Time-Space Trade-Offs for 2D Range Minimum Queries

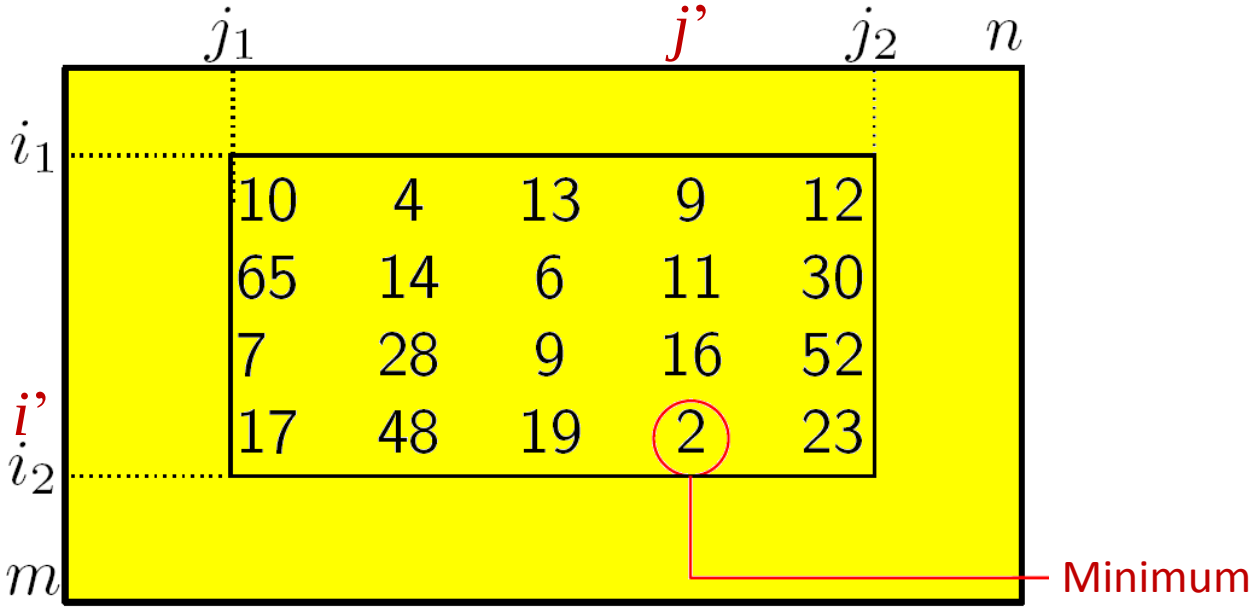
Gerth Stølting Brodal
Aarhus University

MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS

Join work with Pooya Davoodi and S. Srinivasa Rao

Dagstuhl Seminar on Data Structures, February 28 - March 5, 2010

The 2D Range Minimum Problem



Preprocess an m x n-matrix of size $N = n \cdot m$, $m \leq n$, to efficiently support range minimum queries

$$\text{RMQ}([i_1, i_2]\text{x}[j_1, j_2]) = (i', j')$$

$$A_{i',j'} = \min\{ A_{i'',j''} \mid (i'', j'') \in [i_1,i_2] \text{ x } [j_1,j_2] \}, \quad (i', j') \in [i_1,i_2]\text{x}[j_1,j_2]$$

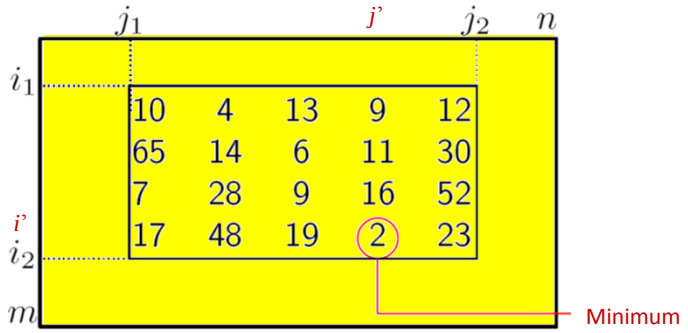
Models

Encoding model

- Queries can access **data structure** but not input matrix

Indexing model

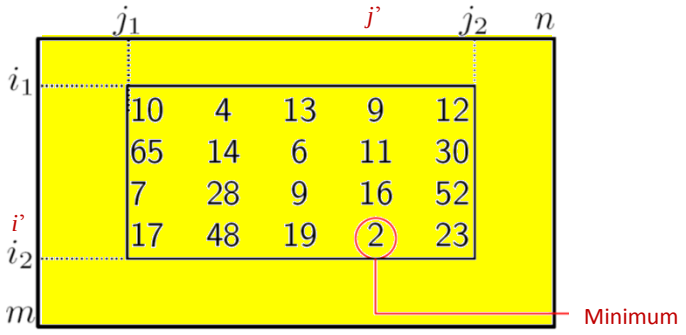
- Queries can access **data structure** and read **input matrix**



Some Trivial Examples...

