

CS161: Design and Analysis of **Alligators** Algorithms

Summer 2022

Overview

How can we choose the best route to commute to work, or determine whether our favorite sports team can still make the playoffs, or efficiently remember which URLs are spammy? And how can we do such things correctly, efficiently, and confidently? This course introduces some powerful building blocks and techniques that will empower you to reason about (and solve!) these and many other problems. As a side bonus, the class should be useful preparation for technical interviews. As an even better side bonus, we will cover some truly beautiful algorithms and data structures, and I can't wait to share them with you!

Instructor: Ian Tullis, itullis@stanford.edu. (Please call me "Ian" rather than "Prof. Tullis" etc.!) I used to work at Google on Search and on Code Jam (an algorithms contest, which I still help out with), but I left industry a couple years ago because I want to become a CS lecturer. There's a bit more about me here: <https://itullis.people.stanford.edu/> (this site is currently down, will fix eventually) (and don't worry, no competitive programming experience is required for CS161!)

Wonderful TAs: Their bios are on the [course site](#)!

Golrokh Emami (golrokh@)
Rishu Garg (rishu@)
Ricky Grannis-Vu (rickygv@)
Ziang Liu (ziangliu@)
Ivan Villa-Renteria (ivillar@)
Lucas Xia (lxia12@)

When: MWF 1:30-3, 6/20 - 8/12.

Where: NVIDIA Auditorium (basement level of Huang Engineering Center).

Prerequisites: 103, 109, and (transitively) 106B, or their equivalents. There are [prereq review sheets](#) on some of the most important topics that you will need from these courses (e.g., induction, indicator random variables, binary search), so these are not hard prerequisites... but you should be prepared to put in some extra work if you lack one or more of them.

Units: 3-5. Undergrads and SCPD students *must* take the class for 5 units per department/SCPD policy – no exceptions, unfortunately! The course content and expectations are the same regardless of how many units you take it for.

OAE and other accommodations: If you require academic accommodations, or if there is anything we can do to help you personally succeed in the course or that we should be aware of, please email our OAE

coordinator (Ziang, ziangliu@stanford.edu) directly. If you have an OAE letter, please send it to us early on in the quarter so that we can best plan how to meet your needs.

Illness / Isolation: If you need to self-isolate because, e.g., you contract COVID or expect that you have been exposed, please let the staff know at your earliest convenience. Fortunately, all of the lectures are recorded, and all of the problem session docs will be uploaded to the site. But we understand that it may take some time to catch up – just let us know your situation and we will work with you. Please email Ian and Rishu (rishu@stanford.edu).

Auditors: If you would like to audit the class, please email Ian to be added to Canvas. Auditors are welcome to attend lecture in person (especially since NVIDIA Auditorium is huge); please use your best judgment. Auditors will also have access to lecture recordings, assignments, assignment solutions, and Ed. However, I ask that auditors not attend office hours / exams and not submit assignments for grading, since the staff bandwidth is limited. (Exception: Feel free to submit Pre-Homeworks and coding problems, since they are graded automatically.)

Course Structure

I'm a firm believer in learning by doing (and practicing!), and the structure of the class reflects this. Each of the six units consists of two lecture days (four mini-lectures), an optional (but strongly recommended) problem session day, a pre-homework, and a homework. There are two exams: a midterm and a final.

Lectures: Most class meetings will consist of two 40-minute minilectures (with a much-needed 10-minute break in between). Lectures will feature frequent opportunities for us to stop and think. The lectures will be recorded and available on Canvas.

The SCPD team has asked me to include the following, verbatim:

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.

Problem Sessions: After every two lecture meetings, we will have an optional problem session with more chances to practice the material. Since these sessions will mostly consist of individuals or groups working through problems at their own pace, they will not be recorded, but the questions (and detailed answers) will be posted. Although these questions will not be directly tested on exams, it is strongly recommended that you work through them for the extra practice.

Whenever there is a live Problem Session, there is also an evening version from 7:30-8:30 on Zoom, on the same day.

Problem Sessions are also office hours, and questions about homework, lecture, etc. are fine!

Pre-homeworks: These are sets of 15 multiple-choice problems designed to give you some practice with each unit's concepts before you tackle the tougher homework problems. They can be completed directly on Gradescope. You can make as many attempts as you want (there is no penalty!), and you get immediate feedback. You can tell that you got a question right when an explanation appears.

