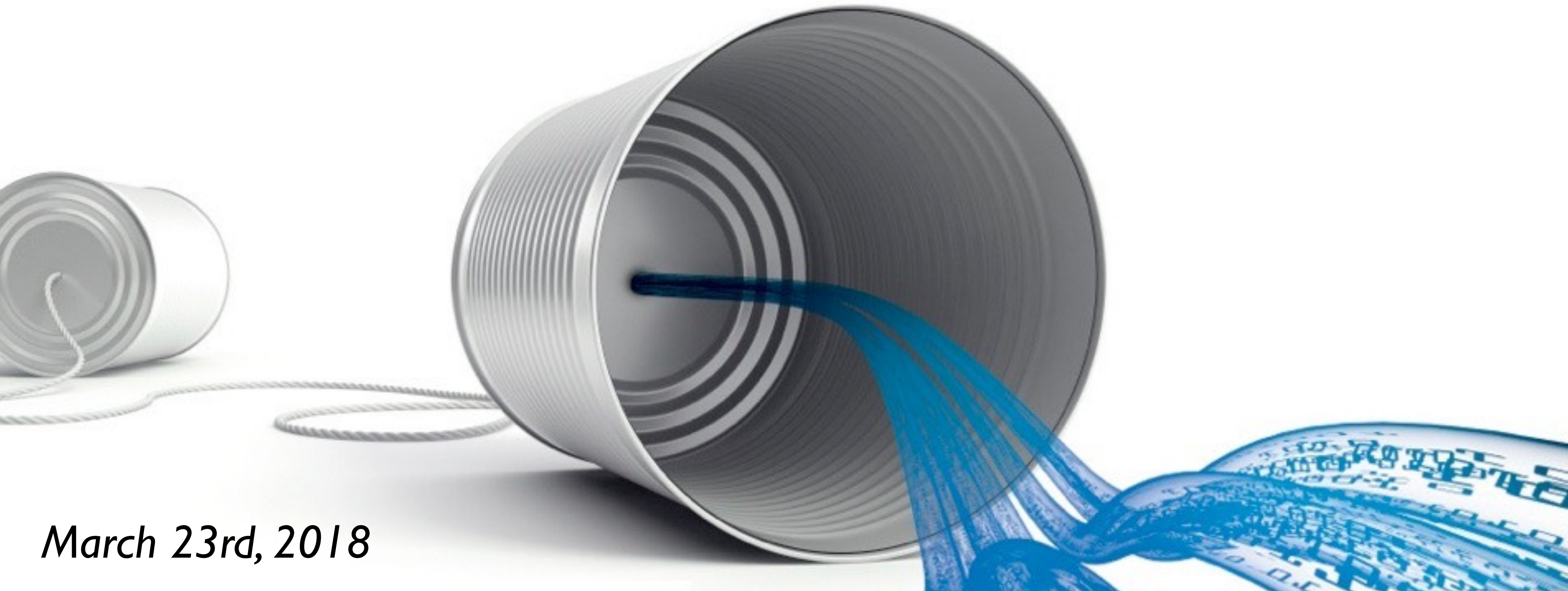
15-252
More Great Ideas in Theoretical Computer Science

Lecture 7: Communication Complexity



March 23rd, 2018

What are the limitations to what computers can learn?

Do certain mathematical theorems have short proofs?

Can quantum mechanics be exploited to speed up computation?

Is every problem whose solution is efficiently verifiable also efficiently solvable? *i.e.* P = NP?

What are the limitations to what computers can learn?

Do certain mathematical theorems have short proofs?

Can quantum mechanics be exploited to speed up computation?

Is every problem whose solution is efficiently verifiable also efficiently solvable? *i.e.* P = NP?

Communication complexity

Cool Things About Communication Complexity

Many useful applications:

machine learning, proof complexity, quantum computation, pseudorandom generators, data structures, game theory,…

The setting is simple and neat.

