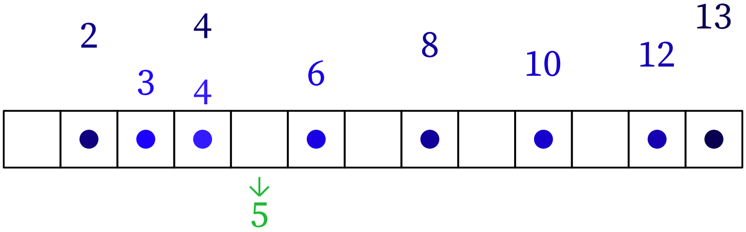
Streaming Algorithms for the Missing Item Finding Problem

Manuel Stoeckl

Department of Computer Science, Dartmouth College*

Symposium on Discrete Algorithms 2023



*This work was supported in part by the National Science Foundation under award 2006589.

Overview

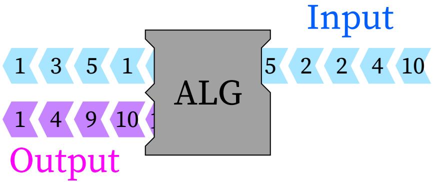
About models for streaming algorithms

- Setting & type of randomness
- Missing Item Finding
- Basic results
- An open question

Proof and algorithm sketches

Setting of a streaming algorithm [Ben-Eliezer, Jayaram, Woodruff, and Yogev 2020]

Static setting

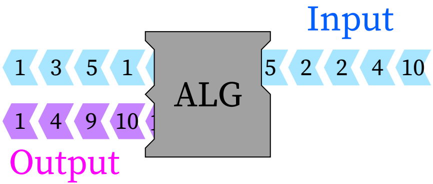


Adversarial setting

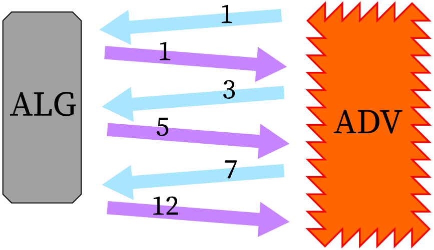
- Makes no difference if deterministic

Setting of a streaming algorithm [Ben-Eliezer, Jayaram, Woodruff, and Yogev 2020]

Static setting



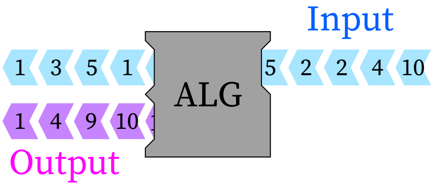
Adversarial setting



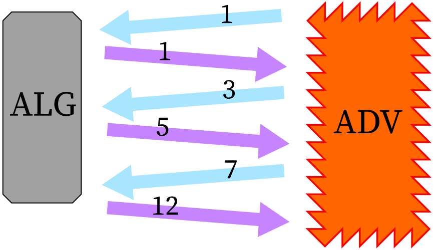
- Makes no difference if deterministic

Setting of a streaming algorithm [Ben-Eliezer, Jayaram, Woodruff, and Yogev 2020]

Static setting



Adversarial setting



- Makes no difference if deterministic

Access to randomness

Deterministic

Random seed

Ex: Linear sketches (with PRG, per Indyk 2006)

Random tape

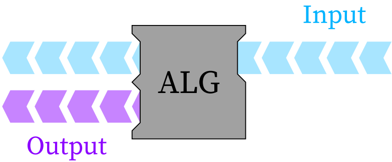
Ex: Morris Counter, reservoir sampling

Random oracle

Ex: Linear sketches

Access to randomness

Deterministic ( )



Random tape ( )

Ex: Morris Counter, reservoir sampling

Random seed ( )

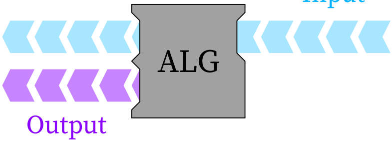
Ex: Linear sketches (with PRG, per Indyk 2006)

