Complexity Theory tries to understand the power and limitations of different resources of computation, as well as their tradeoffs.

So far discussed time, memory, access to input (read-once; random access), communication. There is much more!

There is much more to study and say.
Highly recommended – CS 254

This video – random as a resource!
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, …
Randomness In Computation (2)

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

### Byzantine Agreement

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

**Dining Philosophers:** breaking symmetry

**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 $x \to y$, select node $z$ at random, send $x \to z \to y$.

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 $\text{Lap}(\Delta/\varepsilon)$
Cryptography:

Good Pseudorandom Generators are Crucial

- With them, we have one-time pad (and more):

\[
\text{ciphertext} = \text{plaintext} \oplus K = \text{01101011}
\]

- Without, keys are bad, algorithms are worthless (theoretical & practical)

plaintext data:
00001111

short key \(K_0\): 110

derived key \(K\): 01100100

plaintext data:
01100100

ciphertext = plaintext \oplus K = 01101011

plaintext data:
00001111

plaintext data:
00001111

plaintext data:
00001111

plaintext data:
00001111
Parting thoughts:
Randomness is powerful and weak
Randomness and Pseudorandomness play a role “everywhere” in CS