The Knuth-Yao Quadrangle Inequality Speedup is a Consequence of Total Monotonicity

Wolfgang W. Bein (University of Nevada)
Mordecai J. Golin (Hong Kong UST)
Lawrence L. Larmore (University of Nevada)
Yan Zhang (Hong Kong UST)
Motivation

- Nothing new: material here goes back 20-30 years.
- There are two classic Dynamic Programming Speedups in the literature
  - Knuth-Yao Quadrangle Inequality Speedup
  - SMAWK Algorithm for Totally Monotone Matrices
- They “feel” similar. Are they related?
- Both techniques have been used quite often in improving DP algorithms for various type of constrained source coding.
Outline

- **Background**
  - Kunth-Yao (KY) Quadrangle Inequality (QI) Speedup
  - SMAWK Algorithm for finding Row Minima of Totally Monotone (TM) Matrices

- **The $D^d$ Decomposition**
  A transformation from QI to TM such that SMAWK solves KY problem as quickly as KY.

- **The $L^m$ and $R^m$ Decompositions**
  Another transformation from QI to TM that (1) implies KY speedup and (2) enables online solution.
Outline

- Background
  - Kunth-Yao (KY) Quadrangle Inequality (QI) Speedup
  - SMAWK Algorithm for finding Row Minima of Totally Monotone (TM) Matrices

- The $D^d$ Decomposition
  A transformation from QI to TM such that SMAWK solves KY problem as quickly as KY.

- The $L^m$ and $R^m$ Decompositions
  Another transformation from QI to TM that (1) implies KY speedup and (2) enables online solution.
Background

- Kunth-Yao Quadrangle Inequality Speedup
  - $\Theta(n)$ speedup: $O(n^3)$ down to $O(n^2)$
Background

- Kunth-Yao Quadrangle Inequality Speedup
  - $\Theta(n)$ speedup: $O(n^3)$ down to $O(n^2)$

- SMAWK Algorithm for finding Row Minima of Totally Monotone Matrices
  - $\Theta(n)$ speedup: $O(n^2)$ down to $O(n)$
Background

- Kunth-Yao Quadrangle Inequality Speedup
  - $\Theta(n)$ speedup: $O(n^3)$ down to $O(n^2)$

- SMAWK Algorithm for finding Row Minima of Totally Monotone Matrices
  - $\Theta(n)$ speedup: $O(n^2)$ down to $O(n)$

- How are the two techniques related?
Quadrangle Inequality
Quadrangle Inequality

Original Motivation
Computing Optimal Binary Search Trees (Optimal BST)
[Gilbert and Moore (1959)]
Quadrangle Inequality

- **Original Motivation**
  Computing *Optimal Binary Search Trees (Optimal BST)*
  [Gilbert and Moore (1959)]

- **Optimal BST**
  - Construct a search tree for $n$ keys
  - $n$ internal nodes corresponds to successful search
  - $n + 1$ external nodes corresponds to unsuccessful search
  - Minimize the expected number of comparisons
Quadrangle Inequality

Original Motivation
Computing Optimal Binary Search Trees (Optimal BST)
[Gilbert and Moore (1959)]

Optimal BST

- Construct a search tree for $n$ keys
- $n$ internal nodes corresponds to successful search
- $n + 1$ external nodes corresponds to unsuccessful search
- Minimize the expected number of comparisons

Solution: Dynamic Programming

$$B_{i,j} = \begin{cases} 
    w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} & (i < j) \\
    0 & (i = j)
\end{cases}$$

for some $w(i, j)$ that can be computed in $O(1)$ time.
Quadrangle Inequality

Standard Calculation

\[ B_{i,j} = \begin{cases} 
    w(i, j) + \min_{i < t \leq j} \{B_{i,t-1} + B_{t,j}\} & (i < j) \\
    0 & (i = j) 
\end{cases} \]
Quadrangle Inequality

Standard Calculation

\[ B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} & (i < j) \\
  0 & (i = j) 
\end{cases} \]

Diagonal by diagonal
Quadrangle Inequality

Standard Calculation

\[ B_{i, j} = \begin{cases} 
  w(i, j) + \min_{i < t \leq j} \{ B_{i, t-1} + B_{t, j} \} & (i < j) \\
  0 & (i = j) 
\end{cases} \]

- Diagonal by diagonal
- An example:

\[ n = 6 \]
# Quadrangle Inequality

## Standard Calculation

\[ B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i \leq t \leq j} \{ B_{i,t-1} + B_{t,j} \} & (i < j) \\
  0 & (i = j) 
\end{cases} \]

- **Diagonal by diagonal**

- **An example:**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Quadrangle Inequality

\[ B_{i,j} = \begin{cases} 
    w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} & \text{if } i < j \\
    0 & \text{if } i = j
\end{cases} \]

Diagonal by diagonal

An example:

\[ n = 6 \]

\[ \begin{array}{ccccccc}
   & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
0 & 0 & 230 & & & & & \\
1 & 0 & 146 & & & & & \\
2 & 0 & 75 & & & & & \\
3 & 0 & 43 & & & & & \\
4 & 0 & 44 & & & & & \\
5 & 0 & 52 & & & & & \\
6 & & & & & & & 0
\end{array} \]
Quadrangle Inequality

