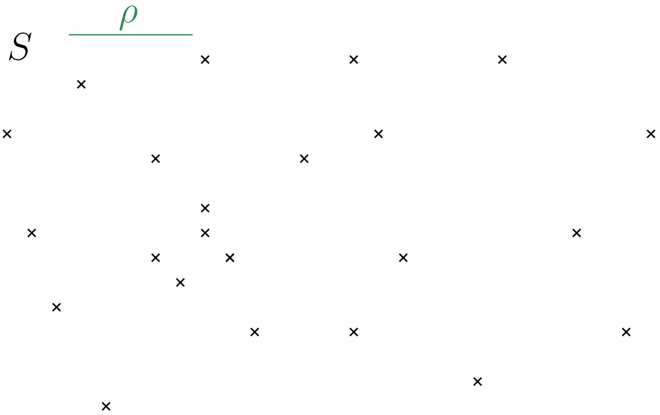
Range Reporting for Time Series via Rectangle Stabbing

Lotte Blank and Anne Driemel

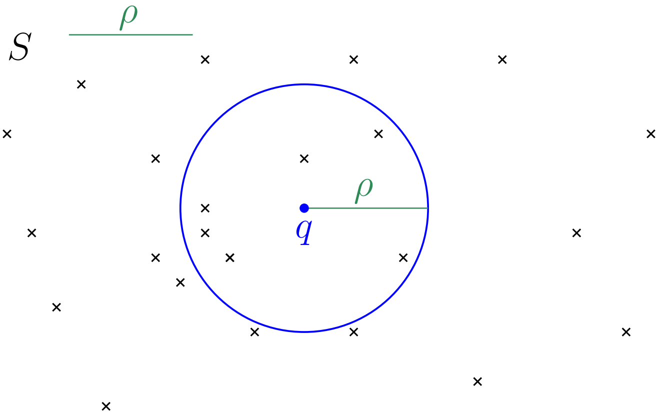
Institute of Computer Science
University of Bonn
Germany

March 14th, 2024

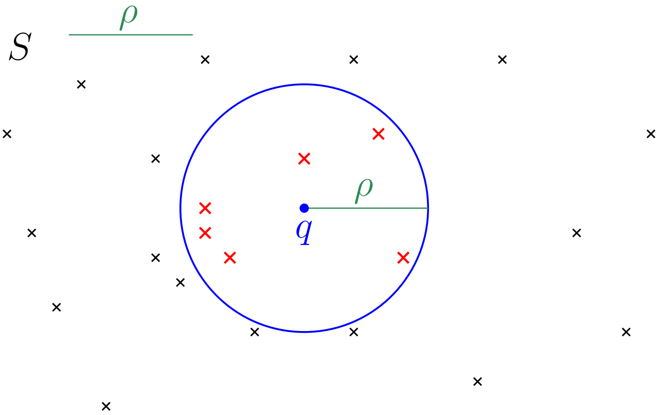
Range Reporting for Points



Range Reporting for Points



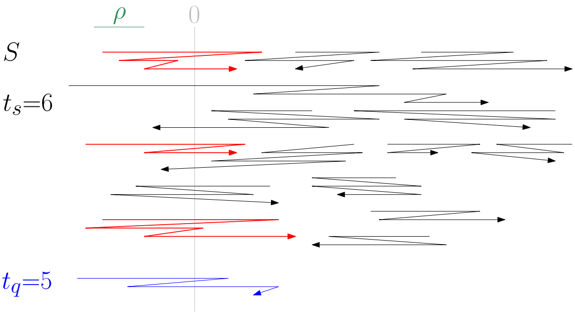
Range Reporting for Points



Problem (Fréchet Queries)

Given:
- set S of n time series all of complexity at most t_s,
- complexity of query time series t_q,
- distance parameter $\rho \in \mathbb{R}_{\geq 0}$

Query task: for time series q of complexity t_q
- Return $\forall s \in S$ with $d_F(s, q) \leq \rho$

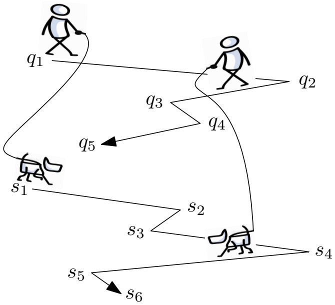


Definition: continuous Fréchet distance

Intuitive Definition:

Continuous Fréchet distance is **length of the shortest leash** such that person and dog traverse the paths

- From start to finish
- Can very speed
- Can't go backward



Rectangle Stabbing

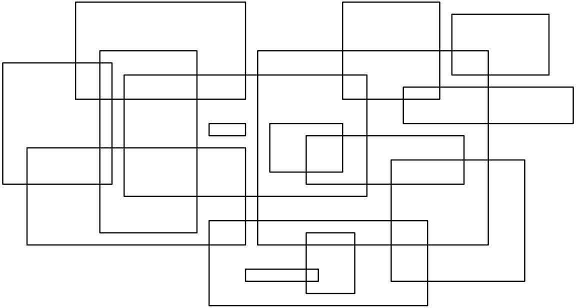
Problem (Rectangle Stabbing)

Given:

- *set S of n axis-aligned d-dimensional rectangles in $\mathbb{R}^d$*

Query task: for point $q \in \mathbb{R}^d$

- *Return $\forall R \in S$ with $q \in R$*



Rectangle Stabbing

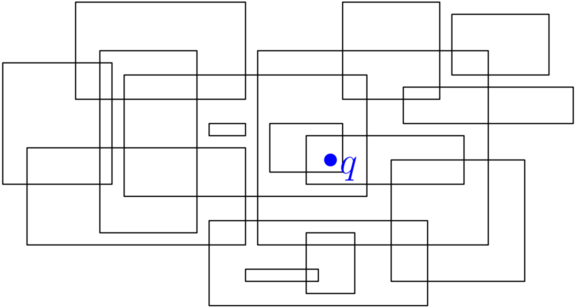
Problem (Rectangle Stabbing)

Given:

- *set S of n axis-aligned d-dimensional rectangles in $\mathbb{R}^d$*

Query task: for point $q \in \mathbb{R}^d$

- *Return $\forall R \in S$ with $q \in R$*



Rectangle Stabbing

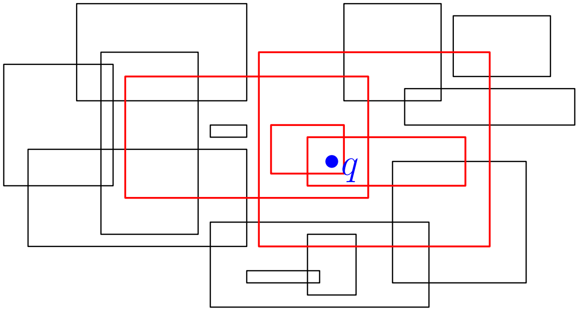
Problem (Rectangle Stabbing)

Given:

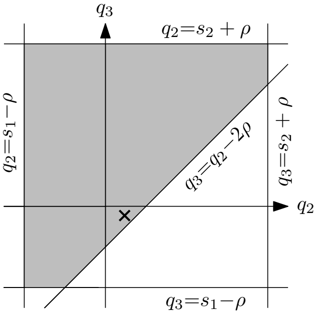
- *set S of n axis-aligned d-dimensional rectangles in $\mathbb{R}^d$*

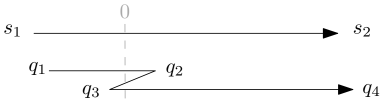
Query task: for point $q \in \mathbb{R}^d$

- *Return $\forall R \in S$ with $q \in R$*



Example: Range of a Time Series s





$d_F(q,s) \leq \rho$ if and only if

- $q_1 \in [s_1-\rho, s_1+\rho]$ and $q_4 \in [s_2-\rho, s_2+\rho]$
- (q_2, q_3) in grey area

