The Order of the Data Stream

Andrew McGregor (UCSD)

includes work with Sudipto Guha (UPenn)
Data Stream Model
Data Stream Model

- **Stream**: Length $m$ list of numbers from range $[n]$: $S = 2, 30, 10, 2, 66, 43, 240, 3, 12, 492, ...$
Data Stream Model

- **Stream**: Length $m$ list of numbers from range $[n]$: $S = 2, 30, 10, 2, 66, 43, 240, 3, 12, 492, ...$

- **Goal**: Estimate a function $f$ of $S$ such as median, entropy, distinct elements count, longest increasing subsequence, ...
Data Stream Model

- **Stream**: Length $m$ list of numbers from range $[n]$:
  
  $S = 2, 30, 10, 2, 66, 43, 240, 3, 12, 492, ...$

- **Goal**: Estimate a function $f$ of $S$ such as median, entropy, distinct elements count, longest increasing subsequence, ...

- **The Catch**:
  
  Sequential access to data in fixed order
  Limited memory, typically $O(\text{polylog}(m,n))$ bits
  Limited number of passes over the data
Data Stream Model

- **Stream**: Length $m$ list of numbers from range $[n]$:
  $S = 2, 30, 10, 2, 66, 43, 240, 3, 12, 492, ...$

- **Goal**: Estimate a function $f$ of $S$ such as median, entropy, distinct elements count, longest increasing subsequence, ...

- **The Catch**: Sequential access to data in fixed order
  - Limited memory, typically $O(\text{polylog}(m,n))$ bits
  - Limited number of passes over the data

- **Applications**: Monitoring network traffic, achieving I/O efficiency, database query-planning, sensor networks...

[Morris '78] [Munro, Paterson '78] [Flajolet, Martin '85]
[Alon, Matias, Szegedy '96] [Henzinger, Raghavan, Rajagopalan '98]
Order of the Data Stream
Order of the Data Stream

- **Order Dependent Function:** \( f(S) \neq f(\pi(S)) \), e.g., longest-increasing seq., histograms of time-series data, ...
  
  [Guha, Koudas, Shim ’01] [Liben-Nowell, Vee, Zhu ’05]
Order of the Data Stream

- **Order Dependent Function:** $f(S) \neq f(\pi(S))$, e.g., longest-increasing seq., histograms of time-series data, ...  
  [Guha, Koudas, Shim ’01] [Liben-Nowell, Vee, Zhu ’05]

- **Order Invariant Function:** $f(S) = f(\pi(S))$, e.g., median, frequency moments, number of distinct items, ...  
  Want guarantees for worst-case ordering.  
  What about average/random orderings?  
  e.g., processing streams of samples.  
  [Munro, Paterson ’78] [Demaine, López-Ortiz, Munro ’02]  
  [Guha, McGregor, Venkatasubramanian ’06]  
  [Guha, McGregor ’06] [Guha, McGregor ’07a]
Order of the Data Stream

• **Order Dependent Function:** \( f(S) \neq f(\pi(S)) \), e.g., longest-increasing seq., histograms of time-series data, ...
  
  [Guha, Koudas, Shim '01] [Liben-Nowell, Vee, Zhu '05]

• **Order Invariant Function:** \( f(S) = f(\pi(S)) \), e.g., median, frequency moments, number of distinct items, ...
  
  Want guarantees for worst-case ordering.
  What about average/random orderings?
  e.g., processing streams of samples.
  
  [Munro, Paterson '78] [Demaine, López-Ortiz, Munro '02]
  [Guha, McGregor, Venkatasubramanian '06]
  [Guha, McGregor '06] [Guha, McGregor '07a]

• **Today's Focus:** Understanding via lower-bounds...
• **Thm:** Finding median in 1 pass (w/p $3/4$) requires $\Omega(m)$ space.  
[Henzinger, Raghavan, Rajagopalan '99]
• **Thm:** Finding median in 1 pass (w/p 3/4) requires $\Omega(m)$ space. [Henzinger, Raghavan, Rajagopalan '99]

• **Proof:** Assume an alg. $A$ exists using $o(m)$ space.
• **Thm:** Finding median in 1 pass (w/p 3/4) requires $\Omega(m)$ space. [Henzinger, Raghavan, Rajagopalan '99]

• **Proof:** Assume an alg. $\mathcal{A}$ exists using $o(m)$ space. Use it to solve the “INDEX” problem.
Alice

length
t=\frac{m+1}{2}

binary string x

Bob

index i in range [t]

• **Thm:** Finding median in 1 pass (w/p 3/4) requires \(\Omega(m)\) space. [Henzinger, Raghavan, Rajagopalan '99]

• **Proof:** Assume an alg. \(A\) exists using \(o(m)\) space.
  Use it to solve the “INDEX” problem.
INDEX: “What’s the value of $x_i$?”

Any one-way protocol that works w/p 3/4 requires $\Omega(t)$ bits sent.

Alice

length
t=$(m+1)/2$
binary string $x$

Bob

index $i$ in range $[t]$

Thm: Finding median in 1 pass (w/p 3/4) requires $\Omega(m)$ space. [Henzinger, Raghavan, Rajagopalan ’99]

Proof: Assume an alg. $\mathcal{A}$ exists using $o(m)$ space. Use it to solve the “INDEX” problem.
INDEX: “What’s the value of $x_i$?”

Any one-way protocol that works w/p 3/4 requires $\Omega(t)$ bits sent.

Alice
length
t=$(m+1)/2$
binary string $x$

Bob
index $i$ in range $[t]$

• **Thm:** Finding median in 1 pass (w/p 3/4) requires $\Omega(m)$ space.  
  [Henzinger, Raghavan, Rajagopalan ’99]

• **Proof:** Assume an alg. $A$ exists using $o(m)$ space.
  Use it to solve the “INDEX” problem.
Alice
length
t=(m+1)/2
binary string x

Bob
index i in range [t]

INDEX: “What’s the value of \( x_i \)?”
Any one-way protocol that works w/p 3/4 requires \( \Omega(t) \) bits sent.

- **Thm:** Finding median in 1 pass (w/p 3/4) requires \( \Omega(m) \) space. [Henzinger, Raghavan, Rajagopalan ’99]
- **Proof:** Assume an alg. \( \mathcal{A} \) exists using \( o(m) \) space. Use it to solve the “INDEX” problem.
INDEX: “What’s the value of $x_i$?”
Any one-way protocol that works w/p $3/4$ requires $\Omega(t)$ bits sent.

Alice
length $t = (m+1)/2$
binary string $x$

Bob
index $i$ in range $[t]$

- **Thm:** Finding median in 1 pass (w/p 3/4) requires $\Omega(m)$ space.  
  [Henzinger, Raghavan, Rajagopalan ’99]

- **Proof:** Assume an alg. $\mathcal{A}$ exists using $o(m)$ space.  
  Use it to solve the “INDEX” problem.
INDEX: “What’s the value of $x_i$?”

Any one-way protocol that works w/p 3/4 requires $\Omega(t)$ bits sent.

Alice
length
t=$\frac{(m+1)}{2}$
binary string $x$

Bob
index i in range $[t]$

• **Thm:** Finding median in 1 pass (w/p 3/4) requires $\Omega(m)$ space. [Henzinger, Raghavan, Rajagopalan '99]

• **Proof:** Assume an alg. $\mathcal{A}$ exists using $o(m)$ space. Use it to solve the “INDEX” problem. Protocol uses $o(m)$ bits. Contradiction!
Our Talk

“Proving Lower Bounds When You Can’t Be Too Mean to the Algorithms”
Our Talk

“Proving Lower Bounds When You Can’t Be Too Mean to the Algorithms”

Selection:

One-Pass, LB (Random Order)
(general random-order lower bound)

Multi-Pass LB (Advers. Order)
(exponential separation between order)
Our Talk

“Proving Lower Bounds When You Can’t Be Too Mean to the Algorithms”

Selection:
One-Pass, LB (Random Order) (general random-order lower bound)
Multi-Pass LB (Advers. Order) (exponential separation between order)

Longest Increasing Subsequence:
Multi-Pass LB (doubly exponential pass/space trade-off)
Our Talk

“Proving Lower Bounds When You Can’t Be Too Mean to the Algorithms”

Selection:

One-Pass, LB (Random Order)  
(general random-order lower bound)

Multi-Pass LB (Advers. Order)  
(exponential separation between order)

Longest Increasing Subsequence:

Multi-Pass LB  
(doubly exponential pass/space trade-off)

Post-Order Tree Traversal:

The “Pass Elimination” Technique
Our Talk

“Proving Lower Bounds When You Can’t Be Too Mean to the Algorithms”

Selection:
One-Pass, LB (Random Order)  
(general random-order lower bound)
Multi-Pass LB (Advers. Order)  
(exponential separation between order)

Longest Increasing Subsequence:  
Multi-Pass LB  
(doubly exponential pass/space trade-off)

Post-Order Tree Traversal:  
The “Pass Elimination” Technique

Other Applications: 3D Linear Programming, Histograms, ...
1: Selection  
2: Tree Traversal  
3: Applications
I: Selection

a) Previous Work
b) Single-Pass LB (Random Order)
c) Multi-Pass LB (Advers. Order)
Previous Work
Previous Work

- **t-approx median**: Any element of rank = m/2±t
Previous Work

- **t-approx median**: Any element of rank $= m/2 \pm t$
- Previous Work (The Story So Far...):
Previous Work

- **t-approx median**: Any element of rank = m/2±t

- **Previous Work (The Story So Far...):**

<table>
<thead>
<tr>
<th>Approx.</th>
<th>Passes</th>
<th>Space</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>m^α</td>
<td>1</td>
<td>Œ(m^{1-α})</td>
<td>Advers.</td>
</tr>
</tbody>
</table>

[Greenwald, Khanna ‘01]
• **t-approx median**: Any element of rank = \( m/2 \pm t \)

• **Previous Work (The Story So Far...)**:

