Algorithm Engineering

CPSC 335-01,03

Spring 2018

Description & Objectives

Algorithm design using classical patterns: exhaustive search, divide and conquer, randomization, hashing, reduction, dynamic programming, and the greedy method. Asymptotic and experimental efficiency analysis. NP-completeness and decidability. Implementing algorithms to solve practical problems.

Prerequisites

CPSC 301/EPP and MATH 270B and declared major/minor in computer science/engineering

Instructor

Professor Kevin Wortman

Phone:	657-278-2968
Email:	kwortman@fullerton.edu
Office:	CS-536
Office Hours:	Monday 2:30-4:30 pm & Thursday 2:30-3:30 pm
(Spring 2018)	& by appointment. During final exam week, office hours are by appointment only.

Meeting Information

Section	**Meets**
335-01	MoWe, 1 - 2:15 pm, E-202
335-03	TuTh, 4 - 5:15 pm, CS-110B

Important Dates

CSUF's Academic Calendar is posted online at «http://apps.fullerton.edu/AcademicCalendar/». The Academic Calendar contains all the campus closures and holidays you should be aware of.
CSUF's Admissions Calendar is posted online at «http://www.fullerton.edu/admissions/Resources/Calendars.asp». The Admissions Calendar contains all the major dates with respect to adding, dropping, and withdrawing from your classes.

Section	**Midterm**	**Final**
335-01	Monday, March 19, in class	Monday, May 14, 2:30 - 4:20 pm
335-03	Tuesday, March 20, in class	Tuesday, May 15, 5 - 6:50 pm

Textbooks

Required

Algorithm Design in Three Acts, Kevin Wortman, Beta Edition, available as a free PDF in Titanium.

This semester is a beta test of an open-culture textbook authored by yours truly.

Optional

1. *Introduction to the Design and Analysis of Algorithms,* Anany Levitin, Addison Wesley, 3rd Ed., 2011.
2. *Introduction to Algorithms,* Cormen, Leiserson, Rivest and Stein, 3rd Ed., 2009.

These books provide an alternative perspective on the material of the course. They are not required, but are suggested to students who feel they would benefit from varied explanations of the material. The Cormen et al. book is an excellent reference and is a worthwhile investment for those students who expect to study algorithms further.

Development Tool Resources

Students interested in using Microsoft® development tools may request a Dreamspark account at «http://dsreqform.ecs.fullerton.edu/». A student may, at no monetary cost, download full featured versions of Microsoft Visual Studio.

Students interested in using Apple® development tools can freely download Xcode through the App Store application bundled with OS X. Students may download Xcode directly from «https://developer.apple.com/xcode/».

A Debian-based GNU/Linux OS virtual machine ready for students use and Debian-style installation scripts are posted online at «https://gamble.ecs.fullerton.edu/resources/».

A CentOS-based shell server is available through secure shell (ssh) and secure file transfer protocol (sftp). The hostname is ecs.fullerton.edu. If your email address is malcolm@csu.fullerton.edu, then your username is ACAD\malcolm. If you are using a command-line ssh client, then your command to connect to ecs.fullerton.edu will be `ssh 'ACAD\malcolm@ecs.fullerton.edu'`. Your password is the same password as your CSUF Portal password.

Please consider adopting a package management system for your personal computer to facilitate adding, updating and removing the various software development tools you may wish to use.

- Apple OS X
  - MacPorts «http://www.macports.org/»
  - Fink «http://www.finkproject.org/»
  - Homebrew «http://brew.sh/»
- Microsoft Windows
  - Chocolatey NuGet «https://chocolatey.org/»
  - Cygwin «http://www.cygwin.com/»
  - Npackd «https://npackd.appspot.com/»
- GNU/Linux OS
  - dpkg «https://www.debian.org/doc/manuals/debian-faq/ch-pkgtools.en.html»
  - rpm «http://fedoranews.org/alex/tutorial/rpm/»

Learning Goals

1. Analyze the efficiency of algorithms. We will define efficiency with mathematical precision, and learn experimental and mathematical methods for analyzing the efficiency of algorithms.
2. Use asymptotic analysis to make design decisions about tradeoffs between runtime, space consumption, and programming effort.
3. Apply common algorithmic problem-solving strategies. We will cover seven classical patterns for designing algorithms, and noteworthy algorithms based upon those strategies. You will practice designing new algorithms using each strategy.
4. Implement algorithms and observe how our model of efficiency relates to real-world performance.
5. Reduce new problems to classical problems with known solutions.
6. Understand important complexity classes of problems: P, NP, NP-complete, and undecidable.