Results for the Fréchet Queries Problem

	d	Storage	Query Time
[Afshani, Driemel 18]	2	$\mathcal{O}\left(n(\log\log n)^{\mathcal{O}\left({t_s}^2\right)}\right)$	$\mathcal{O}\left(\sqrt{n}\log^{\mathcal{O}\left({t_s}^2\right)} n + k\right)$
[Cheng, Huang 23]	d	$\mathcal{O}\left(t_q t_s n\right)^{\mathcal{O}\left(d^4 {t_q}^2 \log(d t_q)\right)}$	$\mathcal{O}\left((d t_q)^{\mathcal{O}(1)} \log(t_q t_s n) + k\right)$
Our Results	1	$\mathcal{O}\left(n\left(\frac{\log n}{\log\log n}\right)^{t_s-1}\right)$	$\mathcal{O}\left(\log n\left(\frac{\log n}{\log\log n}\right)^{t_s-3} + k\right)$
	1	$\mathcal{O}\left(n\log^{t_q-2} n\right)$	$\mathcal{O}\left(\log^{t_q-1} n + k\right)$

Results for the Fréchet Queries Problem

	d	Storage	Query Time
[Afshani, Driemel 18]	2	$\mathcal{O}\left(n(\log\log n)^{\mathcal{O}\left({t_s}^2\right)}\right)$	$\mathcal{O}\left(\sqrt{n}\log^{\mathcal{O}\left({t_s}^2\right)} n + k\right)$
[Cheng, Huang 23]	d	$\mathcal{O}\left(t_q t_s n\right)^{\mathcal{O}\left(d^4 {t_q}^2 \log(d t_q)\right)}$	$\mathcal{O}\left((d t_q)^{\mathcal{O}(1)} \log(t_q t_s n) + k\right)$
Our Results	1	$\mathcal{O}\left(n\left(\frac{\log n}{\log\log n}\right)^{t_s-1}\right)$	$\mathcal{O}\left(\log n\left(\frac{\log n}{\log\log n}\right)^{t_s-3} + k\right)$
	1	$\mathcal{O}\left(n\log^{t_q-2} n\right)$	$\mathcal{O}\left(\log^{t_q-1} n + k\right)$

Lower bounds:[1]	Storage		Query Time
d=1	nh	$\Rightarrow$	$\Omega\left(\log n\left(\frac{\log n}{\log h}\right)^{\lfloor t_{\min}/2\rfloor-2} + k\right)$
d=1	$\Omega\left(n\left(\frac{\log n}{\log\log n}\right)^{\lfloor t_{\min}/2\rfloor-1}\right)$	$\Leftarrow$	$\mathcal{O}(\log^c n + k)$

[1] in pointer machine model

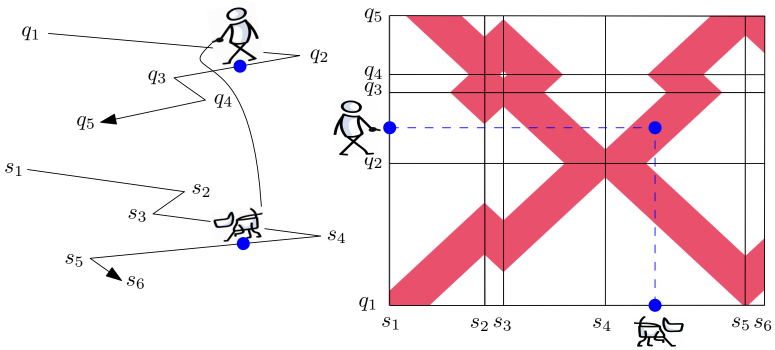
Results for the Fréchet Queries Problem

	d	Storage	Query Time
[Afshani, Driemel 18]	2	$\mathcal{O}\left(n(\log\log n)^{\mathcal{O}\left({t_s}^2\right)}\right)$	$\mathcal{O}\left(\sqrt{n}\log^{\mathcal{O}\left({t_s}^2\right)} n + k\right)$
[Cheng, Huang 23]	d	$\mathcal{O}\left(t_q t_s n\right)^{\mathcal{O}\left(d^4 {t_q}^2 \log(d t_q)\right)}$	$\mathcal{O}\left((d t_q)^{\mathcal{O}(1)} \log(t_q t_s n) + k\right)$
Our Results	1	$\mathcal{O}\left(n\left(\frac{\log n}{\log\log n}\right)^{t_s-1}\right)$	$\mathcal{O}\left(\log n\left(\frac{\log n}{\log\log n}\right)^{t_s-3} + k\right)$
	1	$\mathcal{O}\left(n\log^{t_q-2} n\right)$	$\mathcal{O}\left(\log^{t_q-1} n + k\right)$

