# CS161 Summer 2026 Syllabus

All policies, etc., here are subject to change. Attend lecture to ensure you receive updates.

## Course Logistics

| | |
|---|---|
| Lectures: | MWF 10:30 AM – 12:15 PM, NVIDIA Auditorium |
| Prerequisites: | 106B or 106X; 103 or 103B; 109 or STATS 116 |
| Website: | `https://cs161.stanford.edu` |
| Office Hours: | Link to calendar on website |
| Instructor: | Matthew Sotoudeh, CoDa W326 |
| Email: | sotoudeh@stanford.edu |

## About the Course

After taking 106B, you've become quite an expert at taking real-world problems and writing programs to solve them. But not all programs are equal, even if they solve the same problem! Your program might be fast but use a lot of memory, while your friend's is slow but uses only a little memory. You might write an accounting program that performs wonderfully when you input your financial details, but runs for hours when you share it with your friend and they input their complicated startup stock options.

In 161, we'll explore questions like:

- What does it even mean for a program to solve a problem well, or better than another program? Can we even talk about such concepts in general, outside of specific inputs or computers?
- Can we predict how well a program will solve a problem before implementing and running it? Can we convince others that our approach to solving the problem is good?
- Once you have a problem you want to solve, how do you design a good program for solving it (whatever we decide good means)?
- Is it ever possible to know that we've truly found the best solution to a problem? Or must we always keep searching for better algorithms?

Believe it or not, the skills you learned in CS 103 and 109 are exactly the tools needed to attack these questions! Proofs will let us make guarantees about how well a program does on all possible inputs, even when there are so many that there's no way we could possibly test them all. We will even write proofs about every possible program solving a particular problem, which will help us prove that certain programs are "as good as it gets" (no other program does better). Probability will help us understand how the performance of a program changes depending on the distribution of inputs it will see in the real world.

As the cherry on top, you will get to see some of the jewels of computer science: the most beautiful and fundamental algorithms that form the absolute core of both the theory and practice of computers.

All in all, taking 161 will help you mature from "solved the problem" to "solved the problem *well*."

## Concrete Learning Goals

My learning goals for this course include:

- **Rigorously Formulate and Answer Critical Questions About Algorithms:** This is the big one. Given an algorithm and a question about it (how fast is it? how much memory does it use? does it always give the correct answer?) you will be able to rigorously formulate the question and answer it with very high levels of confidence.
- **Solve problems:** given a problem and setting, be able to design and implement an algorithm to solve that problem efficiently and effectively.
- **Communicate your thoughts:** convince others of the efficiency and efficacy of algorithms. Be able to converse comfortably with other computer scientists about "table stakes" concepts such as Dijkstra's algorithm, balanced trees, and quicksort.

## Textbooks

While the course lectures are the "source of truth" you are expected to use for this course, there are a number of very good algorithms textbooks available that you could use for extra practice. The very best introductory algorithms textbook, and the one I suggest you buy as a reference for this class, is:

- Robert Sedgewick and Kevin Wayne; *Algorithms* (SW)

Other good algorithms textbooks include:

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein; *Algorithms* (CLRS)
- Donald E. Knuth; *The Art of Computer Programming* (TAOCP)
- Jeff Erickson; *Algorithms* (JE)

If you already know what mergesort and red-black trees are, I would suggest picking up volume 3 (and maybe 1) of TAOCP. CLRS and TAOCP are available for free online from Stanford Libraries. JE is available for free online from the author.

## How to Succeed in CS 161

- **Attend Lectures!** Lectures will be where you get to see worked examples of the skills we want you to learn in this class, where you will get to practice thinking through problems with your classmates (and-soon-to-become-friends!), where you will have your understanding checked via regular assessments, and where we will inform you of the homework, assessments, etc., for the week.
- **Follow the Per-Lecture To-Do Lists!** At the start of every lecture I will pass out a "to-do" list for that lecture, which tells you exactly what you need to do before the next lecture. Follow it religiously, marking off tasks as you go!

## Grading

2026 is shaping up to become a year of great change in higher education, and 161 is no exception. This summer, we are going to experiment with a significantly more in-person, engaging course and grading scheme. This is the first quarter we're trying this, so we are going to make mistakes. Please be understanding and flexible as we work to get this right! What we know for sure is that **the majority of your grade will come from two in-class midterm exams and one final.** Beyond that, it depends on how you are enrolled:

1. In-person, letter-graded: Approximately 30-40% of the grade will come from 'regular assessments' that combine tasks including in-class quizzes, attendance, and one-on-one meetings with the CAs. You must pass both the exams and regular assessments to pass the class.
2. For CGOE (remote) students, the grade will be based on the three exams plus a small (about 10%) contribution from required online 'participation quizzes' graded for completion.
3. In-person credit/no-credit students will have the option to pick between the two grading schemes.

The exact mapping of grades to letters will be determined later in the quarter, once we have more data on how well the regular assessments are working. To ensure you get full credit, be sure to attend every lecture and follow the instructions on the to-do list passed out during every lecture.

## Course Policies, Expectations, and Guidelines

- Please, please, please ask questions! If you are confused about something, I guarantee that many others are as well. I understand that many students are concerned about slowing down lecture too much. **Don't be:** keeping lecture on track is my job. If we don't have time to answer your question on-the-spot, I will let you know, continue lecturing, and post the answer on Ed afterwards.