Beautiful mathematics

combinatorics, algebra, analysis, information theory, …

Motivating Example 1: Checking Equality

010010101110101 $\stackrel{?}{=}$ 010010100110101

$\longleftarrow$ n bits $\longrightarrow$ $\qquad$ $\longleftarrow$ n bits $\longrightarrow$

How many bits need to be communicated?

Naively: n $\qquad$ Actually: n

What if we allow 0.0000000001% probability of error?

Naively: $\Omega(n)$ $\qquad$ Actually: $O(\log n)$

Motivating Example 2: Auctions

Alice

Bob

$100

$1000

Defining the model a bit more formally

2 Player Model of Communication Complexity

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$$

known to
both players

(Alice)

$x \in \{0,1\}^n$

(Bob)

$y \in \{0,1\}^n$

Goal: Compute $F(x,y)$. (both players should know the value)

How: Sending bits back and forth according to a protocol.

Resource: Number of communicated bits.

(We assume players have unlimited computational power individually.)

Poll 1

$x, y \in \{0,1\}^n, \quad PAR(x,y) =$ parity of the sum of all the bits.

(i.e. it's 1 if the parity is odd, 0 otherwise.)

How many bits do the players need to communicate?
Choose the tightest bound.

$O(1)$

$O(\log n)$

$O(\log^2 n)$

$O(\sqrt{n})$

$O(n/\log n)$

$O(n)$

Poll 1 Answer

$x, y \in \{0,1\}^n$, $PAR(x,y) =$ parity of the sum of all the bits.

(i.e. it's 1 if the parity is odd, 0 otherwise.)

How many bits do the players need to communicate?
Choose the tightest bound.

Once Bob knows the parity of x, he can compute $PAR(x,y)$.

- Alice sends $PAR(x)$ to Bob. 1 bit

- Bob computes $PAR(x,y)$ and sends it to Alice. 1 bit

2 bits in total

2 Player Model of Communication Complexity

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$$

known to both players

(Alice)

$x \in \{0,1\}^n$

(Bob)

$y \in \{0,1\}^n$

Goal: Compute $F(x,y)$. (both players should know the value)

How: Sending bits back and forth according to a protocol.

Resource: Number of communicated bits.

2 Player Model of Communication Complexity

Goal: Compute $F(x, y)$. (both players should know the value)

How: Sending bits back and forth according to a protocol.

Resource: Number of communicated bits.

A protocol P is the "strategy" players use to communicate.

It determines what bits the players send in each round.

$P(x, y)$ denotes the output of P.

2 Player Model of Communication Complexity

Goal: Compute $F(x, y)$. (both players should know the value)

How: Sending bits back and forth according to a protocol.

Resource: Number of communicated bits.

A (deterministic) protocol P computes F if

$$\forall (x, y) \in \{0,1\}^n \times \{0,1\}^n, \qquad \underbrace{P}_{\text{algorithm (TM)}}(x, y) = \underbrace{F}_{\text{decision problem}}(x, y)$$

Analogous to: algorithm (TM) decision problem

$$\forall x \in \Sigma^* \quad A(x) = F(x)$$

2 Player Model of Communication Complexity

Goal: Compute $F(x, y)$. (both players should know the value)

How: Sending bits back and forth according to a protocol.

Resource: Number of communicated bits.

A randomized protocol P computes F with ϵ error if

$$\boxed{\forall (x, y) \in \{0,1\}^n \times \{0,1\}^n}, \quad \Pr[P(x, y) \neq F(x, y)] \leq \epsilon$$

2 Player Model of Communication Complexity

Goal: Compute $F(x, y)$. (both players should know the value)

How: Sending bits back and forth according to a protocol.

Resource: Number of communicated bits.

$$\text{cost}(P) = \max_{(x,y)} \# \text{ bits } P \text{ communicates for } (x, y)$$

if P is randomized, you take max over the random choices it makes.

Deterministic communication complexity

$\mathbf{D}(F) =$ min cost of a (deterministic) protocol computing F.

Randomized communication complexity

$\mathbf{R}^{\epsilon}(F) =$ min cost of a randomized protocol computing F with ϵ error.

2 Player Model of Communication Complexity

Goal: Compute $F(x, y)$. (both players should know the value)

How: Sending bits back and forth according to a protocol.