Random oracle ( )

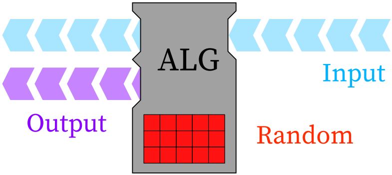
Ex: Linear sketches

Access to randomness

Deterministic ( )



Random seed ( )



Ex: Linear sketches (with PRG, per Indyk 2006)

Random tape ( )

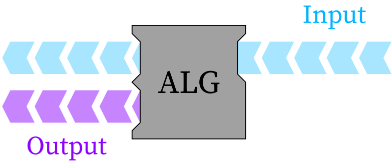
Ex: Morris Counter, reservoir sampling

Random oracle ( )

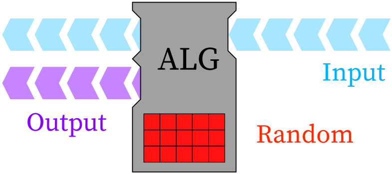
Ex: Linear sketches

Access to randomness

Deterministic ( )

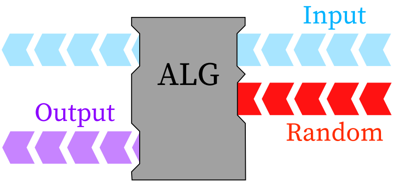


Random seed ( )



Ex: Linear sketches (with PRG, per Indyk 2006)

Random tape ( )



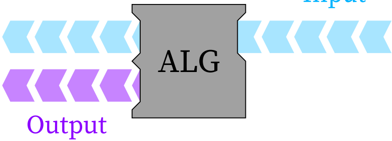
Ex: Morris Counter, reservoir sampling

Random oracle ( )

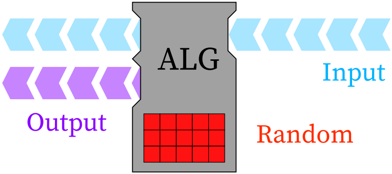
Ex: Linear sketches

Access to randomness

Deterministic ( )

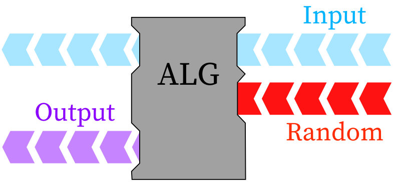


Random seed ( )



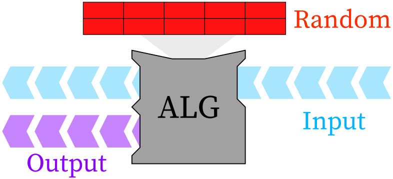
Ex: Linear sketches (with PRG, per Indyk 2006)

Random tape ( )



Ex: Morris Counter, reservoir sampling

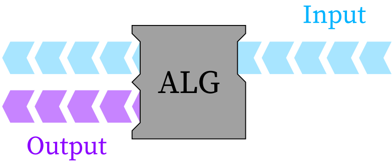
Random oracle ( )



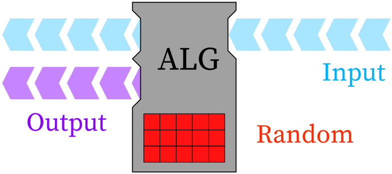
Ex: Linear sketches

Access to randomness

Deterministic ( )

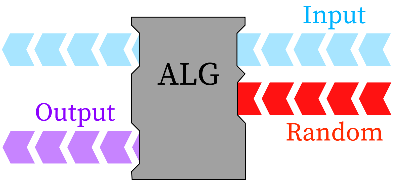


Random seed ( )



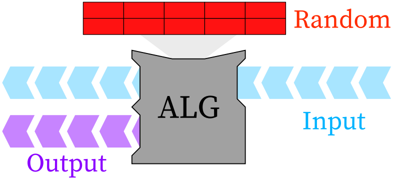
Ex: Linear sketches (with PRG, per Indyk 2006)

