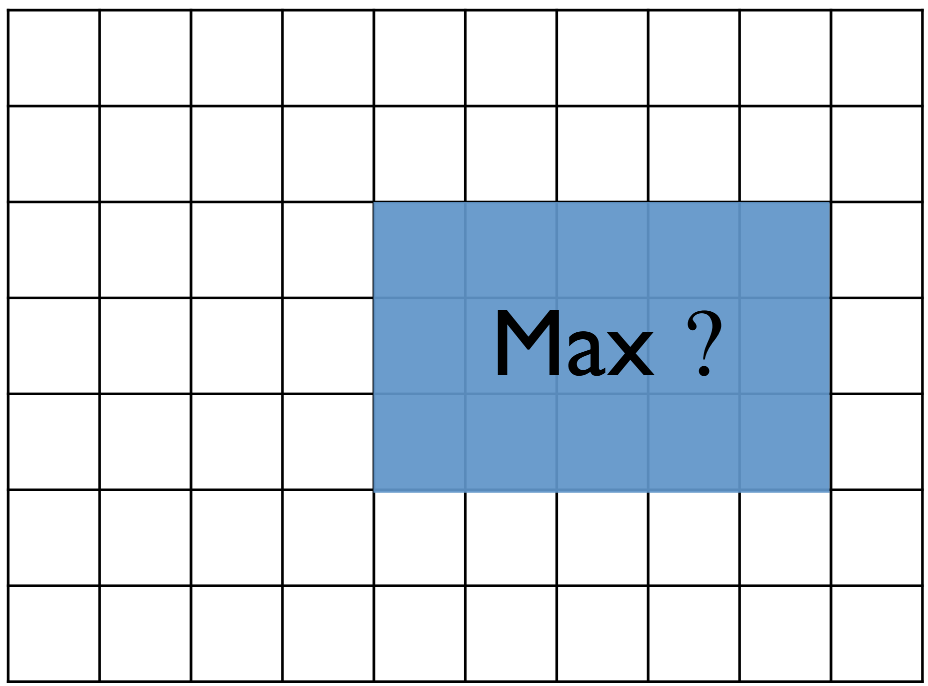
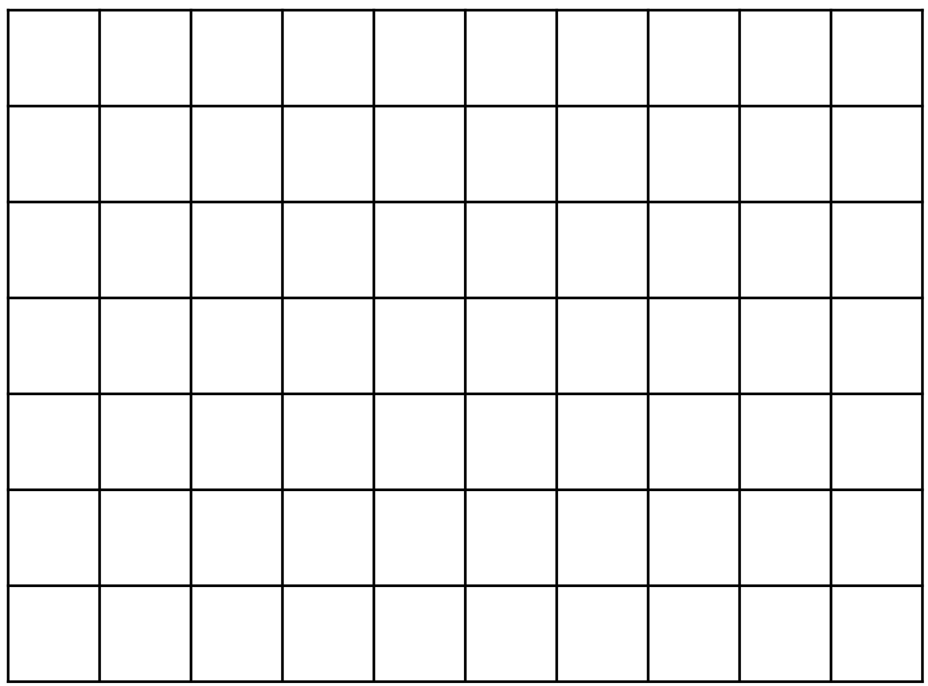
Submatrix Maximum Queries in Monge Matrices are Equivalent to Predecessor Search

Pawel Gawrychowski, Shay Mozes, Oren Weimann



Submatrix Maximum Queries in Monge Matrices are Equivalent to Predecessor Search

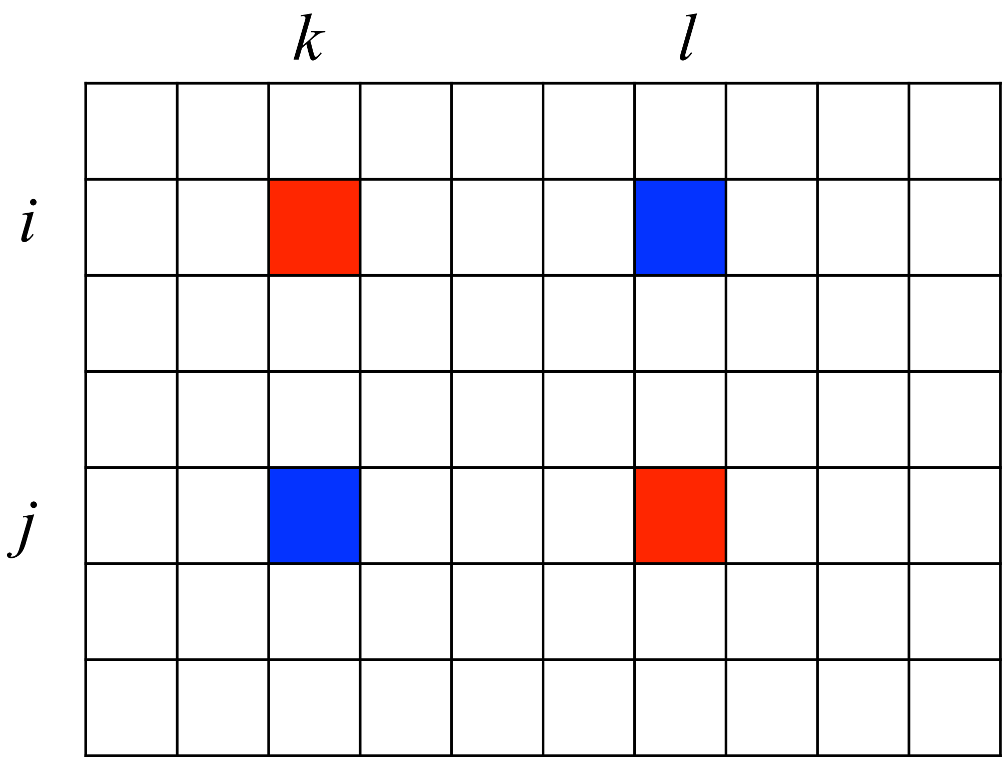
1746 - 1818



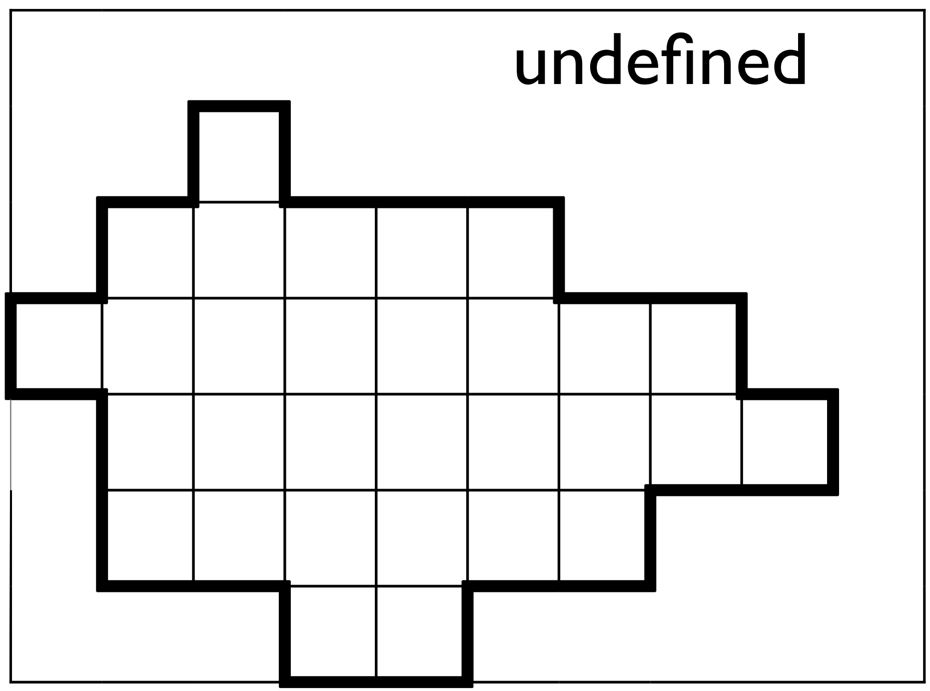
Submatrix Maximum Queries in Monge Matrices are Equivalent to Predecessor Search

$$M_{ik} + M_{jl} \geq M_{il} + M_{jk}$$

1746 - 1818

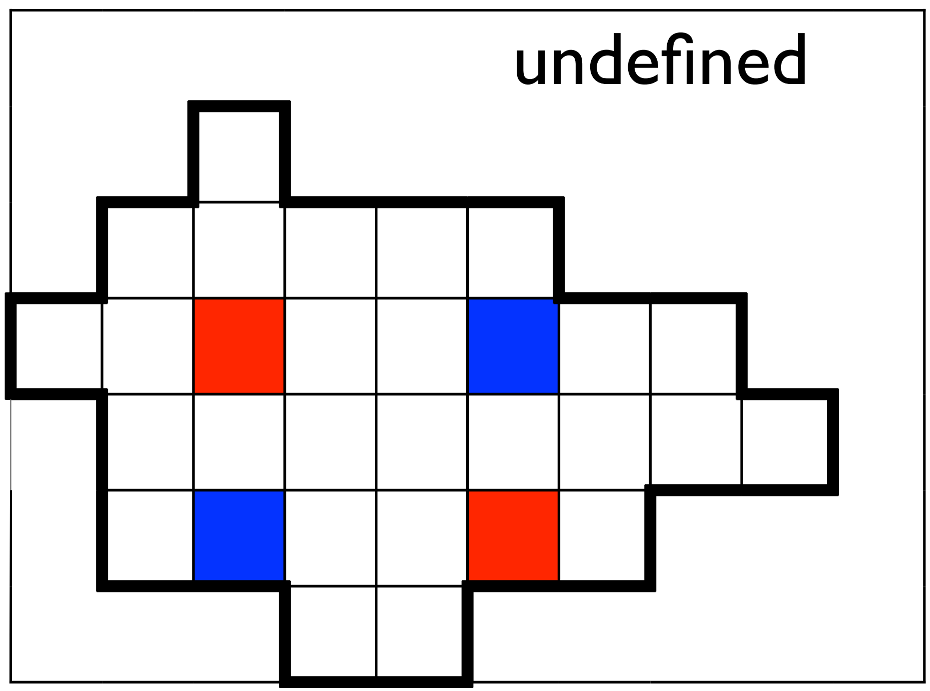


Submatrix Maximum Queries in Partial Monge Matrices are Equivalent to Predecessor Search