Resource: Number of communicated bits.

$$\text{cost}(P) = \max_{(x} \; \# \text{ bits } P \text{ communicates for } (x, y)$$

es.

Deterministic co

$\mathbf{D}(F)$ = mi ...uting F.

Randomized co

$\mathbf{R}^{\epsilon}(F)$ = min ...uting F with ϵ error.

We usually fix ϵ to some constant.

e.g. $\epsilon = 1/3$

We can always boost the success probability if we want.

What is considered hard or easy?

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$$

$$0 \leq \quad \mathbf{R}_2^{\epsilon}(F) \leq \mathbf{D}_2(F) \quad \leq n+1$$

$$c \qquad \log^c(n) \qquad\qquad n^\delta \qquad \delta n$$

Example

Equality: $$EQ(x,y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathbf{D}(EQ) = n + 1. \qquad \mathbf{R}^{1/3}(EQ) = O(\log n).$$

Poll 2

$MAJ(x, y) =$ 1 iff majority of all the bits in x and y are set to 1.

What is $\mathbf{D}(MAJ)$? Choose the tightest bound.

$O(1)$

$O(\log n)$

$O(\log^2 n)$

$O(\sqrt{n})$

$O(n/\log n)$

$O(n)$

Poll 2 Answer

$MAJ(x,y) =$ 1 iff majority of all the bits in x and y are set to 1.

What is $\mathbf{D}(MAJ)$? Choose the tightest bound.

The result can be computed from

$$\sum_{i \in \{1,2,\ldots,n\}} x_i + \sum_{i \in \{1,2,\ldots,n\}} y_i$$

- Alice sends $\sum_i x_i$ to Bob. ~ log n bits
- Bob computes $MAJ(x,y)$ and sends it to Alice. 1 bit

$O(\log n)$ in total

Another example: Disjointness function

Can view the input string as a subset of $\{1,2,3,\ldots,n\}$

$$S_x = \{2, 4, 5\}$$

$$x = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline \end{array}$$
$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array}$$

Disjointness: $DISJ(S_x, S_y) = \begin{cases} 1 & S_x \cap S_y = \emptyset \\ 0 & \text{otherwise} \end{cases}$

$$\mathbf{R}^{1/3}(DISJ) = \Omega(n).$$ hard!

The plan

1. Efficient randomized communication protocol for checking equality.

2. An application of communication complexity.

3. A few words on proving lower bounds.

Efficient randomized communication protocol for checking equality

The Power of Randomization

$$\mathbf{R}^{1/3}(EQ) = O(\log n).$$

The Protocol:

Alice's Input:	$a_0 a_1 a_2 \ldots a_{n-1} \in \{0,1\}^n$
Bob's Input:	$b_0 b_1 b_2 \ldots b_{n-1} \in \{0,1\}^n$

Alice picks a prime $p \in [n^2, 2n^2]$ and a random $t \in \mathbb{Z}_p$.

Alice builds polynomial

$$A(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} \in \mathbb{Z}_p[x]$$

<u>Alice sends Bob:</u> $p, \ t, \ A(t) \rightarrow O(\log n)$ bits

The Power of Randomization

$$\mathbf{R}^{1/3}(EQ) = O(\log n).$$

The Protocol:

Alice picks a prime $p \in [n^2, 2n^2]$ and a random $t \in \mathbb{Z}_p$.

Alice builds polynomial

$$A(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} \in \mathbb{Z}_p[x]$$

Alice sends Bob: $p,\ t,\ A(t) \rightarrow O(\log n) \text{ bits}$

Bob builds polynomial $B(x) \in \mathbb{Z}_p[x]$

Output: If $A(t) = B(t)$, output 1. Otherwise, output 0.

The Power of Randomization

$$\mathbf{R}^{1/3}(EQ) = O(\log n).$$

Analysis:

<u>*Want to show:*</u> For all inputs (a,b), probability of error is $\leq \epsilon$.

