A Tight Lower Bound for Dynamic Membership in the External Memory Model

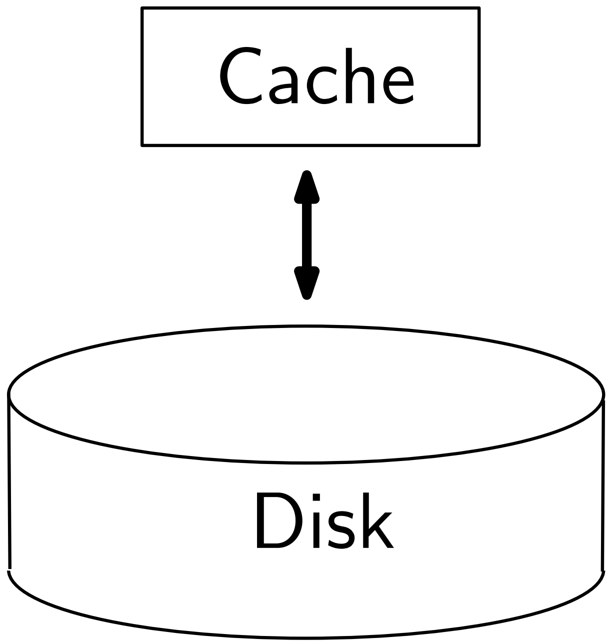
Elad Verbin
ITCS, Tsinghua University

Qin Zhang
Hong Kong University of Science & Technology

April 2010

The computational model

- External memory (EM) model (or I/O model) (Aggarwal and Vitter 1988):



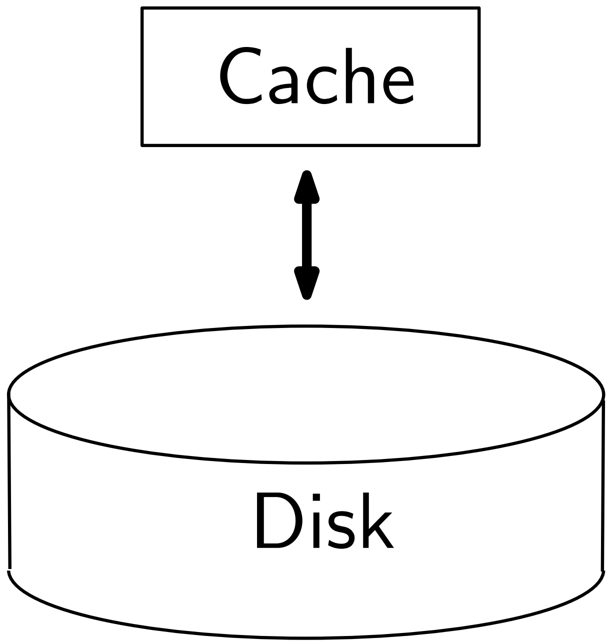
Cache of size $M = \Omega(B)$ (M, B in words)

Each I/O reads/writes a cell

Disk partitioned into cells of size B

The computational model

- External memory (EM) model (or I/O model) (Aggarwal and Vitter 1988):



Cache of size $M = \Omega(B)$ (M, B in words)

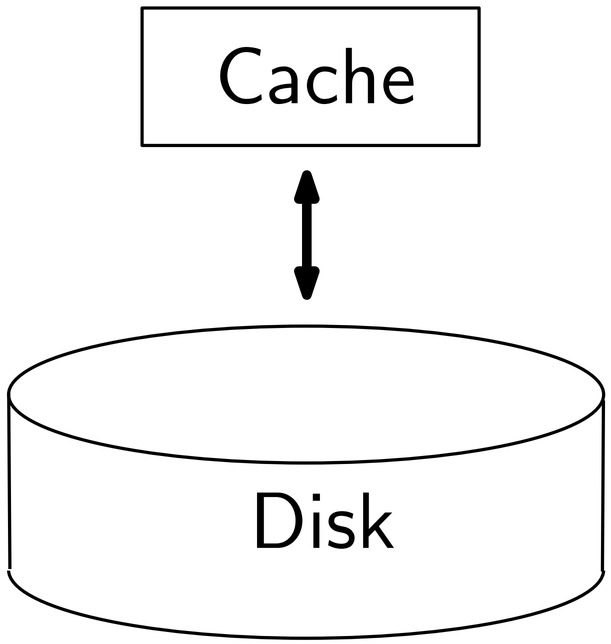
Each I/O reads/writes a cell

Disk partitioned into cells of size B

Cost of an operation: # of cells read/changed; accessing the cache is free of charge.

The computational model

- External memory (EM) model (or I/O model) (Aggarwal and Vitter 1988):



Cache of size $M = \Omega(B)$ (M, B in words)

Each I/O reads/writes a cell

Disk partitioned into cells of size B

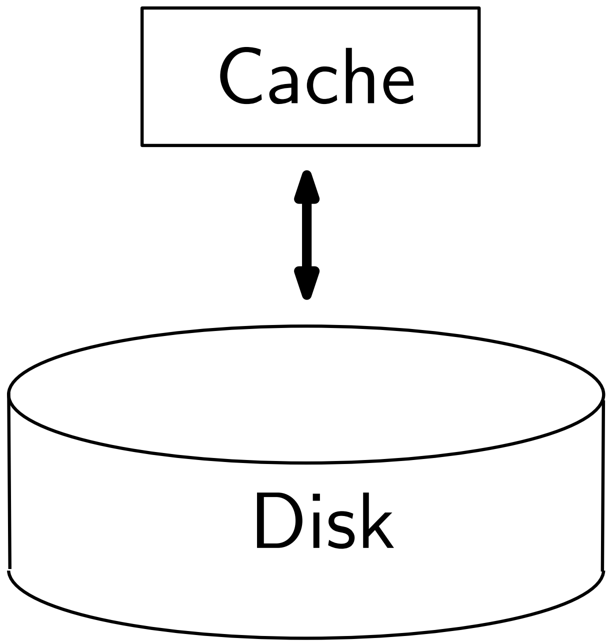
Cost of an operation: # of cells read/changed;

accessing the cache is free of charge.

- Motivated by the real-world applications:
  accessing the cache is much faster than accessing higher level memory hierarchies.

The computational model

- External memory (EM) model (or I/O model) (Aggarwal and Vitter 1988):



Cache of size $M = \Omega(B)$ (M, B in words)

Each I/O reads/writes a cell

Disk partitioned into cells of size B

Cost of an operation: # of cells read/changed; accessing the cache is free of charge.

- Similar to Yao's *cell probe* model, only that in the EM model (1) cell size is large; (2) has an explicit cache.

The power of buffering

- For numerous dynamic data structure problems in external memory, updates can be buffered.
  - Buffer tree [Arge 1995]
  - Logarithmic method [Bentley 1980] + B-tree

The power of buffering

- For numerous dynamic data structure problems in external memory, updates can be buffered.
  - Buffer tree [Arge 1995]
  - Logarithmic method [Bentley 1980] + B-tree

problem	update	query	cache-oblivious
stack	$O(1/B)$	/	trivial
queue	$O(1/B)$	/	trivial
priority-queue	$O(\frac{1}{B}\log_B n)$	/	[Arge et. al. STOC 02]
predecessor	$O(\frac{1}{B}\log n)$	$O(\log n)$	trivial
range-sum range-reporting . . .	$O(\frac{B^\epsilon}{B}\log n)$	$O(\log_B n)$	[Brodal et. al. SODA 10]

B: size of a block/cell (in words)

How about Dictionary and Membership?