Solution	Additional space (bits)	Query time	Model
No data structure	0	$O(N)$	Indexing
Tabulate answers	$O(N^2 \log N)$	$O(1)$	Encoding
Store permutation	$O(N \log N)$	$O(N)$	Encoding

....



Results

1D Range Minimum Queries

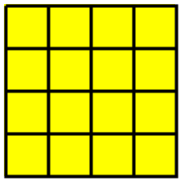
Indexing

- **Upper Bound** (Fischer and Heun (2007))
  - Time $= O(1)$
  - Space $= 2n + o(n) + |A|$ bits
- **NEW** **Lower Bound** (matching upper bound)
  - Time $= \Omega(c)$
  - Space $= O(n/c) + |A|$ bits

Encoding

- **Upper Bound** Fischer (Latin 2010)
  - Time $= O(1)$
  - Space $= 2n + o(n)$ bits
- **Lower Bound:** Space $= 2n - \Theta(\log n)$ bits

2D Range Minimum Queries



Indexing

NEW Upper Bound:
- Time $= O(1)$
- Space $= O(N) + |A|$ bits

- Time $= O(c \log^2 c)$
- Space $= O(N/c) + |A|$ bits

NEW Lower Bound:
- Time $= \Omega(c)$
- Space $= O(N/c) + |A|$ bits

Encoding

Upper Bound:
- Time $= O(1)$
- Space $= O(N \log n)$ bits

NEW Proof

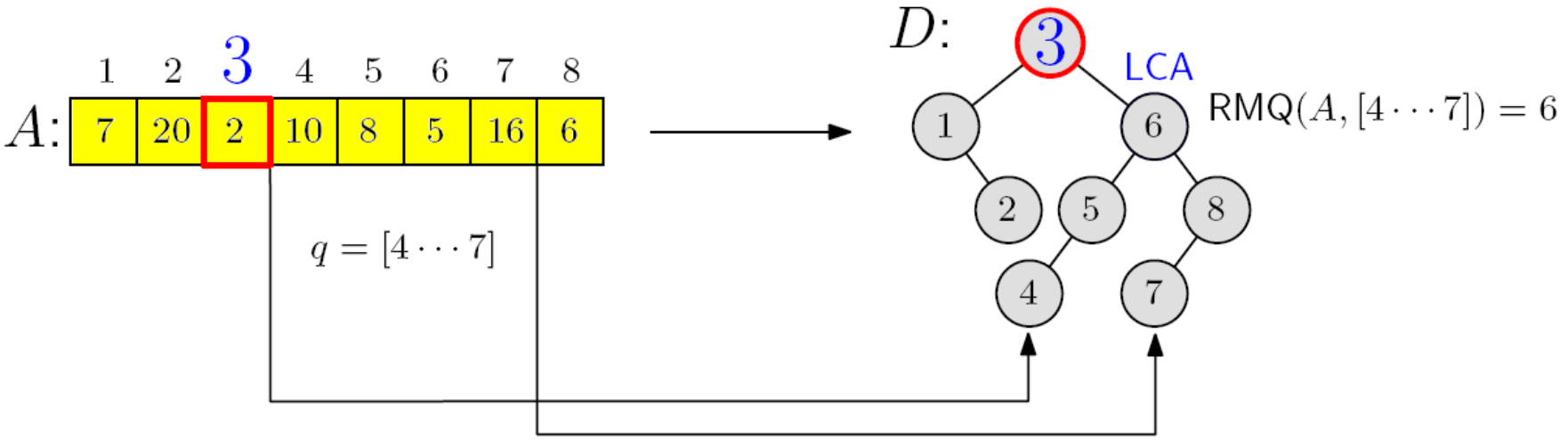
Lower Bound: Space $= \Omega(N \log m)$ bits

