CS1200 Spring 2026:
Introduction to Algorithms and their Limitations

Syllabus

Basics

Course website (including past years' materials): https://harvard-cs-1200.github.io/cs1200/

Shopping/registration information:

- Watch the Fall 2024 course preview video
- Read this syllabus carefully, which includes a number of important course processes and policies, some of which have changed from Fall 2025
- Send any questions to cs1200-spring26@g.harvard.edu
- Anurag's office hours: Mon, Wed 12:45-1:45 pm

Lecture times: MonWed 11:15 pm-12:30 pm Eastern, Science & Engineering Complex (SEC), 150 Western Ave., Allston, Room 1.321. In-person attendance at lectures is required. Lectures are recorded and can be (re)watched asynchronously, except the "Sender-Receiver Exercises": see below.

Course Staff

Instructor: Anurag Anshu (he/they)

Head teaching fellow: Aarna-Pal Yadav

Course Heads Email (*please use this rather than emailing us individually*):
cs1200-spring26@g.harvard.edu

Other Teaching Fellows:
Audrey Cherilyn
Bridget Ma
Emily Gao
Prisha Seth
Vihaan Gupta

Whenever possible, please post questions for the course staff on Ed (privately if needed) rather than emailing us, so that we can all see the question and responses.

Overview

Looking at the world around us, we see computers solving problems on incredibly large scales: finding webpages relevant to our internet searches and returning them in sorted order, computing the quickest way to reach a destination given current traffic conditions, matching people on dating sites. How is this possible? More computing power? Intensive application-specific engineering? While these certainly have had a role to play, in CS1200, you will see and learn how to use general algorithm design principles that cut across application domains and remain relevant even as computing technology changes.

First among these principles is mathematical abstraction, whereby we capture the essence of a computational problem (as well as the notion of a "computer") so that we can develop and analyze solutions independent of an implementation. Given these mathematical abstractions, we can apply a toolkit of basic algorithmic techniques in the search for solutions, and then gain certainty in their correctness and efficiency through rigorous mathematical proofs. Furthermore, the powerful concept of reductions will allow us to identify relationships between computational problems that seem very different on the surface, and thus automatically transfer solutions from one to another.

At the same time, some important computational problems have defied the search for algorithmic solutions. Computer scientists would love to have debugging tools that determine whether their programs can crash, natural scientists would love to have simulators that quickly determine the energy-minimizing states of physical or biological systems, and university registrars would love to be able to automatically schedule classes in a way that optimally maximizes the use of the best classrooms. Why have no scalable algorithms been found for these problems?

In the last part of CS1200, we will see that many important computational problems are inherently unsolvable—they have no general algorithmic solution whatsoever! Others are solvable, but have no efficient algorithm—the minimum computation time inherently grows exponentially with the size of the problem instance. Uncovering these phenomena (known as "uncomputability" and "intractability", respectively) are unique benefits of a mathematically rigorous approach to algorithms. While we may sometimes be satisfied with empirical demonstrations of the performance of an algorithm we have found, a proof seems to be the only way to convince ourselves that there is no algorithm whatsoever.

To summarize, CS1200 aims to give you the power of using mathematical abstraction and rigorous proof to understand computation. Thus equipped, you should be able to design and use algorithms that apply to a wide variety of computational problems, with confidence about their correctness and efficiency, as well as recognize when a problem may have no algorithmic solution. At the same time, we hope you will gain an appreciation for the beautiful mathematical theory of computation that is independent of (indeed, predates) the technology on which it is implemented.

Learning Outcomes

By the end of the course, we hope that you will all have the following skills:

- To mathematically abstract computational problems and models of computation
- To design and implement algorithms using a toolkit of algorithmic techniques
- To recognize and formalize inherent limitations of computation
- To rigorously analyze algorithms and their limitations via mathematical proof
- To appreciate the technology-independent mathematical theory of computation as an intellectual endeavor as well as its relationship with the practice of computing
- To engage effectively in a collaborative theoretical computer science learning community, supporting your peers' learning as well as your own
- To clearly communicate mathematical proofs about computation to peers, conveying both high-level intuition and formal details.