For all (a,b) with $a = b$:

$$\Pr_t[\text{error}] = \Pr_t[A(t) \neq B(t)] = 0$$

For all (a,b) with $a \neq b$:

$$\Pr_t[\text{error}] = \Pr_t[A(t) = B(t)] = \Pr_t[(A-B)(t) = 0]$$

$$= \Pr_t[\underbrace{t \text{ is a root of } A - B}_{\text{degree}(A-B) \leq n-1}] \leq \frac{n-1}{p} \leq \frac{n-1}{n^2} \leq \frac{1}{n}$$

An application of communication complexity

Applications of Communication Complexity

- circuit complexity
- time/space tradeoffs for Turing Machines
- VLSI chips
- machine learning
- game theory
- data structures
- proof complexity
- pseudorandom generators
- pseudorandomness
- branching programs
- data streaming algorithms
- quantum computation
- lower bounds for polytopes representing NP-complete problems

Applications of Communication Complexity

- circuit complexity

- time/space tradeoffs for Turing Machines

- VLSI chips

- machine learning

- game theory

- data structures

- proof complexity

- pseudorandom generators

- pseudorandomness

- branching programs

- data streaming algorithms

- quantum computation

- lower bounds for polytopes representing NP-complete problems

How Communication Complexity Comes In

Setting: Solve some **task** while minimizing some **resource**.

e.g. *find a fast algorithm, design a small circuit, find a short proof of a theorem, …*

Goal: Prove lower bounds on the **resource** needed.

Sometimes we can show:

efficient solution to our problem $\longrightarrow$

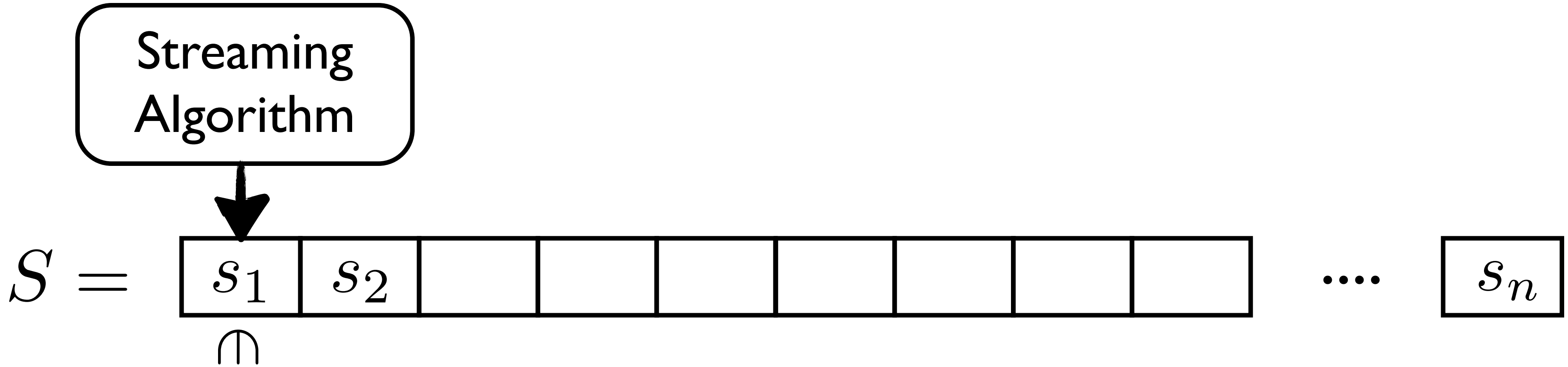
efficient communication protocol for a certain function.

i.e. no efficient protocol for the function $\longrightarrow$

no efficient solution to our problem.

