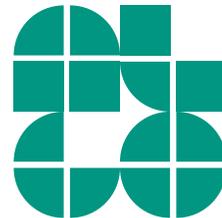


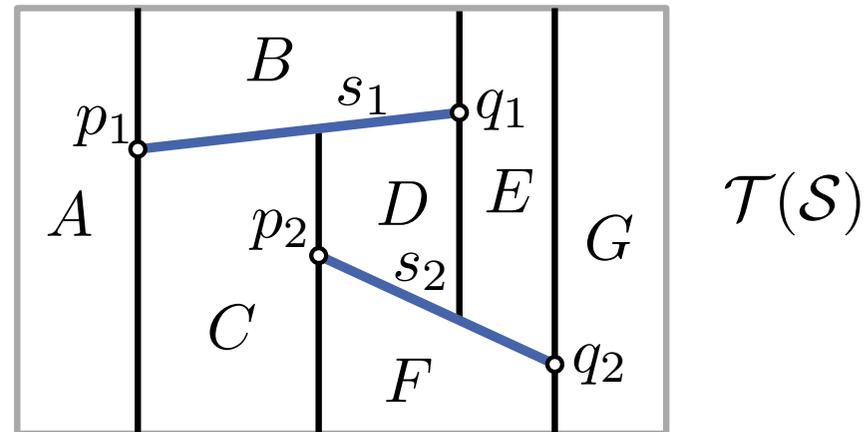
Vorlesung Algorithmische Geometrie

Voronoi-Diagramme

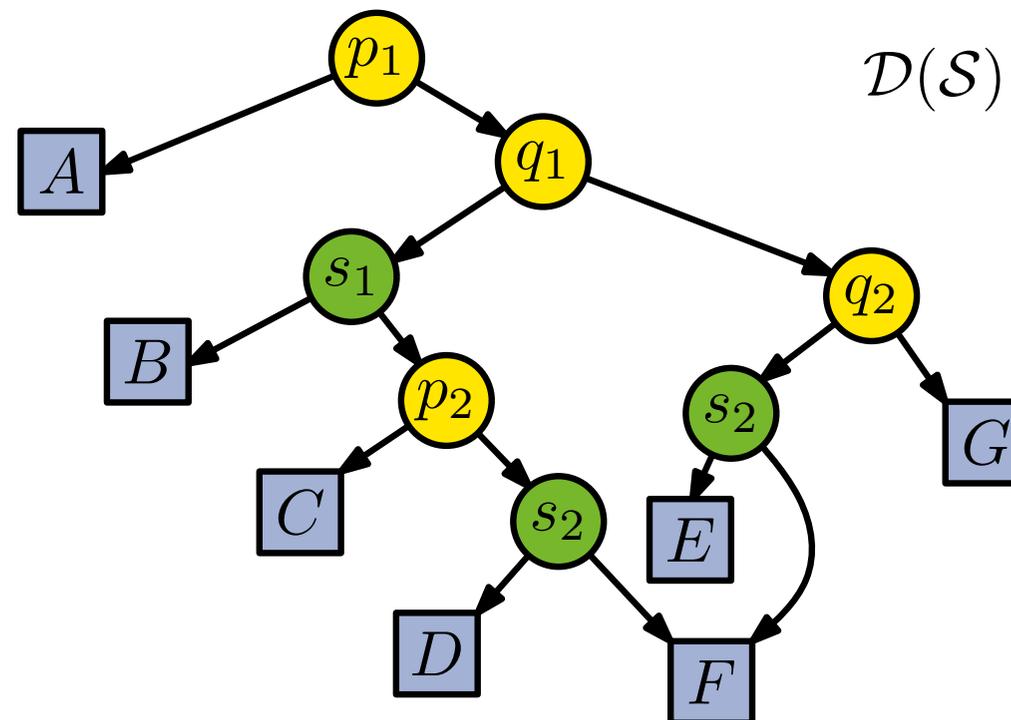
LEHRSTUHL FÜR ALGORITHMIK I · INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Martin Nöllenburg
31.05.2011





Nachtrag Trapezzerlegungen



Degenerierte Eingaben

Zwei Annahmen bislang:

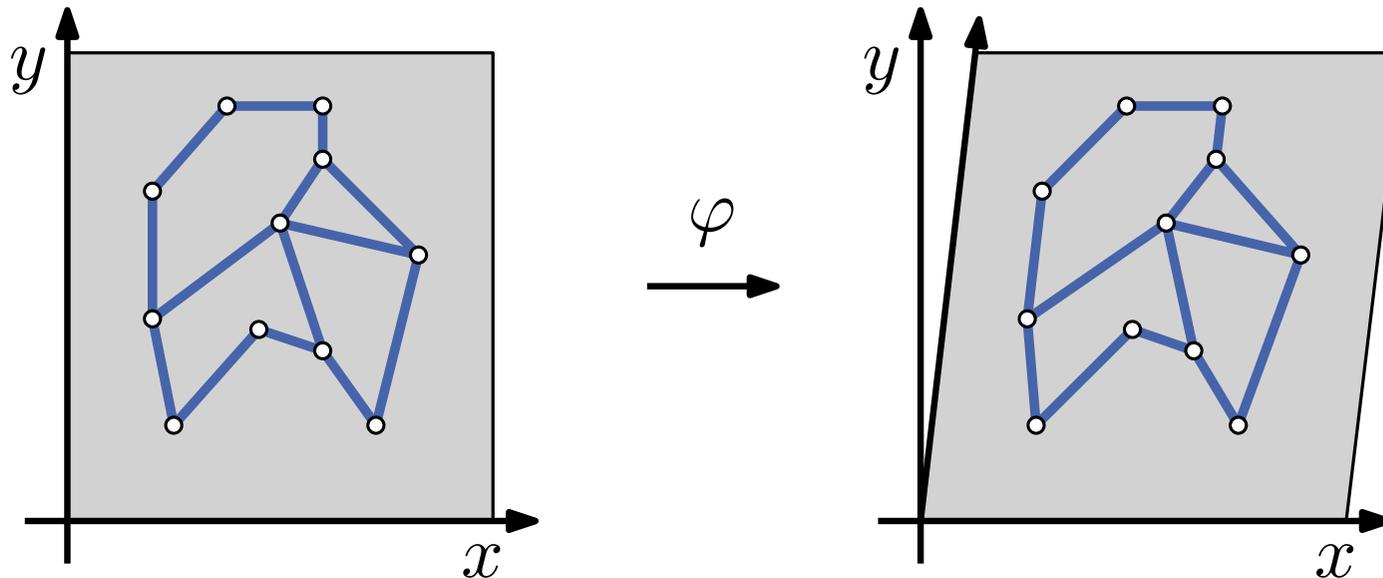
- keine zwei Streckenendpunkte habe gleiche x -Koordinate
- immer eindeutige Antworten (links/rechts) im Suchpfad

Degenerierte Eingaben

Zwei Annahmen bislang:

- keine zwei Streckenendpunkte habe gleiche x -Koordinate
- immer eindeutige Antworten (links/rechts) im Suchpfad

Ausweg: symbolische Scherung $\varphi : (x, y) \mapsto (x + \varepsilon y, y)$

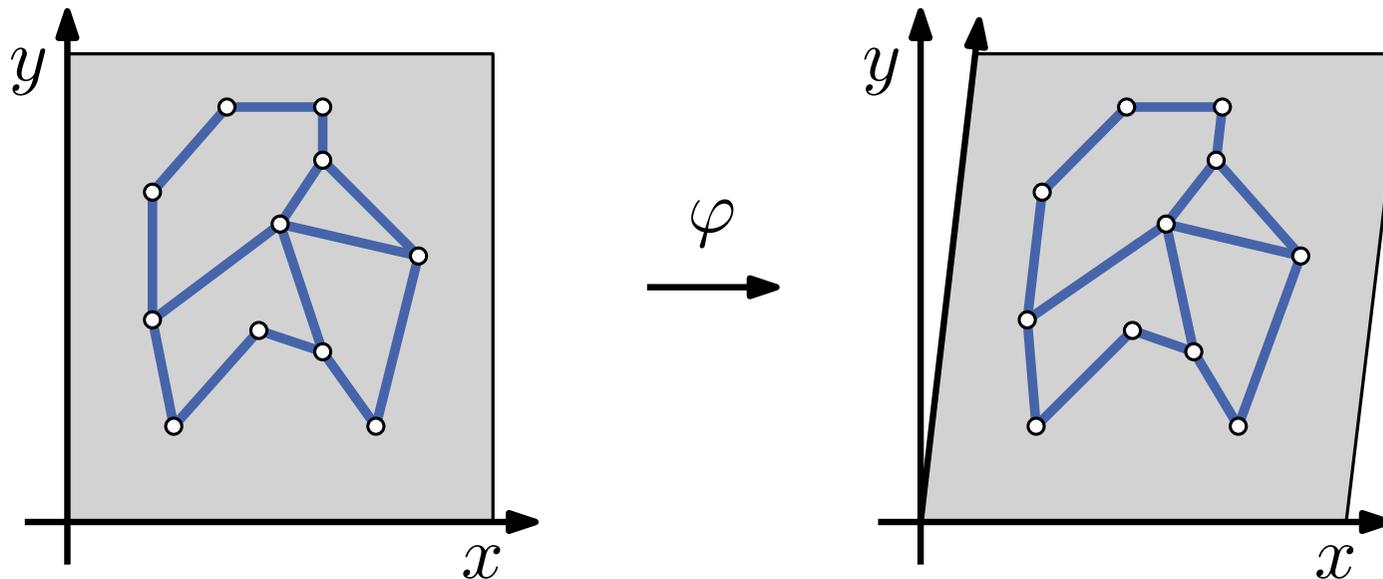


Degenerierte Eingaben

Zwei Annahmen bislang:

- keine zwei Streckenendpunkte habe gleiche x -Koordinate
- immer eindeutige Antworten (links/rechts) im Suchpfad

Ausweg: symbolische Scherung $\varphi : (x, y) \mapsto (x + \varepsilon y, y)$



Dabei ist $\varepsilon > 0$ so gewählt, dass sich die x -Ordnung < der Punkte nicht ändert.

Degenerierte Eingaben

- Effekt 1: lexikographische Ordnung
- Effekt 2: affine Abb. φ erhält Beziehung Punkt–Gerade

Degenerierte Eingaben

- Effekt 1: lexikographische Ordnung
- Effekt 2: affine Abb. φ erhält Beziehung Punkt–Gerade
- Führe Algorithmus für $\varphi\mathcal{S} = \{\varphi s \mid s \in \mathcal{S}\}$ und φp aus.

Degenerierte Eingaben

- Effekt 1: lexikographische Ordnung
- Effekt 2: affine Abb. φ erhält Beziehung Punkt–Gerade
- Führe Algorithmus für $\varphi\mathcal{S} = \{\varphi s \mid s \in \mathcal{S}\}$ und φp aus.
- Zwei elementare Operationen beim Aufbau von \mathcal{T} und \mathcal{D} :
 1. liegt q links oder rechts der Vertikalen durch p ?
 2. liegt q oberhalb oder unterhalb der Strecke s ?

Degenerierte Eingaben

- Effekt 1: lexikographische Ordnung
- Effekt 2: affine Abb. φ erhält Beziehung Punkt–Gerade
- Führe Algorithmus für $\varphi\mathcal{S} = \{\varphi s \mid s \in \mathcal{S}\}$ und φp aus.
- Zwei elementare Operationen beim Aufbau von \mathcal{T} und \mathcal{D} :
 1. liegt q links oder rechts der Vertikalen durch p ?
 2. liegt q oberhalb oder unterhalb der Strecke s ?
- Auch Lokalisierung eines Punktes q in $\mathcal{T}(\mathcal{S})$ geht durch Lokalisierung von φq in $\mathcal{T}(\varphi\mathcal{S})$.

Worst-Case Abschätzung

Bislang: erwartete Anfragezeit für bel. Punkt ist $O(\log n)$

Aber: jede Permutation könnte einen sehr schlechten (worst-case) Anfragepunkt haben

Worst-Case Abschätzung

Bislang: erwartete Anfragezeit für bel. Punkt ist $O(\log n)$

Aber: jede Permutation könnte einen sehr schlechten (worst-case) Anfragepunkt haben

Lemma 1: Sei S Menge von n kreuzungsfreien Strecken, q Anfragepunkt und $\lambda > 0$. Dann gilt
$$\Pr[\text{Suchpfad für } q \text{ länger als } 3\lambda \ln(n + 1)] \leq 1/(n + 1)^{\lambda \ln 1.25 - 1}.$$

Worst-Case Abschätzung

Bislang: erwartete Anfragezeit für bel. Punkt ist $O(\log n)$

Aber: jede Permutation könnte einen sehr schlechten (worst-case) Anfragepunkt haben

Lemma 1: Sei \mathcal{S} Menge von n kreuzungsfreien Strecken, q Anfragepunkt und $\lambda > 0$. Dann gilt
$$\Pr[\text{Suchpfad für } q \text{ länger als } 3\lambda \ln(n+1)] \leq \frac{1}{(n+1)^{\lambda \ln 1.25 - 1}}.$$

Lemma 2: Sei \mathcal{S} Menge von n kreuzungsfreien Strecken und $\lambda > 0$. Dann gilt
$$\Pr[\text{max. Suchpfad in } \mathcal{D} \text{ länger als } 3\lambda \ln(n+1)] \leq \frac{2}{(n+1)^{\lambda \ln 1.25 - 3}}.$$

Worst-Case Abschätzung

Bislang: erwartete Anfragezeit für bel. Punkt ist $O(\log n)$

Aber: jede Permutation könnte einen sehr schlechten (worst-case) Anfragepunkt haben

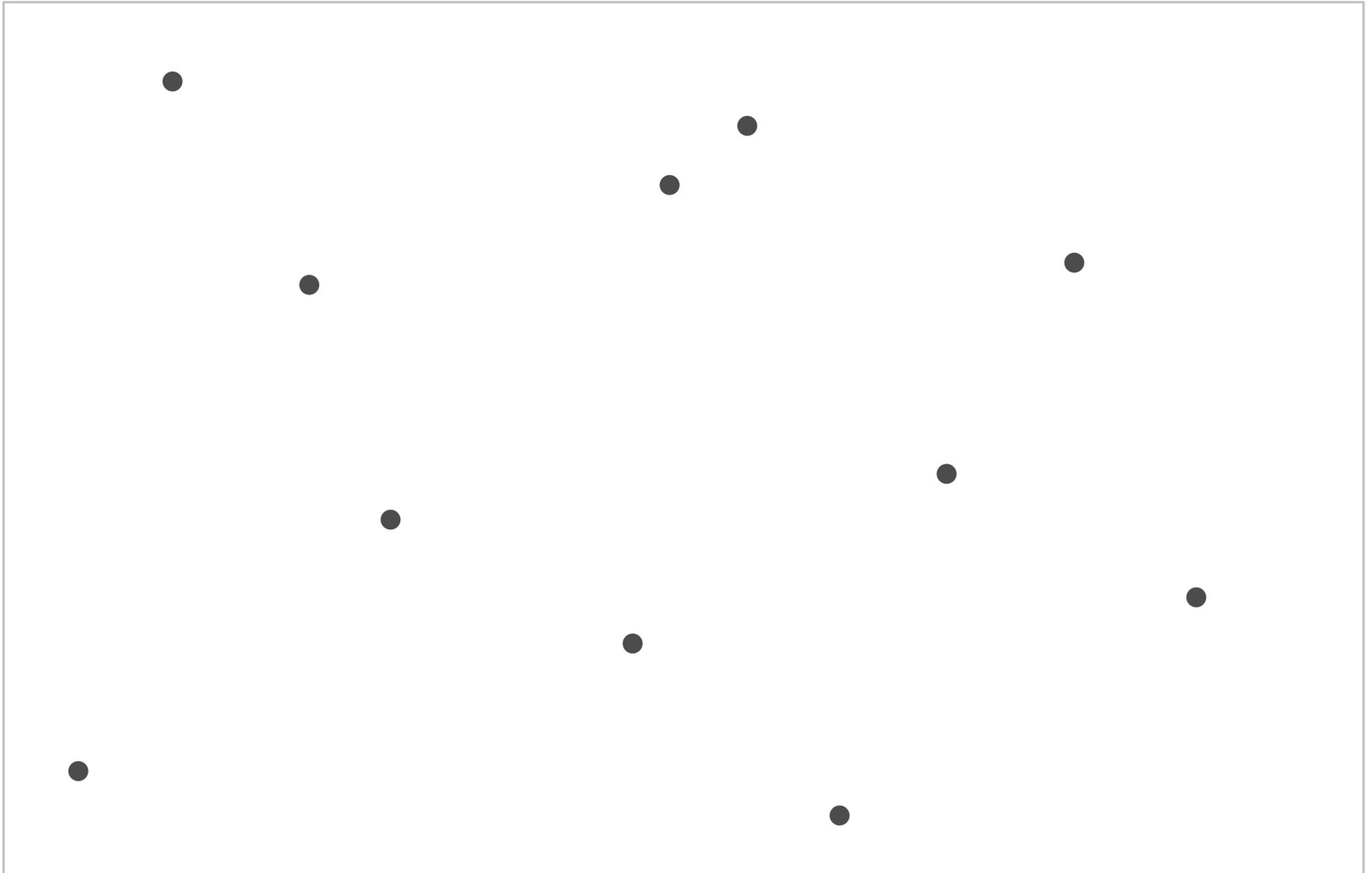
Lemma 1: Sei \mathcal{S} Menge von n kreuzungsfreien Strecken, q Anfragepunkt und $\lambda > 0$. Dann gilt
 $\Pr[\text{Suchpfad für } q \text{ länger als } 3\lambda \ln(n+1)] \leq 1/(n+1)^{\lambda \ln 1.25 - 1}$.

Lemma 2: Sei \mathcal{S} Menge von n kreuzungsfreien Strecken und $\lambda > 0$. Dann gilt
 $\Pr[\text{max. Suchpfad in } \mathcal{D} \text{ länger als } 3\lambda \ln(n+1)] \leq 2/(n+1)^{\lambda \ln 1.25 - 3}$.

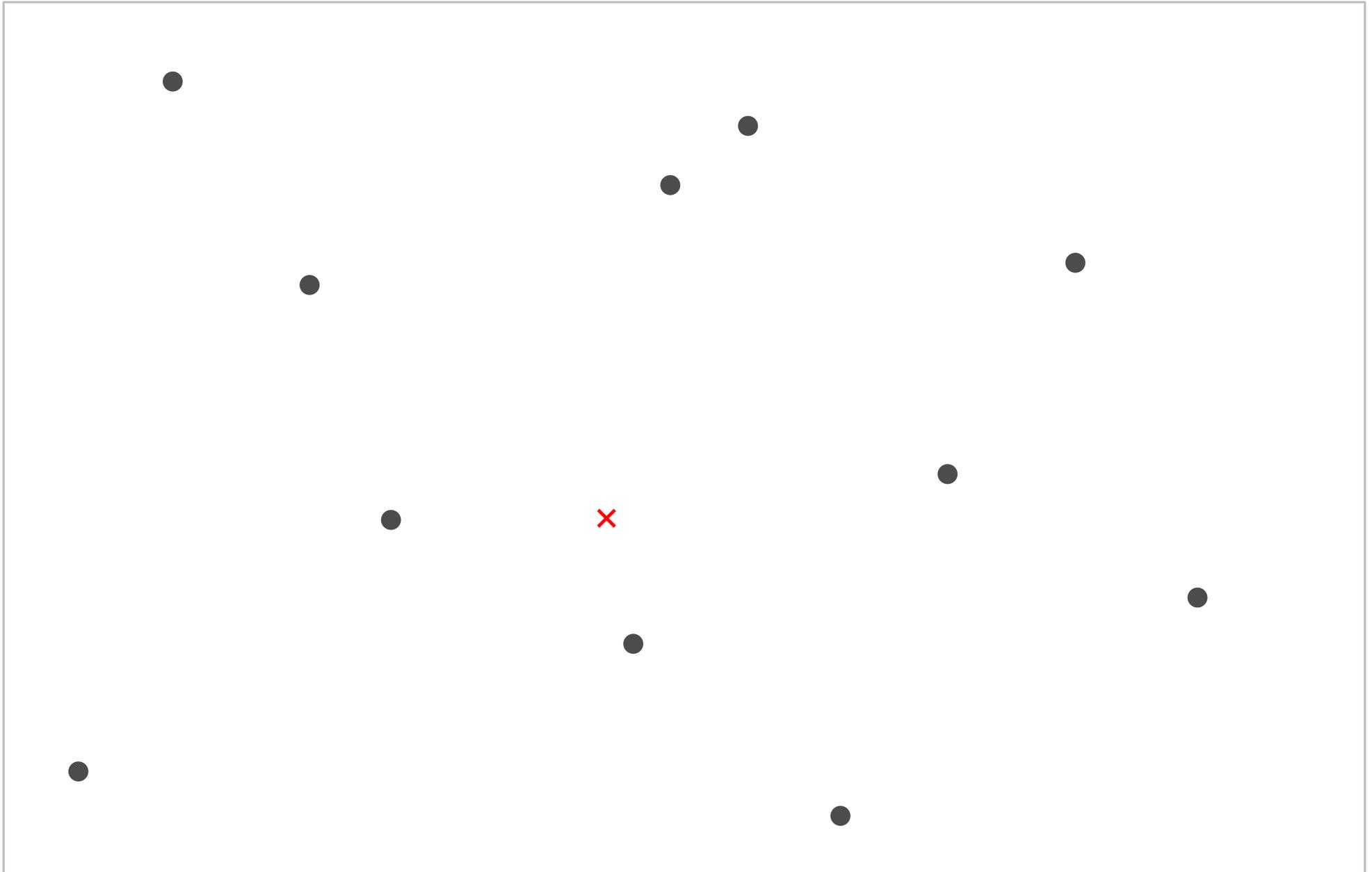
Satz: Sei \mathcal{S} Unterteilung der Ebene mit n Kanten. Es gibt eine Suchstruktur zur Punktlokalisierung in \mathcal{S} mit Platzbedarf $O(n)$ und Anfragezeit $O(\log n)$.

