COS 454/554: Data Structures and Algorithms

Sudarshan S. Chawathe

University of Maine

Fall 2022

This course is an introduction to algorithms. Data structures play an important role, both in enabling efficient algorithms and in using others for their own implementation. Topics include the analysis of algorithms (analytical and experimental), algorithm design techniques (such as dynamic programming), advanced data structures (such as Fibonacci heaps), algorithms for specific problems (such as shortest paths in graphs, and string matching), and an introduction to NP completeness and related topics. An important theme is abstraction and its application to programming.

Prerequisites: COS 226 (data structures); COS 250 (discrete structures); programming maturity.

COS 554 is the graduate version of this course, which shares most class meetings and coursework with the undergraduate version (COS 454) but that includes additional coursework and is assessed to a higher standard.

News and Reminders:

- Until recently, this course was numbered as **COS 350**. Please bear this in mind when referring to or searching for older material.[1]
- Some sections below point to material in separate documents that are found on the class Web site, linked from the online version of this document.
- The most recent version of this document may be found at `http://chaw.eip10.org/cos454/`.
- Please use the PDF version of this document for printing and reference: `cos454.pdf`

Goals and Outcomes

Goals

- Learn several well-known algorithms.
- Learn some of the major algorithm design techniques.
- Continue study of data structures and, in particular, their relation to algorithms.
- Gain experience in implementing algorithms that are described abstractly.
- Study abstractions of a computer that permit simplified analysis of the running times and other characteristics of algorithms.
- Develop the ability to perform simple analyses of algorithms using such abstractions.
- Understand the growth rates of functions and learn how to bound functions that describe running times and other properties of algorithms.
- Understand the basics of computational complexity classes, and the class of NP complete problems.
- Gain experience in conducting and documenting experimental studies.
- Improve programming skills.
- Improve communication skills, with particular emphasis on written communication and, further, well-written programs.

[1] This is no hoax, nor joke, although 454 is in fact both a *hoax* and a *joke* (Smith) number. See `https://oeis.org/A019506` `https://oeis.org/A006753`.

Student Learning Outcomes

Upon successful completion of this course, students should be able to

- List well-known algorithmic problems and explain them in abstract and concrete forms.
- Explain the workings of well-known algorithms (and associated data structures) for solving such problems.
- Describe various algorithm design techniques and the kinds of problems they address.
- Explain the benefits and drawbacks of different abstractions, and understand how they map to actual computing environments.
- Perform simple analyses of algorithms.
- Determine suitable algorithms for solving a given concrete or abstract problem by connecting it to well-known abstract problems.
- Explain the significance of the class of NP-complete problems and recognize members of that class.
- Effectively read suitable publications related to the topic.
- Use resources such as others' code and writing in an ethical and professional manner.
- Contribute to the body of knowledge at an undergraduate level.
- Analyze the running times of programs and abstractly described algorithms using simple methods.
- Perform simple experimental studies of programs.
- Program with attention to community standards and good practices.
- Communicate their programming work effectively.
- Meet Quantitative Literacy General Education requirements, such as being able to [following text is from U. Maine Gen. Ed. documents]:
  - Translate problems from everyday spoken and written language to appropriate quantitative questions.
  - Interpret quantitative information from formulas, graphs, tables, schematics, simulations, and visualizations, and draw inferences from that information.
  - Solve problems using arithmetical, algebraic, geometrical, statistical, or computational methods.
  - Analyze answers to quantitative problems in order to determine reasonableness. Suggest alternative approaches if necessary.
  - Represent quantitative information symbolically, visually, and numerically.
  - Present quantitative results in context using everyday spoken and written language as well as using formulas, graphs, tables, schematics, simulations, and visualizations.

Contact Information

Class meetings:

Time: Mondays, Wednesdays, and Fridays, 09:00–09:50 a.m.
Location: Boardman Hall, Room 115.