Membership: Maintain a set $S \subseteq U$ with $|S| \leq n$.
Given an $x \in U$, is $x \in S$? Yes or No.

Dictionary: If $x \in S$, return associated info, otherwise say No. Often assumes "indivisibility".

Objective: Tradeoff between update cost t_u and query cost t_q

Two of the most fundamental data structure problems in computer science!

How about Dictionary and Membership (Cont.)?

- Dictionary and membership (selected)
  - Knuth, 1973: External hashing
    Expected average cost of an operation is $1 + 1/2^{\Omega(b)}$, provided the load factor α is less than a constant smaller than 1. (truly random hash function)
  - Data structures like Arge's Buffer tree:
    Update $= O(\frac{b^{\epsilon}}{b} \log n)$, Query $= O(\log_b n)$.

How about Dictionary and Membership (Cont.)?

- Dictionary and membership (selected)
  - Knuth, 1973: External hashing
    Expected average cost of an operation is $1 + 1/2^{\Omega(b)}$, provided the load factor α is less than a constant smaller than 1. (truly random hash function)
  - Data structures like Arge's Buffer tree:
    Update $= O(\frac{b^{\epsilon}}{b} \log n)$, Query $= O(\log_b n)$.
- **Question**: can we improve the amortized update cost to $o(1)$ in external memory, without sacrificing the query speed by much?

The conjecture

A long-time folklore conjecture inEM community:
(explicitly stated by Jensen and Pagh, 2007)

t_u must be $\Omega(1)$ if t_q is required to be $O(1)$

t_u: expected amortized update cost

t_q: expected average query cost

The conjecture

A long-time folklore conjecture inEM community:

(explicitly stated by Jensen and Pagh, 2007)

t_u must be $\Omega(1)$ if t_q is required to be $O(1)$

Our result: ≥ 0.99 $\leq 10^{-5} \cdot \log_{b \log n} n$

t_u: expected amortized update cost

t_q: expected average query cost

The conjecture

A long-time folklore conjecture inEM community:
(explicitly stated by Jensen and Pagh, 2007)

t_u must be $\Omega(1)$ if t_q is required to be $O(1)$

Our result: ≥ 0.99 $\leq 10^{-5} \cdot \log_{b \log n} n$

t_u: expected amortized update cost
t_q: expected average query cost

holds even when

- $u = O(n)$,
- no deletion
- randomization

Consequences

- A strong dichotomy result:

  when designing an external memory data structure for dynamic membership,

  - either use external hash ($t_u = t_q = 1 + o(1)$)
  - or use buffer tree ($t_u = o(1), t_q = O(\log_b n)$)

Consequences

- A strong dichotomy result:

  when designing an external memory data structure for dynamic membership,

  - either use external hash ($t_u = t_q = 1 + o(1)$)
  - or use buffer tree ($t_u = o(1), t_q = O(\log_b n)$)

- Striking implications:

  the query complexities of many problems such as

  1D-range reporting, predecessor, partial-sum, etc.,

  are all the same in the regime where the update time is less than 1.

Compared with the rich results in RAM!

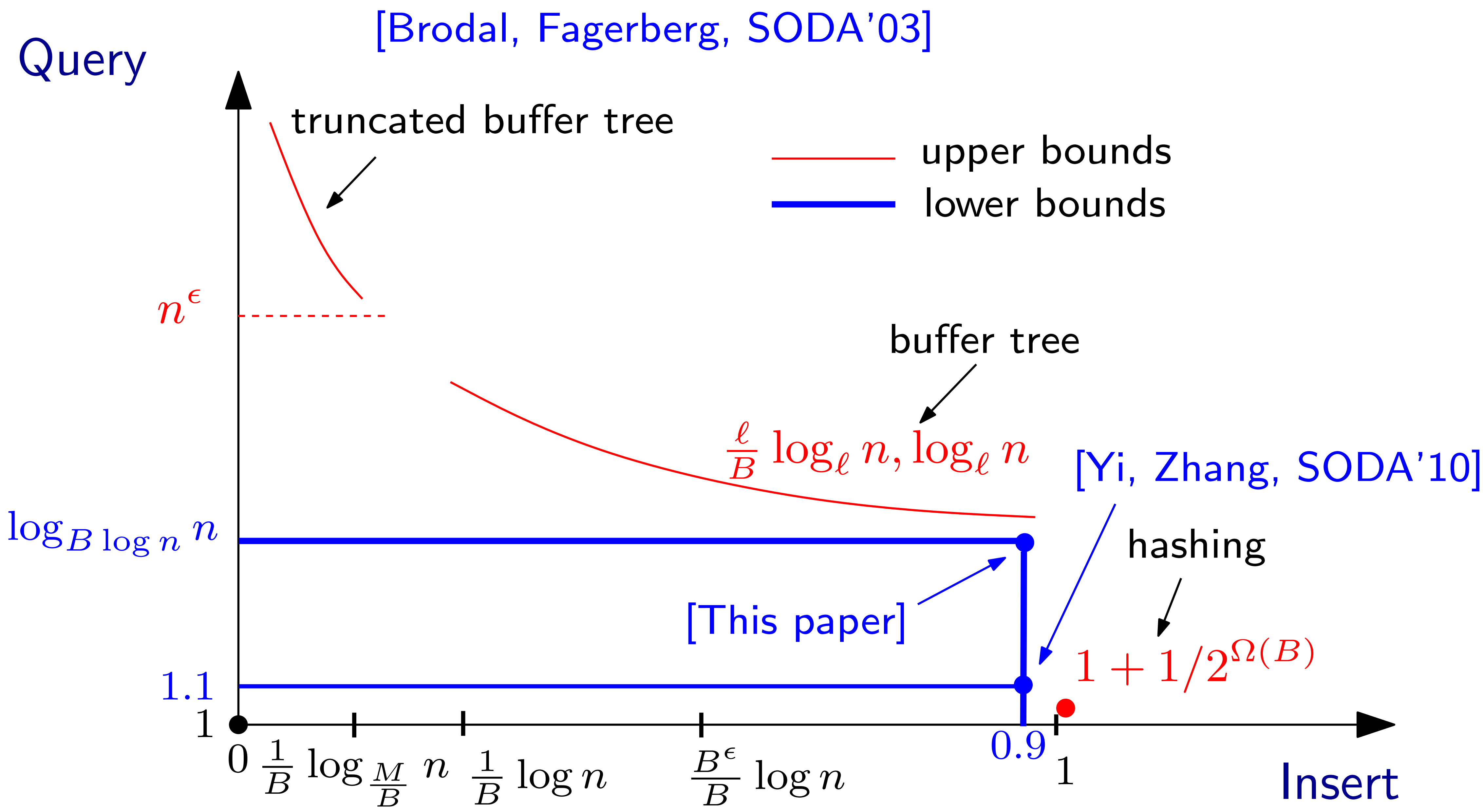
- 1D-range reporting
  - $O(\sqrt{\log N / \log\log N})$ insertion and query [Andersson, Thorup, JACM'07]
  - $O(\log N / \log\log N)$ insertion and $O(\log\log N)$ query [Mortensen, Pagh, Pătrașcu, STOC'05]
  - Other results that depend on the word size w
- Predecessor
  - $\Theta(\sqrt{\log N / \log\log N})$ insertion and query [Andersson, Thorup, JACM'07]
- Partial-sum
  - $\Theta(\log N)$ insertion query [Pătrașcu, Demaine, SODA'04]

Two useful messages

1. **Buffering is impossible to achieve in the EM model with sublogarithmic query time.**

2. **EM model is a “clearner” model than RAM in certain perspectives**.

