Computational Geometry – Exercise

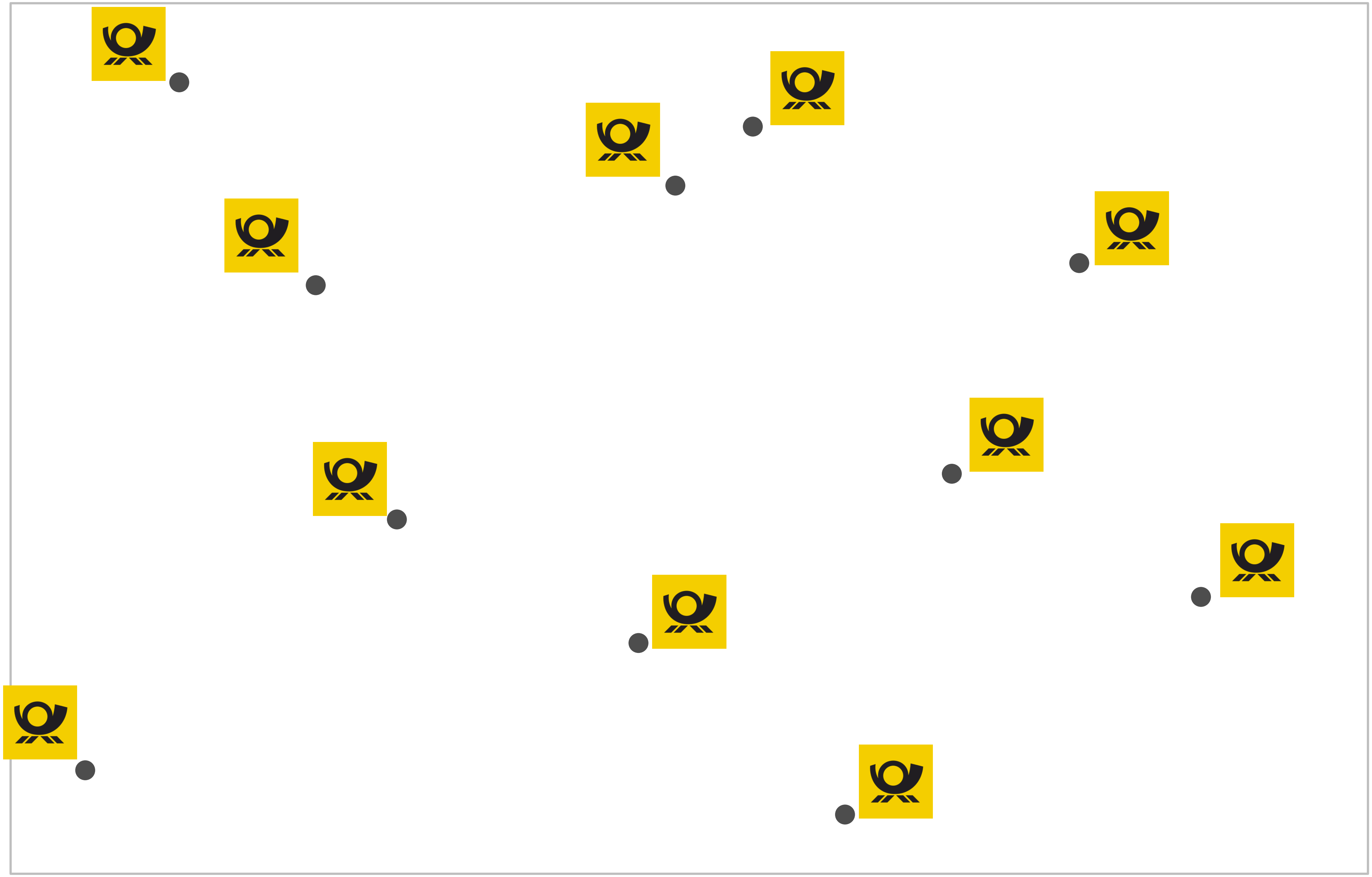
Voronoi-Diagrams

LEHRSTUHL FÜR ALGORITHMIK I · INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