<table>
<thead>
<tr>
<th>Approx.</th>
<th>Passes</th>
<th>Space</th>
<th>Order</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m^\alpha )</td>
<td>1</td>
<td>( \tilde{O}(m^{1-\alpha}) )</td>
<td>Advers.</td>
<td>[Greenwald, Khanna '01]</td>
</tr>
<tr>
<td>( \tilde{O}(m^{1/2}) )</td>
<td>1</td>
<td>( \tilde{O}(1) )</td>
<td>Random</td>
<td>[Guha, McGregor '06]</td>
</tr>
</tbody>
</table>
Previous Work

- **t-approx median:** Any element of rank = m/2±t

- **Previous Work (The Story So Far...):**

<table>
<thead>
<tr>
<th>Approx.</th>
<th>Passes</th>
<th>Space</th>
<th>Order</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>m^α</td>
<td>1</td>
<td>Õ(m^{1-α})</td>
<td>Advers.</td>
<td>[Greenwald, Khanna '01]</td>
</tr>
<tr>
<td>Õ(m^{1/2})</td>
<td>1</td>
<td>Õ(1)</td>
<td>Random</td>
<td>[Guha, McGregor '06]</td>
</tr>
<tr>
<td>Exact</td>
<td>O(lg m/lg lg m)</td>
<td>Õ(1)</td>
<td>Advers.</td>
<td>[Munro, Paterson '78]</td>
</tr>
</tbody>
</table>
**Previous Work**

- **t-approx median:** Any element of rank $= m/2 \pm t$

- **Previous Work (The Story So Far...):**

<table>
<thead>
<tr>
<th>Approx.</th>
<th>Passes</th>
<th>Space</th>
<th>Order</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m^\alpha$</td>
<td>1</td>
<td>$\tilde{O}(m^{1-\alpha})$</td>
<td>Advers.</td>
<td>[Greenwald, Khanna ‘01]</td>
</tr>
<tr>
<td>$\tilde{O}(m^{1/2})$</td>
<td>1</td>
<td>$\tilde{O}(1)$</td>
<td>Random</td>
<td>[Guha, McGregor ‘06]</td>
</tr>
<tr>
<td>Exact</td>
<td>$O(lg m/lg lg m)$</td>
<td>$\tilde{O}(1)$</td>
<td>Advers.</td>
<td>[Munro, Paterson ‘78]</td>
</tr>
<tr>
<td>Exact</td>
<td>$O(lg lg m)$</td>
<td>$\tilde{O}(1)$</td>
<td>Random</td>
<td>[Guha, McGregor ‘06]</td>
</tr>
</tbody>
</table>
**Previous Work**

- **t-approx median:** Any element of rank = \( m/2\pm t \)

---

**Previous Work (The Story So Far...):**

<table>
<thead>
<tr>
<th>Approx.</th>
<th>Passes</th>
<th>Space</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m^\alpha )</td>
<td>1</td>
<td>( \tilde{O}(m^{1-\alpha}) )</td>
<td>Advers. [Greenwald, Khanna '01]</td>
</tr>
<tr>
<td>( \tilde{O}(m^{1/2}) )</td>
<td>1</td>
<td>( \tilde{O}(1) )</td>
<td>Random [Guha, McGregor '06]</td>
</tr>
<tr>
<td>Exact</td>
<td>( O(\lg m/\lg \lg m) )</td>
<td>( \tilde{O}(1) )</td>
<td>Advers. [Munro, Paterson '78]</td>
</tr>
<tr>
<td>Exact</td>
<td>( O(\lg \lg m) )</td>
<td>( \tilde{O}(1) )</td>
<td>Random [Guha, McGregor '06]</td>
</tr>
<tr>
<td>Exact</td>
<td>1</td>
<td>( \tilde{O}(m^{1/2}) )</td>
<td>Random [Munro, Paterson '78]</td>
</tr>
</tbody>
</table>
Previous Work

- **$t$-approx median:** Any element of rank $= m/2 \pm t$

**Previous Work (The Story So Far...):**

<table>
<thead>
<tr>
<th>Approx.</th>
<th>Passes</th>
<th>Space</th>
<th>Order</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m^\alpha$</td>
<td>1</td>
<td>$\tilde{O}(m^{1-\alpha})$</td>
<td>Advers.</td>
<td>[Greenwald, Khanna '01]</td>
</tr>
<tr>
<td>$\tilde{O}(m^{1/2})$</td>
<td>1</td>
<td>$\tilde{O}(1)$</td>
<td>Random</td>
<td>[Guha, McGregor '06]</td>
</tr>
<tr>
<td>Exact</td>
<td>$O(\lg m/\lg \lg m)$</td>
<td>$\tilde{O}(1)$</td>
<td>Advers.</td>
<td>[Munro, Paterson '78]</td>
</tr>
<tr>
<td>Exact</td>
<td>$O(\lg \lg m)$</td>
<td>$\tilde{O}(1)$</td>
<td>Random</td>
<td>[Guha, McGregor '06]</td>
</tr>
<tr>
<td>Exact</td>
<td>1</td>
<td>$\tilde{O}(m^{1/2})$</td>
<td>Random</td>
<td>[Munro, Paterson '78]</td>
</tr>
</tbody>
</table>

**Questions:** Is the exponential separation in passes “real”? Can the random order results be improved?
**t-approx median:** Any element of rank $= m/2 \pm t$

**Previous Work (The Story So Far...):**

<table>
<thead>
<tr>
<th>Approx.</th>
<th>Passes</th>
<th>Space</th>
<th>Order</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m^\alpha$</td>
<td>1</td>
<td>$\tilde{O}(m^{1-\alpha})$</td>
<td>Advers.</td>
<td>[Greenwald, Khanna '01]</td>
</tr>
<tr>
<td>$\tilde{O}(m^{1/2})$</td>
<td>1</td>
<td>$\tilde{O}(1)$</td>
<td>Random</td>
<td>[Guha, McGregor '06]</td>
</tr>
<tr>
<td>Exact</td>
<td>$O(lg m/lg lg m)$</td>
<td>$\tilde{O}(1)$</td>
<td>Advers.</td>
<td>[Munro, Paterson '78]</td>
</tr>
<tr>
<td>Exact</td>
<td>$O(lg lg m)$</td>
<td>$\tilde{O}(1)$</td>
<td>Random</td>
<td>[Guha, McGregor '06]</td>
</tr>
<tr>
<td>Exact</td>
<td>1</td>
<td>$\tilde{O}(m^{1/2})$</td>
<td>Random</td>
<td>[Munro, Paterson '78]</td>
</tr>
</tbody>
</table>

**Questions:** Is the exponential separation in passes “real”? Can the random order results be improved?

**Today’s Results:**

| Exact   | $\Omega(lg m/lg lg m)$ | $\tilde{O}(1)$       | Advers.   | [Guha, McGregor '07]      |
| Exact   | 1                    | $\Omega(m^{1/2})$    | Random    | [Guha, McGregor '07]      |
Random-Order LB
Random-Order LB

- **Thm:** Finding the median of a randomly ordered stream (w/ p 3/4) requires $\Omega(m^{1/2})$ space.
Random-Order LB

• **Thm:** Finding the median of a randomly ordered stream (w/ p 3/4) requires $\Omega(m^{1/2})$ space.

• **Proof:**
  Reduction from size $t$ instance of $\text{INDEX}(x,j)$:

  - **Special Items:** $\{2i+x_i : i=1, \ldots, t\}$
  - **Small Items:** $(m+1)/2-j$ copies of “0”
  - **Large Items:** $(m-1)/2-t+j$ copies of “2t+x_t”
  - **Bias:** $j=|\# \text{ Small Items} - \# \text{ Large Items}|/2$
Random-Order LB

• **Thm:** Finding the median of a randomly ordered stream (w/ p 3/4) requires $\Omega(m^{1/2})$ space.

• **Proof:**

  Reduction from size $t$ instance of $\text{INDEX}(x,j)$:

  - **Special Items:** $\{2i+x_i : i=1, \ldots, t\}$
  - **Small Items:** $(m+1)/2-j$ copies of “0”
  - **Large Items:** $(m-1)/2-t+j$ copies of “$2t+x_t$”
  - **Bias:** $j=|# \text{ Small Items} - # \text{ Large Items}|/2$

  Assume a $t$-space alg. $A$ that finds the median of a randomly ordered stream w/p 3/4.
Random-Order LB

- **Thm:** Finding the median of a randomly ordered stream (w/p 3/4) requires $\Omega(m^{1/2})$ space.

- **Proof:**
  Reduction from size t instance of $\text{INDEX}(x,j)$:
  
  *Special Items:* $\{2i+x_i : i=1, ..., t\}$
  
  *Small Items:* $(m+1)/2-j$ copies of “0”
  
  *Large Items:* $(m-1)/2-t+j$ copies of “2t+x_t”
  
  *Bias:* $j=|\# \text{ Small Items} - \# \text{ Large Items}|/2$

  Assume a t-space alg. $A$ that finds the median of a randomly ordered stream w/p 3/4.

  **Key Challenge:** Must simulate alg. on random perm.
• **Thm:** Finding the median of a randomly ordered stream (w/ p 3/4) requires $\Omega(m^{1/2})$ space.

• **Proof:**

Reduction from size $t$ instance of $\text{INDEX}(x,j)$:

- **Special Items:** $\{2i+x_i : i=1, \ldots, t\}$
- **Small Items:** $(m+1)/2-j$ copies of “0”
- **Large Items:** $(m-1)/2-t+j$ copies of “2t+x_t”
- **Bias:** $j=|#\text{ Small Items} - #\text{ Large Items}|/2$

Assume a $t$-space alg. $\mathcal{A}$ that finds the median of a randomly ordered stream w/p 3/4.

