Polytechnic School of Engineering, NYU
CS 6033: Design & Analysis of Algorithms 1
Course Syllabus
Fall 2018

Prof: Dr. Linda Sellie
Contact Information: Please send a message to me on NYU Classes. **Do not send email.**
OFFICE HOURS: Office hours will be Thursdays 12:00 - 2:00 and by appointment in my office 10.047 on the 10th floor of 2 Metrotech Center.
Class communication: We will use the course site on PIAZZA. NYU Classes is for the syllabus, posting homework assignments, lecture notes, and grade posting.

Catalog description: Review of basic data structures and mathematical tools. Data structures: priority queues, binary search trees, balanced search trees. B-trees. Algorithm design and analysis techniques illustrated in searching and sorting: heapsort, quicksort, sorting in linear time, medians and order statistics. Design and analysis techniques: dynamic programming, greedy algorithms. Graph algorithms: elementary graph algorithms (breadth-first search, depth-first search, topological sort, connected components, strongly connected components), minimum spanning trees, shortest paths. String algorithms. Geometric algorithms. Linear progsramming. Brief introduction to NP-completeness.

Prerequisites: You *must* have taken these courses to take this course.

- CS5403: Data Structures and Algorithms, or equivalent knowledge of fundamental data structures.
- CS6003: Foundations of Computer Science, or equivalent knowledge of discreet mathematics for computer science.
- A programming course beyond "Introduction to Programming".

Additionally, you should not take this course if you have taken a similar course with a B or better grade.

Textbook: Cormen, Leiserson, Rivest, and Stein, Introduction to Algorithms, 3rd Edition, MIT Press, 2009; ISBN-13: 9780262033848; ISBN: 0262033844. It is known as CLRS.

We have free access to CLRS on BOOKS24x7 on the NYU library web site
`http://library.nyu.edu`, search for books24x7).

Approximate Grading Scale Your final grade will be determined roughly as follows: homework assignments 10%, quizzes 15%, midterm 35%, and final 40%. (Extra consideration in determining your grade will be based by your exam scores, especially the final exam.)

Attendance at exams is mandatory. Make-up exams will only be given in the case of a emergency, such as illness, which must be documented, e.g. with a doctor's note. In such cases, you must notify me as early as possible, preferably before the exam is given. If you miss an exam without a valid excuse, you will receive a grade of *zero* for that exam. The exams will be closed book and no notes.

Course Work and Grading: Quizzes (approximately weekly) will be on the required background material of the course, and on topics covered in the homework assignments/lectures. The first quiz is optional. The grade for this quiz can replace the grade for any other quiz in the semester.

Homework assignments (approximately weekly) will be posted on NYU Classes. Announcements, and the occasional helpful hint will be posted on Piazza. You are responsible for being aware of any information posted there, so you should check it regularly.

Although the homework makes up a relatively small percentage of the final grade and is a lot of work, it is a key component to mastering the course material. Experience has shown that you will not do well on the exams if you have not done the homework. Please experiment with writing code for the algorithms; no formal programming assignments will be given.

Homework assignments are due an hour before the start of the lecture. **No** late homework assignments are accepted.

Academic Dishonesty: Cheating will not be tolerated. Absolutely no communication with other students is permitted on exams. I advise you that I will seek a F in the course for any cheating on an exam. So, if you copy a single answer from someone else, I will be seeking an F in the course for you. There are possible additional actions at my discretion including involving the CS department and the administration.

Please see the university policy:https://engineering.nyu.edu/academics/code-of-conduct/academic-dishonesty

Policy on Collaboration: You may discuss general approaches on how to do the homework assignments with other students. You may work with **one** other student to work out the details of the questions, and to write up the solution. If you work with another student, you **must** put both names and netID's on top of the assignment. Additionally, if you work with a partner, **only one** of you will *submit* the assignment on NYU Classes (but both of you are responsible to make sure it was submitted). If you do not have your name on top of the assignment, you will not receive any points for that assignment. If you work together, you must fully understand the work you submit. Your submission must be your (and your partner's) work. If there is any evidence that the work is not yours (and your partner's) work (such as copying from others, from the Internet, paying a third party to carry out the work, etc) it will be considered academic dishonesty. You will receive a 0 for the assignment, you will be reported to the department and the Dean of Student Affairs, and potentially receive a F for this course.(See http://cis.poly.edu/policies/)

APPROXIMATE SCHEDULE Please check for updates during the semester. Lectures slides will be posted on NYU classes a day or two after the lecture.
The precise order and content, especially of the later parts of the course, may change. We will work down the list of topics given in the syllabus.

- Introduction: What's an algorithm? Why do we want to study algorithms? Termination. Correctness. Performance. How to measure performance of an algorithm? Models of computation, abstract machines. RAM. Best-, worst-, and average-case performance. Review of asymptotic notation: big-O, big-Ω and big-Θ; little-o, and little-Ω. Chapters 1, 2 & 3.
- Review of basic data structures. Abstract data types (ADTs). Common ADTs: arrays, stacks, queues, linked lists, priority queues, heaps, heapsort. Chapters 10 and 6.
- Dictionaries ADT. Hashing. Balanced search trees (tentatively, 2-3 trees, 2-3-4 trees, and more generally (a, b)-trees, red-black trees). B-trees. Augmenting a balanced search tree. Chapters 11, 12, 13, 14, and 18.
- Implementation: using Union Find problem and Data Structures. Chapter 21.
- Divide-and-conquer algorithms. Review of recurrences and how to solve them. Master's theorem. Binary search. Mergesort. Quicksort. Median and order statistics. Deterministic linear-time selection. Fast integer multiplication (Karatsuba's algorithm). Fast matrix multiplication (Strassen's algorithm). Closest-pair problem. Chapter 4, 7, 9 and 33.4.
- Randomized algorithms and geometric algorithms (not covered as separate topics).

- Graph algorithms: elementary graph algorithms (breadth-first search, depth-first search, topological sort, connected components, strongly connected components), minimum spanning trees, shortest paths. Some graph algorithms will be presented later in the course as illustrations for different algorithm design paradigms. Chapters 22 and 23.
- Dynamic programming: Rod cutting. Matrix chain product. Longest common subsequence. Optimal binary search trees. Shortest path problems in graphs. Transitive closure. Chapters 15 and 24.
- Greedy algorithms: Activity selection. Huffman coding. Minimum spanning trees. Chapter 16.
- Undecidability and fundamentals of NP-completeness (both very briefly; one lecture). Chapter 34.