Course Information

Meets: Tu and Th, 5:00–6:20PM in WLH 2005.

Instructor: Mihir Bellare

Instructor's Office: AP&M 4230

Instructor's E-mail: mihir@cs.ucsd.edu

Instructor's web page: `http://www-cse.ucsd.edu/users/mihir/`

Course Web Page: `http://www-cse.ucsd.edu/users/mihir/cse200/`. This is the *only* source of all course related handouts such as problem sets, problem set solutions, exam solutions, and notes. Hardcopies of these items will not be distributed.

Course mailing list: The course mailing list is used by the instructor to inform students about any adminstrative updates (eg. where and when they can pick up graded homeworks or exams) or updates or corrections to assignments, and all students should be on this list. If you are taking this course, please add yourself to this list via `https://www.cse.ucsd.edu/mailman/listinfo/cse200`.

Office hour and meetings: My office hour is Tu 1–2PM. If you cannot make that time, you can send me email, with several options of times or intervals that work for you, so that we can schedule an appointment.

TA: The course has a TA whose only responsibility is grading. (The TA grades homeworks, and might grade parts of exams.) Unfortunately, the TA does not have office hours and is not available to see students. (The department only allocates a 25% TA for this course, and grading takes up all that.) For questions about graded homeworks, refer to the instructor, who will either resolve it directly or in consultation with the TA.

Relation to CSE requirements: CSE 200 is a required ccourse for the Computer Science Ph.D program. It is *not* required for Computer Engineering Ph.D students, and it is *not* required for MS students.

Contents: This course is an introduction to computational complexity theory, a branch of the theory of computation whose goal is to classify problems according to the resources (computation time or memory) required to solve them, and attempt to derive an understanding of which computational problems are "easy," which are "hard," and how different problems relate to each other. We shall first review concepts such as Turing machines, decidability, and undecidability. Then we will cover time complexity, space complexity, hierarchy theorems, and various complexity classes, including **P**, **NP**, and **PSPACE**. We will then do reductions and **NP**-completeness. Finally we will look at randomized algorithms and the classes **RP**, **BPP**. Further topics as time permits: probabilistically checkable proof systems, approximation algorithms.

Text: Michael Sipser, *Introduction to the Theory of Computation*, PWS 1997. Available at UCSD bookstore. Relevant chapters are 4,5,7,8,9,10, but not everything in these chapters is covered.

Pre-requisites: An undergraduate introduction to the theory of computation that has covered Chapters 0–5 of the above-mentioned Sipser text, or equivalent, meaning the following topics: regular languages, context-free languages, Turing machines, decidability and undecidability. (At UCSD, this corresponds to course CSE 105.) Also an introduction to algorithms (CSE 101 at UCSD) and Discrete Mathematics (CSE 20,21 at UCSD).

Requirements and grade computation: The course has some number of problem sets (also called homeworks), a midterm exam, and a final exam. Your grade is based on your *course score*. This is:

$$49 \cdot HS/HT + 0.17 \cdot MS + 0.34 \cdot FS$$

where

- HS is the sum, across all homeworks, of the score you got on that homework, and HT is the sum, across all homeworks, of the maximum possible number of points for that homework. (This maximum, which may vary from homework to homework, is indicated on the homework when it is handed out.) Thus homeworks are 49% of the grade, but homeworks are not all equally weighted.
- MS is your midterm exam score, out of 100
- FS is your final exam score, out of 100

It will not be possible for me to say, publicly, to what grade a certain score on the midterm corresponds, for, as you can see, the grade is a function of the total course score, and the mapping of that to a grade is not determined until the end of the course. If you are concerned about how you are doing, however, you are encouraged to meet with me personally and I will try to give you some sense of where you stand.

Exam calendar: The midterm exam is on Thursday February 19th, in the lecture room and in the lecture time slot. Final exam is on Monday March 15, 7–10PM, again in the lecture room.

Rules and grading policies: The exams are in class, closed book and notes, no calculators. "Blue-books" are not required: you are to write on the exam itself. However, bring scratch paper.

There are no makeup exams under any circumstances. If you do not take the midterm you get a zero on it unless your absence is due to a demonstrated personal health problem at the time, in which case the weight of the midterm will be shifted to the final. If there is any anticipable reason for which you cannot take the final exam at the scheduled time, don't take the course. If you do not take the final, you get a zero on it.

Each student must do their homeworks on their own, without any help from other students and without consulting any sources other than their own course notes, course handouts and the course text. In particular, you are not allowed to use any material from previous years of this course and you are not allowed to use the Internet. However, students are allowed to study together for exams.

Solutions to homework or exam problems should use without proof only results from class or class notes, and results of previous homework or exam problems. You may not use other results without proof, even if they are in the text book. If you are doubtful about what can or cannot be used without proof, ask me.

Problem sets are due in class on the day indicated on the problem set. Late problem sets are not accepted. Please do not turn in problem sets at any place other than in class. (If you can't make it to class, give it to someone else to turn in for you.) Re-grade requests on any problem set or exam are only accepted until two weeks after the graded object in question has been returned.

If your problem-set solution has more than one sheet of paper, the sheets should be stapled together, not clipped or folded at the corner. Turn in neat, readable solutions. (Either handwritten or typeset.) Points can be deducted otherwise.

Academic honesty: Cheating, including failure to abide by the above course rules, is taken very seriously. Academic dishonesty cases are prosecuted in conjunction with the Dean of Student Affairs and can result in probation or dismissal. Students have been caught cheating in this course, and other graduate courses in this department, in the past, and have been so prosecuted.

Some particular actions that we have seen attempted and warn against are the following. Sometimes students modify their graded exams or homeworks after they are returned, and then bring them back to us for regrades, thinking we won't notice that they have been modified. You would be surprised how well a grader can remember a single answer across several hundred. Students have been caught doing this and reported. Don't modify your exam or homeworks after they are handed back. Don't copy from others during exams or bring in un-allowed materials. Don't share electronic or other versions of your homework solutions with other students. Don't use the Internet to try to find solutions to homework problems. Don't use solutions from previous years of this course, something that will be quite tempting to those of you who may have dropped out of this course in earlier years and still have old materials in your possession.

Mathematical writing: This course involves mathematical abstraction and proofs. Being able to deal with these is one of the important things to learn. In both problems sets and exams, you will be graded based on the correctness, clarity and accuracy of mathematical exposition. Make sure what you write makes sense. Define notation before you use it; distinguish between different types of objects like machines, languages and strings. Answers that "don't make sense" will not get much credit. Your solutions should have a logical flow from beginning to end. Remember, you are graded on what you write, not on what you think you "meant." So make sure you write what you mean.

Mathematics is a language. Learn its grammar and semantics, and get used to using it correctly. Like any language, its goal is communication, and when properly used, it is a precise and unambiguous tool to this end. When you mis-use the language, you will not be understood, and you will loose points.

Write top to bottom, left to right on the page. Don't scatter information all over. Be as concise as possible.

Read through whatever you write before turning it in. Try to make sure there is an argument with a clear flow. If you paper says lots of different things, you are *not* going to get points just because one of them is right; indeed, you will get *less* points for a jumble which sort of includes something right than for something clear even if not the entire answer.

For problem sets, first write a rough draft, then write a new, final draft to actually turn in. Think about it from the point of view of a grader: how are you making sure that person will understand?

Spend time on the writeup. Re-read it after it is written, trying to be in the mindset of someone who does not know what you are thinking, but only has your solution to look at. Try to make sure it can be understood by such a person.

The articles under *Mathematical and technical writing* available via `http://www-cse.ucsd.edu/users/mihir/education.html` provide more information about mathematical exposition.