Today: Data streaming lower bounds from communication complexity.

View problem as a communication game.

Data stream

current state of working memory sufficient to start processing $\sigma_2$

final output/state

$M_1$ [ writelines]

$\sigma_1$, $\sigma_2$
Alice & Bob each have an input \( \sigma_1, \sigma_2 \) respectively & want to compute \( f(\sigma_1, \sigma_2) \) with min # of bits communicated.

Intuitively, if we have a streaming alg with space \( s \) bits, \( I \) a comm. protocol in which Alice sends the \( s \) bits & Bob on \( s + \sigma_2 \) outputs the answer.

So if can show a reduction from a communication problem to a streaming problem, \( I \) have a lower bound for comm. problem then have lower bound for the amount of space of streaming problem.
Some basic communication complexity:

2 (or more parties) cooperate to compute \( f \) on their inputs.

Say \( f: X \times Y \rightarrow \{0, 1\} \)

Alice gets input \( x \in X \)

Bob

Goal: output \( f(x, y) \) with min amount of bits exchanged.

Alice \( x \rightarrow \) Bob \( y \)

Bob \( y \rightarrow \) Alice \( f(x, y) \)

Trivial: \( \min \{ \log |X|, \log |Y| \} + 1 \)

Want: sublinear in \( |X| \)

(send entire msg)
Parties follow a protocol represented by a binary decision tree. Nodes are associated with the decision tree, where leaves output $0/1 = f(x, y)$.

$\overline{\text{Def}}$: $\overline{\Pi}$ computes $\overline{f}$ if $\text{out}_{\overline{\Pi}}(x, y) = \overline{f}(x, y)$ for all $x, y \in X \times Y$. 

$\overline{\text{msg}}_{A, \text{round } i} : x \rightarrow \overline{y}_{i, A}$

$\overline{\text{msg}}_{B, \text{round } i} : y \rightarrow \overline{y}_{i, B}$
Cost of protocol $\pi$: worst case # of bits communicated over all inputs.

Types of protocols:

- deterministic: $D(f)$: min cost of a protocol that is correct on every input.

- Randomized:
  - private coin - players have private random bits
  - public coin - players use a public string of random bits

  stronger version

  usual notion of randomized protocol here
Def: $T_1$ computes $f$ with error $\delta$ if

$$
\forall (x,y) \in X \times Y \exists \delta \left( \Pr[\text{out} \ T_1(x,y;R) \neq f(x,y)] \leq \delta \right).
$$

Def: $R_\delta(f)$ is the min cost of a

$\delta$-error protocol for $f$.

Claim: $0 < \delta < \frac{1}{3}$

$$
R_\delta(f) = O \left( R(f)^{\frac{1}{3}} \right).
$$

Def: $D^2(f)$

$R^2(f)$ > one way protocols.

Alice only talks.

$D^k(f)$; $R^k(f)$ k-round communication
Basic Obstructions:

1. \( R(f) \leq D(f) \leq \min \{ \log |x|, \log |y| \} \)
2. \( D(f) \leq D(f)^k, \forall k > 0 \)
3. \( R(f) \leq R^e(f) \leq R^e(f)^k \)

Specific com. problems:

\[ \text{EQ}_N(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \]

\[ D(\text{EQ}_N) = N \]
\[ R^{\text{priv}} = O(\log N) \]
\[ \text{IND}(x, i) = x_i \]
\[ \in \{0, 1\}^N \quad \forall \in [N] \]

One way:
\[ D(\text{IND}_N) > N \]
\[ R^*(\text{IND}_N) \geq \Omega(N) \quad \forall (\text{IND}_N) \leq \Omega(N) \]
\[ D^2(\text{IDX}) \leq \log N \]

\[ \text{DIST}_N(x, y) = 1 \quad \text{if} \quad x \cap y = \emptyset \]
\[ \forall \in \{0, 1\}^N \]
\[ R(\text{DIST}_N) \geq \Omega(N) \]
Claim: $D(\text{id}_N^x) \supset N$

Pf: Alice $\xrightarrow{\text{msg}(x)}$ Bob.

$X \xrightarrow{\text{msg}(x)} j$

$? x_j$

Deterministic: Bob, no matter what $j \in N$ he has, must recover $x_j$ from $m(x)$, exactly.

Alice has $2^N$ many msgs. $x$

Bob needs to recover every bit of $x$ from $m(x)$. So Bob can recover $x$ from $\text{msg}(x)$.

$x \xrightarrow{\text{msg}(x)} \xrightarrow{\text{Bob}(\text{msg}(x))} X$

So both mappings are 1-1 & onto.

How many msg can be sent of length $\leq N$
\[2^{N-1} + 2^{N-2} + \cdots + 2 + 1 = 2^N - 1 < 2^N \leq\]

So if \( \exists \) protocol of length \( \leq N-1 \)

the \( \text{msg} \) wouldn't be a bijection.

(contradiction)

Claim: \( R \overset{\text{priv}}{\to} (\text{EQ}_N) = O(\log N) \)

Alice 
\[x, r \]

Bob 
\[y\]

\( \text{msg}(x, r) \rightarrow y \)

\( \text{EQ}_N (x, y) = \sum_{i \in [N]} x_i t^i \)?

