Lower bounds for succinct data structures

Emanuele Viola

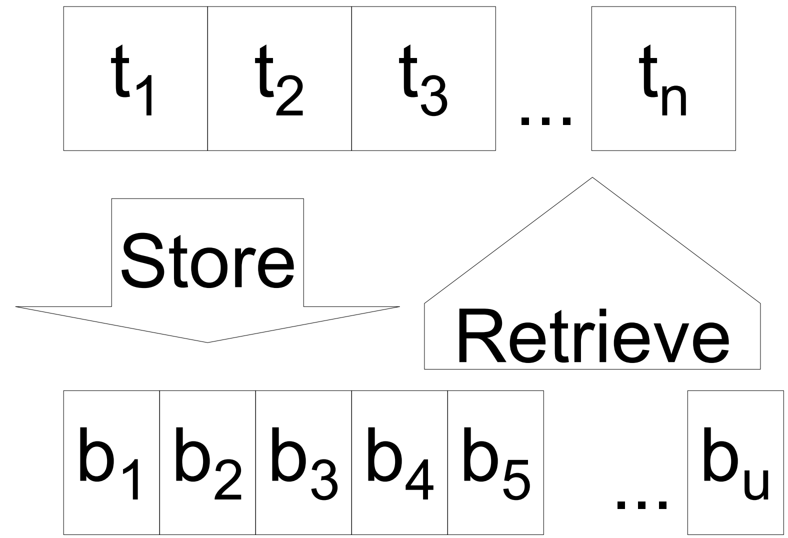
Northeastern University

December 2009

Bits vs. trits

- Store n “trits” $t_1, t_2, \ldots, t_n \in \{0,1,2\}$

  In u bits $b_1, b_2, \ldots, b_u \in \{0,1\}$



- Want:

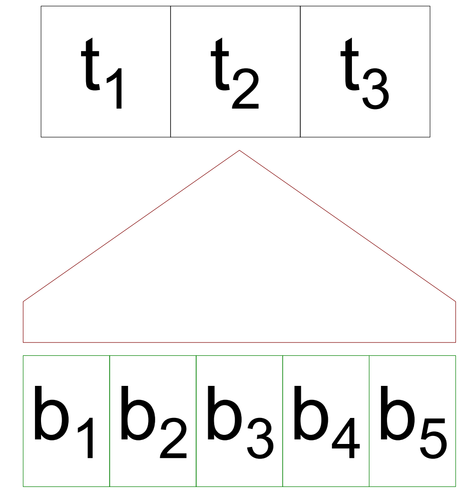
  Small space u $\left(\text{optimal} = \lceil n \lg_2 3 \rceil\right)$

  Fast retrieval: Get t_i by probing few bits $\left(\text{optimal} = 2\right)$

Two solutions

- Arithmetic coding:

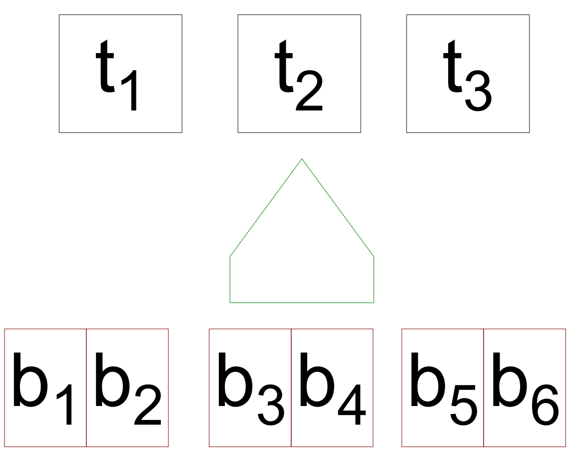
  Store bits of $(t_1, \ldots, t_n) \in \{0, 1, \ldots, 3^n - 1\}$