Demain et al. (2009)

1D

1D	Encoding model	Index model
Upper bound		
Lower bound		

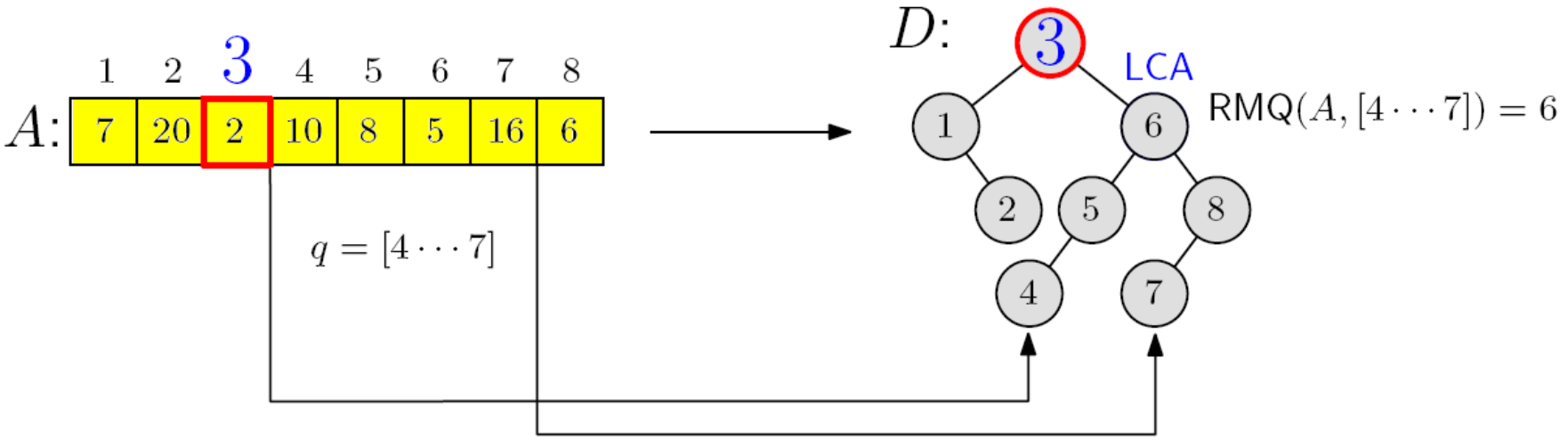
Lower Bound (1D, Encoding)



- For each input array consider the Cartesian tree
- Each binary tree is a possible Cartesian tree
- RMQ queries can reconstruct the Cartesian tree
- # Cartesian trees is $\binom{2n}{n}/(n+1)$
- # bits ≥ $\log\binom{2n}{n}/(n+1) = 2n - \Theta(\log n)$

1D	Encoding model	Index model
Upper bound		
Lower bound	●	

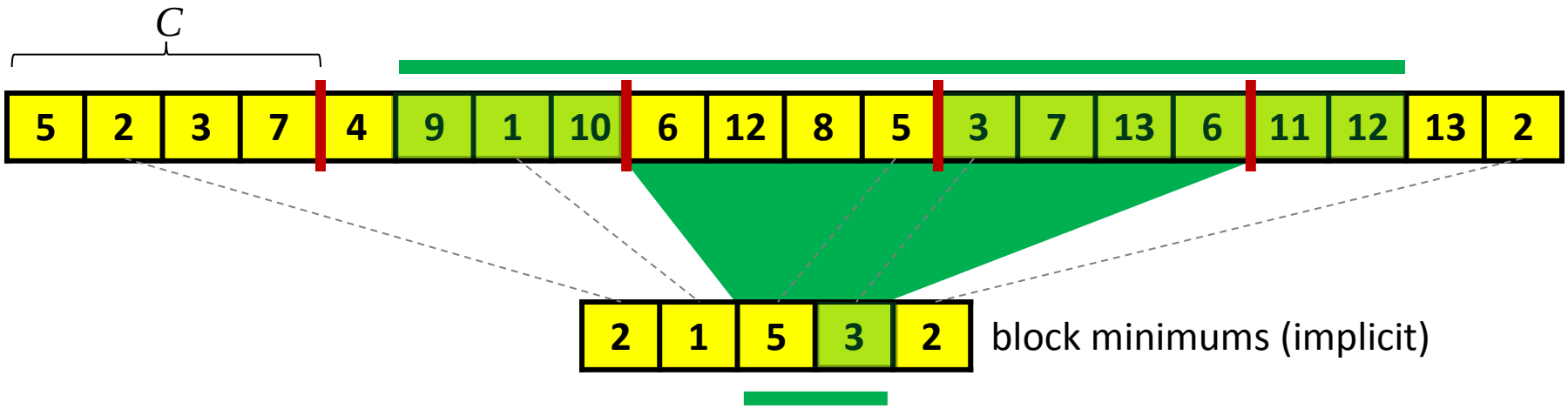
Upper Bound (1D, Encoding)



