Algorithms for Hotspot Computation on Trajectory Data

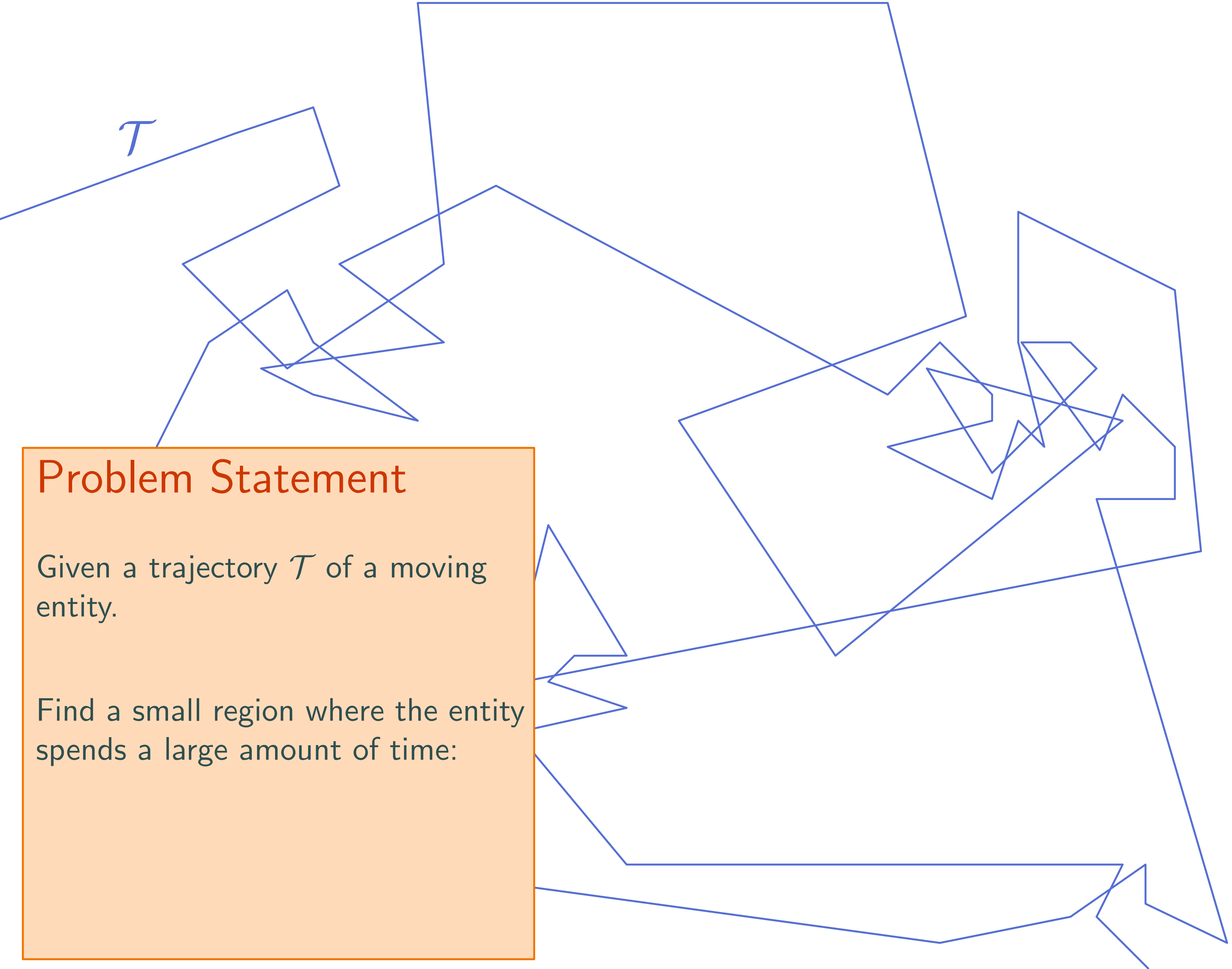
Joachim Gudmundsson

Marc van Kreveld

Frank Staals

University of Sydney

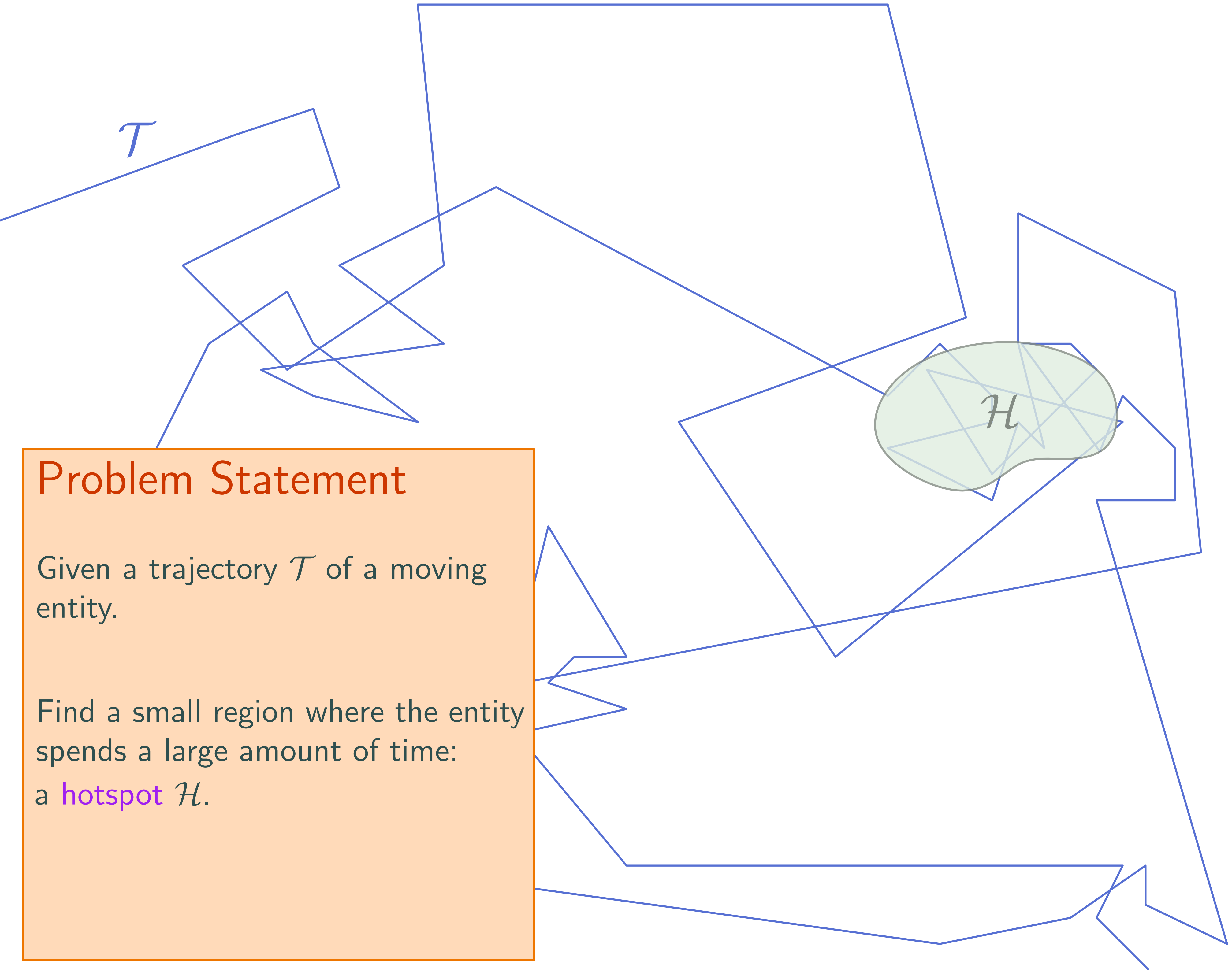
Utrecht University



Problem Statement

Given a trajectory $\mathcal{T}$ of a moving entity.

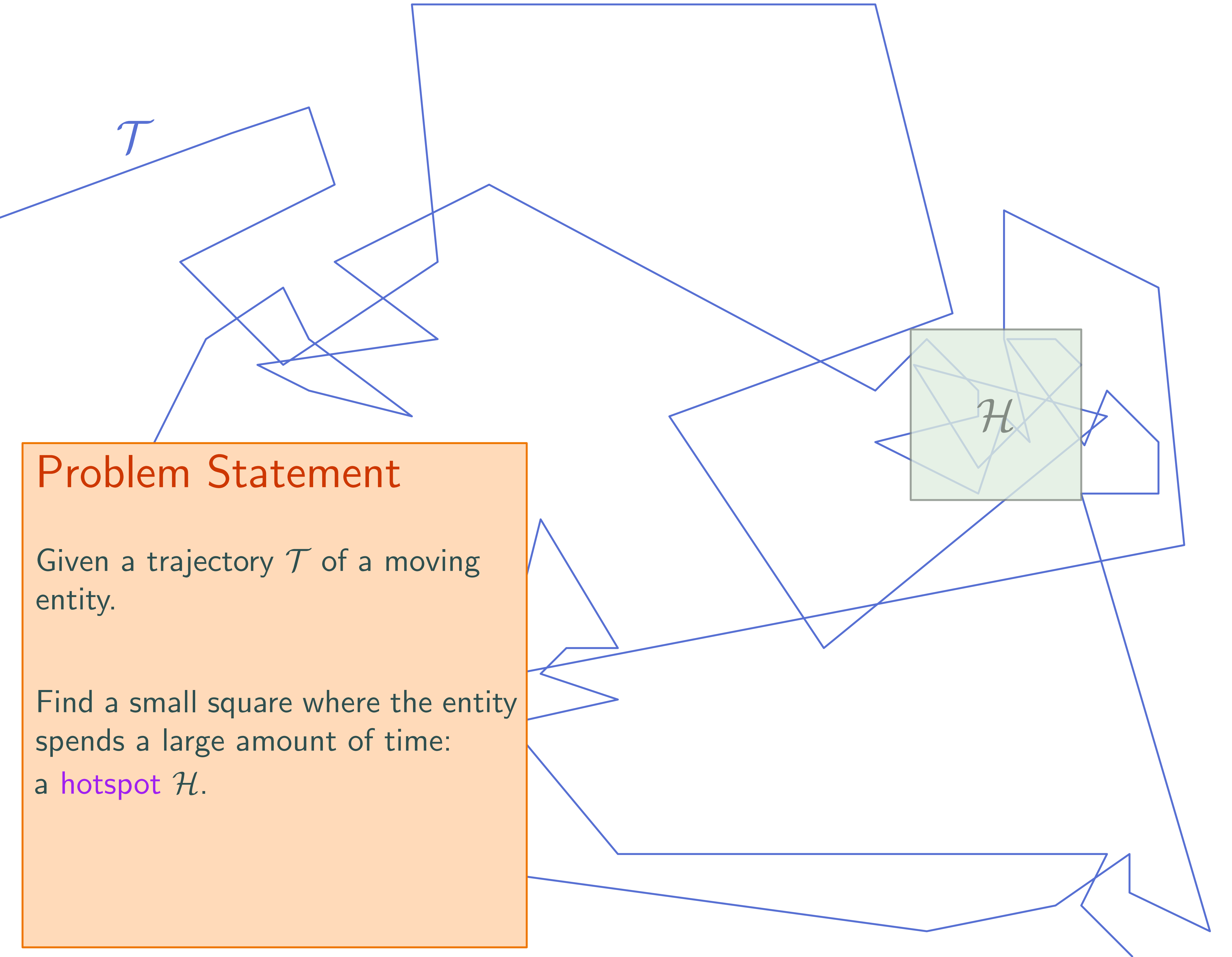
Find a small region where the entity spends a large amount of time:



Problem Statement

Given a trajectory $\mathcal{T}$ of a moving entity.

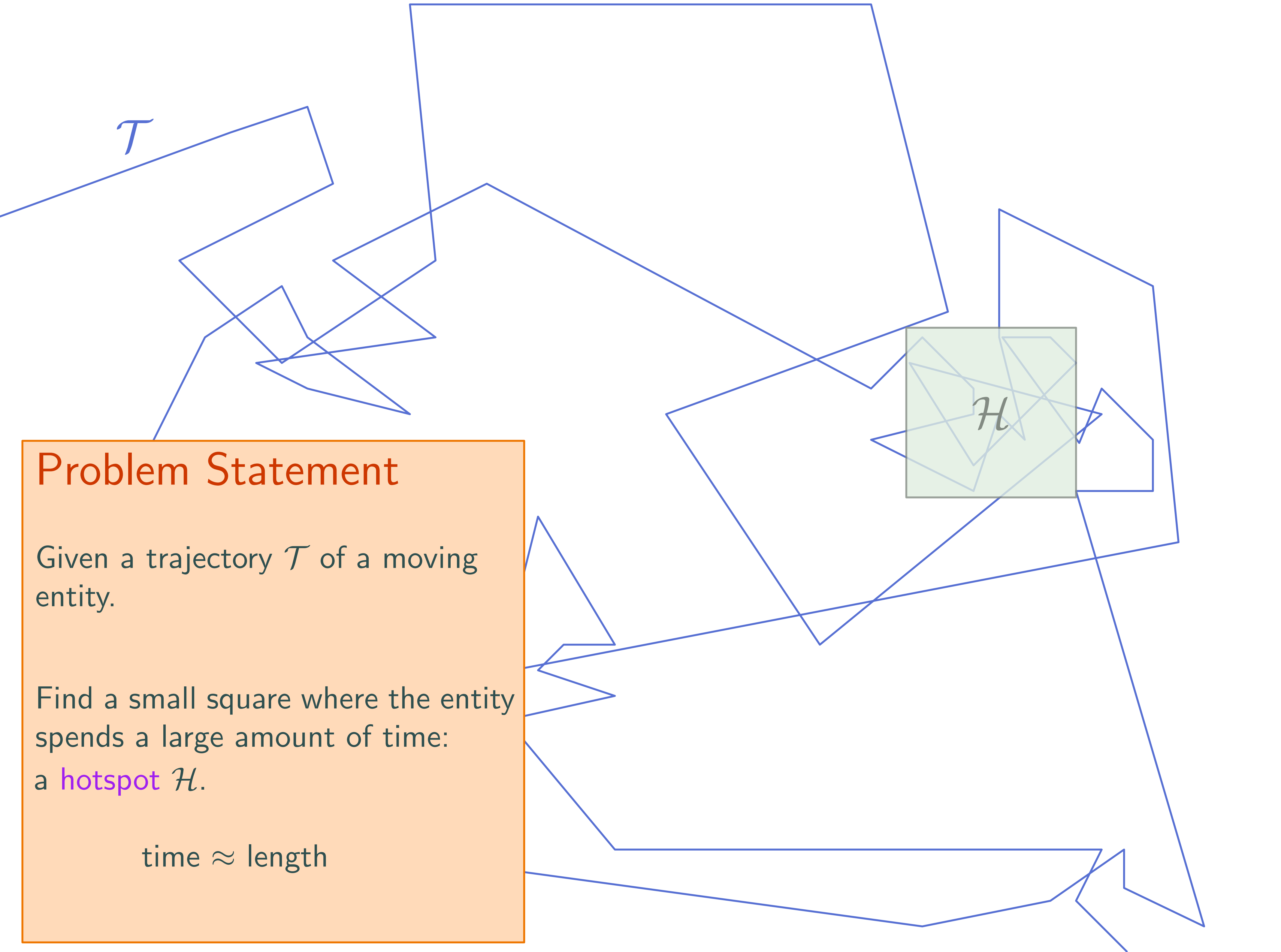
Find a small region where the entity spends a large amount of time: a hotspot $\mathcal{H}$.



Problem Statement

Given a trajectory $\mathcal{T}$ of a moving entity.

Find a small square where the entity spends a large amount of time: a hotspot $\mathcal{H}$.

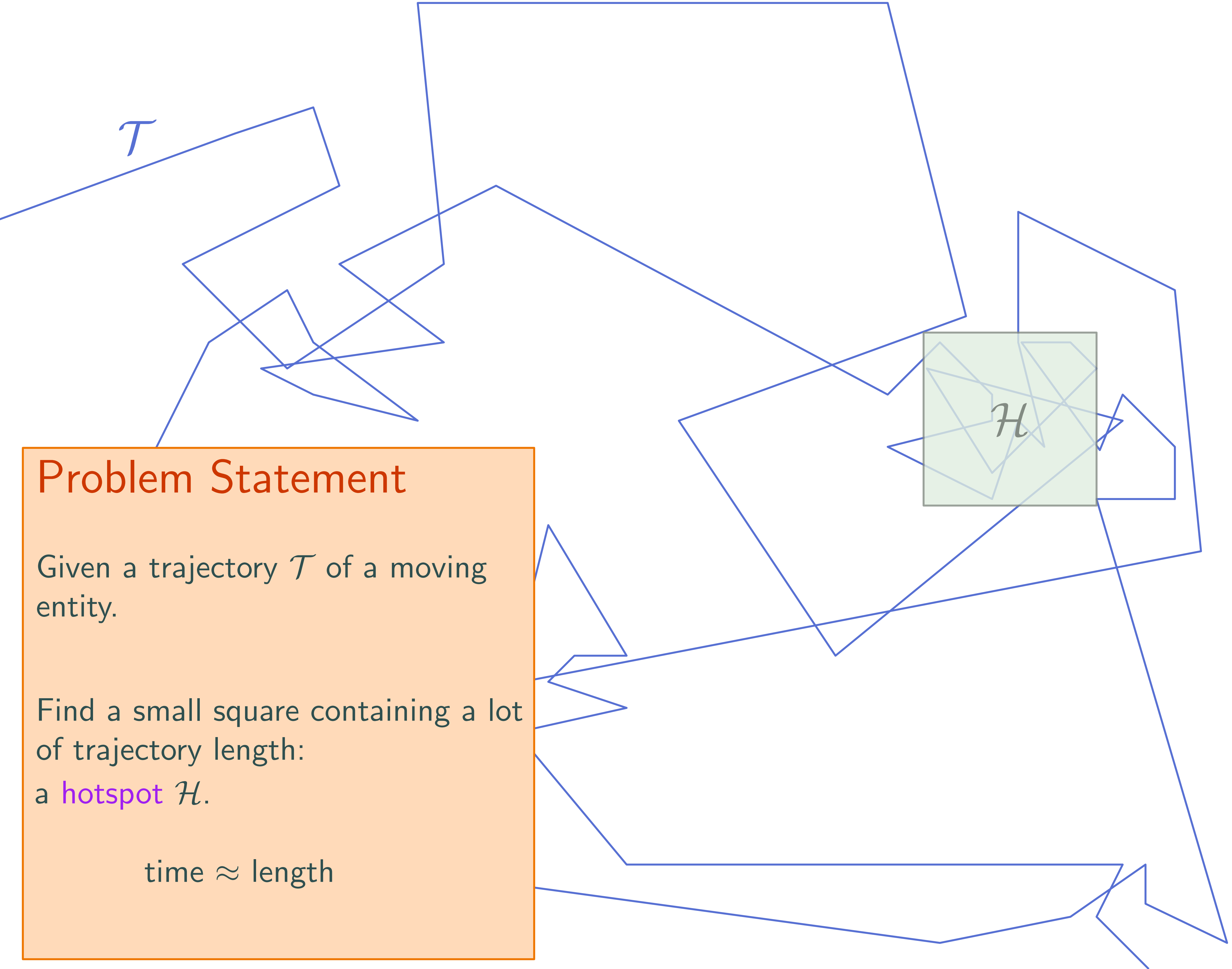


Problem Statement

Given a trajectory $\mathcal{T}$ of a moving entity.

Find a small square where the entity spends a large amount of time: a hotspot $\mathcal{H}$.

time $\approx$ length

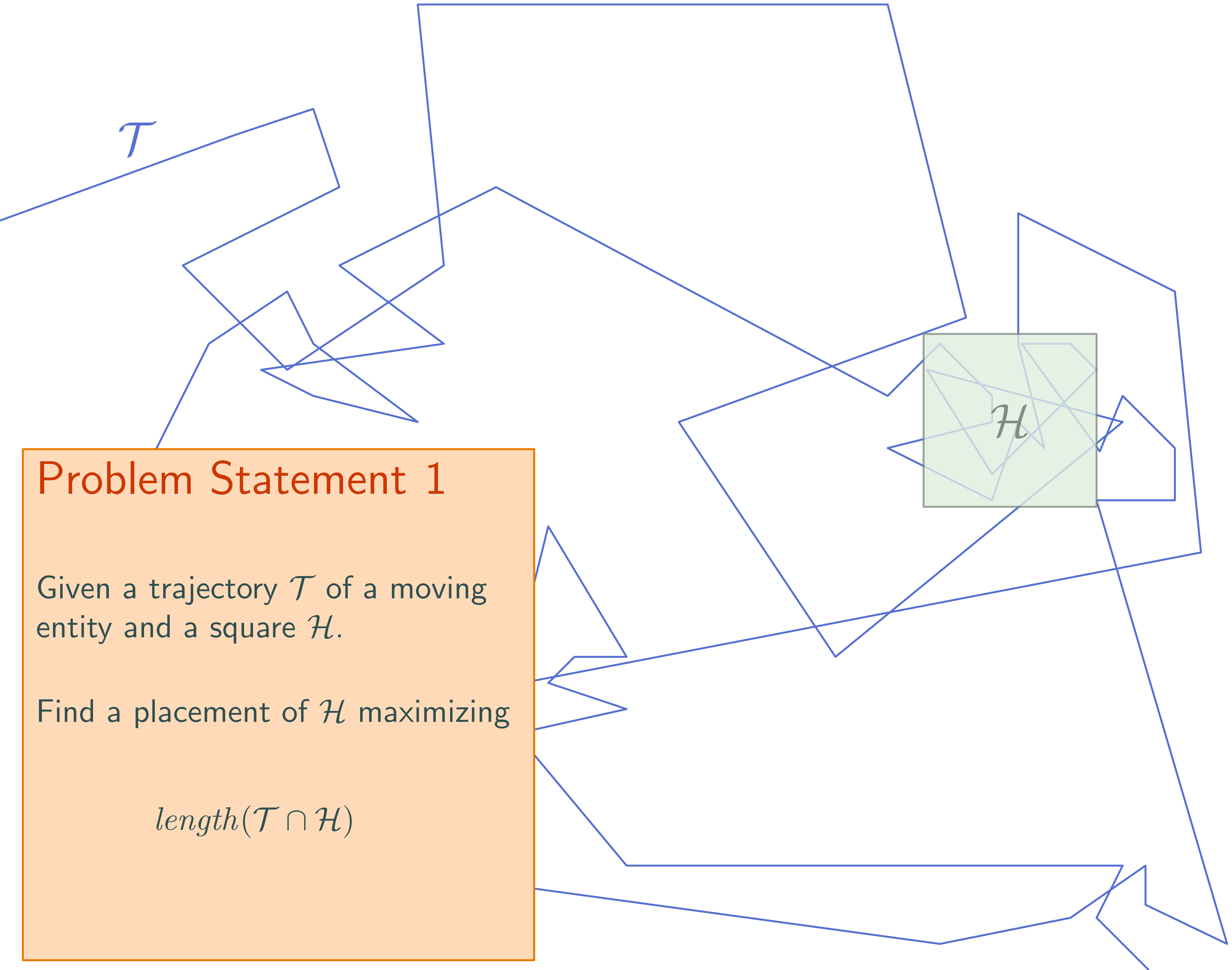


Problem Statement

Given a trajectory $\mathcal{T}$ of a moving entity.

Find a small square containing a lot of trajectory length:
a hotspot $\mathcal{H}$.

time $\approx$ length

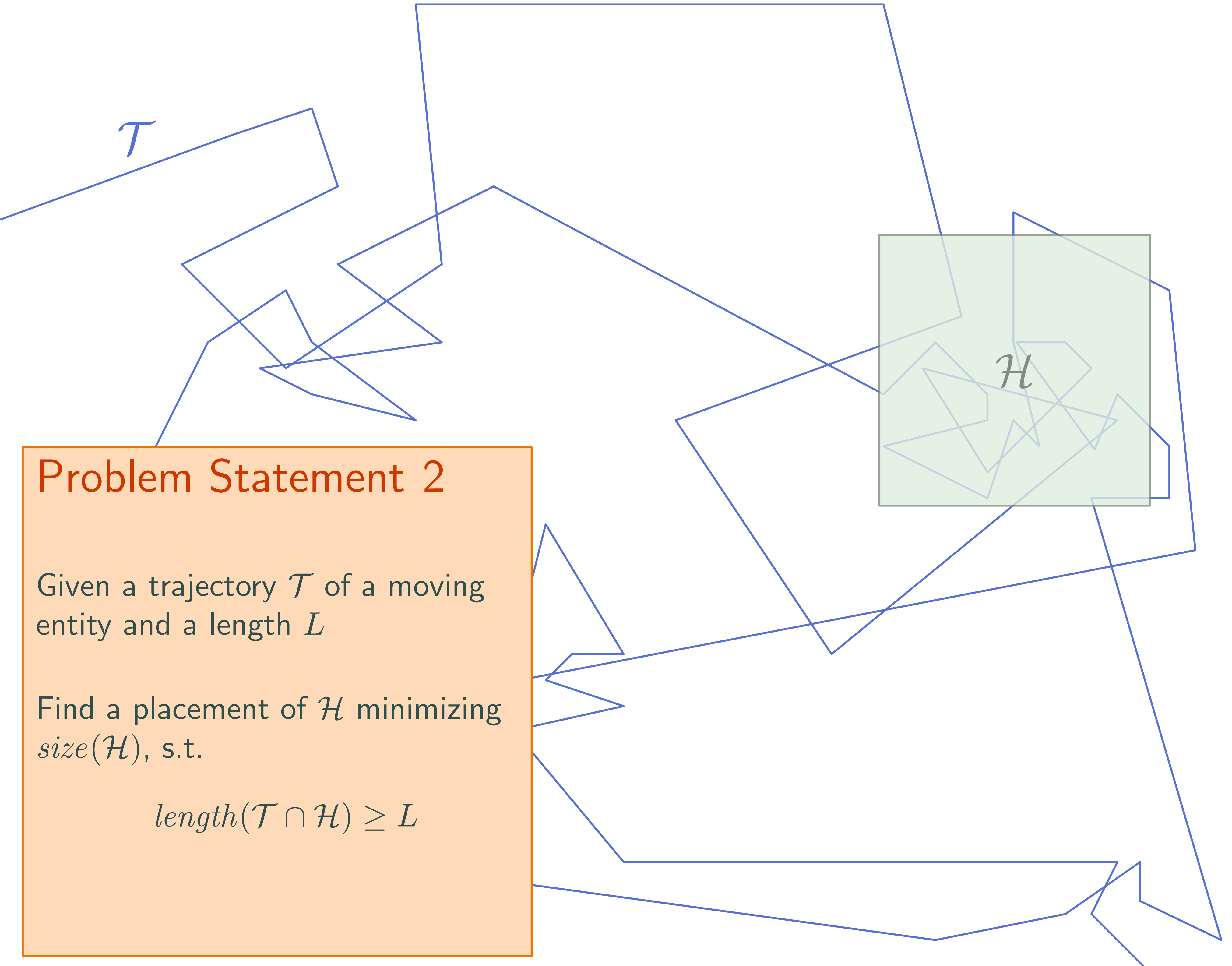


Problem Statement 1

Given a trajectory $\mathcal{T}$ of a moving entity and a square $\mathcal{H}$.