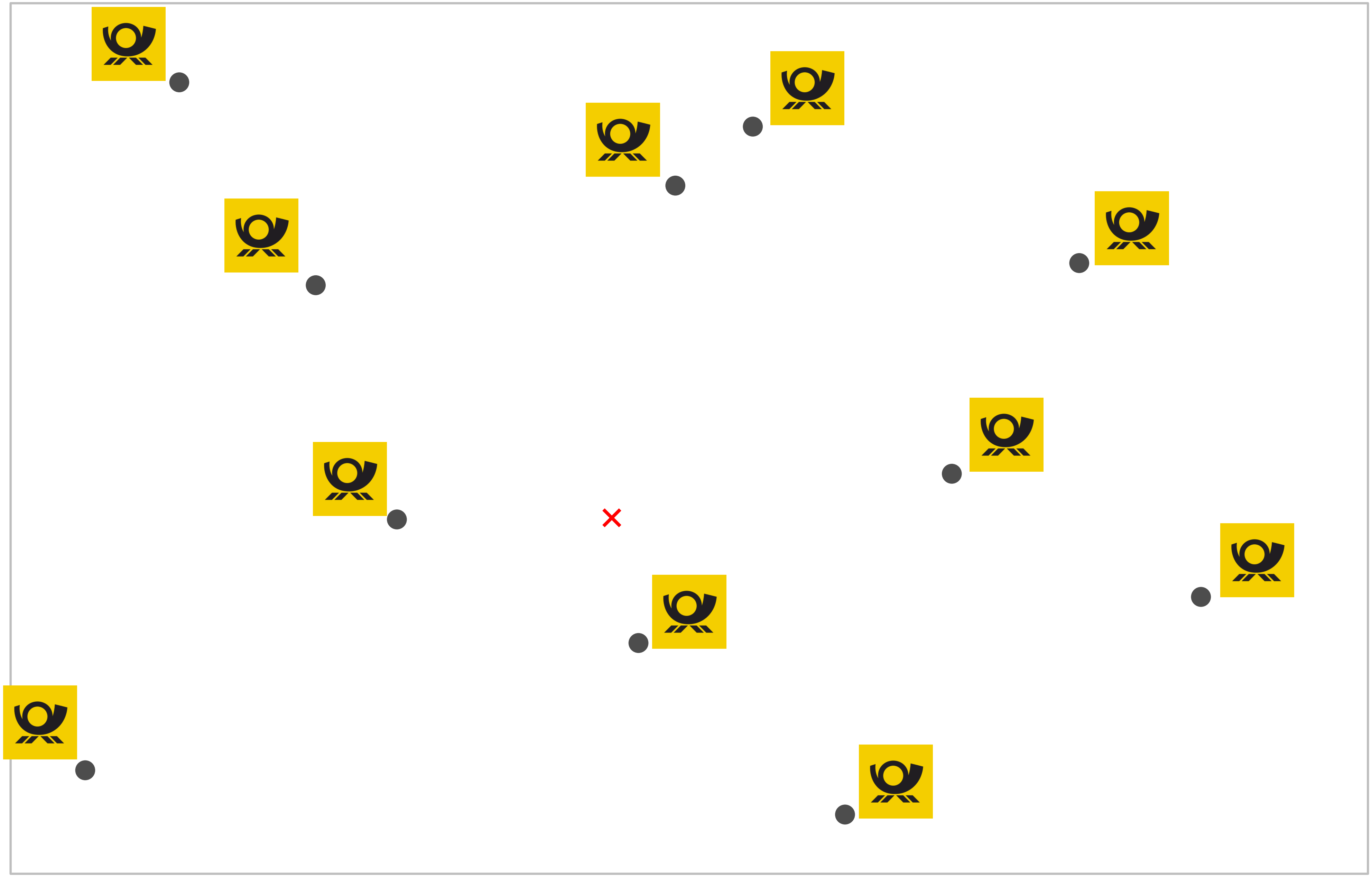
Guido Brückner

15.06.2018

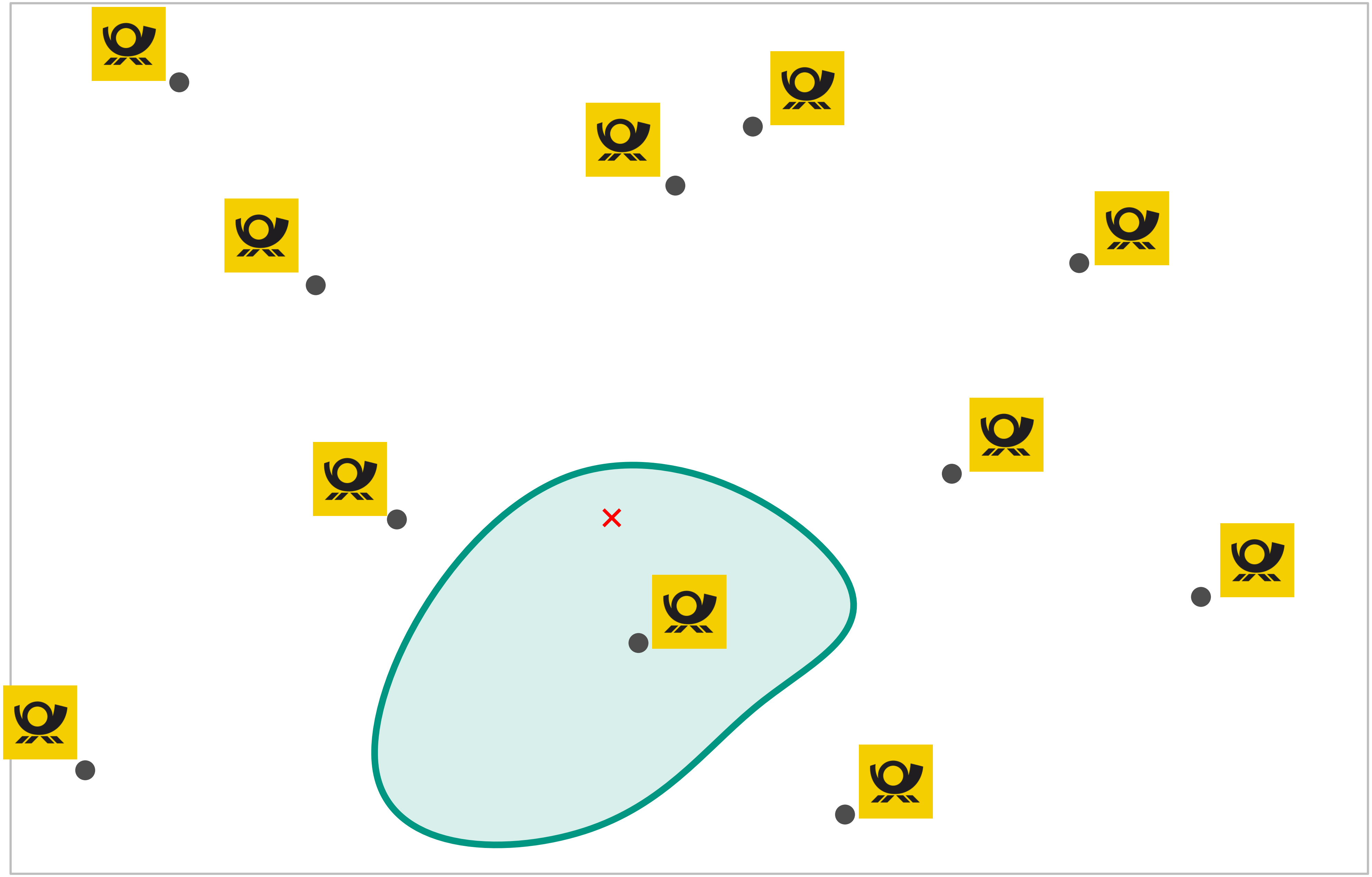
The Post Office Problem



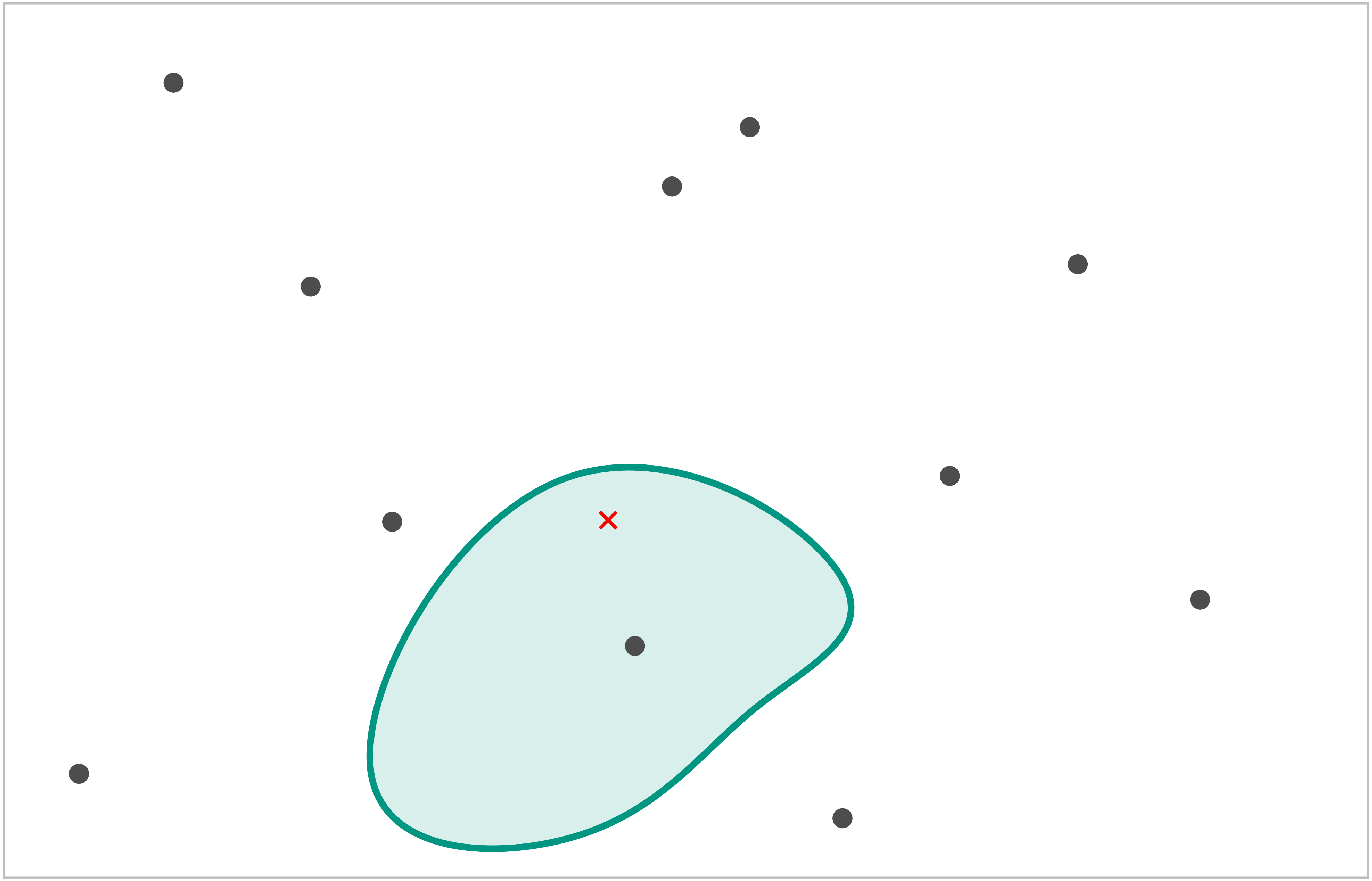
The Post Office Problem



The Post Office Problem

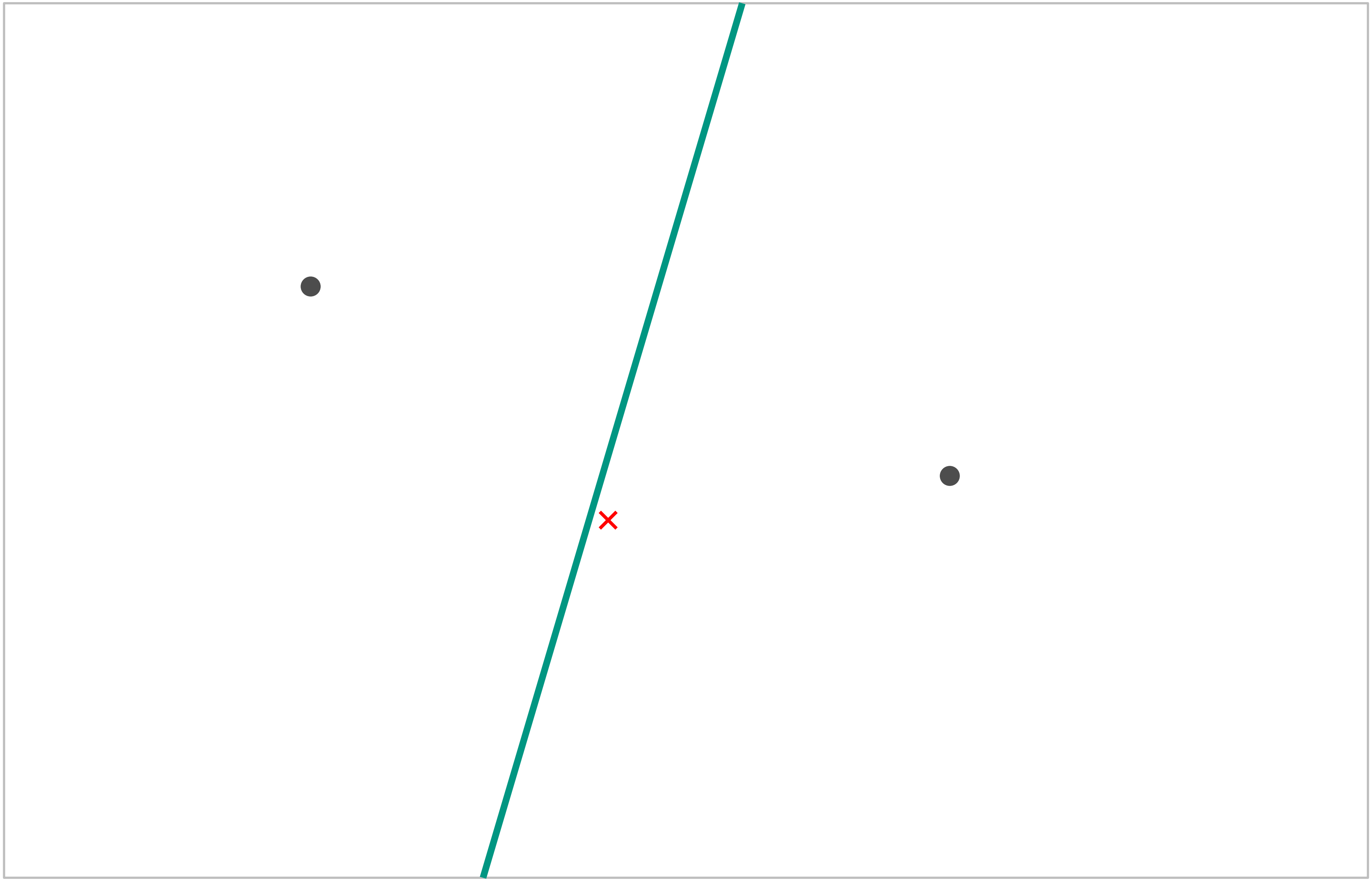


The Post Office Problem

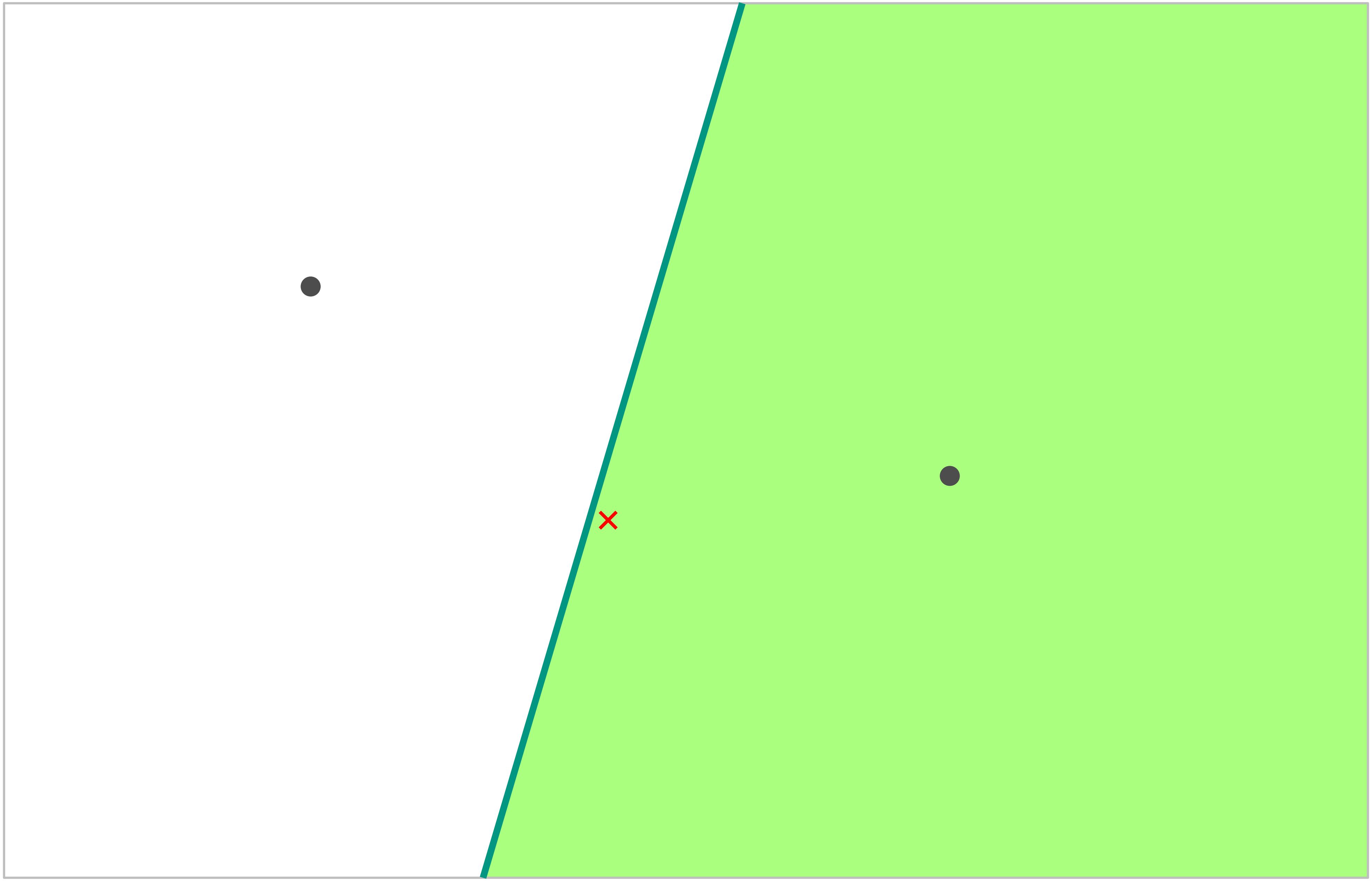


The Post Office Problem

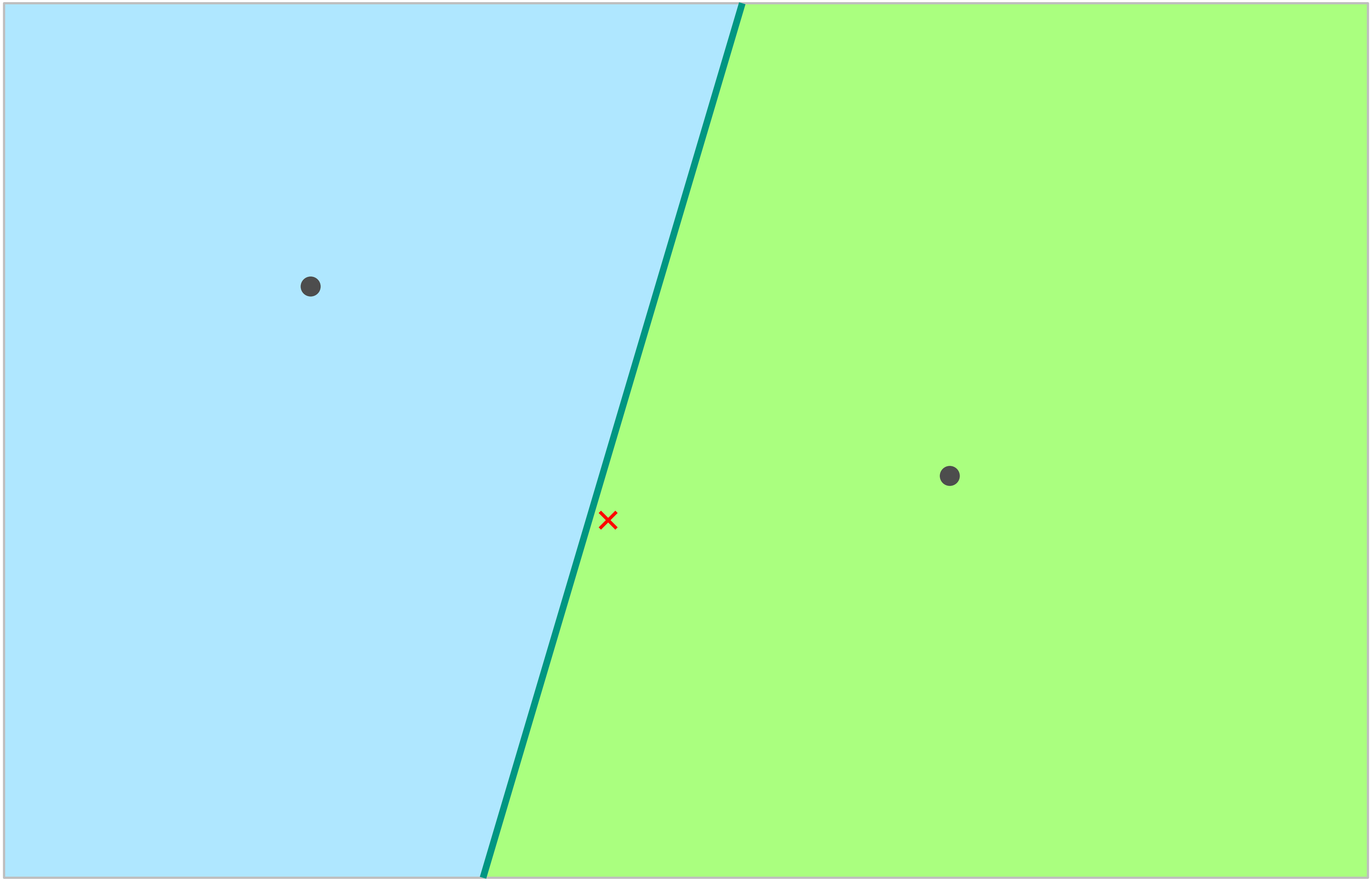
The Post Office Problem



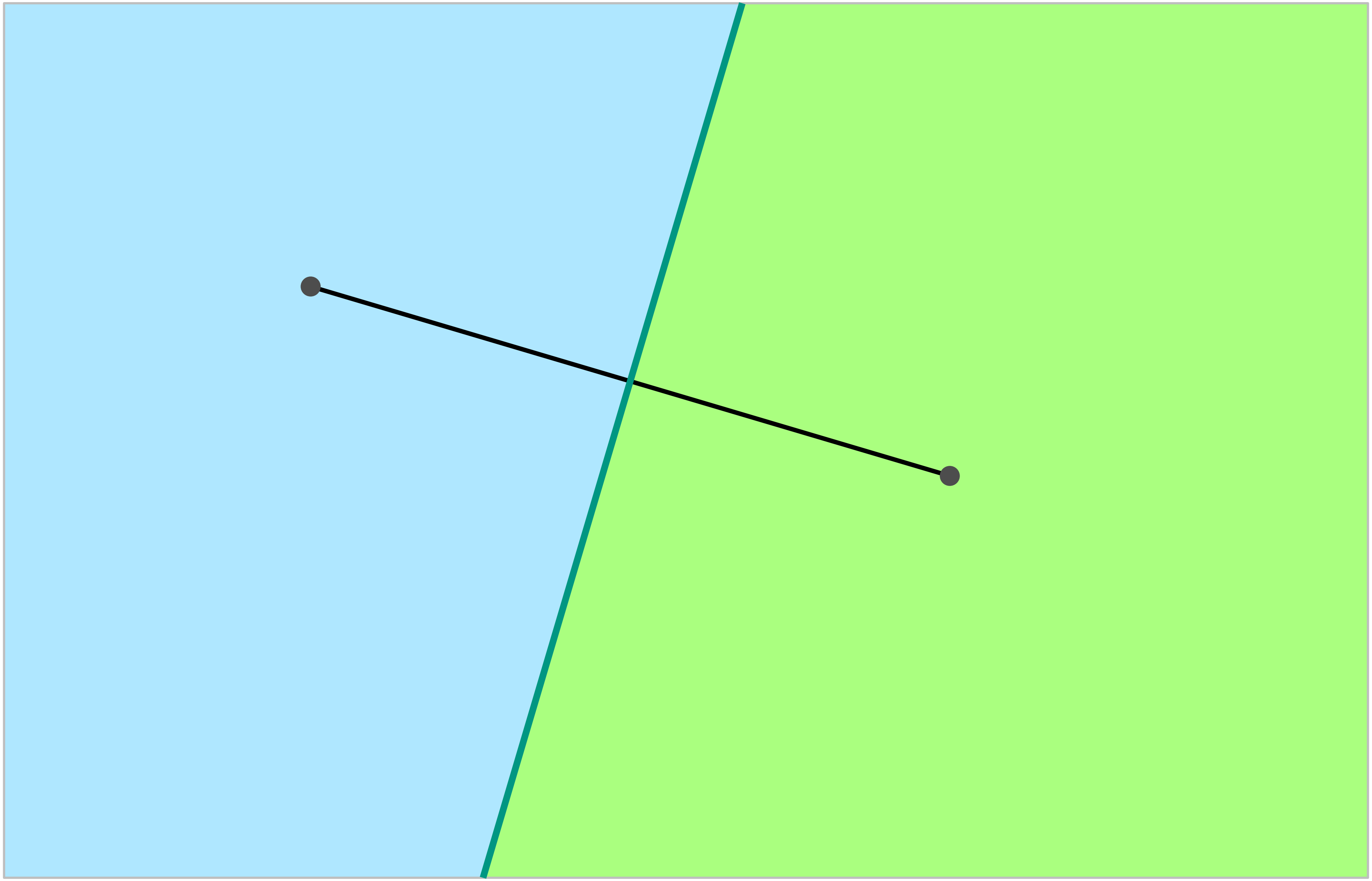
The Post Office Problem



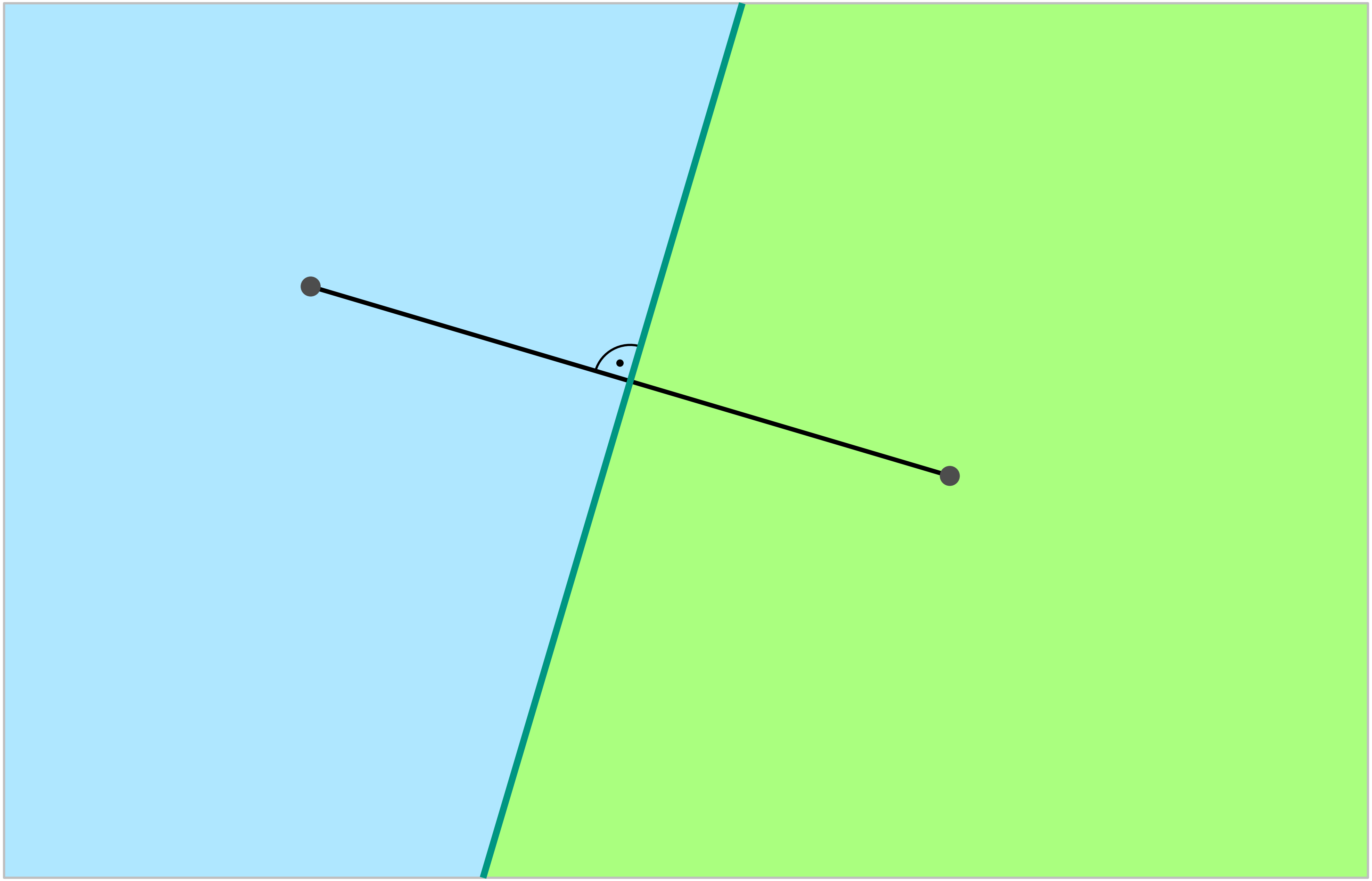
The Post Office Problem



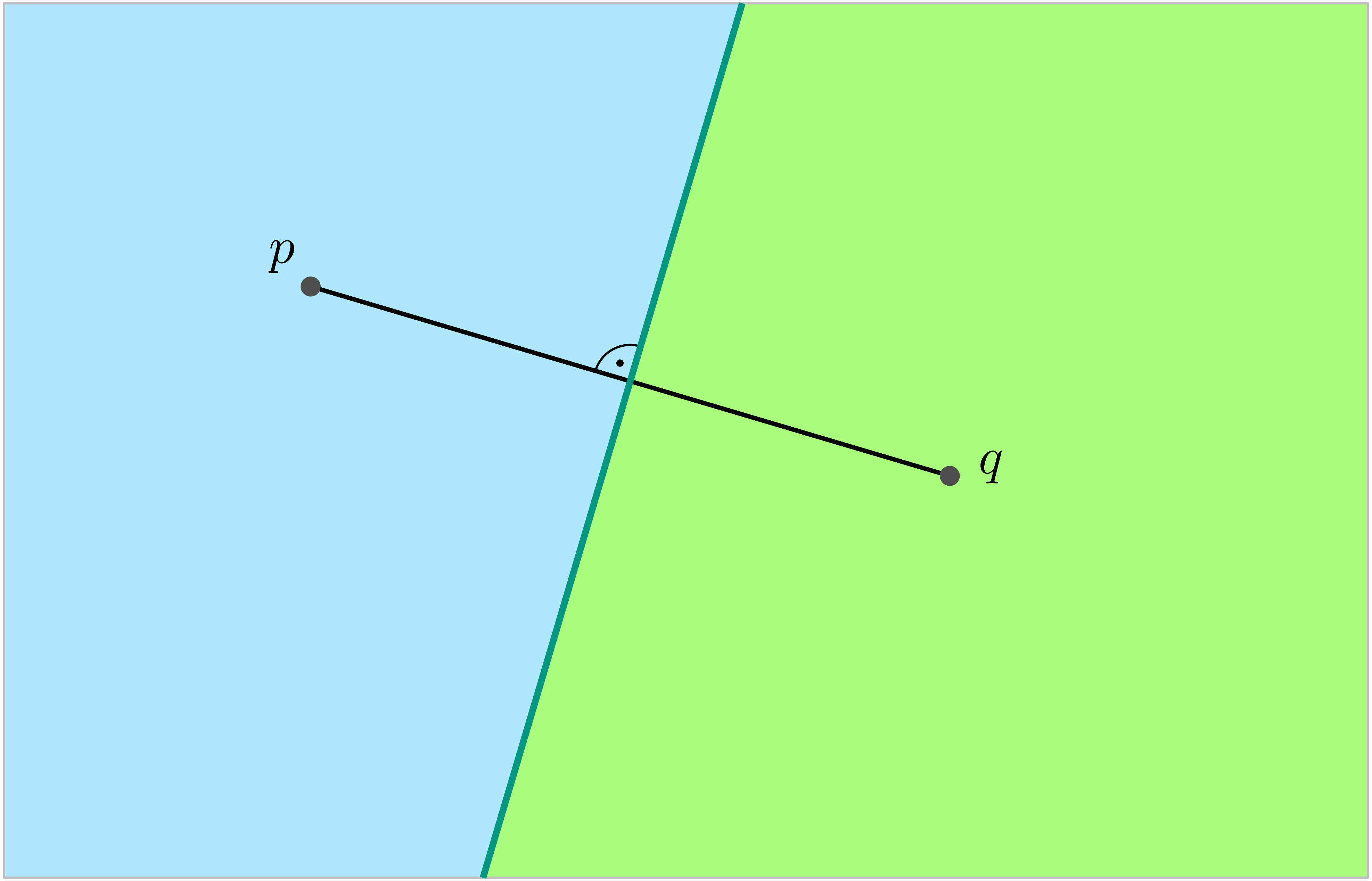
The Post Office Problem



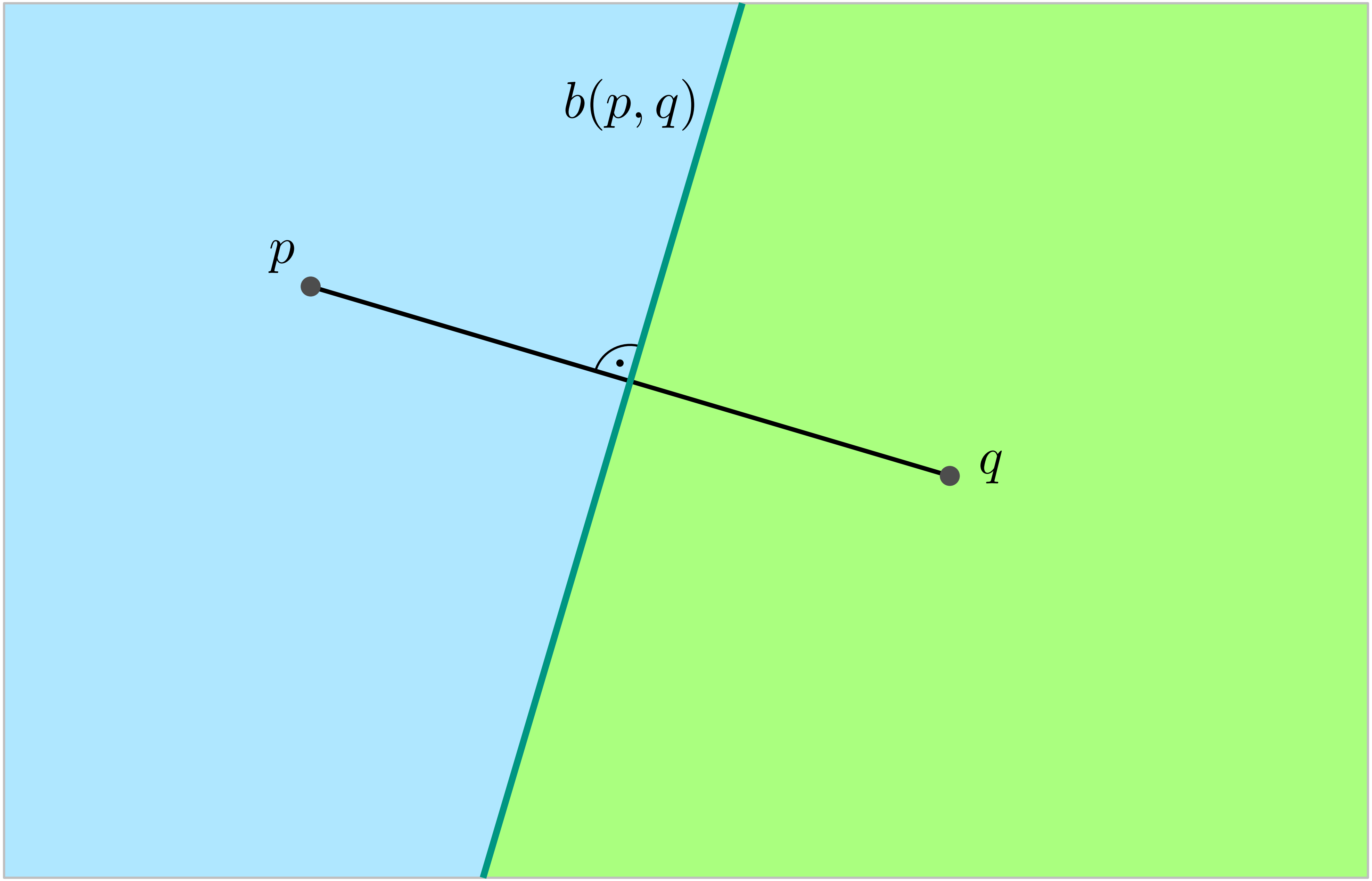
The Post Office Problem



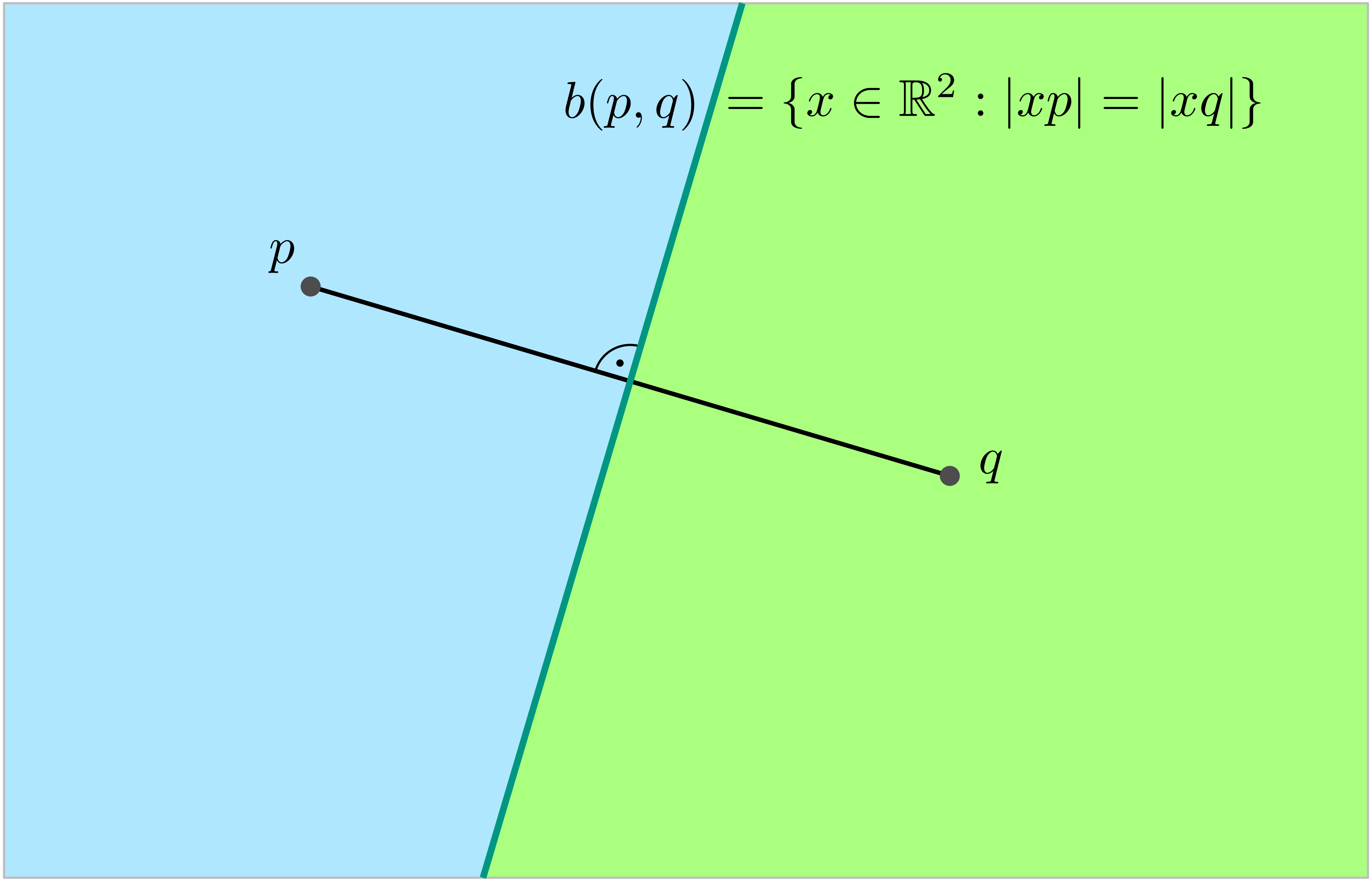
The Post Office Problem



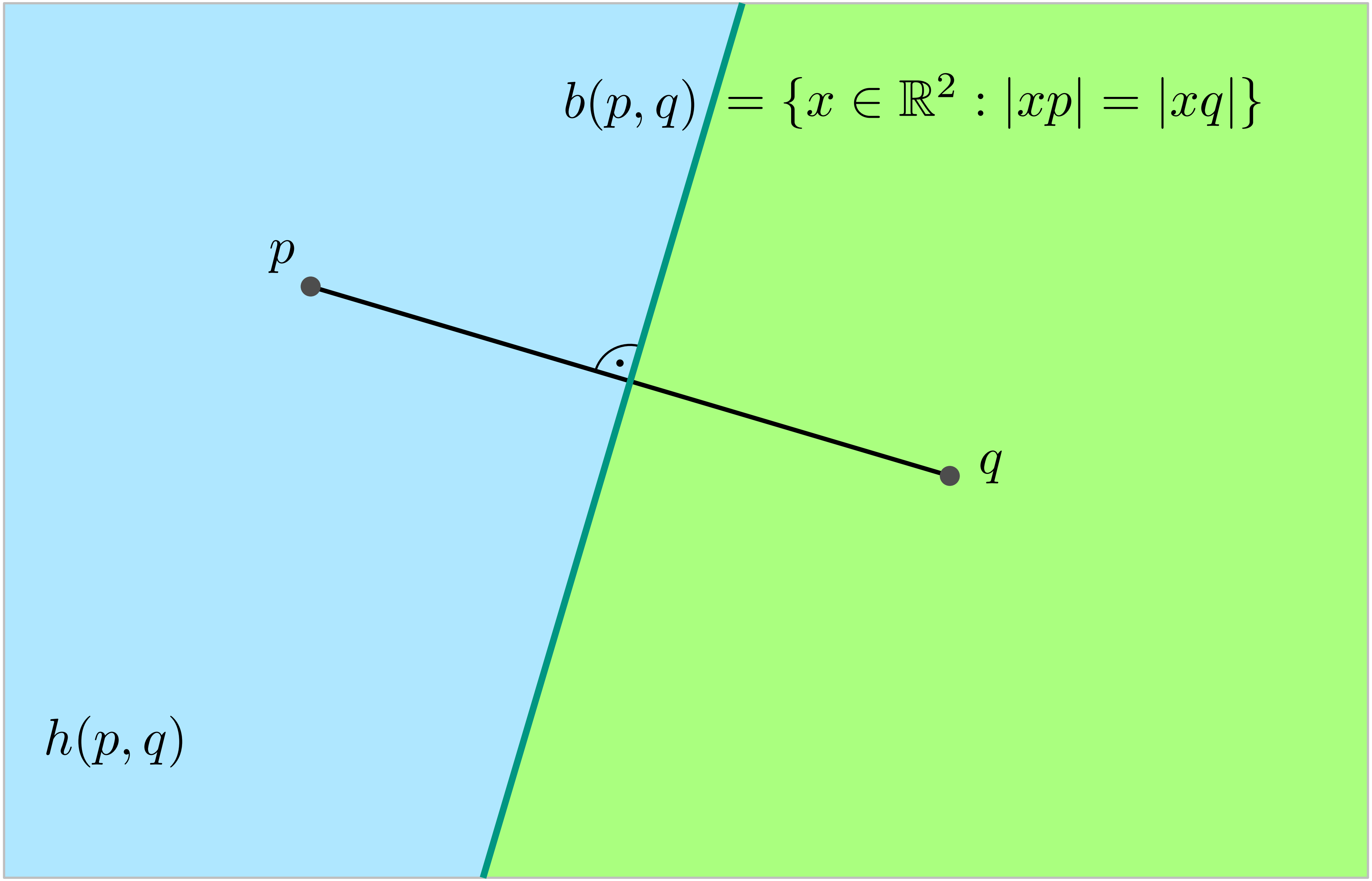
The Post Office Problem



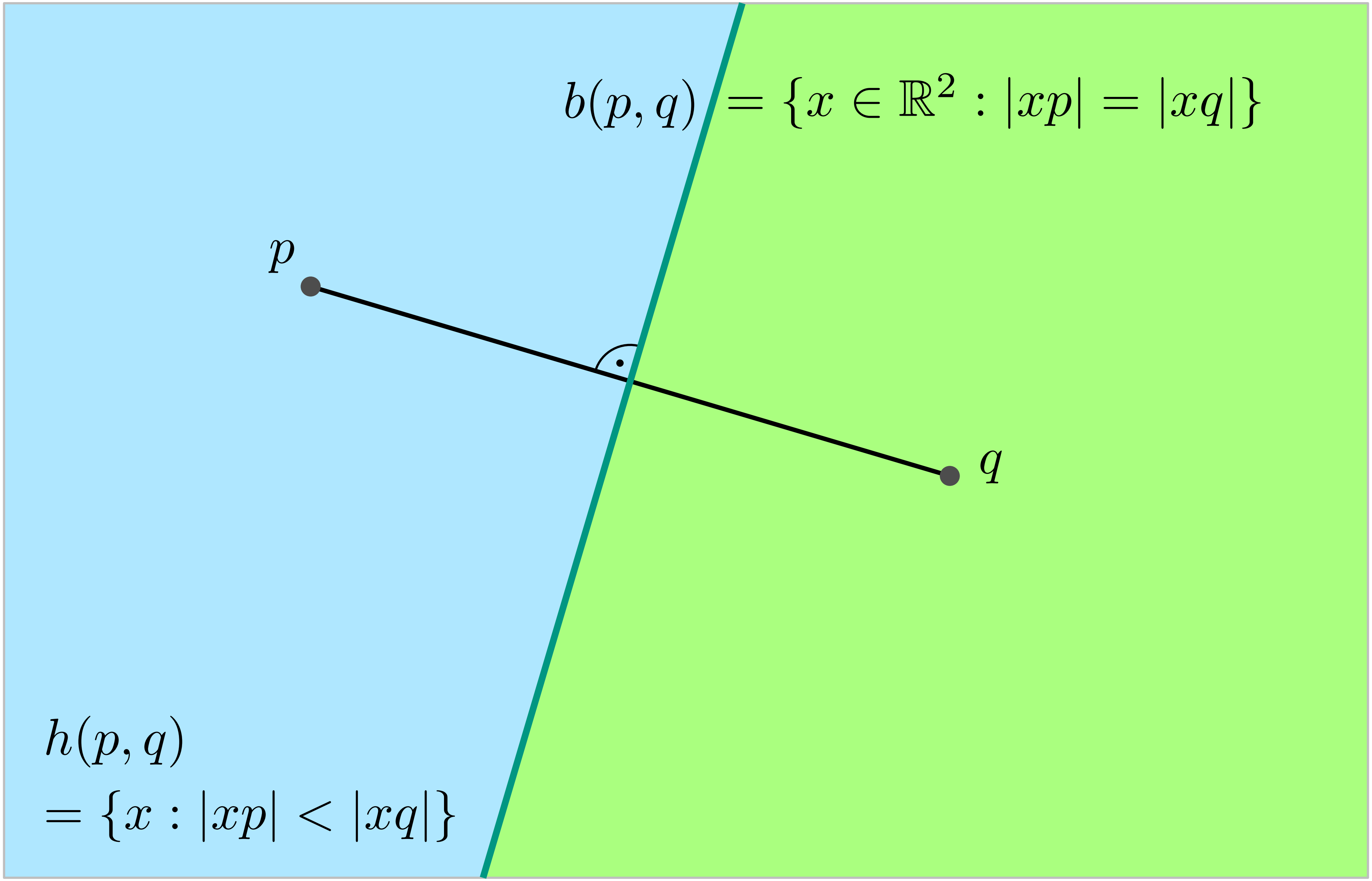
The Post Office Problem



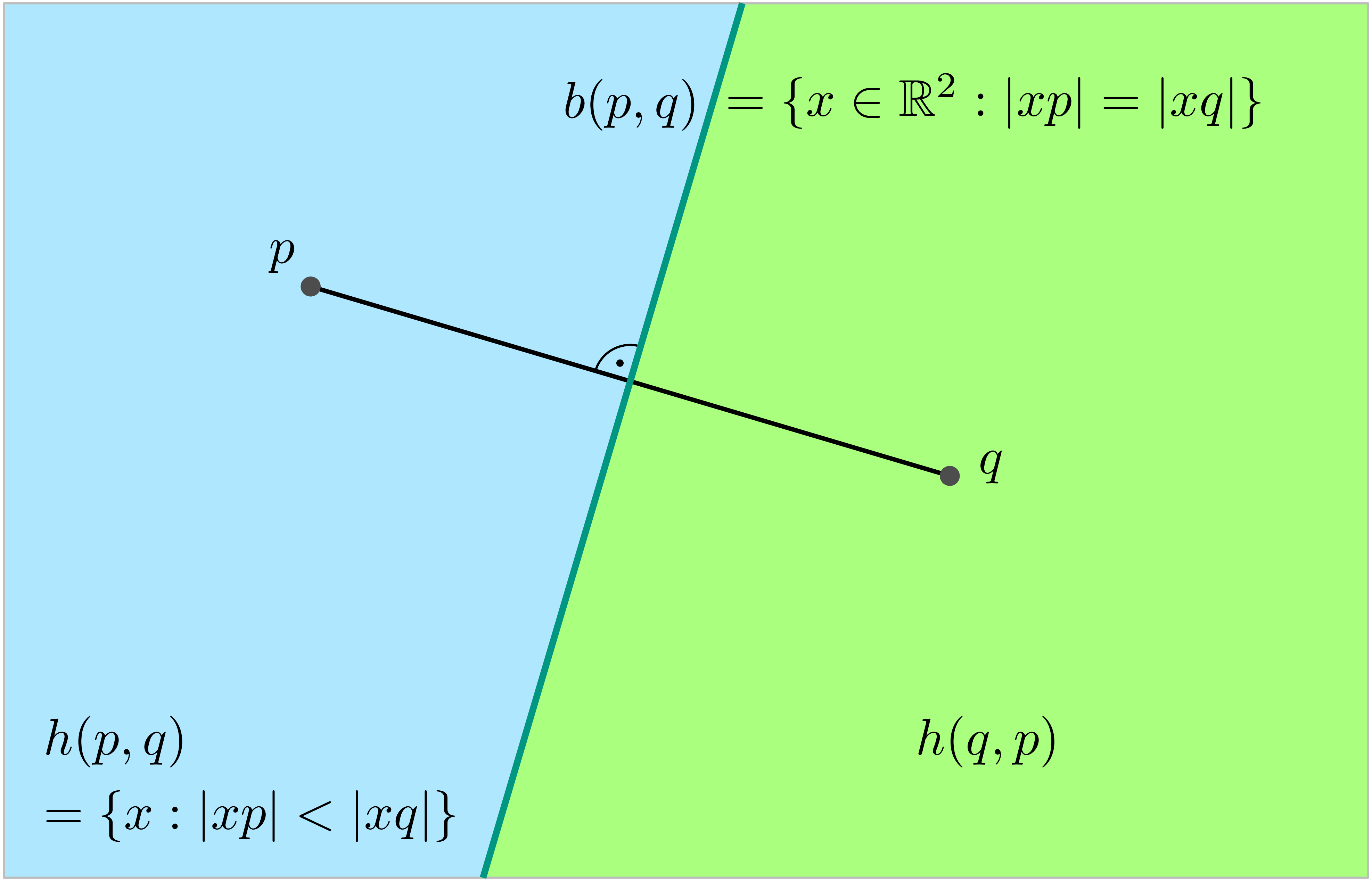
The Post Office Problem



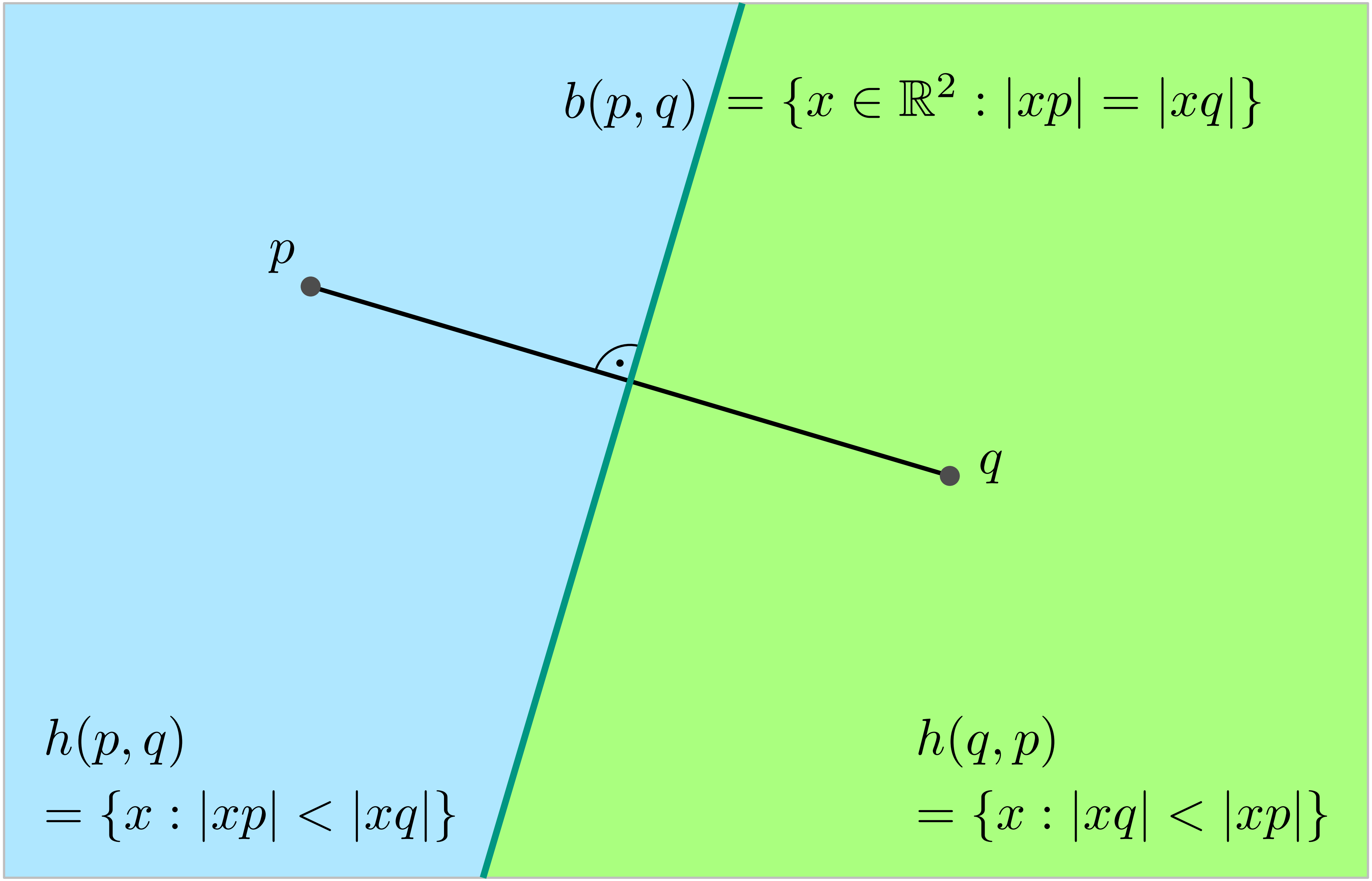
The Post Office Problem



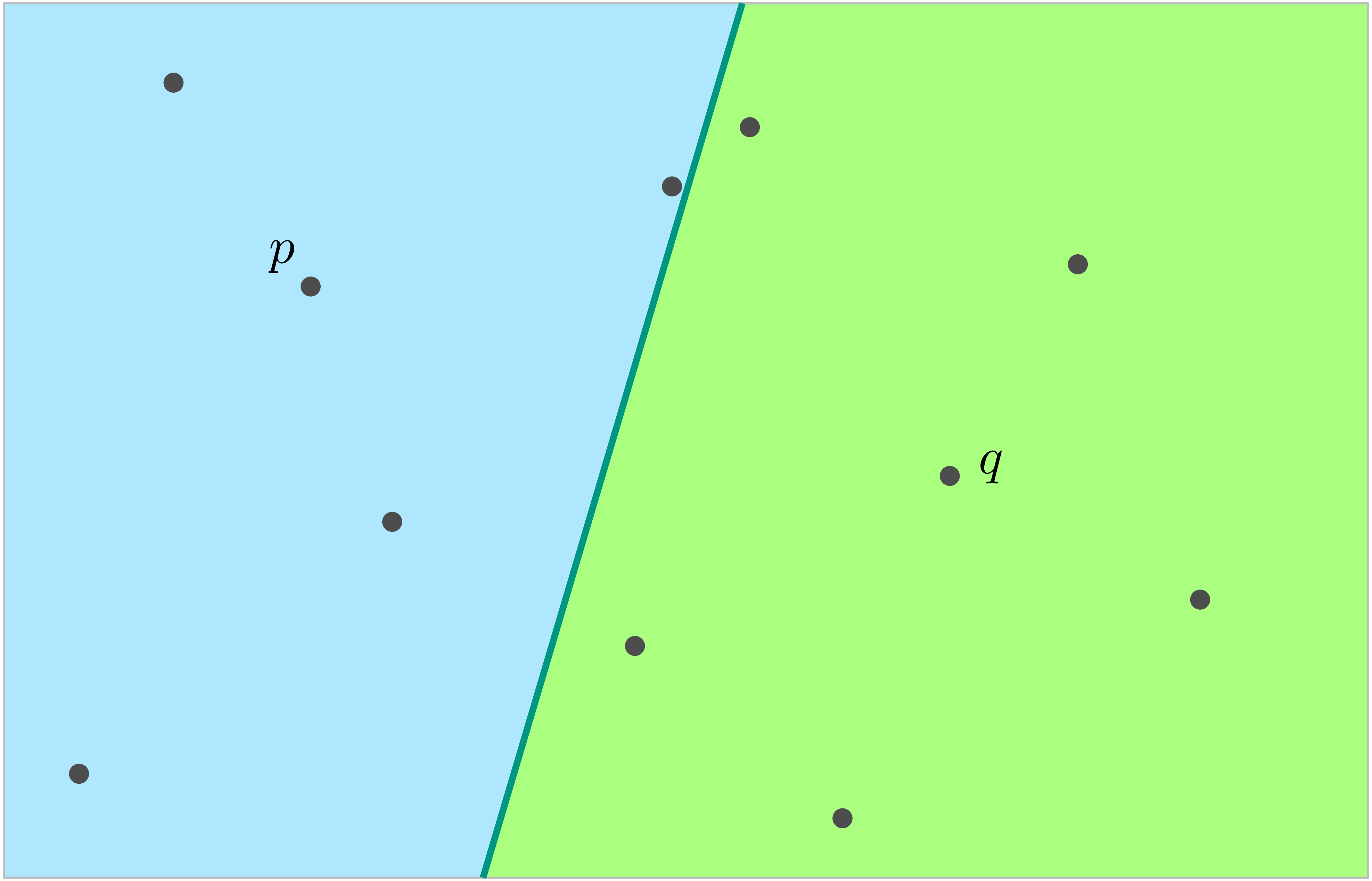
The Post Office Problem



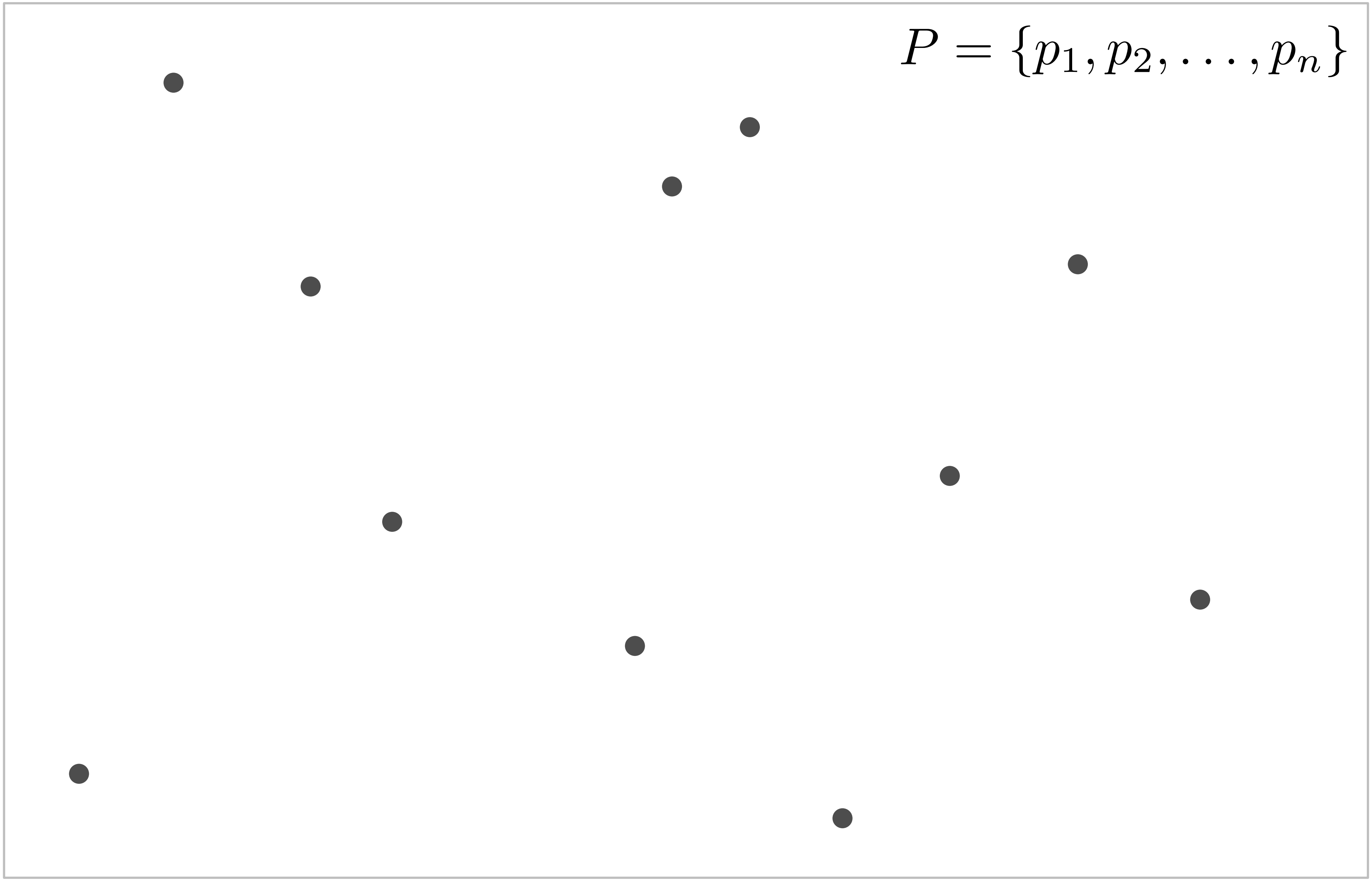
The Post Office Problem



The Post Office Problem

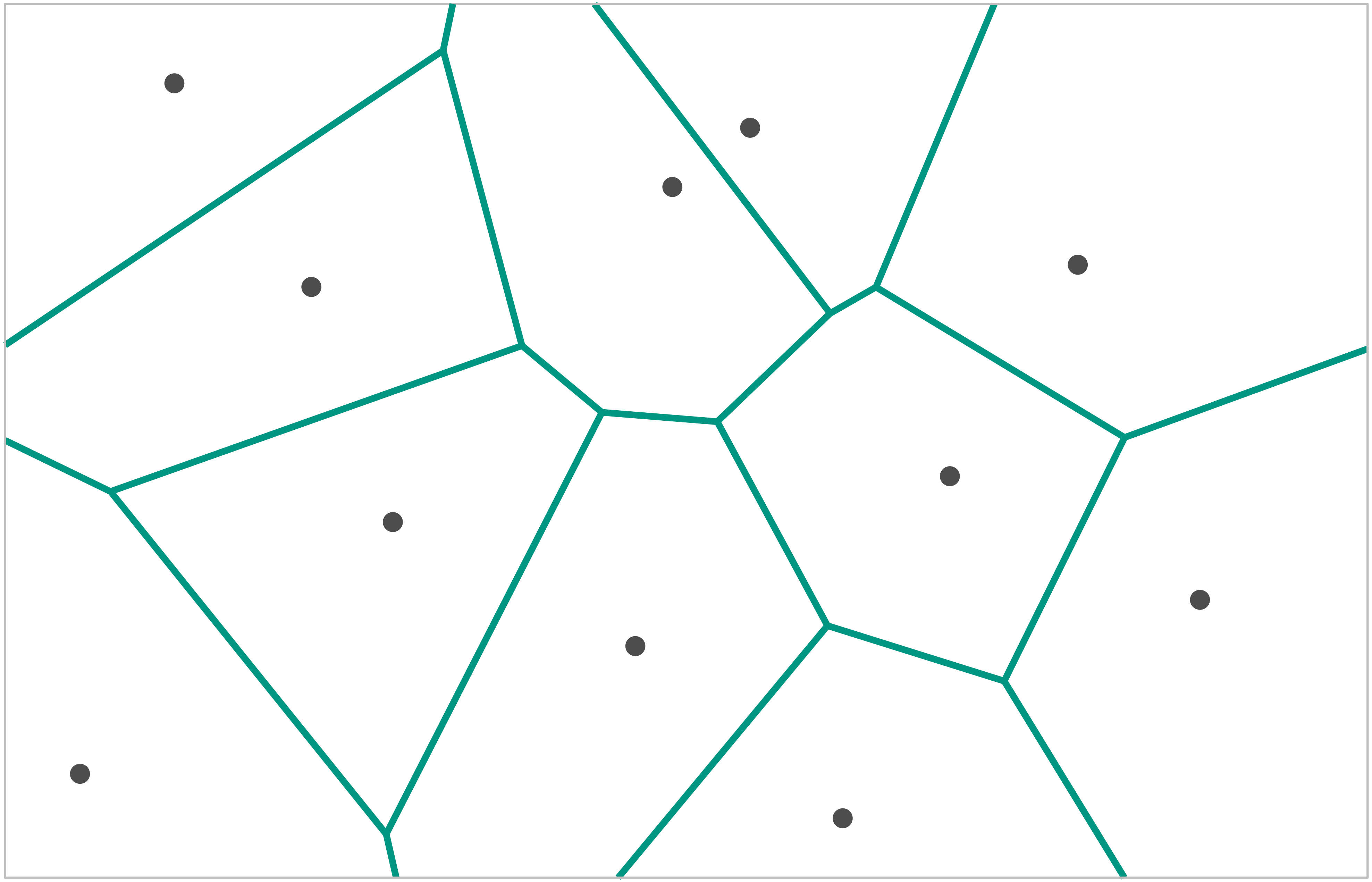


The Post Office Problem

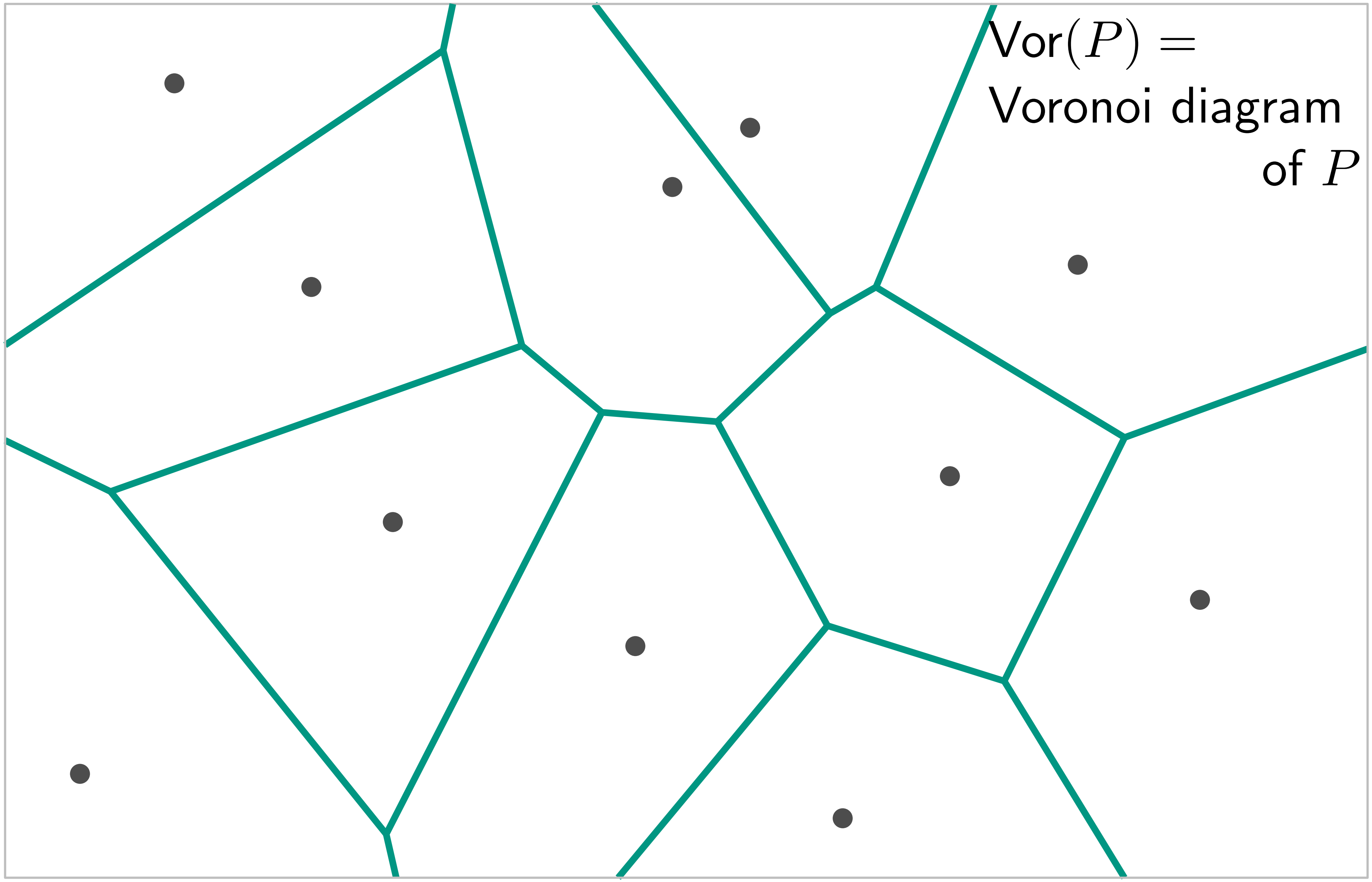


$$P = \{p_1, p_2, \ldots, p_n\}$$

The Post Office Problem

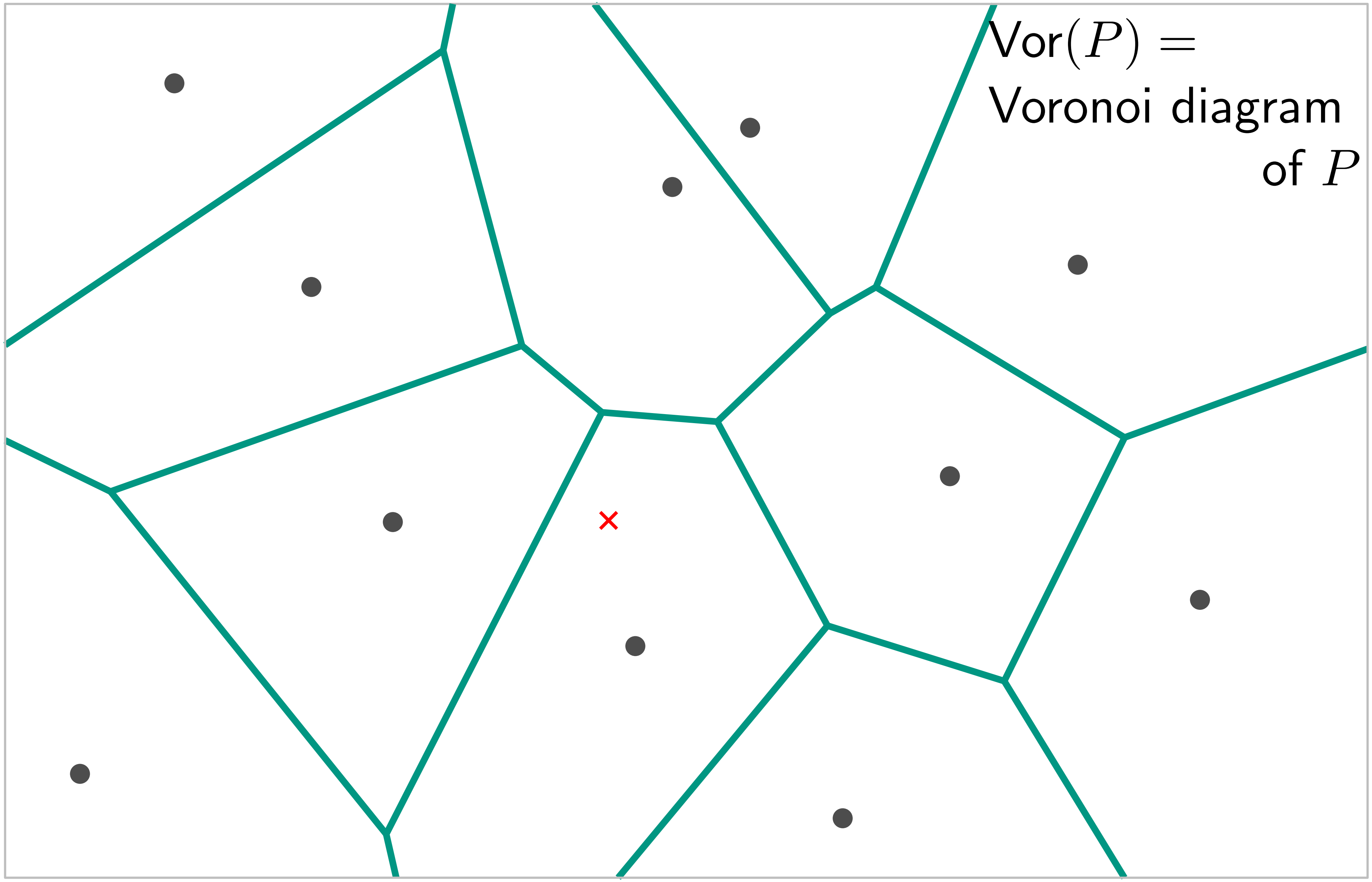


The Post Office Problem

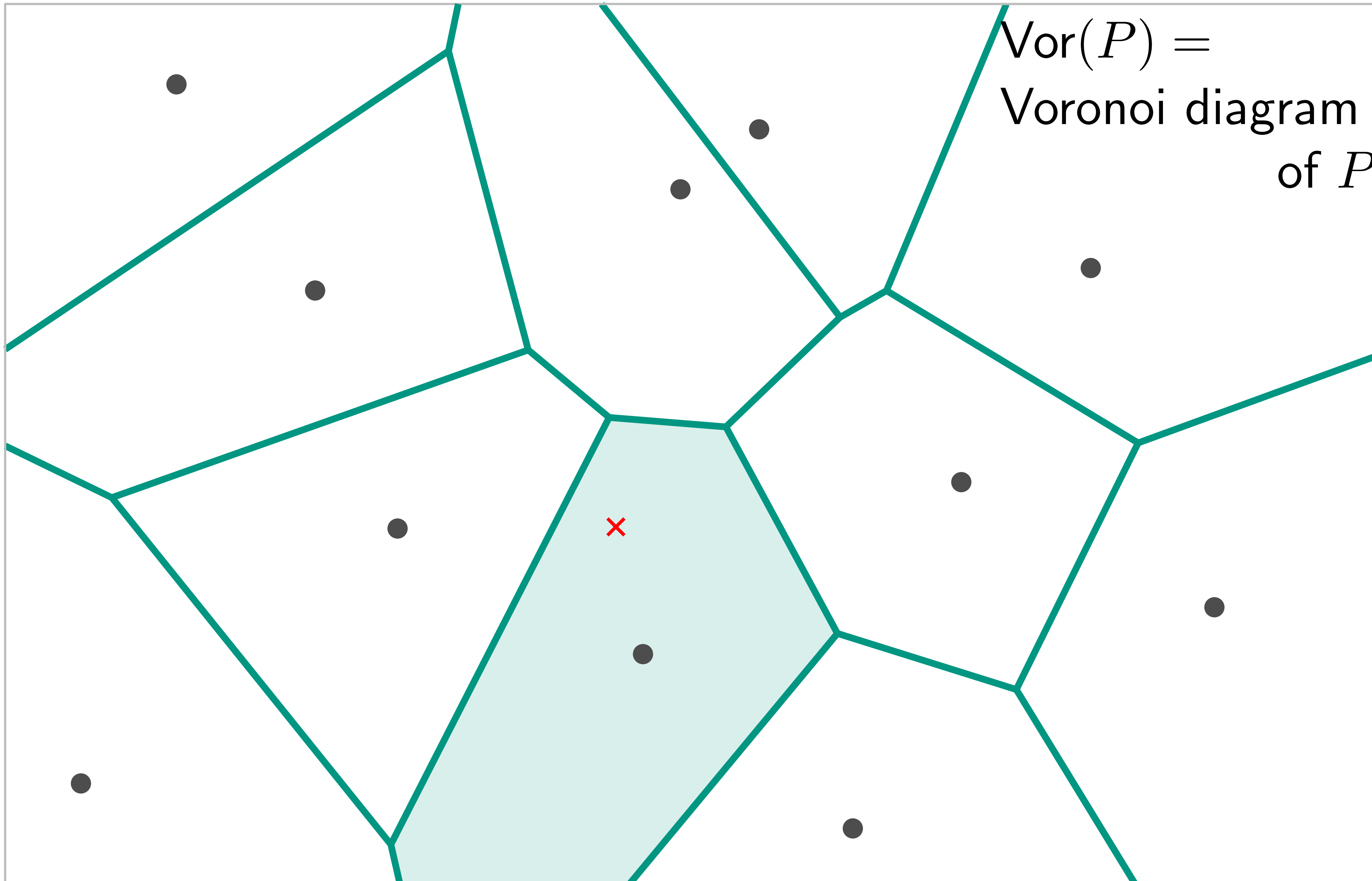


Vor(P) =
Voronoi diagram
of P

The Post Office Problem

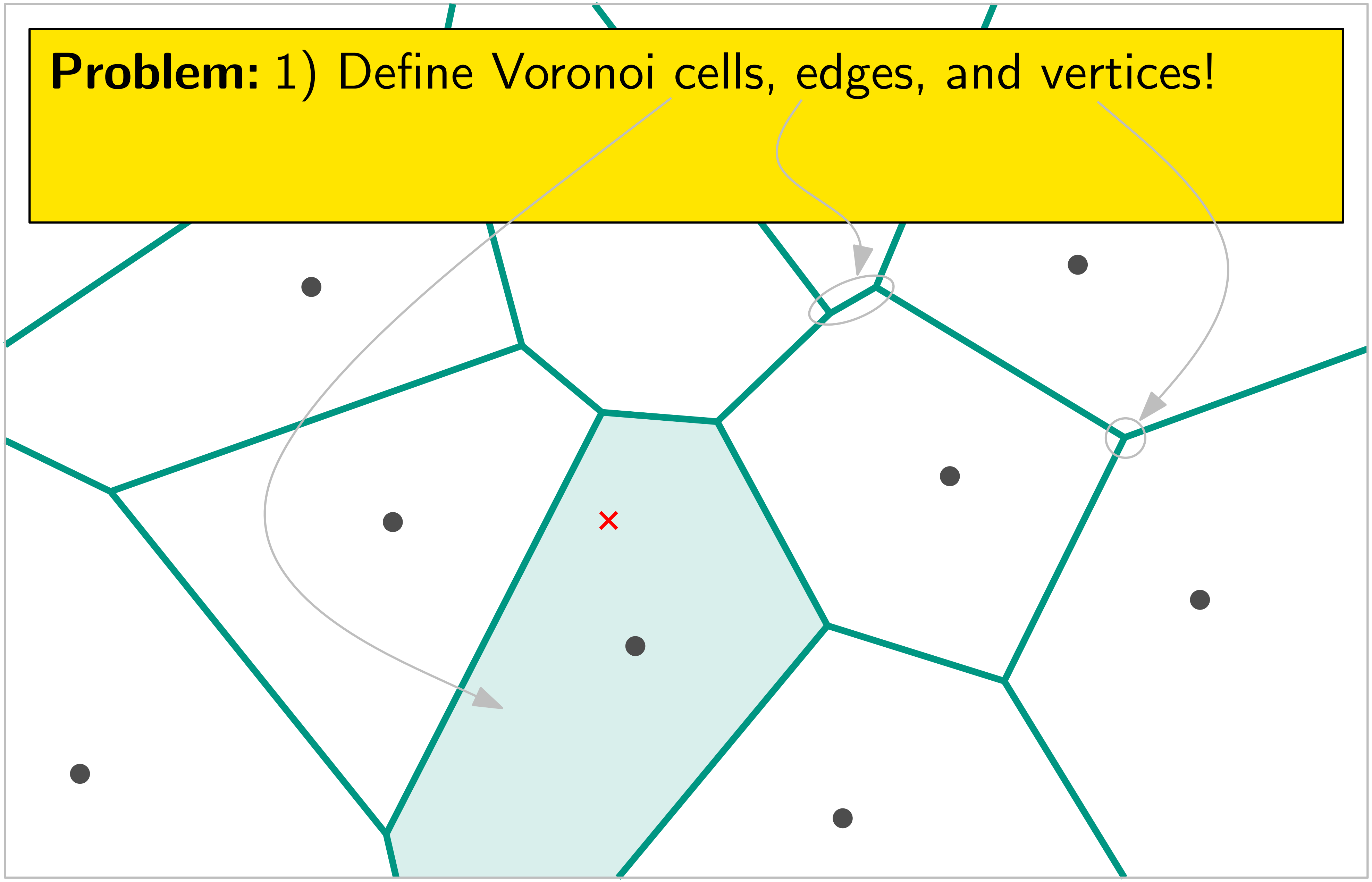


The Post Office Problem



The Post Office Problem

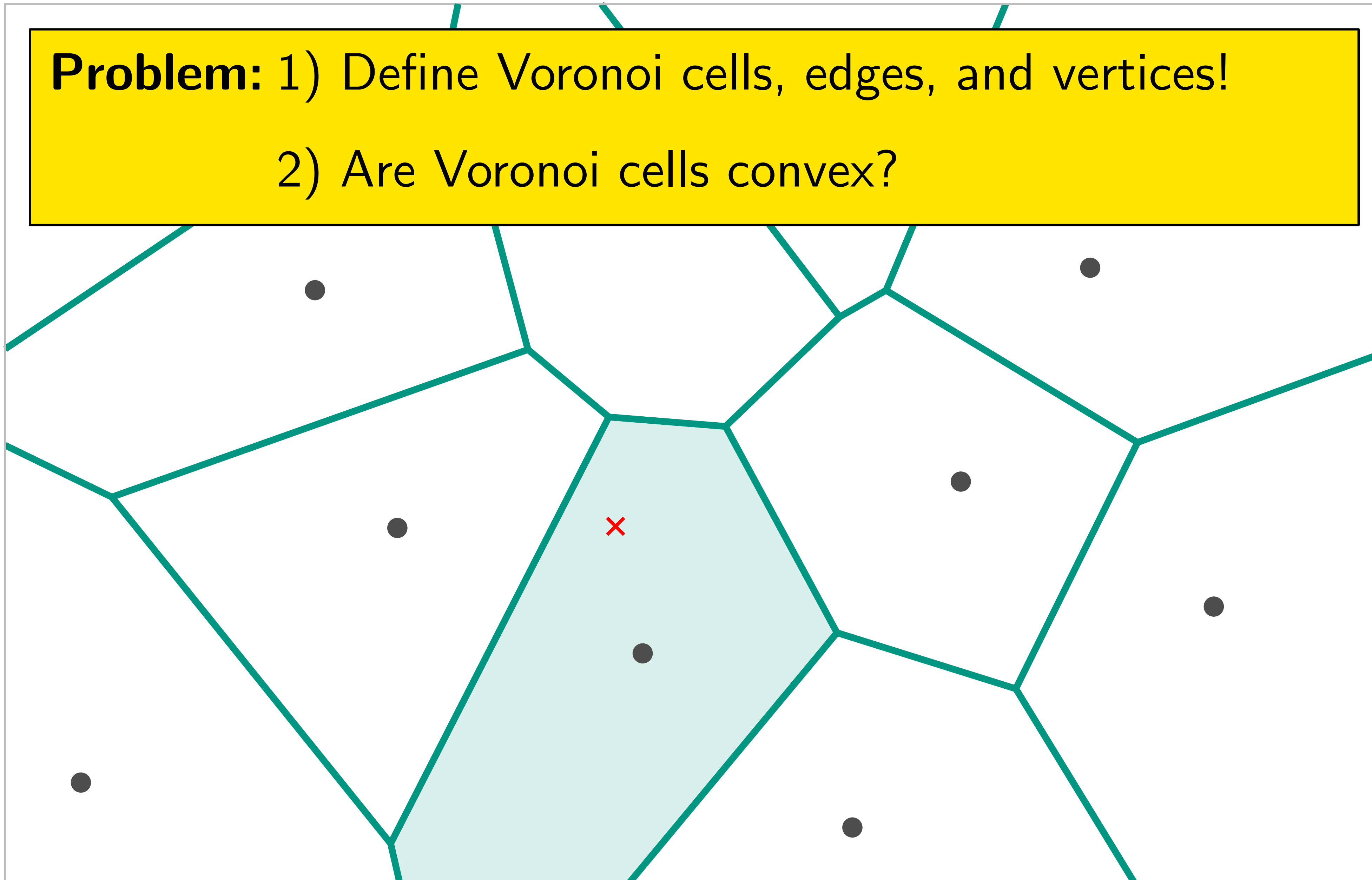
Problem: 1) Define Voronoi cells, edges, and vertices!



The Post Office Problem

Problem: 1) Define Voronoi cells, edges, and vertices!

2) Are Voronoi cells convex?

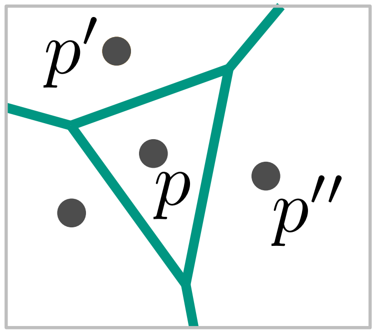


The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

The Voronoi Diagram

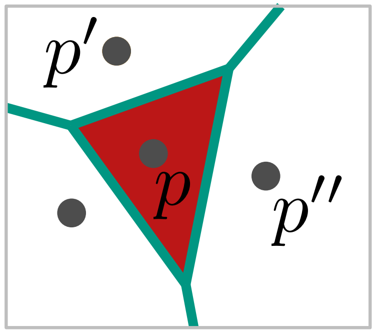
Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram  $\mathrm{Vor}(P)$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram

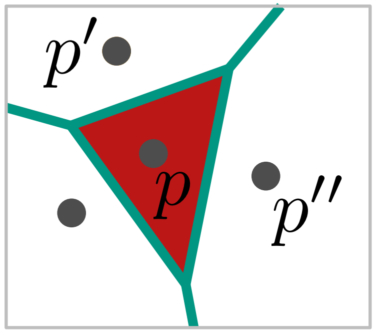


Vor(P)

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



$\text{Vor}(P)$

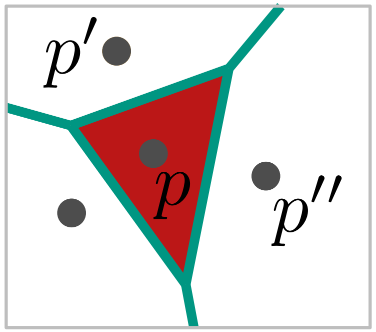
- Voronoi cell

$\mathcal{V}(\{p\}) =$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



$\text{Vor}(P)$

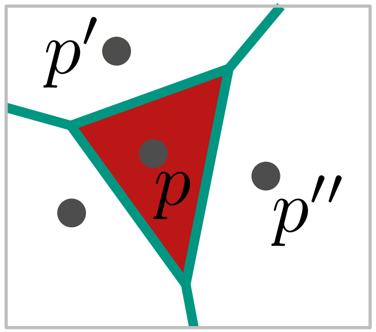
- Voronoi cell

$\mathcal{V}(\{p\}) = \mathcal{V}(p) =$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



$\text{Vor}(P)$

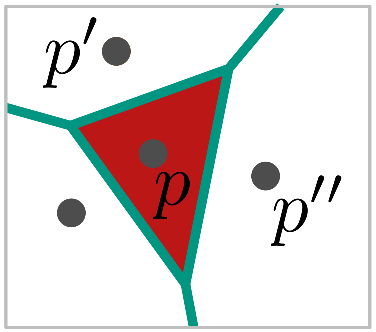
- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \left\{ x \in \mathbb{R}^2 : |xp| < |xq| \, \forall q \in P \setminus \{p\} \right\}$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



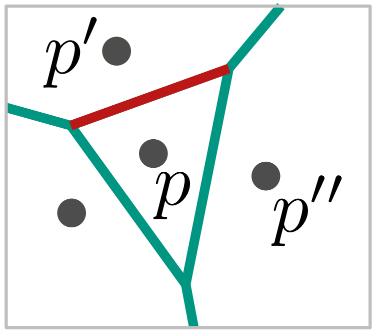
$\mathrm{Vor}(P)$

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \, \forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

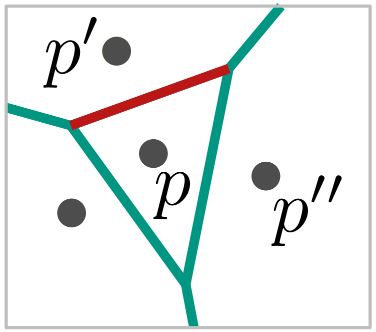
Voronoi Diagram  Vor(P)

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \, \forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram  Vor(P)

- Voronoi cell

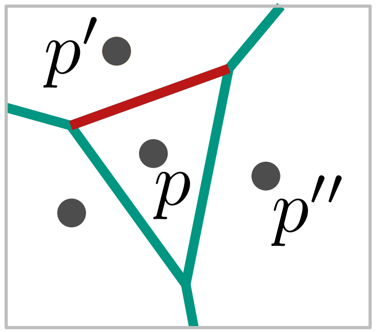
$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \, \forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

- Voronoi edges

$$\mathcal{V}(\{p, p'\}) =$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram  $\text{Vor}(P)$

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \, \forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

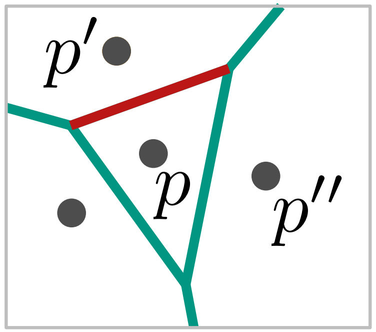
- Voronoi edges

$$\mathcal{V}(\{p, p'\}) = \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \;\; \forall q \neq p, p'\}$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



$\mathrm{Vor}(P)$

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \,\forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

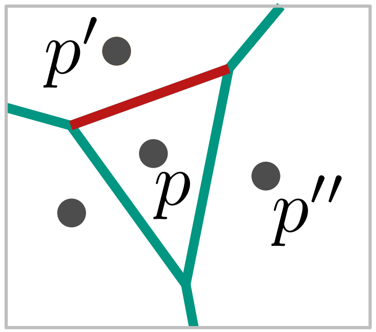
- Voronoi edges

$$\mathcal{V}(\{p, p'\}) = \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \quad \forall q \neq p, p'\}$$
$$= \partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



$\text{Vor}(P)$

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \, \forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

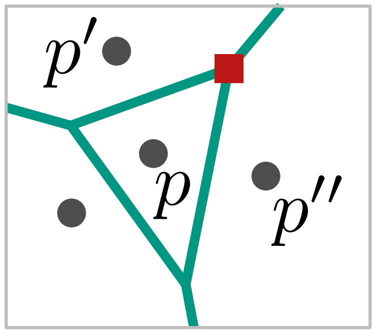
- Voronoi edges

$$\mathcal{V}(\{p, p'\}) = \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \;\; \forall q \neq p, p'\}$$
$$= \text{rel-int}\big(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')\big), \text{ d.h. without endpoints}$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



$\text{Vor}(P)$

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \left\{ {\color{red}x} \in \mathbb{R}^2 : |{\color{red}x}p| < |{\color{red}x}q| \, \forall q \in P \setminus \{p\} \right\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

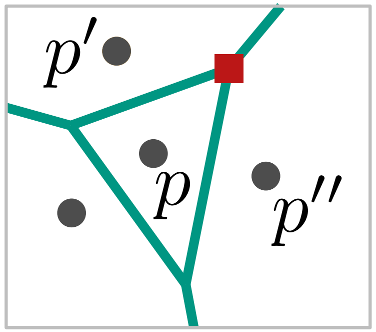
- Voronoi edges

$$\mathcal{V}(\{p, p'\}) = \{{\color{red}x} : |{\color{red}x}p| = |{\color{red}x}p'| \text{ and } |{\color{red}x}p| < |{\color{red}x}q| \;\; \forall q \neq p, p'\}$$
$$= \text{rel-int}\big(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')\big), \text{ d.h. without endpoints}$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



$\mathrm{Vor}(P)$

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \, \forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

