Algorithm Engineering, September 2013

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

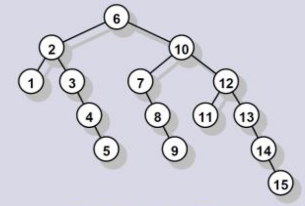
Gerth Stølting Brodal
Aarhus University
MADALGO – CENTER FOR MASSIVE DATA ALGORITHMICS

Join work with Pooya Davoodi and S. Srinivasa Rao

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

Data Structures, February-March 2010

Skewed Binary Search Trees



Gerth Stølting Brodal
University of Aarhus

Joint work with Gabriel Moruz

Dagstuhl seminar on Data Structures, February 26-March 3, 2006. 1

Data Structures, February-March 2006

On the Adaptiveness of Quicksort

Gabriel Moruz
BRICS
University of Aarhus

Joint work with Gerth Stølting Brodal and Rolf Fagerberg

Schloss Dagstuhl, Germany, July 22, 2004

Cache-Oblivious Data Structures and Algorithms for Undirected BFS and SSSP

Rolf Fagerberg
University of Southern Denmark - Odense

with G.S. Brodal, U. Meyer, N. Zeh

The Cost of Cache-Oblivious Searching

Alex López-Ortiz

University of Waterloo

Joint work with Bender, Brodal, Fagerberg, Ge, He, Hu, and Iacono
Presented at FOCS'03

Dagstuhl Seminar on Cache-Oblivious and Cache-Aware Algorithms, July 22, 2004

Cache-Oblivious and Cache-Aware Algorithms, July 2004

Optimal Finger Search Trees in the Pointer Machine

Gerth Stølting Brodal
BRICS
University of Aarhus

George Lagogiannis, Christos Makris
Athanasios Tsakalidis, Kostas Tsichlas
University of Patras

Dagstuhl seminar on "Data Structures", March 1, 2002

Data Structures, February-March 2002

Experimental Algorithmics, September 2000

Dynamic Planar Convex Hull

Gerth Stølting Brodal
BRICS
University of Aarhus
Denmark

Joint work with Riko Jacob, BRICS

Dagstuhl-Seminar on Data Structures
February 28 – March 3, 2000

BRICS

Data Structures, February-March 2000

Worst-Case Efficient External-Memory Priority Queues

Gerth Stølting Brodal
MPI Saarbrücken

Joint work with
Jyrki Katajainen
University of Copenhagen

Data Structures, March 1998

Constant Time Priority Queues

Priority Queues on Parallel Machines

Gerth S. Brodal

BRICS

Aarhus University

Data Structures, February-March 1996

Range Minimum Queries (Part II)

Gerth Stølting Brodal
Aarhus University

CENTER FOR MASSIVE DATA ALGORITHMICS

Join work with Andrej Brodnik and Pooya Davoodi (ESA 2013)

Dagstuhl Seminar on Data Structures and Advanced Models of Computation on Big Data, February 23-28, 2014

The Problem

Assumption

$m \leq n$

	1	2	3	4	⋯	n	
1	3	1	3	42	12	8	
2	7	14	6	11	15	37	i_1
3	13	99	21	27	44	16	i_2
⋮	23	28	5	13	4	47	
m	34	24	1	24	9	11	
		j_1		j_2			

