Welcome to CS 154

Andreas Garcia  
Duligur Ibeling Duligur  
Natalie Ng  
Omer Reingold  
Li-Yang Tan
Introduction to Automata and the Complexity Theory of Computation
Textbook

Additional Reading: Boaz Barak, Introduction to Theoretical Computer Science (online)
This class is about formal models of computation

What is computation?
What can and cannot be computed?
What can and cannot be efficiently computed?

Philosophy, Mathematics, and Engineering (... Science, ...)
Computing:

“to find out (something) by using mathematical processes”

Merriam-Webster.
Computers
Algorithms
Computing:

“evolution of an environment via repeated application of simple, local rules”

~Avi Wigderson
Computational Lens
Limited Resources
What you Need to Know About the Course
Meet us on Piazza:
piazza.com/stanford/fall2018/cs154/home

Sign up!
piazza.com/stanford/fall2018/cs154
Other Resource Pages

**Submitting Homework:** Gradescope

**Presentations and reading**


**Lecture recordings:** Stanford Canvas

- What? 4 platforms?
- Yes: you are Stanford students, you can handle it
Grades (lower bound)
Homework / Problem Sets

Homework will be assigned every Tuesday (except for the week before the midterm) and will be due one week later at the beginning of class. No late submission. We will drop your lowest homework grade.

Assigned reading: Integral part of course work. Classes good for organizing, reminding, highlighting, giving important perspective. Not so good for replacing a textbook. Doubly true if you forgot CS 103.
Homework Collaboration and Submission

You may (even encouraged to) collaborate with others, but you **must:**

- Try to solve all the problems by yourself *first*
- List your **collaborators on each problem**
- If you receive a significant idea from somewhere, you must acknowledge that source in your solution.
- **Write your own solutions**

Assignments and submissions through [gradescope.com](http://gradescope.com)

- Best to write in LaTex
Thinking of taking another course at the same time?

“Students must not register for classes with conflicting end quarter exams.”

BUT if the other course:
- Allows you to take our midterm in class (Ours is on Tuesday 10/30).
- Does not have its own final exam (Ours is Wednesday, December 12, 12:15-3:15 p.m)

Then we’ll allow it. Not necessarily recommended.
What You’ll Say in December

• Some will say: “a review of 103 with slightly more depth”

• Some: “CS 103 to be pretty easy … but I really struggled in CS 154 just because it moved at a much faster pace”

• All the range from easy to hard; from boring to fascinating

• One will say: “I'm just really fond of Omer's accent.”

• Others will be too polite to say
The Big Challenge and Opportunity

- This class is the first non-mandatory theory class:
  - Offers a pick into some cool stuff
  - Requires more maturity

- My advice:
  - Keep with the lectures, psets, reading
  - Come to office hours
  - Concentrate on knowledge rather than grades (grades will follow).
Back to the Theory of Comutation

We’re Back!
Gödel’s Incompleteness Theorem completely blew my mind ...
Chapter I

Finite Automata (40s-50s):
Very Simple Model (constant memory)
• Characterize what can be computed (through closure properties)
• First encounter: non-determinism (power of verified guessing)
• Argue/characterize what cannot be computed
• Optimization, learning

More modern (complexity-theoretic) perspective:
streaming algorithms, communication complexity
Chapter II

Computability Theory 30’s – 50’s

Very Powerful Models: Turing machines and beyond

(U)n) decidability – what cannot be computed at all

• Foot in the door – an unrecognizable language
• Many more problems, through reductions
• Hierarchy of exceedingly harder problems

The foundations of mathematics & computation

Kolmogorov complexity (universal theory of information)
Chapter III

Complexity Theory: 60’s –
Time complexity, P vs. NP, NP-completeness
• Non-determinism comes back
• Our foot in the door – SAT, a problem that is likely hard to compute
• Many more problems through (refined) reductions
• An hierarchy of hard problems

Other Resources: space, randomness, communication, power,... Crypto, Game Theory, Computational Lens
Map Coloring

Given a map, can we “legally” color the countries with 3 colors? (4-coloring always exists)

This is the central question of computer science (and one of the central questions of math). Really?
A good proof should be:
- **Clear** – easy to understand
- **Correct** and **convincing**
- Like an ogre (or a parfait)
Sipser: In writing mathematical proofs, it can be very helpful to provide three levels of detail

- **1st**: a short phrase/sentence giving a “hint” of the proof
  (e.g. “Proof by contradiction,” “Proof by induction,” “Follows from the pigeonhole principle”)

- **2nd**: a short, one paragraph description of the main ideas

- **3rd**: the full proof
During lectures, my proofs will usually contain intuitive descriptions and only a little of the gory details (the third level) if at all.

You should think about how to fill in the details!

You aren’t required to do this (except on some assignments) but it can help you learn.
Some Methods of Proof

Construction, Contradiction, Induction (and strong induction), and our BFFTC (if not BFF):

Reduction
Proof Example

Suppose \( A \subseteq \{1, 2, \ldots, 2n\} \) with \( |A| = n+1 \)

Show that there are always two numbers in \( A \) such that one number divides the other number

Example: \( A \subseteq \{1, 2, 3, 4\} \)

1 divides every number.

If 1 isn’t in A then \( A = \{2, 3, 4\} \), and 2 divides 4
HINT 1: THE PIGEONHOLE PRINCIPLE
If 6 pigeons occupy 5 holes, then at least one hole will have more than one pigeon
THE PIGEONHOLE PRINCIPLE

If 6 pigeons occupy 5 holes, then at least one hole will have more than one pigeon.
HINT 1: THE PIGEONHOLE PRINCIPLE
If \( n+1 \) pigeons occupy \( n \) holes, then at least one hole will have more than one pigeon.

HINT 2: Every integer \( a \) can be written as \( a = 2^k m \), where \( m \) is an odd number (\( k \) is an integer). Call \( m \) the “odd part” of \( a \).
Proof Idea:

Given $A \subseteq \{1, 2, \ldots, 2n\}$ and $|A| = n+1$

Using the pigeonhole principle, we’ll show there are elements $a_1 \neq a_2$ of $A$ such that $a_1 = 2^i m$ and $a_2 = 2^k m$ (for some odd $m$ and integers $i$ and $k$)
Proof:

\[ A \subseteq \{1, 2, \ldots, 2n\} \text{ with } |A| = n+1 \]

Each element of \( A \) can be written as \( a = 2^k m \Rightarrow m \) is an odd number in \( \{1, \ldots, 2n\} \).

Observe there are \( n \) odd numbers in \( \{1, \ldots, 2n\} \).

Since \( |A| = n+1 \), there must be two distinct numbers in \( A \) with the same odd part.

Let \( a_1 \) and \( a_2 \) have the same odd part \( m \).

\[ \Rightarrow a_1 = 2^i m \text{ and } a_2 = 2^k m \Rightarrow \text{one must divide the other (e.g., if } k > i \text{ then } a_1 \text{ divides } a_2) \]
Parting thoughts:

Computation a powerful notion

Good theory is about the right level of abstraction

Let's have some fun

Questions?