- Voronoi edges

$$\mathcal{V}(\{p, p'\}) = \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \;\; \forall q \neq p, p'\}$$
$$= \text{rel-int}\big(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')\big), \text{ d.h. without endpoints}$$

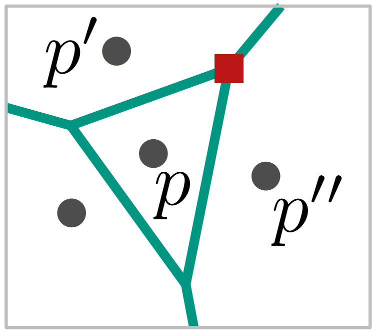
- Voronoi vertices

$$\mathcal{V}(\{p, p', p''\})$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



Vor(P)

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \,\forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

- Voronoi edges

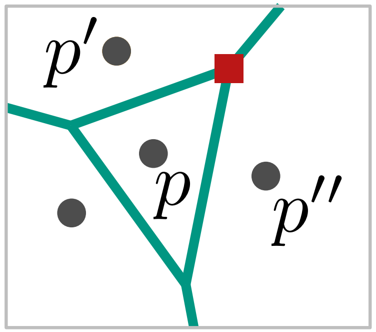
$$\mathcal{V}(\{p, p'\}) = \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \;\; \forall q \neq p, p'\}$$
$$= \text{rel-int}\big(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')\big), \text{ d.h. without endpoints}$$

- Voronoi vertices

$$\mathcal{V}(\{p, p', p''\}) = \partial\mathcal{V}(p) \cap \partial\mathcal{V}(p') \cap \partial\mathcal{V}(p'')$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram  Vor(P)

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \, \forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p, q)$$

- Voronoi edges

$$\mathcal{V}(\{p, p'\}) = \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \quad \forall q \neq p, p'\}$$
$$= \text{rel-int}\big(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')\big)\text{, d.h. without endpoints}$$

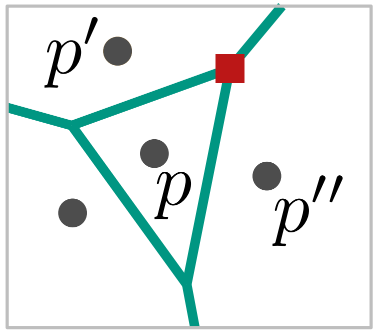
- Voronoi vertices

$$\mathcal{V}(\{p, p', p''\}) = \partial\mathcal{V}(p) \cap \partial\mathcal{V}(p') \cap \partial\mathcal{V}(p'')$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



Vor(P) → subdivision

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \, \forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p,q)$$

- Voronoi edges

$$\mathcal{V}(\{p,p'\}) = \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \;\; \forall q \neq p, p'\}$$
$$= \text{rel-int}\big(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')\big), \text{ d.h. without endpoints}$$

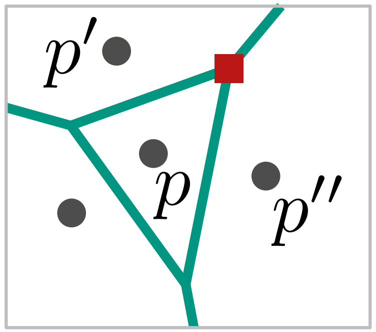
- Voronoi vertices

$$\mathcal{V}(\{p,p',p''\}) = \partial\mathcal{V}(p) \cap \partial\mathcal{V}(p') \cap \partial\mathcal{V}(p'')$$

The Voronoi Diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

Voronoi Diagram



Vor(P) → subdivision
→ geometric graph

- Voronoi cell

$$\mathcal{V}(\{p\}) = \mathcal{V}(p) = \{x \in \mathbb{R}^2 : |xp| < |xq| \; \forall q \in P \setminus \{p\}\}$$
$$= \bigcap_{q \neq p} h(p,q)$$

- Voronoi edges

$$\mathcal{V}(\{p,p'\}) = \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \;\; \forall q \neq p, p'\}$$
$$= \text{rel-int}\big(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')\big), \text{ d.h. without endpoints}$$

