CS154, Lecture 18:
CS 154 – Final Exam

Wednesday December 13, 12:15-3:15 pm
Skilling Auditorium

You’re allowed one double-sided sheet of notes
Exam is comprehensive (but will emphasize post-midterm topics)
Look for practice final and solutions
Evaluate CS 154

Your Input Really Matters
Finite Automata: 40’s – 50’s
Very Simple Models (constant memory)
Non-determinism (power of guessing)
Learning, communication complexity,
Streaming algorithms

Complexity Theory: 60’s –
Time complexity,
P vs. NP, NP-completeness,
Other Resources: space,
randomness, communication,
power,…
Crypto, Game Theory, …
Computational Lens

Computability Theory 30’s – 50’s
Very Powerful Models
(Turing machines and more)
Decidability,
Kolmogorov Complexity

Some Topics
Computing:

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

Somebody
Computational Game Theory

Markets computing an equilibrium.
Simple dynamics (best response)?

Bounded Rationality:
- Prisoners Dilemma
- Repeated Games, infinite, finite
- Backward Induction
- Finite Automata
- Always Cooperate, Always Defect, Tit for Tat, Trigger
Limited Resources

Example 3.12: Recall Theorem 1.1, where we proved the if-then statement with hypotheses $H = \text{"U is an infinite set, } S \text{ is a finite subset of } U, \text{ and } T \text{ is the complement of } S \text{ with respect to } U." \text{ The conclusion } C = \text{"T is infinite."} \text{ We proceeded to prove this theorem by contradiction. We assumed } \text{ not } C, \text{ that is, we assumed } T \text{ was finite.}

Our proof was to derive a falsehood from } H \text{ and not } C. \text{ We first showed from the assumptions that } S \text{ and } T \text{ are both finite, that } C \text{ also must be finite. But since } U \text{ is stated in the hypothesis } H \text{ to be infinite, and a set cannot be both finite and infinite, we have proved the logical statement } \text{"false." In logical terms, we have both a proposition } p (U \text{ is finite}) \text{ and its negation, not } p (U \text{ is infinite}). \text{ We then used the fact that } p \text{ and not } p \text{ is logically equivalent to } \text{"false."} \square
Factoring & One-Way Functions

Given two primes $P$ and $Q$ easy to compute $N=PQ$. For random such $N$, assume it is hard to find $P$ and $Q$. Special case of One-Way Functions (the most basic cryptographic primitives).

Random Instances of SAT that are hard
Zero-Knowledge Proofs
Hardness of learning
Pseudorandom Generators
  Deterministically increasing entropy
Randomness is weak
Randomness and Pseudorandomness

- When Randomness is Useful
- When Randomness can be reduced or eliminated – derandomization
- Basic Tool: Pseudorandomness
  - An object is pseudorandom if it “looks random” (indistinguishable from uniform), though it is not.
Randomness In Computation (1)

- Distributed computing (breaking symmetry)
- Cryptography: Secrets, Semantic Security, ...
- Sampling, Simulations, ...

Can’t live without you
Randomness In Computation (2)

- Communication Complexity (e.g., equality)
- Routing (on the cube [Valiant]) - drastically reduces congestion
Randomness In Computation (3)

In algorithms – useful design tool, but many times can derandomize (e.g., PRIMES in P). Is it always the case?

BPP=P means that every randomized algorithm can be derandomized with only polynomial increase in time

RL=L means that every randomized algorithm can be derandomized with only a constant factor increase in memory
In Distributed Computing

Dining Philosophers: breaking symmetry

<table>
<thead>
<tr>
<th>Byzantine Agreement</th>
<th>Deterministic</th>
<th>Randomized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronous t failures</td>
<td>t+1 rounds</td>
<td>O(1)</td>
</tr>
<tr>
<td>Asynchronous</td>
<td>impossible</td>
<td>possible</td>
</tr>
</tbody>
</table>

Don’t Attack

Attack Now
Randomness Saves Communication

- **Deterministic**: need to send the entire file!
- **Randomness in the Sky**: $O(1)$ bits (or log in 1/error)
- **Private Randomness**: Logarithmic number of bits (derandomization).
In Cryptography

**Private Keys**: no randomness - no secrets and no identities

**Encryption**: two encryptions of same message with same key need to be different

**Randomized (interactive)**

**Proofs**: Give rise to wonderful new notions: Zero-Knowledge, PCPs, …
Random Walks and Markov Chains

- When in doubt, flip a coin:
- Explore graph: minimal memory
- Page Rank: stationary distribution of Markov Chains
- Sampling vs. Approx counting. Estimating size of Web
- Simulations of Physical Systems
- ...
Communication network (n-dimensional cube)

Every deterministic routing scheme will incur exponentially busy links (in worse case)

Valiant: To send a message from $x \rightarrow y$, select node $z$ at random, send $x \rightarrow z \rightarrow y$.

Now: $O(1)$ expected load for every edge

Another example – randomized quicksort

Smoothed Analysis: small perturbations, big impact
In Private Data Analysis

Hide Presence/Absence of Any Individual

How many people in the database have the BC1 gene?
Add random noise to true answer distributed as $\text{Lap}(\Delta/\varepsilon)$
What’s next?

A few possibilities (more to come):

- CS161 – Design and Analysis of Algorithms
- CS168 - The Modern Algorithmic Toolbox (spring)
- CS250 – Algebraic Error Correcting Codes (Winter)
- CS254 – Complexity Theory (next year)
- CS255 - Introduction to Cryptography (winter)
- CS264 - Beyond Worst Case Analysis
- CS296G - Almost Linear Time Graph Algorithms
- CS 352 - Pseudorandomness (next year)
- CS250 - Error Correcting Codes: Theory and Applications
- CS261 - Optimization and Algorithmic Paradigms (Winter)
- CS265 - Randomized Algorithms and Probabilistic Analysis
- CS368 - Algorithmic Techniques for Big Data
Parting thoughts:

• Computation is a powerful notion, becoming increasingly central
• Theory allows us to model and analyse computation, reaching non-trivial understanding
• Much is still open – waiting for you
That's all Folks!