Find a placement of $\mathcal{H}$ maximizing

$$length(\mathcal{T} \cap \mathcal{H})$$

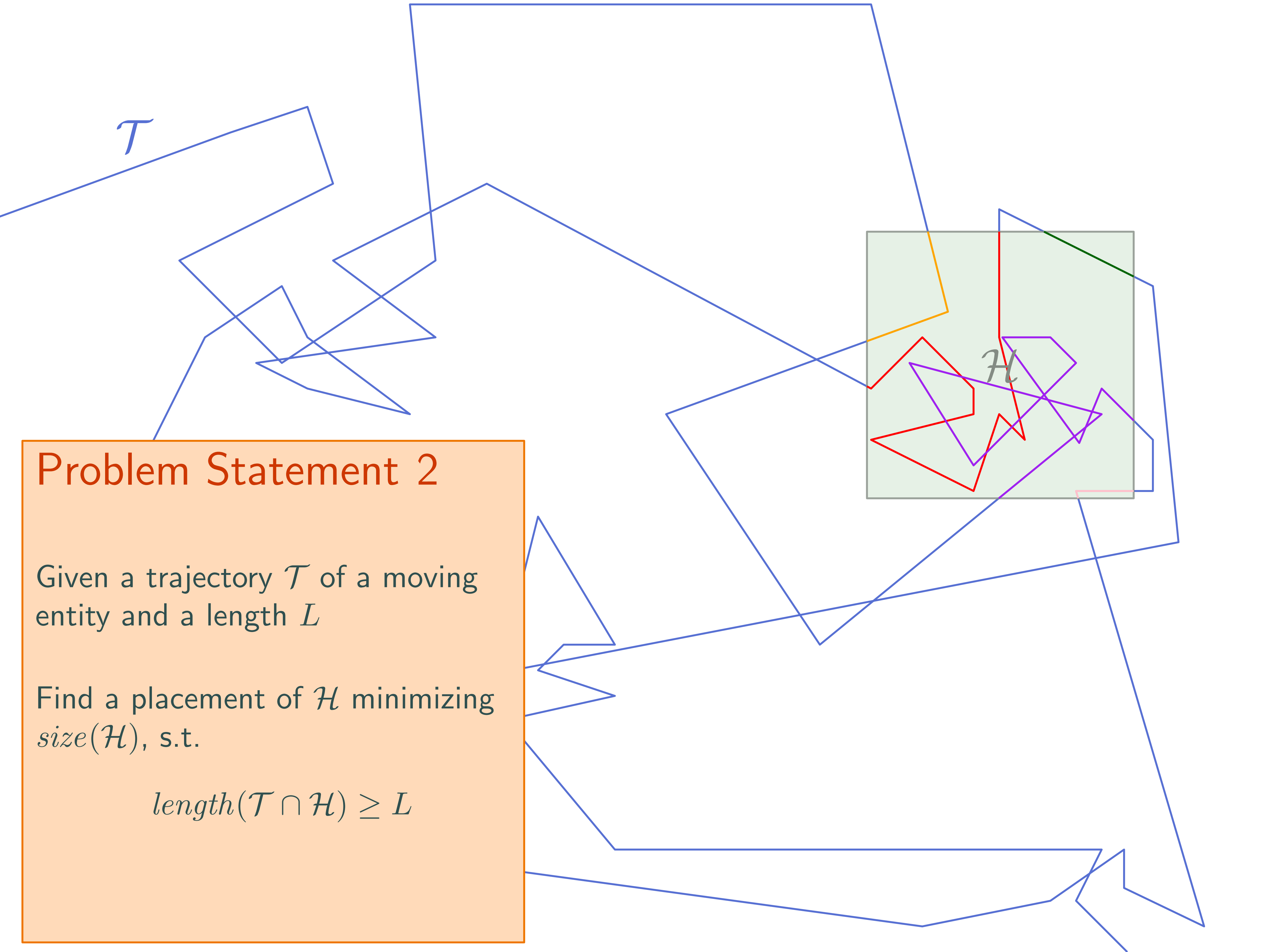


Problem Statement 2

Given a trajectory $\mathcal{T}$ of a moving entity and a length L

Find a placement of $\mathcal{H}$ minimizing $size(\mathcal{H})$, s.t.

$$length(\mathcal{T} \cap \mathcal{H}) \geq L$$

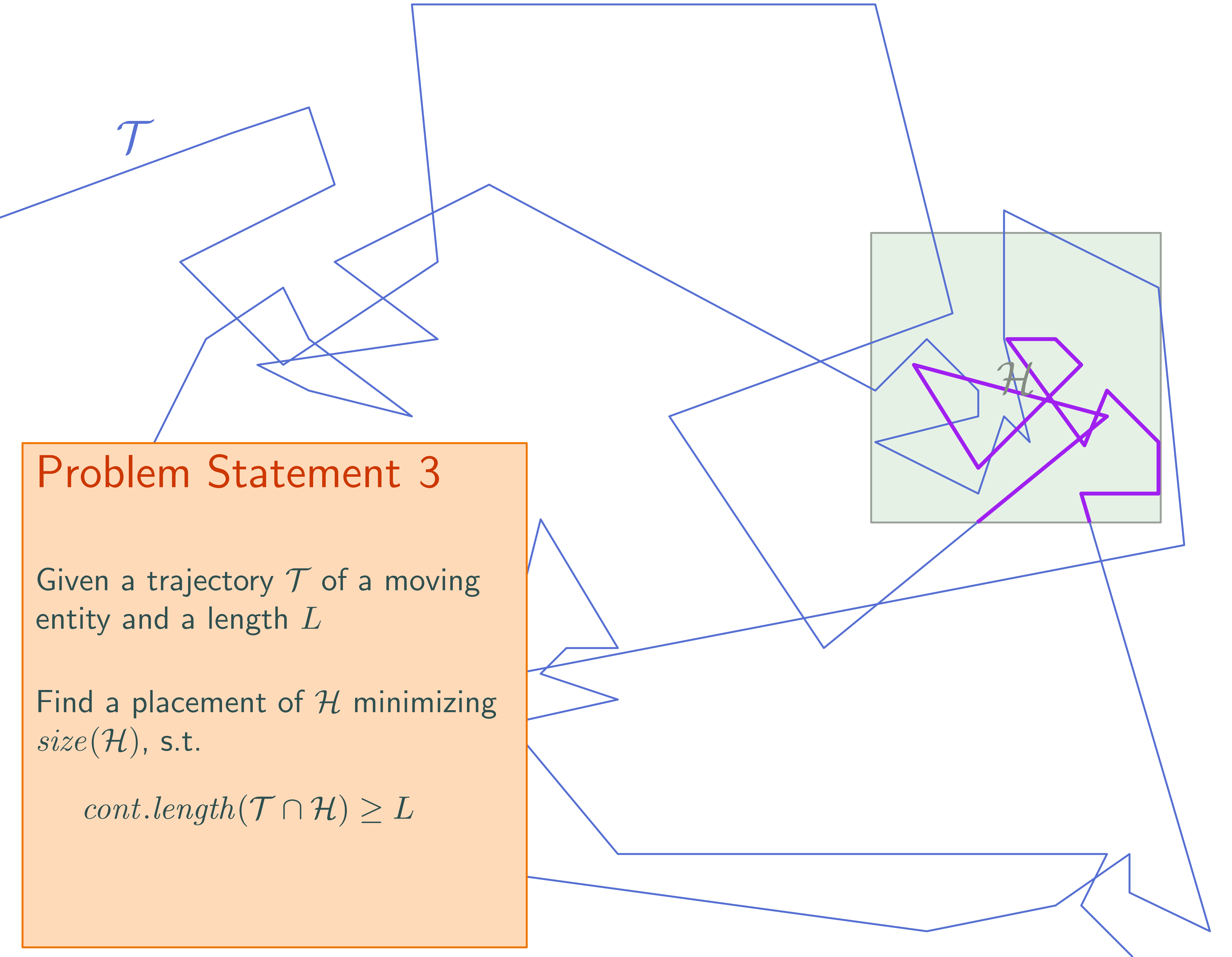


Problem Statement 2

Given a trajectory $\mathcal{T}$ of a moving entity and a length L

Find a placement of $\mathcal{H}$ minimizing $size(\mathcal{H})$, s.t.

$$length(\mathcal{T} \cap \mathcal{H}) \geq L$$

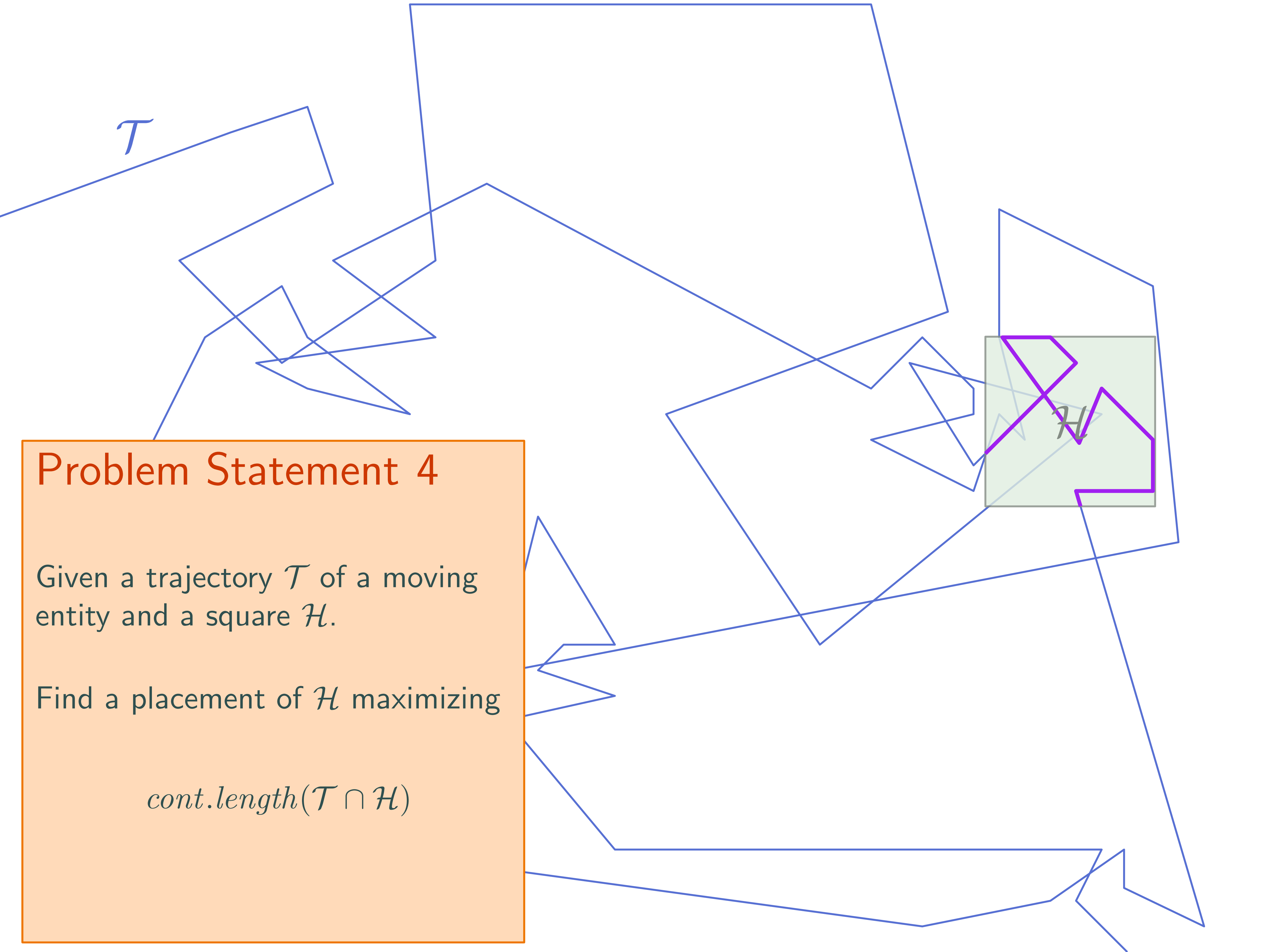


Problem Statement 3

Given a trajectory $\mathcal{T}$ of a moving entity and a length L

Find a placement of $\mathcal{H}$ minimizing $size(\mathcal{H})$, s.t.

$$cont.length(\mathcal{T} \cap \mathcal{H}) \geq L$$

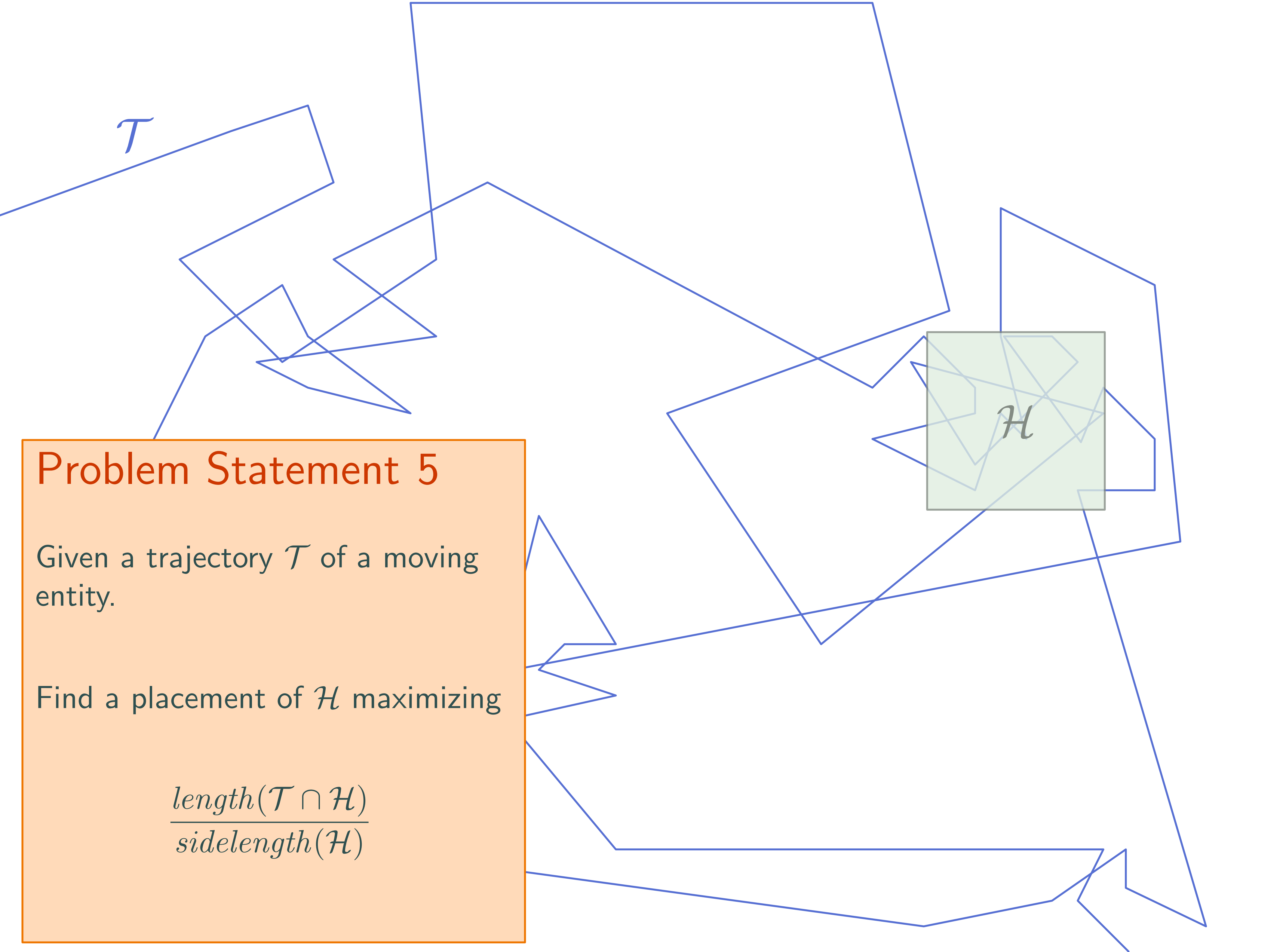


Problem Statement 4

Given a trajectory $\mathcal{T}$ of a moving entity and a square $\mathcal{H}$.

Find a placement of $\mathcal{H}$ maximizing

$$cont.length(\mathcal{T} \cap \mathcal{H})$$

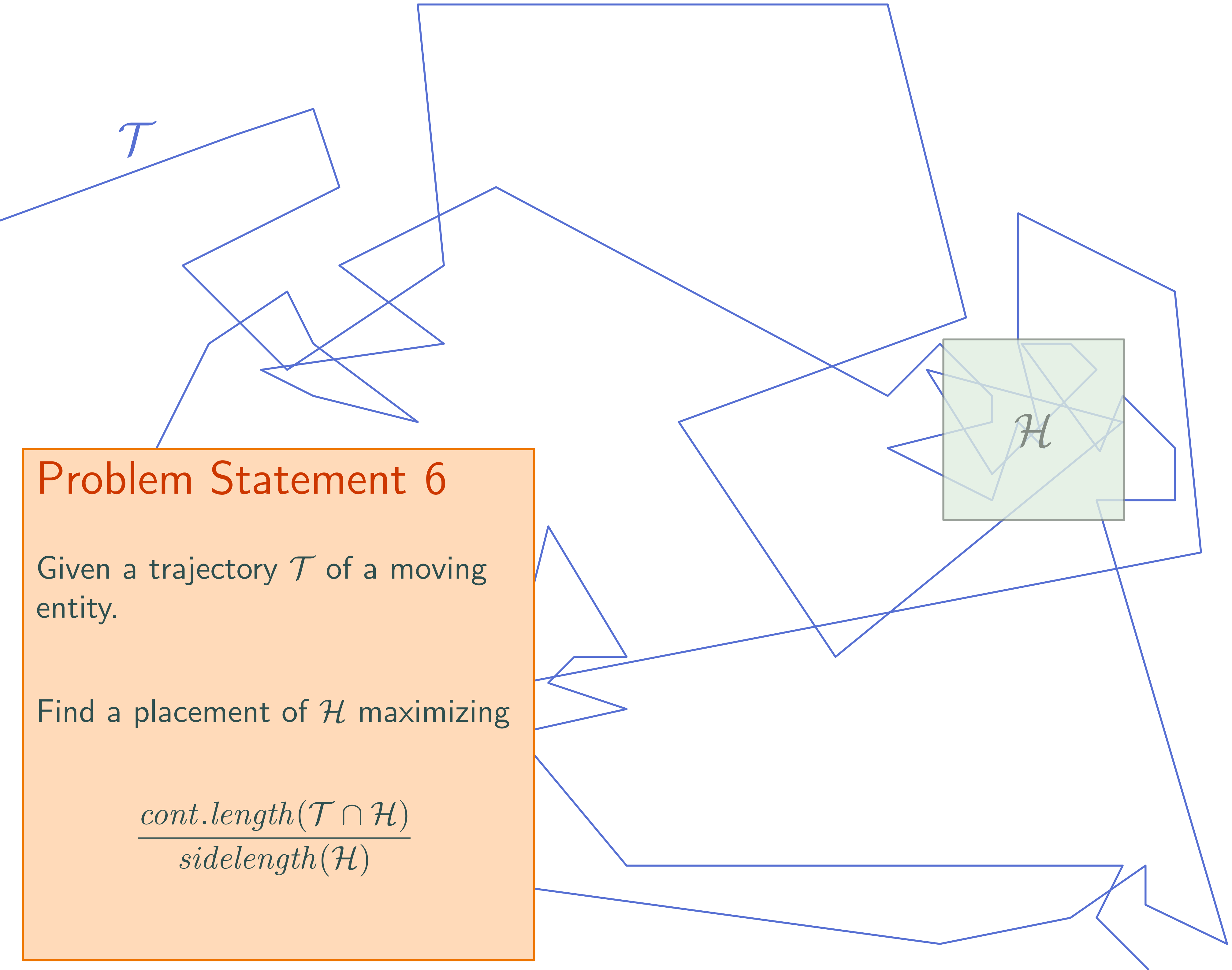


Problem Statement 5

Given a trajectory $\mathcal{T}$ of a moving entity.

Find a placement of $\mathcal{H}$ maximizing

$$\frac{length(\mathcal{T} \cap \mathcal{H})}{sidelength(\mathcal{H})}$$



Problem Statement 6

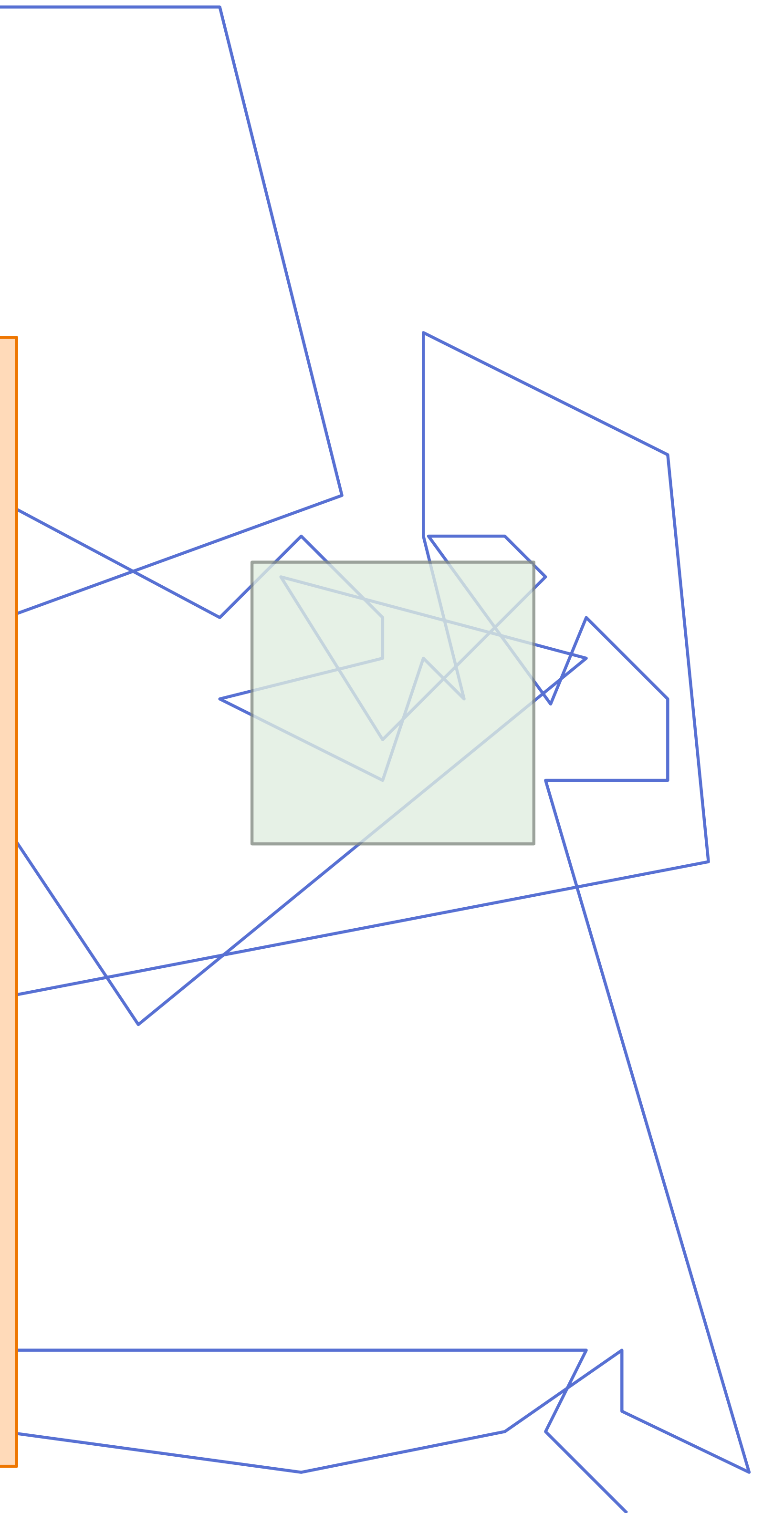
Given a trajectory $\mathcal{T}$ of a moving entity.

Find a placement of $\mathcal{H}$ maximizing

$$\frac{cont.length(\mathcal{T} \cap \mathcal{H})}{sidelength(\mathcal{H})}$$

Results

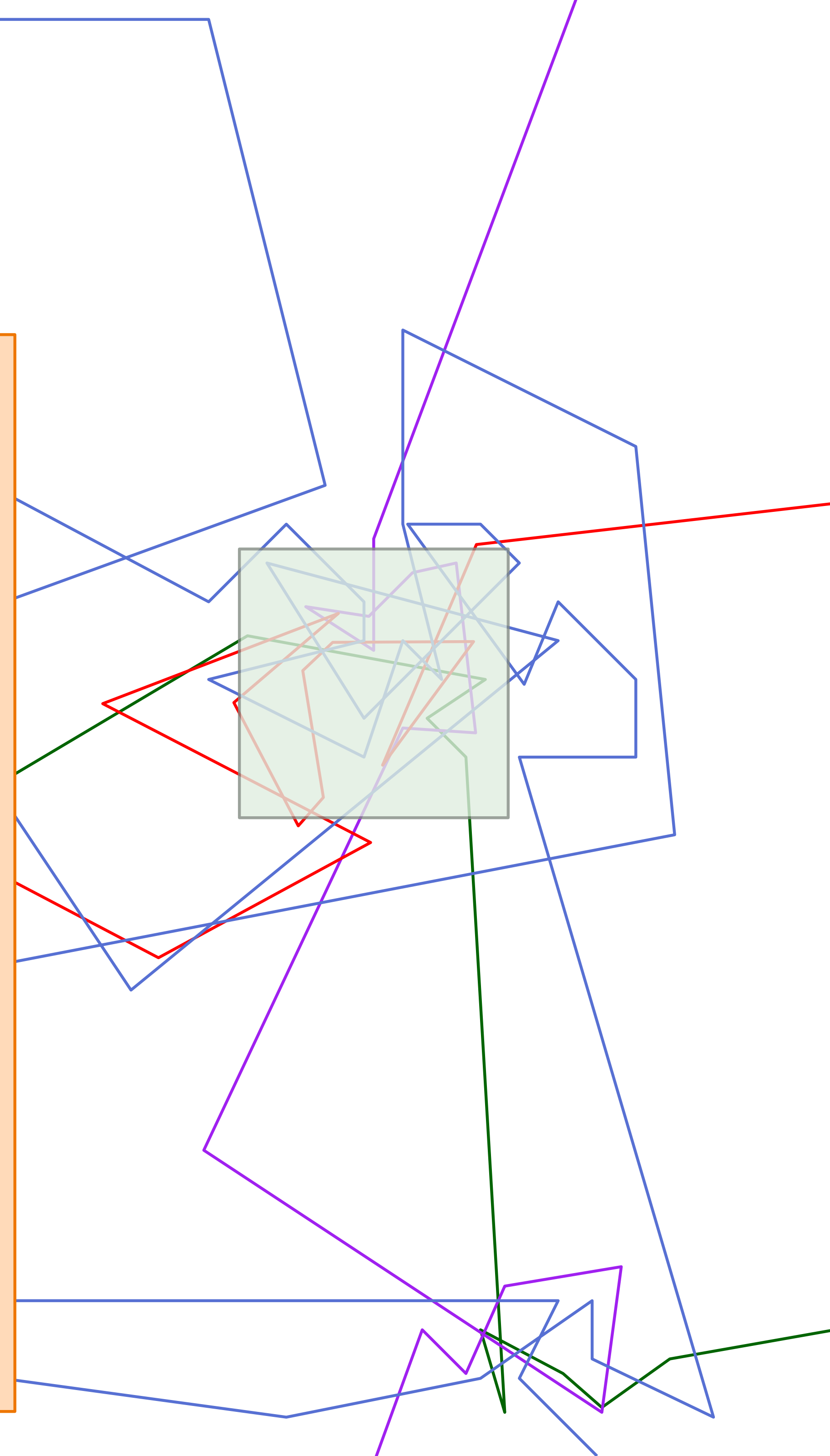
	Fixed Size	Fixed Length	Relative
$length$:	$O(n^2)$	$O(n^2 \log^2 n)$	$O(n^3)$
$cont.length$:	$O(n \log n)$	$O(n \log n)$	-



Results

	Fixed Size	Fixed Length	Relative
$length$:	$O(n^2)$	$O(n^2 \log^2 n)$	$O(n^3)$
$cont.length$:	$O(n \log n)$	$O(n \log n)$	-

Our algoritms also work for multiple trajectories,

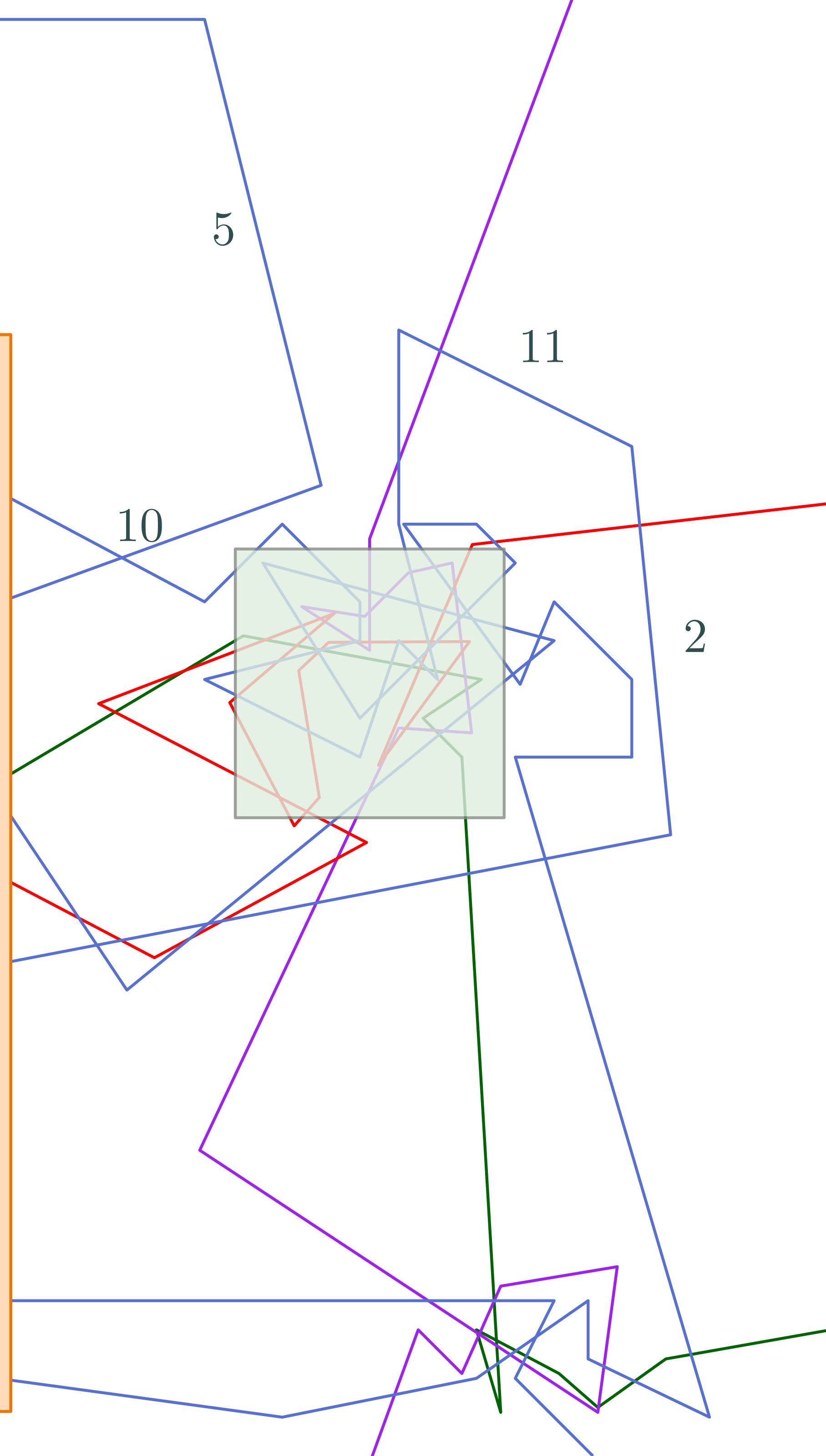


Results

	Fixed Size	Fixed Length	Relative
$length:$	$O(n^2)$	$O(n^2 \log^2 n)$	$O(n^3)$
$cont.length:$	$O(n \log n)$	$O(n \log n)$	-

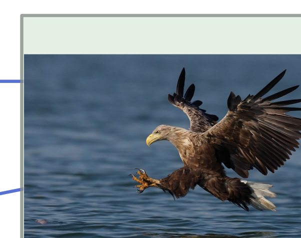
Our algoritms also work for multiple trajectories,

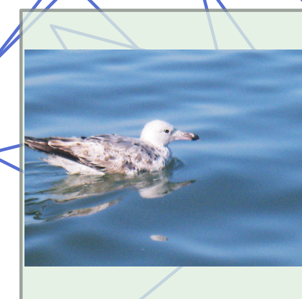
and for weighted edges.



