Policies

Miscellaneous

- Homework FAQ
- Piazza Etiquette
- Latex Guide

Homeworks

Every Wednesday evening, homework is posted on Piazza, and it is due the following Wednesday by **10:00 PM**. There will be a 30-minute grace period afterwards, with a late deadline at 10:30 PM. **We strongly recommend you use the late deadline to protect against technical issues and have your homework finished by 10:00 PM**. We will not accept homeworks that were delayed for technical issues after 10:30 PM.

There are no exceptions for homework deadlines. At the end of the semester, we will automatically drop your two lowest homeworks, including any you may have received a 0 on. You should save these drops for when you need them, as there are no exceptions for number of drops either.

We want you to focus on understanding the material for homeworks, not necessarily on maximizing your score. For this reason, we will give you **full credit on a homework if you score above 80%**.

Programming Project

There will be one challenging, yet (hopefully) fun programming project. The programming project cannot be dropped. Details on the project will be announced after the second midterm.

Exams

There will be two midterms and one final exam.

- Midterm 1: 9/30, 8pm-10pm
- Midterm 2: 11/7, 8pm-10pm
- Final: 12/19, 8am-11am

DSP exams will be 6pm-10pm the day of the midterm, and 8am-2pm the day of the final.

Grading

We will compute grades from a weighted average, as follows:

- Homeworks: 15%
- Programming Project: 5%
- Midterm 1: 20%
- Midterm 2: 25%
- Final exam: 35%

The class will be curved *at the end of the semester*. There is no clobbering policy, but we *may* take into account participation on Piazza, discussion, office hours, HW Party and student performance *if the grade is on a boundary.*

Collaboration

The guiding principle for homework collaboration is that your writeup of the solutions should be **yours only**.

You are encouraged to work on homework problems in study groups of up to five people; however, **you must always write up the solutions on your own.** Similarly, you may use books or online resources to help solve homework problems, but **you must always credit all such sources** in your write-up and you must never copy material verbatim. However, as a general rule of thumb, **you should never possess solutions to exact homework questions other**

than those solutions you have written yourself. We realize that it is sometimes possible to stumble upon solutions on accident. If this happens, please cite the source and write up solutions in your own words.

We believe that most students can distinguish between helping other students and cheating. Explaining the meaning of a question, discussing a way of approaching a solution, or collaboratively exploring how to solve a problem within your group are types of interaction that we strongly encourage. But you should write your homework solution strictly by yourself so that your hands and eyes can help you internalize this material. At no time should you be in possession of another student's solution. You may discuss approaches but your solution must be written by you and you only. **You should explicitly acknowledge everyone whom you have worked with or who has given you any significant ideas about the homework.**

Further, it is your responsibility to ensure that your solutions will not be visible to other students. If you use Github or another source control system to store your solutions electronically, you must ensure your account is configured so your solutions are not publicly visible. Many popular version control systems provide free repositories to students; staff members or fellow students can help you obtain one.

As a final note, we'd like to point out that collaboration on homework, while encouraged, can be detrimental to your learning if misused. In particular, avoid collaborations where you do not contribute enough to your own satisfaction. Such a collaboration not only cheats you out of an opportunity to learn through homework, but can also affect your confidence. If you feel that you are not contributing enough to your group, then try to spend time thinking about the problems alone before working with your group. If you end up solving the problem all by yourself that's great! And if not, you'll still be prepared to better contribute to your group.

If you're ever in doubt about what constitutes academic dishonesty, always ask a TA or on Piazza.

> **Warning**: Your attention is drawn to the Department's Policy on Academic Dishonesty. In particular, you should be aware that copying or sharing solutions, in whole or in part, from other students in the class (or any other source without acknowledgment) constitutes cheating. Any student found to be cheating risks automatically failing the class and being referred to the Office of Student Conduct.

Other Policies and Information

Announcements

All announcements will be posted to Piazza. **You are responsible for staying up-to-date with announcements made on Piazza.**

Contacting Staff

Please contact course staff through Piazza. Be careful about posting homework questions (do not give away the answers or parts of the answers). If you want only the instructors to see your post, mark it as private.

If you want to contact the Professors and Head TAs for special requests, email cs170@berkeley.edu.

Prerequisites

The prerequisites for CS 170 are CS 61B and CS70. You will need to be comfortable with mathematical induction, big-O notation, basic data structures, and programming in a standard imperative language (e.g., Java/C/Python). Another "prerequisite" for doing well in the course is mathematical maturity, or the ability to think about and work with proof-based math (which CS70 can help build).

Textbook

The required textbook is Algorithms by Dasgupta, Papadimitriou, and Vazirani, also known as DPV.

Discussion sections

Attendance at discussion sections is not required, although sections cover important material not covered in lecture that will appear on exams. You may attend any discussion section. Outside of your discussion section, you should feel free to attend any of the staff office hours (not just your section TA's office hours) and ask any of us for help.

Office Hours

TA office hours are for homework- and midterm-related questions, or minor concerns. Prof. Rao's and Prof. Raghavendra's office hours are intended for major concerns or high-level questions about the course material.

> We prioritize conceptual questions over homework questions in office hours.

Doing Homeworks

Homework parties

We will hold two weekly homework parties: **Mondays 2-4PM in Wozniak Lounge in Soda Hall and Tuesdays 5-8PM in Cory 540AB**. Homework parties are completely optional, but you are encouraged to come! This is a fantastic opportunity to meet people and form a group to study for the class and work on homework.