Lower bounds for data streaming algorithms

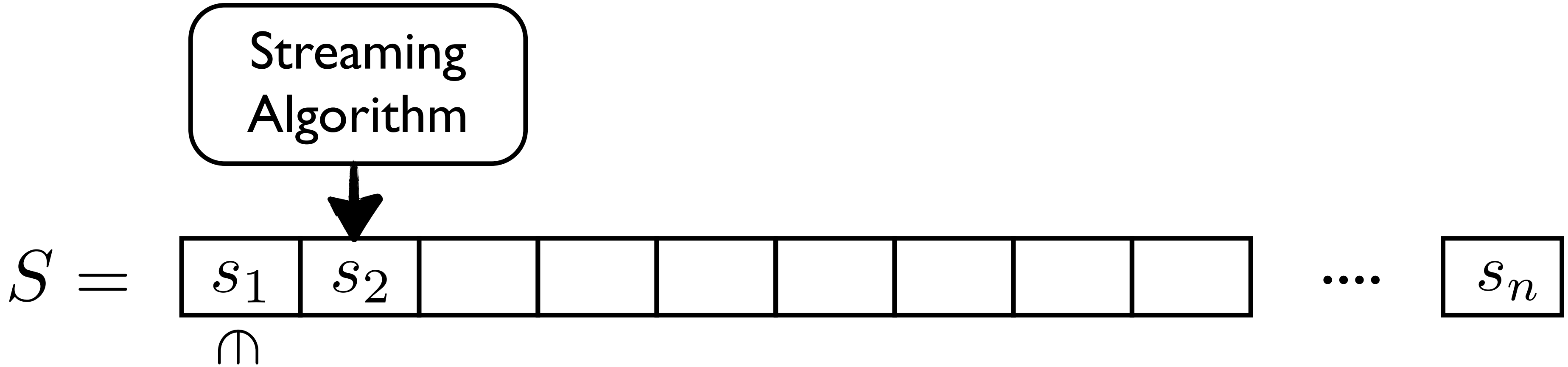
Data Streaming Algorithms



$$[n] = \{1, 2, \ldots, n\}$$

$$S \in [n]^n$$

Data Streaming Algorithms

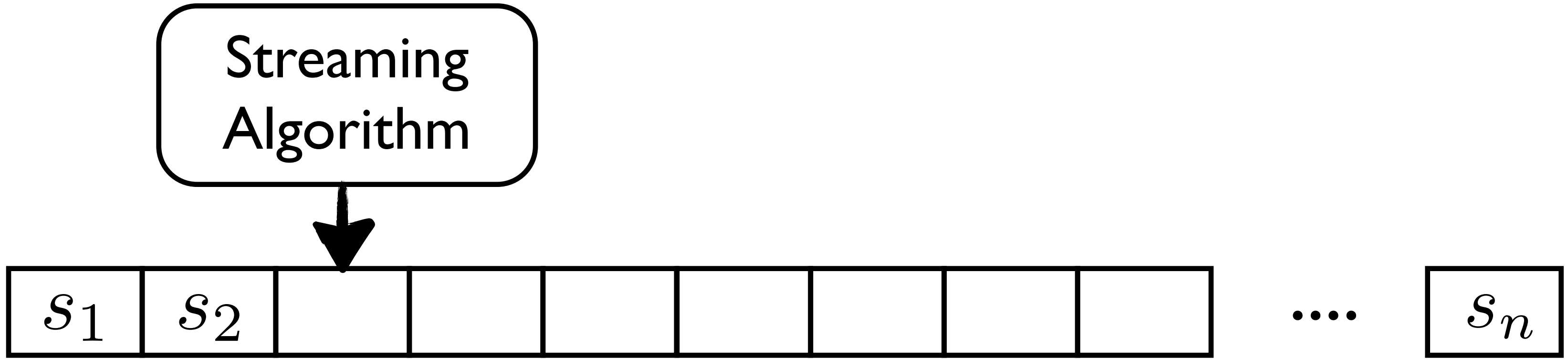


$$[n] = \{1, 2, \ldots, n\}$$

$$S \in [n]^n$$

Data Streaming Algorithms

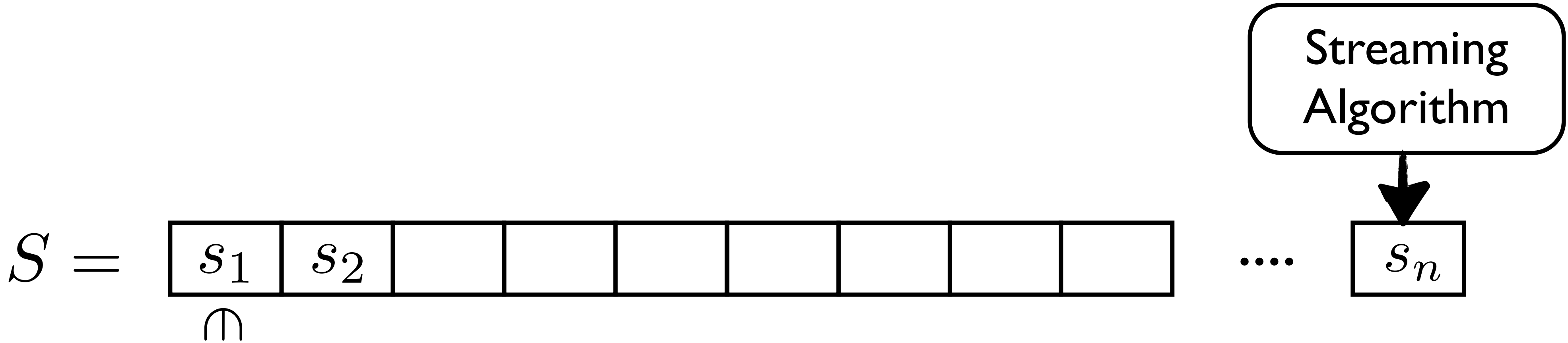
$S =$



$$[n] = \{1, 2, \ldots, n\}$$

$$S \in [n]^n$$

Data Streaming Algorithms



$$[n] = \{1, 2, \ldots, n\}$$

$$S \in [n]^n$$

Fix some function $f : [n]^n \to \mathbb{Z}.$

e.g. $f(S) = \#$ most frequent symbol in S

Goal: On input S, compute (or approximate) $f(S)$
while minimizing space usage.

Lower Bounds via Communication Complexity

$$f(S) = \#\ \text{most frequent symbol in } S$$

Space efficient streaming algorithm computing f $\longrightarrow$ communication efficient protocol computing $DISJ$.

Disjointness: $DISJ(S_x, S_y) = \begin{cases} 1 & S_x \cap S_y = \emptyset \\ 0 & \text{otherwise} \end{cases}$

Lower Bounds via Communication Complexity

$$f(S) = \#\ \text{most frequent symbol in } S$$

Space efficient streaming algorithm computing f $\longrightarrow$ communication efficient protocol computing $DISJ$.

$$S_x = \{2, 4, 5\} \qquad\qquad S_y = \{1, 5, 7, 8\}$$

