University of California at San Diego
Computer Science and Engineering
Course: CSE 200
Handout 1: Course Information
Instructor: Russell Impagliazzo — Winter, 2019

CSE 200: COMPUTABILITY AND COMPLEXITY

Meets: MWF, Warren Lecture Hall, 2204 **Instructor:**Russell Impagliazzo **Office:** 4248 CSE building (EBU3b) **E-mail:** russell@cs.ucsd.edu **My Web Page:** http://www-cse.ucsd.edu/users/russell **Course Web Page:** http://www-cse.ucsd.edu/classes/wi19/cse200-a

Office Hours: Tue, Thursday, 3-4, or by appointment. Tuesday 2-3 is undergrad advising; stop by if it's not crowded. **TAs:** Jiawei Gao. Office hours:TBA , Kenneth Hoover, OH: TBA **Texts:** Arora and Barak, Computational Complexity: A Modern Approach.

Recommended: Garey and Johnson, NP-Completeness: A Guide to Intractibility

Recommended for those uncomfortable with proofs: Solow, How to Read and Do Proofs.

Purpose: This course will introduce you to computational complexity theory, which studies and classifies problems according to the resources required to solve them. In particular, we will try to understand which problems are "easy," which are "hard," and how different problems relate to each other. We will attempt to relate both the known results and the unsolved problems from complexity theory to various areas of computer science. In particular, we will show how computationally hard (intractible) problems are an obstacle to many areas of computing, but are actually useful for computer security. We will examine how various answers to P vs NP and related questions would radically change the nature of computer science. We will introduce new models of computation, such as randomized algorithms, and relate them to complexity classes for deterministic models.

Outline (28 lectures total):

1. Models of computation: Turing Machines and variants, RAM model, Boolean circuits. Simulating one machine model with another. The Church-Turing thesis and versions for efficient computation. Digression into communication complexity. (5 lectures; see Chapters 1 and 6 of Arora Barak, with a digression into all of Chapter 13 except multiparty communication complexity).

2. Universal machines and diagonalization. Undecidability. Recursive enumerability. The Halting Problem. Time and Space Hierarchy Theorems. Reductions. (Mostly chapter 3. 4 lectures, but intermingled with the above.)

3. NP-completeness and the P vs NP question. Consequences of $P = NP$. The polynomial-time hierarchy. (5 lectures, Chapters 2 and 5).

4. Space Complexity, (3 lectures, Chapter 4)

5. Randomized computing (Chapter 7, 2-3 lectures)

6. Interactive proofs, cryptographic applications, and hardness of approximation (3 lectures)

7. Circuit lower bounds and pseudo-randomness (3 lectures)

8. Time permitting: Fine-grained complexity; average-case complexity and the theory of cryptography (Chapters 18, 9). Or other topic by request. Up to 3 lectures.

Prerequisites : We assume some undergraduate exposure to algorithms and their analysis, boolean logic, formal language and automata theory, and, most importantly, the ability to read, recognize and write a valid proof. If you have done well in CSE 105 (Theory of computation), CSE 101 (Design and analysis of algorithms) and CSE 20 (Discrete mathematics), or equivalent, you should have the necessary background for this course. If your preparation in these areas is weak, you may need to do some extra work to catch up. Please talk to me about this.

The material for this class is not that difficult, but can be confusing due to its abstract nature. We will not just be discussing particular algorithms for particular problems, but pondering whether, say, each problem of a certain form has some (unspecified) efficient algorithm that solves it. CSE 105 is less important for any particular material covered in it, as it is as an introduction to such abstract thinking about categories of computational problems.

Assignments There will be five homework assignments, which may be done in groups of up to 5, and an individual take-home exam. The first homework is on prerequisite material, and will be graded both as if it were a normal assignment, but also based solely on for which problems an honest attempt was made. Only the latter grade will be recorded; the first is for your information only.

Evaluation: Homework will account for 50 % of the grade, and the final will account for the remaining 50 % of the grade.

