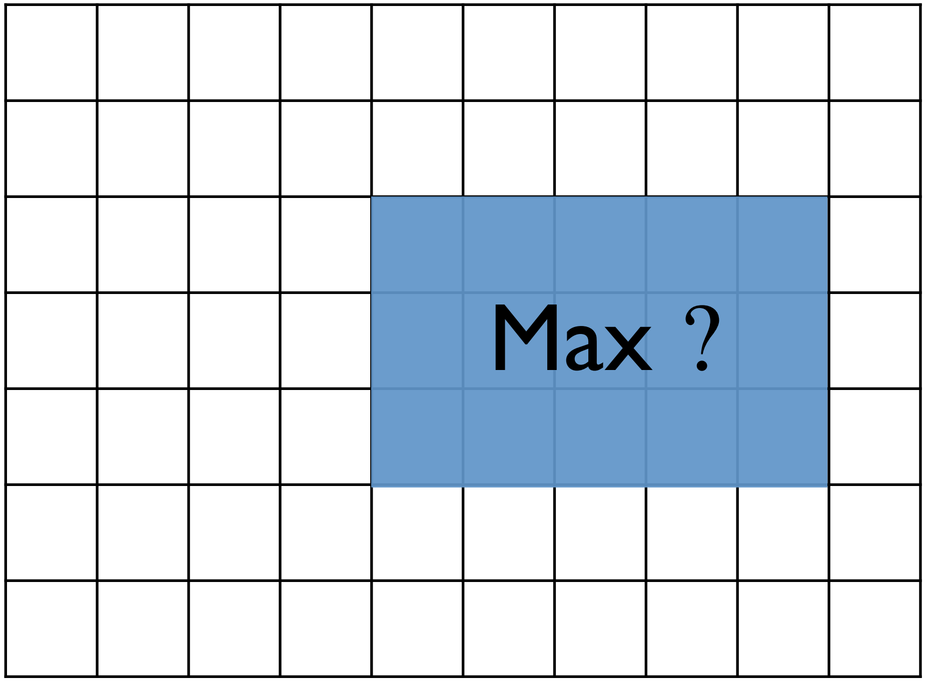
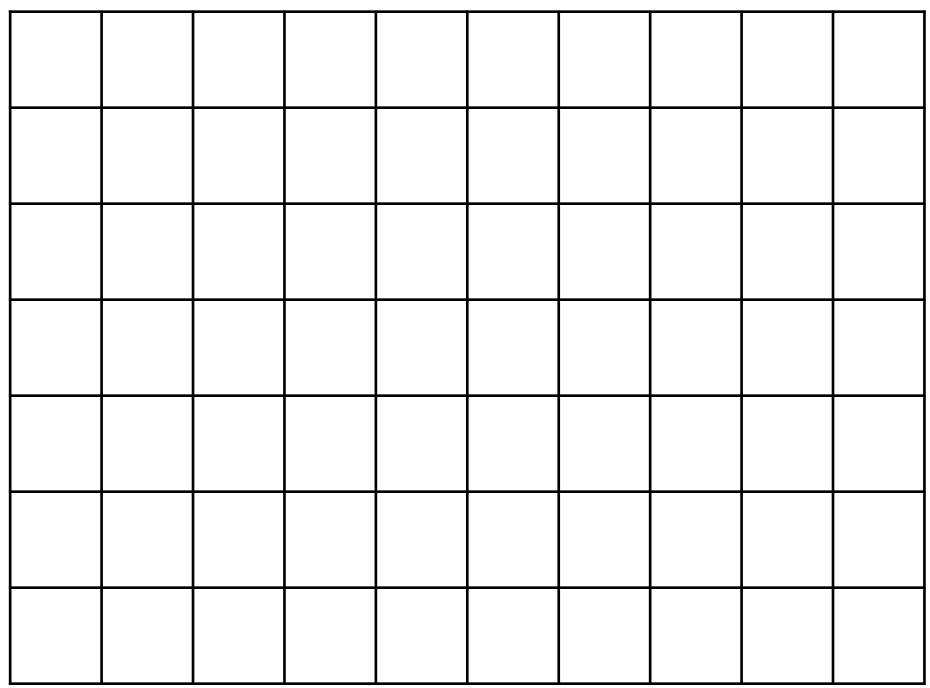
Improved Submatrix Maximum Queries in Monge Matrices

Pawel Gawrychowski, Shay Mozes, Oren Weimann



Improved Submatrix Maximum Queries in Monge Matrices

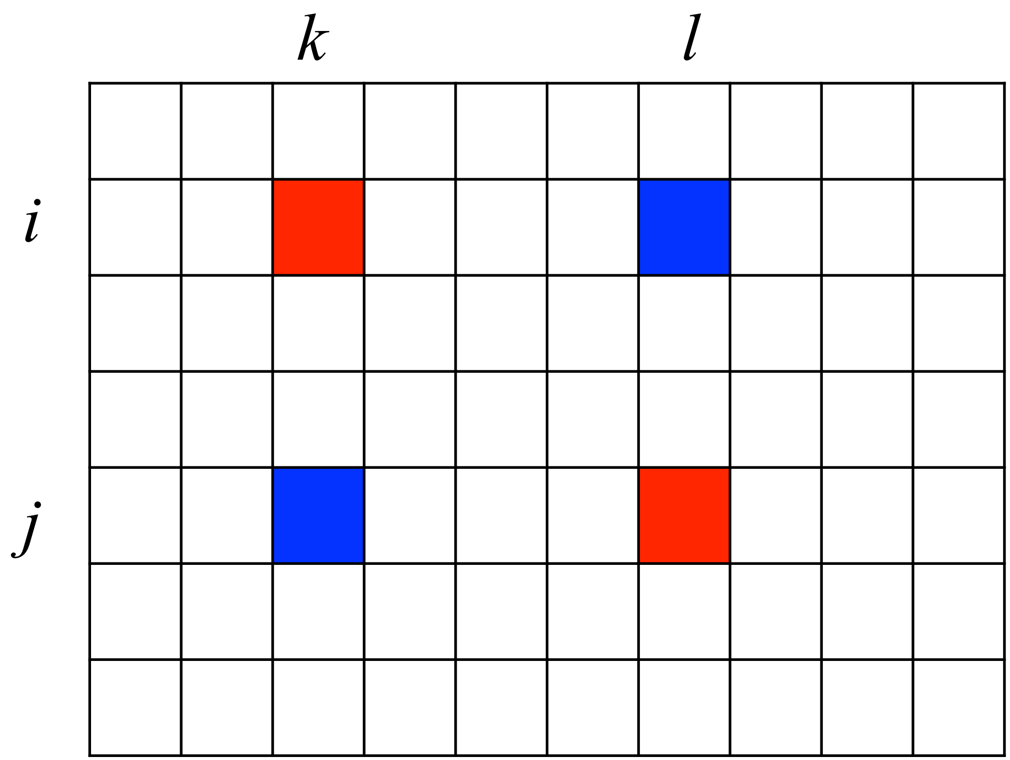
1746 - 1818



Improved Submatrix Maximum Queries in Monge Matrices

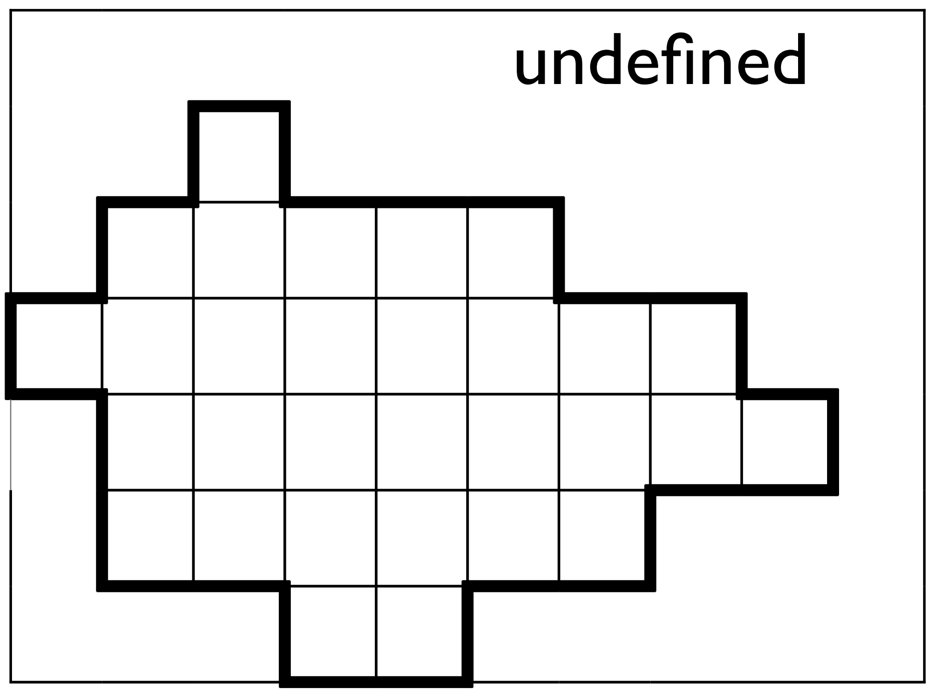
$$M_{ik} + M_{jl} \geq M_{il} + M_{jk}$$