- Voronoi vertices

$$\mathcal{V}(\{p,p',p''\}) = \partial\mathcal{V}(p) \cap \partial\mathcal{V}(p') \cap \partial\mathcal{V}(p'')$$

Properties

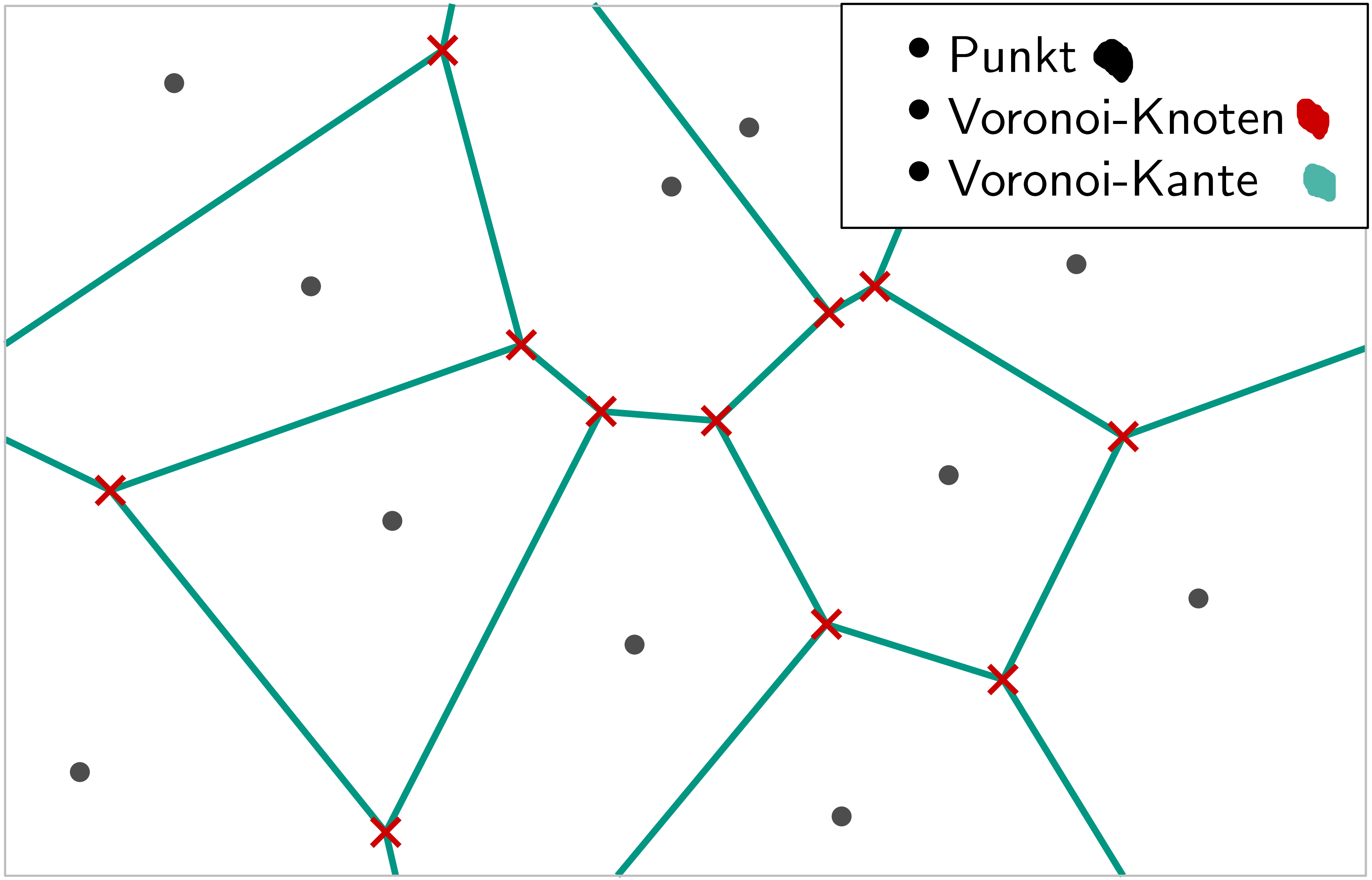
Theorem 1: Let $P \subset \mathbb{R}^2$ be a set of n points. If all points are collinear, then $\mathrm{Vor}(P)$ consists of $n-1$ parallel lines. Otherwise $\mathrm{Vor}(P)$ is connected and its edges are either segments or half lines.

Find a set P so that a cell in $\mathrm{Vor}(P)$ has linear complexity.

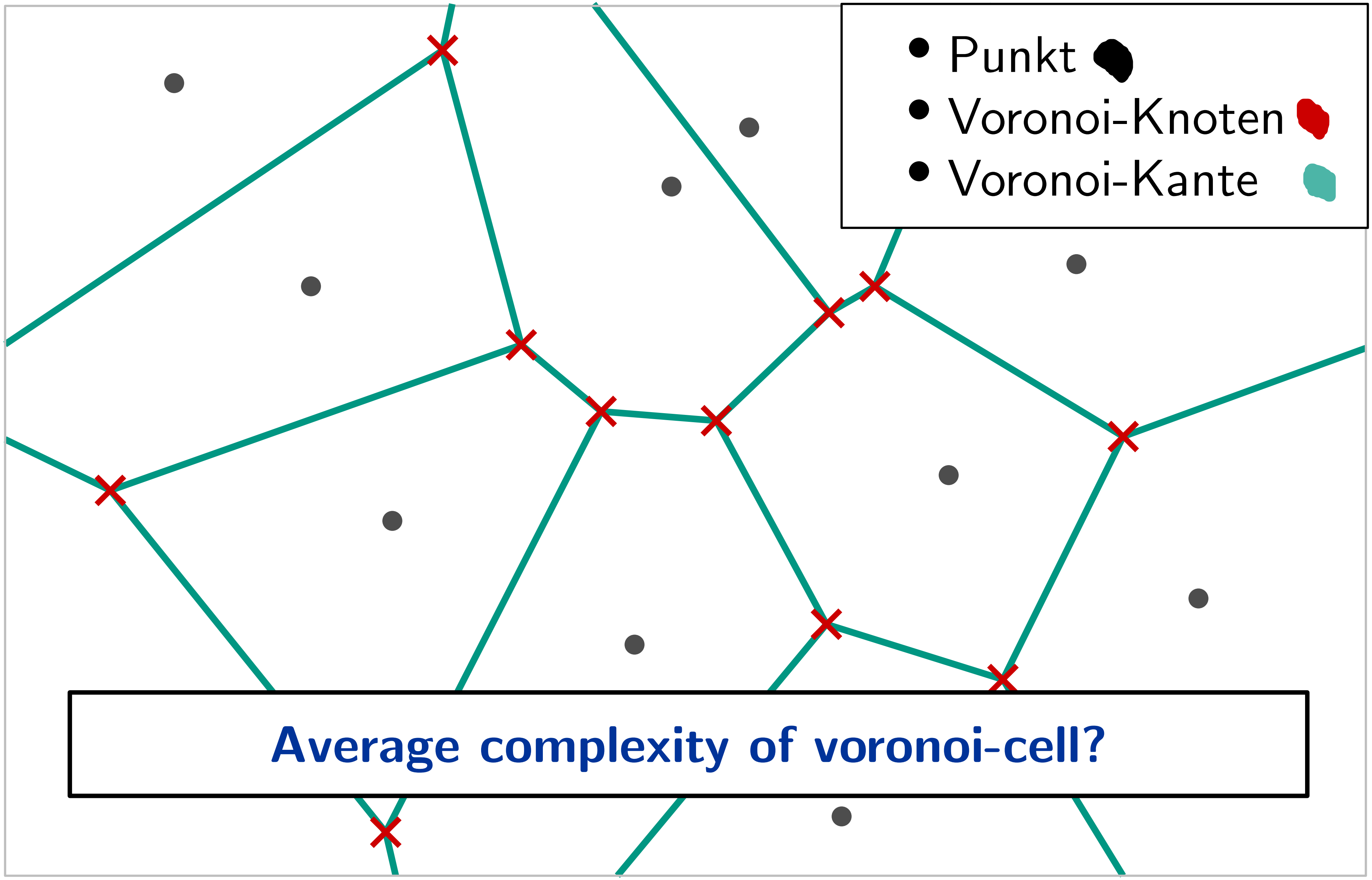
Can this happen with (almost) all cells?

Theorem 2: Let $P \subset \mathbb{R}^2$ be a set on n points. $\mathrm{Vor}(P)$ has at most $2n-5$ nodes and $3n-6$ edges.

Exercise 1



Exercise 1



Average complexity of voronoi-cell?

In the Direction of the Sweep Line

Obviously it's the intersection of $\mathrm{Vor}(P)$ and sweep line ℓ at the current time point is not known yet.

In the Direction of the Sweep Line

Obviously it's the intersection of $\mathrm{Vor}(P)$ and sweep line ℓ at the current time point is not known yet.

Instead, we view the part above ℓ as already fixed!

In the Direction of the Sweep Line

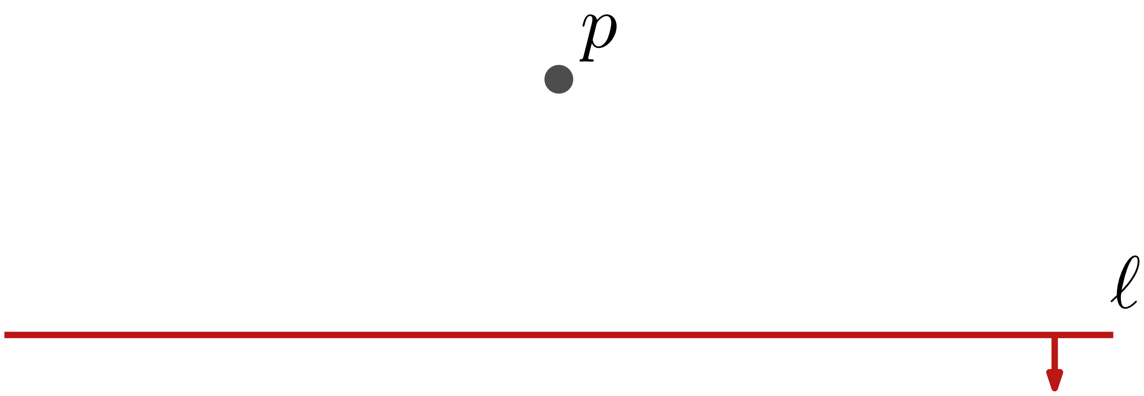
Obviously it's the intersection of $\mathrm{Vor}(P)$ and sweep line ℓ at the current time point is not known yet.

Instead, we view the part above ℓ as already fixed!

In the Direction of the Sweep Line

Obviously it's the intersection of $\mathrm{Vor}(P)$ and sweep line ℓ at the current time point is not known yet.

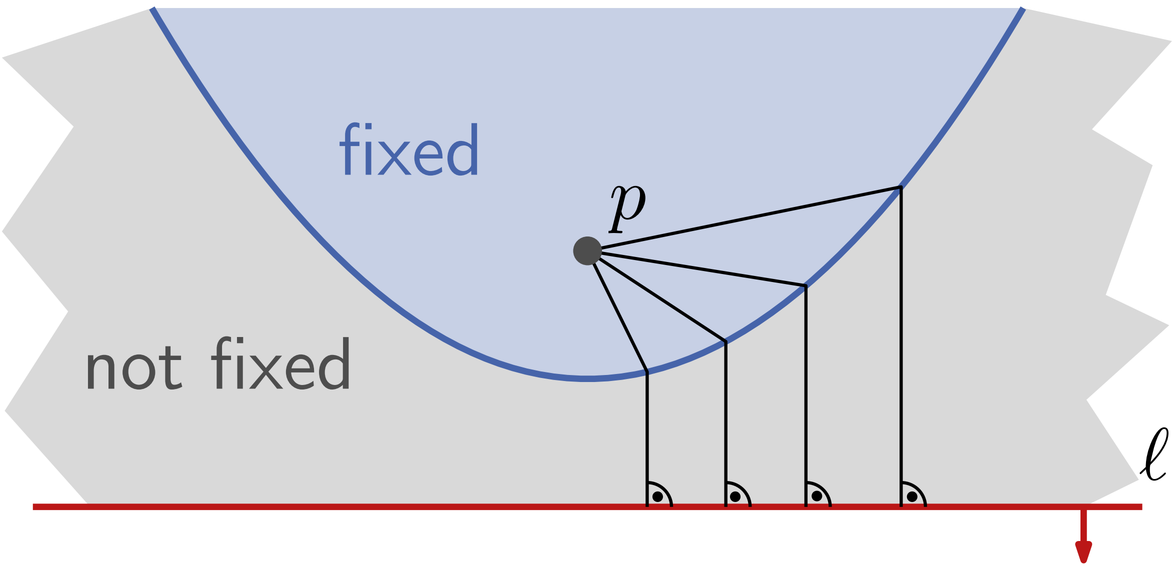
Instead, we view the part above ℓ as already fixed!



In the Direction of the Sweep Line

Obviously it's the intersection of $\mathrm{Vor}(P)$ and sweep line ℓ at the current time point is not known yet.

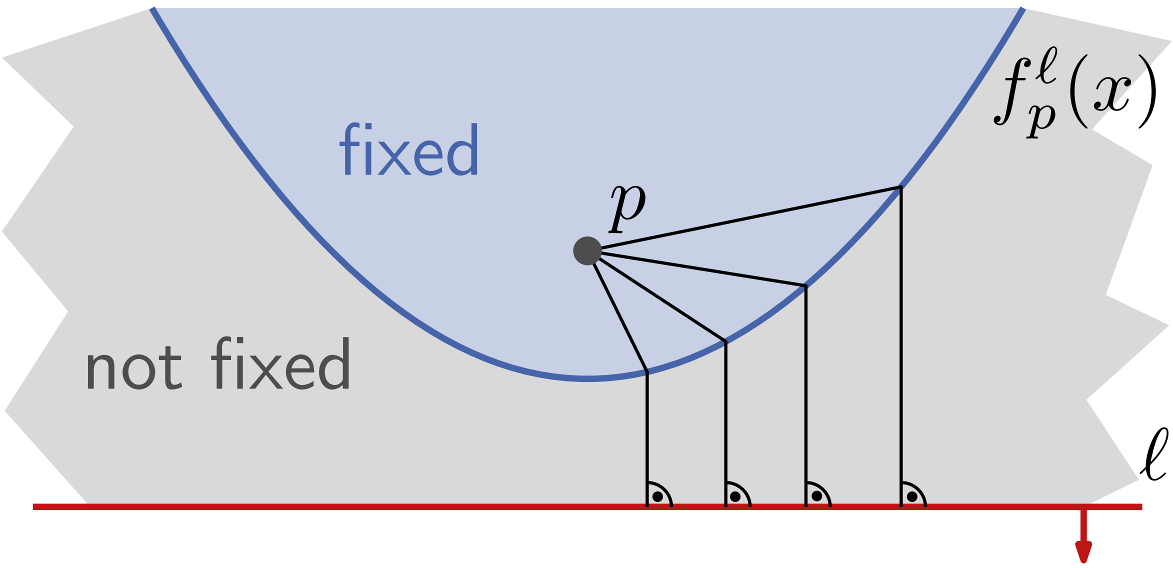
Instead, we view the part above ℓ as already fixed!



In the Direction of the Sweep Line

Obviously it's the intersection of $\mathrm{Vor}(P)$ and sweep line ℓ at the current time point is not known yet.

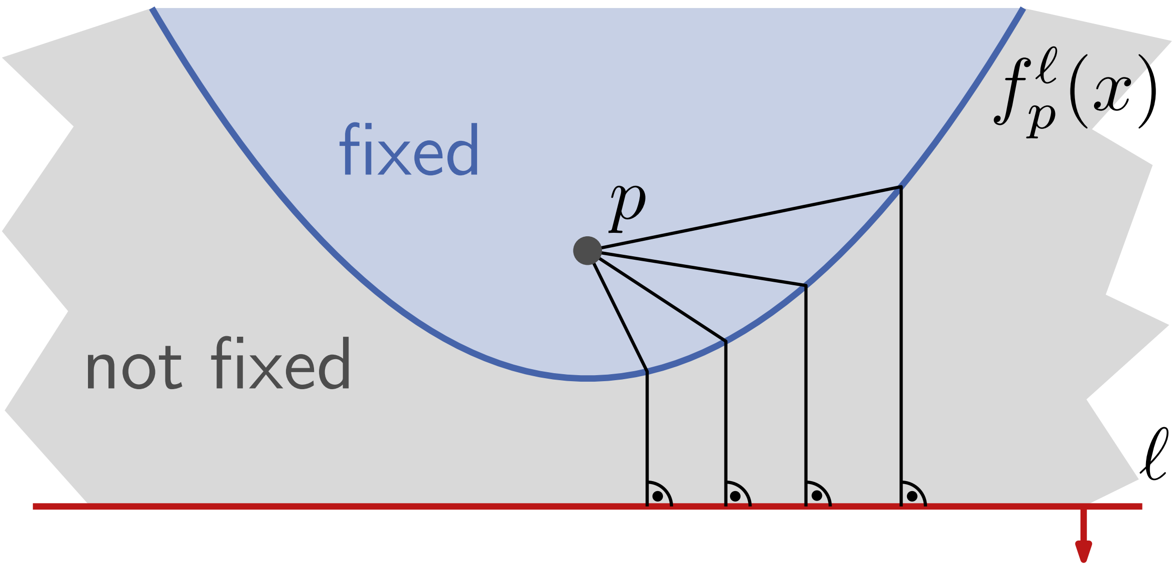
Instead, we view the part above ℓ as already fixed!



In the Direction of the Sweep Line

Obviously it's the intersection of $\mathrm{Vor}(P)$ and sweep line ℓ at the current time point is not known yet.

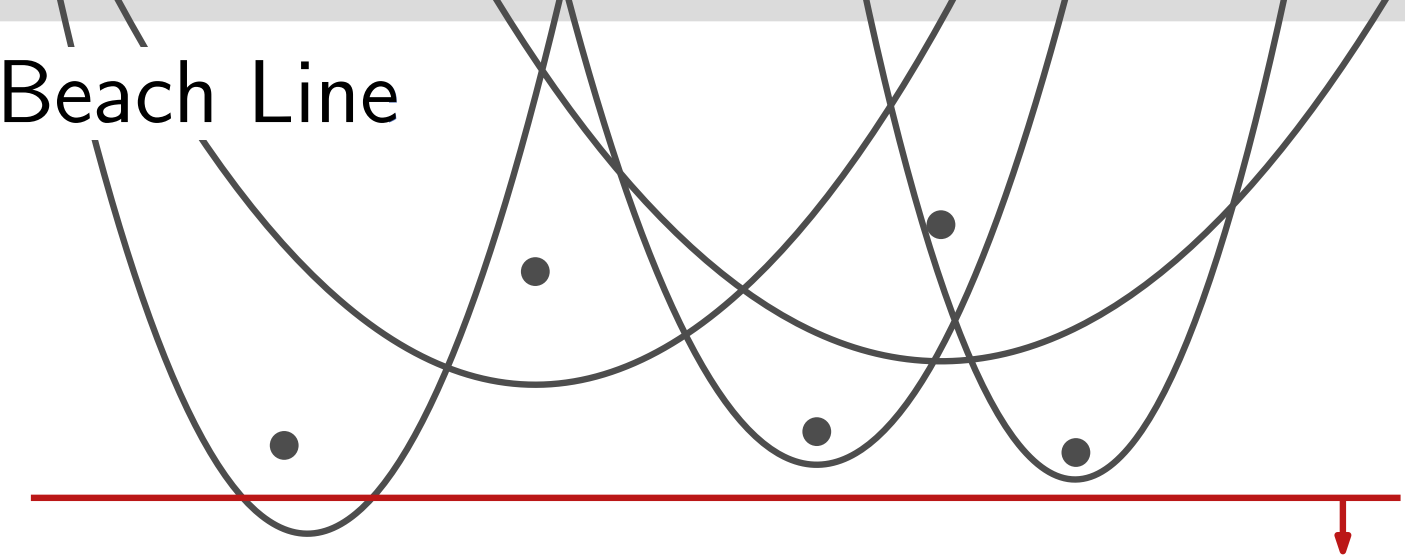
Instead, we view the part above ℓ as already fixed!



Enforcing the equality $|pq| = |q\ell|$ gives

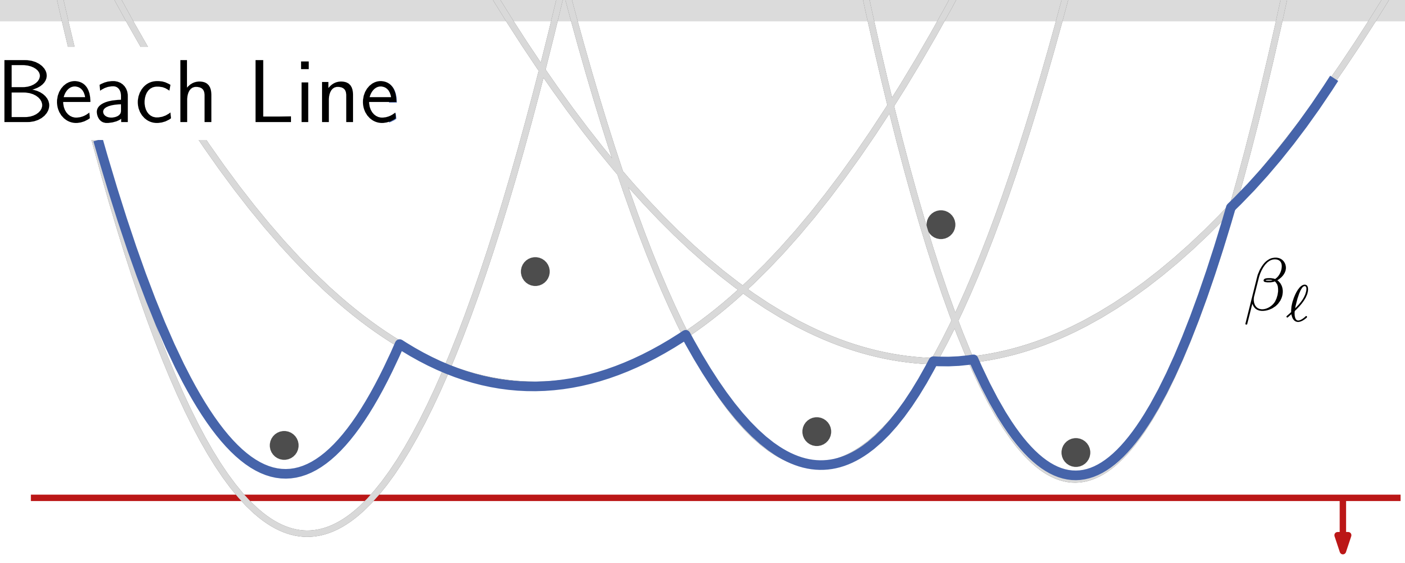
$$f_p^\ell(x) = \frac{1}{2(p_y - \ell_y)}(x - p_x)^2 + \frac{p_y + \ell_y}{2}$$

The Beach Line



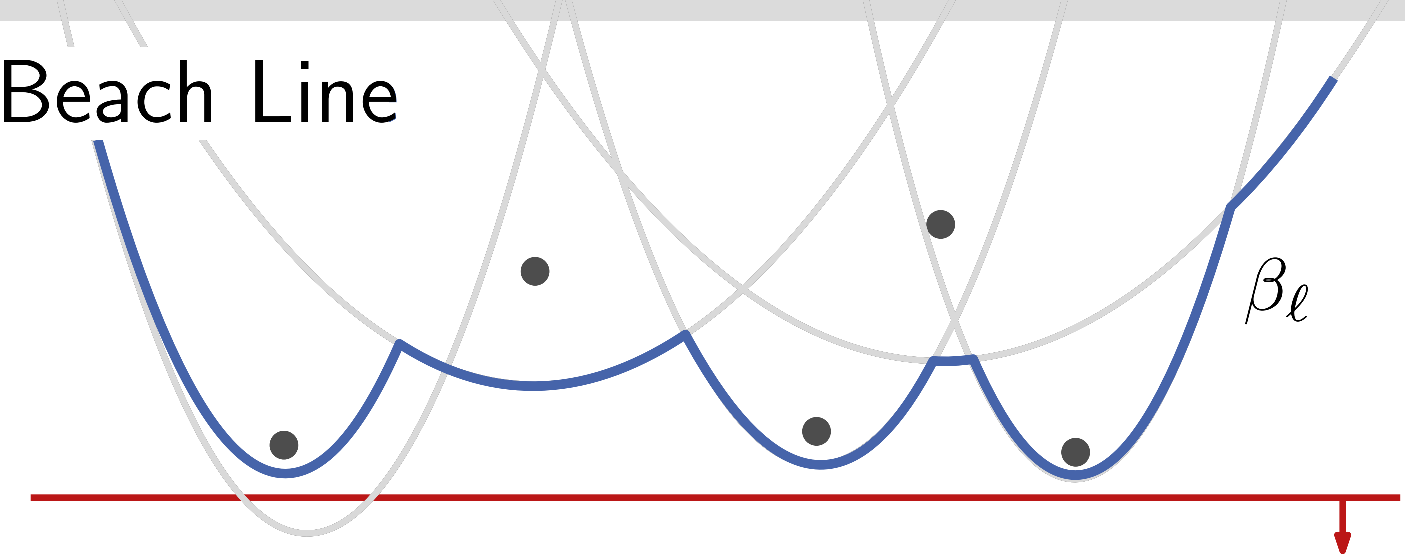
Definition: The **beach line** β_ℓ is the lower envelope of parabolas f_p^ℓ for the points already found.

The Beach Line



Definition: The **beach line** β_ℓ is the lower envelope of parabolas f_p^ℓ for the points already found.

