University of California, Irvine: Computer Science

CS 161: Algorithm Design and Analysis (SU24)

Teaching Staff	Instructor: Shion Fukuzawa (fukuzaws@uci.edu) Office hours: TBA TA: Ryuto Kitagawa (ryutok@uci.edu) Office hours: TBA TA: Kourosh Mirsohi (mirsohi@gmail.com) Office hours: TBA
Class Information	SSL 140, MWF 11:00-12:00
Website	https://www.shionfukuzawa.com/courses/161su24.html

About Me

Hi all! My name is Shion (pronounced she-own) and I use he/him pronouns. I am a fourth year PhD student doing research in quantum algorithms and information theory. I'm also very passionate about CS education, and I hope that I get to continue being involved in education throughout my career. Outside of class I love baking, taking care of my plants, and playing valorant. I am excited to meet you all and learn together, and if there is anything I can be doing better throughout the quarter to help you please let me know! I'll be providing opportunities to share feedback throughout the course (more on that later).

About This Course

It's no secret that computers now play a crucial part in all of our lives. By now, you have some experience writing algorithms, and have started developing a procedural way of thinking to solve problems. In any field, it is important to have a unified language to communicate the quality of the results that are developed. In this course you will learn about the language computer scientists use to communicate the quality of their algorithms. To discuss this, we will be honing our mathematical tools, and learn when this language works well, and what challenges we face trying to use this language.

We will also learn about some standard algorithm design paradigms, which include divide and conquer, greedy algorithms, and dynamic programming. We will think about various scenarios where these paradigms can help design efficient algorithms.

Prerequisites

The official prerequisites listed in the catalogue are: (I&C SCI 46 or CSE 46) and I&C SCI 6B and I&C SCI 6D and MATH 2B. I&C SCI 46 with a grade of C or better. CSE 46 with a grade of C or better.

With that said, what I generally expect you to know are the following topics:

- Fundamental data structures (arrays, linked lists, trees, graphs)
- Proof techniques
- Basic familiarity with sorting algorithms

Student Learning Outcomes

After successful completion of this course, you will be able to:

1. **Communicate the time and speed efficiency of your algorithm using Big-Oh notation.** This is the basic language used by computer scientists when discussing algorithms.
2. **Use mathematical techniques to prove the quality of an algorithm using Big-Oh notation.** We will see that there is a lot we can say about a problem using the language we develop. This includes improvements we can add, as well as limitations that we will never break past.
3. **Design algorithms using standard algorithm design paradigms to solve various problems.** Here, standard algorithm design paradigms refer to divide and conquer, dynamic programming, and greedy algorithms.

Reading

No book purchase is required as I will be providing course notes throughout the quarter which should be complete for our purposes. I will be handing out notes for you to fill out throughout the class, of which I will provide digital copies for on the course website in case you lose them. As such, please bring a writing utensil or digital tablet to class. A lot of the content from my lecture notes is based on course notes from other professors and textbooks. If you like an external reference, here are a few suggestions:

- Algorithm Design and Applications, by Goodrich and Tamassia, ISBN: 978-1-118-33591-8
- Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein (often referred to in the literature as CLRS), ISBN: 978-0-262-04630-5

For any textbooks, you definitely should not search online for them on websites like Library Genesis where full PDFs are available for download. There are also github repositories which house textbook PDFs as well that you should definitely not download from.

Lecture, Discussion, and Class Participation

Students in this class are encouraged to speak up and participate during class meetings and on Ed Discussion. Because the students in the class will have a diversity of backgrounds and experiences, every member of this class must show respect for every other member of this class. Diverse teams have been shown to exhibit more creativity, social cohesion, and success which I hope we can agree are things that all of us want.

Throughout lecture, we will pause 2-3 times to solve some questions in the notes. You should attempt this question on a notebook or tablet, and I will open up a live quiz during this time where I ask you to participate in. These questions should only take a few minutes to solve, and I encourage you to attempt them individually. Once the quiz is closed, we will discuss the solution as a whole class.