1746 - 1818



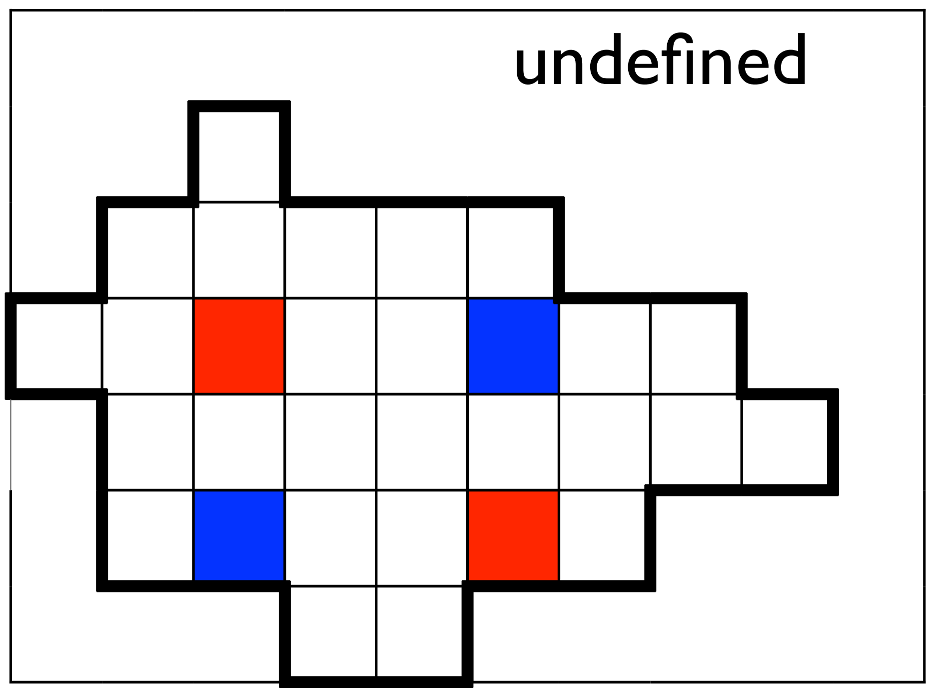
Improved Submatrix Maximum Queries

Partial Monge Matrices



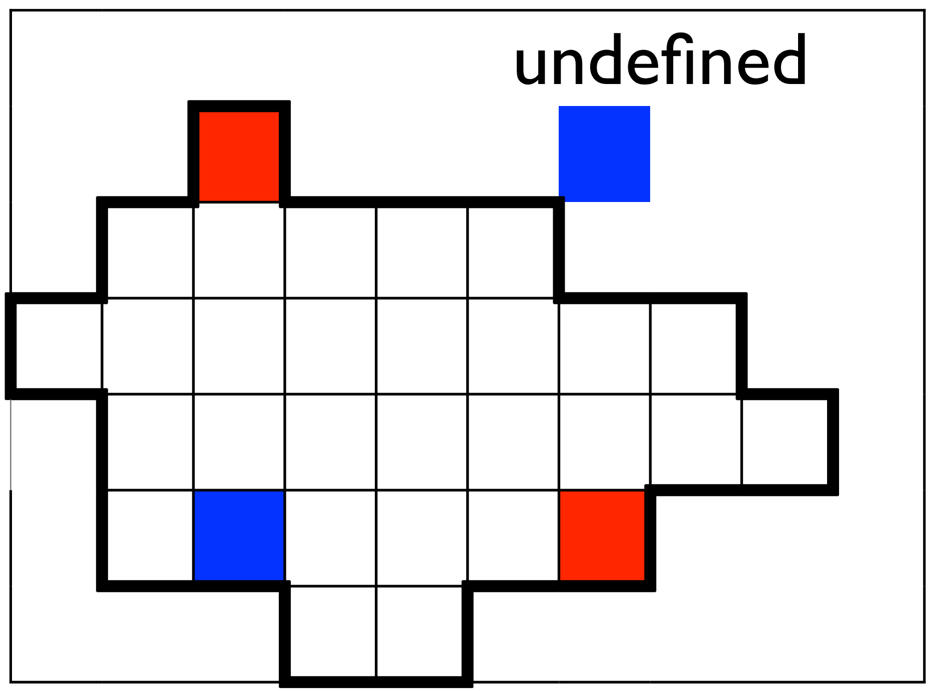
Partial Monge Matrices

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

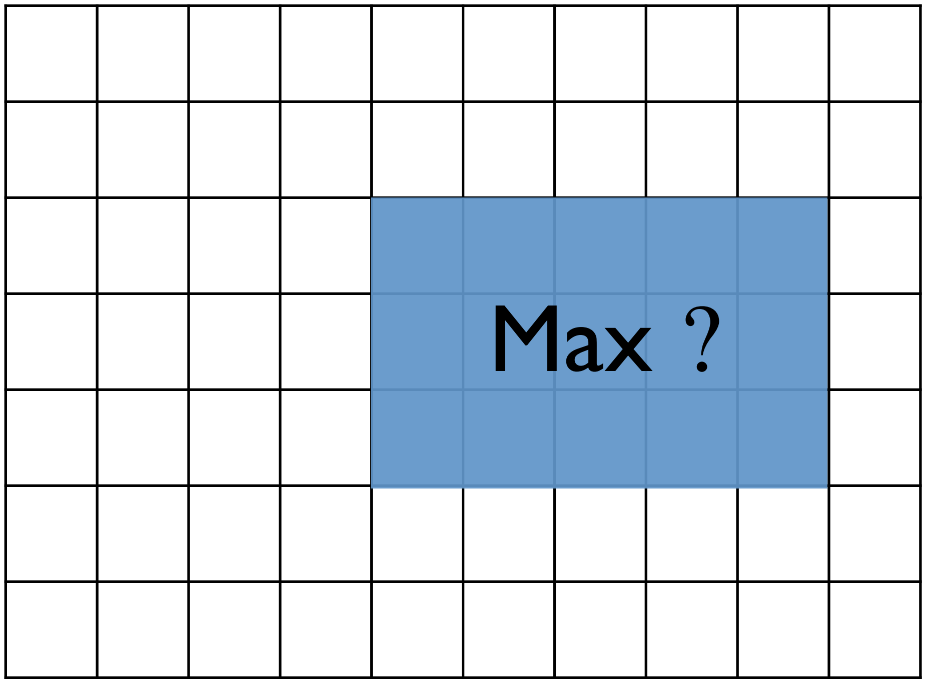


Partial Monge Matrices

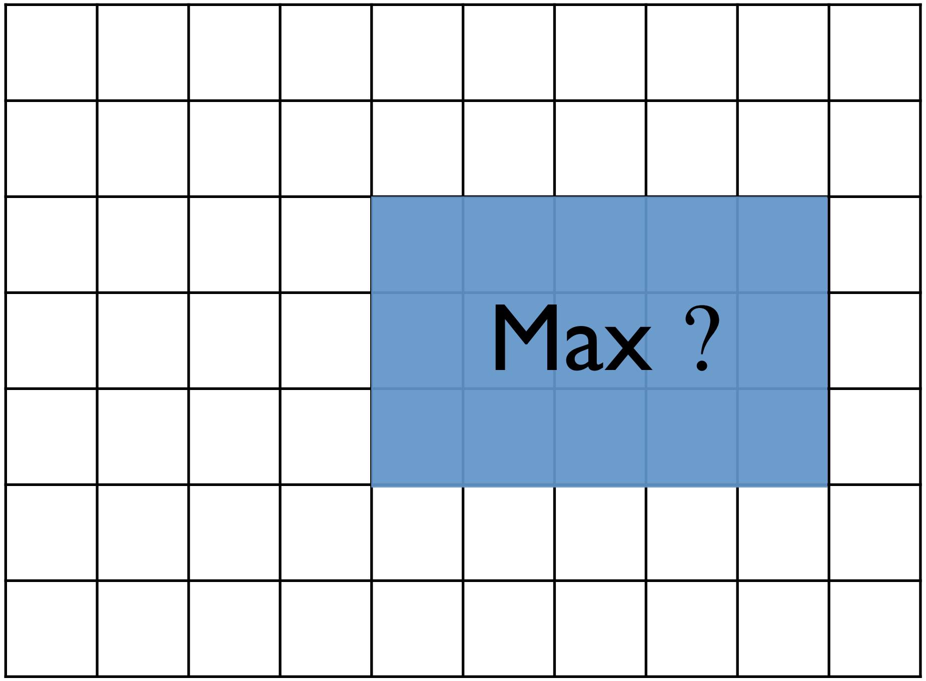
$$M_{ik} + M_{jl} \ ? \ M_{il} + M_{jk}$$



Improved Submatrix Maximum Queries in Monge Matrices



Improved Submatrix Maximum Queries in Monge Matrices



Improved Submatrix Maximum Queries in Monge Matrices

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

For an n x n matrix:

Space	$O(n \log n)$	$O(n)$
Query	$O(\log^2 n)$	$O(\log n)$

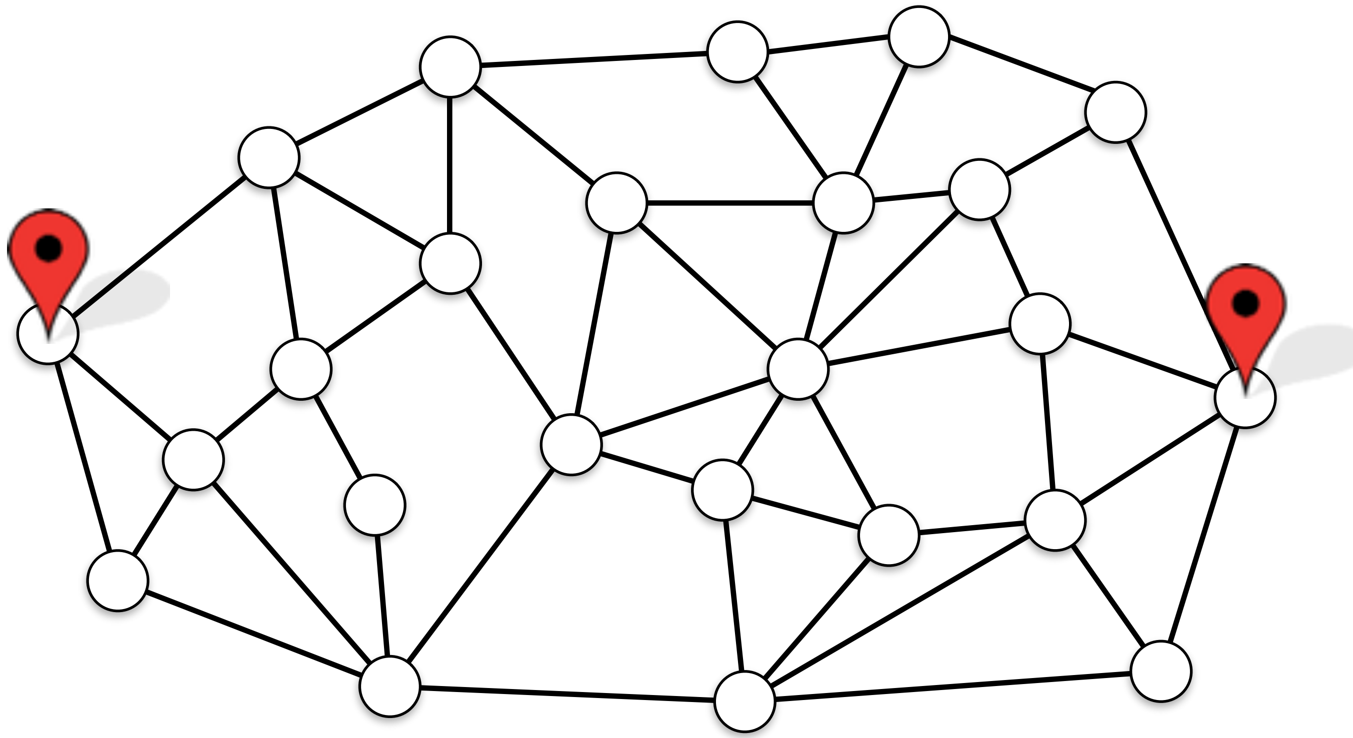
For an n x n partial matrix:

Space	$O(n \log n \, \alpha(n))$	$O(n)$
Query	$O(\log^2 n)$	$O(\log n \, \alpha(n))$

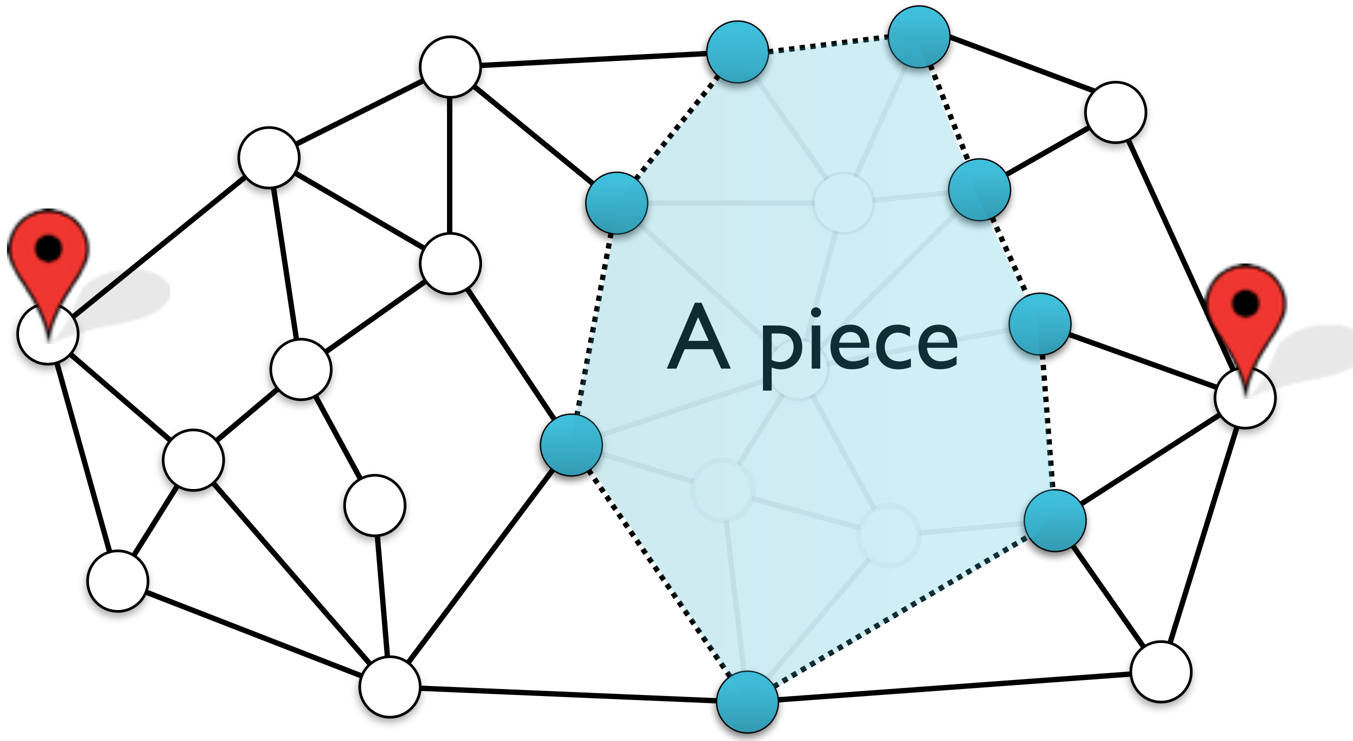
Applications

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

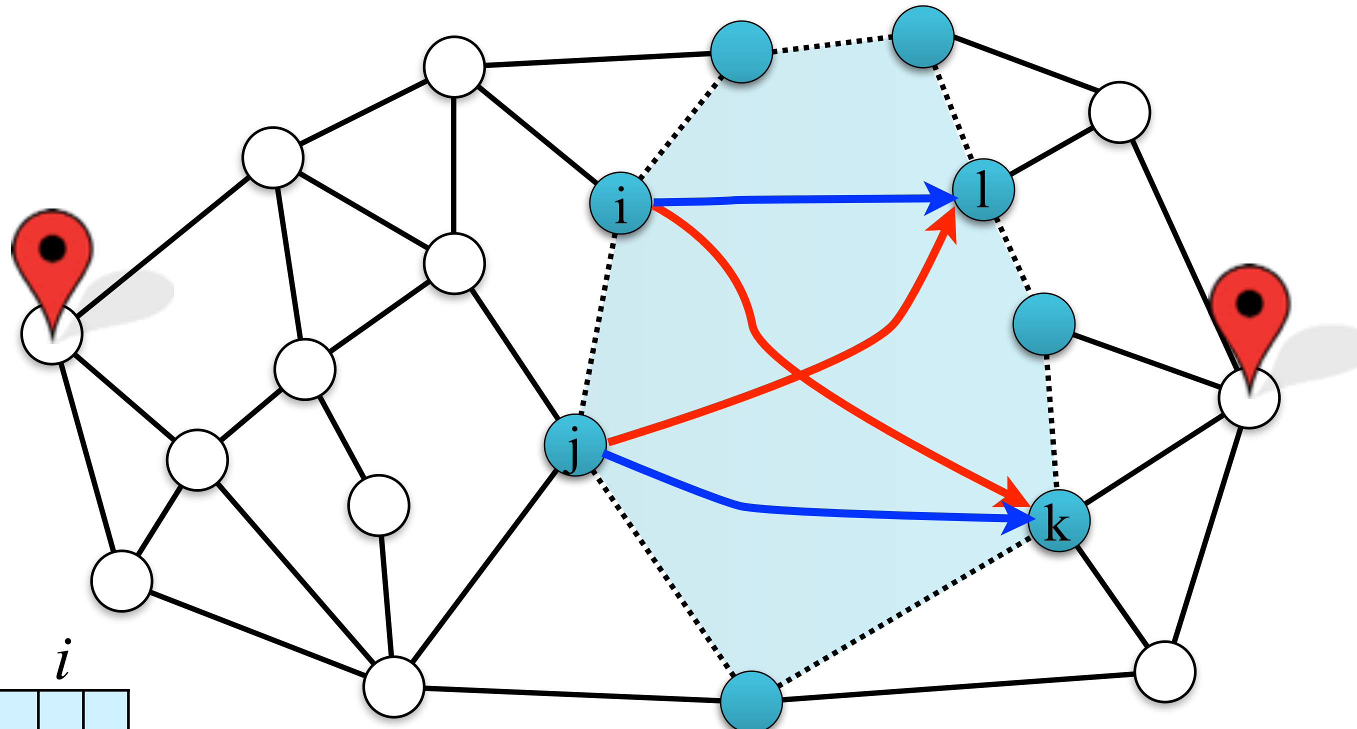
Application I
Shortest paths in planar graphs