The landscape of the membership problem



Now, technical details ...

Outline

- Preliminaries

Outline

- Preliminaries
- Deterministic algorithm + random update sequence

Outline

- Preliminaries
- Deterministic algorithm + random update sequence

Can be extended to:

- randomized algorithm

Outline

- Preliminaries
- Deterministic algorithm + random update sequence

  Can be extended to:

  - randomized algorithm
- Future work

Preliminaries

- $U = \{0, 1, \ldots, u-1\}$: universe. $|U| = u$.

- m: size of cache. In bits.

  b: size of one cell. In bits.

  n: total number of inserted elements.

- S: set of elements we are maintaining. $|S| \leq n$

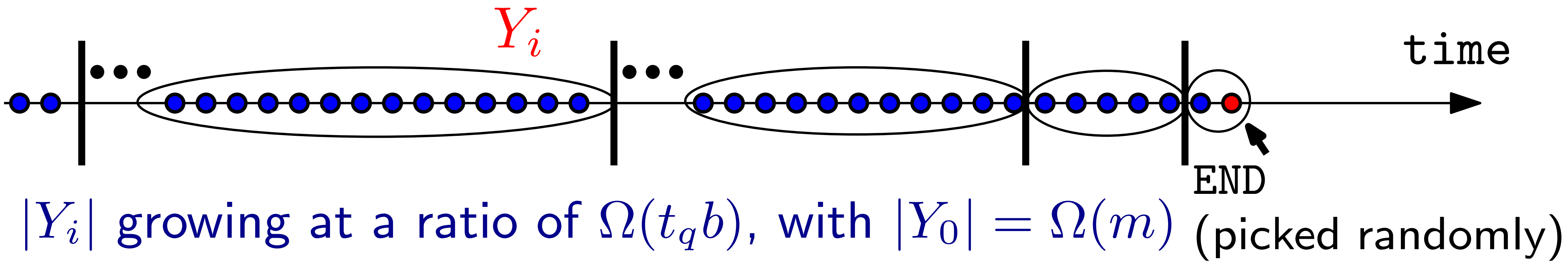
- $q_{\texttt{snapshot}}(x)$: query path of x at time snapshot `state`

Preliminaries

- $U = \{0, 1, \ldots, u-1\}$: universe. $|U| = u$.
- m: size of cache. In bits.
  b: size of one cell. In bits.
  n: total number of inserted elements.
- S: set of elements we are maintaining. $|S| \leq n$
- $q_{\texttt{snapshot}}(x)$: query path of x at time snapshot `state`
- A very mild assumption
  - $u \geq c \cdot n$ for sufficiently large c

Framework of the proof

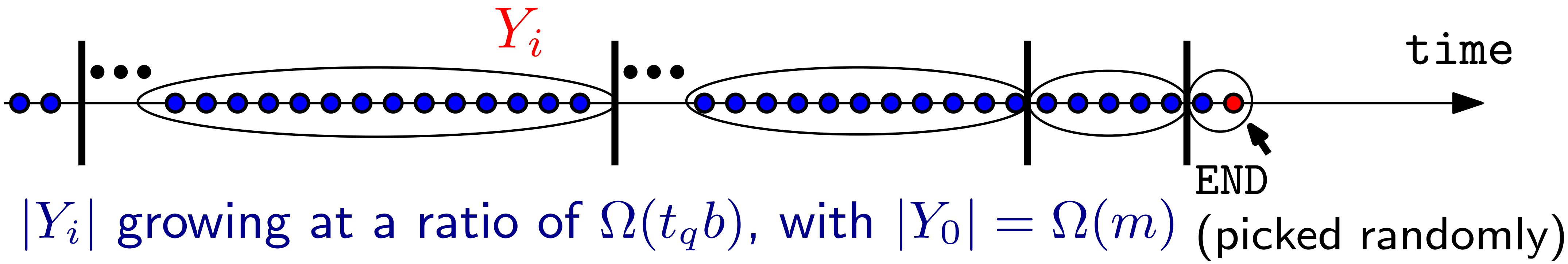
Exponentially growing epoches



$|Y_i|$ growing at a ratio of $\Omega(t_q b)$, with $|Y_0| = \Omega(m)$

Framework of the proof

Exponentially growing epoches

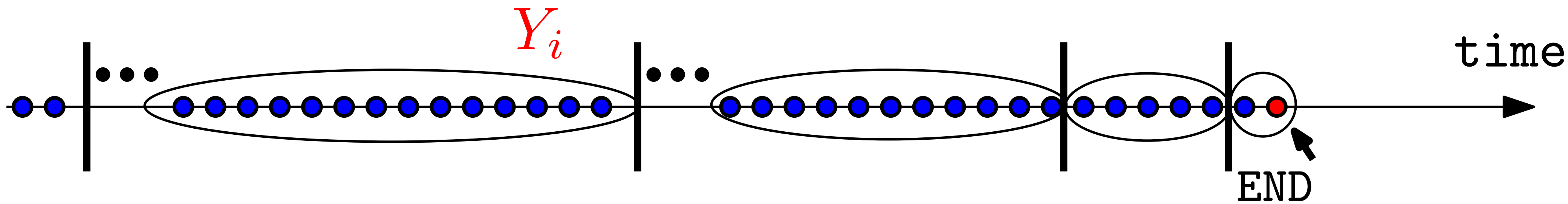


$|Y_i|$ growing at a ratio of $\Omega(t_q b)$, with $|Y_0| = \Omega(m)$

- C_i^u: set of cells probed by inserting Y_i.
- C_i^q: set of cells probed when querying all elements in Y_i at END. (cheat a bit here.)
- $C_i^* = C_i^u \cup C_i^q$, $C_{<i}^* = \bigcup_{j<i} C_j^*$.

Framework of the proof

Exponentially growing epoches



$|Y_i|$ growing at a ratio of $\Omega(t_q b)$, with $|Y_0| = \Omega(m)$ (picked randomly)

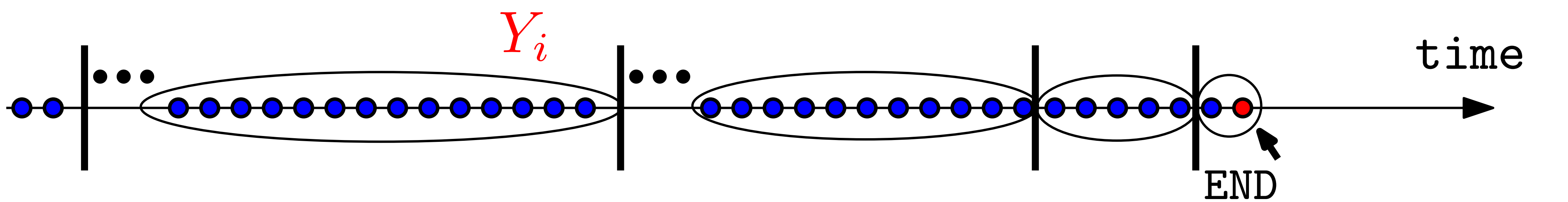
- C_i^u: set of cells probed by inserting Y_i.
- C_i^q: set of cells probed when querying all elements in Y_i at END. (cheat a bit here.)
- $C_i^* = C_i^u \cup C_i^q$, $C_{<i}^* = \bigcup_{j<i} C_j^*$.

