Lower Bounds and Open Problems in Streams

Raphaël Clifford

Joint work with
Markus Jalsenius and Benjamin Sach
The CPU does not remember anything in between operations.
The CPU does not remember anything in between operations.

The CPU has unlimited computational power.
Data Structure Lower Bounds

Yao - FOCS ’78

Predecessor (static)
- Ajtai - Combinatorica ’88 (incorrect) (Communication complexity)
- Miltersen - STOC ’94
- Miltersen, Nisan, Safra, Wigderson - STOC ’95
- Beame, Fich - STOC ’99
- Sen - ICALP ’01

Dynamic problems (partial sums, union find)
- Fredman, Saks - STOC ’89 (Chronogram technique)
- Ben-Amram, Galil - FOCS ’91
- Miltersen, Subramanian, Vitter, Tamassia - TCS ’94
- Husfeldt, Rauhe, Skyum - SWAT ’96 (planar connectivity)
- Fredman, Henzinger - Algorithmica ’98 (non-determinism)
- Alstrup, Husfeldt, Rauhe - FOCS ’98 (marked ancestor)
- Alstrup, Husfeldt, Rauhe - SODA ’01 (2D NN)
- Alstrup, Ben-Amram, Rauhe - STOC ’99 (union-find)
Data Structure Lower Bounds

Yao - FOCS ’78
Predecessor (static)
- Ajtai - Combinatorica ’88 (incorrect) (Communication complexity)
- Miltersen - STOC’ 94
- Miltersen, Nisan, Safra, Wigdersen - STOC ’95
- Beame, Fich - STOC ’99
- Sen - ICALP ’01

Dynamic problems (partial sums, union find)
- Fredman, Saks - STOC ’89 (Chronogram technique)
- Ben-Amram, Galil - FOCS ’93 (Dynamization technique)
- Miltersen, Subramanian - TCS ’94
- Husfeldt, Rauhe, Skyum - SWAT ’96 (planar connectivity)
- Fredman, Henzinger - Algorithmica ’98 (non-determinism)
- Alstrup, Husfeldt, Rauhe - FOCS ’98 (marked ancestor)
- Alstrup, Husfeldt, Rauhe - SODA ’01 (2D NN)
- Alstrup, Ben-Amram, Rauhe - STOC ’99 (union-find)

Best lower bound
\(\Omega\left(\frac{\log n}{\log \log n}\right)\)
Data Structure Lower Bounds

Yao - FOCS '78

Predecessor (static)
- Ajtai - Combinatorica '88 (incorrect) (Communication complexity)
- Miltersen - STOC ’94
- Miltersen, Nisan, Safra, Wigdersen - STOC ’95
- Beame, Fich - STOC ’99
- Sen - ICALP '01

Dynamic problems (partial sums, union find)
- Fredman, Saks - STOC '89 (Chronogram technique)
- Ben-Amram, Galil - FOCS '91
- Miltersen, Subramanian, Vitter, Tamassia - TCS '94
- Husfeldt, Rauhe, Skyum - SWAT ’96 (planar connectivity)
- Fredman, Henzinger - Algorithmica ’98 (non-determinism)
- Alstrup, Husfeldt, Rauhe - FOCS ’98 (marked ancestor)
- Alstrup, Husfeldt, Rauhe - SODA ’01 (2D NN)
- Alstrup, Ben-Amram, Rauhe - STOC ’99 (union-find)

First $\Omega (\log n)$ lower bound using information transfer.

M. Pătrașcu and E. Demaine
Tight bounds for the partial-sums problem
SODA 2004
Convolution

Stream of numbers from \([q]\)

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fixed vector

\(V \in [q]^n\)

Output dot product (modulo \(q\)):

\[
V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x(i + \text{leftmost aligned index})
\]
Convolution

Stream of numbers from \([q]\)

Fixed vector \(V \in [q]^n\)

Output dot product (modulo \(q\)):

\[
V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x(i + \text{leftmost aligned index})
\]
Convolution

Stream of numbers from $[q]$

Fixed vector $V \in [q]^n$

Output dot product (modulo $q$):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x(i + \text{leftmost aligned index})$$
Stream of numbers from $[q]$

Fixed vector $V \in [q]^n$

Output dot product (modulo $q$):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x(i + \text{leftmost aligned index})$$
Convolution

Stream of numbers from $[q]$

Fixed vector $V \in [q]^n$

Output dot product (modulo $q$):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x(i + \text{leftmost aligned index})$$
Convolution

Stream of numbers from $[q]$

$\begin{array}{cccccccccccccc}
  x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12}
\end{array}$