$$x = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} & \multicolumn{1}{c}{7} & \multicolumn{1}{c}{8} \end{array} \qquad y = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ \hline \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} & \multicolumn{1}{c}{7} & \multicolumn{1}{c}{8} \end{array}$$

Protocol: Alice runs streaming algorithm on S_x .
She sends the state and memory contents to Bob.
Bob continues to run the algorithm on S_y .
If $f(S_x \cdot S_y) = 2$, Bob outputs 0, otherwise 1.

Correctness ✓ Cost ✓

A few words on showing lower bounds

The function matrix

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$$

$$M_F = \begin{array}{cc} & y \\ x & \boxed{\begin{matrix} 011010111010111111001010010101000 \\ 010101010110101001010100101111100 \\ 010100001010111010101010111101011 \\ 000101110101111010101100101010101 \\ 001010101010101010110100010101011 \\ 010111010111010010110101011110100 \\ 010101110101010101000101000101101 \\ 010101010111010101101101101110101 \\ 111010101011010101010101011111100 \\ 111011101010101010101010101001111 \\ 110101101010101000101010101000101 \\ 011110000111110000000001110101111 \\ 011010111010111111001010010101000 \\ 001010101010100101010101111110000 \\ 101010101000001110101011101011000 \end{matrix}} \end{array}$$

