New York University Tandon School of Engineering
Computer Science Department
Course: CS-GY CS6033 INET Design and Analysis of Algorithms
Instructor: Dr. Raj Rajagopalan

Instructor contact info: srr355@nyu.edu
Phone: [available by email]
Office hours: TBD on Google Hangout or by Skype appointment

Course Pre-requisites

This is a graduate-level course in Algorithms. There are several topics you must be familiar with in order to successfully handle the course material.

* Programming. You should be a reasonably competent programmer in some high-level programming language (C,C++,Java,Python, etc). While no programming will be required in this course, without programming experience it is likely you will get the maximum benefit from the course.
* Basic data structures: stacks, queues, linked lists, hashing, priority queues (heaps), and hashing. How to measure runtime through the use of big-O. We offer CS5403 if you do not have this background. Similarly, the course is NOT intended for people who have taken and passed (with a grade of B or higher, or equivalent) an introductory undergraduate or graduate course on algorithms.
* Discrete math. In addition to basic algebra (especially logarithms and exponents), you should be familiar with discrete math, especially proof by induction, summation notation (and in particular summing arithmetic and geometric progressions), solving recurrence relations with Master's method, and summation notation. We offer CS6003 if you do not have this background.

It would be helpful to review/refresh the above background before the start this course.

Course Description This is a foundational course and a core requirement for a Masters in Computer Science. In this course we will cover the basics of asymptotic analysis of algorithm performance, tools for designing algorithms for various categories, and tools for comparing different algorithms for a given problem. The particular focus of this class is on thinking for yourself and you should be careful not to confuse quantity with quality.

Course Objectives
1. Learn the abstract algorithm model
2. Learn to use asymptotic analysis
3. Learn the use of algorithmic design strategies: divide-and-conquer, greedy, etc.
4. Learn to apply algorithmic design ideas to solve new problems.

Course Structure
This is an entirely online course, including all classes, assignments, and exams. The general weekly structure for each topic will look like below.
1. Watch the assigned pre-recorded videos for the week.
2. Read the assigned portion of the book.
3. Attempt the homework with the assigned team and submit it jointly.
4. Participate in a discussion of the solutions of the homework online.

Lectures

There will be a “synchronous” (i.e. live) lecture in most weeks on specific topics of relevance in that week. These lectures are tentatively scheduled for Wednesdays 8-1000PM Eastern Time, subject to this being mutually convenient to a majority of class students. In some weeks the class may be moved (or canceled in the rare case) depending on the professor’s travel schedule. Lectures will include material from the professor and discussion about homework.

Live lectures will follow a Socratic method and class participation is highly encouraged. The purpose of the lectures is to dive deep into a specific topic of relevance and gauge how well the material is being absorbed by the students.

There will be a tutorial section led by the TA roughly one hour every week at a time TBD.

Office hours

The professor will hold office hours on Google Hangout. Time TBD.

Online materials

There is a universe of materials available to you on the Internet in the form of presentations and videos. You are encouraged to look online and if you particularly like something you are invited to share that with the class or bring it to the professor’s attention.

Readings

The required text for the course is:

<table>
<thead>
<tr>
<th>Title</th>
<th>Algorithms 3rd Ed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Thomas H. Cormen; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein</td>
</tr>
<tr>
<td>Publisher</td>
<td>MIT Press</td>
</tr>
</tbody>
</table>
This is the standard textbook on Algorithms. You are strongly encouraged to purchase it and you will need to read a substantial portion of it. Be sure to make sure that you have the THIRD edition of the book as the chapter numbering and exercises are slightly changed. Time will only allow us to cover roughly half of this book’s chapters. Coverage of some of the later chapters in the book will depend on the maturity of the students.

Availability NYU bookstore, Dibner, Dibner reserves.

Course requirements

Your grade will be based on 25% homework, 25% midterm, 35% final and 15% term project. The instructor may assign at his discretion 2% on an “off the curve” basis for sustained class participation and 1% for submitting high quality homework answers.

Exams
There will be one midterm exam and one final exam. The mid-term will be soon after the halfway point in the semester and the final will be either during finals week or the last day of classes. These will be both online exams conducted as per university policy under the supervision of ProctorU. Missing an exam (without a valid excuse with a suitable supporting documentation as allowed by University rules) will result in a grade of ZERO for the exam and an automatic grade of F in the course. All exams will be online and closed book/notes.

Homework Assignments [25%]
These will typically be assigned from the textbook. There will be a homework assignment every two weeks and will be due ten days after issue.

For each homework assignment, you will be assigned to a random team of 2 or 3 students from the class. One submission is expected from each team and its members will share the resulting points equally. This is done to foster collaboration in an online class. Before doing the homework you should watch the video and read the associated part of the book. You should look at the problems at the end of each section in the book; these problems are meant to be very easy. If you have any questions about the material you should post a question online.
All homework assignments will be assigned and must be submitted in pdf format online. You must complete the homework on your own or with your assigned team. You must not use the internet to explicitly find the solutions. Every word of what you hand in must be your own. If it is not you are committing academic dishonesty.

Homework assignments submitted late will be penalized 1% per day with a maximum of 3 days. We will discuss some (not all) homework problems online. If possible some students may be asked to discuss their high quality solution to a homework problem in class. It is hoped that this method will help students learn how to create and present algorithms in a much better way than simply handing in the homework.

Course Project/Term Paper [15%]
All students will be required to submit a term paper of up to 10 pages (11pt with 1 inch margins) on a topic of their choice. Students can optionally pair up with at most two partners of their choice and submit term projects as a team. The list of topics will be posted at the beginning of the term. Every student can choose his/her topic from the list (a student may also propose a topic outside the list but must get approval from the Professor). Term paper will be due the last day of classes.

**Week 1** Introduction to Algorithm design
- Chapters 1, 2, and 3

**Week 2** Solving Recurrence Relations
- Chapter 4

**Week 3** Sorting
- Chapters 2.1, 2.3, 7

**Week 4** Average-case analysis
- Chapters 5, 7.3, 7.4, 6

**Week 5** Sorting II
- Chapter 8

**Week 6** Hashing and Randomized algorithms
- Chapters 12, 13, 11

**Week 7** Graphs: Introduction
- Chapters 22

**Week 8** Minimum Spanning Tree algorithms
- Chapters 21, 23

**Week 9** Shortest path algorithms
- Chapter 24
Week 10 Dynamic Programming
   • Chapters 15-1, 15-2, 15-3, 15-4

Week 11 String Algorithms
   • Chapter 32.1, 32.2

Moses Center Statement of Disability

If you are 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. Information about the Moses Center can be found at www.nyu.edu/csd. The Moses Center is located at 726 Broadway on the 2nd floor.

NYU School of Engineering Policies and Procedures on Academic Misconduct
   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:

   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.