Grading

In this course, your grades will be primarily dependent on how you perform for your exams. There will be three exams throughout the quarter covering each of the three modules (mathematical foundations, quantum information, quantum computing). The final exam will be split into three sections which will go over the same three major modules. Your final grade for each module will be the larger grade you get between the midterms and the final exam. See the below table for an example.

	Midterm 1	Midterm 2	Midterm 3	Final
Divide and Conquer	C-			A-
Dynamic Programming		B+		B-
Greedy Algorithms			B-	B

After this, I will average your letter grade to get a final grade for the course. The numerical values of the letter grade will use the familiar GPA point scale system.

A	4.0
A-	4.0 – 1/3
B+	3.0 + 1/3
B	3.0
B-	3.0 – 1/3
C+	2.0 + 1/3
C	2.0
C-	2.0 – 1/3
D	1.0
F	0.0

The above student would get a (4.0 – 1/3 + 3.0 + 1/3 + 3.0) / 3 = (3 + 1/3) or a B+ for their final grade. If a student gets a grade in between two letter grades, it will be rounded up. For example, having a 3.2 average will be a B+.

We will also have about 8 homework assignments throughout the quarter, which will be graded for completion of all the problems, combined with correctness for one randomly selected problem. Getting a full score on 6 out of 8 assignments will increment your final letter grade by a "sign" (i.e., C+ would go to B-).

Note that this means that you can technically get an A in this course without attempting any of the assignments. The purpose of the homework is to give you a medium where you are encouraged to make mistakes and can receive feedback without the risk of reducing your grade. Our goal is to provide helpful feedback when you make mistakes so that we can avoid them when the stakes are a bit higher. As such, tackling the exams without having done any of the homework will be extremely difficult, even though you could technically do so.

Modules

The following is the detailed breakdowns of the three modules. Note that I may update these as the quarter progresses, but I will make sure to make an announcement if I do so. For an up to date list refer to the course website linked at the top of the document.

Foundations/Divide and Conquer:

- Review of basic data structures
    - Arrays, Linked lists, trees, graphs
- Big-Oh Notation
- Sorting Algorithms including but not limited to:
    - Bubble sort
    - Insertion sort
    - Merge sort
    - Quicksort
    - Bucket sort
- Proving the run time of merge sort and quicksort
- Sorting lowerbounds

Dynamic Programming:

- Optimal substructure property
- Analyzing the size of the search space
- Top-down (memoization) vs. Bottom-up

Greedy Algorithms:

- Greedy choice property
- Optimal substructure property
- When can we use a greedy algorithm instead of a dynamic programming approach?

Communication

The course staff are all here to help you and provide what you need to succeed in the course. However, we do ask that in return you follow the communication guidelines below so that we can best support you.

Ed Discussion

We will use an online question and answer platform called Ed Discussion. You will all have an account linked to your UCInetID. **Please do not use Canvas for communication.**

There is a link to the Ed Discussion page for this class in the column on the left of the Canvas page [Add link]. If you have a question about course content, you can go to see if your question has already been asked by another students. If not, you can post the question yourself. The course staff make sure you have an answer within 24 hours. There is also a way for students to collectively edit a response to a question and for instructors to indicate whether the answer is a good/correct answer.

Ed Discussion is also a good place to post general administrative questions about the class. If you send me an email with a technical or administrative question about the class, I am likely to refer you to Ed Discussion. These type of questions get repeated a lot, so I would prefer to broadcast the response so other students can also see.

Before posting a question on Ed Discussion, you should look through the course materials or previous Ed Discussion posts to see if you can find the answer to your question. If your question is about the technical content of the course, check the text. If you have an administrative question, check the material in this Canvas course space. Try and make your questions on Ed Discussion as specific as possible. I am happy to explore more in-depth conversations through office hours or scheduled meetings.

Ed Discussion provides a way for you to post anonymously. However, your posts will only be anonymous to other students. The instructors will be able to see the name and UCINetID of any individual who posts to Ed Discussion.

