The Encoding Complexity of Two Dimensional Range Minimum Data Structures

Assumption
$1 \le m \le n$

	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)

Gerth Stølting Brodal
Aarhus

Andrej Brodnik
Primorska & Ljubljana

Pooya Davoodi
Polytechnic New York

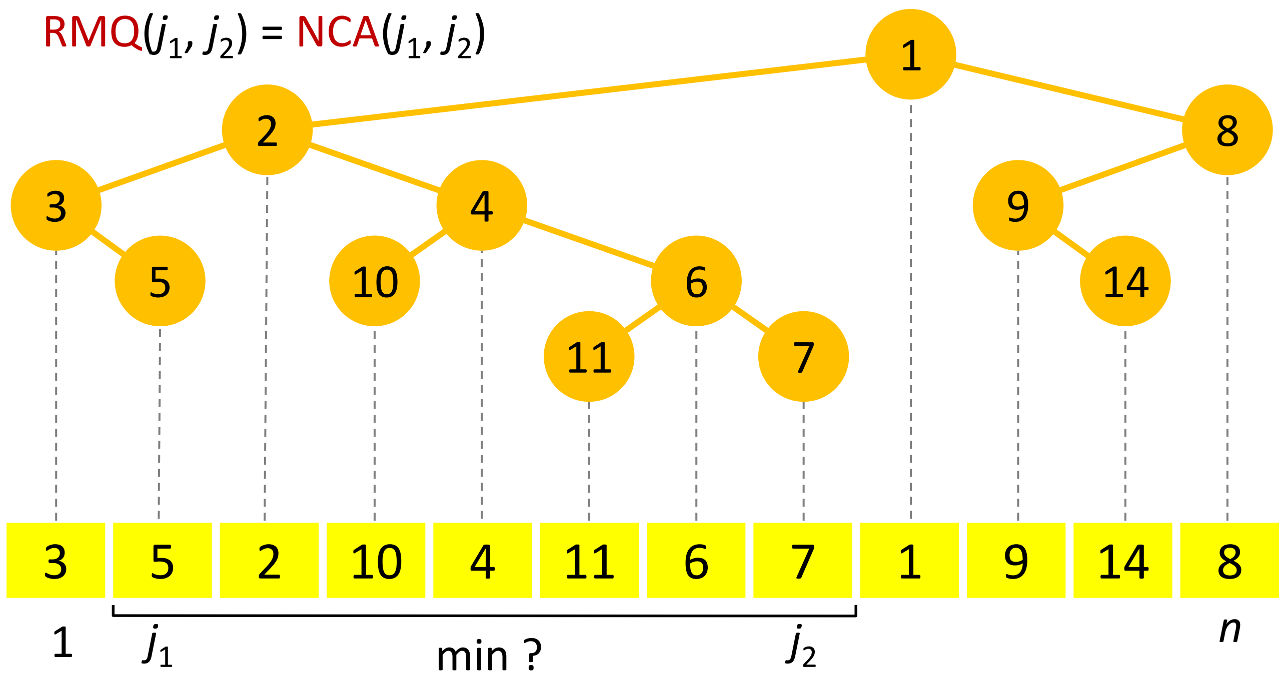
MADALGO, October 2, 2013 (Work presented at ESA 2013)

	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 $O(m^2n^2)$ Space $O(m^2n^2 \cdot \log n)$ bits Queries $O(1)$ Store rank of all elements Preprocessing $O(mn \cdot \log n)$ Space $O(mn \cdot \log n)$ bits Queries $O(mn)$

Encoding *m* = 1 (Cartesian tree)



To support RMQ queries we need...

- tree structure (1111010011001100000100100)
- mapping between nodes and cells (inorder)

	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 (Less Trivial) 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 [BDR10] n/c bits, $O(c)$ time [BDR10]	$\geq 2n$ - $O(\log n)$ 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 [BDR10] mn/c bits $\Rightarrow$ $\Omega(c)$ time [BDR10] $O(c \cdot \log^2 c)$ time [BDR10] $O(c \cdot \log c \cdot (\log\log c)^2)$ time [BDLRR12]	$\Omega(mn \cdot \log m)$ bits [BDR10] $O(mn \cdot \log n)$ bits, $O(1)$ time [BDR10] $O(mn \cdot \log m)$ bits, $O(mn)$ time [NEW]
$m = n$ squared		$\Omega(mn \cdot \log n)$ bits [DLW09] $O(mn \cdot \log n)$ bits, $O(1)$ time [AY10]

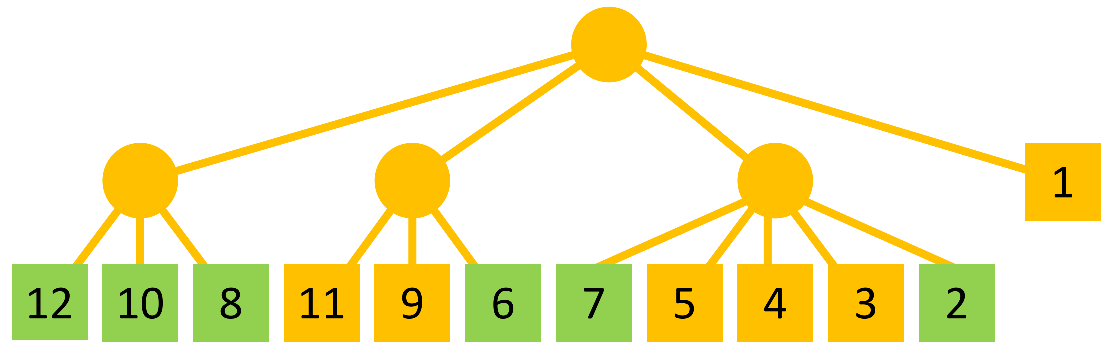
	1	2	3	4	⋯	n	
1	3	1	3	42	12	8	
2	7	14	6	11	15	37	i_1
3	13	99	21	27	44	16	i_2
⋮	23	28	5	13	4	47	
m	34	24	1	24	9	11	
		j_1		j_2			

