COS 350: Data Structures and Algorithms

Sudarshan S. Chawathe

University of Maine

Spring 2018

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.

News and Reminders:

- **Please read the newsgroup for timely announcements:**
  Local group `umaine.cos350` on NNTP server `creak.um.maine.edu`.
  Web interface: `http://chaw.eip10.org/news/`.
- Please use the PDF version of this document for printing and reference: `cos350.pdf`
- The most recent version of this document may be found at `http://chaw.eip10.org/cos350/`.
- 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.

Goals and Outcomes

Goals

- Learn several well-known algorithms.
- Learn the principal 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 analysis 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.

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:** Tuesdays & Thursdays, 9:30–10:45 a.m.
> **Location:** Neville Hall, Room 208.

Instructor: Sudarshan S. Chawathe

> **Office:** Neville Hall, Room 224.
> **Office hours:** (Please check for changes.)
> - Tuesdays and Thursdays:
>   - 10:45–11:30 a.m.
>   - 3:15–4:00 p.m.
> **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 newsgroup. (See below.) Please use only this email address and put the string *COS350* near the beginning of the Subject header of the message. *All other messages may be ignored.*
> **Web:** http://chaw.eip10.org/

Teaching Assistant: Mark Royer

> **Office:** East Annex, Room 225.

Office hours: (Please check for changes.)
Mondays: 1:30–3:00 p.m.
Thursdays: 4:30–6:00 p.m.
Phone: +1-207-581-3946.
Email: `mark.royer@maine.edu`

Online Resources

Class Web site:
`http://chaw.eip10.org/cos350/`
We will use the class Web site for posting assignments, readings, notes, and other material. Please monitor it.

Class Newsgroup: We will use the local USENET newsgroup `umaine.cos350` on the NNTP (net news) server `creak.um.maine.edu` for electronic discussions. The Web interface at `http://chaw.eip10.org/news/` provides convenient access. Some further, more general, information on USENET appears at `http://en.wikipedia.org/wiki/Usenet`. The newsgroup is the primary forum for electronic announcements and discussions, so please monitor it regularly, and post messages there as well. Unless there is a reason for not sharing a question or comment, please *use the newsgroup, not email*, for questions and comments related to this course.

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 newsgroup. I anticipate fewer than a dozen messages on this list over the semester.

Grading Scheme

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**.

component	% of grade
class participation	5
newsgroup activities	15
quick checks	5
classroom exercises	5
portfolio presentations	5
homeworks	25
two quizzes (short exams)	8
two midterm exams	16
final exam	16

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

Newsgroup activities Discussions on the class newsgroup are an important mode of learning. To encourage everyone to participate actively in these discussions, there is a portion of the grade assigned to how well, and how often, students participate on the newsgroup. Details of how newsgroup activity is scored will be described in class. Briefly, students' scores will reflect the number and regularity of their messages, the significance of the observations they make (including questions), and the initiative and self-directed learning they reflect.

Quick checks: In order to encourage everyone to keep up with reading the textbook and other assigned work, there will be a number of very short (under five minutes) tests. Quick checks are **closed book (and notes, etc.)**, unlike exams and quizzes (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.

Portfolio presentations: The portfolio portion of the grade is intended to encourage reflection and discussions on the learning process of individuals and groups. The portfolio consists of a learning journal (see below), class work and home work samples, and other artifacts that demonstrate student progress. There will be two in-class presentations of student portfolios, in a poster-session format.

Students should maintain a journal of their progress through the course, noting things such as material learned, difficulties encountered, successes and problems in resolving these difficulties, and plans for improvement.

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 (but not quick checks, noted above) 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.

Midterm exams will be held during regular class meetings, and will be roughly an hour long. Each quiz is a short exam, roughly half an hour long, held during part of a class meeting. The final exam follows the usual university schedule, and is thus held outside of regular class meeting times, and often at a different location.

Policies

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 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 both the instructor's and TA's 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/.]

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.

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 Java 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 newsgroup.

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 Java and wish to use this opportunity to master another language, not for students weak in Java 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. The use of any specific literate-programming or documentation tool is neither necessary nor sufficient for this requirement.

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, 3rd edition, 2009. 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 newsgroup. 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).

TUESDAY	THURSDAY
January 23rd C1 Introduction; rod-cutting. §§ 1.*; 15.1.	25th C2 Fundamentals of alg. analysis; dynamic programming. §§ 2.*, 3.*; .15.2, 15.3.
30th C3 Dynamic programming. §§ 15.4, 15.5.	**February** 1st C4 Divide and conquer. §§ 4.1, 4.2.
6th C5 ★ **Quiz 1**, regular class time & place.	8th C6 Solving recurrences. §§ 4.{3,4,5}.
13th C7 Probabilistic analysis. §§ 5.1, 5.2.	15th C8 Randomized algorithms. §§ 5.3.
20th C9 Augmenting data structures. §§ 14.*.	22nd C10 ★ **Midterm Exam 1**, regular class time & place.
27th C11 Elementary graph algorithms. §§ 22.*	**March** 1st C12 Minimum spanning trees. §§ 23.*
6th C13 **Portfolio presentations v1.**	8th C14 (snow day) Shortest paths. §§ 24.{1,2}.
13th ✕ *No class.* Spring break Mar. 12th–18th.	15th ✕ *No class.* Spring break Mar. 12th–18th.
20th C15 Elementary graph algorithms. §§ 22.*	22nd C16 Minimum spanning trees. §§ 23.*
27th C17 ★ **Quiz 2**, regular class time & place.	29th C18 Shortest paths. §§ 24.{1,2}.
April 3rd C19 Shortest paths. §§ 24.{3,4,5}.	5th C20 All-pairs shortest paths. §§ 25.*.
10th C21 NP completeness. §§ 34.{1,2,3}.	12th C22 NP completeness. §§ 34.{4,5}.
17th C23 Catch-up and review.	19th C24 ★ **Midterm Exam 2**, regular class time & place.
24th C25 **Portfolio presentations v2.**	26th C26 Greedy algorithms. §§ 16.{1,2,3}.
May 1st C27 Synthesis and review.	3rd C28 Synthesis and review.
8th ✕ *No class.* Finals week May 7th–11th. ★ **Final exam: May 8th 8:00–10:00 a.m.**	10th ✕ *No class.* Finals week May 7th–11th. ★ **Check Univ. schedule for final exams.**

