CS 514: Advanced Algorithms II – Sublinear Algorithms **Rutgers: Fall 2021**

Lecture 8

October 26, 2021

Instructor: Sepehr Assadi

Topics of this Lecture

1 Introduction to Communication Complexity

Recall from Lecture 3 that we can use query complexity to prove lower bounds for sublinear *time* algorithms. In a similar spirit, we can use **communication complexity** to prove lower bounds on the *space* of streaming algorithms. We explore this approach in this lecture.

Suppose we have a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ (as before, one can consider different domains for the function, different ranges, relations instead of Boolean functions, or even functions with more than two arguments corresponding to *multi-party* communication models; for brevity, we stick to the most basic setting in this lecture). Function f defines a *communication problem* as follows: there are two players, say Alice and Bob, who get inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$, respectively, and their goal is to compute $f(x,y)$. For instance, in the *equality problem*, Alice and Bob would like to evaluate $EQ(x,y)$ which is defined as:

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

namely, determine whether or not their inputs are equal.

Considering neither Alice nor Bob has the entire input on their own, the players need to *communicate* with each other to determine the value of $f(x,y)$. The communication happens according to a *protocol* and is done as follows: Alice sends a single bit to Bob based solely on her input; after receiving this bit, Bob responds back with a single bit of his own which is a function of his input *and* the message of Alice; the players continue this throughout the protocol; we assume that the last bit communicated by any player is

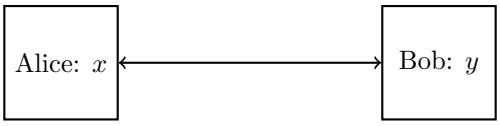

Figure 1: Alice and Bob computing $f(x, y)$

the output of the problem (again note that for simplicity, we assume Alice and Bob are sending alternating bits but one may also consider the case when players are communicating longer messages – we will discuss this later in this lecture).

The main measure of efficiency in this model is the communication cost of the protocols, defined as follows.

Definition 1 (**Communication cost**). *The communication cost of a protocol π, denoted by $||\pi||$, is the* worst-case *number of bits communicated between Alice and Bob in π over any choice of inputs x, y.*

Definition 2 (**Deterministic communication complexity**). *The* deterministic *communication complexity of a function f is defined as $D(f) = \min_\pi \|\pi\|$, where π ranges over all protocols that can solve f.*

Note that $D(f) = O(n)$ for any function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ as Alice can send all her n-bit input to Bob, while Bob responding with $n-1$ 0's between each one (to have the alternating bit property) and then Bob can output the final answer in another one bit.

Protocol Trees

An easy way of describing a protocol between Alice and Bob is by using a **protocol tree** (this should be compared and contrasted with decision trees in query complexity defined in Lecture 3).

The protocol tree for a protocol π is a *binary tree* defined as follows. Each node of the tree has a *subset* S of possible combination of x and y with the root having all possible combinations. We number the *levels* of the tree from 1 at root down to the leaf-nodes. Every node of the tree has either two or zero child-nodes; moreover, the edges going to child-nodes are labeled 0 for the left-child node and 1 with the right-child node.

At the root of the tree, the left-child node contains all (x, y) pairs such that Alice sends bit 0 for them and the right child-node contains the (x, y) pairs where Alice sends 1 for them. We continue this way so that whenever we are at a node N, the root-to-node path corresponds to the bits communicated so far, and the set S of this node contains all (x, y) that are consistent with these communicated bits. Finally, the leaf-nodes of the tree are such that for $(x, y) \in S$, $f(x, y)$ has the same value (thus the answer is now fixed and can be inferred directly from the transcript of the protocol).

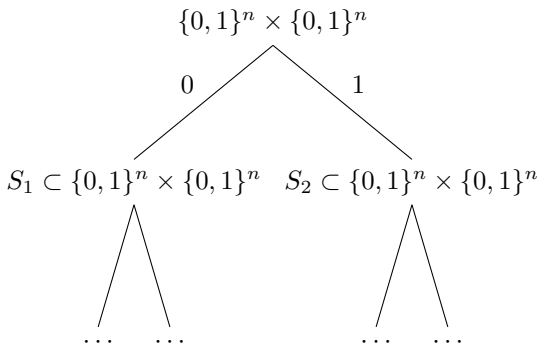

Figure 2: Example of a protocol tree

An extremely important property of communication protocols (and protocol trees) is the following so-called rectangle property.

Theorem 3 (Rectangle Property)**.** *For a node N in the protocol tree, the set $S \subseteq \{0,1\}^n \times \{0,1\}^n$ assigned to it forms a* combinatorial rectangle*, i.e., $S = X \times Y$ for $X, Y \subseteq \{0,1\}^n$ (as opposed to S being some arbitrary subset of $\{0,1\}^n \times \{0,1\}^n$). In other words,*

$$\begin{matrix}(x,y) \in S \\ (a,b) \in S\end{matrix} \implies \begin{matrix}(x,b) \in S \\ (x,a) \in S\end{matrix}.$$

Proof. The intuition behind the proof is that, by the definition of the protocol, any odd level node only partitions that set of inputs to Alice in its child-nodes into two parts based on the message of Alice, and any even level node do the same thing for the input of Bob.