  Optimal space: $\lceil n \lg_2 3 \rceil \approx n \cdot 1.584$

  Bad retrieval: To get t_i probe all > n bits

- Two bits per trit

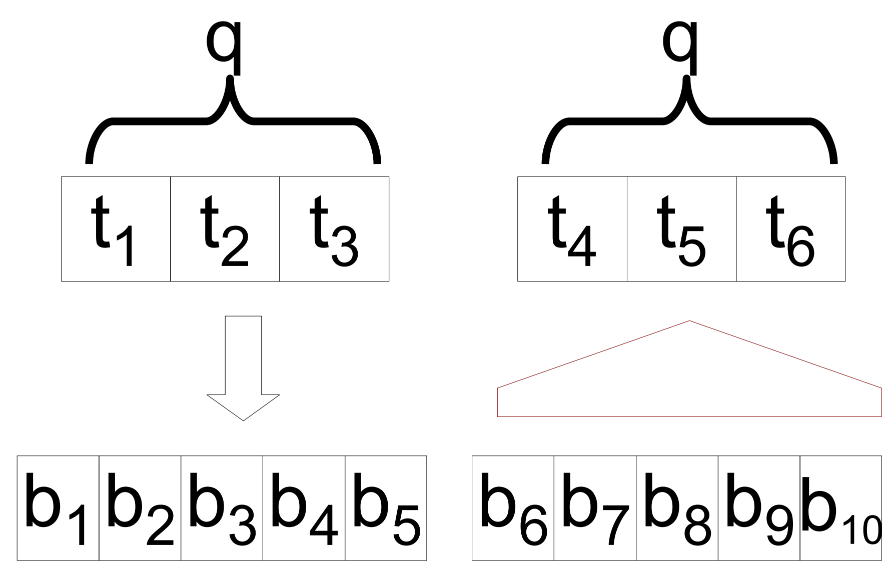


  Bad space: $n \cdot 2$

  Optimal retrieval: Probe 2 bits

Polynomial tradeoff

- Divide n trits $t_1, \ldots, t_n \in \{0,1,2\}$ in blocks of q



- Arithmetic-code each block

Space: $\lceil q \lg_2 3 \rceil n/q < (q \lg_2 3 + 1)\, n/q$

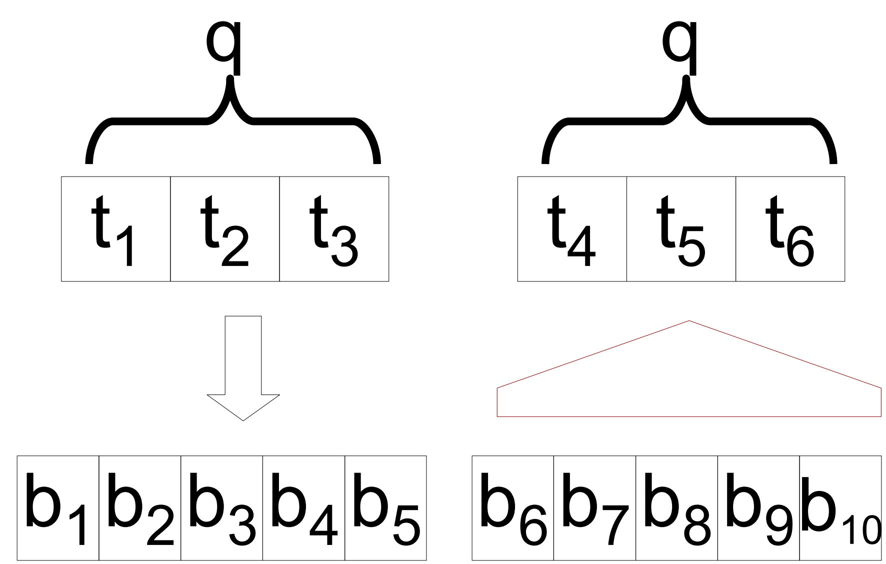
$= n \lg_2 3 + n/q$

Retrieval: Probe O(q) bits

polynomial tradeoff between redundancy, probes

Polynomial tradeoff

- Divide n trits $t_1, \ldots, t_n \in \{0,1,2\}$ in blocks of q
- Arithmetic-code each block



Space: $\lceil q \lg_2 3 \rceil n/q = (q \lg_2 3 + 1/q^{\Theta(1)})\ n/q$

$= n \lg_2 3 + n/q^{\Theta(1)}$

Retrieval: Probe O(q) bits

polynomial tradeoff between redundancy, probes

Logarithmic forms

Exponential tradeoff

- Breakthrough [Pătraşcu '08, later + Thorup]

Space: $n \lg_2 3 + n/2^{\Omega(q)}$

Retrieval: Probe q bits

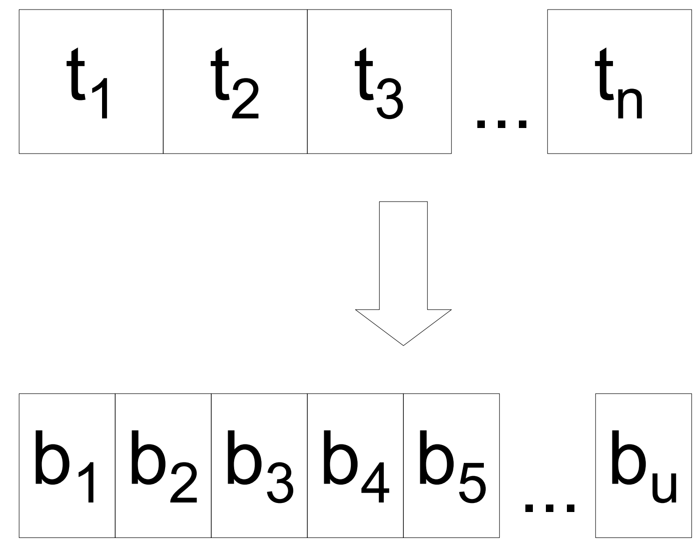
exponential tradeoff between redundancy, probes