New Results

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

Tree Representation

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

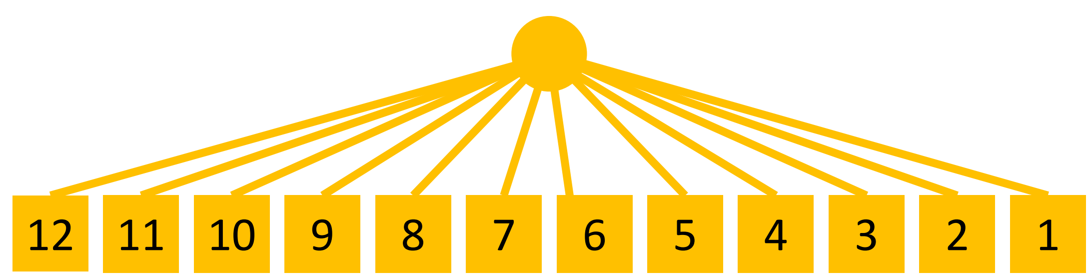


Requirements

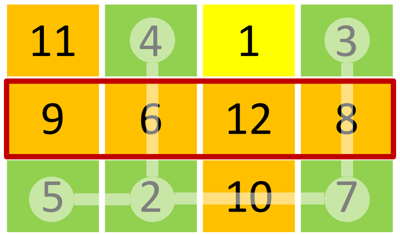
- (Index of) Cells ↔ leafs
- Query ⇒ Answer = rightmost leaf

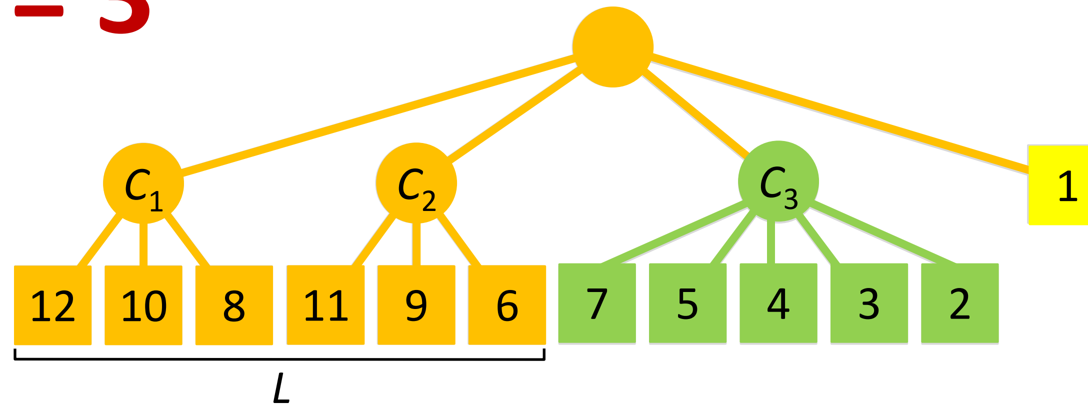
Trivial solution

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



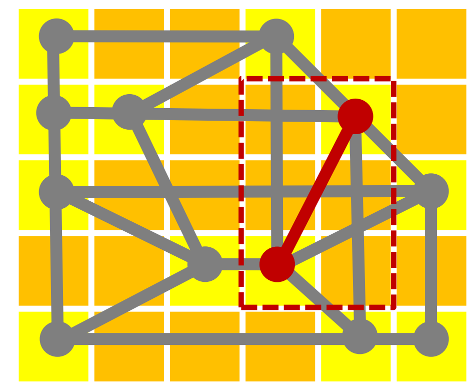
Components $\alpha = 3$





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

$$\underbrace{(((()))((()))((())()())())}_{\text{structure of } T}.\underbrace{7.5.9.2.}_{\text{roots}}\underbrace{\overbrace{\text{L1.L2.}}^{C_1}\overbrace{\text{L1.L0.}}^{C_2}\overbrace{\text{R2.L0.L2.L0.}}^{C_3}}_{\text{spanning tree edges}}\underbrace{\overbrace{0.2.1.}^{C_1}\overbrace{0.1.2.}^{C_2}\overbrace{4.2.0.1.3}^{C_3}}_{\text{local leaf index}}$$

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

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 [BDR10] n/c bits, $O(c)$ time [BDR10]	$\geq 2n - O(\log n)$ 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 [BDR10] mn/c bits $\Rightarrow \Omega(c)$ time [BDR10]	$\Omega(mn \cdot \log m)$ bits [BDR10] $O(mn \cdot \log n)$ bits, $O(1)$ time [BDR10] $O(mn \cdot \log m)$ bits, $O(mn)$ time [NEW]
$m = n$ squared	$O(c \cdot \log^2 c)$ time [BDR10] $O(c \cdot \log c \cdot (\log\log c)^2)$ time [BDLRR12]	$\Omega(mn \cdot \log n)$ bits [DLW09] $O(mn \cdot \log n)$ bits, $O(1)$ time [AY10]

better upper or
lower bound ?

Thank you