More formally, we can prove this by induction on the levels of the tree. The theorem statement is clearly true for the root node. Let N be any node in the tree and suppose the set S of node N is a combinatorial rectangle by the induction hypothesis. Consider child-nodes N_0, N_1 of N (corresponding to bit 0 and 1, respectively), and their corresponding sets S_0 and S_1.

The set S_0 is obtained by taking all $(x, y) \in S$ such that the protocol sends 0 as the next step for them. Suppose Alice is the sender of this bit (i.e., N is an odd level node); the other case is symmetric. In this case, $S_0 \subseteq S$ consists of all $(x', y') \in S = X \times Y$ such that Alice sends 0 for them next. As the bit sent by Alice is only a function of x and the message communicated, we can define $X_0 \subseteq X$ as all x's for which Alice next sends bit 0 – all $y \in Y$ are still consistent with this extra bit. As such $S_0 = X_0 \times Y$, finalizing the proof for N_0. By symmetry this also holds for $S_1 = X_1 \times Y$, concluding the proof. □

2 Communication Complexity of the Equality Problem

Recall the equality function $EQ(x, y)$ from Eq (1) which checked whether or not the inputs x, y to Alice and Bob are equal. We will prove that the communication complexity of EQ problem is $\Omega(n)$.

Theorem 4. *The deterministic communication complexity of Equality is $D(EQ) \geq n$.*

Proof. Define the **communication matrix** of EQ as the following $2^n \times 2^n$ dimensional matrix $M := M^{EQ}$: the rows are indexed by inputs $x \in \{0,1\}^n$ to Alice, and the columns are indexed by the inputs $y \in \{0,1\}^n$ to Bob. The entry $M_{x,y} = EQ(x, y)$, i.e., the value of the function on inputs x, y.

It is easy to see that the communication matrix M of EQ is the identity matrix, $\mathbb{I}_{2^n}$ (e.g., the following matrix:

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

for $n = 2$.)

Now consider a protocol π for EQ. As the protocol tree of π is a binary tree with depth $||\pi||$, it can have at most $2^{||\pi||}$ many leaf-nodes. Moreover, by the rectangle property of communication protocols, the set assigned to each these leaf-nodes is a combinatorial rectangle. Moreover, by definition of the protocol tree, these combinatorial rectangles are *disjoint* and *monochromatic*, where the latter means that the value of f over them is fixed (either all 0 or all 1).

The above discussion then simply means that a protocol π will partition the entries of the matrix M into $2^{||\pi||}$ monochromatic combinatorial rectangles. On the other hand, note that no monochromatic combinatorial rectangle in M can contain two or more entries on the diagonal. Otherwise, for (x, y) and (a, b) in the diagonal mapped to the same rectangle, we have $EQ(x, y) = 1$ and $EQ(a, b) = 1$ but then $EQ(x, b) \neq 1$, hence this rectangle is not monochromatic. This in particular means that there is no way of partitioning the entries of M into monochromatic (combinatorial) rectangles, using less than 2^n rectangle. As such, we should have $2^{||\pi||} \geq 2^n$ which means $||\pi|| \geq n$, as desired. □

Remark. The techniques we used in proving the lower bound for $D(EQ)$ is quite general and can be used for many other problems f. Basically, "all" one has to do to bound $D(f)$, is to lower bound the number of disjoint monochromatic (combinatorial) rectangles that can partition all entries of the communication matrix M_f for f – the lower for $D(f)$ is then the logarithm of this number.

2.1 A Streaming Lower Bound for the Distinct Element Problem

We now use our lower bound on $D(EQ)$ from the last section to prove a lower bound on the space complexity of *deterministic* streaming algorithms for the distinct element problem from Lecture 7.

This problem was defined as follows.

Problem 1. Given a stream of n elements from the universe $[m]$, output the number of *distinct* elements in the stream, denoted by DE.

We prove that any deterministic algorithm cannot solve this problem exactly in the streaming setting, unless it uses $\Omega(\min\{n, m\})$ bits. This (almost) matches the trivial bound upper bounds that either store the entire stream or store one bit per each element in $[m]$ to check whether or not the element is visited.

Theorem 5. *Any deterministic streaming algorithm for computing* DE *exactly requires* $\Omega(\min\{n, m\})$ *bits.*

Proof. The proof comes from a reduction from Equality. Assume there is a deterministic streaming algorithm $\mathcal{A}$ for computing DE that only uses s bits of space.

Now consider an instance (x, y) of EQ problem. Define the sub-streams

$$\sigma_A := \{i : x_i = 1\} \qquad \text{and} \qquad \sigma_B := \{j : y_j = 1\},$$

and let $\sigma := \sigma_A \circ \sigma_B$ be the concatenation of these sub-streams. Note that while σ is a function of both x and y, σ_A only depends on the input of Alice and σ_B depends only on the input of Bob. Another important property is that when $EQ(x, y) = 1$, we have $\mathsf{DE}(\sigma) = \mathsf{DE}(\sigma_A) = \mathsf{DE}(\sigma_B)$, while when $EQ(x, y) = 0$, either $\mathsf{DE}(\sigma) \neq \mathsf{DE}(\sigma_A)$ or $\mathsf{DE}(\sigma) \neq \mathsf{DE}(\sigma_B)$.