Questions and Announcements

The best way to get your questions answered is by coming to lectures, office hours or discussion and asking them there. In particular, office hours and discussion are the best place to ask questions that require a longer answer or some dialog to get resolved.

The table below shows where to go to for different kinds of questions. Please try to follow the directions there. I really do want to be available for significant problems or issues that may arise. I also enjoy meeting students in my office hours. However, the amount of email generated from smaller, routine questions can be overwhelming, so I would like you to try and find answers through other sources first.

Question type	***Example***	***Where to go***
Questions about course content	*How does mergesort work?*	*Try and find the answer to your question through the course materials.* *If your question is not answered, then post on Ed Discussion.*
General administrative questions	*What topics does test 1 cover?*	*Check the Canvas course space first.* *If your question is not answered, then post on Ed Discussion.*
Questions on how your HW or test was graded.	*Why didn't I get full credit on the selected question from HW 3?*	*Submit a regrade request through Gradescope. Regrade windows remain open for 1 week after the grades are released.*
A question that requires including personal information about you.	*Can I get approval to miss Test 2? Here is the reason I need to be absent...*	*Email Shion* *Note that these should happen very infrequently.*
Course-related announcements.	*An update or correction to a HW problem.*	*Posted as an announcement on Ed Discussion. Time-critical announcements will trigger an email as well.*
Non-course-related announcements.	*Info about an upcoming UCI hackathon.*	*Posted as an announcement on Ed Discussion, tagged as non-course-related.*

Academic Dishonesty

The Bren School of ICS and the University have already established an academic honesty policy. Which you can read here.

Violators of academic honesty policies are subject to the penalties described in the Bren School of ICS policy. They are also subject to an immediate course grade of F, and you will not be allowed to drop the course to avoid the grade. Also be aware that a single documented case of academic dishonesty may preclude you from switching into computing majors, registering for computing minors, joining the ICS Honors Program, and graduating from a computing major with honors.

Guidelines to avoid plagiarism:

- **Do not look at another person's homework.** Instead you should prefer to discuss the problem in plain English. This helps you to communicate clearly, practice technical jargon as it applies to your problem, and to identify how your solution exhibits behavior different from what you expect.
- **Do not write down the solution in your notes.** It is perfectly fine (and encouraged) to collaborate on work. Working in a group is a rewarding experience, and definitely a necessary skill in any professional career. The collaboration can include drawing diagrams and perhaps

solving the problem on a whiteboard. However, you should avoid writing the solution in your notes and instead practice reconstructing it using your own words afterwards. It is very useful to rethink the problem and go through the details and logic when you solve it again on your own, and it will help avoid plagiarism.

We expect that:

- **You can monitor each other and enforce these rules among yourselves (even over Zoom).** Making sure that others follow these guidelines will help to ensure that they don't pass off your work as their own.
- **Your work honestly represents your efforts.** The entire purpose of obtaining an education is so that you can accumulate a body of skills and experience that will help you later on. If you do not perform the work yourself, then you have cheated yourself out of the education. Employers in our field can (and do) screen applicants for skills and knowledge. You will perform poorly (and discredit UCI) if you do not practice now by doing your own work.

Disability Services

If you need any accommodations, please contact the Disability Services Center (DSC) and make the appropriate arrangements through them. If you are starting the class aware of necessary accommodations, I highly suggest you have prepared them by the middle of week 2, which will be a bit over one week before the first in class test.

Tentative Schedule

Please refer to the course website for a more up to date schedule.

Week	Topic	Midterms
1	Intro and Foundations	
2	Sorting algorithms and Divide and Conquer	
3	More Divide and Conquer	1
4	Introduction to Dynamic Programming	
5	Dynamic Programming	
6	Dynamic Programming	2
7	Greedy Algorithms	
8	Greedy Algorithms	
9	Greedy Algorithms	3
10	Miscellaneous topics and/or review	

Important dates:

- Midterm 1: July 12
- Midterm 2: July 31 or August 2
- Midterm 3: August 23
- Selfcare day: July 26 or August 2 (No class! Do something to help yourself)
- Final: August 30