Standard Calculation

\[ B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} & (i < j) \\
  0 & (i = j)
\end{cases} \]

- Diagonal by diagonal
- An example:

\[ n = 6 \]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>230</td>
<td>433</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>146</td>
<td>260</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>75</td>
<td>141</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>43</td>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>44</td>
<td>121</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quadrangle Inequality

Standard Calculation

\[ B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} & (i < j) \\
  0 & (i = j)
\end{cases} \]

Diagonal by diagonal

An example:

\[ n = 6 \]

\[
\begin{array}{ccccccc}
  & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
 0 & 0 & 230 & 433 & \text{586} & & & \\
 1 & & 0 & 146 & 260 & \text{349} & & \\
 2 & & & 0 & 75 & 141 & \text{250} & \\
 3 & & & & 0 & 43 & 119 & 204 \\
 4 & & & & & 0 & 44 & 121 \\
 5 & & & & & & 0 & 52 \\
 6 & & & & & & & 0 \\
\end{array}
\]
Quadrangle Inequality

Standard Calculation

\[ B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} & (i < j) \\
  0 & (i = j)
\end{cases} \]

Diagonal by diagonal

An example:

\[ n = 6 \]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>230</td>
<td>433</td>
<td>586</td>
<td>698</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>146</td>
<td>260</td>
<td>349</td>
<td>491</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
<td>75</td>
<td>141</td>
<td>250</td>
<td>357</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td></td>
<td>43</td>
<td>119</td>
<td>204</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
<td>44</td>
<td>121</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
**Quadrangle Inequality**

- **Standard Calculation**

\[
B_{i,j} = \begin{cases} 
    w(i,j) + \min_{i < t \leq j} \{B_{i,t-1} + B_{t,j}\} & (i < j) \\
    0 & (i = j)
\end{cases}
\]

- **Diagonal by diagonal**

- **An example:**

\[
\begin{array}{ccccccc}
  & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
0 & 0 & 230 & 433 & 586 & 698 & 862 & \\
1 & 0 & 146 & 260 & 349 & 491 & 624 & \\
2 & 0 & 75 & 141 & 250 & 357 & \\
3 & 0 & 43 & 119 & 204 & \\
4 & 0 & 44 & 121 & \\
5 & 0 & 52 & \\
6 & 0 & 
\end{array}
\]
## Quadrangle Inequality

### Standard Calculation

\[
B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} & \text{for } i < j \\
  0 & \text{for } i = j
\end{cases}
\]

### Diagonal by diagonal

### An example:

\(n = 6\)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>230</td>
<td>433</td>
<td>586</td>
<td>698</td>
<td>862</td>
<td>1002</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>146</td>
<td>260</td>
<td>349</td>
<td>491</td>
<td>624</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>75</td>
<td>141</td>
<td>250</td>
<td>357</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>43</td>
<td>119</td>
<td>204</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>44</td>
<td>121</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Quadrangle-Inequality and Total-Monotonicity – p.7/30
Quadrangle Inequality

Standard Calculation

\[ B_{i,j} = \begin{cases} \quad w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} & (i < j) \\ \quad 0 & (i = j) \end{cases} \]

Diagonal by diagonal

An example:

\[ n = 6 \]

Running time:

\[ O(n^3) \]
Quadrangle Inequality

Speedup: $O(n^3) \rightarrow O(n^2)$  [Knuth (1971)]

$$B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i < t \leq j} \{B_{i,t-1} + B_{t,j}\} & (i < j) \\
  0 & (i = j)
\end{cases}$$
Quadrangle Inequality

- **Speedup:** $O(n^3) \rightarrow O(n^2)$ [Knuth (1971)]

$$B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i < t \leq j} \{B_{i,t-1} + B_{t,j}\} & (i < j) \\
  0 & (i = j)
\end{cases}$$

$K_B(i, j)$ the index $t$ that achieves the minimum.
Quadrangle Inequality

Speedup: $O(n^3) \rightarrow O(n^2)$ [Knuth (1971)]

$$B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i<t\leq j}\{B_{i,t-1} + B_{t,j}\} & (i < j) \\
  0 & (i = j)
\end{cases}$$

$K_B(i, j)$ the index $t$ that achieves the minimum.

Theorem in [Knuth (1971)]

$$K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$$
Quadrangle Inequality

Speedup: \( O(n^3) \rightarrow O(n^2) \) [Knuth (1971)]

\[
B_{i,j} = \begin{cases} 
  w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} & (i < j) \\
  0 & (i = j)
\end{cases}
\]

\( K_B(i, j) \) the index \( t \) that achieves the minimum.

