CS 170: Efficient Algorithms and Intractable Problems

Course Overview

Prof. Luca Trevisan

1 Introduction

Whereas CS 61B was a bare introduction to the theory of computer science, CS 170 is a full exploration of it. In CS 170, you will study the design and analysis of graph algorithms, greedy algorithms, dynamic programming, randomized algorithms, linear programming, fast matrix multiplication, Fourier transforms, complexity, and NP-completeness.

Please read this document carefully. It contains answers to most of the questions that students ask during the first few weeks of class. The subjects include: how to contact the staff, prerequisites, textbooks, grading, late penalties, and policies on academic misconduct. There is a class Web page at `http://www.cs.berkeley.edu/~luca/cs170`.

A list of discussion sections and the TAs who run them is linked from the Web page. A tentative syllabus, which includes reading assignments, exam dates, and due dates, is also available. Please check the class Web page at the beginning of the semester and regularly throughout to learn about new information and syllabus changes.

If you have a general question about something *not* covered herein, the best option is to post a message in the `ucb.class.cs170` newsgroup. We (the instructors and TAs) will check the newsgroup regularly, and other students will be able to help you too. Other students will also be able to benefit from the answers. If you don't want to make your question public, you may send email to `cs170@cory.eecs`. Your email will be forwarded to both instructors and all the TAs. Do not send email directly to the instructor and/or your TA to ask questions or clarifications. If you wish to talk with one of us individually, you are welcome to come to our office hours, posted on our doors and linked from the Web page. If the office hours are not convenient, you may make an appointment with any of us by email. There are about 60 of you to every one of us, so please reserve email for the questions you can't get answered in office hours, in discussion sections, or through the class newsgroup.

Instructor:
Prof. Luca Trevisan
Office: 615 Soda Hall
Phone: 642-8006
Email: luca@cs

Course Secretary:
Jenny Gonzalez
Office: 395 Soda Hall
Phone: 642-9044
Email: csoffice@cs

Teaching Assistants:

- Scott Aaronson
  Office: 595 Soda Hall
  Phone: 643-4006

Email: aaronson@cs

- Shyam Lakshmin
  Office: 594 Soda Hall
  Phone: 642-5422
  Email: lakshmin@cs

- Iordanis Kerenidis
  Office: 587 Soda Hall
  Phone: 642-3560
  Email jkeren@cs

- Joe Polastre
  Office: 464 Soda Hall
  Phone: 642-9076
  Email: polastre@eecs

- Beini Zhou
  Office: 592 Soda Hall
  Phone: 642-5422
  Email: beini@cs

Lectures are Tuesdays and Thursdays from 3:30 pm to 5:00 pm in 10 Evans Hall.

2 Prerequisites

The prerequisites for CS 170 are CS 61B and one of Mathematics 55 or CS 70.

If you have not satisfied all the prerequisites, but you have taken a course you feel is very similar to CS 61B or Math 55, fill out an appeal form in the main CS office, 393 Soda Hall. The instructor does not handle appeals, so please do not attempt to lobby him for admission to the course. If you try to stay in the course when you have not satisfied the prerequisites, you will receive an F as your final grade.

You should be comfortable with mathematical induction, big-O notation, sorting algorithms, basic data structures, and binary heaps. In particular, if you are a transfer student and did not obtain a thorough understanding of binary heaps from CS 61B or a similar course, you should read Chapter 7 of Cormen, Leiserson, and Rivest in preparation for this course.

You will be using C or C++ to program one project in this course, and you should have prior knowledge of one of these languages.

If you are not familiar with the Unix operating system and basic tools, it is important that you learn. Some student groups, including CSUA, teach help sessions on Unix. See the CSUA Web pages.

3 Readings

There is one required class reader for CS 170, and one required textbook universally known to computer scientists as "CLR":

- Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, Introduction to Algorithms, MIT Press, 1990. ISBN 0-262-53091-0.

A second edition ("CLRS"), coauthored also by Cliff Stein, has been published a couple of weeks ago. While CLR is the "official" textbook, it's ok to buy CLRS instead. Reading assignments will be given with references to both textbooks.

The text should be available at the ASUC bookstore or across Bancroft at either Ned's or the Campus Textbook Exchange. The instructor's lecture notes will be made available on the web, and they will typically be posted a week in advance.