RMQ(i_1, i_2, j_1, j_2) = (2,3)
= **position** of min

Cost
- Space (bits)
- Query time
- Preprocessing time

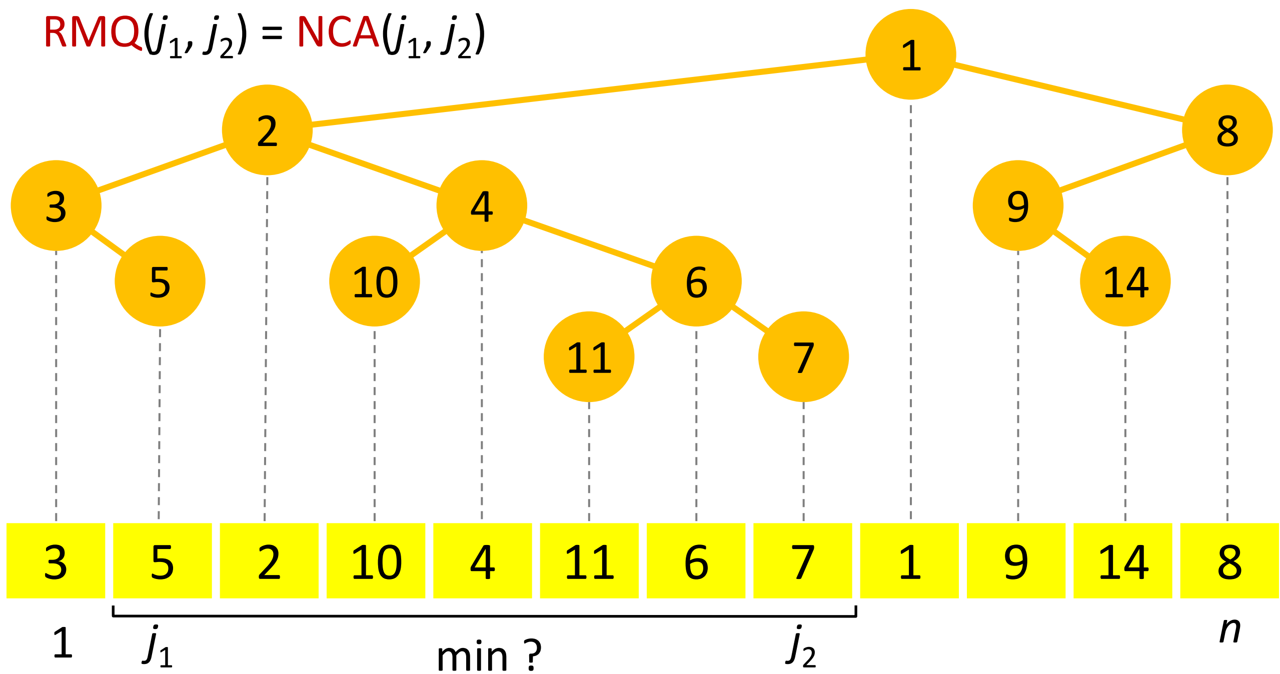
Models
- Indexing (input accessible)
- Encoding (input not accessible)

	1	2	3	4	⋯	n	
1	3	1	3	42	12	8	
2	7	14	6	11	15	37	i_1
3	13	99	21	27	44	16	i_2
⋮	23	28	5	13	4	47	
m	34	24	1	24	9	11	
		j_1		j_2			

Some (Trivial) Results

	Indexing Model (input accessible)	**Encoding Model** (input not accessable)
$m \leq n$	Preprocessing: Do nothing *!* Very fast preprocessing Very space efficient Queries $O(mn)$	Tabulate the answer to all ~ m^2n^2 possible queries Preprocessing & space $O(m^2n^2 \cdot \log n)$ bits Queries $O(1)$
		Store rank of all elements Preprocessing & space $O(mn \cdot \log n)$ bits Queries $O(mn)$

Encoding m = 1 (Cartesian tree)



To support RMQ queries we need...

- tree structure (111101001100110000100100)
- mapping between nodes and cells (inorder)

Some (Less Trivial) Results

	1	2	3	4	⋯	n	
1	3	1	3	42	12	8	
2	7	14	6	11	15	37	i_1
3	13	99	21	27	44	16	i_2
⋮	23	28	5	13	4	47	
m	34	24	1	24	9	11	
		j_1		j_2			