- For an input array consider the Cartesian tree
- Succint representation using $4n+o(n)$ bits and O(1) query time (Sadakane 2007)
- Improved to $2n+o(n)$ (Fischer 2010)

1D	Encoding model	Index model
Upper bound	●	
Lower bound	●	

Upper Bounds (1D, Indexing)

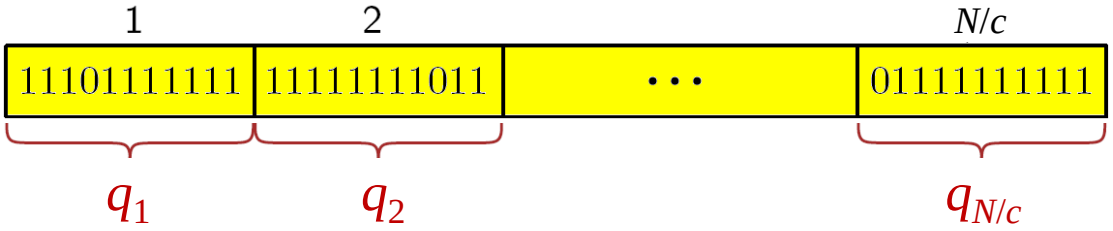


- Build encoding $O(N/c)$ bit structure for block minimums
- RMQ = query to encoding structure + $3c$ elements, i.e. query time $O(c)$

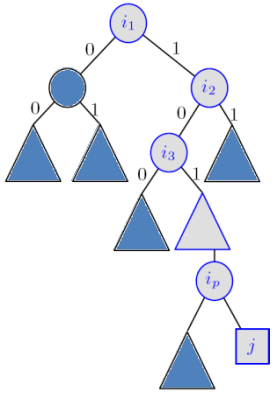
1D	Encoding model	Index model
Upper bound	●	●
Lower bound	●	

Lower Bounds (1D, Indexing)

Thm Space N/c bits implies $\Omega(c)$ query time



- Consider N/C queries for $c^{N/c}$ different {0,1} inputs with exactly one zero in each block
- $c^{N/c} \,/\, 2^{N/c}$ inputs share some data structure
- Every query is a decision tree of height $\leq d$



