General Information:

- Instructor: Xi Chen CSB 503, Office hours: MW 5:30pm -- 6:30pm
- Location: Mudd 1127
- Time: MW 7:10pm -- 8:25pm
- Sign up for Piazza using this link
- Textbook: Introduction to Algorithms (third edition), by Cormen, Leiserson, Rivest and Stein, with a list of errata given here
- More references and links will be posted on the Lectures page
- Supplementary reading:
  - The Design and Analysis of Computer Algorithms, by Aho, Hopcroft, and Ullman
  - Data Structures and Algorithms, by Aho, Hopcroft, and Ullman
  - Algorithms, by Dasgupta, Papadimitriou, and Vazirani
  - Algorithm Design, by Kleinberg and Tardos

Teaching Assistants:

- Irit Huq-Kuruvilla, Office hours: Friday 2pm -- 4pm (TA room), Email: ih2271@columbia.edu
- Ying Sheng, Office hours: Monday 9am -- 11am (TA room), Email: ys2982@columbia.edu
- Weston Jackson, Office hours: Tuesday 10am -- 12pm (TA room), Email: wjj2106@columbia.edu
- Edo Roth, Office hours: Tuesday 4pm -- 6pm (TA room), Email: enr2116@columbia.edu
- Kailash Karthik Meiyappan, Office hours: Thursday 3pm -- 5pm and Friday 10am -- 12pm (TA room): Email: kkm2142@columbia.edu

Course Description:

- In this course, we focus on principles and techniques that are widely used in the design and analysis of efficient computer algorithms. Topics covered include:
  - Asymptotics and recurrences
  - Sorting and searching
  - Greedy algorithms
  - Amortized analysis
  - Dynamic programming
  - Graph algorithms
  - Randomized algorithms
  - Approximation algorithms
  - NP-completeness
Grading:

- Homework (35%) [Best six out of seven], in-class midterm (30%), final (35%)

Prerequisites:

- The course is mostly theoretical, so you should feel comfortable reading and working with math and proofs. Some knowledge of discrete mathematics and basic probability (e.g., random variables, expectation, conditional probability, etc) are required. Some of the required mathematical background (including basic counting and probability) can be found in the appendix of the textbook.
- We will not use any particular programming language in the course, but will describe algorithms either in English or in a simple pseudocode that should be readable to anyone with a little programming experience.

Homework:

- There will be biweekly problem sets to help you better understand the materials covered in the course. They will be assigned on Mondays, posted here, and due two weeks later. (Leave your homework in one of the two boxes labelled Analysis of Algorithms section 1 and section H1 outside my office CSB 503 no later than 6pm on the due date.)
- Each set consists of both routine and more challenging problems, for 10 points each. You are encouraged to start early, and to make effective use of the office hours of the instructor and teaching assistants.
- Most of the problems ask you to either design an efficient algorithm for a specific problem, prove or disprove the correctness of a given algorithm, or to analyze the complexity of a given algorithm. Most of the problems require one or two key ideas. Be concise when writing up the solutions, but make sure to give enough details to clearly present those key ideas. Learn to mathematically reason about algorithms, and learn from the writing style of the textbook how to rigorously and succinctly describe an algorithm. Understandability will be an important factor considered in grading. You are discouraged from writing up long answers which you suspect are incorrect, in the hopes of picking up a point or two. You will get no point by doing this, and will receive a warning for the first violation. Any subsequent violation will be penalized by 10% off the whole set, due to the waste of TA time.

Homework policy:

- Late homeworks will not be accepted. Exceptions will be made only for exceptional circumstances (e.g., serious illness or family crisis).
- The homework submitted must be written by yourself independently, without looking at anybody else’s solutions. You may discuss the problems with fellow students taking the class, teaching assistants, and the instructor. If you collaborate, you must acknowledge clearly, at the beginning of each problem, the people you discussed with on that problem. You are expected to fully understand every single step of the solution, and must write up the solutions by yourself independently.
- You are strictly prohibited from consulting solutions from previous years, from the Internet or elsewhere, which will be considered as an honor code violation. You are expected to adhere to the Academic Integrity policy of the Computer Science Department.