Voronoi-Diagramme

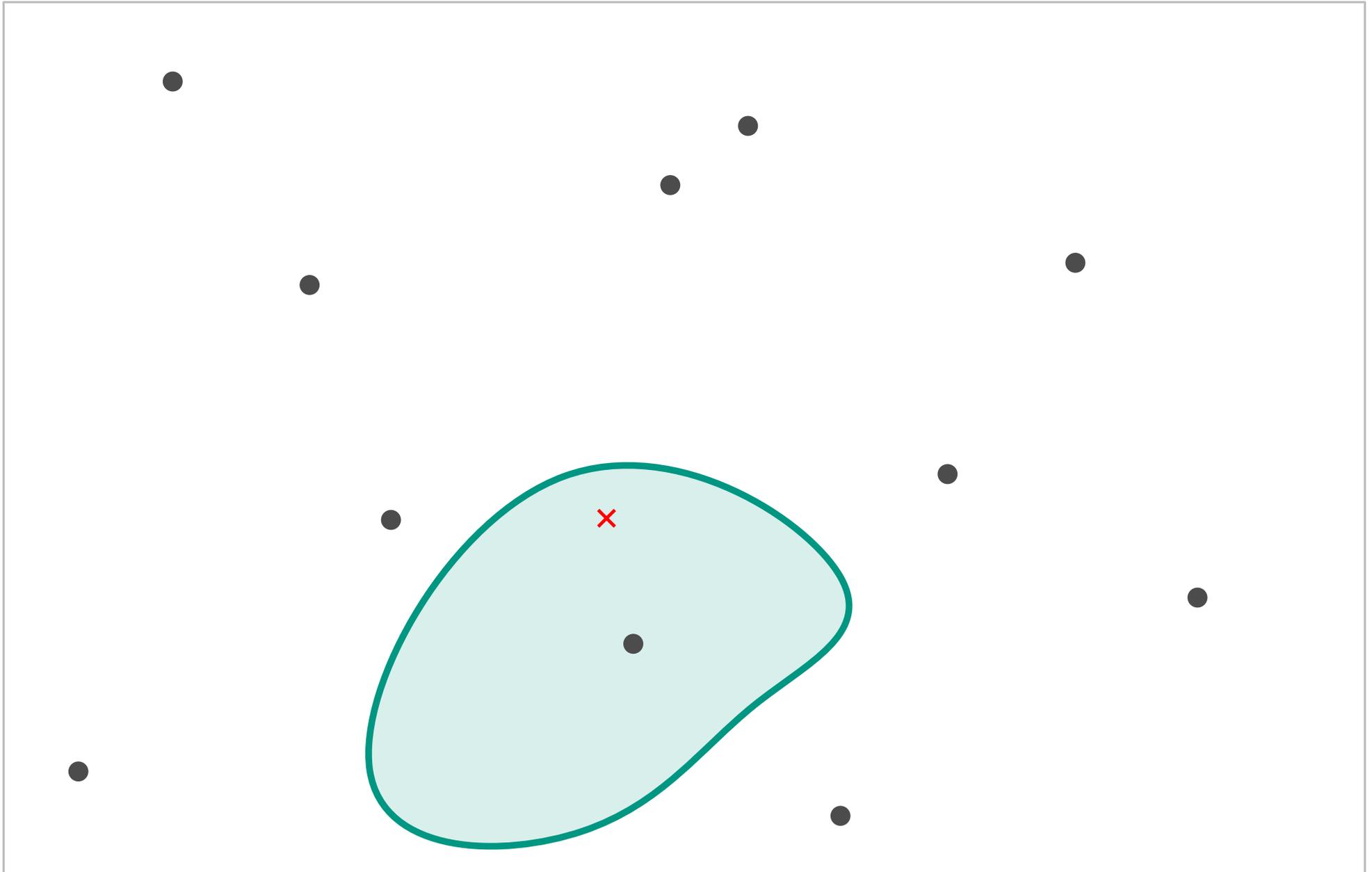
Das Postamt-Problem



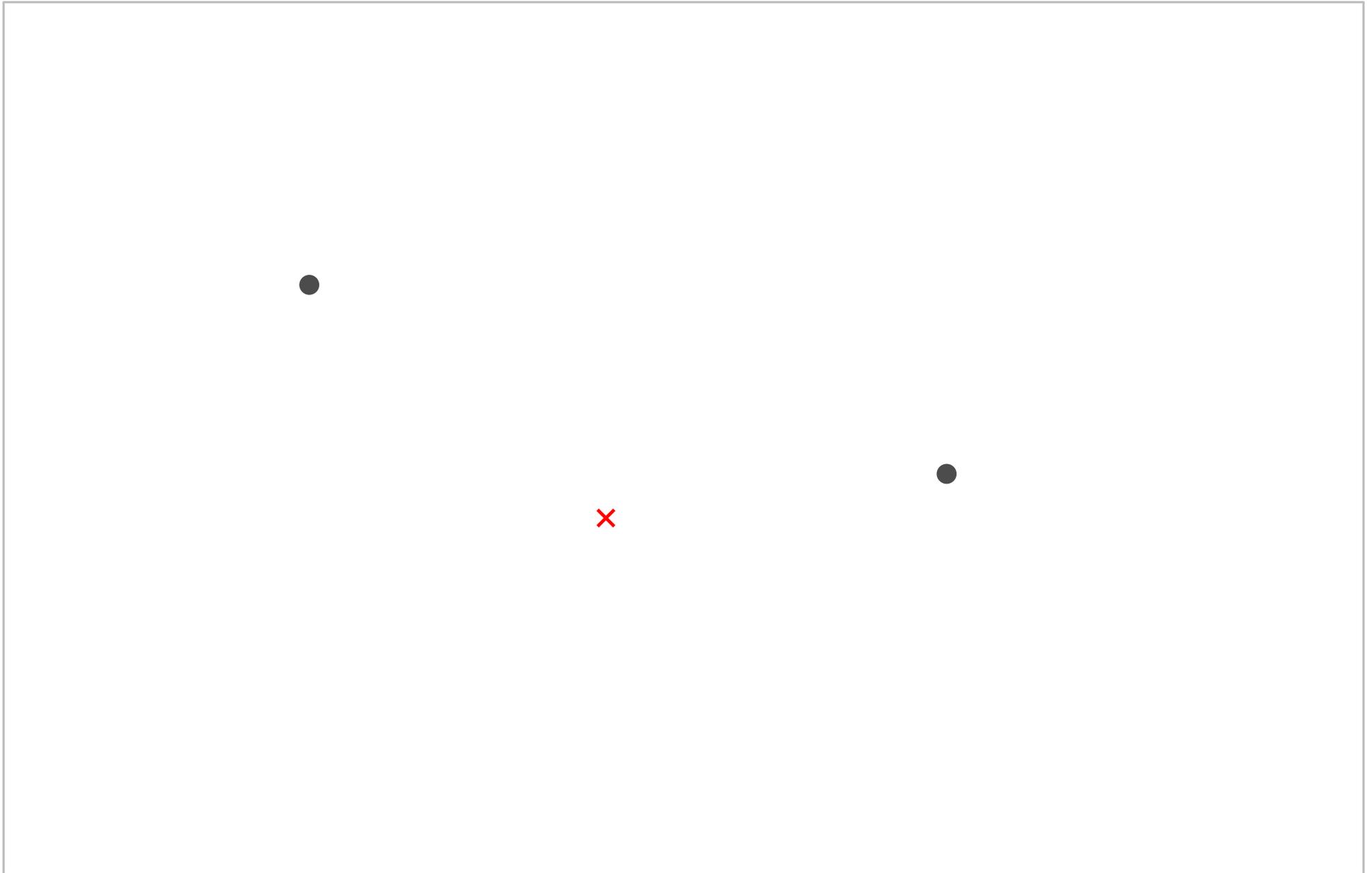
Das Postamt-Problem



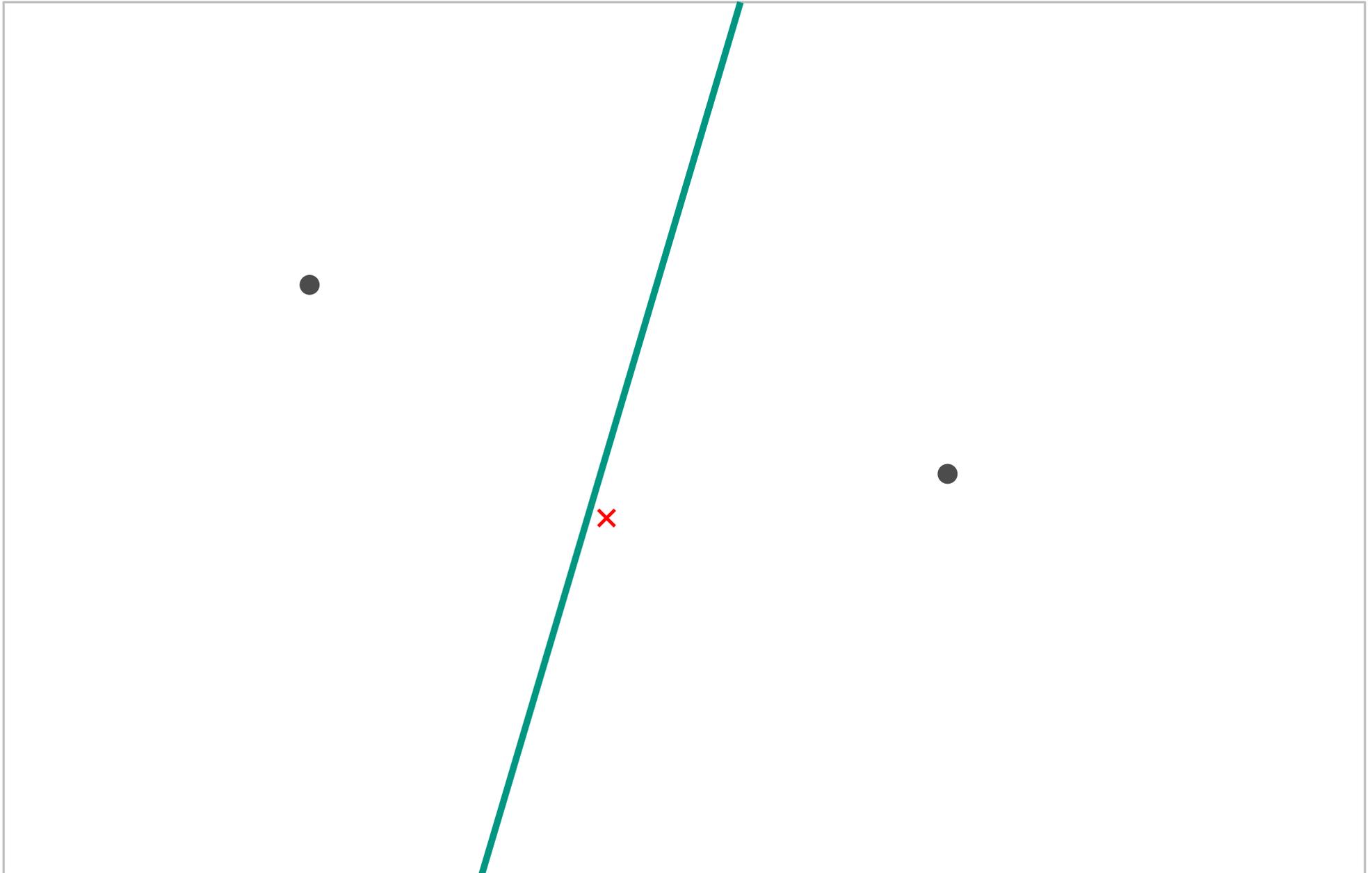
Das Postamt-Problem



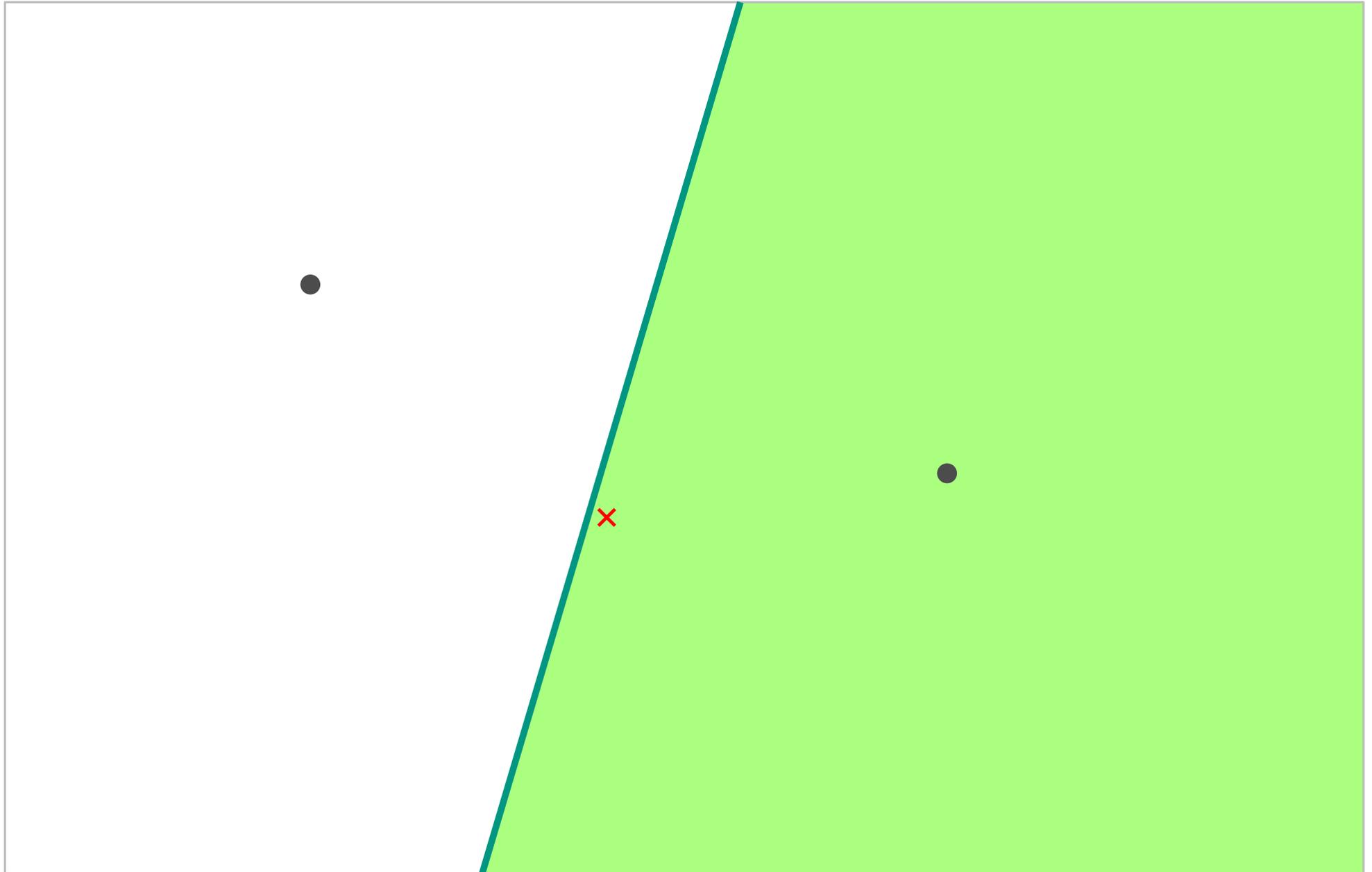
Das Postamt-Problem



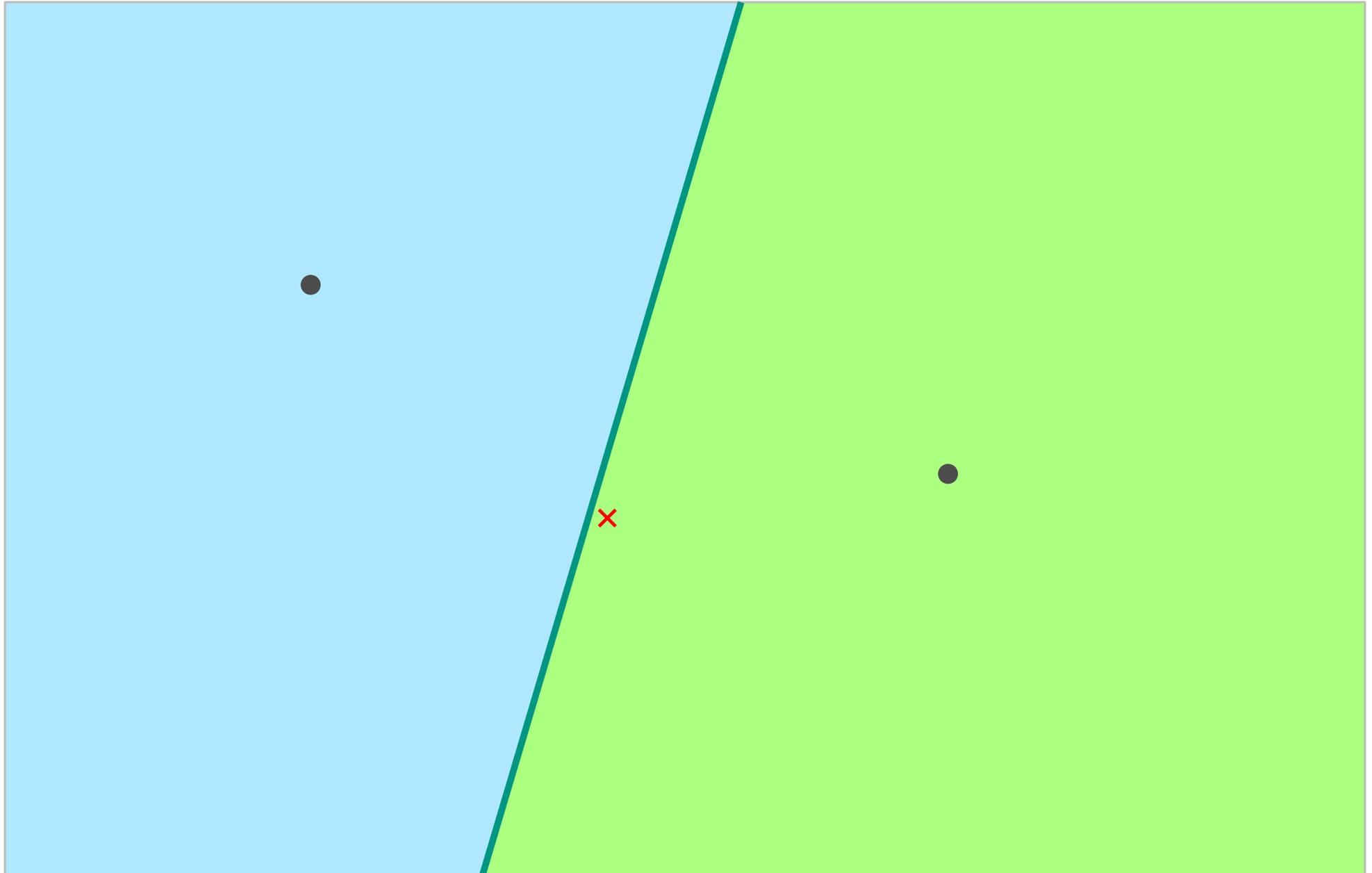
Das Postamt-Problem



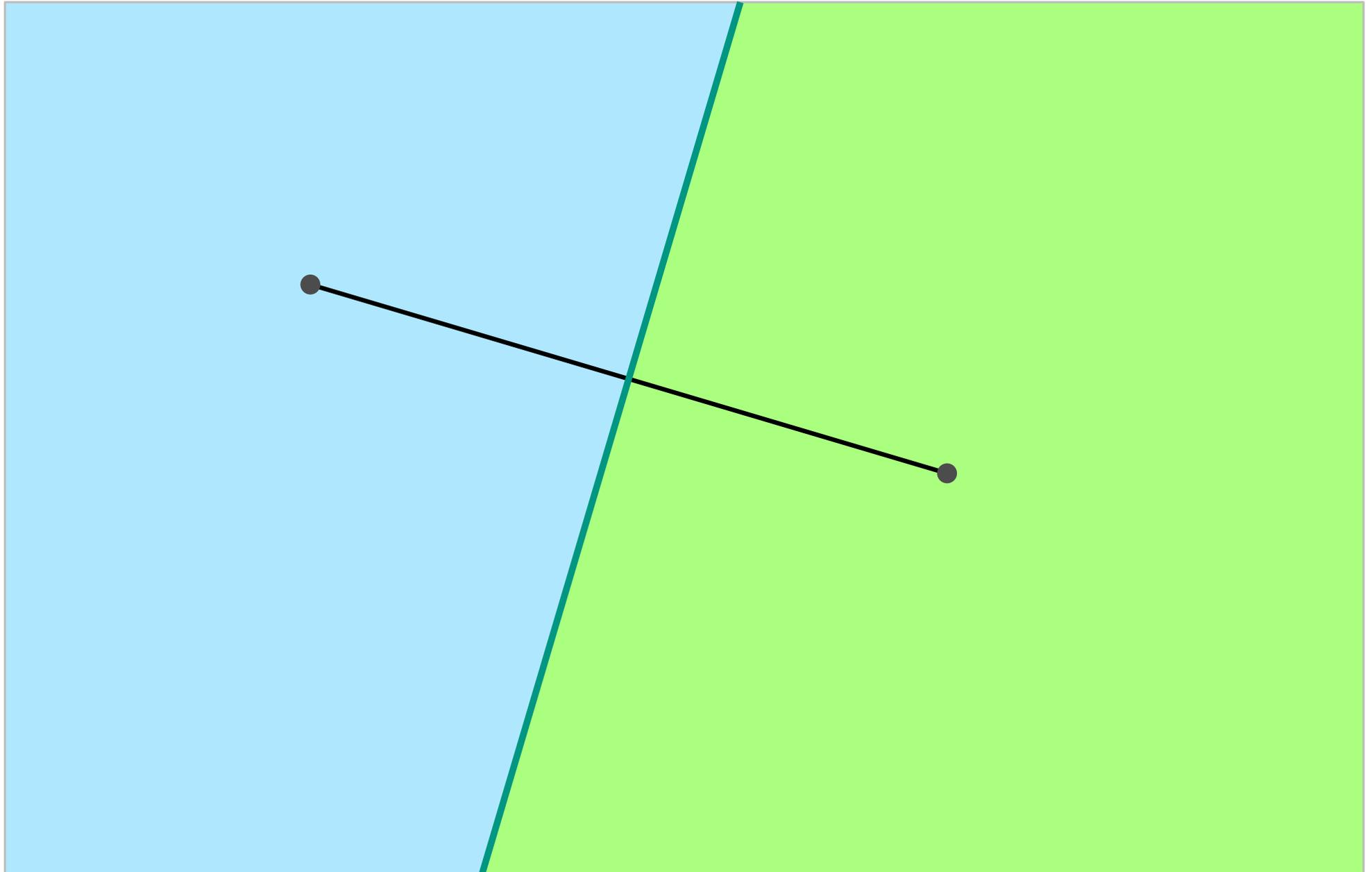
Das Postamt-Problem



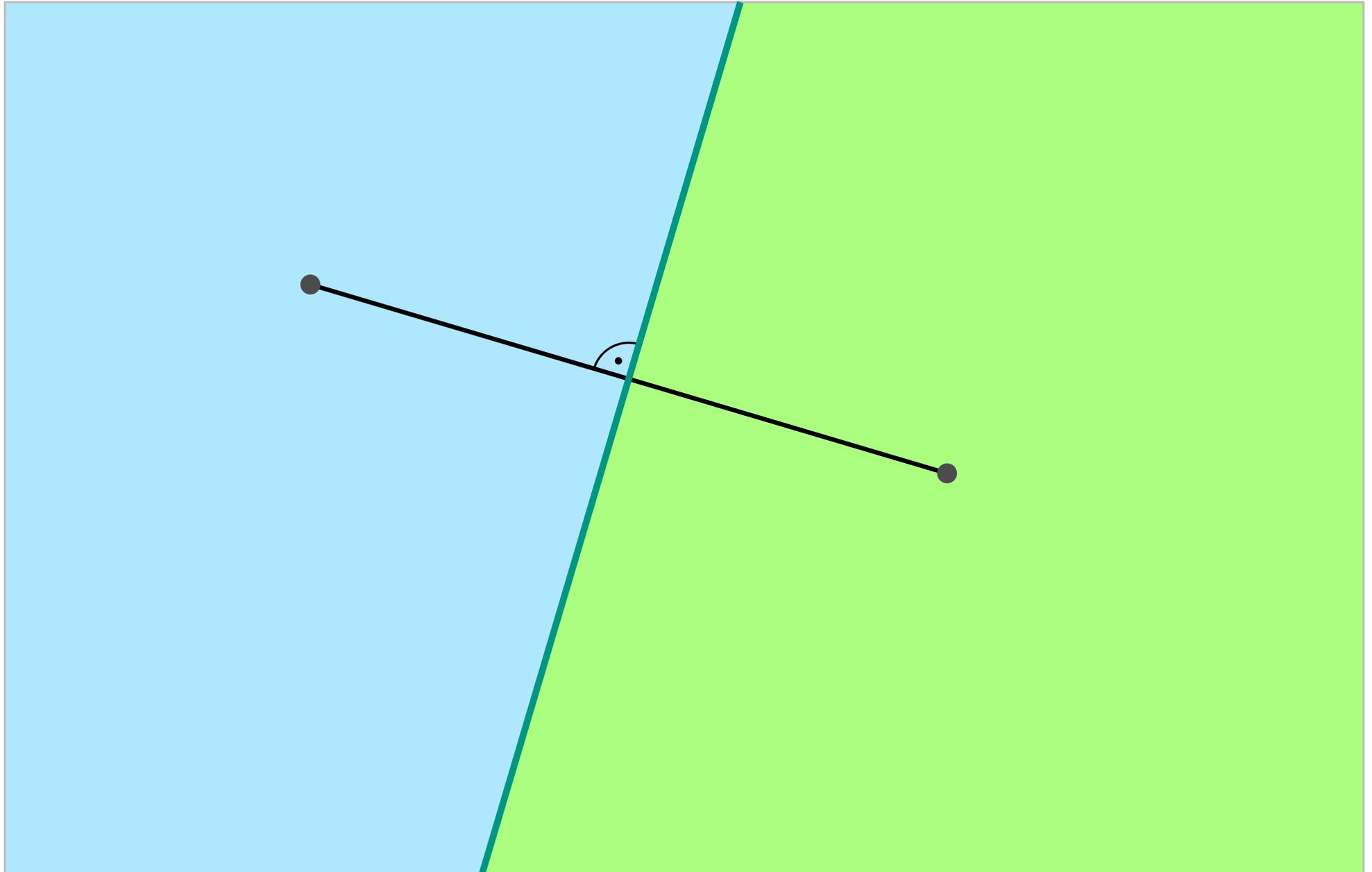
Das Postamt-Problem



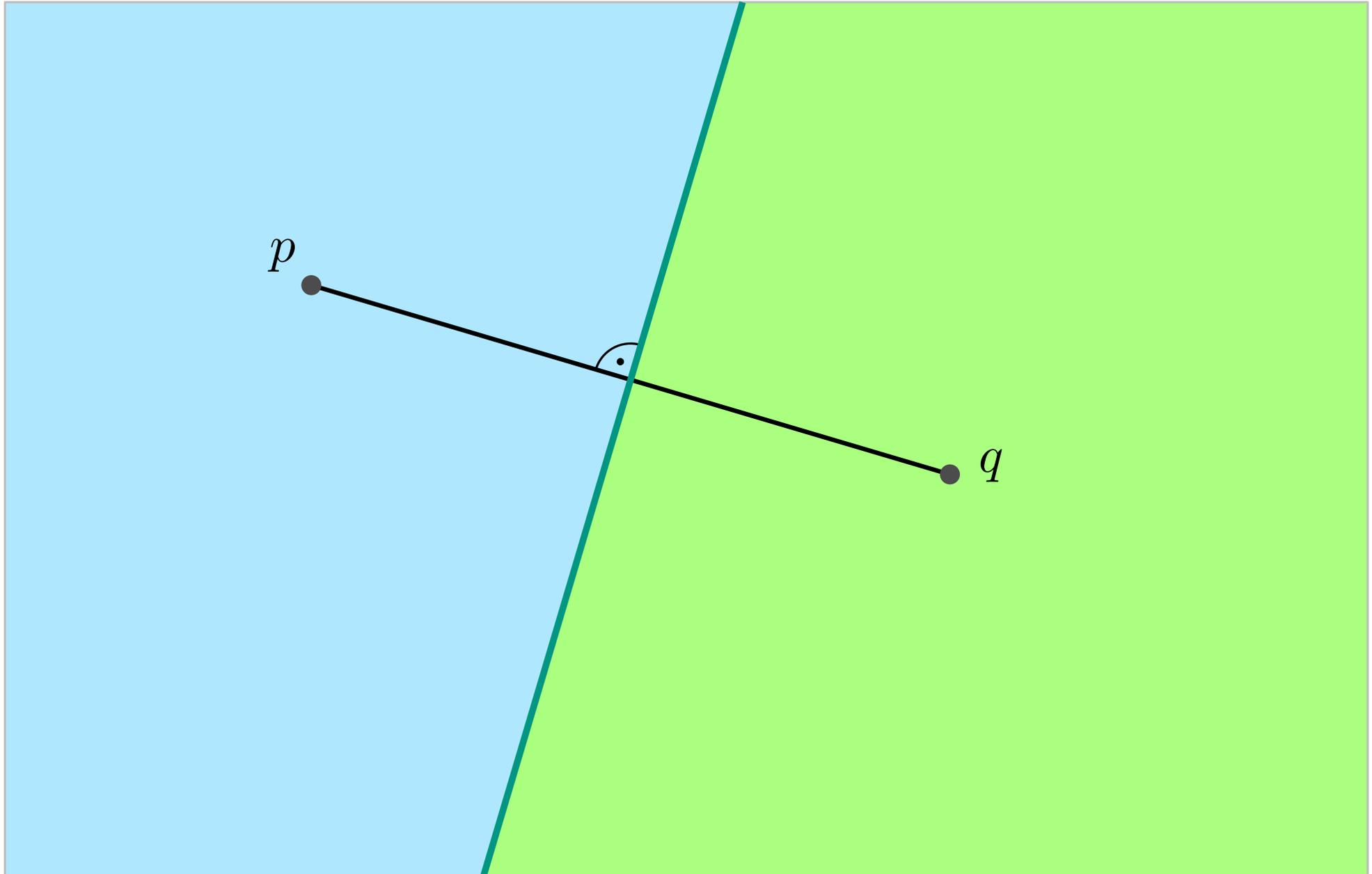
Das Postamt-Problem



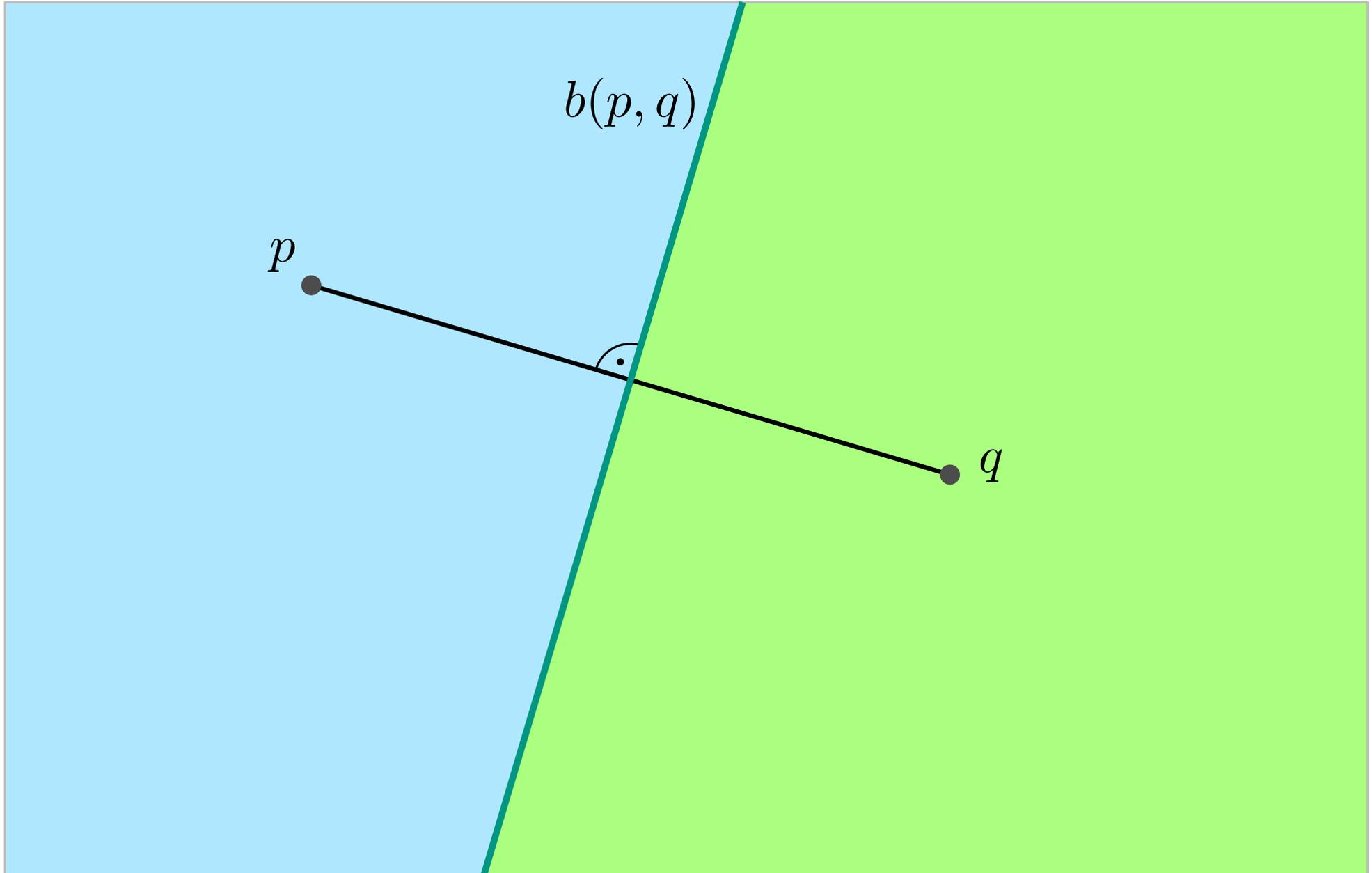
Das Postamt-Problem

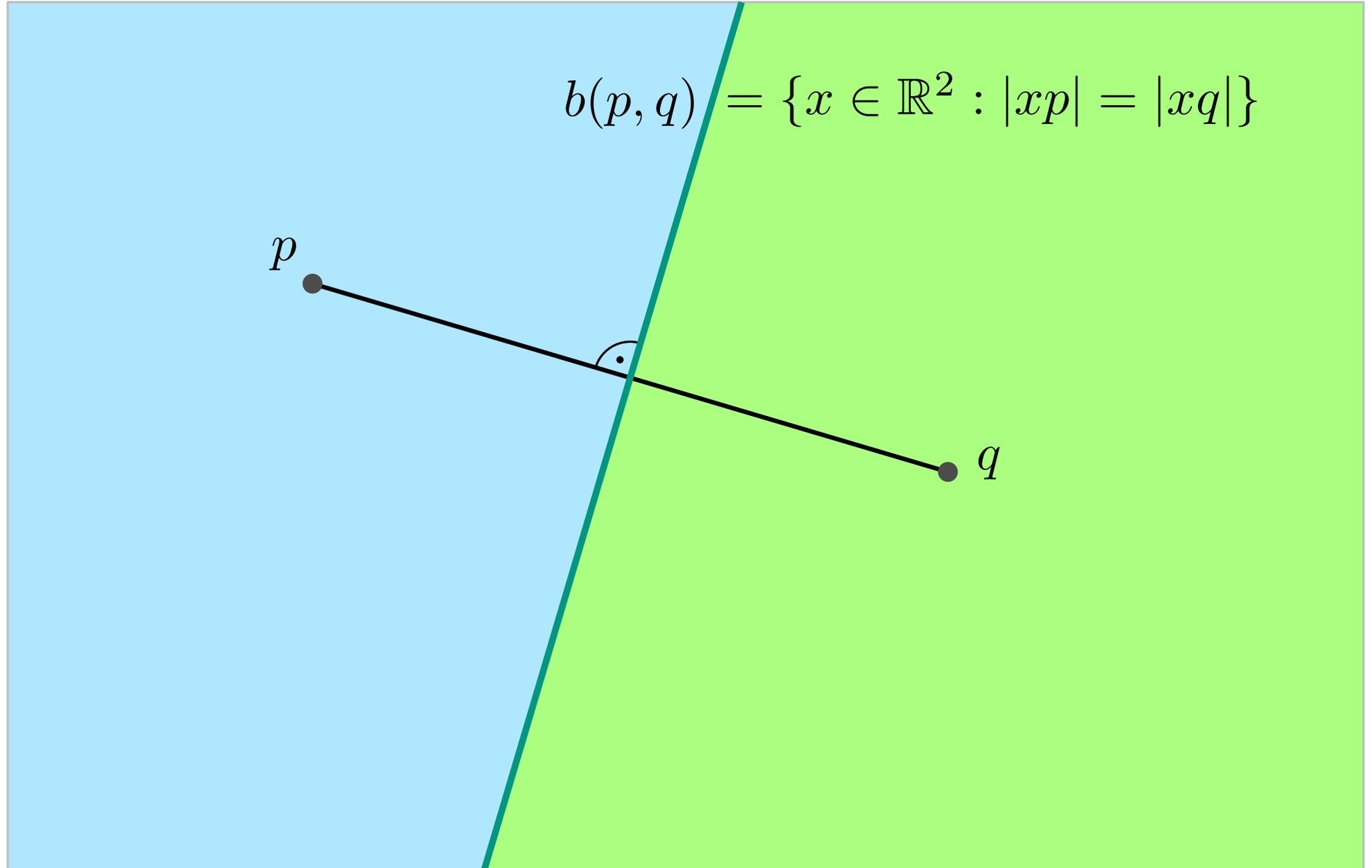


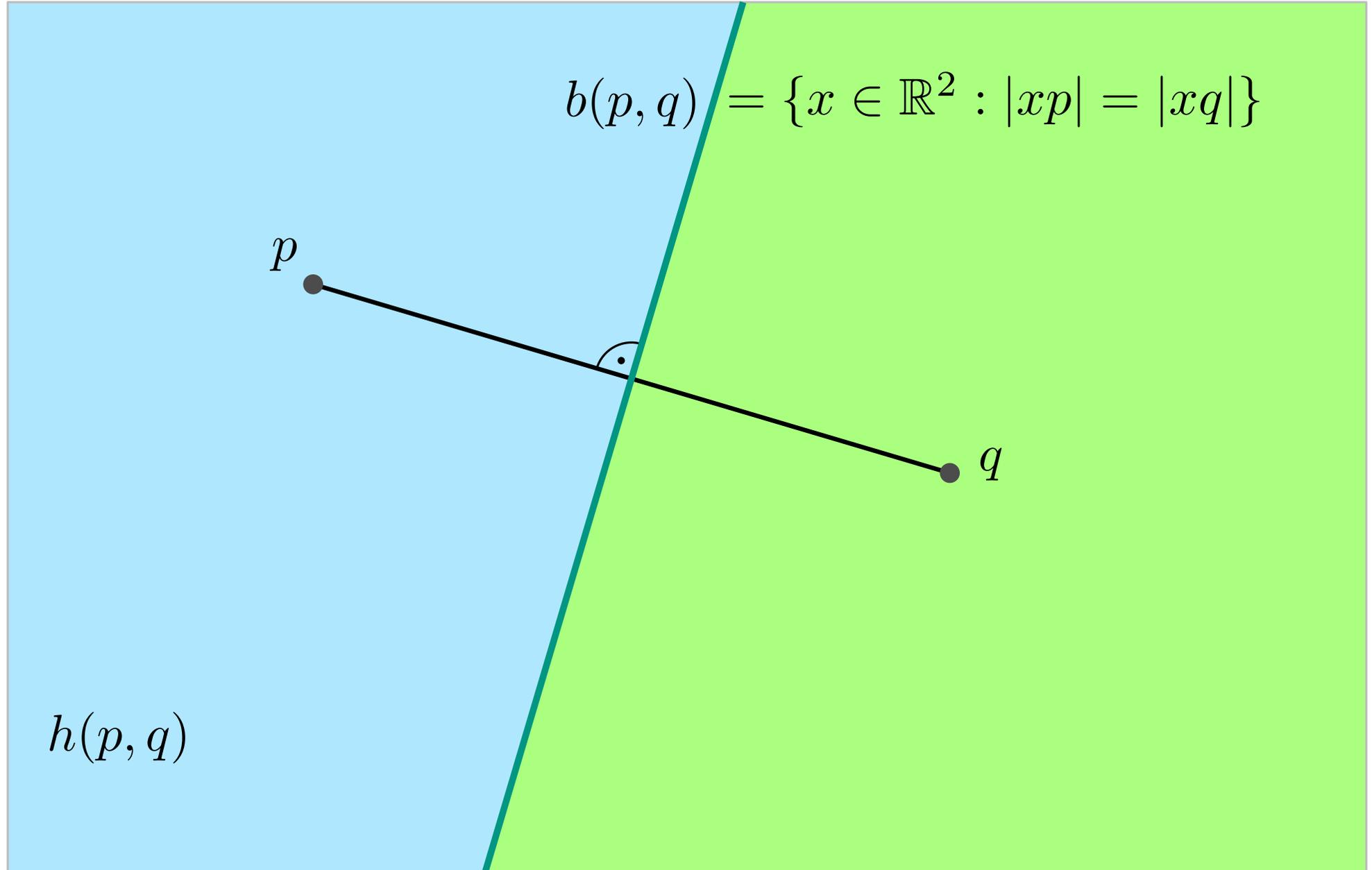
Das Postamt-Problem

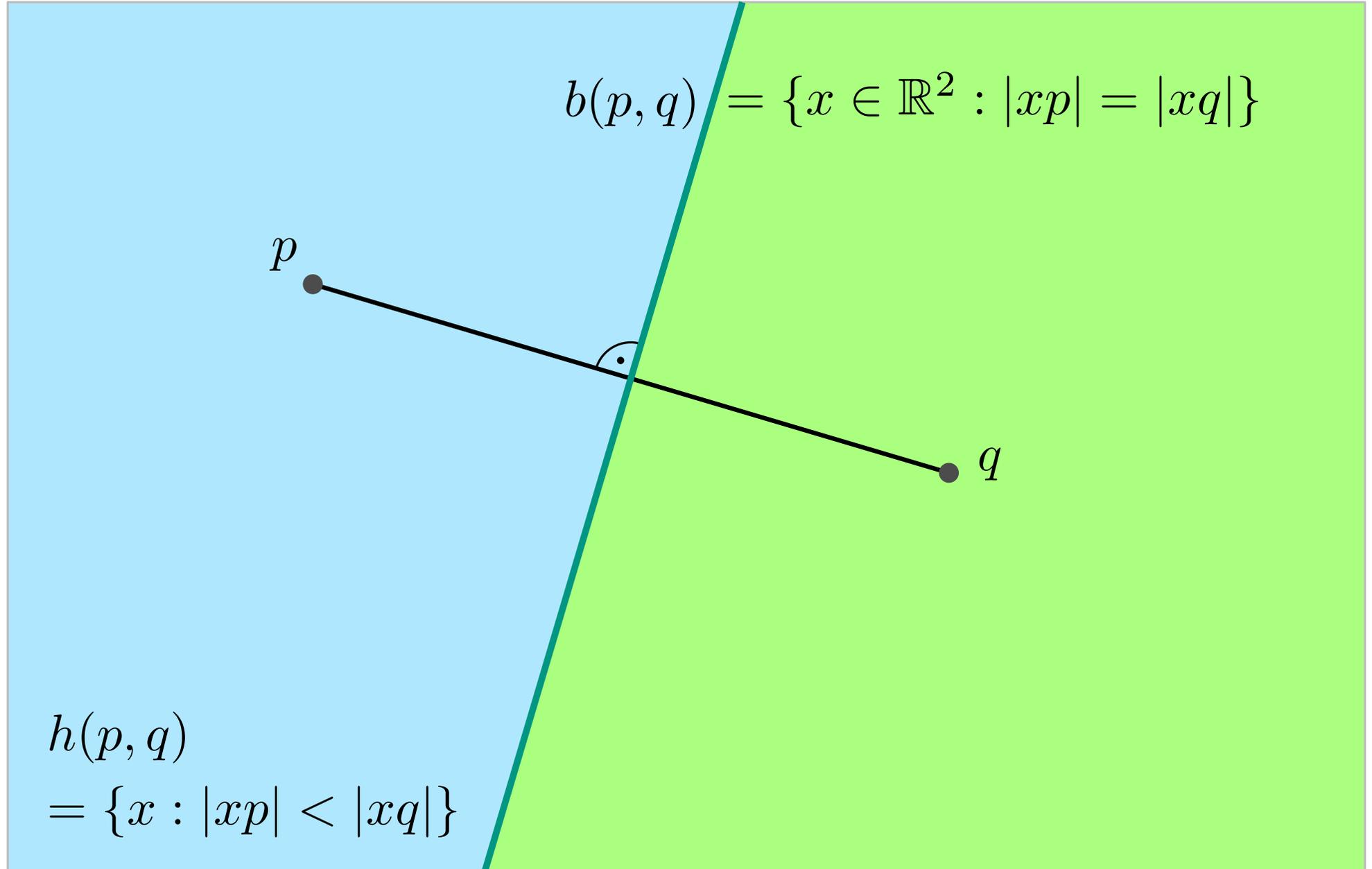


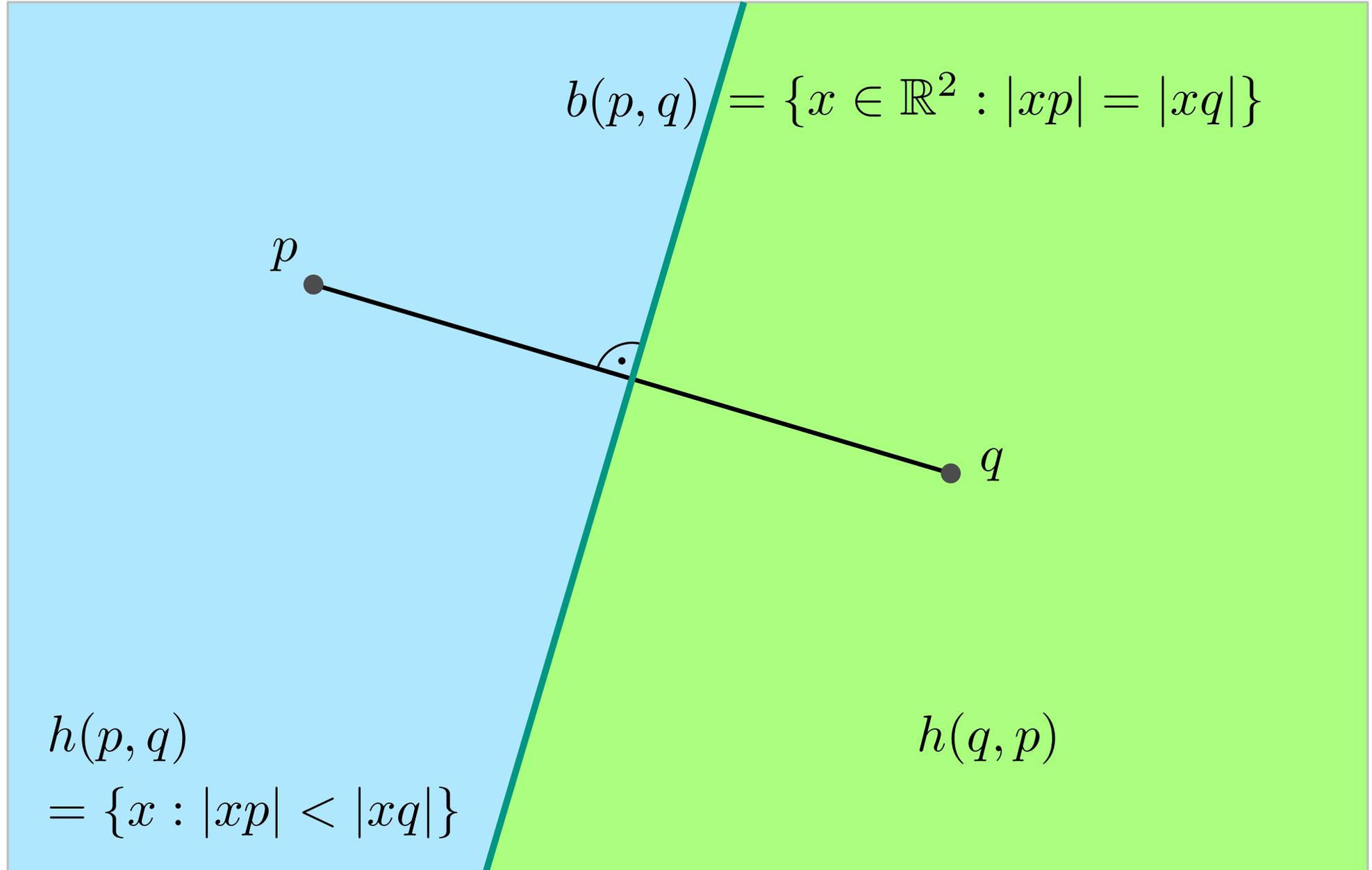
Das Postamt-Problem

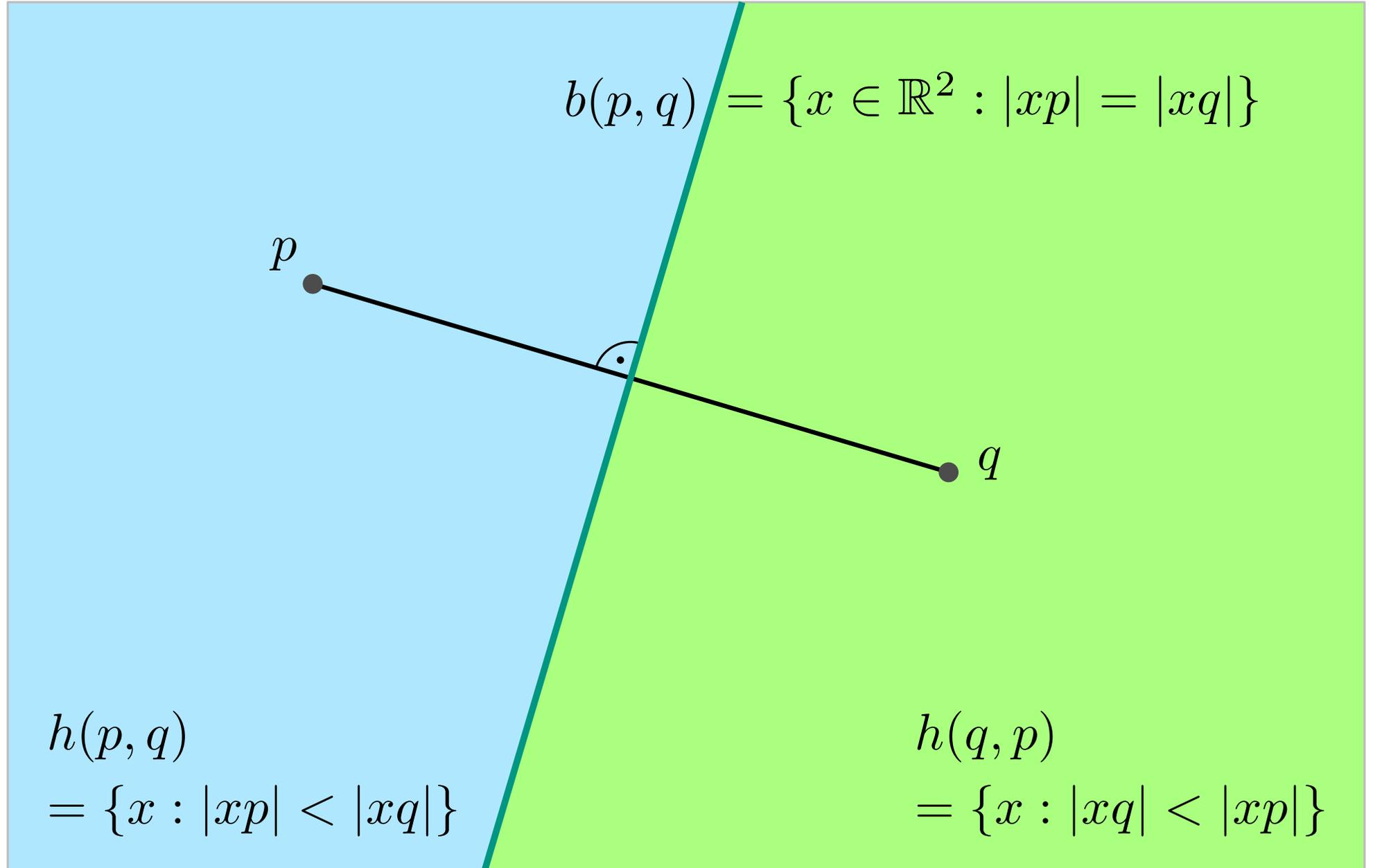




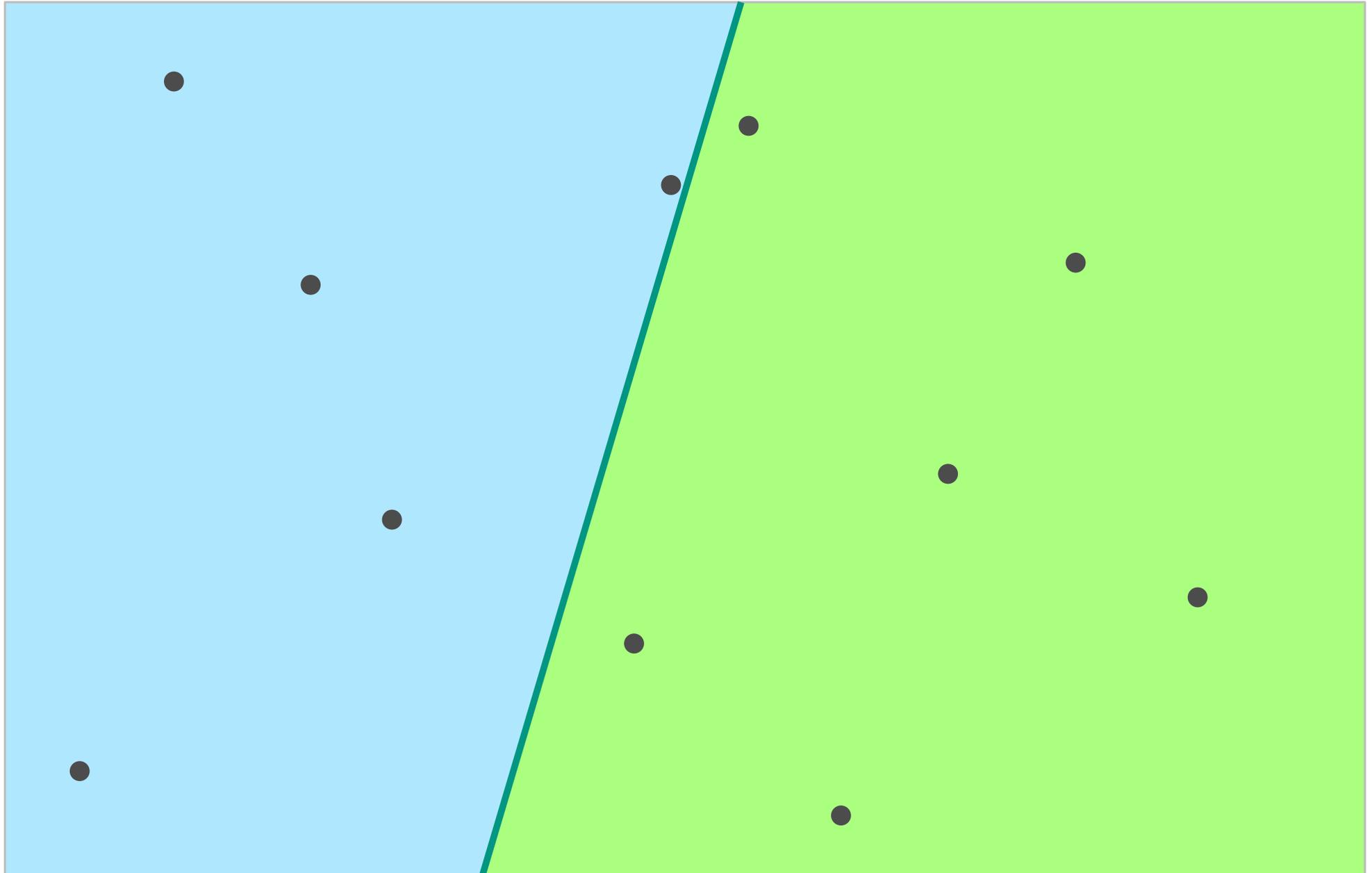






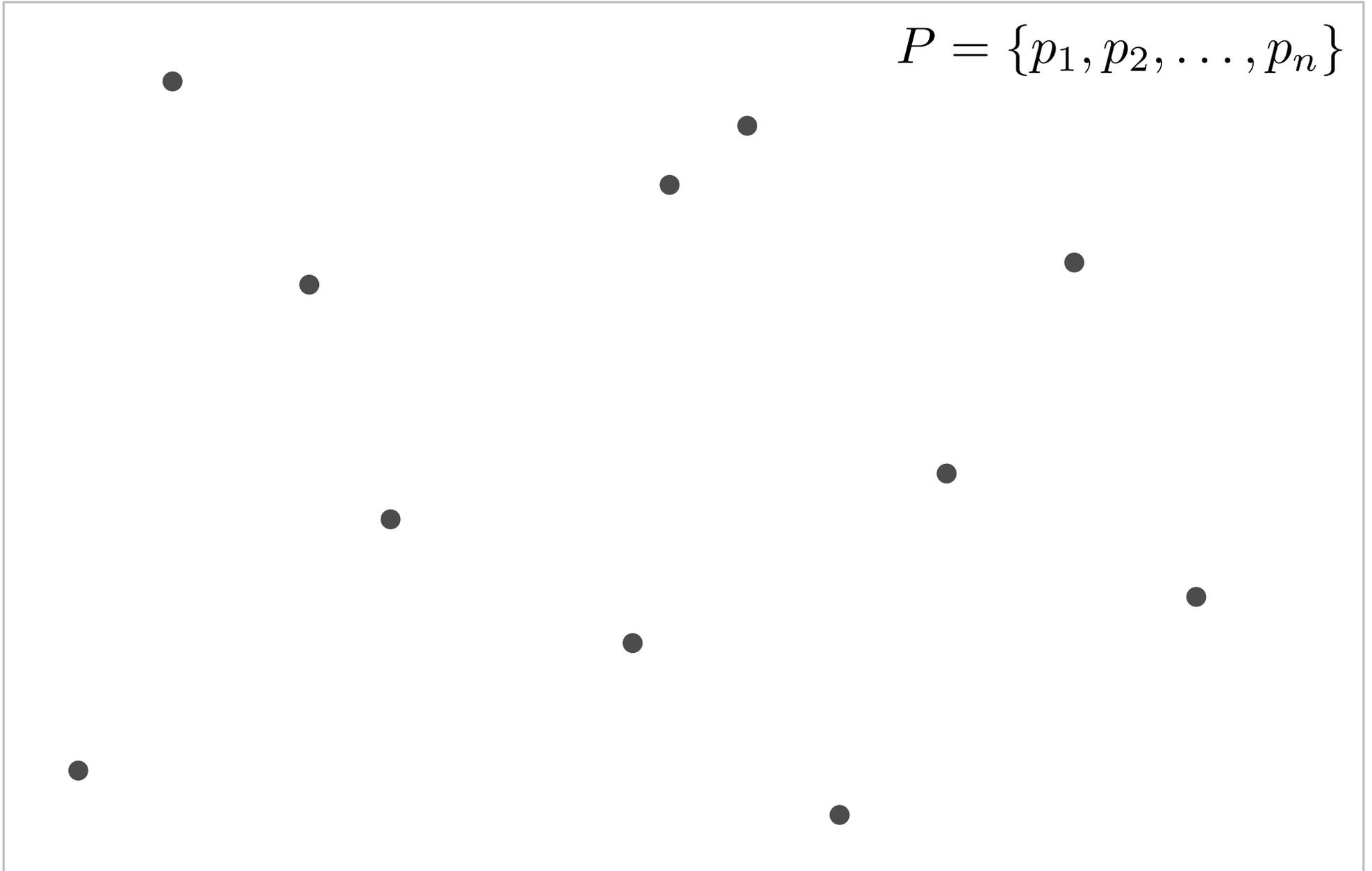