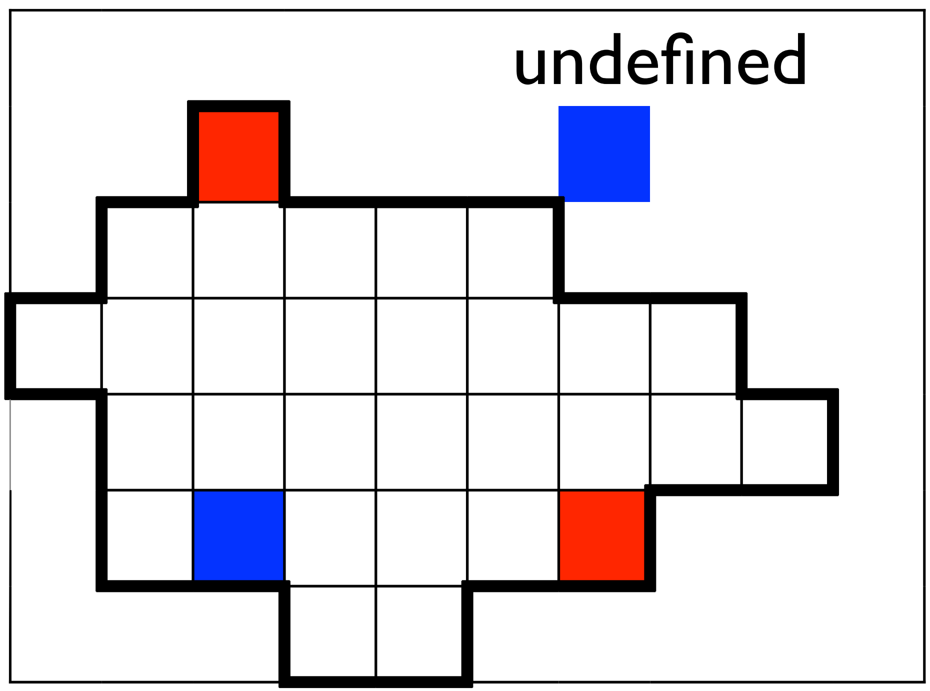
Submatrix Maximum Queries in Partial Monge Matrices are Equivalent to Predecessor Search

$$M_{ik} + M_{jl} \geq M_{il} + M_{jk}$$



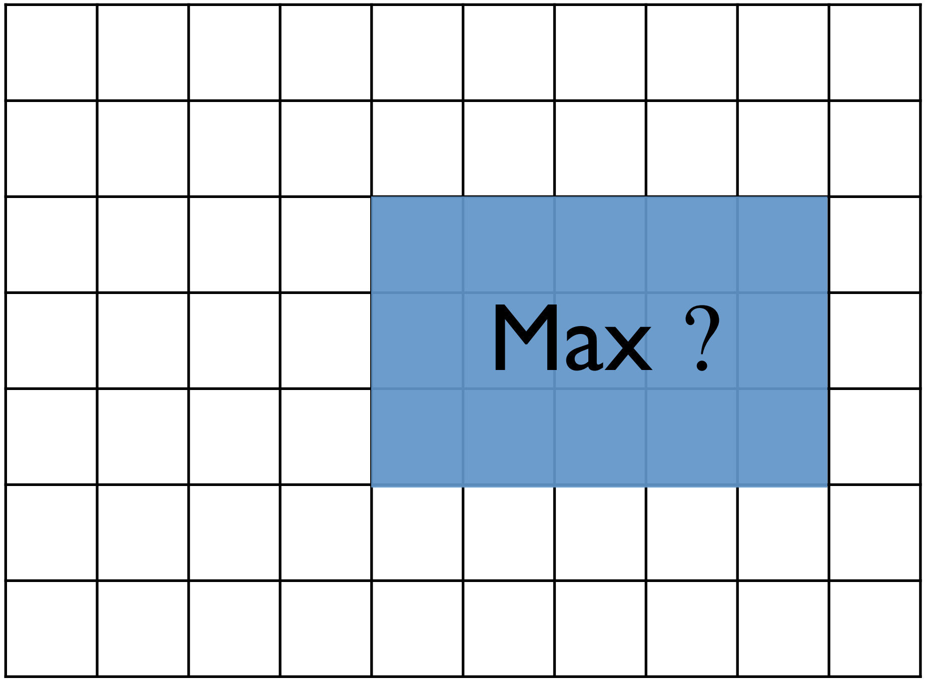
Submatrix Maximum Queries in Partial Monge Matrices are Equivalent to Predecessor Search

$$M_{ik} + M_{jl} \ ? \ M_{il} + M_{jk}$$



Submatrix Maximum Queries in Monge Matrices

are Equivalent to Predecessor Search

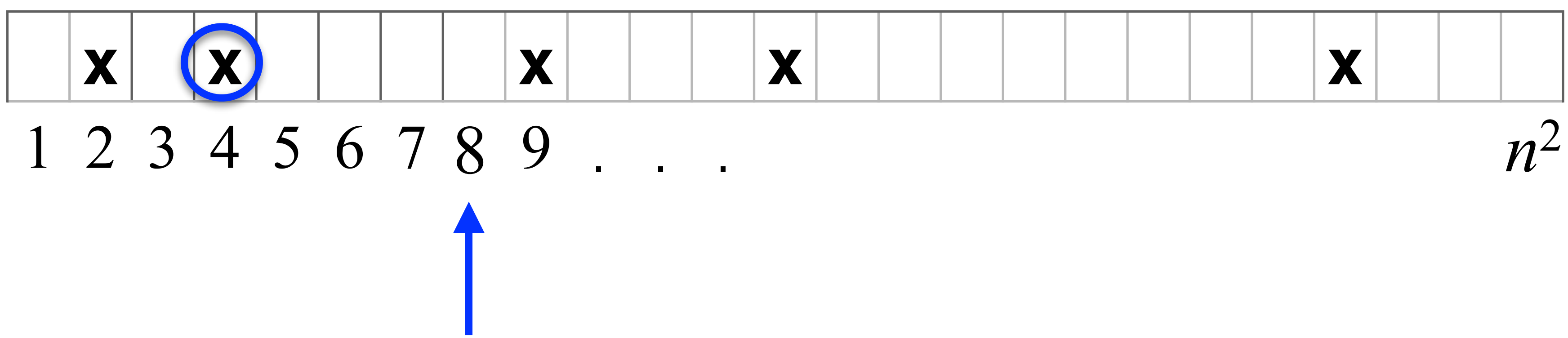


Submatrix Maximum Queries in Monge Matrices are Equivalent to Predecessor Search

Submatrix Maximum Queries in Monge Matrices are Equivalent to Predecessor Search

O(n *polylog* n) space requires Ω(loglog$n$) query-time
[Patraşcu, Thorup STOC'06]

Given n integers in $\{1,2,\ldots, n^2\}$



Submatrix Maximum Queries in Monge Matrices are Equivalent to Predecessor Search

[Kaplan, Mozes, Nussbaum, Sharir SODA'12]

[Gawrychowski, Mozes, W. ICALP'14]

[Gawrychowski, Mozes, W. ICALP'15]

Submatrix Maximum Queries in Monge Matrices are Equivalent to Predecessor Search

	[Kaplan, Mozes, Nussbaum, Sharir SODA'12]	[Gawrychowski, Mozes, W. ICALP'14]	[Gawrychowski, Mozes, W. ICALP'15]
n x n Monge:			
Space	$O(n \log n)$	$O(n)$	$O(n)$
Query	$O(\log^2 n)$	$O(\log n)$	$\Theta(\log\log n)$

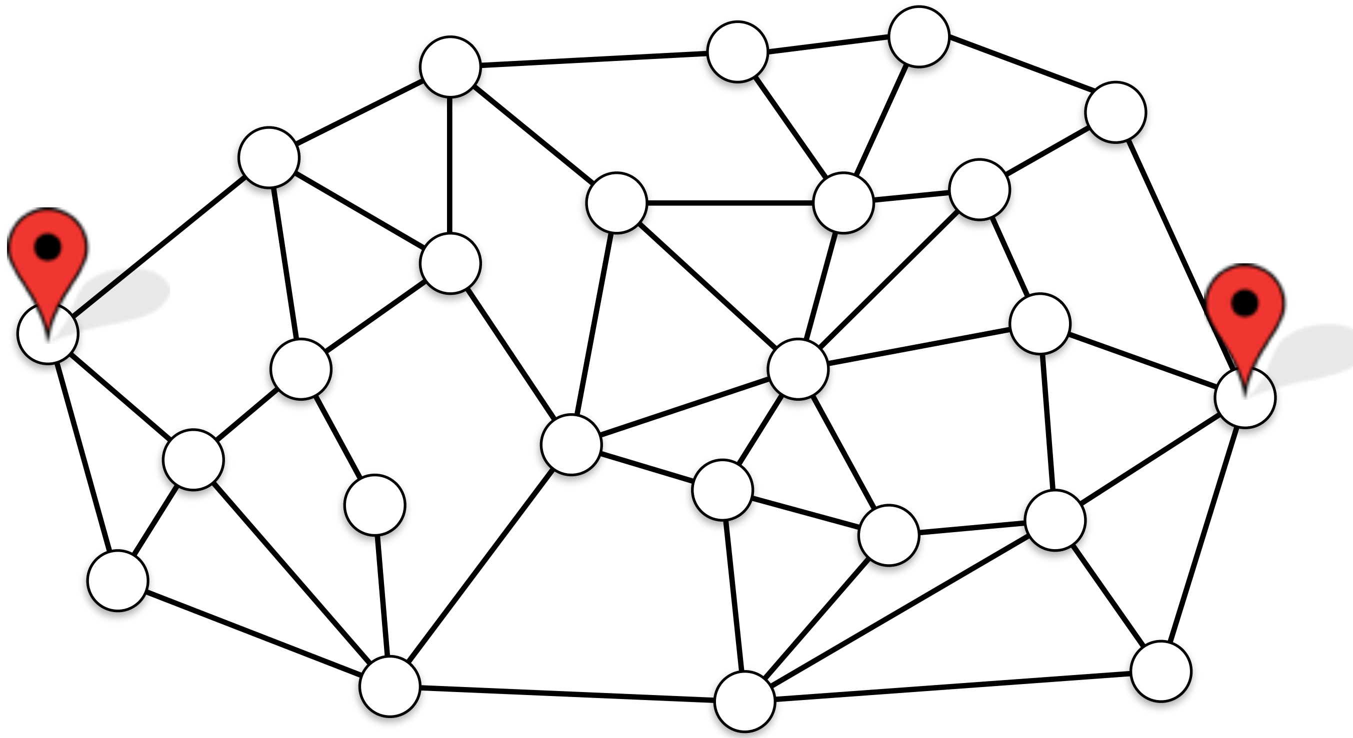
Submatrix Maximum Queries in Monge Matrices are Equivalent to Predecessor Search

	[Kaplan, Mozes, Nussbaum, Sharir SODA'12]	[Gawrychowski, Mozes, W. ICALP'14]	[Gawrychowski, Mozes, W. ICALP'15]
n x n Monge:			
Space	$O(n \log n)$	$O(n)$	$O(n)$
Query	$O(\log^2 n)$	$O(\log n)$	$\Theta(\log\log n)$
n x n partial:			
Space	$O(n \log n\, \alpha(n))$	$O(n)$	$O(n)$
Query	$O(\log^2 n)$	$O(\log n\, \alpha(n))$	$\Theta(\log\log n)$

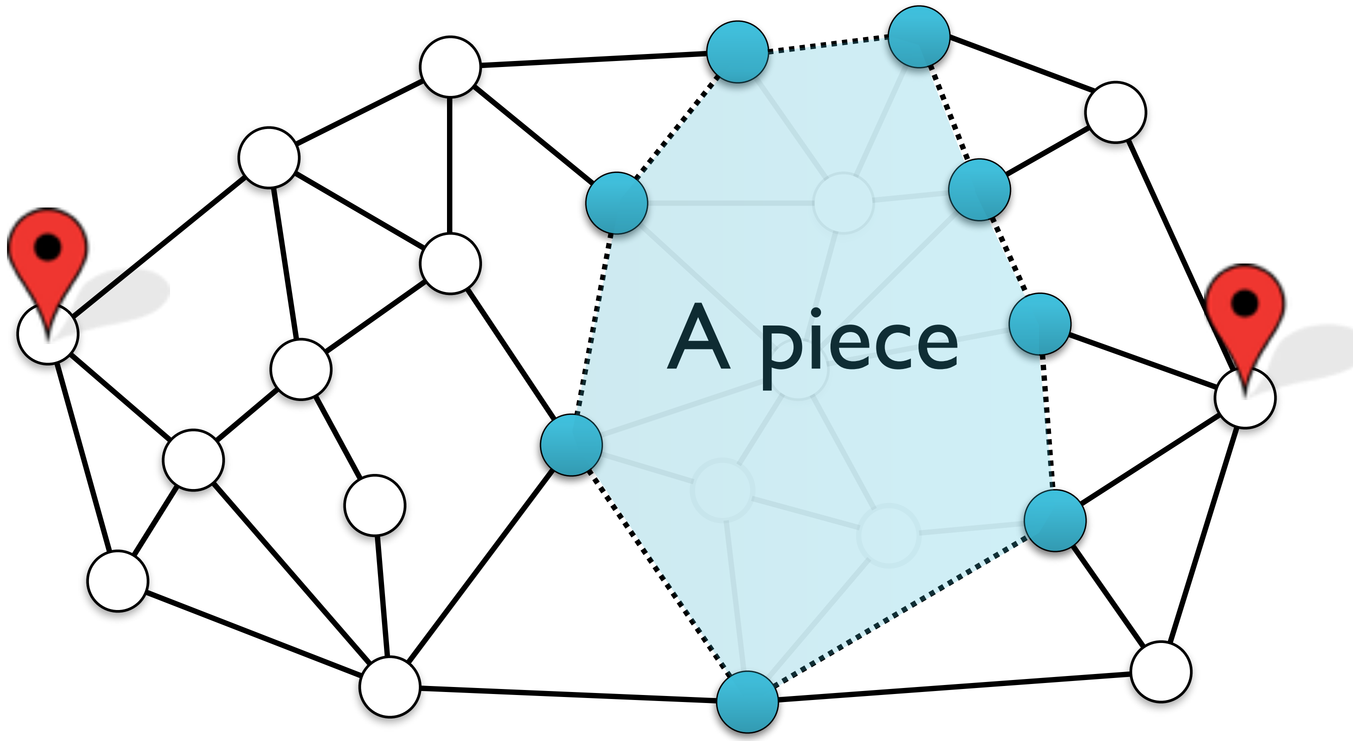
Applications

[Kaplan,Mozes,Nussbaum,Sharir SODA'12]