Random tape ( )



Ex: Morris Counter, reservoir sampling

Random oracle ( )



Ex: Linear sketches

The Missing Item Finding Problem

- MIF (n, r) is given stream over $[n]$ of length $\leq r$ for $r < n$
- For stream $a_1, \ldots, a_i$, output $v \in [n] \setminus \{a_1 \ldots, a_i\}$

Background

- MIF special cases in: [Chakrabarti, Ghosh, and Stoeckl 2022; Tarui 2007].
- See also:
  - Graph coloring in edge arrival streams [Assadi, Y. Chen, and Khanna 2019; Assadi, A. Chen, and G. Sun 2022; Chakrabarti, Ghosh, and Stoeckl 2022]
  - Card Guessing Game and Mirror Game [Garg and Schneider 2018; Menuhin and Naor 2022]
  - Static/adversarial separation [Kaplan, Mansour, Nissim, and Stemmer 2021]

The Missing Item Finding Problem

- MIF (n, r) is given stream over $[n]$ of length $\leq r$ for $r < n$
- For stream $a_1, \ldots, a_i$, output $v \in [n] \setminus \{a_1 \ldots, a_i\}$

Background

- MIF special cases in: [Chakrabarti, Ghosh, and Stoeckl 2022; Tarui 2007].
- See also:
  - Graph coloring in edge arrival streams [Assadi, Y. Chen, and Khanna 2019; Assadi, A. Chen, and G. Sun 2022; Chakrabarti, Ghosh, and Stoeckl 2022]
  - Card Guessing Game and Mirror Game [Garg and Schneider 2018; Menuhin and Naor 2022]
  - Static/adversarial separation [Kaplan, Mansour, Nissim, and Stemmer 2021]

Space complexity for Missing Item Finding†

Model	Space
Static setting, random seed ( )	$\tilde{O}(1)$
Adversarial setting, random oracle ( )	$\tilde{\Theta}\left(1 + \frac{r^2}{n}\right)$
Deterministic ( )	$\tilde{\Theta}\left(\sqrt{r} + \frac{r}{\log n}\right)$

† Error $\delta = \Theta(1)$; $r \leq n/2$; and $\tilde{O}(\cdot)$ hides $\mathrm{polylog}\, r$ factors

Do stronger lower bounds hold for random seed/tape algorithms in adversarial setting?

- Adversarial setting, random oracle ()
  - $\tilde{\Omega}\left(1+\frac{r^2}{n}\right)$ lower bound applies to all random models
  - $\tilde{O}\left(1+\frac{r^2}{n}\right)$ algorithm uses $\tilde{\Omega}(r)$ oracle random bits. Open: random seed/tape

In adversarial setting, is random oracle necessary for least space?

- Answer is NO in static setting ($\tilde{O}(\log m)$ random seed sufficient by [Newman 1991])
- Impact:
  - Random seed / tape models: use hardware random generator
  - Random oracle: use cryptographic pseudo-random generator

Do stronger lower bounds hold for random seed/tape algorithms in adversarial setting?

- Adversarial setting, random oracle ( )
  - $\tilde{\Omega}\left(1 + \frac{r^2}{n}\right)$ lower bound applies to all random models
  - $\tilde{O}\left(1 + \frac{r^2}{n}\right)$ algorithm uses $\tilde{\Omega}(r)$ oracle random bits. Open: random seed/tape

In adversarial setting, is random oracle necessary for least space?

- Answer is NO in static setting ($\tilde{O}(\log m)$ random seed sufficient by [Newman 1991])
- Impact:
  - Random seed / tape models: use hardware random generator
  - Random oracle: use cryptographic pseudo-random generator

Do stronger lower bounds hold for random seed/tape algorithms in adversarial setting?

- Adversarial setting, random oracle ()
  - $\tilde{\Omega}\left(1 + \frac{r^2}{n}\right)$ lower bound applies to all random models
  - $\tilde{O}\left(1 + \frac{r^2}{n}\right)$ algorithm uses $\tilde{\Omega}(r)$ oracle random bits. Open: random seed/tape

In adversarial setting, is random oracle necessary for least space?