Applications

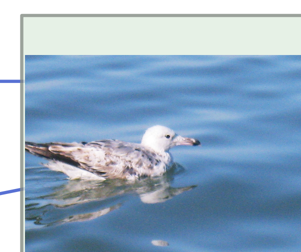
- Finding places





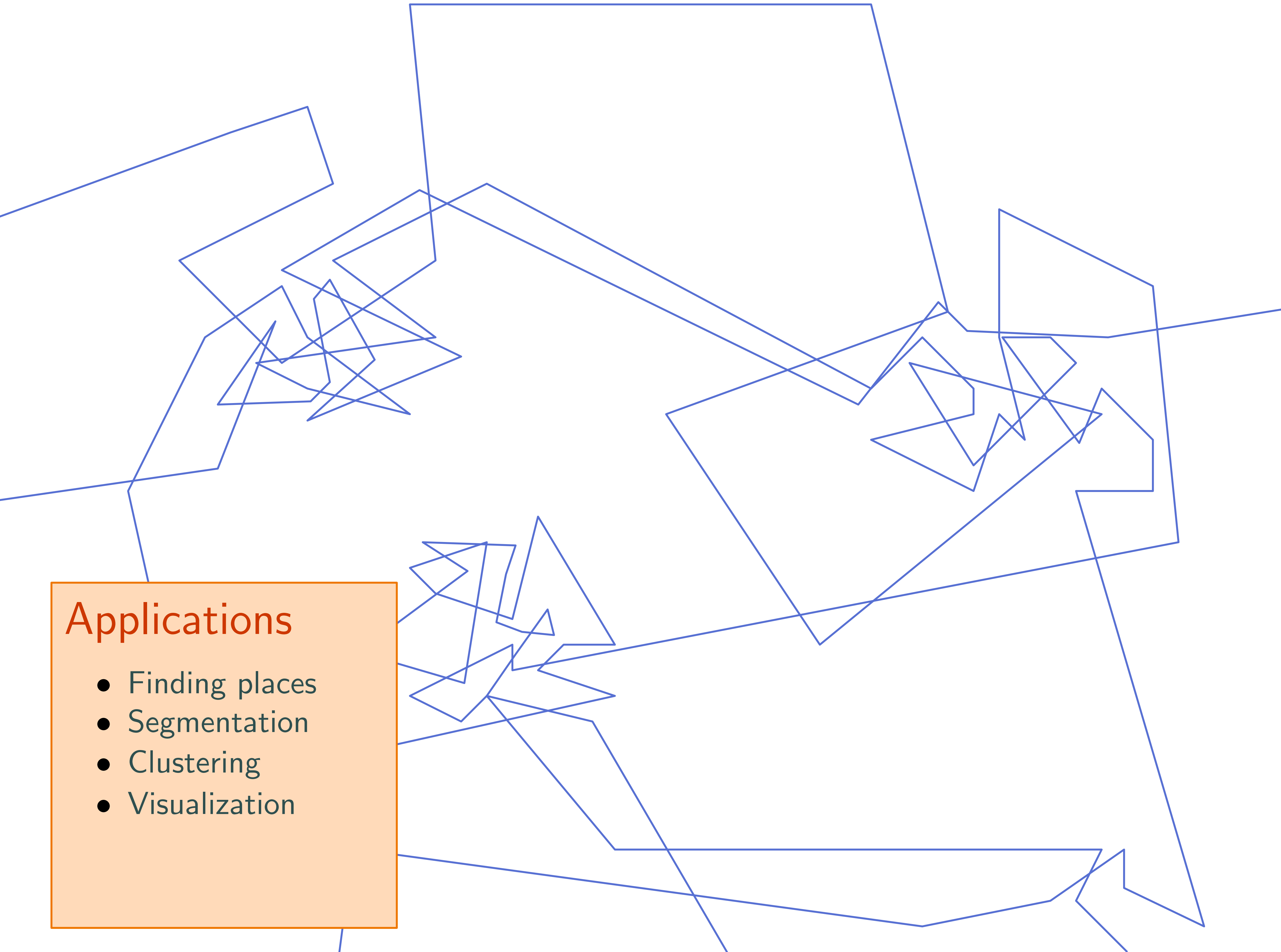
Applications

- Finding places
- Segmentation



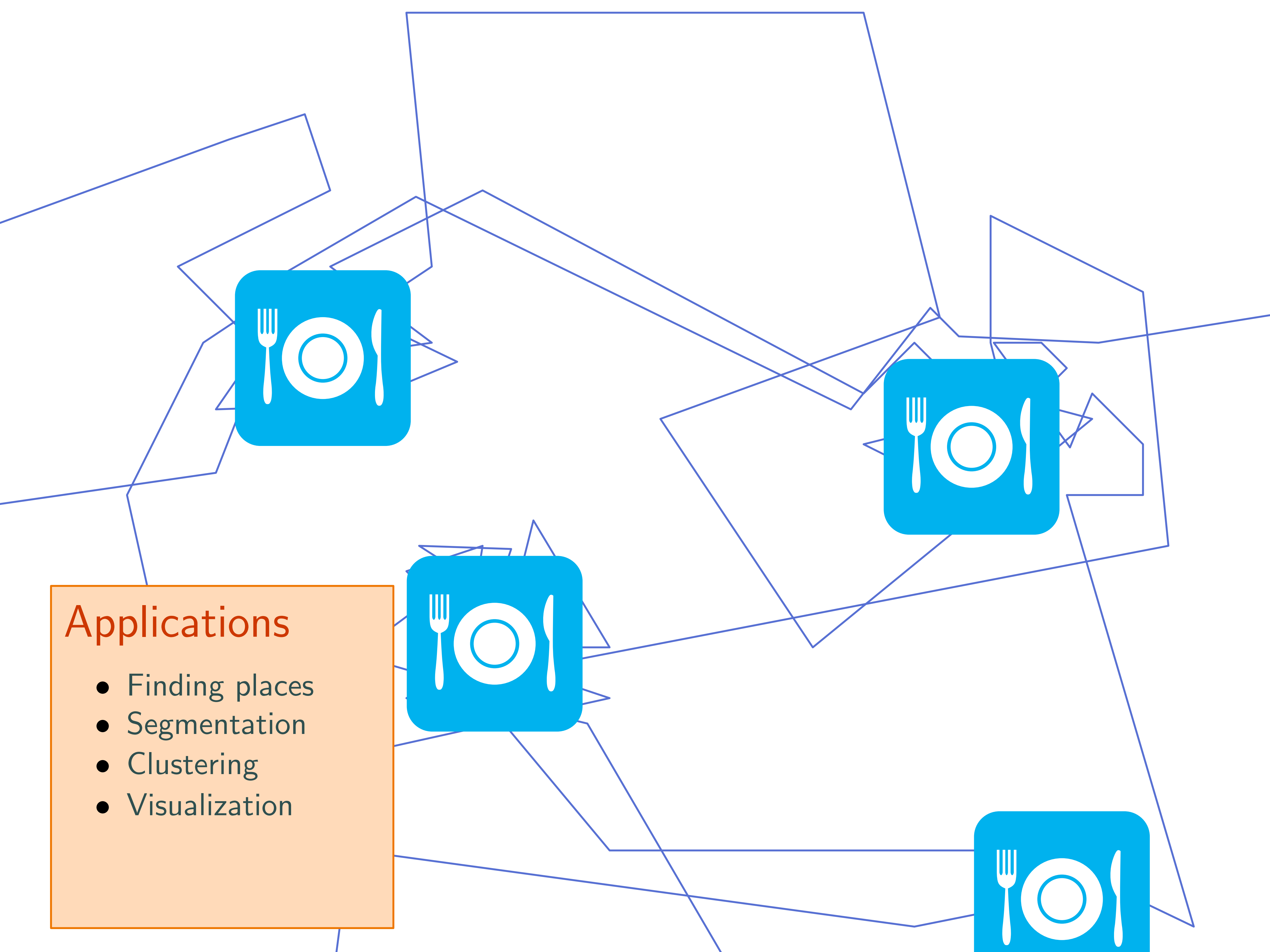
Applications

- Finding places
- Segmentation
- Clustering



Applications

- Finding places
- Segmentation
- Clustering
- Visualization

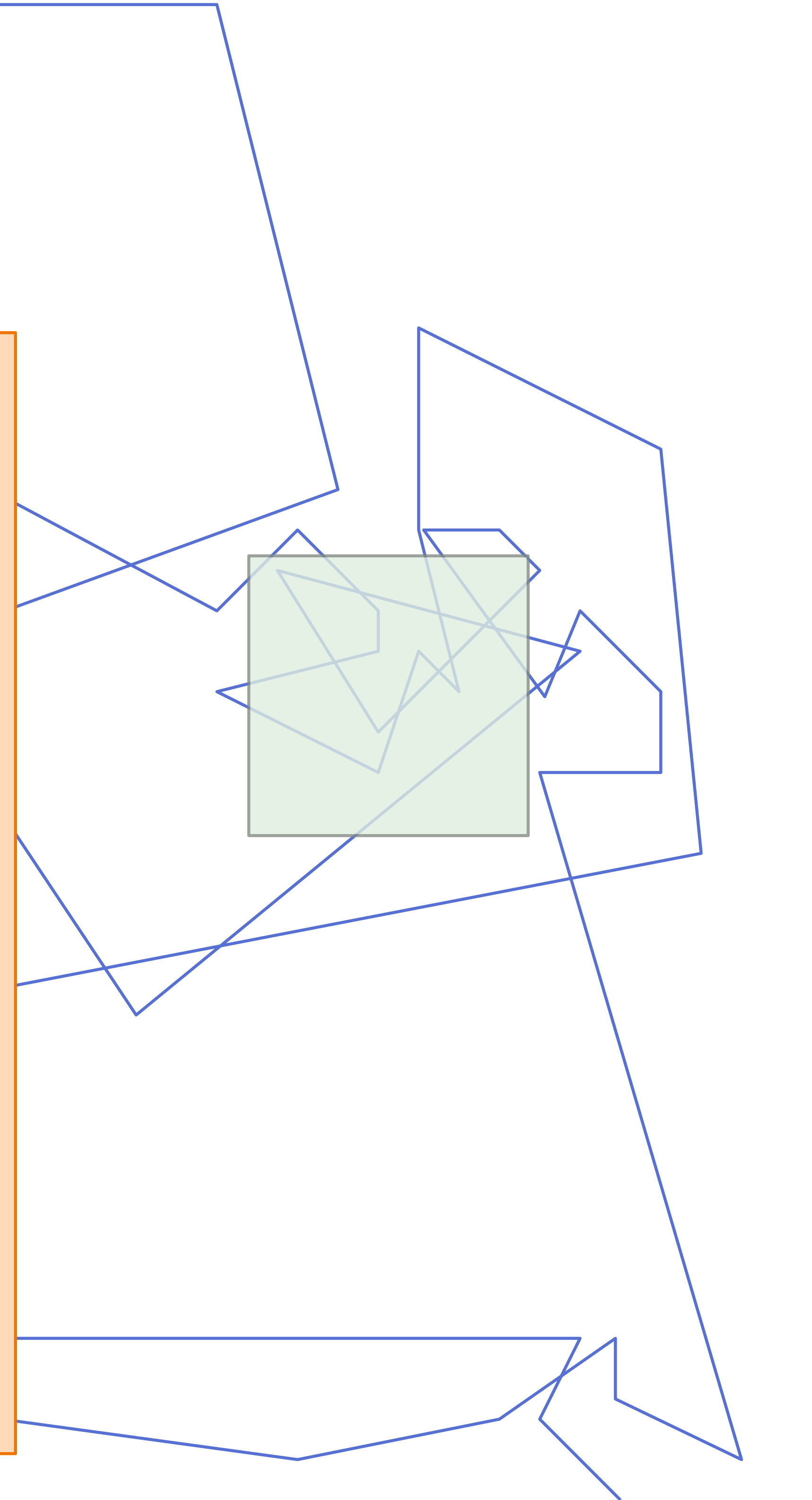


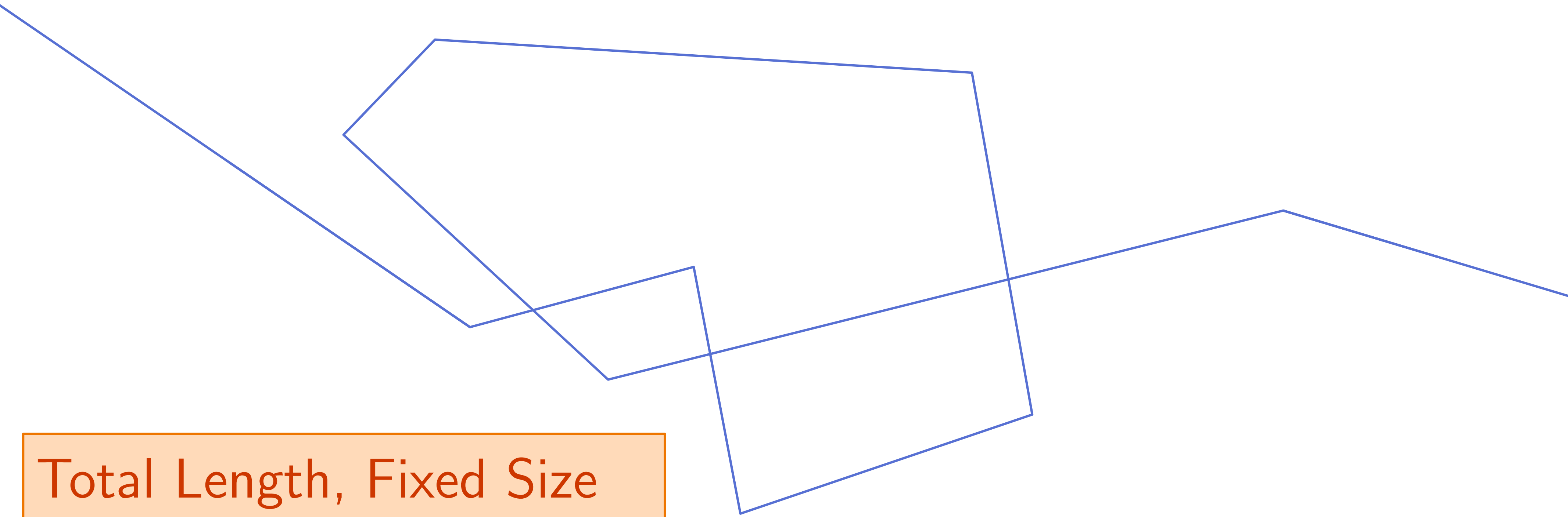
Applications

- Finding places
- Segmentation
- Clustering
- Visualization

Results

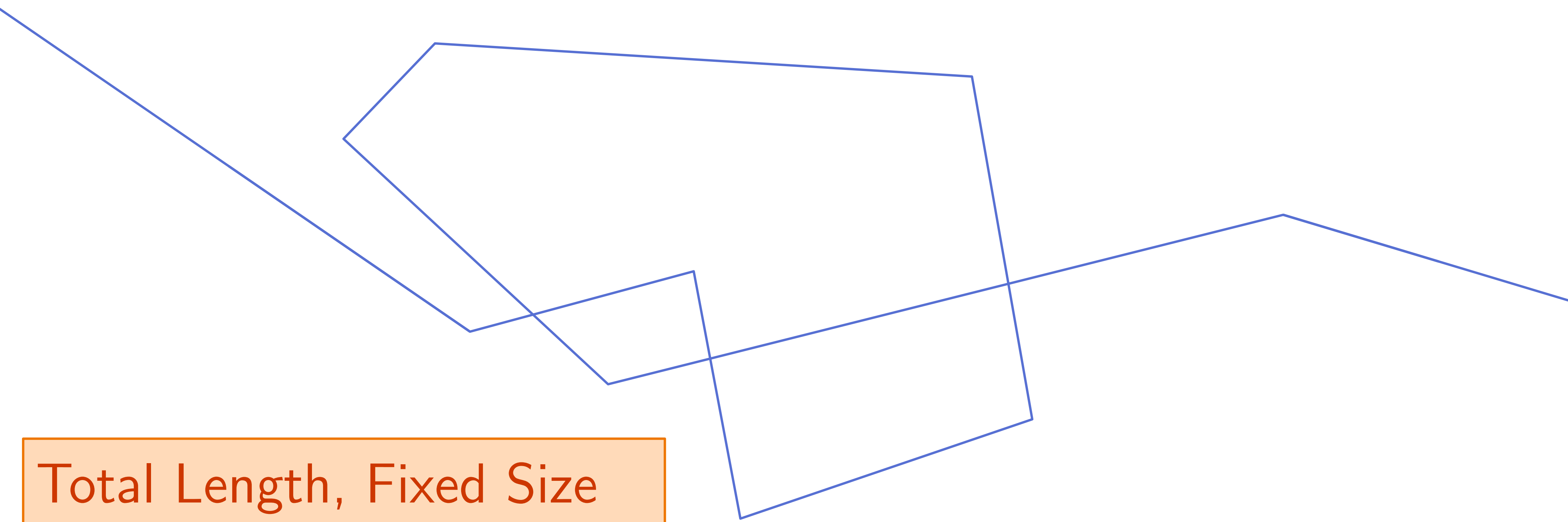
	Fixed Size	Fixed Length	Relative
$length$:	$O(n^2)$	$O(n^2 \log^2 n)$	$O(n^3)$
$maxlength$:	$O(n \log n)$	$O(n \log n)$	-





Total Length, Fixed Size

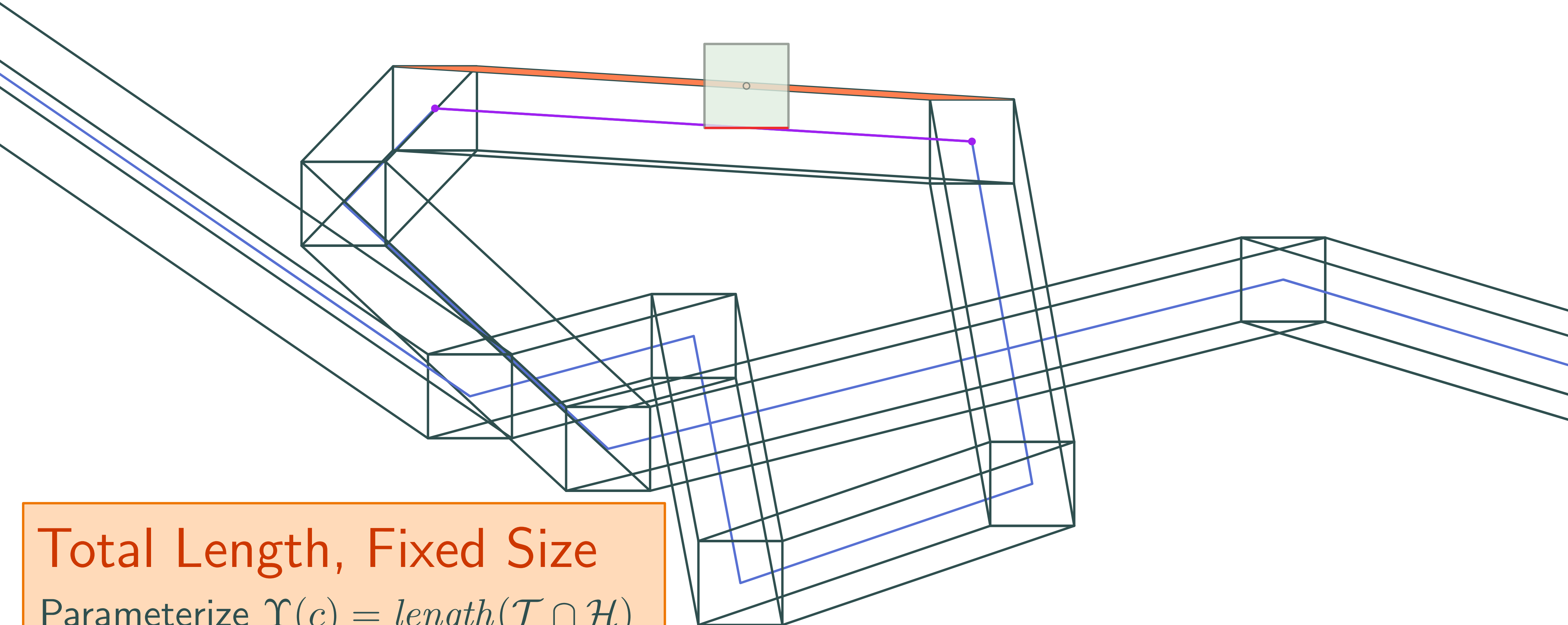
Parameterize $\Upsilon(c) = length(\mathcal{T} \cap \mathcal{H})$ by the center c of $\mathcal{H}$.



Total Length, Fixed Size

Parameterize $\Upsilon(c) = length(\mathcal{T} \cap \mathcal{H})$ by the center c of $\mathcal{H}$.

Lemma 1. Υ *is piecewise linear.*

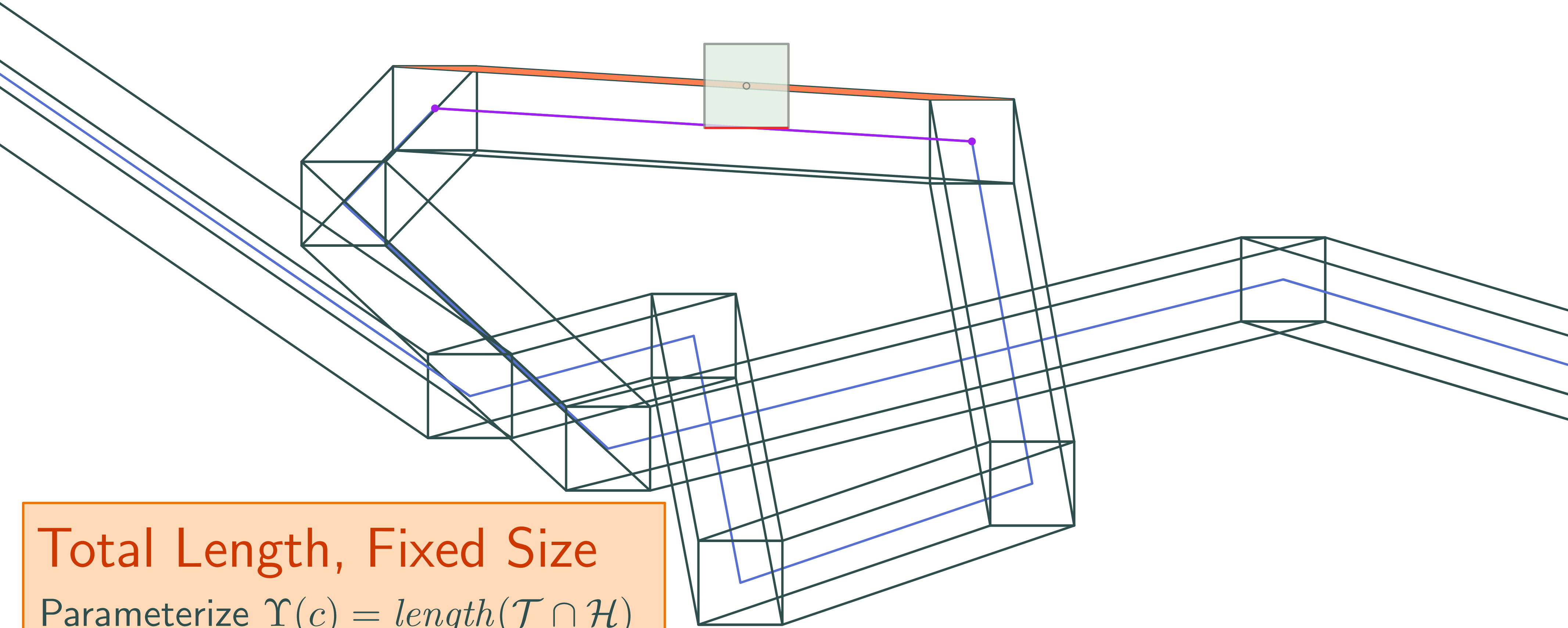


Total Length, Fixed Size

Parameterize $\Upsilon(c) = length(\mathcal{T} \cap \mathcal{H})$ by the center c of $\mathcal{H}$.

Lemma 1. Υ *is piecewise linear.*

Consider the subdivision $\mathcal{A}$ of the parameter space of Υ.



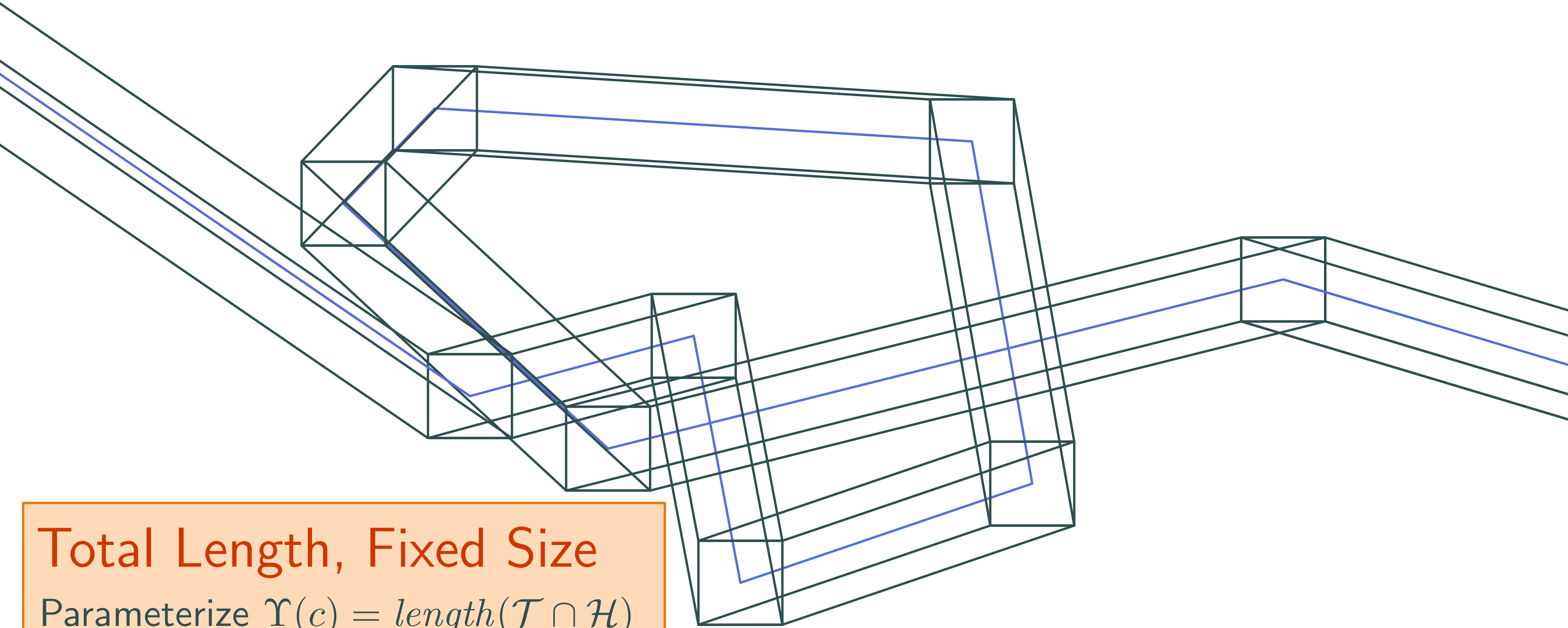
Total Length, Fixed Size

Parameterize $\Upsilon(c) = length(\mathcal{T} \cap \mathcal{H})$ by the center c of $\mathcal{H}$.

Lemma 1. Υ *is piecewise linear.*

Consider the subdivision $\mathcal{A}$ of the parameter space of Υ.

$\max \Upsilon$ occurs at a vertex of $\mathcal{A}$. So, compute Υ at each vertex of $\mathcal{A}$.



Total Length, Fixed Size

Parameterize $\Upsilon(c) = length(\mathcal{T} \cap \mathcal{H})$ by the center c of $\mathcal{H}$.

Lemma 1. Υ *is piecewise linear.*

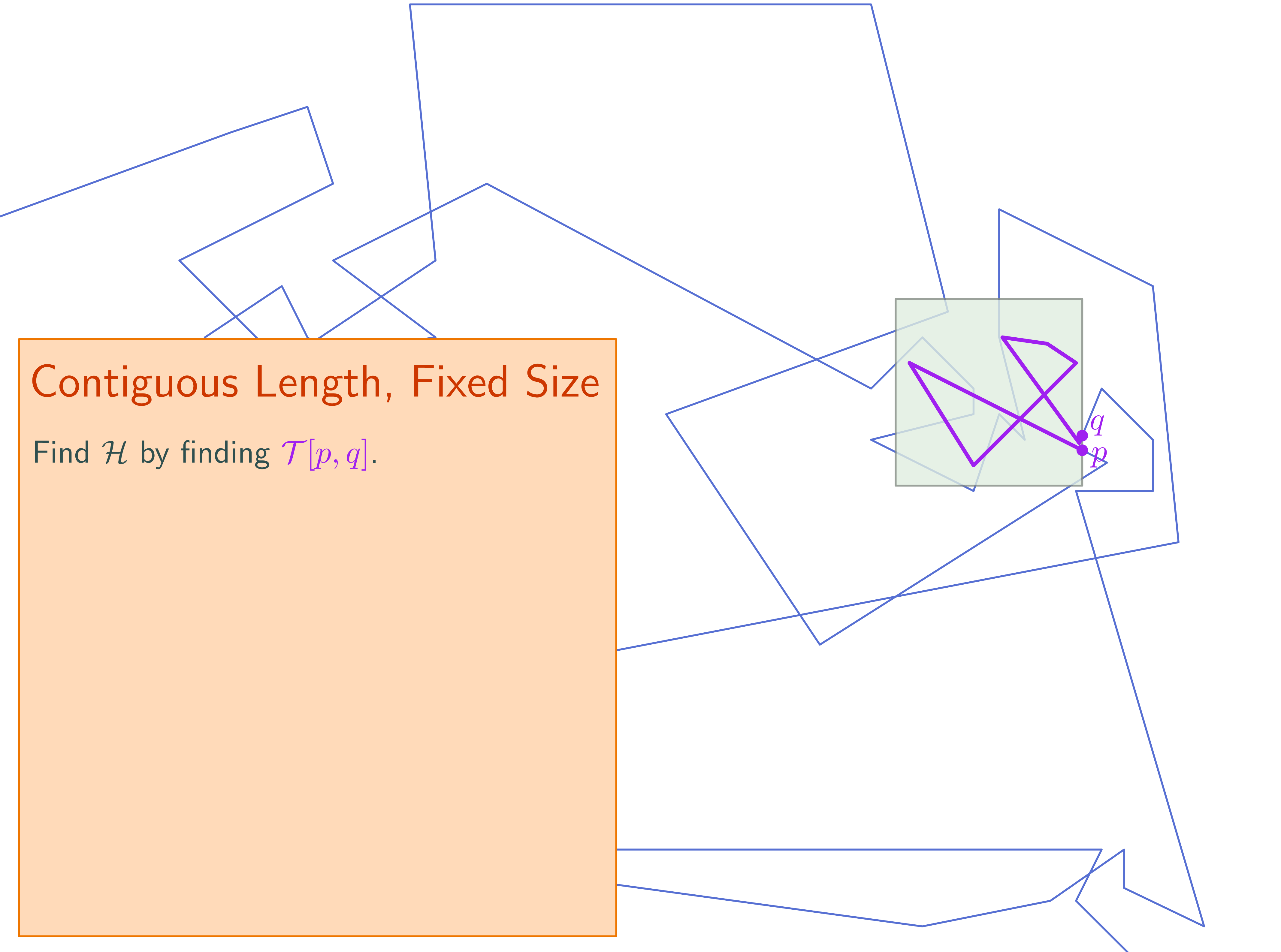
Consider the subdivision $\mathcal{A}$ of the parameter space of Υ.

