Introduction to Analysis of Algorithms

CS 4820/5820 Spring 2022 Syllabus

General Information

- **Instructor:** Eshan Chattopadhyay, Gates Hall 319, eshan@cs.cornell.edu.

  Office hours: Monday 10:30am-11:30am, Thursday 1:30pm-2:30pm.
- **Lectures:** MWF 9:05am-9:55am, Uris Hall G01.

Course Description

This course develops techniques used in the design and analysis of algorithms, with an emphasis on problems arising in computing applications. Example applications are drawn from systems and networks, artificial intelligence, computer vision, data mining, and computational biology. This course covers four major algorithm design techniques (greedy algorithms, divide and conquer, dynamic programming, and network flow), computability theory focusing on undecidability, computational complexity focusing on NP-completeness, and algorithmic techniques for intractable problems, including identification of structured special cases, approximation algorithms, and local search heuristics. This course continues to build on work in previous courses on proofwriting and asymptotic runtime analysis of algorithms.

Learning Objectives

On completing this course, students should be able to:

- Identify problems solvable with a **greedy algorithm**, design and prove the correctness of such an algorithm, and supply asymptotic running time for a variety of given algorithms.
- Recognize problems to which **divide and conquer** or **dynamic programming** approaches may apply, design algorithms with these approaches, and analyze their computational efficiency;
- Apply **randomization** to produce tractable algorithms for several specific computationally challenging problems;
- Reduce resource management as well as partition problems to **network flow or cut problems**, implement correct strategies for finding optimal flows/cuts, and understand the properties of these strategies;
- Recognize whether or not certain problems are computationally intractible (e.g. **NP-complete, uncomputable**), and use reductions to known problems to prove intractability;

- Use **approximation algorithms** to efficiently produce near-optimal solutions for intractable problems, and bound how close these algorithms are to being optimal;
- Use **online algorithms** to produce near-optimal solutions when only partial information about a problem is available, and bound how close these algorithms are to being optimal; and
- Be able to recognize, implement, and understand the properties of several famous and important algorithms including
  - **Gale-Shapley method** for stable matchings,
  - **Prim's and Kruskal's algorithms** for finding minimum spanning trees,
  - **Bellman-Ford's algorithm** for finding shortest paths in a graph, and
  - **Ford-Fulkerson's algorithm** for finding max flows in networks.

Course Material

The textbook for the course is *Algorithm Design* by Jon Kleinberg and Eva Tardos (available at Cornell Store). Although this book was designed for this course, there will be topics covered in lecture that are not in the text and there will be topics in the text that are not covered in lecture. You are responsible for topics covered in lecture and for any assigned reading in the text.

The following books are also useful references.

- T. Cormen, C. Leiserson, R. Rivest. *Introduction to Algorithms.*
- S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms.*
- A. Aho, J. Hopcroft, J. Ullman. *The Design and Analysis of Computer Algorithms.*
- M. Garey and D. Johnson. *Computers and Intractability.*
- D. Kozen. *The Design and Analysis of Algorithms.*

Prerequisites

The prerequisites for the course are, either having an A- or better in both CS 2800 and CS 2110, or having successfully completed all three of CS 2800, CS 2110, and CS 3110. We assume that everyone is familiar with the material in CS 2110, CS 3110, and CS 2800, and we will use it as necessary in CS 4820. This includes elementary data structures, probability (conditional probability, expectation, variance), sorting, and basic terminology involving graphs (including the concepts of depth-first search and breadth-first search), and coding (in Java, or one of the other languages supported – see Programming Assignment Instructions). Some of these are reviewed in the text. The lectures and homework involve the analysis of algorithms at a fairly mathematical level. A few of the homework exercises consist of writing code in Java. We expect everyone to be **comfortable reading and writing proofs at the level of CS 2800, as well as writing code in Java**.

Grading

Your grade will based on weekly homework, participation (based on class participation, participation on Ed, and completion of a course evaluation), two prelims, and one final exam. Each of these components have the following weights:

- homework: 35%
- participation: 5%
- prelim 1: 15%
- prelim 2: 15%
- final exam (cumulative): 30%

Homework

Homework is an important part of the course. We will have weekly homework assignments. All homework assignments will be posted on Canvas. Most homework assignments will be due on Thursday at 11:59pm.

Graduate students (i.e., those students enrolled in CS 5820) will be assigned some advanced problems over the course of the semester, that require students to synthesize different components of the course material.

Typesetting

We will require problem sets to be typeset and submitted as a PDF. This requirement is for everyone's benefit. In general, we recommend that you first develop your solutions in draft form, and then write or type your solution in a concise way. Typesetting not only makes the last step essential (instead of handing in solution in draft form), it also makes it much easier for you to edit and improve your writeup, as well as easier for your TAs to read your proofs. It is up to you which tool you use; though we recommend LaTeX, tools like the Equation Editor in Microsoft Word can be surprisingly effective as an alternative.