- Answer is NO in static setting ($\tilde{O}(\log m)$ random seed sufficient by [Newman 1991])
- Impact:
  - Random seed / tape models: use hardware random generator
  - Random oracle: use cryptographic pseudo-random generator

Do stronger lower bounds hold for random seed/tape algorithms in adversarial setting?

- Adversarial setting, random oracle ()
  - $\tilde{\Omega}\left(1 + \frac{r^2}{n}\right)$ lower bound applies to all random models
  - $\tilde{O}\left(1 + \frac{r^2}{n}\right)$ algorithm uses $\tilde{\Omega}(r)$ oracle random bits. Open: random seed/tape

In adversarial setting, is random oracle necessary for least space?

- Answer is NO in static setting ($\tilde{O}(\log m)$ random seed sufficient by [Newman 1991])
- Impact:
  - Random seed / tape models: use hardware random generator
  - Random oracle: use cryptographic pseudo-random generator

Overview

About models for streaming algorithms

Proof and algorithm sketches

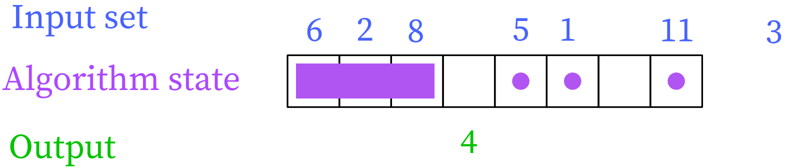
- Full result table
- Algorithm example
- Lower bounds

Table of results[‡]

Model	Space
Static setting, random seed	$\tilde{O}(1)$
Adversarial setting, random oracle	$\tilde{\Theta}\left(1 + \frac{r^2}{n}\right)$
Deterministic	$\tilde{\Theta}\left(\sqrt{r} + \frac{r}{\log n}\right)$
White box adversarial, random tape	$\Omega\left(\frac{r}{\operatorname{polylog} n}\right)$ if $\delta \leq \frac{1}{2^{\operatorname{polylog} n}}$
Zero error, random oracle	$\tilde{\Theta}\left(1 + \frac{r^2}{n}\right)$ expected
Adversarial setting, random seed	conditional on pseudo-deterministic

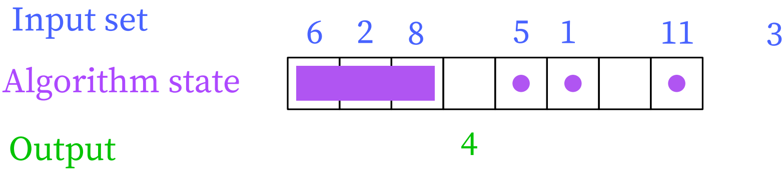
[‡] Error $\delta = \Theta(1)$; $r \leq n/2$; and $\tilde{O}(\cdot)$ hides $\operatorname{polylog} r$ factors

Algorithm for adversarial setting with random oracle ( )



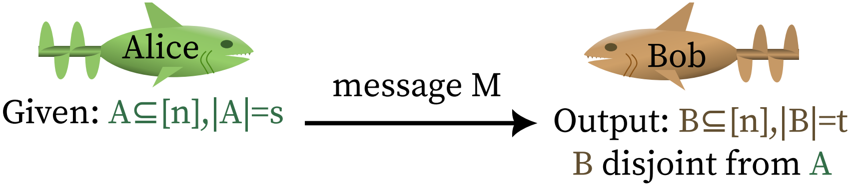
- Using $\tilde{O}(r)$ bits from oracle, pick random list of length $r+1$
- Track which of elements in list have been seen
- Report first available element
- After analysis: $\tilde{O}\left(1+\frac{r^2}{n}\right)$ space needed w.h.p.

