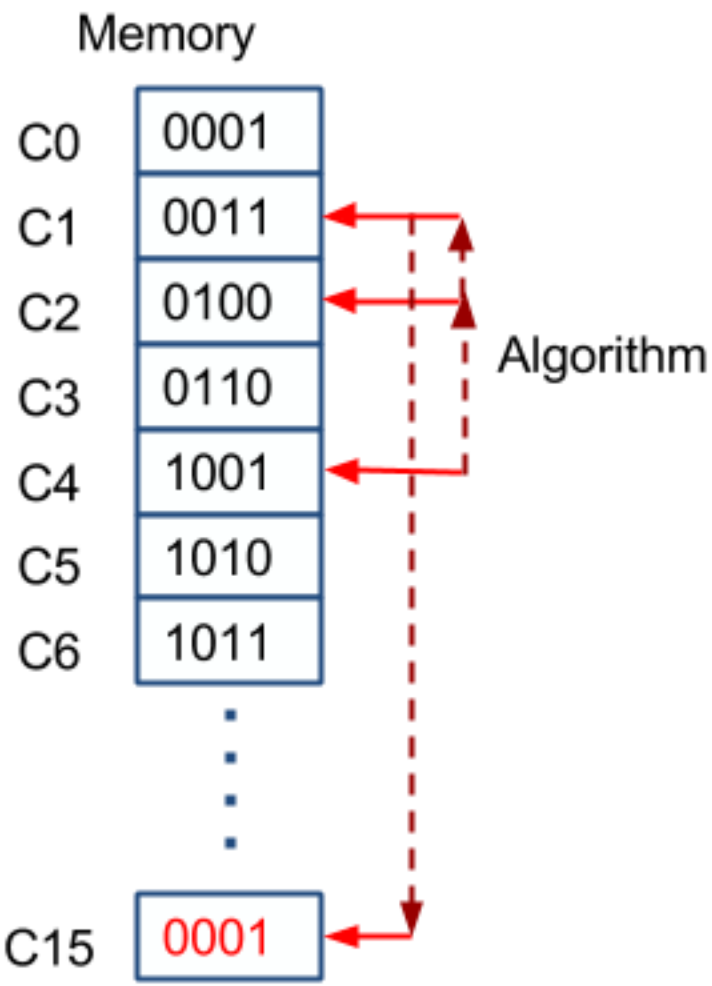
Non-Adaptive Data Structure Bounds for Dynamic Predecessor Search

Joe Boninger, **Joshua Brody**, Owen Kephart

Swarthmore College

Cell Probe Model [Yao81]

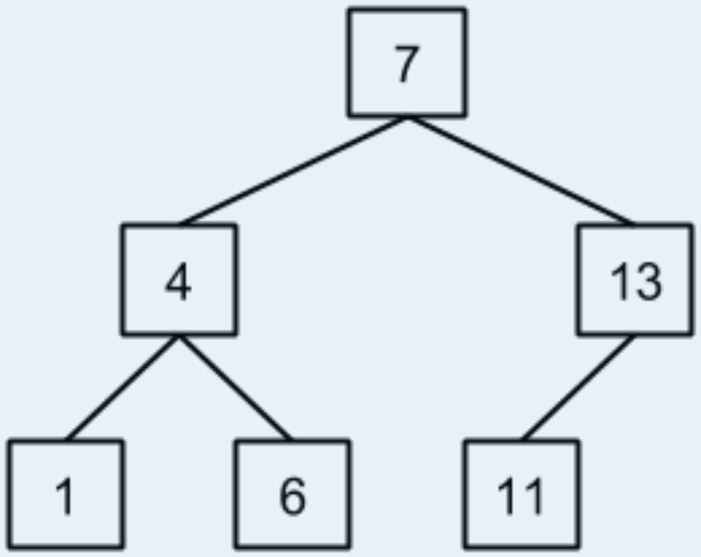


- Memory consists of **w-bit** cells
- Updates/queries charged for **# probes**
- All other computation *for free*
- Cell Probe Complexity: **# probes** required to maintain DS.

