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

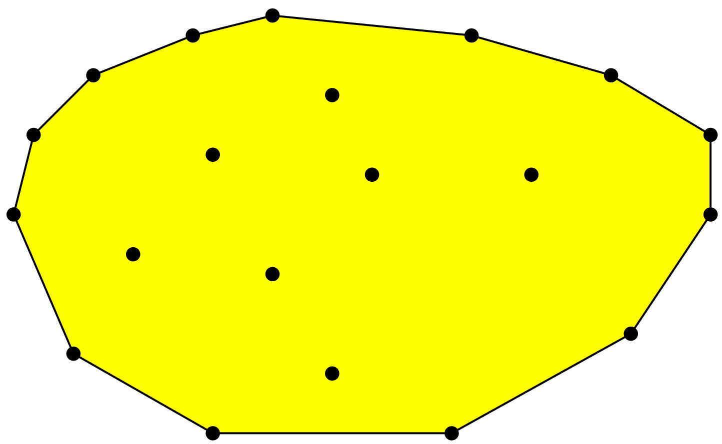
Outline of Talk

- **Convex hull definitions and results**
- **Observations on static convex hull**
- **Deletions-only data structure**
- **Fully dynamic data structure**
  - **General dynamization technique**
  - **Duality: convex hulls and envelopes of lines**
  - **Dual queries**
  - **Data structure**
- **Application**
- **Conclusion**

Planar Convex Hull

Input A set of points $P \subseteq \mathbb{R}^2$

Output The points on the convex hull $\mathbf{CH}(P)$ in clockwise order



$n = |P|$ $h = |\mathbf{CH}(P)|$

Known results

Optimal $O(n \log n)$ Graham 1972; ...

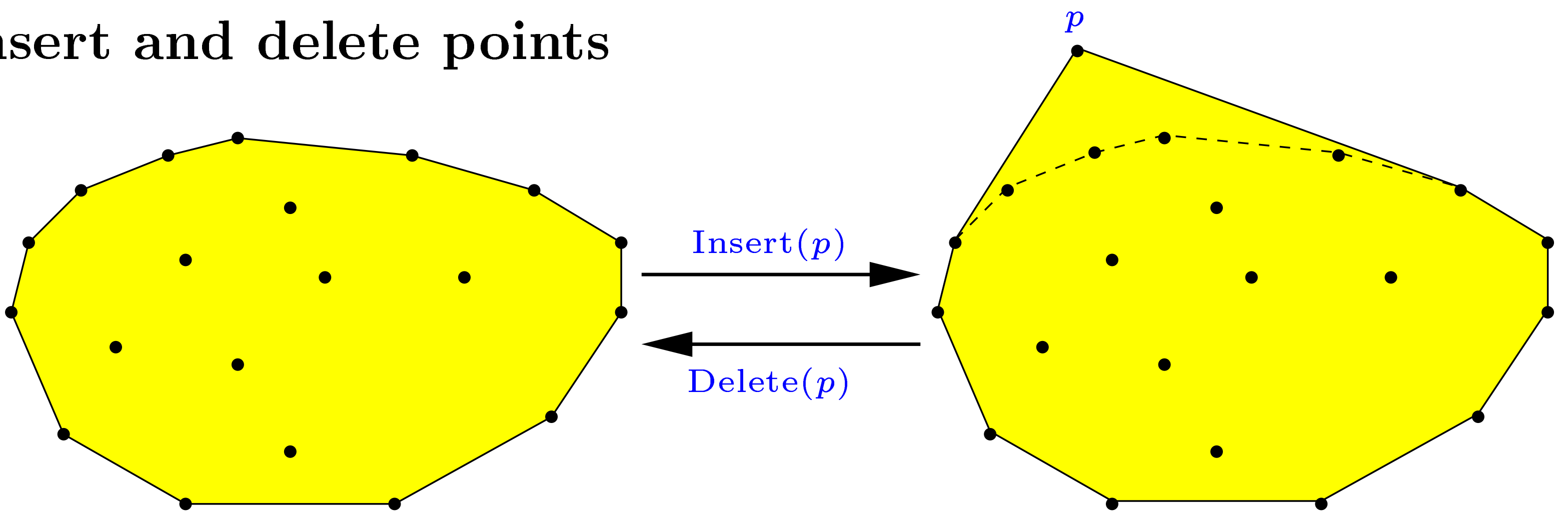
Output-sensitive $O(n \log h)$ Kirkpatrick, Seidel 1986; Chan 1996

BRICS

Dynamic Planar Convex Hull

Updates

Insert and delete points



Queries

(a) The extreme point in a direction

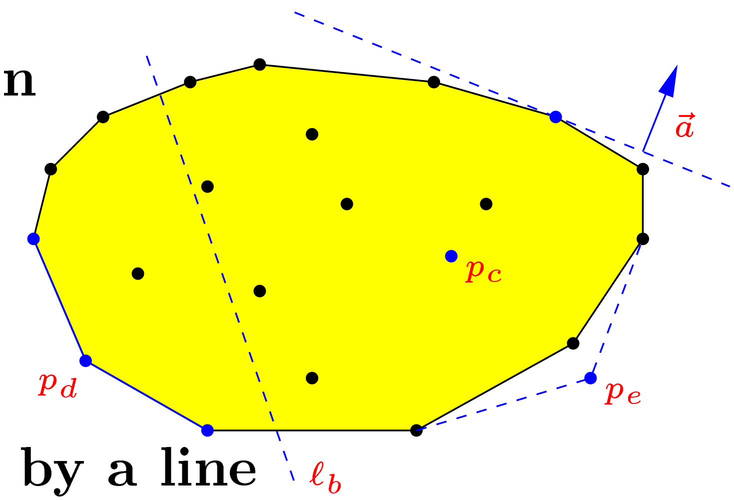
(b) Does a line intersect CH(P)?

(c) Is a point inside CH(P)?