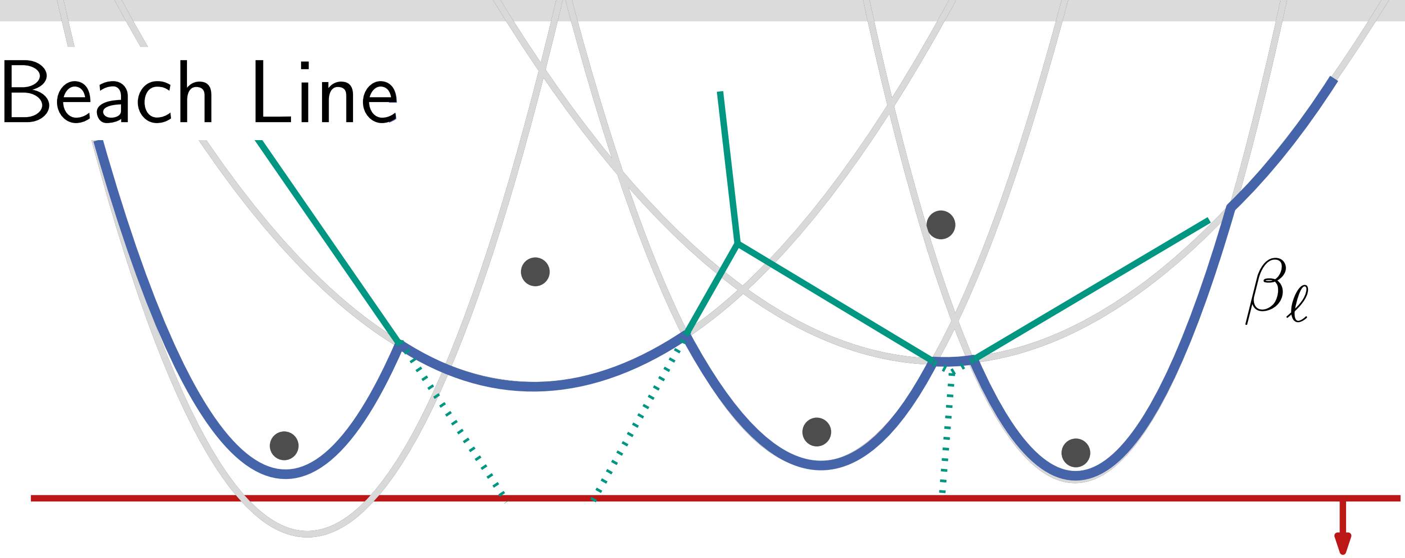
The Beach Line



Definition: The **beach line** β_ℓ is the lower envelope of parabolas f_p^ℓ for the points already found.

What does it have to do with $\mathrm{Vor}(P)$?

The Beach Line

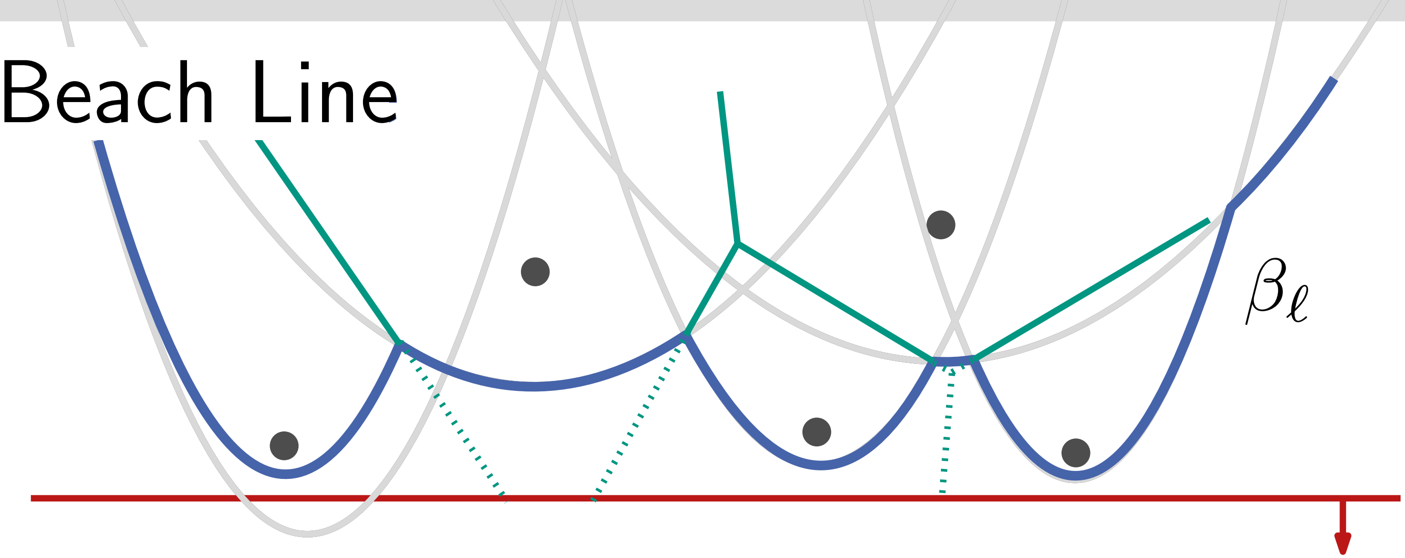


Definition: The **beach line** β_ℓ is the lower envelope of parabolas f_p^ℓ for the points already found.

What does it have to do with $\mathrm{Vor}(P)$?

Obs.:
- The beach line is x-monotone
- Intersection points in the beach line lie on Voronoi edges

The Beach Line



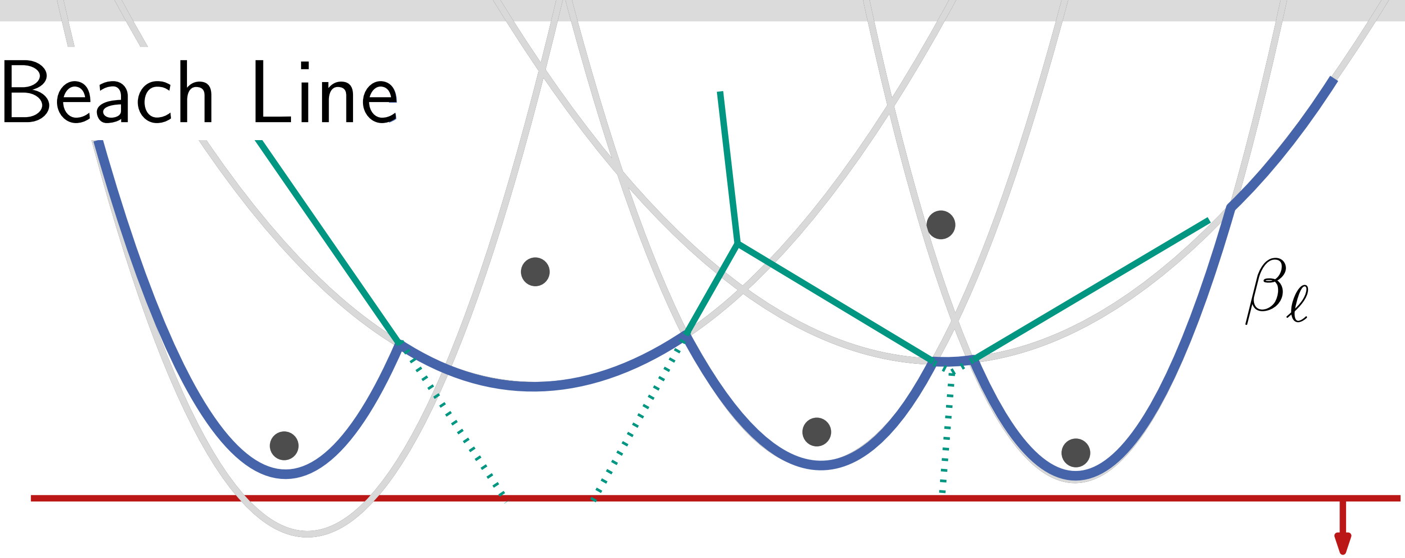
Definition: The **beach line** β_ℓ is the lower envelope of parabolas f_p^ℓ for the points already found.

What does it have to do with $\mathrm{Vor}(P)$?

Obs.:
- The beach line is x-monotone
- Intersection points in the beach line lie on Voronoi edges

Even: Intersection points run along $\mathrm{Vor}(P)$

The Beach Line



Definition: The **beach line** β_ℓ is the lower envelope of parabolas f_p^ℓ for the points already found.

What does it have to do with $\mathrm{Vor}(P)$?

Obs.:
- The beach line is x-monotone
- Intersection points in the beach line lie on Voronoi edges

Even: Intersection points run along $\mathrm{Vor}(P)$

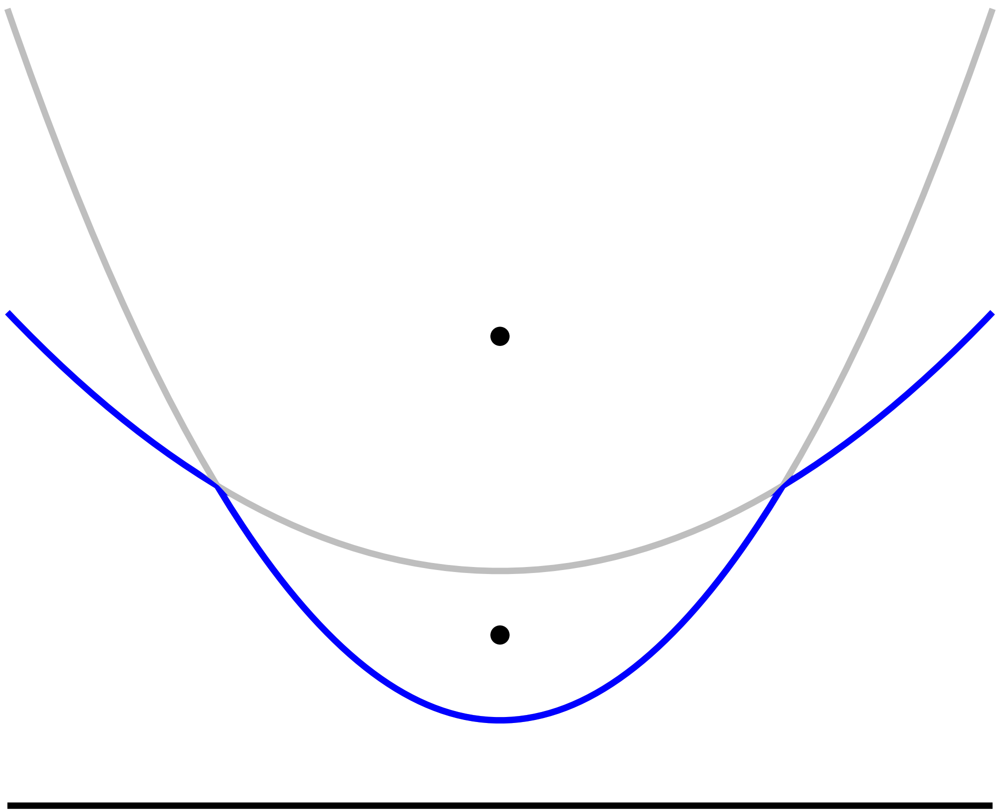
Goal: Store (implicit) contour β_ℓ instead of $\mathrm{Vor}(P) \cap \ell$

Exercise 2

Give an example where a parabola contributes more than one arc to the beach line.

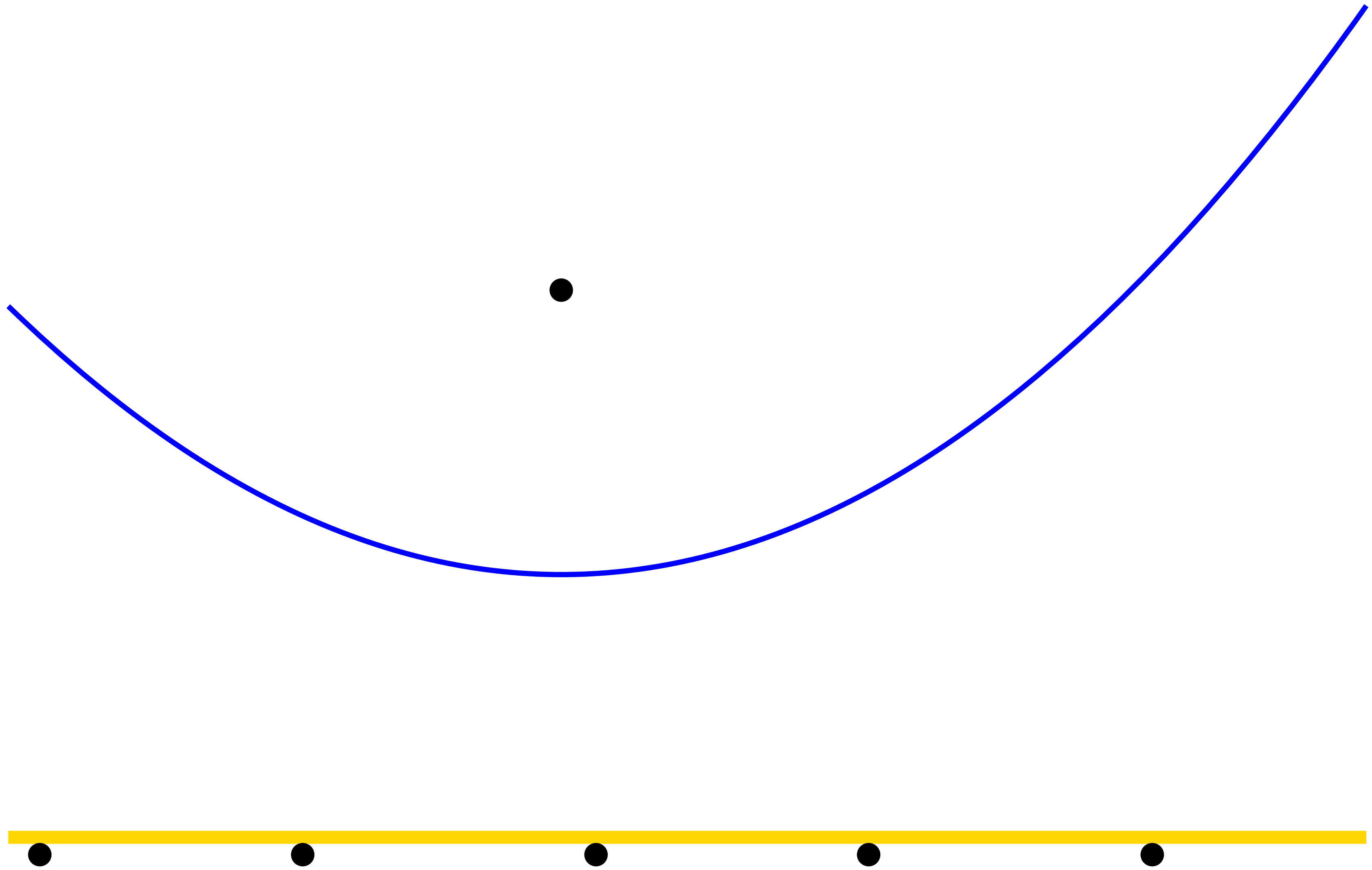
Exercise 2

Give an example where a parabola contributes more than one arc to the beach line.



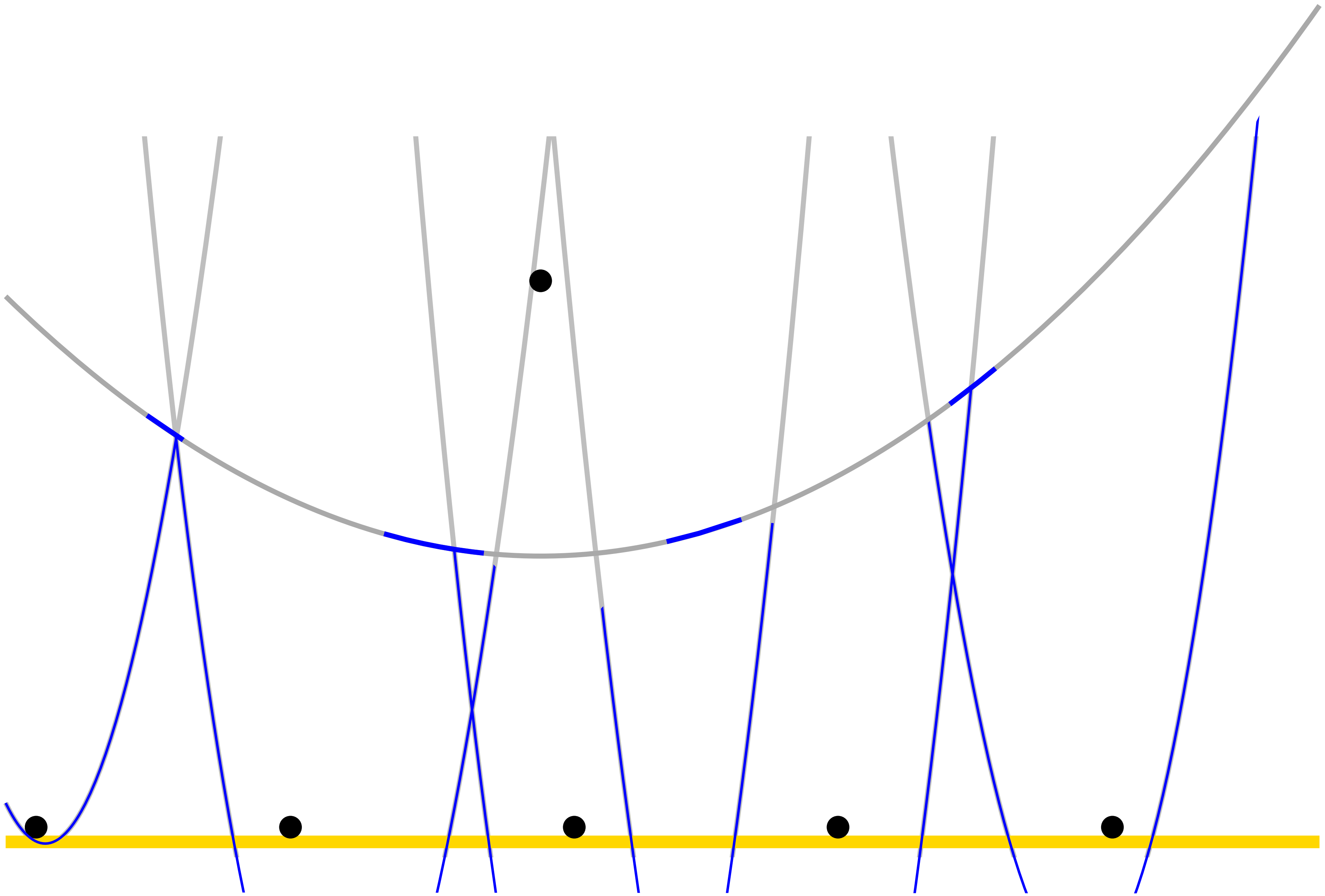
Exercise 2

A parabola, that contributes linear many arcs to the beach line.



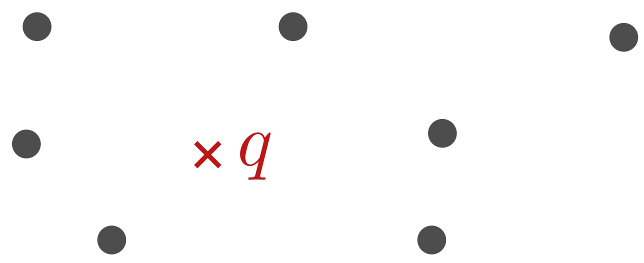
Exercise 2

A parabola, that contributes linear many arcs to the beach line.



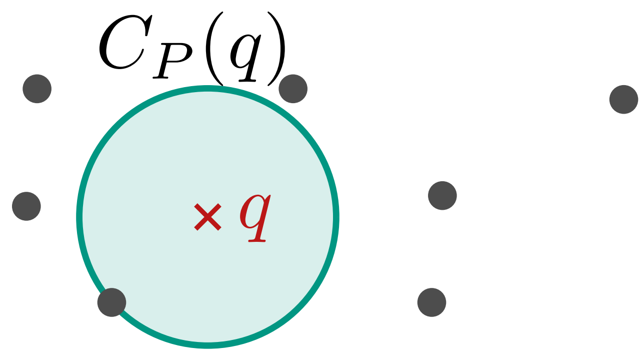
Characterization

Definition: Let q be a point. Define $C_P(q)$ to be the points in P that lie on the empty circle with center q.



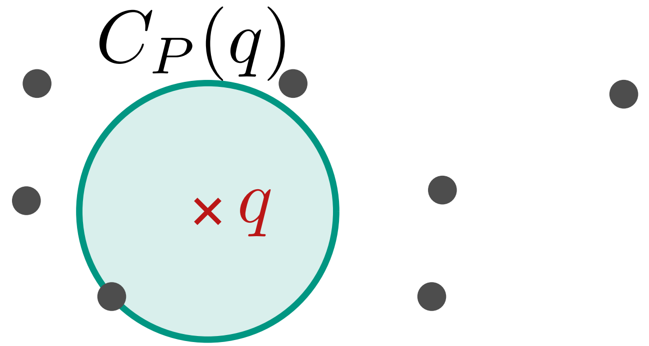
Characterization

Definition: Let q be a point. Define $C_P(q)$ to be the points in P that lie on the empty circle with center q.



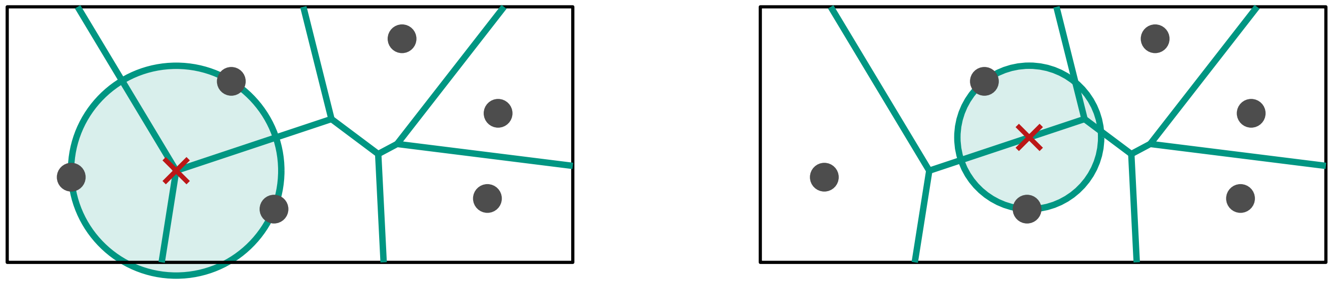
Characterization

Definition: Let q be a point. Define $C_P(q)$ to be the points in P that lie on the empty circle with center q.



Theorem 3:

- A point q is a Voronoi vertex $\Leftrightarrow |C_P(q) \cap P| \geq 3$,
- the bisector $b(p_i, p_j)$ defines a Voronoi edge $\Leftrightarrow \exists q \in b(p_i, p_j)$ with $C_P(q) \cap P = \{p_i, p_j\}$.



Exercise 3 – Next Neighbor

Given: Set P of n points.

Describe algorithm that computes for each point $p \in P$ its next neighbor $a(p)$ in P in $O(n \log n)$ time.

Exercise 3 – Next Neighbor

Given: Set P of n points.

Describe algorithm that computes for each point $p \in P$ its next neighbor $a(p)$ in P in $O(n \log n)$ time.

Idea:

1. Construct Voronoi-Diagramm $\mathrm{Vor}(P)$ in $O(n \log n)$ time.
2. Using $\mathrm{Vor}(P)$ to find for each point p its next neighbor in $\mathcal{O}(n)$ time.

Proof:

Show that $a(p)$ lies in a adjacent cell of the cell of p.

Exercise 3 – Next Neighbor

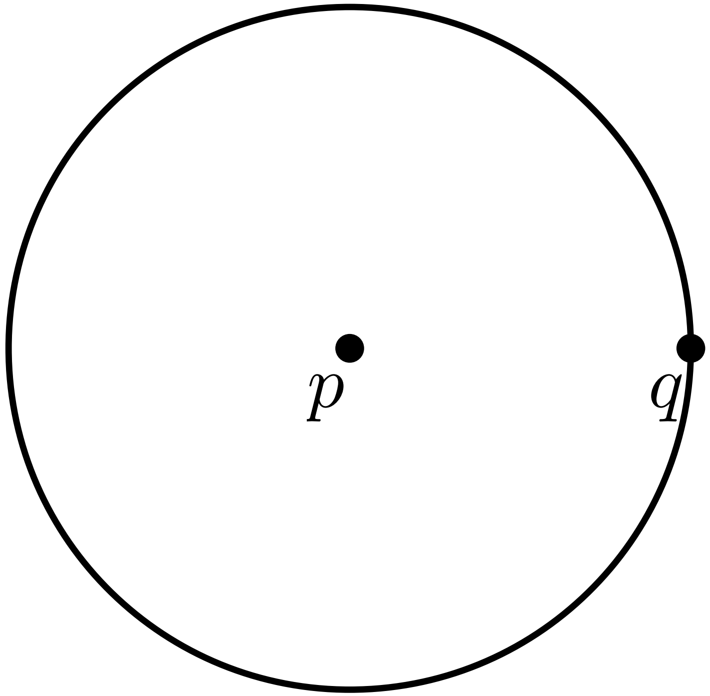
Given: Set P of n points.

Describe algorithm that computes for each point $p \in P$ its next neighbor $a(p)$ in P in $O(n \log n)$ time.

Idea:
1. Construct Voronoi-Diagramm $\mathrm{Vor}(P)$ in $O(n \log n)$ time.
2. Using $\mathrm{Vor}(P)$ to find for each point p its next neighbor in $\mathcal{O}(n)$ time.

Proof:

Show that $a(p)$ lies in a adjacent cell of the cell of p.