Theorem in [Knuth (1971)]

\[
K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)
\]

<table>
<thead>
<tr>
<th></th>
<th>( j - 1 )</th>
<th>( j )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i )</td>
<td>( K_B(i, j - 1) )</td>
<td>( K_B(i, j) )</td>
</tr>
<tr>
<td>( i + 1 )</td>
<td>( K_B(i + 1, j) )</td>
<td></td>
</tr>
</tbody>
</table>
Quadrangle Inequality

Speedup: $K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$
Quadrangle Inequality

- Speedup: $K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$
- The index table
Quadrangle Inequality

- Speedup: $K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$
- The index table

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Quadrangle Inequality

- **Speedup:** \( K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j) \)

- **The index table**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quadrangle Inequality

- **Speedup:** $K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$

- **The index table**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quadrangle Inequality

- Speedup: $K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$
- The index table

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quadrangle Inequality

- Speedup: $K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$

- The index table

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quadrangle Inequality

- Speedup: $K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$
- The index table

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quadrangle Inequality

- **Speedup:** \( K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j) \)

- The index table

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Quadrangle Inequality

- **Speedup:** \( K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j) \)
- **The index table**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quadrangle Inequality

- **Speedup:** \( K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j) \)

- **The index table**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Quadrangle Inequality**

- **Speedup:** \( K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j) \)
- **The index table**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Quadrangle Inequality**

- Speedup: $K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$
- The index table

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quadrangle Inequality

- Speedup: $K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j)$
- The index table

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Running time: $O(n^3)$ down to $O(n^2)$
Quadrangle Inequality
Quadrangle Inequality

Definition [Yao (1980, 1982)]
Quadrangle Inequality

Definition [Yao (1980, 1982)]

Function \( f(i, j), (0 \leq i \leq j \leq n) \)
satisfies a Quadrangle Inequality (QI), if \( \forall i \leq i' \leq j \leq j' \)

\[
f(i, j) + f(i', j') \leq f(i', j) + f(i, j')
\]
Quadrangle Inequality

Definition [Yao (1980, 1982)]

Function \( f(i, j), (0 \leq i \leq j \leq n) \)
satisfies a Quadrangle Inequality (QI), if \( \forall i \leq i' \leq j \leq j' \)

\[
f(i, j) + f(i', j') \leq f(i', j) + f(i, j')
\]
Speedup using Quadrangle Inequality

\[ B_{i,j} = w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} \]
Speedup using Quadrangle Inequality

\[ B_{i,j} = w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} \]

- Lemmas from [Yao (1980)]
Speedup using Quadrangle Inequality

\[ B_{i,j} = w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} \]

Lemmas from [Yao (1980)]

(A) If \( w(i, j) \) satisfies QI (and some additional constraints),
\[ \Rightarrow B_{i,j} \text{ satisfies QI.} \]
Speedup using Quadrangle Inequality

\[ B_{i,j} = w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} \]

- Lemmas from [Yao (1980)]
  - (A) If \( w(i, j) \) satisfies QI (and some additional constraints),
    \[ \Rightarrow B_{i,j} \text{ satisfies QI.} \]
  - (B) If \( B_{i,j} \) satisfies QI,
    \[ \Rightarrow K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j) \]
Speedup using Quadrangle Inequality

\[ B_{i,j} = w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} \]

- Lemmas from [Yao (1980)]
  - (A) If \( w(i, j) \) satisfies QI (and some additional constraints),
    \[ \Rightarrow B_{i,j} \] satisfies QI.
  - (B) If \( B_{i,j} \) satisfies QI,
    \[ \Rightarrow K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j) \]

- In optimal BST problem,
  \[ B_{i,j} = w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} \]
Speedup using Quadrangle Inequality

\[ B_{i,j} = w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} \]

- Lemmas from [Yao (1980)]
  - (A) If \( w(i, j) \) satisfies QI (and some additional constraints),
    \[ \Rightarrow B_{i,j} \text{ satisfies QI.} \]
  - (B) If \( B_{i,j} \) satisfies QI,
    \[ \Rightarrow K_B(i, j - 1) \leq K_B(i, j) \leq K_B(i + 1, j) \]

- In optimal BST problem,
  \[ B_{i,j} = w(i, j) + \min_{i < t \leq j} \{ B_{i,t-1} + B_{t,j} \} \]
  - The specific \( w(i, j) \) satisfies QI (and the additional constraints).
Outline

- Background
  - Kunth-Yao (KY) Quadrangle Inequality (QI) Speedup
  - SMAWK Algorithm for finding Row Minima of Totally Monotone (TM) Matrices

- The $D^d$ Decomposition
  A transformation from QI to TM such that SMAWK solves KY problem as quickly as KY.

- The $L^m$ and $R^m$ Decompositions
  Another transformation from QI to TM that (1) implies KY speedup and (2) enables online solution.
Totally Monotone Matrices

Definition
Totally Monotone Matrices

Definition \( M \) is an \( m \times n \) matrix.
Definition

\( M \) is an \( m \times n \) matrix

\( \text{RM}_M(i) \) is index of minimum item of row \( i \) of \( M \).
Totally Monotone Matrices

Definition

$M$ is an $m \times n$ matrix

- $RM_M(i)$ is index of minimum item of row $i$ of $M$.
- $M$ is Monotone if $\forall i \leq i', \quad RM_M(i) \leq RM_M(i')$. 
Totally Monotone Matrices

**Definition**

$M$ is an $m \times n$ matrix

- $RM_M(i)$ is index of minimum item of row $i$ of $M$.
- $M$ is Monotone if $\forall i \leq i'$, $RM_M(i) \leq RM_M(i')$.