- E.g., optimal space $\lceil n \lg_2 3 \rceil$, probe $O(\lg n)$

Our results

- Theorem[V.]:
  Store n trits $t_1, \ldots, t_n \in \{0,1,2\}$
  in u bits $b_1, \ldots, b_u \in \{0,1\}$.

  If get t_i by probing q bits
  then space $u > n \lg_2 3 + n/2^{O(q)}$.



- Matches [Pătraşcu Thorup]: space $< n \lg_2 3 + n/2^{\Omega(q)}$

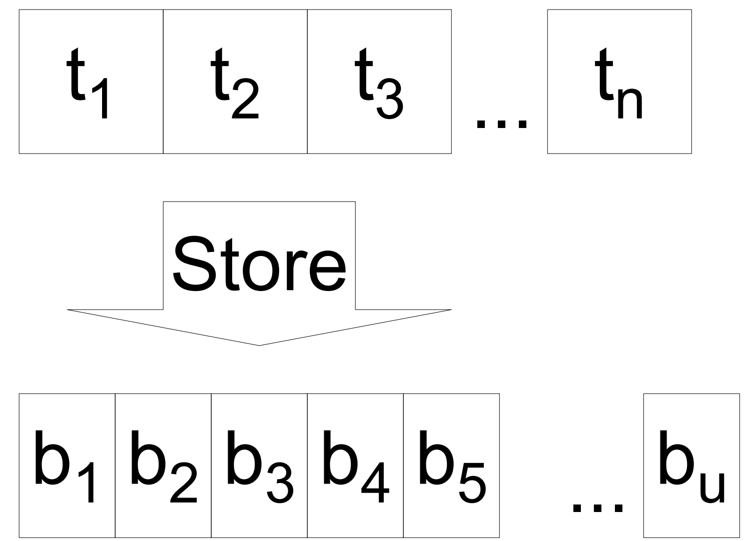
Outline

- Bits vs. trits
- Proof bits vs. trits
- Bits vs. sets
- Cells vs. prefix sums

Recall our results

- Theorem:
  Store n trits $t_1, \ldots, t_n \in \{0,1,2\}$
  in u bits $b_1, \ldots, b_u \in \{0,1\}$.

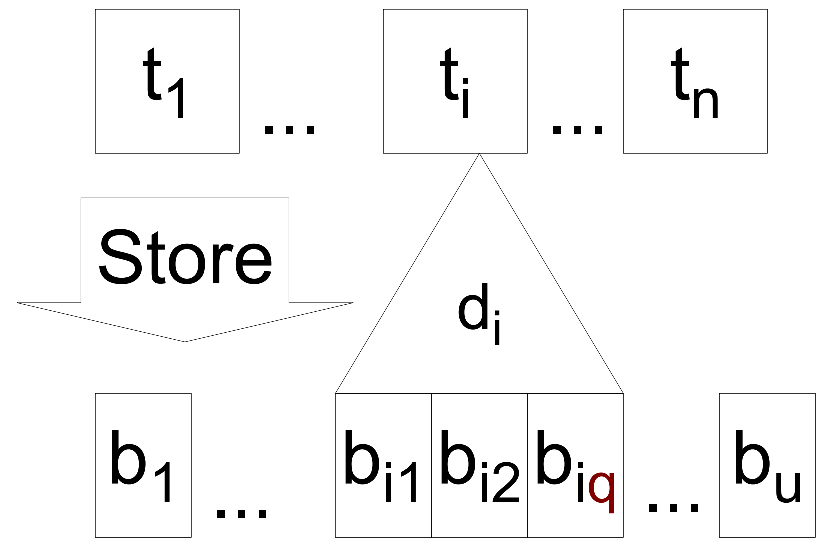
  If get t_i by probing q bits
  then space $u > n \lg_2 3 + n/2^{O(q)}$.