G.E. Requirements

This class does not meet any CSU General Education requirements.

Course Outline

Week	Topics	Reading Assignment
1	**Introduction**; review syllabus; administer wait list Terminology, pseudocode, and algorithm patterns	Syllabus Chapters 1-2
2	**Efficiency analysis**; functions measuring resources, asymptotic notation, experimental analysis Mathematical analysis, the standard model, step counting	Sections 3.1-3.5 Sections 3.6.1-3.6.3
3	Proving efficiency classes Amortized analysis; **review essential data structures**	Sections 3.6.4 - 3.6.6 Section 3.7, Chapter 4
4	The **greedy pattern**; analyzing nested loops; sequential search; change making; greedy sorting and selection sort In-place selection sort	Sections 5.1 - 5.5
5	President's Day holiday, Professor for a Day The Prim-Jarník algorithm for minimum spanning trees	Section 5.6
6	Dijkstra's algorithm for nonnegative single-source shortest paths **Exhaustive search and optimization**; generating and verifying candidates; minimum spanning trees by exhaustive search	Section 5.7 Sections 6.1 - 6.6
7	Circuit satisfaction, traveling salesperson, knapsack problem **Decrease by half**; recursive algorithms, master theorem, merge sort	Sections 6.7 - 6.9 Sections 7.1 - 7.4
8	Binary search; indivisible problems **Randomization**; pure quicksort	Sections 7.5 - 7.6 Sections 8.1 - 8.5.2
9	**Midterm exam** Analysis of quicksort; in-place quicksort	Covers chapters 1-7 Sections 8.5.3 - 8.5.5
	Spring Recess - no classes	
10	**Reduction**; reduction to sorting hash tables	Sections 9.1 - 9.2 Section 9.3
11	Priority search queues; optimizing Prim-Jarník and Dijkstra **Dynamic programming**; Fibonacci numbers; change making; backtracking	Section 9.4 Sections 10.1 - 10.4
12	**Limitations of algorithms**; **lower bounds**; sorting lower bound; reduction arguments	Chapter 11

13	**Intractable problems**; complexity classes; class *P* Unsolvable problems and **decidability**; halting problem	Sections 12.1 - 12.2 Section 12.3
14	Verifiable problems; *NP*, *NP*-hard, and *NP*-complete *NP*-completeness reduction proofs	Sections 12.4 - 12.5 Section 12.7
15	More *NP*-completeness; the P vs. NP question *Overflow/review time*	Section 12.8
finals	**Final exam**	Covers chapters 7 - 12

Technical Proficiency

Technical proficiency in programming and software engineering should correspond to the prerequisite(s) of the course. Students are expected to be intimately familiar with their development platform of choice and be able to write and debug code in C++ at a level of proficiency that corresponds to the prerequisites of the course.

Technical proficiency with information technology, such as, but not limited to, the use of web-based online services, sending and receiving electronic mail, and desktop computer file systems, is assumed.

Grading

Plus and minus grading is not used when determining final grades.

Final grades are computed by first finding the average score in each category described in the table below on the right. All scores are normalized to a scale of 0 to 100 before being averaged. The average score for each category is then used to compute the weighted average according to the weights in the second table below.

Grade	**% of Total Points**
A	90–100%
B	80–89%
C	70–79%
D	60–69%
F	Below 59%

Category	% of Final Grade
Reading Assignments	0%
Homework	15%
Projects	25%
Midterm	30%
Final	30%

Assignments

Programming and written assignments will be discussed in class and posted to the course website in advance of their due dates. Each assignment description will include the assignment's grading rubric. Reading assignments are outlined in the syllabus and it is the responsibility of the student to stay up to date with the reading.