(d) Neighbor points on CH(P)

(e) Tangent points on CH(P)

(f) The edges of CH(P) intersected by a line



Dynamic Planar Convex Hull Results

Fully dynamic	Update	Query
Overmars, van Leeuwen 1981	$O(\log^2 n)$	$O(\log n)$
Chan 1999	$O_{\mathrm{A}}(\log^{1+\epsilon} n)$	$O(\log n)$
Brodal, Jacob 2000	$O_{\mathrm{A}}(\log n \cdot \log\log n)$	$O(\log n)$

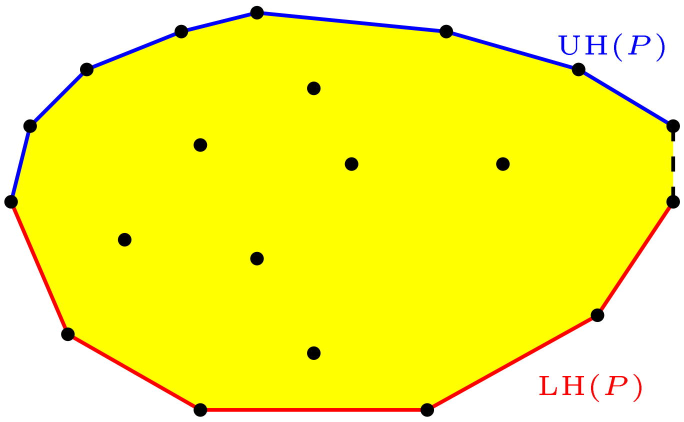
Insertions only	Insert	Query
Preparata, Shamos 1985	$O(\log n)$	$O(\log n)$

Deletions only	Preprocessing	Delete	Query
Hershberger, Suri 1992	$O_{\mathrm{A}}(n \log n)$	$O_{\mathrm{A}}(\log n)$	$O(\log n)$
Brodal, Jacob 2000	$O_{\mathrm{A}}(n)^*$	$O_{\mathrm{A}}(\log n \cdot \log\log n)$	—

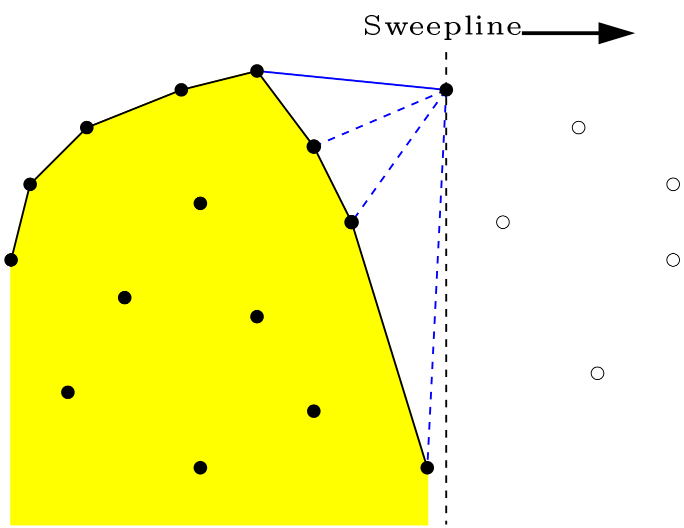
O_{A}=Amortized time Query=Queries (a)–(e) $*$=Points presorted

BRICS

Sweepline Algorithm for Convex Hull



Sufficient to consider the
upper hull UH(P)
The lower hull LH(P) is symmetric



If P lexicographically sorted

$\Downarrow$

Computing UH(P) takes $O(n)$ time

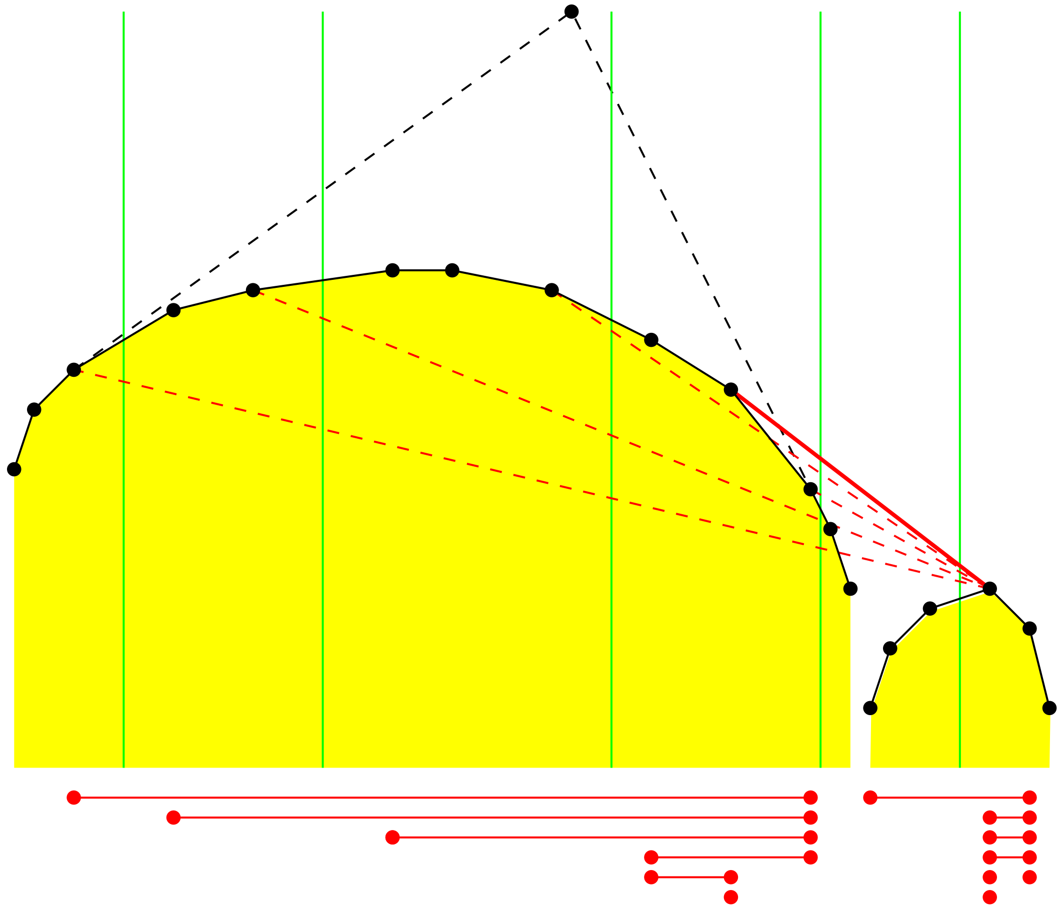
BRICS

Deletions-Only Data Structure

Goal A deletions-only data structure for storing n points

$O_{\mathbf{A}}(n)$ **preprocessing** and $O_{\mathbf{A}}(\log n \cdot \log\log n)$ **deletion** time

Deletions return the **changes** in the convex hull



Deletions-Only Data Structure

Goal **A deletions-only data structure for storing n points**

$O_{\mathbf{A}}(n)$ preprocessing and $O_{\mathbf{A}}(\log n \cdot \log\log n)$ deletion time

Deletions return the changes in the convex hull

Data structure

– Partition points in $\frac{n}{\log n}$ blocks

– For each block P_i construct a binary tree storing $\mathrm{UH}(P_i)$ in $O(\log n)$ time