Prerequisites

Experience with proofs and discrete mathematics at the level of Computer Science 20, and (Python) programming at the level of CS32/CS50. If you haven't taken these courses, we recommend using the CS20 Placement Self-Assessment and Problem Set 0 (PS0) to gauge your preparation. You can see past years' PS0 on the course website; this year's will be posted before the start of the semester, on Jan 22. Note that while PS0 does not require any new material from CS1200, it is a full-fledged problem set that may require substantial effort to complete. The course staff will hold review sessions and office hours during the first week of classes to support you in doing PS0.

Enrollment

Students who enroll in the course on or before Feb 2 are expected to submit PS0 on time or use up to three of their late days. College students who wish to enroll after Feb 2 must email course heads for permission to enroll. If you wish to switch into CS 1200 from another course, you should include in your email evidence of work that you completed in the class you are leaving. If you wish to join CS 1200 late for any other reason, we request a note from your resident dean or academic advisor.

We are happy to have college students as auditors. Just email the courseheads and we will grant you access to the course platforms. However, please note that current auditors may not take the course for credit in future offerings.

The last day that course heads will approve a change of grading basis (for example, from letter grade to pass/fail), is April 6.

Curricular Context

CS1200 is meant to provide a gentler entry into theoretical computer science than CS1210 (limits & models of computation) and CS1240 (algorithms). CS1200 can be taken before CS1210 or CS1240, but these courses will *not* assume CS1200 as a prerequisite. See this FAQ for a more detailed comparison between CS 1200, 1210, and 1240, and the CS Advising site for how CS1200 fits into the CS concentration requirements.

Expectations and Format

This is the sixth offering of CS1200, and the course is still somewhat experimental. By joining the class, you will be partners in the continued development and improvement of the course. We will solicit feedback from you periodically and expect to make adjustments based on it, so everything below is subject to change.

The main planned components of the course are as follows:

Main class meetings (MonWed 11:15am-12:30pm): About a third of our class meetings will begin with an approx. 30min *Sender-Receiver Exercise (SRE)* done in pairs, where the partners, the “sender” and “receiver,” engage in an interactive dialogue to develop a joint understanding of a mathematical proof (related to recent class content) that the sender has studied beforehand. We recommend that senders spend at least an hour reviewing the proof ahead of the SRE. In the last 5min of each SRE, each of you will fill out a short reflection survey on

takeaways from the SRE and ways you can improve them in the future.

The remainder of our class time will consist of interactive lectures on new material.

Regular attendance in class is required. In particular, the SREs will develop important skills such as deconstructing and communicating proofs. Additionally, your peers will be depending on your participation in the SREs and these will be a part of participation grading (described below).

If you will miss a lecture with an SRE due to a pre-planned academic, extracurricular, or religious activity, then you should notify the course staff via email at least 4 days before the missed class, and cc your resident or faculty dean (or advisor, for graduate students); see here for a list of resident deans. You will be responsible for organizing and completing a make-up with another College student. The course staff will give you access to a spreadsheet to find another College student who couldn't attend lecture, and schedule a time to do it with them outside of class. Your SRE survey deadline on Gradescope will be extended by 3 days.

If you miss a lecture with an SRE due to an illness, injury, or personal emergency, you will be completely excused from that SRE if you provide a note from HUHS (illness/injury) or an email from your resident dean or faculty dean (or advisor, for graduate students). We strongly recommend that you still review the material on the SRE with a classmate as soon as you are able; problem set and exam questions will rely on the SREs.

If you miss a lecture with an SRE for any other reason, such as oversleeping or missing the shuttle, there is no opportunity for make-up, but we will drop the lowest grade received among the 7 SRE surveys.

If you are absent from a lecture which does not contain an SRE or an exam, you do not need to contact the course staff about an excused absence or make-up, but you will be expected to review all missed material.

Sections (weekly, times TBD): Each week, a section handout consisting of practice problems and review of difficult concepts will be provided. TF-led sections will be held weekly, at times and places TBD to accommodate as many students as possible. Attendance is optional, but highly recommended. Section provides an opportunity for real problem-solving practice, which is important for developing the course skills, as well as preparing you for problem sets and exams. For students who don't go to one of those sections, we recommend working through the section materials independently. You will find sections most useful if you have read and started thinking about the problem sets and the corresponding material in advance.