**Key Challenge:** Must simulate alg. on random perm.

**Solution:** Make perm. “almost random” by choosing $t$ small and allow Alice to determine all but last “few” elements...
Alice

length $t$

binary string $x$

Bob

index $i$ in range $[t]$
**Alice**: Randomly guess bias $b$ in range $[t]$. Run algorithm on a random perm. of,
$$\{ 0, \ldots , 0, 2 + x_1, \ldots , 2t + x_t, 2t + 2, \ldots , 2t + 2 \}$$

\[ \frac{(m+1-m_2)}{2-b} \]

**Bob**
index $i$ in range $[t]$
**Alice**: Randomly guess bias $b$ in range $[t]$. Run algorithm on a random perm. of,

$$\{0, \ldots, 0, 2 + x_1, \ldots, 2t + x_t, 2t + 2, \ldots, 2t + 2\}$$

\[
\left(\frac{m+1-m_2}{2}-b\right)
\]

\[
\left(\frac{m+1-m_2}{2}+b-t-1\right)
\]

**Memory State of Algorithm and “$b$”**

**Alice**
- length $t$
- binary string $x$

**Bob**
- index $i$ in range $[t]$
**Alice**: Randomly guess bias $b$ in range $[t]$. Run algorithm on a random perm. of,

$$\{ 0, \ldots, 0, 2 + x_1, \ldots, 2t + x_t, 2t + 2, \ldots, 2t + 2 \}$$

$$\frac{(m+1-m_2)}{2-b}$$

**Bob**: inserts a random permutation of,

$$\{ 0, \ldots, 0, 2t + 2, \ldots, 2t + 2 \}$$

$$\frac{m_2}{2+b-i}$$

**MEMORY STATE OF ALGORITHM** and “$b$”

**Alice** length $t$ binary string $x$

**Bob** index $i$ in range $[t]$
Alice: Randomly guess bias \( b \) in range \([t]\). Run algorithm on a random perm. of,

\[
\{ 0, \ldots, 0, 2 + x_1, \ldots, 2t + x_t, 2t + 2, \ldots, 2t + 2 \}
\]

\[
(m + 1 - m_2)/2 - b
\]

Bob: inserts a random permutation of,

\[
\{ 0, \ldots, 0, 2t + 2, \ldots, 2t + 2 \}
\]

\[
m_2/2 + b - i
\]

\[
m_2/2 - b + i
\]

• If \( t \ll \sqrt{m_2} \) then bias \( b \) is not apparent
Alice: Randomly guess bias $b$ in range $[t]$. Run algorithm on a random perm. of,

$$\{0, \ldots, 0, 2 + x_1, \ldots, 2t + x_t, 2t + 2, \ldots, 2t + 2\}$$

Bob: inserts a random permutation of,

$$\{0, \ldots, 0, 2t + 2, \ldots, 2t + 2\}$$

- If $t << \sqrt{m_2}$ then bias $b$ is not apparent
- If $t m_2 << m$ then no special elements in suffix.
Alice:

Randomly guess bias $b$ in range $[t]$. Run algorithm on a random perm. of,

$$\{ 0, \ldots, 0, 2 + x_1, \ldots, 2t + x_t, 2t + 2, \ldots, 2t + 2 \}$$

$$\frac{(m+1-m_2)}{2-b}$$

Bob:

inserts a random permutation of,

$$\{ 0, \ldots, 0, 2t + 2, \ldots, 2t + 2 \}$$

$$\frac{m_2}{2+b-i}$$

• If $t<<\sqrt{m_2}$ then bias $b$ is not apparent

• If $t m_2<<m$ then no special elements in suffix.

• If $t=O(m^{1/3})$ and $m_2=O(m^{2/3})$ then $A$ succeeds w/p $2/3$ since ordering is “random enough.”
Alice

length $t$

binary string $x$

Bob

index $i$ in range $[t]$.

**Alice:** Randomly guess bias $b$ in range $[t]$. Run algorithm on a random perm. of,

\[
\{0, \ldots, 0, 2 + x_1, \ldots, 2t + x_t, 2t + 2, \ldots, 2t + 2\}
\]

(memory state of algorithm and "b")

**Bob:** inserts a random permutation of,

\[
\{0, \ldots, 0, 2t + 2, \ldots, 2t + 2\}
\]

- If $t<<\sqrt{m_2}$ then bias $b$ is not apparent
- If $t \cdot m_2<<m$ then no special elements in suffix.
- If $t=\mathcal{O}(m^{1/3})$ and $m_2=\mathcal{O}(m^{2/3})$ then $A$ succeeds w/p $2/3$ since ordering is "random enough."
- **Thm:** Space required is $\Omega(m^{1/3})$
Random-Order LB
Random-Order LB

... can be extended to:

- **Thm:** Finding the median of a randomly ordered stream (with probability at least 3/4) requires $\Omega(m^{1/2-\delta})$ space.
Random-Order LB

... can be extended to:

- **Thm:** Finding the median of a randomly ordered stream (with probability at least 3/4) requires $\Omega(m^{1/2-\delta})$ space.

- **Open Questions:**
  ? Extending to multiple passes: is $\Omega(lg lg m)$ passes required when we only have polylog($m,n$) space?
Random-Order LB

... can be extended to:

• **Thm:** Finding the median of a randomly ordered stream (with probability at least 3/4) requires $\Omega(m^{1/2-\delta})$ space.

• **Open Questions:**
  
  ? Extending to multiple passes: is $\Omega(lg lg m)$ passes required when we only have polylog($m, n$) space?

  **BREAKING NEWS:** See SODA ’08 [Chakrabarti, Jayram, Patrascu ’08]
Random-Order LB

... can be extended to:

• **Thm:** Finding the median of a randomly ordered stream (with probability at least 3/4) requires $\Omega(m^{1/2-\delta})$ space.

• **Open Questions:**
  ? Extending to multiple passes: is $\Omega(\lg \lg m)$ passes required when we only have polylog$(m,n)$ space?

  **BREAKING NEWS:** See SODA '08 [Chakrabarti, Jayram, Patrascu '08]

  ? For exact selection, result is tight. But more generally...?
Adversarial-Order LB

- **Thm:** Find the median of an adversarially ordered stream in $p$ passes requires $\Omega(m^{1/p})$ space.
Adversarial-Order LB

- **Thm:** Find the median of an adversarially ordered stream in $p$ passes requires $\Omega(m^{1/p})$ space.

- **Proof Steps:**
Adversarial-Order LB

• **Thm:** Find the median of an adversarially ordered stream in \( p \) passes requires \( \Omega(m^{1/p}) \) space.

• **Proof Steps:**
  
  \( p \)-pass, \( (p+1) \)-level, \( t \)-ary Tree Pointer Chasing needs \( \Omega(t) \) space
Adversarial-Order LB

- **Thm:** Find the median of an adversarially ordered stream in $p$ passes requires $\Omega(m^{1/p})$ space.

- **Proof Steps:**
  
  - $p$-pass, $(p+1)$-level, $t$-ary Tree Pointer Chasing needs $\Omega(t)$ space
  
  Reduction from $(p+1)$-level, $m^{1/p}$-ary Tree Pointer Chasing.
Tree Pointer Chasing
Tree Pointer Chasing

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, 
  \(f(v)=i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node 
  \(f(v)\) in \(\{0,1\}\) if \(v\) is a leaf
Tree Pointer Chasing

- **Tree Pointers:** Function on nodes of \((p+1)\)-level, \(t\)-ary tree, 
  \(f(v) = i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node 
  \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf
Tree Pointer Chasing

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, \(f(v) = i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node.
  \(f(v)\) in \{0,1\} if \(v\) is a leaf.

\[
\begin{align*}
f(v_1) &= 0 & f(v_2) &= 1 & f(v_3) &= 1 & f(v_4) &= 1 & f(v_5) &= 0 & f(v_6) &= 1 & f(v_7) &= 1 & f(v_8) &= 0 & f(v_9) &= 1
\end{align*}
\]
Tree Pointer Chasing

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, \(f(v)=i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node
- \(f(v)\) in \(\{0,1\}\) if \(v\) is a leaf

\[
\begin{align*}
 f(v_1) &= 0 & f(v_2) &= 1 & f(v_3) &= 1 & f(v_4) &= 1 & f(v_5) &= 0 & f(v_6) &= 1 & f(v_7) &= 1 & f(v_8) &= 0 & f(v_9) &= 1 \\
 f(v_{10}) &= 1 & f(v_{11}) &= 3 & f(v_{12}) &= 1
\end{align*}
\]
Tree Pointer Chasing

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, 
  \(f(v)=i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node 
  \(f(v)\) in \(\{0,1\}\) if \(v\) is a leaf

\[
\begin{align*}
  f(v_1) &= 0 & f(v_2) &= 1 & f(v_3) &= 1 & f(v_4) &= 1 & f(v_5) &= 0 & f(v_6) &= 1 & f(v_7) &= 1 & f(v_8) &= 0 & f(v_9) &= 1 \\
  f(v_{10}) &= 1 & f(v_{11}) &= 3 & f(v_{12}) &= 1 & f(v_{13}) &= 3
\end{align*}
\]
Tree Pointer Chasing

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, \(f(v)=i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node, \(f(v)\) in \(\{0,1\}\) if \(v\) is a leaf.

- **Level-by-level Problem**: Compute \(f(f(... f(v_{\text{root}})....))\), i.e., follow the pointer from root to leaf, when the data is level-by-level, in the wrong order: \(f(v_1), f(v_2), f(v_3), f(v_4), ..., f(v_{13})\)

\[
\begin{align*}
\text{f(v)} & \text{= value of child}
\end{align*}
\]
Tree Pointer Chasing

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, 
  \(f(v) = i\), specifies \(i\)th child of \(v\) if \(v\) is an internal node, 
  \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf

- **Level-by-level Problem**: Compute \(f(f(\ldots f(v_{\text{root}})\ldots))\), i.e., follow the 
  pointer from root to leaf, when the data is level-by-level, in 
  the wrong order: \(f(v_1), f(v_2), f(v_3), f(v_4), \ldots, f(v_{13})\)

\[
\begin{align*}
f(v_1) &= 0 \\
f(v_2) &= 1 \\
f(v_3) &= 1 \\
f(v_4) &= 1 \\
f(v_5) &= 0 \\
f(v_6) &= 1 \\
f(v_7) &= 1 \\
f(v_8) &= 0 \\
f(v_9) &= 1 \\
f(v_{10}) &= 1 \\
f(v_{11}) &= 3 \\
f(v_{12}) &= 1 \\
f(v_{13}) &= 3
\end{align*}
\]
Tree Pointer Chasing

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, \(f(v)=i\), specifies \(i\)th child of \(v\) if \(v\) is an internal node, \(f(v)\) in \(\{0,1\}\) if \(v\) is a leaf.