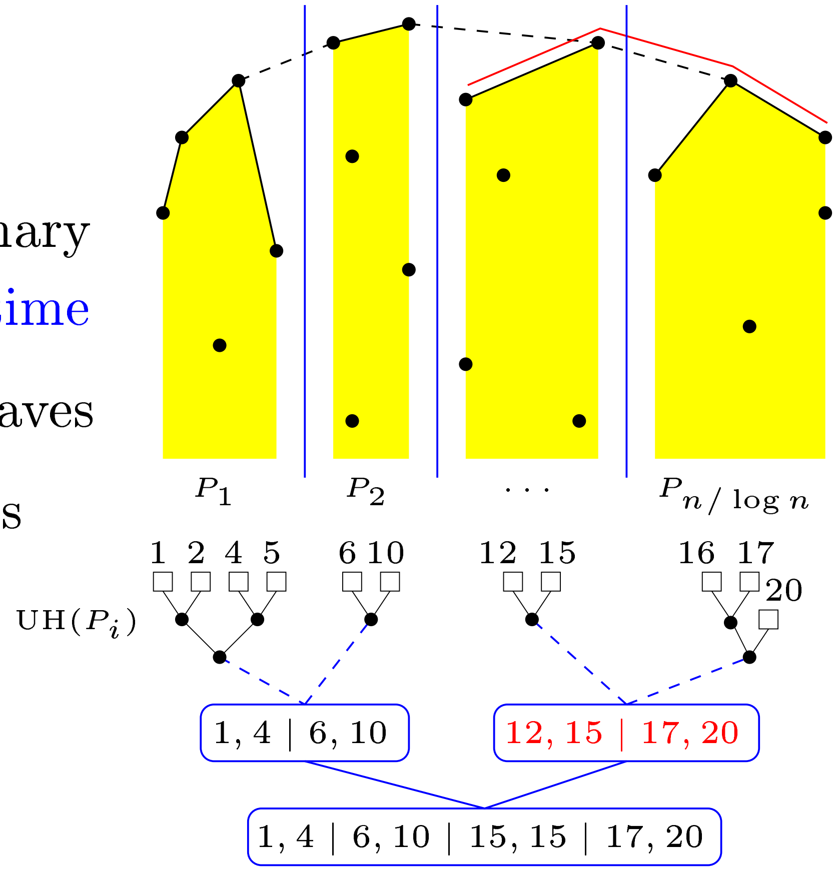
– Binary tree with blocks at the leaves

– $\mathrm{UH}(P_v)$ as pointer-pairs to blocks

– $\frac{n}{\log n}$ pointer-pairs per level

– Space $O(n)$

– Preprocessing $O(n)$



BRICS

Deletions

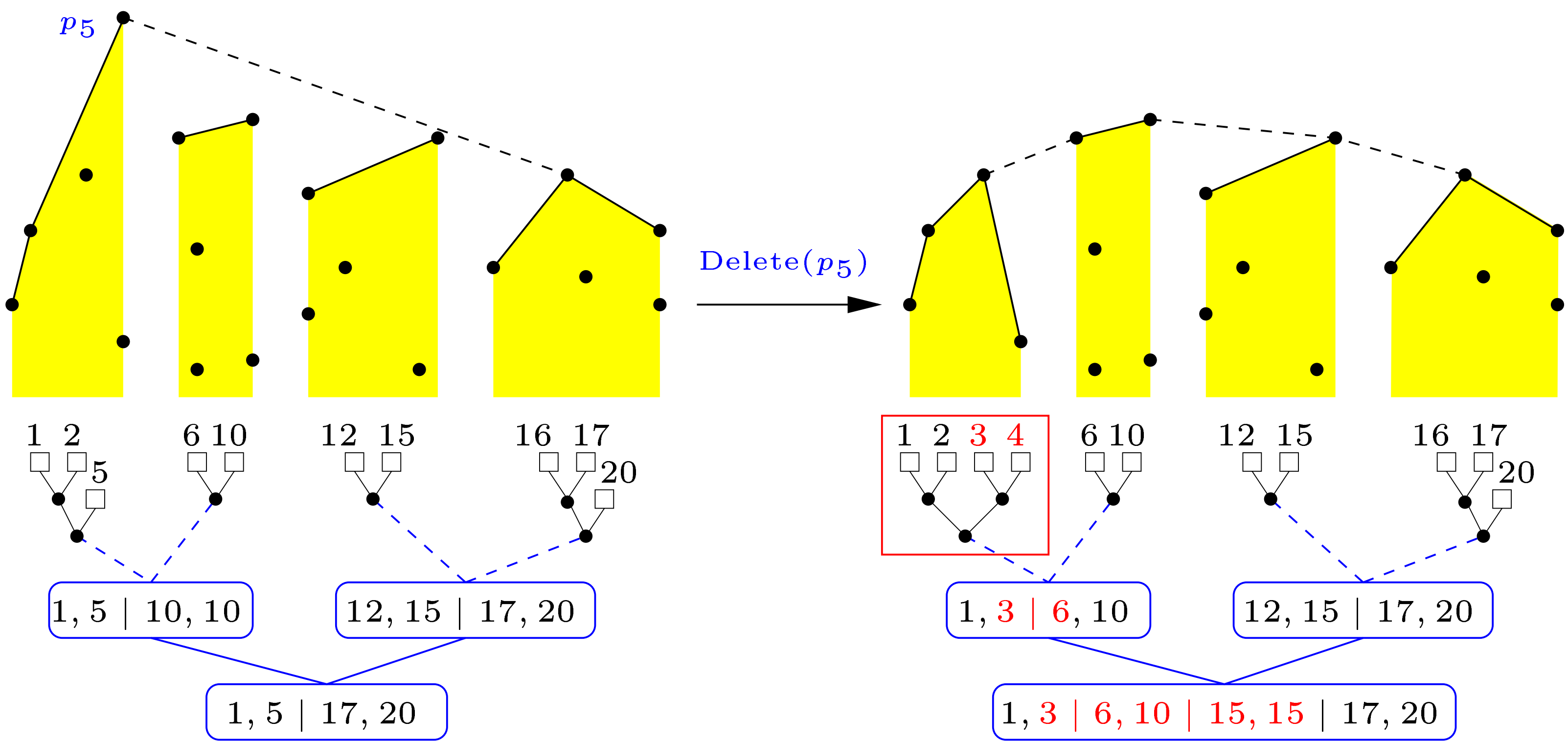
Delete point and rebuild UH(P_i) $O(\log n)$

Update UH(P_v) for each v on the path to the root $O(\log n) \times$

– **Delete p — possibly deleting a pointer-pair** $O(1)$

– **Insert/copy x pointer-pairs from the children** $O(x)$

– **Find a new bridge between two blocks** $O(\log \log n)$



Analysis - Deletions-Only

D deletions

$$O(X + D \cdot \log n \cdot \log\log n)$$

where

$$\begin{aligned} X &= \#\ \textbf{pointer-pairs inserted} \\ &\leq \frac{n}{\log n}\log n + D \cdot \log n \\ &= n + D \cdot \log n \end{aligned}$$