1D	Encoding model	Index model
Upper bound	●	●
Lower bound	●	●

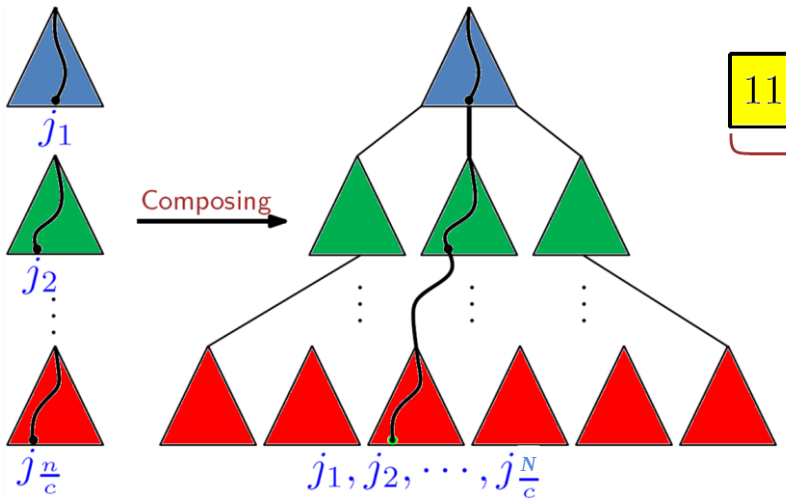
Lower Bounds (1D, Indexing) *cont.*

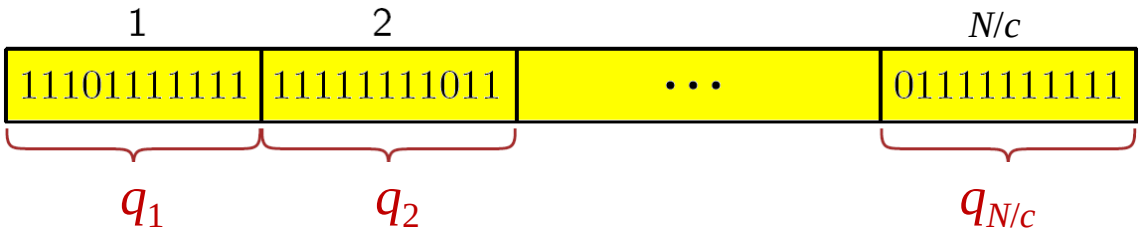
- Combine queries to decision tree identifying input
- Prune non-reachable branches

➡ # zeroes on any path ≤ N/c

➡ $$\frac{c^{N/c}}{2^{N/c}} \leq \#\text{inputs} = \#\text{leaves} \leq \binom{d \cdot N/c}{N/c}$$

➡ query time $d = \Omega(c)$





1D	Encoding model	Index model
Upper bound	●	●
Lower bound	●	●

2D

2D	Encoding model	Index model
Upper bound		
Lower bound		

Upper Bounds (2D, Indexing)

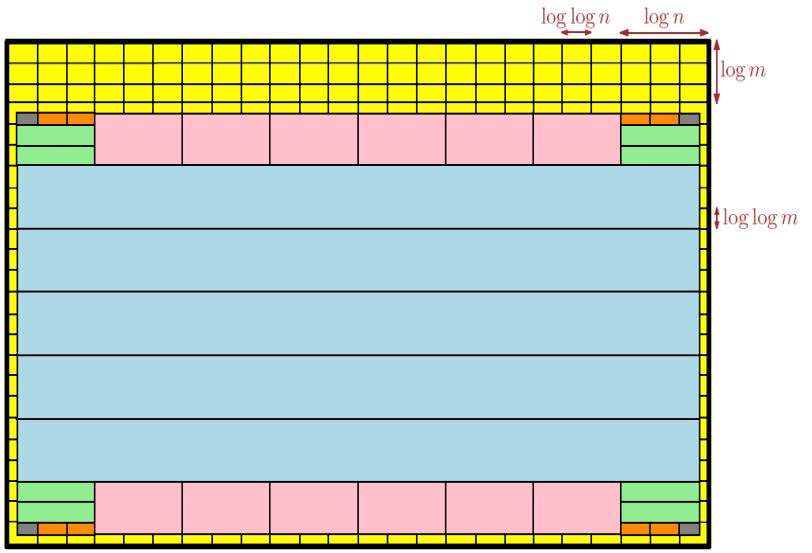
$O(1)$ time using $O(N)$ words

Atallah and Yuan (SODA 2010)