Instructor: Sudarshan S. Chawathe

Office: Boardman Hall, Room 329.
Office hours: (Please check for changes.)
Mondays, Wednesdays, and Fridays; 08:15 a.m.–08:45 a.m. and 09:55 a.m.–10:15 a.m. (No appointment needed, but advance notification is useful.)
Others by appointment, possibly remote/online.
Phone: +1-207-581-3930.
Please avoid calling except for truly urgent matters.
Email: `sudarshan.chawathe@maine.edu`
Use email only for messages unsuitable for the discussion forum. (See below.) Please use only this email address and put the string *COS454* near the beginning of the Subject header of the message. *Responses to all other messages may be very significantly delayed.*
Web: `http://chaw.eip10.org/`.

Online Resources

Class Web site:
`http://chaw.eip10.org/cos454/`
We will use the class Web site for posting assignments, readings, notes, and other material. Please monitor it.

Class discussion forum: We will use the university's *Brightspace* installation for class discussions outside class meeting times.

Class mailing list: *Please make sure you are on the class mailing list.* The mailing list will use the email address for each student as recorded in the official university records (*MaineStreet* system). We will use this mailing list only for urgent messages because all other messages will go on the class discussion forum. I anticipate fewer than a dozen messages on this list over the semester.

Grading Scheme

Likely to change based on class discussions during the first week of classes.

Grade components: *Students are expected to complete and submit all assigned coursework in good faith; those who fail to do so will earn a* ***failing grade, regardless of overall numerical score****.* Cut-offs for final letter grades D, C, B, A are, respectively, 35, 55, 70, and 85.

component	% of grade
class participation	5
class exercises	5
online quizzes	10
homeworks	20
two quizzes	10
two midterm exams	30
final exam	20

Class participation: Students are expected to contribute to learning by asking questions and making relevant comments in class and participating in the specified online components of the class. Quality is more important than quantity. Disruptive activity contributes negatively. See policies below.

Classroom exercises: Our work in the classroom will include a number of short group exercises, meant to solidify understanding of the concepts being discussed. One or more such exercises are likely to be part of most class meetings. The exercises will be graded primarily for effort, group work, and other contributions, and less so for simple correctness.

Homeworks: Homeworks include programming and non-programming ones, often mixed. No collaboration is permitted. Everyone is encouraged to discuss the problems and solution strategies *at a high level*, but the final solution and details must be individual work. If the boundary between permissible and non-permissible interactions is unclear, please ask for clarifications.

Exams and quizzes: All exams and quizzes are *open book, open notes.* You are free to bring with you any resources that you find useful. However, no communications are permitted other than between students and me. The use of computers during exams is strongly discouraged, but brief use may be permitted provided it does not cause a disturbance, at the discretion of the proctor. You may use the Internet, but only as a library to look up material you may find useful. Ask for clarifications in case of any doubt. The exams are designed to require no equipment other than a pen and paper, along with the textbook and assigned readings.

Policies

Fall 2022 COVID-19 issues: Everyone is expected to follow the University policies related to the COVID-19 pandemic. See `https://umaine.edu/return`.

Due dates: All due dates and times, as announced in class, are strict, to the second. If you believe your work was delayed by truly exceptional circumstances, let me know as soon as those circumstances are known to you and I will try to make a fair allowance. However, *the default is that you get a zero if you don't turn in the work on time*, and fail the class if you don't turn it in at all (cf. Grade Components above).

Attendance: Although I expect students to attend all class meetings, I will not be taking attendance. *If you miss a class meeting, you are responsible for catching up on the lost material, including any important announcements made in class, on your own.* If you have a valid reason for missing a class, let me know early and I will try to help you make up the class. There will be no make-up exams or quizzes. A missed test earns zero credit. If you have a valid reason for missing a test, let me know as early as that reason is known to you and I will make a fair allowance but there will be no make-up tests in any case.

