CALIFORNIA STATE UNIVERSITY
FULLERTON

Syllabus for CPSC 335 Algorithm Engineering (Spring 2023)

Department of Computer Science, College of Engineering and Computer Science

Course	Instructor	Place	Meeting Time	Final Exam
CPSC 335.01	Dr. Sampson Akwafuo	CS 102B	MonWed, 1:00 - 2:15 pm	Monday, 5/15/2023 1:00pm – 2:50pm

Phone: (657) 278 4676 E-mail: sakwafuo@fullerton.edu. Office: CS-550

Office hours: Tuesday 5:30 pm - 6:30pm, Wednesday 10:30 am - 12:30pm or by appointment. During the final week, office hours will be held by appointment only. **Office Hours** will be held both virtually through zoom and in-person. https://fullerton.zoom.us/j/85767583386 is the office hour link.

Attendance and Participation

Attending the lectures is not mandatory during scheduled physical or virtual meetings. However, attendance may be randomly taken during some classes. Students are responsible for all course materials regardless of whether they are present or absent.

IMPORTANT DATES: CSUF's Academic Calendar is posted online here and contains all the campus closures and holidays you should be aware of. CSUF's Admissions Calendar is posted online here and contains all the major dates with respect to adding, dropping, and withdrawing from your classes.

Monday February 20: Presidents' Day, no classes or office hours

Monday March 27 - Sunday April 2: Spring Recess, no classes or office hours

The last day to drop the course without a grade of 'W' is Monday, February 6. See detailed calendar.

CATALOG DESCRIPTION: 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. Implementing algorithms to solve practical problems. A shortcut to the course description is http://csufcs.com/cs335.

Schedule

Week	List of Topics	Reading Assignment

1	Introduction to Course: Syllabus, Algorithmic Terminologies, Algorithmic Problem Solving	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 of essential data structures**	Sections 3.6. - 3.6 Chapter 4
4	The **naïve pattern**, sequential search, the sorting problem, one-at-a-time sorting. Selection sort; in-place selection sort	Section 5.3 Section 5.5
5	The **greedy pattern**; greedy sorting; change making; The **Prim-Jarník algorithm,** Kruskal algorithm for minimum spanning trees Dijkstra's algorithm for nonnegative single-source shortest paths	Sections 6.1 - 6.2 Section 6.3 Section 6.4
6	**Exhaustive search and optimization**; generating and verifying candidates; minimum spanning trees by exhaustive search	Sections 7.1 - 7.6
7	Circuit satisfaction, traveling salesperson, knapsack problem **Decrease by half**; recursive algorithms, master theorem, merge sort	Sections 7.7 - 7.9 Sections 8.1 - 8.4
8	Binary search; indivisible problems **Randomization**; Monte Carlo pattern, Las Vegas pattern; pure quicksort	Sections 8.5 - 8.6 Sections 9.1 - 9.5.2
9	**Midterm exam** Analysis of quicksort; in-place quicksort	Covers chapters 1 – 7 Sections 9.5.3 - 9.5.5
10	**Reduction**; reduction to sorting Hash tables	Sections 10.1 - 10.2 Section 10.3
11	Priority search queues; optimizing Prim-Jarník and Dijkstra; **HeapSort.m** **Dynamic programming**; Fibonacci numbers; change making; backtracking	Section 10.4 Sections 11.1 - 11.4

12	**Limitations of algorithms**; **lower bounds**; sorting lower bound; reduction arguments	Chapter 12
13	**Intractable problems**; complexity classes; class *P* Unsolvable problems and **decidability**; halting problem	Sections 13.1 - 13.2 Section 13.3
14	Verifiable problems; *NP*, *NP*-hard, and *NP*-complete *NP*-completeness reduction proofs	Sections 13.4 - 13.5 Section 13.7
15	The P vs. NP question; The NP-Completeness problem Review	Section 13.8
16	**Final exam**	Covers chapters 8 – 13

Prerequisites: MATH 150A, MATH 170A, and CPSC 131; Computer Science or Computer Engineering major or minor; or Computer Science or Computer Engineering graduate standing.

Course Communication: All course announcements and individual emails are sent to your CSUF email accounts through Canvas. Therefore, you must check your CSUF email on a regular basis for the duration of the course.

Response Time: The instructor responds to email questions usually within 48 hours except on weekends and holidays.

TECHNICAL PROFICIENCY: Students are expected to be intimately familiar with their development platform of choice and be able to write and debug code in C/C++and python at a level of proficiency that corresponds to the prerequisites of the course. Specifically, students are expected to

1. Have basic computer competency which includes:
   a. the ability to use a personal computer to locate, create, move, copy, delete, name, rename, and save files and folders on hard drives and on secondary storage devices;
   b. the ability to use a software program namely C, C++, Python, or Java for their programming assignments;
   c. the ability to use an electronic mail system to receive, create, edit, print, save, and send an e-mail message with and without an attached file; and
   d. the ability to use an Internet browser to upload their assignments to Canvas.
2. Have ongoing reliable access to a computer with internet connectivity and a web camera for regular course lectures, quizzes and exams
3. Maintain and regularly access the CSUF email account
4. Apply his/her educational technology skills to complete expected competencies
5. Utilize other software applications as course requirements dictate

6. Utilize Canvas to access course materials and complete assignments

STUDENT LEARNING GOALS

The students will expand their knowledge in algorithm's efficiency (gained from CPSC 131) and learn how to compare algorithms solving the same problem in terms of efficiency. Students will learn various methods for solving practical problems and generalize these methods to solve other problems. Through a combination of discussion, class activities, written assignments, programming projects, and exams, students will develop a solid understanding of the role of algorithmic-based problem solving.