$\max \Upsilon$ occurs at a vertex of $\mathcal{A}$. So, compute Υ at each vertex of $\mathcal{A}$.

Complexity $\mathcal{A}$:	$O(n^2)$
Find $\max \Upsilon$:	$O(n^2)$
Total:	$O(n^2)$

Contiguous Length, Fixed Size

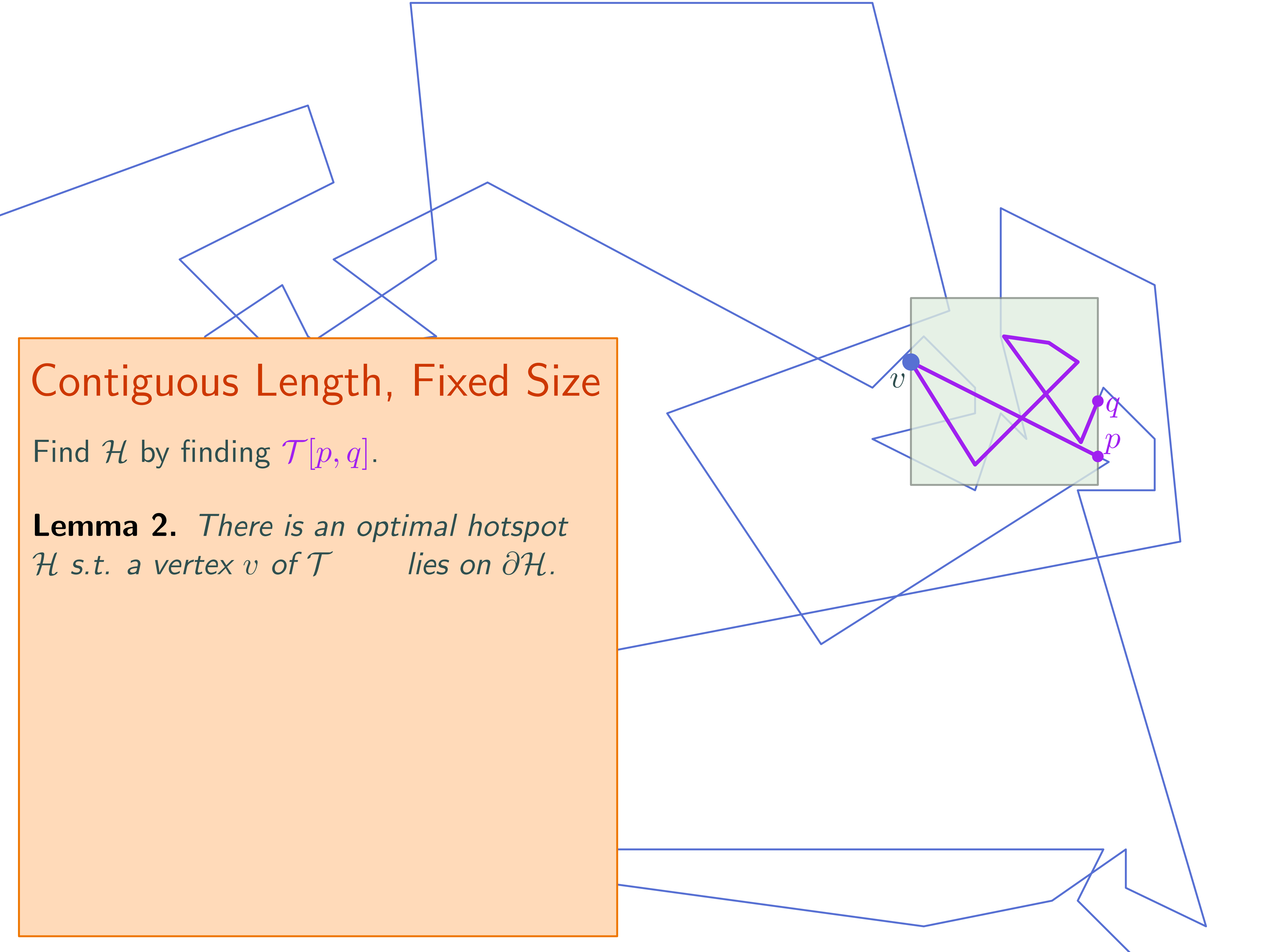
Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.



Contiguous Length, Fixed Size

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

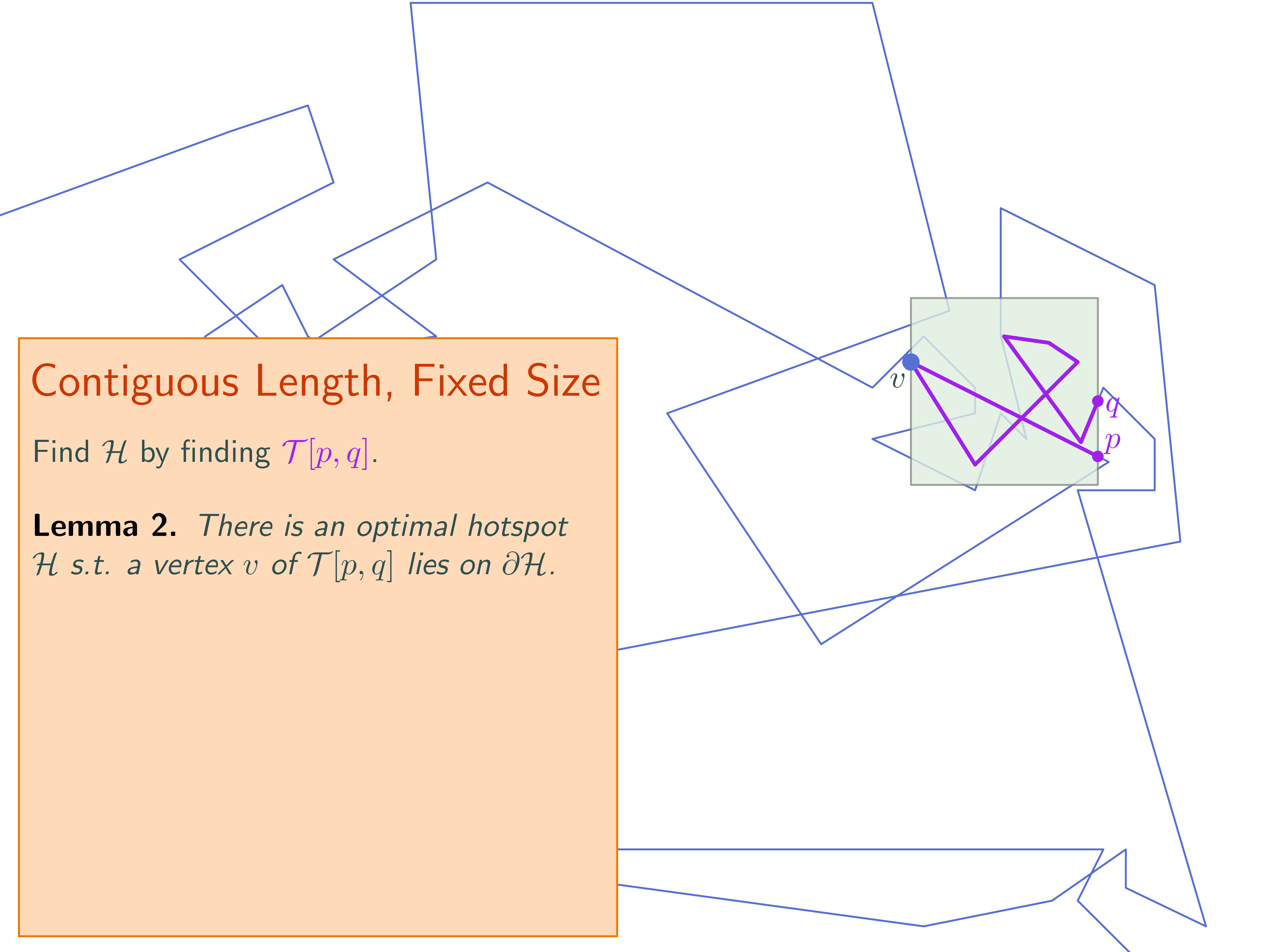
Lemma 2. *There is an optimal hotspot $\mathcal{H}$ s.t. a vertex v of $\mathcal{T}$ lies on $\partial\mathcal{H}$.*



Contiguous Length, Fixed Size

Find $\mathcal{H}$ by finding $\mathcal{T}[p,q]$.

Lemma 2. *There is an optimal hotspot $\mathcal{H}$ s.t. a vertex v of $\mathcal{T}[p,q]$ lies on $\partial\mathcal{H}$.*

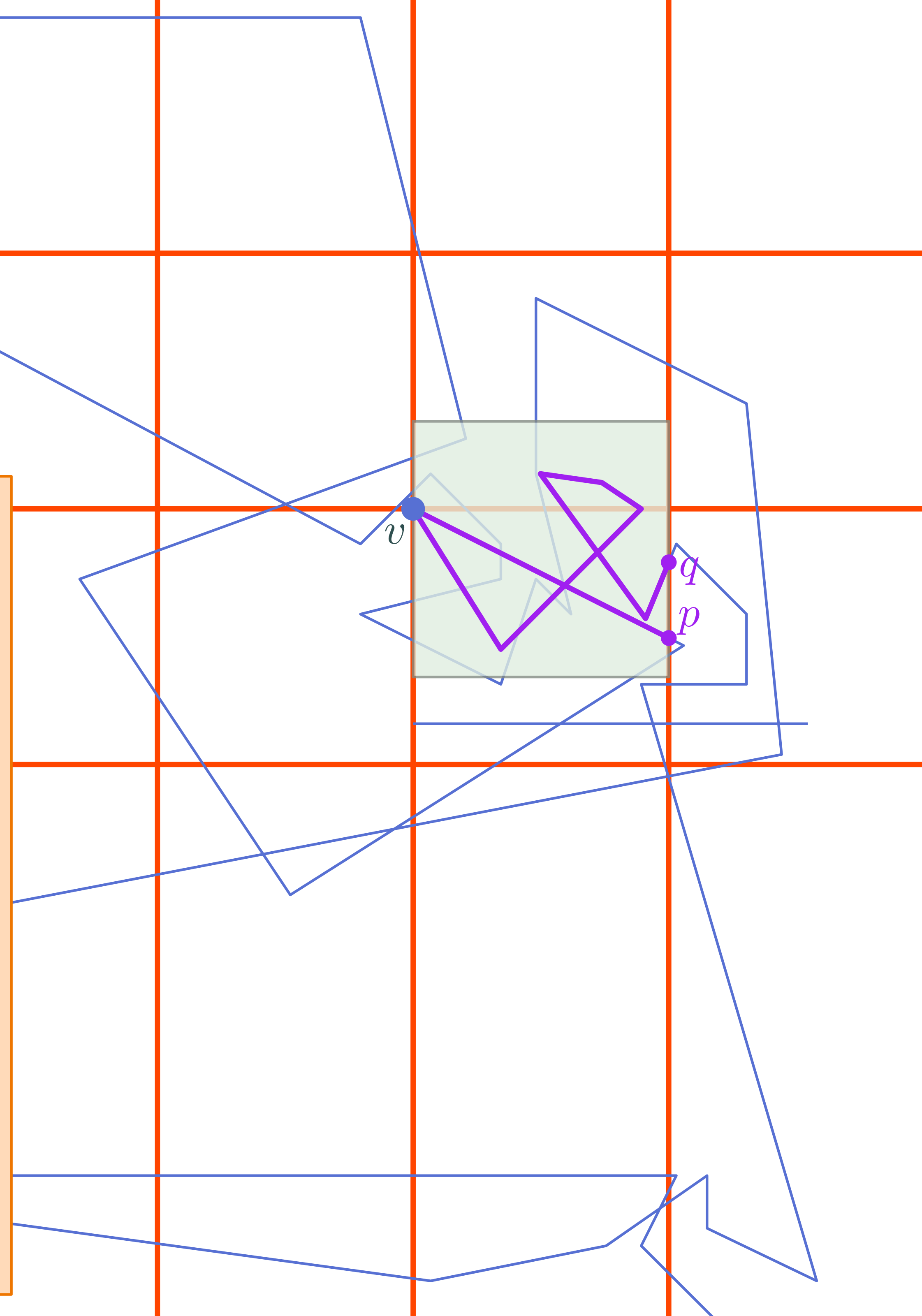


Contiguous Length, Fixed Size

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

Lemma 2. *There is an optimal hotspot $\mathcal{H}$ s.t. a vertex v of $\mathcal{T}[p, q]$ lies on $\partial\mathcal{H}$.*

Corollary 3. *Starting point p on one of the horizontal or vertical lines.*



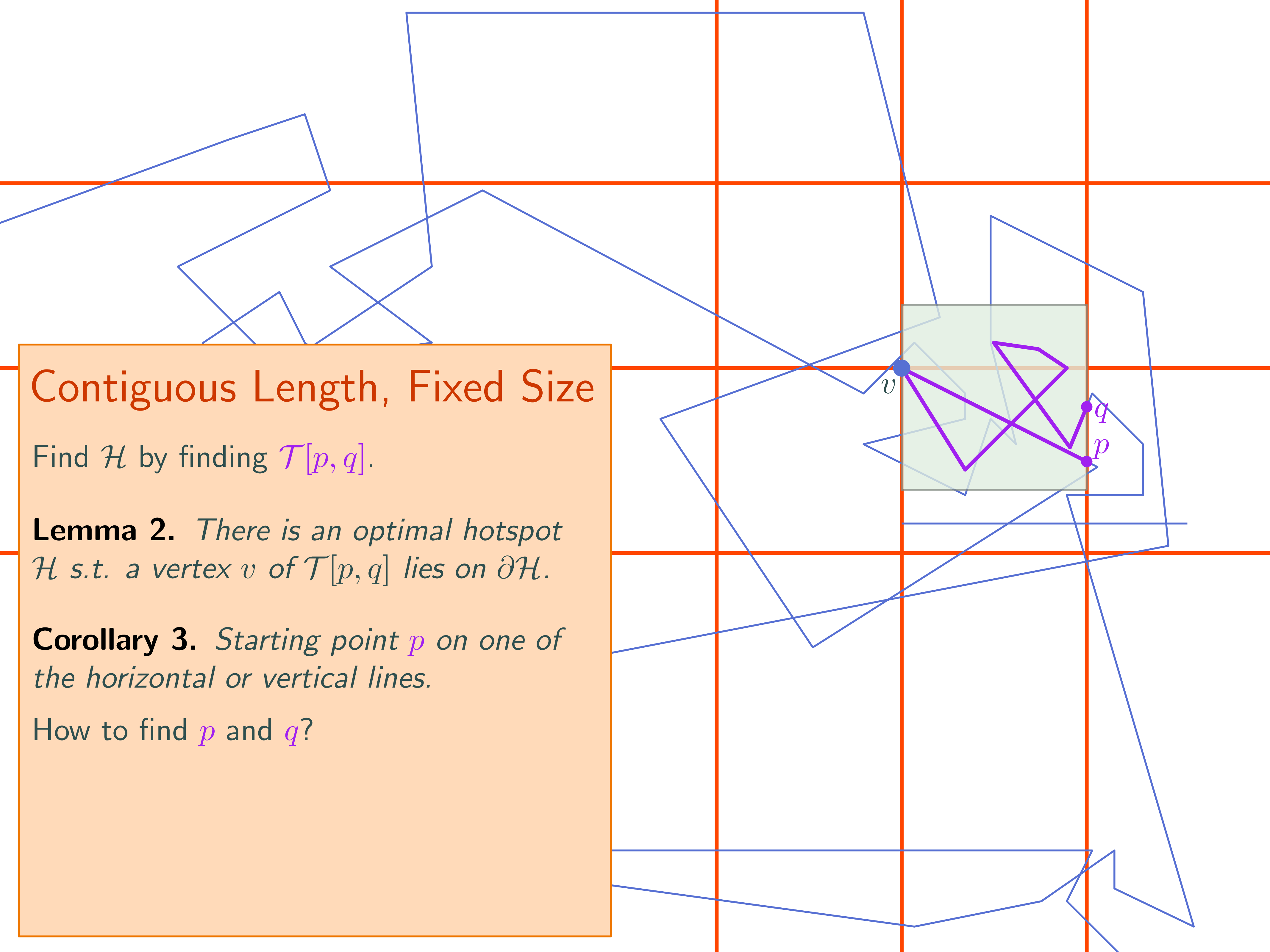
Contiguous Length, Fixed Size

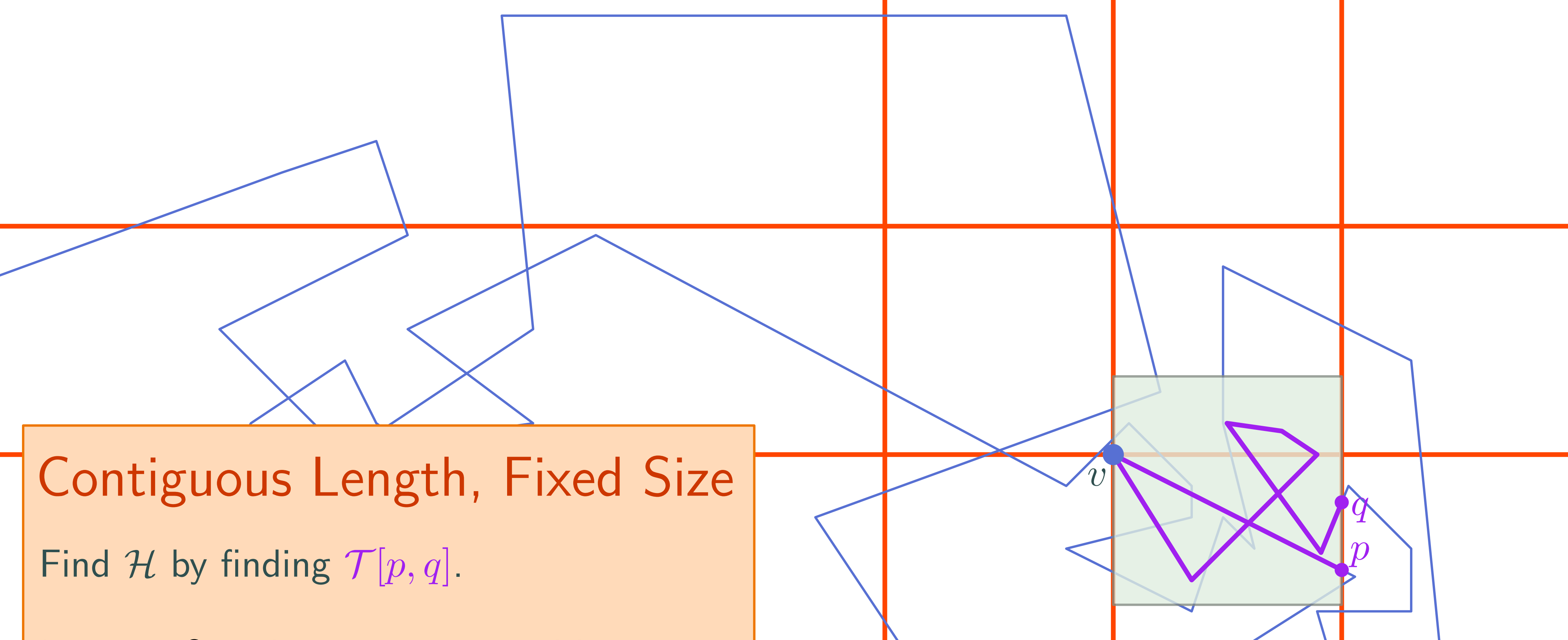
Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

Lemma 2. *There is an optimal hotspot $\mathcal{H}$ s.t. a vertex v of $\mathcal{T}[p, q]$ lies on $\partial\mathcal{H}$.*

Corollary 3. *Starting point p on one of the horizontal or vertical lines.*

How to find p and q?





Contiguous Length, Fixed Size

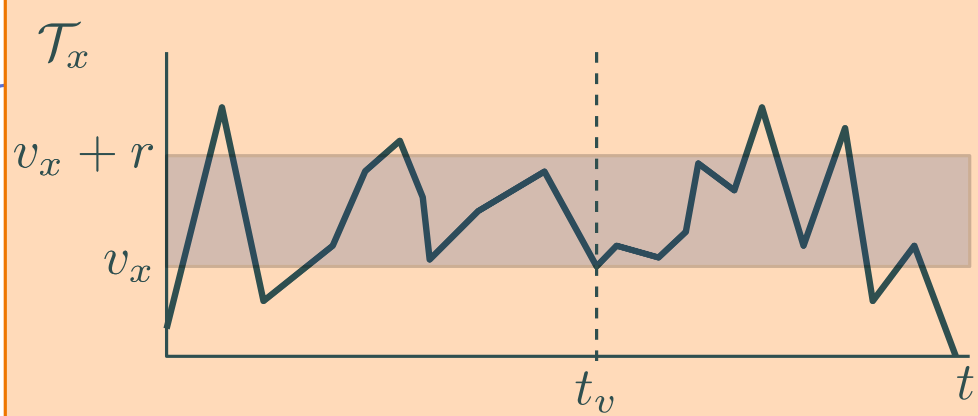
Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

Lemma 2. *There is an optimal hotspot $\mathcal{H}$ s.t. a vertex v of $\mathcal{T}[p, q]$ lies on $\partial\mathcal{H}$.*

Corollary 3. *Starting point p on one of the horizontal or vertical lines.*

How to find p and q?

Consider $\mathcal{T}_x$ and $\mathcal{T}_y$ separately.



Contiguous Length, Fixed Size

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

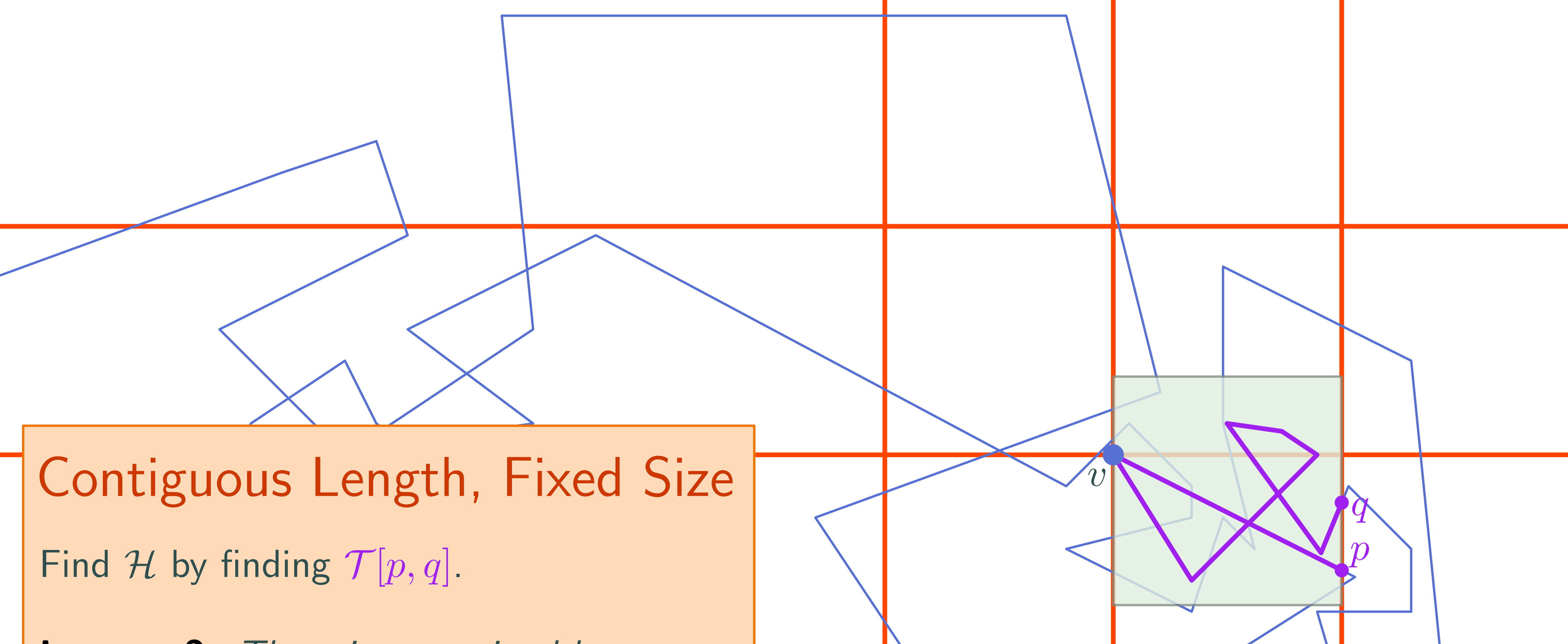
Lemma 2. *There is an optimal hotspot $\mathcal{H}$ s.t. a vertex v of $\mathcal{T}[p, q]$ lies on $\partial\mathcal{H}$.*

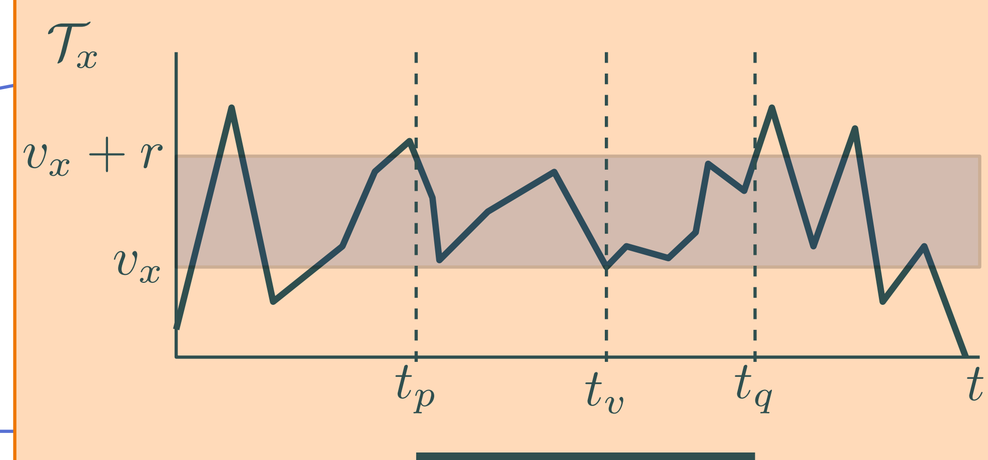
Corollary 3. *Starting point p on one of the horizontal or vertical lines.*

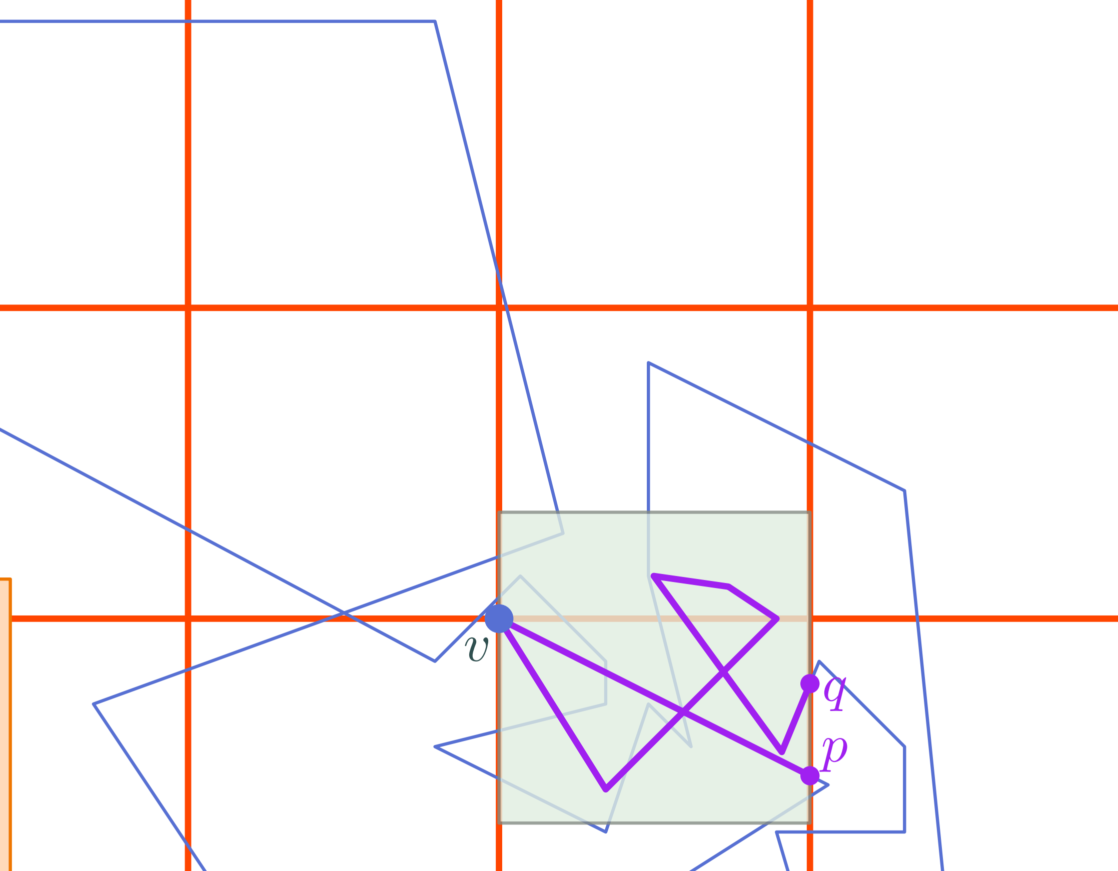
How to find p and q?