Das Postamt-Problem

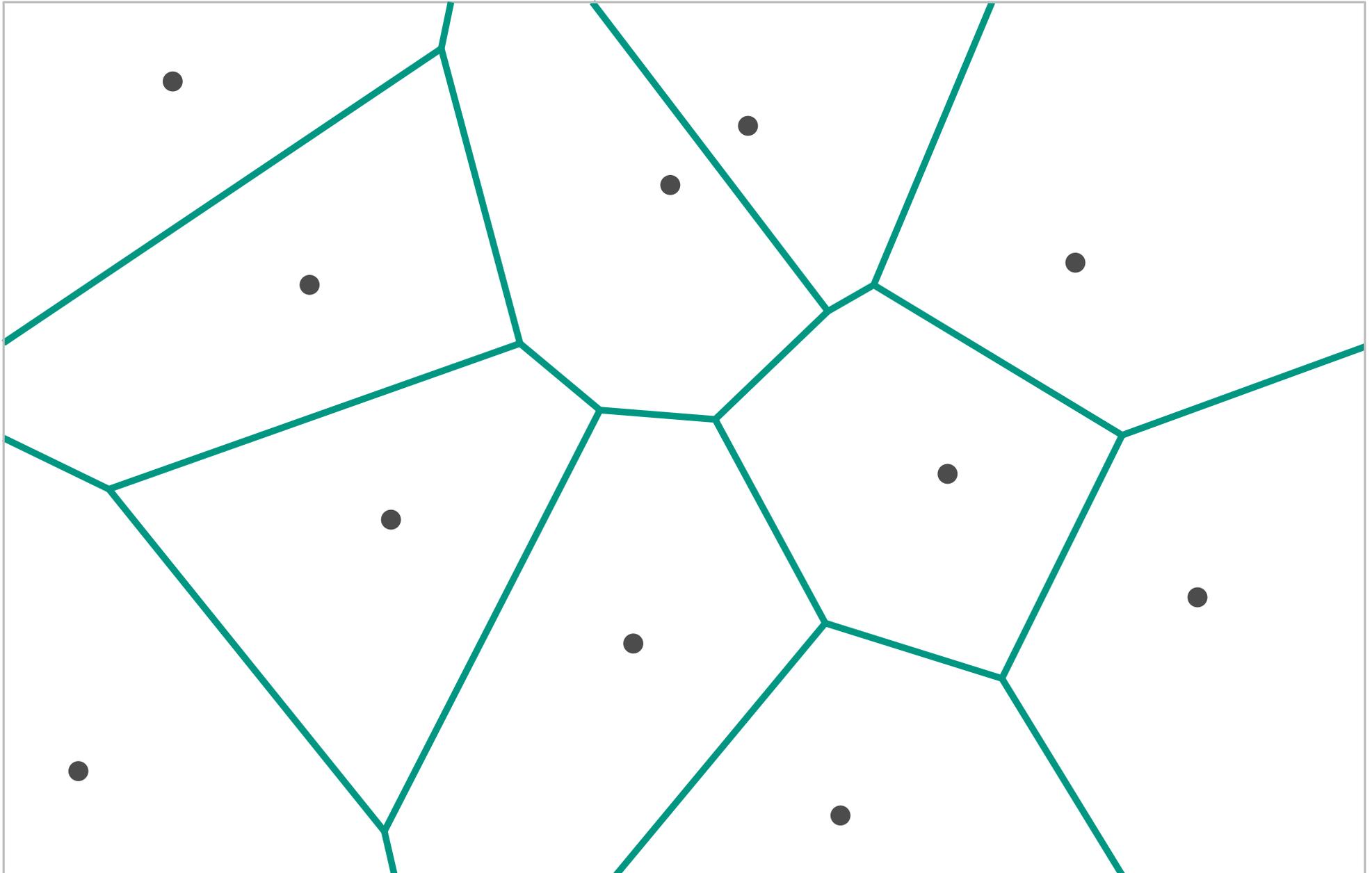


Das Postamt-Problem

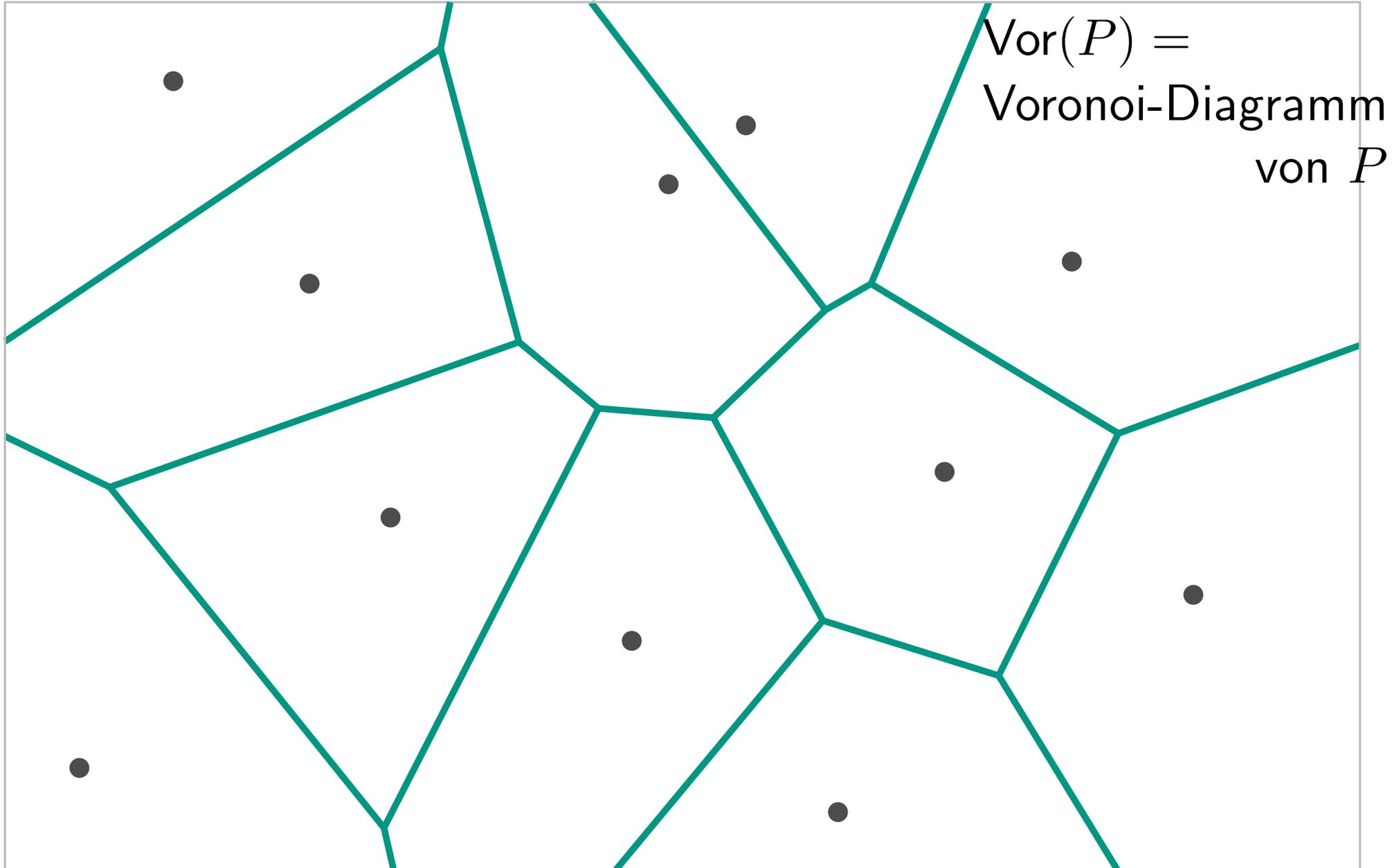
$$P = \{p_1, p_2, \dots, p_n\}$$



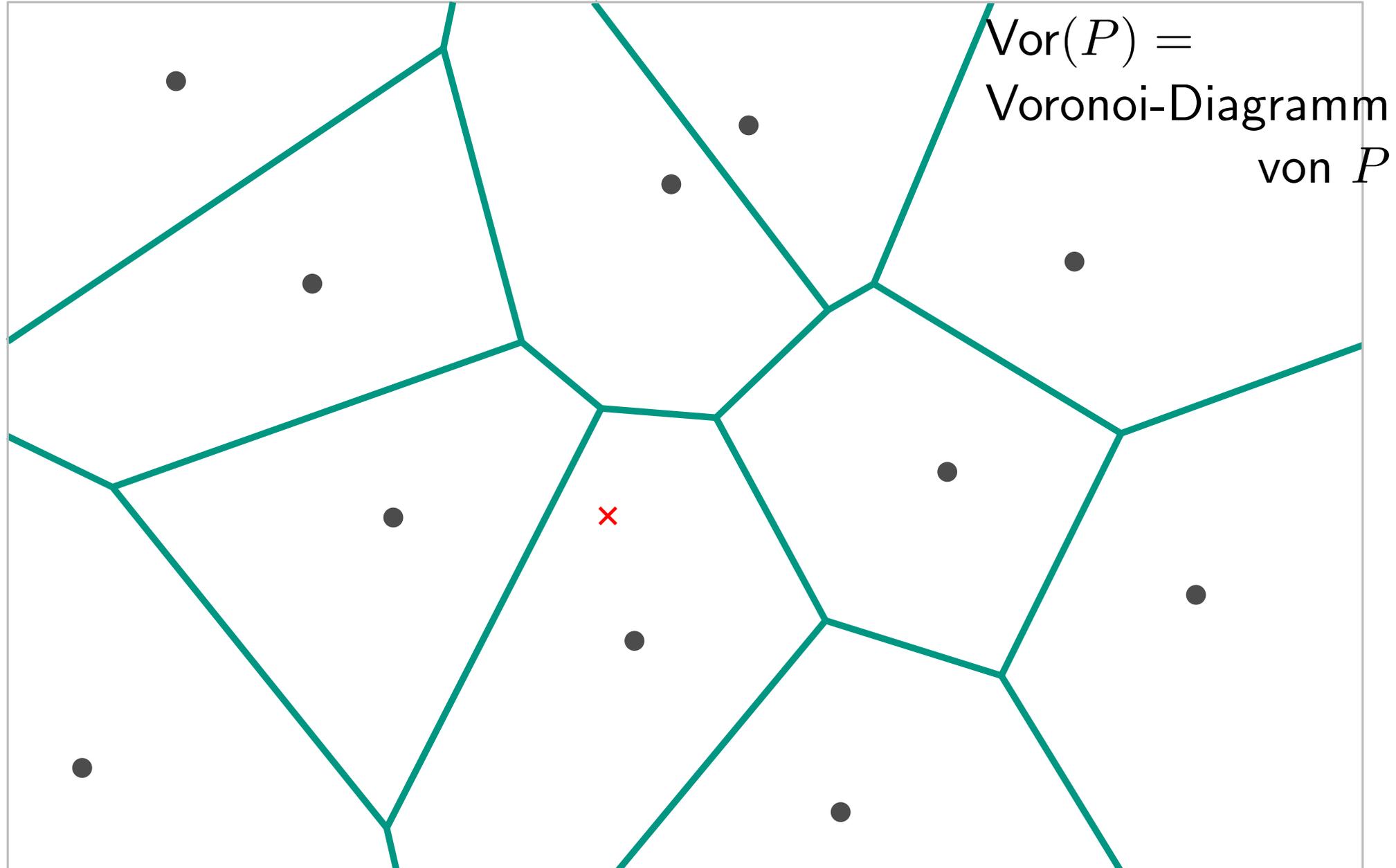
Das Postamt-Problem



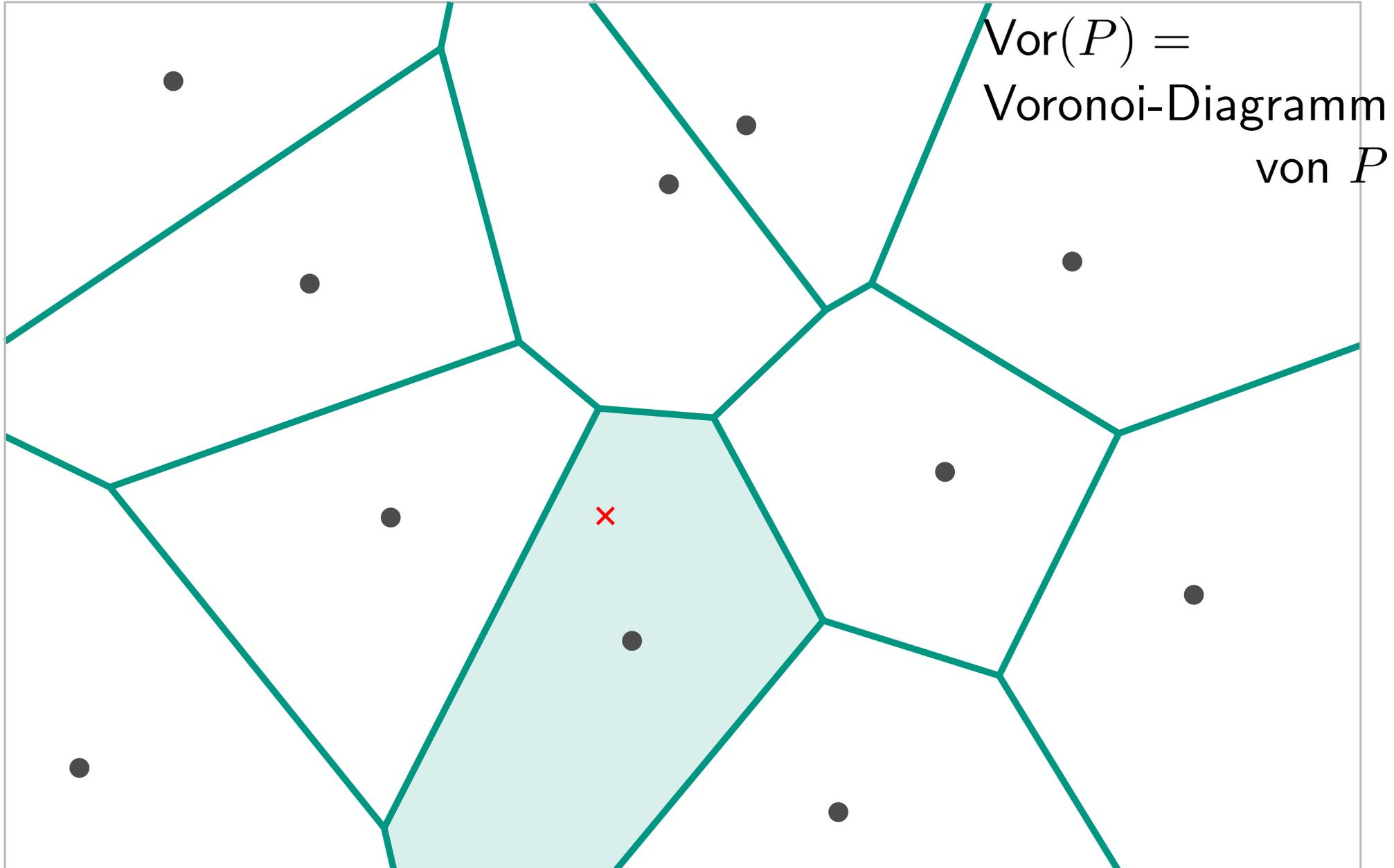
Das Postamt-Problem



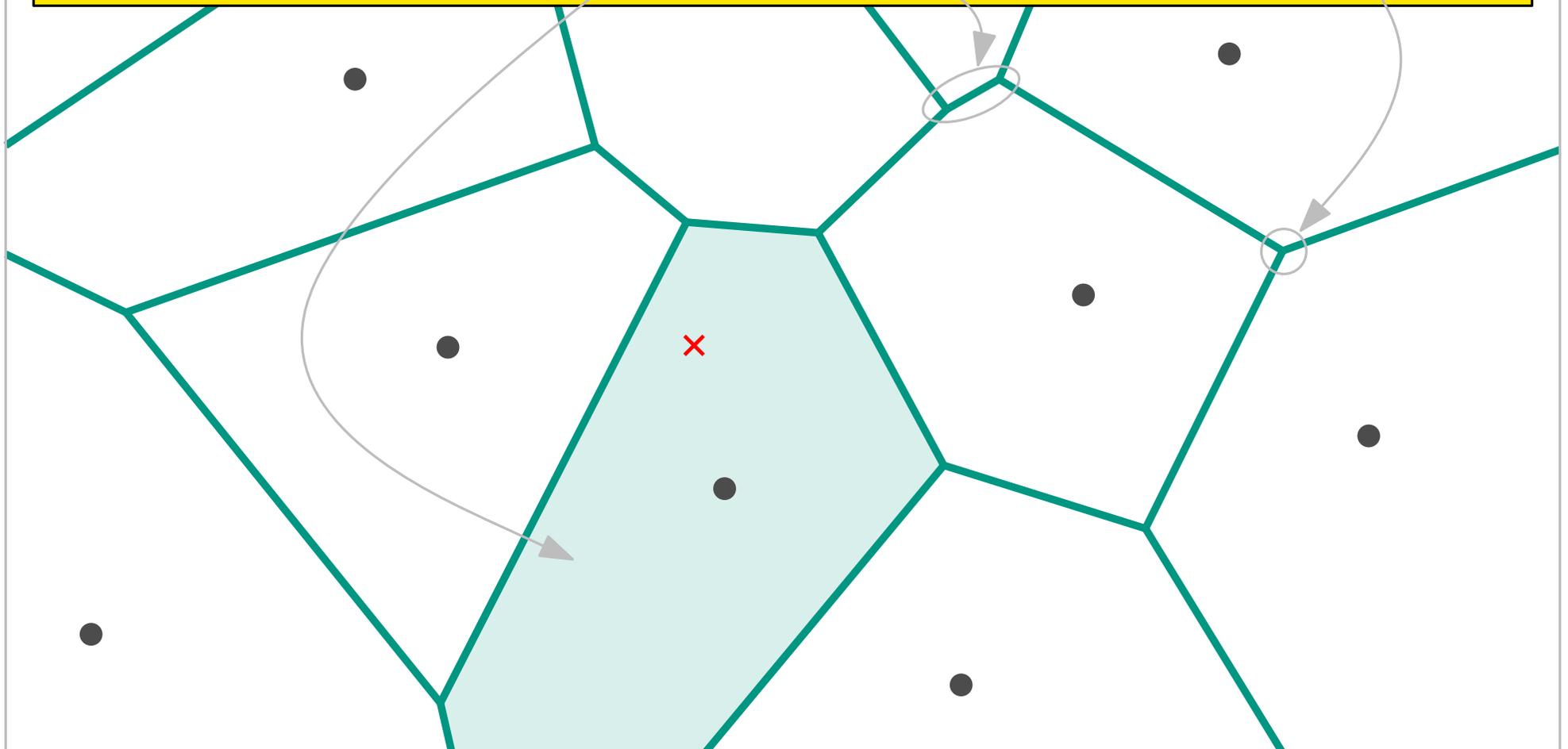
Das Postamt-Problem



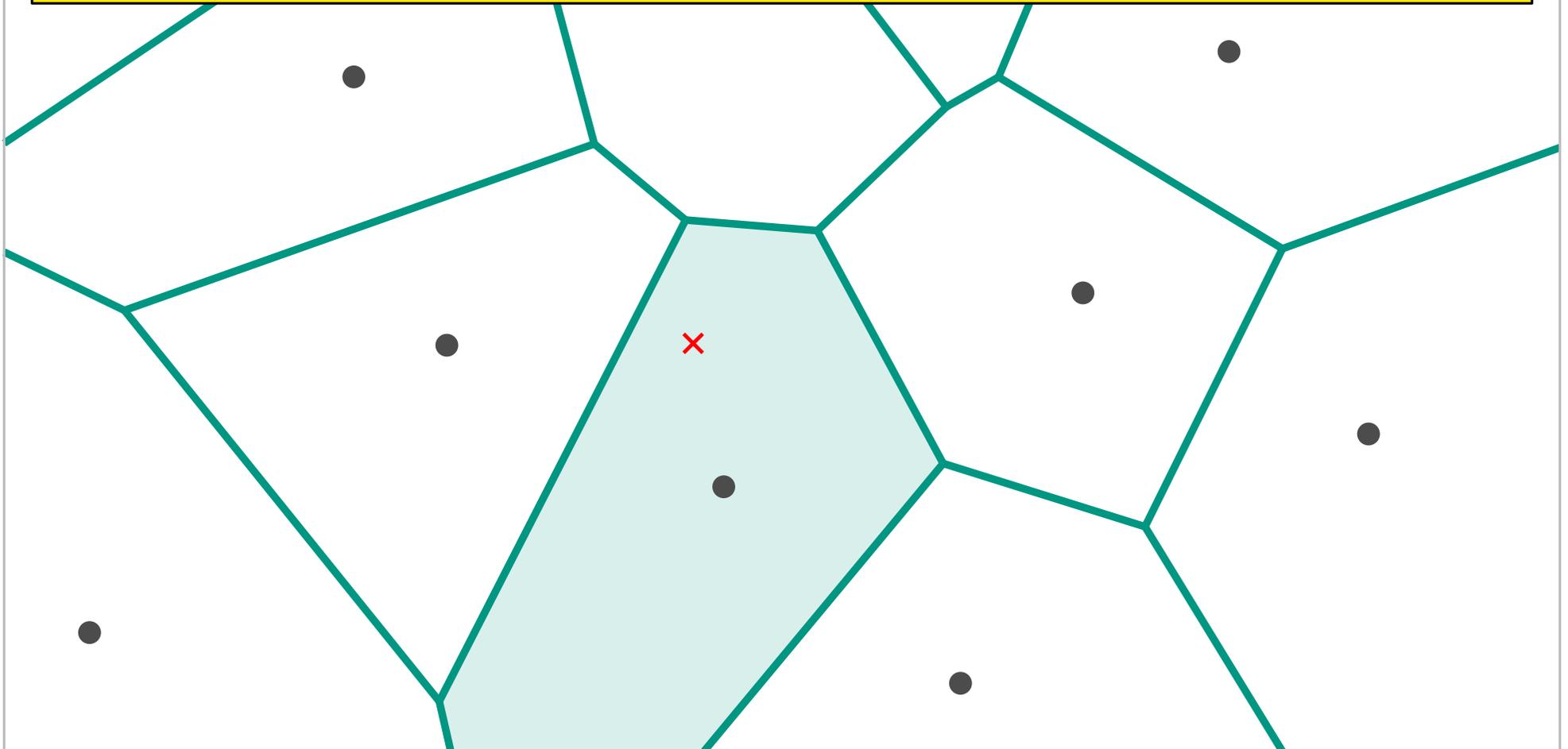
Das Postamt-Problem



Aufgabe: 1) Definiere Voronoi-Zellen, Kanten und Knoten!



Aufgabe: 1) Definiere Voronoi-Zellen, Kanten und Knoten!
2) Sind Voronoi-Zellen konvex?



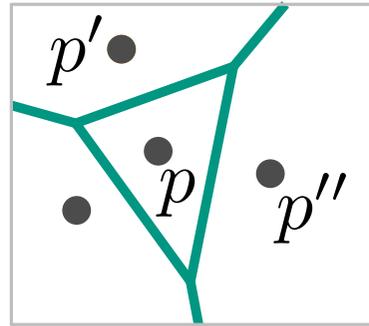
Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm

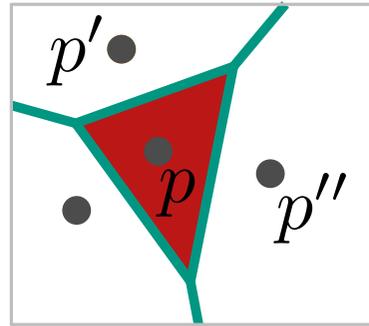


$\text{Vor}(P)$

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm

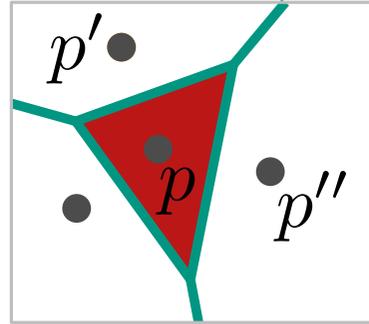


$\text{Vor}(P)$

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

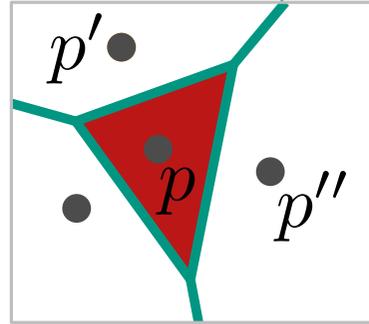