MS comprehensive exam The take-home final can be used as a component of the MS comprehensive exam. A B+ (75%) or better is required to pass. Remember that this is strictly an individual exam, using only the professor, TA, class materials from the website, and textbooks. No external sources are permitted, and plagiarism will be grounds for dismissal from the MS program.

Guidelines for homework assignments Students will be allowed to solve all homework assignments in groups of size up to 5. All group members should be acknowledged on the first page of the assignment. Group members are responsible for participating in the solution to all problems submitted, unless a note is attached to the assignment, specifying which problems a group member was not involved with. (Non-participating members will not get credit for the problem, but will also not be responsible for academic honesty for that problem.)

Students should not look for answers to homework problems in other texts, from people other than their study group and the professor, or on the web. However, students may use other texts as a general study tool, and may accidentally see solutions to homework problems. In this case, the student should write up the final solution without consulting the text or other source, and should give an acknowledgement of the text or other source on the first page of their solutions. Such a solution may be given partial or no credit if it too closely follows the text.

Academic Integrity The golden rule of academic integrity is: **If you use someone else's ideas, you must give them credit**. This rule is only relaxed in the case of people whose job it is to give you insights for this class, such as myself, the TA's, and the textbook.

Be sure to follow the following ethical guidelines:

1. Do not discuss problems with people outside your group (except me of course).
2. Do not share written solutions or partial solutions with other groups.
3. Prepare your final written solution without consulting any written material except class notes and the class text.
4. Acknowledge all supplementary texts or other sources that helped you solve any problems. This particularly includes sources from the internet, such as videos or class notes from similar courses elsewhere, research papers, or hints from other students.
5. Each member of the group bears responsibility for the honesty of all parts of an assignment. If you did not contribute to a particular part of an assignment, you must acknowledge this on the first page of the assignment. In that case, you will not recieve credit for that part,

but you also will not have any responsibility for dishonesty regarding that part.

6. Be respectful of the teaching staff and your classmates, and expect others to treat you with respect. If any aspect of the course is making you feel unwelcome, please tell me, possibly with an anonymous piazza post if you do not want to draw attention to yourself.

Standards for evaluation. THIS IS IMPORTANT!!! Grading of all problems (homework and exam) will be both on the basis of correctness and on logical consistency and completeness, i.e., "mathematical style". It is your obligation to provide a compelling argument that forces the reader to believe the result, not just notes from which a proof theoretically could be reconstructed. Describe your answer in a way you might present this material to a skeptical undergraduate honors class.

A few problems will involve performing experiments using

your own and others' programs. The standards for these assignments involve performing a meaningful experiment, and clearly documenting the process and results. This involves describing the algorithm implemented, describing the architecture and environment that the programs were run in, acknowledging any code used written by others, and presenting a clear summary of timing and other data. Do not hand in code; we will not look at it. We may ask you to give a personal demo of your program if something is odd.

Lateness Policy Late homework will be accepted until I give out an answer key and no later. So you have to be no later than me.

Reading You are expected to do a **lot** of independent reading for this course. Most topics I will be presenting are also in the text, and it is good to try to follow along using the Outline above. . I may not give as many details as the text does for some topics, and may use slightly different notation. Some (but not all) topics will also have lecture notes on the class website.

Tentative Schedule

- Jan. 7. Homework 1 (calibration homework) available
- Jan. 14. Homework 1 due.
- Jan. 14 -18. Rusell is away. Classes and TA OH as usual, but no prof. OH. (See announcements of make up OH).
- Jan. 21: MLK holiday.
- Jan. 22: Homework 2 (models of computation, computability) available.
- Jan 28: Homework 2 due. Homework 3 (diagonalization, reductions, NP, NP-completeness) available.

- Feb. 11: Homework 3 due. Homework 4 (Polynomial hierarchy, relativized computation, space complexity) available.
- Feb. 18: President's day holiday.
- Feb. 25: Homework 4 due. Homework 5 (Probabilistic algorithms and complexity classes, counting classes, cryptography, interactive proofs) available.
- March 11: Homework 5 due.
- March 13: Take-home final exam available.
- March 20: Take home final due.