Algorithm for adversarial setting with random oracle ( )



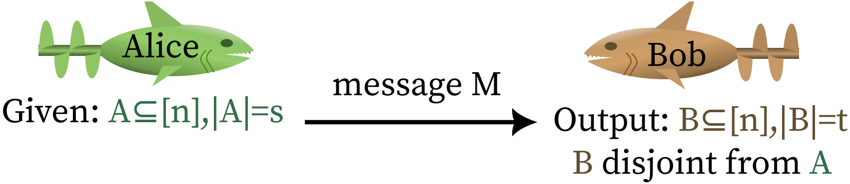
- Using $\tilde{O}(r)$ bits from oracle, pick random list of length $r + 1$
- Track which of elements in list have been seen
- Report first available element
- After analysis: $\tilde{O}\left(1 + \frac{r^2}{n}\right)$ space needed w.h.p.

Lower bound: Adversarial setting, random oracle ()



- Implement AVOID$\left(n, s = \frac{r}{2}, t = \frac{r}{2} + 1\right)$ using MIF(n, r)
  - Needs $\Omega(st/n) = \Omega\left(r^2/n\right)$ bitsChakrabarti, Ghosh, and Stoeckl 2022
  - M = MIF algorithm state on $a_1, \ldots, a_s = A$
  - B = set formed by repeatedly asking algorithm for output and feeding output back into algorithm

Lower bound: Adversarial setting, random oracle ()



- Implement AVOID$\left(n, s = \frac{r}{2}, t = \frac{r}{2} + 1\right)$ using MIF$\left(n, r\right)$
  - Needs $\Omega\left(st/n\right) = \Omega\left(r^2/n\right)$ bitsChakrabarti, Ghosh, and Stoeckl 2022
  - $M =$ MIF algorithm state on $a_1, \ldots, a_s = A$
  - $B =$ set formed by repeatedly asking algorithm for output and feeding output back into algorithm

Pseudo-deterministic (PD) streaming algorithms [Goldwasser, Grossman, Mohanty, and Woodruff 2020]

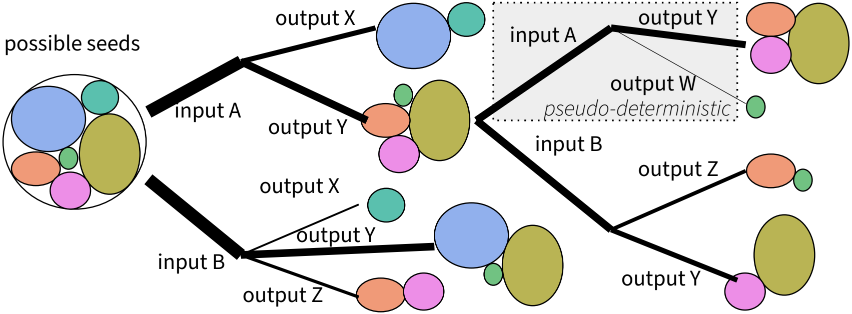
- For any input stream, will give exact same output with probability $\geq 1 - \delta$
- $\tilde{\Omega}\left(\sqrt{r}\right)$ lower bound for random seed, adversarial setting IF pseudo-deterministic requires $\tilde{\Omega}\left(r\right)$ space:
- Design adversary for random seed algorithm over a number of epochs. Either:
  - Adversary can learn information about random seed (happens only $O\left(\text{space}\right)$ times)
  - Algorithm behaves pseudo-deterministically on a short stretch of the stream

Pseudo-deterministic (PD) streaming algorithms [Goldwasser, Grossman, Mohanty, and Woodruff 2020]

- For any input stream, will give exact same output with probability $\geq 1 - \delta$
- $\tilde{\Omega}\left(\sqrt{r}\right)$ lower bound for random seed, adversarial setting IF pseudo-deterministic requires $\tilde{\Omega}\left(r\right)$ space:
- Design adversary for random seed algorithm over a number of epochs. Either:
  - Adversary can learn information about random seed (happens only $O\left(\text{space}\right)$ times)
  - Algorithm behaves pseudo-deterministically on a short stretch of the stream