Fixed vector $V \in [q]^n$

Output dot product (modulo $q$):

$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x(i + \text{leftmost aligned index})$

$\begin{array}{ccccccccccc}
  v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7
\end{array}$
Convolution

Stream of numbers from $[q]$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
<th>$x_9$</th>
<th>$x_{10}$</th>
<th>$x_{11}$</th>
<th>$x_{12}$</th>
<th>$x_{13}$</th>
<th>?</th>
</tr>
</thead>
</table>

Fixed vector $V \in [q]^n$

Output dot product (modulo $q$):

$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x(i + \text{leftmost aligned index})$
Convolution

Stream of numbers from $[q]$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
<th>$x_9$</th>
<th>$x_{10}$</th>
<th>$x_{11}$</th>
<th>$x_{12}$</th>
<th>$x_{13}$</th>
</tr>
</thead>
</table>

Fixed vector $V \in [q]^n$

Output dot product (modulo $q$):

$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x(i + \text{leftmost aligned index})$

Lower bound: $\Omega\left(\frac{\delta}{w} \log n\right)$

$\delta = \log q$, word size $w$.

C., Jalsenius. Lower Bounds for Online Integer Multiplication and Convolution in the Cell-Probe Mode. ICALP 2011
Previous bounds

M. J. Fischer and L. J. Stockmeyer
Fast on-line integer multiplication
STOC ’73

C., K. Efremenko, B. Porat and E. Porat
A black box for online approximate pattern matching

• $O(\log^2 n)$ time per arriving symbol (pair)
Previous bounds

M. J. Fischer and L. J. Stockmeyer
Fast on-line integer multiplication
STOC ’73

C., K. Efremenko, B. Porat and E. Porat
A black box for online approximate pattern matching

• $O(\log^2 n)$ time per arriving symbol (pair)

Offline cell probe complexity is linear!

⇒

online upper bound of $O(\log n)$
Previous bounds

M. J. Fischer and L. J. Stockmeyer
Fast on-line integer multiplication
STOC ’73

C., K. Efremenko, B. Porat and E. Porat
A black box for online approximate pattern matching

• $O(\log^2 n)$ time per arriving symbol (pair)

Better online lower bound

$\Rightarrow$

super linear lower bound for offline convolution and multiplication
Yao’s minimax principle

A lower bound on the expected running time for

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Random</td>
</tr>
</tbody>
</table>

implies that the same lower bound holds for

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomised</td>
<td>Worst case</td>
</tr>
</tbody>
</table>
Yao’s minimax principle

A lower bound on the expected running time for

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Random</td>
</tr>
</tbody>
</table>

implies that the same lower bound holds for

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Worst case</td>
</tr>
</tbody>
</table>
Information transfer

- Fixed value
- Unknown value chosen uniformly at random from $[q]$
Information transfer

- **Fixed value**
- **Unknown value** chosen uniformly at random from \([q]\)

Memory cells
Information transfer

- Fixed value
- Unknown value chosen uniformly at random from $[q]$
Information transfer

Fixed value

Unknown value
chosen uniformly at random from $[q]$
Information transfer

- Fixed value
- Unknown value chosen uniformly at random from $[q]$
Information transfer

[Diagram showing green boxes for fixed values and yellow question mark for unknown values chosen uniformly at random from \([q]\).]
Information transfer

Fixed value

Unknown value
chosen uniformly at random from $[q]$

Cell written during the $?$-inputs
Information transfer

- Fixed value
- Unknown value chosen uniformly at random from $[q]$
Information transfer

Fixed value

Unknown value
chosen uniformly
at random from $[q]$

Cell written during the $t$-inputs

Memory cells
Information transfer

Fixed value

Unknown value chosen uniformly at random from $[q]$.

Cell written during the $\ell$-inputs.
Information transfer

- **Fixed value**
- **Unknown value** chosen uniformly at random from \([q]\)

Cell written during the \(\ell\)-inputs

Cells read during the next \(\ell\) inputs
Information transfer

Cells read during the next \( \ell \) inputs

Unknown value chosen uniformly at random from \([q]\)

Fixed value

Cell written during the \( ? \)-inputs

Cells read during the next \( \ell \) inputs
Information transfer

Cells read during the next $\ell$ inputs

Unknown value chosen uniformly at random from $[q]$
Information transfer

Cells read during the next $\ell$ inputs

Unknown value chosen uniformly at random from $[q]$?