- Using two-levels of recursion, tabulating micro-blocks of size

  $\log\log m \times \log\log n$

  $O(1)$ time using $O(N)$ bits

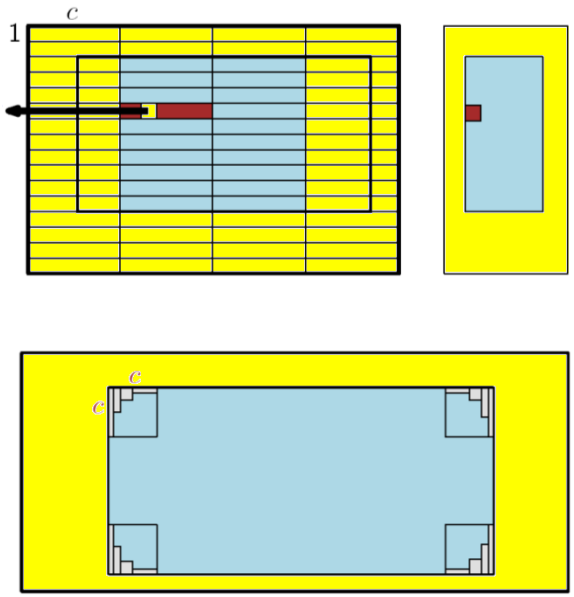


2D	Encoding model	Index model
Upper bound		●
Lower bound		

Upper Bounds (2D, Indexing) *cont.*

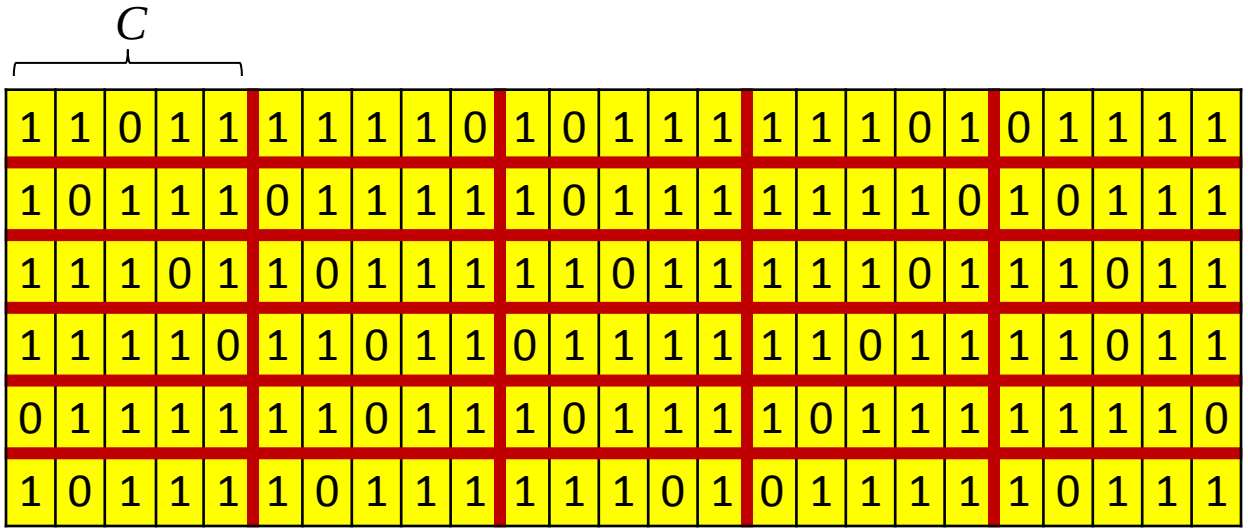
Thm $O(N/c \cdot \log c)$ bits and $O(c \log c)$ query time

- Build $\log c$ indexing structures for compressed matrices for block sizes 2^i x $c/2^i$, each using $O(N/c)$ bits and can locate $O(1)$ blocks with minimum key in $O(1)$ time
- Query: $O(1)$ blocks for each block size in time $O(c)$ + elements not covered by blocks in time $O(c \log c)$



2D	Encoding model	Index model
Upper bound		●
Lower bound		

Lower Bounds (2D, Indexing)



- As for 1D consider {0,1} matrices and partition the array into blocks of c elements each containing exactly one zero
- As for 1D an algorithm being able to identify the zero in each block using N/c bits will require time $\Omega(c)$

2D	Encoding model	Index model
Upper bound		●
Lower bound		●

Upper Bounds (2D, Encoding)

29	-14	10	15
2	7	0	13
-4	-5	-1	21
5	20	-17	32

input matrix

15	2	10	12
7	9	6	11
4	3	5	14
8	13	1	16

rank matrix

- Translate input matrix into rank matrix using $O(N \log N)$ bits
- Apply index structure to rank matrix using $O(N)$ bits achieving $O(1)$ query time

2D	Encoding model	Index model
Upper bound	●	●
Lower bound		●

NEW Proof

Lower Bound (2D, Encoding)