- For now, assume non-adaptive probes:
  $t_i = d_i(b_{i1}, b_{i2}, \ldots, b_{iq})$

Proof idea

- $t_i = d_i\ (b_{i1}, b_{i2}, \ldots, b_{iq})$



- Uniform $(t_1, \ldots, t_n) \in \{0,1,2\}^n$

  Let $(b_1, \ldots, b_u) := \text{Store}(t_1, \ldots, t_n)$

- Space $u \approx$ optimal $\Rightarrow (b_1, \ldots, b_u) \in \{0,1\}^u \approx$ uniform $\Rightarrow$

$$\boxed{1/3 = \Pr[\, t_i = 2 \,] = \Pr[\, d_i\ (b_{i1}, \ldots, b_{iq}) = 2 \,] \approx A / 2^q \neq 1/3}$$

Contradiction, so space u >> optimal Q.e.d.

Information-theory lemma

[Edmonds Rudich Impagliazzo Sgall, Raz, Shaltiel V.]

Lemma: Random $(b_1, \ldots, b_u)$ uniform in $B \subseteq \{0,1\}^u$

$|B| \approx 2^u \Rightarrow$ there is large set $G \subseteq [u]$:

for every $i_1, \ldots, i_q \in G$: $(b_{i_1}, \ldots, b_{i_q}) \approx$ uniform in $\{0,1\}^q$

Proof: $|B| \approx 2^u \Rightarrow H(b_1, \ldots, b_u)$ large

$\Rightarrow H(b_i \mid b_1, \ldots, b_{i-1})$ large for many i ($\in G$)

Closeness[$(b_{i_1}, \ldots, b_{i_q})$, uniform] $\geq H(b_{i_1}, \ldots, b_{i_q})$

$\geq H(b_{i_q} \mid b_1, \ldots, b_{i_q-1}) + \ldots + H(b_{i_1} \mid b_1, \ldots, b_{i_1-1})$, large Q.e.d.

Proof

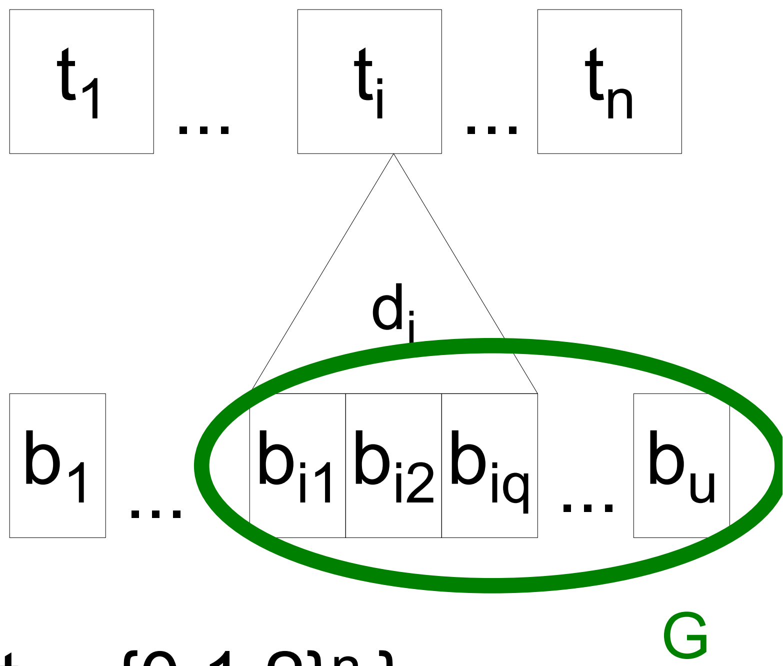
- Argument OK **if** probes in G

- $t_i = d_i\,(b_{i1}, b_{i2}, \ldots, b_{iq})$

- Uniform $(t_1, \ldots, t_n) \in \{0,1,2\}^n$

$$\Downarrow$$

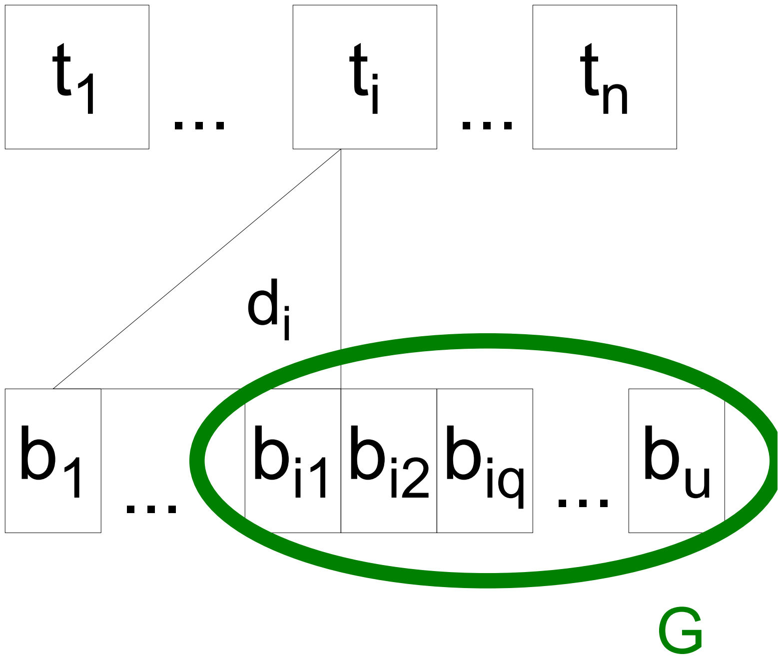
uniform $(b_1, \ldots, b_u) \in B := \{\mathrm{Store}(t) \mid t \in \{0,1,2\}^n\}$