■ Voronoi-Zelle

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

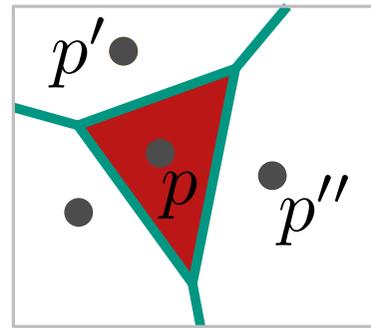
■ Voronoi-Zelle

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

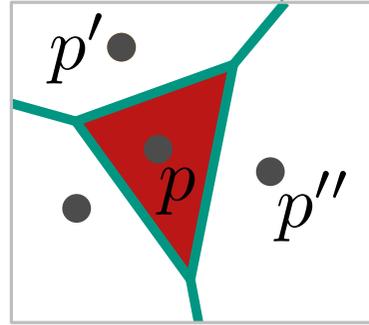
- Voronoi-Zelle

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

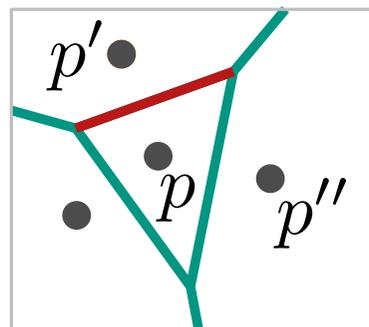
- Voronoi-Zelle

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

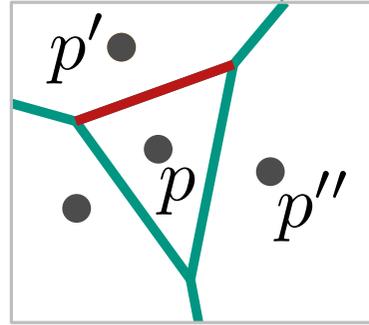
■ Voronoi-Zelle

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

- Voronoi-Zelle

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

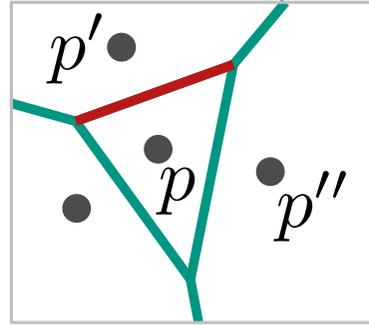
- Voronoi-Kante

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

- Voronoi-Zelle

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

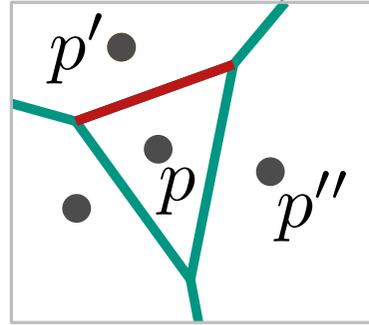
- Voronoi-Kante

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

- Voronoi-Zelle

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