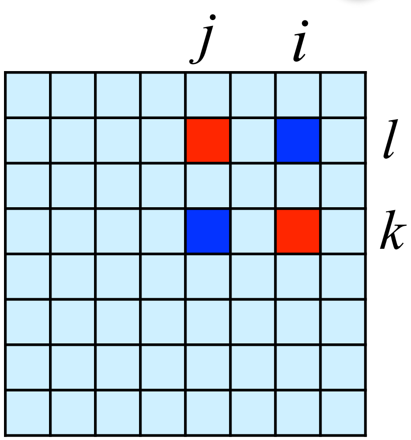


Application I
Shortest paths in planar graphs



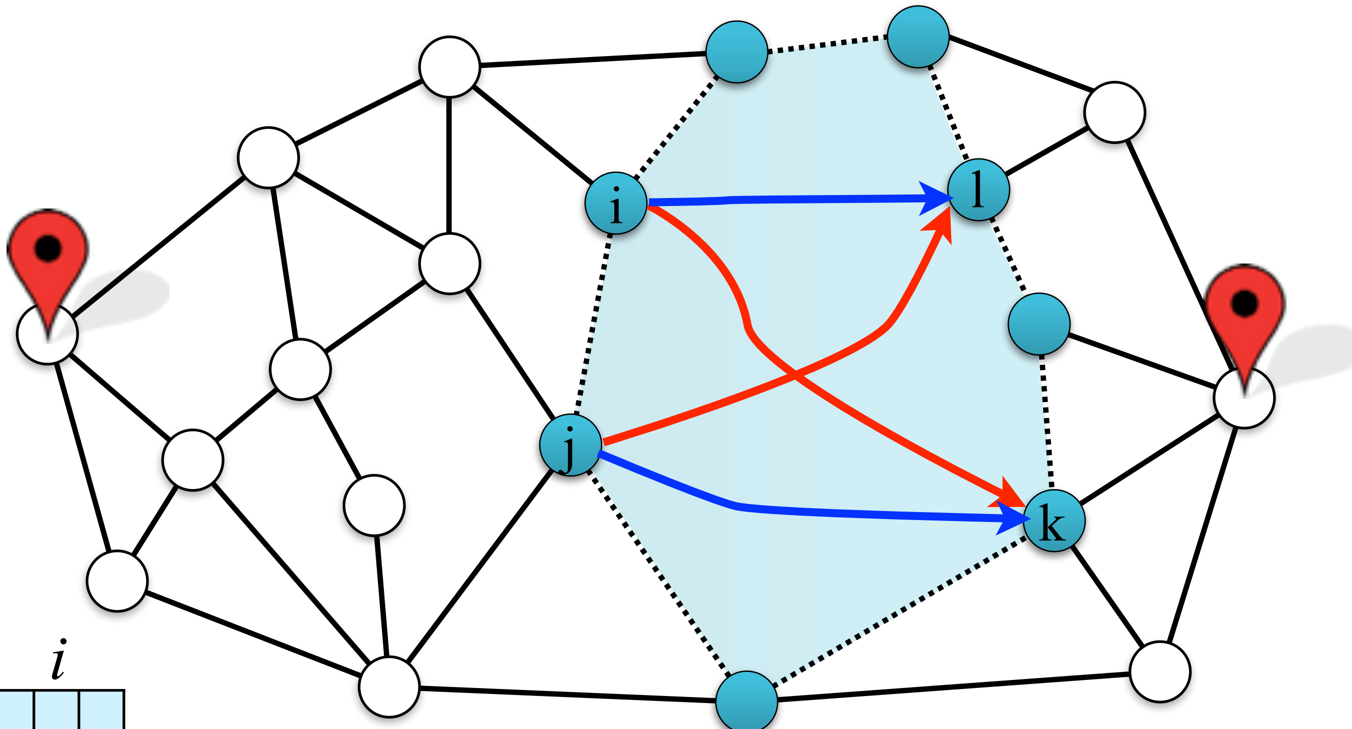
Application I
Shortest paths in planar graphs

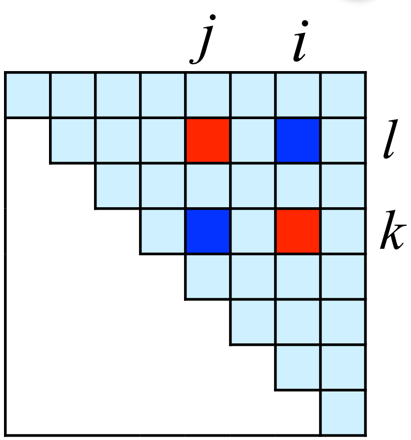




$$M_{ik} + M_{jl} \geq M_{il} + M_{jk}$$

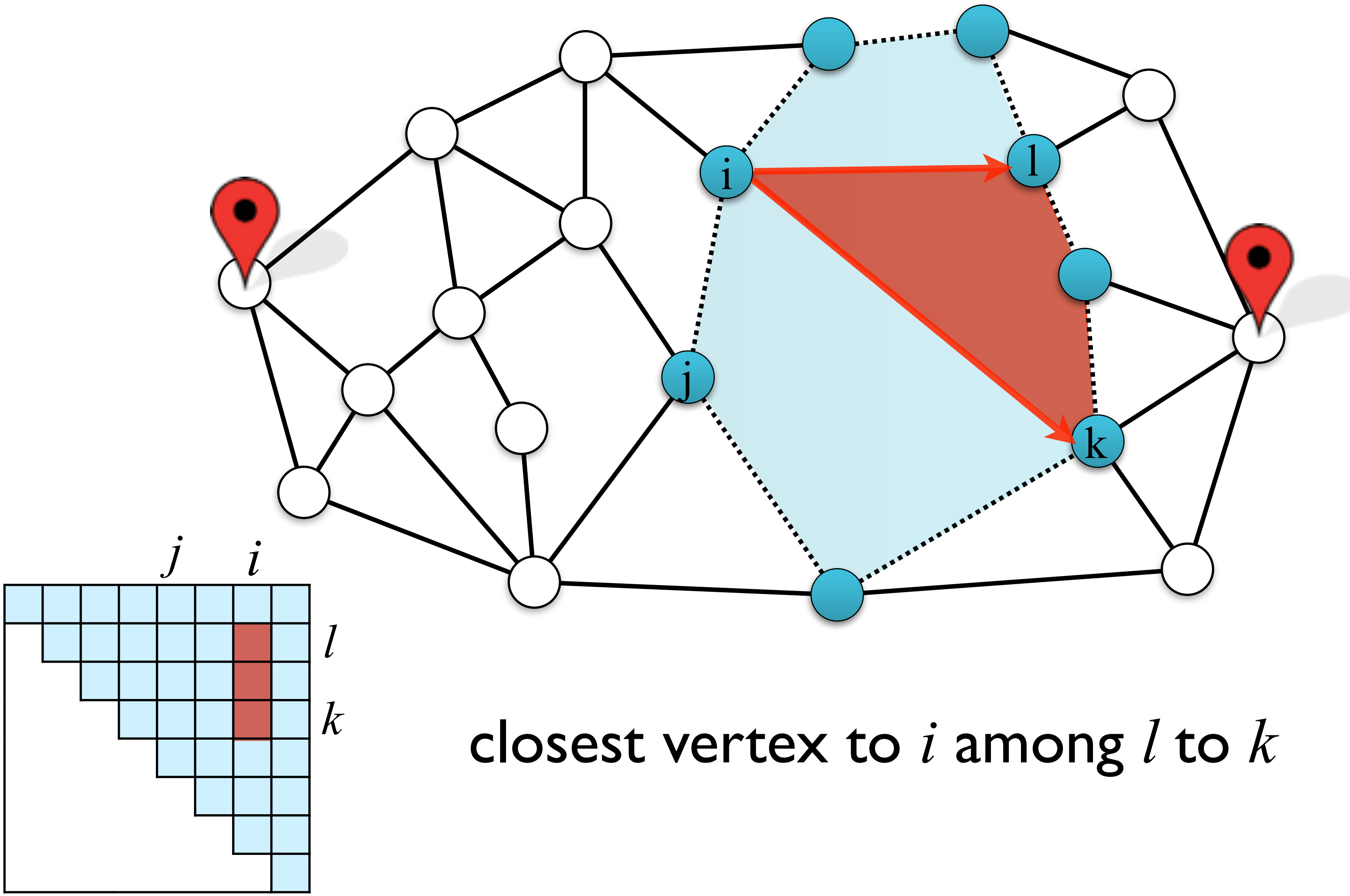
Application I
Shortest paths in planar graphs





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

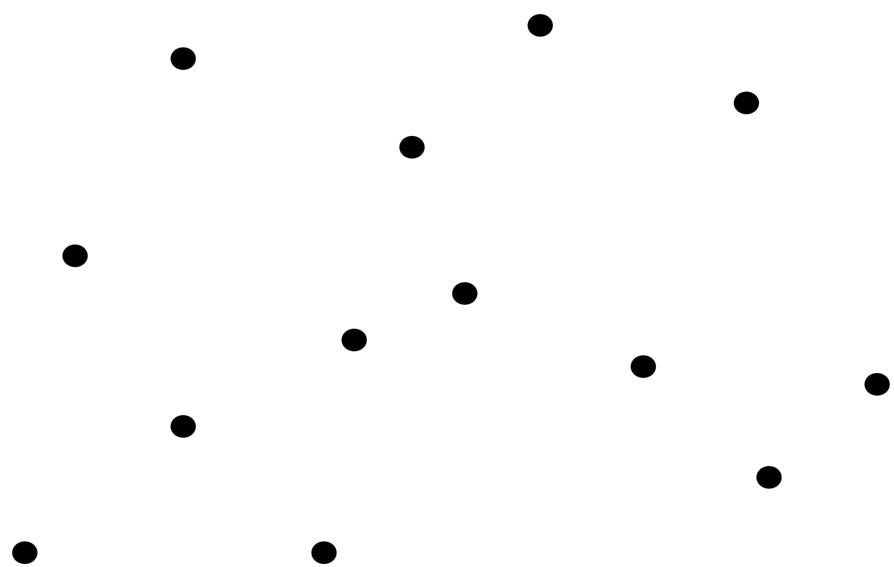
Application I
Shortest paths in planar graphs