$|B| = 3^n \approx 2^u \Rightarrow$ (Lemma) $\Rightarrow (b_{i1}, \ldots, b_{iq}) \approx$ uniform $\Rightarrow$

$$\boxed{1/3 = \Pr[\, t_i = 2 \,] = \Pr[\, d_i\,(b_{i1}, \ldots, b_{iq}) = 2 \,] \approx A / 2^q \neq 1/3}$$

Probes not in G

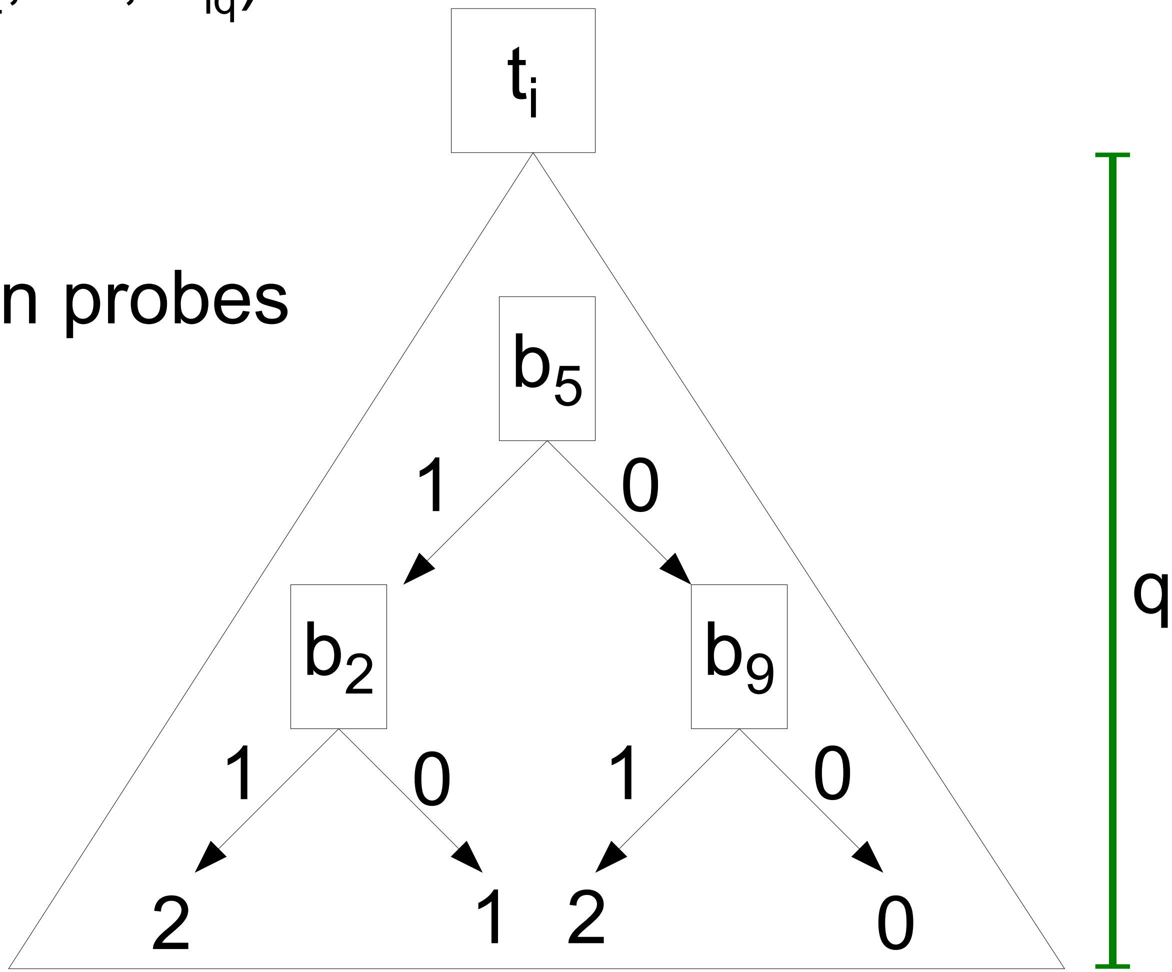


- If every t_i probes bits not in G
- Argue as in [Shaltiel V.]:
- Condition on heavy bits := probed by many t_i
- Can find $t_i \approx$ uniform in {0,1,2}, all probes in G