The instructor and TAs will post announcements, clarifications, hints, and other information in the `ucb.class.cs170` newsgroup, which you should read regularly whether you post questions to it or not, so that you're not the last to find out about major changes to assignments and midterm dates. Send questions or information of general class-related interest to this newsgroup using the Netscape Web browser, which has facilities for reading and posting messages.

4 Enrollment

Make sure that you are enrolled in a discussion section that has space for you. To find a section that has space, see the online class schedule for discussion section enrollment levels.

If you are on the waiting list for the course, the reason is that you are waiting to be admitted to a discussion section that is currently full. You need to choose a section that is not full if you want to take the class. If all sections are full, it is because we do not yet have enough TA support to add every student to the class. We will try to resolve by the end of the first week whether or not every student with the prerequisites will be admitted to the course. We cannot let you into sections that are full, so if you insist on waiting for a full section to have space, you will most likely never be admitted to the course. If you wish to switch sections, there must be room for you in the section you want to switch to.

If you are something other than a regular Berkeley undergraduate, then you probably need a signature on a form admitting you to the course. We cannot promise to admit those of you who are not regular Berkeley students. In particular, we will not sign any concurrent enrollment or UC Extension forms until after the *second* week of classes.

CS and EECS majors should already have named accounts for the lab machines from Instructional Facilities. Lab machines are in 271, 273, 275, and 277 Soda, and in 199 Cory. The Instructional Facilities staff may be found in 333 Soda (which is typically staffed from 8 am to 6 pm).

It is important that you make sure your instructional account is registered in the CS 170 grade book during the first two weeks; we use the registration information along with the TeleBears roster to determine who will receive grades in the class. To ensure you are registered, go to the URL `https://saidar.eecs.berkeley.edu/~register`, enter the login and password of your instructional account, and follow any further instructions. Of course, you can only do this if you have already obtained an instructional account.

5 Discussion Sections

In addition to the lectures, you will attend a discussion section for one hour each week. The discussion sections meet between the Tuesday and Thursday lectures. Discussions sections are not mandatory, and nothing done in section will directly affect your grade. On the other hand, discussion sections are your best opportunity to ask questions and learn interactively, and some examples worked out in section will be helpful on the graded assignments. Your section TA will return your assignments and midterms, and help with grading corrections.

You may attend a section other than that for which you are registered only if the TA of the section you are attending agrees to it.

Outside of your discussion section, You should feel free to attend any of the staff office hours (not just your own TA's) and ask any of us for help.

6 Course Work and Grading

6.1 Exams

There will be two midterm exams, each worth 15% of your final grade, and a final exam worth 30%. The two midterms are scheduled to be held in lecture on Thursday, September 27 and Thursday, November 15. The final exam will take place Wednesday, December 19, from 12:30 pm to 3:30 pm, in a location to be announced by the University later in the semester. You are required to bring a student photo ID and a blue-covered exam book to the exams. Exams are closed book, but you are allowed to bring one 8 1/2"-by-11" sheet of paper containing any notes of your choice (you can write on both sides of the sheet). The test coverage is cumulative, so material from the beginning of the course may be tested in either midterm or the final exam. All exams will be graded by the TAs and professors.

If you miss a midterm, you will be assigned a percentage score for that exam equal to the percentage score you earn on the final exam. (There will be no make-up midterms.) If you miss the final exam, you will receive a grade of F in the class unless you missed it because of a circumstance beyond your control, documented by a physician or equivalent authority, or if it conflicts with another scheduled University of California exam. If we decide to forgive your missing the final exam, you will receive an Incomplete grade and have the opportunity to make it up when we choose to give you an alternate exam, quite possibly at the end of the following semester.

A course grade of Incomplete will be granted only for dire medical or personal emergencies that cause you to miss the final, and only if your work up to that point has been satisfactory.

6.2 Assignments

In addition to exams, there are two types of assignments: homeworks and a project. Homeworks (roughly one per week) are handwritten and involve algorithm analysis and design. Homeworks are worth 35% of your final score, and each homework is equally weighted. Written homeworks may be submitted to Jenny Gonzalez in the CS main office, or to the CS 170 box in 283 Soda. Since we expect you to have conflicts from time to time during the semester, we will ignore your three lowest homework grades. There is one programming project, which you will submit electronically. It is worth 5% of your final score. Homeworks and projects must be done individually, are due Mondays at 4 pm, and will be graded by one of the class readers.