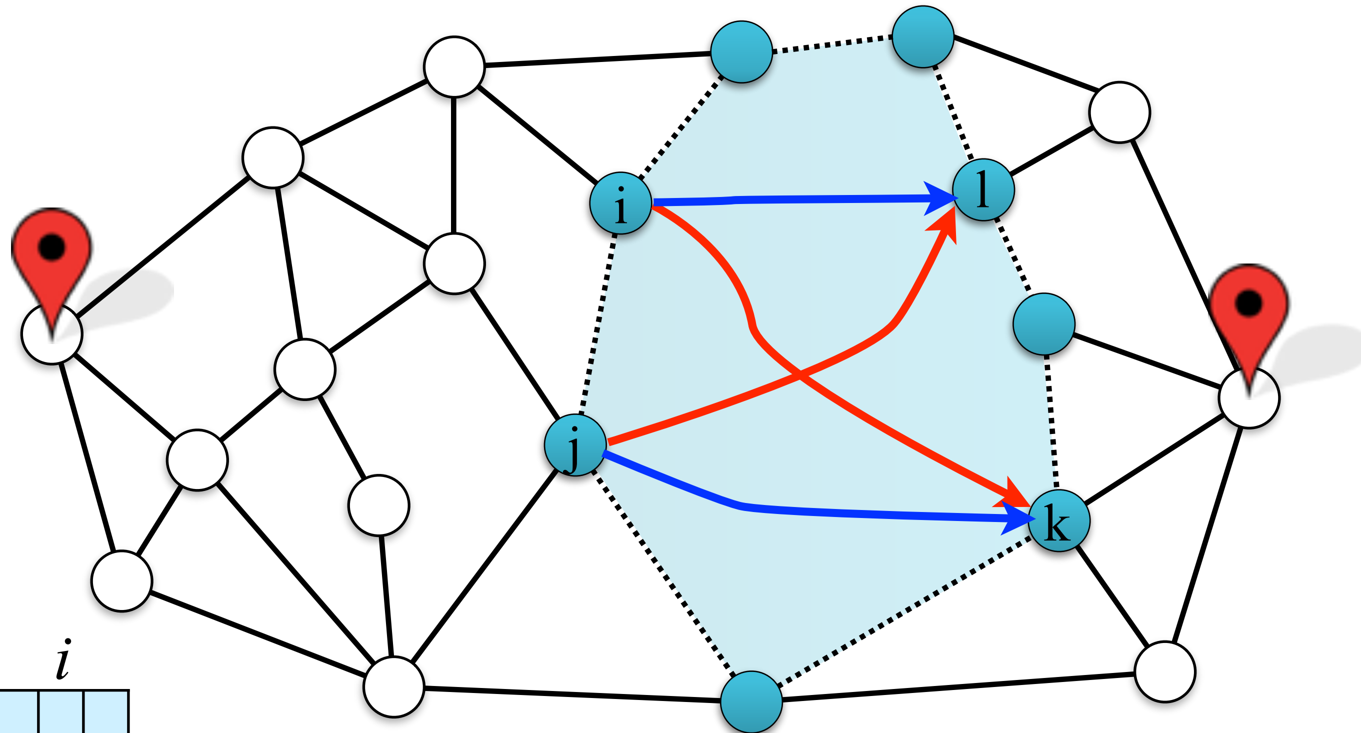
Application I
Shortest paths in planar graphs

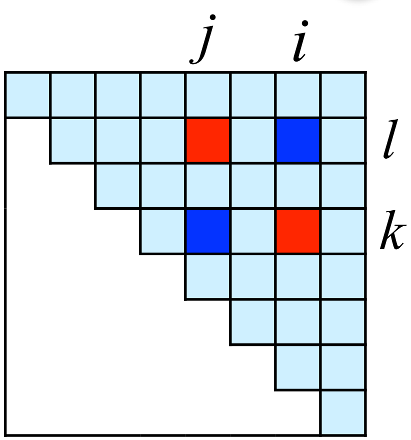


Application I
Shortest paths in planar graphs



Application I
Shortest paths in planar graphs

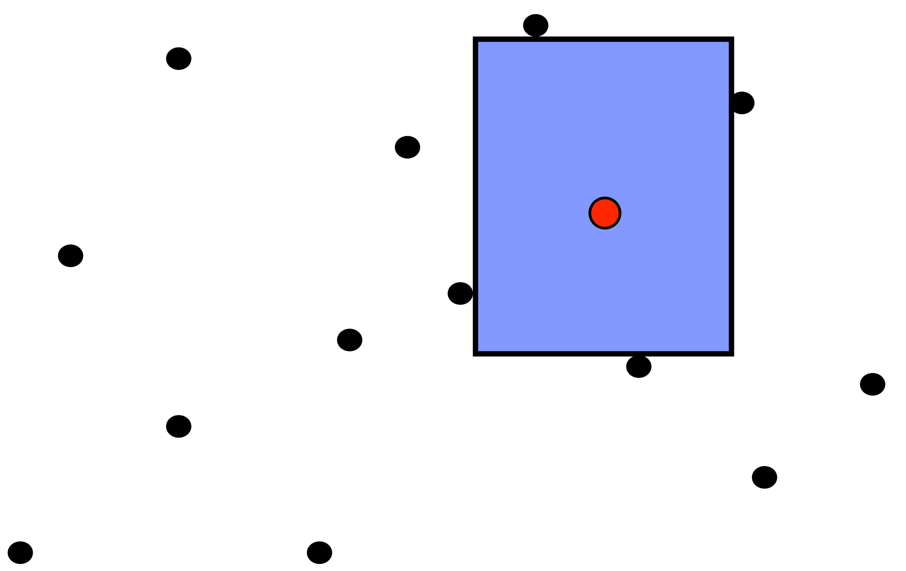




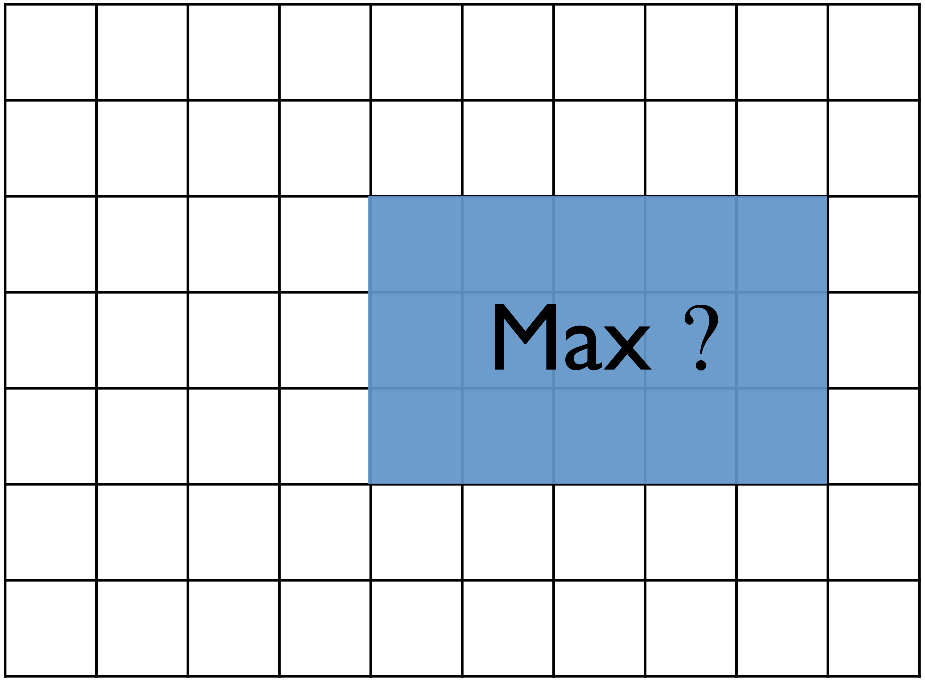
$$M_{ik} + M_{jl} \geq M_{il} + M_{jk}$$

Application II: Largest empty rectangle

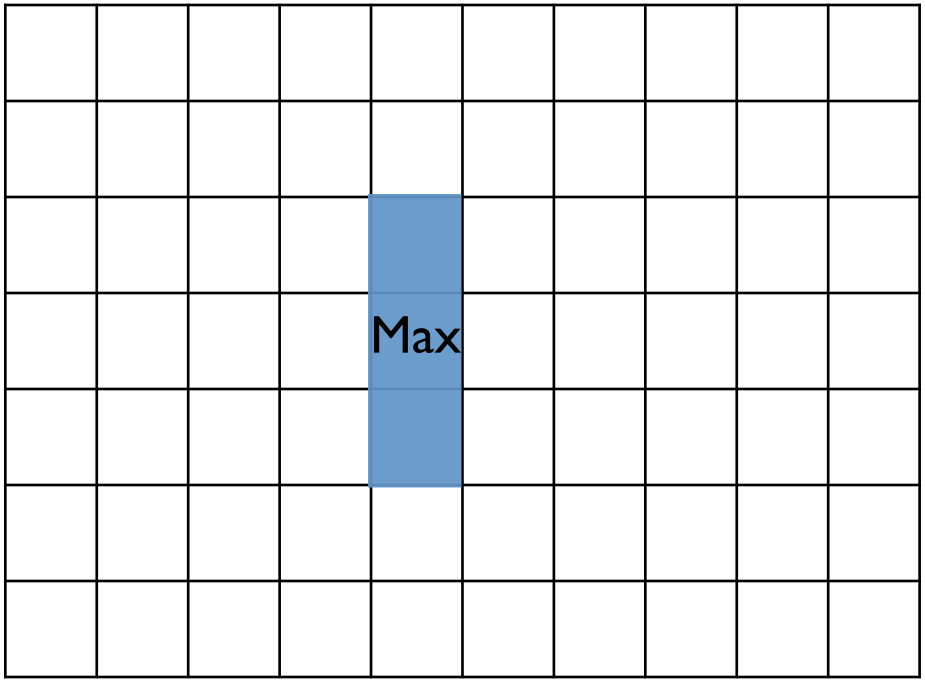
- Input: a set of *n* points
- Query: find largest empty rectangle containing a point