closest vertex to i among l to k

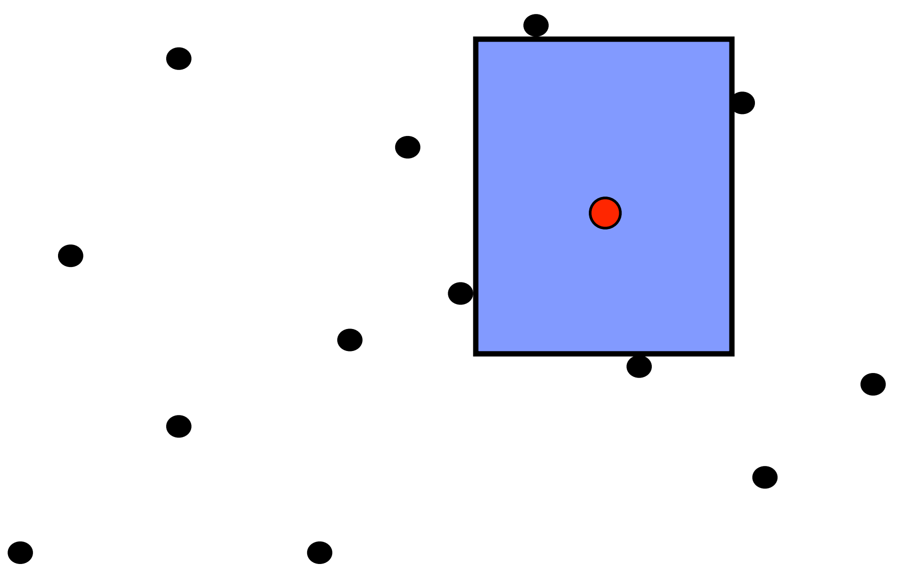
Application II: Largest empty rectangle

- Input: a set of *n* points



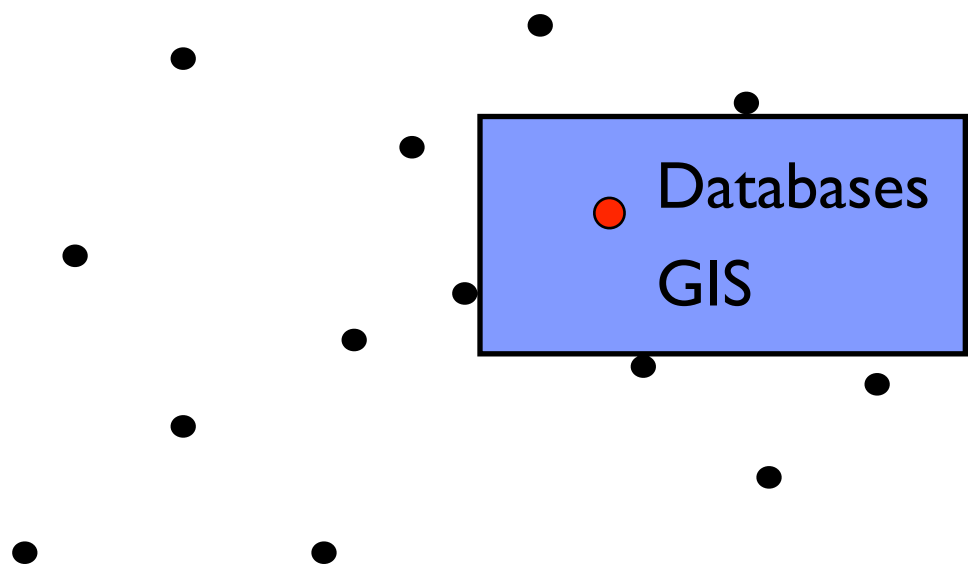
Application II: Largest empty rectangle

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

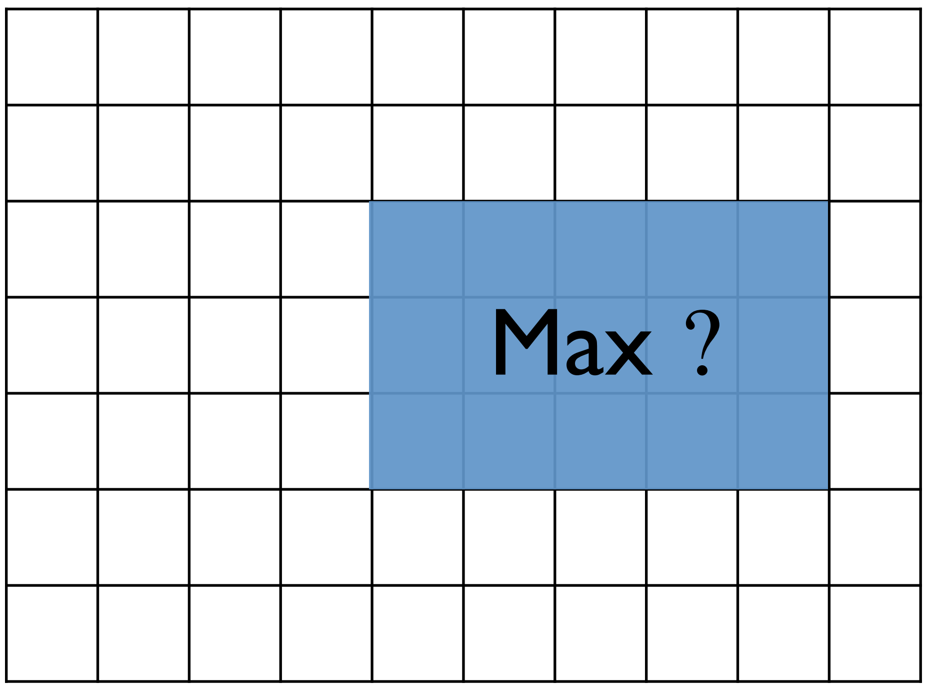


Application II:
Largest empty rectangle

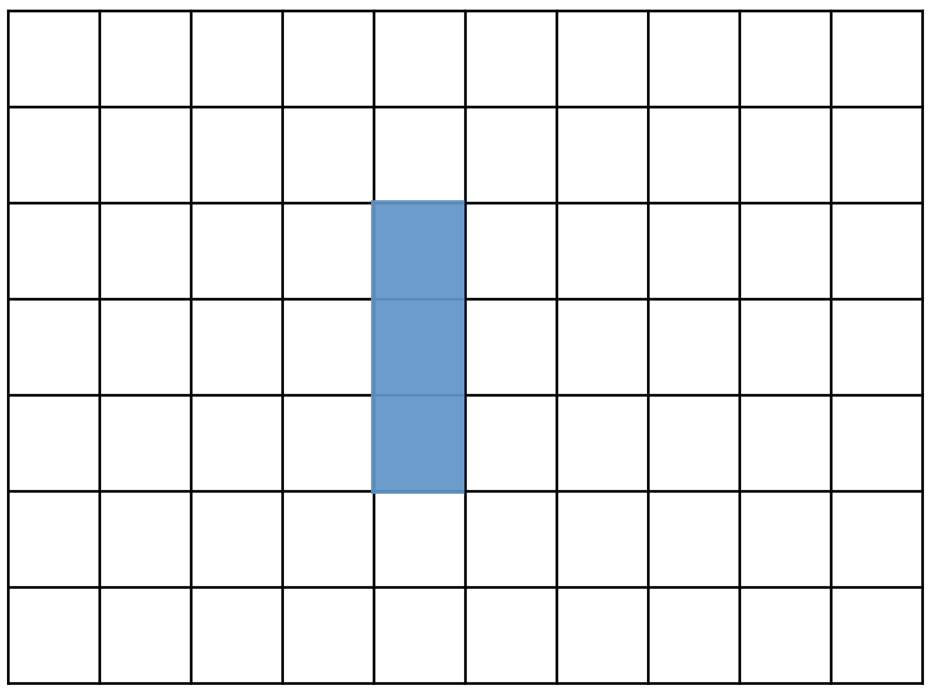
- Input: a set of *n* points
- Query: find largest empty rectangle containing a point



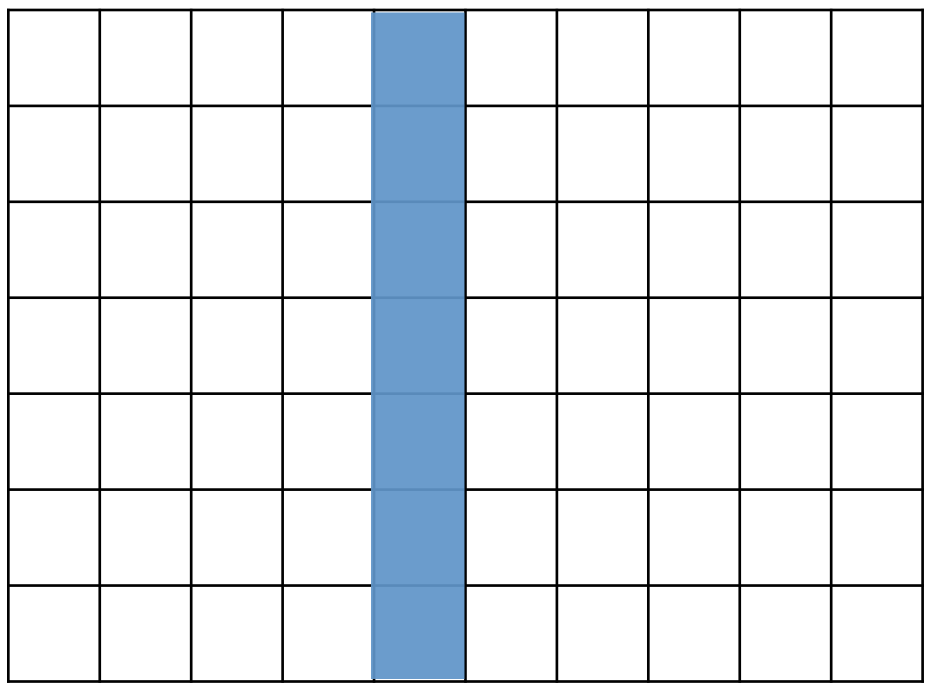
Improved Submatrix Maximum Queries in Monge Matrices