You may program the project at home, in the Soda Hall labs (271, 273, 275, and 277), or in 199 Cory Hall. The Soda labs are open from 7:00 am to 6:30 pm Monday through Friday. Outside these hours, the doors to the building and lab are locked. You will need to obtain a card key for after-hours access.

Everything you turn in for grading must show your name and your discussion section number. *You might receive no credit for assignments that are turned in without this information.* The project (which are submitted electronically) should also show your account name (login ID).

6.3 Grades

If you believe we have misgraded a midterm exam question or project, return it to your section TA with a *written note on a separate piece of paper* explaining the problem. If you're requesting an exam regrade, please staple this paper to the *front* of the exam. The entire exam or project will be regraded, so be sure to check the solutions to confirm that your final grade will go up after regrading. All requests for regrades and recording corrections must be made within two weeks after you receive the graded assignment or exam. By University policy, final exams may *not* be regraded.

Because of the difficulty of evaluating students in a course where much of the homework is based on proofs, the grading scale (conversion of scores to letter grades) will not be established until the end of the semester.

6.4 Lateness

We will give no credit for written homework turned in after the deadline, so that we can make online solutions available promptly and so that you can discuss those solutions in your discussion sections. Please do not ask for extensions for homework–the three lowest homework grades are dropped, so you can safely miss three. However, the material in this class can only be learned by doing lots of problems, so the homework is very important. Late homework will not be accepted for any reason whatsoever, even emergencies.

We do allow the programming project to be turned in late, but there is a penalty. If the project is N hours late, we'll reduce your earned score by ceiling(N) percent. While this gives you some leeway for putting the final touches on the project, do not stretch the deadlines too far. A project that is one day late will lose 24% of your earned score. After three days, even a perfect solution won't earn a passing grade.

7 Policy on Collaboration and Cheating

Cheating on a homework or project will earn you the maximum negative grade on that assignment. For example, if you cheat on a project worth 5% of your final grade, your grade on that project will be -5%. Cheating on an exam, or cheating twice in any way, will earn you an F in the course. We reserve the right to assign an F in the course to anyone who cheats on a project once, though we will probably not exercise this right. All incidents of cheating will be reported to the Office of Student Conduct, who will maintain records of your academic misconduct throughout your undergraduate career.

We encourage you to help each other learn the material by discussing the work *before* you do each assignment. For most assignments, explaining the meaning of a question or a way of approaching a solution is an interaction that we encourage. On the other hand, you should **never** read another student's solution or partial solution, nor have it in your possession, either electronically or on paper. You should write your homework strictly by yourself. For some problems, we may instruct you not to discuss the problem with other students at all. If you receive a significant idea from someone in the class, explicitly acknowledge that person in your solution. Not only is this a good scholarly conduct, it also protects you from accusations of theft of your colleagues' ideas.

Presenting another person's work as your own constitutes cheating, whether that person is a friend, an unknown student in this class or a previous semester's class, or an anonymous person on the Web who happens to have solved the problem you've been asked to solve. Everything you turn in must be your own doing, and it is your responsibility to make it clear to the graders that it really is your own work. The following activities are specifically forbidden in all graded course work:

- Possession (or theft) of another student's solution or partial solution in any form (electronic, handwritten, or printed).
- Giving a solution or partial solution to another student, even with the explicit understanding that it will not be copied.
- Working together to develop a single solution and then turning in copies (or modified versions) of that solution under multiple names.

Cheating on the project will be policed by advanced cheating-detection software (no, it's not a joke). If you share code with another student, you will be caught, even if you take steps to hide your cheating. ¡p¿In our experience, nobody begins the semester with the intention of cheating. Students who cheat do so because they fall behind gradually and then panic. Some students get into this situation because they are afraid of an unpleasant conversation with a professor if they admit to not understanding something. We would much rather deal with your misunderstanding early than deal with its consequences later. Even if you are convinced that you are the only person in the class that doesn't understand the material, and that it is entirely your fault for having fallen behind, please overcome your feeling of guilt and ask for help as soon as you need it. Remember that the other students in the class are working under similar constraints–they are taking multiple classes and are often holding down outside employment.

Mail inquiries to `cs170@cory.eecs.berkeley.edu`.