# CMSC 858M: Algorithmic Lower Bounds: Fun with Hardness Proofs
# Fall 2020

**Instructor:** Mohammad T. Hajiaghayi
**Scribe:** Kiarash Banihashem

## 1 Overview

This chapter considers straming algorithms which are very similar to online algorithms. In both cases, the algorithm needs to make decisions before it sees all the data. Unlike online algorithms however, the streaming algorithm can defer making its decisions and it the focus of designing streaming algorithms is maintaining low memoery.
The book first introduces the tool of communication complexity. The most important problem in this tool is the INDEX problem. In this problem, there are two agents Alice and Bob. Alice has access to an array of binary values $X = [X_1, \dots, X_n]$ and Bob has an index $i \in \{1, \dots, n\}$. The array is only known to Alice and the index is only known to Bob. Alice wants to send some information to Bob such that Bob can determine $X_i$.
A naive way to do this is to send the entire array $X$. As the chapter shows, there is really no smarter way to solver the problem, in other words, any communication protocol for solving this problem requires $\mathcal{O}(n)$ bits to be sent by Alice.

As we will see, this simple problem allows us to prove lower bounds for a wide class of streaming algorithms. The main idea in the proofs is that in many streaming problems, the input seen until some point may be both insufficient to solve the problem and have enough information such that the problem cannot be solved with a compressed version of it. This allows us to build reductions from the INDEX problem such that the partially seen input encodes the array $X$ and the unseen part of the input determines the index $i$.

## 2 Introduction to streaming algorithms

This section formally defines streaming algorithms.

**Definition 1** *Streaming Algorithms are algorithms for processing data streams in which the input is presented as a sequence of items and can be examined in only a few passes (typically just one) by using relatively little memory (much less than the input size), and also limited processing time per item. Often produce approximate answers based on a summary or "sketch" of the data stream in memory. A common technique for creating a "sketch" is sampling at random.*

As an example, assume that the input is going to be all of the integers in the set $\{1, \dots, n\}$ except for one integer that will not appear in the input sequence. We want to design a streaming algorithm that outputs this number. A simple algorithm that needs only $O(\log(n))$ space is to keep the running sum $s$ of the items seen so far. The output can then be calculated via $\frac{n(n+1)}{2} - s$. This is a large improvement over the $O(n)$ space naive algorithms that keep track of the numbers seen so far via a binary array or hashset.

The book lists four important streaming problems. In all of these problems, the goal is often to solve them approximately. An algorithm is said to be $(\epsilon, \delta)$-accurate if it achieves an error less than $\epsilon$ with probability at least $1 - \delta$. The mentioned problems are as follows.

1. For a given value of $k$, evaluate the $k$th frequency moment: $F_k(a) = \sum_{i=1}^{n} a_i^k$ where $a_i$ denotes the number of times an element has appeared.
2. Find heavy hitters, that is elementes that appear with frequency $a_i > T$.
3. Counte the number of distinct elements.
4. Calculate the entropy $E(a) = \sum_{i=1}^{n} \frac{a_i}{m} \log(\frac{a_i}{m})$ where $m = \sum a_i$.

# 3 Streaming for Graph Algorithms

For graph algorithms, there are two main versions of streaming considered.

1. **Insertion only**. In this version, the edges are added to the graph over time.
2. **Dynamic**. In this version, the edges are added to the grapph but can also be deleted.

# 4 Maximum matching

This section provides semi-streaming algorithms for maximum matching.

**Definition 2** *Maximum Matching (MM) Let $G = (V, E)$ be a graph. Th maximum matching problem is finding the largest set of disjoint edges in $G$.*

**Theorem 1** *There is a semi-streaming algorithm for MM that* ***(a)*** *decides whether or not an edge* $e$ *is in the matching as soon as* $e$ *is added to the graph ,* ***(b)*** *only requires one pass over the data,* ***(c)*** *uses at most* $\widetilde{O}(n)$ *space and* ***(d)*** *has approximation gaurantee* $\frac{1}{2}$.

The proof of the above theorem is provided in the proof. The algorithm they consider adds a new edge $e$ to the matching if and only if none of its endpoints are already present in an edge of the running matching. To keep track of these vertices, it is sufficient to keep an array of vertices and mark the endpoints of each edge whenver it is addded to the matching.

Several improved results are known for this problem.

1. There is a one-pass semi-streaming algorithm for MM in bipartite graphs that uses $\widetilde{O}(n)$ space and has approximation factor $0.6$.
2. There is a one-pass semi-streaming algorithm for MM in general graphs that requires $\widetilde{O}(n)$ space and has approximation factor $0.545$.