- Auditors are welcome and encouraged to ask questions during lectures, but due to the very limited course staff please understand that we cannot take OH or grading time away from enrolled students.
- SCPD students are encouraged to attend anything they want (lectures, OHs, exams) in person.
- Please do not use electronic devices during lecture. If you need paper, pencils, or pens, let me know and I would be happy to provide you with some. If you must use a laptop, tablet, or other electronic device during lecture, please consider sitting towards the back of the lecture hall to avoid distracting those behind you.
- Regrade requests are for when the grader made a factually incorrect comment, not for when you disagree with the amount of partial credit given to a wrong answer. Instructions for requesting regrade requests will be sent out with each exam. Regrade requests sent after the regrade request window closes, or otherwise not following the regrade request instructions, will not be considered.
- Students who arrive more than 10 minutes late to an exam without prior approval from the instructor, or continue writing after time, will be subject to a point deduction. If you have an accommodation allowing exam breaks, they must be taken within view of the proctors. Students needing to take unusually long bathroom breaks during exams should request an OAE letter indicating this.
- Late work is subject to a point deduction to be determined at the end of the quarter. Following the per-lecture to-do lists is your best way to avoid such deductions.
- The Stanford Honor Code and CS Honor Code apply to this class. For the exams, no assistance is permitted (including other people, websites, notes, chatbots) unless explicitly approved by the instructor. Please observe the copyright notice in the preamble section of the lecture notes. If you have a question about these policies, please contact the instructor.

These policies may be changed or extended throughout the quarter. Changes affecting the whole class will be announced in lecture. I reserve the right to make one-off exceptions to this syllabus for students on a case-by-case basis, but this occurs very rarely.

## Regular Assessment Recording Notice

Some of the regular assessment tasks you will need to complete involve oral examinations with a course staff member. We will video and/or audio record these sessions for course use only.

## SCPD Recording Notice

Stanford-written notice:

> Video cameras located in the back of the room will capture the instructor presentations in this course. For your convenience, you can access these recordings by logging into the course Canvas site. These recordings might be reused in other Stanford courses, viewed by other Stanford students, faculty, or staff, or used for other education and research purposes. Note that while the cameras are positioned with the intention of recording only the instructor, occasionally a part of your image or voice might be incidentally captured. If you have questions, please contact a member of the teaching team.

## Access and Accommodations

Stanford-written notice:

> Stanford is committed to providing equal educational opportunities for disabled students. Disabled students are a valued and essential part of the Stanford community. We welcome you to our class.
>
> If you experience disability, please register with the Office of Accessible Education (OAE). Professional staff will evaluate your needs, support appropriate and reasonable accommodations, and

> prepare an Academic Accommodation Letter for faculty. To get started, or to re-initiate services, please visit oae.stanford.edu.
>
> If you already have an Academic Accommodation Letter, we invite you to share your letter with us. Academic Accommodation Letters must be shared at the earliest possible opportunity so we may partner with you and OAE to identify any barriers to access and inclusion that might be encountered in your experience of this course.

## AIWG Pilot Program Notice

AIWG-written notice:

> This course is participating in the proctoring pilot overseen by the Academic Integrity Working Group (AIWG). The purpose of this pilot is to determine the efficacy of proctoring and develop effective practices for proctoring in-person exams at Stanford. To find more details on the pilot or the working group, please visit the AIWG's webpage.

You can view the AIWG student guide at `http://aiwg.stanford.edu/studentinfo`.

## Tentative Course Schedule

*Barring major typos (e.g., an exam landing on a holiday I didn't realize) the midterm dates on the below calendar are fixed and you should email me ASAP if you have a conflict with either of them.*

| Day | Topic(s) and/or deadlines |
|---|---|
| M, June 22nd | Introduction to CS 161. Selection sort. Best- and worst-case analyses. |
| W, June 24th | Big-O. Insertion sort. Average case analysis. |
| F, June 26th | Divide-and-conquer. Merge sort. Analyzing recurrences. |
| M, June 29th | Quicksort and quickselect. Expected time analysis. |
| W, July 1st | Heaps and heapsort. |
| F, July 3rd | HOLIDAY |
| M, July 6th | Comparison sorting lower bounds. Intro to dynamic sorting and search. BSTs and the fundamental BST operations: best, average, and worst-case analysis. |
| W, July 8th | AVL trees. |
| F, July 10th | **Midterm 1: Sorting** |
| M, July 13th | 2–3 trees. Red–black trees. |
| W, July 15th | Amortization. Dynamic arrays. Aggregate analysis. Potential method. |
| F, July 17th | Splay trees. Union-find. |
| M, July 20st | Hash tables. |
| W, July 22rd | (Catch-up) |
| F, July 24th | Overview of algorithm design patterns we've seen and those we will see. Algorithm design via fundamental graph algorithms: DFS and topological sort. |
| M, July 27th | Greedy algorithms: Prim, Kruskal, activity selection. |
| W, July 29th | (Catch-up) |
| F, July 31st | **Midterm 2: Searching (trees and hashing)** |
| M, August 3rd | Dynamic programming: Fibonacci and Bellman-Ford. |
| W, August 5th | More dynamic programming: Dijkstra (and BFS), Floyd-Warshall. |
| F, August 7th | Tarjan's strongly connected components algorithm. |
| M, August 10th | (Barely on final) Knuth-Morris-Pratt |
| W, August 12th | (Not on final) design and analysis of algorithms beyond CS 161. |
| **Sa, August 15th** | **Final Exam** |