Written assignments must be typeset and presented in a professional manner. Presentation, spelling and grammar can be worth up to 30% of a written assignment's grade.

All programming assignments must be written in the C++. Coding style must conform to professional norms. At a minimum, code must be commented, have descriptive names for identifiers, and contain a comment at the top of each file with pertinent information such as the student's name, email address, and assignment name. For students unfamiliar with coding style, Google's style guides are an excellent starting point, «https://github.com/google/styleguide», particularly their C++ style guide, «https://google.github.io/styleguide/cppguide.html».

At the start of the semester, the instructor will detail the platform and tools used to grade student assignments. It is the student's responsibility to ensure that the assignments execute to his or her satisfaction on the instructor's grading platform.

Reading Assignments

For each lecture (except exam days) there is a specific reading assignment. The reading assignments are listed in the course outline above. You are expected to read these assignments *before* class. Lecture time will focus on discussing critical material and answering questions; there is not enough time to present every single concept from the text.

Homework

Homework will involve approximately three word problems involving algorithm design and analysis. Homework will be due every week, except for the first week, midterm week, and last week, so there will be approximately 13 homeworks. You must turn in homework submissions as hardcopy in class. You may work in groups of up to three on homework; make one submission with the names of all group members. Late submissions will not be accepted. Your lowest homework score will be dropped.

On each submission, write a *star* next to one problem. You will receive feedback on that answer and it will be the basis of your grade. Feedback will be given for the un-starred problems on a time-permitting basis. Homework will be graded according to a coarse rubric with significant partial credit for effort.

Homework problems are intended to provide review and reinforcement of the material in class, and practice for the kind of problems that are asked on exams. You are expected to complete all assigned problems. If you want further practice, complete the textbook problems that were not assigned, or problems from the optional textbooks.

Projects

There will be four programming projects. The first project will be posted after we have covered the foundational material of weeks 1-3, and the remaining projects will be roughly evenly spaced. The tentative topics for the projects are:

1. Empirical analysis, comparing efficiency classes
2. Exhaustive search, exponential versus polynomial time
3. Dynamic programming
4. Approximation

Each project will involve designing, implementing, and analyzing a substantial C++ program, and writing a brief report document. Projects will be graded according to a written rubric involving both the form and function of the C++ code, and also the algorithm analysis and technical writing in the report.

We will use GitHub and Travis CI to distribute, submit, and grade projects. This approach has the benefit of offering you grading feedback before the deadline, expediting grading, and teaching you marketable skills. Please refer to our GitHub Education Instructions.

You may work in groups of up to three on projects; make one submission with the names of all group members.

Late submissions will not be accepted.

Exams

Exams are written, paper-packet, closed-book, individually-completed, non-cumulative, in-class examinations. You do not need any special materials beyond a writing implement (no Scantron form etc.). A review sheet listing fair-game material will be posted prior to each exam.

You may bring a memory aid (“crib sheet”) to each exam. Your sheet must be a single piece of paper, letter/A4 sized, optionally double-sided, with your name and CWID on it, and you must turn it in with your exam packet.

Attendance Policy

Attending class is mandatory. Missing class is not allowed unless it is excused by the instructor. Missing class as part of a documented accommodation is guaranteed to be excused. The ADA accommodated student must make a reasonable effort to coordinate any absences with the instructor.

Make Up Policy

Exams and quizzes cannot be taken after they have been given in class. Due to an act of nature, personal medical emergency, a family crisis, an act of terrorism, severe civil unrest, etc. students have 10 calendar days to petition the instructor to retake any exam/quiz or submit an assignment without late penalty.

Exceptions shall be made on a case by case basis, provided there is time to evaluate the merits of such an application.

Participation

In the context of this course, participation is defined as the following:

- Arriving to class prepared and on time.
- Taking notes.
- Actively listening to the lecture and asking questions when appropriate.
- Annotating code listings and handouts.
- Bringing any required materials to class.
- When needed/desired, seeking assistance to complete assignments.
- Barring an emergency, not leaving the class session early unless the instructor consents.
- Not distracting oneself or others with smartphones, games, online diversions, etc.
- Respecting and treating the instructor and the student's peers civilly.