<table>
<tr><th></th><th>Indexing Model
(input accessible)</th><th>Encoding Model
(input not accessable)</th></tr>
<tr><td>$m = 1$
1D</td><td>$2n+o(n)$ bits, $O(1)$ time [FH07]
n/c bits $\Rightarrow$ $\Omega(c)$ time [BDS10]
n/c bits, $O(c)$ time [BDS10]</td><td>$\geq 2n$ - $O(\log n)$ bits
$2n+o(n)$ bits, $O(1)$ time [F10]</td></tr>
<tr><td>$1<m<n$</td><td rowspan="2">$O(mn \cdot \log n)$ bits, $O(1)$ time [AY10]
$O(mn)$ bits, $O(1)$ time [BDS10]
mn/c bits $\Rightarrow$ $\Omega(c)$ time [BDS10]
$O(c \cdot \log^2 c)$ time [BDS10]
$O(c \cdot \log c \cdot (\log\log c)^2)$ time [BDLRR12]</td><td>$\Omega(mn \cdot \log m)$ bits [BDS10]
$O(mn \cdot \log n)$ bits, $O(1)$ time [BDS10]
$O(mn \cdot \log m)$ bits, ? time [BBD13]</td></tr>
<tr><td>$m = n$
squared</td><td>$\Omega(mn \cdot \log n)$ bits [DLW09]
$O(mn \cdot \log n)$ bits, $O(1)$ time [AY10]</td></tr>
</table>

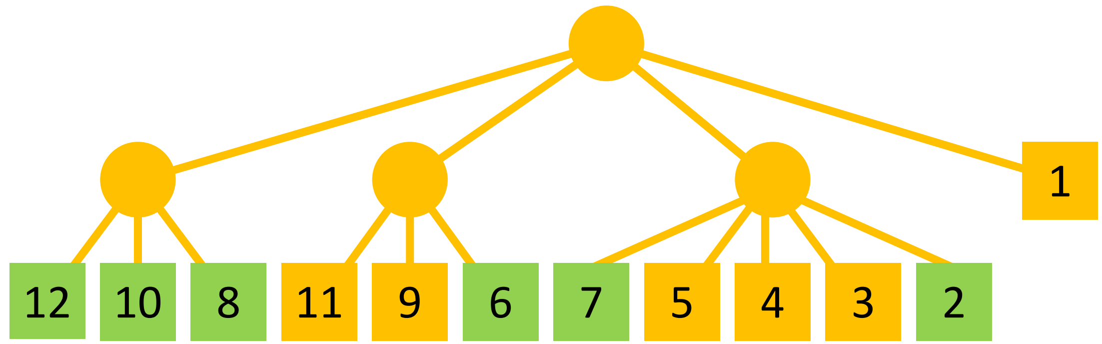
	1	2	3	4	…	n	
1	3	1	3	42	12	8	
2	7	14	6	11	15	37	i_1
3	13	99	21	27	44	16	i_2
⋮	23	28	5	13	4	47	
m	34	24	1	24	9	11	
		j_1		j_2			

New Results

1. $O(nm\cdot(\log m + \log\log n))$ bits
   - tree representation
   - component decomposition
2. $O(nm\cdot\log m\cdot\log^* n))$ bits
   - bootstrapping
3. $O(nm\cdot\log m)$ bits
   - relative positions of roots
   - refined component construction

Tree Representation

11	4	1	3
9	6	12	8
5	2	10	7

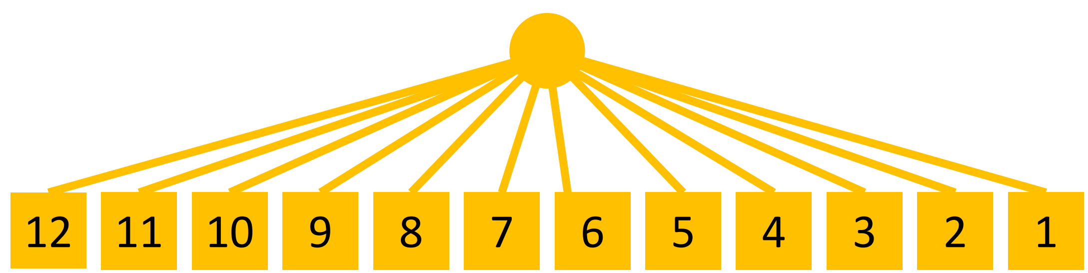


Requirements

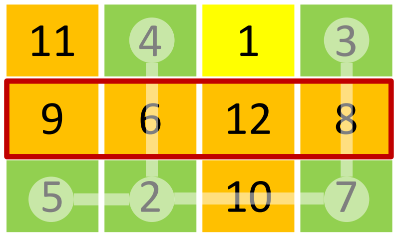
- Cells $\leftrightarrow$ leafs
- Query $\Rightarrow$ Answer = rightmost leaf