Homeworks: These are sets of 6 meatier problems; they can range from mathematical proofs to algorithm design and analysis. There will always be one coding problem that can be submitted on Gradescope, and for which you can get immediate feedback on correctness; we currently support C++ and Python. You do not necessarily need to solve all of the problems; see the grading breakdown below.

You may discuss homework questions with other students at a high level, but you must write up your own answers, and you must never look at another student's written work or code. See the Honor Code section below for more information. I require you to write up your own work because doing this reveals tricky aspects of a solution that you may otherwise miss, and prepares you better for exams. (Also, in classes I've taken in which students can submit a joint writeup, in practice it seems that people often divide the work and only think about their own parts. "Divide and conquer" *is* one of the strategies you will practice in CS161, but... not like this!)

Homework submissions must be typed. It's OK to include some pictures of handwritten equations, diagrams, etc. – for instance, you can take a picture with a phone and then put that in – but these must be neat enough for us to read.

We do not require you to typeset your homework, but we encourage you to do so. (It's good to learn LaTeX if you are planning to take additional courses in CS theory, AI, or math, for instance.) The Overleaf site makes it relatively easy to get the hang of it, and we provide templates that you can use if you'd like.

Regrade requests must be submitted on Gradescope within one week of homework being returned. You should not hesitate to submit these – sometimes graders make mistakes! However, in the absence of a mistake, we will not award additional points when, e.g. the grade was assigned consistently with our rubric / how we graded other students.

Due times and late days: Pre-homeworks and homeworks are due at 11:59 PM on the indicated days.

You also have 6 late days to use throughout the quarter. Each late day extends the deadline by 24 hours. You may use at most 2 late days on each assignment.

Work submitted after the deadline is penalized as follows:

- By 11:59 PM on the day after the due date: -20% of points earned (e.g., a 35 on a homework would become 28)
- By 11:59 PM on the day two days after the due date: -40% of points earned (e.g., a 35 on a homework would become 21)

Be aware that outside of exceptional circumstances (such as health or family emergencies) or OAE accommodations, we will not accept any work 48 hours past its deadline, or grant additional late days.

This is for a few reasons:

- Summer courses move fast, and we really *really* don't want you to fall too far behind by focusing on earlier work to the exclusion of newer topics.
- We also want to be able to release solutions and grade assignments relatively quickly.
- Between the late days and the additional coursework points available, there is already a lot of flexibility built into the system – you can in theory miss an entire homework and still get all 330 coursework points by doing everything else perfectly

Even if you think you will be using one or more late days, you might consider submitting what you have before the deadline and before the 24-hour mark, just in case. However, we will only grade your final submission.

You do not need to let us know when you are using late days. If you end up submitting something late even after using all your late days, we will recalculate the penalties in the way that hurts your grade the least. **Be aware that because of this, late penalties will not show up on Gradescope until the end of the quarter, so you are responsible for tracking your own late day usage.** If you have any doubts about where you stand regarding late days, please email Ian.

Exams: There will be two in-class exams: a midterm covering the first four units (with emphasis on the first three), and a cumulative final covering the entire class (but with a bit more weight on material covered since the midterm). You are responsible for understanding the content of the lectures, pre-homeworks, and homeworks (including any problems you skipped or did not solve – you must still read and digest the solutions). The problem sessions are truly optional – they will surely help to prepare you for exams, but there will not be exam problems directly based on problem session problems.

You should expect the exams to be challenging, since I want there to be room to demonstrate deep understanding and mastery, but they will not be gratuitously difficult (e.g., as in some classes that do this to create a sharp spread in grades).

You may bring up to 5 pages of handwritten or typed notes to the midterm, and 10 to the final; in both cases, "pages" means front and back. The limit is to encourage you to review and distill what you have learned. However, in general, the exams will test your accumulated understanding rather than the strength of your notes.

If you are in the Bay Area, I expect you to take the two exams in person. If you are an SCPD student who is taking the class remotely, you will take the exams with your chosen exam monitor, with the same length of time that students get in person, but within a 24-hour window on the day of the exam. (In extreme cases, a 48-hour window may be possible; please reach out to Ian.)

If you are not an SCPD student but wish to take one or both exams remotely, I expect you to meet the same standards as SCPD students: for each exam you must have an exam monitor (e.g. a coach, a

librarian, a manager – *not* a family member / friend) who can administer the exam and send me the results – see [this page](#) for what SCPD students need to do, although you should not use that exact process if you are not an SCPD student. (Here is a [form](#) for making these requests.)