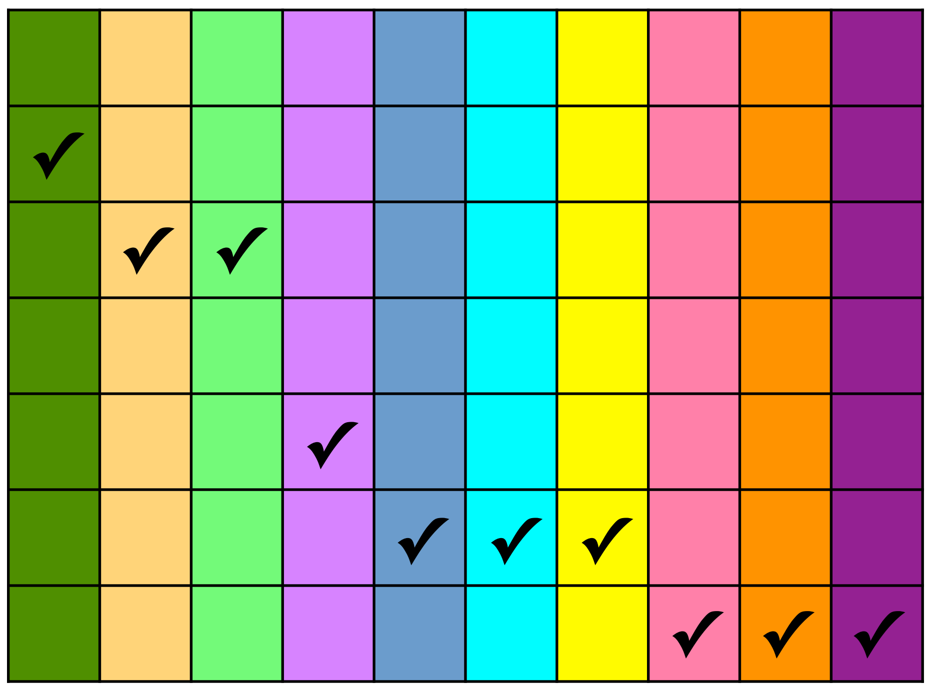
Easier: sub-column ranges



Even easier: entire-column ranges

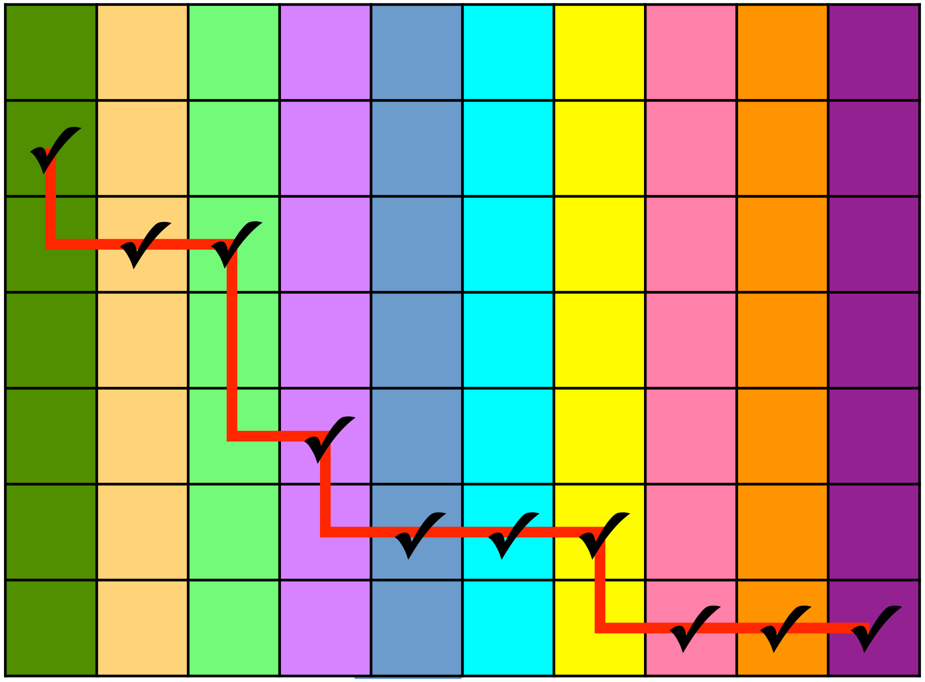


Even easier: entire-column ranges



The rows of the column maxima increase monotonically

Even easier: entire-column ranges

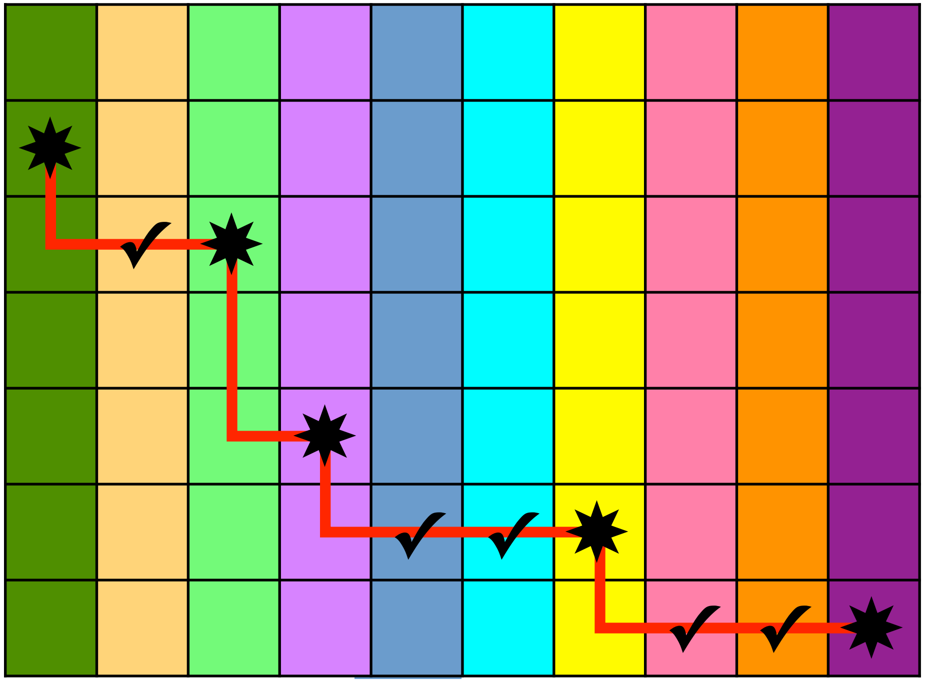


The rows of the column maxima increase monotonically

Even easier: entire-column ranges

Enough to compute list of breakpoints ✸

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

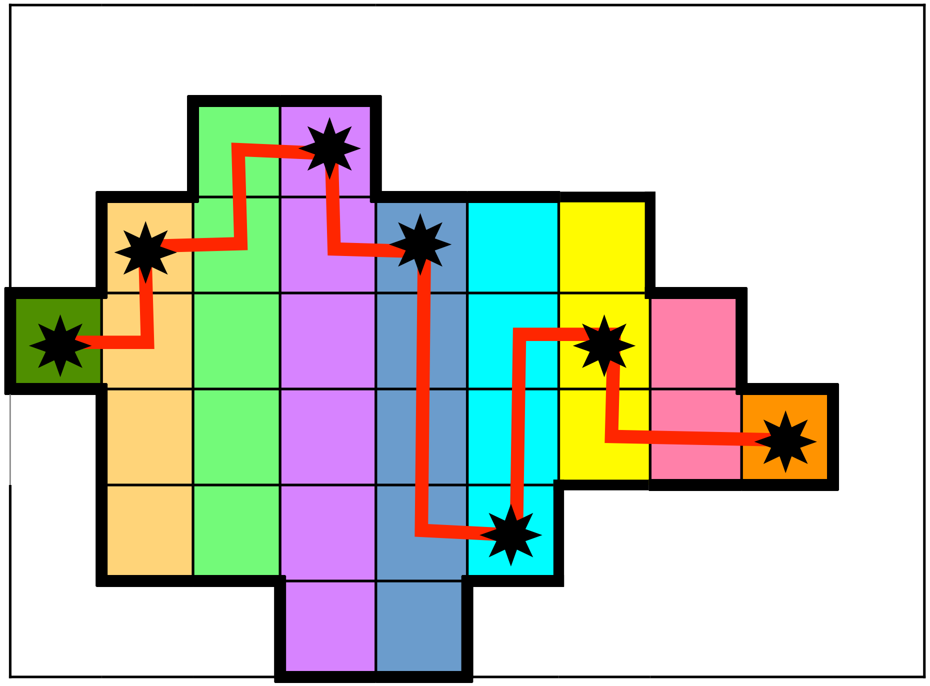


The rows of the column maxima increase monotonically

Even easier: entire-column ranges

Enough to compute list of breakpoints ✸

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



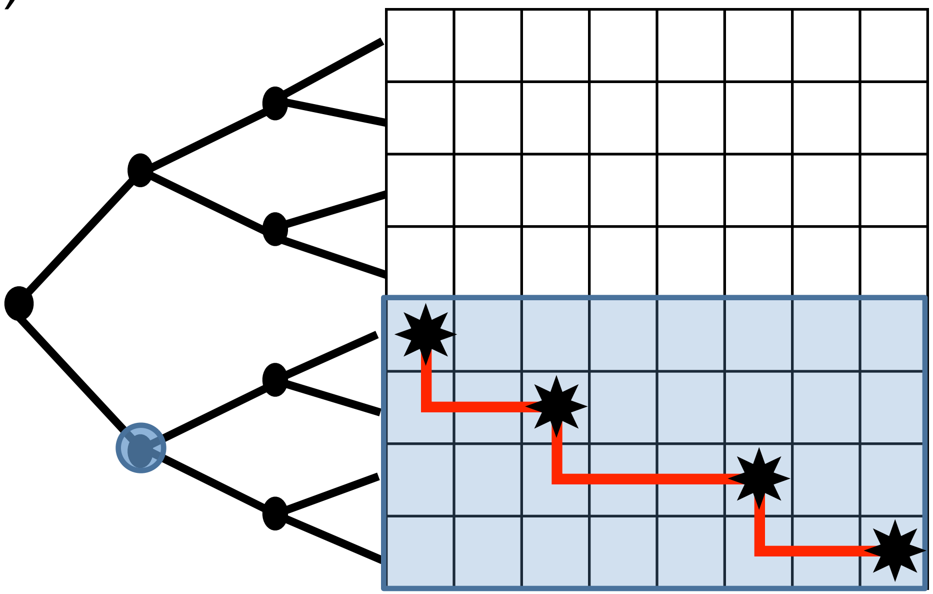
~~The rows of the column maxima increase monotonically~~

The tree of breakpoints

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

Each node computes the breakpoints of its submatrix

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

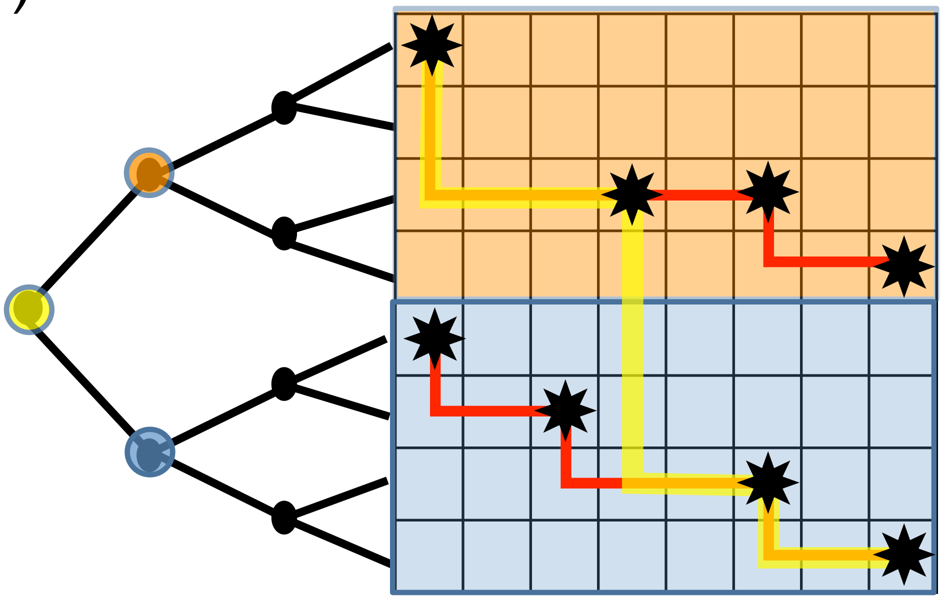


The tree of breakpoints

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

Each node computes the breakpoints of its submatrix

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



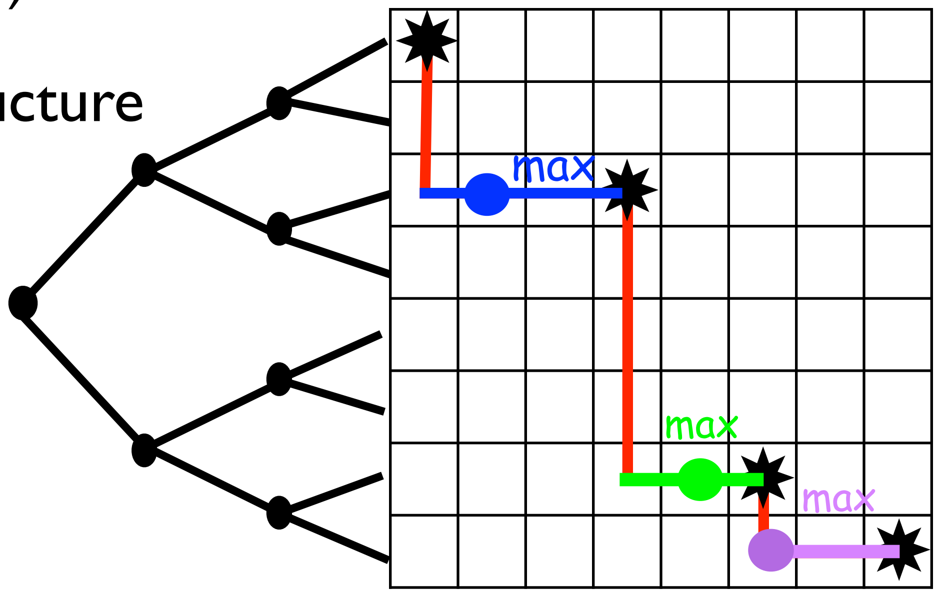
The tree of breakpoints

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

Each node computes the breakpoints of its submatrix

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

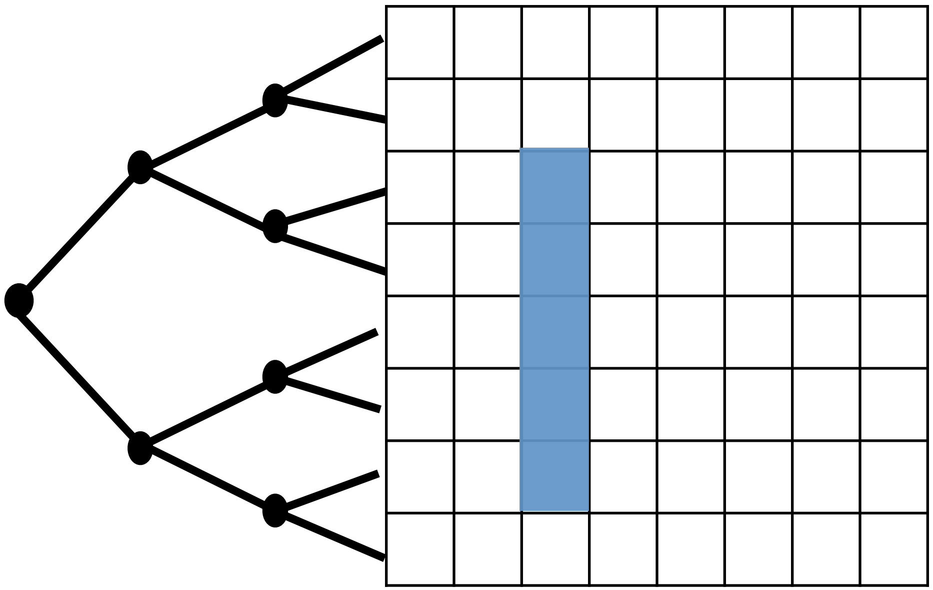
Each node stores RMQ data structure
on max's between breakpoints



The tree of breakpoints

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

A subcolumn query

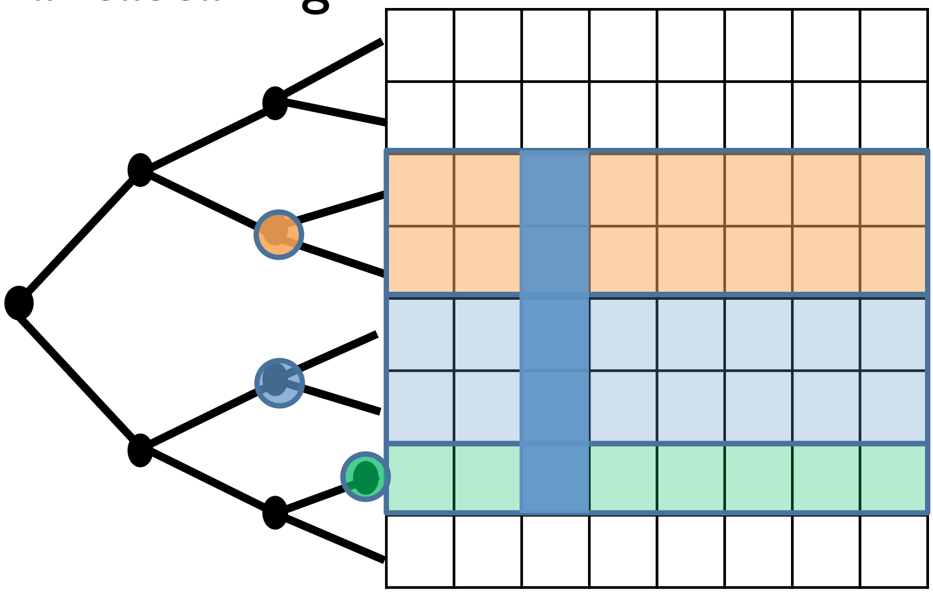


The tree of breakpoints

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

A subcolumn query is covered by $O(\log n)$ *canonical* nodes.
Search the breakpoints of each canonical node

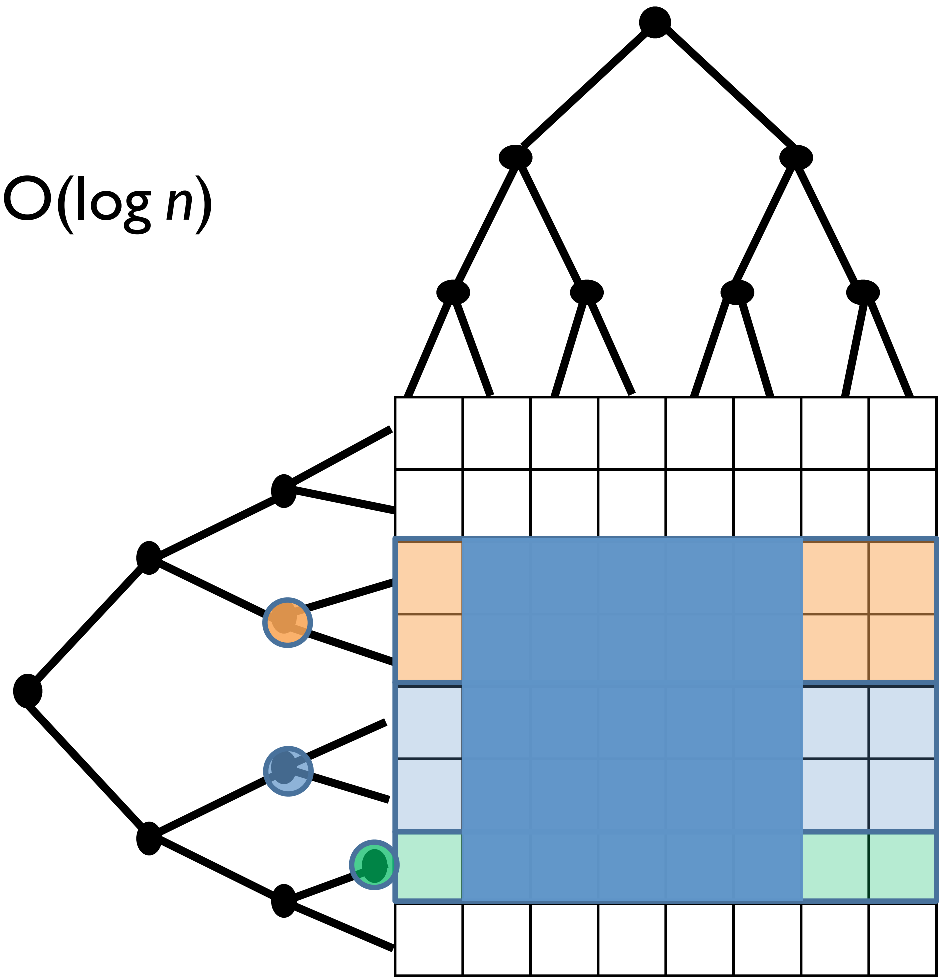
$O(\log^2 n)$ time, $O(\log n)$ via fractional cascading



The tree of breakpoints

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

A submatrix query is covered by O(log *n*) canonical nodes.