Lower bounds:[1]	Storage		Query Time
d=1	nh	$\Rightarrow$	$\Omega\left(\log n\left(\frac{\log n}{\log h}\right)^{\lfloor t_{\min}/2\rfloor-2} + k\right)$
d=1	$\Omega\left(n\left(\frac{\log n}{\log\log n}\right)^{\lfloor t_{\min}/2\rfloor-1}\right)$	$\Leftarrow$	$\mathcal{O}(\log^c n + k)$

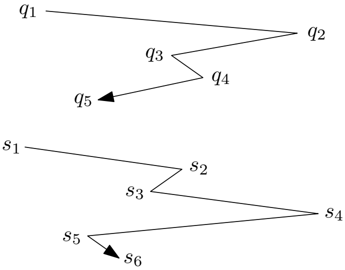
[1] in pointer machine model

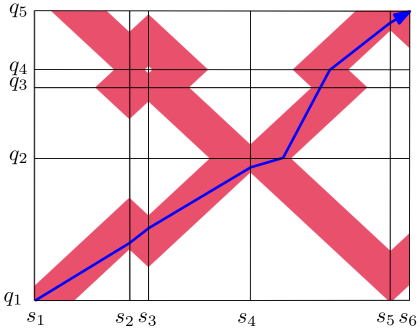
Free Space Diagram $F_\rho(q, s)$



$$F_\rho(q, s) = \{(x, y) \in [1, t_s] \times [1, t_q] \mid |q(x) - s(y)| \leq \rho\}$$

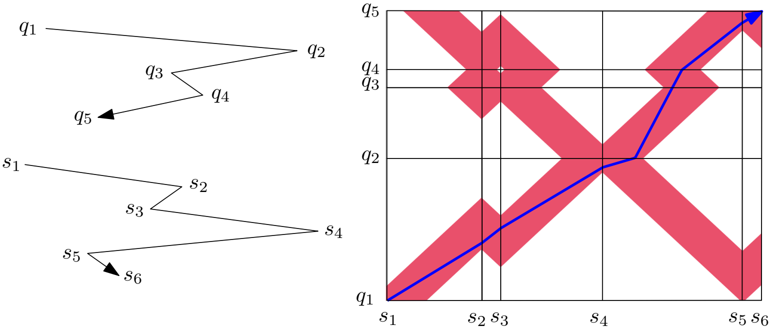
Free Space Diagram $F_\rho(q, s)$





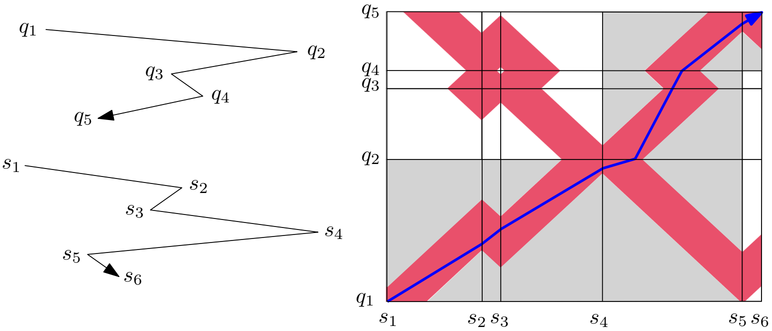
- $d_F(q, s) \leq \rho \Leftrightarrow \exists$ a path monotone in both coordinates from $(1, 1)$ to (t_s, t_q) in the free space.
- Such a path is called **feasible path**.

Free Space Diagram $F_\rho(q, s)$



- $d_F(q, s) \leq \rho \Leftrightarrow \exists$ a path monotone in both coordinates from $(1, 1)$ to (t_s, t_q) in the free space.
- Such a path is called **feasible path**.