- **Level-by-level Problem**: Compute \(f(f(... f(v_{\text{root}})....))\), i.e., follow the pointer from root to leaf, when the data is level-by-level, *in the wrong order*: \(f(v_1), f(v_2), f(v_3), f(v_4), ... , f(v_{13})\)

\[\begin{align*}
f(v_1) &= 0 \\
f(v_2) &= 1 \\
f(v_3) &= 1 \\
f(v_4) &= 1 \\
f(v_5) &= 0 \\
f(v_6) &= 1 \\
f(v_7) &= 1 \\
f(v_8) &= 0 \\
f(v_9) &= 1 \\
f(v_{10}) &= 1 \\
f(v_{11}) &= 3 \\
f(v_{12}) &= 1 \\
f(v_{13}) &= 3
\end{align*}\]
Tree Pointer Chasing

- **Tree Pointers:** Function on nodes of \((p+1)\)-level, \(t\)-ary tree, \(f(v)=i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node \(f(v)\) in \(\{0,1\}\) if \(v\) is a leaf

- **Level-by-level Problem:** Compute \(f(f(... f(v_{\text{root}})....))\), i.e., follow the pointer from root to leaf, when the data is level-by-level, in the wrong order: \(f(v_1), f(v_2), f(v_3), f(v_4), \ldots, f(v_{13})\)

- **Thm:** Any \(p\)-pass algorithm requires \(\Omega(t)\) space (\(p\) small).

![Diagram showing tree pointer chasing with node values]

- \(f(v_{13})=3\)
- \(f(v_{10})=1\)
- \(f(v_{11})=3\)
- \(f(v_{12})=1\)
- \(f(v_1)=0\)
- \(f(v_2)=1\)
- \(f(v_3)=1\)
- \(f(v_4)=1\)
- \(f(v_5)=0\)
- \(f(v_6)=1\)
- \(f(v_7)=1\)
- \(f(v_8)=0\)
- \(f(v_9)=1\)
Reduction to Selection
Reduction to Selection

- Reconsider INDEX problem as 2-level tree pointer chasing:
Reduction to Selection

- Reconsider INDEX problem as 2-level tree pointer chasing:

\[ f(v_1) = 1 \quad f(v_2) = 0 \quad f(v_3) = 0 \quad f(v_4) = 1 \quad f(v_5) = 0 \]

\[ f(v_6) = 2 \]

Corresponds to \( x = 10010 \) and \( j = 2 \).
Reduction to Selection

- Reconsider INDEX problem as 2-level tree pointer chasing:

\[ f(v_1) = 1 \quad f(v_2) = 0 \quad f(v_3) = 0 \quad f(v_4) = 1 \quad f(v_5) = 0 \]

\[ f(v_6) = 2 \]

corresponds to \( x = 10010 \) and \( j = 2 \).
Reduction to Selection

- Reconsider INDEX problem as 2-level tree pointer chasing:
  \[ f(v_6) = 2 \]
  \[ f(v_1) = 1 \]
  \[ f(v_2) = 0 \]
  \[ f(v_3) = 0 \]
  \[ f(v_4) = 1 \]
  \[ f(v_5) = 0 \]

  corresponds to \( x = 10010 \) and \( j = 2 \).

- \textbf{Special Items:} \( i^{th} \) leaf becomes \( i(t+2) + x_i \)
Reduction to Selection

- Reconsider INDEX problem as 2-level tree pointer chasing:

  \[ f(v_6) = 2 \]

  \[
  \begin{array}{l}
  f(v_1) = 1 \\
  f(v_2) = 0 \\
  f(v_3) = 0 \\
  f(v_4) = 1 \\
  f(v_5) = 0
  \end{array}
  \]

  corresponds to \( x = 10010 \) and \( j = 2 \).

- **Special Items:** \( i^{th} \) leaf becomes \( i(t+2)+x_i \)

- **Large/Small Items:** Root becomes \((j-1)\) copies of “0” and \((t-j)\) copies of “(t+1)(t+2)”
Reduction to Selection

• Reconsider INDEX problem as 2-level tree pointer chasing:

```
Reconstruction process:
```

- **Special Items**: $i$th leaf becomes $i(t+2)+x_i$
- **Large/Small Items**: Root becomes $(j-1)$ copies of “0” and $(t-j)$ copies of “$(t+1)(t+2)$”

• **Example**:
  - Tree-Traversal Stream: 1,0,0,1,0,2
  - Induced Stream: 8, 14, 21, 29, 35, 0, 0, 0, 42
Reduction to Selection

• Reconsider INDEX problem as 2-level tree pointer chasing:

  \[ f(v_6) = 2 \]
  \[ f(v_1) = 1 \]
  \[ f(v_2) = 0 \]
  \[ f(v_3) = 0 \]
  \[ f(v_4) = 1 \]
  \[ f(v_5) = 0 \]

  corresponds to \( x = 10010 \) and \( j = 2 \).

• **Special Items**: \( i^{th} \) leaf becomes \( i(t+2) + x_i \)

• **Large/Small Items**: Root becomes \( (j-1) \) copies of “0” and \( (t-j) \) copies of “(t+1)(t+2)”

• **Example**:
  * Tree-Traversal Stream*: 1, 0, 0, 1, 0, 2
  * Induced Stream*: 8, 14, 21, 29, 35, 0, 0, 0, 42

• ... extend to deeper trees via “representation in base \( (t+2) \)”
Reduction to Selection

\[
\begin{align*}
    f(v_1) &= 0 \\
    f(v_2) &= 1 \\
    f(v_3) &= 1 \\
    f(v_4) &= 1 \\
    f(v_5) &= 0 \\
    f(v_6) &= 1 \\
    f(v_7) &= 1 \\
    f(v_8) &= 0 \\
    f(v_9) &= 1
\end{align*}
\]
Reduction to Selection

- Let \((a_p, ..., a_0)\) denote \(a_p(t+2)^p + a_{p-1}(t+2)^{p-1} + ... + a_0(t+2)^0\)
- Let \(v[i_p,...,i_j]\) denote the \(i_j\)-th child of \(v[i_p,...,i_{j-1}]\) where \(v[]\) is \(v_{\text{root}}\)
Reduction to Selection

- Let \((a_p, \ldots, a_0)\) denote \(a_p(t+2)^p + a_{p-1}(t+2)^{p-1} + \ldots + a_0(t+2)^0\)
- Let \(v[i_p, \ldots, i_j]\) denote the \(i_j\)-th child of \(v[i_p, \ldots, i_{j-1}]\) where \(v[\cdot]\) is \(v_{\text{root}}\)
- For leaf \(v[i_p, \ldots, i_l]\), replace \(f(v)\) by \((i_p, \ldots, i_l, f(v))\)
Reduction to Selection

- Let \((a_p, ..., a_0)\) denote \(a_p(t+2)^p + a_{p-1}(t+2)^{p-1} + ... + a_0(t+2)^0\)
- Let \(v[i_p, ..., i_j]\) denote \(i_j\)-th child of \(v[i_p, ..., i_{j-1}]\) where \(v[]\) is \(v_{\text{root}}\)
- For leaf \(v[i_p, ..., i_1]\), replace \(f(v)\) by \((i_p, ..., i_1, f(v))\)
- For internal node \(v[i_p, ..., i_j]\), replace \(f(v)\) by:
  \[ (f(v)-1)(2t-1)^{j-2} \text{ copies of } (i_p, ..., i_j, 0, ..., 0) \]
  \[ (t-f(v))(2t-1)^{j-2} \text{ copies of } (i_p, ..., i_j, t+1, ..., 0) \]

\[
\begin{array}{c}
\text{f}(v_{10})=2 \\
\text{f}(v_{11})=3 \\
\text{f}(v_{12})=1 \\
\text{f}(v_{13})=2 \\
\text{f}(v_1)=0 \\
\text{f}(v_2)=1 \\
\text{f}(v_3)=1 \\
\text{f}(v_4)=1 \\
\text{f}(v_5)=0 \\
\text{f}(v_6)=1 \\
\text{f}(v_7)=1 \\
\text{f}(v_8)=0 \\
\text{f}(v_9)=1 \\
\end{array}
\]
• Let \((a_p, ..., a_0)\) denote \(a_p(t+2)^p + a_{p-1}(t+2)^{p-1} + ... + a_0(t+2)^0\)

• Let \(v[i_p, ..., i_j]\) denote \(i_j\)-th child of \(v[i_p, ..., i_{j-1}]\) where \(v[]\) is \(v_{\text{root}}\)

• For leaf \(v[i_p, ..., i_l]\), replace \(f(v)\) by \((i_p, ..., i_l, f(v))\)

• For internal node \(v[i_p, ..., i_j]\), replace \(f(v)\) by:
  \((f(v)-1)(2t-1)^{j-2}\) copies of \((i_p, ..., i_j, 0, ..., 0)\)
  \((t-f(v))(2t-1)^{j-2}\) copies of \((i_p, ..., i_j, t+1, ..., 0)\)
• Let \((a_p, \ldots, a_0)\) denote \(a_p(t+2)^p + a_{p-1}(t+2)^{p-1} + \cdots + a_0(t+2)^0\)

• Let \(v[i_p, \ldots, i_j]\) denote \(i_j\)-th child of \(v[i_p, \ldots, i_{j-1}]\) where \(v[]\) is \(v_{\text{root}}\)

• For leaf \(v[i_p, \ldots, i_l]\), replace \(f(v)\) by \((i_p, \ldots, i_l, f(v))\)

• For internal node \(v[i_p, \ldots, i_j]\), replace \(f(v)\) by:
  \[(f(v)-1)(2t-1)^{j-2} \text{ copies of } (i_p, \ldots, i_j, 0, \ldots, 0)\]
  \[(t-f(v))(2t-1)^{j-2} \text{ copies of } (i_p, \ldots, i_j, t+1, \ldots, 0)\]
Reduction to Selection

• Let \((a_p, \ldots, a_0)\) denote \(a_p(t+2)^p + a_{p-1}(t+2)^{p-1} + \ldots + a_0(t+2)^0\)

• Let \(v[i_p,\ldots,i_j]\) denote \(i_j\)-th child of \(v[i_p,\ldots,i_{j-1}]\) where \(v[]\) is \(v_{\text{root}}\)

• For leaf \(v[i_p, \ldots, i_l]\), replace \(f(v)\) by \((i_p, \ldots, i_l, f(v))\)

• For internal node \(v[i_p, \ldots, i_j]\), replace \(f(v)\) by:

  \[(f(v)-1)(2t-1)^{j-2} \text{ copies of } (i_p, \ldots, i_j, 0,\ldots, 0)\]

  \[5 \times (0,0,0), 5 \times (4,0,0)\]

  \[(2,4,0), (2,4,0)\]

  \[(3,0,0), (3,0,0)\]

  \[(3,3,1)\]

  \[(3,1,1)\]

  \[(3,2,0)\]

  \[(1,1,0)\]

  \[(1,2,1)\]

  \[(1,3,1)\]

  \[(2,1,1)\]

  \[(2,2,0)\]

  \[(2,3,1)\]

  \[(3,1,1)\]

  \[(3,2,0)\]

  \[(3,3,1)\]

  \[(5,0,0), 5 \times (4,0,0)\]
Summary

• **Thm:** Find an $m^\delta$-approx median of a randomly ordered stream requires $\Omega(m^{1/2-3\delta/2})$ space.