Office hours (every week, times TBD): The teaching staff will have regular office hours, both in-person and on Zoom, to help guide you as you work through the material. Before coming to office hours with questions about the problem set (other than clarifications on what is being asked), it is important to invest real effort in trying to solve the problems without staff assistance.

If you get stuck, the teaching staff will help you get unstuck, not by giving hints, but by probing your thought process and understanding of the content and/or reviewing material and related problems. The goal is to help you build the skills that will enable you to discover solutions for yourself.

Problem sets: There will be approximately 11 weekly problem sets, with release and due dates provided on the course schedule and Google Calendar. Some of the problems may require deep thought, so you are strongly encouraged to begin them early and to brainstorm in groups of 2-3 students. (See Collaboration Policy below.) Most problem sets will include a qualitative question for you to reflect on how you are engaging with the course (see Participation below) and/or on what you are learning in the course.

Asynchronous discussions: We will use the *Ed* platform for discussions, Q&A, and announcements.

Participation: The last three learning objectives listed above ("To appreciate…", "To engage effectively…", "To clearly communicate…") go beyond the technical content of CS1200 and involve the development of virtues (collaboration, communication, thoughtful learning, etc.) that will serve you well beyond the confines of this course, as a student, as a computer scientist, and as a citizen. To develop these virtues, it is important that you are an active participant in the class, reflecting on how you are engaging with the course, and what you can do to better support your own and your classmates' progress in the class. The qualitative questions on the problem sets and the 5min in-class surveys after each SRE are meant to foster such reflection.

Exams: There will be an in-class, 75min midterm exam on March 11, and a standard 3-hour final exam scheduled by the Registrar.

Students who have conflicts with the midterm exam should email course heads by cs1200-spring26@g.harvard.edu. We cannot guarantee a make-up time but will try to accommodate valid, unavoidable conflicts. College students who have conflicts with the final exam must contact the Registrar (your resident dean can help you with this process.)

College students with DAO testing accommodations must sign up to take the midterm exam at the FAS Testing Center.

Grading

Breakdown. 40% of your grade in CS1200 will be based on the problem sets, with your lowest problem set score dropped (this includes the content questions *AND* related reflection question). The midterm and final exams will comprise 40% of your overall grade in CS1200, weighted proportionally to their length (75:180). The remaining 20% of your grade will be based on your participation, as evaluated by the reflection questions on the problem sets and the SRE surveys.

Problem set grading rubric. The problem sets will be graded using the following rubric:

- N: Not assessable. The assignment is too fragmentary or incomplete for us to assess the level of effort and understanding reached while working on it.
- L: Learning. Significant effort is evident, but there remain important misconceptions or gaps that will limit your ability to apply the skills and/or knowledge in the future (whether in the rest of CS1200, in later CS courses, or outside of your CS education).
- R-: Nearly ready to move on, but with review and revision recommended. You have achieved the main learning objectives of the assignment, but there are some gaps that should be filled in on your path to mastery.
- R: Ready to move on. Your work meets all of the learning objectives of this assignment, and contains only minor errors (which we will note in our feedback to you).
- R+: Beyond ready. The work is nearly perfect, goes beyond expectations in its clarity or insight, and/or explores optional extensions.

Exams will be graded on a numerical scale.

The reason for this coarse grading scale is to move away from nitpicking over point deductions and focus on the end goal of acquiring sufficient understanding to competently apply what you have learned.

Revisions. The material in CS1200 is largely cumulative, so we wish to see you all reach an R-range grade on all assignments. But mistakes and misunderstandings are a part of the learning process, so we will allow you to submit short revisions (in the form of short videos) that clearly explain the misconceptions or gaps in your problem sets and the corrected understanding you have developed by studying the solutions. These revisions can increase up to five of your problem set grades from L to R- or R- to R (not both). On problem sets 0 and 1, these revisions can instead increase your problem set grades from N/L/R- to R. We encourage you to submit revisions on all of your L and R- grades (even if you receive more than five) to ensure that you have achieved the learning objectives of the assignments; at the end of the course, we will use your budget of five in the way that maximizes your final letter grade. To make sure you do not fall behind, there will be a deadline for revisions on each problem set, approximately 1.5 weeks after graded assignments are returned. You may use up to 3 late days on a revision submission. We will not accept revisions on assignments that receive a grade of N (except problem sets 0 and 1) or R.