Amortized bounds

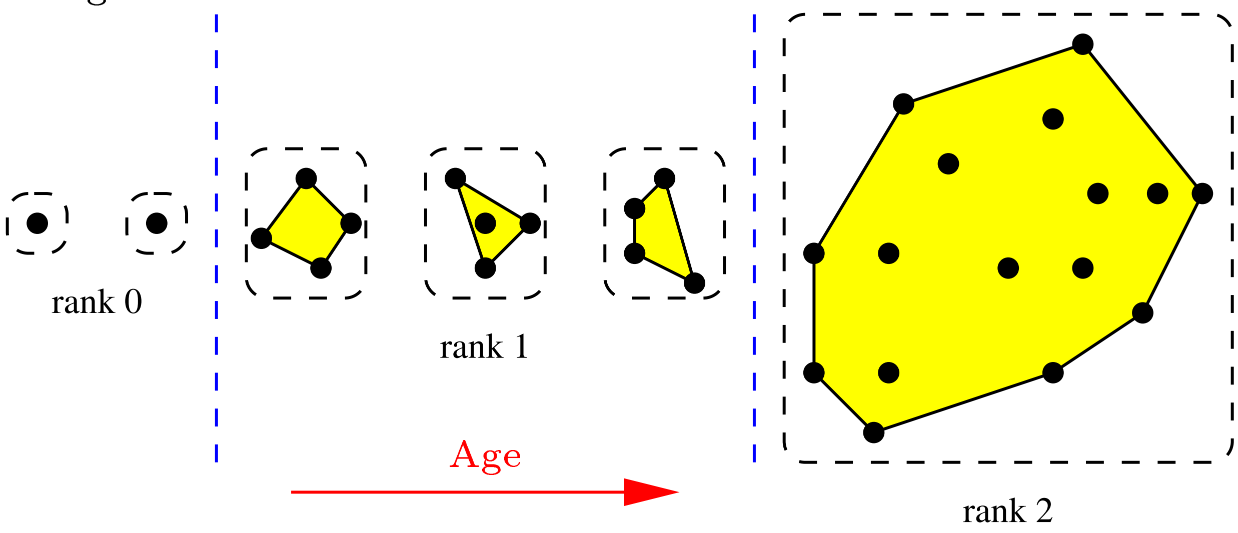
Preprocessing	$O_{\mathbf{A}}(n)$
Deletions	$O_{\mathbf{A}}(\log n \cdot \log\log n)$

BRICS

Fully Dynamic Data Structure

General dynamization technique

- **Collection $\mathcal{C}$ of sets of points**
- **Each set has a rank**
- **Store each set as a deletions-only data structure**
- **Insertions create new rank 0 sets**
- **If $\log n$ sets have rank r, merge them to a rank $r+1$ set**
- **Max rank** $= \log_{\log n} n$
- $|\mathcal{C}| \leq \log_{\log n} n \cdot \log n$



Outline of Operations

Insertions

Create new rank 0 sets and merge sets whenever overflowing

Merging $\equiv$ MergeSort using $O_{\mathbf{A}}(n)$ preprocessing $O_{\mathbf{A}}(\log n)$

Deletions

Delete point from deletions-only data structure

$O_{\mathbf{A}}(\log n \cdot \log\log n)$

Queries

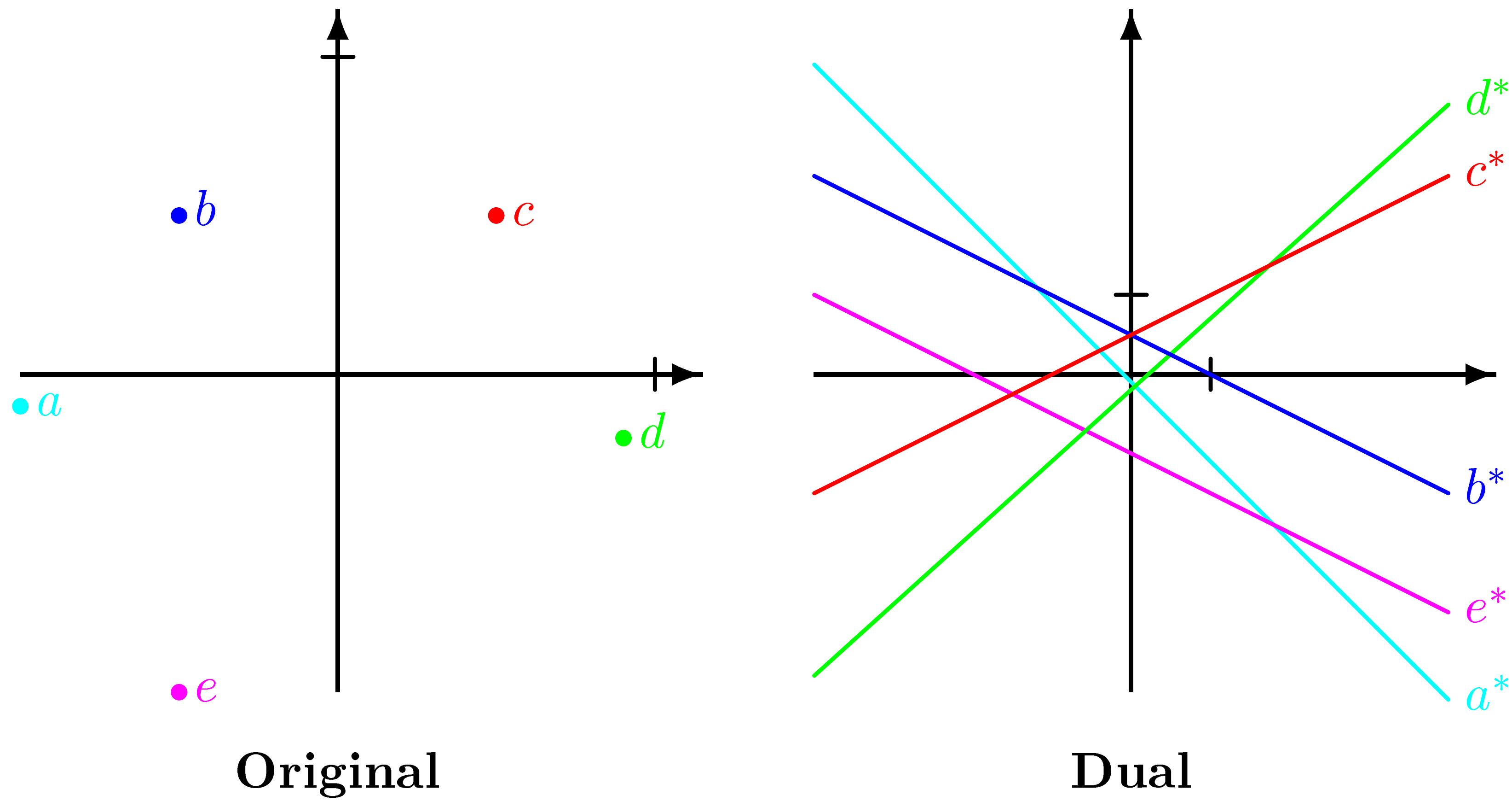
Query the convex hulls of the $O(\log^2 n)$ sets "simultaneously"