Handling adaptivity

- So far $t_i = d_i\,(b_{i1}, b_{i2}, \ldots, b_{iq})$

- In general,
  q adaptively chosen probes
  = decision tree
    2^q bits
    depth q



$1/3 = \Pr[\,t_i = 2\,] = \Pr[\,d_i\,(b_{i1}, \ldots, b_{i2^q}) = 2\,] \approx A / 2^q \neq 1/3$

Outline

- Bits vs. trits
- Proof bits vs. trits
- Bits vs. sets
- Cells vs. prefix sums

Bits vs. sets

- Store $S \subseteq \{1, 2, \ldots, n\}$ of size $|S| = k$

01001001101011

In u bits $b_1, \ldots, b_u \in \{0,1\}$

b_1	b_2	b_3	b_4	b_5	...	b_u

- Want:

  Small space u $\left(\text{optimal} = \lceil \lg_2 (n \text{ choose } k) \rceil\right)$

  Answer "$i \in S$?" by probing few bits $\left(\text{optimal} = 1\right)$

Previous results

- Store $S \subseteq \{1, 2, \ldots, n\}$, $|S| = k$ in bits, answer "$i \in S$?"
- [Minsky Papert '69] Average-case study
- [Buhrman Miltersen Radhakrishnan Venkatesh; Pagh '00]

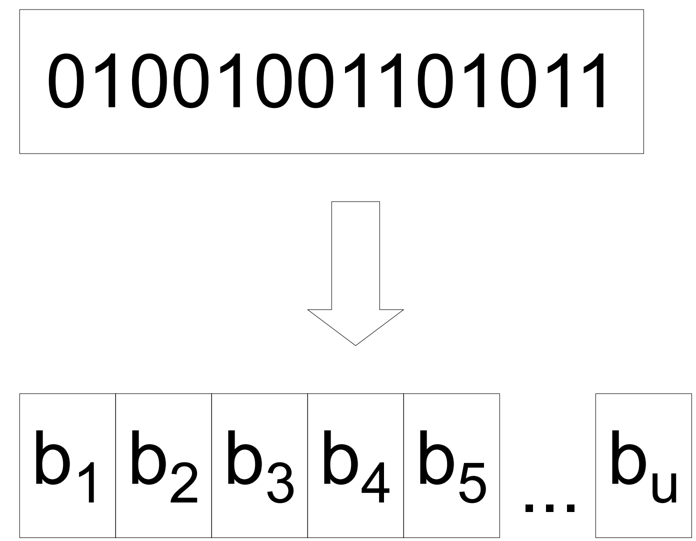
  Space O(optimal), probe O(lg(n/k))

  Lower bounds for $k < n^{1-\varepsilon}$
- No lower bound was known for $k = \Omega(n)$

Our results

- Theorem[V.]:
  Store $S \subseteq \{1, 2, \ldots, n\}$, $|S| = n/3$
  in u bits $b_1, \ldots, b_u \in \{0,1\}$

  If answer "$i \in S$?" probing q bits
  then space $u >$ optimal $+ n/2^{O(q)}$.