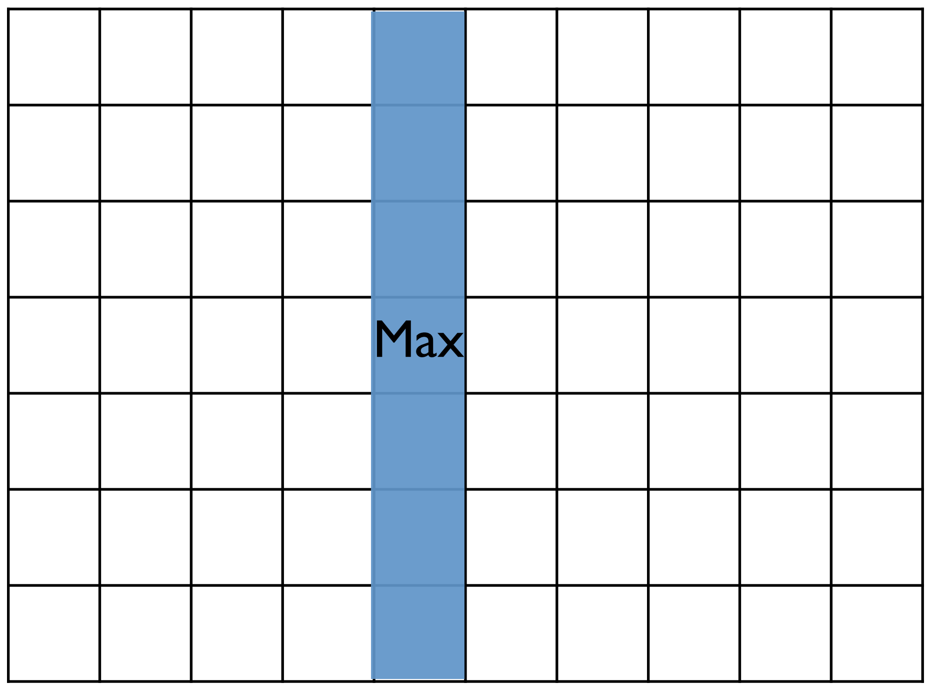
Submatrix Maximum Queries



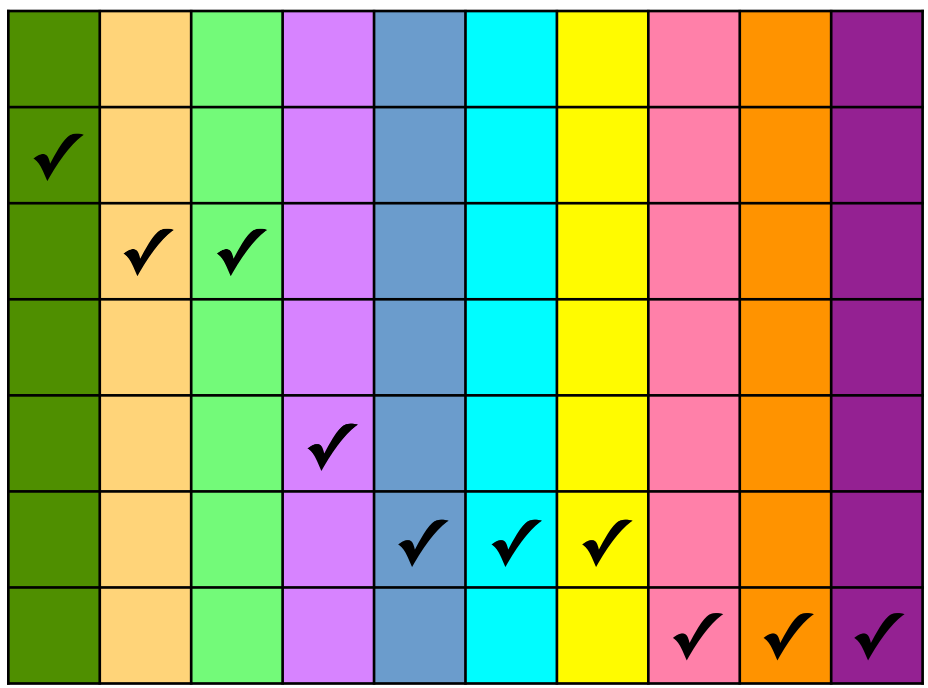
Today: Subcolumn Maximum Queries



Even easier: entire-column ranges

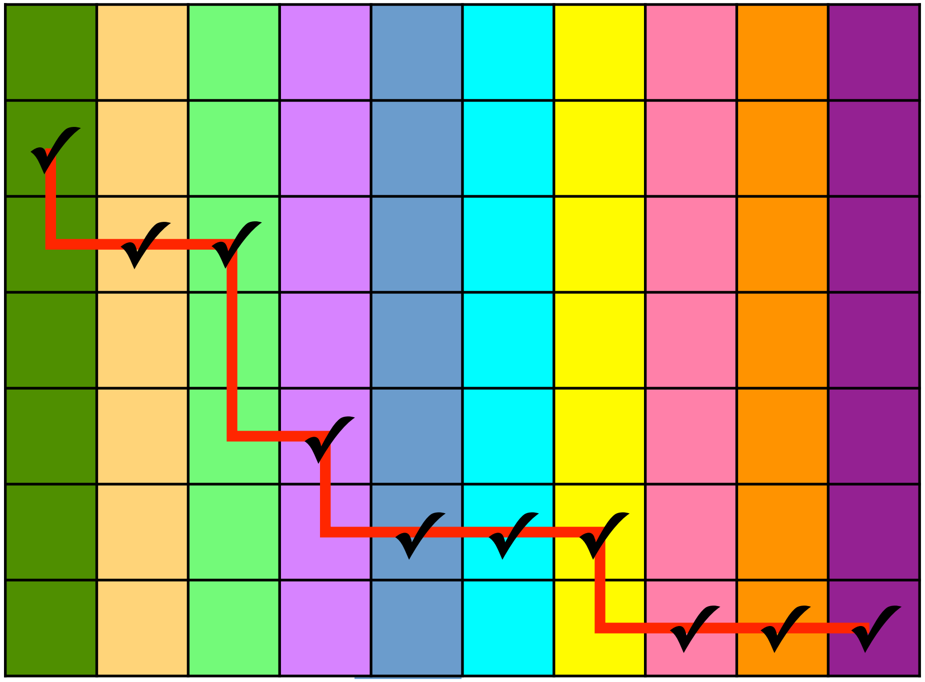


Even easier: entire-column ranges



The rows of the column maxima increase monotonically

Even easier: entire-column ranges

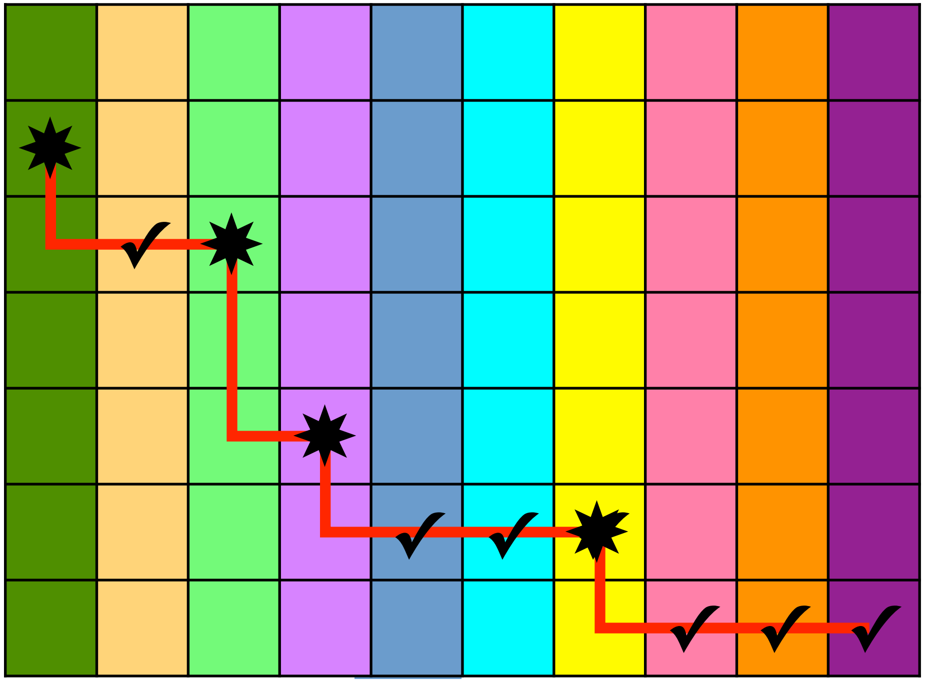


The rows of the column maxima increase monotonically

Even easier: entire-column ranges

Enough to compute list of breakpoints ✸ (predecessor search)

O(n) time SMAWK [Shor, Moran, Aggarwal, Wilber, Klawe 1987]

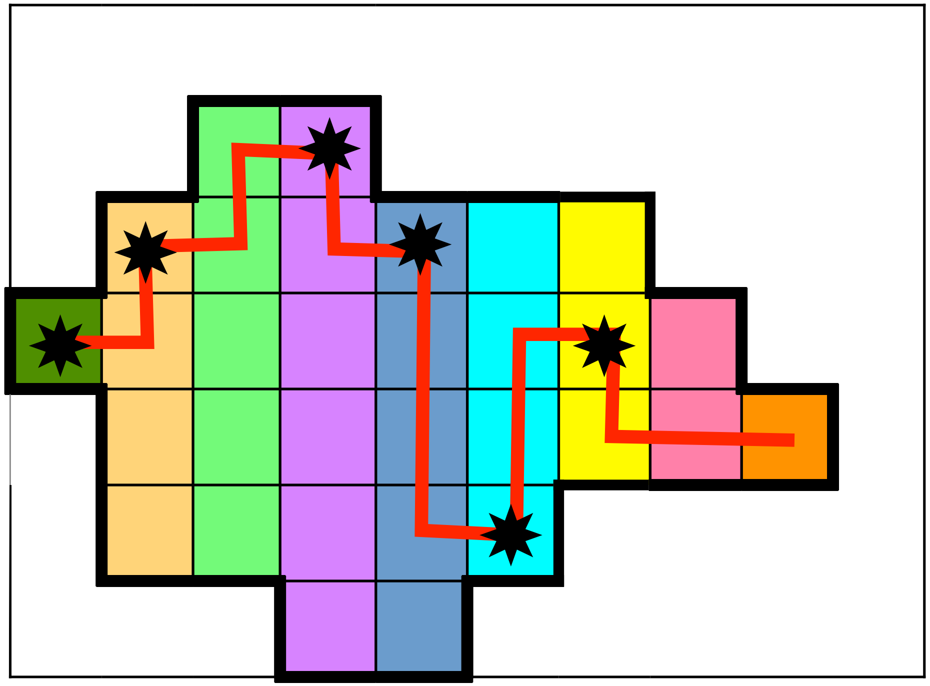


The rows of the column maxima increase monotonically

Even easier: entire-column ranges

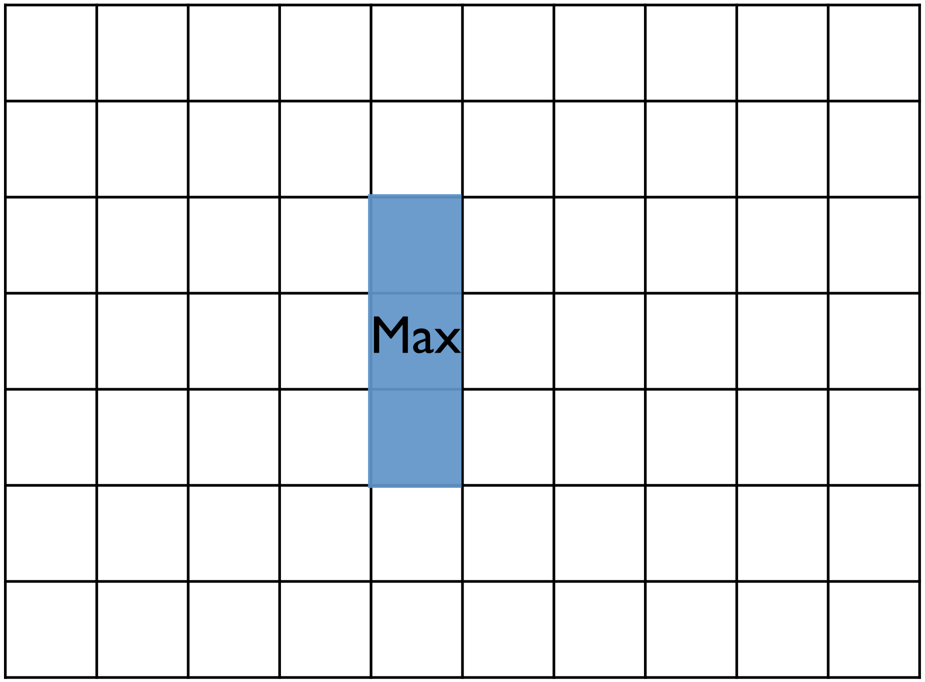
Enough to compute list of breakpoints ✸ (predecessor search)

- $O(n)$ time SMAWK [Shor, Moran, Aggarwal, Wilber, Klawe 1987]
- $O(n\,\alpha(n))$ time for partial matrices [Klawe, Kleitman 1990]



~~The rows of the column maxima increase monotonically~~

Subcolumn Maximum Queries

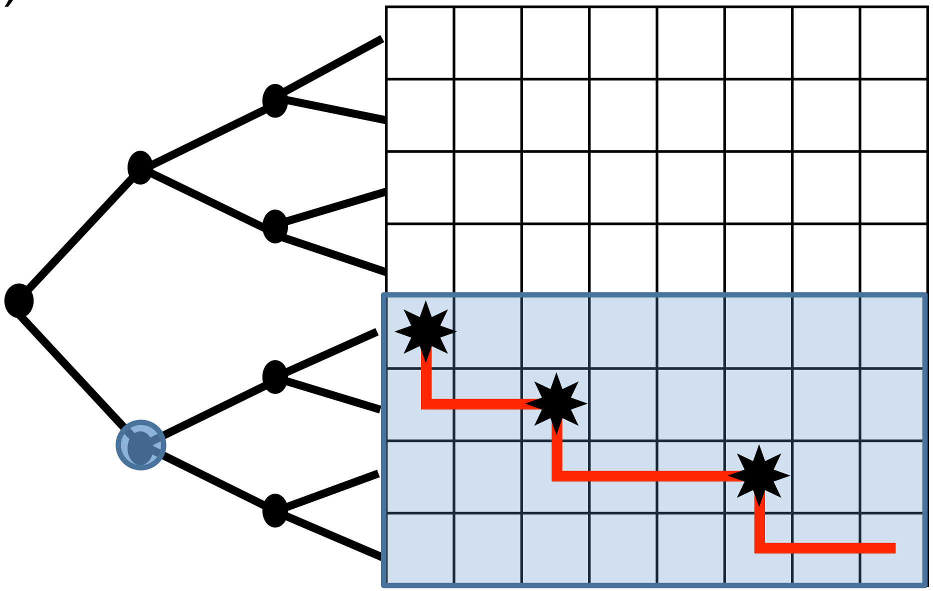


The tree of breakpoints

[Kaplan, Mozes, Nussbaum, Sharir SODA'12]
[Gawrychowski, Mozes, W. ICALP'14]

