CSCI 270: Introduction to Algorithms and the Theory of Computing (Fall 2022)

Most recent announcement posted: 08/19/2022.

1. Basic Information

1.1 Course Staff

1.1.1 Instructors

Name	E-mail (@usc)	Office	Office Hours
David Kempe	David.M.Kempe@Gmail.com	SAL 232	Monday 14:00–15:30 (SAL 232, masked) Tuesday 15:45–17:00 (SAL 232, masked) Thursday 15:45–17:00 (Zoom and SAL 232, masked)

1.1.2 Teaching Assistants

Name	E-mail (@usc)	Office	Office Hours
Curtis Bechtel	bechtel	TBD	TBD
Calvin Leng	cleng	TBD	TBD
Neel Patel	neelbpat	TBD	TBD
Serban Stan	sstan	TBD	TBD

1.1.3 Course Producers

Name	E-mail (@usc)	Office	Office Hours
Natalie Abreu	nrabreu	TBD	TBD
Aadesh Bajaj	aadeshba	TBD	TBD
Rohen Chawla	rrchawla	TBD	TBD
Sammie Fan	sxfan	TBD	TBD
Em Fang	ejfang	TBD	TBD
Aditya Hariharan	adityaha	TBD	TBD
Catriona Jung	cyjung	TBD	TBD
Vikram Kher	vkher	TBD	TBD
Max li	maxl	TBD	TBD
Yutong Li	yli81711	TBD	TBD
Phuong Pham	phuongp	TBD	TBD
Aditya Prasad	aprasad4	TBD	TBD
Bruno Segovia	segovia	TBD	TBD
Grace Shan	gracesha	TBD	TBD
Dongming (Alvin) Shen	alvin.shen	TBD	TBD
Lucy Shi	lucys	TBD	TBD
Jarrett Spino	spino	TBD	TBD
Albert Yan	albertzy	TBD	TBD
Tianyi (Lorena) Yan	tianyiy	TBD	TBD
Karina Yang	karinaya	TBD	TBD
Yuxing (Tom) Zhou	yuxingz	TBD	TBD

1.2 Lectures, Discussion Sections, Textbook, Important Links

1.2.1 Lectures, Discussions

Event	Time	Location
Lecture	T/Th 11:00–12:20	SGM 101
Lecture	T/Th 14:00–15:20	SGM 124
Discussion	F 10:00–11:50	MHP 101
Discussion	F 12:00–13:50	MHP 101
Discussion	F 14:00–15:50	MHP 101

1.2.2 Quizzes, Exams

Event	Time	Location
Prerequisite Quiz	Thursday, August 25, 19:00–20:00	online

Midterm Exam	Thursday, October 6, 19:00-20:50	TBD
Final Exam	Wednesday, December 14, 14:00-16:00	TBD

Apart from these scheduled exams, the quiz section will not be used.

1.2.3 Textbook and other material

The course textbook is **Algorithm Design** by Jon Kleinberg and Eva Tardos, referred to as KT below. We will lean heavily on the textbook, though we will occasionally assign additional reading which will be linked here.

1.2.4 Important Links

- Course Website.
- Course Piazza Page.

2. Announcements

- **08/19/2022:** The course website is up.

3. Course Schedule by Week

- Week 1: Introduction, Stable Matching (2 lectures)
  - Required Reading: KT chapter 1
  - Review Reading: KT chapters 2 and 3
  - Recommended Reading: Shaddin's note on optimizing and analyzing runtime, using the Gale-Shapley algorithm as an example.
  - Optional reference: Chapter 10.4 from the Algorithmic Game Theory book (To freely view the contents, you'd have to go through the USC library proxy).
  - Additional optional references: This paper by Irving et al on optimizing over stable matchings (no paywall if accessing on campus, or online through USC library proxy), and the book by Gusfield and Irving on algorithmic questions in stable matching (available online through USC library proxy).
- Weeks 2-3: Greedy Algorithms (4 lectures)
  - Required Reading: KT chapter 4
- Weeks 4-5: Divide and Conquer (4 lectures)
  - Required Reading: KT chapter 5
- Weeks 6-8: Dynamic Programming (5 lectures)
  - Required Reading: KT chapter 6
- Weeks 9-10: Max Flow / Min cut (4 lectures)
  - Required Reading: KT chapter 7
  - Additional reference: Non-terminating example for the Ford-Fulkerson algorithm with real-valued capacities can be found in the Wikipedia article.
  - Additional reference: Bobby Kleinberg's notes on the Edmonds-Karp algorithm and its analysis.
  - Additional reference: David Kempe's notes on the Edmonds-Karp algorithm.
- Weeks 11-13: NP-Completeness (5 lectures)
  - Required Reading: KT chapter 8
  - Additional Reference: Compendium of NP optimization problems
  - Additional Reference: Garey and Johnson
  - Additional reference: Paper on the possible worlds of complexity theory
- Weeks 13-15: Computability theory (4 lectures)
  - Required Reading: Chapter 1 of Arora/Barak
  - Recommended Reading:
    - Lecture notes by Jeff Erikson: Turing Machines, Undecidability, Universal Models of Computation
    - Wikipedia has a nice summary of Cantor's diagonal argument.
    - David Kempe's brief notes on recursion theory.
  - Additional references: Turing's original 1936 paper
  - Fun reading: The graphic novel Logicomix for a fictionalized account of the development of formal logic and the foundations of computation