Now consider the following protocol π for EQ based on $\mathcal{A}$. Alice generates σ_A and Bob generates σ_B (this can be done locally). Then, Alice runs $\mathcal{A}$ on σ_A and sends the content of the memory of the streaming algorithm to Bob. Additionally, Alice also sends $\mathsf{DE}(\sigma_A)$ to Bob. Bob continue running $\mathcal{A}$ on $\sigma = \sigma_A \circ \sigma_B$ by using the fact that $\mathcal{A}$ is a streaming algorithm and thus only needs its memory content on σ_A to proceed for the computation on σ_B. At the end, Bob gets the answer of $\mathcal{A}$ on σ and outputs the answer to $EQ(x, y)$ based on whether $\mathsf{DE}(\sigma) = \mathsf{DE}(\sigma_A) = \mathsf{DE}(\sigma_B)$ or not (recall that Alice also sends $\mathsf{DE}(\sigma_A)$ to Bob).

Using the above reduction, we obtain a deterministic protocol for EQ with communication cost $s + O(\log n)$ where s is the space complexity of $\mathcal{A}$. Since $D(EQ) \geq n$ by Theorem 4, we should have $s = \Omega(n)$, finalizing the proof. □

Remark. Again, the technique used in proving the lower bound for streaming algorithms in the above theorem based on communication complexity is quite general. Indeed, almost all streaming lower bounds are proven (or can be proven) through communication complexity lower bounds (although one may need to consider much more general communication models than we considered so far).

Remark. One can extend the lower bound above to prove that any deterministic algorithm that can even output a 1.1-approximation to DE requires $\Omega(n)$ space. In order to do this, consider a subset $\mathcal{F} \subseteq \{0,1\}^n$ such that for all $a, b \in \mathcal{F}$, if $a \neq b$, then the support of a and b are different in at least $0.2n$ entries. Using a simple probabilistic argument (or a combinatorial one), we can prove that size of $\mathcal{F}$ can be as large as $2^{\Omega(n)}$ (see Lemma 13).

We can then consider the $EQ(x, y)$ problem where $x, y \in \mathcal{F}$ instead of being arbitrary 0/1-strings and prove that its communication complexity is still $\Omega(n)$ (the proof is straightforward as communication complexity of EQ is $\Omega(\log |\mathcal{F}|)$ by our previous proof). Finally, we can show that using this promise variant of the equality problem, the reduction above works even for 1.1-approximation algorithms for DE and not necessarily only the exact ones. (It is a good exercise for the reader to formalize this proof.)

3 Randomized Communication Complexity

We now consider randomized communication complexity wherein Alice and Bob have access to random bits. There are two ways of introducing random bits to the communication model: the **private coin** model where Alice and Bob have access to separate sources of randomness on their own, and the **public coin** model, where players have access to a shared source of randomness. Similar to other settings, we require that a randomized protocol for a problem f to output the correct answer to $f(x, y)$ for any given x, y to Alice and Bob, with probability at least $2/3$ (or some other constant strictly more than half).

At first glance, the notion of public coin may sound rather strange. However, there are multiple reasons that motivate considering public coins in addition to (or even instead of) private coins. One particularly important reason is that public coin protocols are perhaps "mathematically nicer" to work with as they can be considered as distributions over deterministic protocols (and hence, among many other things, follow Yao's Minimax Principle of Lecture 3). However, can it be that by allowing public coins, we give "too much power" to the protocols? The answer, perhaps surprisingly at first, is *No*!

Theorem 6 (Newman's Theorem)**.** *Any public coin protocol π for a communication problem f with probability of success at least $1-\epsilon$ can be simulated by a private coin protocol θ such that $||\theta|| \leq ||\pi|| + O(\log n + \log 1/\delta)$ and θ outputs the correct answer to f with probability at least $1 - \epsilon - \delta$.*

Proof. Recall the following additive chernoff bound (or Hoeffding inequality). Given n independent random variables $Z_1, \ldots, Z_t \in [0, 1]$ and $\bar{Z} = \sum_i^t Z_i/t$, for any $\alpha > 0$, we have

$$Pr(\bar{Z} - E[\bar{Z}] > \alpha) \leq \exp\left(-2\alpha^2 t\right). \tag{2}$$

Let $\pi(x, y, r)$ denote the *deterministic* output of the protocol π on inputs x and y, and public randomness r. Define the indicator random variable $Z(x, y, r) \in \{0, 1\}$ which is 1 iff $\pi(x, y, r) \neq f(x, y)$, i.e., the protocol outputs errs in the answer. By the guarantee of the protocol π, we have that for any x, y:

$$\mathop{\mathbb{E}}_{r}[Z(x, y, r)] = \Pr(\pi \text{ errs on } (x, y)) \leq \epsilon.$$