Key Lemma

Suppose that $t_u \leq 0.9$ and pick $x \in U$ uniformly at random.
$\mathcal{E}_i$ is an indicator random variable $= 1$ if $q_{\texttt{END}}(x)$ intersects $C_i^* \backslash C_{<i}^*$.
Then $\mathbf{E}[\mathcal{E}_i] \geq \Omega(1)$ for all $i = 1, 2, \ldots, d$.

Framework of the proof

Exponentially growing epoches



$|Y_i|$ growing at a ratio of $\Omega(t_q b)$, with $|Y_0| = \Omega(m)$

- C_i^u: set of cells probed by inserting Y_i.
- C_i^q: set of cells probed when querying all elements in Y_i at END. (cheat a bit here.)
- $C_i^* = C_i^u \cup C_i^q$, $C_{<i}^* = \bigcup_{j<i} C_j^*$.

Key Lemma $\Rightarrow t_q \geq \Omega(\log_{b \log n}(n/m))$

Suppose that $t_u \leq 0.9$ and pick $x \in U$ uniformly at random.
$\mathcal{E}_i$ is an indicator random variable $= 1$ if $q_{\texttt{END}}(x)$ intersects $C_i^* \backslash C_{<i}^*$.
Then $\mathbf{E}[\mathcal{E}_i] \geq \Omega(1)$ for all $i = 1, 2, \ldots, d$.

Proof for the key lemma

Key Lemma

Suppose that $t_u \leq 0.9$ and pick $x \in U$ uniformly at random.
$\mathcal{E}_i$ is an indicator random variable $= 1$ if $q_{\mathtt{END}}(x)$ intersects $C_i^* \backslash C_{<i}^*$.
Then $\mathbf{E}[\mathcal{E}_i] \geq \Omega(1)$ for all $i = 1, 2, \ldots, d$.

Proof for the key lemma

Conceptually, we put $C^*_{<i}$ into the cache. Let $m_i = m + |C^*_{<i}|$

Key Lemma

Suppose that $t_u \leq 0.9$ and pick $x \in U$ uniformly at random.
$\mathcal{E}_i$ is an indicator random variable $= 1$ if $q_{\mathtt{END}}(x)$ intersects ~~$C^*_i \setminus C^*_{<i}$~~ C^*_i.
Then $\mathbf{E}[\mathcal{E}_i] \geq \Omega(1)$ for all $i = 1, 2, \ldots, d$.

Proof for the key lemma

Conceptually, we put $C^*_{<i}$ into the cache. Let $m_i = m + |C^*_{<i}|$

Key Lemma

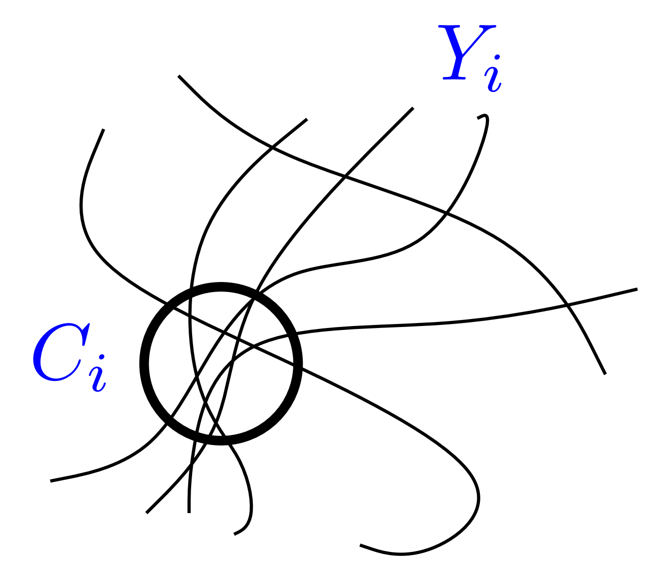
Suppose that $t_u \le 0.9$ and pick $x \in U$ uniformly at random.
$\mathcal{E}_i$ is an indicator random variable $= 1$ if $q_{\mathtt{END}}(x)$ intersects ~~$C^*_i \setminus C^*_{<i}$~~ C^*_i.
Then $\mathbf{E}[\mathcal{E}_i] \ge \Omega(1)$ for all $i = 1, 2, \ldots, d$.

Proof. We prove for each epoch i.

Step 1: **(the encoding lemma)**

If $m_i \le \gamma |Y_i|$ (γ: small const), then with probability at least 0.9 over the choice of Y_i, it holds that

$|\{y \in Y_i \mid q_{\mathtt{END}}(y) \cap C^u_i \ne \emptyset\}| \ge 0.99\,|Y_i|$.



Proof for the key lemma

Conceptually, we put $C^*_{<i}$ into the cache. Let $m_i = m + |C^*_{<i}|$

Key Lemma

Suppose that $t_u \le 0.9$ and pick $x \in U$ uniformly at random.
$\mathcal{E}_i$ is an indicator random variable $= 1$ if $q_{\texttt{END}}(x)$ intersects ~~$C_i^* \setminus C^*_{<i}$~~ C_i^*.
Then $\mathbf{E}[\mathcal{E}_i] \ge \Omega(1)$ for all $i = 1, 2, \ldots, d$.

Proof. We prove for each epoch i.

Step 1: **(the encoding lemma)**

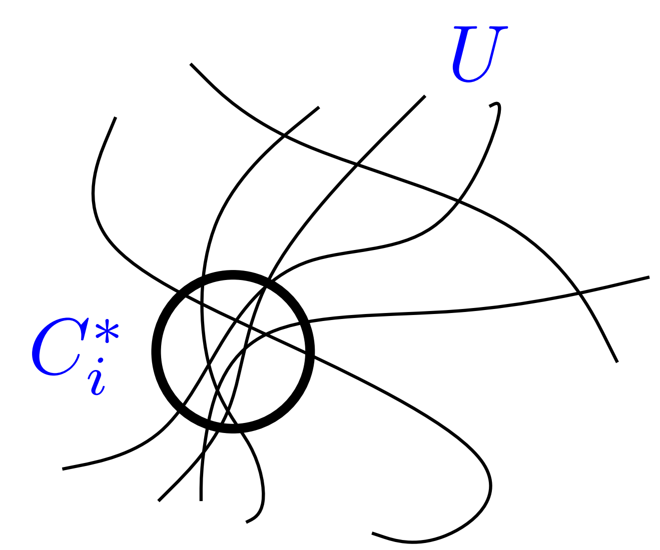
If $m_i \le \gamma |Y_i|$ (γ: small const), then with probability at least 0.9 over the choice of Y_i, it holds that

$$|\{y \in Y_i \mid q_{\texttt{END}}(y) \cap C_i^u \ne \emptyset\}| \ge 0.99\,|Y_i|.$$

Step 2: **(the "snake" lemma)**

If $|C_i^u| \le 10/11 \cdot |Y_i|$ and $m_i \le \gamma |Y_i|$, then

$$\mathbf{E}[|\{x \in U \mid q_{\texttt{END}}(x) \cap C_i^* \ne \emptyset\}|] \ge \Omega(u).$$



Proof for the key lemma

Conceptually, we put $C_{<i}^*$ into the cache. Let $m_i = m + |C_{<i}^*|$

Key Lemma

Suppose that $t_u \le 0.9$ and pick $x \in U$ uniformly at random.
$\mathcal{E}_i$ is an indicator random variable $= 1$ if $q_{\mathtt{END}}(x)$ intersects ~~$C_i^* \setminus C_{<i}^*$~~ C_i^*.
Then $\mathbf{E}[\mathcal{E}_i] \ge \Omega(1)$ for all $i = 1, 2, \ldots, d$.