Participation grading rubric. For consistency, we will use the same N/L/R-/R/R+ scale for grading the elements of your participation grades (namely, the reflection questions on the problem sets and the sender-receiver exercises), but with the following interpretations:

- N: Your submission is missing or shows no engagement with the question or exercise.
- L: Your submission shows only superficial engagement with the question or exercise.
- R-: We will not use this grade for participation.
- R: Your submission demonstrates solid engagement with the question or exercise. We expect that the vast majority of participation grades will fall in this category.

- R+: Your submission goes well beyond expectations, in the thoughtfulness of your reflections and/or your contributions to the course's learning community. We expect that R+ grades on participation will be rarer than on problem sets.

Note that the problem set with the lowest score will be automatically dropped; this drop includes both the content questions *and* associated reflection question. We accept no regrade requests on reflection questions. Also note that, unlike reflection question grades which will be released when the problem set grade is released a week after the regular deadline, we will not be returning SRE grades during the semester, but we will automatically drop your lowest SRE survey grade.

Reflection grade bumps for regular section attendance. At the start of each section (whether it's in-person, hybrid, or remote), the TFs will record a list of people in attendance. For every 4 sections you attend, you will earn one reflection question grade bump (this means that at the end of the year, we will automatically revise any one L grade on a reflection question to an R-; if you have no Ls on any reflection question, we will automatically revise any one R- to an R). Note that you can attend multiple sections in a section cycle for additional support, but we will not double-count your attendance. In summary, you can earn at most two reflection question grade bumps over the course of the semester by attending sections in 8 or more cycles.

Letter grades. Your final letter grade will be obtained by calculating a grade for problem sets, participation, and exams, each on a 4.0 on scale, taking a weighted average of those (.40*pset+.20*participation+.40*exams), and then rounding to the nearest letter grade (using Harvard's conversion between letter grades and GPA).

Your problem set and participation GPAs are determined as follows. First, we will drop the lowest grade as described above. Then we will average your R, R+, and R- grades, with R as a 4.0, R+ as 4.33, and R- as 3.0. Then each L will reduce your GPA for that category by .333 (i.e. one modified letter grade) and each N will reduce it by .666 (i.e. two modified letter grades).

The exams do not have a predetermined translation between exam score and the 4.0 scale. This translation is determined by the instructors assessing the level of proficiency shown by the submitted exams at different scores, using the College's rubric.

Late days. You have a total of ten late days that you can use for problem sets or revision videos (not counting late days used on your dropped pset), using at most three on any one assignment. (That is, a problem set beyond 3 days late will automatically receive an N grade, and a revision video will not be accepted beyond 3 days late.) Late days are counted from 12:00 AM after an assignment's due date. There are no partial late days. If you exceed your total late day budget, each additional late day used will reduce your pset GPA being reduced by .0333 (equivalent to a .333 drop on one pset).

Do not needlessly use up your late days at the beginning of the semester; they are meant for

accommodating unforeseen circumstances or crunches from your other classes or other aspects of your life. Any extensions beyond these late days require a note from your resident dean (or advisor, in the case of graduate students).

If you ever find yourself racing against a problem set deadline or out of late days, remember that it is far better to submit something than to submit nothing. If you score an L or R-, then you can improve your grade later through a revision video. Though we cannot guarantee that a certain amount of submitted work will translate to a particular grade, we advise that you demonstrate significant effort on every problem. Your lowest problem set score will be dropped at the end of the semester.

Diversity and Inclusion[1]

We would like to create a learning environment in our class that supports a diversity of thoughts, perspectives and experiences, and honors your identities (including race, gender, class, sexuality, socioeconomic status, religion, ability, etc.). We (like many people) are still in the process of learning about diverse perspectives and identities. If something was said in class (by anyone) that made you feel uncomfortable, please talk to us about it. If you feel like your performance in the class is being impacted by your experiences outside of class, please don't hesitate to come and talk with us. As a participant in course discussions, you should also strive to be open-minded and respectful of your classmates.

Health Accommodations[2]

If you have a physical or mental health condition that affects your learning or classroom experience, please let us know as soon as possible so that we can do our best to support your learning (at minimum, providing all of the accommodations listed in your DAO letter if you have one).