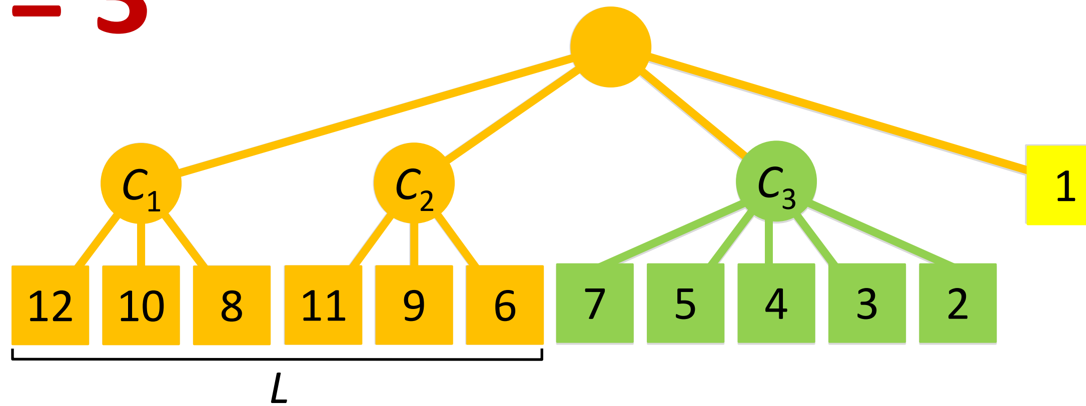
Trivial solution

- Sort leafs
- $\Omega(mn \cdot \log n)$ bits



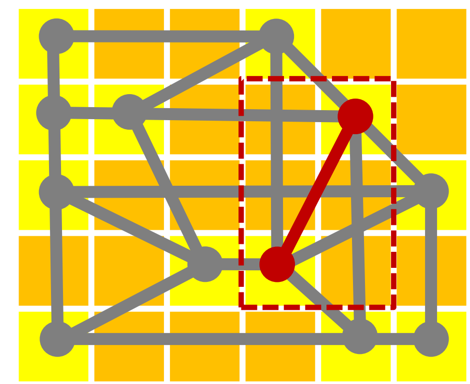
Components $\alpha = 3$





Construction

- Consider elements in decreasing order
- Find connected components with size ≥ α
- L-adjacency $\Rightarrow$ $|C_1| \leq 4\alpha\text{-}3$, $|C_i| \leq 2m\alpha$



L-adjacency

Representation

C_1 C_2 C_3 C_1 C_2 C_3

(((()))(((()))((())()())()).7.5.9.2.L1.L2.L1.L0.R2.L0.L2.L0.0.2.1.0.1.2.4.2.0.1.3

structure of T — roots — spanning tree edges — local leaf index

$O(mn + mn/\alpha \cdot \log \mathrm{n} + mn \cdot \log m + mn \cdot \log(m\alpha))$

Spanning tree structures | Component root positions | Spanning tree edges | Local leaf ranks in components