Pseudo-deterministic (PD) streaming algorithms [Goldwasser, Grossman, Mohanty, and Woodruff 2020]

- For any input stream, will give exact same output with probability $\geq 1 - \delta$
- $\tilde{\Omega}\left(\sqrt{r}\right)$ lower bound for random seed, adversarial setting IF pseudo-deterministic requires $\tilde{\Omega}\left(r\right)$ space:
- Design adversary for random seed algorithm over a number of epochs. Either:
  - Adversary can learn information about random seed (happens only $O\left(\text{space}\right)$ times)
  - Algorithm behaves pseudo-deterministically on a short stretch of the stream



Pseudo-deterministic (PD) streaming algorithms [Goldwasser, Grossman, Mohanty, and Woodruff 2020]

- For any input stream, will give exact same output with probability $\geq 1 - \delta$
- $\tilde{\Omega}\left(\sqrt{r}\right)$ lower bound for random seed, adversarial setting IF pseudo-deterministic requires $\tilde{\Omega}\left(r\right)$ space:
- Design adversary for random seed algorithm over a number of epochs. Either:
  - Adversary can learn information about random seed (happens only $O\left(\text{space}\right)$ times)
  - Algorithm behaves pseudo-deterministically on a short stretch of the stream

Updates

Space lower bound for PD, given a random seed adversarial lower bound
Explicit $\tilde{O}\left(\sqrt{r} + \frac{r^2}{n}\right)$ random seed upper bound in adversarial setting

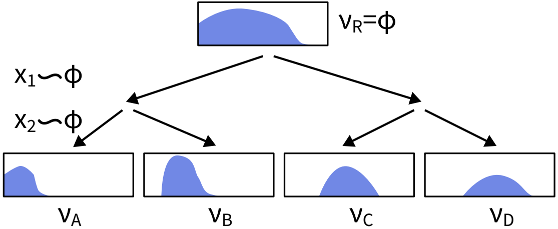
White box adversarial setting [Ajtai, Braverman, Jayram, Silwal, A. Sun, Woodruff, and Zhou 2022], with random tape ()

- White box adversary sees algorithm state, but not future random tape
- Lower bound: $\tilde{\Omega}(r)$ if $\delta \leq 2^{-\mathrm{polylog}\, n}$, proven by contradiction using adversary:
- Want to sample next few inputs from ϕ, where ϕ also equals expected output distribution
  - Use Brouwer's fixed point theorem

- Output distribution at state avoids inputs leading to it
- "concentration" of output distributions must increase beyond limit: contradiction

White box adversarial setting [Ajtai, Braverman, Jayram, Silwal, A. Sun, Woodruff, and Zhou 2022], with random tape

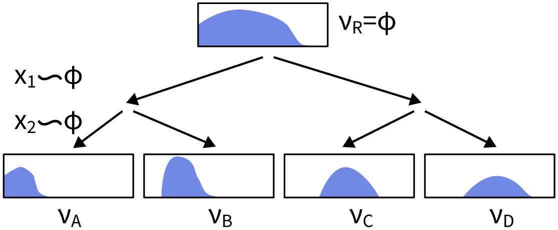
- White box adversary sees algorithm state, but not future random tape
- Lower bound: $\tilde{\Omega}(r)$ if $\delta \leq 2^{-\mathrm{polylog}\, n}$, proven by contradiction using adversary:
- Want to sample next few inputs from ϕ, where ϕ also equals expected output distribution
  - Use Brouwer's fixed point theorem



- Output distribution at state avoids inputs leading to it
- "concentration" of output distributions must increase beyond limit: contradiction

White box adversarial setting [Ajtai, Braverman, Jayram, Silwal, A. Sun, Woodruff, and Zhou 2022], with random tape

- White box adversary sees algorithm state, but not future random tape
- Lower bound: $\tilde{\Omega}(r)$ if $\delta \leq 2^{-\text{polylog}\, n}$, proven by contradiction using adversary:
- Want to sample next few inputs from ϕ, where ϕ also equals expected output distribution
  - Use Brouwer's fixed point theorem



- Output distribution at state avoids inputs leading to it
- "concentration" of output distributions must increase beyond limit: contradiction

Summary

- Missing Item Finding(n, r): find element not in stream so far
- In adversarial setting, is random oracle necessary?