The tree of breakpoints

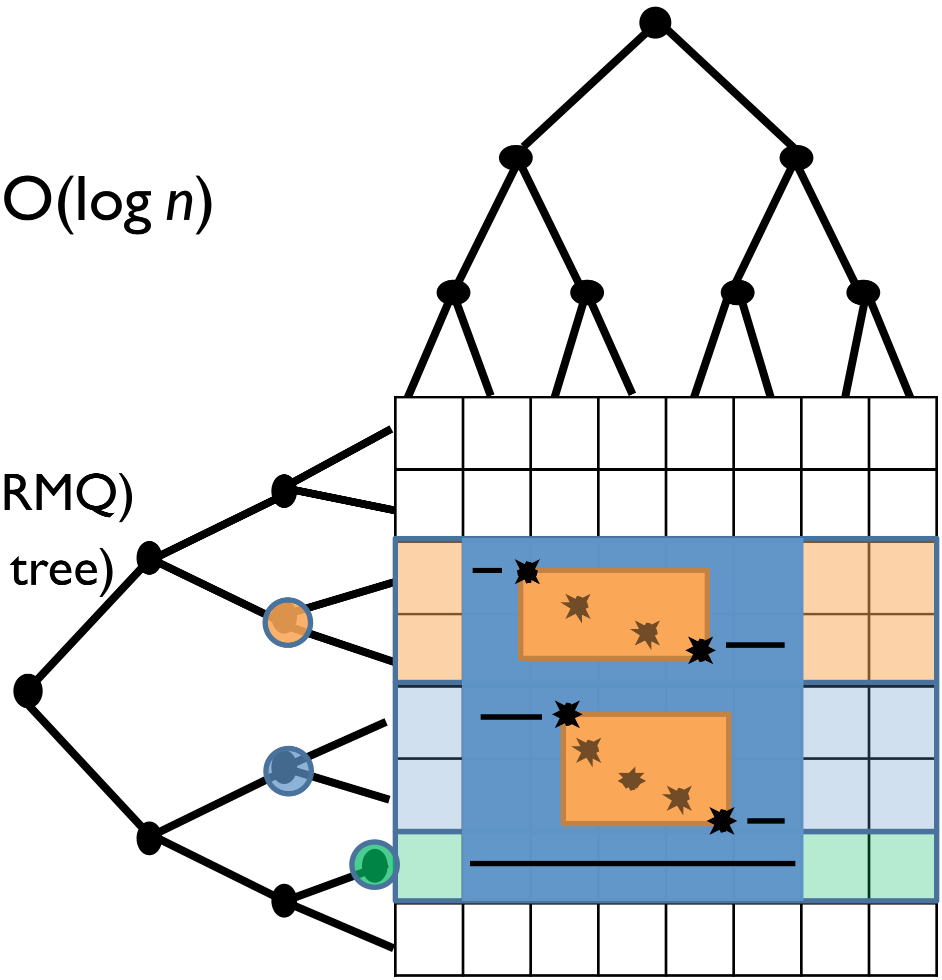
[Kaplan,Mozes,Nussbaum,Sharir SODA'12]

A submatrix query is covered by O(log *n*) canonical nodes.

The range is covered by:

- submatrices bounded by breakpoints (RMQ)
- two row intervals per submatrix (row tree)

Total query: O($\log^2 n$)
(no fractional cascading)



Improving the query-time

A submatrix query is covered by O(log *n*) canonical nodes.

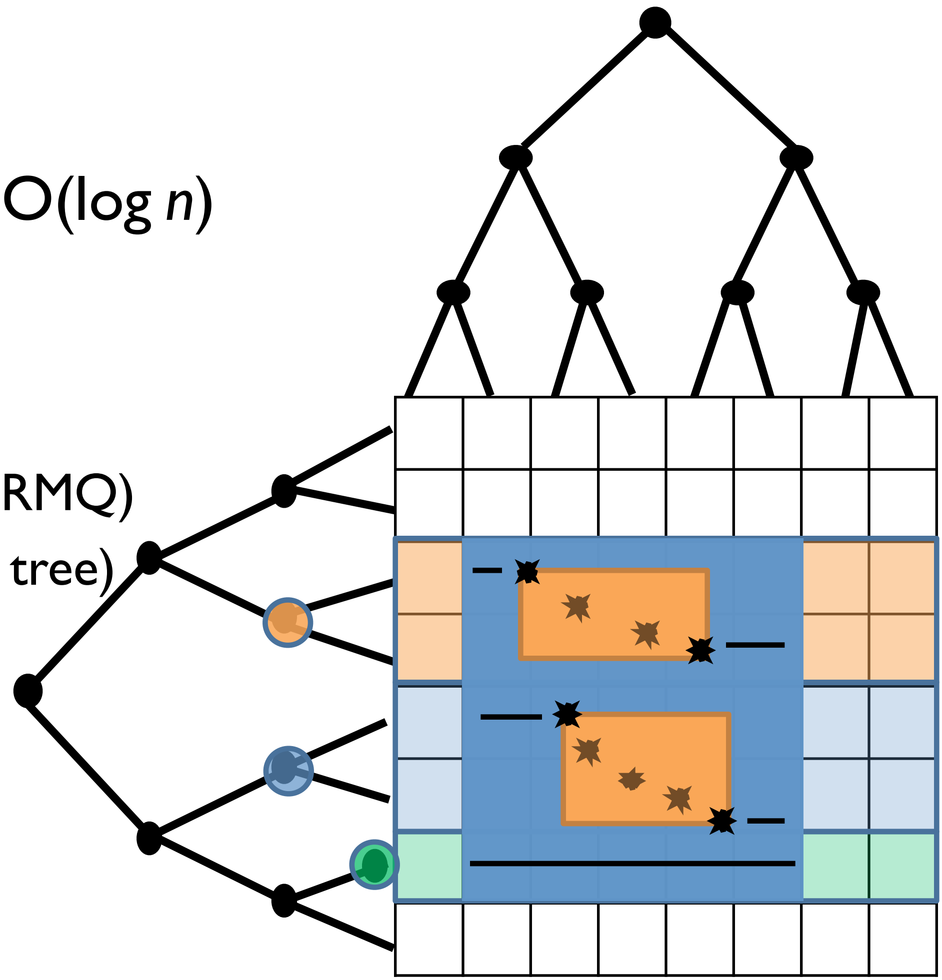
The range is covered by:

✓ - submatrices bounded by breakpoints (RMQ)

- two row intervals per submatrix (row tree)

Total query: O(log *n*)

Fractional cascading



Improving the query-time

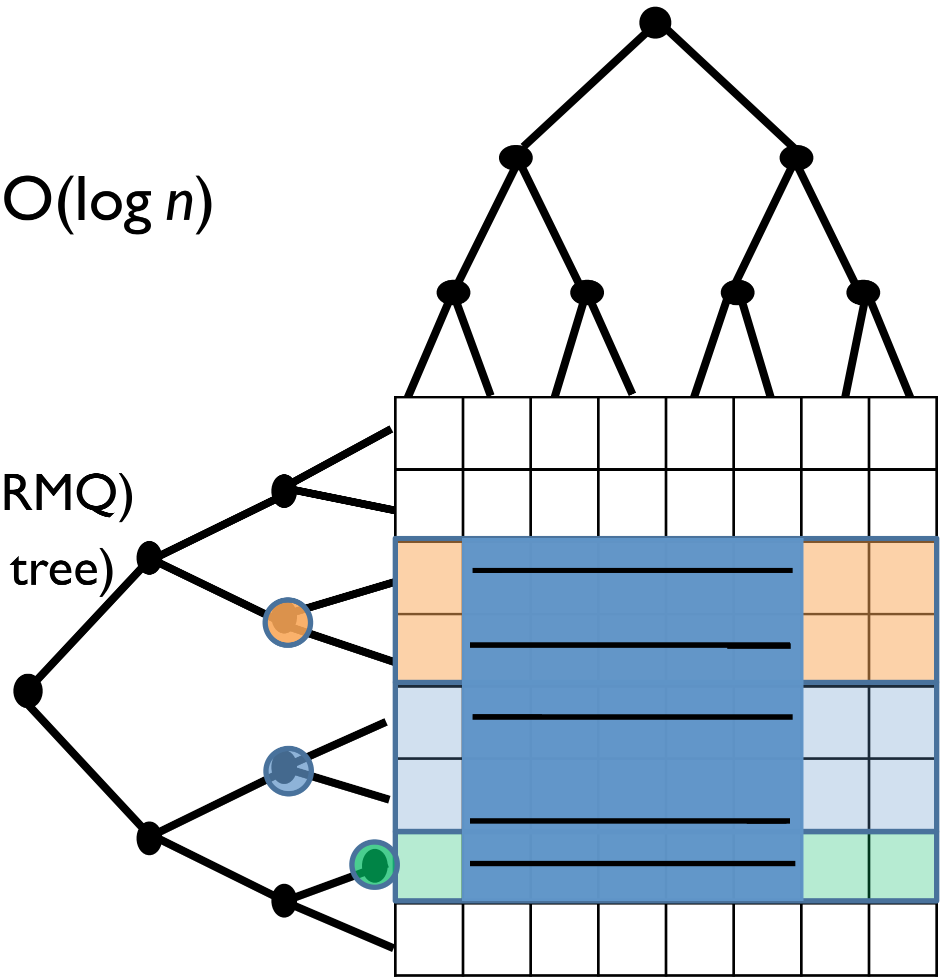
A submatrix query is covered by O(log n) canonical nodes.

The range is covered by:

✓ - submatrices bounded by breakpoints (RMQ)

- two row intervals per submatrix (row tree)

Total query: O(log n)



Improving the query-time

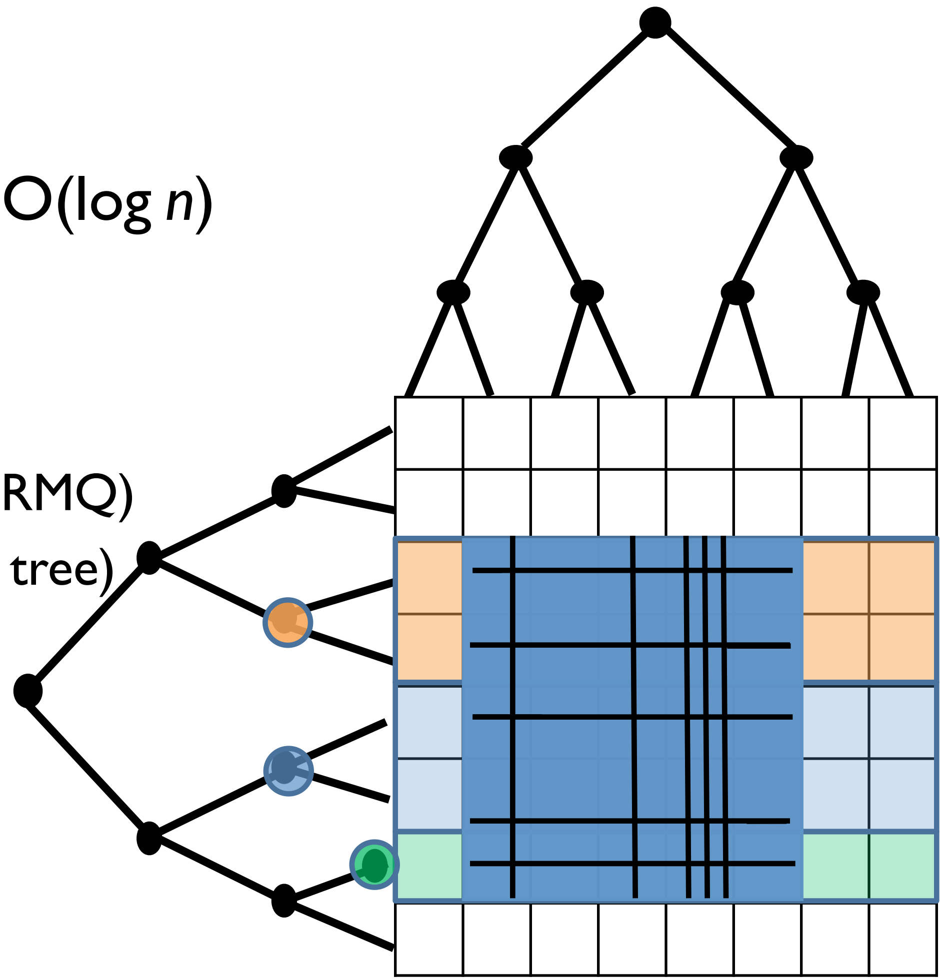
A submatrix query is covered by O(log n) canonical nodes.