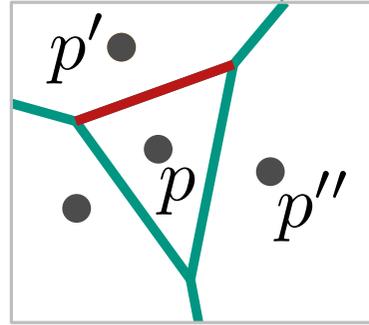
- Voronoi-Kante

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

- Voronoi-Zelle

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

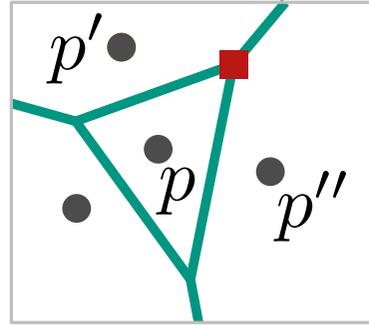
- Voronoi-Kante

$$\begin{aligned}\mathcal{V}(\{p, p'\}) &= \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \forall q \neq p, p'\} \\ &= \text{rel-int}(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')), \text{ d.h. ohne Endpunkte}\end{aligned}$$

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

- Voronoi-Zelle

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

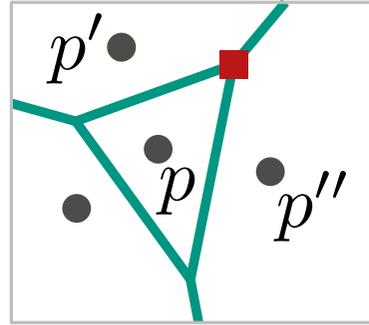
- Voronoi-Kante

$$\begin{aligned}\mathcal{V}(\{p, p'\}) &= \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \forall q \neq p, p'\} \\ &= \text{rel-int}(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')), \text{ d.h. ohne Endpunkte}\end{aligned}$$