Each node computes the breakpoints of its submatrix

By merging the breakpoints of its two children
(overall $O(n \log n)$ time and space)

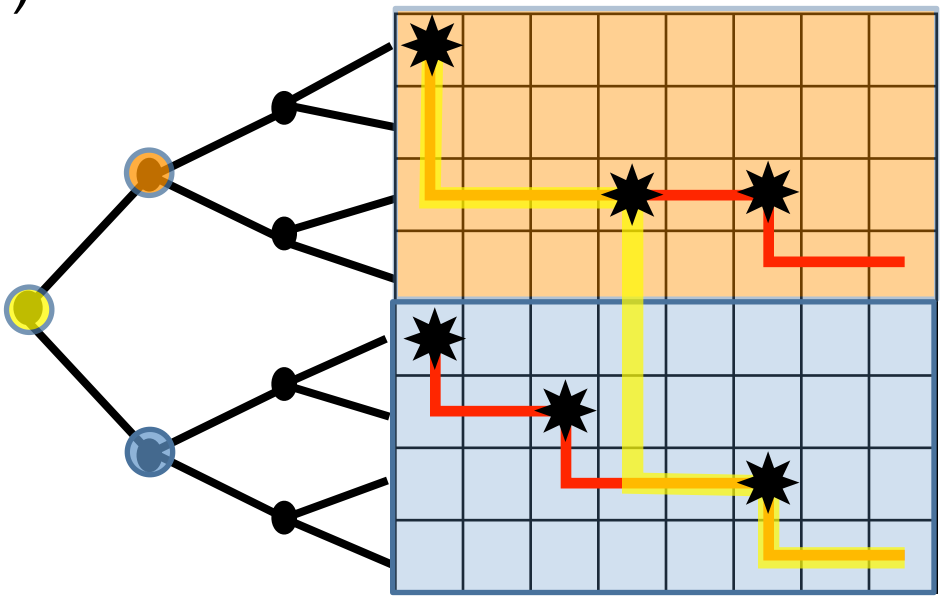


The tree of breakpoints

[Kaplan, Mozes, Nussbaum, Sharir SODA'12]
[Gawrychowski, Mozes, W. ICALP'14]

Each node computes the breakpoints of its submatrix

By merging the breakpoints of its two children
(overall $O(n \log n)$ time and space)

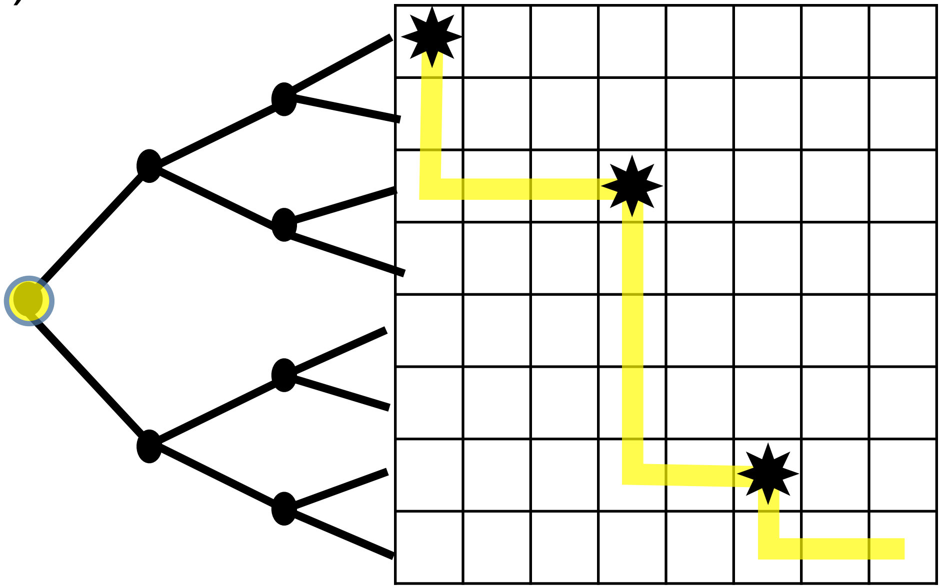


The tree of breakpoints

[Kaplan,Mozes,Nussbaum,Sharir SODA'12]
[Gawrychowski, Mozes, W. ICALP'14]

Each node computes the breakpoints of its submatrix

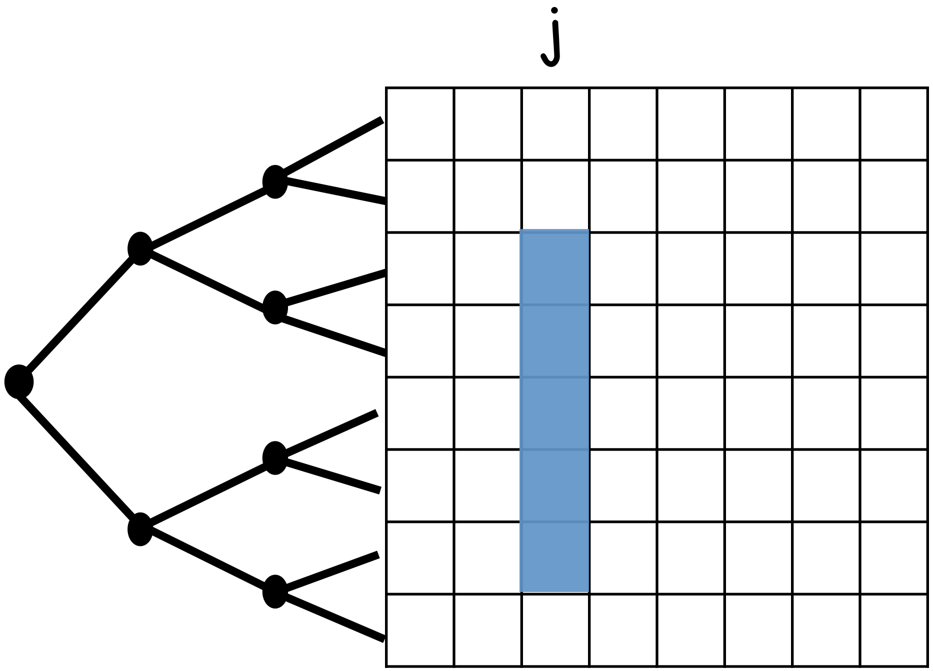
By merging the breakpoints of its two children
(overall $O(n \log n)$ time and space)



The tree of breakpoints

[Kaplan,Mozes,Nussbaum,Sharir SODA'12]
[Gawrychowski, Mozes, W. ICALP'14]

A subcolumn query



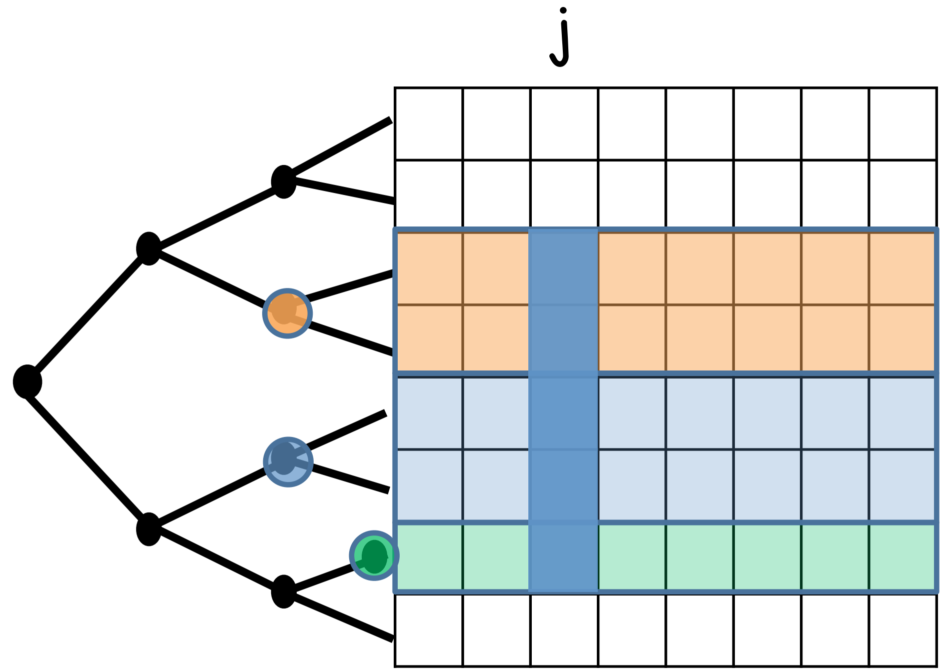
The tree of breakpoints

[Kaplan,Mozes,Nussbaum,Sharir SODA'12]
[Gawrychowski, Mozes, W. ICALP'14]

A subcolumn query is covered by O(log *n*) *canonical* nodes.
In each canonical node, find predecessor(j) in its breakpoints.