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>2</th>
<th>4</th>
<th>3</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

$RM_M(1) = 2$
$RM_M(2) = 4$
$RM_M(3) = 4$
$RM_M(4) = 4$
$RM_M(5) = 6$
$RM_M(6) = 6$
Totally Monotone Matrices

Definition $M$ is an $m \times n$ matrix

- $RM_M(i)$ is index of minimum item of row $i$ of $M$.
- $M$ is Monotone if $\forall i \leq i'$, $RM_M(i) \leq RM_M(i')$.

<table>
<thead>
<tr>
<th>7</th>
<th>2</th>
<th>4</th>
<th>3</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

- $RM_M(1) = 2$
- $RM_M(2) = 4$
- $RM_M(3) = 4$
- $RM_M(4) = 4$
- $RM_M(5) = 6$
- $RM_M(6) = 6$

An $m \times n$ matrix $M$ is Totally Monotone (TM) if every $2 \times 2$ submatrix is Monotone.
SMAWK Algorithm
SMAWK Algorithm

Motivation

Find all $m$ row minima of an implicitly given $m \times n$ matrix $M$
SMAWK Algorithm

Motivation
Find all $m$ row minima of an implicitly given $m \times n$ matrix $M$

Naive Algorithm: $O(mn)$
SMAWK Algorithm

- Motivation
  Find all $m$ row minima of an *implicitly* given $m \times n$ matrix $M$

- Naive Algorithm: $O(mn)$

- SMAWK Algorithm
  [Aggarwal, Klawe, Moran, Shor, Wilber (1986)]
SMAWK Algorithm

Motivation

Find all $m$ row minima of an implicitly given $m \times n$ matrix $M$

Naive Algorithm: $O(mn)$

SMAWK Algorithm

[Aggarwal, Klawe, Moran, Shor, Wilber (1986)]

If $M$ is Totally Monotone,

all $m$ row minima can be found in $O(m + n)$ time.
SMAWK Algorithm

Motivation

Find all \( m \) row minima of an implicitly given \( m \times n \) matrix \( M \)

Naive Algorithm: \( O(mn) \)

SMAWK Algorithm

[Aggarwal, Klawe, Moran, Shor, Wilber (1986)]

If \( M \) is Totally Monotone,

all \( m \) row minima can be found in \( O(m + n) \) time.

Usually \( \Theta(n) \) speedup: \( O(n^2) \) down to \( O(n) \).
The Monge Property
The Monge Property

Motivation

TM property is often established via Monge property.
The Monge Property

Motivation

TM property is often established via Monge property.

Definition

An \( m \times n \) matrix \( M \) is Monge if \( \forall i \leq i' \) and \( \forall j \leq j' \)

\[
M_{i,j} + M_{i',j'} \leq M_{i',j} + M_{i,j'}
\]
The Monge Property

- **Motivation**
  TM property is often established via Monge property.

- **Definition**
  An $m \times n$ matrix $M$ is Monge if $\forall i \leq i'$ and $\forall j \leq j'$
  \[
  M_{i,j} + M_{i',j'} \leq M_{i',j} + M_{i,j'}
  \]

- **Theorems**
  $M$ is Monge $\Rightarrow$ $M$ is Totally Monotone
  $M$ is Monge $\not\Leftrightarrow$ $M$ is Totally Monotone
The Monge Property

Quadrangle Inequality

Function \( f(i, j) \)

\[ f(i, j) + f(i', j') \leq f(i', j) + f(i, j') \]

Monge Matrix \( M \)

\[ M_{i,j} + M_{i',j'} \leq M_{i',j} + M_{i,j'} \]
The Monge Property

Quadrangle Inequality

Function $f(i, j)$

$\forall i \leq i' \leq j \leq j'$

$f(i, j) + f(i', j') \leq f(i', j) + f(i, j')$

QI vs. Monge

Monge

Matrix $M$

$\forall i \leq i'$ and $\forall j \leq j'$

$M_{i,j} + M_{i',j'} \leq M_{i',j} + M_{i,j'}$
The Monge Property

Quadrangle Inequality

Function $f(i, j)$

$\forall i \leq i' \leq j \leq j'$

$f(i, j) + f(i', j') \leq f(i', j) + f(i, j')$

Monge

Matrix $M$

$\forall i \leq i'$ and $\forall j \leq j'$

$M_{i,j} + M_{i',j'} \leq M_{i',j} + M_{i,j'}$

QI vs. Monge

Different names for same type of inequality.
The Monge Property

Quadrangle Inequality

Function \( f(i, j) \)
\[ \forall i \leq i' \leq j \leq j' \]
\[ f(i, j) + f(i', j') \leq f(i', j) + f(i, j') \]

Monge

Matrix \( M \)
\[ \forall i \leq i' \text{ and } \forall j \leq j' \]
\[ M_{i,j} + M_{i',j'} \leq M_{i',j} + M_{i,j'} \]

QI vs. Monge

- Different names for same type of inequality.
- Used differently in literature.
The Monge Property

Quadrangle Inequality

Function $f(i, j)$

$\forall i \leq i' \leq j \leq j'$

$f(i, j) + f(i', j') \leq f(i', j) + f(i, j')$

Monge

Matrix $M$

$\forall i \leq i'$ and $\forall j \leq j'$

$M_{i,j} + M_{i',j'} \leq M_{i',j} + M_{i,j'}$

QI vs. Monge

Different names for same type of inequality.