Dynamic data structures

- Maintain a set of data S, support updates and queries. e.g.
  - $\mathbf{S \subseteq \{1, \dots, m\}}$
  - Updates: insert/delete element
  - Query(x): is $\mathbf{x \in S}$?
  - **t_u,t_q:** update/query time
- Goal: show **max{t_u,t_q} >= poly(m)**

S = {1, 4, 6, 7, 11, 13}



Current State of the art: $max\{t_u, t_q\} = \Omega\left(\left(\frac{\log m}{\log \log m}\right)^2\right)$ [Larsen12]

Previous results/hard problems

- [Larsen 12a, 12b]: $\boldsymbol{\Omega}\left(\left(\frac{\log m}{\log\log m}\right)^2\right)$ for 2D-range counting, polynomial evaluation
- [CGL15, WY16]: $\boldsymbol{\Omega}\left(\left(\frac{\log m}{\log\log m}\right)^2\right)$ amortized bounds
- [Patrascu10]: polynomial lower bounds from CC of Multiphase
- [CEEP12]: strongest Multiphase conjecture false, but weaker version still shows polynomial DDS lower bounds
- [**B**L15]: polynomial lower bounds for non-adaptive DS

Non-Adaptive Data Structures

- **Non-Adaptive Queries:** cells probed by query algorithm chosen in advance
- **Non-Adaptive Updates:** cells probed by update algorithm chosen in advance
- **Memoryless Updates:** non-adaptive, plus contents of each write depend only on update, prev. contents.

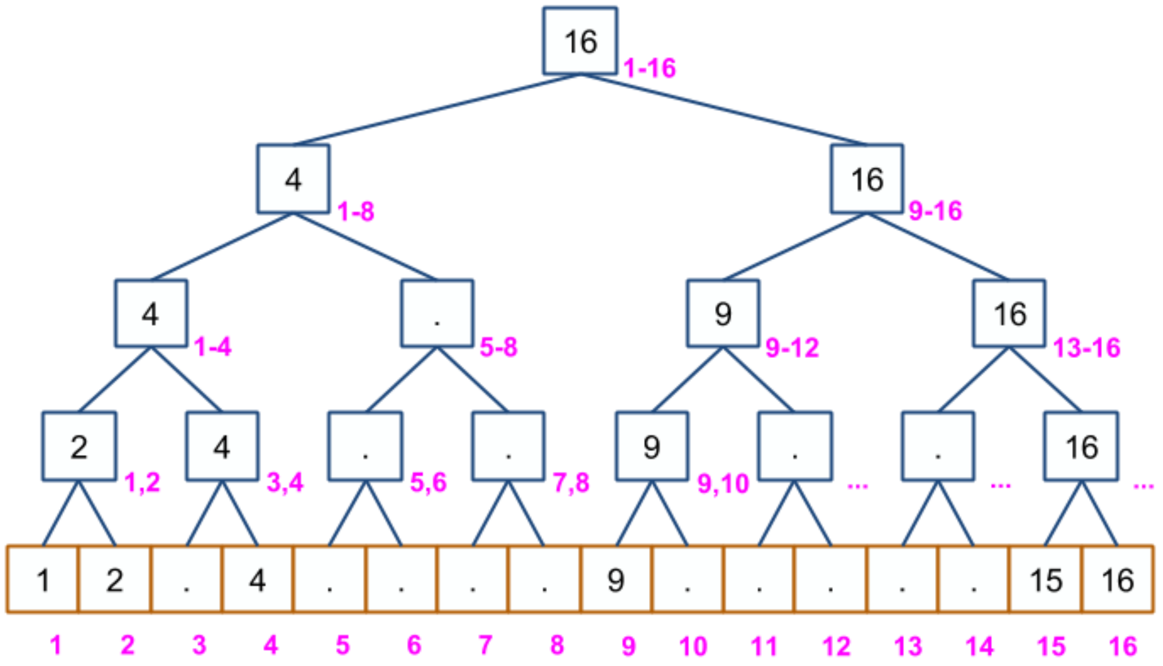
Non-Adaptive Data Structure: non-adaptive queries, updates