# 5 Communication complexity

This section introduces the notion of communication complexity which will be used to prove lower bounds for streaming. In the problems considered in the section, there are two agents Alice and Bob that wish to work together to solve a problem instance. The difficulty in doing so however is that each of them has access to only a part of the input and as such, the need to communicated with each other to obtain a solution. To do so, they are free to design a communication protocol which specifies what information each party should send and in what manner. Often, we will restrict these protocols and e.g, require them to be one-way from Alice to Bob, i.e, only Alice can send information to Bob and not the other way round.
The section introduces the following important problems and provides lower bounds on their *communication complexity*, that is, the maximum number of bits any protocol may need in the worst case.

**Definition 3 (INDEX)** *Alice has a string* $x \in \{0,1\}^n$ *and bob has a natural number* $i \in [n]$. *Alice can communicate to Bob in a one-way model so that Bob can find* $x_i$.

**Definition 4 (INDEXSAME)** *Alice has a string* $x \in \{0,1\}^n$ *and bob has a natural number* $i \in [n-1]$. *Alice can communicate to Bob in a one-way model so that Bob can determine whether* $x_i = x_{i+1}$.

**Definition 5 (DISJ)** *Alice has a string* $x \in \{0,1\}^n$ *and bob has a string* $y \in \{0,1\}^n$. *They want to know if the sets represented by* $x, y$ *are disjoint, i.e, if* $\neg x_i \vee \neg y_i$ *holds for all* $i$. *The communication is 2-way and there are no restrictions on the number of communication rounds.*

The next theorem proves lower bounds for these problems which will be useful for showing hardness results for streaming algorithms.

**Theorem 2**

*INDEX requires* $\Omega(n)$ *bits. The lower bound holds even for randomized algorithms that have the correct output with probability* $\geq \frac{2}{3}$.

*INDEXSAME requires* $\Omega(n)$ *bits. The lower bound holds even for randomized algorithms that have the correct output with probability* $\geq \frac{2}{3}$.

*DISJ requires* $\Omega(n)$ *bits both in the deterministic and randomized cases with success probability* $\geq \frac{2}{3}$. *It also holds when input is gauranteed to satisfy* $\sum x_i = \sum y_i = \lfloor \frac{n}{4} \rfloor$.

# 6 Lower Bounds on Graph Streaming Problems

In this section, the book provides lower bounds on several graph streaming problems which we describe below.

**Definition 6** *Max-Conn-comp(k) Given a forest* $G = (V, E)$*, we want to determine whether there is a fully connected component of size* $\geq k$. *Note that* $k$ *is not part of the input; rather, it is a parameter that specifes the problem.*

**Theorem 3** *For* $k \geq 3$*, any single pass algorithm for Max-Conn-comp(k) requires* $\Omega(n)$ *space.*

**Definition 7** *Is-Tree Given a graph* $G = (V, E)$*, we want to determine whether the graph is a tree.*

**Theorem 4** *Any single pass algorithm for Is-Tree requires* $\Omega(n)$ *space.*

**Definition 8** *Perfect-Matching Given a graph* $G = (V, E)$*, we want to determine whether the graph has a perfect matching, i.e, a disjoint set of edges that covers the graph.*

**Theorem 5** *Any single pass algorithm for Perfect-Matching requires* $\Omega(m) = \Omega(n^2)$ *space.*

**Definition 9** *Shorest-path Given an unweighted graph* $G = (V, E)$ *and two vertices* $v, w$*, what is the length of the shortest path from* $v$ *to* $w$*?*

**Theorem 6** *Any single pass algorithm that approximates Shorest-Path with a factor better than*

In adition, the book provides a lower bound for the Frequency Moments problem outlined below.

**Definition 10** *Given a data stream of numbers $y_1, \dots, y_n \in [m]$, define the frequency of each $k \in [m]$ as $x_k = |\{j : y_j = k\}|$ and define the frequency vector as $x = (x_1, \dots, x_m)$. For $p \in \mathcal{N} \cup \{\infty\}$, define the $p$th frequency moment as*

$$F_p = \begin{cases} \sum x_i^p & \text{if } p \in \mathcal{N} \\ \max_i x_i & \text{if } p = \infty \end{cases}$$

**Theorem 7** *Let $p > 2$.*

1. *There exist a randomized streeaming algorithm which approximates $F_p$ with a factor of $1+\epsilon$ and requires $\mathcal{O}(m^{1-2/p})$ space.*
2. *Any randomized streaming algorithm that $1+\epsilon$ approximates $F_p$ requires at least $\Omega(m^{1-2/p})$ space.*