Therefore, the class can be taken fully remotely even as a non-SCPD student, but you must take the monitoring requirements (and Honor Code) seriously and make arrangements well in advance of the exams.

Grading

Coursework: 330 points. (Additional points earned in this section beyond 330 go into the "extra credit" category below, but at 1/3 the value. This is to prevent what is supposed to be extra, optional work from carrying undue weight.)

90 pts 6 pre-homeworks x 15.

240 pts 6 homeworks x 40. (However, because there are six problems on each and they are each worth 8 points, up to 48 points are available per homework.)

Exams: 270 points.

90 pts Midterm

180 pts Final

Extra credit, capped at 24 points: From, e.g., extra homework problems, helping your peers on Ed, or other positive contributions to the course community.

There are no fixed grade cutoffs, although it is guaranteed that 90% and above will always be some form of A, and 80% and above will always be at least some form of B, etc. I will set grades based on performance and not on historical CS161 trends or on a curve; I encourage you to think of your classmates as colleagues rather than as competitors. I will also not incorporate extra credit until the grade cutoffs have been determined, so that it truly is extra and not something you *de facto* have to do. (Exception: A+ grades are given sparingly, per department guidelines, and might not be attainable without getting some extra credit.)

Incompletes: These are granted on a case-by-case basis for personal emergencies; please reach out to Ian directly. Be advised that all coursework completed prior to taking the Incomplete still carries over to when you complete the Incomplete (those assignments, exams, etc. cannot be replaced in that new quarter).

Honor Code Policies

Collaboration (on both pre-homeworks and homeworks):

- You can feel free to discuss the questions with other students in the class, including study groups.
- For pre-homeworks, you may even directly compare answers with each other as long as you explain your reasoning to each other.
- For homework problems, you may not directly share answers. It is fine to talk about high-level approaches and even to dip into some details. But you cannot, e.g., just say "did you get $O(n^2)$ for

question 4a?" Put another way, I am fine with you collaborating as long as you are teaching – and learning from – each other. Just checking answers is not that.

- ***You may never look at another student's written work or code.***
 - This includes, e.g., writing down someone else's solution from a study group and then referring to it later. Feel free to take notes based on your discussions, but you should *not* be arriving at complete written solutions together.
 - It is OK to discuss small snippets of work in office hours, i.e., you can share an equation. We will let you know if too much is being shared. If you want to discuss code, you should attend a 1:1 session or otherwise meet privately with one of us.
 - When using Ed (the course forum), please make your post private if it contains written work or code or would otherwise spoil significant solution details for other students.
- You are always welcome to ask for help from the course staff, e.g., in office hours and on Ed. You do not need to cite us. Otherwise, for homework problems, whenever you work with or otherwise receive significant help from a fellow student, or from any other source, you must acknowledge them by name in your submission.

In addition to the collaboration policies above, you must abide by the following rules:

- You may never post pre-homework or homework problems online or otherwise solicit help (e.g., from forums such as StackOverflow).
- For homework problems, you may not search for solutions online. Past CS161 solution materials are specifically forbidden, but the same also goes for, e.g., assignments from other algorithms classes.
 - It is fine to search for background information that will help you solve the problems on your own. However, a problem may ask you to e.g. not look up the running time details of an algorithm, in which case you must not do that. You should also not perform searches that might reasonably turn up an answer to a homework problem.
 - If you *accidentally* run across something that gives away part or all of a solution, please let us know in your submission (and don't worry – this will not impact your grade and is not considered an Honor Code violation). (Of course, do not intentionally fish for happy accidents!)
- For coding problems in particular, you may not include code that you did not write. It is fine to, e.g., adapt an algorithm from class, but your work must ultimately be your own.
 - Using library functions (e.g. Python's `.sort()`) is fine unless the problem asks you not to.
- For pre-homework problems, you may search however you like.

Official violations of the Honor Code (as determined by the Office of Community Standards) will result in – at a bare minimum – a zero on the problem(s) in question and at least a one-step grade deduction in your overall course grade (e.g. A- to B+), even if the lost points from the problem(s) would not have dropped the grade that far.

We will be using plagiarism detection software, but it is ultimately open to interpretation by the staff – no decision will be made purely by an algorithm! We understand when, e.g., two pieces of code might look similar by chance.