4. Prerequisites

- CSCI 170
- CSCI 104L

These prerequisites will be tested in a quiz in Week 1. More specifically, you should be familiar and comfortable with the following:

- Writing rigorous proofs, including using techniques such as induction and contradiction.
- Big-O notation (the meanings of O, Ω, Θ, their differences and how to apply them).
- Basic set, function, sum, and product notation, and how to use/manipulate them.
- Basics about relations and their properties.
- Some basic series, specifically, arithmetic, geometric, and Harmonic series.
- Runtime analysis, and what it means to prove upper and lower bounds on the runtime of an algorithm.
- Basic combinatorics: counting, pairs, permutations, subsets, inclusion/exclusion, Pigeon Hole Principle.
- Basic probability: probability, events, independence, expectations, conditioning, Birthday Paradox.
- Data Structures and the opertions they support, as well as the different implementations and their running times for the operations.
- In particular, you should know stacks, queues, heaps (priority queues), and maps. For maps, you should be comfortable with implementations as balanced search trees and hashtables, and understand their relative advantages.
- Breadth-First Search, Depth-First Search, Dijkstra's Algorithm
- The basic sorting algorithms: Insertion/Selection/Bubble Sort as well as Quick Sort, Merge Sort, and Heap Sort.
- **Importantly, the quiz will also contain a question on the CSCI 270 academic integrity policy.**

  If you want to check how well you've reviewed the material, a sample/practice quiz has been posted. We recommend that you only take the sample quiz at a point when you feel you have already reviewed the material well, so it serves as a useful calibration.

5. Grading and Homeworks

5.1 Grade percentage overview

- Homework (40%): There will be homework assignments roughly every week (a total of about 10-11). This is a theory class, and therefore homeworks will be proof-based and will not involve any coding. The homeworks are intended to be challenging, with individual problems requiring possibly several days of thought, so plan ahead. We expect the **average** student to spend about 10-15 hours on a homework - if this class is difficult for you, it might take you longer.
- Midterm Exam (24%): The midterm exam will be held on Thursday, October 6, during the scheduled quiz section from 7:00pm to 8:50pm. It will cover the class material up to that point.
- Final Exam (34%): The final exam will be held on Wednesday, December 14, from 2:00-4:00pm. It will be cumulative (i.e., cover the material of the entire semester), though more focus will be on the second half of the semester.

- Prerequisite Quiz (2%): The prerequisite quiz will be held online on Thursday, August 25 – from 7:00pm until about 8:00pm. It will cover material you should be very comfortable with from CSCI 104 and CSCI 170. See above for more detail on the material.

Based on the weighted percentage, the following table tells you the final grade. If our exams are more difficult than anticipated, we might make the scale slightly more lenient, but it will definitely not become more strict.

p	Grade
86%	A
79%	A–
72%	B+
65%	B
58%	B–
54.5%	C+
51%	C
47.5%	C–
40%	D
<40%	F

Notice that we do not give D+ or D– grades.

5.2 Regrades

If you feel that you were graded unfairly (too high or too low) on an exam or homework, you are welcome to request a regrade. Here are the relevant policies:

- Regrade requests must be submitted in writing to one of the TAs. For exams, they must be submitted in writing to one of the instructors.
- Except on the final exam, you must wait for at least 48 hours after the homework/exam is returned before you can ask for a regrade. (This is to give you time to think through the grading and make sure you still feel that the grading was wrong.)
- You have at most two weeks after the return of the homework/exam to submit your regrade request. If you have an insanely busy 2 weeks but know you want a regrade, you can send a brief "Heads up. I will submit a detailed regrade request on such–and–such date" e–mail instead. (This is to prevent desperate attempts to submit all homeworks for regrades at the end of the semester if you just need 2 more points for a higher grade.)
- When you ask for a regrade, we will regrade your entire homework/exam, not just the question which you feel was misgraded. This is to ensure fairness overall.