Formatting Homework

You can write your homework in one of the following ways:

1. Typeset your answers using LaTeX. We strongly recommend doing this.
2. Type your answers using a mainstream word processor.
3. Neatly handwrite, then use a flatbed scanner or smartphone app (such as CamScanner) to produce a PDF. **Do not simply take a photo and dump it into a PDF.**

The readers work hard to grade your homeworks in a timely manner, and with useful feedback. Given the size of the course, we require a certain format of all homeworks: **each problem must start on a new page, and clearly display the problem number; if a problem has multiple parts, make sure that each part fits on a page.**

List your collaborators on the first page of your submission. If you have no collaborators, you must explicitly write "none".

Responding to Algorithm Problems

Oftentimes, a problem will ask you to "give/design/devise/formulate/create/etc. an algorithm" or to "show how" to solve some computational task. In this case, write your solution in the 3-part algorithm format:

1. **Algorithm description**. This can come in terms of pseudocode, or a description in English. It must be unambiguous, as short as possible (but no shorter), and precise.
    - Your pseudocode does not need to be executable. You should use notation such as "add X to set S" or "for each edge in graph G". Remember you are writing your pseudocode to be read by a human, not a computer.
    - See DPV for examples of pseudocode.
2. **Proof of correctness**. Give a formal proof (as in CS 70) of the correctness of your algorithm. Intuitive arguments are not enough.
    - Again, see DPV for examples of proofs of correctness.
3. **Runtime analysis**. You should use big-O notation for your algorithm's runtime, and justify this runtime with a runtime analysis. This may involve a recurrence relation, or simply counting the complexity and number of operations your algorithm performs.

Submitting Homework

Homework must be submitted via Gradescope. See the homework deadline policy.

Re-grading policies

Regrading of homeworks or exams will only be undertaken in cases where you believe there has been a genuine error or misunderstanding. Any requests for grade changes or re-grading must be made by the deadline. Submit regrade requests via Gradescope.

Electronics Policy

Students may use laptops and other electronics in the class. But if you do so, please sit only in the rows at the back of the class, so as to not disturb anyone else.

Extra credit opportunities

The instructors may offer optional activities that provide a small amount of extra credit, for those eager for a tougher challenge. We recommend you do them only if you enjoy these problems, as spending effort on them is likely not the most efficient way to maximize your final course grade.

Advice

The following tips are offered based on our experience with CS 170:

Don’t fall behind!

In a conceptual class such as this, it is particularly important to maintain a steady effort throughout the semester, rather than hope to cram just before homework deadlines or exams. This is because it takes time and practice for the ideas to sink in. Make sure you allocate a sufficient number of hours every week to the class, including enough time for reading and understanding the material as well as for doing assignments. (As a rough guide, you should expect to do at least one hour of reading and two hours of problem solving for each hour of lecture.) Even though this class does not focus on projects, you should plan to spend as much time on it as your other technical classes.

Do the homeworks, and go over homework solutions

The effort you spend on the homework assignments is the largest factor in your mastery of the material, and thus performance in the class as a whole. The staff posts detailed solutions of the homework problems each week, often with multiple solution approaches and commentary. Regardless of how well you did on the homework, make it a priority to carefully read the solutions and compare them with your own work, even for the problems you got right. The main reason to do so is that sample solutions will often be simpler and cleaner. You will have a higher chance of solving similar problems under exam pressure if you have a simple picture. You may well learn a different way of looking at the problem, and you may also benefit from emulating the style of the solutions. It might seem attractive in the short term to just ignore homework solutions, or delay reading them until the day before the final. Some students will even try to finish the homework faster by looking up solutions to problems online. But these practices are detrimental to understanding in the long term, and thoroughly understanding solutions is a critical step to mastering the material. These homeworks are provided to you not as a chore, but as an opportunity for you to challenge yourself and explore the wonderful world of algorithms. Don’t short-change yourself by skimping on them. As an added incentive, we will draw from the homework problems when writing exams.

Don’t wait until the last minute to start homeworks!

Read through the homework problems as soon as they are available, and let them percolate in your brain. Schedule multiple times throughout the week to attempt the problems. Often, the solution will only come to you the second or third time you try a problem. Definitely do not wait until the night before it is due to start working on the homework. If you do this (alas, procrastination is all too common), (a) the stress will make it harder to be creative and solve the problems; (b) you will be tempted to ask your homework group for too many hints (or outright solutions) to the problems, which borders on cheating and impedes your learning, and (c) the stress will sap the joy of finding algorithms!

Make use of office hours and discussion sections

The instructor and TAs hold office hours expressly to help you. You are free to attend as many office hours as you wish! Discussion sections are an opportunity for interactive learning. The success of a discussion section depends largely on the willingness of students to participate actively in it. As with office hours, the better prepared you are for the discussion, the more you are likely to get out of it.

Form study groups!

As stated above, you are encouraged to form small groups (e.g. of four students) to work together on homeworks and on understanding the class material on a regular basis. If you're unsure about how to find a study group, attending homework parties is a great way to getting started. In addition to being fun, this can save you a lot of time by generating ideas quickly and preventing you from getting hung up on some point or other. Of course, it is your responsibility to ensure that you contribute actively to the group; passive listening will likely not help you much. And recall the caveat above that you must write up your solutions on your own. Spend some time on your own thinking about each problem before you meet with your study partners; this way you will be in a position to compare ideas with your partners, and it will get you in practice for the exams.