- First lower bound for $|S| = \Omega(n)$

Outline

- Bits vs. trits

- Proof bits vs. trits

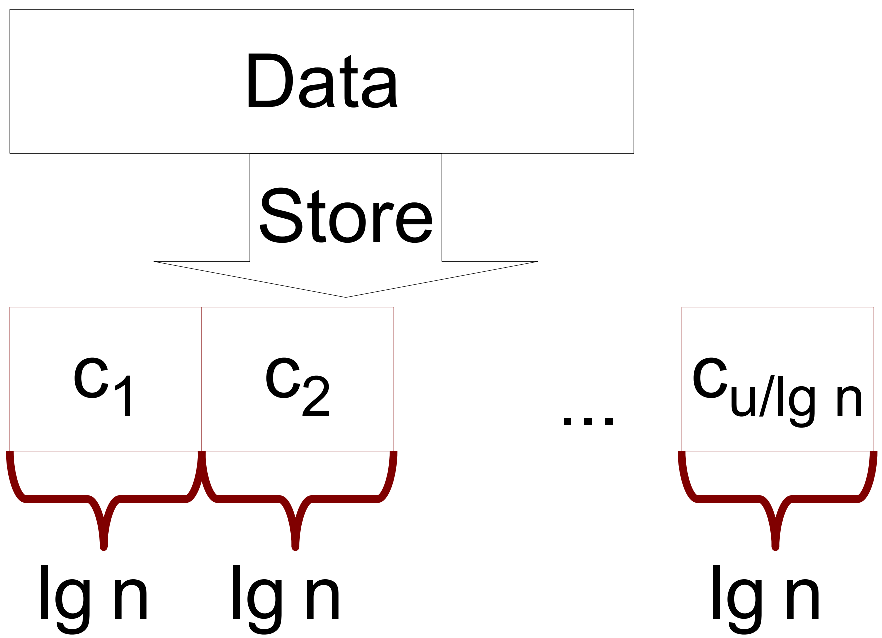
- Bits vs. sets

- Cells vs. prefix sums

Cell-probe model

- So far: q = number of bit probes

- Cell model: q = number of probes in cells of lg(n) bits



- Relationship: q bit $\subseteq$ q cell $\subseteq$ q lg(n) bit

Results in cell-probe model

- Cells vs. trits:

  $q = O(1)$, optimal space = $\lceil n \lg_2 3 \rceil$ [Pătraşcu Thorup]

  $q = 1 \Rightarrow$ space > $n \lg_2 3 + n/\lg^{O(1)} n$ [this work]

- Cells vs. sets:

  q probes, space = optimal + $n / \lg^{\Omega(q)} n$ [Pagh, Pătraşcu]

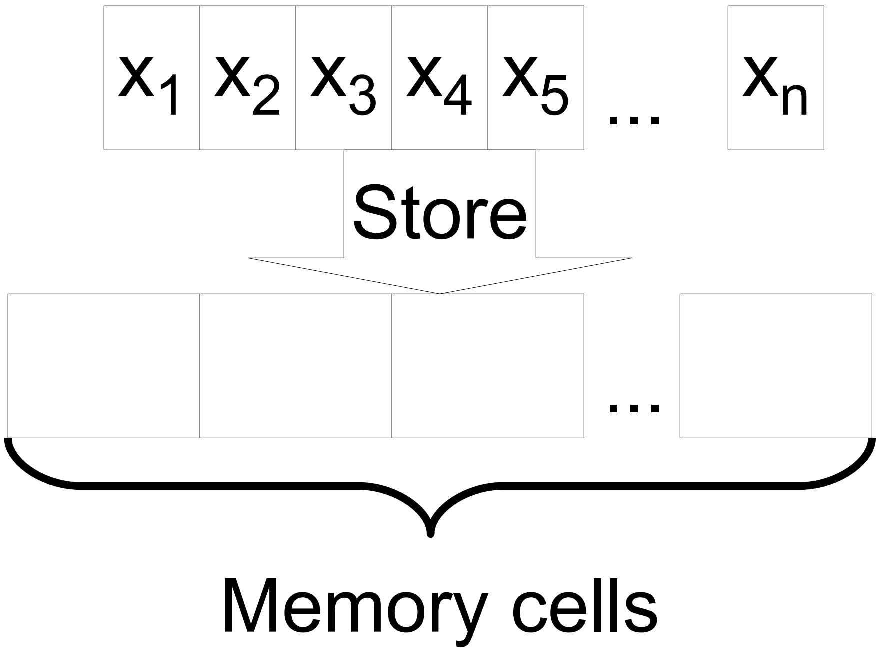
  Lower bounds?

Outline

- Bits vs. trits
- Proof bits vs. trits
- Bits vs. sets
- Cells vs. prefix sums

Prefix sums

- Store n bits $x_1, x_2, \ldots, x_n \in \{0,1\}$ in memory cells



- Want:

  Small space

  Fast answer prefix sum (a.k.a. Rank) queries:

$$\mathrm{Sum}(i) := \sum_{k \leq i} x_k \in \{0, 1, 2, \ldots, n\}$$

History

- Fundamental problem: succinct trees, sets, ...
- Trivial — Space = $n \lg n$; Time = 1 cell probe
- [Jacobson '89] — Space = $n + O(n / \lg n)$; Time = $O(1)$ cell probes
- [Pătraşcu '08] — Space = $n + n / \lg^q n$; Time = $O(q)$ cell probes

Our results

- Theorem[Pătraşcu V.]:
  Store n bits in memory

  If answer Sum(i) := $\sum_{k \leq i} x_k$ queries

  by probing q cells then space > n+ $n/\lg^{O(q)} n$.