$\alpha = \log n \;\Rightarrow\; O(mn \cdot (\log m + \log\log n))$

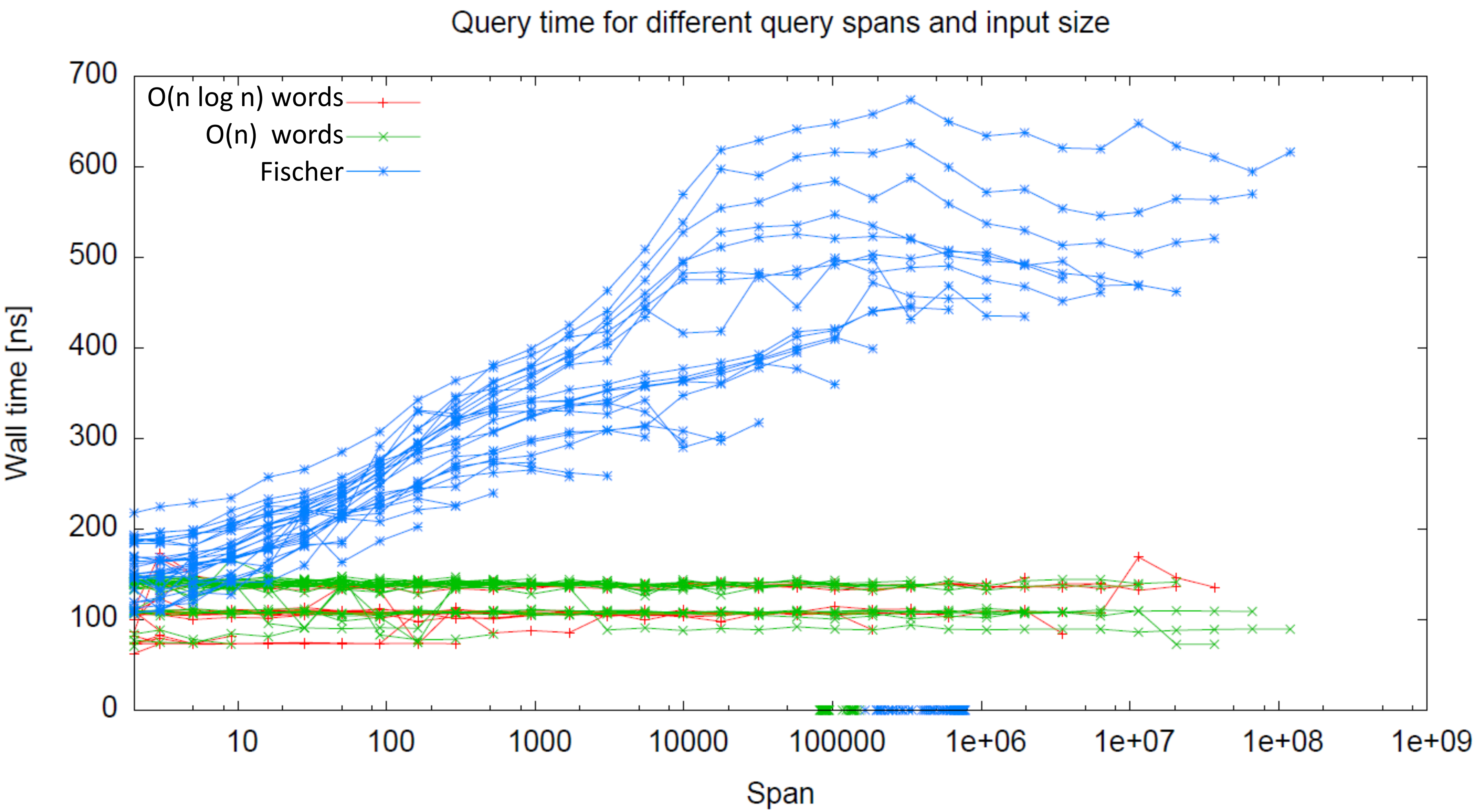
	1	2	3	4	…	n	
1	3	1	3	42	12	8	
2	7	14	6	11	15	37	i_1
3	13	99	21	27	44	16	i_2
⋮	23	28	5	13	4	47	
m	34	24	1	24	9	11	
		j_1		j_2			

Results

	Indexing Model (input accessible)	**Encoding Model** (input not accessable)
$m = 1$ 1D	$2n+o(n)$ bits, $O(1)$ time [FH07] n/c bits $\Rightarrow \Omega(c)$ time [BDS10] n/c bits, $O(c)$ time [BDS10]	$\geq 2n - O(\log n)$ bits $2n+o(n)$ bits, $O(1)$ time [F10]
$1<m<n$	$O(mn \cdot \log n)$ bits, $O(1)$ time [AY10] $O(mn)$ bits, $O(1)$ time [BDS10] mn/c bits $\Rightarrow \Omega(c)$ time [BDS10] $O(c \cdot \log^2 c)$ time [BDS10] $O(c \cdot \log c \cdot (\log\log c)^2)$ time [BDLRR12]	$\Omega(mn \cdot \log m)$ bits [BDS10] $O(mn \cdot \log n)$ bits, $O(1)$ time [BDS10] $O(mn \cdot \log m)$ bits, ? time [BBD13]
$m = n$ squared		$\Omega(mn \cdot \log n)$ bits [DLW09] $O(mn \cdot \log n)$ bits, $O(1)$ time [AY10]