Proof. We prove for each epoch i.

Step 1: **(the encoding lemma)**

If $m_i \le \gamma\,|Y_i|$ (γ: small const), then with probability at least 0.9 over the choice of Y_i, it holds that

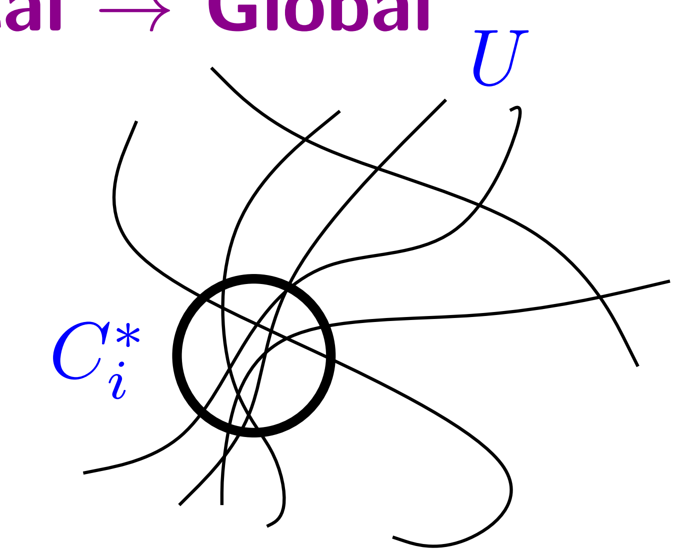
$$|\{y \in Y_i \mid q_{\mathtt{END}}(y) \cap C_i^u \ne \emptyset\}| \ge 0.99\,|Y_i|.$$

Step 2: **(the "snake" lemma)**

If $|C_i^u| \le 10/11 \cdot |Y_i|$ and $m_i \le \gamma\,|Y_i|$, then

$$\mathbf{E}[|\{x \in U \mid q_{\mathtt{END}}(x) \cap C_i^* \ne \emptyset\}|] \ge \Omega(u).$$

Local $\rightarrow$ Global



Idea of the proof for the encoding lemma

The encoding lemma (subscripts i are omitted)

If $m \leq \gamma |Y|$ (γ: a small const), then with probability at least 0.9 over the choices of Y, it holds that

$|\{y \in Y \mid q_{\mathtt{END}}(y) \cap C^u \neq \emptyset\}| \geq 0.99 |Y|$.

Proof.

Z: the set of $y \in Y$ such that $q(y)$ intersect C^u at END.

Idea of the proof for the encoding lemma

The encoding lemma (subscripts i are omitted)

If $m \leq \gamma |Y|$ (γ: a small const), then with probability at least 0.9 over the choices of Y, it holds that

$|\{y \in Y \mid q_{\mathtt{END}}(y) \cap C^u \neq \emptyset\}| \geq 0.99 |Y|$.

Proof.

Z: the set of $y \in Y$ such that $q(y)$ intersect C^u at END.

Idea: If $|Z| < 0.99|Y|$, then there is an encoding of Y of small size, contradicting to the fact that Y is chosen from U randomly (thus has a large entropy $\approx \log \binom{u}{|Y|}$).

Idea of the proof for the encoding lemma

The encoding lemma (subscripts i are omitted)

If $m \leq \gamma\,|Y|$ (γ: a small const), then with probability at least 0.9 over the choices of Y, it holds that

$|\{y \in Y \mid q_{\mathtt{END}}(y) \cap C^u \neq \emptyset\}| \geq 0.99\,|Y|$.

Proof.

Z: the set of $y \in Y$ such that $q(y)$ intersect C^u at END.

Idea: If $|Z| < 0.99|Y|$, then there is an encoding of Y of small size, contradicting to the fact that Y is chosen from U randomly (thus has a large entropy $\approx \log\binom{u}{|Y|}$).

The encoding:

- The state of the cache, m bits.
- The elements Z, $\log\binom{u}{|Z|}$ bits.

Idea of the proof for the encoding lemma

The encoding lemma (subscripts i are omitted)

If $m \leq \gamma\,|Y|$ (γ: a small const), then with probability at least 0.9 over the choices of Y, it holds that

$$|\{y \in Y \mid q_{\mathtt{END}}(y) \cap C^u \neq \emptyset\}| \geq 0.99\,|Y|.$$

Proof.

Z: the set of $y \in Y$ such that $q(y)$ intersect C^u at END.

Idea: If $|Z| < 0.99|Y|$, then there is an encoding of Y of small size, contradicting to the fact that Y is chosen from U randomly (thus has a large entropy $\approx \log\binom{u}{|Y|}$).

The encoding:

- The state of the cache, m bits.
- The elements Z, $\log\binom{u}{|Z|}$ bits.

But,
$m + \log\binom{u}{|Z|} < \log\binom{u}{|Y|}$
A contradiction.

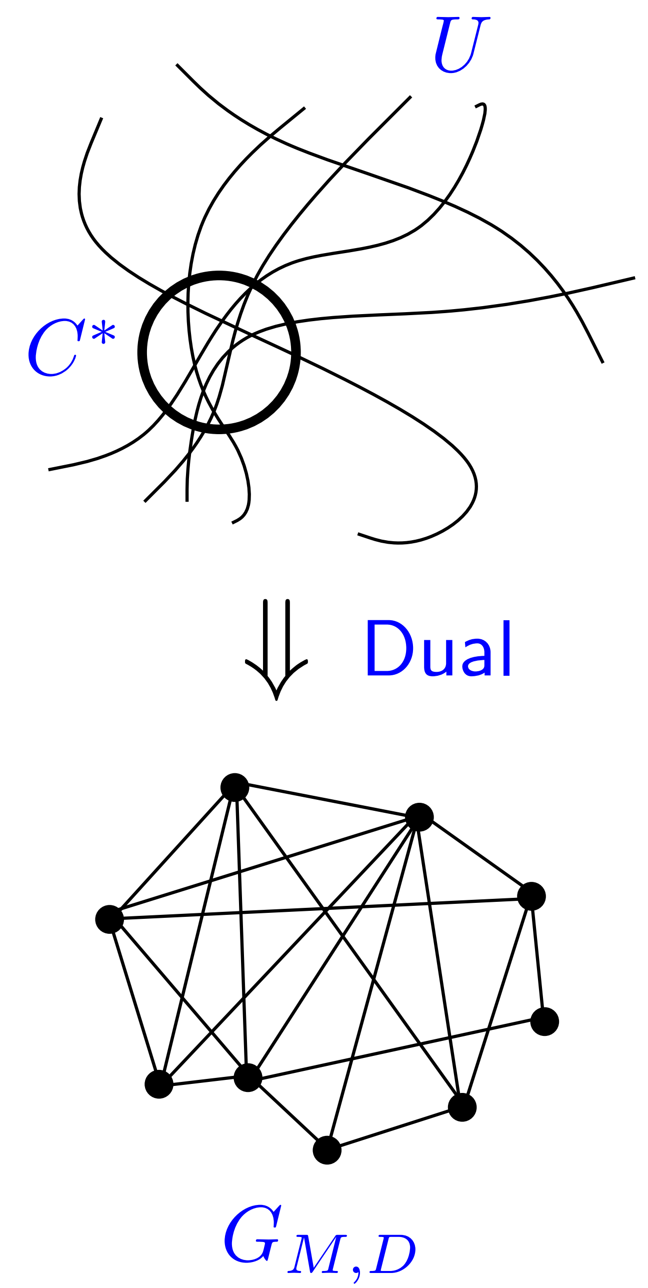
Idea of the proof for the “snake” lemma

The “snake” lemma

If $|C^u| \leq 10/11 \cdot |Y|$ and $m \leq \gamma\, |Y|$, then $\mathbf{E}[|\{x \in U \mid q_{\texttt{END}}(x) \cap C^* \neq \emptyset\}|] \geq \Omega(u)$.

