1 Communication Complexity

**Model:** Two parties, called Alice and Bob, receive inputs $x, y$ respectively. They can exchange messages, in rounds, until one of them (or both) reports an output $f(x, y)$.

Main measure is communication complexity, i.e., total communication between the parties (in bits, worst-case).

Variants of randomization: none (deterministic), shared/public, or private.

Number of rounds: zero (simultaneous, i.e., each sends a message to a referee and not directly to each other), one (one-way communication), or more/unbounded.

Many other variants, like more players communicating in series (or broadcast etc.), with different input model (e.g., number on forehead instead of number in hand).

**Equality as an Example:**

Problem definition: Alice and Bob’s inputs are $x, y \in \{0, 1\}^n$, and their goal is to compute $EQ(x, y) = 1_{\{x = y\}}$.

Public randomness: There is a (simultaneous) protocol with $O(1)$ bits.

Private randomness: There is a (one-way) protocol with $O(\log n)$ bits.

Deterministic one-way: Every protocol requires $\Omega(n)$ communication bits.

2 Indexing

**Problem definition:** Alice has input $x \in \{0, 1\}^n$ and Bob has as input an index $i \in [n]$. Their goal is to output $INDEX(x, i) = x_i$.  

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.*
This function would be easy if Bob could send his (short) input to Alice. But we shall consider one-way communication from Alice to Bob, and her input is much longer.

**Theorem 1 [Kremer, Nisan, and Ron, 1999]:** The randomized one-way communication complexity of indexing is $\Omega(n)$, even with shared randomness.

It’s therefore a “canonical” problem for reductions (in this model).

We skipped the proof of this theorem (those interested can find a simple proof by [Jayram, Kumar and Sivakumar, 2008] that uses an error correcting code and some averaging arguments).

### 3 Streaming Lower Bounds: Exact $\ell_0$

**Theorem 2:** Every streaming algorithm for computing $\ell_0$ exactly in $\mathbb{R}^n$, even a randomized one with error probability $1/6$, requires storage of $\Omega(n)$ bits.

Remark: This is true even for insertions-only streams.

**Proof:** Was seen in class, by reduction from the indexing problem.

Remark: Notice that our proof works even if random coins are not counted in the storage of the streaming algorithm (because we rely on a communication lower bound with public coins).

**Exer:** Show a similar lower bound for exact $\ell_1$.

Hint: You obviously must use a stream with deletions.

**Exer:** Prove that every streaming algorithm for graph connectivity on $n$ vertices (i.e., deciding whether a stream of edge-insertions gives a connected graph), even a randomized one with error probability $1/3$, requires storage of $\Omega(n)$ bits.

### 4 Gap Hamming Distance (GHD)

**Problem definition:** Alice and Bob’s inputs are $x, y \in \{0, 1\}^n$, respectively, and their goal is to determine whether the hamming distance between $x, y$ is $\leq \frac{n}{2} - \sqrt{n}$ or $\geq \frac{n}{2} + \sqrt{n}$.

**Theorem 3 [Woodruff, 2004]:** The randomized one-way communication complexity of GHD is $\Omega(n)$, even with shared randomness.

We skipped the proof of this theorem (those interested can find a proof by [Jayram, Kumar and Sivakumar, 2008] that uses a reduction from Indexing).

We mention in passing a stronger result, where the number of rounds is unbounded.

**Theorem [Chakrabarti and Regev, 2011]:** The communication complexity (with unbounded number of rounds) of GHD is $\Omega(n)$, even with shared randomness.
5 Streaming Lower Bounds: Approximate $\ell_0$

**Theorem 4:** Every streaming algorithm that $(1 + \varepsilon)$-approximates $\ell_0$ in $\mathbb{R}^n$ for $1/\sqrt{n} \leq \varepsilon < 1$, even a randomized one with error probability $1/6$, requires storage of $\Omega(1/\varepsilon^2)$ bits.

Remark: For smaller $0 < \varepsilon < 1/\sqrt{n}$, the required storage is $\Omega(n)$; to see this, observe that an algorithm for such “smaller” $\varepsilon$ “solves” $\varepsilon = 1/\sqrt{n}$ which is covered by the above theorem.

We skipped the proof of this theorem (for those interested, it is by reduction from GHD).

6 Current Research Directions

We concluded with a brief mention of research topics related to the course.

**Streaming matrices:** Different update models, different problems

**Streaming (and sampling) edit distance:** Different models of the input

**Fast algorithms:** in classic sense, like near-linear time

**Dynamic algorithms:** fast update time (no space constraints)