$\Rightarrow$ additional data structure — for the dual problem

BRICS

Duality

$$(a , b) \mapsto y = a \cdot x + b$$

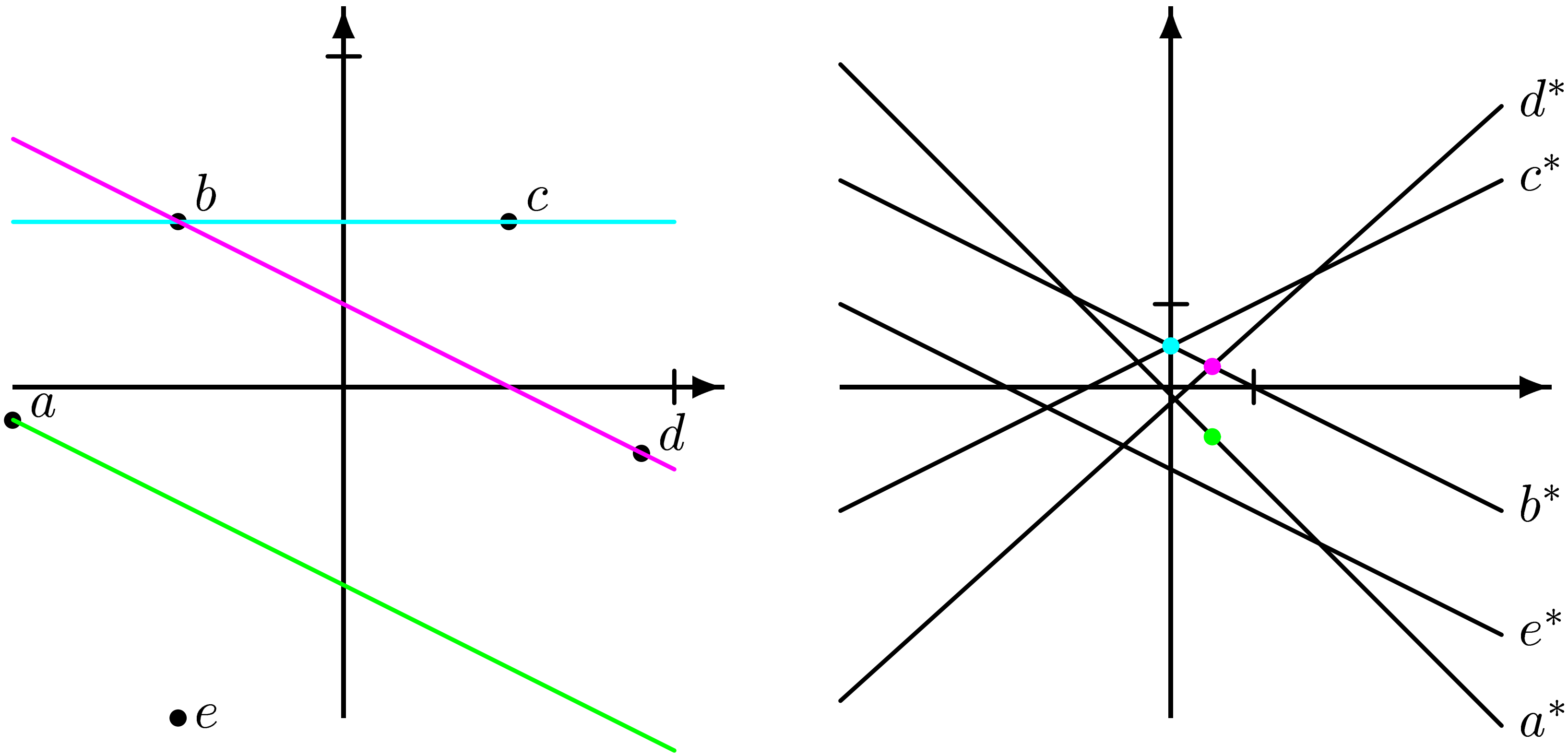


BRICS

Duality

$$(a , b) \mapsto y = a \cdot x + b$$

$$y = a \cdot x + b \mapsto (-a , b)$$

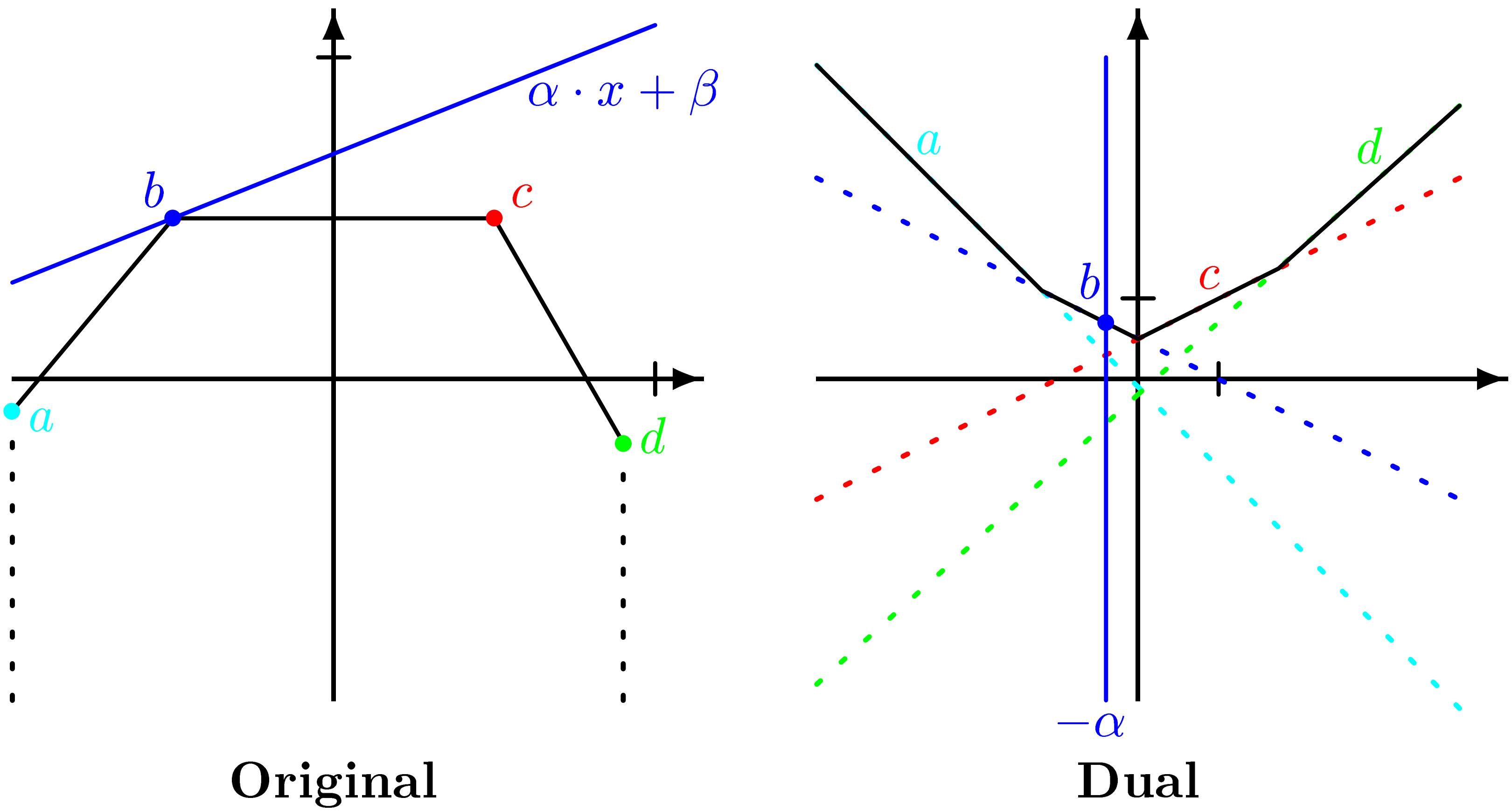


Original **Dual**

BRICS

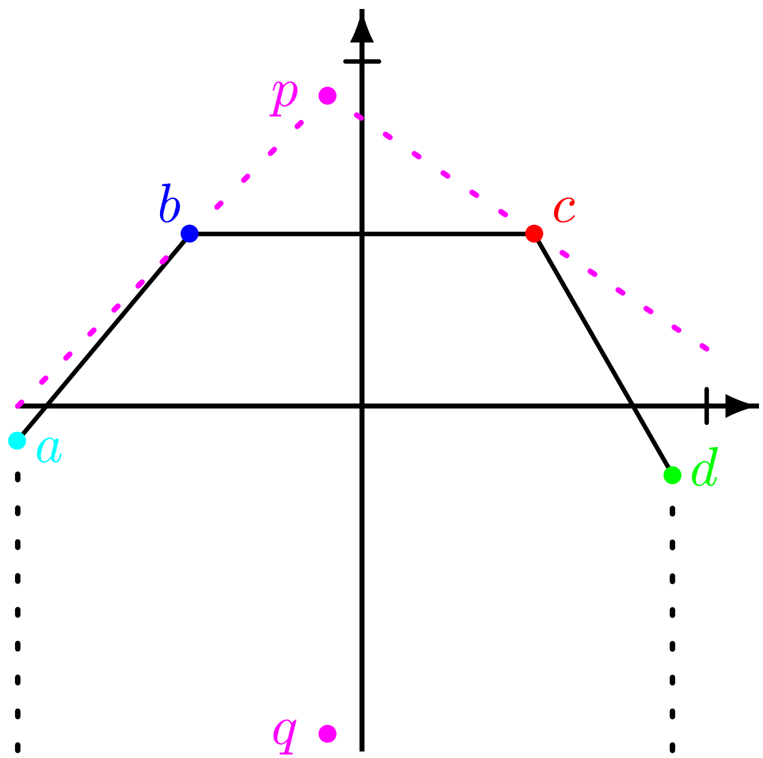
Upper Hulls vs. Upper Envelopes

Tangent with slope α $\mapsto$ **Intersection with** $x = -\alpha$