Demaine et al. 2009

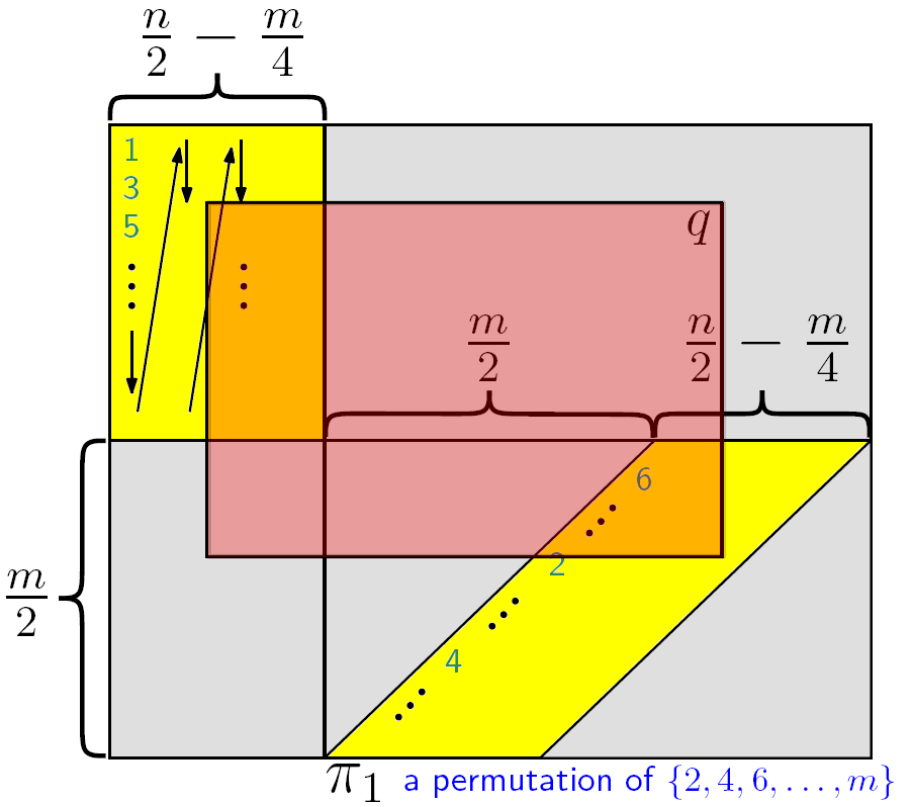
- Define a set of

$$\left(\frac{m}{2}!\right)^{\frac{n}{2}-\frac{m}{4}}$$

matrices where the RMQ answers differ among all matrices

- Bits required is at least

$$\log\left(\frac{m}{2}!\right)^{\frac{n}{2}-\frac{m}{4}} = \Omega(N \log m)$$



2D	Encoding model	Index model
Upper bound	●	●
Lower bound	●	●

Conclusion

1D Range Minimum Queries

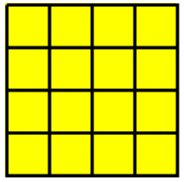
Indexing

- **Upper Bound** (Fischer and Heun (2007))
  - Time $= O(1)$
  - Space $= 2n + o(n) + |A|$ bits
- **NEW Lower Bound** (matching upper bound)
  - Time $= \Omega(c)$
  - Space $= O(n/c) + |A|$ bits

Encoding

- **Upper Bound** Fischer (Latin 2010)
  - Time $= O(1)$
  - Space $= 2n + o(n)$ bits
- **Lower Bound:** Space $= 2n - \Theta(\log n)$ bits

2D Range Minimum Queries



Indexing

NEW Upper Bound:
- Time $= O(1)$
- Space $= O(N) + |A|$ bits

- Time $= O(c \log^2 c)$
- Space $= O(N/c) + |A|$ bits

?

NEW Lower Bound:
- Time $= \Omega(c)$
- Space $= O(N/c) + |A|$ bits

Encoding

Upper Bound:
- Time $= O(1)$
- Space $= O(N \log n)$ bits

?

NEW Proof

Lower Bound: Space $= \Omega(N \log m)$ bits

Demain et al. (2009)

Tau · Jërë-jëf · Tashakkur · S.aHHa · Sag olun
Giihtu · Djakujo · Dâkujem vám · Thank you
Tesekkür ederim · To-siä · Merci · Tashakur
Taing · Dankon · Efharistó · Shukriya · Kiitos
Dhanyabad · Rakhmat · Trugarez · Asante
Köszönöm · Blagodarya · Dziekuje · Eskerrik asko
Grazie · Tak · Bayarlaa · Miigwech · Dank u
Spasibo · Dêkuji vám · Ngiyabonga · Dziakuj
Obrigado · Gracias · A dank aych · Salamat
Takk · Arigatou · Tack · Tänan · Aciu
Korp kun kah · Multumesk · Terima kasih · Danke
Rahmat · Gratias · Mahalo · Dhanyavaad
Paldies · Faleminderit · Diolch · Hvala
Kam-sa-ham-ni-da · Xìe xìe · Mèrcie · Dankie

Thank You