  $\therefore$ Algorithm for exact selection is essentially optimal.

• **Thm:** Find an $m^\delta$-approx median of a adversarially ordered stream in $p$ passes requires $\Omega(m^{(1-\delta)/p})$ space.
Summary

• **Thm:** Find an $m^\delta$-approx median of a randomly ordered stream requires $\Omega(m^{1/2-3\delta/2})$ space.

  \[ \therefore \] Algorithm for exact selection is essentially optimal.

• **Thm:** Find an $m^\delta$-approx median of a adversarially ordered stream in $p$ passes requires $\Omega(m^{(1-\delta)/p})$ space.

  \[ \therefore \] Selection in $\tilde{O}(1)$ space needs $\Omega(lg m/lg \ lg m)$ passes while $O(lg \ lg m)$ passes suffices for random-order.
Summary

• **Thm:** Find an $m^{\delta}$-approx median of a randomly ordered stream requires $\Omega(m^{1/2-3\delta/2})$ space.

  \[ \therefore \text{ Algorithm for exact selection is essentially optimal.} \]

• **Thm:** Find an $m^{\delta}$-approx median of a adversarially ordered stream in $p$ passes requires $\Omega(m^{(1-\delta)/p})$ space.

  \[ \therefore \text{ Selection in } \tilde{O}(1) \text{ space needs } \Omega(lg \frac{m}{lg lg m}) \text{ passes while } O(lg lg m) \text{ passes suffices for random-order.} \]

• **Further Directions**
  Is improvement as dramatic for other problems?
  Notions of “semi-random” order?
  Trade-offs between space and sample-complexity...
1: Selection  
2: Tree Traversal  
3: Applications
2: Tree Traversal

a) Post-Order Tree Traversal LB
b) Pass Elimination Lemma
Post-Order Traversal
Post-Order Traversal

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, 
  \(f(v) = i\), specifies \(i\)th child of \(v\) if \(v\) is an internal node
  \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf
Post-Order Traversal

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, \(f(v)=i\), specifies \(i\)th child of \(v\) if \(v\) is an internal node, \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf

- **Post-Order Traversal**: Compute \(f(f(... f(v_{\text{root}})....))\), when the data is ordered as a post-order traversal:
\[ f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13}) \]
Post-Order Traversal

- **Tree Pointers:** Function on nodes of \((p+1)\)-level, t-ary tree, \(f(v) = i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node
  \(f(v)\) in \([0, 1]\) if \(v\) is a leaf

- **Post-Order Traversal:** Compute \(f(f(\ldots f(v_{\text{root}})\ldots))\), when the data is ordered as a post-order traversal:
  \(f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\)
• **Tree Pointers:** Function on nodes of \((p+1)\)-level, \(t\)-ary tree,
  \(f(v) = i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node
  \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf

\[
\begin{align*}
  f(v_1) &= 0 \\
  f(v_2) &= 1
\end{align*}
\]

• **Post-Order Traversal:** Compute \(f(f(\ldots f(v_{root}) \ldots ))\), when the data is ordered as a post-order traversal:
  \(f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\)
Post-Order Traversal

- **Tree Pointers:** Function on nodes of \((p+1)\)-level, t-ary tree, \(f(v)=i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node
- \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf

- **Post-Order Traversal:** Compute \(f(f(\ldots f(v_{\text{root}})\ldots))\), when the data is ordered as a post-order traversal:

\(f(v_1), f(v_2), f(v_3), f(v_10), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\)
• **Tree Pointers:** Function on nodes of \((p+1)\)-level, t-ary tree, 
  
  \[ f(v) = i, \text{ specifies } i^{th} \text{ child of } v \text{ if } v \text{ is an internal node} \]
  
  \[ f(v) \in \{0, 1\} \text{ if } v \text{ is a leaf} \]

- **Post-Order Traversal:** Compute \(f(f(\ldots f(v_{\text{root}})\ldots)))\), when the data is ordered as a post-order traversal:
  
  \[ f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13}) \]
• **Tree Pointers:** Function on nodes of \((p+1)\)-level, t-ary tree, 
  \(f(v) = i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node 
  \(f(v)\) in \(\{0,1\}\) if \(v\) is a leaf

- **Post-Order Traversal:** Compute \(f(f(\ldots f(v_{\text{root}})\ldots))\), when the data 
  is ordered as a post-order traversal: 
  \(f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\)
• **Tree Pointers:** Function on nodes of \((p+1)\)-level, \(t\)-ary tree, 
\(f(v)=i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node 
\(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf

```
1: f(v_{10})=1
2: f(v_1)=0 f(v_2)=1 f(v_3)=1 f(v_4)=1 f(v_5)=0
```

• **Post-Order Traversal:** Compute \(f(f(... f(v_{\text{root}})....))\), when the data is ordered as a post-order traversal:

\[ f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13}) \]
Post-Order Traversal

- **Tree Pointers:** Function on nodes of \((p+1)\)-level, \(t\)-ary tree, 
  \(f(v) = i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node 
  \(f(v)\) in \(\{0,1\}\) if \(v\) is a leaf

![Tree Diagram]

- **Post-Order Traversal:** Compute \(f(f(... f(v_{root}) ....))\), when the data 
  is ordered as a post-order traversal:
  \(f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\)
Post-Order Traversal

- **Tree Pointers:** Function on nodes of \((p+1)\)-level, \(t\)-ary tree, 
  \(f(v) = i\), specifies \(i\)th child of \(v\) if \(v\) is an internal node
  \(f(v) \) in \(\{0, 1\}\) if \(v\) is a leaf

- **Post-Order Traversal:** Compute \(f(f(... f(v_{\text{root}})....))\), when the data is ordered as a post-order traversal:
  \(f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\)
Post-Order Traversal

- **Tree Pointers:** Function on nodes of \((p+1)\)-level, \(t\)-ary tree, 
  \(f(v) = i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node 
  \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf

- **Post-Order Traversal:** Compute \(f(f(... f(v_{\text{root}})....))\), when the data 
is ordered as a post-order traversal:
  \(f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\)
Post-Order Traversal

- **Tree Pointers:** Function on nodes of \((p+1)\)-level, t-ary tree, \(f(v) = i\), specifies \(i\)th child of \(v\) if \(v\) is an internal node \(f(v) \in \{0, 1\}\) if \(v\) is a leaf

- **Post-Order Traversal:** Compute \(f(f(... f(v_{\text{root}}) ...))\), when the data is ordered as a post-order traversal:
  \[ f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13}) \]
Post-Order Traversal

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, t-ary tree, 
  \(f(v) = i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node 
  \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf

- **Post-Order Traversal**: Compute 
  \(f(f(... f(v_{\text{root}})....))\), when the data is ordered as a post-order traversal:
  
  \[f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\]
Post-Order Traversal

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, t-ary tree, \(f(v)=i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node
  \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf

- **Post-Order Traversal**: Compute \(f(f(\ldots f(v_{\text{root}})\ldots))\), when the data is ordered as a *post-order traversal*:
  \(f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\)
Post-Order Traversal

- **Tree Pointers**: Function on nodes of \((p+1)\)-level, \(t\)-ary tree, \(f(v)=i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node. \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf.

- **Post-Order Traversal**: Compute \(f(f(\ldots f(v_{\text{root}})\ldots))\), when the data is ordered as a post-order traversal:
  \[f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})\]
Post-Order Traversal

- **Tree Pointers:** Function on nodes of \((p+1)\)-level, \(t\)-ary tree, \(f(v) = i\), specifies \(i^{th}\) child of \(v\) if \(v\) is an internal node, \(f(v)\) in \(\{0, 1\}\) if \(v\) is a leaf.

- **Post-Order Traversal:** Compute \(f(f(\ldots f(v_{\text{root}})\ldots))\), when the data is ordered as a post-order traversal:

\[
f(v_1), f(v_2), f(v_3), f(v_{10}), f(v_4), f(v_5), f(v_6), f(v_{11}), f(v_7), f(v_8), f(v_9), f(v_{12}), f(v_{13})
\]

- ... will prove space-complexity via “pass-elimination.”
Pass-Elimination Lemma

cf. Round-Elimination [Miltersen, Nisan, Safra, Wigderson ’98], [Sen ’03]
Pass-Elimination Lemma
cf. Round-Elimination [Miltersen, Nisan, Safra, Wigderson ’98], [Sen ’03]

- **Meta-Problem:** Let $f$ be a function on $[n]^d$. For $x^i$ from $[n]^d$, $P_{t,f} (x^1, \ldots, x^t,i) = f(x^i)$
Pass-Elimination Lemma

cf. Round-Elimination [Miltersen, Nisan, Safra, Wigderson '98], [Sen '03]

• **Meta-Problem:** Let \( f \) be a function on \([n]^d\). For \( x^i \) from \([n]^d\),

\[
P_{t,f}(x^1, ..., x^t,i) = f(x^i)
\]

• **Pass-Elimination Lemma:** If \( t/s \) is large constant:

  If there's a \( p \)-pass, \( s \)-space alg. \( A \) for \( P_{t,f} \) (w/p 1-\( \delta \)) then

  there's a \((p-1)\)-pass, \((2s \lg \delta^{-1})\)-space alg. \( A' \) for \( f \) (w/p 1-\( \delta \)).
Pass-Elimination Lemma

cf. Round-Elimination [Miltersen, Nisan, Safra, Wigderson ’98], [Sen ’03]

- **Meta-Problem:** Let $f$ be a function on $[n]^d$. For $x^i$ from $[n]^d$,  
  $P_{t,f}(x^1, ..., x^t,i)=f(x^i)$

- **Pass-Elimination Lemma:** If $t/s$ is large constant:
  If there’s a $p$-pass, $s$-space alg. $A$ for $P_{t,f}$ (w/p $1-\delta$) then
  there’s a $(p-1)$-pass, $(2s \log \delta^{-1})$-space alg. $A'$ for $f$ (w/p $1-\delta$).