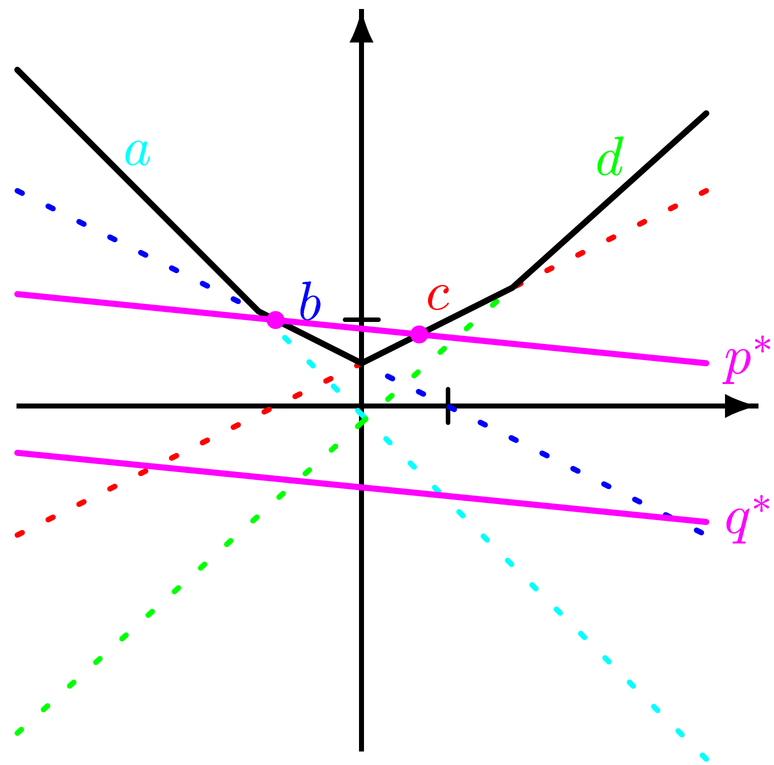


Upper Hulls vs. Upper Envelopes

Tangents through a point p $\mapsto$ **Intersections with** p^*



Original



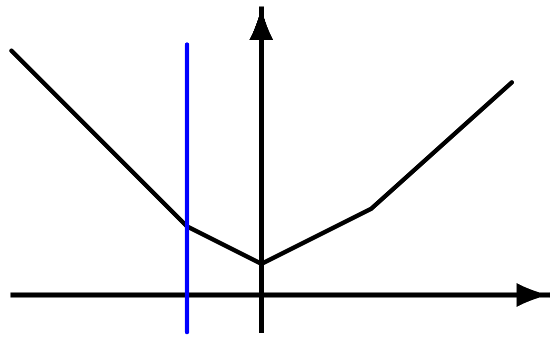
Dual

BRICS

Dual Queries

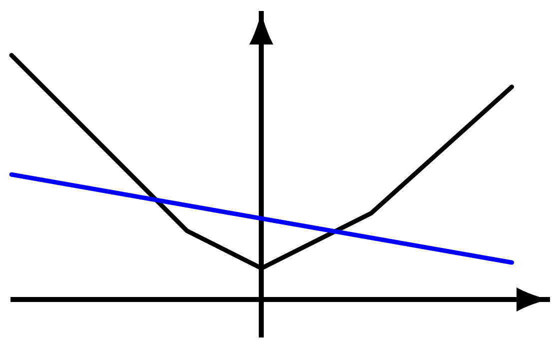
(a) The extreme point in a direction
(b) Does a line intersect CH(P)?

} **Vertical line intersection queries**



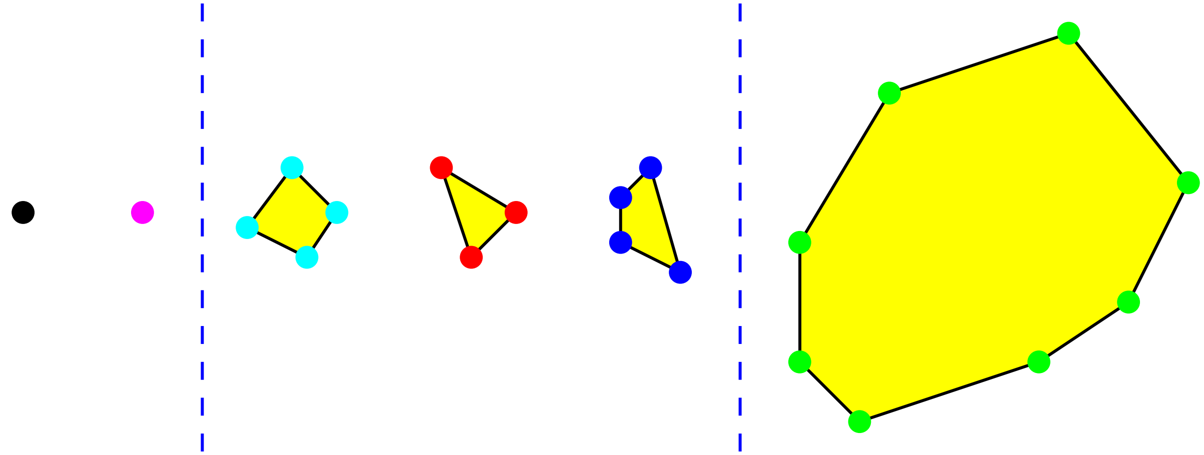