Classroom activities: This course is based on an active learning format, so effective classroom activities are critical to its success. Students are expected to contribute to their own learning and that of their classmates, and to devote 100% of their attention to these activities while in class. On a similar note, all electronic and other distractions (computers, phones, assorted gizmos, etc.) must be completely silenced and put away for the entire duration of the class. (Students who need any such devices for disability accommodations should follow the guidelines outlined below. Others who need any accommodation in this regard due to special circumstances should make advance arrangements with the instructor.) No food or drink is allowed in class, other than water, tea, coffee, and similar, in a spill-proof container. Students who violate these rules or otherwise cause distractions in class will be asked to leave with *no warning*; habitual violators will face disciplinary action.

Office hours: All students are encouraged to make use of office hours to further their learning, obtain assistance on homework assignments, obtain feedback on their class performance, etc. However, office hours are not to be used as a substitute for attending and participating in class meetings (see above). Similarly, assistance with homework assignments will be limited to what is appropriate based on fairness to all; students are expected to demonstrate substantial effort on the assignment before seeking assistance.

Make-up classes: I may have to reschedule a few classes due to my other professional commitments. I will make every attempt to minimize the number of such occurrences and to reschedule for a time that works for most students. Further, I will make sure no student is penalized by such occurrences.

University of Maine administrative policy statements: [Verbatim, standard wording from `https://umaine.edu/citl/teaching-resources-2/required-syllabus-information/`. Please refer to that site for further details.]

Academic Honesty Statement Academic honesty is very important. It is dishonest to cheat on exams, to copy term papers, to submit papers written by another person, to fake experimental results, or to copy or reword parts of books or articles into your own papers without appropriately citing the source. Students committing or aiding in any of these violations may be given failing grades for an assignment or for an entire course, at the discretion of the instructor. In addition to any academic action taken by an instructor, these violations are also subject to action under the University of Maine Student Conduct Code. The maximum possible sanction under the student conduct code is dismissal from the University.

Students Accessibility Services Statement If you have a disability for which you may be requesting an accommodation, please contact Student Accessibility Services, 121 East Annex, 581.2319, as early as possible in the term. Students who have already been approved for accommodations by SAS and have a current accommodation letter should meet with me (the instructor of the course) privately as soon as possible.

Course Schedule Disclaimer (Disruption Clause) In the event of an extended disruption of normal classroom activities, the format for this course may be modified to enable its completion within its programmed time frame. In that event, you will be provided an addendum to the syllabus that will supersede this version.

Observance of Religious Holidays/Events The University of Maine recognizes that when students are observing significant religious holidays, some may be unable to attend classes or labs, study, take tests, or work on other assignments. If they provide adequate notice (at least one week and longer if at all possible), these students are allowed to make up course requirements as long as this effort does not create an unreasonable burden upon the instructor, department or University. At the discretion of the instructor, such coursework could be due before or after the examination or assignment. No adverse or prejudicial effects shall result to a student's grade for the examination, study, or course requirement on the day of religious observance. The student shall not be marked absent from the class due to observing a significant religious holiday. In the case of an internship or clinical, students should refer to the applicable policy in place by the employer or site.

Sexual Violence Policy Sexual Discrimination Reporting

The University of Maine is committed to making campus a safe place for students. Because of this commitment, if you tell a teacher about an experience of sexual assault, sexual harassment, stalking, relationship abuse (dating violence and domestic violence), sexual misconduct or any form of gender discrimination involving members of the campus, your teacher is required to report this information to the campus Office of Sexual Assault & Violence Prevention or the Office of Equal Opportunity.

If you want to talk in confidence to someone about an experience of sexual discrimination, please contact these resources:

For confidential resources on campus: Counseling Center: 207-581-1392 or Cutler Health Center: at 207-581-4000. For confidential resources off campus: Rape Response Services: 1-800-310-0000 or Partners for Peace: 1-800-863-9909.