Now, suppose we sample public coins $r_1, \ldots, r_t$ for $t = \lceil 2n/\delta^2 \rceil$. By additive Chernoff bound, we have that:

$$\begin{aligned}\Pr_{r_1,\ldots,r_t}\left(\frac{\sum_{i=1}^t Z(x, y, r_i)}{t} \geq \varepsilon + \delta\right) &\leq \Pr_{r_1,\ldots,r_t}\left(\frac{\sum_{i=1}^t Z(x, y, r_i)}{t} - \mathbb{E}\left[\frac{\sum_{i=1}^t Z(x, y, r_i)}{t}\right] \geq \delta\right) \\ &\leq \exp\left(-2\delta^2 t\right) \leq \exp\left(-4n\right).\end{aligned}$$

As such, by union bound over all choices of $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, we have,

$$\Pr_{r_1,\ldots,r_t}\left(\text{exists } (x, y) \text{ s.t. } \frac{\sum_{i=1}^t Z(x, y, r_i)}{t} \geq \varepsilon + \delta\right) \leq 2^{2n} \cdot \exp\left(-4n\right) < 1.$$

Thus, one can find a collection of t choices of public coins $r_1, \ldots, r_t$ such that $\frac{\sum_{i=1}^{t} Z(x,y,r_i)}{t} < \varepsilon + \delta$ for all possible inputs x, y. In the following we fix such choice of $r_1, \ldots, r_t$.

The protocol θ works as follows. Alice first *privately* samples $r' \in \{r_1, \ldots, r_t\}$ uniformly at random and sends it to Bob using $O(\log t) = O(\log n + \log(1/\delta))$ bits. The players then together run the *deterministic* protocol $\pi(x, y, r)$ from now on and output the same answer.

The communication cost of this protocol is clearly at most $O(\log n + \log(1/\delta))$ more than the communication cost of π. Moreover, for any input (x, y):

$$\Pr_{r'}(\theta(x,y,r') \text{ errs}) = \Pr_{r' \in \{r_1,\ldots,r_t\}}(\pi(x,y,r') \text{ errs}) = \frac{1}{t} \cdot \sum_{i=1}^{t} \Pr(\pi(x,y,r_i) \text{ errs}) = \frac{1}{t} \sum_{i=1}^{t} Z(x,y,r_i) \leq \varepsilon + \delta,$$

by the definition of $r_1, \ldots, r_t$. This concludes the proof. □

Remark. Newman's theorem can be seen as a very basic **pseudo-random number generator** (PRG): We were able to reduce the entire random bits needed by π to only $O(\log n + \log(1/\delta))$ bits (and thus communicate it between the players) at the cost of only paying an additive factor of δ in the algorithm. Note that aside from transforming public coins to private coins, this theorem also implies that any constant-error protocol can be made to work with only $O(\log n)$ bits of randomness.

Equipped with Newman's theorem, we can henceforth only focus on public coin protocols and present our definitions only for such protocols (and if needed, one can infer the results for private coin protocols by applying Newman's theorem and "pay" a minimal penalty in the bounds). In the following, by randomized protocols, we always mean public coin protocols (note that a public coin protocol does *not* need access to private coins in this setting).

Definition 7 (**Communication cost**). *The communication cost of a randomized protocol π is the* worst-case *number of bits communicated between Alice and Bob in π over any choice of inputs x, y and the randomness of the protocol.*

We can view a public coin protocol as a distribution over deterministic protocols obtained by first using the public coins to sample the deterministic protocol and then running the deterministic protocol on the input. As such, we can alternatively define the communication cost of π as the maximum communication cost of any deterministic protocol in the support of this distribution.

Definition 8 (**Randomized communication complexity**). *The* randomized *communication complexity of function f is defined as $R(f) = \min_\pi \|\pi\|$, where π ranges over all randomized protocols that can solve f with probability of success at least $2/3$.*

Similar to the query complexity setting, we also define a distributional version of communication complexity.

Definition 9. *Let μ be a distribution on inputs (x, y). We define the distributional communication complexity of f over distribution μ as $D_\mu(f) = \min_\pi ||\pi||$ where π ranges overall deterministic protocols that output the correct answer on inputs sampled from μ with probability at least $2/3$.*

Exactly as in the query complexity model, we also have the following implication of Yao's minimax principle.

Proposition 10 (Yao's minimax principle). *For any communication problem $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$,*

(i) $D_\mu(f) \leq R(f)$ for all input distributions μ; and,

(ii) $D_{\mu^\star}(f) = R(f)$ for some input distribution $\mu^\star$.

Again, Yao's minimax principle gives us a way of proving lower bounds for randomized protocols by instead considering deterministic ones on random inputs (we typically only use the first part which follows from a simple averaging argument – the second part implies that this approach can always gives us the tightest possible bound *if* we are able to find the "right" distribution $\mu^\star$).

3.1 A Detour: Randomized Communication Complexity of Equality

Before getting to proving lower bounds for randomized protocols, let us first see that how they can dramatically be stronger than the deterministic ones.

The following randomized protocol uses only 1 bit of communication (!) to solve the equality problem with probability of success 2/3. As such, $R(EQ) = O(1)$ in sharp contrast with the fact that $D(EQ) = \Omega(n)$.