Consider $\mathcal{T}_x$ and $\mathcal{T}_y$ separately.

Try to find $[t_p, t_q]$.







Contiguous Length, Fixed Size

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

Lemma 2. *There is an optimal hotspot $\mathcal{H}$ s.t. a vertex v of $\mathcal{T}[p, q]$ lies on $\partial\mathcal{H}$.*

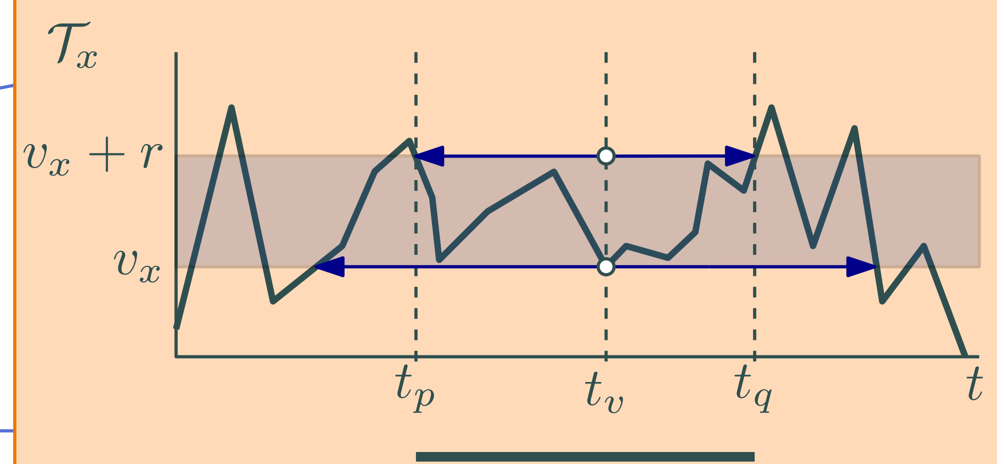
Corollary 3. *Starting point p on one of the horizontal or vertical lines.*

How to find p and q?

Consider $\mathcal{T}_x$ and $\mathcal{T}_y$ separately.

Try to find $[t_p, t_q]$.

Use ray shooting queries.



Contiguous Length, Fixed Size

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

Lemma 2. *There is an optimal hotspot $\mathcal{H}$ s.t. a vertex v of $\mathcal{T}[p, q]$ lies on $\partial\mathcal{H}$.*

Corollary 3. *Starting point p on one of the horizontal or vertical lines.*

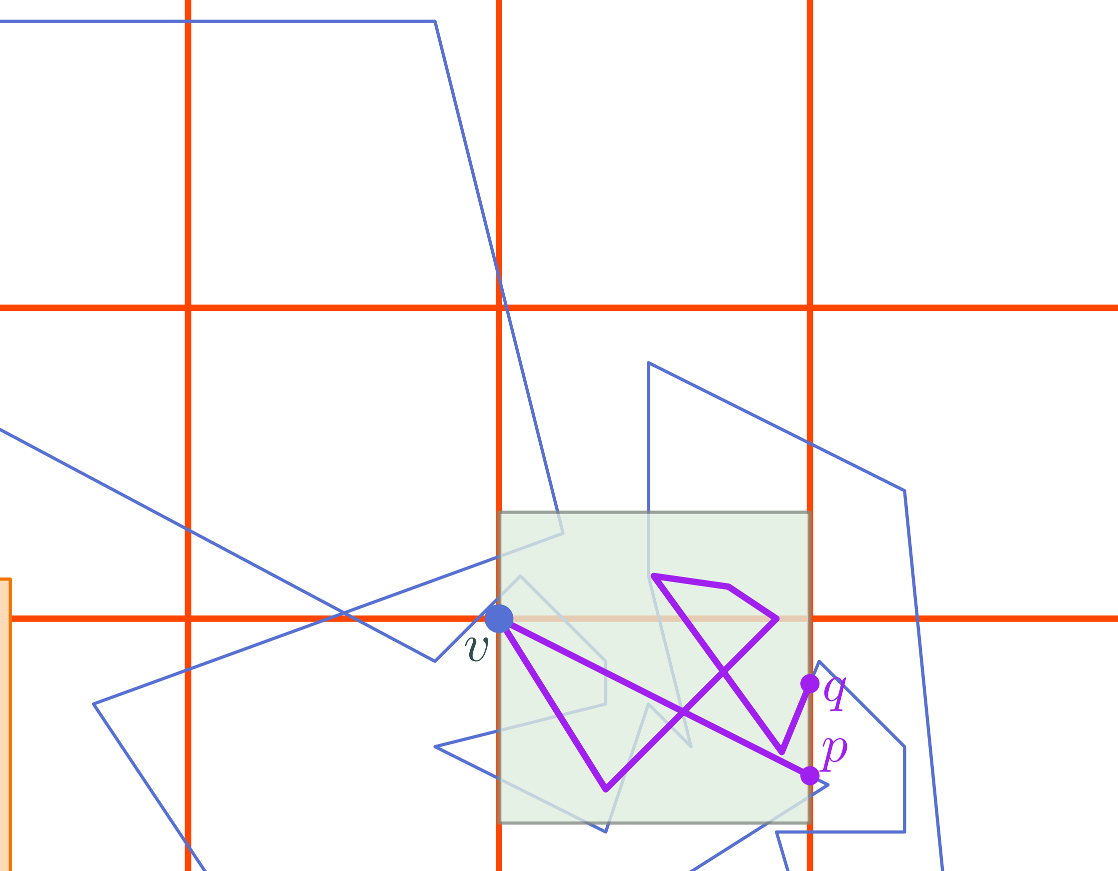
How to find p and q?

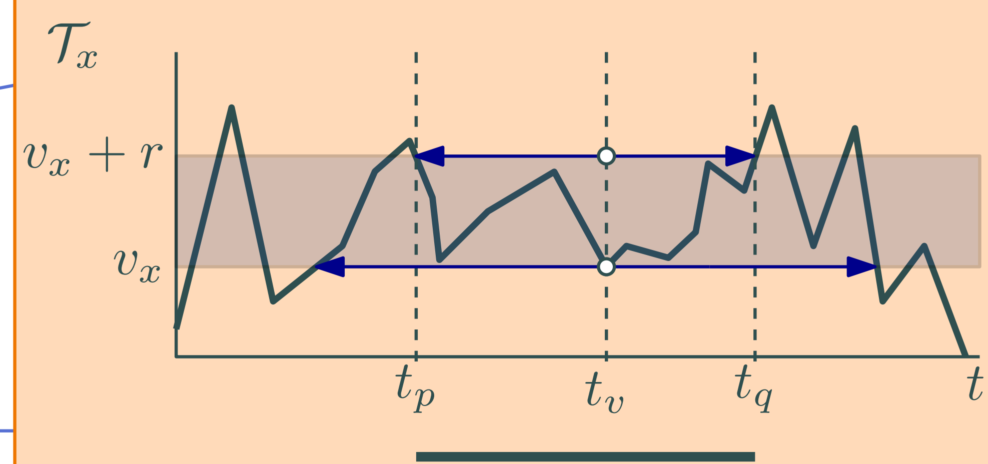
Consider $\mathcal{T}_x$ and $\mathcal{T}_y$ separately.

Try to find $[t_p, t_q]$.

Use ray shooting queries.

Running time: $O(n \log n)$



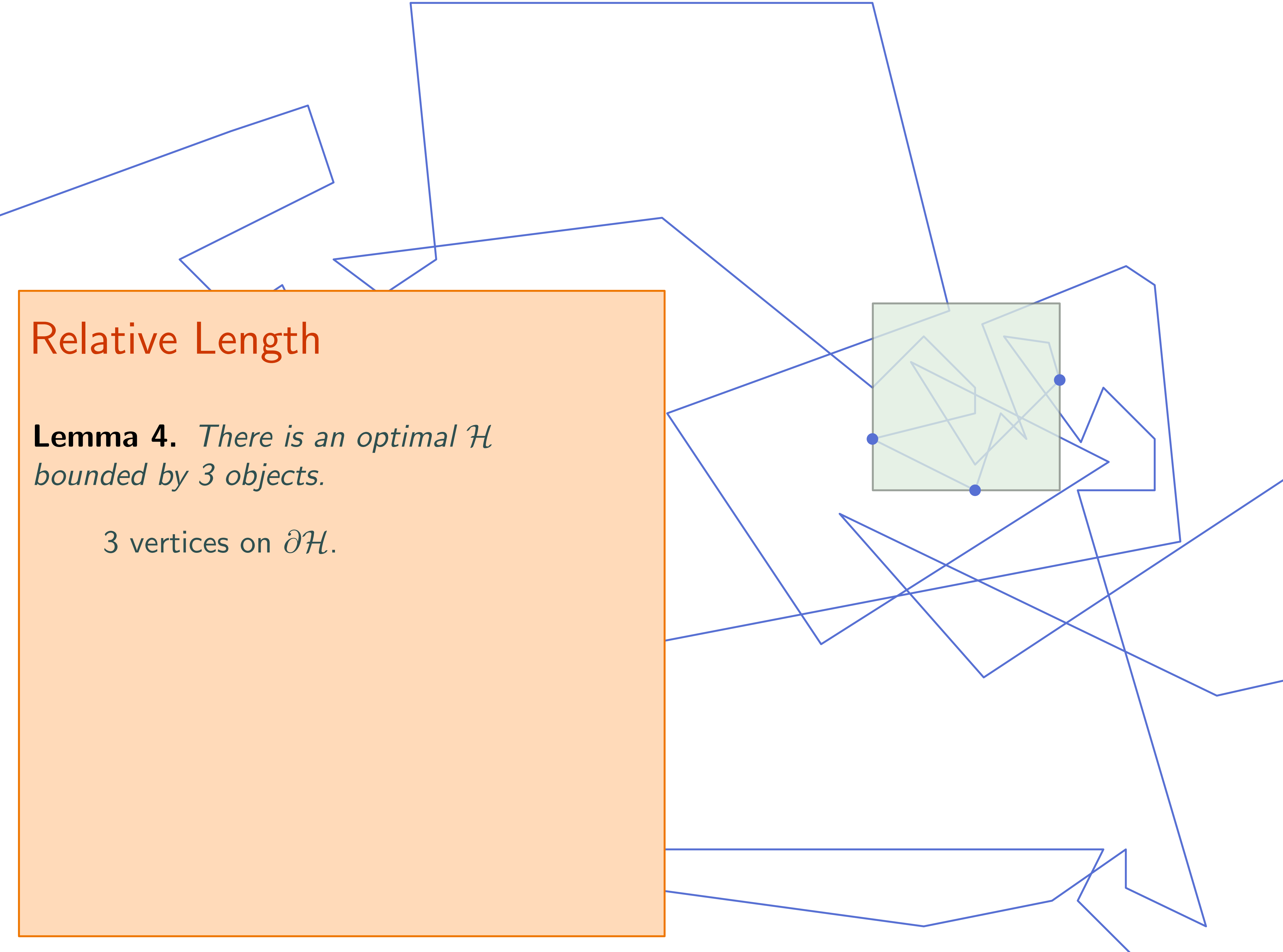


Relative Length

Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

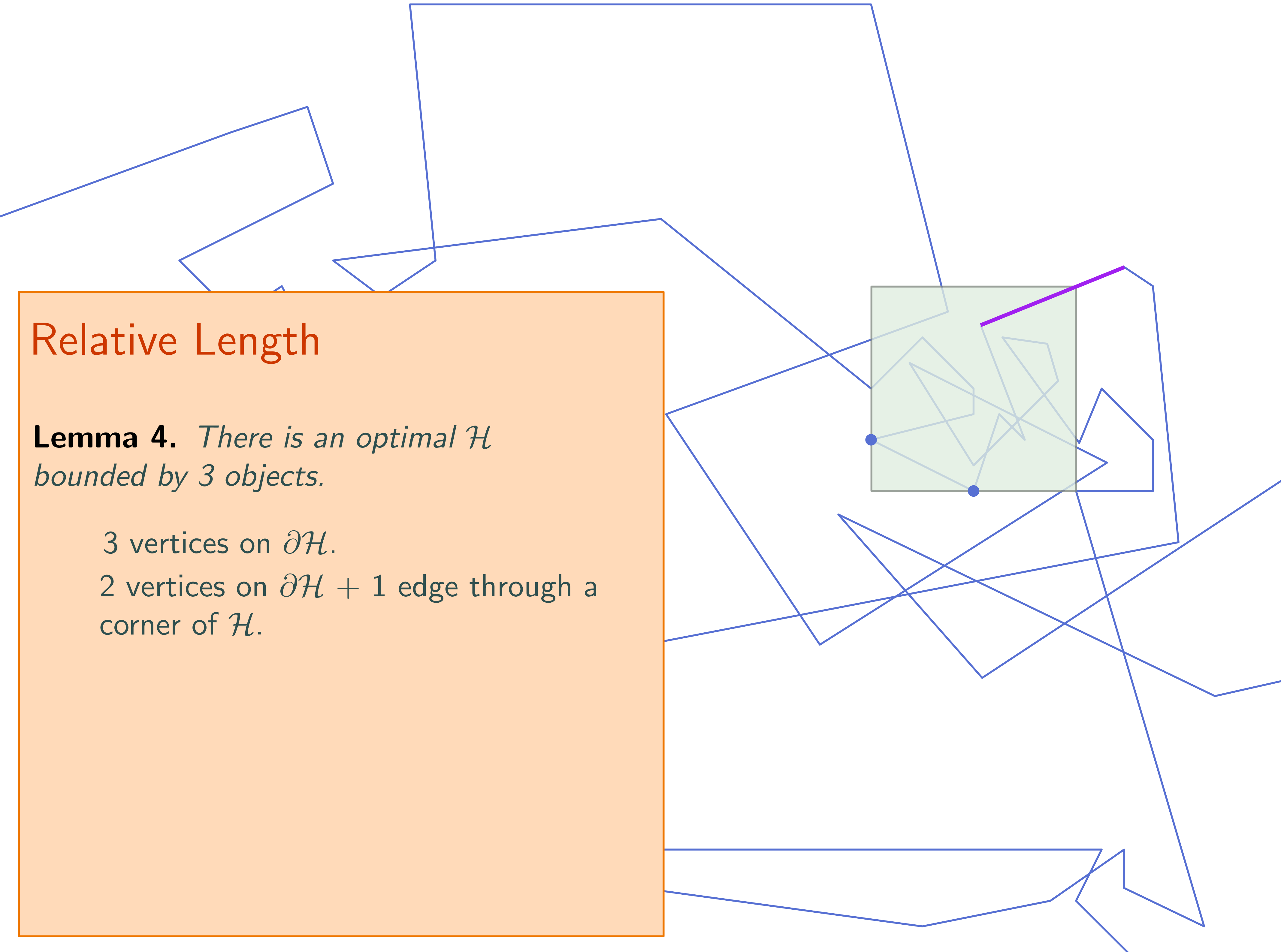
3 vertices on $\partial\mathcal{H}$.



Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

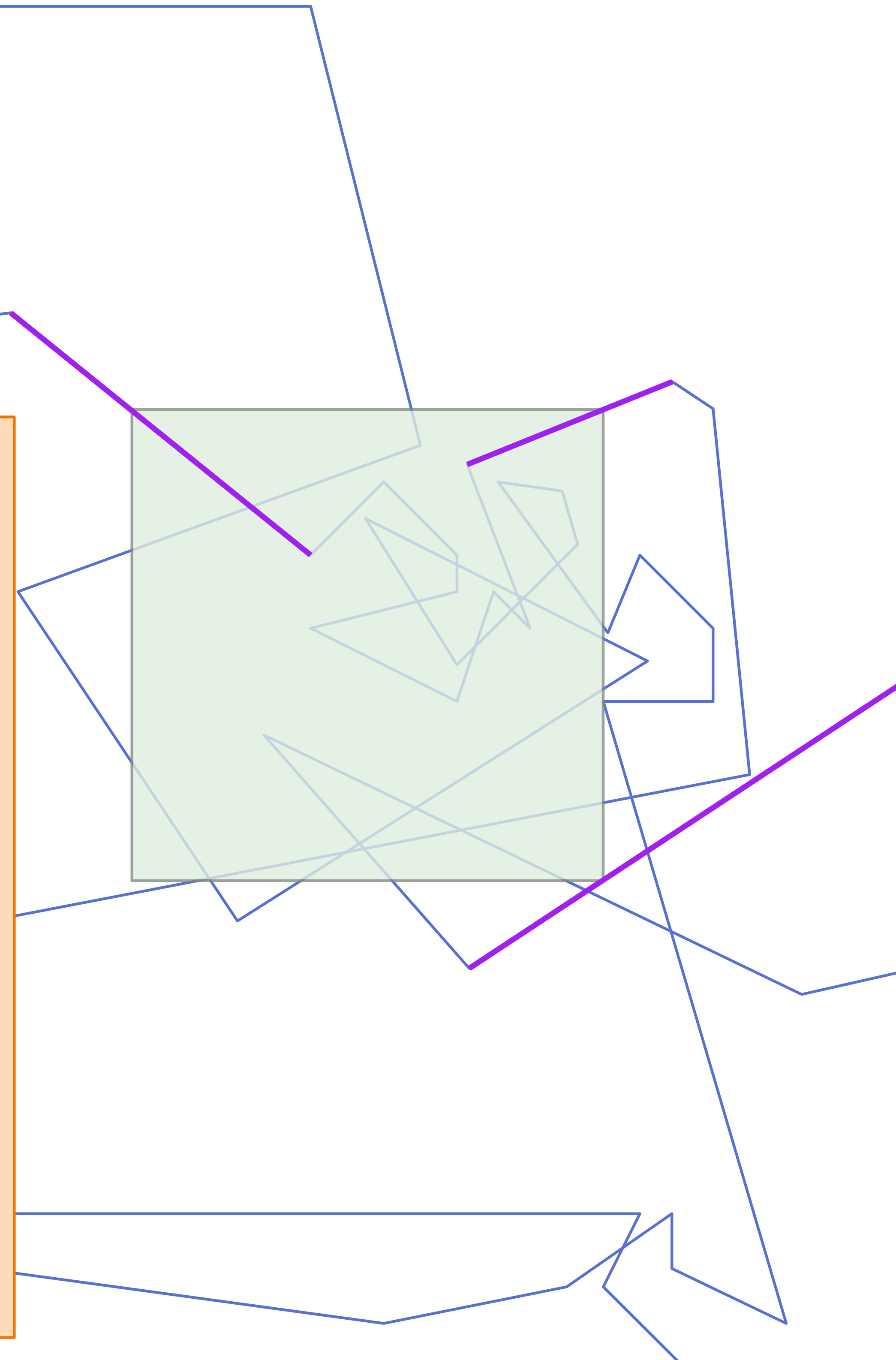
- 3 vertices on $\partial\mathcal{H}$.
- 2 vertices on $\partial\mathcal{H}$ + 1 edge through a corner of $\mathcal{H}$.



Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

- 3 vertices on $\partial\mathcal{H}$.
- 2 vertices on $\partial\mathcal{H} + 1$ edge through a corner of $\mathcal{H}$.
- ...
- 3 edges through corners of $\mathcal{H}$.



Relative Length

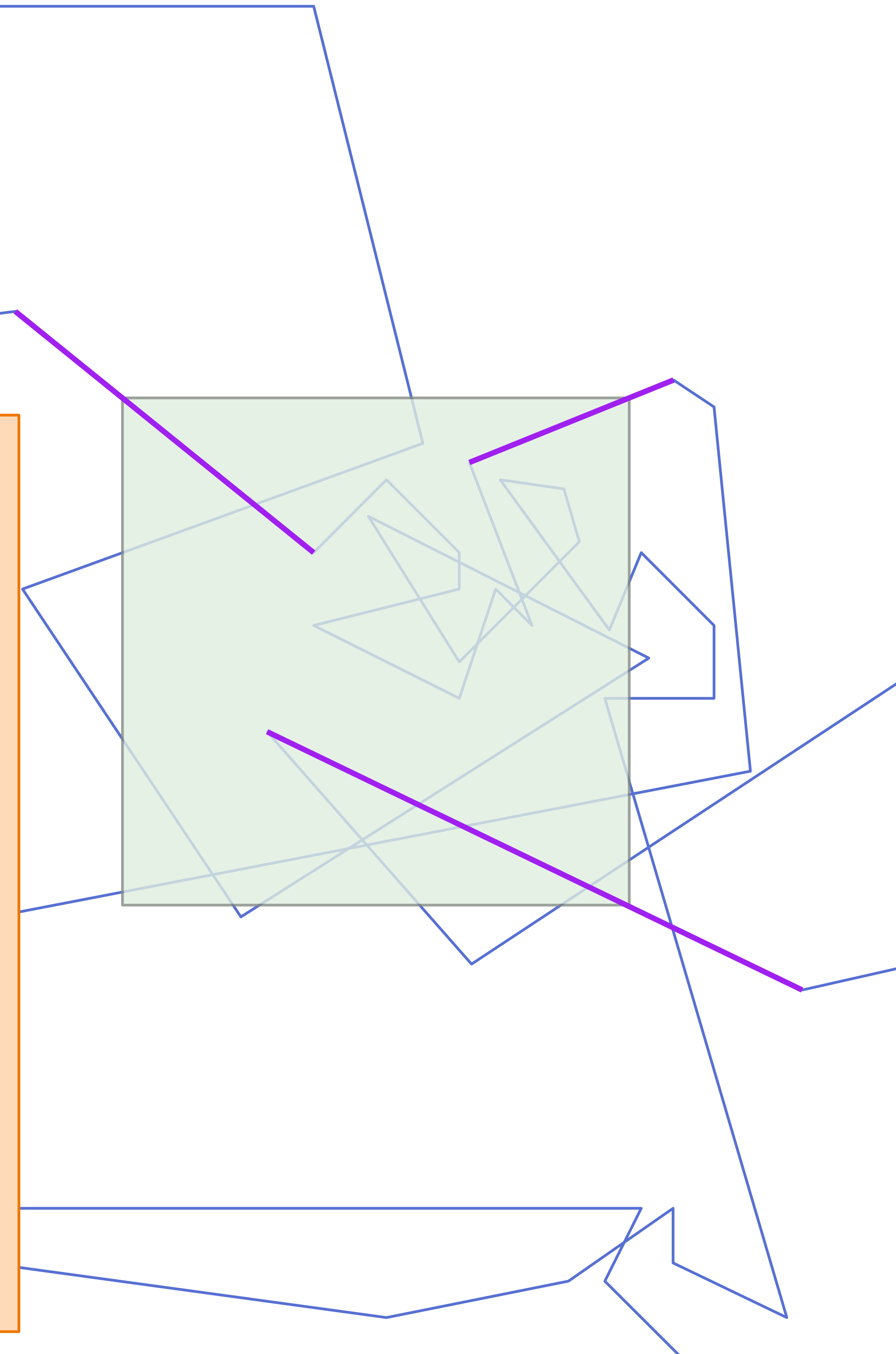
Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

3 vertices on $\partial\mathcal{H}$.

2 vertices on $\partial\mathcal{H}$ + 1 edge through a corner of $\mathcal{H}$.

...

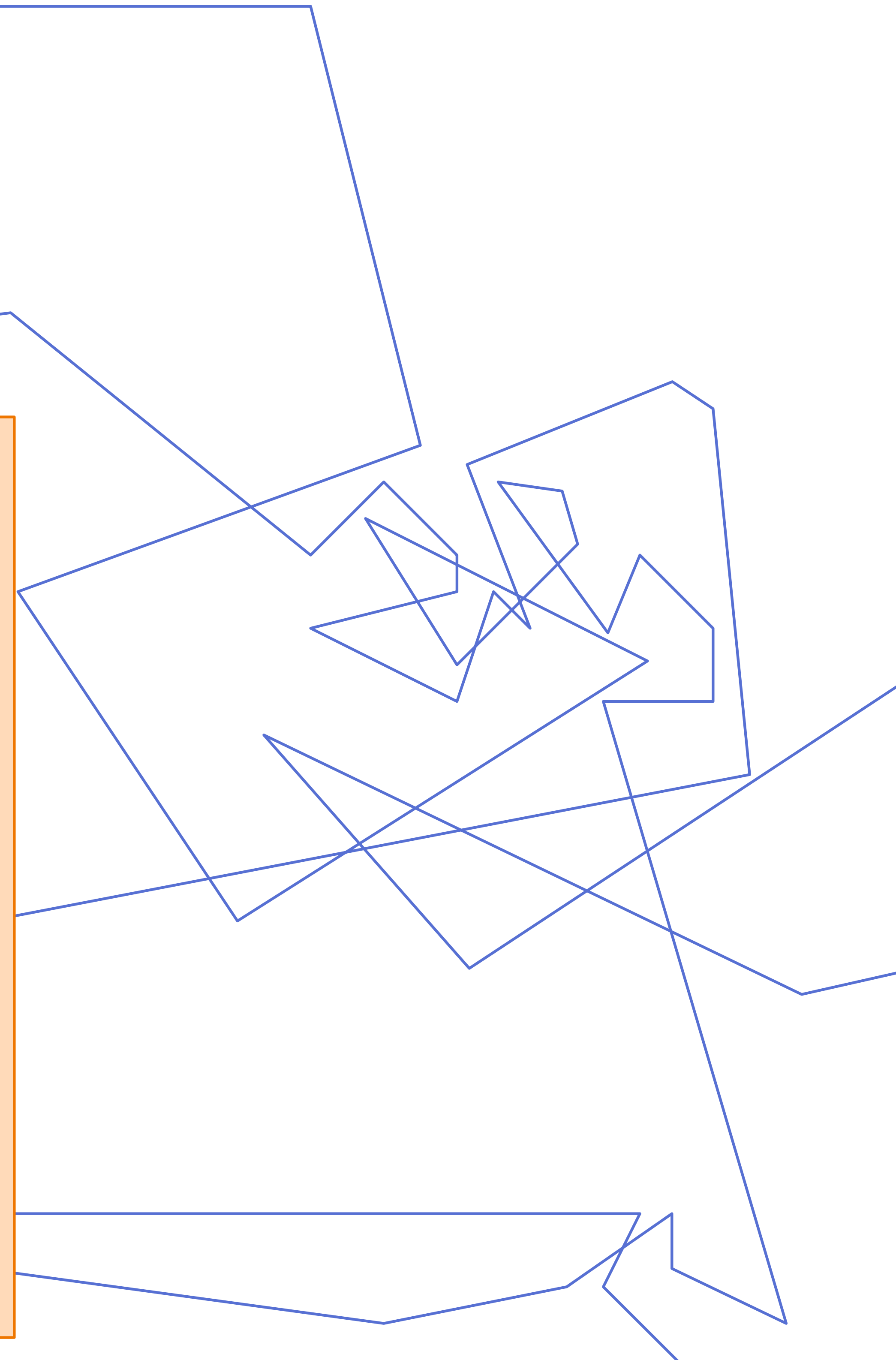
3 edges through corners of $\mathcal{H}$.



Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

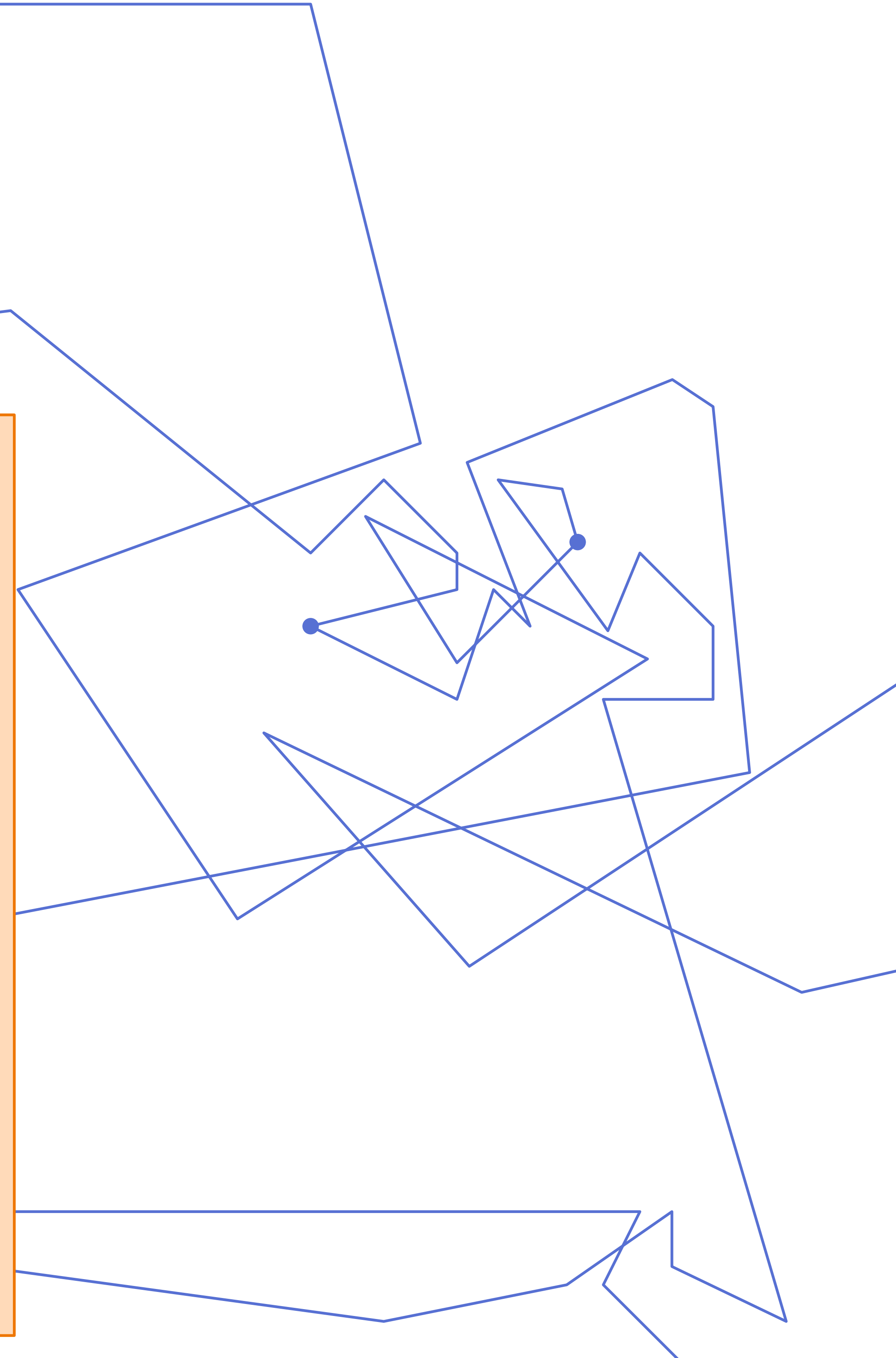


Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

Case: 3 vertices on $\partial\mathcal{H}$.