5.3 Homework Submission and Late Submission

We will use Gradescope for homework submission and grading. Your Gradescope accounts will be set up by us, and you will receive email notification.

A guide for homework submission is available here.

If you have any questions about Gradescope and homework submission, please ask on Piazza. Contact the TAs and the professors if you have questions about grading or homework policies.

All homework submissions must be PDF files. Starting on Homework 3, the solutions must be typed up. Homeworks 1 and 2 can be submitted as photographed/scanned handwritten solutions, and the graders will make a reasonable attempt to read them. If as a result of the handwriting quality or the capturing, the solution is illegible, it may result in lost points.

For typing up solutions, we highly recommend the (free) LaTeX document processing language. It is a flexible markup language you can use with any editor, and the de facto standard for most CS and math papers. The output looks **much** nicer than MS–Word or others. You may really enjoy using this opportunity to learn it.

5.3.1 Late Homework Policy

You will be allowed 10 total late days for homework, to be used in integer amounts and distributed over homeworks as you see fit. A late day is 24 hours from the original homework deadline, or any part thereof. Additional late submissions will only be considered in exceptional circumstances. In order to allow us to distribute sample solutions in a timely manner, no homework can be submitted more than 5 days late; in some cases, we may further constrain late submissions (e.g., close to an exam).

There are no special procedures to use a late day. We will make the final deadline 5 days after the regular deadline. Notice that Gradescope does not keep track of your late days. We will not accept late submissions that exceed your total late day limit without explicit prior approval.

We stress that late days are meant as an unbureaucratic way for you to give yourself extensions when you have legitimate reasons to submit late. You can also use them as "cushion" if a homework is giving you exceptional trouble or you have a lot going on. However, this does not mean that you are entiteled to 10 "cushion days" **in addition** to late days for legitimate reasons, such as illness, family emergency, etc. In particular, please do not make the (common) mistake of using up all your late days on the first three homeworks because you underestimated the difficulty, then finding yourself struggling a lot later in the semester.

5.4 Homeworks

All homeworks (and subsequently HW solutions) will be posted on Piazza once they are assigned.

6. Collaboration and Academic Integrity

We will expect the utmost in academic integrity from each of you, and we will trust you to abide by the commensense norms of academic integrity (see the statement on academic integrity below). A more detailed discussion of what is and is not appropriate in this class (as well as some general advice on plagiarism) is given below. Most importantly, if you have any doubts as to whether a certain practice is considered a violation of academic integrity, it is your responsibility to preemptively ask the instructors or graduate TAs (not the CPs) before engaging in the practice.

While we will not go out of our way to identify violations of academic integrity, those who are found to be in violation will be referred to the office of judicial affairs without exception. Furthermore, consistent with USC's recommended policy, we will suggest to SJACS a grade of F in the class for any confirmed violation. While this may be stricter than some introductory classes, it reflects the fact that by now, we expect you to be more mature, and have gotten the "high–school mindset" out of your system.

6.1 Plagiarism

Aside from a handful of cases that are obviously cheating (e.g., consulting a textbook or other source in a closed–book exam, using a computer or calculator when you were not allowed to), most cases of cheating involve what is called "plagiarism" which is defined as "passing someone else's work off as your own without attribution" Plagiarim can thus take many forms:

- Copying solutions from a classmate.
- Copying solutions from a friend who took the class earlier.
- Copying solutions from a manual you found online, or some other textbook which happens to have the solution.
- Paying someone to do the homework for you.

It does not change anything if you make small changes while/before copying a solution, if the gist is still the same. A solution does not have to be copied verbatim to be considered plagiarism.

Notice from the definition that there is a surefire way to avoid committing plagiarism: attribute all sources you use. Even if you copied a solution word for word from an online source, you are not committing plagiarism if you clearly and visibly write on top "I copied this solution from the following source: ... This is not my own solution." In this scenario, you will likely receive a 0 on the question (since you did not plagiarism if you clearly and visibly write on top "I copied this solution from the following source: ... This is not my own solution." In this scenario, you will likely receive a 0 on the question (since you did not actually solve it), but you did not commit plagiarism.

Zooming out a bit: in any paper or assignment you write, any project report in your professional life, or even a presentation, it is a great idea to attribute all ideas and text you took from somewhere else. Not only does it avoid plagiarizing in an academic environment, but your colleagues will also really appreciate you for giving them credit for the slides/text/ideas you borrowed from them if you make it clear that it was theirs.

Zooming back in: on your homeworks, you should err on the side of safety, and cite all sources. That is, list all websites where you looked up ideas (even if you didn't copy them verbatim), and any classmates or other people that you had substantial discussions with. That way, we (the course staff) can make the call how much you added beyond your sources, but in any case, you will not be accused of plagiarizing.