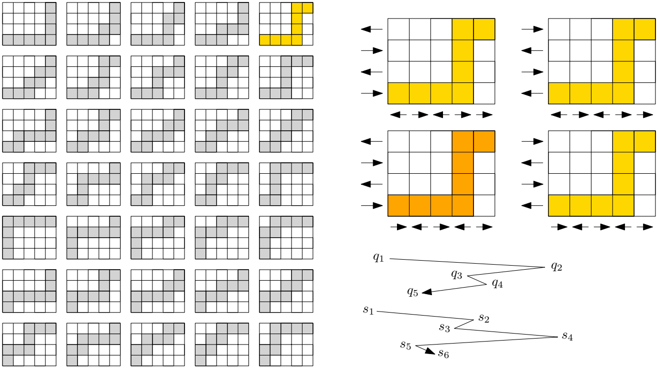
Free Space Diagram $F_\rho(q, s)$



- $d_F(q, s) \leq \rho \Leftrightarrow \exists$ a path monotone in both coordinates from $(1, 1)$ to (t_s, t_q) in the free space.
- Such a path is called **feasible path**.

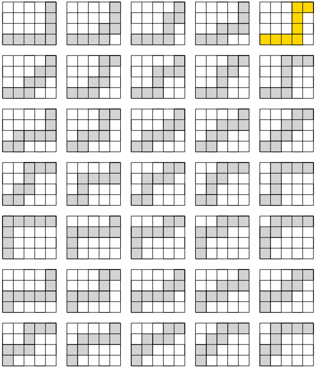
The Data Structure

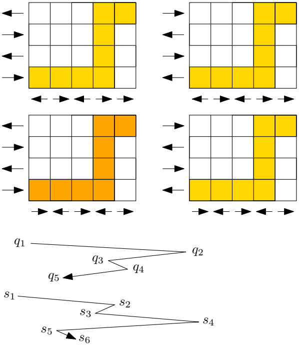
For every (valid) sequence of cells: Store 4 rectangle stabbing data structures



The Data Structure

For every (valid) sequence of cells: Store 4 rectangle stabbing data structures



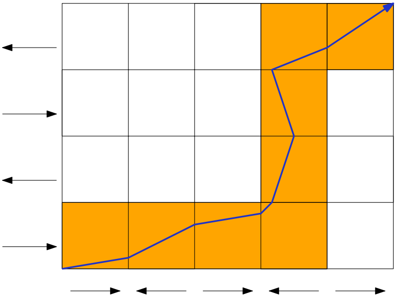


In every such data structure: Store at most one t_q-dim. rectangle per time series in S.

Defining the Intervals

Lemma

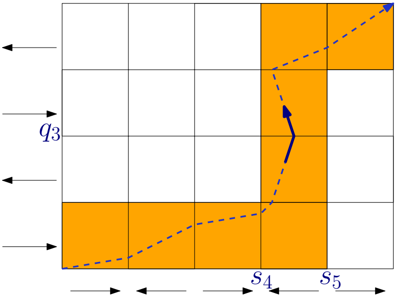
Given: sequence of cells $\mathscr{C}$*, time series* s*, distance* ρ*, edge directions of* q
Then: Exists intervals $I_1, \ldots, I_{t_q}$ *such that* $(q_1, \ldots, q_{t_q}) \in I_1 \times \ldots \times I_{t_q} \Leftrightarrow \exists$ *feasible path traversing* $\mathscr{C}$ *except for monotonicity x-coordinate*



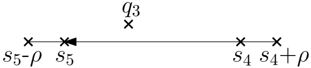
Defining the Intervals

Lemma

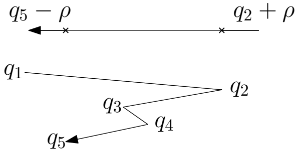
Given: sequence of cells $\mathscr{C}$, time series s, distance ρ, edge directions of q
Then: Exists intervals $I_1, \ldots, I_{t_q}$ such that $(q_1, \ldots, q_{t_q}) \in I_1 \times \ldots \times I_{t_q} \Leftrightarrow \exists$ feasible path traversing $\mathscr{C}$ except for monotonicity x-coordinate



For example,
$q_3 \in [s_5 - \rho, s_4 + \rho]$.