Fixed value

Cell written during the $\ell$-inputs

Memory cells

Cells read during the next $\ell$ inputs
Information transfer

- Fixed value
- Unknown value chosen uniformly at random from $[q]$

Cell written during the $\ell$-inputs

Cells read during the next $\ell$ inputs

Information transfer $IT(t, \ell)$

Not including cells that were overwritten before being read
The cells in $IT(t, \ell)$ provide sufficient information in order to give correct output during inputs.

Not including cells that were overwritten before being read.
Information transfer

Fixed value

Unknown value
chosen uniformly
at random from \([q]\)

The conditional entropy

\[ H(\text{the outputs during } | \text{all fixed}) \leq w + 2w \cdot E[|IT(t, \ell)| | \text{all fixed}] \]

\(w\) bits per cell
The conditional entropy

\[ H(\text{the outputs during } | \text{all } \text{fixed}) \leq w + 2w \cdot \mathbb{E}[|IT(t, \ell)| \mid \text{all } \text{fixed}] \]

\( w \) bits per cell
Information transfer

- **Fixed value**
- **Unknown value** chosen uniformly at random from \([q]\)

The conditional entropy

\[ H(\text{the outputs during } | IT(t, \ell)| \mid \text{all fixed}) \leq w + 2w \cdot E[|IT(t, \ell)| \mid \text{all fixed}] \]

\(w\) bits per cell
How much information about $\ell$ do we need in order to give correct outputs during $t$?
Information transfer

How much information about \( ? ? ? ? ? \) do we need in order to give correct outputs during \( ? \)?

Depends on the fixed vector
Output is always 0 (no information)
Information transfer

Contributes to the dot product with the same value at each alignment

\((\delta = \log q \text{ bits of information})\)
Information transfer

if the position is a power of 2
Information transfer

If the position is a power of 2
if the position is a power of 2
if the position is a power of 2
Information transfer

\[ \ell = 8 \]

1 if the position is a power of 2
if the position is a power of 2
Information transfer

If the position is a power of 2

\[ \ell / 2 = 4 \]

\[ \ell = 8 \]

\[ R \]

= a recovered value

(recall that \(?\) is chosen uniformly at random from \([q]\), hence contributes with \(\delta = \log q\) bits to the entropy)
if the position is a power of 2

\( R \) = a recovered value

(recall that \( ? \) is chosen uniformly at random from \( [q] \), hence contributes with \( \delta = \log q \) bits to the entropy)
if the position is a power of 2

\( \text{R} \) = a recovered value

(recall that ? is chosen uniformly at random from \([q]\), hence contributes with \( \delta = \log q \) bits to the entropy)
if the position is a power of 2

R = a recovered value

(recall that ? is chosen uniformly at random from \([q]\), hence contributes with \(\delta = \log q\) bits to the entropy)
1 if the position is a power of 2

R = a recovered value

(recall that ? is chosen uniformly at random from \([q]\), hence contributes with \(\delta = \log q\) bits to the entropy)

Conclusion: If \(\ell\) is a power of 2 then we recover \(\frac{\ell}{2}\) values
The conditional entropy

\[ H(\text{the outputs during } | \text{ all fixed}) \geq \frac{l}{2} \delta \]

**Conclusion:** If \( l \) is a power of 2 then we recover \( \frac{l}{2} \) values
The conditional entropy

\[ H(\text{the outputs during } | \text{all } \square \text{ fixed}) \geq \frac{\ell}{2} \delta \]

The conditional information transfer

\[ \mathbb{E} [ | IT(t, \ell) | | \text{all } \square \text{ fixed}] \geq \frac{\delta}{4w} \ell - \frac{1}{2} \]

\( w \) bits per cell
Suppose that all values (■ and ?) from the stream are chosen uniformly at random from \([q]\).

By linearity of expectation…

The conditional information transfer

\[
\mathbb{E} \left[ |IT(t, \ell)| \mid \text{all } \square \text{ fixed} \right] \geq \frac{\delta}{4w} \ell - \frac{1}{2}
\]

\(w\) bits per cell
Suppose that all values (and) from the stream are chosen uniformly at random from $[q]$.

By linearity of expectation...

The conditional information transfer

$$\mathbb{E}[|IT(t, \ell)|] \mid \text{all fixed} \geq \frac{\delta}{4w} \ell - \frac{1}{2}$$

$w$ bits per cell
Feed the algorithm with $n$ values chosen uniformly at random from $[q]$. 
Feed the algorithm with $n$ values chosen uniformly at random from $[q]$. 