The range is covered by:

✓ - submatrices bounded by breakpoints (RMQ)

- two row intervals per submatrix (row tree)

Total query: O(log n)



Improving the query-time

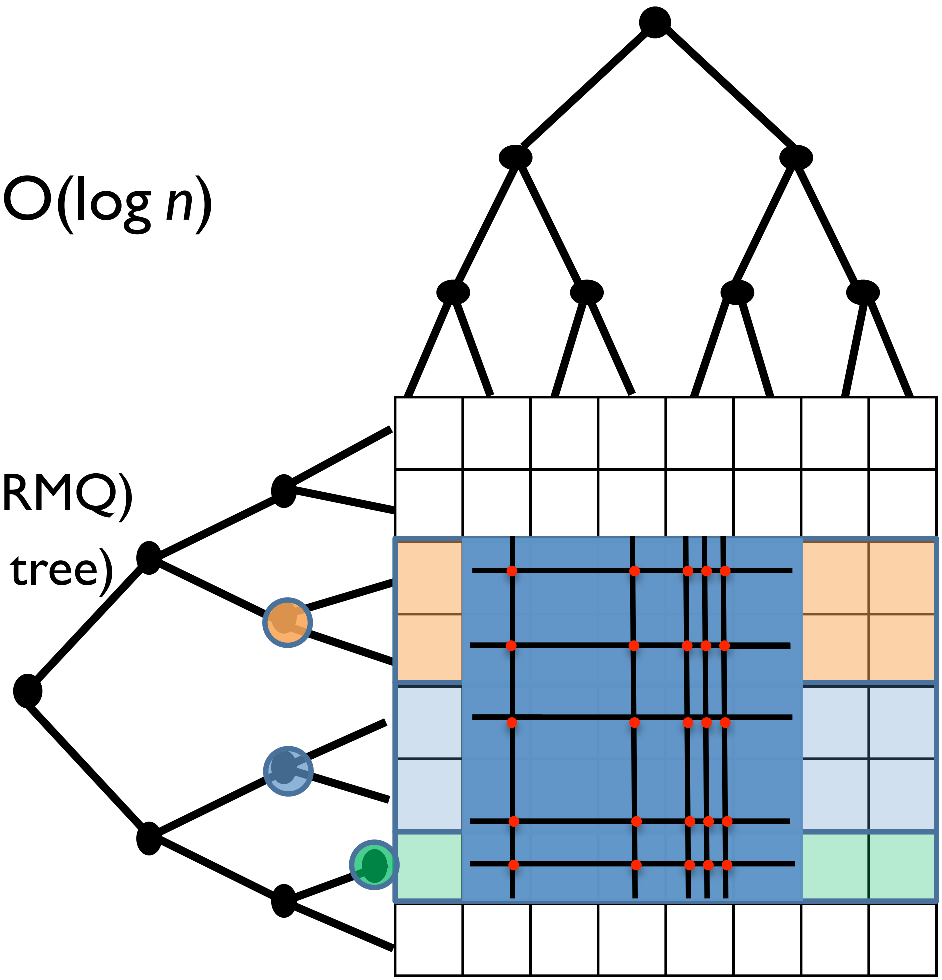
A submatrix query is covered by O(log *n*) canonical nodes.

The range is covered by:

✓ - submatrices bounded by breakpoints (RMQ)

- two row intervals per submatrix (row tree)

Total query: O(log *n*)



Improving the query-time

A submatrix query is covered by O(log n) canonical nodes.

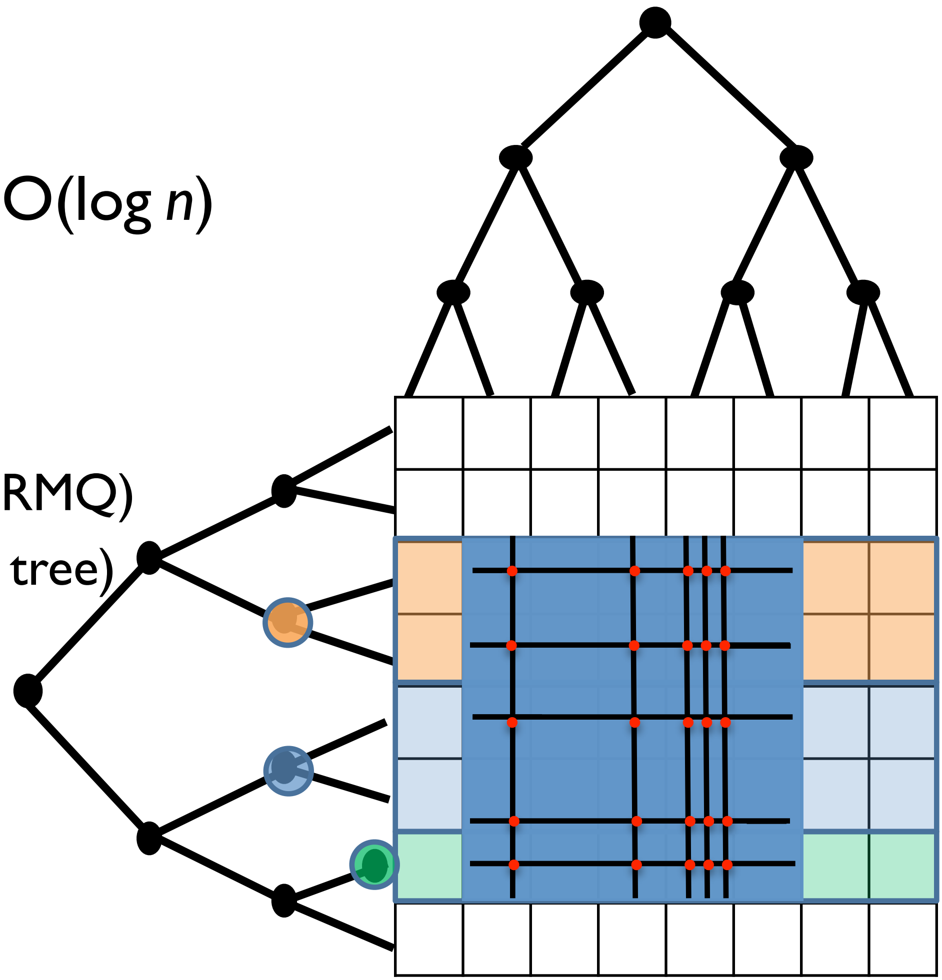
The range is covered by:

✓ - submatrices bounded by breakpoints (RMQ)

✓ - two row intervals per submatrix (row tree)

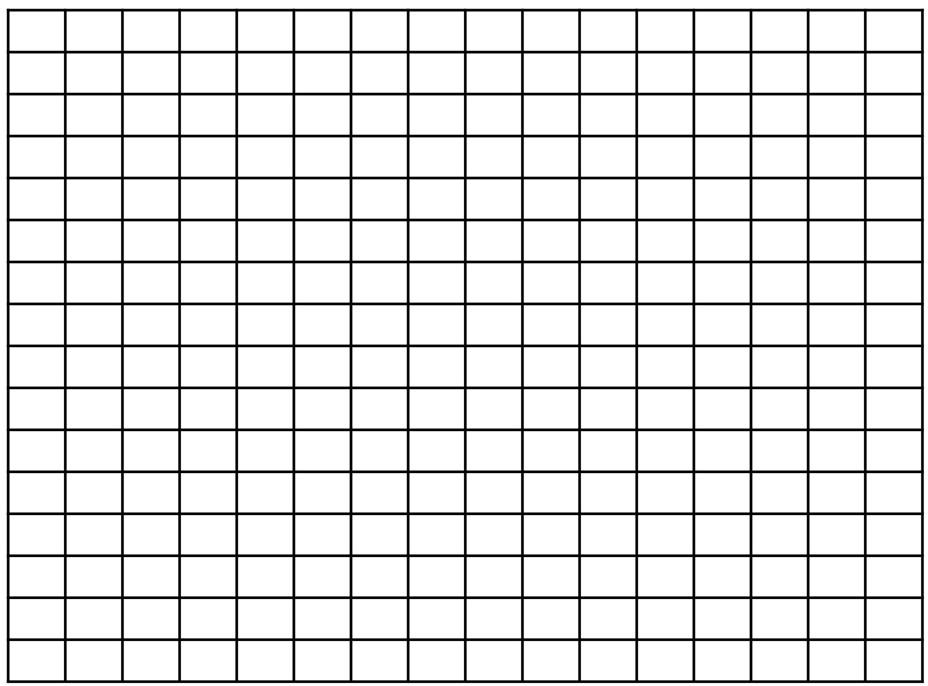
Total query: O(log n)

SMAWK



Improving the space

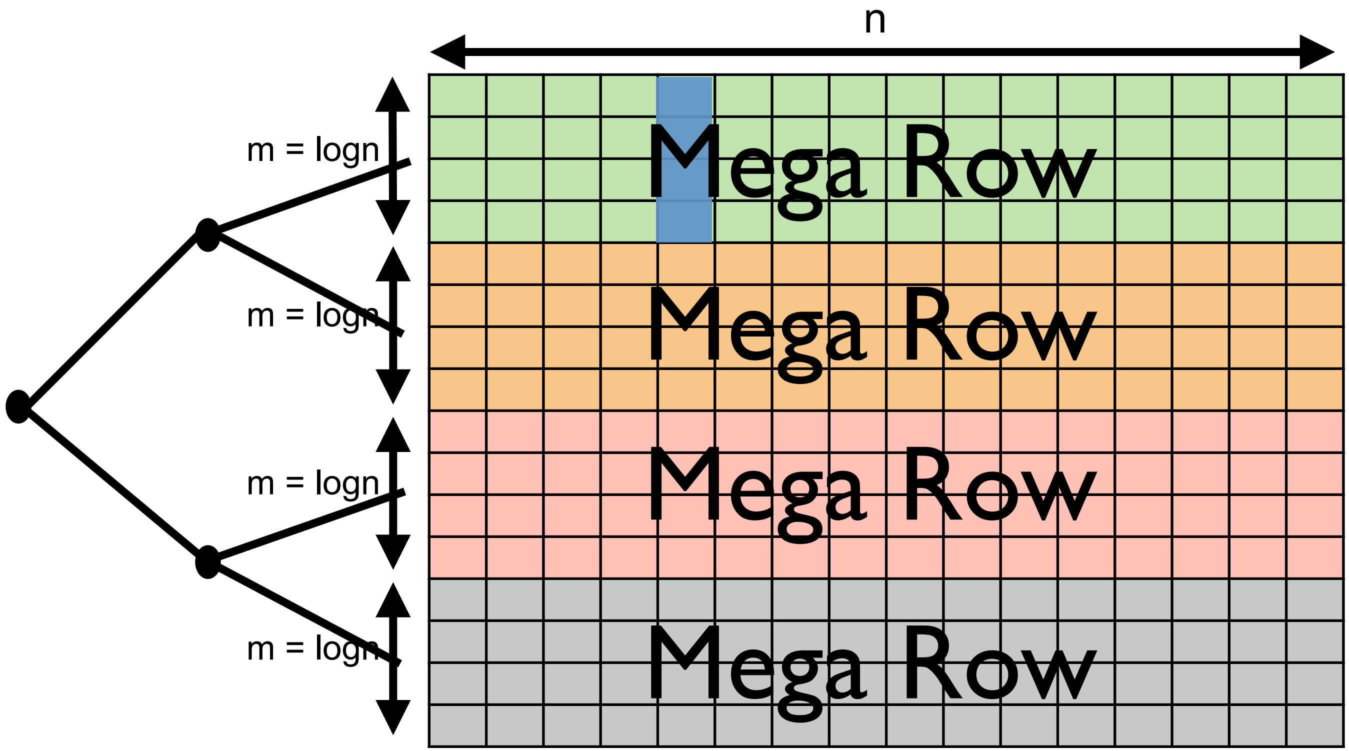
from O(n log n) to O(n)



Improving the space

<u>Theorem</u>: Given an m-by-n matrix, after O(m log n) time and O(m) space we can answer entire-column queries in O(log m) time.