Algorithm

1. Generate a public random string $a \in_R \{0,1\}^n$ uniformly at random.
2. Alice send $c = \sum_{i=1}^{n} a_i \cdot x_i \mod 2$ to Bob.
3. Bob also computes $c' = \sum_{i=1}^{n} a_i \cdot y_i \mod 2$, and output $\begin{cases} 0 & \text{with } prob. = 1/3 \\ \begin{cases} 1 & c' = c \\ 0 & o.w. \end{cases} & \text{with } prob. = 2/3 \end{cases}$

As we already show in Lecture 6, whenever $x \neq y$,

$$\Pr_{a \sim \{0,1\}^n} \left(\sum_{i=1}^{n} a_i \cdot x_i = \sum_{i=1}^{n} a_i \cdot y_i \mod 2 \right) = \frac{1}{2}.$$

Moreover, whenever $x = y$, we clearly have $c' = c$. As such, when $x = y$, the protocol is correct with probability 2/3 and when $x \neq y$, the protocol is correct with probability $1/3 + 2/3 \cdot 1/2 = 2/3$.

Remark. Recall that by using Newman's theorem, we can turn the above protocol for EQ to a one that only uses private coins and has communication cost $O(\log n)$. One can also design such a protocol directly by considering the input of each player as number between 0 and 2^n in the binary representation and compare them mod some $\Theta(\log n)$-bit large random prime (this is left as an exercise for the reader). It is also worth mentioning that one can prove an $\Omega(\log n)$ bit lower bound on the communication cost of any *private* coin protocol for the equality problem. This in turn implies the tightness of Newman's theorem (for constant δ).

4 One-Way Communication Complexity: The Index Problem

The equality problem should act as a reminder that proving lower bound for randomized protocols can be considerably more challenging than deterministic ones simply because they are inherently much stronger. In this section, we prove a randomized communication complexity lower bound for one of the most important communication problems in the context of streaming algorithms, the **Index** problem.

Problem 2. In the index communication problem Ind, Alice gets a string $x \in \{0,1\}^n$ and Bob gets an index $i \in [n]$; the goal for the players is to output x_i, i.e., $Ind(x, i) = x_i$.

Before studying the communication complexity of this problem, let us see how it can be used to prove streaming lower bounds for the distinct element problem.

Theorem 11. *Any randomized streaming algorithm for computing* DE *exactly, requires* $\Omega(R(Ind) - \log n)$ *bits of space, where Ind is the index problem on domain* $\{0,1\}^n$.

Proof. Assume there is a randomized streaming algorithm $\mathcal{A}$ for computing DE that uses s bits of space. Now consider an instance (x, i) of Ind problem. Define the sub-streams

$$\sigma_A := \{i : x_i = 1\} \qquad \text{and} \qquad \sigma_B := \{i\}\,,$$

and let $\sigma := \sigma_A \circ \sigma_B$ be the concatenation of these sub-streams. Note that while σ is a function of both x and y, σ_A only depends on the input of Alice and σ_B depends only on the input of Bob. Another important property is that when $Ind(x, i) = 1$, we have $\mathsf{DE}(\sigma) = \mathsf{DE}(\sigma_A)$, while when $Ind(x, i) = 0$, $\mathsf{DE}(\sigma) = \mathsf{DE}(\sigma_A) + 1$.

Now consider the following protocol π for Ind based on $\mathcal{A}$. Alice generates σ_A and Bob generates σ_B (this can be done locally). Then, Alice runs $\mathcal{A}$ on σ_A and sends the content of the memory of the streaming algorithm to Bob. Additionally, Alice also sends $\mathsf{DE}(\sigma_A)$ to Bob. Bob continues running $\mathcal{A}$ on $\sigma = \sigma_A \circ \sigma_B$ by using the fact that $\mathcal{A}$ is a streaming algorithm and thus only needs its memory content on σ_A to proceed for the computation on σ_B. At the end, Bob gets the answer of $\mathcal{A}$ on σ and outputs the answer to $Ind(x, i)$ based on whether $\mathsf{DE}(\sigma) = \mathsf{DE}(\sigma_A)$ or not.

Using the above reduction, we obtain a randomized protocol for Ind with communication cost $s + O(\log n)$ where s is the space complexity of $\mathcal{A}$, finalizing the proof. □

The above theorem suggests that we can prove a lower bound for space complexity of distinct element by lower bounding $R(Ind)$ instead. Alas, it is easy to see that even $D(Ind) = O(\log n)$ as Bob can simply sends his index to Alice and Alice solves the problem. This will not allows us to prove any meaningful lower bound for the streaming distinct element problem.

But before entirely giving up, let us re-examine the protocol π obtained in the reduction of Theorem 11. This protocol did not involve Bob communicating with Alice at all – only Alice sent a single message to Bob and then Bob output the solution. Such a protocol is called a **one-way protocol**. As such, as long as we can lower bound randomized communication complexity of such protocols, we can still apply Theorem 11 to get a "good" lower bound (note that in particular, the $O(\log n)$ communication protocol described above is not a one-way protocol with only Alice speaking).