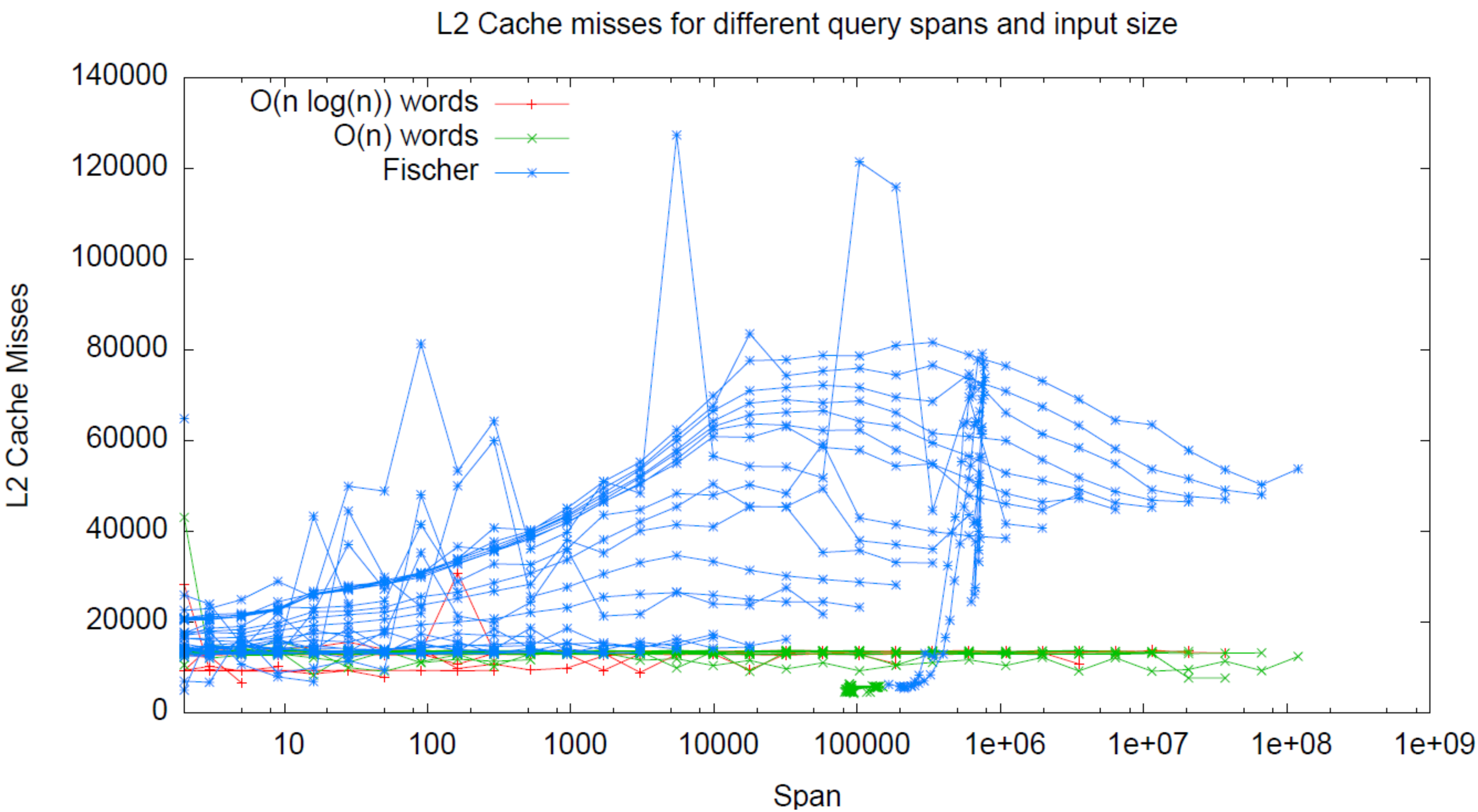
better upper or lower bound ?