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

- Voronoi-Zelle

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

- Voronoi-Kante

$$\begin{aligned}\mathcal{V}(\{p, p'\}) &= \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \forall q \neq p, p'\} \\ &= \text{rel-int}(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')), \text{ d.h. ohne Endpunkte}\end{aligned}$$

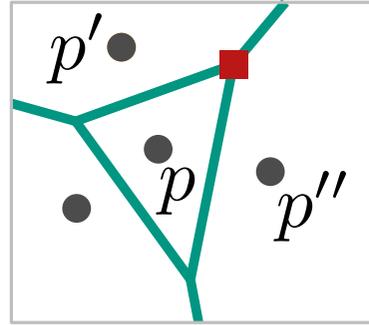
- Voronoi-Knoten

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$

- Voronoi-Zelle

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

- Voronoi-Kante

$$\begin{aligned}\mathcal{V}(\{p, p'\}) &= \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \forall q \neq p, p'\} \\ &= \text{rel-int}(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')), \text{ d.h. ohne Endpunkte}\end{aligned}$$

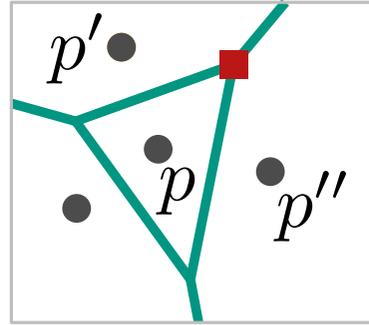
- Voronoi-Knoten

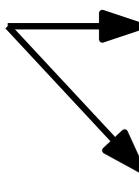
$$\mathcal{V}(\{p, p', p''\}) = \partial\mathcal{V}(p) \cap \partial\mathcal{V}(p') \cap \partial\mathcal{V}(p'')$$

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$ 

■ Voronoi-Zelle

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

■ Voronoi-Kante

$$\begin{aligned}\mathcal{V}(\{p, p'\}) &= \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \forall q \neq p, p'\} \\ &= \text{rel-int}(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')), \text{ d.h. ohne Endpunkte}\end{aligned}$$

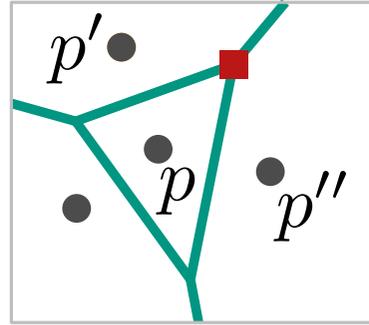
■ Voronoi-Knoten

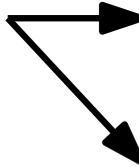
$$\mathcal{V}(\{p, p', p''\}) = \partial\mathcal{V}(p) \cap \partial\mathcal{V}(p') \cap \partial\mathcal{V}(p'')$$

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$  Unterteilung

■ Voronoi-Zelle

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

■ Voronoi-Kante

$$\begin{aligned}\mathcal{V}(\{p, p'\}) &= \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \forall q \neq p, p'\} \\ &= \text{rel-int}(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')), \text{ d.h. ohne Endpunkte}\end{aligned}$$

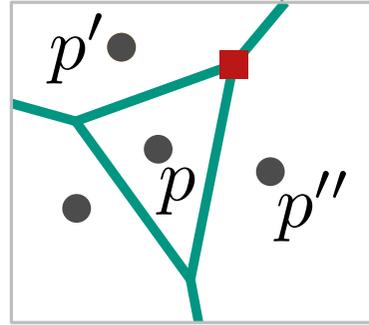
■ Voronoi-Knoten

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

Das Voronoi-Diagramm

Sei P eine Menge von Punkten in der Ebene und $p, p', p'' \in P$.

Voronoi-Diagramm



$\text{Vor}(P)$ $\begin{cases} \rightarrow \text{Unterteilung} \\ \rightarrow \text{geometr. Graph} \end{cases}$

■ Voronoi-Zelle

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

■ Voronoi-Kante

$$\begin{aligned} \mathcal{V}(\{p, p'\}) &= \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \forall q \neq p, p'\} \\ &= \text{rel-int}(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')), \text{ d.h. ohne Endpunkte} \end{aligned}$$

■ Voronoi-Knoten

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

Satz: Sei $P \subset \mathbb{R}^2$ eine Menge von n Punkten. Sind alle Punkte kollinear, besteht $\text{Vor}(P)$ aus $n - 1$ parallelen Geraden. Sonst ist $\text{Vor}(P)$ zusammenhängend und die Kanten sind Strecken oder Halbgeraden.

Satz: Sei $P \subset \mathbb{R}^2$ eine Menge von n Punkten. Sind alle Punkte kollinear, besteht $\text{Vor}(P)$ aus $n - 1$ parallelen Geraden. Sonst ist $\text{Vor}(P)$ zusammenhängend und die Kanten sind Strecken oder Halbgeraden.

Finde eine Menge P , so dass $\text{Vor}(P)$ eine Zelle linearer Komplexität hat.

Satz: Sei $P \subset \mathbb{R}^2$ eine Menge von n Punkten. Sind alle Punkte kollinear, besteht $\text{Vor}(P)$ aus $n - 1$ parallelen Geraden. Sonst ist $\text{Vor}(P)$ zusammenhängend und die Kanten sind Strecken oder Halbgeraden.

Finde eine Menge P , so dass $\text{Vor}(P)$ eine Zelle linearer Komplexität hat.

Kann das für (fast) jede Zelle passieren?

Satz: Sei $P \subset \mathbb{R}^2$ eine Menge von n Punkten. Sind alle Punkte kollinear, besteht $\text{Vor}(P)$ aus $n - 1$ parallelen Geraden. Sonst ist $\text{Vor}(P)$ zusammenhängend und die Kanten sind Strecken oder Halbgeraden.

Finde eine Menge P , so dass $\text{Vor}(P)$ eine Zelle linearer Komplexität hat.

Kann das für (fast) jede Zelle passieren?

Satz: Sei $P \subset \mathbb{R}^2$ Menge von n Punkten. $\text{Vor}(P)$ besteht aus

Satz: Sei $P \subset \mathbb{R}^2$ eine Menge von n Punkten. Sind alle Punkte kollinear, besteht $\text{Vor}(P)$ aus $n - 1$ parallelen Geraden. Sonst ist $\text{Vor}(P)$ zusammenhängend und die Kanten sind Strecken oder Halbgeraden.

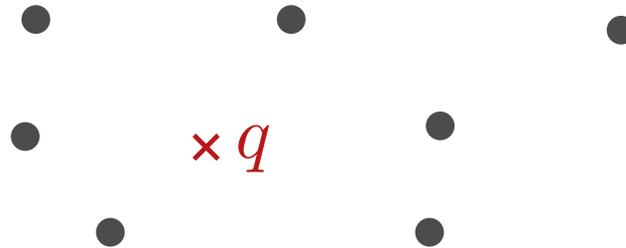
Finde eine Menge P , so dass $\text{Vor}(P)$ eine Zelle linearer Komplexität hat.

Kann das für (fast) jede Zelle passieren?

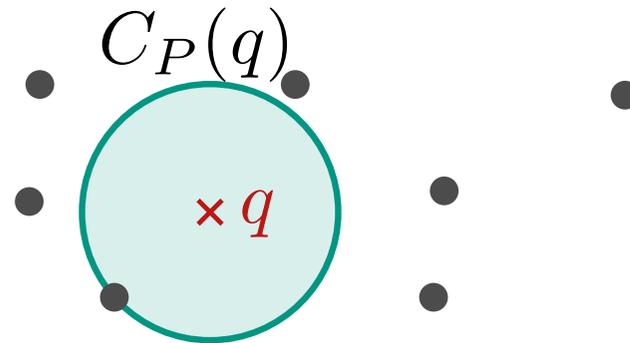
Satz: Sei $P \subset \mathbb{R}^2$ Menge von n Punkten. $\text{Vor}(P)$ besteht aus höchstens $2n - 5$ Knoten und $3n - 6$ Kanten.

Charakterisierung

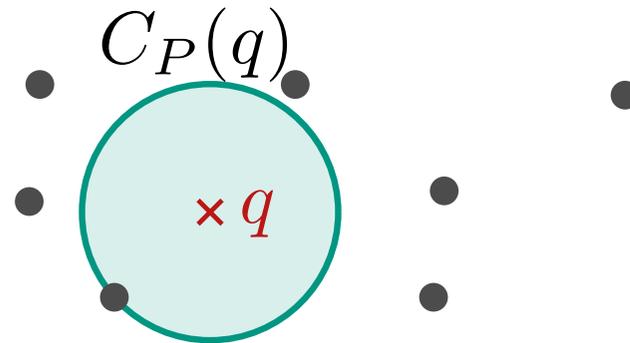
Definition: Sei q ein Punkt. Definiere $C_P(q)$ als den bzgl. P größten im Inneren leeren Kreis mit Mittelpunkt q .



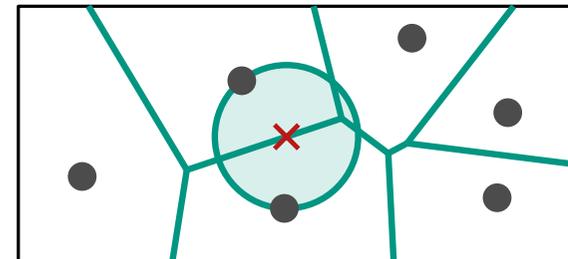
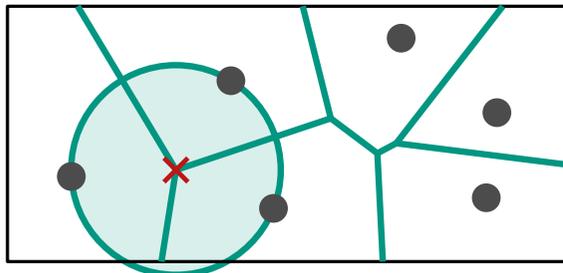
Definition: Sei q ein Punkt. Definiere $C_P(q)$ als den bzgl. P größten im Inneren leeren Kreis mit Mittelpunkt q .



Definition: Sei q ein Punkt. Definiere $C_P(q)$ als den bzgl. P größten im Inneren leeren Kreis mit Mittelpunkt q .



- Satz:**
- Ein Punkt q ist ein Voronoi-Knoten
 $\Leftrightarrow |C_P(q) \cap P| \geq 3$,
 - der Bisektor $b(p_i, p_j)$ definiert eine Voronoi-Kante
 $\Leftrightarrow \exists q \in b(p_i, p_j)$ mit $C_P(q) \cap P = \{p_i, p_j\}$.



Berechnung von $\text{Vor}(P)$

Für jedes $p \in P$ ist $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$ der Schnitt von $n - 1$ Halbebenen.

Wie könnte man $\text{Vor}(P)$ mit schon bekannten Algorithmen berechnen?

Berechnung von $\text{Vor}(P)$

Für jedes $p \in P$ ist $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$ der Schnitt von $n - 1$ Halbebenen.

Wie könnte man $\text{Vor}(P)$ mit schon bekannten Algorithmen berechnen?

foreach $p \in P$ **do**

└ berechne $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$

Berechnung von $\text{Vor}(P)$

Für jedes $p \in P$ ist $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$ der Schnitt von $n - 1$ Halbebenen.

Wie könnte man $\text{Vor}(P)$ mit schon bekannten Algorithmen berechnen?

foreach $p \in P$ **do**

└ berechne $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$ $O(n \log n)$ [VL 4]

Berechnung von $\text{Vor}(P)$

Für jedes $p \in P$ ist $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$ der Schnitt von $n - 1$ Halbebenen.

Wie könnte man $\text{Vor}(P)$ mit schon bekannten Algorithmen berechnen?

```
foreach  $p \in P$  do  $O(n^2 \log n)$   
└ berechne  $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$   $O(n \log n)$  [VL 4]
```

Berechnung von $\text{Vor}(P)$

Für jedes $p \in P$ ist $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$ der Schnitt von $n - 1$ Halbebenen.

Wie könnte man $\text{Vor}(P)$ mit schon bekannten Algorithmen berechnen?

```
foreach  $p \in P$  do  $O(n^2 \log n)$   
└ berechne  $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$   $O(n \log n)$  [VL 4]
```

Ist $O(n^2 \log n)$ Laufzeit für ein Objekt linearer Größe nötig?

Berechnung von $\text{Vor}(P)$

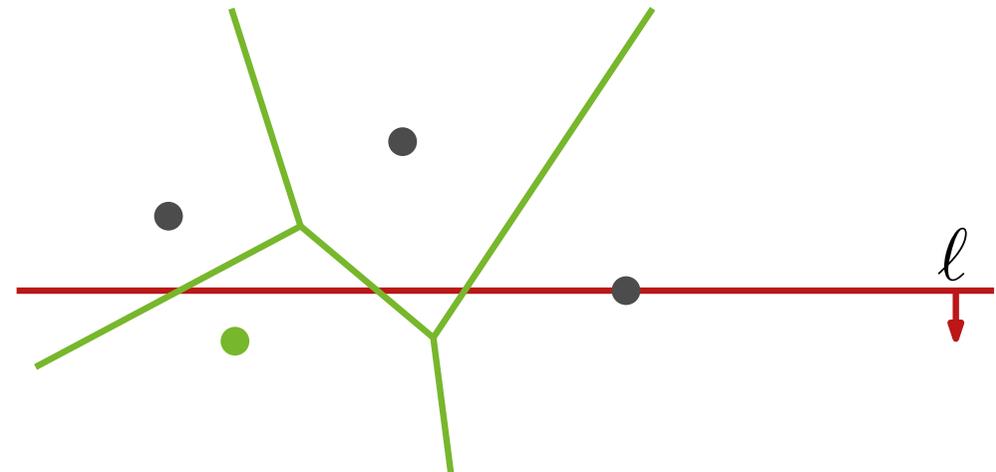
Für jedes $p \in P$ ist $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$ der Schnitt von $n - 1$ Halbebenen.

Wie könnte man $\text{Vor}(P)$ mit schon bekannten Algorithmen berechnen?

```
foreach  $p \in P$  do  $O(n^2 \log n)$   
└ berechne  $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$   $O(n \log n)$  [VL 4]
```

Ist $O(n^2 \log n)$ Laufzeit für ein Objekt linearer Größe nötig?

Idee 2: Sweep-Verfahren



Berechnung von $\text{Vor}(P)$

Für jedes $p \in P$ ist $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$ der Schnitt von $n - 1$ Halbebenen.

Wie könnte man $\text{Vor}(P)$ mit schon bekannten Algorithmen berechnen?

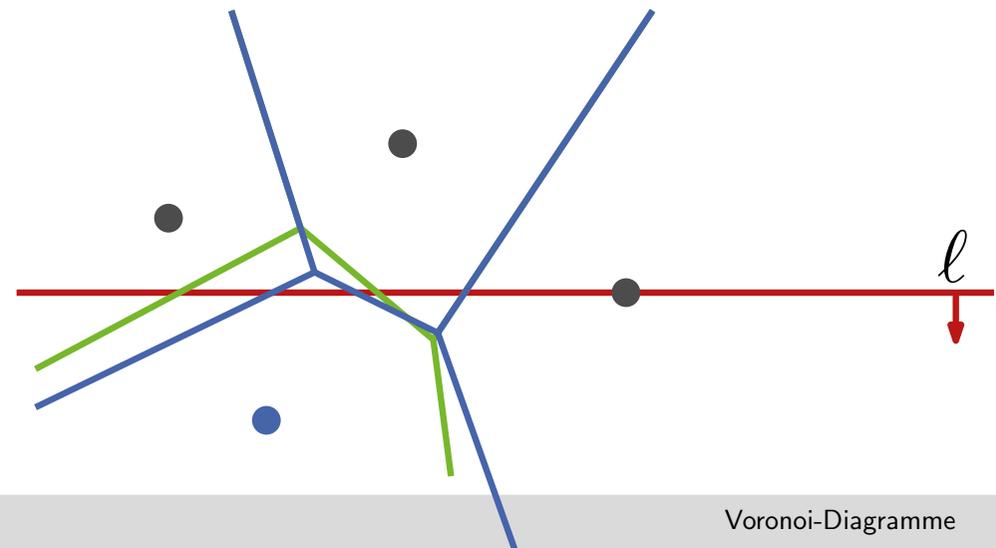
```
foreach  $p \in P$  do  $O(n^2 \log n)$   
└ berechne  $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$   $O(n \log n)$  [VL 4]
```

Ist $O(n^2 \log n)$ Laufzeit für ein Objekt linearer Größe nötig?

Idee 2: Sweep-Verfahren

Problem:

$\text{Vor}(P)$ oberhalb ℓ hängt von Punkten unterhalb ℓ ab!



Berechnung von $\text{Vor}(P)$

Für jedes $p \in P$ ist $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$ der Schnitt von $n - 1$ Halbebenen.

Wie könnte man $\text{Vor}(P)$ mit schon bekannten Algorithmen berechnen?

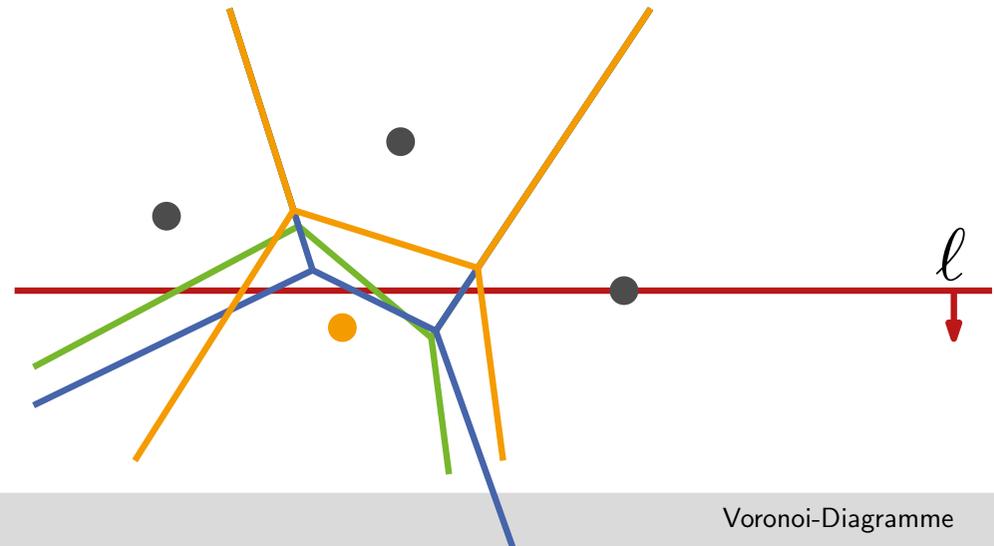
```
foreach  $p \in P$  do  $O(n^2 \log n)$   
└ berechne  $\mathcal{V}(p) = \bigcap_{p' \neq p} h(p, p')$   $O(n \log n)$  [VL 4]
```

Ist $O(n^2 \log n)$ Laufzeit für ein Objekt linearer Größe nötig?

Idee 2: Sweep-Verfahren

Problem:

$\text{Vor}(P)$ oberhalb ℓ hängt von Punkten unterhalb ℓ ab!



In Richtung Sweep-Verfahren

Offensichtlich ist der Schnitt von $\text{Vor}(P)$ und Sweep Line ℓ zum aktuellen Zeitpunkt noch nicht bekannt.

In Richtung Sweep-Verfahren

Offensichtlich ist der Schnitt von $\text{Vor}(P)$ und Sweep Line ℓ zum aktuellen Zeitpunkt noch nicht bekannt.

Betrachte stattdessen den Teil oberhalb ℓ , der schon fest ist!

In Richtung Sweep-Verfahren

Offensichtlich ist der Schnitt von $\text{Vor}(P)$ und Sweep Line ℓ zum aktuellen Zeitpunkt noch nicht bekannt.

Betrachte stattdessen den Teil oberhalb ℓ , der schon fest ist!

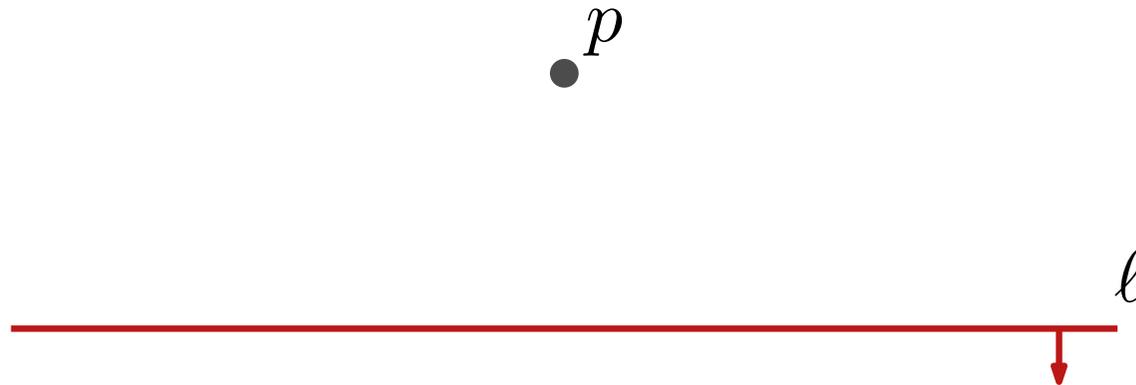
Wie sieht der aus?

In Richtung Sweep-Verfahren

Offensichtlich ist der Schnitt von $\text{Vor}(P)$ und Sweep Line ℓ zum aktuellen Zeitpunkt noch nicht bekannt.

Betrachte stattdessen den Teil oberhalb ℓ , der schon fest ist!

Wie sieht der aus?



In Richtung Sweep-Verfahren

Offensichtlich ist der Schnitt von $\text{Vor}(P)$ und Sweep Line ℓ zum aktuellen Zeitpunkt noch nicht bekannt.

Betrachte stattdessen den Teil oberhalb ℓ , der schon fest ist!