Figure 1: **Approximate** *schedule, likely to change. Textbook items are in* §§ *chapter.section* *format.*

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

1. Pam A. Mueller and Daniel M. Oppenheimer. The pen is mightier than the keyboard: Advantages of longhand over laptop note taking. *Psychological Science*, 25(6):1159–1168, 2014. PMID: 24760141.

2. Moritz Baum, Julian Dibbelt, Dorothea Wagner, and Tobias Zündorf. Modeling and Engineering Constrained Shortest Path Algorithms for Battery Electric Vehicles. In Kirk Pruhs and Christian Sohler, editors, *25th Annual European Symposium on Algorithms (ESA 2017)*, volume 87 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

3. Amariah Becker, Philip N. Klein, and David Saulpic. A Quasi-Polynomial-Time Approximation Scheme for Vehicle Routing on Planar and Bounded-Genus Graphs. In Kirk Pruhs and Christian Sohler, editors, *25th Annual European Symposium on Algorithms (ESA 2017)*, volume 87 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

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 session (recursively): `http://chaw.eip10.org/201701/cos350/`.

- 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`.
  - Class Exercise 8: `hwq/ce08.pdf`.
  - Class Exercise 9: `hwq/ce09.pdf`.
  - Class Exercise 10: `hwq/ce10.pdf`.
  - Class Exercise 11: `hwq/ce11.pdf`.
  - Class Exercise 12: `hwq/ce12.pdf`.
  - Class Exercise 13: `hwq/ce13.pdf`.
  - Class Exercise 14: `hwq/ce14.pdf`.
  - Class Exercise 15: `hwq/ce15.pdf`.
  - Class Exercise 16: `hwq/ce16.pdf`.
  - Class Exercise 17: `hwq/ce17.pdf`.
  - Class Exercise 18: `hwq/ce18.pdf`.
  - Class Exercise 19: `hwq/ce19.pdf`.
  - Class Exercise 20: `hwq/ce20.pdf`.
  - Class Exercise 21: `hwq/ce21.pdf`.
  - Class Exercise 22: `hwq/ce22.pdf`.

- Class Exercise 23: `hwq/ce23.pdf`.

- Homework assignments:

  - Homework 1: `hwq/hw01.pdf`.
  - Homework 2: `hwq/hw02.pdf`.
  - Homework 3: `hwq/hw03.pdf`.
  - Homework 4: `hwq/hw04.pdf`.
  - Homework 5: `hwq/hw05.pdf`.

- Quick Checks:

  - Quick Check 1: `hwq/qc01.pdf`.
  - Quick Check 2: `hwq/qc02.pdf`.
  - Quick Check 3: `hwq/qc03.pdf`.
  - Quick Check 4: `hwq/qc04.pdf`.
  - Quick Check 5: `hwq/qc05.pdf`.
  - Quick Check 6: `hwq/qc06.pdf`.
  - Quick Check 7: `hwq/qc07.pdf`.
  - Quick Check 8: `hwq/qc08.pdf`.
  - Quick Check 9: `hwq/qc09.pdf`.
  - Quick Check 10: `hwq/qc10.pdf`.

- Quizzes and Exams:

  - Quiz 1: `hwq/q01.pdf`.
  - Midterm Exam 1: `hwq/mt01.pdf`.
  - Quiz 2: `hwq/q02.pdf`.
  - Midterm Exam 2: `hwq/mt02.pdf`.
  - Final Exam: `hwq/fin.pdf`.

Homework Submissions

Handwritten answers to non-programming problems should be submitted in class on the due date, at the beginning of class (within the first five minutes), unless prior alternate arrangements are made. If you prefer to type your answers, please make sure the result uses the proper symbolic notation for mathematical constructs. *Illegible, hard to read, or otherwise messy submissions, whether handwritten on typed, are likely to be returned without grading, for zero credit.* Answers to programming problems should be submitted electronically, using the packaging and submission procedure that will be described in class and on the class newsgroup.

All electronic submissions must be made using the upload interface at `http://chaw.eip10.org/u/`. Electronic submissions in **all other forms**, such as email or physical media, will be **discarded and receive no credit.**

If your upload is successful, you will be presented with a confirmation Web page similar to the following sample. You should record the reported MD5 checksum and timestamp (important in case there are undetected problems).

```
SUCCESS: Please note the following for your records.

Successfully saved cos350-hw01-aardvark-alice-4233.tgz.
MD5 checksum: 09ee098b83d94c7c046d6b55ebe84ae1
Timestamp: 2018-01-22 17:42:42
```

If you do not see something very similar then your submission is unsuccessful.

Contingency 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.

Do not submit your work by email; it will be discarded; really.