- The **intersection graph**.

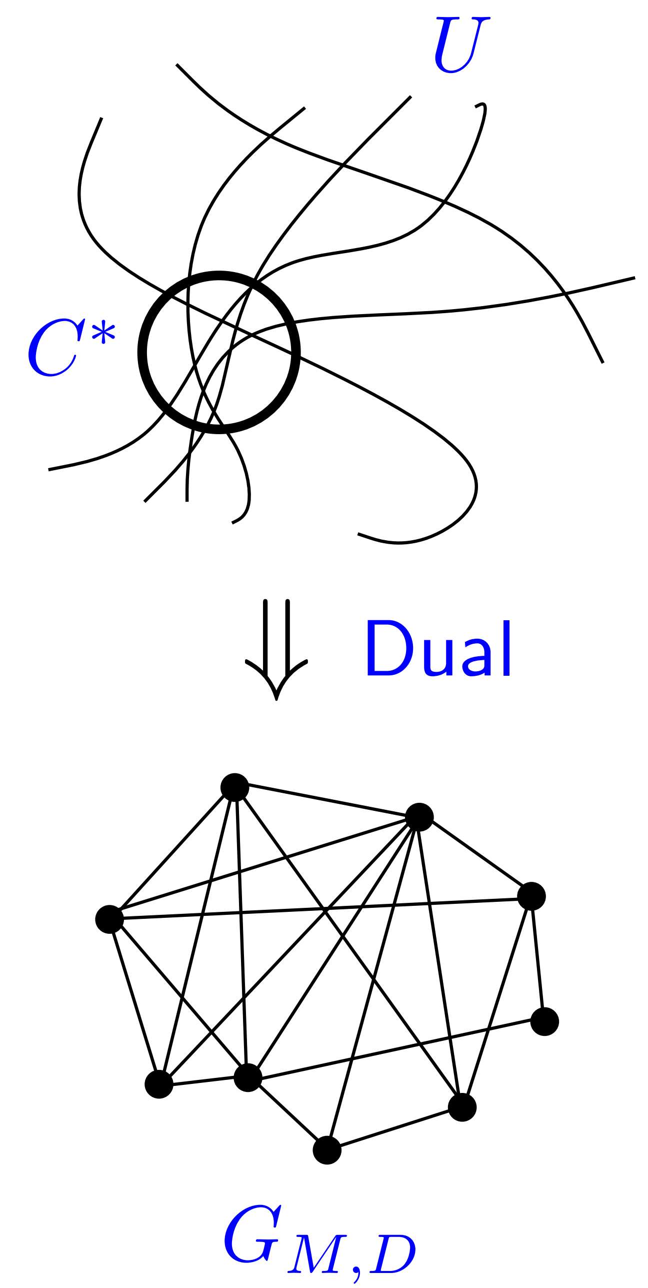
  Under disk state D and cache state M, the graph $G_{M,D} = (U, E)$, where $x, y \in U$ are connected by an edge if $q_{M,D}(x) \cap q_{M,D}(y) \neq \emptyset$



Idea of the proof for the “snake” lemma

The “snake” lemma

If $|C^u| \leq 10/11 \cdot |Y|$ and $m \leq \gamma\, |Y|$, then
$\mathbf{E}[|\{x \in U \mid q_{\mathtt{END}}(x) \cap C^* \neq \emptyset\}|] \geq \Omega(u)$.



- The **intersection graph**.

  Under disk state D and cache state M, the graph $G_{M,D} = (U, E)$, where $x, y \in U$ are connected by an edge if $q_{M,D}(x) \cap q_{M,D}(y) \neq \emptyset$

- **Idea of the proof.**

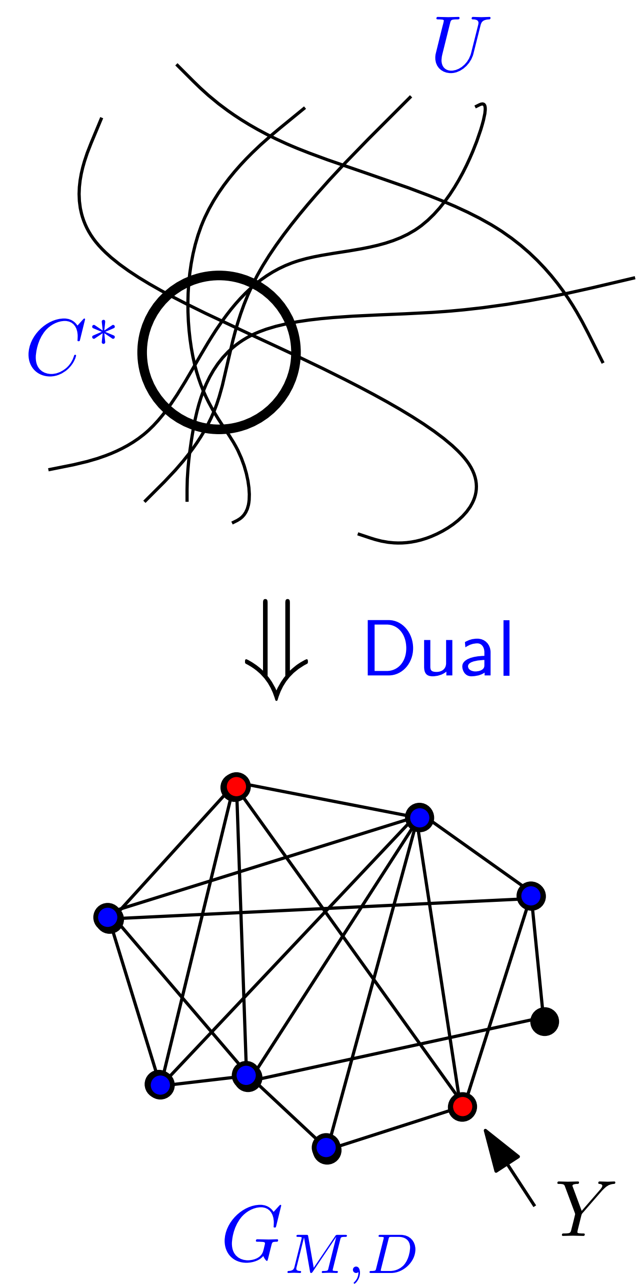
  Intersection graph must be dense at END if $t_u \leq 0.9$. (next slides)

Idea of the proof for the "snake" lemma

The "snake" lemma

If $|C^u| \leq 10/11 \cdot |Y|$ and $m \leq \gamma\,|Y|$, then

$\mathbf{E}[|\{x \in U \mid q_{\mathtt{END}}(x) \cap C^* \neq \emptyset\}|] \geq \Omega(u)$.



