Course Information

Course Overview

Data structures are ubiquitous in 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

Benjamin Plaut
Mitchell Douglass
Rafael Musa
Sam Redmond

Email

The course staff can be reached at cs166-spr1718-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.

Lectures

Tuesdays and Thursdays from 3:00PM – 4:20PM in Herrin T175. 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

We'll be holding a few sets of office hours each week. We'll post a calendar to the course website once they're set up.

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 related concepts like universal families of hash functions); 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. Additionally, we expect that you'll be comfortable writing code in lower-level languages like C and higher-level languages like Java.

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 recommended 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 copies of this book on reserve in the Engineering Library.

Additionally, there will be a variety of readings posted online (papers, course notes, slides, 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 six 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. More details are in an upcoming handout.

Everyone has ***two*** free late days they can use on the assignments. Each late day is an automatic 24-hour extension on the assignment. If you're working in a pair and submit late, both members of the pair will be charged the late day. If you submit past the due date and are out of late days, we'll apply a flat 30% penalty to your submission (so it's better to submit late than never). No work will be accepted more than 48 hours past the initial assignment deadlines; this will enable us to release solutions as soon as possible.

Midterm

There will be a midterm exam on Tuesday, May 29th from 7:00PM – 10:00PM, location TBA. The exam will be cumulative and serve as a review of all the topics we've covered.

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 in around the eighth week of the quarter. We will release more information about the final project as we draw closer to the end of the quarter.

Grading

The problem sets, midterm, and final project are each worth one-third of your total grade in this course. We do not curve individual assignment or exam scores, so your grade will be determined using the raw scores you receive.

Unlike some other courses, we do not drop the lowest problem set or midterm score and do not offer any make-up work. Your raw score will be computed purely by weighting your raw scores by the above amounts, and we will determine final grades based on those raw scores.

Incompletes

If you have a medical or family emergency and cannot complete the work in this course, you may contact Keith (not the TAs) to request an incomplete. We 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 be eligible for an incomplete, you must have completed all of the assignments (except possibly the most-recently-due assignment) and must have a performance that is roughly on par with a passing grade, as determined by the course instructor.

All incompletes are worked out on a case-by-case basis, and the instructor retains final discretion to approve or reject any requests for an incomplete.