LEARNING OBJECTIVES

Upon completion of Algorithm Engineering, students will be able to

- Analyze the efficiency of algorithms. We will define efficiency precisely, and learn empirical and mathematical methods for analyzing the efficiency of algorithms.
- Use asymptotic analysis to make engineering decisions about tradeoffs between runtime, space consumption, and programming effort.
- Apply common algorithmic problem-solving strategies. We will cover seven classical strategies for designing algorithms, and noteworthy algorithms based upon those strategies. You will practice designing new algorithms using each of the strategies.
- Implement algorithms and observe how our model of efficiency relates to real-world performance.
- Reduce new problems to classical problems with known solutions.
- Understand important complexity classes of problems: P, NP, NP-complete, and undecidable.
- Handle hard problems appropriately; make an informed decision about whether to settle for an approximation, limit input size, or give up.

Required Textbook

Algorithm Design in Three Acts, Kevin Wortman, Beta Edition, free and available here and on Canvas.

Recommended materials

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

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.

DESCRIPTION OF ASSESSED WORK

Exams: There will be a midterm exam, scheduled in **week 8** or **9**, and a final exam, scheduled according to the University Final Exam schedule. A review material will be posted prior to each exam. Exams will be done through canvas. The exams are individually completed and non-cumulative. There will be **no makeups** for missed exams except through advance request for documented exceptional circumstances.

Projects: There will be three or four programming projects, due before 11:59 pm on published deadline. See Late Policy for late submissions. Every project score is weighted equally, regardless of the number of points available on each project. Each project will involve designing, implementing, and analyzing a substantial algorithm, using Python or C++, and writing a brief report document. Projects will be graded according to the accompanying rubric. You may work **individually** or in groups of **up to three** on projects; make one submission with the names of all group members. The following attract penalties:

- Source code that cannot be compiled successfully
- Input/output that is falsified.
- Submissions that are plagiarized or otherwise violate the collaboration guidelines

Students may use Tuffix to do their projects. Tuffix is the Titan's Linux distribution. You can set up Tuffix as a native install on a dedicated computer, or as a virtual machine (VM), by following the Tuffix Installation Instructions. You can use your own computer, or borrow a computer from CSUF for free through the Long-Term Laptop Checkout process. Another option is CSUF's Virtual Computing Lab (VCL). This allows you to connect remotely to a Tuffix-like environment. Keep in mind a VCL session only lasts a limited time (up to 4 hours) and *is erased at the end of the session,* so you *must save your work elsewhere before your session ends* (e.g. git push all your work before the session ends). Students using Tuffix should join the CSUF TUFFIX slack workspace at https://csuf-tuffix.slack.com. Please use the #general channel to ask about troubleshooting, installing, and using Tuffix. The smoothest and best-performing option is to install Tuffix natively on a computer, so that is our first recommendation.

Be advised that automated tools may be used to detect plagiarized work.

Quizzes: There will be 4 or 5 quizzes, to be taken on Canvas. Every quiz score will be weighted equally, regardless of the number of points available on each quiz. Every quiz is timed and must be taken with the specified time frame.

Homework: Homework will involve questions on algorithm design and analysis. About 2 or 3 HomeWorks will be given during non-quiz weeks. You must upload your homework to Canvas on or before the deadline. You may work individually or in groups of up to three on homework; make one submission with the names of all group members.

In-class Presentations: There will be one group presentation on pre-assigned algorithmic topics. Each group should upload their presentation on or before the due date. Further details will be provided later.

Reading lists: 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.

Late Submission Policy: Late submission of an assignment attracts 50% penalty for up to 48 hours. No submission will be accepted after 48 hours. Under exceptional circumstances, the instructor may allow additional extensions on a case-by-case basis.

GRADING STANDARDS AND CRITERIA

Grades will be based on following:

- Quizzes 15%
- Projects 20%
- Homework and Presentations 15%
- Midterm exam 20%
- Final exam 30%

Each student's weighted numerical average is translated into letter grade, as shown below.