$IT(t = 1, \ell = 1)$
Total number of cell reads

0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0

$n$

Feed the algorithm with $n$ values chosen uniformly at random from $[q]$. 

$IT(t = 3, \ell = 1)$
Total number of cell reads

\[
\begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0
\end{array}
\]

Feed the algorithm with \( n \) values chosen uniformly at random from \([q]\).
Feed the algorithm with $n$ values chosen uniformly at random from $[q]$. 
Total number of cell reads

\[
\begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
\end{array}
\]

\[n\]

Feed the algorithm with \(n\) values chosen uniformly at random from \([q]\).
Feed the algorithm with $n$ values chosen uniformly at random from $[q]$. 

$IT(t = 5, \ell = 2)$
Total number of cell reads

Feed the algorithm with $n$ values chosen uniformly at random from $[q]$. 

$IT(t = 1, \ell = 4)$
Total number of cell reads

Feed the algorithm with $n$ values chosen uniformly at random from $[q]$. 

$IT(t = 1, \ell = 8)$

Feed the algorithm with $n$ values chosen uniformly at random from $[q]$. 

$IT(t = 1, \ell = 8)$
Total number of cell reads

The number of cell reads during the $n$ inputs is at least

$$\sum_{\text{internal node } v} |IT(t_v, \ell_v)|$$
Total number of cell reads

The number of cell reads during the $n$ inputs is at least

$$\sum_{\text{internal node } v} |IT(t_v, \ell_v)|$$

random from $[q]$.

No double counting of a cell read!
Total number of cell reads

The number of cell reads during the \( n \) inputs is at least

\[
\sum_{\text{internal node } v} |IT(t_v, \ell_v)|
\]

The expected number of cell reads is at least

\[
\mathbb{E} \left[ \sum_{\text{internal node } v} |IT(t_v, \ell_v)| \right] = \sum_{\text{internal node } v} \mathbb{E} [ |IT(t_v, \ell_v)| ] \\
\geq \sum_{\text{internal node } v} \frac{\delta}{4w} \ell_v - \frac{1}{2} \\
= \Omega \left( \frac{\delta}{w} \cdot n \log n \right)
\]
Total number of cell reads

The number of cell reads during the $n$ inputs is at least

$$\sum_{\text{internal node } v} |IT(t_v, \ell_v)|$$

The expected number of cell reads is at least

$$\mathbb{E}\left[\sum_{\text{internal node } v} |IT(t_v, \ell_v)|\right] = \sum_{\text{internal node } v} \mathbb{E}[|IT(t_v, \ell_v)|]$$

So...

The amortised time lower bound per output is $\Omega\left(\frac{\delta}{w} \log n\right)$
What happens if the alphabet is binary?

For binary alphabet and sensible word size, we get useless

\[ \Omega \left( \frac{\log n}{w} \right) = \Omega(1). \]
What happens if the alphabet is binary?

For binary alphabet and sensible word size, we get useless

$$\Omega \left( \frac{\log n}{w} \right) = \Omega(1).$$

But...

▶ What if each output is in \{0, \ldots, n\}?  

▶ Total entropy of \( n/\log n \) outputs could therefore be \( \Omega(n) \).

▶ We could then use a new *lop-sided information transfer* technique instead.
Pattern matching with address errors

Message sent: *eleven plus two*

The $L_2$-rearrangement distance defined to be

$$\min_{\pi \in \Pi} \sum_{j=0}^{n-1} \left( j - \pi(j) \right)^2$$

(AABLLPSV:2009)

Online: $O(\log_2 n)$ time per arriving symbol (CS:2011).

Example

The $L_2$-rearrangement distance of 11100 and 10110 is $0^2 + 1^2 + 1^2 + 2^2 + 0^2 = 6$. 
Pattern matching with address errors

Message sent: eleven plus two
Message received: twelve plus one
Pattern matching with address errors

Message sent: *eleven plus two*
Message received: *twelve plus one*

- The $L_2$-rearrangement distance defined to be
  \[ \min_{\pi \in \Pi} \sum_{j=0}^{n-1} (j - \pi(j))^2 \]  
  (AABLLPSV:2009)
- Online: $O(\log^2 n)$ time per arriving symbol (CS:2011).