Wie sieht der aus?

p



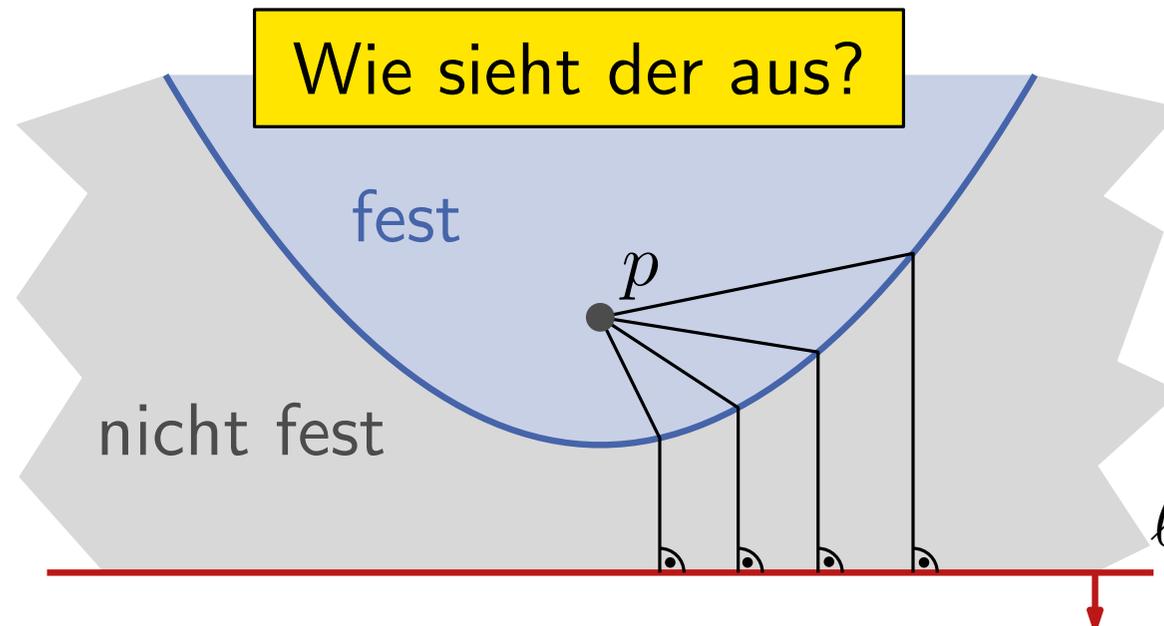
Punkte näher an p als an ℓ sind schon fest zugeordnet



In Richtung Sweep-Verfahren

Offensichtlich ist der Schnitt von $\text{Vor}(P)$ und Sweep Line ℓ zum aktuellen Zeitpunkt noch nicht bekannt.

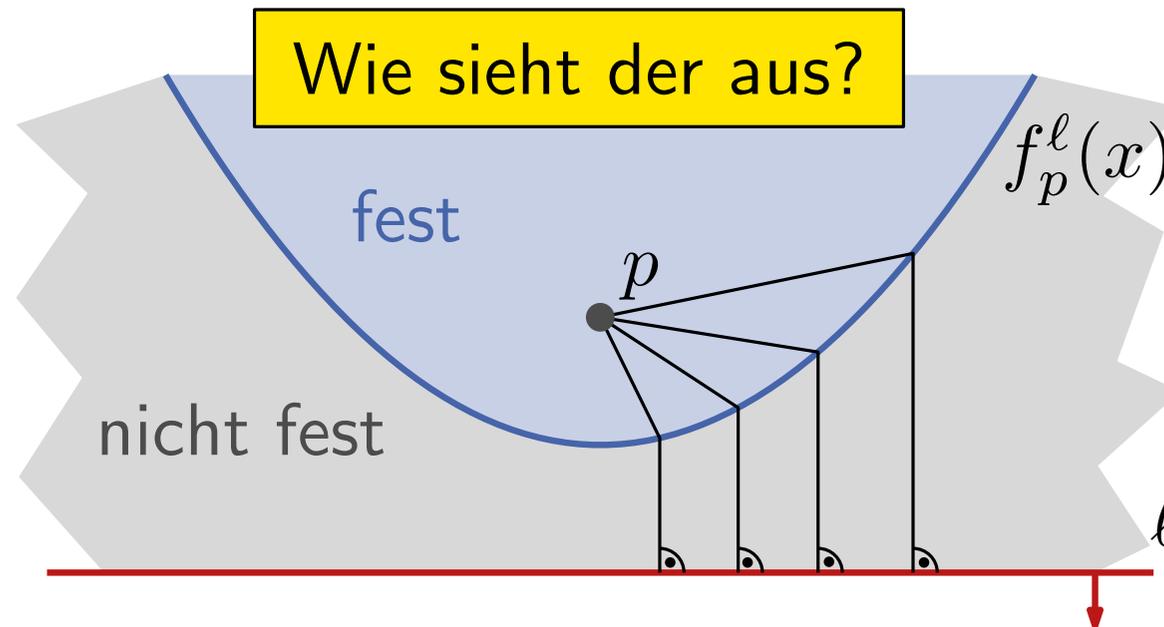
Betrachte stattdessen den Teil oberhalb ℓ , der schon fest ist!



In Richtung Sweep-Verfahren

Offensichtlich ist der Schnitt von $\text{Vor}(P)$ und Sweep Line ℓ zum aktuellen Zeitpunkt noch nicht bekannt.

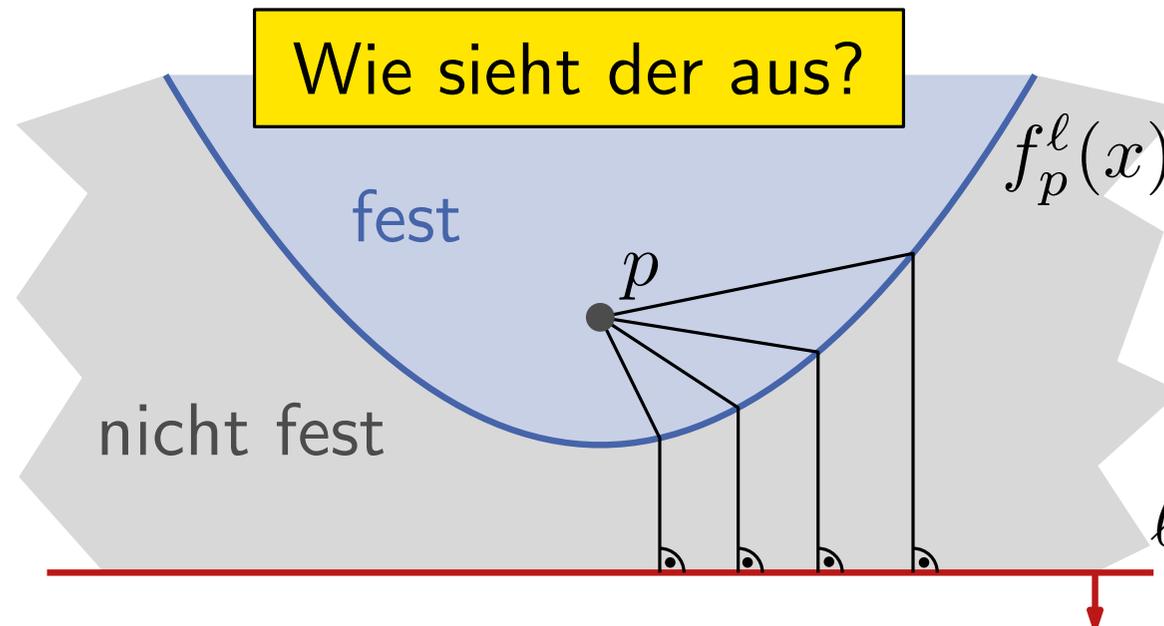
Betrachte stattdessen den Teil oberhalb ℓ , der schon fest ist!



In Richtung Sweep-Verfahren

Offensichtlich ist der Schnitt von $\text{Vor}(P)$ und Sweep Line ℓ zum aktuellen Zeitpunkt noch nicht bekannt.

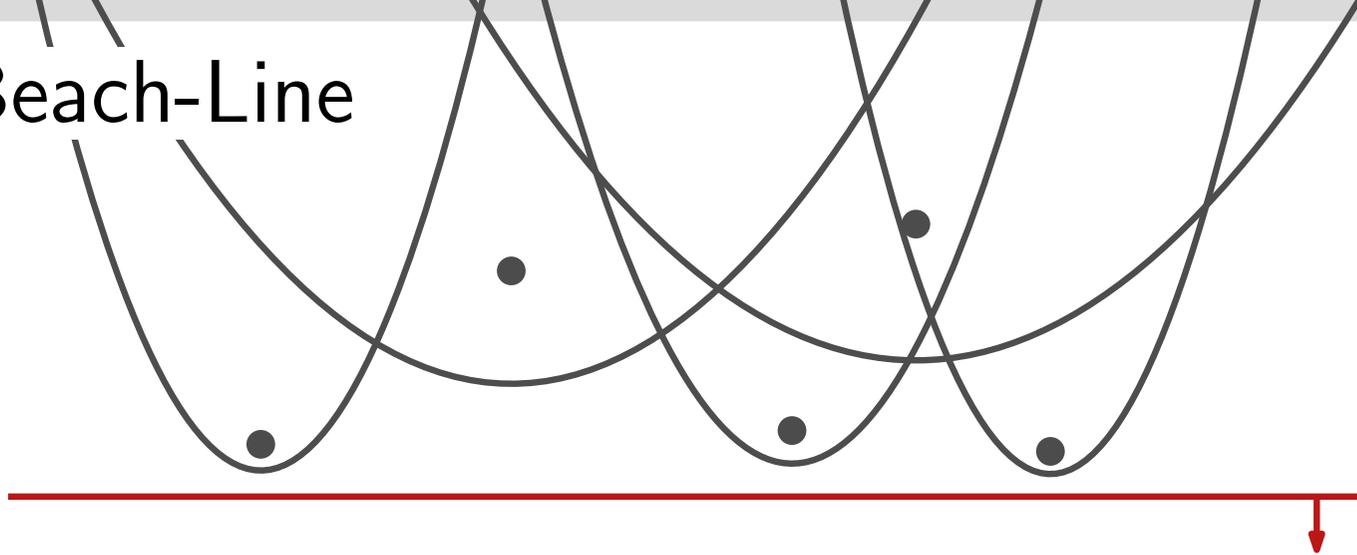
Betrachte stattdessen den Teil oberhalb ℓ , der schon fest ist!



Lösen der Gleichung $|pq| = |q\ell|$ liefert

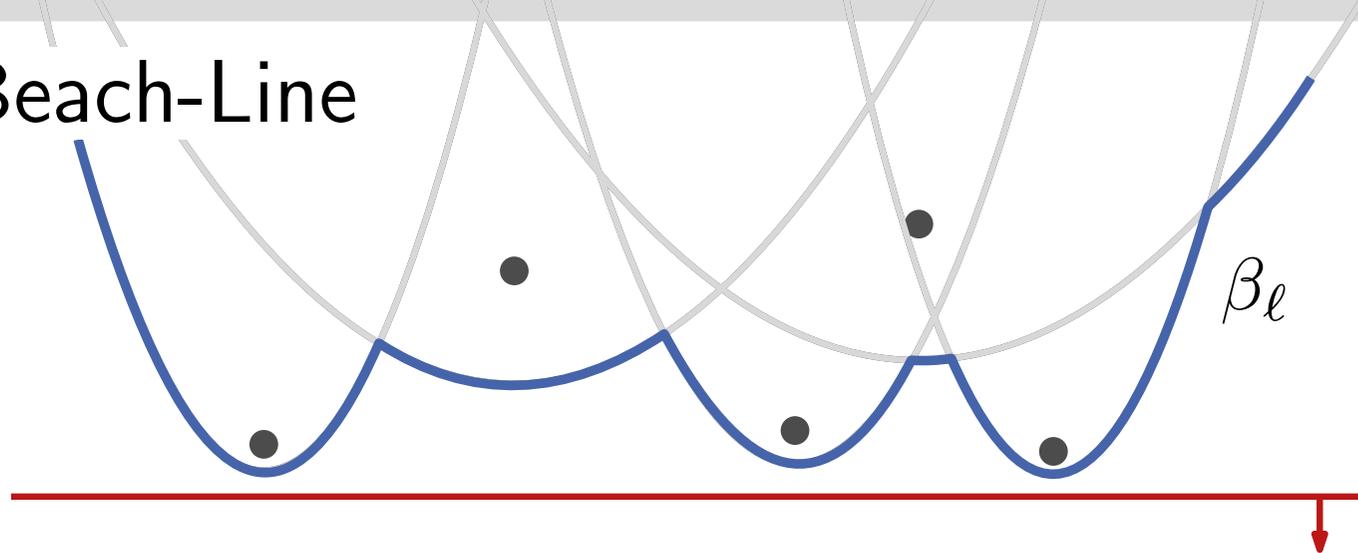
$$f_p^\ell(x) = \frac{1}{2(p_y - \ell_y)} (x - p_x)^2 + \frac{p_y + \ell_y}{2}$$

Die Beach-Line



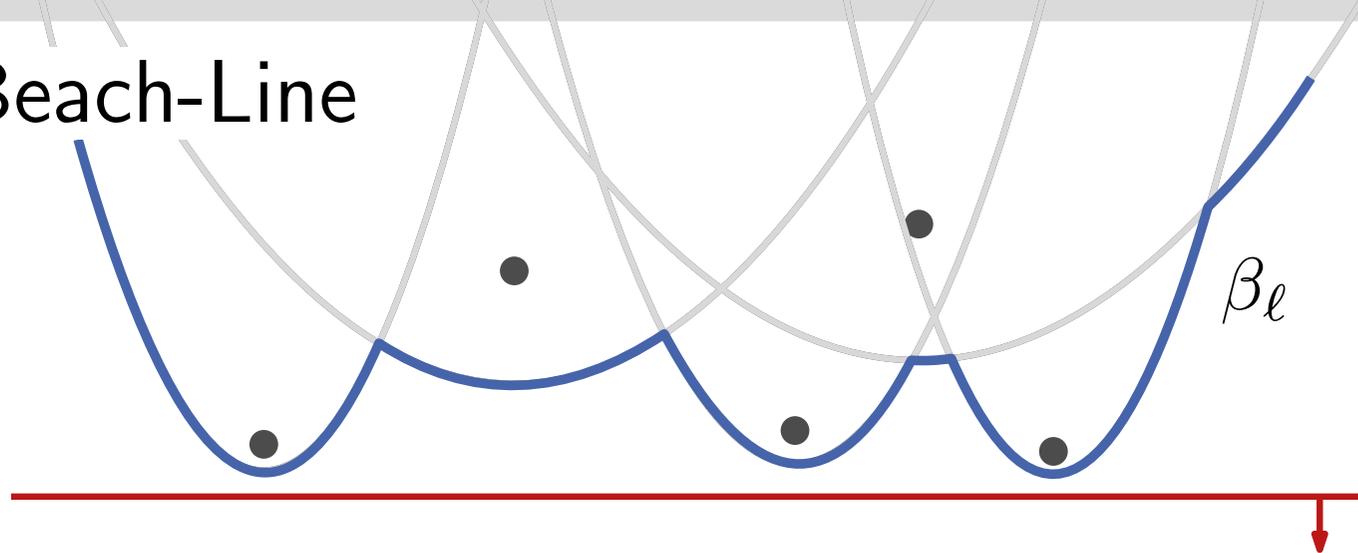
Definition: Die **Beach-Line** β_ℓ ist die untere Kontur der Parabeln f_p^ℓ für die bereits besuchten Punkte.

Die Beach-Line



Definition: Die **Beach-Line** β_ℓ ist die untere Kontur der Parabeln f_p^ℓ für die bereits besuchten Punkte.

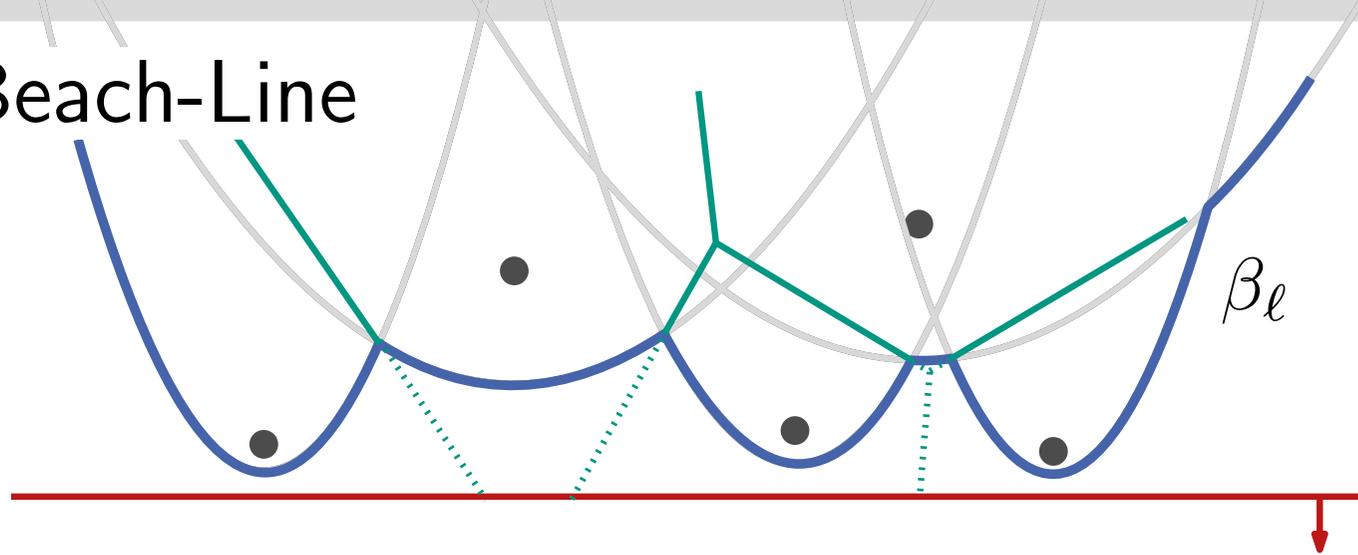
Die Beach-Line



Definition: Die **Beach-Line** β_ℓ ist die untere Kontur der Parabeln f_p^ℓ für die bereits besuchten Punkte.

Was hat das mit $\text{Vor}(P)$ zu tun?

Die Beach-Line

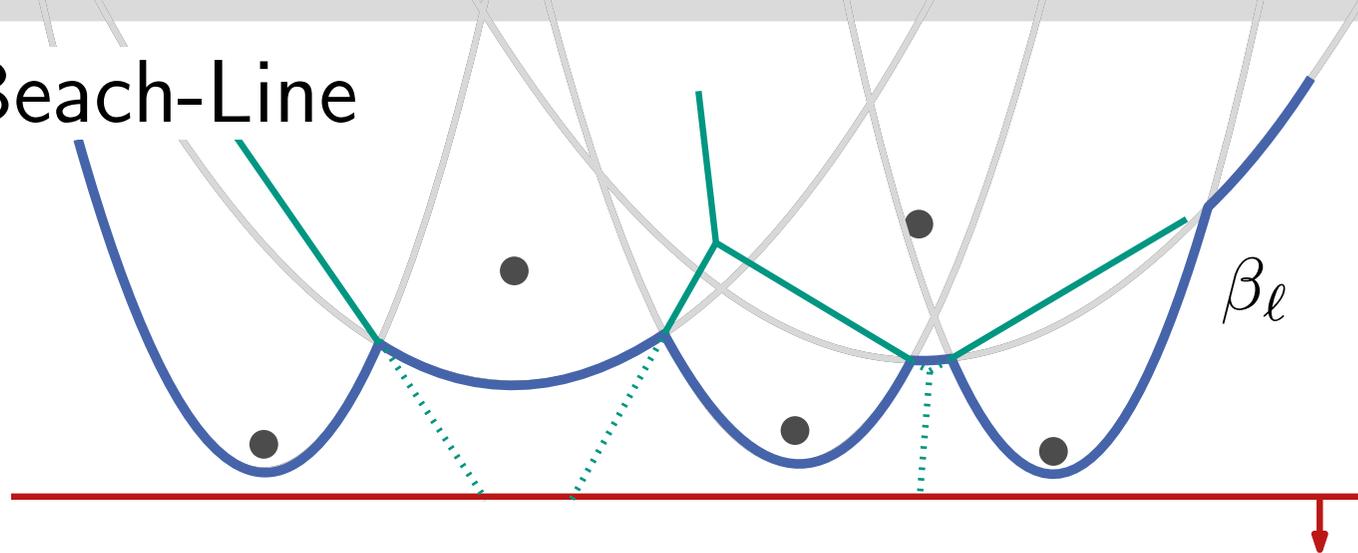


Definition: Die **Beach-Line** β_ℓ ist die untere Kontur der Parabeln f_p^ℓ für die bereits besuchten Punkte.

Was hat das mit $\text{Vor}(P)$ zu tun?

- Beob.:**
- Beach-Line ist x -monoton
 - Schnittpunkte der Beach-Line liegen auf Voronoi-Kanten

Die Beach-Line

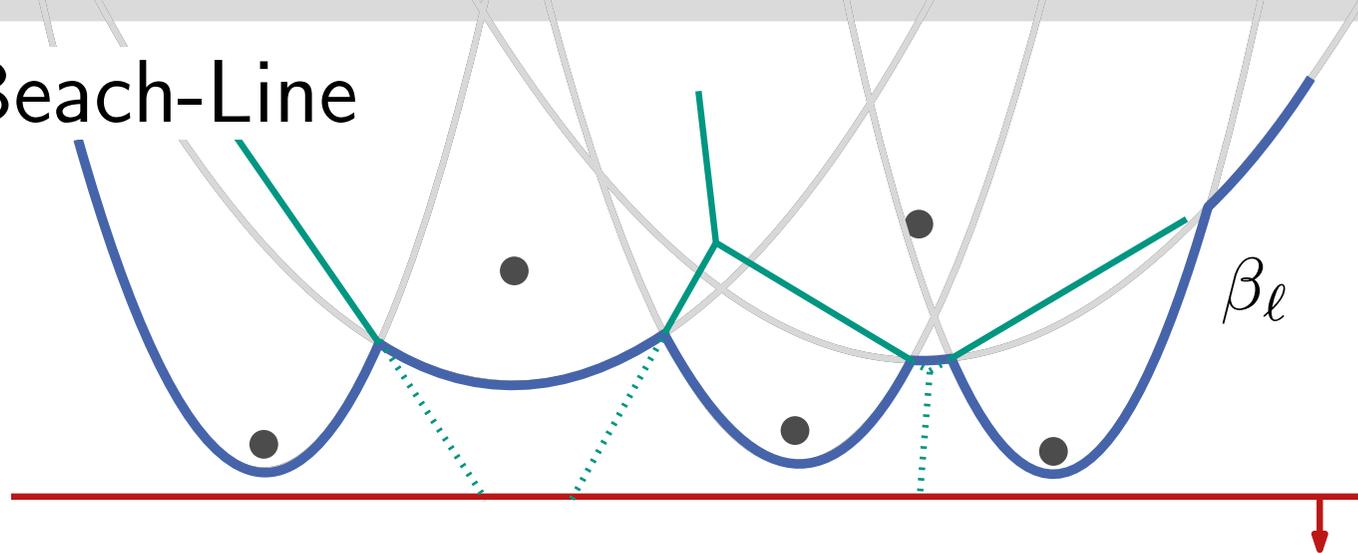


Definition: Die **Beach-Line** β_ℓ ist die untere Kontur der Parabeln f_p^ℓ für die bereits besuchten Punkte.

