http://library.nyu.edu/

You can access NYU Tandon's Bern Dibner Library here: http://library.poly.edu/

Assignments and Exams

Exams Administered and Proctored Online

Exams in this course are administered through NYU Classes. You are required to arrange an online proctor for your exams via ProctorU. More information on ProctorU and scheduling proctoring sessions can be found on Tandon Online's website.

Exams Administered On Paper and Proctored Remotely

Exams in this course are administered via paper and pencil. If you are not able to attend an exam session on-campus, you are required to secure in-person proctoring arrangements near your location. Tandon Online's website.

University Policies

Moses Center Statement of Disability

Academic accommodations are available for students with disabilities. Please contact the Moses Center for Students with Disabilities (212-998-4980 or mosescsd@nyu.edu) for further information. Students who are requesting academic accommodations are advised to reach out to the Moses Center as early as possible in the semester for assistance.

NYU Tandon School of Engineering Policies and Procedures on Academic Misconduct[1]

A. 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.

B. 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:

   a. 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.
   b. Fabrication: including but not limited to, falsifying experimental data and/or citations.
   c. 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.
   d. Unauthorized collaboration: working together on work that was meant to be done individually.
   e. Duplicating work: presenting for grading the same work for more than one project or in more than one class, unless express

[1] Excerpted from the [Tandon School of Engineering Student Code of Conduct]

and prior permission has been received from the course instructor(s) or research adviser involved.

f. Forgery: altering any academic document, including, but not limited to, academic records, admissions materials, or medical excuses.