- **Thm:** A $p$-pass alg. for $(p+1)$-level, $t$-ary PO-Traversal (w/p $2/3$) requires $\Omega(t/2^{O(p)})$ space.

- **Proof:** Repeated elimination of passes implies:
Pass-Elimination Lemma
cf. Round-Elimination [Miltersen, Nisan, Safra, Wigderson ’98], [Sen ’03]

- **Meta-Problem:** Let \( f \) be a function on \([n]^d\). For \( x^i \) from \([n]^d\),
  \[ P_{t,f} (x^1, \ldots, x^t,i) = f(x^i) \]

- **Pass-Elimination Lemma:** If \( t/s \) is large constant:
  If there’s a \( p \)-pass, \( s \)-space alg. \( \mathcal{A} \) for \( P_{t,f} (w/p \ 1-\delta) \) then
  there’s a \((p-1)\)-pass, \((2s \log \delta^{-1})\)-space alg. \( \mathcal{A}'\) for \( f (w/p \ 1-\delta) \).

- **Thm:** A \( p \)-pass alg. for \((p+1)\)-level, \( t \)-ary PO-Traversal \((w/p \ 2/3)\) requires \( \Omega(t/2^{O(p)}) \) space.

- **Proof:** Repeated elimination of passes implies:
  \((p-1)\)-pass alg. for \( p \)-level in \( O(s \ 2^{O(1)}) \) space
Pass-Elimination Lemma
cf. Round-Elimination [Miltersen, Nisan, Safra, Wigderson ’98], [Sen ’03]

- **Meta-Problem:** Let $f$ be a function on $[n]^d$. For $x^i$ from $[n]^d$, $P_{t,f}(x^1, ..., x^t,i) = f(x^i)$

- **Pass-Elimination Lemma:** If $t/s$ is large constant:
  
  If there’s a $p$-pass, $s$-space alg. $A$ for $P_{t,f}(w/p 1-\delta)$ then
  there’s a $(p-1)$-pass, $(2s \log \delta^{-1})$-space alg. $A'$ for $f (w/p 1-\delta)$.

- **Thm:** A $p$-pass alg. for $(p+1)$-level, $t$-ary PO-Traversal (w/p 2/3) requires $\Omega(t/2^O(p))$ space.

- **Proof:** Repeated elimination of passes implies:
  
  $(p-1)$-pass alg. for $p$-level in $O(s 2^O(1))$ space
  .... 1-pass alg. for 2-level PO-Traversal in $O(s 2^O(p))$
Pass-Elimination Lemma
cf. Round-Elimination [Miltersen, Nisan, Safra, Wigderson ’98], [Sen ’03]

- **Meta-Problem:** Let \( f \) be a function on \([n]^d\). For \( x^i \) from \([n]^d\),
  \[ P_{t,f}(x^1, \ldots, x^t,i) = f(x^i) \]

- **Pass-Elimination Lemma:** If \( t/s \) is large constant:
  If there’s a \( p \)-pass, \( s \)-space alg. \( \mathcal{A} \) for \( P_{t,f}(w/p 1-\delta) \) then
  there’s a \((p-1)\)-pass, \((2s \log \delta^{-1})\)-space alg. \( \mathcal{A}' \) for \( f\) \((w/p 1-\delta)\).

- **Thm:** A \( p \)-pass alg. for \((p+1)\)-level, \( t \)-ary PO-Traversal \((w/p 2/3)\) requires \( \Omega(t/2^{O(p)}) \) space.

- **Proof:** Repeated elimination of passes implies:
  (\(p-1\))-pass alg. for \( p \)-level in \( O(s \, 2^{O(1)}) \) space
  .... 1-pass alg. for 2-level PO-Traversal in \( O(s \, 2^{O(p)}) \)
  But 2-level is INDEX and 1-pass alg. requires \( \Omega(t) \) space
Pass-Elimination Lemma
cf. Round-Elimination [Miltersen, Nisan, Safra, Wigderson ’98], [Sen ’03]

• Meta-Problem: Let $f$ be a function on $[n]^d$. For $x^i$ from $[n]^d$, 
  $P_{t,f}(x^1,\ldots,x^t,i)=f(x^i)$

• Pass-Elimination Lemma: If $t/s$ is large constant:
  If there’s a $p$-pass, $s$-space alg. $A$ for $P_{t,f}(w/p\ 1-\delta)$ then
  there’s a $(p-1)$-pass, $(2s \ lg \ \delta^{-1})$-space alg. $A'$ for $f$ $(w/p\ 1-\delta)$.

• Thm: A $p$-pass alg. for $(p+1)$-level, $t$-ary PO-Traversal $(w/p\ 2/3)$ requires $\Omega(t/2^{O(p)})$ space.

• Proof: Repeated elimination of passes implies:
  $(p-1)$-pass alg. for $p$-level in $O(s\ 2^{O(1)})$ space
  .... $1$-pass alg. for $2$-level PO-Traversal in $O(s\ 2^{O(p)})$
  But $2$-level is INDEX and $1$-pass alg. requires $\Omega(t)$ space
  Hence, $\Omega(t/2^{O(p)})$ space is required.
Pass-Elimination Proof
Pass-Elimination Proof

- **Pass-Elimination Lemma**: If $t/s$ is large constant:
  If there's a $p$-pass, $s$-space alg. $\mathcal{A}$ for $P_{t,f}(w/p 1-\delta)$ then there's a $(p-1)$-pass, $(2s \lg \delta^{-1})$-space alg. $\mathcal{A}'$ for $f$ (w/p $1-\delta$).
Pass-Elimination Proof

• **Pass-Elimination Lemma:** If $t/s$ is large constant:
  
  If there's a $p$-pass, $s$-space alg. $\mathcal{A}$ for $P_{t,f}(w/p \ 1-\delta)$ then there's a $(p-1)$-pass, $(2s \ lg \ \delta^{-1})$-space alg. $\mathcal{A}'$ for $f$ $(w/p \ 1-\delta)$.

• **Proof Idea:**
  
  To solve $f(x)$, emulate $\mathcal{A}$ on $(x^1, \ldots, x^{r-1}, x, x^{r+1}, \ldots, x^t, r)$ for some $r, x^i$
Pass-Elimination Proof

- **Pass-Elimination Lemma:** If $t/s$ is large constant:
  If there’s a $p$-pass, $s$-space alg. $A$ for $P_{t,f}(w/p \ 1-\delta)$ then there’s a $(p-1)$-pass, $(2s \ lg \ \delta^{-1})$-space alg. $A'$ for $f$ (w/p 1-\delta).

- **Proof Idea:**
  To solve $f(x)$, emulate $A$ on $(x^1, \ldots, x^{r-1}, x, x^{r+1}, \ldots, x^t, r)$ for some $r, x^i$
  Choose $x^{r+1}, \ldots, x^t$ such that state of $A$ at the end of the first pass is always the same.
Pass-Elimination Proof

- **Pass-Elimination Lemma:** If $t/s$ is a large constant:
  
  If there's a $p$-pass, $s$-space alg. $\mathcal{A}$ for $P_{t,f}(w/p, 1-\delta)$ then there's a $(p-1)$-pass, $(2s \log \delta^{-1})$-space alg. $\mathcal{A}'$ for $f$ (w/p $1-\delta$).

- **Proof Idea:**
  
  To solve $f(x)$, emulate $\mathcal{A}$ on $(x^1, \ldots, x^{r-1}, x, x^{r+1}, \ldots, x^t, r)$ for some $r, x^i$.
  
  Choose $x^{r+1}, \ldots, x^t$ such that state of $\mathcal{A}$ at the end of the first pass is always the same.

**Key Idea “Average Encoding Thm”:**

Consider deterministic alg. and random input (Yao’s Lemma)

Let $M^i$ be memory state after $(X^1, \ldots, X^i)$

Exists $r$ with $M^r$ and $M^t$ almost indept., i.e., $I(M^r, M^t)$ is small
Pass-Elimination Proof

- **Pass-Elimination Lemma:** If t/s is large constant:
  If there’s a p-pass, s-space alg. \( A \) for \( P_{t,f} \) (w/p \( 1-\delta \)) then there’s a \((p-1)\)-pass, \((2s \lg \delta^{-1})\)-space alg. \( A' \) for \( f \) (w/p \( 1-\delta \)).

- **Proof Idea:**
  To solve \( f(x) \), emulate \( A \) on \((x^1, ..., x^r, x, x^{r+1}, ..., x^t, r)\) for some \( r, x^i \)
  Choose \( x^{r+1}, ..., x^t \) such that state of \( A \) at the end of the first pass is always the same.

**Key Idea “Average Encoding Thm”:**
Consider deterministic alg. and random input (Yao’s Lemma)
Let \( M^i \) be memory state after \((X^1, ..., X^i)\)
Exists \( r \) with \( M^r \) and \( M^t \) almost indept., i.e., \( I(M^r, M^t) \) is small

\( A' \) pass 1: Emulate first two passes of \( A \).
Pass-Elimination Proof

- **Pass-Elimination Lemma**: If \( t/s \) is large constant:
  - If there’s a \( p \)-pass, \( s \)-space alg. \( A \) for \( P_{t,f} \) (w/p 1-\( \delta \)) then
    - there’s a \( (p-1) \)-pass, \((2s \lg \delta^{-1})\)-space alg. \( A' \) for \( f \) (w/p 1-\( \delta \)).