$$M_F[x, y] = F(x, y)$$

2^n by 2^n matrix

The function matrix

Equality: $EQ(x, y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$

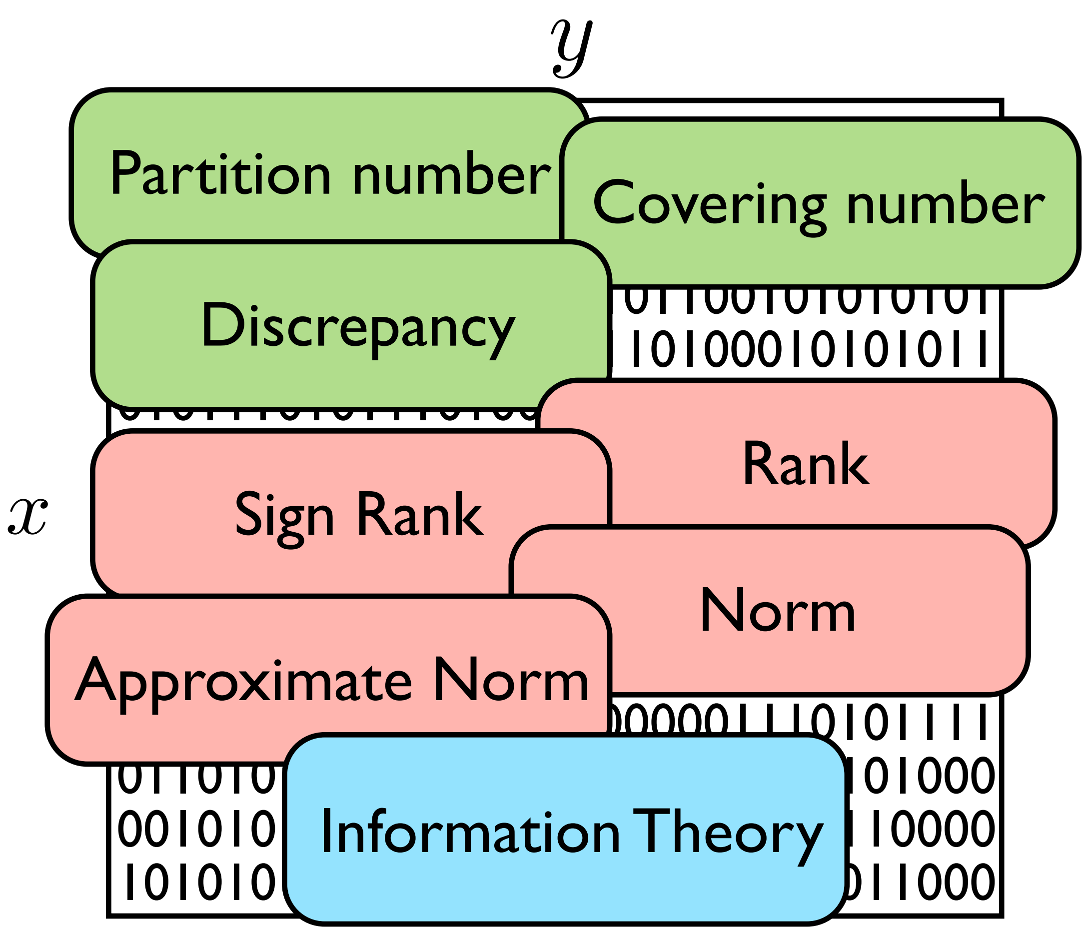
n = 3

$$M_{EQ} = \begin{array}{cc|cccccccc|} & & \multicolumn{8}{c}{y} \\ & & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ & 000 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 001 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 010 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ & 011 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ x & 100 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & 101 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ & 110 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ & 111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

2^n by 2^n matrix

The function matrix

How do you prove lower bounds on comm. complexity?



$$M_F[x,y] = F(x,y)$$

You study this matrix!

Take-Home Message

Communication complexity studies natural distributed tasks.

Communication complexity (lower bounds) has **<u>many</u>** interesting applications.

Lower bounds can be proved using a variety of tools: *combinatorial, algebraic, analytic, information theoretic,…*