For some proofs or writeups, it may be helpful to use a figure to explain your thinking more concisely. This is encouraged! Again, it is up to you how you want to include that in your writeup, whether it is a picture of a drawing in your notebook that you took with your phone or something you made digitally, as long as the figure was produced by you personally and is clear enough to see, it's a great idea to include it.

Late Submissions

You have **six late days**. Grades from a late submission count only if you have late days left. You can use *at most three late days per homework*. The purpose of late days is to help handle unforseen circumstances such as illness, interviews, or conflicting deadlines that may prevent you from turning in your assignment on time. **Requests for additional late days will not be granted except in truly extraordinary circumstances**, such as hospitalization or the death of a loved one. If such an emergency arises, contact the instructor as soon as possible.

Collaboration

In the real world of algorithms research, collaboration and conversation is an important part of how ideas get generated. So too in this course; we encourage you to discuss with your peers in the course to brainstorm ideas for how to get through homework. However, **your solution must be written up completely on your own; you are not allowed to share digital or written notes or images of your work in any form with each other**. Just like in research, your work must also include acknowledgements of all students with whom you collaborated. Both the physical or digital distribution of information about solutions and the failure to acknowledge collaborators are serious violations of academic integrity.

Admissible Resources

For the homework, **it is not admissible to use resources beyond course material and student discussions**. In particular, you may not use Wikipedia, or search the Web, or look at any textbook, other than the ones assigned/recommended in the course. Using such additional resources is a violation of academic integrity. If you feel the resources available to you are insufficient, talk to course staff or ask questions on Ed.

Advice for Success

Algorithms assignments can often require creative insights and complex proofs beyond what previous courses have required. Here are a few tips for succeeding in your writeups:

- **Start your assignments early**. Even if you aren't writing anything down yet, looking over the problem set well in advance of the due date can ensure you have enough time to brainstorm possible solutions and to clear up confusion about how to interpret a problem. Creativity doesn't work well on a deadline.
- **Talk with classmates at a similar level about ideas**. As previously stated, while you cannot share physical or digital solutions of any kind to these problems, we actively encourage you to talk to classmates while you work through them. In particular, we recommend finding a group of students to meet with throughout the semester in advance of the deadline to talk about ideas. For best results, make sure those students are at the same level of understanding of the material as you; talking through your ideas with colleagues with a similar

level of understanding will make talking through ideas with each other easier and more equitable, and is more likely to leave you prepared for course exams.

- **Ask questions in class, in office hours, and on Ed Discussions.** The material in this class moves quickly and is often cumulative. If you find yourself scratching your head after a lecture, even after consulting the textbook and course notes, you're certainly not alone, and it's better to seek help then than to wait until you are more confused.

Course Conduct

We understand that our members represent a rich variety of backgrounds and perspectives. Cornell University is committed to providing an atmosphere for learning that respects diversity. We expect students to communicate in a respectful manner with the instructors, course staff, and fellow students, in a way the honors the unique experiences, values, and beliefs represented by different members of our community.

Academic Integrity

Any violation of academic integrity will be severely penalized. You are allowed to collaborate on the homework to the extent of formulating ideas as a group. However, you are expected to write up (and understand) the homework on your own, and you should acknowledge the names of the students with whom you collaborated.

From Cornell's code of academic integrity:

> Absolute integrity is expected of every Cornell student in all academic undertakings. Integrity entails a firm adherence to a set of values, and the values most essential to an academic community are grounded on the concept of honesty with respect to the intellectual efforts of oneself and others. Academic integrity is expected not only in formal coursework situations, but in all University relationships and interactions connected to the educational process, including the use of University resources. [...]
>
> A Cornell student's submission of work for academic credit indicates that the work is the student's own. All outside assistance should be acknowledged, and the student's academic position truthfully reported at all times. In addition, Cornell students have a right to expect academic integrity from each of their peers.

Course Material Copyright

Course materials posted on Canvas, gradescope, or Ed Discussions are intellectual property belonging to the author. Students are not permitted to buy or sell any course materials without the express permission of the instructor. Such unauthorized behavior constitutes academic misconduct.

Inclusiveness

You should expect and demand to be treated by your classmates and the course staff with respect. You belong here, and we are here to help you learn and enjoy this course. If any incident occurs that challenges this commitment to a supportive and inclusive environment, please let the instructor or a TA know so that the issue can be addressed. We are personally committed to this, and subscribe to the Computer Science Department's Values of Inclusion.

Accommodations

This course complies with the Cornell University policy and equal access laws to ensure that students with disabilities can still participate fully in this course. Requests for academic accommodations should be made during the first three weeks of the semester, except for unusual circumstances, so arrangements can be made as soon as possible. Students are encouraged to register with Student Disability Services, as we may require verification of eligibility to provide appropriate accommodations.