\( \sum_{i \in [N]} y_i t^i \)

Pf: Alice, Bob can view their inputs as polys:
\( x \rightarrow P_x(t) = \sum_{i \in [N]} x_i t^i \)
\( y \rightarrow P_y(t) = \sum_{i \in [N]} y_i t^i \)
\( P_x, P_y \in \mathbb{F}[t] \) — they agree on beforehand.

Alice picks random \( r \in \mathbb{F} \) and sends \( r, p_x(r) \) to Bob.

Bob compares \( p_y(r) \) with \( p_x(r) \).

If \( "\neq" \) output 1.

else output 0.

\# bits communicated:

\[
|r| + |p_x(r)| + 1
\]

\[
\log |\mathbb{F}| + \log |\mathbb{F}| = 2 \log |\mathbb{F}|
\]

\[
\Pr[\text{error} \neq 0] \Rightarrow O(\log N)
\]
\[ x: y \quad P_r \left[ P_x (r) = P_y (r) \right] \leq \frac{N}{\| F \|} \]

\[ |F| = 20N \]

or \[ |F| = N^2 \]

**Communication Matrix**

\[ \begin{pmatrix} x & x' & y & y' \\ x & & & \\ x' & & & \\ y & & & \\ y' & & & \end{pmatrix} \]

**Detour into Log-Rank Conjecture**

Def: \( C \subseteq x \times y \) is a combinatorial rectangle if \( C = x' \times y' \)

\[ x' \leq x \quad y' \leq y \]

Def: \( C \) is monochromatic wrt \( f \) if \( f(x, y) = f(x', y') \) for \( (x, y), (x', y') \in C \).

Def: Let \( C^D (f) \) be \( \min \) \# of monochromatic rectangles that can partition \( M_f \).

Thm: \( D (f) \geq \log C (C^D (f)) \).
**Thm:** \( f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\} \)

\[ D(f) \geq \log (\text{rank}(M_f)) \]

**Claim:** Let \( l \) be a leaf of the decision tree produced

Let \( C_e = \xi(x, y) \) whose transcript ends at \( l \).

then \( C_e \) is a combinatorial rectangle:

\[ \forall (x, y) \in C_e : \exists (x', y') \in C_e. \]

Thus \( D(f) \) NP hard to compute

- Bigconj: \( D(f) = O(\text{poly log rank} M_f) \)

Lovett: \( D(f) = O(\sqrt{\text{rank} M_f \cdot \text{log rank} M_f}) \)
Thm: Any single-pass det algorithm for deciding if a graph on $n$ vertices has a perfect matching needs $\Omega(n^2)$ bits of memory.

\[ D(i|D_{\text{N}}(x,i)) = \Omega(N) \]

\[ R(i|D_{\text{N}}(x,i)) = \Omega(N) \]

Given
\[
\begin{array}{cccc}
0 & 0 & 1 \\
0 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\]

Alice gets $x$, Bob gets $(i,j)$

\[ \Omega(i|D_{\text{N}}(x,i)) = \Omega(n^2) \]
E.g.,

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

\((i,j) = (3,4)\)

Claim: \(M(i,j) = 1\) if and only if \(G\) has a perfect matching, \(\nu\).

So, stream Alice's edges before Bob's edges. Can come up with protocol in which Alice sends...
memory content to Bob as soon as her stream of edges was processed.

So, if one can solve streaming problem in \( o(n^2) \) space then one can solve \( \text{IND}_{n^2}_n \) problem with \( o(n^2) \) bits communicated, a contradiction!

**Thm:** Any streaming deterministic one-pass approx alg for shortest Path \((v,w)\) with approx factor \( \leq \frac{5}{3} \) uses \( \Omega(n^2) \) space. 

**Pr:** Reduce from \( \text{ID} \times (M_{n \times n}, i, j) \).

Alice's edges

\[
M(i,j) = 1
\]

Bob's edges

\[
G \quad G_{n'}
\]
Use the streaming alg with \( o(n^2) \) space to compute \( \frac{5}{3} \) approx for \( d(v, w) \).

Output \( S \). Stream Alice's edges before Bob's. In the cc protocol Alice sends her memory content to Bob.

If \( M(i, j) = 1 \) then \( d(v, w) = 3 \)

\[ \text{ow } d(v, w) \geq 5 \]

(\text{could be } \infty )

\[
\begin{array}{cccc}
3 & ? & 5 & ? \\
\hline
\end{array}
\]

Use output \( S \) to decide whether edge \((i, j)\) exists.

If \( S < 5 \) output \( \text{IDX}(M, i, j) = 1 \)

\[ \text{ow } \text{output } \text{IDX}(M, i, j) = 0 \]

(Indeed, if \( \text{OPT} = 3 \Rightarrow S \leq 5 \)

\[ \text{ow } S > \text{OPT} = 5 \)

So, can solve \( \text{IND}_{n^2} \) in \( O(n^2) \). Contradiction.