Memoryless Data Structure: non-adaptive queries, memoryless updates

Predecessor Search

Maintain set $T \subseteq [m]$ of $\leq n$ *items*, support

- Insert(j)
- Delete(j)
- **Pred(i) = $\max\{j \leq i : j \in T\}$**



Our Results

- Adaptive DS for Pred with **t_u, t_q = O(log log m)** [van Emde Boas 75]
- Non-Adaptive: $t_u, t_q = O\left(min\left\{\frac{n \log m}{w}, \frac{\log m}{\log\left(w / \log m\right)}\right\}\right)$
- Non-Adaptive: $\max\{t_u, t_q\} = \Omega\left(min\left\{\frac{n \log m}{w \log w}, \frac{\log m}{\log w}\right\}\right)$

Recent independent work [Rao, Ramamoorthy 17]:

- Either $t_q = \Omega\left(\frac{\log m}{\log\log m + \log w}\right)$ or $t_u = \Omega\left(\frac{t_q\, m^{1/2(1+t_q)}}{\log(m)}\right)$
- Only requires non-adaptive queries

Theorem: Let $\alpha = \min\{n, w/2\}$. Then, any non-adaptive data structure solving dynamic predecessor with $t_u = O(\log m)$ must have

$$t_q \geq \frac{\alpha \log m}{2w \log(w \cdot t_u)}$$

Theorem: Let $\alpha = \min\{n, w/2\}$. Then, any non-adaptive data structure solving dynamic predecessor with $t_u = O(\log m)$ must have

$$t_q \geq \frac{\alpha \log m}{2w \log(w \cdot t_u)}$$

Idea: grow set of cells **C**, maintain query set **A** such that each query in **A** probes every cell in **C**

Setup:
- Predecessor w/wraparound: **Pred*(i) = min(Pred(i), Pred(m))**
- $\mathbf{Q_i}$: cells probed by **Pred(i)**
- $\mathbf{U_j}$: cells probed by **Insert(j)**

Theorem: Let $\alpha = \min\{n, w/2\}$. Then, any non-adaptive data structure solving dynamic predecessor with $t_u = O(\log m)$ must have

$$t_q \geq \frac{\alpha \log m}{2w \log(w \cdot t_u)}$$

Main Technical Lemma: Let C be a set of cells in the data structure and $A \subseteq [m]$. If

1. $|A| \geq \sqrt{m}$

2. $|C| \leq \frac{\alpha \log m}{5w}$, and

3. For all $i \in A$, Pred(i) probes each cell in C , then

There is $j \in A$ and subset $A' \subseteq A$ with $|A'| \geq \frac{|A|}{w^2}$ such that for all $i \in A'$,

U_j and **Q_i** intersect *outside of C*.

Claim: For all $1 \le k \le \frac{\alpha \log m}{2w \log(w \cdot t_u)}$, there is a set of k cells C_k and a set of queries $A_k \subseteq [m]$ such that

1. $|A_k| \ge \frac{m}{w^{2(k-1)} {t_u}^k}$
2. $C_k \subseteq Q_i$ for all $i \in A_k$

Proof: *induction*

Base Case:

- **Q_i, U_j** intersect for each **i,j**
- *Pigeonhole Principle:* there is cell **c** probed by **Insert(j)** and **m/t_u Pred(i)**
- **$C_1 = \{c\}$, $A_1 = \{i:$ Pred(i) probes c$\}$**

Claim: For all $1 \le k \le \frac{\alpha \log m}{2w \log(w \cdot t_u)}$, there is a set of k cells C_k and a set of queries $A_k \subseteq [m]$ such that

1. $|A_k| \ge \frac{m}{w^{2(k-1)} {t_u}^k}$
2. $C_k \subseteq Q_i$ for all $i \in A_k$

Induction Hypothesis: There is $\mathbf{A_k}$, $\mathbf{C_k}$ such that $\mathbf{C_k \subseteq Q_i}$ for all $\mathbf{i} \in \mathbf{A_k}$

Induction Step:

- **MTL:** there is **Insert(j)**, subset $A_k' \subseteq Ak$ with $|A_k'| \ge \frac{|A_k|}{w^2}$ s.t. for all $i \in A'$, $\mathbf{U_j}$ and $\mathbf{Q_i}$ intersect *outside of* C_k.
- *Pigeonhole:* there is cell $\mathbf{c} \in \mathbf{U_j} \setminus \mathbf{C_k}$ probed by $\frac{|A_k'|}{t_u} \ge \frac{|A_k|}{w^2 t_u}$ queries
- $\mathbf{C_{k+1} = \{c\} \cup Ck}$, $\mathbf{A_{k+1} = \{i \in A_k' : \text{Pred}(i) \text{ probes } c\}}$

Main Technical Lemma: Let C be a set of cells in the data structure and $A \subseteq [m]$. If

1. $|A| \geq \sqrt{m}$
2. $|C| \leq \frac{\alpha \log m}{5w}$, and
3. For all $i \in A$, Pred(i) probes each cell in C , then

There is $j \in A$ and subset $A' \subseteq A$ with $|A'| \geq \frac{|A|}{w^2}$ such that for all $i \in A'$,

$\mathbf{U_j}$ and $\mathbf{Q_i}$ intersect *outside of C*.

Proof: suppose implication false. Then

- For every update j, $\mathbf{U_j} \cap \mathbf{Qi} \subseteq \mathbf{C}$ *for all but* $\frac{|A|}{w^2}$ *queries*
- For any set **T** of **α** updates, for all but $\frac{|A|\alpha}{w^2} \leq \frac{|A|}{2w}$ queries,

$$\mathbf{U_j} \cap \mathbf{Qi} \subseteq \mathbf{C} \textit{ for all } \mathbf{j} \in \mathbf{T}$$

- When DS stores T, can use **C** to compute **Pred(i)** for *most* i.
- Use **C** to encode **T**.

Encode arbitrary *spread out* subset $T \subseteq A$ with

1. $|T| = \alpha$
2. $|j - j'| \geq |A|/w$ for all $j, j' \in T$

Fact: There are $2^{\frac{\alpha \log m}{4}(1-o(1))}$ spread out T

1				…				35				…			80			…			123	…	T = {1,35,80,123}

```
Encoding Procedure {
   in
   se
}
```

```
Decoding Procedure {
                       empty DS
                       i
                       |A|/2w times
```

Encoding Length: $|C| \cdot w \leq \frac{\alpha \log m}{5}$ bits

Coding Lower Bound:

encoding arbitrary T requires $\geq \frac{\alpha \log m}{4}(1 - o(1))$ bits

1	1	…	Pred(i)

~~1~~	~~1~~	~~1~~	1	…	1	~~1~~	1	35	35	35	35	…	~~35~~	~~35~~	80	80	80	…	80	80	123	123	Pred(i) w/errors

Less than $|A|/2w$ total errors ➔ Decoder recovers T

Theorem: Let $\alpha = \min\{n, w/2\}$. Then, any non-adaptive data structure solving dynamic predecessor with $t_u = O(\log m)$ must have

$$t_q \geq \frac{\alpha \log m}{2w \log(w \cdot t_u)}$$

Claim: For all $1 \leq k \leq \frac{\alpha \log m}{2w \log(w \cdot t_u)}$, there is a set of k cells C_k and a set of queries $A_k \subseteq [m]$ such that

1. $|A_k| \geq \frac{m}{w^{2(k-1)} {t_u}^k}$
2. $C_k \subseteq Q_i$ for all $i \in A_k$

Main Technical Lemma: Let C be a set of cells in the data structure and $A \subseteq [m]$. If

1. $|A| \geq \sqrt{m}$
2. $|C| \leq \frac{\alpha \log m}{5w}$, and
3. For all $i \in A$, Pred(i) probes each cell in C , then

There is $j \in A$ and subset $A' \subseteq A$ with $|A'| \geq \frac{|A|}{w^2}$ such that for all $i \in A'$,

U_j and **Q_i** intersect *outside of C*.

Acknowledgements

- Joe Boninger ‘16
  - **Highest Honors** *(top 2% of Swarthmore students)*
  - Math/CS double major

- Owen Kephart ‘18
  - CS/Engineering double major
  - Honors candidate