Used differently in literature.

QI: $f(i, j)$ is function to be calculated.

Monge: $M_{i,j}$ implicitly given.
The Monge Property

Quadrangle Inequality

Function $f(i, j)$

$\forall i \leq i' \leq j \leq j'$

$f(i, j) + f(i', j') \leq f(i', j) + f(i, j')$

Monge

Matrix $M$

$\forall i \leq i'$ and $\forall j \leq j'$

$M_{i,j} + M_{i',j'} \leq M_{i',j} + M_{i,j'}$

QI vs. Monge

- Different names for same type of inequality.
- Used differently in literature.
  - QI: $f(i, j)$ is function to be calculated.
    - Need all $f(i, j)$ entries.
  - Monge: $M_{i,j}$ implicitly given.
    - Only need the row minima, but not other entries.
Relationship?

Quadrangle Inequality  Totally Monotone (Monge)
Relationship?

Quadrangle Inequality  
A matrix to be calculated

Totally Monotone (Monge)  
A matrix given implicitly
Relationship?

**Quadrangle Inequality**
A matrix to be calculated
Need all $O(n^2)$ entries

**Totally Monotone (Monge)**
A matrix given implicitly
Need only $O(n)$ row minima
## Relationship?

<table>
<thead>
<tr>
<th>Quadrangle Inequality</th>
<th>Totally Monotone (Monge)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A matrix to be calculated</td>
<td>A matrix given implicitly</td>
</tr>
<tr>
<td>Need all $O(n^2)$ entries</td>
<td>Need only $O(n)$ row minima</td>
</tr>
<tr>
<td>$O(n^3)$ to $O(n^2)$ speedup</td>
<td>$O(n^2)$ to $O(n)$ speedup</td>
</tr>
</tbody>
</table>
**Quadrangle Inequality**
A matrix to be calculated
Need all $O(n^2)$ entries
$O(n^3)$ to $O(n^2)$ speedup

**Totally Monotone (Monge)**
A matrix given implicitly
Need only $O(n)$ row minima
$O(n^2)$ to $O(n)$ speedup

This talk
Relationship?

Quadrangle Inequality
A matrix to be calculated
Need all $O(n^2)$ entries
$O(n^3)$ to $O(n^2)$ speedup

Totally Monotone (Monge)
A matrix given implicitly
Need only $O(n)$ row minima
$O(n^2)$ to $O(n)$ speedup

This talk
- QI instance is decomposed into $\Theta(n)$ TM instances
## Relationship?

<table>
<thead>
<tr>
<th>Quadrangle Inequality</th>
<th>Totally Monotone (Monge)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A matrix to be calculated</td>
<td>A matrix given implicitly</td>
</tr>
<tr>
<td>Need all $O(n^2)$ entries</td>
<td>Need only $O(n)$ row minima</td>
</tr>
<tr>
<td>$O(n^3)$ to $O(n^2)$ speedup</td>
<td>$O(n^2)$ to $O(n)$ speedup</td>
</tr>
</tbody>
</table>

**This talk**

- QI instance is decomposed into $\Theta(n)$ TM instances
- Each TM instance requires $O(n)$ time
**Quadrangle Inequality**
A matrix to be calculated
- Need all $O(n^2)$ entries
- $O(n^3)$ to $O(n^2)$ speedup

**Totally Monotone (Monge)**
A matrix given implicitly
- Need only $O(n)$ row minima
- $O(n^2)$ to $O(n)$ speedup

**This talk**
- QI instance is decomposed into $\Theta(n)$ TM instances
- Each TM instance requires $O(n)$ time
- $\Rightarrow$ QI instance requires $O(n^2)$ time in total
Outline

Background
- Kunth-Yao (KY) Quadrangle Inequality (QI) Speedup
- SMAWK Algorithm for finding Row Minima of Totally Monotone (TM) Matrices

The $D^d$ Decomposition
A transformation from QI to TM such that SMAWK solves KY problem as quickly as KY.

The $L^m$ and $R^m$ Decompositions
Another transformation from QI to TM that
- (1) implies KY speedup and (2) enables online solution.
Decompositions

QI instance $\rightarrow \Theta(n)$ TM instances
Decompositions

QI instance $\rightarrow \Theta(n)$ TM instances

- $D^d$ decomposition

- $L^m$ and $R^m$ decompositions
Decompositions

QI instance $\rightarrow \Theta(n)$ TM instances

- $D^d$ decomposition
  - Each diagonal $\rightarrow$ TM instance

- $L^m$ and $R^m$ decompositions
Decompositions

QI instance $\rightarrow \Theta(n)$ TM instances

- $D^d$ decomposition
  - Each diagonal $\rightarrow$ TM instance

- $L^m$ and $R^m$ decompositions
  - $L^m$: Each row $\rightarrow$ TM instance
  - $R^m$: Each column $\rightarrow$ TM instance
$D^d$ Decomposition