Grading Scale

Grade	Percentage		Grade	Percentage		Grade	Percentage
A	92-100%		B-	72-77%		D	50-57%
A-	88-91%		C+	68-71%		F	below 50%
B+	82-87%		C	62-67%			
B	78-81%		C-	58-61%			

Academic Dishonesty Policy

By submitting work for evaluation, the student acknowledges that he/she has adhered to the spirit of the university's academic honesty policy and that his/her submission is an original work done by the student unless otherwise directed to work in groups. It is the student's responsibility to be aware of and follow the spirit of CSU Fullerton's academic honesty policy found at Academic Dishonesty Policy. Academic dishonesty includes such things as plagiarism, cheating, inventing false information or citations, and helping someone else commit an act of academic dishonesty. It usually involves an attempt by a student to show possession of a level of knowledge or skill, which he/she in fact does not possess. Plagiarism is defined as the act of taking the specific substance of another and offering it as one's own without giving credit to the source (e.g., copying other person's program). When sources are used, acknowledgment of the original author or source must be made following standard scholarly practice. Cheating is defined as the act of obtaining or attempting to obtain credit for work by the use of any deceptive, dishonest, fraudulent or unauthorized means. Examples of cheating include, but are not limited to using notes or aids, help other students on tests and examinations in ways other than those expressly permitted by the instructor, plagiarism as defined above, tampering with grading procedure, collaborating with others on any assignment where such collaboration is expressly forbidden by the instructor. **You are not allowed to use any material from any website that provides solutions to the assignments given in class for a fee or free of charge. Instructors may be using software to detect similarity and plagiarism.** Failure to follow the spirit of the academic honesty policy will result in a severely negative evaluation of the work in question. Each offense will be reported to the Department Chair and the Dean of Students

office, Student Conduct. A first offence will result in a zero score on the offending assignment. A subsequent offence will result in an F in the course.

Collaboration: Collaboration is not allowed on any exam or quiz. For homework or projects, you may work freely with your fellow students, but must limit the input you get from sources outside your group:

- You may help each other understand the assignment and brainstorm general solutions, but each group must develop and submit their own distinct work.
- You may give each other technical support, for instance troubleshooting, installing the compiler or logging in to Canvas.
- You must separate to develop your own detailed solution to the problem, and type in your own source code and project report.
- Given these requirements, any submissions with identical excerpts, or excerpts that are identical up to superficial rearrangements, will be considered highly suspect of plagiarism.

POLICY ON RETENTION OF STUDENT WORK

Work is submitted through Canvas and shall be retained on the course website for a reasonable time after the semester is completed (see UPS 320.005).

Administrative drops: Any student who misses the first class meeting may be dropped from the class, unless they contact the instructor within 24 hours.

Religious Holidays: If you will miss a class or an exam due to observance of a religious holiday, please notify the instructor by email (sakwafuo@fullerton.edu) in the first week of class.

UNIVERSITY INFORMATION

As a registered student you are enrolled in Canvas. We will be using only Canvas. Problems? Contact the student help desk at (657) 278-8888 or email StudentITHelpDesk@fullerton.edu.

ADA Accommodations for Students with Special Needs

Please inform the instructor during the first week of classes about any disability or special needs that you may have that may require specific arrangements related to attending class sessions, carrying out class assignments, or writing papers or quizzes, tests or examinations. Any student who, because of a disability, may require special arrangements in order to meet course requirements must contact the instructor and the Office of Disability Support Services as soon as possible to make the necessary arrangements. The instructor may request verification of need from the Dean of Students Office. Students are encouraged to contact the Office of Disability Support Services within the first week of the semester to best ensure that the appropriate accommodations are implemented in a timely fashion. 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 or email 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. Ms. Lindsay O'Neill in the Pollak Library <jloneill@Exchange.FULLERTON.EDU> will be able to answer technical questions about accessibility of specific library-provided resources.

ECS Student Resources

Any student who wishes to discuss any concern may contact the student support services and resource center of the college. There are student advocates who will help you navigate the university's policies and procedures and assist with resolving any conflict. http://www.fullerton.edu/ecs/resources/

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

Software for Students

Did you know you can get FREE and low-cost software for being an active CSUF student? Software can be requested from the CSUF Student Technology Services website.

Library Support: The Pollak Library has many services to offer students. Assistance available for online students includes online instruction guidelines available on the library website.

EMERGENCY PROCEDURES

Each student is expected to read and understand the guidelines published here. Should an emergency occur, follow the instructions given by faculty, staff, and public safety officials, or contact the University Police at (657) 278-3333. An emergency information recording is available by calling the Campus Operation and Emergency Closure line at 657-278-4444. In an effort to keep our campus community informed and to comply with the California State Education Code, Chapter 16, of the Donahue Higher Education Act, Section 67380; the California State University, Fullerton Police Department prepares the California Campus Safety Plan annually. The plan can be found on the University Police website under the Jeanne Clery-Crime Prevention tab or by clicking on this link.

RECORDING & TRANSCRIPTION OF CLASS CONTENT

Recording class content is governed by UPS 330.230. The instructor permits 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

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

- Attendance, although not mandatory, is essential.
- If it makes noise, silence it.
- The student is responsible to be aware of any course announcements including changes to due dates and requirements.

- Third party work (code, artwork, etc.) may not be used in student work without prior instructor's 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 fully document the work's origin, copyright and license will be construed as willful academic dishonesty.