Exercise 3 – Next Neighbor

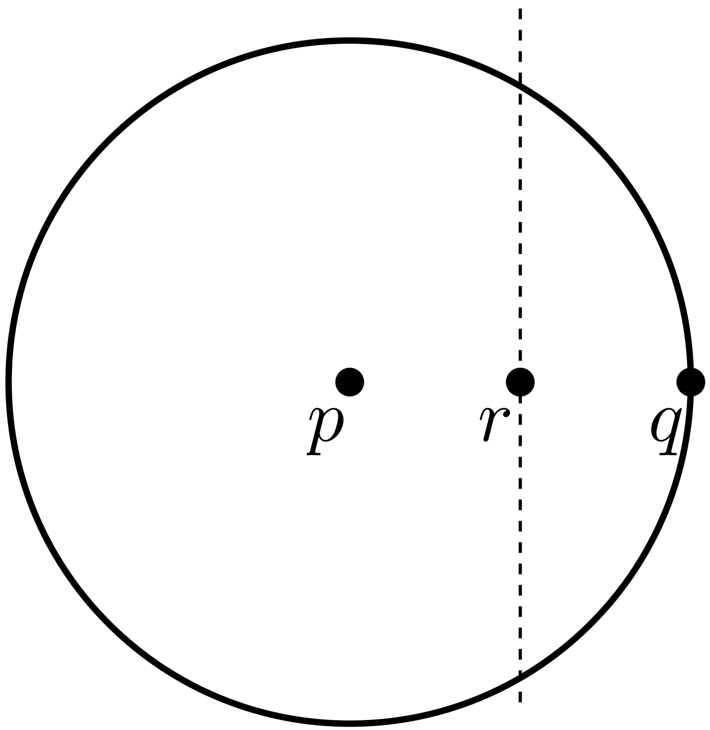
Given: Set P of n points.

Describe algorithm that computes for each point $p \in P$ its next neighbor $a(p)$ in P in $O(n \log n)$ time.

Idea:
1. Construct Voronoi-Diagramm $\mathrm{Vor}(P)$ in $O(n \log n)$ time.
2. Using $\mathrm{Vor}(P)$ to find for each point p its next neighbor in $\mathcal{O}(n)$ time.

Proof:

Show that $a(p)$ lies in a adjacent cell of the cell of p.



Exercise 3 – Next Neighbor

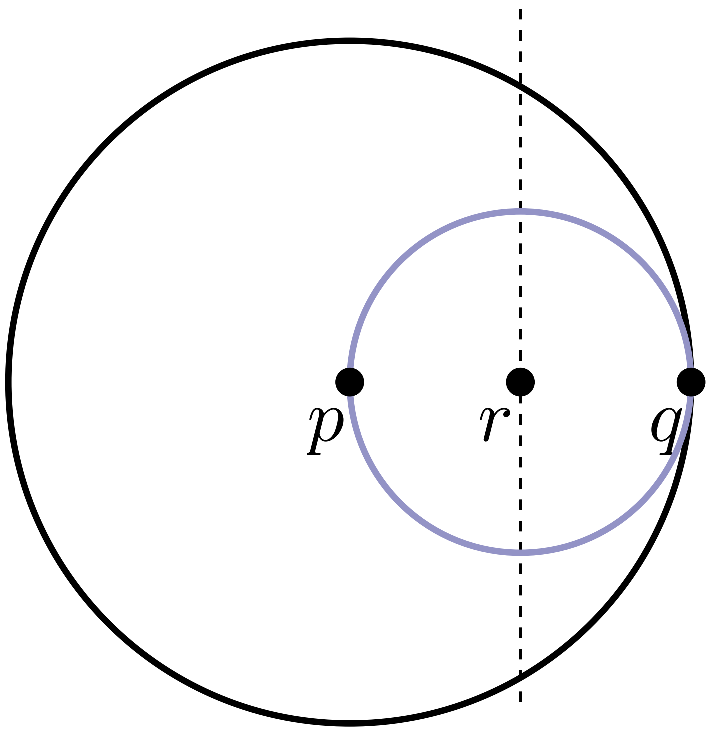
Given: Set P of n points.

Describe algorithm that computes for each point $p \in P$ its next neighbor $a(p)$ in P in $O(n \log n)$ time.

Idea:
1. Construct Voronoi-Diagramm $\mathrm{Vor}(P)$ in $O(n \log n)$ time.
2. Using $\mathrm{Vor}(P)$ to find for each point p its next neighbor in $\mathcal{O}(n)$ time.

Proof:

Show that $a(p)$ lies in a adjacent cell of the cell of p.



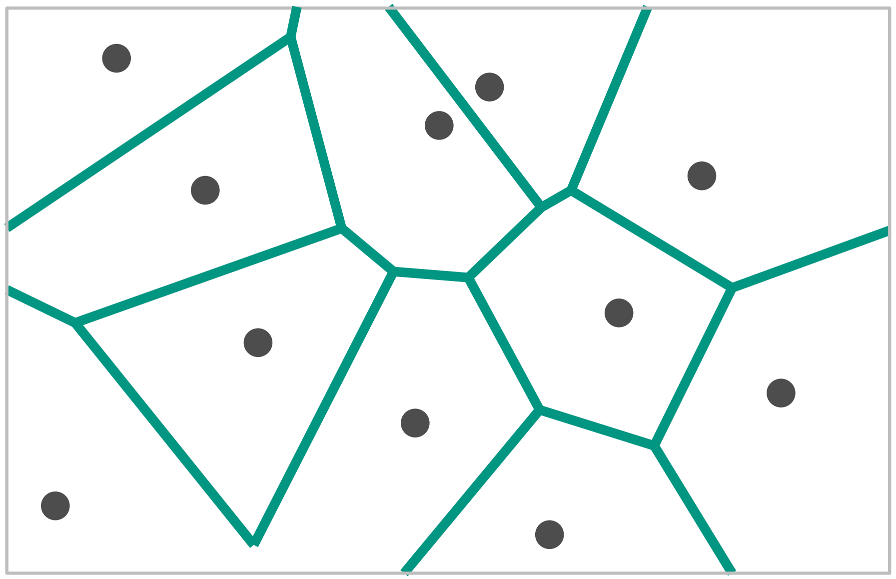
Exercise 4 – Nuclear Power Plants

Given:

Find: Point $p \in R$ with maximum distance to nuclear power plants.

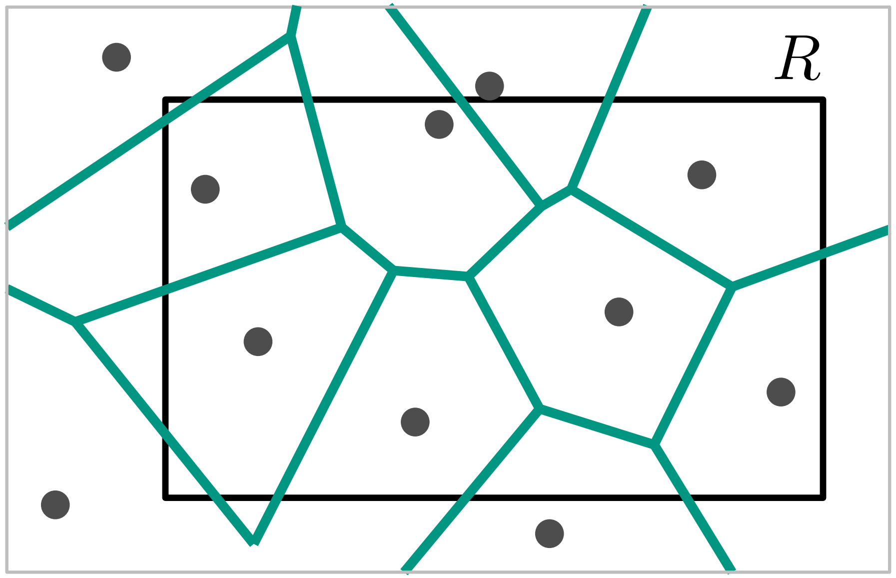
Exercise 4 – Nuclear Power Plants

Given: nuclear plants (set P of points) and preferred region (rectangle R).
Find: Point $p \in R$ with maximum distance to nuclear power plants.



Exercise 4 – Nuclear Power Plants

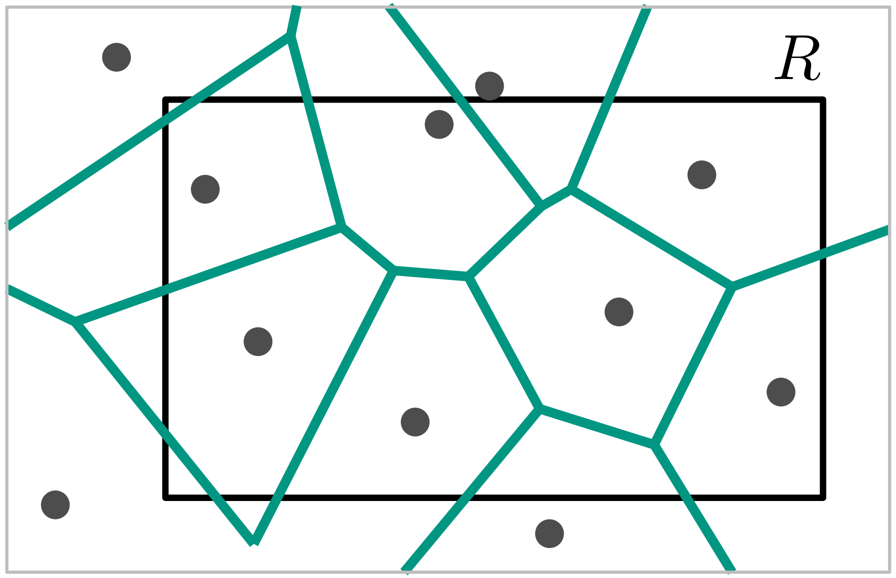
Given: nuclear plants (set P of points) and preferred region (rectangle R).
Find: Point $p \in R$ with maximum distance to nuclear power plants.



Exercise 4 – Nuclear Power Plants

Given: nuclear plants (set P of points) and preferred region (rectangle R).
Find: Point $p \in R$ with maximum distance to nuclear power plants.
Candidates:
- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten

a) Candidates are sufficient
b) Computation in $\mathcal{O}(n)$ time

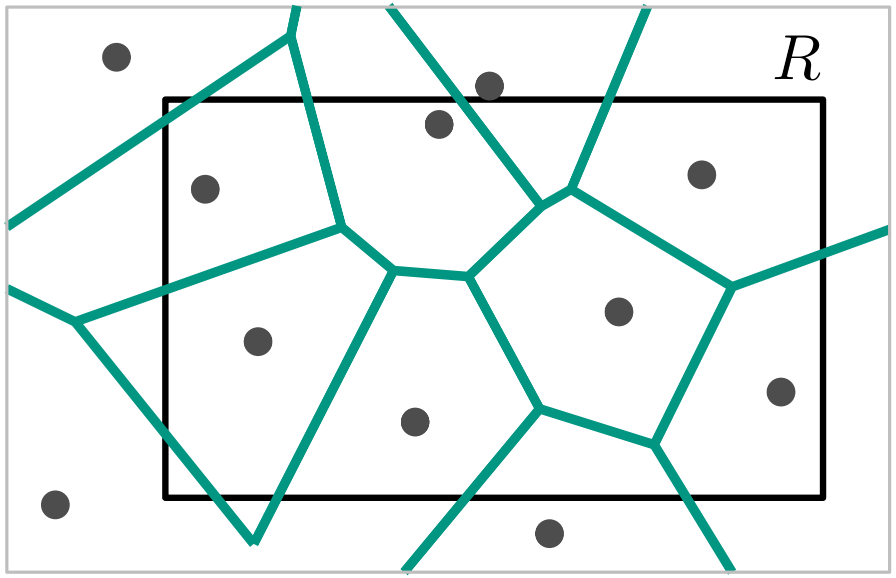


Exercise 4 – Nuclear Power Plants

Given: nuclear plants (set P of points) and preferred region (rectangle R).
Find: Point $p \in R$ with maximum distance to nuclear power plants.
Candidates:

- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten

a) Candidates are sufficient
b) Computation in $\mathcal{O}(n)$ time



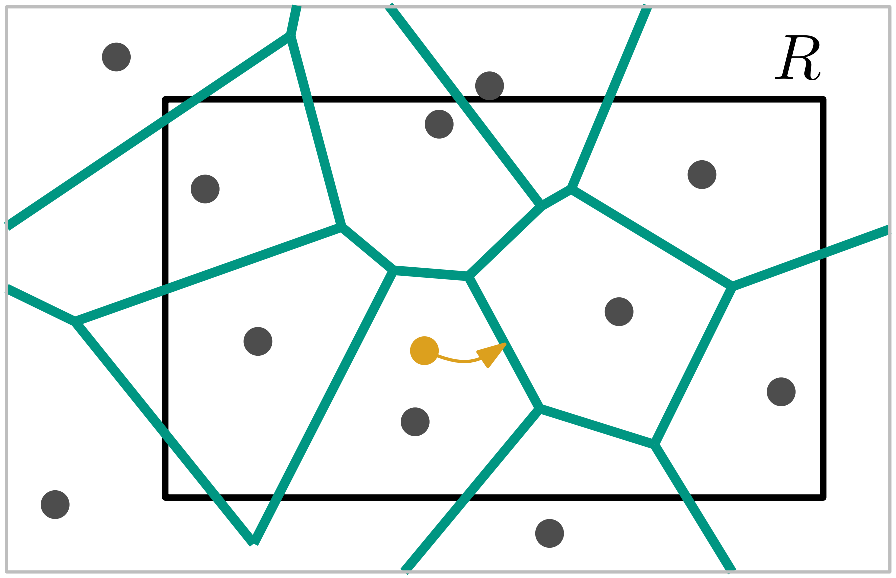
Assume: optimal Point r that does not satisfy criteria.

Construct better solution:

Exercise 4 – Nuclear Power Plants

Given: nuclear plants (set P of points) and preferred region (rectangle R).
Find: Point $p \in R$ with maximum distance to nuclear power plants.
Candidates:
- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten

a) Candidates are sufficient
b) Computation in $\mathcal{O}(n)$ time



Assume: optimal Point r that does not satisfy criteria.

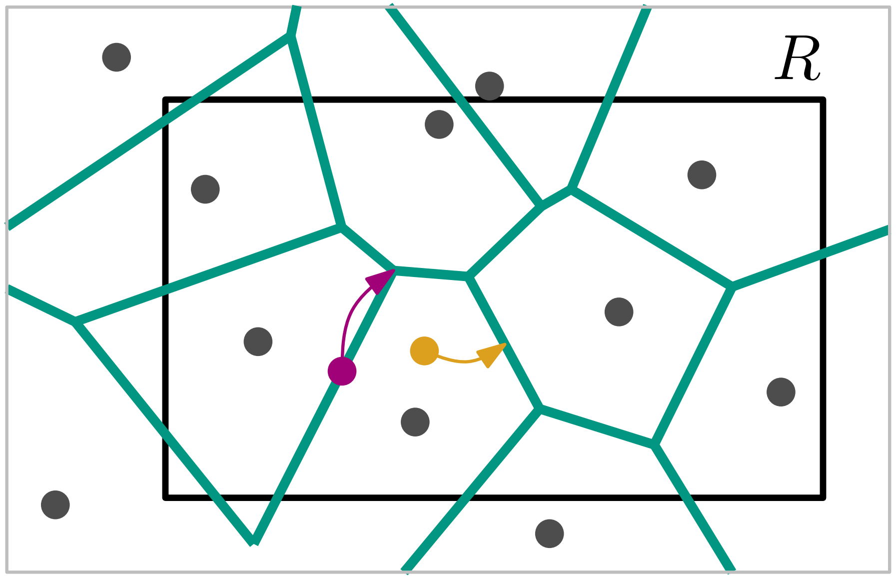
Construct better solution:

r is contained in interior of cell.
$\longrightarrow$ Move r towards boundary.

Exercise 4 – Nuclear Power Plants

Given: nuclear plants (set P of points) and preferred region (rectangle R).
Find: Point $p \in R$ with maximum distance to nuclear power plants.
Candidates:
- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten

a) Candidates are sufficient
b) Computation in $\mathcal{O}(n)$ time



Assume: optimal Point r that does not satisfy criteria.

Construct better solution:

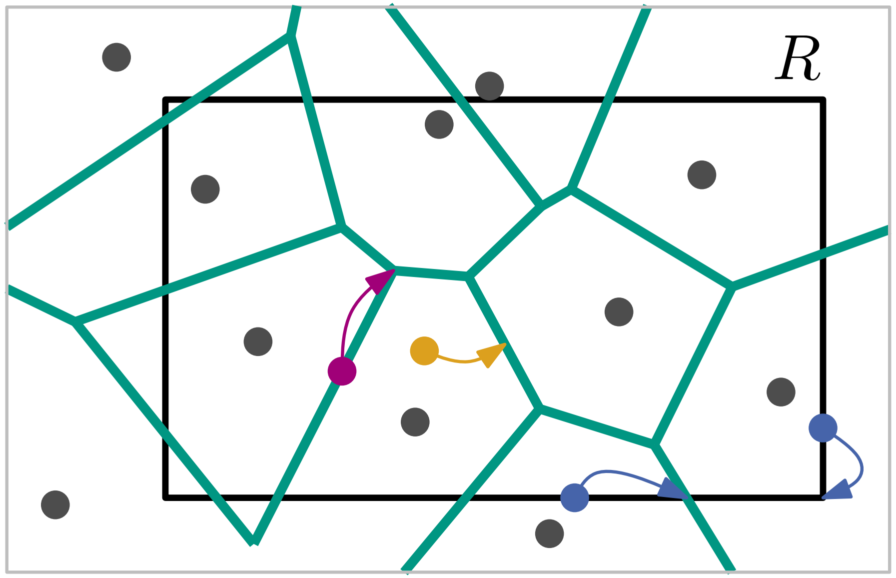
r is contained in interior of cell. → Move r towards boundary.

r lies on voronoi edge. → Push towards to one of both end points.

Exercise 4 – Nuclear Power Plants

Given: nuclear plants (set P of points) and preferred region (rectangle R).
Find: Point $p \in R$ with maximum distance to nuclear power plants.
Candidates:
- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten

a) Candidates are sufficient
b) Computation in $\mathcal{O}(n)$ time



Assume: optimal Point r that does not satisfy criteria.

Construct better solution:

r is contained in interior of cell.
→ Move r towards boundary.

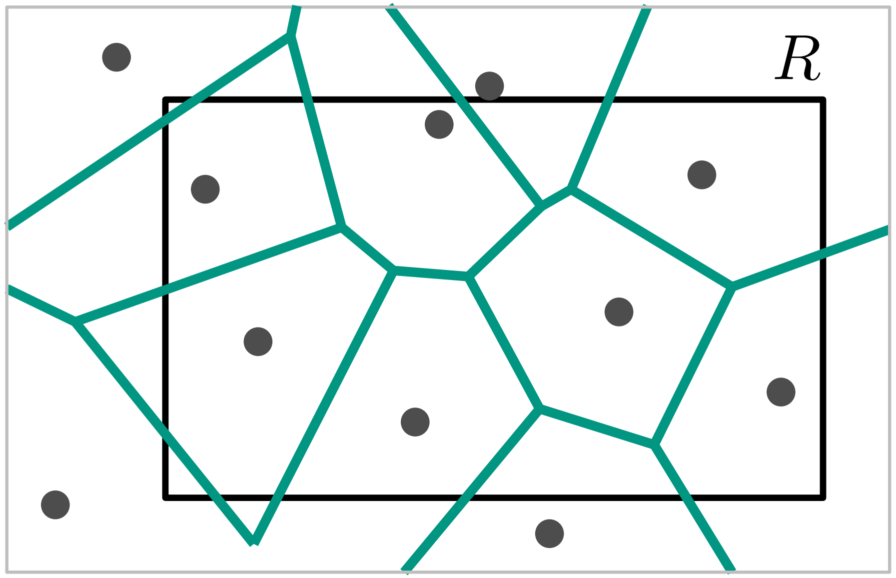
r lies on voronoi edge. → Push towards to one of both end points.

r lies on edge of R → Push towards intersection of R & Vor.-edges, or
→ Push towards corner of R

Exercise 4 – Nuclear Power Plants

Given: nuclear plants (set P of points) and preferred region (rectangle R).
Find: Point $p \in R$ with maximum distance to nuclear power plants.
Candidates:
- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten

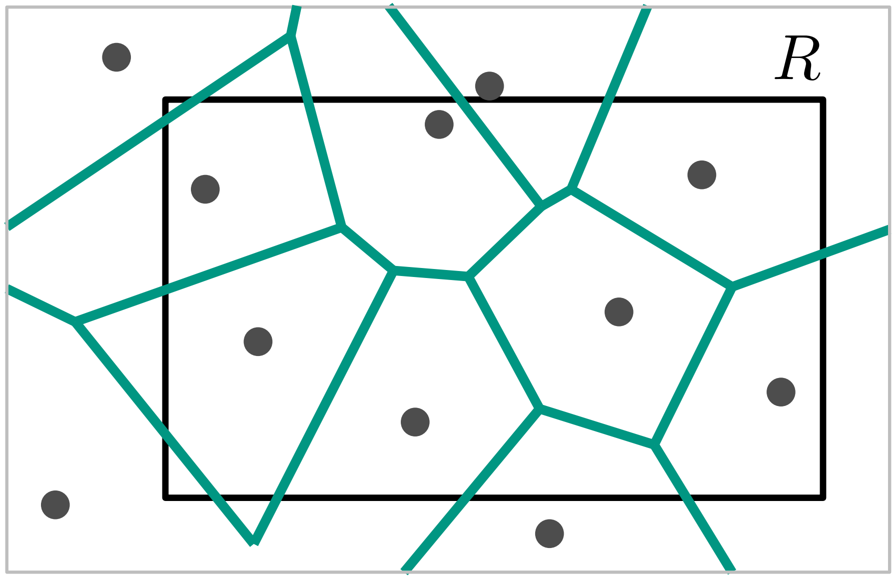
a) Candidates are sufficient
b) Computation in $\mathcal{O}(n)$ time



Exercise 4 – Nuclear Power Plants

Given: nuclear plants (set P of points) and preferred region (rectangle R).
Find: Point $p \in R$ with maximum distance to nuclear power plants.
Candidates:
- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten

a) Candidates are sufficient
b) Computation in $\mathcal{O}(n)$ time



Idea: Check all candidates.

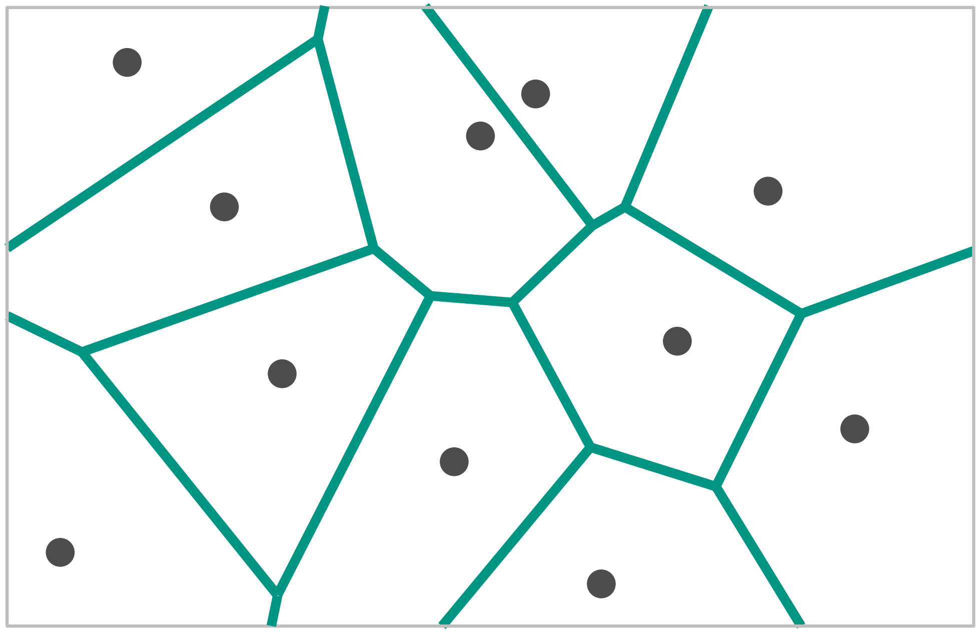
There are $O(n)$ many:

- $O(n)$ many Vor.–edges & –vertices
- R intersects each edge in at most two intersection points.

⟶ Computation in $O(n)$ linear time.

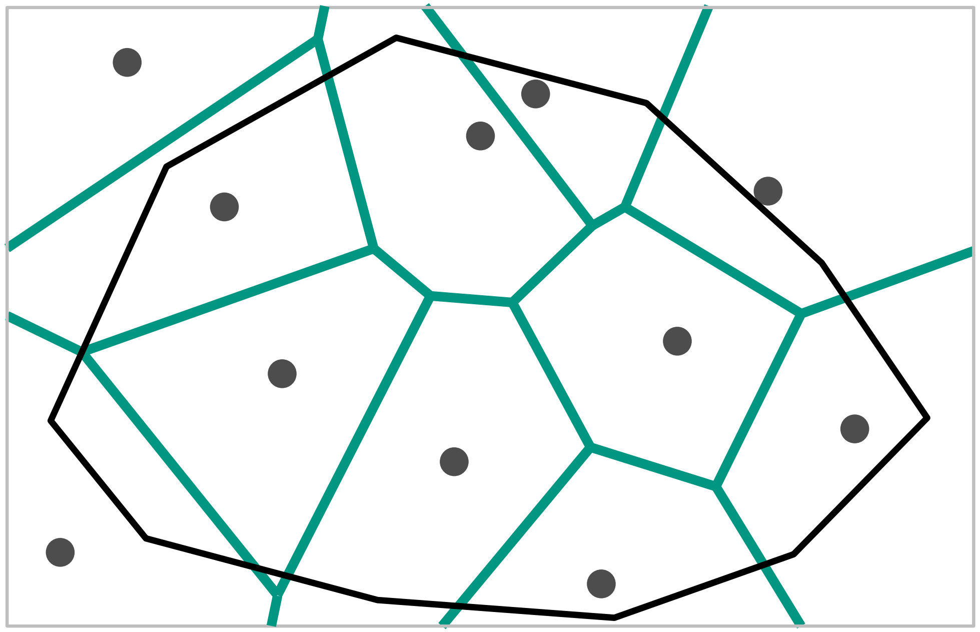
Exercise 4 – Nuclear Power Plants

Now: Convex polygon P with m nodes instead of rectangle R.