One-way Communication Complexity. Let us formalize the discussion above. In the one-way communication model, we only allow Alice to send a single message to Bob and Bob then needs to output the answer. We again define the communication cost of the protocol as the bit-length of the message of Alice.

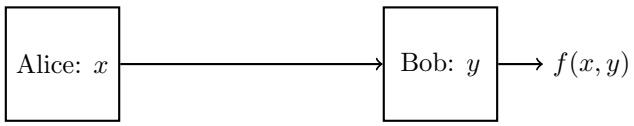

Figure 3: One way communication.

We can then define:

- One-way deterministic communication complexity – $\overrightarrow{D}(f) := \min_\pi ||\pi||$ where π ranges over all deterministic *one-way* protocols for f;
- One-way randomized communication complexity – $\overrightarrow{R}(f) := \min_\pi ||\pi||$ where π ranges over all randomized *one-way* protocols for f with probability of success at least $2/3$;
- One-way distributional communication complexity – $\overrightarrow{D}_\mu(f) := \min_\pi ||\pi||$ where π ranges over all deterministic *one-way* protocols for f with probability of success at least $2/3$ over the distribution μ;

Let us examine the one-way communication complexity of Ind. It is easy to prove that $\overrightarrow{D}(Ind) \geq n$ as follows: By pigeonhole principle, if the message of Alice has size less than n, then at least two different

strings $x \neq y \in \{0,1\}^n$, are mapped to the same message M. Now let i be an index where $x_i \neq y_i$. The output of Bob is a deterministic function of M and i and is thus wrong for one of x_i and y_i, a contradiction.

This argument however does not work for randomized algorithms when the answer is allowed to be wrong with probability $1/3$ (in fact, it is easy to see that $\overrightarrow{R}(Ind) \leq 2n/3$). In the following, we are going to use a more general argument to prove that $\overrightarrow{R}(Ind) = \Omega(n)$.

Theorem 12. *The one-way randomized communication complexity of Index is* $\overrightarrow{R}(Ind) = \Omega(n)$.

Before getting to the proof of this theorem, we need to first prove an auxiliary lemma about a construction of "large" family of strings in $\{0,1\}^n$ that are "far from" each other. Let us define the **Hamming distance** between two strings x, y as the number of indices they differ from each other, i.e.,

$$\Delta(x,y) = |\{i \mid x_i \neq y_i\}|.$$

Lemma 13. *For any parameter $\delta \in (0,1/4)$, there exists a subset $\mathcal{F} \subseteq \{0,1\}^n$ with size $\exp\left((\delta^2/4)\cdot n\right)$ such that for any two $x \neq y \in \mathcal{F}$, $\Delta(x,y) > n/2 - \delta \cdot n$.*

Proof. The proof is by probabilistic argument. Consider sampling two strings x, y independently and uniformly at random from $\{0,1\}^n$. Then,

$$\mathbb{E}\left[\Delta(x,y)\right] = \sum_{i=1}^{n} \Pr\left(x_i \neq y_i\right) = n \cdot (1/2) = n/2.$$

Moreover, $\Delta(x,y)$ is a sum of n independent random variables (one for each coordinate in $[n]$) and thus, by Chernoff bound,

$$\Pr\left(\Delta(x,y) \leq n/2 - \delta\cdot n\right) \leq \Pr\left(|\Delta(x,y) - \mathbb{E}\left[\Delta(x,y)\right]| \geq \delta \cdot n\right) \leq \exp\left(-\frac{\delta^2}{2}\cdot n\right).$$

Let $t := \exp\left((\delta^2/4)\cdot n\right)$ and suppose we sample strings $x_1, \ldots, x_t$ uniformly to form $\mathcal{F}$. By union bound,

$$\Pr\left(\text{exists } i \neq j \text{ with } \Delta(x_i,x_j) \leq n/2 - \delta\cdot n\right) \leq \binom{t}{2}\cdot \exp\left(-\frac{\delta^2}{2}\cdot n\right) < t^2 \cdot \exp\left(-\frac{\delta^2}{2}\cdot n\right) = 1.$$

As such, there exists a choice of $x_1, \ldots, x_t$ that satisfy the bounds above.

Note. An alternative proof for this lemma is as follows. Greedily pick any x from $\{0,1\}^n$ and include it in $\mathcal{F}$; remove all strings z from $\{0,1\}^n$ with $\Delta(x,z) < n/2 - \delta n$; recurse on the *remaining* strings in $\{0,1\}^n$. Considering the number of strings that can be removed in each step is at most $\binom{n}{n/2-\delta n+1}$, we obtain that

$$|\mathcal{F}| \geq \frac{2^n}{\binom{n}{n/2-\delta n+1}};$$

some calculation to simplify the above bound then allows us to prove the lemma (with similar bounds). □

Remark. The family of strings constructed in Lemma 13 and its many variants and generalizations is extremely useful in various problems in TCS and mathematics (two entirely unrelated applications of such family are discussed in this lecture alone). They are also closely related to some fundamental problems such as *error correcting codes* or *combinatorial design*.