1D Range Minimum Queues



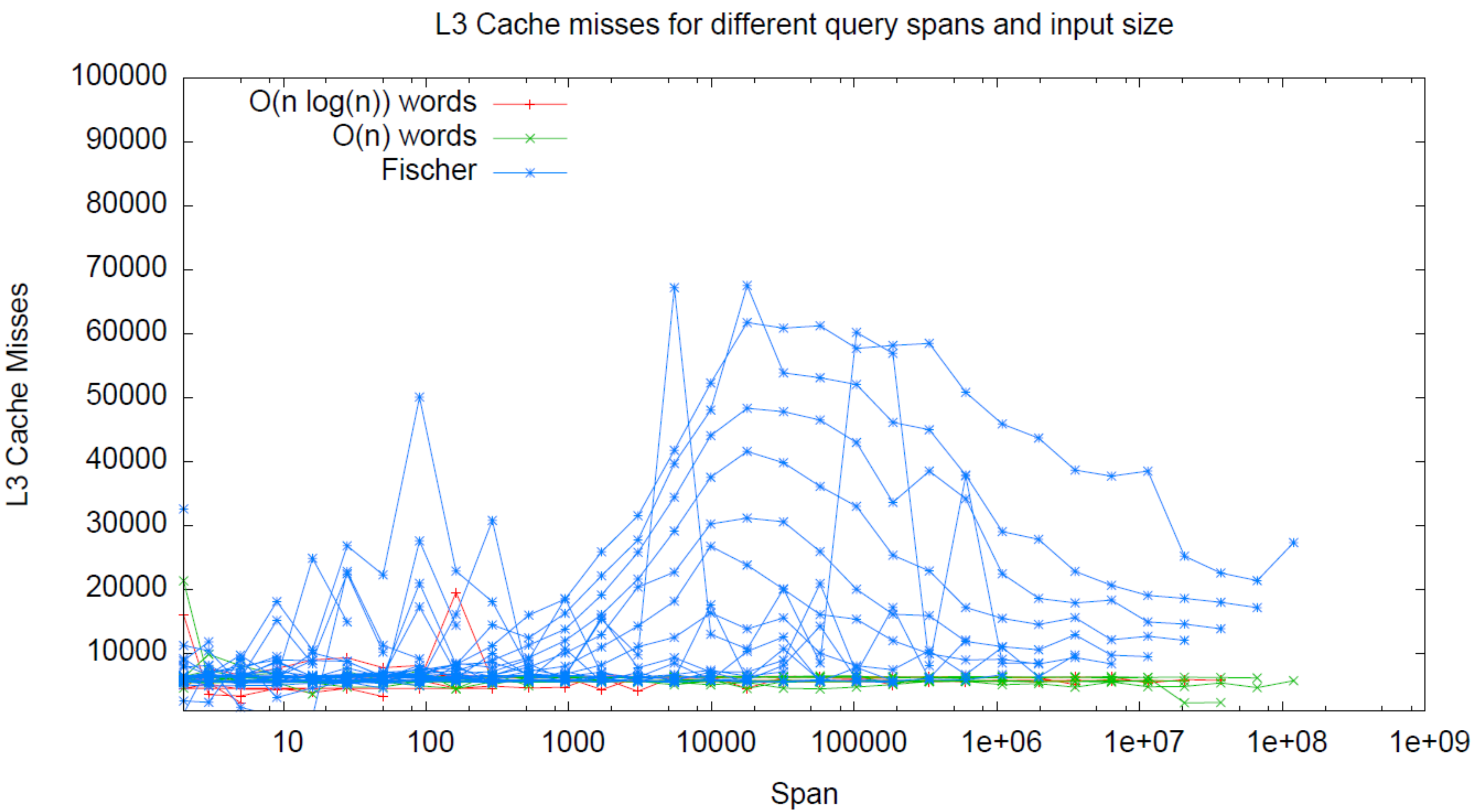
Kasper Nielsen, Mathieu Dehouck, Sarfraz Raza 2013