This course has a default plan for students with the Deadline and Attendance Adjustment (DAA) accommodation:

- You have a total of fifteen late days that you can use for problem sets or revision videos, however the maximum number of late days that may be used on any one assignment is still 3.
- If you miss an SRE due to a disability-related event, please email course heads that you need to use your DAA accommodation, and your absence will be excused. You won't need to provide a HUHS note or email from your resident dean. We strongly recommend that you review the material on the SRE with a classmate as soon as you are able; problem set and exam questions will rely on the SREs.

---

[1] Based on text by Dr. Monica Linden at Brown University.
[2] Based on text by Prof. Krzysztof Gajos at Harvard University.

- If a disability-related event prevents you from submitting a regrade request, we can grant a short extension if you email the course heads before its deadline.

If you are concerned that this DAA plan might not support your needs, please arrange a meeting with the instructor and your DAO advisor.

Support Structures[3]

Everyone can benefit from support during challenging times. If you experience significant stress or worry, changes in mood, or problems eating or sleeping this semester, whether because of CS1200 or other courses or factors, please do not hesitate to reach out to Anurag or other members of the course staff. Not only are we happy to listen and discuss how we can help you cope in CS1200, we can also refer you to additional support structures on campus, including, but not limited to, the below.

- Academic Resource Center (ARC)
- InTouch (for SEAS graduate students)
- Counseling and Mental Health Services, 617-495-2042 (24/7 support line)
- Peer Counseling, including Room 13, 617-366-7375

Collaboration & Generative AI Policies

Students are encouraged to discuss the course material and the homework problems with each other in small groups (2-3 people). Discussion of homework problems may include brainstorming and talking through possible solutions, but should not include one person telling the others how to solve the problem. In addition, each person must write up their solutions independently of each other, and these write-ups should not be checked against each other or passed around. Generative AI tools may be used to better understand the course material, or the concepts in a homework problem, but you should not use an AI tool for brainstorming or writing up your solution. In particular, you should not enter the homework problem itself as part of the prompt to an AI model. While working on your problem sets, you should not refer to existing solutions, whether from other students, past offerings of this course, materials available on the internet, or elsewhere. All sources of ideas, including the names of any collaborators (including AI tools, people outside of the course, and ARC tutors), must be listed on your submitted homework along with a brief description of how they influenced your work. You do not need to cite course material. If you use any concepts, terminology, or problem-solving approaches not covered in the course material by that point in the semester, you must describe the source of that idea. If you credit an AI tool for a particular idea, then you should also provide a primary source that corroborates it. Github Copilot and similar tools should be turned off when working on programming assignments.

[3] Based on text in the Harvard CS50 Syllabus.

In general, we expect all students to abide by the Harvard College Honor Code. We view us all (teaching staff and students) as engaged in a *shared mission* of learning and discovery, not an adversarial process. The assignments we give and the rules we set for them (such as the collaboration policy) are designed with the aim of maximizing what you take away from the course. We trust that you will follow these rules, as doing so will maximize your own learning (and thus performance on exams) and will maintain a positive educational environment for everyone in the class. We welcome and will solicit feedback from you about what more we can do to support your learning.

Content and Textbooks

The content covered in CS1200 this semester will be very similar to the previous offerings, with some minor changes. We will have the drafts of the textbook chapters available to you on Perusall. The textbook drafts, lectures, lecture notes, SREs, problem sets, and section notes will capture all of the content that you are expected to learn during the semester, but you may want to refer to the following textbooks for supplementary reading and practice exercises:

1. Background (to review or fill in material from CS 20 and CS32/CS 50)
    a. Harry Lewis and Rachel Zax. Essential Discrete Mathematics for Computer Science.
    b. CS50 OpenCourseWare and the Python documentation.
2. Algorithms (approx. the first two-thirds of the course).
    a. Tim Roughgarden. Algorithms Illuminated, especially Volumes I & II.
    b. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. Introduction to Algorithms, Fourth Edition.
3. Limits of Algorithms (approx. last third of the course)
    a. John MacCormick. What can be Computed?
    b. Michael Sipser. Introduction to the Theory of Computation.