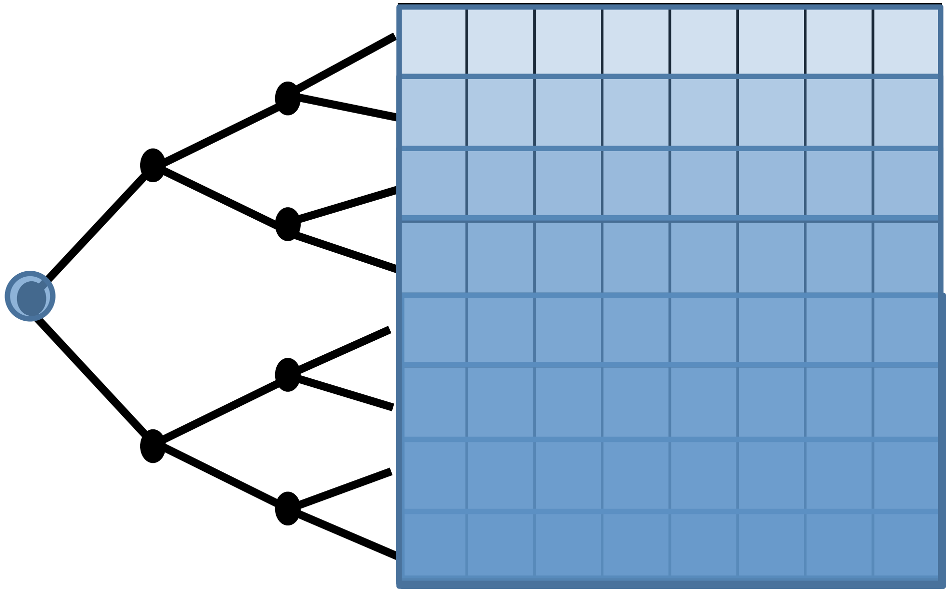
O(log *n* loglog *n*) time

O(log *n*) via fractional cascading



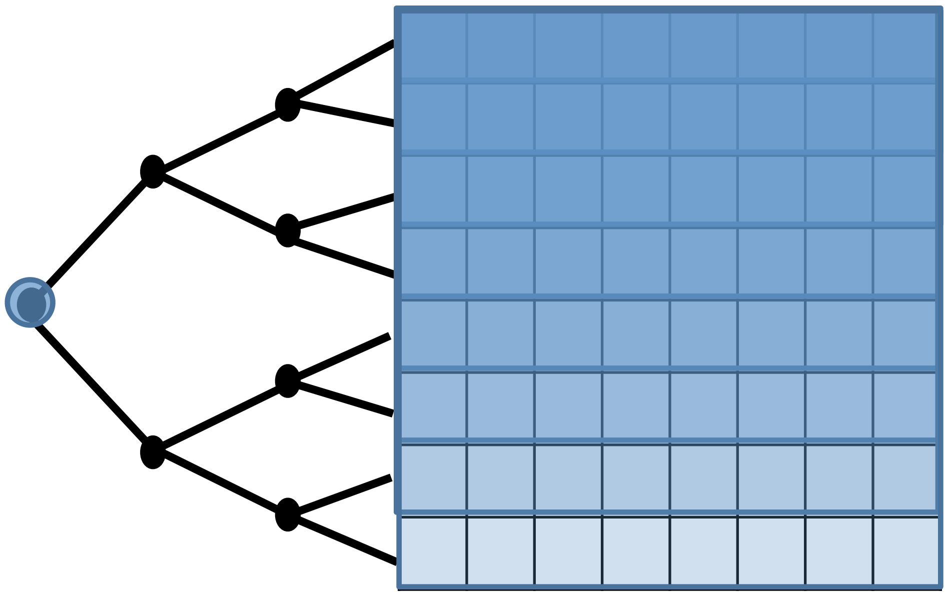
Our Tree

Each node stores breakpoints of every suffix/prefix of rows



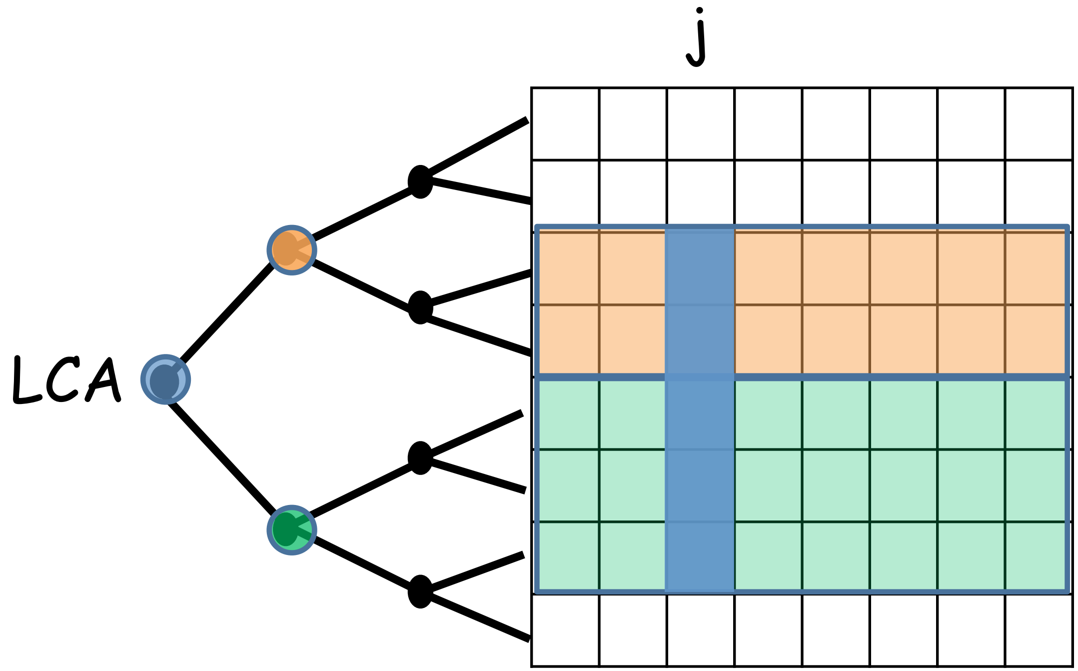
Our Tree

Each node stores breakpoints of every suffix/prefix of rows

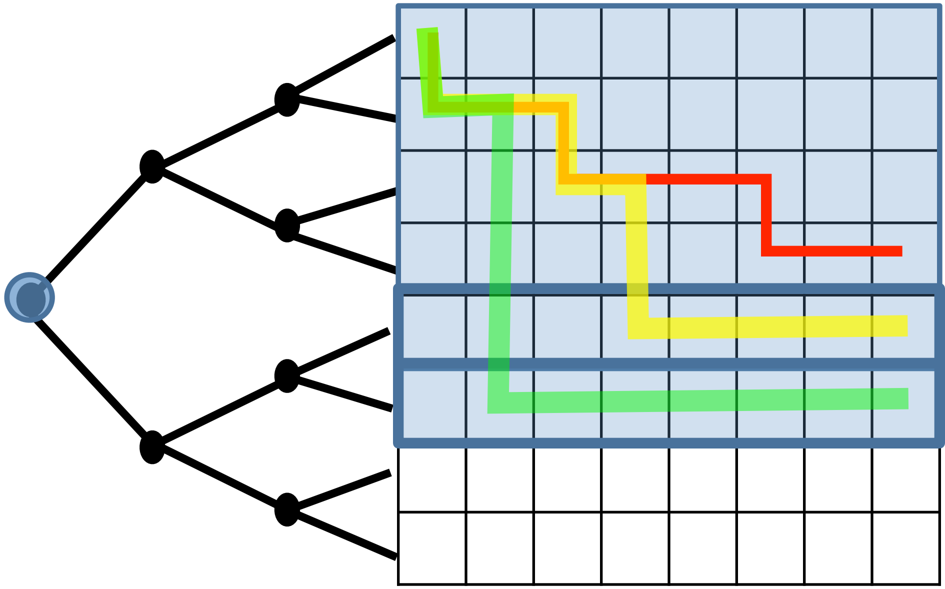


Our Tree - Query

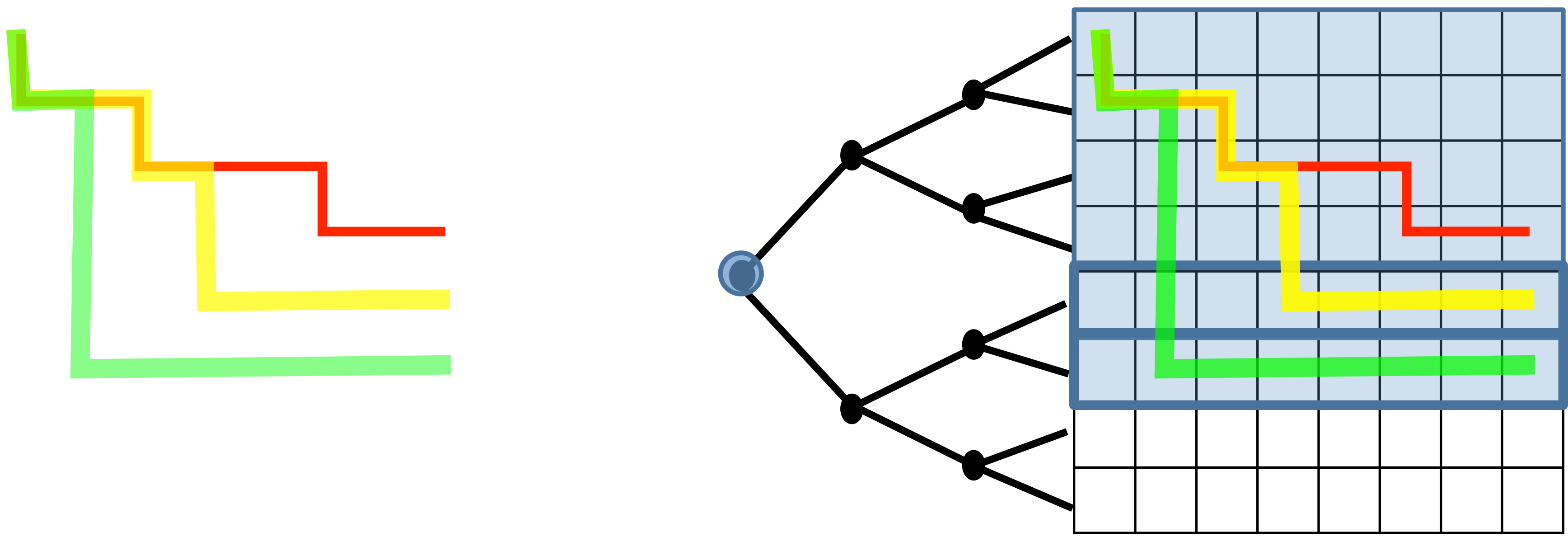
A subcolumn query only two predecessor(j) searches