# 7 Extra problems

1. [9] show consider the problem of monotone submodular maximization under a cardinality constraint. In this problem, there is a ground set $V$ and a function $f : 2^V \to \mathbb{R}^+$ such that it is monotone, i.e, $f(A) \geq f(B)$ for all $B \subseteq A$ and submodular, i.e, $f(A \cup \{x\}) - f(A) \leq f(B \cup \{x\}) - f(B)$ for all $B \subseteq A$ and $x \notin A$.
   They show that any algorithm that has an approximation factor $\alpha > 0.5$ for this problem in the streaming setting requires $\Omega(n/k)$ memory.
2. [4] consider the problem of dynamic submodular maximization where the goal is to maintain a solution to the optimization problem subject to both insertions and deletions. They show that for any $\epsilon \geq 0$, a $\frac{1}{2} + \epsilon$ approximation algorithm requires at least $n^{\widetilde{\Omega}(\epsilon)}/k^3$ amorotized queries to the submodular oracle in expectation.
3. [10] study low rank approximation in the streaming model in which the rows of an $n \times d$ matrix A are presented one at a time in an arbitrary order. In the end the algorithm needs to output a $k \times d$ matrix $R$ satisfying $\|A - AR^\dagger R\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$. They show a space lower bound of $\Omega(dk/\epsilon)$ bits for the problem.
4. [5] consider a variety of Numerical Linear Algebra problems in the Streaming Model. They provide upper and lower boudns on the space complexity of one-pass algorithms. In what follows, $A$ is an $n \times d$ matrix, $B$ is an $n \times d'$ matrix and $c = d + d'$ and the input is assumed to be integers of $\mathcal{O}(\log(nc))$ bits or $\mathcal{O}(\log(nd))$ bits.
   (a) For outputing a matrix $C$ such that $\|A^T B - C\| \leq \epsilon \|A\| \cdot \|B\|$, they show that $\Theta(c\epsilon^{-2}\log(nc))$ space is needed.
   (b) For $d' = 1$, i.e, when $B$ is a vector $b$, finding an $x$ such that $\|Ax - b\| \leq (1+\epsilon)\min_{x' \in \mathbb{R}^d} \|Ax' - b\|$ requires $\Theta(d^2\epsilon^{-1}\log(nd))$ space.

5. [2] consider the gap cycle counting problem in the streaming model. The edges of a 2-regular n-vertex graph G are arriving one-by-one and it is guaranteed that G is a disjoint union of either k-cycles or 2k-cycles for some small k. The goal is to determine which of these two cases has happend. They show that any $p$-pass streaming algorithm requires $n^{1-1/k^{\Omega 1/p}}$ space.

6. [3] show that two-pass graph streaming algorithm for the s-t reachability problem in n-vertex directed graphs requires near-quadratic space of $n^{2-o(1)}$ bits.

7. [6] consider the The approximate null vector problem where given $x_1, \dots x_{d-1}$ vectors in $\mathbb{R}^d$, the goal is to output a vector that is approximately orthogonal to all of them. They show that the problem has an $\Omega(d^2)$ lower bound.

8. [7] consider the maximum matching problem. They show that any single pass algorithm cannot achieve better than $2/3$ approximation. There have been improvements to the bound since this work and most recently, [8] showed a $\frac{1}{1+\ln 2}$ bound.

9. [1] consider approximating the maximum matching problem for two pass algorithms and show that any such algorithm has approximation ratio at least $1 - \Omega(\frac{\log RS(n)}{\log n})$ where $RS(n)$ denotes maximum number of disjoint induced matchings of size $\theta(n)$ in any n-vertex graph.

# References

[1] S. Assadi. A two-pass lower bound for semi-streaming maximum matching. *arXiv preprint arXiv:2108.07187*, 2021.

[2] S. Assadi, G. Kol, R. R. Saxena, and H. Yu. Multi-pass graph streaming lower bounds for cycle counting, max-cut, matching size, and other problems. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 354–364. IEEE, 2020.

[3] S. Assadi and R. Raz. Near-quadratic lower bounds for two-pass graph streaming algorithms. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 342–353. IEEE, 2020.

[4] X. Chen and B. Peng. On the complexity of dynamic submodular maximization. *arXiv preprint arXiv:2111.03198*, 2021.

[5] K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 205–214, 2009.

[6] Y. Dagan, G. Kur, and O. Shamir. Space lower bounds for linear prediction in the streaming model. In *Conference on Learning Theory*, pages 929–954. PMLR, 2019.

[7] A. Goel, M. Kapralov, and S. Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 468–485. SIAM, 2012.

[8] M. Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1874–1893. SIAM, 2021.

[9] A. Norouzi-Fard, J. Tarnawski, S. Mitrovic, A. Zandieh, A. Mousavifar, and O. Svensson. Beyond 1/2-approximation for submodular maximization on massive data streams. In *International Conference on Machine Learning*, pages 3829–3838. PMLR, 2018.

[10] D. Woodruff. Low rank approximation lower bounds in row-update streams. *Advances in neural information processing systems*, 27, 2014.