Required Material

- A writing instrument
- A notebook
- A personal computer with the requisite development tools or regular access to a computer lab

Academic Dishonesty

Students are encouraged to assist one another and discuss the course materials with your peers. It is your responsibility to be aware of and follow the spirit of CSU Fullerton's academic honesty policy which can be found at «http://www.fullerton.edu/senate/publications_policies_resolutions/ups/UPS%20300/UPS%20300.021.pdf

». Academic dishonesty will not be tolerated. The University Catalog and the Class Schedule provide a detailed description of Academic Dishonesty under *University Regulations*.

By submitting work for evaluation, you acknowledge that you have adhered to the spirit of the university's academic honesty policy and that your submission is an original work by you unless otherwise directed to work in groups. Failure to follow the spirit of the academic honesty policy will result in a severely negative evaluation of the work in question and may result in involving the Department Chair and the Judicial Affairs office to seek a disciplinary remedy.

ADA Accommodations

Any student who, because of a disability, may require special arrangements in order to meet course requirements must register with the Office of Disability Support Services within the first week of classes. The Office of Disability Support Services' website is «http://www.fullerton.edu/DSS/». They can be reached by phone at 657-278-3117 or TDD at 657-278-2786. Their email address is «dsservices@fullerton.edu». Their office is located in University Hall, room 101. The instructor may request verification of need from the Dean of Students Office. Students requesting accommodations shall inform their instructors during the first week of classes about any disability or special needs that may require specific arrangements/accommodations related to attending class sessions, completing course assignments, writing papers or quizzes, tests or examinations.

Student Resources

Any student who wishes to discuss any concern may contact the assistant deans of the college. Assistant deans are student advocates who will help you navigate the university's policies and procedures and assist with resolving any conflicts.
Assistant Dean for Student Affairs Carlos Santana
CS-206A (657) 278-4407 «csantana@fullerton.edu»
Assistant Dean International Programs and Global Engagement Lillybeth Sasis
CS-206A (657) 278-4881 «lsasis@fullerton.edu»

Emergency Procedures

For your own safety and the safety of others, each student is expected to read and understand the guidelines published at «http://prepare.fullerton.edu/campuspreparedness/». Should an emergency occur, follow the instructions given to you by faculty, staff, and public safety officials. An emergency information recording is available by calling the Campus Operation and Emergency Closure line at 657-278-4444.

Instructional Continuity

Due to an event such as an epidemic or a natural disaster that disrupts normal campus operations, students must monitor the course Titanium site and their campus email address for any instructions and assignments that the instructor announces.

Laboratory Safety

Safety is no accident. Learning and following the appropriate safety practices and protocols is an integral part to all laboratory courses. Following the appropriate safety practices and protocols minimizes the chances of repetitive stress injuries, mishandling hazardous materials, and injury to self and others. Additional campus laboratory safety information regarding hazardous materials is online at «http://riskmanagement.fullerton.edu/laboratorysafety/».

Extra Credit

There are no opportunities for extra credit.

Recording & Transcription of Class Content

Recording class content is governed by UPS 330.230, «http://www.fullerton.edu/senate/publications_policies_resolutions/ups/UPS%20300/UPS%20330.230.pdf». Each instructor must permit class content to be recorded or transcribed by students when mandated to do so by the Americans with Disabilities Act or by other federal or state laws. Any recording of class content is for private use and study and shall not be made publicly accessible without the written consent of the instructor and students in the class.

Course Rules & Classroom Management

Unless an agreement or accommodation is reached between the student and the instructor, these rules must be followed.

- Attendance at all regularly scheduled lecture and discussion section is mandatory.
- Do not eat during lecture.
- If it makes noise, silence it.
- Computer use is not allowed in lecture except for taking notes.
- The student is responsible to be aware of any course announcements including changes to due dates and requirements.
- Homework, programming assignments, etc. may not be submitted late.

- Third party work (code, artwork, etc.) may not be used in student work without prior instructor consent. Failure to gain and document instructor consent will be construed as willful academic dishonesty.
- When a third party's work is incorporated into student work after gaining instructor consent, failure to wholly document the work's origin, copyright and license will be construed as willful academic dishonesty.