Proof of Theorem 12. We prove the lower bound for protocols of Ind that output the correct answer with probability of success at least 0.9 as opposed to 2/3; this is without loss of generality as the communication cost of these protocols are within a constant factor of "standard" protocols by simply running the 1/3-error protocol in parallel $O(1)$ time independently and taking the majority value.

We will use Yao's minimax principle in part (i) of Proposition 10 by showing that there is a distribution μ where $\overrightarrow{D}_{\mu}(Ind) = \Omega(n)$ where with a slight abuse of notation, we interpret $\overrightarrow{D}_{\mu}(Ind)$ here as the distributional complexity of protocols over μ that output the correct answer with probability 0.9 as opposed to 2/3 (note that in general, unlike randomized communication complexity which only changes by a constant factor if we change the required probability of success by a constant, the change in the distributional communication complexity can be unbounded because we can no longer rely on boosting the probability of success).

The distribution μ is as follows. Let $\mathcal{F}$ be the family of strings in Lemma 13 for parameter $\delta = 0.1$ and thus for all $x, y \in \mathcal{F}$, $\Delta(x, y) > 0.4 \cdot n$ and $|\mathcal{F}| = 2^{\Omega(n)}$. In the distribution μ, we sample $x \in \mathcal{F}$ uniformly at random and give it to Alice, and sample $i \in [n]$ uniformly at random and independently and give it to Bob.

Now consider any deterministic one-way protocol π for Ind on the distribution μ with probability of success at least 2/3. By definition,

$$\Pr_{x \in \mathcal{F}} \Pr_{i \in [n]} (\pi \text{ errs on input } (x, i)) \leq 0.1;$$

Let $\mathcal{F}'$ be the subset of $\mathcal{F}$ where for any $x \in \mathcal{F}'$, π errs on at most 0.2 choices of index i, i.e.:

$$\mathcal{F}' := \left\{ x \in \mathcal{F} \mid \Pr_{i \in [n]} (\pi \text{ errs on input } (x, i)) \leq 0.2 \right\}.$$

We have that $|\mathcal{F}'| \geq |\mathcal{F}|/2$ as otherwise (note that choice of i and x are independent and both are uniform over their support):

$$\begin{aligned}
\Pr_{x \in \mathcal{F}} \Pr_{i \in [n]} (\pi \text{ errs on input } (x, i)) &= \Pr_{x \in \mathcal{F}'} \Pr_{i \in [n]} (\pi \text{ errs on input } (x, i)) + \Pr_{x \in \mathcal{F} \setminus \mathcal{F}'} \Pr_{i \in [n]} (\pi \text{ errs on input } (x, i)) \\
&> \frac{|\mathcal{F} \setminus \mathcal{F}'|}{|\mathcal{F}|} \cdot (0.2) && \text{(by the definition of } x \in \mathcal{F} \setminus \mathcal{F}') \\
&> \frac{1}{2} \cdot (0.2) = 0.1,
\end{aligned}$$

which contradicts the first equation above.

For any $x \in \mathcal{F}'$, let $M(x)$ denote the message sent by Alice to Bob when her input was x (note that $M(x)$ is a deterministic function of x). We claim that given only $M(x)$ for any $x \in \mathcal{F}'$, we can recover the string x entirely (this is *not* necessarily true for all $x \in \mathcal{F}$). The proof is as follows.

Given $M(x)$, define the vector $y(x) \in \{0, 1\}^n$ such that for any $i \in [n]$, $y_i := \pi(x, i)$, i.e., the output of the protocol when the input to Alice is x and input to Bob is i. Note that $y(x)$ is a deterministic function of $M(x)$ only. Moreover, we have (recall that Δ is the Hamming distance):

$$\Delta(y(x), x) \leq 0.2 \cdot n.$$

by the definition $\mathcal{F}'$ as only 0.2 fraction of entries of $y(x)$ can be different from x. On the other hand, for any $z \in \mathcal{F}$ (and thus $z \in \mathcal{F}'$), we have,

$$\begin{aligned}
\Delta(y(x), z) &\geq \Delta(z, x) - \Delta(y(x), x) && \text{(by the triangle inequality as } \Delta(\cdot, \cdot) \text{ is a distance metric)} \\
&> 0.4 \cdot n - 0.2 \cdot n > 0.2 \cdot n \geq \Delta(y(x), x).
\end{aligned}$$

As such, given $y(x)$, we can simply find the string $w \in \mathcal{F}$ which minimizes $\Delta(y(x), w)$; by the two equations above, this string has to be x, thus allowing us to recover x from $y(x)$. Since $y(x)$ itself was a deterministic function of $M(x)$, we obtained a *one-to-one* mapping from $x \to M(x)$ for all $x \in \mathcal{F}'$. This implies that there needs to be $|\mathcal{F}'| = 2^{\Omega(n)}$ *different* messages sent by Alice, which means length of her message needs to be $\Omega(n)$ bits, concluding the proof. $\square$

4.1 Detour: A Lower Bound for Index via Information Theory

Let us give an alternative proof of Theorem 12 using information theory tools for the interested readers and to show case how much these tools can prove streaming lower bounds easier.