Key Concept: Backward and Forward Numbers

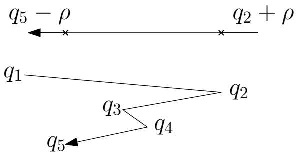


Here, $b_2(q) = 5$.

Backward number $b_i(q)$: highest index $k \in \{i, \ldots, t_q\}$ s.t.

- $d_F(\langle q_i, \ldots, q_k \rangle, \longleftarrow) \leq \rho$

Key Concept: Backward and Forward Numbers



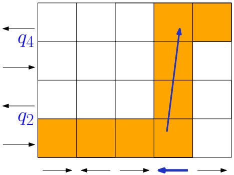
Here, $b_2(q) = 5$.

Backward number $b_i(q)$: highest index $k \in \{i, \ldots, t_q\}$ s.t.

- $d_F(\langle q_i, \ldots, q_k \rangle, \longleftarrow) \leq \rho$

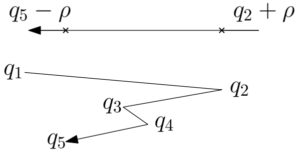
Backward number $b_i(\mathscr{C})$: highest index $k \in \{i, \ldots, t_q\}$ s.t.

- $(i-1, j), \ldots, (k, j) \in \mathscr{C}$
- $\overline{s_j s_{j+1}}$ oriented backward



Here, $b_2(\mathscr{C}) = 4$.

Key Concept: Backward and Forward Numbers



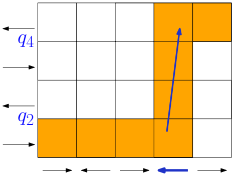
Here, $b_2(q) = 5$.

Backward number $b_i(q)$: highest index $k \in \{i, \ldots, t_q\}$ s.t.

- $d_F(\langle q_i, \ldots, q_k \rangle, \longleftarrow) \leq \rho$

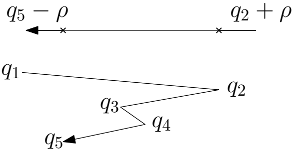
Backward number $b_i(\mathscr{C})$: highest index $k \in \{i, \ldots, t_q\}$ s.t.

- $(i-1, j), \ldots, (k, j) \in \mathscr{C}$
- $\overline{s_j s_{j+1}}$ oriented backward



Here, $b_2(\mathscr{C}) = 4$.

Similarly, define forward numbers $f_i(q)$ and $f_i(\mathscr{C})$.



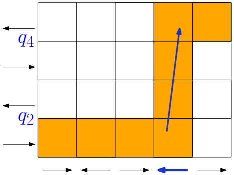
Here, $b_2(q) = 5$.

Backward number $b_i(q)$: highest index $k \in \{i, \ldots, t_q\}$ s.t.

- $d_F(\langle q_i, \ldots, q_k \rangle, \longleftarrow) \leq \rho$

Backward number $b_i(\mathscr{C})$: highest index $k \in \{i, \ldots, t_q\}$ s.t.

- $(i-1, j), \ldots, (k, j) \in \mathscr{C}$
- $\overline{s_j s_{j+1}}$ oriented backward



Here, $b_2(\mathscr{C}) = 4$.

Similarly, define forward numbers $f_i(q)$ and $f_i(\mathscr{C})$.

Lemma (Equivalent statements)

- *$\exists$ feasible path in $F_\rho(q, s)$ traversing exactly cells in $\mathscr{C}$*
- *$b_i(q) \geq b_i(\mathscr{C})$, $f_i(q) \geq f_i(\mathscr{C})$ and $q_i \in I_i$ for all i*

The Query Algorithm

The Query Algorithm.

Compute $f_1(q), \ldots, f_{t_q}(q), b_1(q), \ldots, b_{t_q}(q)$

for *all valid sequences of cells* $\mathscr{C}$ **do**

if $f_i(\mathscr{C}) \leq f_i(q)$ *and* $b_i(\mathscr{C}) \leq b_i(q)$ *for all* i **then**

Query search in associated rectangle stabbing data structure with $(q_1, q_2, \ldots, q_{t_q})$

Output all time series associated with rectangle containing this point

The Query Algorithm.