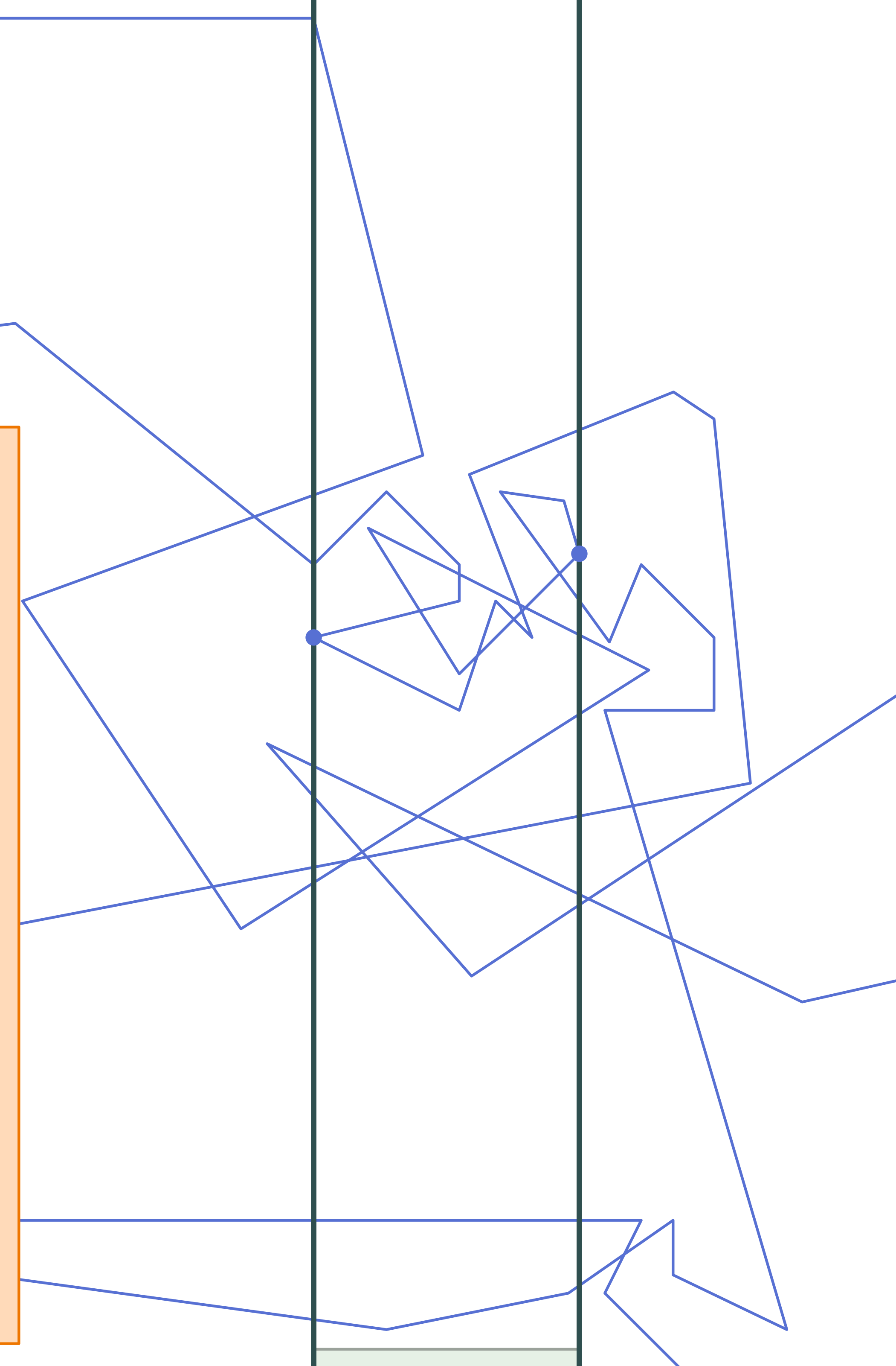


Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

Case: 3 vertices on $\partial\mathcal{H}$.

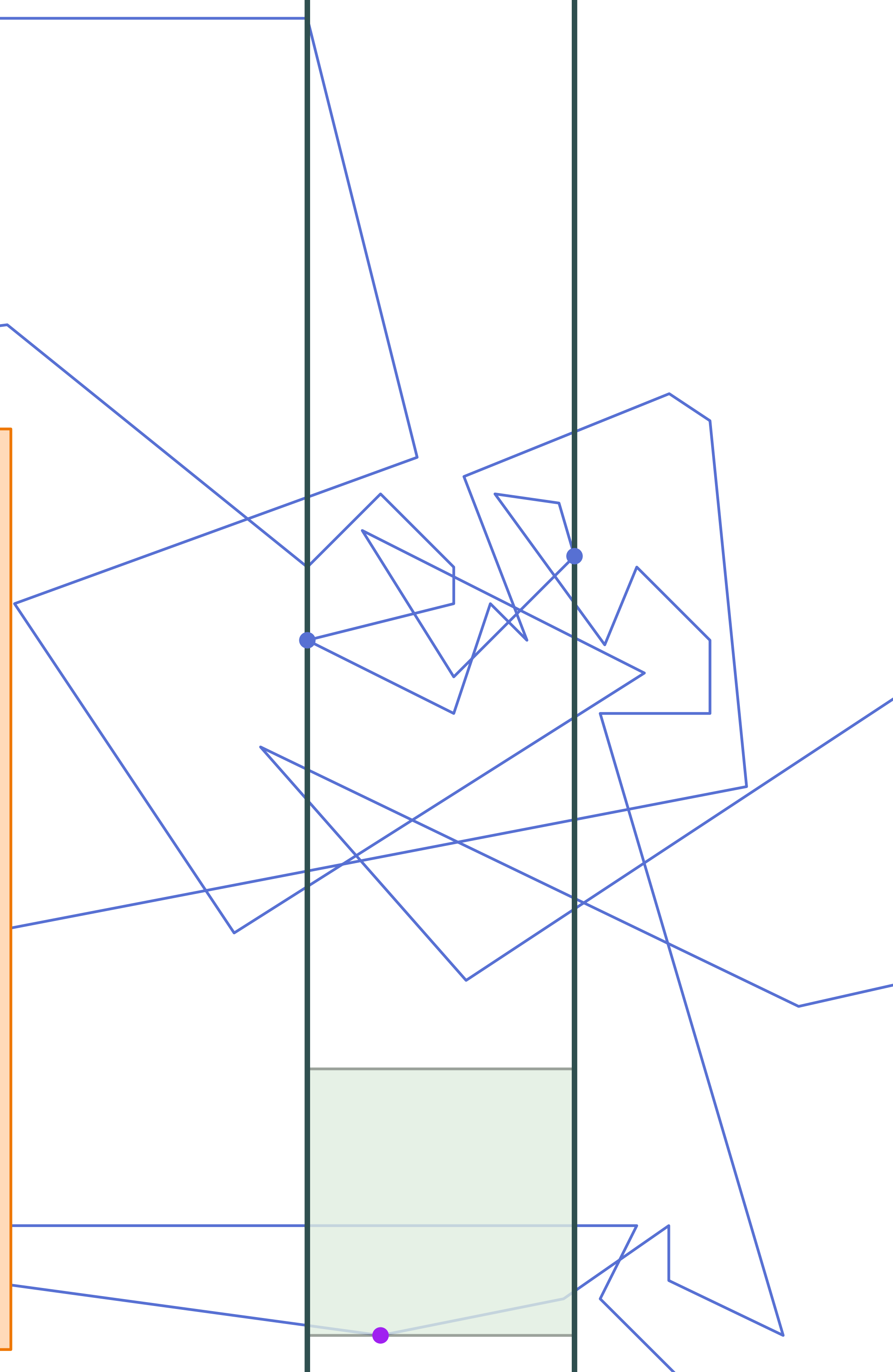


Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

Case: 3 vertices on $\partial\mathcal{H}$.

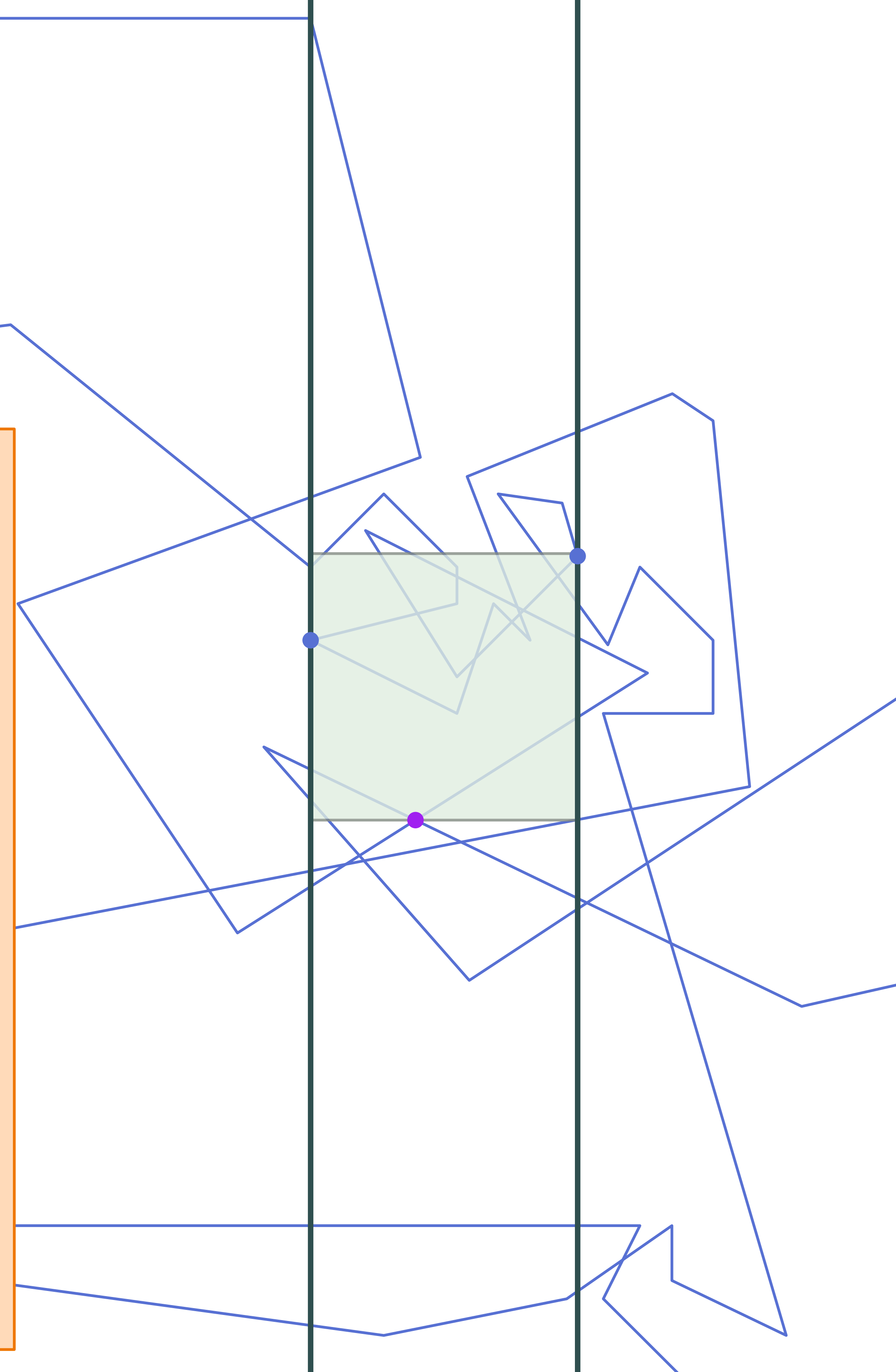


Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

Case: 3 vertices on $\partial\mathcal{H}$.

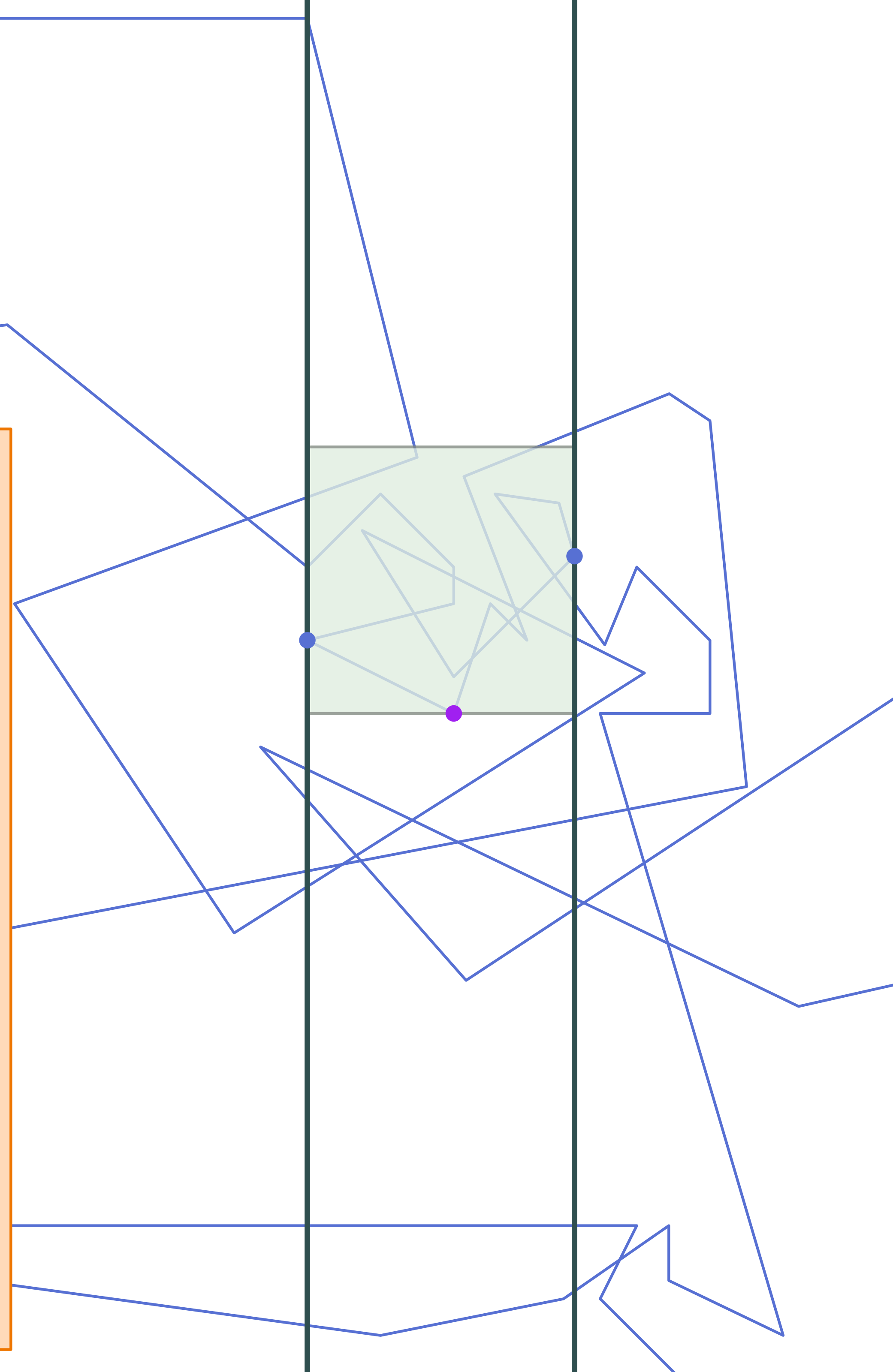


Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

Case: 3 vertices on $\partial\mathcal{H}$.

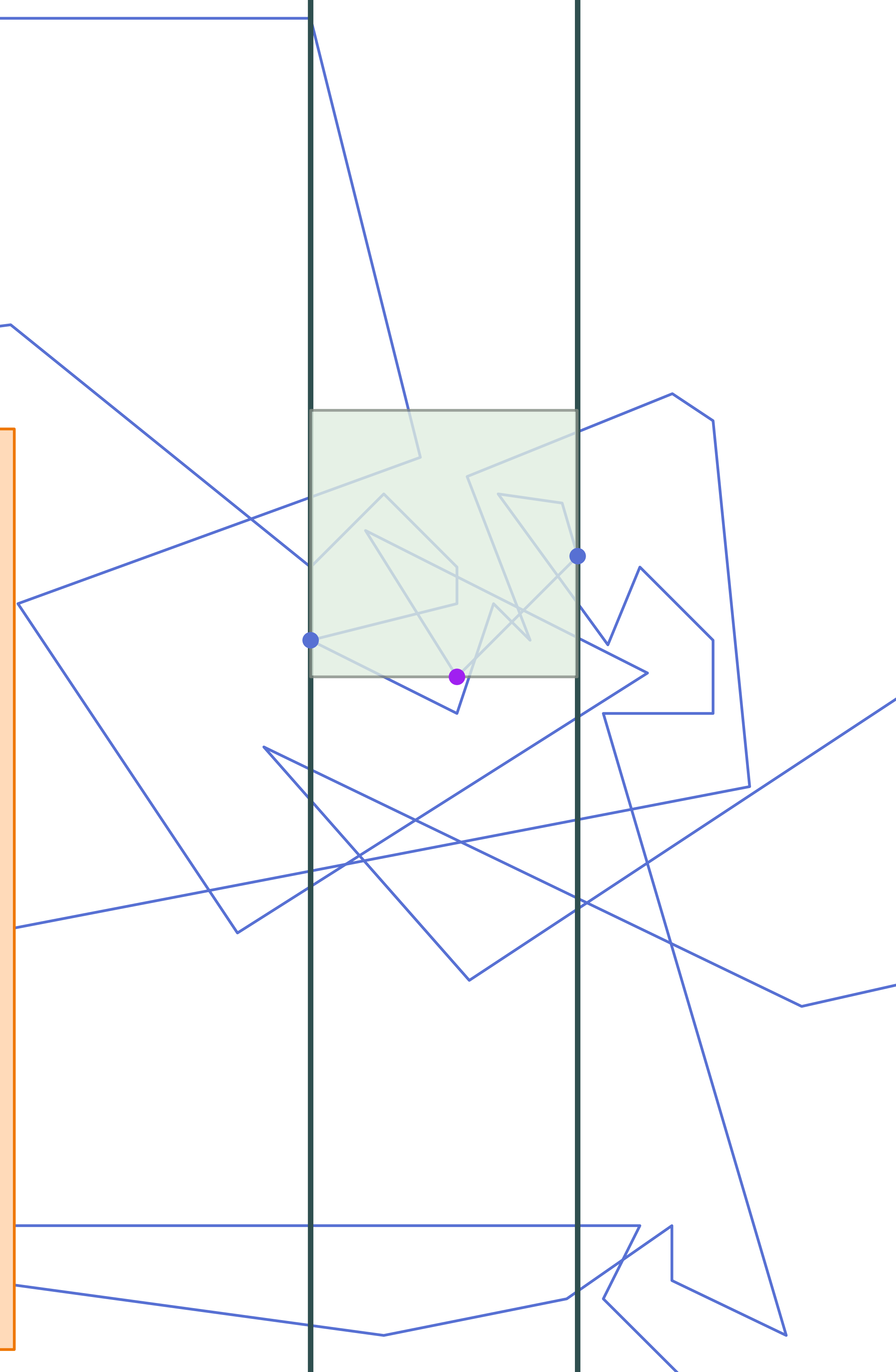


Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

Case: 3 vertices on $\partial\mathcal{H}$.



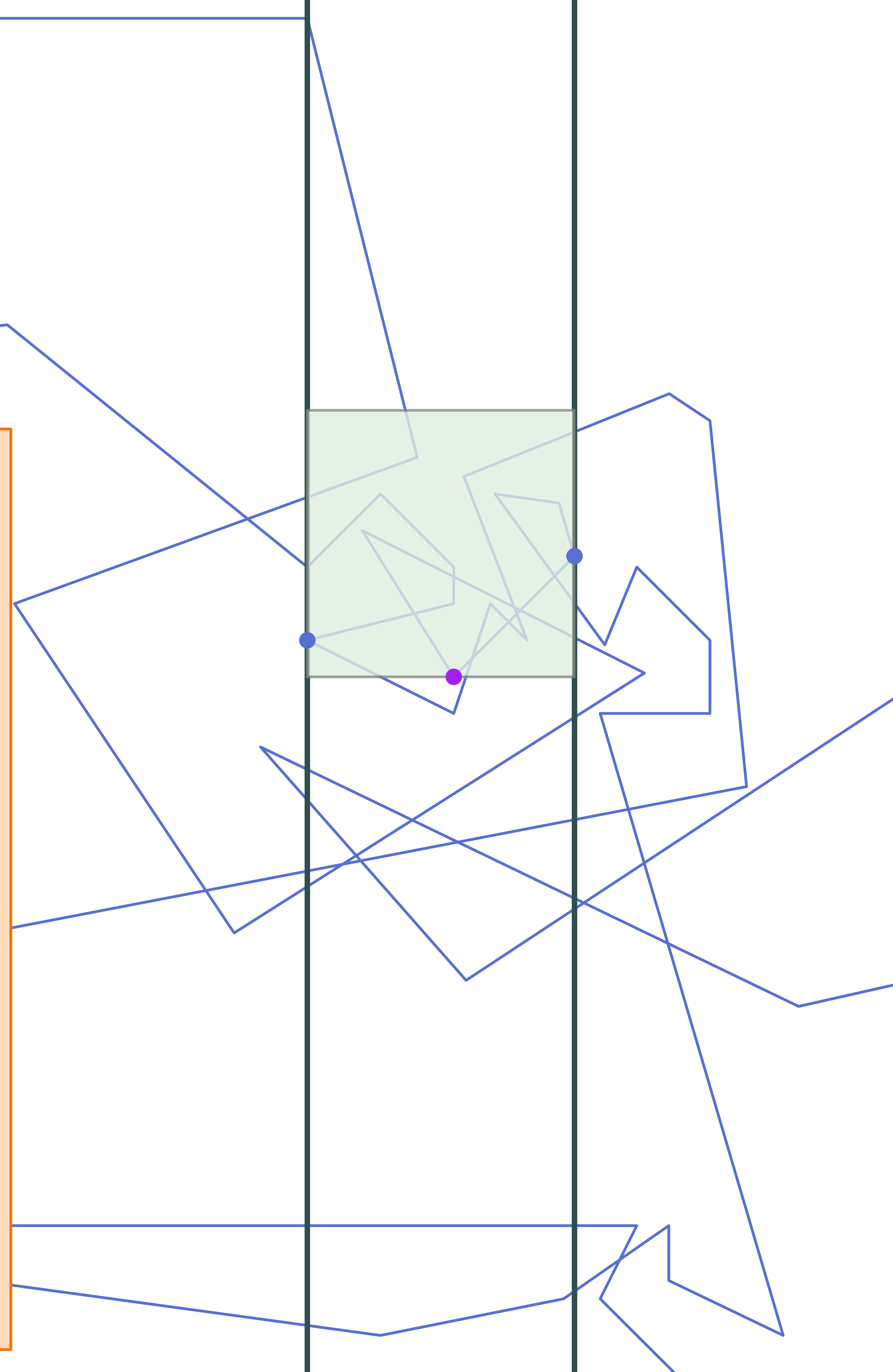
Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

Case: 3 vertices on $\partial\mathcal{H}$.

$O(n)$ breakpoints/events: $O(n)$ time.



Relative Length

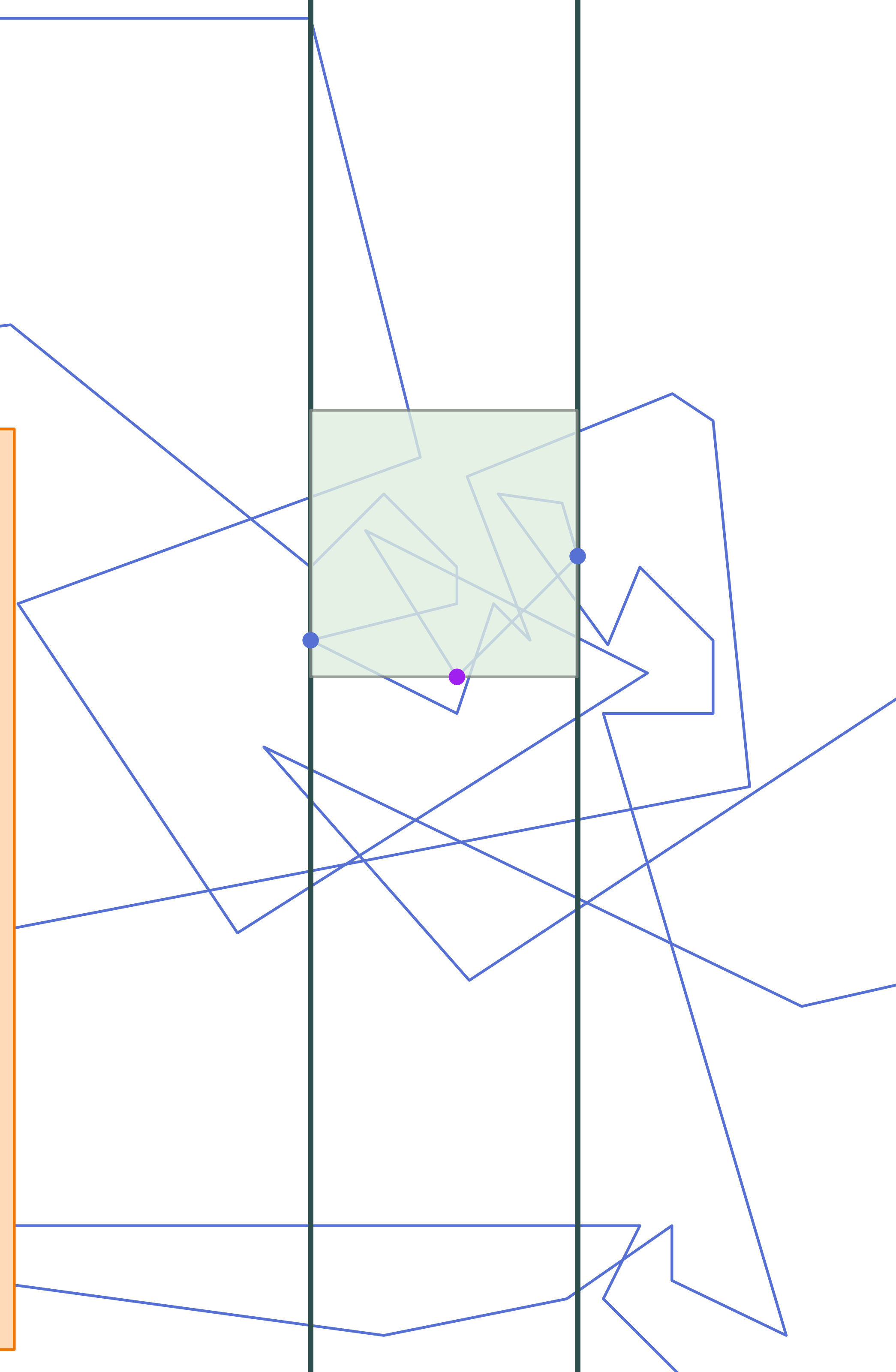
Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

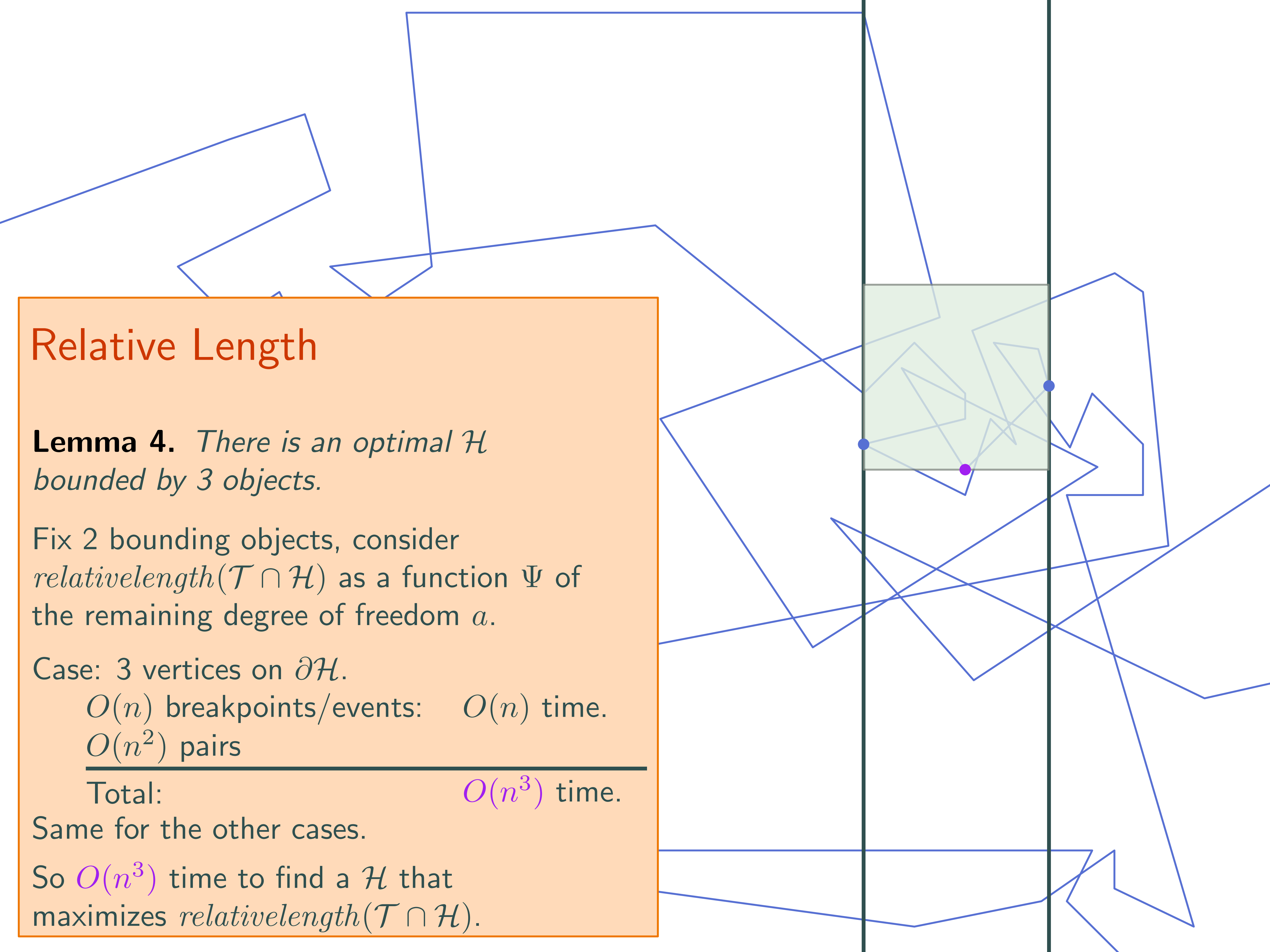
Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

Case: 3 vertices on $\partial\mathcal{H}$.