Other resources: The resources listed below can offer support but may have to report the incident to others who can help:

For support services on campus: Office of Sexual Assault & Violence Prevention: 207-581-1406, Office of Community Standards: 207-581-1409, University of Maine Police: 207-581-4040 or 911. Or see the OSAVP website for a complete list of services at http://www.umaine.edu/osavp/

Programming

This course focuses on high-level concepts that are mostly oblivious to choices of programming languages and environments. However, in order to provide concrete realizations of these concepts, we will use Python as the primary programming environment and a POSIX environment as the primary operating system. Submissions will be in the form of packaged, well documented *source* files. *Proper documentation and packaging of source code and other material is a crucial component of assigned work and submissions failing in this regard will receive no credit.*

Programming Environment and Tools: You are free to choose details such as operating system, development environment, and editor based on your preferences. However, no matter what you use, the submission should be a *source*-code package that works on the host *aturing* (see below). Further details on the packaging, submission, and testing procedure will be provided in class and on the discussion forum.

Other Languages: If you prefer to use other programming languages or systems, please contact me by the second class meeting. I am quite open to the idea, and encourage interested students to explore it further. However, please check with me very early in the semester so that we can determine the specifics to make sure your submissions can be tested and graded fairly. You should avail of this option only if you are confident enough of your programming skills to not require any programming help, and are prepared to take on additional work. *This option is designed for students who are proficient in Python and wish to use this opportunity to master another language, not for students weak in Python and who wish to avoid them.* Anyone granted this option will still be responsible for all the material related to the default languages and systems used in the course.

Literate Programming: All submitted work must use a *literate programming style*: Your programs must be designed with *a human as the intended reader*, although they must also compile and run correctly. *Programs that do not meet this requirement are likely to receive a zero score with no further consideration.* Details will be discussed in class.

Class accounts: Shell accounts will be generated on the host `aturing.umcs.maine.edu` based on registration records. These accounts are required for successful completion of homeworks and other assignments. You should be able to access your accounts from anywhere on the Internet by using *ssh*. On most Unix-like hosts (GNU/Linux, Mac OS), the command `ssh -l` *`username`* `aturing.umcs.maine.edu` should suffice. For Windows hosts, the freely available *Putty* program works well.

Schedule

A rigid schedule is not conducive to effective learning, since it would limit our flexibility in exploring ideas as they arise in class. The actual schedule (both the timing and the selection of topics) will be determined by in-class interactions. Nevertheless, a partial and *approximate* schedule, to serve as a baseline, appears in Figure 1; it will be updated as we progress. Please use it only as a rough guide to plan your studies. *Do not use it to schedule travel or other events.* If you need a definite answer on when something will or will not occur, you should check with me.

At the beginning and end of each class, I typically announce the topics and textbook sections covered in that class and those due at the next class. It is important that students read the material *before* the class in which it is discussed and, in general, keep up with readings and studies.

Textbook and Readings

Textbook: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 4th edition, 2022. *Please note the edition and year.* The university bookstore carries this book, which is a *required textbook* for this course. The book is very popular and there are many resources on the Web. These resources include solutions to exercises by the authors and others, video lectures and tutorials, and more. You are welcome, and encouraged, to use these resources (unless specifically directed otherwise), and to share and discuss them with classmates on the discussion forum. However, *you must prominently attribute any help from such or other resources in all your work.* Failure to do so is a serious offense (see policies).

Readings: A few supplemental readings will be added here based on class preferences.

Exercises, Homeworks, Tests, and Notes

Material will appear here as we move along the semester.

It may be useful to refer to material from the previous sessions (recursively): `http://chaw.eip10.org/202109/cos350/`. However please bear in mind that each session is different based on a variety of factors (most recently pandemic-related), and this one is much more so due to several other significant changes.