- Matches [Pătraşcu]: space < n + $n / \lg^{\Omega(q)} n$

Proof idea

- Efficient data structure $\Rightarrow$ Break queries' correlations

- For $i < j$, $A \subseteq \{0,1\}^n$

$$0 = \Pr_{x \in A} [\ \mathrm{Sum}(i) > t \text{ AND } \mathrm{Sum}(j) < t]$$

$$\approx \Pr_{x \in A} [\ \mathrm{Sum}(i) > t\ ]\ \ \Pr_{x \in A}[\mathrm{Sum}(j) < t]$$

$$> \qquad (1/10) \quad (1/10) \qquad >> 0$$

- Contradiction, so data structure cannot be efficient

Proof idea

$0 = \Pr_{x \in A} [\text{ Sum(i)} > t \text{ AND Sum(j)} < t]$

$\approx \Pr_{x \in A} [\text{ Sum(i)} > t\ ] \ \ \Pr_{x \in A}[\text{Sum(j)} < t]$ (1)

$>$ (1/10) (1/10) (2)

- Reasoning:

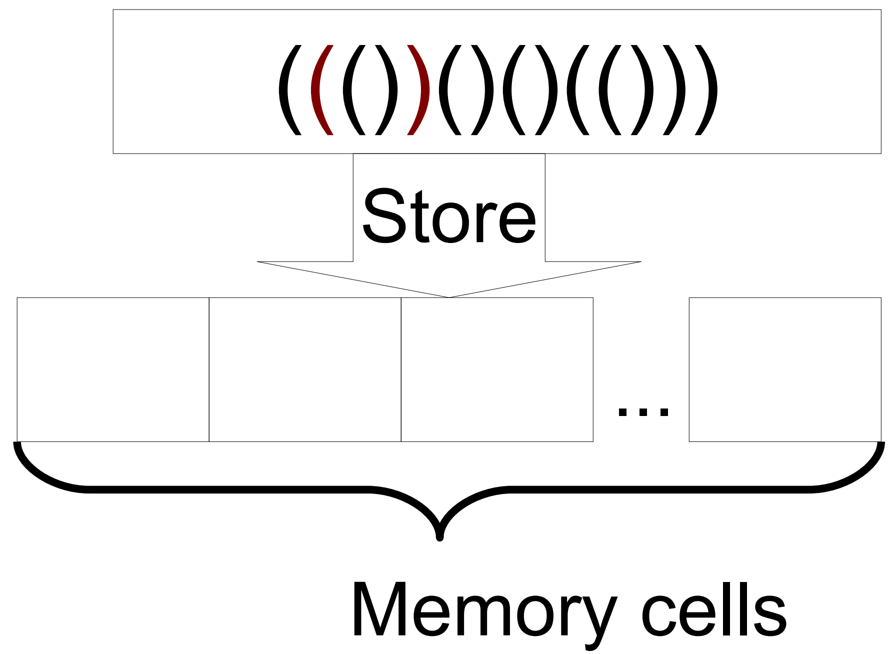
  Fix heavy cells. Then ∃ i, j s.t. Sum(i) and Sum(j):

  (1) depend on disjoint, nearly uniform cells ⇒ independent

  (2) have high entropy

Balanced brackets

- Store n balanced brackets



- Want:

  Small space

  Fast answer match queries:

- Theorem[V.]: space > optimal + $n/\lg^{2^{O(q)}} n$.

  for non-adaptive q probes

- [Pătraşcu]: space < optimal + $n / \lg^{\Omega(q)} n$ non-adaptive

Summary

- New lower bounds for basic data structures:

  Representing trits, sets, prefix sums, balanced brackets using space = optimal + redundancy

- Sometimes matching [Pătraşcu]

- Open problems: storing sets:
  2 cell probes and optimal space?
  Bit-probe lower bounds for set-size n/4 ? (have n/3)

Future directions

- Lower bounds for generating distributions

- Example: $f : \{0,1\}^r \to \{0,1\}^n$
  each bit f_i depends on $\leq q$ input bits
  prove f(uniform) far from uniform on sets of size n/4

- Known[V.]: distance $\geq 1/2^{O(q)}$

- Open: distance $\geq 1 - o(1)$
  $\Rightarrow$ Lower bound for storing sets of size n/4

- $\sum\prod\surd\cap\notin\cup\supset\supseteq\not\subset\subset\subseteq\in\Downarrow\Rightarrow\Uparrow\Leftarrow\Leftrightarrow\vee\wedge\geq\leq\forall\exists\Omega\alpha\beta\varepsilon\gamma\delta\rightarrow$
- $\neq\approx$