The Big Picture

Please evaluate this course on Axess.

Your feedback really makes a difference.

Final Exam Logistics

- Our final exam is Monday, March 18th from 3:30PM – 6:30PM, location here! Hewlett 200
  - Sorry about how soon that is – the registrar picked this time, not us.
- The exam is cumulative. You're responsible for topics from PS1 – PS9 and all of the lectures.
- As with the midterms, the exam is closed-book, closed-computer, and limited-note. You can bring one double-sided sheet of 8.5" × 11" notes with you to the exam.

Preparing for the Final

- On the course website you'll find
  - ***four*** practice final exams, which are all real exams with minor modifications, with solutions, and
- My recommendation:
  - Look back over the exams and problem sets and redo any problems that you didn't really get the first time around.
  - Redo our in-class polleverywhere questions
- Ask for help, and help each other, on Piazza

Where to Go From Here

We've gone to the absolute limits of computing.

We've probed the limits of efficient computation.

Congratulations on making it this far!

What's next in CS theory?

Formal languages

What problems can be solved by computers?

Regular languages
Context-Free Languages
R and **RE**
P and **NP**

DFAs
NFAs
Regular Expressions
Context-Free Grammars
Recognizers
Deciders
Verifiers
Poly-time TMs/Verifiers

Function problems (CS254)
Counting problems (CS254)

What problems can be solved by computers?

Interactive proof systems (CS254)
Approximation algorithms (CS261/369A)
Average-case efficiency (CS264)
Randomized algorithms (CS265/254)
Parameterized complexity (CS266)
Communication complexity (CS369E)

Nondeterministic TMs (CS154)
Enumerators (CS154)
Oracle machines (CS154)
Space-Bounded TMs (CS154/254)
Machines with Advice (CS254/354)
Streaming algorithms (CS263)
μ-Recursive functions (CS258)
Quantum computers (CS259Q)
Circuit complexity (CS354)

How do we actually get the computer to effectively solve problems?

DFA design intuitions
Guess-and-check
Massive parallelism
Myhill-Nerode lower bounds
Verification
Polynomial-time reductions

How do we actually get the computer to effectively solve problems?

Algorithm design (CS161)
Efficient data structures (CS166)
Modern algorithmic techniques (CS168)
Approximation algorithms (CS261/CS369A)
Average-case efficient algorithms (CS264)
Randomized algorithms (CS265)
Parameterized algorithms (CS266)
Geometric algorithms (CS268)
Game-theoretic algorithms (CS364A/B)

What mathematical structures arise in computer science?

Sets
Propositional and First-Order Logic
Equivalence Relations
Strict Orders
Functions
Injections, Surjections, Bijections
Graphs
Planar and Bipartite Graphs
Polynomial-Time Reductions

What mathematical structures arise in computer science?

Groups, Rings, and Fields (Math 120, CS255)
Trees (Math 108, CS161)
Graphs (Math 107, Math 108)
Hash Functions (CS109, CS161, CS255)
Permutations (Math 120, CS255)
Monoids (CS149)
Lattices and Semilattices (CS143)
Control-Flow Graphs (CS143)
Vectors and Matrices (Math 113, EE103, CS205A)
Modal Logic (Phil 154, CS224M)
Mapping Reductions (CS154)

Where does CS theory meet CS practice?

Finite state machines
Regular expressions
CFGs and programming languages
Password-checking
Secure voting
Polynomial-time reducibility
NP-hardness and **NP**-completeness

Where does CS theory meet CS practice?

Compilers (CS143)
Computational logic (CS157)
Program optimization (CS243)
Data mining (CS246)
Cryptography (CS255)
Programming languages (CS258)
Network protocol analysis (CS259)
Techniques in big data (CS263)
Graph algorithms (CS267)
Computational geometry (CS268)
Algorithmic game theory (CS364)

A Whole World of Theory Awaits!

What's being done here at Stanford?

Algorithms ∩ Game theory

(Tim Roughgarden)

As compute power stretches closer to the infinte-tape TM, and in particular reaches human brain sophistication, how do we protect and care for humanity?

(Fei-Fei Li)

Approximating NP-hard problems

(Moses Charikar)

Optimizing programs... randomly

(Alex Aiken)

Computing on encrypted data
(Dan Boneh)

Correcting errors automatically

(Mary Wootters)

Keep on exploring! There's so much more to learn!


Theory
Practice

So many options – what to do next?

Final Thoughts

There is so much more to explore and so many big questions to ask – ***many of which haven't been asked yet!***

You now know what problems we can solve, what problems we can't solve, and what problems we believe we can't solve efficiently.

Our questions to you:

What problems will you ***choose*** to solve?
Why do those problems matter to you?
And how are you going to solve them?

Parting thought:

These tools we teach you in this course and in this department are powerful.

Please use them wisely.

Your Questions