- **Proof Idea**:
  - To solve \( f(x) \), emulate \( A \) on \((x^1, \ldots, x^{r-1}, x, x^{r+1}, \ldots, x^t,r)\) for some \( r, x^i \)
  - Choose \( x^{r+1}, \ldots, x^t \) such that state of \( A \) at the end of the first pass is always the same.
  
  **Key Idea “Average Encoding Thm”**:
  - Consider deterministic alg. and random input (Yao’s Lemma)
    - Let \( M^i \) be memory state after \((X^1, \ldots, X^i)\)
    - Exists \( r \) with \( M^r \) and \( M^t \) almost indept., i.e., \( I(M^r, M^t) \) is small
  
  \( A' \) pass 1: Emulate first two passes of \( A \).
  \( A' \) pass \( i \): Emulate pass \( i-1 \) of \( A \).
Pass-Elimination Proof

- **Pass-Elimination Lemma:** If $t/s$ is large constant:

  If there's a $p$-pass, $s$-space alg. $A$ for $P(t,f)$ (w/p $1-\delta$) then there's a $(p-1)$-pass, $(2s \log_\delta - 1)$-space alg. $A'$ for $f$ (w/p $1-\delta$).

  **Proof Idea:** To solve $f(x)$, emulate $A$ on $(x_1, \ldots, x_{r-1}, x, x_{r+1}, \ldots, x_t)$ for some $r$, $x_i$.

  Choose $x_{r+1}, \ldots, x_t$ such that state of $A$ at the end of the first pass is always the same.

- **Key Idea “Average Encoding Thm”:**

  Consider deterministic alg. and random input (Yao’s Lemma)

  Let $M_i$ be memory state after $(X_1, \ldots, X_i)$

  Exists $r$ with $M_r$ and $M_t$ almost indept., i.e., $I(M_r, M_t)$ is small

  $A'$ pass 1: Emulate first two passes of $A$.

  $A'$ pass $i$: Emulate pass $i-1$ of $A$.

  Why is this different from round-elimination?

  Our approach adapts ideas directly for streams rather than proving lower bounds via communication.

  Gives *tighter* results and *avoids order-dependency issues*.

  E.g., consider a stream formed by concatenating two length $m$ binary strings $x$ and $y$. Given $p$ passes, is $x < y$?

  Round-elimination implies $\Omega(m^{1/(2p-1)})$ space

  Pass-elimination implies $\Omega(m^{1/p})$
Summary

• **Lemma (Pass-Elimination):** If t/s is large constant:
  
  If there’s a p-pass, s-space alg. $A$ for $P_{t,f}(w/p 1-\delta)$ then there’s a (p-1)-pass, $(2s \log \delta^{-1})$-space alg. $A'$ for $f$ (w/p 1-\delta).

• **Thm (Post-Order Traversal):** A p-pass algorithm for (p+1)-level, t-ary PO-Traversal (w/p 2/3) requires $\Omega(t/2^{O(p)})$ space.

∴ **Consequences:** Multi-pass space lower-bound for median finding, and ...
3: Applications

a) Summary of Applications

b) Longest Increasing Subsequences
Applications

- **Fixed-Dimensional Linear Programming:**
  Generalizes 3D LP requires $\Omega(n^{1/p})$  
  [Chan, Chen '05]

- **Histogram Learning and “Minimum Missing Element”:**
  Improves: $\Omega(n^{1/(2p-1)})$ becomes $\Omega(n^{1/p})$  
  [Chang, Kannan '06]

- **Selection and Quantile Estimation:**
  Simplifies (small loss of optimality)  
  [Guha, McGregor '07]

- **Longest Increasing Subsequence (LIS):**
  Extends to multiple passes  
  [Sun, Woodruff '07]
Increasing Subsequences

- **Find length of LIS:**
  
  1-pass $O(LIS)$ space  
  [Liben-Nowell et al. ’06]

  1-pass $(1+\varepsilon)$-approx. $O(\sqrt{m/\varepsilon})$ space  
  [Gopalan et al. ’06]

  $O(1)$-pass deterministic $(1+\varepsilon)$-approx. requires $\Omega(\sqrt{m})$  
  [Gal, Gopalan ’07], [Ergun, Jowhari ’08]
Increasing Subsequences

- **Find length of LIS:**
  - 1-pass $O(LIS)$ space [Liben-Nowell et al. '06]
  - 1-pass $(1+\epsilon)$-approx. $O(\sqrt{m/\epsilon})$ space [Gopalan et al. '06]
  - $O(1)$-pass deterministic $(1+\epsilon)$-approx. requires $\Omega(\sqrt{m})$ [Gal, Gopalan '07], [Ergun, Jowhari '08]

- **Find elements of LIS (given LIS$<k$):**
  - $p$-pass in $O(k^{1+1/(2^p-1)})$ space [Liben-Nowell et al. '06]
  - 1-pass in $\Omega(k^2)$ space [Sun, Woodruff '07]
Increasing Subsequences

- **Find length of LIS:**
  - 1-pass $O(LIS)$ space \([\text{Liben-Nowell et al. '06}]\)
  - 1-pass \((1+\varepsilon)\)-approx. $O(\sqrt{m/\varepsilon})$ space \([\text{Gopalan et al. '06}]\)
  - $O(1)$-pass deterministic \((1+\varepsilon)\)-approx. requires $\Omega(\sqrt{m})$ \([\text{Gal, Gopalan '07}, \text{Ergun, Jowhari '08}]\)

- **Find elements of LIS (given LIS}k):**
  - $p$-pass in $O(k^{1+1/(2^p-1)})$ space \([\text{Liben-Nowell et al. '06}]\)
  - 1-pass in $\Omega(k^2)$ space \([\text{Sun, Woodruff '07}]\)

- **Today's Result (Lower-Bounds):** Finding the elements of LIS in $p$ passes requires $\Omega(k^{1+1/(2^p-1)})$ space.
LIS Lower-Bound
LIS Lower-Bound

- **Single pass**: Reduction from $\text{INDEX}(x,j)$ [Sun, Woodruff '07]

Stream consist of $t+1$ blocks: $S_0, \ldots, S_{t-1}, T$

- $S_i = \{ 2(t-i)t+2j+x_j : j=\text{“a”}, \ldots, \text{“b”} \}$ of length
- $T = \{ 2(t-j)t+t+1, 2(t-j)t+t+2, \ldots, 2(t-j)t+t+j \}$
LIS Lower-Bound

- **Single pass:** Reduction from \( \text{INDEX}(x,j) \) [Sun, Woodruff ’07]

Stream consist of \( t+1 \) blocks: \( S_0, \ldots, S_{t-1}, T \)

\[
S_i = \{ 2(t-i)t+2j+x_j : j=“a”, \ldots, “b” \} \text{ of length } T = \{ 2(t-j)t+t+1, 2(t-j)t+t+2, \ldots, 2(t-j)t+t+j \}
\]

LIS length is \( t \) and is realized by \( S_j \cup T \)
LIS Lower-Bound

• **Single pass:** Reduction from $\text{INDEX}(x,j)$ [Sun, Woodruff ’07]
  Stream consist of $t+1$ blocks: $S_0, ..., S_{t-1}, T$
  
  $S_i = \{ 2(t-i)t+2j+x_j : j = \text{“}a\text{”}, ..., \text{“}b\text{”} \}$ of length
  $T = \{ 2(t-j)t+t+1, 2(t-j)t+t+2, ..., 2(t-j)t+t+j \}$

  LIS length is $t$ and is realized by $S_j \cup T$

• **$p$-pass:** Embed the above single pass instance into PO-traversal, set parameters carefully, and voila!
LIS Lower-Bound

- **Single pass:** Reduction from $\text{INDEX}(x,j)$ [Sun, Woodruff '07]
  Stream consist of $t+1$ blocks: $S_0, ..., S_{t-1}, T$
  
  \[ S_i = \{ 2(t-i)t+2j+x_j : j = \text{"a"}, \ldots, \text{"b"} \} \text{ of length} \]
  \[ T = \{ 2(t-j)t+t+1, 2(t-j)t+t+2, \ldots, 2(t-j)t+t+j \} \]

  LIS length is $t$ and is realized by $S_j \cup T$

- **p-pass:** Embed the above single pass instance into PO-traversal, set parameters carefully, and voila!
Thanks.

**Highlights**

- First general random-order lower-bound and it’s tight!
- Exponential separation between passes for selection in random-order and adversarial-order.
- First doubly-exponential pass/space trade-off.
- ... many open problems.

Thanks.
A Bit More Detail
A Bit More Detail

- Suffices to consider a deterministic alg. $A$ on $[X^1, \ldots, X^t, i]$ where each $X^i \sim D$ and $i \sim \text{Uni}\{1, \ldots, t\}$ for arbitrary $D$. 
A Bit More Detail

- Suffices to consider a deterministic alg. $\mathcal{A}$ on $[X^1, \ldots, X^t, i]$ where each $X^i \sim D$ and $i \sim \text{Uni}\{1, \ldots, t\}$ for arbitrary $D$. 

Stream: $X^1 \mid X^2 \mid X^3 \mid X^4 \mid X^5 \mid X^6 \mid i$
A Bit More Detail

- Suffices to consider a deterministic alg. $\mathcal{A}$ on $[X^1, \ldots, X^t, i]$ where each $X^i \sim D$ and $i \sim \text{Uni}\{1, \ldots, t\}$ for arbitrary $D$. 

Memory State: $M^1 \quad M^2 \quad M^3 \quad M^4 \quad M^5 \quad M^6$

Stream: $X^1 \quad X^2 \quad X^3 \quad X^4 \quad X^5 \quad X^6 \quad i$
A Bit More Detail

- Suffices to consider a deterministic alg. $\mathcal{A}$ on $[X^1, \ldots X^t,i]$ where each $X^i \sim D$ and $i \sim \text{Uni}\{1, \ldots, t\}$ for arbitrary $D$.