**Example**
The $L_2$-rearrangement distance of 11100 and 10110 is
\[ 0^2 + 1^2 + 1^2 + 2^2 + 0^2 = 6. \]
Pattern matching with address errors

For binary inputs, our new lower bound is:

$$\Omega \left( \frac{\lg^2 n}{w \cdot \lg \lg n} \right)$$

To do this we must find an input distribution such that:

- The conditional entropy of the outputs is high.
- It is possible to sum the contributions from many interval lengths without double counting.
To sum contributions, we introduce a gap:

The lengths $\ell$ are taken from:

$$\left\{ n^{1/4} \cdot (\lg n)^{2i} \mid i \in \{0, 1, 2, \ldots, \frac{\lg n}{4\lg \lg n}\} \right\}.$$
Upper bound on entropy

\[ H(A_{\ell,t} | \tilde{U}_{\ell,t} = \tilde{u}_{\ell,t}) \leq 2w + 2w \cdot \mathbb{E}[l_{\ell,t} + G_{\ell,t} | \tilde{U}_{\ell,t} = \tilde{u}_{\ell,t}] \].
Lop-sided information transfer - Mind the gap

Upper bound on entropy

\[ H(A_{\ell,t} \mid \tilde{U}_{\ell,t} = \tilde{u}_{\ell,t}) \leq 2w + 2w \cdot \mathbb{E}[l_{\ell,t} + G_{\ell,t} \mid \tilde{U}_{\ell,t} = \tilde{u}_{\ell,t}] \]

Lower bound on entropy

**Lemma**

For the \(L_2\)-rearrangement distance problem there exists a hard input distribution such that

\[ H(A_{\ell,t} \mid \tilde{U}_{\ell,t} = \tilde{u}_{\ell,t}) \geq \kappa \cdot \ell \cdot \lg n \]

for any fixed \(\tilde{u}_{\ell,t}\).
We remove the conditioning by taking expectation over $\tilde{U}_{\ell,t}$ under random $U$ giving:

$$\mathbb{E}[l_{\ell,t}] \geq \kappa \cdot \ell \cdot \lg n \cdot \frac{1}{2w} - 1 - \mathbb{E}[G_{\ell,t}].$$

By carefully choosing $T_{\ell}$ we get:

$$\mathbb{E}\left[\sum_{\ell \in L} \sum_{t \in T_{\ell}} l_{\ell,t}\right] \in \Omega\left(\frac{n \cdot \lg^2 n}{w \cdot \lg \lg n}\right).$$
The hard distribution for $L_2$-rearrangement

We let the incoming streaming be randomly sampled from:

$$\{0101, 1010\}^*$$
The hard distribution for $L_2$-rearrangement

We let the incoming streaming be randomly sampled from:

$$\{0101, 1010\}^*$$

Different bits of the output give different bits of the stream.

```
output = 9 = 1001 (in binary)
```

```
\begin{array}{c}
\bullet \quad \bullet \quad \bullet \quad \bullet \\
U_0^{\ell} \quad U_1^{\ell} \quad U_2^{\ell} \quad U_3^{\ell} \quad U_4^{\ell} \quad U_5^{\ell} \quad U_6^{\ell} \quad U_7^{\ell}
\end{array}
```

```
\begin{array}{c}
\bullet \quad \bullet \quad \bullet \quad \bullet \\
F_0^{\ell} \quad F_1^{\ell} \quad \ldots \quad F_3^{\ell}
\end{array}
```

($\cdots$ denotes a repeated stretch of 1001)
A lower bound for convolution?

For convolution we hit a tricky mathematical hurdle.

- What is the entropy of $n/ \log n$ consecutive overlapping inner products?
A lower bound for convolution?

For convolution we hit a tricky mathematical hurdle.

- What is the entropy of $n / \log n$ consecutive overlapping inner products?

$$111011 \longleftrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Conjecture

Let $x \in \{0, 1\}^\ell$ be sampled at random. There exist $\ell / \log \ell$ by $\ell$ Toeplitz matrices $M$ such that $H(Mx) \in \Omega(\ell)$. 
A lower bound for convolution?

For convolution we hit a tricky mathematical hurdle.

▶ What is the entropy of $n/ \log n$ consecutive overlapping inner products?

$111011 \leftrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$

Conjecture

Let $x \in \{0, 1\}^\ell$ be sampled at random. There exist $\ell/ \log \ell$ by $\ell$ Toeplitz matrices $M$ such such that $H(Mx) \in \Omega(\ell)$. 
A lower bound for convolution?

For convolution we hit a tricky mathematical hurdle.

- What is the entropy of $n/\log n$ consecutive overlapping inner products?

\[
111011 \leftrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}
\]

Conjecture

Let $x \in \{0, 1\}^\ell$ be sampled at random. There exist $\ell/\log \ell$ by $\ell$ Toeplitz matrices $M$ such that $H(Mx) \in \Omega(\ell)$.

Thank you!