CS 517: Computational Complexity

Instructor: Dr. Mike Rosulek, <rosulekm@eecs.oregonstate.edu>

TA: Hung Le <lehu@oregonstate.edu>

Meets: MWF 2-3, in KEC 1003

Website: http://eecs.oregonstate.edu/~rosulekm/cs517
Please check often for announcements, homeworks, etc.

Textbook: There is no required text. Lecture notes from various sources and in-class slides will be posted on the course website. You will also be expected to be independently resourceful as needed. If you prefer killing trees, I suggest the following:

- *Computational Complexity: A Conceptual Perspective*, Oded Goldreich

Challenges, Expectations, Scare Tactics

In a computational complexity course, we do lots of things that are potentially very alien and conceptually challenging. Most of our time is spent simply **writing algorithms**, but we write algorithms . . .

. . . in exotic, bizarrely constrained, abstract *models of computation* (e.g., a nondeterministic oracle Turing machine with 1 bit of advice)

. . . to solve abstractly specified problems (e.g., solve an arbitrary, unspecified PSPACE-complete problem; disagree with every $n^2$-time Turing machine)
using weird, abstractly specified assumptions (e.g., assuming that any exponential-
time Turing machine has a polynomial-size circuit).

to understand how powerful different abstract resources are in the context of com-
putation (e.g., randomness, interaction, nondeterminism) and to understand what
properties make problems inherently hard to solve (e.g., alternating quantifiers).

This is an advanced & intensive course designed for CS PhD students. This course is
probably going to challenge you no matter what, but you will face an even more significant
uphill battle in this course unless:

you have mastered the basics of undergraduate algorithms and automata theory.
you are comfortable understanding and producing precise mathematical writing.
you are comfortable with high levels of abstraction.

In particular, if you have not taken a graduate-level algorithms course, I suggest you drop
this course and take cs515 first. If you have not taken a course in automata theory, I
suggest you drop this course and take cs516 instead.

Approximate Schedule

4 lectures: Turing machines, universality, undecidability

2 lectures: Kolmogorov complexity

5 lectures: P, NP, nondeterminism, and NP-completeness

2 lectures: Diagonalization: time hierarchies & Ladner’s theorem

4 lectures: Polynomial hierarchy: oracle computations & alternation

3 lectures: Space complexity classes, configuration graph problems, PSPACE-completeness

3 lectures: Complexity of counting: #P and #P-completeness

2 lectures: Complexity of randomized computations

2 lectures: Non-uniform computation

Assessment

40% problem sets

Expect 6-8 problem sets throughout the quarter. Your submissions must be typeset in
\LaTeX{} (a template will be provided) and submitted on TEACH.

In general, throughout this course:
- A clear and correct justification is just as important as a correct answer/algorithm. “Correct” answers with insufficient justification will receive only minimal partial credit.

- If your answer includes an algorithm (highly likely), then justify both the correctness and efficiency of your algorithm (though efficiency is not so important for the first homework).

60% exams

Expect a midterm exam and a final, each accounting for 30% of the final grade. The final will be mostly non-cumulative.

If you score less than 50% on any exam, you automatically fail the class.

Other Policies

Cheating: Academic dishonesty (including plagiarism and cheating) will not be tolerated. Consult the university’s student conduct code for more details. I will follow the guidelines given there, and seek out the maximum allowable penalty for violations that occur in this course.

If you have a question about what constitutes academic dishonesty, please ask me.

Disabilities: Accommodations are collaborative efforts between students, faculty, and Disability Access Services (DAS). Students with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 737-4098.