Model	Space
Static setting, random seed ( )	$\tilde{O}(1)$
Adversarial setting, random oracle ( )	$\tilde{\Theta}\left(1+\frac{r^2}{n}\right)$
Deterministic ( )	$\tilde{\Theta}\left(\sqrt{r}+\frac{r}{\log n}\right)$
White box adversarial, random tape ( )	$\Omega\left(\frac{r}{\mathrm{polylog}\, n}\right)$ if $\delta \leq \frac{1}{2^{\mathrm{polylog}\, n}}$
Zero error, random oracle ( )	$\tilde{\Theta}\left(1+\frac{r^2}{n}\right)$ expected
Adversarial setting, random seed ( )	conditional on pseudo-deterministic
Adversarial setting, random tape ( )	???

Miklós Ajtai, Vladimir Braverman, T.S. Jayram, Sandeep Silwal, Alec Sun, David P. Woodruff, and Samson Zhou. The white-box adversarial data stream model. In *Proc. 41st ACM Symposium on Principles of Database Systems*, pages 15–27, 2022. doi: 10.1145/3517804.3526228.

Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for $(\Delta + 1)$ vertex coloring. In *Proc. 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 767–786, 2019. doi: 10.1137/1.9781611975482.48.

Sepehr Assadi, Andrew Chen, and Glenn Sun. Deterministic graph coloring in the streaming model. In *Proc. 54th Annual ACM Symposium on the Theory of Computing*, pages 261–274, 2022. doi: 10.1145/3519935.3520016.

Omri Ben-Eliezer, Rajesh Jayaram, David P. Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. In *Proc. 39th ACM Symposium on Principles of Database Systems*, pages 63–80, 2020. doi: 10.1145/3375395.3387658.

Amit Chakrabarti, Prantar Ghosh, and Manuel Stoeckl. Adversarially robust coloring for graph streams. In *Proc. 13th Conference on Innovations in Theoretical Computer Science*, 37:1–37:23, 2022. doi: 10.4230/LIPIcs.ITCS.2022.37.

Shafi Goldwasser, Ofer Grossman, Sidhanth Mohanty, and David P. Woodruff. Pseudo-Deterministic Streaming. In *Proc. 20th Conference on Innovations in Theoretical Computer Science*, volume 151, 79:1–79:25, 2020. doi: 10.4230/LIPIcs.ITCS.2020.79.

Sumegha Garg and Jon Schneider. The Space Complexity of Mirror Games. In *Proc. 10th Conference on Innovations in Theoretical Computer Science*, 36:1–36:14, 2018. doi: 10.4230/LIPIcs.ITCS.2019.36.

Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006.

Haim Kaplan, Yishay Mansour, Kobbi Nissim, and Uri Stemmer. Separating adaptive streaming from oblivious streaming using the bounded storage model. In *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 94–121. Springer, 2021. doi: 10.1007/978-3-030-84252-9_4.

Boaz Menuhin and Moni Naor. Keep that card in mind: card guessing with limited memory. In *Proc. 13th Conference on Innovations in Theoretical Computer Science*, 107:1–107:28, 2022. doi: 10.4230/LIPIcs.ITCS.2022.107.

Ilan Newman. Private vs. common random bits in communication complexity. *Inform. Process. Lett.*, 39(2):67–71, 1991.

Jun Tarui. Finding a duplicate and a missing item in a stream. In *Proc. 4th International Conference on Theory and Applications of Models of Computation*, pages 128–135, 2007. doi: 10.1007/978-3-540-72504-6_11.