CS1200: Intro. to Algorithms and their Limitations Prof. Salil Vadhan

Lecture 25: Conclusions

Harvard SEAS - Fall 2025 *2025-12-02*

1 Announcements

- Final exam: Friday 12/12 9am-12pm, Science Center B.
- Review sessions: 12/5 12-3pm (content review, going through algorithms), 12/7 1-3pm (proof strategies for reductions)
- Final exam cheat sheet: One *handwritten* page, double-sided. (Rationale: educational value of actively summarizing the course content yourself.)
- Madhu Sudan P vs. NP lecture highly recommended: Wed 12/3 5p-6pm, Science Center D.
- My last OH for the term: Mon 12/1 5pm-5:45pm (note earlier start time).
- Fill out Q evaluations!

Recommended Reading:

- Hesterberg–Vadhan 29
- Roughgarden IV, Epilogue
- MacCormick, Chapter 18

2 An Algorithmicist's Workflow

The concepts and skills developed in this book provide you with a powerful toolkit for tackling algorithmic problems that you may encounter in the future. Here we synthesize those tools into a potential workflow, mimicking how we approached the "real-world" algorithmic problems we encountered earlier in the book (e.g. Web Search, Interval Scheduling, Census Demographics, Driving Directions, Magic Maze, Register Allocation, Wireless Spectrum Allocation, Kidney Exchange, Lyber, Arithmetic Overflow). When encountering such a problem, you can tackle it by looping through the following steps:

1. **Mathematically model.** Precisely specify the problem you need to solve, addressing questions like the following.

2. **Look for related problems**. By searching your memory, textbooks, the research literature, and the web, you may be able to find an already-solved problem that is related to yours and try to obtain an algorithm by *reduction* *<u>to</u>* the other problem. Some examples we have seen are:

3. If coming up with a reduction to a related problem does not succeed, you may need to **design a new algorithm** for the problem. For this, you can try to apply some of *algorithmic techniques* you have learned, such as:

4. If you are not able to find a satisfactory algorithm for the problem, perhaps none exists. You can thus try to **show hardness or unsolvability** by *reduction from* other problems

5. If you determine that your original problem formulation has no satisfactory algorithm (or you simply fail to find one), you can **settle for weaker guarantees**, for example by:

Importantly, throughout this process, keep the landscape of complexity classes in view:

3 Other Takeaways

Universality.

A Rigorous Mathematical Theory.

Open Problems.

4 Learning Outcomes

It is a good time to reflect on the extent to which you have acquired the skills we set out to develop(as enumerated in the Syllabus):

- To mathematically abstract computational problems and models of computation
- To design and implement algorithms using a toolkit of algorithmic techniques
- To recognize and formalize inherent limitations of computation
- To rigorously analyze algorithms and their limitations via mathematical proof
- To appreciate the technology-independent mathematical theory of computation as an intellectual endeavor as well as its relationship with the practice of computing.

- To engage effectively in a collaborative theoretical computer science learning community, supporting your peers' learning as well as your own
- To clearly communicate mathematical proofs about computation to peers, conveying both high-level intuition and formal details.

5 Algorithms in a World of AI

How will the massive advances in AI and ML affect theoretical computer science and the theory of algorithms as we have seen it in CS1200?

AI/ML systems differ substantially from what we have studied in CS1200 in several respects:

Possibility: ML will eventually replace some of the manual design of algorithms as we have studied in CS1200. Nevertheless, the theory of algorithms and skills from CS1200 likely to remain very valuable for several reasons:

6 Where to Learn More

- Theory of Computation seminar: `https://toc.seas.harvard.edu/events-seminars`
- Many other CS courses, especially x2x0. Look at grad (2xx0) courses as well. (CS1200 may serve as a sufficient substitute for CS1210/CS1240 in some of them.)
- Read more of our textbooks (Roughgarden, MacCormick, CLRS, and the references therein)
- Come talk to us in office hours!