CpSc 8400: Design and Analysis of Algorithms

Instructor: Dr. Brian Dean | Fall 2024
Handout 1: Course Syllabus | Online / Coursera

1 Overview and Course Goals

The study of algorithms is a significant part of the foundation for the discipline of computing. Over the past several decades, research in algorithmic computer science has advanced at a rapid pace its contributions have had a profound impact on almost every area of science and industry. In this graduate-level course, we aim to provide a modern introduction to the study of algorithms that is both broad and deep. The primary goals of the course are:

- to become proficient in the application of fundamental algorithm design techniques, as well as the main tools used in the analysis of algorithms,
- to study and analyze different algorithms for many of the most common types of "standard" algorithmic problems, and
- to improve one's ability to implement algorithmic ideas in code.

2 Prerequisites

This class is designed to be reasonably approachable for a wide range of potential students, even those with a more distant background in computing / mathematics. Nonetheless, some prior knowledge in areas related to the class will be helpful. In terms of mathematics, we will use very basic terminology and ideas from probability theory and discrete math / combinatorics, and certain modules of the course will use terminology and ideas from calculus (e.g., when we study gradient descent in optimization) and linear algebra. Depending on their background, students may find it necessary to review some of these concepts on an as-needed basis throughout the course. Prior familiarity with algorithms and data structures is of course also helpful but not assumed. Coding will be involved in lectures and assignments, so students are assumed to know how to write code in one of the supported languages of the course (C/C++, Java, or Python).

3 Instructor Information

Instructor: Dr. Brian Dean
Email: mscscoursera@clemson.edu
Office Hours: To be determined

4 Tentative List of Topics

The field of algorithms includes far too much interesting material than we can hope to cover in just a single course. In order to cover as much ground as possible, we will move at a fairly fast pace through the semester. The following is a list of the 10 high-level "modules" that will make up the course.

- **Fundamentals.** How to analyze performance of algorithms; models of computation; basic data structures; algorithm design techniques; common sorting algorithms.
- **Randomization.** Methods for analyzing expected running time and average-case performance; randomized quicksort and quickselect; examples of randomized algorithms and data structures; "high probability" bounds.
- **Recursion.** Techniques for analyzing recursive (e.g., "divide and conquer") algorithms; recursive thinking; examples of recursion in algorithm design.
- **Data Structure Fundamentals.** Priority queues; Binary search trees for representing sets, maps, and sequences; randomized and amortized tree balancing mechanisms; sweep line methods.
- **Further Ideas in Data Structures.** Multi-dimensional search structures; amortization; memory-conscious structures (e.g., B-trees); skip lists.
- **Hashing.** Hash tables and distributed hash tables; universal hashing; polynomial hashing of large objects; applications of hashing across different computing subfields.
- **Discrete Optimization**. Greedy algorithms; dynamic programming; heuristics based on iterative refinement; multi-scale methods; pruned exhaustive search; hard problems and approximation algorithms.
- **Continuous Optimization**. Gradient descent and Newton's method; stochastic gradient descent; gradient-free methods; continuous relaxations of discrete problems; convexity; optimization with constraints; common types of optimization problems.
- **Graph Algorithms**. Connectivity and related problems; shortest paths; matchings.
- **Networks and Data Analytics.** Clustering; centrality; approaches based on random walks; use of linear algebra in data analytics; multiplicative weight updates (e.g., learning from experts and boosting).

5 Grading

The following items contribute to your grade in this course:

- **Practice Exercises (10%).** In many modules you will be given short exercises that reinforce the course material. Some of these can be taken multiple times (problems on these are often randomized).

- **Programming Assignments (4 × 10% each).** In each of these assignments, you will submit code in one of the supported languages of the class (C/C++, Java, or Python) to solve one or more algorithmic problems related to the current module under study. Solutions will be automatically graded, with instant feedback available.
- **Design and Analysis Assignments (2 × 15% each).** These assignments require students to design and analyze algorithms for solving simple problems. Solutions must be *typeset*. Algorithms should be described concisely at a high level (but still in sufficient detail) using a combination of either prose or pseudocode. Analysis of these algorithms should make proper use of terminology, notation, and mathematical concepts.
- **Final Exam (20%).** A final exam will be held at the end of the semester. It is cumulative, covering all material from the entire course.

Final letter grades are assigned on a scale of A (90 or above), B (75-89), C (60-74), and F (below 60). Grades on individual items above may possibly be curved upwards.

As a general policy, an assignment submitted late or lack or participation in a quiz/exam will result in a grade of zero for that particular item. However, the instructor acknowledges that additional flexibility may be required in some situations (e.g., if a medical issue arises). Please contact the course staff (as far in advance as possible) in the event of an exceptional circumstance in which you may require special accommodation.

If you feel one of the design and analysis assignments graded by a TA has been graded incorrectly, you may request a re-grade from the instructor. However, note that the entire assignment will be re-graded (which could potentially cause your grade to drop).

6 Course Materials

There is no required text for this course. Lecture slides will be available alongside lecture videos on the course website.

7 Course Delivery

This class is taught on-line in an asynchronous fashion. Course staff will be available to help during both scheduled office hours (e.g., on zoom) and also email / discussion board interaction.