6.2 Collaboration on Homeworks

Naturally, the prerequisite quiz, midterm and final will be individual work. However, since much of CS (and math) work is often collaborative, we allow collaboration and discussion among students for the homework. The focus should be on discussing high-level ideas, whereas you should not spell out all details in your collaborative work. More specifically, here are concrete guidelines for your collaborations:

- You must acknowledge everyone who you collaborated with. If you worked with different people on different problems, then list them on the specific problems. In the spirit of the preceding section, you should err on the side of over-reporting, and list people even if you just briefly chatted and exchanged high-level ideas. Feel free to specify how you collaborated, e.g., "I gave Edsger Dijkstra a little hint about the loop invariant." or "Donald Knuth suggested to use Dynamic Programming on Question 3."
- Your discussions should focus on high-level ideas. You should stop your group collaboration well short of writing out the details of an algorithm or proof.
- If you meet in a group to work on the homework, your group must include no more than 5 individuals.
- You are not allowed to take any written material with you out of the meeting, including screenshots or photographs. In other words, reconstruct the key ideas by yourself after you've worked them out with your classmates.
- You must wait at least 30 minutes after any discussion with fellow students, and then *write up your solutions independently*. If your discussion was accidentally too detailed, then you should significantly extend this 30-minute interval, and avoid thinking about CSCI 270 material in between. (If you want to see the rationale for this rule, look at a homework solution you wrote, and try to memorize it. Wait for a day, and then try to write it again as closely as possible. If you compare the solutions, you will likely see that they are extremely different.)

Of course, asking the instructors, teaching assistants, or course producers for hints is always ok. If we feel that a hint would reveal too much, we will simply not answer your questions.

6.3 Consulting outside sources

You may not discuss problems with (or seek solutions or hints from) anyone outside the class, be it strangers on the Internet (e.g., in discussion forums or homework-for-hire websites), or a friend who took the class previously.

You also must not consult solution manuals or google for solutions to similar problems. Both are absolutely considered cheating. That being said, you are allowed (and encouraged) to consult general sources (such as Wikipedia, books, or course notes) pertaining to course content in order to broaden your perspective, so long as you do not specifically seek out material related to a particular homework problem. As discussed above, you should cite all such materials that you consulted.

6.4 USC's text on Academic Conduct

Plagiarism - passing someone else's ideas as your own, either verbatim or recast in your own words - is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in *SCampus* in Section 11, *Behavior Violating University Standards* https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions/. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct/.

7. Additional Course Syllabus Items

7.1 Course Description

The course covers the fundamentals of algorithm design and the theory of computing. Much of the work of a computer scientist is problem solving using computational artifacts (such as modern computers), and problem solving involves describing step-by-step procedures that can be followed by machines. These are called algorithms. Many fundamental techniques for algorithm design, as well as specific algorithms themselves, recur throughout all areas of computer science, and computer scientists must be able to apply design and analysis techniques to devise efficient algorithms and compare them.

The flipside of algorithm design is knowing the limits of algorithm design: when are problems so intractable that they either cannot be solved at all, or can --- to the current state of scientific knowledge --- only be solved by very inefficient algorithms? Such knowledge is necessary to avoid searching for solutions to problems that simply cannot be solved.

The course CSCI 270 provides an introduction to both of these complementary pieces: it covers greedy algorithms, Divide&Conquer algorithms, Dynamic Programming and their corresponding analysis techniques. It also contains an introduction to the theory of NP-completeness and computability theory. At the discretion of the instructor, it will contain a discussion of a subset of the following topics: algorithms for flows and cuts, linear programming, the role of randomization in computing, approximation algorithms, number-theory based cryptographic algorithms, and non-standard models of computing.

7.2 Non-Discrimination

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the *Office of Equity and Diversity* http://equity.usc.edu/ or to the *Department of Public Safety* http://capsnet.usc.edu/department/department-public-safety/online-forms/contact-us. This is important for the safety whole USC community. Another member of the university community - such as a friend, classmate, advisor, or faculty member - can help initiate the report, or can initiate the report on behalf of another person. *The Center for Women and Men* http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage sarc@usc.edu describes reporting options and other resources.

7.3 Support Systems

A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the *American Language Institute* http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. *The Office of Disability Services and Programs* http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, *USC Emergency Information* http://emergency.usc.edu/ will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.

7.4 Emergency Preparedness / Course Continuity in a Crisis

In case of a declared emergency if travel to campus is not feasible, USC executive leadership will announce an electronic way for instructors to teach students in their residence halls or homes using a combination of Blackboard, teleconferencing, and other technologies.