$O(n)$ breakpoints/events: $O(n)$ time.
$O(n^2)$ pairs

Total: $O(n^3)$ time.





Relative Length

Lemma 4. *There is an optimal $\mathcal{H}$ bounded by 3 objects.*

Fix 2 bounding objects, consider $relativelength(\mathcal{T} \cap \mathcal{H})$ as a function Ψ of the remaining degree of freedom a.

Case: 3 vertices on $\partial\mathcal{H}$.

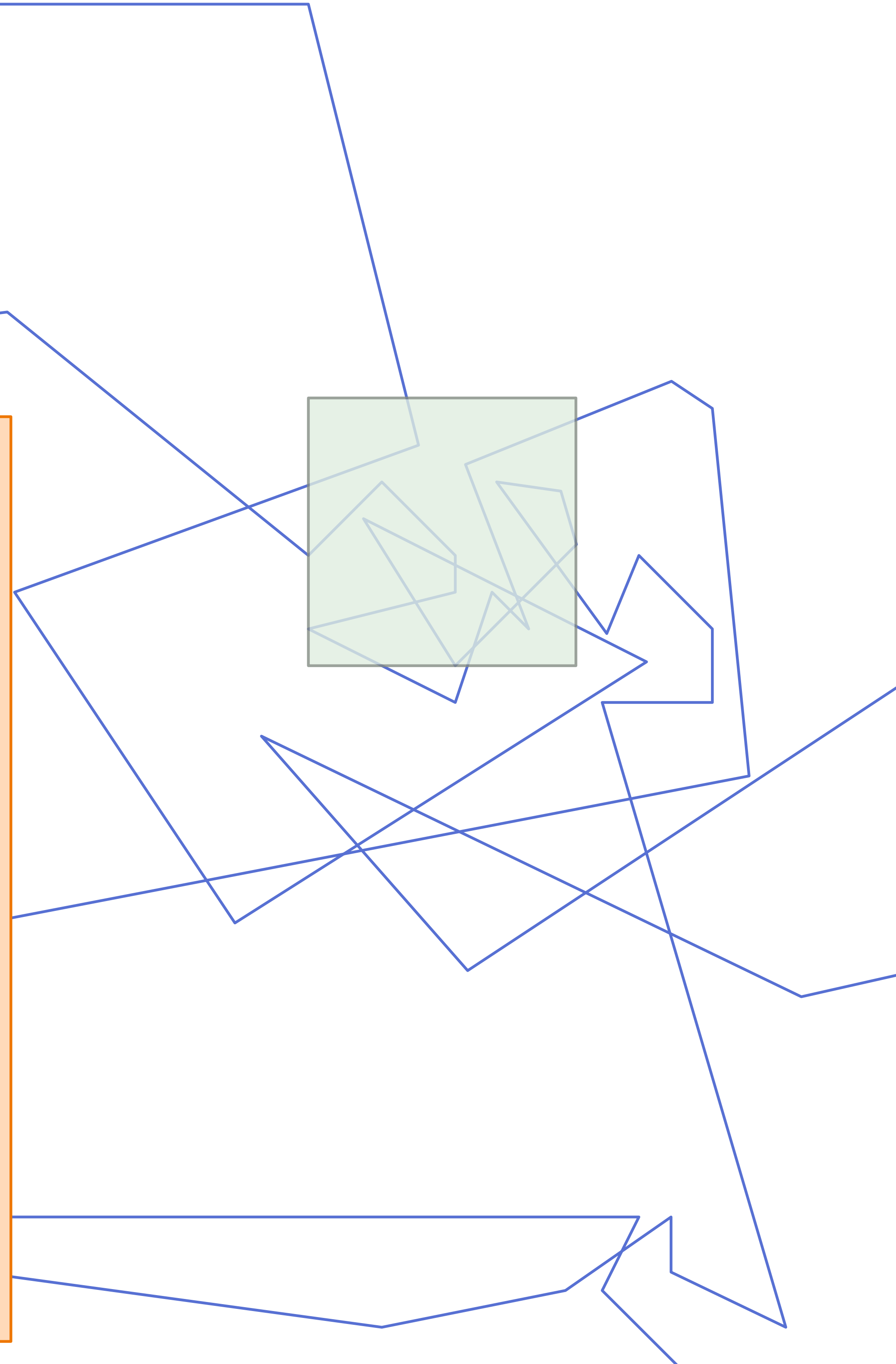
$O(n)$ breakpoints/events:	$O(n)$ time.
$O(n^2)$ pairs	
Total:	$O(n^3)$ time.

Same for the other cases.

So $O(n^3)$ time to find a $\mathcal{H}$ that maximizes $relativelength(\mathcal{T} \cap \mathcal{H})$.

Hotspot Shapes

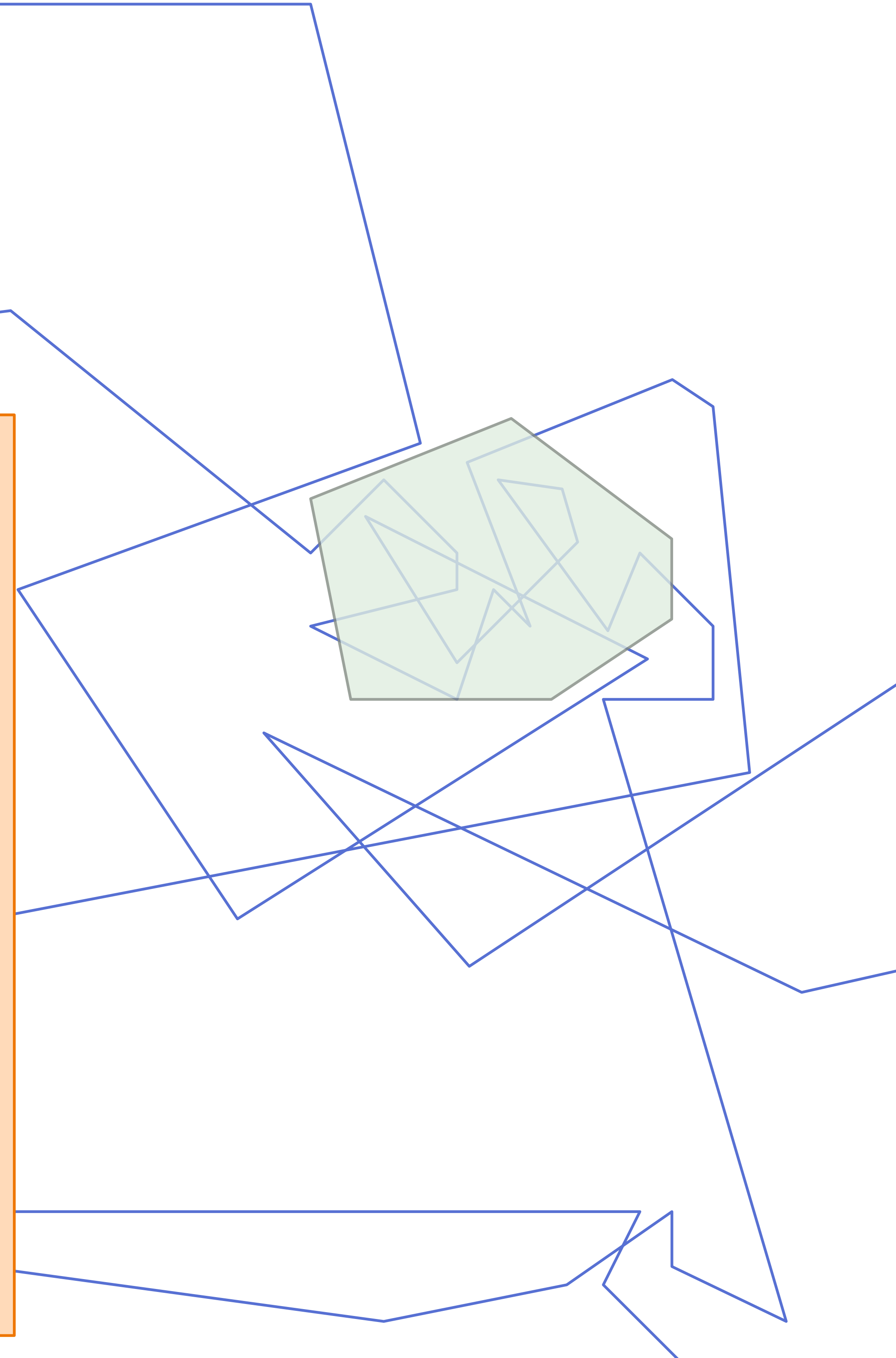
Can we handle hotspots of a different shape?



Hotspot Shapes

Can we handle hotspots of a different shape?

$\mathcal{H}$ is a convex polygon of given shape: yes

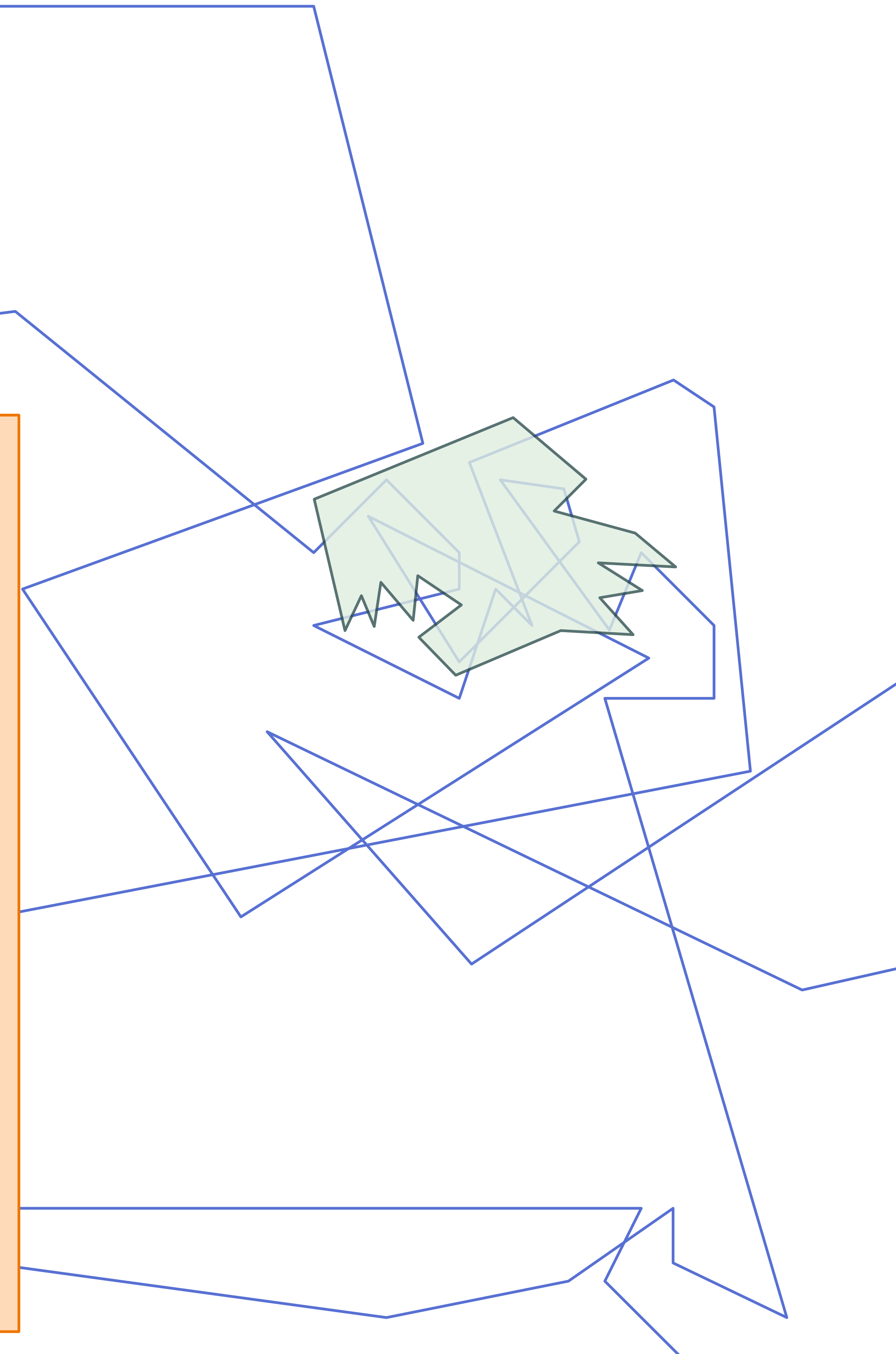


Hotspot Shapes

Can we handle hotspots of a different shape?

$\mathcal{H}$ is a convex polygon of given shape: yes

$\mathcal{H}$ is a polygon of given shape: no



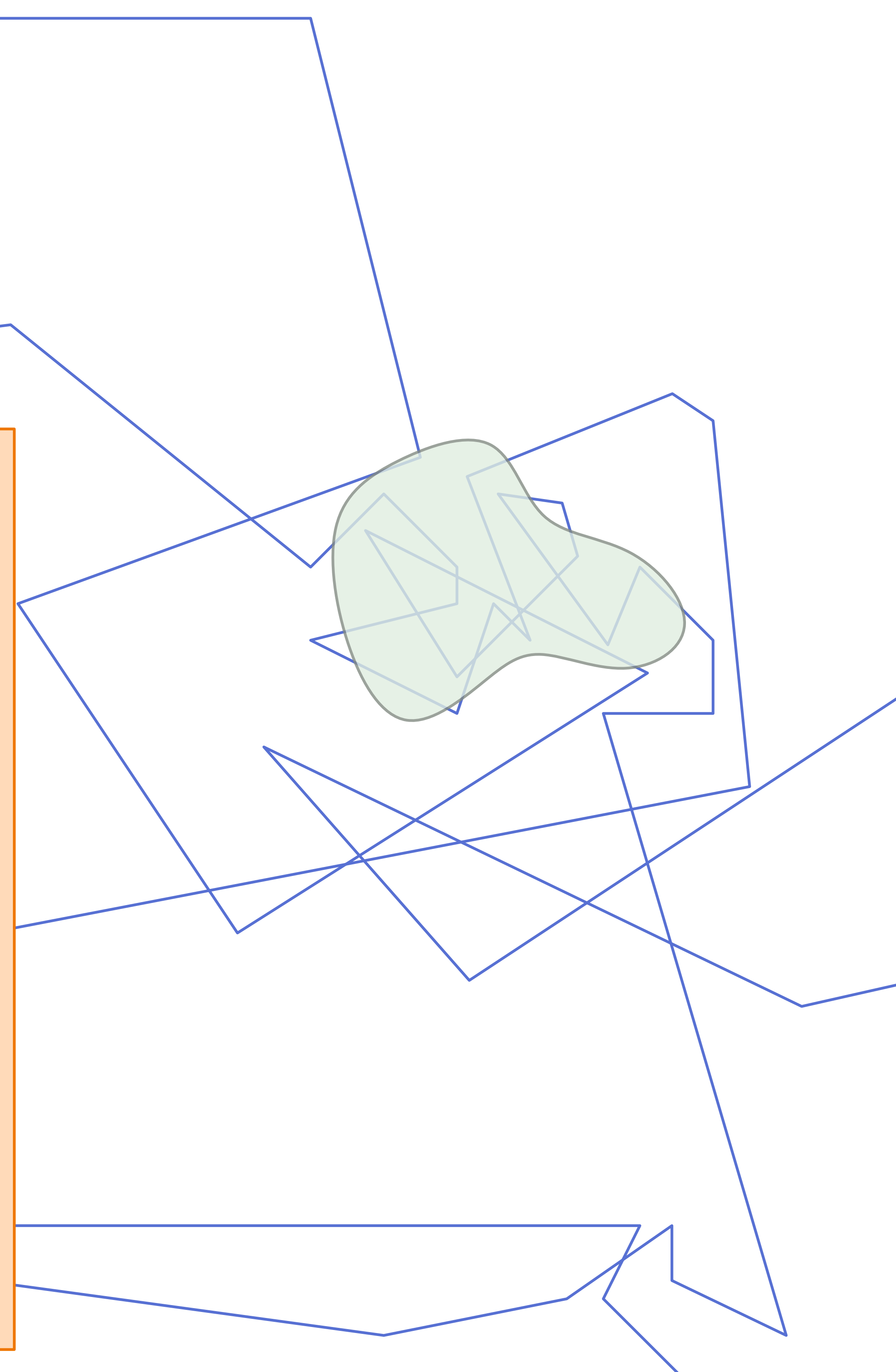
Hotspot Shapes

Can we handle hotspots of a different shape?

$\mathcal{H}$ is a convex polygon of given shape: yes

$\mathcal{H}$ is a polygon of given shape: no

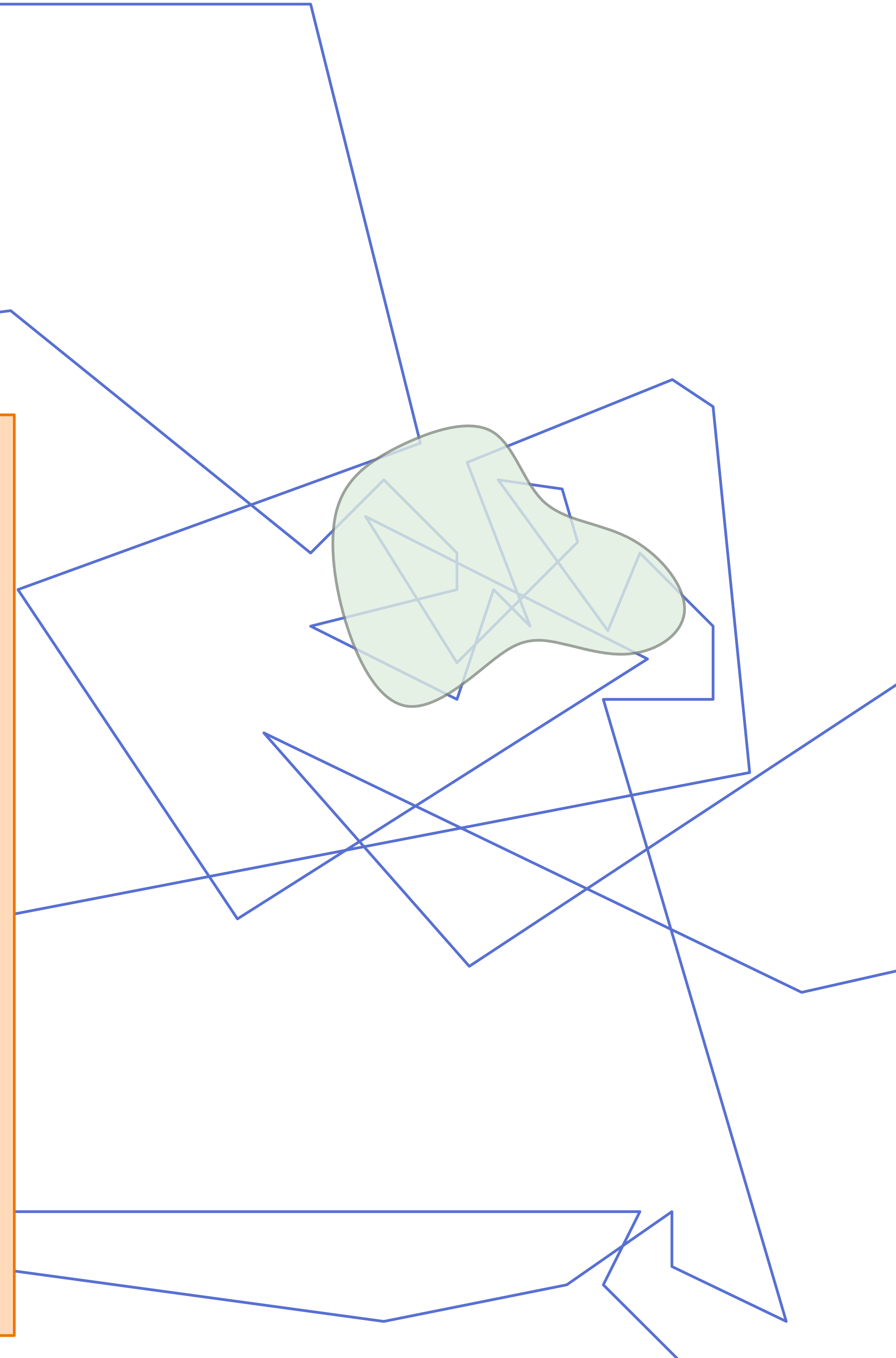
$\mathcal{H}$ has fixed but curved boundaries: no



Hotspot Shapes

Can we handle hotspots of a different shape?

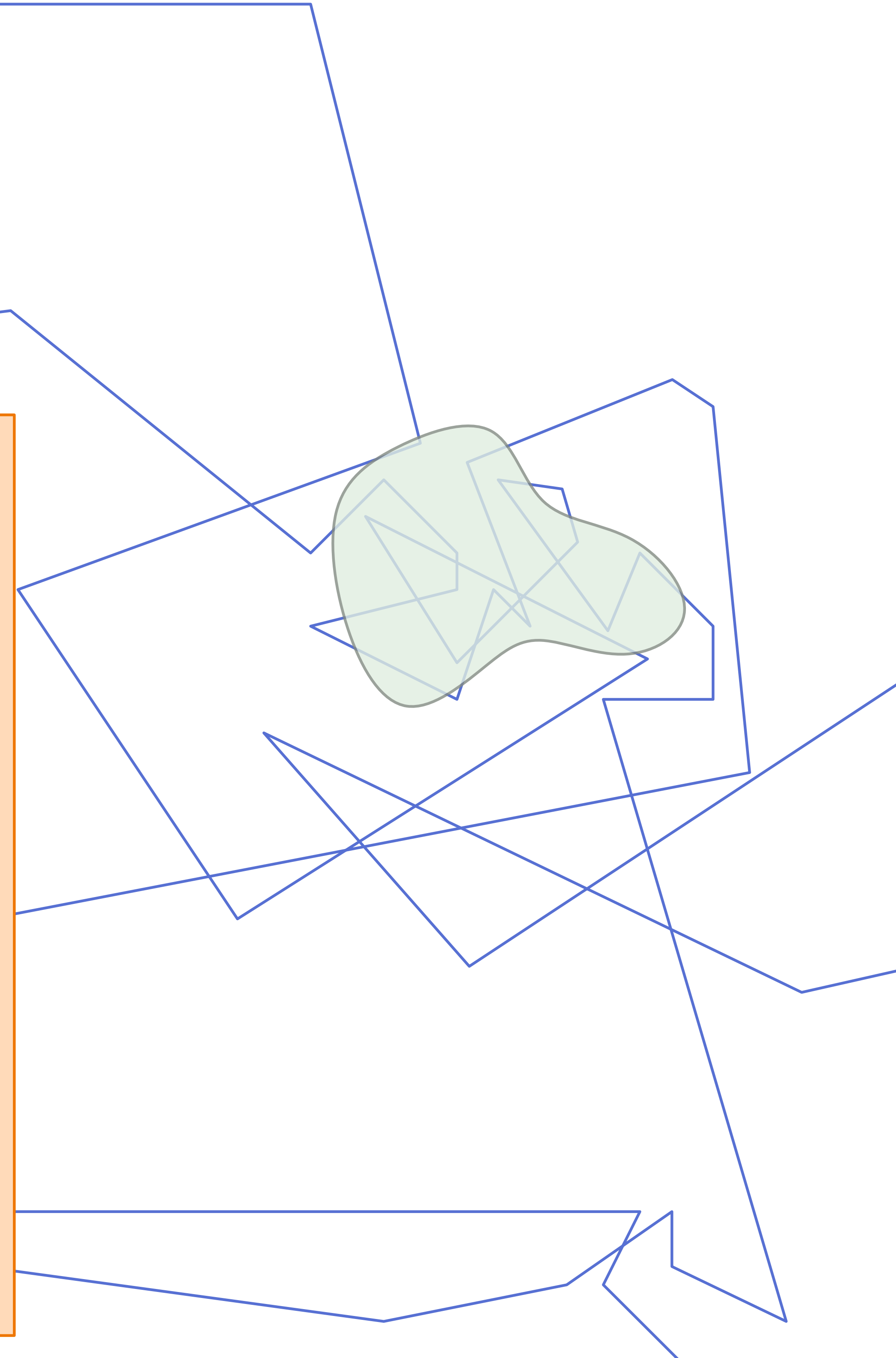
$\mathcal{H}$ is a convex polygon of given shape:	yes
$\mathcal{H}$ is a polygon of given shape:	no
$\mathcal{H}$ has fixed but curved boundaries:	no
The shape of $\mathcal{H}$ is not predefined:	no



~~Hotspot Shapes~~ Future Work

Can we handle hotspots of a different shape?

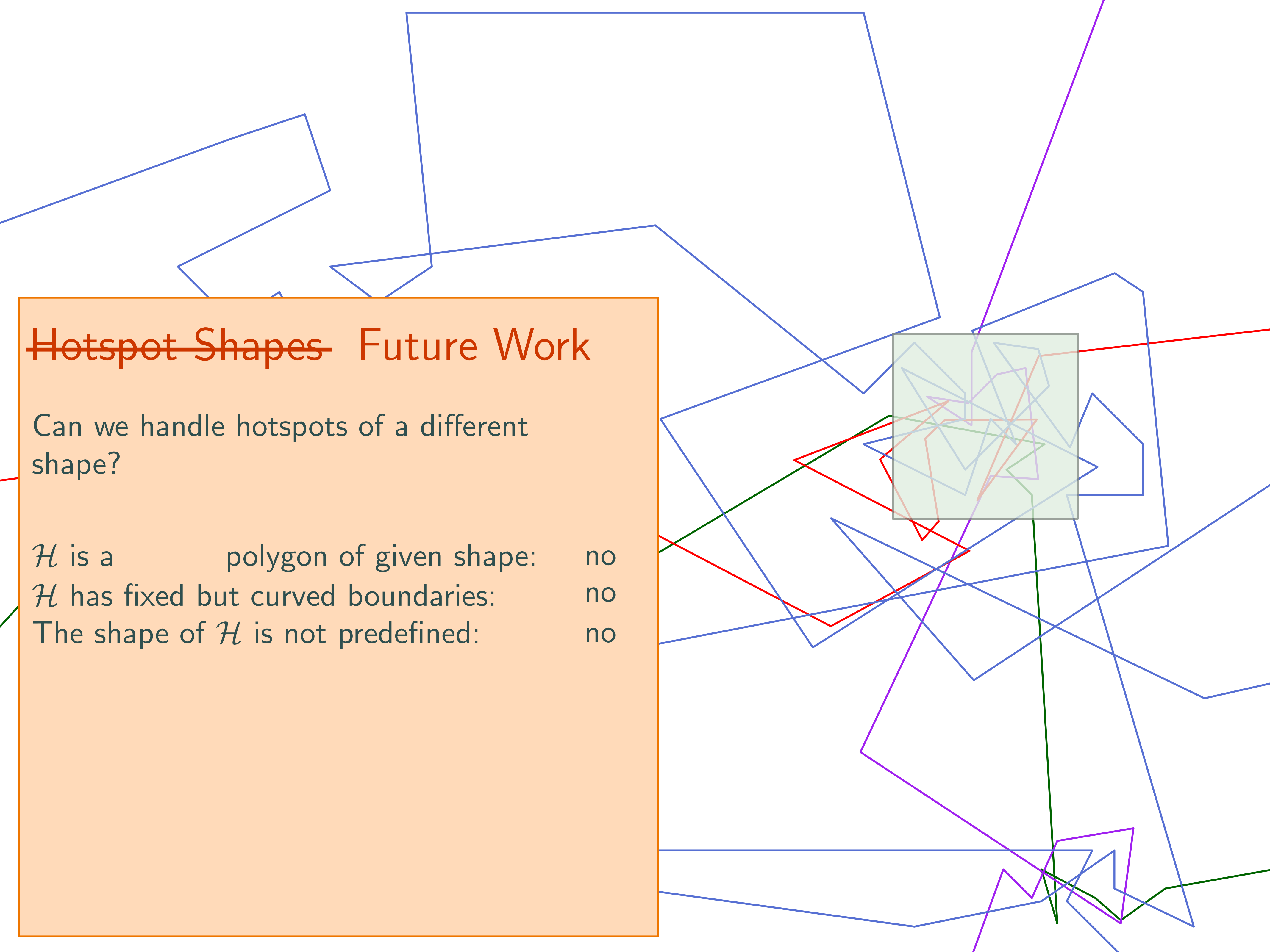
$\mathcal{H}$ is a polygon of given shape:	no
$\mathcal{H}$ has fixed but curved boundaries:	no
The shape of $\mathcal{H}$ is not predefined:	no

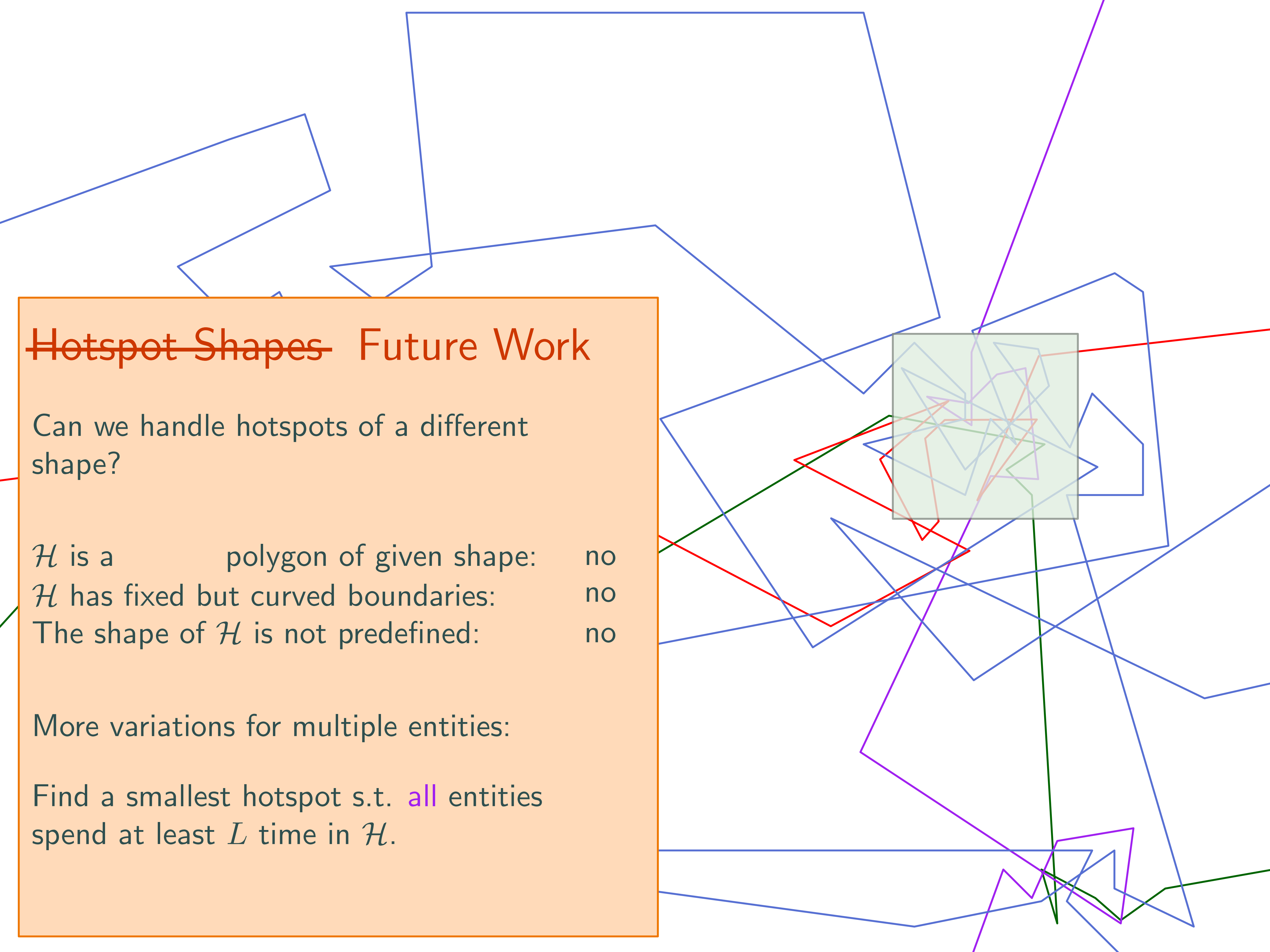


~~Hotspot Shapes~~ Future Work

Can we handle hotspots of a different shape?