$D^d$ decomposition

• Each diagonal $\rightarrow$ TM instance
$D^d$ Decomposition

$D^d$ decomposition

Each diagonal $\rightarrow$ TM instance
$D^d$ Decomposition

- $D^d$ decomposition
- Each diagonal $\rightarrow$ TM instance
$D^d$ Decomposition

$D^d$ decomposition

Each diagonal $\rightarrow$ TM instance
$D^d$ Decomposition

- $D^d$ decomposition
- Each diagonal $\rightarrow$ TM instance
$D^d$ Decomposition

$D^d$ decomposition

Each diagonal $\rightarrow$ TM instance
$L^m$ and $R^m$ Decompositions ($R^m$ shown)

- $L^m$ and $R^m$ decompositions
  - $L^m$: Each row $\rightarrow$ TM instance
  - $R^m$: Each column $\rightarrow$ TM instance
$L^m$ and $R^m$ Decompositions ($R^m$ shown)

- $L^m$ and $R^m$ decompositions
  - $L^m$: Each row $\rightarrow$ TM instance
  - $R^m$: Each column $\rightarrow$ TM instance
$L^m$ and $R^m$ Decompositions ($R^m$ shown)

- $L^m$ and $R^m$ decompositions
  - $L^m$: Each row $\rightarrow$ TM instance
  - $R^m$: Each column $\rightarrow$ TM instance
$L^m$ and $R^m$ Decompositions ($R^m$ shown)

- $L^m$ and $R^m$ decompositions
- $L^m$: Each row $\rightarrow$ TM instance
- $R^m$: Each column $\rightarrow$ TM instance
\( L^m \) and \( R^m \) Decompositions (\( R^m \) shown)

- \( L^m \) and \( R^m \) decompositions
  - \( L^m \): Each row \( \rightarrow \) TM instance
  - \( R^m \): Each column \( \rightarrow \) TM instance
\[ L^m \text{ and } R^m \text{ Decompositions} \quad (R^m \text{ shown}) \]

- \( L^m \) and \( R^m \) decompositions
  - \( L^m \): Each row \( \rightarrow \) TM instance
  - \( R^m \): Each column \( \rightarrow \) TM instance
Definition
D^d Decomposition

**Definition**

For diagonal $d$, $(1 \leq d < n)$

$$B_{i,i+d} = w(i, i + d) + \min_{i < j \leq i + d} \{B_{i,j-1} + B_{j,i+d}\}$$
$D^d$ Decomposition

**Definition**

- For diagonal $d$, $(1 \leq d < n)$
  \[
  B_{i,i+d} = w(i, i + d) + \min_{i < j \leq i + d} \{ B_{i,j-1} + B_{j,i+d} \}
  \]

- Define $(n - d + 1) \times (n + 1)$ matrix $D^d$

\[
D^{d}_{i,j} = \begin{cases} 
  w(i, i + d) + \{ B_{i,j-1} + B_{j,i+d} \} & \text{if } 0 \leq i < j \leq i + d \leq n \\
  \infty & \text{otherwise}
\end{cases}
\]
$D^d$ Decomposition

**Definition**

- For diagonal $d$, $(1 \leq d < n)$
  
  \[ B_{i,i+d} = w(i, i + d) + \min_{i<j\leq i+d}\{B_{i,j-1} + B_{j,i+d}\} \]

- Define \((n - d + 1) \times (n + 1)\) matrix $D^d$
  
  \[
  D^d_{i,j} = \begin{cases} 
  w(i, i + d) + \{B_{i,j-1} + B_{j,i+d}\} & \text{if } 0 \leq i < j \leq i + d \leq n \\
  \infty & \text{otherwise}
  \end{cases}
  \]

- Then, $B_{i,i+d} = \min_{0 \leq j \leq n} D^d_{i,j} = \text{minimum of row } i \text{ of } D^d$
$D^d$ Decomposition

**Definition**

- For diagonal $d$, $(1 \leq d < n)$
  \[ B_{i,i+d} = w(i, i + d) + \min_{i<j \leq i+d} \{ B_{i,j-1} + B_{j,i+d} \} \]

- Define $(n - d + 1) \times (n + 1)$ matrix $D^d$

  \[
  D^d_{i,j} = \begin{cases} 
  w(i, i + d) + \{ B_{i,j-1} + B_{j,i+d} \} & \text{if } 0 \leq i < j \leq i + d \leq n \\
  \infty & \text{otherwise}
  \end{cases}
  \]

- Then, $B_{i,i+d} = \min_{0 \leq j \leq n} D^d_{i,j} = \text{minimum of row } i \text{ of } D^d$

**Lemma**

$D^d$ is Monge, for each $1 \leq d < n$. 

