

Computational Complexity for Data Science

Last modified: 2026-10-12 02:31:22 (PDT)

Welcome

Computational complexity asks how much time, memory or randomness a problem needs, whatever algorithm solves it. Data science meets it whenever fitting a model exactly turns out to be computationally hard, which is one reason practical methods use greedy search or a convex surrogate loss.

These notes collect the computational complexity that data science courses use. They are new. The pages written so far are:

- [Asymptotic notation](#): big-O, big-Omega and big-Theta, and what it means for a computation to cost $O(D)$

The planned topics still to come are:

- models of computation: Turing machines and computability
- time and space complexity, and the classes P and NP
- reductions and NP-completeness
- randomized computation
- communication and information complexity

Information theory has its own library, [Information Theory for Data Science](#). Results that connect the two live in the library that holds their main statement, and the other links to them.

Starting materials

These notes are written fresh, but they draw on the following books and Stanford course pages.

- Sanjeev Arora and Boaz Barak, *Computational Complexity: A Modern Approach*, the main starting point for complexity theory.

- [CS154: Introduction to the Theory of Computation](#) (Omer Reingold and Li-Yang Tan), which covers automata, computability and complexity, using Michael Sipser's *Introduction to the Theory of Computation*, with lecture slides and videos.
- [CS154: Introduction to Automata and Complexity Theory](#) (Spring 2009, with Rajeev Motwani among the course staff), with handouts, homework and a reading list.
- [Introduction to Automata Theory, Languages, and Computation](#), Jeff Ullman's home page for the book, which is the textbook the 2009 CS154 page links. It has solutions to starred exercises, errata, slides, and past CS154 course materials.
- [CS254: Computational Complexity](#) (Li-Yang Tan), which covers space complexity, the polynomial hierarchy, randomized and non-uniform computation, and interactive proofs. It uses Arora and Barak and Avi Wigderson's *Mathematics and Computation*, and links a reading for each lecture.

Using these notes in another site

Course sites link to these pages by URL; they do not include this repository as a git submodule. A host site that keeps a copy of this repository at its root, named `cds`, can still include fragments with paths that start with `cds/`. This site includes its own fragments the same way, through a `cds` symlink that points at the repository root.

Quarto resolves `@id` cross-references only within one rendered page, so a host site that links to a result here uses an explicit link to the page and anchor.