- Class exercises:
  - Class Exercise 1: `hwq/ce01.pdf`.
  - Class Exercise 2: `hwq/ce02.pdf`.
  - Class Exercise 3: `hwq/ce03.pdf`.
  - Class Exercise 4: `hwq/ce04.pdf`.
  - Class Exercise 5: `hwq/ce05.pdf`.
  - Class Exercise 6: `hwq/ce06.pdf`.
  - Class Exercise 7: `hwq/ce07.pdf`.

Monday	Wednesday	Friday
August 29th C1 Introduction; illustrative problem.	31st C2 Illustrative algorithm. §§ 1.*, 2.0–2.2.	**September** 2nd C3 Programming-based synthesis and review.
5th ✕ *No class.* Labor Day.	7th C4 Fundamentals of alg. analysis. §§ 2.*,3.*	9th C5 Programming-based synthesis and review.
12th C6	14th C7	16th C8
19th C9	21st C10	23rd C11 ★ **Quiz 1**, regular class time & place.
26th C12	28th C13	30th C14
October 3rd C15	5th C16	7th C17 ★ **Midterm Exam 1**, regular class time & place.
10th ✕ *No class.* Fall break Oct. 10–11.	12th C18	14th C19
17th C20	19th C21	21st C22
24th C23	26th C24	28th C25
31st C26	**November** 2nd C27	4th C28 ★ **Quiz 2**, regular class time & place.
7th C29	9th C30	11th ✕ *No class.* Veterans Day.
14th C31	16th C32	18th C33 ★ **Midterm Exam 2**, regular class time & place.
21st C34	23rd ✕ *No class.* Thanksgiving break Nov. 23–27.	25th ✕ *No class.*
28th C35 Students' choice of topic.	30th C36	**December** 2nd C37
5th C38 Synthesis and review.	7th C39 Synthesis and review.	9th C40 Synthesis and review.
12th ✕ *No class.* Finals week Dec. 12–16.	14th ✕ *No class.* ★ **Final exam: 2:45 p.m.–4:45 p.m. BD 115**	16th ✕ *No class.* **Check Univ. schedule for final exams.**

Figure 1: **Approximate** schedule, likely to change. Notation: §§ $x.y$ ⇒ textbook chapter x, section y.

  - Class Exercise 8: `hwq/ce08.pdf`.
  - Class Exercise 9: `hwq/ce09.pdf`.
  - Class Exercise 10: `hwq/ce10.pdf`.
  - Class Exercise 11: `hwq/ce11.pdf`.
- Homework assignments:
  - Homework 1: `hwq/hw01.pdf`.
- Quizzes and Exams:
  - Practice Quiz 1: `hwq/pq01.pdf`.
  - Quiz 1: `hwq/q01.pdf`.
  - Midterm Exam 1: `hwq/mt01.pdf`.
  - Quiz 2: `hwq/q02.pdf`.
  - Midterm Exam 2: `hwq/mt02.pdf`.

Homework and Project Submissions

Answers to programming problems should be submitted electronically, using the packaging and submission procedure that will be described in class or on the class discussion forum.

All electronic submissions must be made using the specified procedure. Electronic submissions in **all other forms**, such as email or physical media, will be **discarded and receive no credit.**

Answers to non-programming problems should be submitted **both** electronically as above in PDF format and in physical paper format in class. The PDF file may be either a scanned copy of handwritten work or generated directly from a typed electronic version (your choice). However, it is critically important that your submission have good legibility regardless of how it is produced. Please be very mindful of file sizes especially if you use scanned PDFs. Unless you are adept at a program that typesets math *well*, well written and well scanned hand-written versions are likely to be better than clumsily typed versions. *Illegible, hard to read, or otherwise messy submissions, whether handwritten on typed, are likely to be returned without grading, for zero credit.*

Fallback procedure If (and only if) there are unexpected problems and you are unable to submit your work as above, then you should save your file on your own computer (with some backups), compute its MD5 checksum using the md5sum utility on Unix-like systems (or other similar tools), and submit the file name, time stamp, and MD5 checksum (only, not the file itself) by email with a suitable Subject header.