We do not wish to create a climate in which you have to keep worrying about whether some action might violate the course policies. We think it will be clear when something is over the line; if you are unsure, though, please ask Ian. And if you find yourself in a tight spot, and you are tempted to do something that seems like it would violate the Honor Code, please reach out to Ian to discuss options instead of doing something regrettable.

Please also see:

- The Stanford Honor Code:
<https://communitystandards.stanford.edu/policies-and-guidance/honor-code>
- The Stanford Honor Code (as it pertains to CS courses):
<https://web.stanford.edu/class/archive/cs/cs106b/cs106b.1164/handouts/honor-code.pdf>

Resources

Course Site: cs161.stanford.edu. Links to the resources below will be provided there; enrolled students will be automatically added to all sites.

Gradescope is where you submit pre-homeworks and homeworks, and where graded exams are returned to you.

Ed (AKA "Edstem") is our course forum. We will make any critical course announcements here (in pinned posts), and also post homeworks and homework solutions here. You are otherwise not required to keep up with the content on Ed, but it can be very useful to learn from other students' questions. If you are active and helpful on Ed, you will earn some extra credit for it. (This will be determined by course staff, on the basis of how frequent, helpful, and thorough your answers have been. It is not based simply on a statistic like the number of posts.)

Videos of lectures and problem sessions will be posted on **Canvas**. We won't use it for anything else apart from a roster (so you can see who else is in the course, e.g., for the purpose of finding study partners...) and a calendar.

Office hours are held in person or over Zoom. The canonical calendar of office hours is on Canvas. We will have opportunities for both 1:1 slots and group discussions.

Textbooks: There is no required reading, but I recommend Tim Roughgarden's Algorithms Illuminated, volumes 1-3. These are inexpensive, well-written works by a former Stanford prof who taught 161. Some copies will be available at the Stanford Bookstore.

Rec letters: I am happy to write recommendation letters based on this class, with a couple caveats:

- I am a Master's student and not a faculty member, so a letter from me may carry less weight
- Letters from "core" classes may in general carry less weight

- If the letter is for grad school, be advised that letters based on performance in a class are generally viewed as "low-signal" unless they include evidence of a student's potential for research (e.g., you came up with some novel solution to a homework problem and explored it)

If you intend to ask for a letter, please make it easy for me and the TAs to get to know you throughout the quarter, so that I have specifics to write about. A letter like "Edsger did well on all the homeworks and exams and got an A" conveys no information beyond the grade on your transcript.

The Algorators

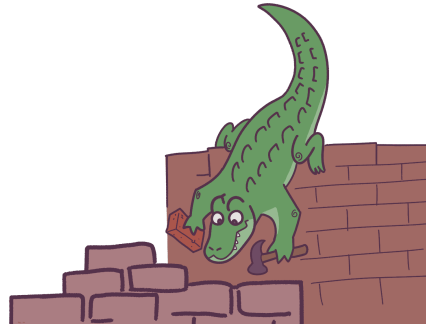
There are five alligator mascots for this class; they are mostly for fun, but they also illustrate different perspectives on studying and using algorithms. They will make appearances in lectures, problems, etc.



BRUTUS is the brute-force gator. Brutus is stronk. Brutus is in no danger of overthinking problems. Brutus is often in danger of underthinking problems.



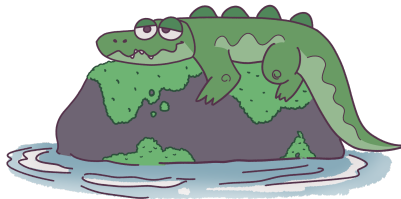
INDY is the industry gator. "When would you ever use red/black trees?", he says, as he uses libraries based on red/black trees and asks candidates interview questions about red/black trees. Then he speeds home in his expensive car and rolls on his piles of quantum coins or whatever.



SISI is the systems gator. She is practical and cares more about implementations and speed than about abstract performance guarantees.



TERRY is the theory / academic gator. They are passionate about proofs and fine details, sometimes to the point of exasperating the other Algorators.



WAVERLY is the intuitive (some might say "handwavy") gator. She has a big-picture, intuitive understanding and does not like to get bogged down with extreme rigor.

(Art by Isaac Hernández. I have no drawing talent of my own!)

Schedule

There was one here, but it has been moved to the Schedule tab of cs161.stanford.edu, just to avoid having two copies that can get out of sync!