(c) Is a point inside CH(P)?
(d) Neighbor points on CH(P)
(e) Tangent points on CH(P)

} **Arbitrary line intersection queries**



BRICS

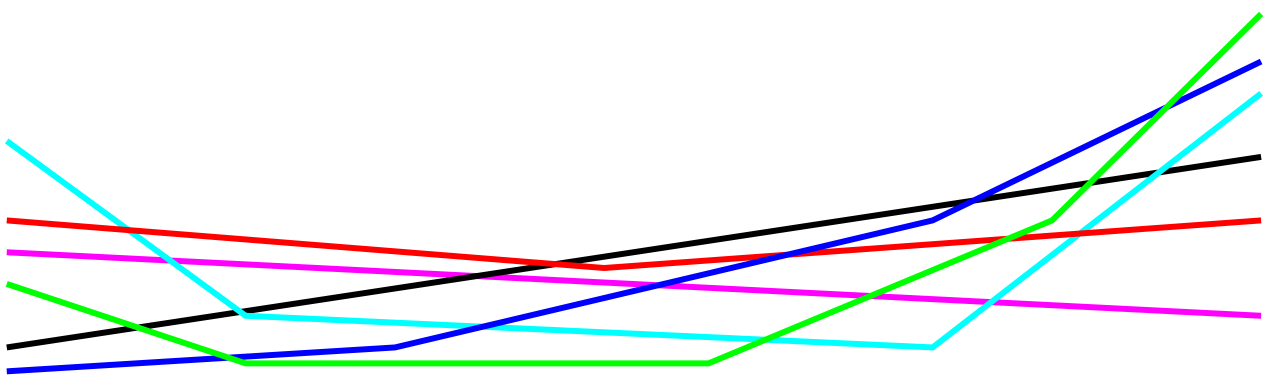
Unions of Envelopes



UH $\equiv$ **UH of points on the $O(\log^2 n)$ convex hulls**

$\equiv$ **the upper envelope in the dual of the points**

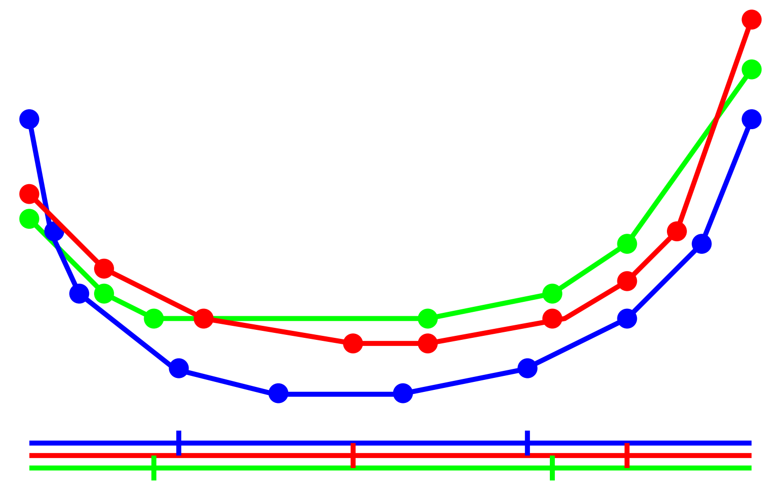
$\equiv$ **the upper envelope of $O(\log^2 n)$ upper envelopes**