Quadrangle-Inequality and Total-Monotonicity – p.22/30
**$D^d$ Decomposition**

**Definition**
- For diagonal $d$, $(1 \leq d < n)$
  \[ B_{i,i+d} = w(i, i + d) + \min_{i<j\leq i+d} \{ B_{i,j-1} + B_{j,i+d} \} \]
- Define $(n - d + 1) \times (n + 1)$ matrix $D^d$
  
  \[
  D^d_{i,j} = \begin{cases} 
  w(i, i + d) + \{ B_{i,j-1} + B_{j,i+d} \} & \text{if } 0 \leq i < j \leq i + d \leq n \\
  \infty & \text{otherwise}
  \end{cases}
  \]
- Then, $B_{i,i+d} = \min_{0 \leq j \leq n} D^d_{i,j}$ = minimum of row $i$ of $D^d$

**Lemma**
- $D^d$ is Monge, for each $1 \leq d < n$.
- For fixed $d$, SMAWK can be used to find all the $B_{i,i+d}$ in $O(n)$ time.
  \[ \Rightarrow O(n^2) \text{ time for all } D^d. \]
$R^m$ Decomposition

• Definition
Definition

For column $m$, $(1 \leq m \leq n)$

$$B_{i,m} = w(i, m) + \min_{i<j \leq m}\{B_{i,j-1} + B_{j,m}\}$$
Definition

For column $m$, $(1 \leq m \leq n)$

$$B_{i,m} = w(i, m) + \min_{i<j\leq m}\{B_{i,j-1} + B_{j,m}\}$$

Define $(m + 1) \times (m + 1)$ matrix $R_m$

$$R_{i,j}^m = \begin{cases} 
  w(i, m) + \{B_{i,j-1} + B_{j,m}\} & \text{if } 0 \leq i < j \leq m \\
  \infty & \text{otherwise}
\end{cases}$$
$R^m$ Decomposition

**Definition**

- For column $m$, $(1 \leq m \leq n)$
  \[
  B_{i,m} = w(i,m) + \min_{i<j \leq m}\{B_{i,j-1} + B_{j,m}\}
  \]

- Define $(m + 1) \times (m + 1)$ matrix $R^m$
  \[
  R^m_{i,j} = \begin{cases} 
  w(i,m) + \{B_{i,j-1} + B_{j,m}\} & \text{if } 0 \leq i < j \leq m \\
  \infty & \text{otherwise}
  \end{cases}
  \]

- Then, $B_{i,m} = \min_{0<j \leq m} R^m_{i,j}$
$R^m$ Decomposition

**Definition**

For column $m$, $(1 \leq m \leq n)$

$$B_{i,m} = w(i, m) + \min_{i < j \leq m} \{B_{i,j-1} + B_{j,m}\}$$

Define $(m + 1) \times (m + 1)$ matrix $R^m$

$$R^m_{i,j} = \begin{cases} 
    w(i, m) + \{B_{i,j-1} + B_{j,m}\} & \text{if } 0 \leq i < j \leq m \\
    \infty & \text{otherwise}
\end{cases}$$

Then, $B_{i,m} = \min_{0 < j \leq m} R^m_{i,j}$

**Lemma**

$R^m$ is Monge, for each $1 \leq m \leq n$. 

Quadrangle-Inequality and Total-Monotonicity – p.23/30
LARSCH Algorithm
LARSCH Algorithm

$D^d$ decomposition
LARSCH Algorithm

- $D^d$ decomposition
  - $D^d_{i,j} = w(i, i + d) + \{B_{i,j-1} + B_{j,i+d}\} \quad (0 \leq i < j \leq i + d \leq n)$
- SMAWK algorithm
LARSCH Algorithm

- $D^d$ decomposition

  \[ D^d_{i,j} = w(i, i + d) + \{B_{i,j-1} + B_{j,i+d}\} \quad (0 \leq i < j \leq i + d \leq n) \]

- SMAWK algorithm

- $L^m$ and $R^m$ decomposition

  \[ R^m_{i,j} = w(i, m) + \{B_{i,j-1} + B_{j,m}\} \quad (0 \leq i < j \leq m) \]
LARSCH Algorithm

- **$D^d$ decomposition**
  \[
  D^d_{i,j} = w(i, i + d) + \{B_{i,j-1} + B_{j,i+d}\} \quad (0 \leq i < j \leq i + d \leq n)
  \]

- **SMAWK algorithm**

- **$L^m$ and $R^m$ decomposition**
  \[
  R^m_{i,j} = w(i, m) + \{B_{i,j-1} + B_{j,m}\} \quad (0 \leq i < j \leq m)
  \]

- Can not use SMAWK algorithm:
  $B_{j,m}$ is row minimum of row $j$ of $R^m$ and is therefore not known.
LARSCH Algorithm

- $D^d$ decomposition
  
  $D^d_{i,j} = w(i, i + d) + \{B_{i,j-1} + B_{j,i+d}\} \quad (0 \leq i < j \leq i + d \leq n)$

- SMAWK algorithm

- $L^m$ and $R^m$ decomposition
  
  $R^m_{i,j} = w(i, m) + \{B_{i,j-1} + B_{j,m}\} \quad (0 \leq i < j \leq m)$

  Can not use SMAWK algorithm:

  $B_{j,m}$ is row minimum of row $j$ of $R^m$ and is therefore not known.

- LARSCH algorithm [Larmore, Schieber (1990)]

  permits calculating row minima of TM matrices in $O(n)$ time, even with this dependency.
LARSCH Algorithm

- $D^d$ decomposition
  \[ D^d_{i,j} = w(i, i + d) + \{B_{i,j-1} + B_{j,i+d}\} \quad (0 \leq i < j \leq i + d \leq n) \]

- SMAWK algorithm

- $L^m$ and $R^m$ decomposition
  \[ R^m_{i,j} = w(i, m) + \{B_{i,j-1} + B_{j,m}\} \quad (0 \leq i < j \leq m) \]

- Can not use SMAWK algorithm:
  $B_{j,m}$ is row minimum of row $j$ of $R^m$ and is therefore not known.