- The **intersection graph**.

  Under disk state D and cache state M, the graph $G_{M,D} = (U, E)$, where $x, y \in U$ are connected by an edge if $q_{M,D}(x) \cap q_{M,D}(y) \neq \emptyset$

- **Idea of the proof.**

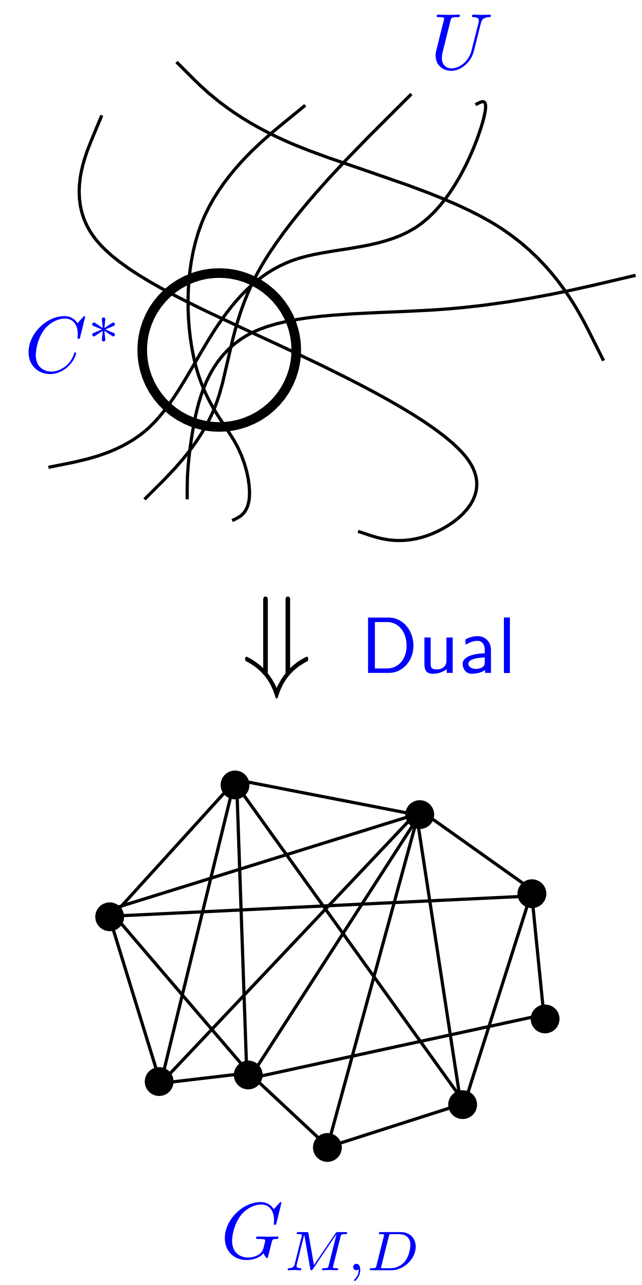
  Intersection graph must be dense at END if $t_u \leq 0.9$. (next slides)

  $\Rightarrow$ For a randomly chosen set Y, the neighborhood of Y must be large. Thus

  $\mathbf{E}[|\{x \in U \mid q_{\mathtt{END}}(x) \cap C^* \neq \emptyset\}|] \geq \Omega(u)$

Idea of the proof for the “snake” lemma (cont.)

- Why intersection graph must be dense at END if $t_u \leq 0.9$?
  - Let D, M and D', M' be the states of the cache and disk before and after the epoch.

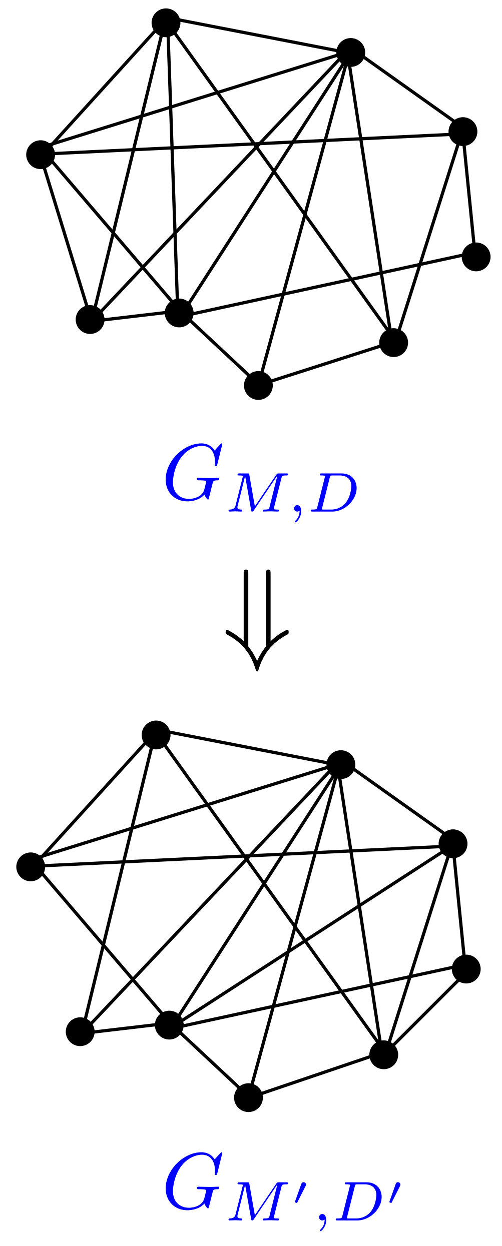


Idea of the proof for the “snake” lemma (cont.)

- Why intersection graph must be dense at `END` if $t_u \le 0.9$?
  - Let D, M and D', M' be the states of the cache and disk before and after the epoch.

If $t_u \le 0.9$

$\Rightarrow$ $|C^u|$ is small and
D' cannot differ from D by much.



Idea of the proof for the "snake" lemma (cont.)

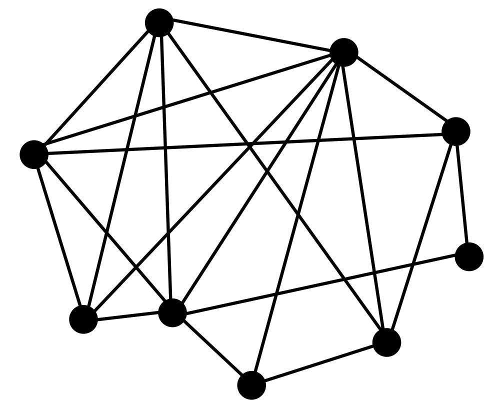
- Why intersection graph must be dense at END if $t_u \leq 0.9$?
  - Let D, M and D', M' be the states of the cache and disk before and after the epoch.

If $t_u \leq 0.9$

$\Rightarrow$ $|C^u|$ is small and D' cannot differ from D by much.

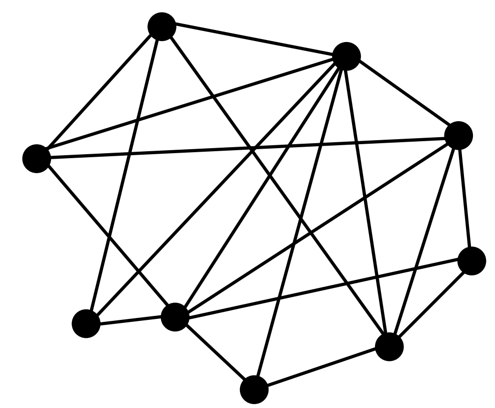
$+$ $|M|$ is small

$\Rightarrow$ Structure of G will not change by much.