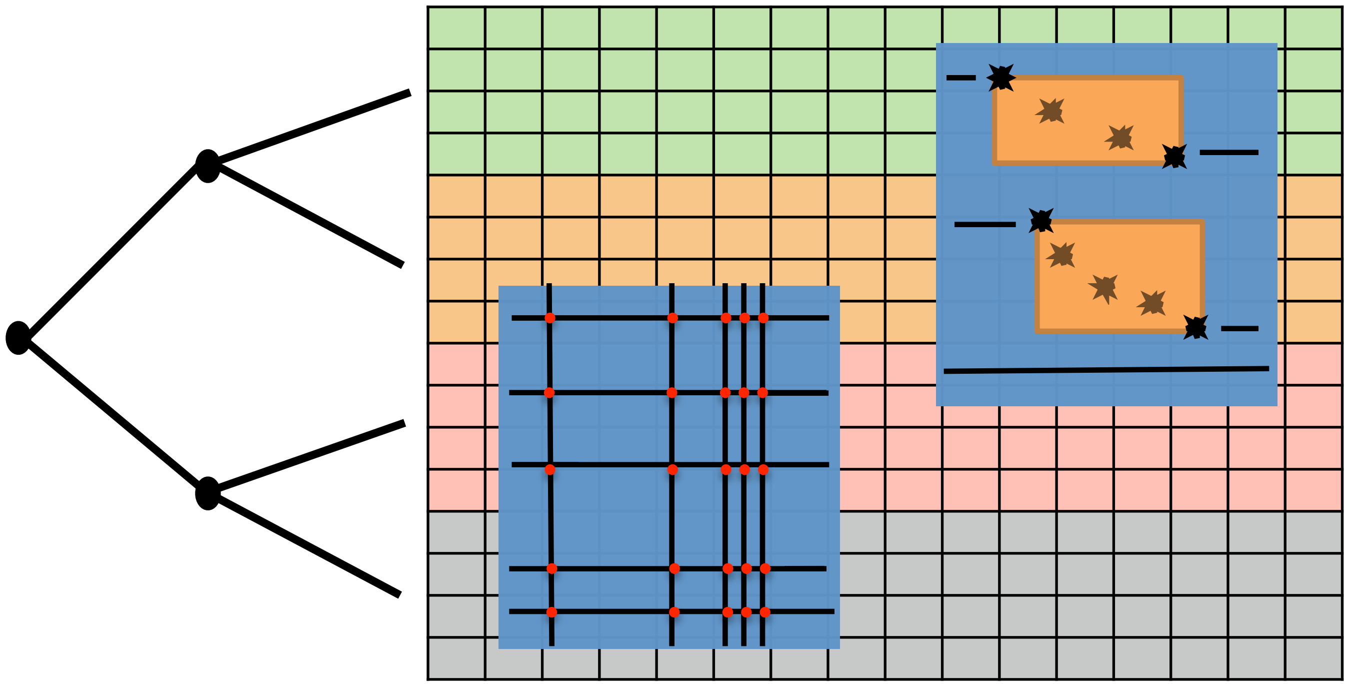
Mega Row entries fetched in O(log m) time using the above



Improving the space

\+

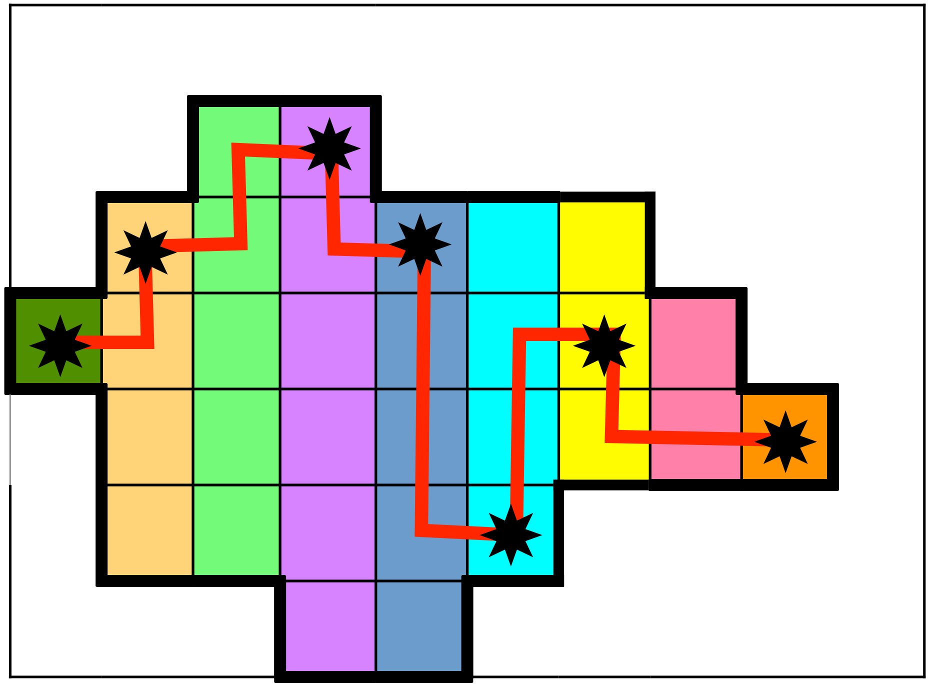
Improving the query-time



Partial Monge matrices

Partial Monge matrices

Theorem: The number of breakpoints of an m-by-n partial matrix is O(m).



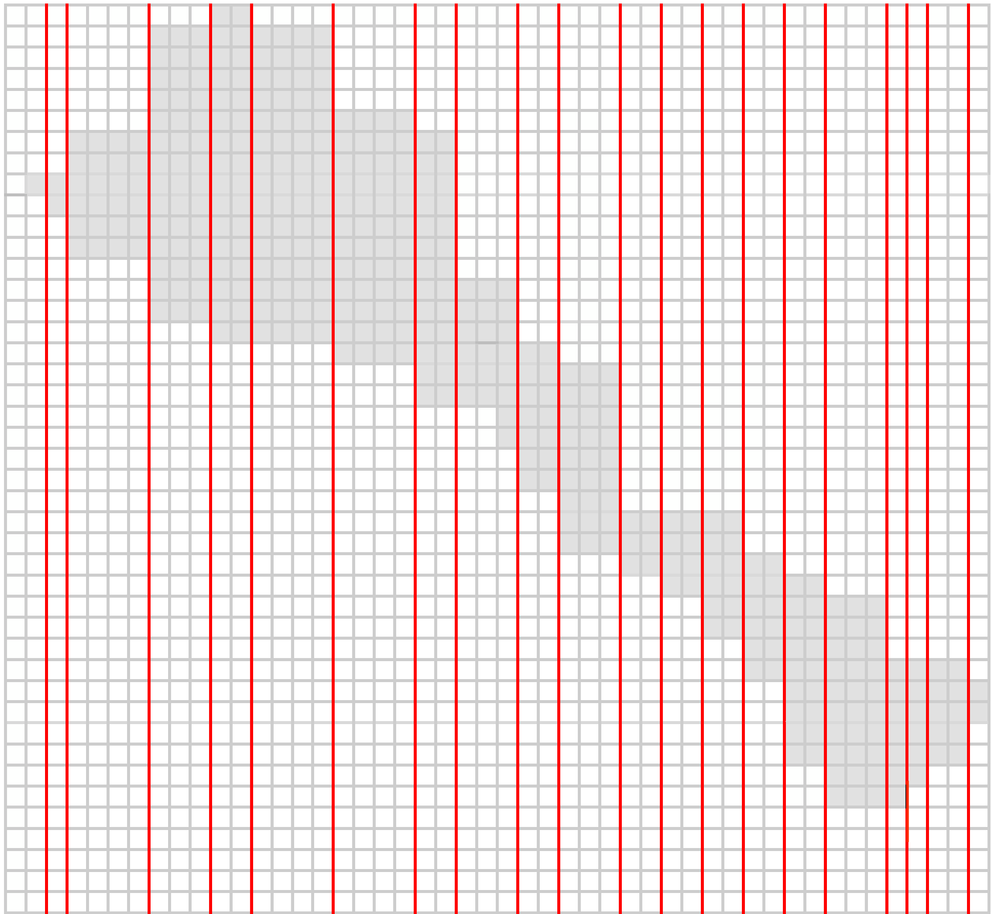
~~The rows of the column maxima increases monotonically~~

Partial Monge matrices

Theorem: The number of breakpoints of an m-by-n partial matrix is O(m).

Partial Monge matrices

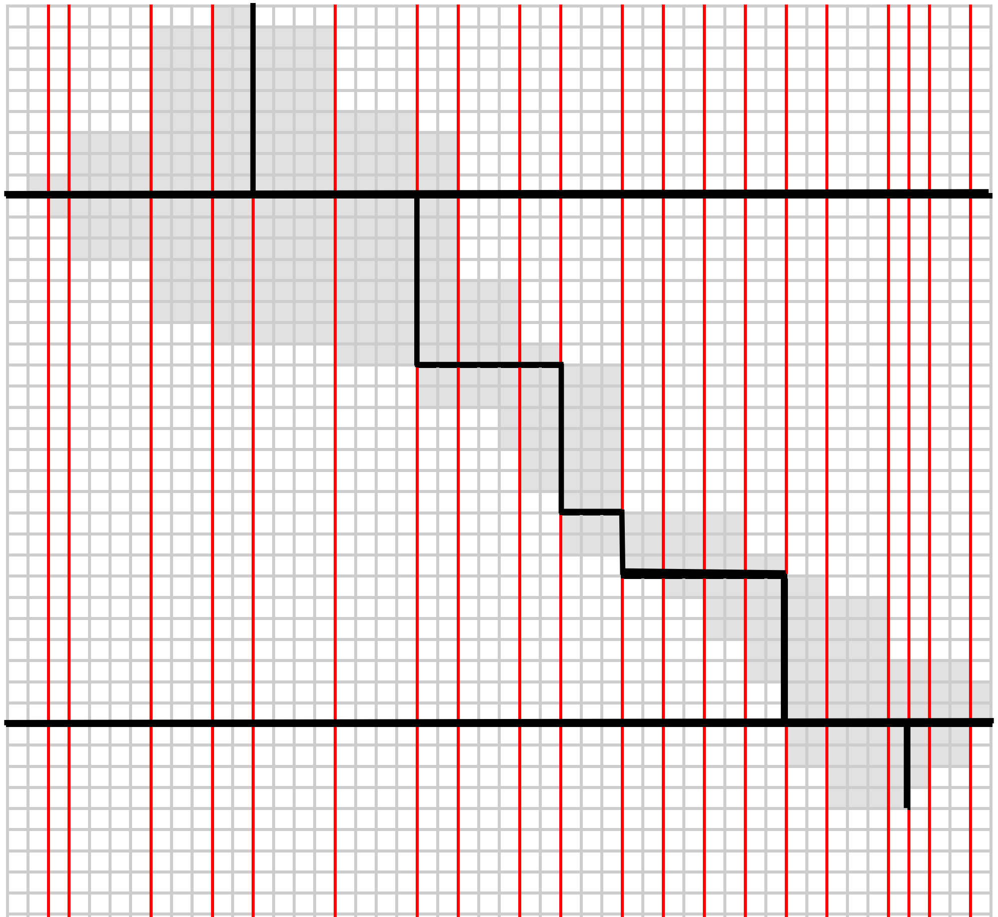
Theorem: The number of breakpoints of an m-by-n partial matrix is O(m).



Partial Monge matrices

Theorem: The number of breakpoints of an m-by-n partial matrix is O(m).

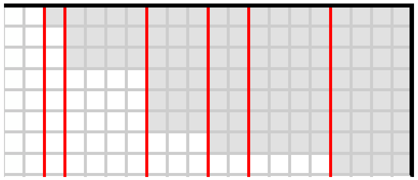
Each row appears in at most three staircase matrices



Partial Monge matrices

Theorem: The number of breakpoints of an m-by-n partial matrix is O(m).

Each row appears in at most three staircase matrices



Open Problems

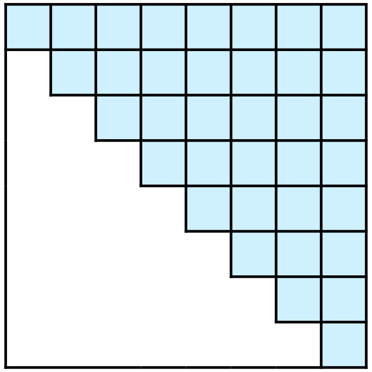
Open Problems

- 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

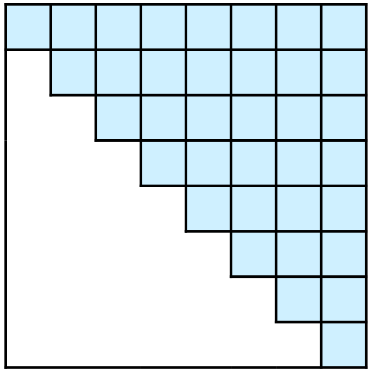
$O(\log^2 n)$ • activate a row and add k to all its entries

$O(\log^2 n)$ • delete column

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

[Fakcharoenphol Rao, 2006]

m–by–n staircase

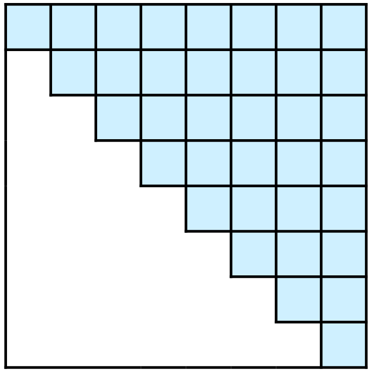


Open Problems

- 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

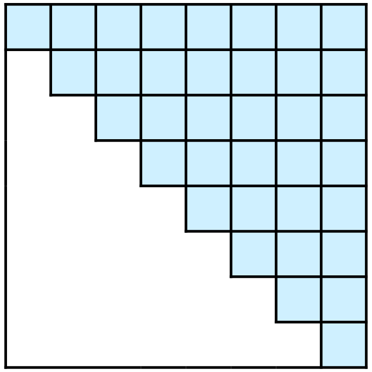


- Find the O(m) breakpoints in linear time

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]
  - $O(m\log n)$ [Here]

Thank You!