$\mathcal{H}$ is a polygon of given shape:	no
$\mathcal{H}$ has fixed but curved boundaries:	no
The shape of $\mathcal{H}$ is not predefined:	no





~~Hotspot Shapes~~ Future Work

Can we handle hotspots of a different shape?

$\mathcal{H}$ is a polygon of given shape: no
$\mathcal{H}$ has fixed but curved boundaries: no
The shape of $\mathcal{H}$ is not predefined: no

More variations for multiple entities:

Find a smallest hotspot s.t. all entities spend at least L time in $\mathcal{H}$.

~~Hotspot Shapes~~ Future Work

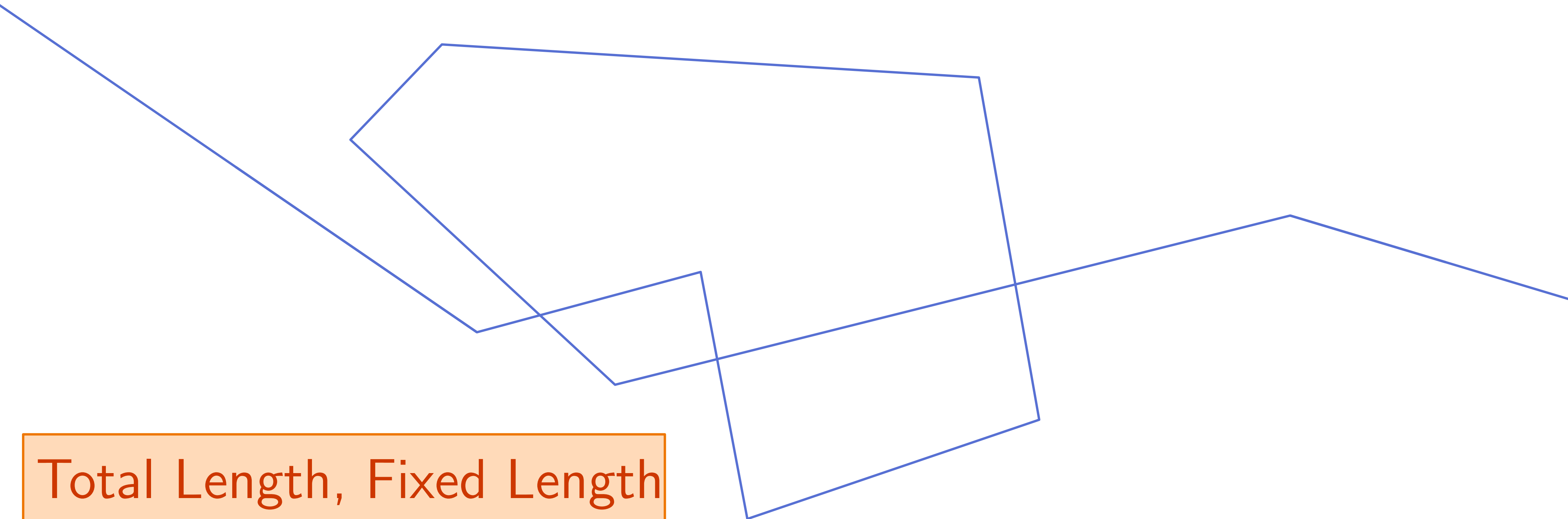
Can we handle hotspots of a different shape?

$\mathcal{H}$ is a polygon of given shape: no
$\mathcal{H}$ has fixed but curved boundaries: no
The shape of $\mathcal{H}$ is not predefined: no

More variations for multiple entities:

Find a smallest hotspot s.t. all entities spend at least L time in $\mathcal{H}$.

Thank you!



Total Length, Fixed Length

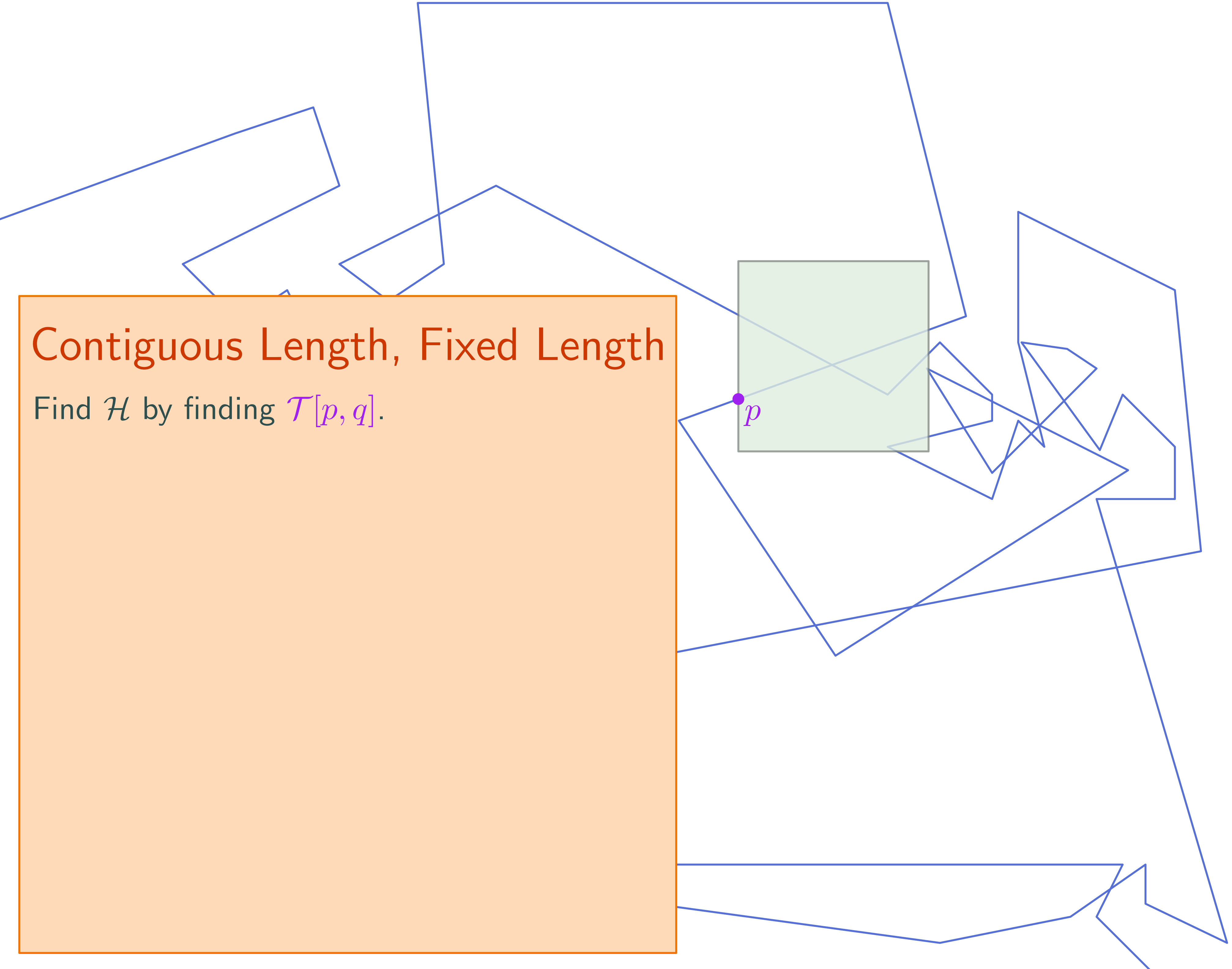
Goal: minimize the side length of $\mathcal{H}$ for a fixed trajectory length L.

Use parametric search, using the Fixed-Size algorithm as a decision algorithm.

Running time: $O(n^2 \log^2 n)$.

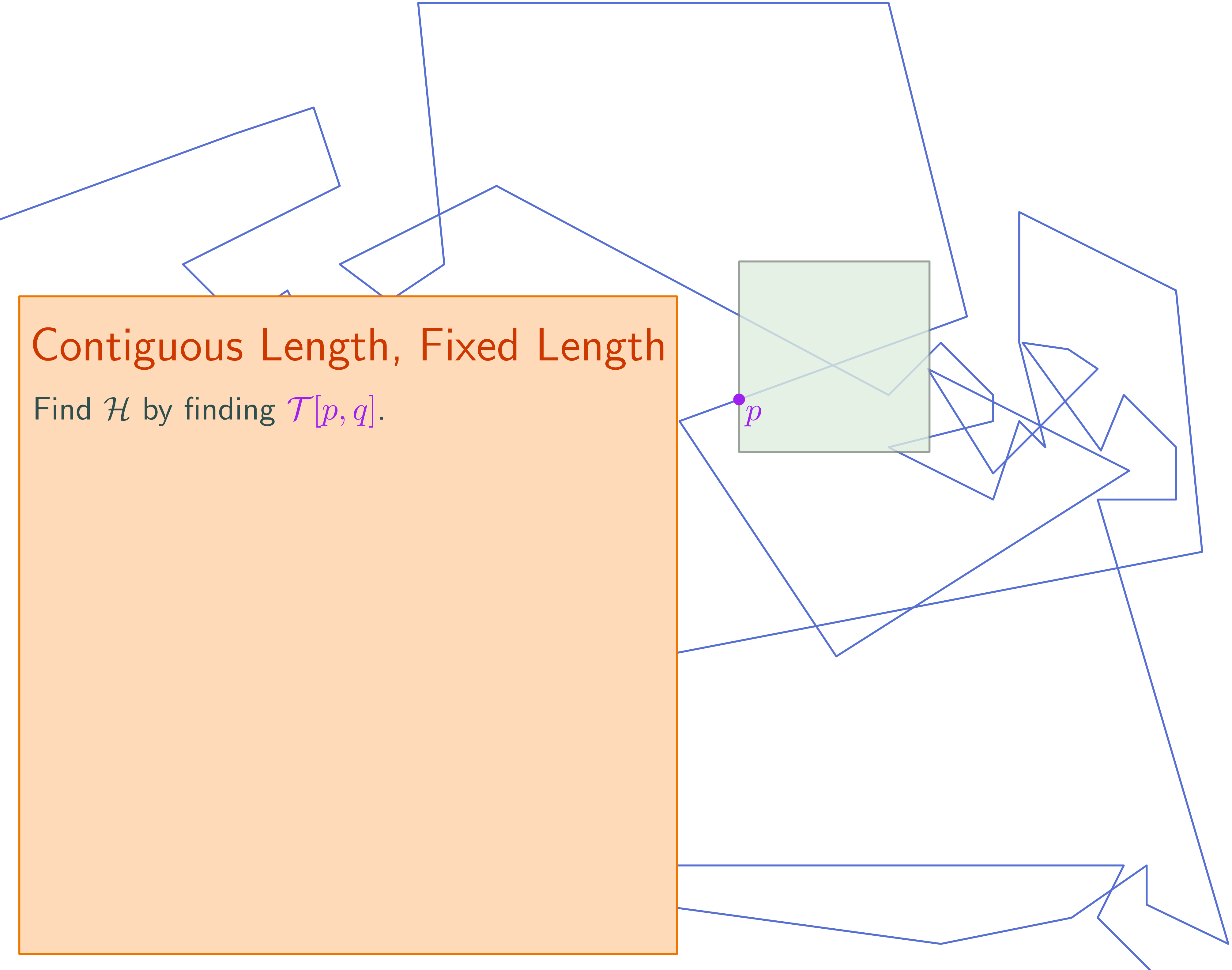
Contiguous Length, Fixed Length

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.



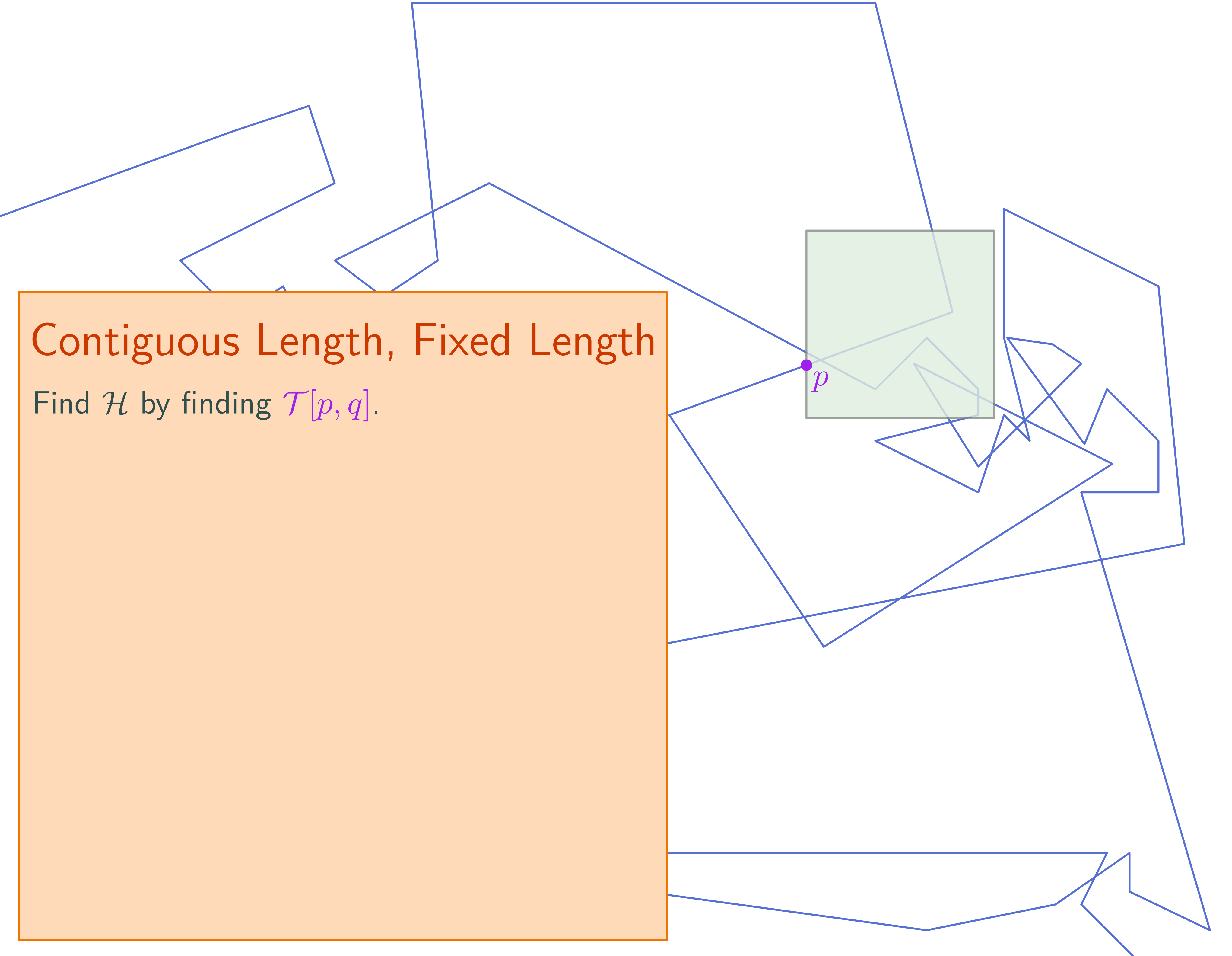
Contiguous Length, Fixed Length

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.



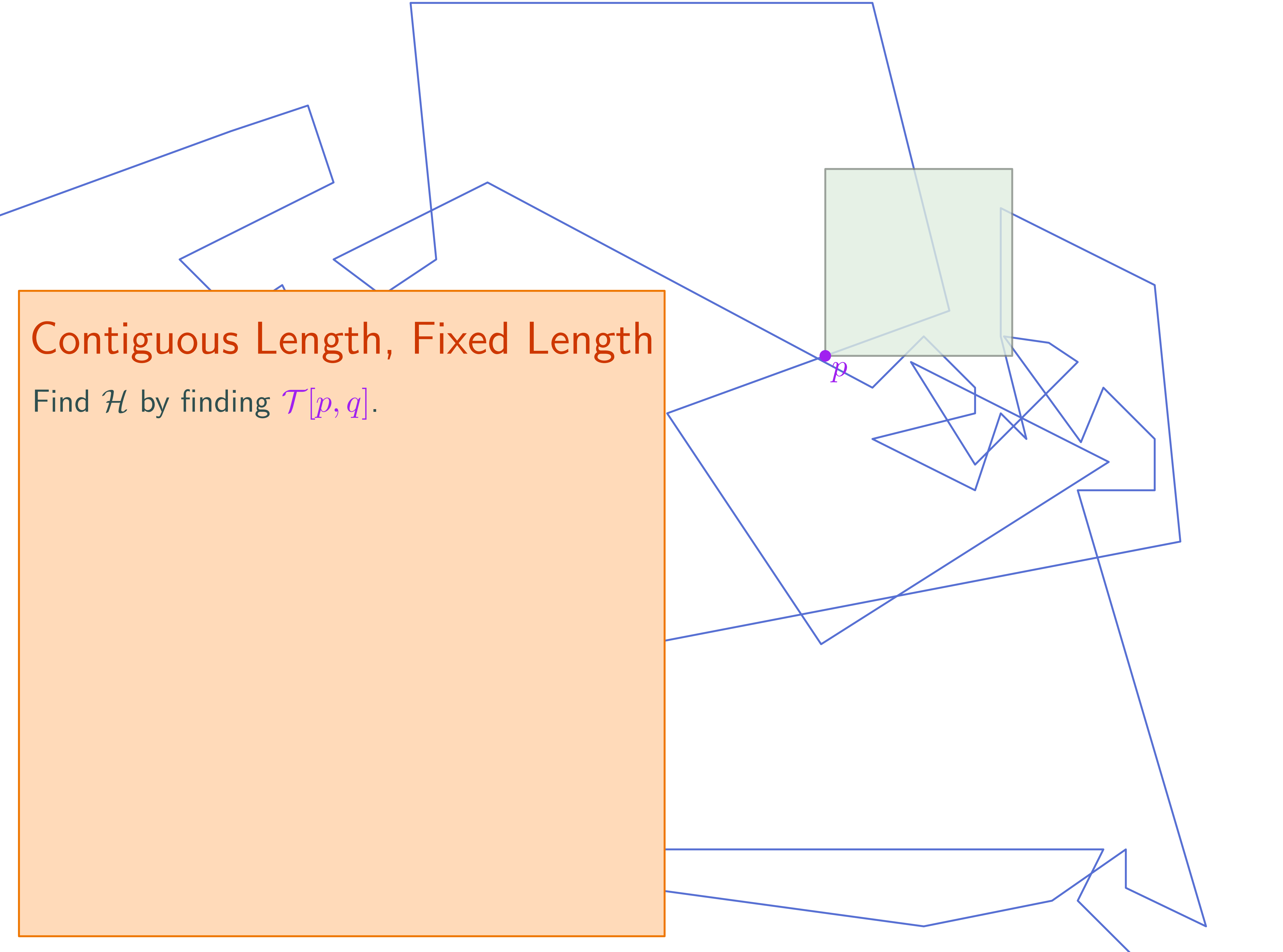
Contiguous Length, Fixed Length

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.



Contiguous Length, Fixed Length

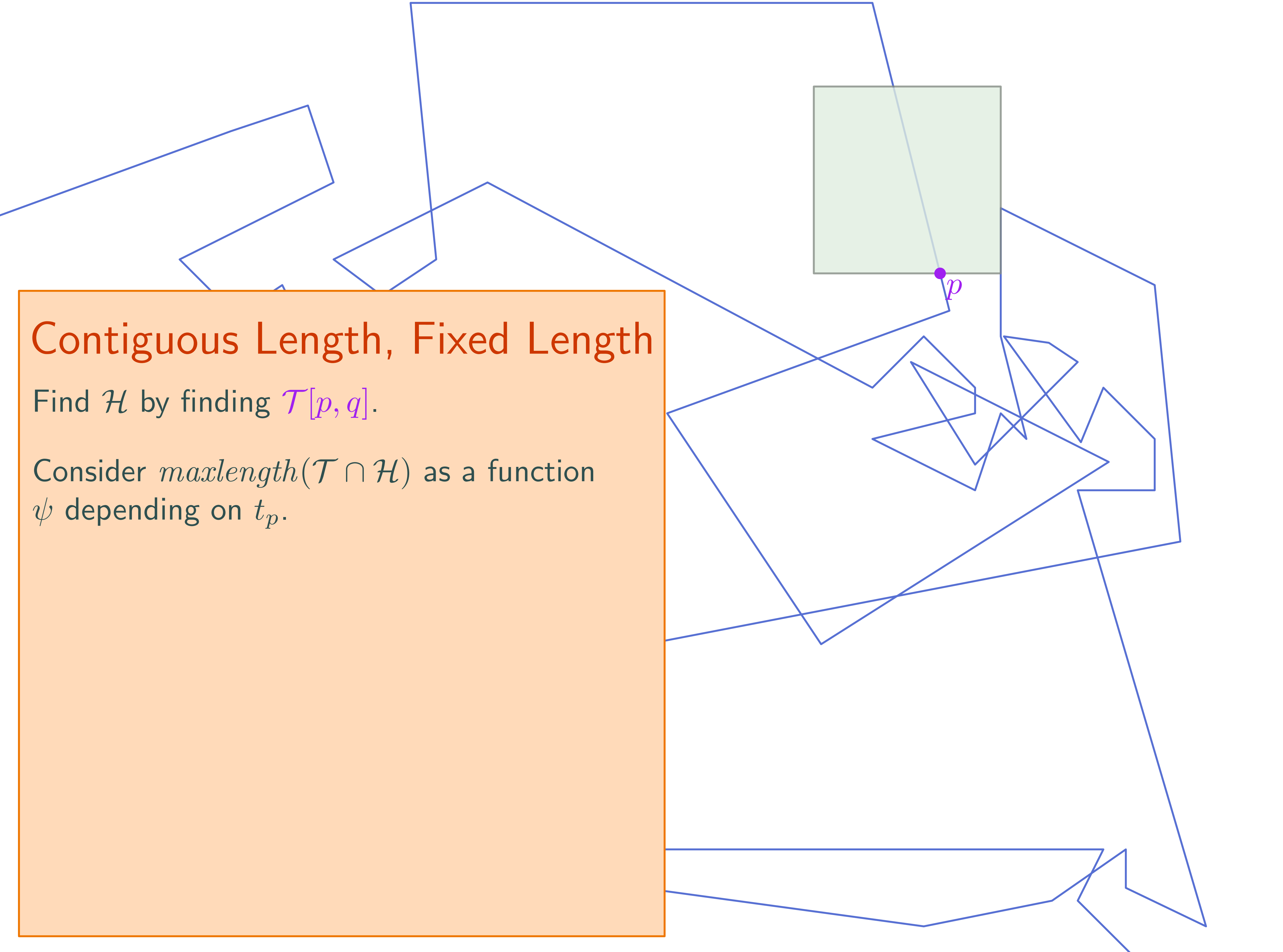
Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

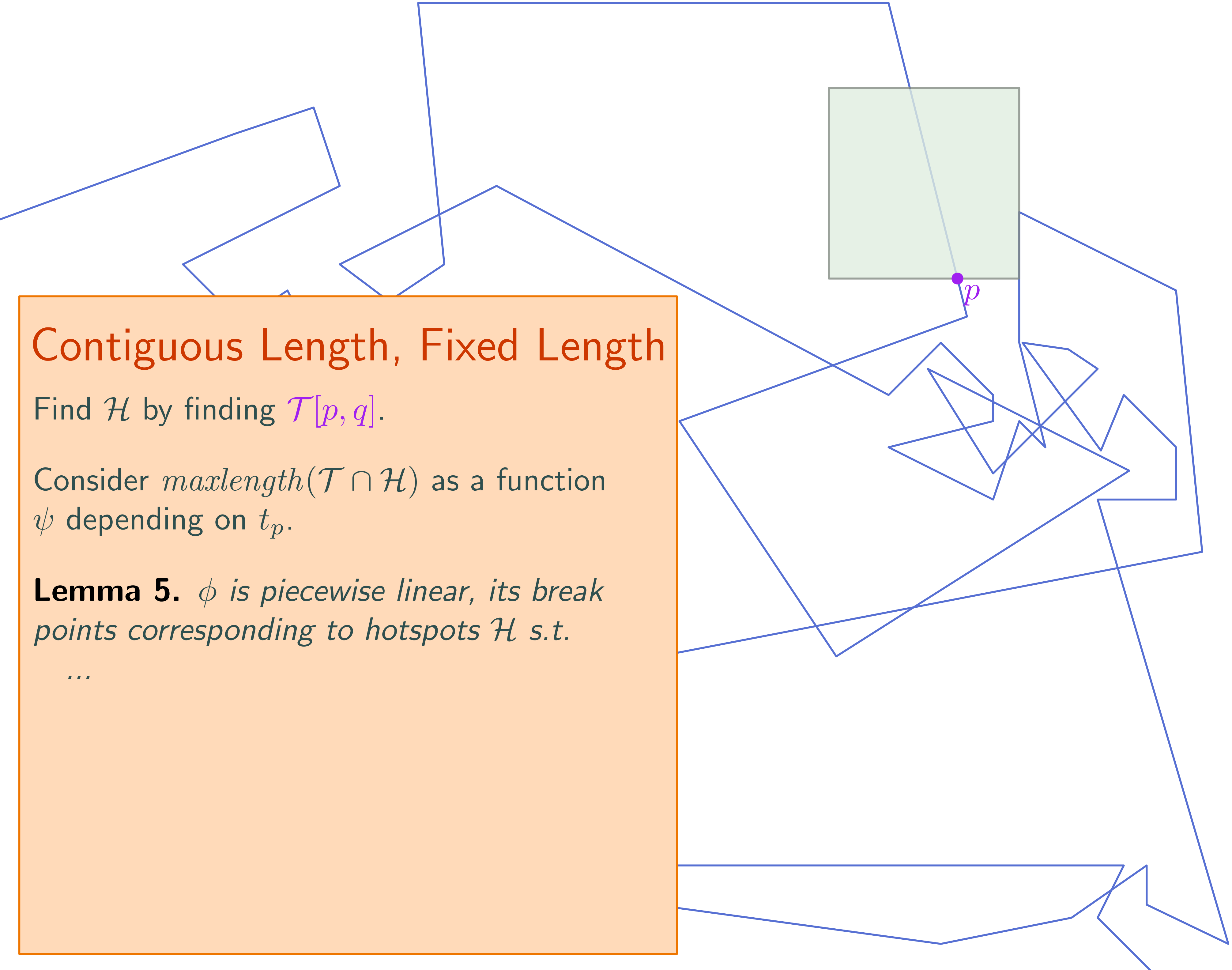


Contiguous Length, Fixed Length

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

Consider $maxlength(\mathcal{T} \cap \mathcal{H})$ as a function ψ depending on t_p.





Contiguous Length, Fixed Length

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

Consider $maxlength(\mathcal{T} \cap \mathcal{H})$ as a function ψ depending on t_p.

Lemma 5. *ϕ is piecewise linear, its break points corresponding to hotspots $\mathcal{H}$ s.t.*

...

Contiguous Length, Fixed Length

Find $\mathcal{H}$ by finding $\mathcal{T}[p, q]$.

Consider $maxlength(\mathcal{T} \cap \mathcal{H})$ as a function ψ depending on t_p.

Lemma 5. *ϕ is piecewise linear, its break points corresponding to hotspots $\mathcal{H}$ s.t.*

...

Compute ϕ at all break points and select the maximum.

Running time: $O(n \log n)$