Course Information

Course Overview

Data structures are absolutely essential to modern software, yet aside from a few fundamental data structures (dynamic arrays, linked lists, binary heaps, binary search trees, and hash tables) aren't typically covered as part of a CS curriculum. This course explores both the theoretical and practical aspects of data structures and data structure design. We'll explore a wide array of data structures designed to solve all sorts of problems, from classical problems like finding minimum spanning trees to newer problems like dynamic graph connectivity and frequency estimation. Additionally, we'll explore a variety of different techniques used to design and analyze data structures. By the time you've finished this course, you should have a much better understanding of both the theoretical and practical techniques that power data structures. Along the way, you'll sharpen your theory and coding skills and learn a bunch of useful problem-solving techniques.

Instructor

Keith Schwarz (htiek@cs.stanford.edu)
Office: Gates 178
Office Phone: (650) 723-4350

TAs

Kyle Brogle (broglek@stanford.edu)
Daniel Hollingshead (dhollingshead@stanford.edu)
Nick Isaacs (nisaacs@stanford.edu)
Aparna Krishnan (aparnak@stanford.edu)
Sen Wu (senwu@stanford.edu)

Email

The course staff can be reached at cs166-spr1314-staff@lists.stanford.edu. Please don't hesitate to send us emails! We're here because we genuinely love this material and want to share it with you. If you have any questions on the material, or if you're interested in exploring more advanced content, please get in touch with us. We'd be happy to help out.

This is a brand-new course and we're hoping to make it as good as it can be. If you have any suggestions on how to improve the course, please feel free to let us know! We will take your feedback seriously.

Lectures

Mondays and Wednesdays from 2:15PM – 3:30PM in Hewlett 201. Lectures will not be recorded, so you're encouraged to attend!

Units

CS166 is offered for either three or four units. Undergraduates are required to enroll for four units, while graduate students can enroll for either three or four units. The course content and requirements are the same in the three-unit and four-unit versions of the course and the unit flexibility is purely to help graduate students stay under unit limits.

Website

The course website is http://cs166.stanford.edu and it's loaded with resources for this course. There, you'll find all the course handouts, the syllabus, links to readings, and all sorts of other resources.

Office Hours

The course staff will hold office hours throughout the week. We'll announce the schedule and post it to the course website during the first week of class.

Prerequisites

The prerequisites for this course are CS161 and CS107.

From CS161, we expect you to feel comfortable designing and analyzing nontrivial algorithms and writing proofs of correctness. Mathematically, you should be comfortable using asymptotic notation (o, O, Θ, Ω, and ω), solving recurrence relations, manipulating inequalities, simplifying summations, and working with probabilities. We'll also expect that you're comfortable using the divide-and-conquer, greedy, and dynamic programming techniques; that you're familiar with randomized algorithms; and that you're comfortable writing correctness proofs for algorithms of each of these types. You should also feel comfortable with standard algorithms like Dijkstra's algorithm, Prim's algorithm, quicksort, etc.

From CS107, we expect that you're comfortable writing and testing nontrivial programs and working from the command line. You should also feel comfortable with binary representations of numbers. We'll expect that you've at least heard of the memory hierarchy and are comfortable with the idea that not all memory accesses take the same amount of time.

If you're unsure whether CS166 is the right place for you, please feel free to get in touch with the course staff.

Readings

The main reading for this course is *Introduction to Algorithms, Third Edition* by Cormen, Leiserson, Rivest, and Stein. This is an excellent textbook to have on-hand if you're doing anything related to algorithms and data structures, and we hope you find it useful. As a note – you will want to use the Third Edition of CLRS in this class, since at least one of the data structures we'll explore this course (van Emde Boas trees) is not covered in the second edition. There are three copies of this book on reserve in the Engineering Library.

Additionally, there will be a variety of readings posted online (research papers, course notes, lecture slides, external articles, etc.) Check the website for details on the readings for each lecture. I will try to present the salient features of each data structure in lecture, so depending on your learning style, you may find it useful to do the readings right before or right after lecture.

Assignments

Over the course of the quarter, there will be seven problem sets. These problem sets will contain a mixture of theoretical questions and coding questions that are relevant to the week's material. You may either work on the problem sets individually or in pairs. If you work in a pair, you will turn in a joint problem set submission and both members of the pair will receive the same grade. If you choose to work individually on an assignment, we will grade your problem set on a slightly easier scale. More details are in an upcoming handout.

To make sure that everyone is following along with the material, there are no late days in CS166. Assignments are due at the times indicated on the problem set and no late submissions will be accepted. If you have a medical or family emergency, or if some other circumstances will prevent you from submitting a problem set on time, please contact the course staff at least 24 hours in advance of the assignment due date and request an extension.

Midterm

There will be a midterm exam on **Wednesday, May 21**, location and time TBA. The exam will be cumulative and serve as a review of all the topics we've covered to help you prepare for the final project.

We'll release more information about the midterm early in the quarter and will offer at least one alternate exam time.

Final Project

At the end of the quarter, you will be asked to complete a final project in which you'll research one or more advanced data structures and present your findings. You should expect to start working on the final project right after the midterm exam. We will release more information about the final project as we draw closer to the end of the quarter.

Grading

Overall, your grade for this course will be determined as

Assignments:	50%
Midterm:	25%
Final Project:	25%

Enrollment Cap

You might have noticed on Axess that enrolling in CS166 requires "consent of instructor." Because this is a new course, we're capping this class at 100 students. If enrollment goes over 100, we'll send out an application and details on how to apply. In order for us to get an approximate headcount, if you're seriously considering this course, we'd appreciate it if you enrolled on Axess.

Incompletes

If you have a serious medical or family emergency and cannot complete the work in this course, you may contact Keith to arrange for an incomplete. Our policy is to reserve incompletes only for emergencies, so we do not grant incomplete grades for poor performance on the assignments or exams, nor do we offer incompletes for busy work schedules.

In order to receive an incomplete in the course, you must have completed all assignments in the course at the time at which you request an incomplete, except for possibly the most-recently-due assignment.