1D Range Minimum Queues



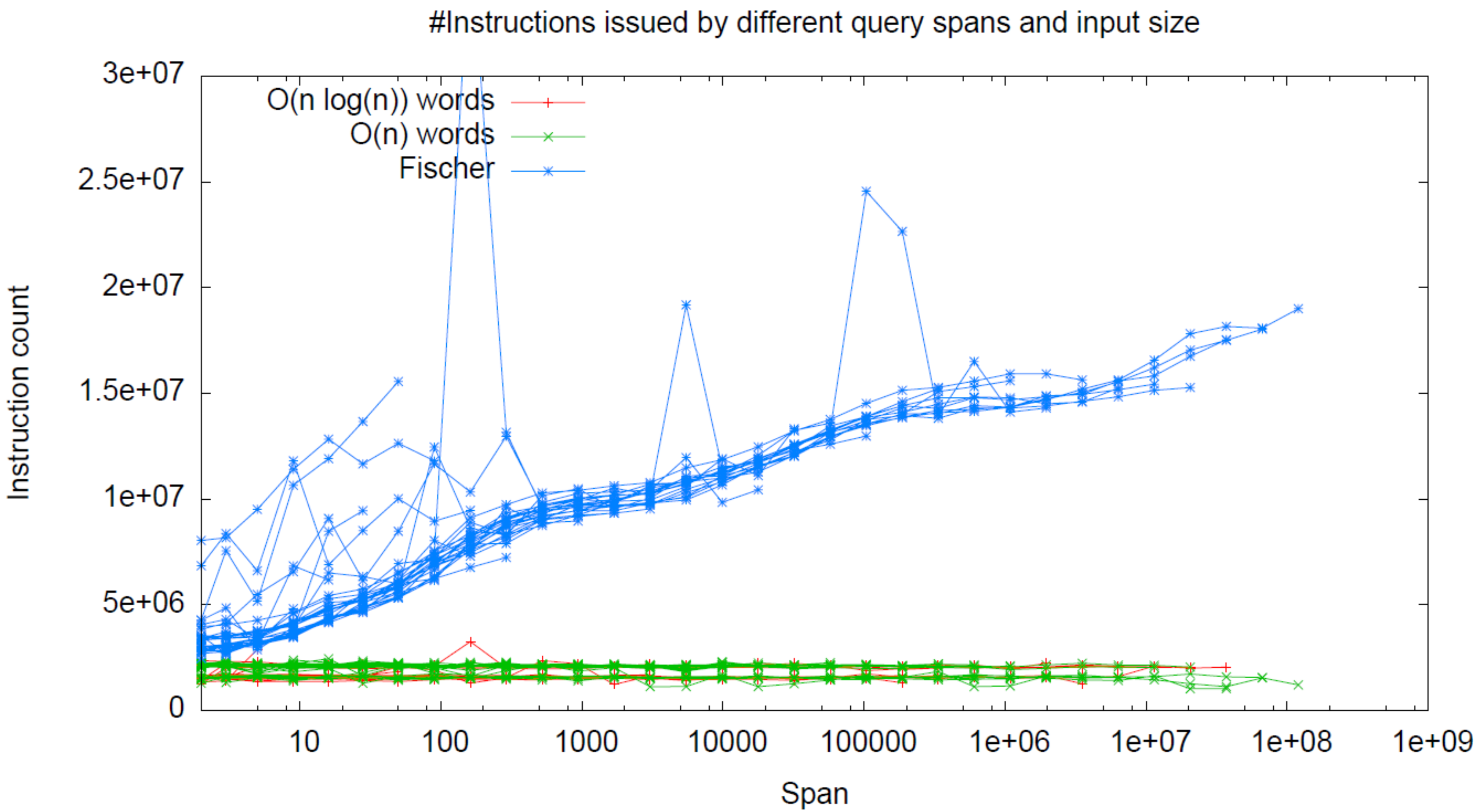
Kasper Nielsen, Mathieu Dehouck, Sarfraz Raza 2013

1D Range Minimum Queues



Kasper Nielsen, Mathieu Dehouck, Sarfraz Raza 2013

1D Range Minimum Queues



Kasper Nielsen, Mathieu Dehouck, Sarfraz Raza 2013