Exercise 4 – Nuclear Power Plants

Now: Convex polygon P with m nodes instead of rectangle R.

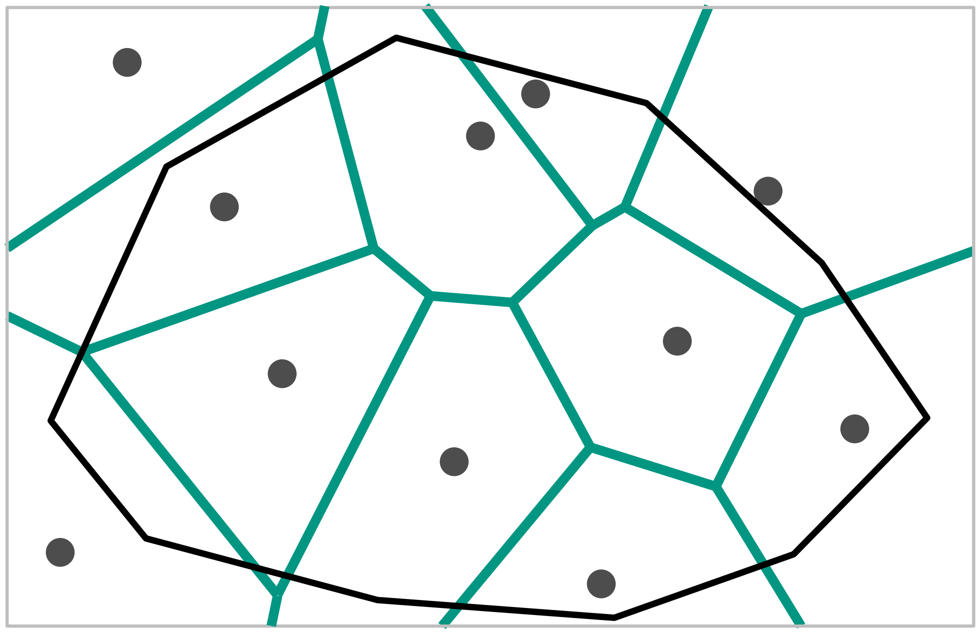


Exercise 4 – Nuclear Power Plants

Now: Convex polygon P with m nodes instead of rectangle R.

Candidates:

- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten



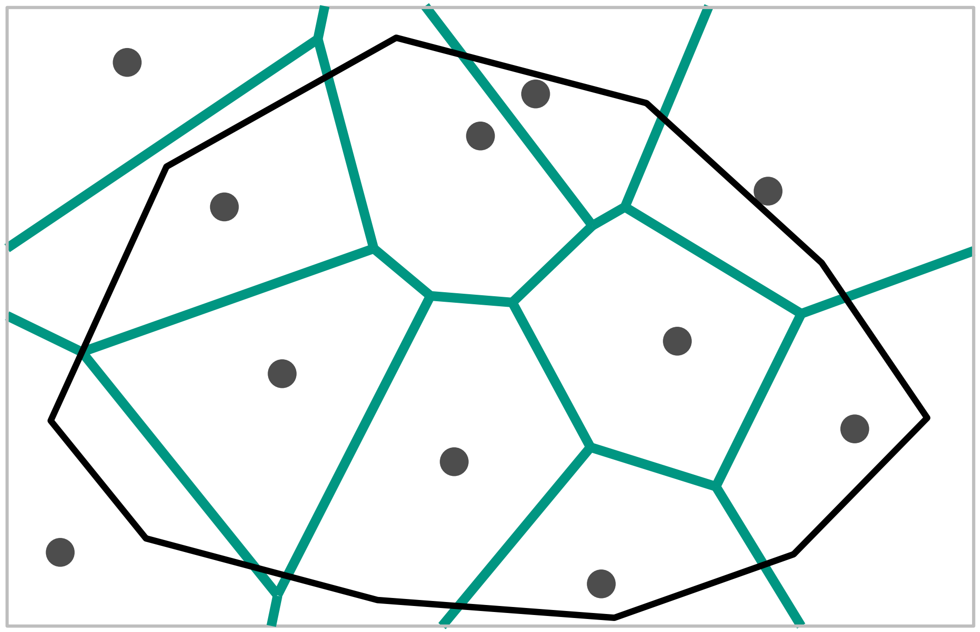
1) Candidates are sufficient
2) Computaiton in $\mathcal{O}(n+m)$ time

Exercise 4 – Nuclear Power Plants

Now: Convex polygon P with m nodes instead of rectangle R.

Candidates:

- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten



1) Candidates are sufficient
2) Computaiton in $\mathcal{O}(n+m)$ time

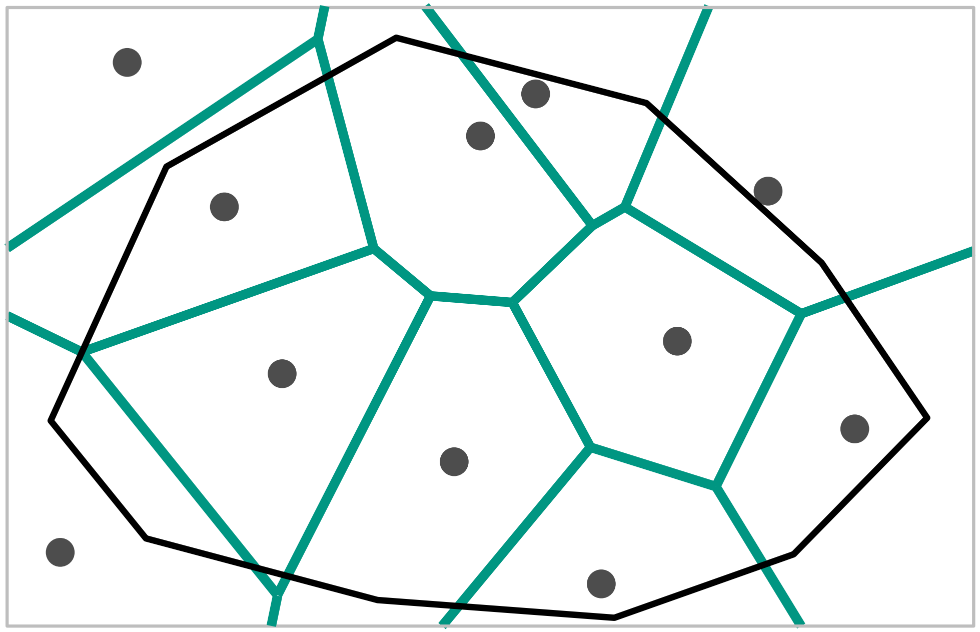
1.) Same arguments as in a).

Exercise 4 – Nuclear Power Plants

Now: Convex polygon P with m nodes instead of rectangle R.

Candidates:

- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten



1) Candidates are sufficient
2) Computaiton in $\mathcal{O}(n+m)$ time

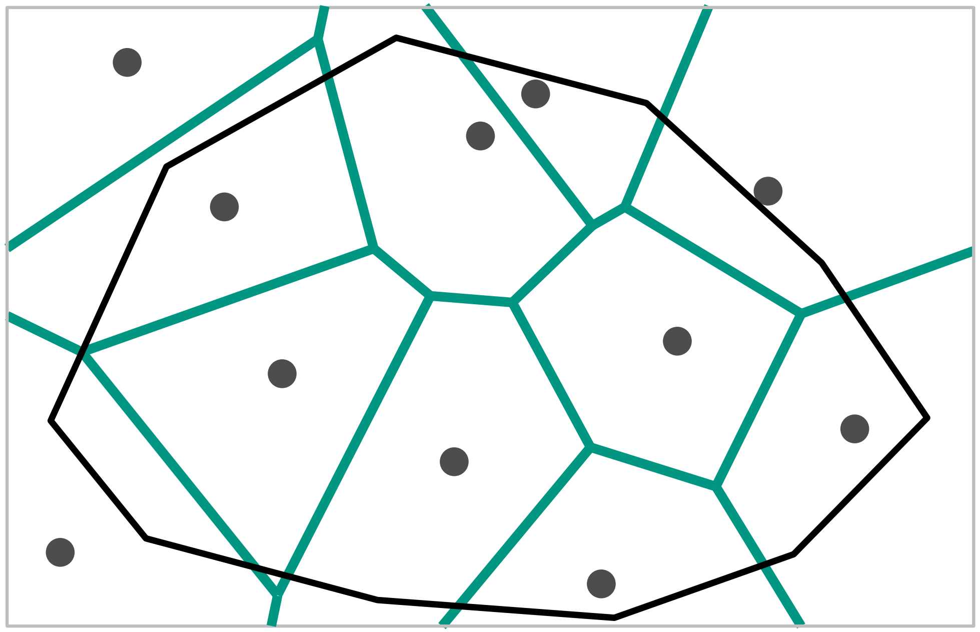
1.) Same arguments as in a).

Exercise 4 – Nuclear Power Plants

Now: Convex polygon P with m nodes instead of rectangle R.

Candidates:

- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten



1) Candidates are sufficient
2) Computaiton in $\mathcal{O}(n+m)$ time

1.) Same arguments as in a).

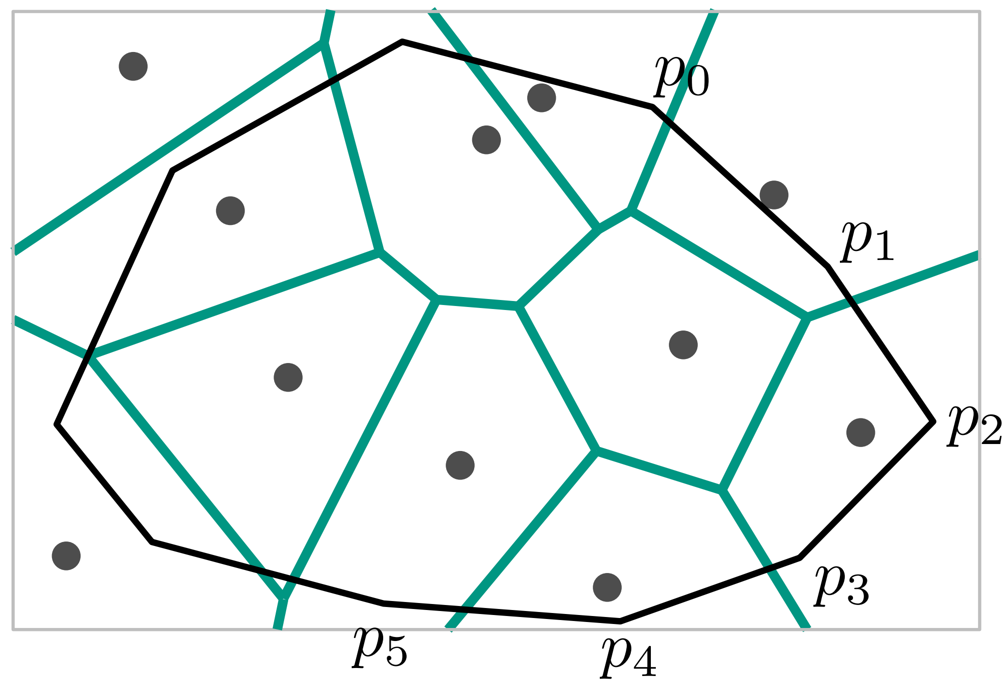
Exercise 4 – Nuclear Power Plants

Now: Convex polygon P with m nodes instead of rectangle R.

Candidates:

- Voronoi vertices
- Corners of R
- Intersections of R & Voronoi-Kanten

1) Candidates are sufficient
2) Computaiton in $\mathcal{O}(n+m)$ time



1.) Same arguments as in a).

2.) Determine set $\mathcal{C}$ of all cells that are intersected by P or contained in P in $O(m+n)$ time.

$\rightarrow$ set $\mathcal{C}$ induces candidates.

Construct Set $\mathcal{C}$

Find cells intersected by P:

Init.: $C \leftarrow$ cell containing p_0, $i \leftarrow 0$.

1) Mark C as visited

2) Determine intersection s of half-line through p_i, p_{i+1} with cell C

Construct Set $\mathcal{C}$

Find cells intersected by P:

Init.: $C \leftarrow$ cell containing p_0, $i \leftarrow 0$.

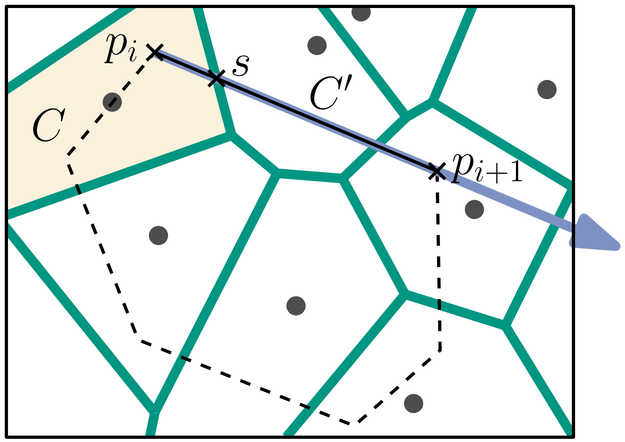
1) Mark C as visited

2) Determine intersection s of half-line through p_i, p_{i+1} with cell C

Case 1: s lies on segment $\overline{p_i p_{i+1}}$.

→ p_{i+1} lies not in C.

$C \leftarrow C'$, where C' is neighbor cell induced by intersection s.



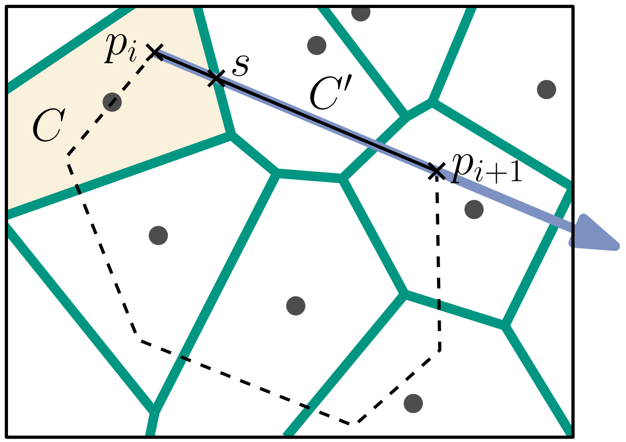
Construct Set $\mathcal{C}$

Find cells intersected by P**:**

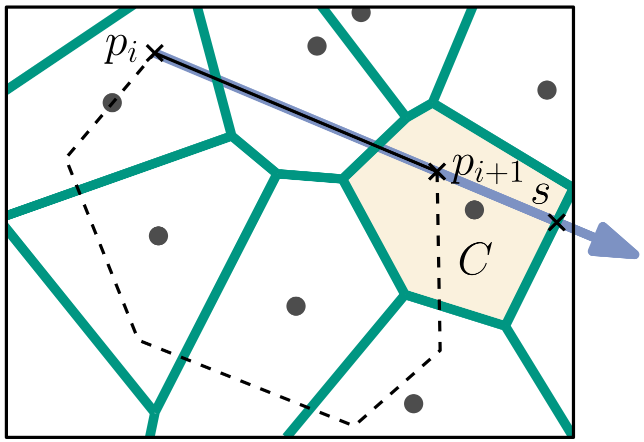
Init.: $C \leftarrow$ cell containing p_0, $i \leftarrow 0$.

1) Mark C as visited
2) Determine intersection s of half-line through p_i, p_{i+1} with cell C

Case 1: s lies on segment $\overline{p_i p_{i+1}}$.
→ p_{i+1} lies not in C.
$C \leftarrow C'$, where C' is neighbor cell induced by intersection s.



Case 2: s lies not on segment $\overline{p_i p_{i+1}}$.
→ p_{i+1} lies in C.
$i \leftarrow i + 1$



Construct Set $\mathcal{C}$

Find cells intersected by P:

Init.: $C \leftarrow$ cell containing p_0, $i \leftarrow 0$.

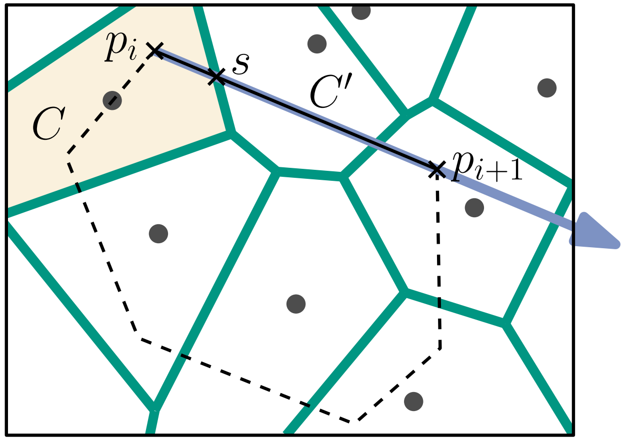
1) Mark C as visited

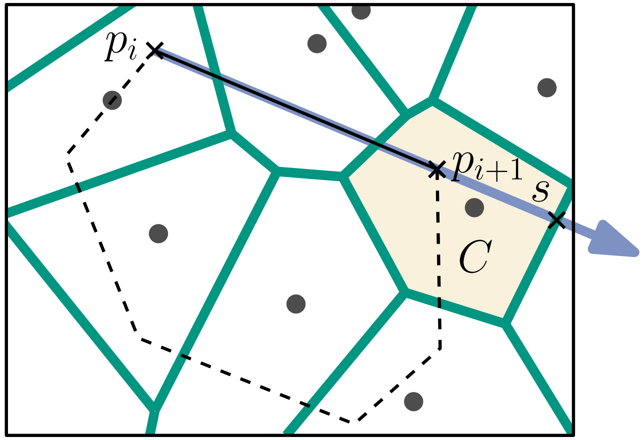
2) Determine intersection s of half-line through p_i, p_{i+1} with cell C

Case 1: s lies on segment $\overline{p_i p_{i+1}}$.
→ p_{i+1} lies not in C.
$C \leftarrow C'$, where C' is neighbor cell induced by intersection s.

Case 2: s lies not on segment $\overline{p_i p_{i+1}}$.
→ p_{i+1} lies in C.
$i \leftarrow i + 1$

3) Repeat 1)–3) until P is traversed.





Construct Set $\mathcal{C}$

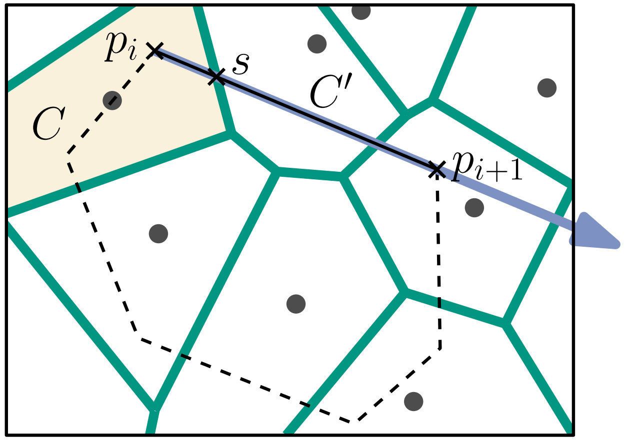
Find cells intersected by P:
Init.: $C \leftarrow$ cell containing p_0, $i \leftarrow 0$.

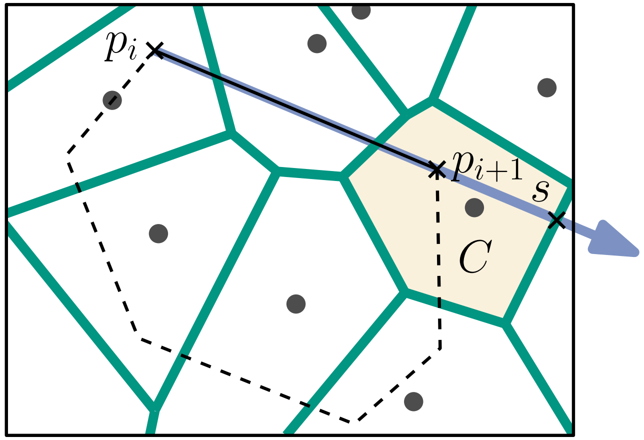
1) Mark C as visited
2) Determine intersection s of half-line through p_i, p_{i+1} with cell C

Case 1: s lies on segment $\overline{p_i p_{i+1}}$.
→ p_{i+1} lies not in C.
$C \leftarrow C'$, where C' is neighbor cell induced by intersection s.

Case 2: s lies not on segment $\overline{p_i p_{i+1}}$.
→ p_{i+1} lies in C.
$i \leftarrow i + 1$

3) Repeat 1)–3) until P is traversed.





Find cells contained in polygon.
depth-first search, bounded by the cells found in the first step.

Construct Set $\mathcal{C}$

Fi

In

Average running time $O(1)$, because each cell has $O(1)$ complexity in average.

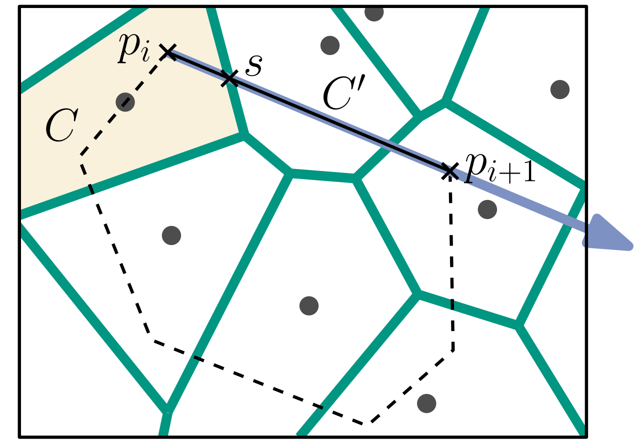
1) Mark C as visited

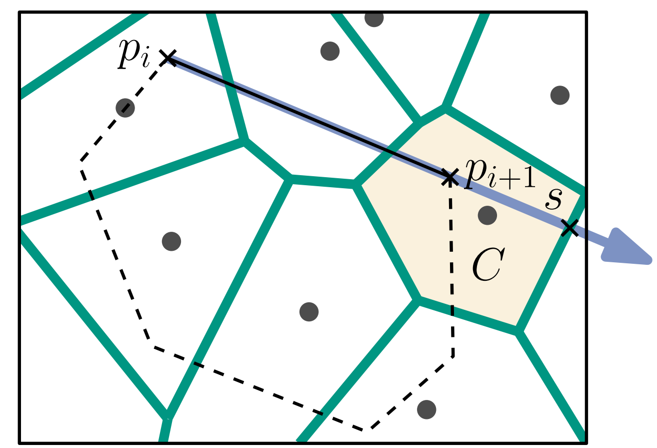
2) Determine intersection s of half-line through p_i, p_{i+1} with cell C

Case 1: s lies on segment $\overline{p_i p_{i+1}}$.
→ p_{i+1} lies not in C.
$C \leftarrow C'$, where C' is neighbor cell induced by intersection s.

Case 2: s lies not on segment $\overline{p_i p_{i+1}}$.
→ p_{i+1} lies in C.
$i \leftarrow i + 1$

3) Repeat 1)–3) until P is traversed.





Find cells contained in polygon.
depth-first search, bounded by the cells found in the first step.

Construct Set $\mathcal{C}$

Fi

In

Average running time $O(1)$, because each cell has $O(1)$ complexity in average.

1) Mark C as visited

2) Determine intersection s of half-line through p_i, p_{i+1} with cell C

Case 1: s lies on segment $\overline{p_i p_{i+1}}$.

$\longrightarrow$ p_{i+1} lies not in C.

$C \leftarrow C'$, where C' is neighbor cell

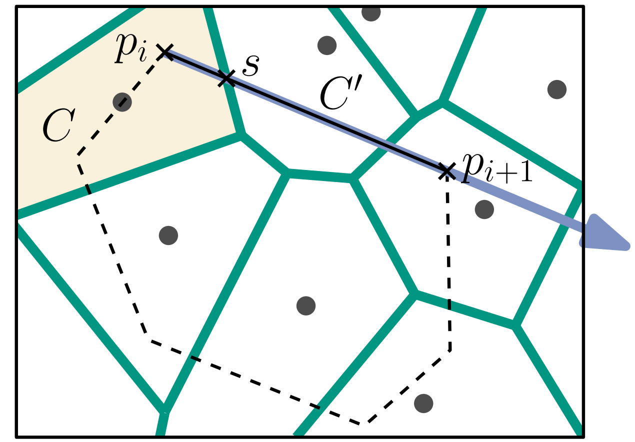
Since P is convex, each Voronoi edge is intersected at most twice.

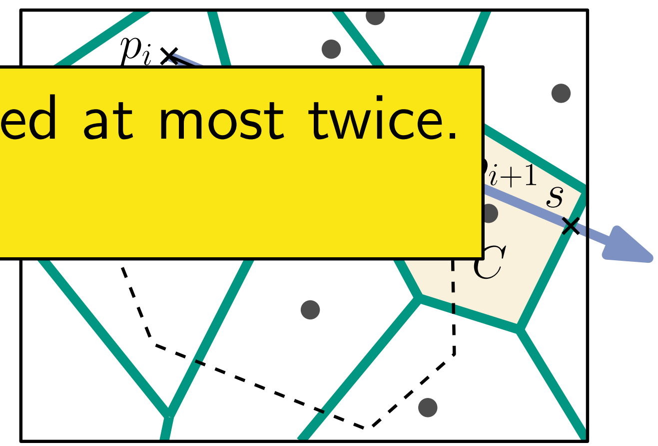
$\longrightarrow$ $O(m+n)$ running time.

$\longrightarrow$ p_{i+1} lies in C.

$i \leftarrow i + 1$

3) Repeat 1)–3) until P is traversed.





Find cells contained in polygon.

depth-first search, bounded by the cells found in the first step.