$G_{M,D}$

$\Downarrow$



$G_{M',D'}$

Idea of the proof for the "snake" lemma (cont.)

- Why intersection graph must be dense at END if $t_u \leq 0.9$?

  - Let D, M and D', M' be the states of the cache and disk before and after the epoch.

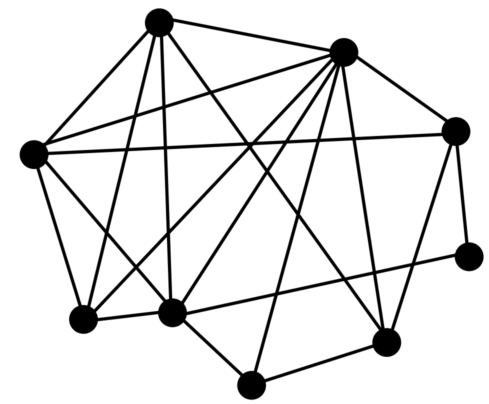
  If $t_u \leq 0.9$

  $\Longrightarrow$ $|C^u|$ is small and D' cannot differ from D by much.

  $+$ $|M|$ is small

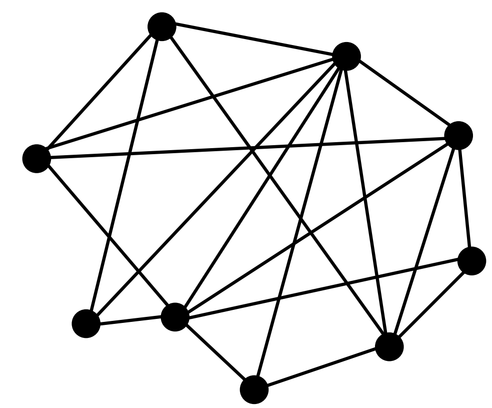
  $\Longrightarrow$ Structure of G will not change by much.

  $\Longrightarrow$ Since Y is *random* at END, $\nexists$ C^u such that 0.99 fraction of Y whose query paths intersect a small set C^u at END, if $G_{M'D'}$ is sparse.



$G_{M,D}$

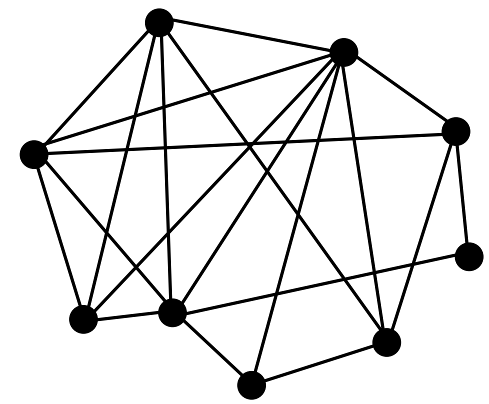
$\Downarrow$



$G_{M',D'}$

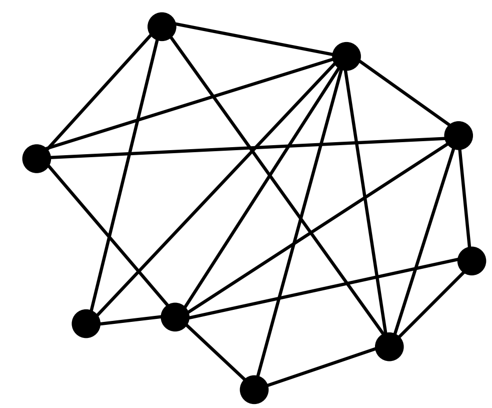
Idea of the proof for the "snake" lemma (cont.)

- Why intersection graph must be dense at END if $t_u \le 0.9$?
  - Let D, M and D', M' be the states of the cache and disk before and after the epoch.



$G_{M,D}$

$\Downarrow$



$G_{M',D'}$

If $t_u \le 0.9$

$\Longrightarrow$ $|C^u|$ is small and
D' cannot differ from D by much.

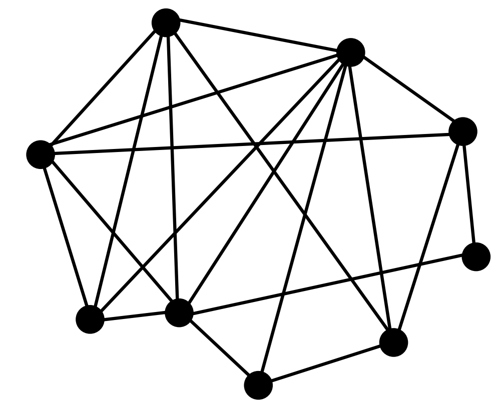
$+$ $|M|$ is small

$\Longrightarrow$ Structure of G will not change by much.

$\Longrightarrow$ Since Y is *random* at END,
$\nexists$ C^u such that 0.99 fraction of Y whose query paths intersect a small set C^u at END,
if $G_{M'D'}$ is sparse. $\Longrightarrow$ $G_{M',D'}$ is dense

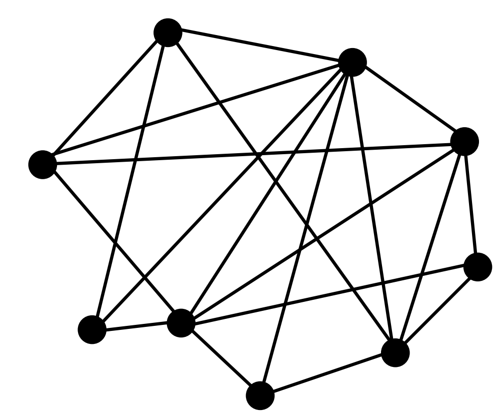
Idea of the proof for the "snake" lemma (cont.)

- Why intersection graph must be dense at END if $t_u \leq 0.9$?
  - Let D, M and D', M' be the states of the cache and disk before and after the epoch.



$G_{M,D}$

$\Downarrow$



$G_{M',D'}$

If $t_u \leq 0.9$

$\Rightarrow$ $|C^u|$ is small and D' cannot differ from D by much.

$+$ $|M|$ is small

$\Rightarrow$ Structure of G will not change by much.

cheat a bit

$\Rightarrow$ Since Y is *random* at END, $\nexists$ C^u such that 0.99 fraction of Y whose query paths intersect a small set C^u at END, if $G_{M'D'}$ is sparse. $\Rightarrow$ $G_{M',D'}$ is dense

Futher work

- We still cannot handle fast updates.
  e.g. if $t_u = O(1/B)$, $t_q = \Omega(n^\epsilon)$?

Futher work

- We still cannot handle fast updates.

  e.g. if $t_u = O(1/B)$, $t_q = \Omega(n^\epsilon)$?

- Lower bounds of other dynamic problems in the external memory.

  e.g., for union-find, need super-log query time
  if we want to batch up the updates?

  Call for new techniques?

Futher work

- We still cannot handle fast updates.

  e.g. if $t_u = O(1/B)$, $t_q = \Omega(n^\epsilon)$?

- Lower bounds of other dynamic problems in the external memory.

  e.g., for union-find, need super-log query time
  if we want to batch up the updates?

  Call for new techniques?

  And the priority queue.

  $\max\{\texttt{delete}, \texttt{deletemin}\} \geq \frac{1}{B} \log_2 n$?

The End

THANK YOU

Q and A