- **Lemma:** Exists $r$ and $x^1, \ldots, x^{r-1}$ s.t. if $X^1 = x^1, \ldots, X^{r-1} = x^{r-1}$, $i = r$: $M^r$ and $M^t$ are “almost independent”
  - $\mathcal{A}$ solves $f(X^r)$ w/p $1 - 9\delta$

\[
\begin{array}{cccccccc}
\text{Stream:} & X^1 & X^2 & X^3 & X^4 & X^5 & X^6 & i \\
\text{Memory State:} & M^1 & M^2 & M^3 & M^4 & M^5 & M^6 &
\end{array}
\]
A Bit More Detail

• Suffices to consider a deterministic alg. $\mathcal{A}$ on $[X^i, \ldots, X^t, i]$ where each $X^i \sim D$ and $i \sim \text{Uni}\{1, \ldots, t\}$ for arbitrary $D$.

• **Lemma:** Exists $r$ and $x^1, \ldots, x^{r-1}$ s.t. if $X^1 = x^1, \ldots, X^{r-1} = x^{r-1}, i = r$: $M^r$ and $M^t$ are “almost independent”

  $\mathcal{A}$ solves $f(X^r)$ w/p $1 - 9\delta$

---

**Memory State:**

\[ M^1 \quad M^2 \quad M^3 \quad M^4 \quad M^5 \quad M^6 \]

**Stream:**

\[ x^1 \quad x^2 \quad X^3 \quad X^4 \quad X^5 \quad X^6 \quad i = r \]
A Bit More Detail

- Suffices to consider a deterministic alg. $A$ on $[X^1, \ldots X^t,i]$ where each $X^i \sim D$ and $i \sim \text{Uni}\{1, \ldots, t\}$ for arbitrary $D$.

- **Lemma**: Exists $r$ and $x^1, \ldots, x^{r-1}$ s.t. if $X^1 = x^1, \ldots, X^{r-1} = x^{r-1}, i = r$: $M^r$ and $M^t$ are “almost independent”
  $A$ solves $f(X^r)$ w/p 1-9$\delta$

Memory State: 

```
M1  M2  M3  M4  M5  M6
```

Stream: 

```
x1  x2  x3  x4  x5  x6
```

almost indep. $i = r$
A Bit More Detail

• Suffices to consider a deterministic alg. \( \mathcal{A} \) on \([X^1, \ldots, X^t, i]\)
  where each \( X^i \sim D \) and \( i \sim \text{Uni}\{1, \ldots, t\} \) for arbitrary \( D \).

• **Lemma:**Exists \( r \) and \( x^1, \ldots, x^{r-1} \) s.t. if \( X^i = x^i, \ldots, X^{r-1} = x^{r-1}, i = r \):
  \( M^r \) and \( M^t \) are “almost independent”
  \( \mathcal{A} \) solves \( f(X^r) \) w/p \( 1 - 9\delta \)

• **Lemma:** \( L_1(M^r, M^r \text{ given } M^t = v) = O(\sqrt{s/t}) \) w/p \( \frac{2}{3} \) if \( v \sim M^t \).
A Bit More Detail

• Suffices to consider a deterministic alg. $\mathcal{A}$ on $[X^i, ... X^t, i]$ where each $X^i \sim D$ and $i \sim \text{Uni}\{1, ..., t\}$ for arbitrary $D$.

• **Lemma:** Exists $r$ and $x^1, ..., x^{r-1}$ s.t. if $X^i = x^i$, ..., $X^{r-1} = x^{r-1}, i = r$: $M^r$ and $M^t$ are “almost independent”
  $\mathcal{A}$ solves $f(X^r)$ w/p $1 - 9\delta$

• **Lemma:** $L_1(M^r, M^r \text{ given } M^t = v) = O(\sqrt{s/t})$ w/p $2/3$ if $v \sim M^t$.

• **Lemma:** Exists $v, x^{r+1}(.), ..., x^t(.)$ s.t., if $M^r = u$ then
  $M^t = v$ on stream $[x^i, ..., x^{r-1}, X^r, x^{r+1}(u), ..., x^m(u)]$
  $\mathcal{A}$ solves $f(X^r)$ w/p $9/10$ on $[x^i, ..., x^{r-1}, X^r, x^{r+1}(u), ..., x^t(u), r]$

Memory State: $M^1 \rightarrow M^2 \rightarrow M^3 \rightarrow M^4 \rightarrow M^5 \rightarrow M^6$
Stream: $x^1, x^2, ..., x^6$
A Bit More Detail

- Suffices to consider a deterministic alg. $\mathcal{A}$ on $[X^1, \ldots, X^t, i]$ where each $X^i \sim D$ and $i \sim \text{Uni}\{1, \ldots, t\}$ for arbitrary $D$.

- **Lemma:** Exists $r$ and $x^1, \ldots, x^{r-1}$ s.t. if $X^i = x^i, \ldots, X^{r-1} = x^{r-1}, i = r$:
  - $M^r$ and $M^t$ are "almost independent"
  - $\mathcal{A}$ solves $f(X^r)$ w/p $1 - 9\delta$

- **Lemma:** $L_1(M^r, M^r \mid M^t = v) = O(\sqrt{s/t})$ w/p $2/3$ if $v \sim M^t$.

- **Lemma:** Exists $v, x^{r+1}(\cdot), \ldots, x^t(\cdot)$ s.t., if $M^r = u$ then $M^t = v$ on stream $[x^1, \ldots, x^{r-1}, X^r, x^{r+1}(u), \ldots, x^m(u)]$
  - $\mathcal{A}$ solves $f(X^r)$ w/p $9/10$ on $[x^1, \ldots, x^{r-1}, X^r, x^{r+1}(u), \ldots, x^t(u), r]$

---

**Memory State:**

- $M^1$
- $M^2$
- $M^3 = u$
- $M^4$
- $M^5$
- $M^6 = v$

---

**Stream:**

- $x^1$
- $x^2$
- $x^3$
- $x^4(u)$
- $x^5(u)$
- $x^6(u)$
- $i = r$
A Bit More Detail

- Suffices to consider a deterministic alg. $A$ on $[X_1, \ldots, X_t, i]$ where each $X_i \sim D$ and $i \in \{1, \ldots, t\}$.

- **Lemma:** Exists $r$ and $x^l, \ldots, x^{r-1}$ s.t. if $X^l = x^l, \ldots, X^{r-1} = x^{r-1}$, $i = r$:
  - $M^r$ and $M^t$ are “almost independent”
  - $A$ solves $f(X^r)$ w/p $1-9\delta$

- **Lemma:** $L_1(M^r, M^r \text{ given } M^t=v) = O(\sqrt{s/t})$ w/p $2/3$ if $v \sim M^t$.

- **Lemma:** Exists $v, x^{r+1}, \ldots, x^t$ s.t., if $M^r = u$ then
  - $M^t = v$ on stream $[x^l, \ldots, x^{r-1}, X^r, x^{r+1}(u), \ldots, x^m(u)]$
  - $A$ solves $f(X^r)$ w/p $9/10$ on $[x^l, \ldots, x^{r-1}, X^r, x^{r+1}(u), \ldots, x^t(u), r]$

**Memory State:**  
$M^1, M^2, M^3 = u, M^4, M^5, M^6 = v$

**Stream:**  
$x^l, x^2, \ldots, x^3, x^4(u), x^5(u), x^6(u), i = r$

---

- a) Chain Rule of Mutual Information
- b) Expect to Satisfy Both Conditions
- c) Markov’s Inequality, Union Bound
A Bit More Detail

• Suffices to consider a deterministic algorithm \( A \) on \( [X_1, ..., X_t, i] \) where each \( X_i \sim D \) and

  \[ i \in \{1, ..., t\} \]

• **Lemma:** Exists \( r \) and \( x_1, ..., x_{r-1} \) s.t. if \( X_1 = x_1, ..., X_{r-1} = x_{r-1}, i = r \):
  \( A \) solves \( f(X_r) \) w/p \( 1 - \delta \)

• **Lemma:** Exists \( v \), \( x_{r+1} \), ..., \( x_t \) s.t., if \( M_r = u \) then
  \( M_t = v \) on stream \( [x^1, ..., x^{r-1}, X_r, x^{r+1}(u), ..., x^t(u)] \)
  \( A \) solves \( f(X_r) \) w/p \( 9/10 \) on \( [x^1, ..., x^{r-1}, X_r, x^{r+1}(u), ..., x^t(u), r] \)

**AverageEncodingTheorem**

  a) Mutual Information = \( E[KL-\text{divergence}] \)
  b) Relate KL-divergence to \( L_1 \)-distance

| Memory State: | \( M^1 \) | \( M^2 \) | \( M^3 = u \) | \( M^4 \) | \( M^5 \) | \( M^6 = v \) |
| Stream: | \( x^1 \) | \( x^2 \) | \( x^3 \) | \( x^4(u) \) | \( x^5(u) \) | \( x^6(u) \) |
A Bit More Detail

- Suffices to consider a deterministic alg. $\mathcal{A}$ on $[X^i, \ldots X^t, i]$ where each $X^i \sim D$ and $i \sim \text{Uni}\{1, \ldots, t\}$ for arbitrary $D$.

- **Lemma:** Exists $r$ and $x^1, \ldots, x^{r-1}$ s.t. if $X^1 = x^1, \ldots, X^{r-1} = x^{r-1}, i=r$: $M^r$ and $M^t$ are “almost independent.”
  - $\mathcal{A}$ solves $f(X^r)$ w/p $1-\delta$.

- **Lemma:** $L_1(M^r, M^r \text{ given } M^t = v) = O(\sqrt{s/t})$ w/p $2/3$ if $v \neq M^t$.

- **Lemma:** Exists $v, x^{r+1}(.), \ldots, x^t(.)$ s.t., if $M^r = u$ then $M^t = v$ on stream $[x^1, \ldots, x^{r-1}, X^r, x^{r+1}(u), \ldots, x^m(u)]$
  - $\mathcal{A}$ solves $f(X^r)$ w/p $9/10$ on $[x^1, \ldots, x^{r-1}, X^r, x^{r+1}(u), \ldots, x^t(u), r]$

---

**Memory State:**

- $M^1$
- $M^2$
- $M^3 = u$
- $M^4$
- $M^5$
- $M^6 = v$

**Stream:**

- $x^1$
- $x^2$
- $X^3$
- $x^4(u)$
- $x^5(u)$
- $x^6(u)$

Almost indep.