Information theory tools. For a random variables X, Y, $\mathbb{H}(X)$ denotes the *Shannon entropy* of X and $\mathbb{H}(X \mid Y) = \mathbb{E}_{y \in Y} \mathbb{H}(H \mid Y = y)$ is the conditional entropy. We use the following properties of entropy:

(*i*) $0 \leq \mathbb{H}(X) \leq \log(|\text{support}(X)|)$ and the right equality holds iff X is uniform over its support.

(*ii*) Chain rule of entropy: $\mathbb{H}(X, Y) = \mathbb{H}(X) + \mathbb{H}(Y \mid X)$

(*iii*) Entropy is subadditive: $\mathbb{H}(X, Y) \leq \mathbb{H}(X) + \mathbb{H}(Y)$.

(*iv*) Fano's inequality: If X is a binary random variable, and there is an estimator random variable Y and function g such that $g(Y) = X$ with probability at least $1 - \delta$ for $\delta < 1/2$, then

$$\mathbb{H}(X \mid Y) \leq H_2(\delta) := \delta \cdot \log(1/\delta) + (1 - \delta) \cdot \log(1/(1 - \delta)).$$

The lower bound for Index. Consider the uniform distribution over $\{0,1\}^n$ for x and uniform (and independent) distribution over $[n]$ for i. Also, consider any one-way deterministic protocol π for Index over this distribution with probability of success $1 - \delta$ for $\delta < 1/2$.

Let X, M, and I denote the random variables for input of Alice, message of Alice, and index i of Bob. Moreover, let $\Theta = X_I$ denote the correct answer to the problem. By Fano's inequality, and since π can find the value of Θ given only M and I with probability at least $1 - \delta$, we have,

$$\mathbb{H}(\Theta \mid M, I) \leq H_2(\delta). \tag{3}$$

On the other hand, by the definition of conditional entropy,

$$\mathbb{H}(\Theta \mid M, I) = \mathop{\mathbb{E}}_{i \in [n]} [\mathbb{H}(\Theta \mid M, I = i)] = \mathop{\mathbb{E}}_{i \in [n]} [\mathbb{H}(X_i \mid M, I = i)],$$

as conditioned on $I = i$, the correct answer to the problem is X_i. Now we are going to use the fact that I is chosen independent of X and thus the one-way message M, i.e., $X, M \perp I$. This implies that the joint distribution of (X_i, M) is independent of the event $I = i$ (note that we are talking about X_i and not X_I – there is no randomness in choice of i in X_i, and the only randomness is in X). As such, we can remove the conditioning on the event $I = i$ above. This implies:

$$\begin{aligned}
\mathbb{H}(\Theta \mid M, I) &= \mathop{\mathbb{E}}_{i \in [n]} [\mathbb{H}(X_i \mid M)] = \frac{1}{n} \cdot \sum_{i=1}^{n} \mathbb{H}(X_i \mid M) && \text{(as the distribution of } i \text{ is uniform over } [n]) \\
&\geq \frac{1}{n} \cdot \mathbb{H}(X \mid M) && \text{(by subadditivity of entropy)} \\
&= \frac{1}{n} \cdot (\mathbb{H}(X, M) - \mathbb{H}(M)) && \text{(by chain rule)} \\
&= \frac{1}{n} \cdot (\mathbb{H}(X) + \mathbb{H}(M \mid X) - \mathbb{H}(M)) = \frac{1}{n} \cdot (\mathbb{H}(X) - \mathbb{H}(M)) \\
&\quad \text{(by chain rule of entropy and since } \mathbb{H}(M \mid X) = 0 \text{ as } M \text{ is deterministic conditioned on } X) \\
&\geq \frac{1}{n} \cdot (n - ||\pi||) \\
&\quad \text{(as } \mathbb{H}(X) = n \text{ since } X \text{ is uniform over } 2^n \text{ strings and by property } (i) \text{ of entropy above)} \\
&= 1 - \frac{||\pi||}{n}.
\end{aligned}$$

Plugging in this bound in the Eq (3), we get that:

$$||\pi|| \geq (1 - H_2(\delta)) \cdot n,$$

which is $\Omega(n)$ for any constant $\delta < 1/2$ by the definition of $H_2(\delta)$. Thus, we proved that the distributional communication complexity of Index is $\Omega(n)$ on the uniform distribution. The lower bound for randomized protocols now follows from the easy direction of Yao's minimax principle. □

This concludes our lecture. We refer the interested reader to the excellent textbooks by Kushilevitz and Nisan [KN97], and by Rao and Yehudayoff [RY20] on the basics of communication complexity, and by Roughgarden [Rou16] for implications of communication complexity in proving lower bounds for different algorithms. An excellent introduction to information theory and proof of the properties above can be found in Chapter 2 of the textbook by Cover and Thomas [CT06].

References

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006. 12

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997. 12

[Rou16] Tim Roughgarden. Communication complexity (for algorithm designers). *Foundations and Trends in Theoretical Computer Science*, 11(3-4):217–404, 2016. 12

[RY20] Anup Rao and Amir Yehudayoff. *Communication complexity and applications*. Cambridge University Press, 2020. 12