Compute $f_1(q), \ldots, f_{t_q}(q), b_1(q), \ldots, b_{t_q}(q)$
for *all valid sequences of cells* $\mathscr{C}$ **do**
    if $f_i(\mathscr{C}) \leq f_i(q)$ *and* $b_i(\mathscr{C}) \leq b_i(q)$ *for all* i **then**
        Query search in associated rectangle stabbing data structure with $(q_1, q_2, \ldots, q_{t_q})$
        Output all time series associated with rectangle containing this point

The Query Algorithm

The Query Algorithm.

Compute $f_1(q), \ldots, f_{t_q}(q), b_1(q), \ldots, b_{t_q}(q)$
for *all valid sequences of cells* $\mathscr{C}$ **do**
  if $f_i(\mathscr{C}) \leq f_i(q)$ *and* $b_i(\mathscr{C}) \leq b_i(q)$ *for all* i **then**
    Query search in associated rectangle stabbing data structure with $(q_1, q_2, \ldots, q_{t_q})$
    Output all time series associated with rectangle containing this point

The Query Algorithm

The Query Algorithm.

Compute $f_1(q), \ldots, f_{t_q}(q), b_1(q), \ldots, b_{t_q}(q)$
for *all valid sequences of cells* $\mathscr{C}$ **do**
  if $f_i(\mathscr{C}) \leq f_i(q)$ *and* $b_i(\mathscr{C}) \leq b_i(q)$ *for all* i **then**
    Query search in associated rectangle stabbing data structure with $(q_1, q_2, \ldots, q_{t_q})$
    Output all time series associated with rectangle containing this point

Result

Theorem

For constant t_s and t_q, there exists a data structure solving the Fréchet Queries problem

- *of size in $\mathcal{O}(n \log^{t_q - 2} n)$ and*
- *query time in $\mathcal{O}(\log^{t_q - 1} n + k)$.*

Proof.

- Correctness by discussion before
- Rectangle stabbing data structure by Chazelle (1986) with size in $\mathcal{O}(n \log^{d-2} n)$ and query time in $\mathcal{O}(\log^{d-1} n + k)$

Result

Theorem

For constant t_s and t_q, there exists a data structure solving the Fréchet Queries problem

- *of size in $\mathcal{O}(n \log^{t_q - 2} n)$ and*
- *query time in $\mathcal{O}(\log^{t_q - 1} n + k)$.*

Proof.

- Correctness by discussion before
- Rectangle stabbing data structure by Chazelle (1986) with size in $\mathcal{O}(n \log^{d-2} n)$ and query time in $\mathcal{O}(\log^{d-1} n + k)$

Thank you for your attention!
Questions?

Results for the Fréchet Queries Problem

	d	Storage	Query Time
[Afshani, Driemel 18]	2	$\mathcal{O}\left(n(\log\log n)^{\mathcal{O}\left({t_s}^2\right)}\right)$	$\mathcal{O}\left(\sqrt{n}\log^{\mathcal{O}\left({t_s}^2\right)} n + k\right)$
[Cheng, Huang 23]	d	$\mathcal{O}\left(t_q t_s n\right)^{\mathcal{O}\left(d^4 {t_q}^2 \log(dt_q)\right)}$	$\mathcal{O}\left((dt_q)^{\mathcal{O}(1)}\log(t_q t_s n) + k\right)$
Our Results	1	$\mathcal{O}\left(n\left(\frac{\log n}{\log\log n}\right)^{t_s-1}\right)$	$\mathcal{O}\left(\log n\left(\frac{\log n}{\log\log n}\right)^{t_s-3} + k\right)$
	1	$\mathcal{O}\left(n\log^{t_q-2} n\right)$	$\mathcal{O}\left(\log^{t_q-1} n + k\right)$

Lower bounds:[1]	Storage		Query Time
d=1	nh	$\Rightarrow$	$\Omega\left(\log n\left(\frac{\log n}{\log h}\right)^{\lfloor t_{\min}/2\rfloor-2} + k\right)$
d=1	$\Omega\left(n\left(\frac{\log n}{\log\log n}\right)^{\lfloor t_{\min}/2\rfloor-1}\right)$	$\Leftarrow$	$\mathcal{O}(\log^c n + k)$

[1] in pointer machine model