Our Tree - Construction

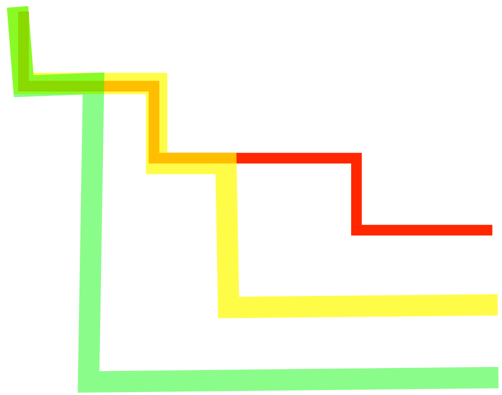


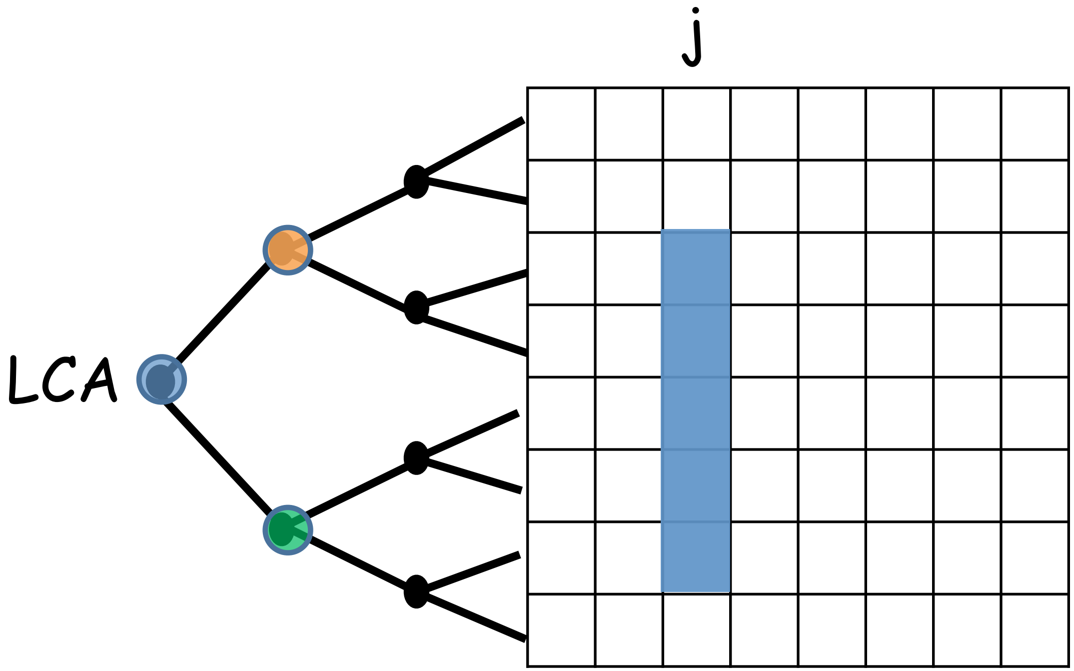
Our Tree - Construction



Our Tree - Construction

A subcolumn query





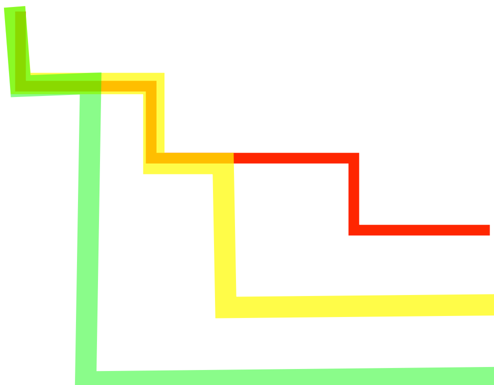
Our Tree - Construction

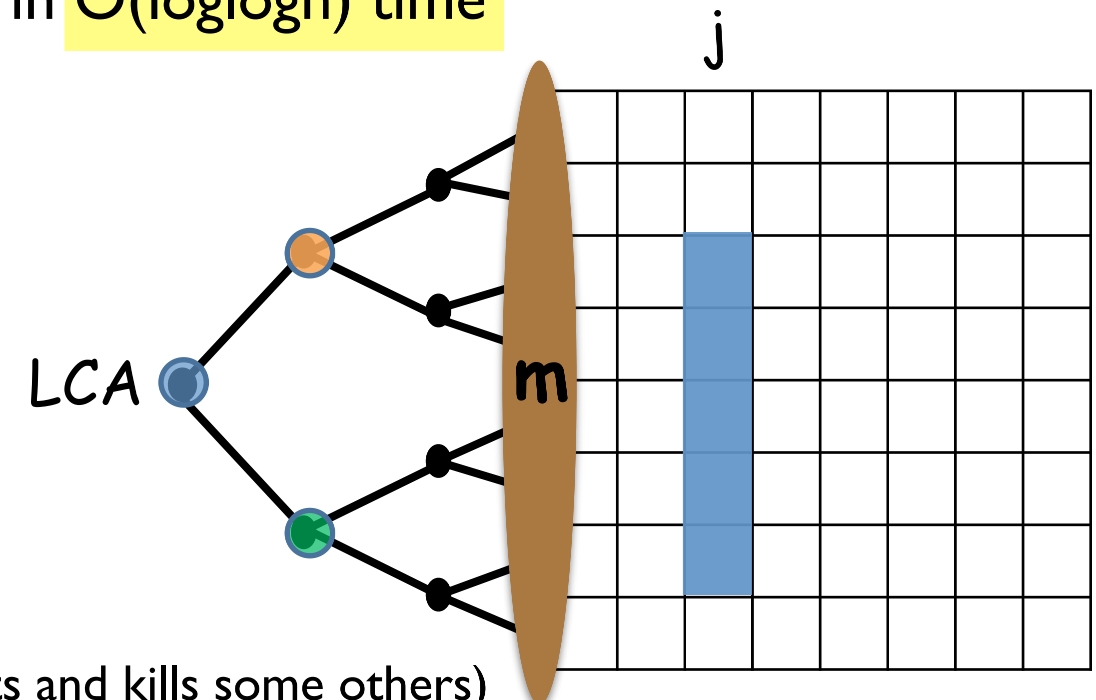
A subcolumn query

= 2 x predecessor queries on a root-to-leaf path

= 2 x weighted ancestor queries

= O(1) predecessor queries in O(loglogn) time





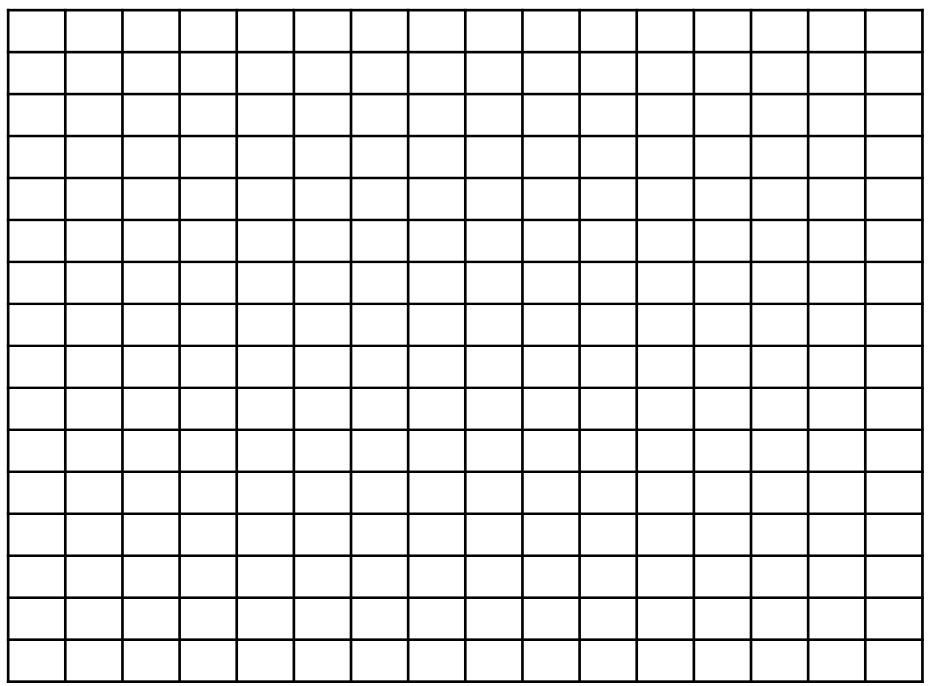
size of this tree = m

(each row adds one breakpoints and kills some others)

size of all trees = O(nlogn) space

Improving the space

from $O(n \log n)$ to $O(n)$

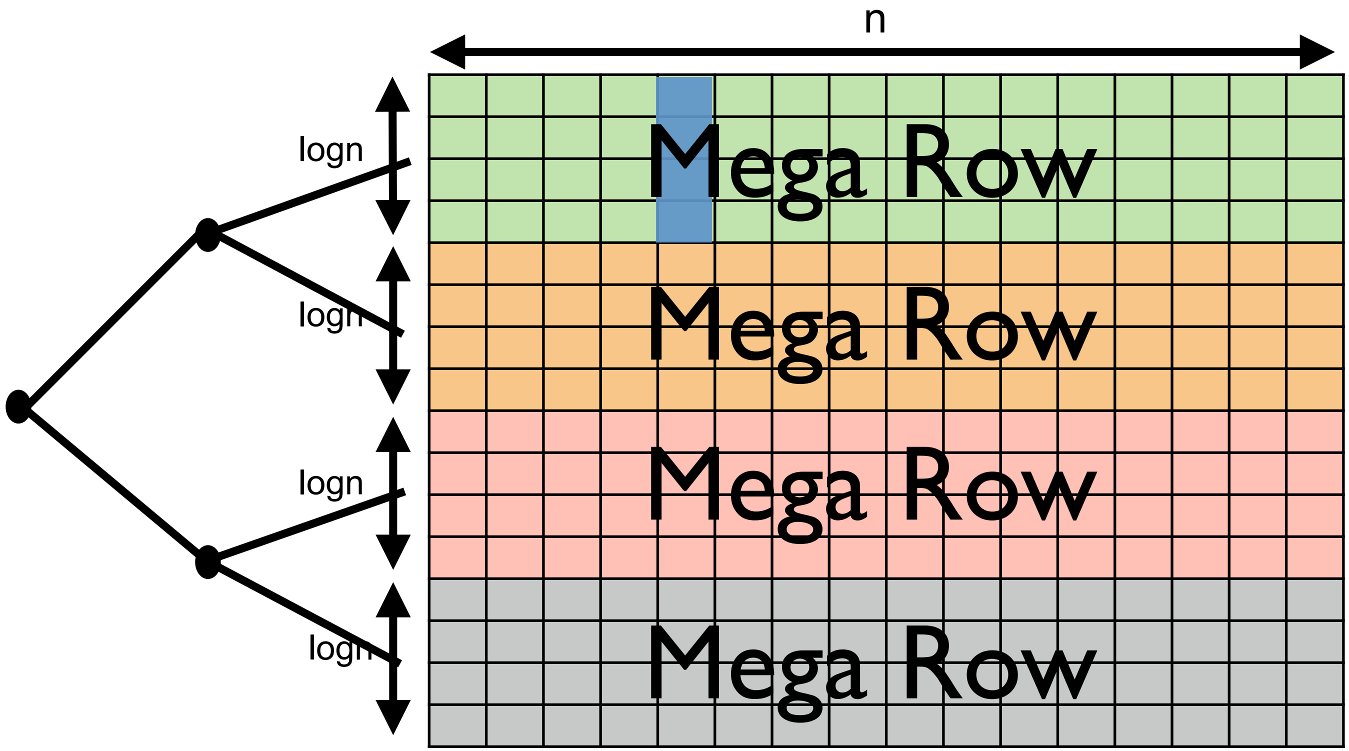


Improving the space

<u>Theorem</u> [Gawrychowski, Mozes, W. ICALP'14]

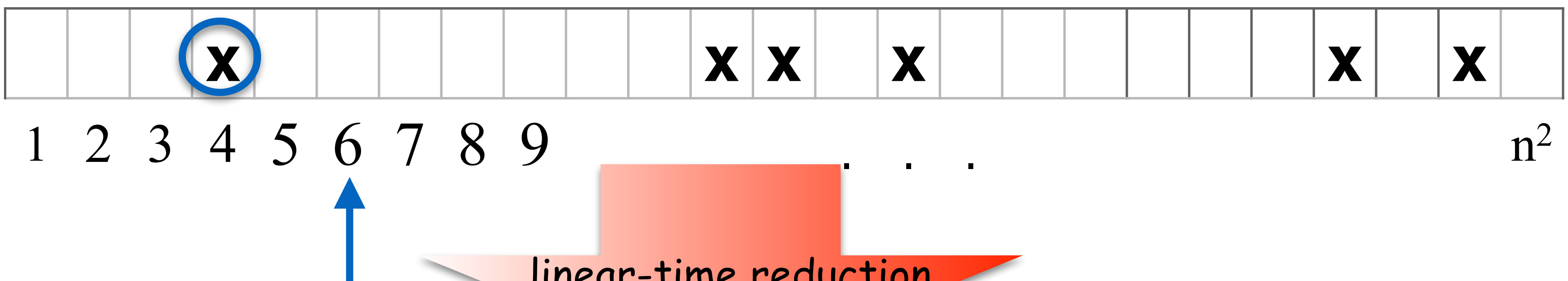
Given a logn-by-n Monge matrix, there is a O(logn) space data structure that answers *entire-column* queries in O(1) time.

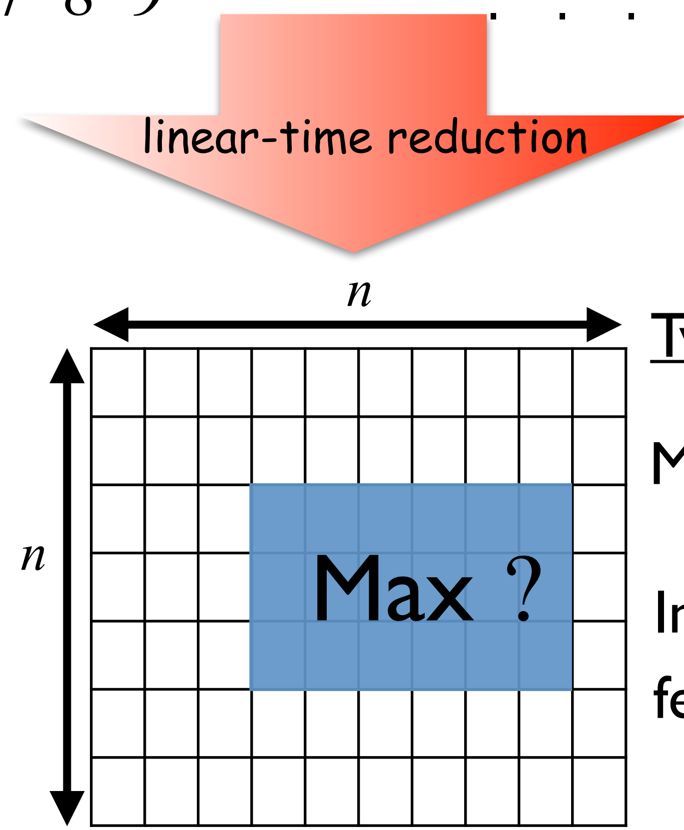
Mega Row entries fetched in O(1) time using the above Theorem.



Lower Bound

Given n integers in $\{1,2,\ldots,n^2\}$:



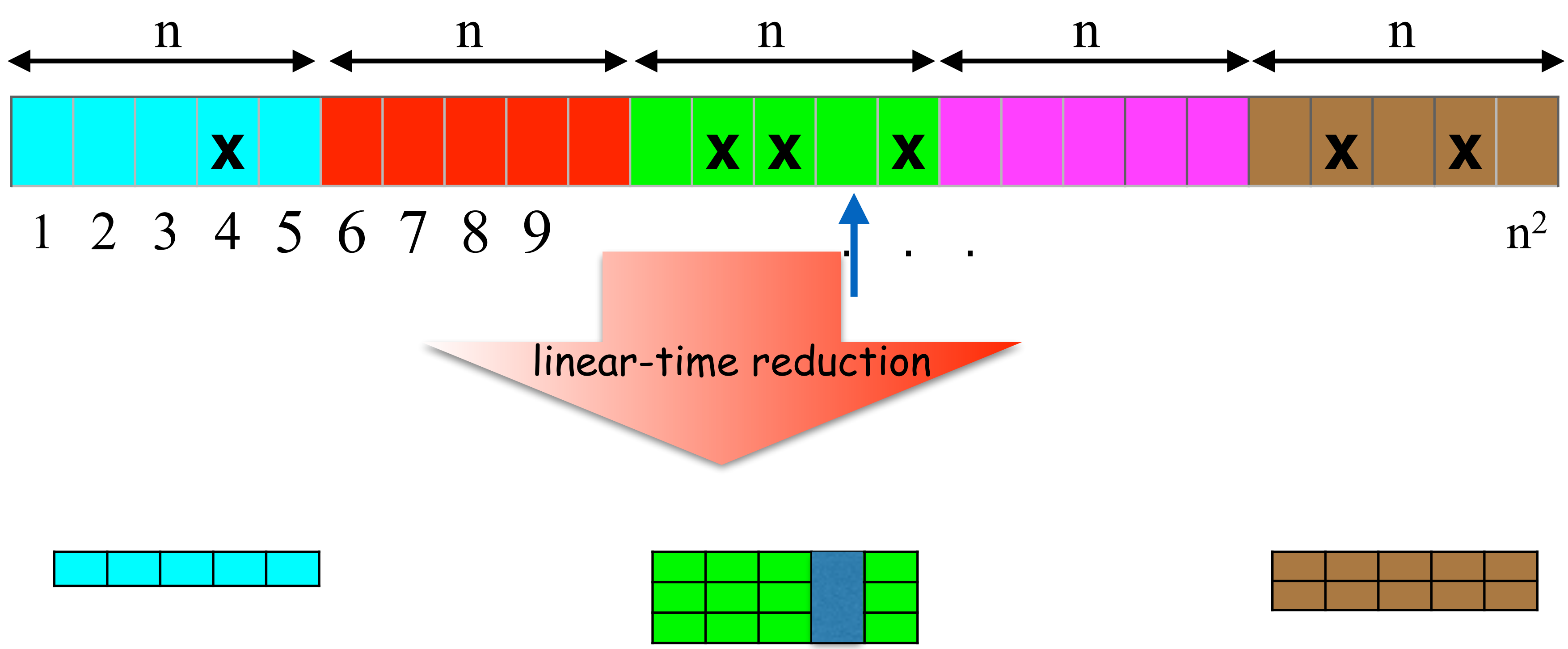


Two challenges:

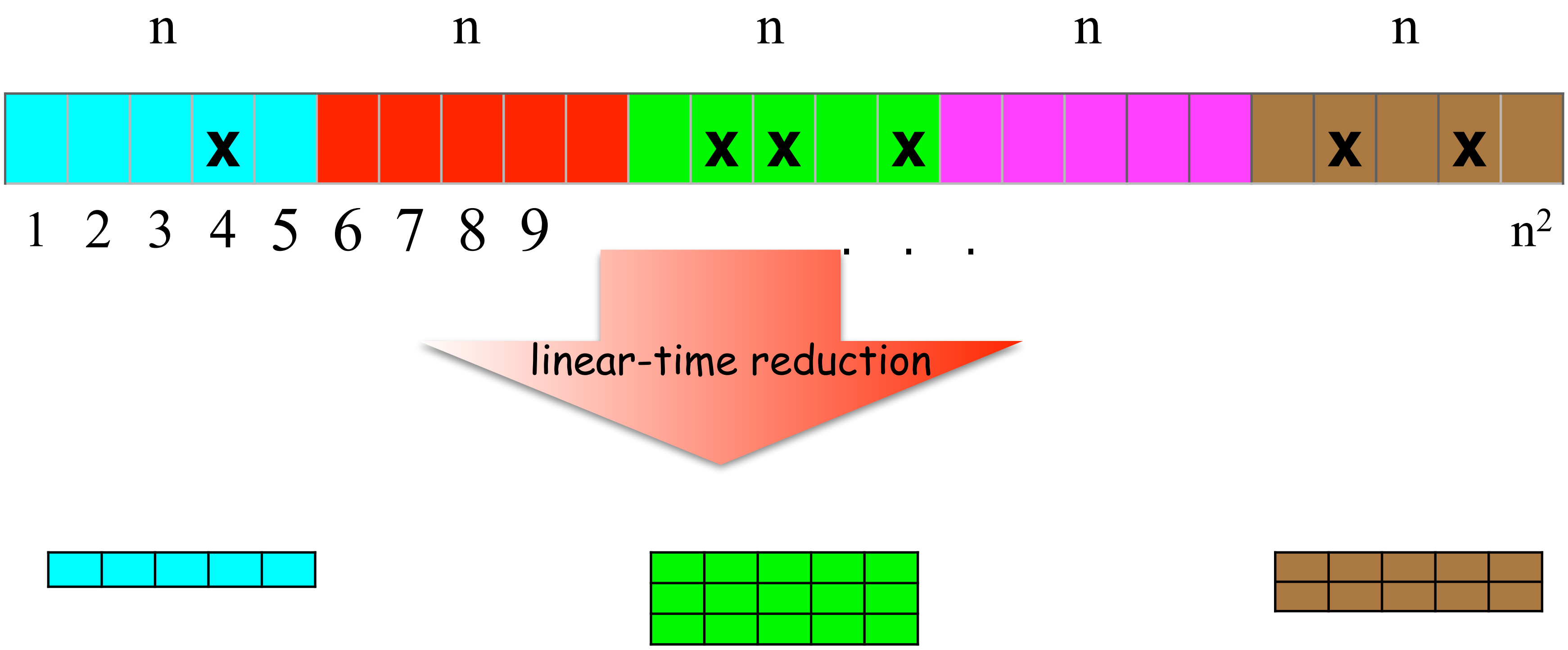
Matrix has to be Monge

Implicit $O(n \operatorname{polylog} n)$ rep. fetching entries in $O(1)$ time

Lower Bound

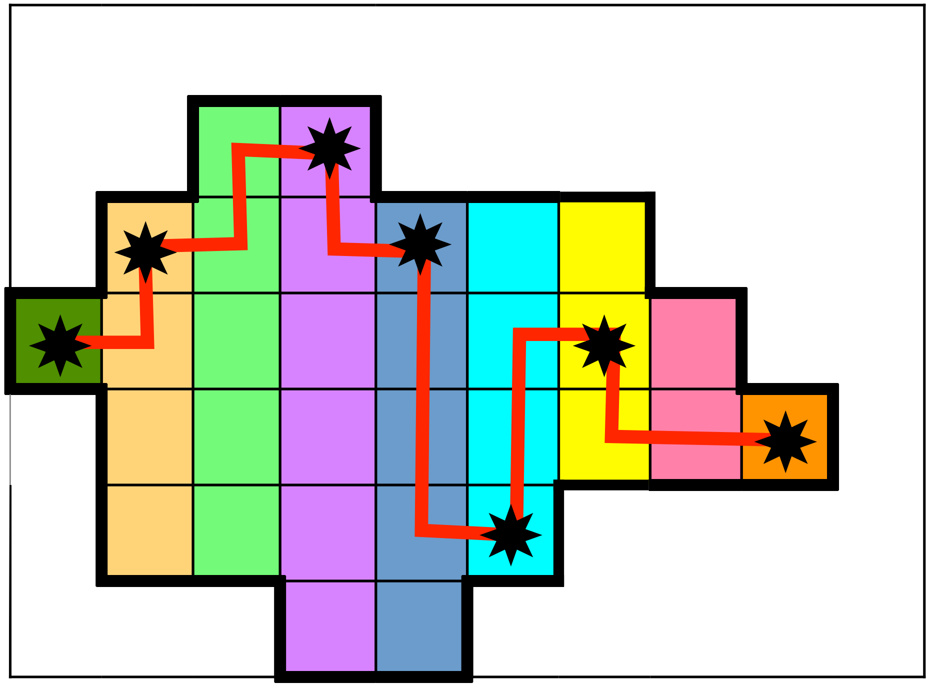


Lower Bound



Partial Monge matrices

Partial Monge matrices

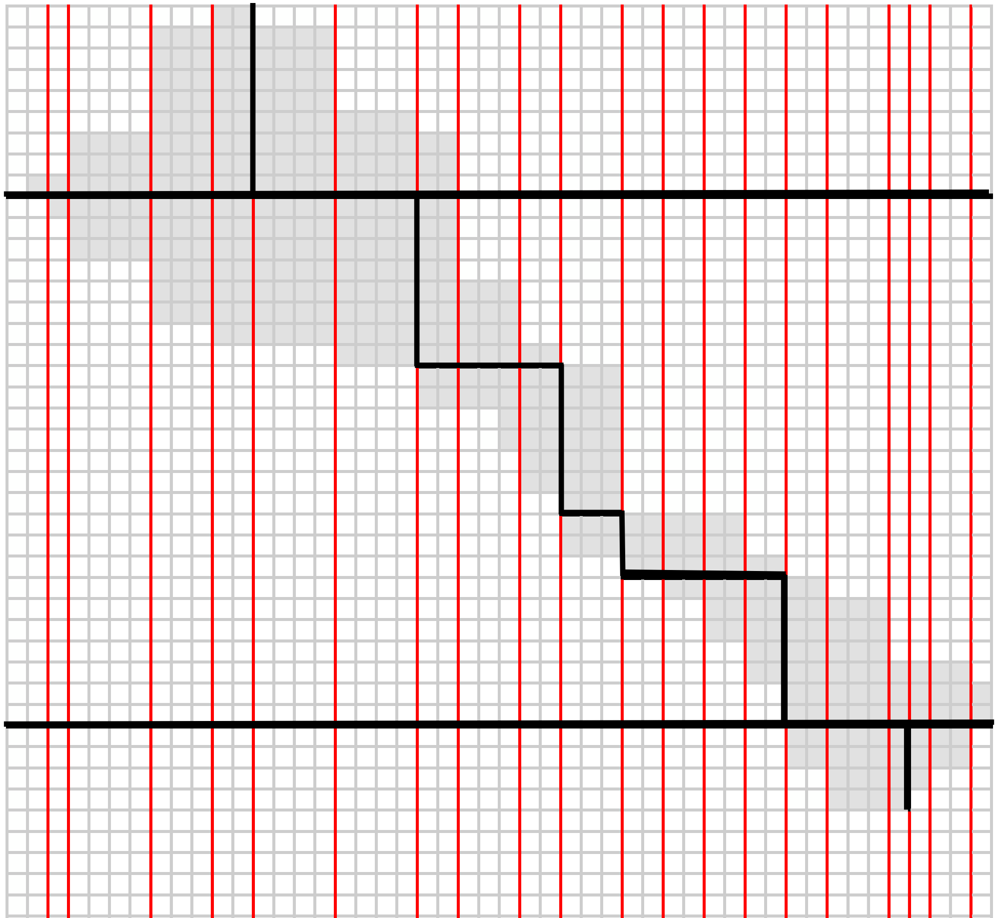


~~The rows of the column maxima increases monotonically~~

Partial Monge matrices

Partial Monge matrices

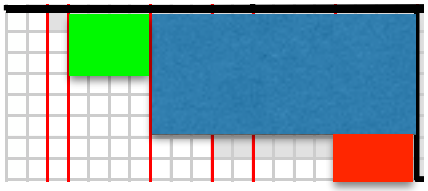
Decomposition I: Reduction to *staircase* matrices



Partial Monge matrices

Decomposition I: Reduction to *staircase* matrices
Decomposition II: Cover staircase matrices by full matrices

Much more…



Open Problems

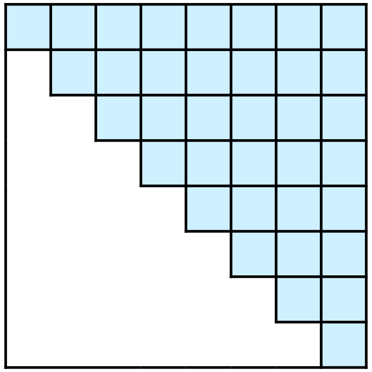
Open Problems

- Shortest paths in planar graphs

Open Problems

- Shortest paths in planar graphs

m-by-n staircase



Open Problems

- Shortest paths in planar graphs
  - In the beginning all rows are deactivated

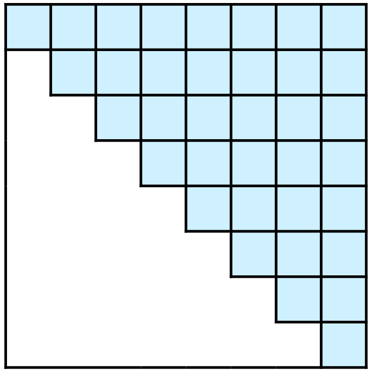
O($\log^2 n$) • activate a row and add k to all its entries

O($\log^2 n$) • delete column

O($\log^2 n$) • report minimum active entry

[Fakcharoenphol Rao, 2006]

m-by-n staircase



Open Problems

- Shortest paths in planar graphs
  - In the beginning all rows are deactivated

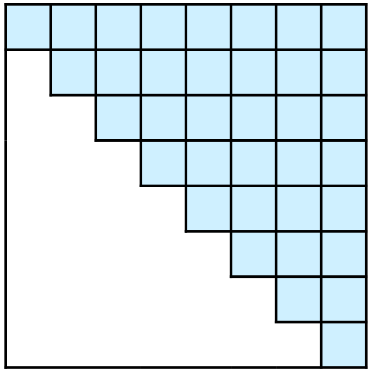
$O(\log^2 n)$ • activate a row and add k to all its entries

$O(\log^2 n)$ • delete column

$O(\log^2 n)$ • report minimum active entry

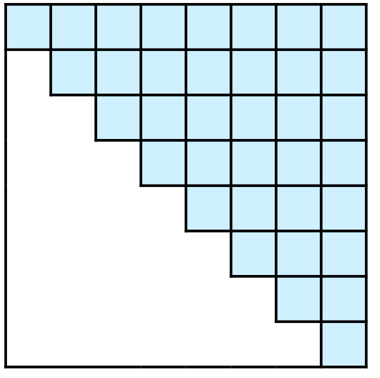
[Fakcharoenphol Rao, 2006]

m-by-n staircase



- Find the O(m) breakpoints in linear time

Open Problems



- Shortest paths in planar graphs
  - In the beginning all rows are deactivated

$O(\log^2 n)$ • activate a row and add k to all its entries

$O(\log^2 n)$ • delete column

$O(\log^2 n)$ • report minimum active entry

[Fakcharoenphol Rao, 2006]

- Find the O(m) breakpoints in linear time
  - $O((m+n)\alpha(n)))$ [Klawe Kleitman, 1990]
  - $O(m\log n)$ [Gawrychowski, Mozes, W. ICALP'14]

Arigato!