- LARSCH algorithm [Larmore, Schieber (1990)]
  permits calculating row minima of TM matrices in $O(n)$ time, even with this dependency.

- $O(n)$ time for each column $\Rightarrow O(n^2)$ in total.
LARSCH Algorithm

Finding row minima in totally monotone matrices with limited dependency.
LARSCH Algorithm

Finding row minima in totally monotone matrices with limited dependency.

Entries of column \( j \) can depend on the row minima of rows \( i \) where \( M_{i,j} = \infty \).

Green: the column \( j \).
Red: rows that column \( j \) can depend on.
LARSCH Algorithm

Finding row minima in totally monotone matrices with limited dependency.

Entries of column $j$ can depend on the row minima of rows $i$ where $M_{i,j} = \infty$.

Green: the column $j$.
Red: rows that column $j$ can depend on.
LARSCH Algorithm

Finding row minima in totally monotone matrices with limited dependency.

Entries of column $j$ can depend on the row minima of rows $i$ where $M_{i,j} = \infty$.

Green: the column $j$.
Red: rows that column $j$ can depend on.
LARSCH Algorithm

Finding row minima in totally monotone matrices with limited dependency.

Entries of column $j$ can depend on the row minima of rows $i$ where $M_{i,j} = \infty$.

Green: the column $j$.
Red: rows that column $j$ can depend on.
LARSCH Algorithm

Finding row minima in totally monotone matrices with limited dependency.

Entries of column $j$ can depend on the row minima of rows $i$ where $M_{i,j} = \infty$.

Green: the column $j$. Red: rows that column $j$ can depend on.

$R^m$ satisfies the condition of LARSCH.
Outline

- Background
  - Kunth-Yao (KY) Quadrangle Inequality (QI) Speedup
  - SMAWK Algorithm for finding Row Minima of Totally Monotone (TM) Matrices

- The $D^d$ Decomposition
  A transformation from QI to TM such that SMAWK solves KY problem as quickly as KY.

- The $L^m$ and $R^m$ Decompositions
  Another transformation from QI to TM that (1) implies KY speedup and (2) enables online solution.
Online Problem
Online Problem

- Definition: Two-sided online problem
Online Problem

Definition: Two-sided online problem

Current step: Optimal BST for $\text{Key}_l, \ldots, \text{Key}_r$
Online Problem

Definition: Two-sided online problem

- Current step: Optimal BST for Key$_l$, . . . , Key$_r$
- Next step: Add either Key$_{l-1}$ or Key$_{r+1}$. 

Quadrangle-Inequality and Total-Monotonicity – p.27/30
Online Problem

Definition: Two-sided online problem
- Current step: Optimal BST for $\text{Key}_l, \ldots, \text{Key}_r$
- Next step: Add either $\text{Key}_{l-1}$ or $\text{Key}_{r+1}$.

An example
Online Problem

Definition: Two-sided online problem
- Current step: Optimal BST for Key\(_l\), \ldots, Key\(_r\)
- Next step: Add either Key\(_{l-1}\) or Key\(_{r+1}\).

An example
Input = (Key\(_l\), \ldots, Key\(_r\))

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>75</td>
<td>141</td>
<td>250</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>43</td>
<td>119</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Online Problem

Definition: Two-sided online problem
- Current step: Optimal BST for \(\text{Key}_l, \ldots, \text{Key}_r\)
- Next step: Add either \(\text{Key}_{l-1}\) or \(\text{Key}_{r+1}\).

An example
Input = (\(\text{Key}_l, \ldots, \text{Key}_r, \text{Key}_{r+1}\))

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td>75</td>
<td>141</td>
<td>250</td>
<td>357</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>0</td>
<td>43</td>
<td>119</td>
<td>204</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>44</td>
<td>121</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Online Problem

Definition: Two-sided online problem
- Current step: Optimal BST for $Key_l, \ldots, Key_r$
- Next step: Add either $Key_{l-1}$ or $Key_{r+1}$.

An example
Input = $(Key_{l-1}, Key_l, \ldots, Key_r, Key_{r+1})$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>146</td>
<td>260</td>
<td>349</td>
<td>491</td>
<td>624</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>75</td>
<td>141</td>
<td>250</td>
<td>357</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>43</td>
<td>119</td>
<td>204</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>44</td>
<td>121</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Using $L^m$ and $R^m$ decomposition
Using $L^m$ and $R^m$ decomposition

$O(n)$ time worst case per step.
Outline

- **Background**
  - Kunth-Yao (KY) Quadrangle Inequality (QI) Speedup
  - SMAWK Algorithm for finding Row Minima of Totally Monotone (TM) Matrices

- **The $D^d$ Decomposition**
  A transformation from QI to TM such that SMAWK solves KY problem as quickly as KY.

- **The $L^m$ and $R^m$ Decompositions**
  Another transformation from QI to TM that (1) implies KY speedup and (2) enables online solution.
Questions?