BRICS

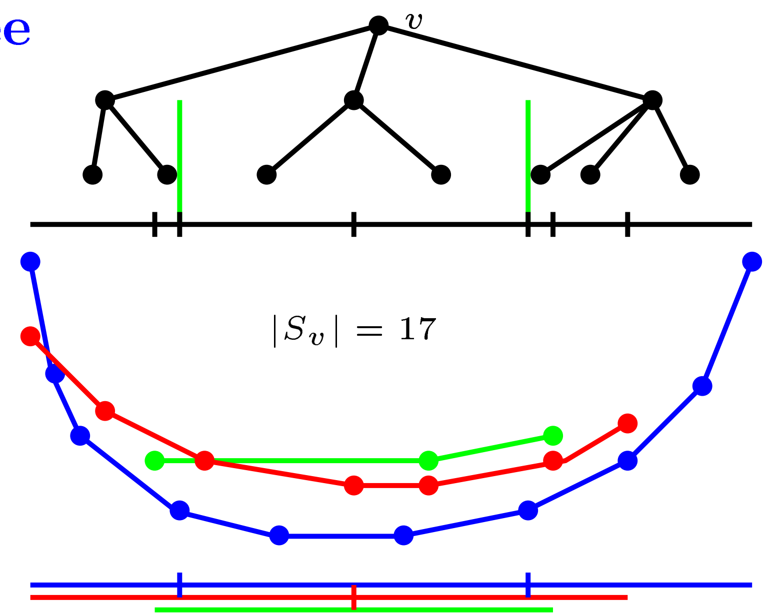
Block Decomposition of Envelopes

Partition upper envelopes in blocks with $O(\frac{\log n}{\log \log n})$ line segments



Each block covers an x-interval

Store the blocks in an interval tree with degree $O(\log n)$ and height $O(\frac{\log n}{\log \log n})$



S_v = line segments in blocks where the block interval contains at least one search-key of v (and not for any ancestor of v)

BRICS

Fully Dynamic Case : Queries

$$|S_v| = O(\log^4 n)$$

Store S_v as secondary data structures using fully dynamic data structures with $O_{\mathbf{A}}(U(n))$ update and $O(\log n)$ query time e.g. $U(n) = \log^2 n$ by Overmars, van Leeuwen 1981

Vertical line queries

One query to a secondary data structure for each level of the interval tree

$$O\left(\frac{\log n}{\log \log n} \cdot \log(\log^4 n)\right) = O(\log n)$$

BRICS

Fully Dynamic Case : Insertions

Inserting/deleting blocks in the interval tree

Searching + (block size) · (update time secondary structures)

$$O\left(\log n + \frac{\log n}{\log\log n} \cdot U(\log^4 n)\right)$$

$\Rightarrow$ $O(U(\log^4 n))$ **per segment**

Insertions

Each point can "pop up" in $\leq \frac{\log n}{\log\log n}$ **convex hulls**

Total cost for n **insertions is bounded by the time for block operations on the interval tree**

$$O\left(n \cdot \frac{\log n}{\log\log n} \cdot U(\log^4 n)\right)$$

$\Rightarrow$ $O_{\mathbf{A}}\left(\frac{\log n}{\log\log n} \cdot U(\log^4 n)\right)$ **per point**

BRICS

Fully Dynamic Case : Deletions

Deletions

Delete the point from one deletions-only data structure
+ perform $O(1)$ block updates on the interval tree
+ new points may "pop up" on a convex hull

Last two terms can be charged to the insertions
The deletion time is inherited from the deletions-only data structure, say denoted $D(n)$

$\Rightarrow O_{\mathbf{A}}(D(n))$ **per point**

BRICS

Transformation Result

Given a deletions-only CH data structure with $O_{\mathbf{A}}(n)$ preprocessing and $O_{\mathbf{A}}(D(n))$ deletion time, and a fully dynamic CH data structure with $O_{\mathbf{A}}(U(n))$ update time and $O(\log n)$ query time, there exists a CH data structure with

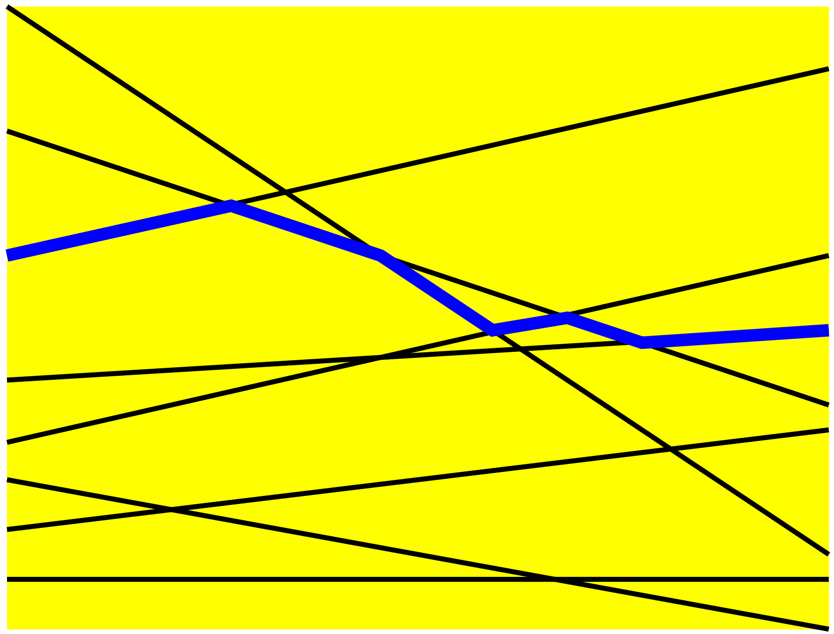
Query	$O(\log n)$
Insert	$O_{\mathbf{A}}\left(\frac{\log n}{\log\log n} \cdot U(\log^4 n)\right)$
Delete	$O_{\mathbf{A}}(D(n))$

Using our deletions-only data structure and the fully dynamic data structure of Overmars and van Leeuwen

Query	$O(\log n)$		$O_{\mathbf{A}}(\log n)$
Insert	$O_{\mathbf{A}}(\log n \cdot \log\log n)$	$\Longrightarrow$	$O_{\mathbf{A}}(\log n \cdot \log\log\log n)$
Delete	$O_{\mathbf{A}}(\log n \cdot \log\log n)$		$O_{\mathbf{A}}(\log n \cdot \log\log n)$

Application: k-Level Problem

Input n **lines and integer** k
Output **The** k**-level of the lines**



$O(n \cdot \log n + m \cdot \log^2 n)$ **using Overmars and van Leeuwen where** m **size of output** Edelsbrunner, Welzl 1986

Corollary $O(n \cdot \log n + m \cdot \log n \cdot \log\log n)$

BRICS

Conclusion and Open Problems

Result

Data structure for the dynamic planar convex hull problem

Query	$O(\log n)$
Insert	$O_{\mathbf{A}}(\log n \cdot \log\log\log n)$
Delete	$O_{\mathbf{A}}(\log n \cdot \log\log n)$

Open Problems

- **Achieve $O(\log n)$ update time**
- **Worst-case time bounds**
- **More advanced queries**

BRICS