8 Course Conduct

- **Attendance.** Since this is an on-line asynchronous class, students are responsible for keeping up with the course material and turning in assignments on time. A suggested schedule mapping course modules to calendar ranges will be provided; as some modules may be opened early, students should feel welcome to work ahead of schedule if they find this beneficial.
- **Academic Integrity.** The official university policy on academic integrity is as follows: "As members of the Clemson University community, we have inherited Thomas Green Clemson's vision of this institution as a "high seminary of learning". Fundamental to this vision is

a mutual commitment to truthfulness, honor, and responsibility, without which we cannot earn the trust and respect of others. Furthermore, we recognize that academic dishonesty detracts from the value of a Clemson degree. Therefore, we shall not tolerate lying, cheating, or stealing in any form." The instructor of this class values academic integrity *very* highly, and will report any infractions for further disciplinary action. Note that one of the most common infractions seen in the past in this class is plagiarism (passing off the work of others as your own); if you have any concerns related to what constitutes plagiarism or about any other issues related to academic integrity, please contact the course staff.

- **Use of Previous Course Material.** Since some problems are re-used from prior semesters in this course, it is not permitted to consult any homework solutions from previous years (written by the instructor or by students who have taken the course in previous years). Do not ask students having previously taken the course for specific details on individual homework problems.

- **Use of Web Content.** Please feel free to consult the web for *general* information (e.g., useful mathematical formulae, definitions, supplemental reading on course material). However, you are not to go looking on the web for information *specifically* related to any particular homework problems. The purpose of the homework is to help you develop your skills at problem-solving, not web searching. Use your best judgment as to what is acceptable and what is not, or if there is any question, you can always ask the instructor.

- **Use of Generative AI.** You are welcome to use generative AI as a tool to help with "low level" technical issues involving programming assignments (e.g., language syntax, examples of usage of library functions). However, you should not be asking generative AI to write code the implements the core logic behind the algorithms and data structures required for your solution. The point of the homework assignments is for you to gain experience translating ideas into code, not just to practice your prompt engineering. Similarly, in write-ups for the algorithm design and analysis assignments, you can use generative AI as a tool to help with low-level phrasing and grammar, but the high-level ideas being expressed in your solution should be yours. On the final exam, generative AI is not to be used.

- **Collaboration.** Collaboration on homework is encouraged to the extent that it is carried out appropriately — with all students involved contributing equally to a discussion of how a concept works or how a problem should be solved, then working independently on writing up solutions. Solution write-ups and code must always be your own individual work. If you do use some material from external sources like books or web pages, you must include appropriate citations for your sources. In addition, on every assignment please include a list of any students with whom you have collaborated.

- **Title IX.** Clemson University is committed to a policy of equal opportunity for all persons and does not discriminate on the basis of race, color, religion, sex, sexual orientation, gender, pregnancy, national origin, age, disability, veteran's status, genetic information or protected activity (e.g., opposition to prohibited discrimination or participation in any complaint process, etc.) in employment, educational programs and activities, admissions and financial aid. This includes a prohibition against sexual harassment and sexual violence as mandated by Title IX of the Education Amendments of 1972. The University is committed to combatting sexual harassment and sexual violence. As a result, you should know that University faculty and staff members who work directly with students are required to report any instances of sexual harassment and sexual violence, to the University's Title IX Coordinator. What this means is

that as your professor, I am required to report any incidents of sexual harassment, sexual violence or misconduct, stalking, domestic and/or relationship violence that are directly reported to me, or of which I am somehow made aware. There are two important exceptions to this requirement about which you should be aware: Confidential Resources and facilitators of sexual awareness programs such as "Take Back the Night and Aspire to be Well" when acting in those capacities, are not required to report incidents of sexual discrimination. Another important exception to the reporting requirement exists for academic work. Disclosures about sexual harassment, sexual violence, stalking, domestic and/or relationship violence that are shared as part of an academic project, a research project, classroom discussion, or course assignment, are not required to be disclosed to the University's Title IX Coordinator. This policy is located at `http://www.clemson.edu/campus-life/campus-services/access/title-ix/`. Ms. Alesia Smith is the Executive Director for Equity Compliance and the Title IX Coordinator. Her office is located at 223 Holtzendorff Hall, phone number is 864.656.3181, and email address is `alesias@clemson.edu`.

9 Accessibility

Clemson University values the diversity of our student body as a strength and a critical component of our dynamic community. Students with disabilities or temporary injuries/conditions may require accommodations due to barriers in the structure of facilities, course design, technology used for curricular purposes, or other campus resources. Students who experience a barrier to full access to a class should let the professor know, and make an appointment to meet with a staff member in Student Accessibility Services as soon as possible. You can make an appointment by calling 864-656-6848 or by emailing `studentaccess@lists.clemson.edu`. Students who receive Academic Access Letters are strongly encouraged to request, obtain and present these to their professors as early in the semester as possible so that accommodations can be made in a timely manner. It is the student's responsibility to follow this process each semester. You can access further information here: `http://www.clemson.edu/campus-life/campusservices/sds/`

10 Syllabus Modifications

Specific lecture topics and assignment details are potentially subject to change. The course syllabus is a general plan for the course. Should deviations be necessary, they will be announced to the class.