Was hat das mit $\text{Vor}(P)$ zu tun?

- Beob.:**
- Beach-Line ist x -monoton
 - Schnittpunkte der Beach-Line liegen auf Voronoi-Kanten
- sogar:** Schnittpunkte laufen entlang $\text{Vor}(P)$

Die Beach-Line



Definition: Die **Beach-Line** β_ℓ ist die untere Kontur der Parabeln f_p^ℓ für die bereits besuchten Punkte.

Was hat das mit $\text{Vor}(P)$ zu tun?

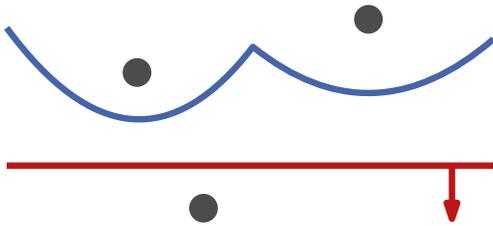
Beob.:

- Beach-Line ist x -monoton
- Schnittpunkte der Beach-Line liegen auf Voronoi-Kanten

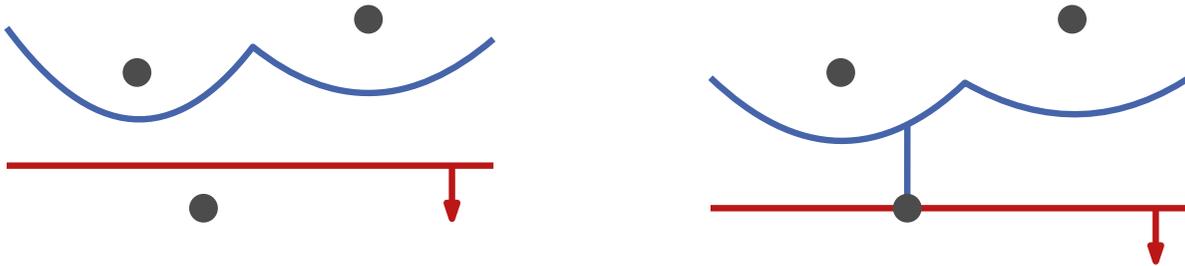
sogar: Schnittpunkte laufen entlang $\text{Vor}(P)$

Ziel: speichere (implizit) aktuelle Kontur β_ℓ statt $\text{Vor}(P) \cap \ell$

Punkt-Events

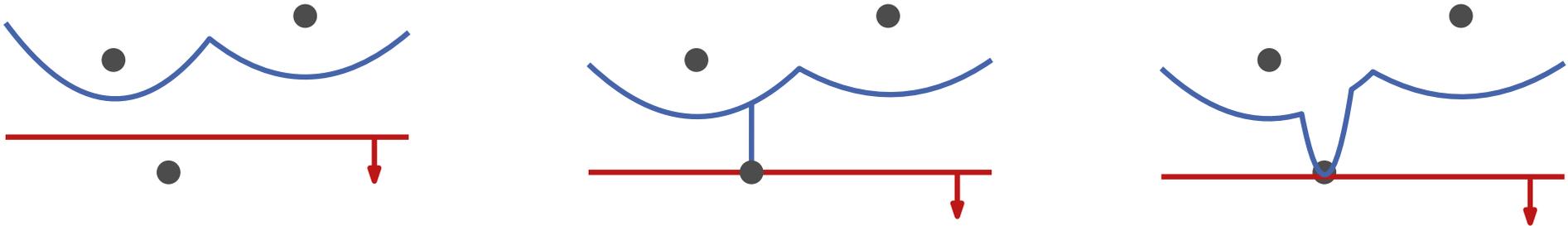


Punkt-Events



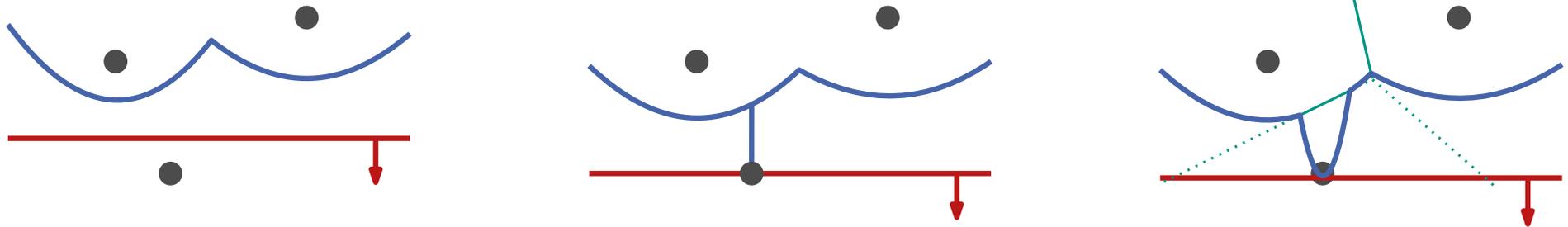
- trifft ℓ auf einen Punkt, kommt neue Parabel zu β_ℓ hinzu

Punkt-Events



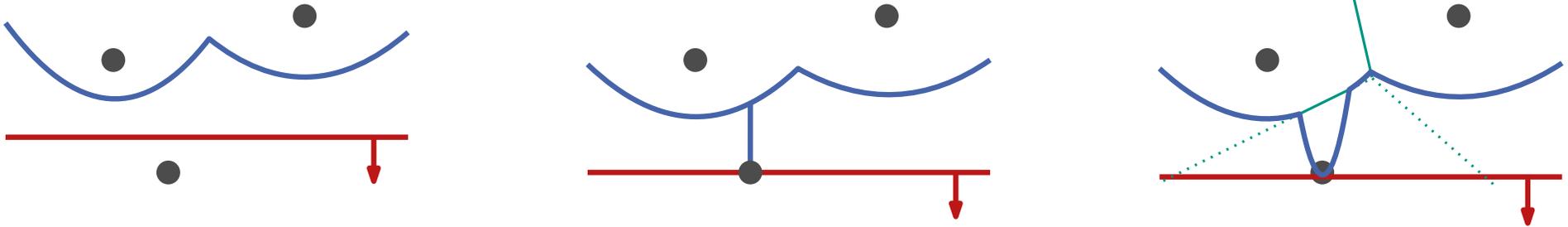
- trifft ℓ auf einen Punkt, kommt neue Parabel zu β_ℓ hinzu

Punkt-Events



- trifft ℓ auf einen Punkt, kommt neue Parabel zu β_ℓ hinzu
- die beiden Schnittpunkte erzeugen neue Teilkante

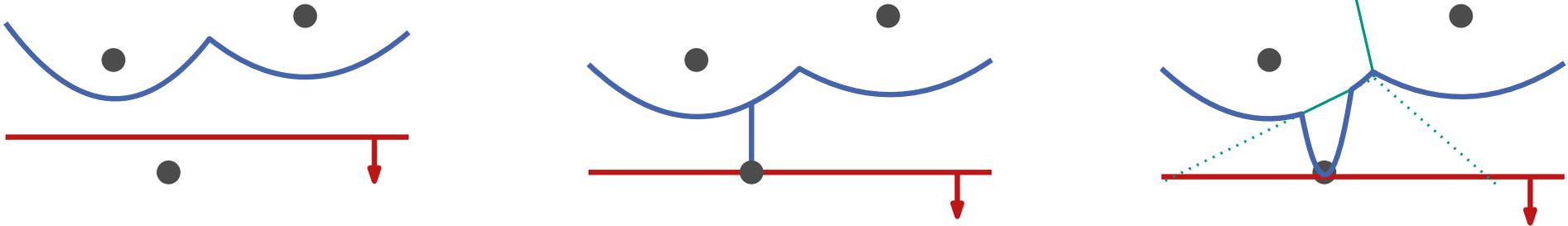
Punkt-Events



- trifft ℓ auf einen Punkt, kommt neue Parabel zu β_ℓ hinzu
- die beiden Schnittpunkte erzeugen neue Teilkante

Lemma: Neue Bögen auf β entstehen nur durch Punkt-Events.

Punkt-Events

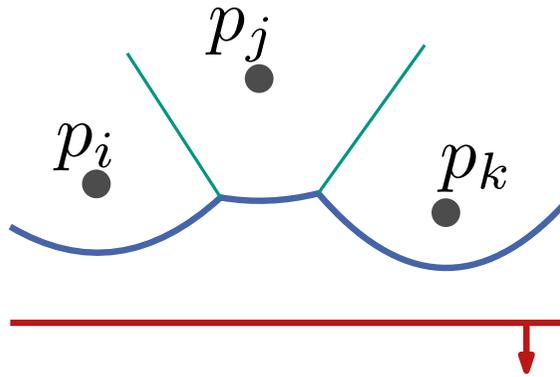


- trifft ℓ auf einen Punkt, kommt neue Parabel zu β_ℓ hinzu
- die beiden Schnittpunkte erzeugen neue Teilkante

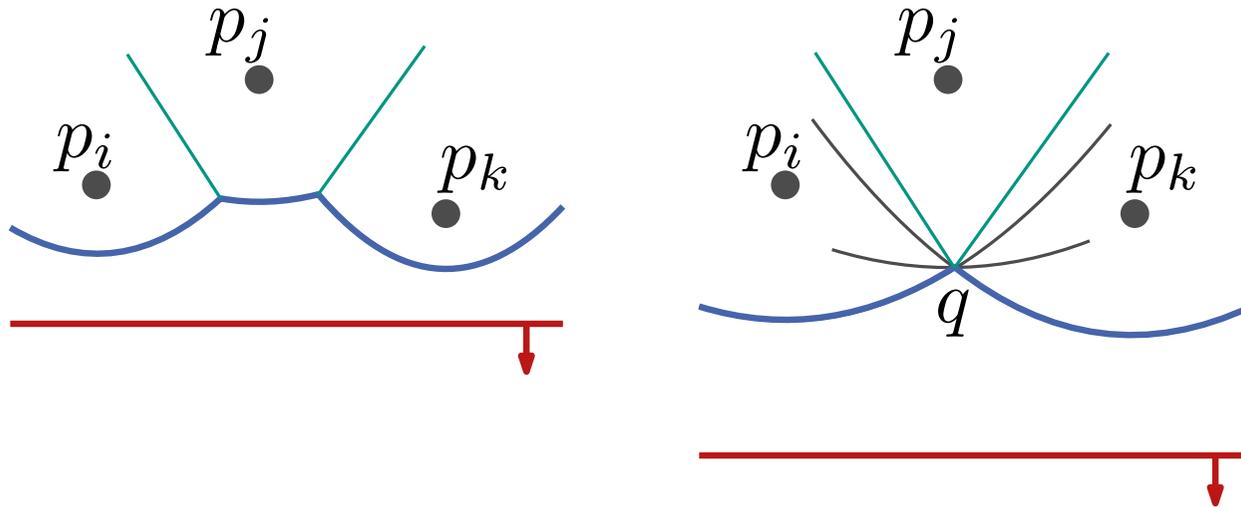
Lemma: Neue Bögen auf β entstehen nur durch Punkt-Events.

Korollar: β besteht aus maximal $2n - 1$ Parabelbögen

Kreis-Events

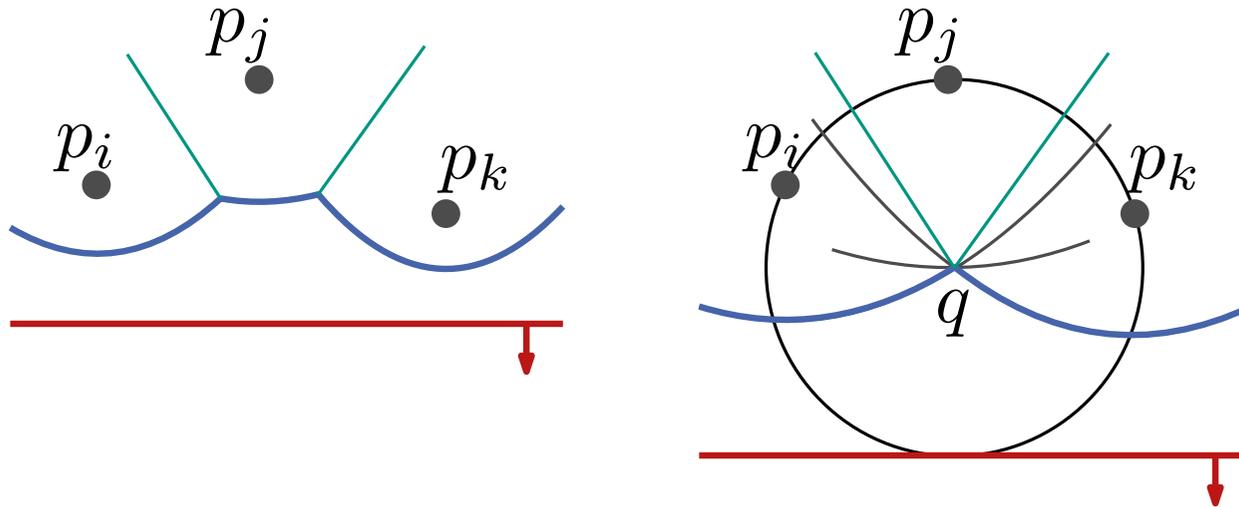


Kreis-Events



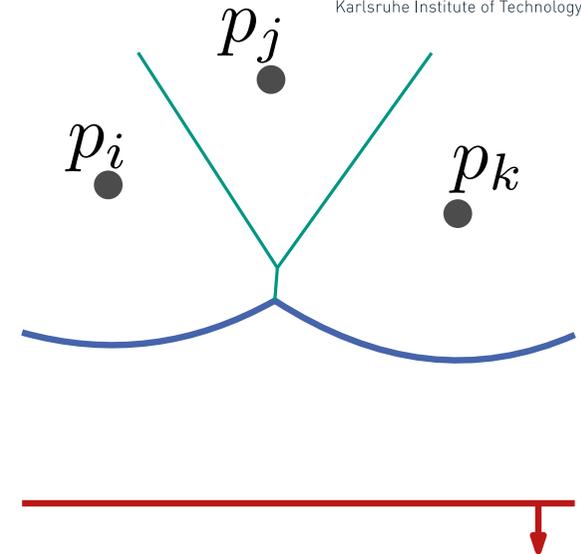
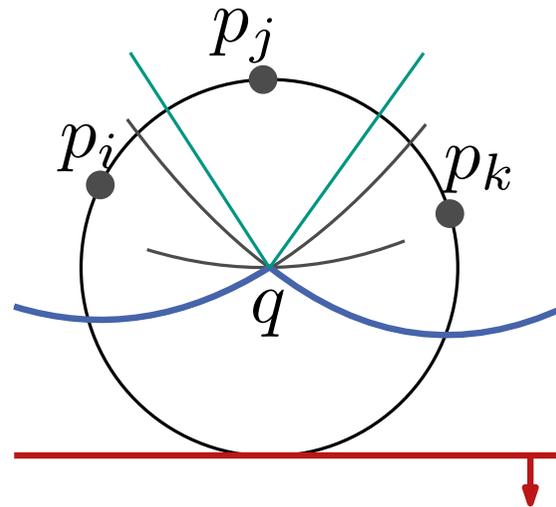
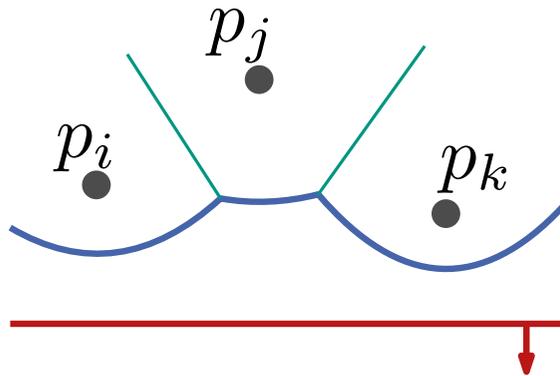
- verschwindet ein Bogen so laufen $f_{p_i}^l$, $f_{p_j}^l$, $f_{p_k}^l$ durch einen gemeinsamen Punkt q

Kreis-Events



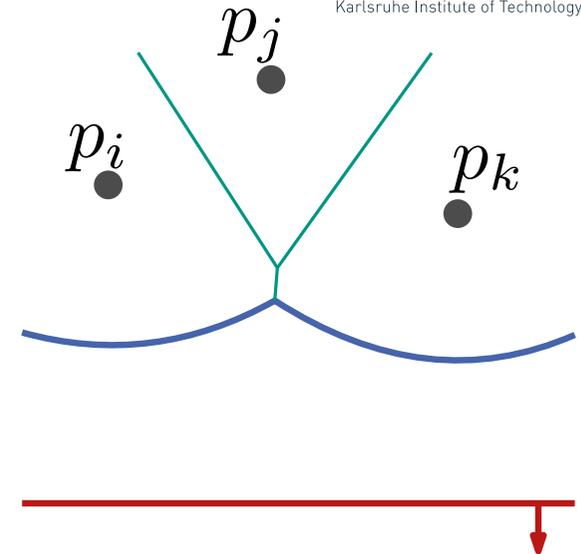
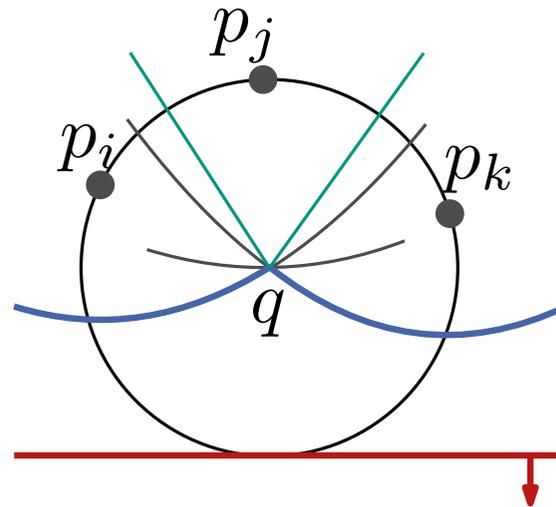
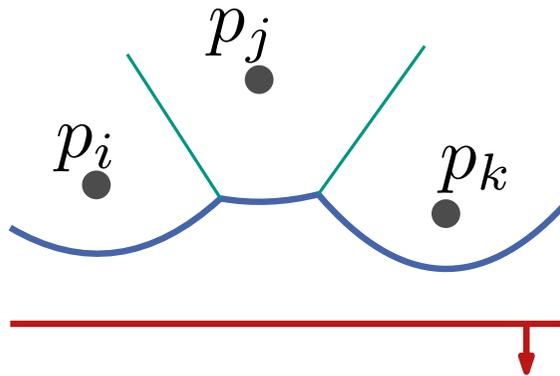
- verschwindet ein Bogen so laufen $f_{p_i}^\ell, f_{p_j}^\ell, f_{p_k}^\ell$ durch einen gemeinsamen Punkt q
- Kreis $C_P(q)$ geht durch p_i, p_j, p_k und berührt ℓ
 $\Rightarrow q$ ist Voronoi-Knoten

Kreis-Events



- verschwindet ein Bogen so laufen $f_{p_i}^l, f_{p_j}^l, f_{p_k}^l$ durch einen gemeinsamen Punkt q
- Kreis $C_P(q)$ geht durch p_i, p_j, p_k und berührt ℓ
 $\Rightarrow q$ ist Voronoi-Knoten

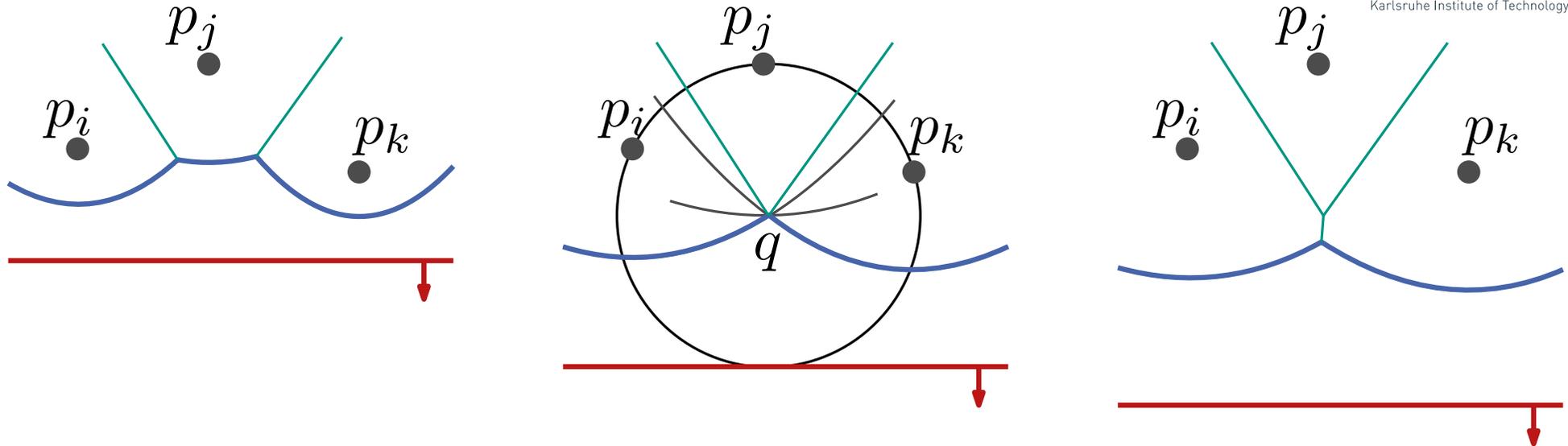
Kreis-Events



- verschwindet ein Bogen so laufen $f_{p_i}^\ell, f_{p_j}^\ell, f_{p_k}^\ell$ durch einen gemeinsamen Punkt q
- Kreis $C_P(q)$ geht durch p_i, p_j, p_k und berührt ℓ
 $\Rightarrow q$ ist Voronoi-Knoten

Def.: Der unterste Punkt des Kreises durch drei Punkte mit konsekutiven Bögen auf β definiert ein **Kreis-Event**.

Kreis-Events

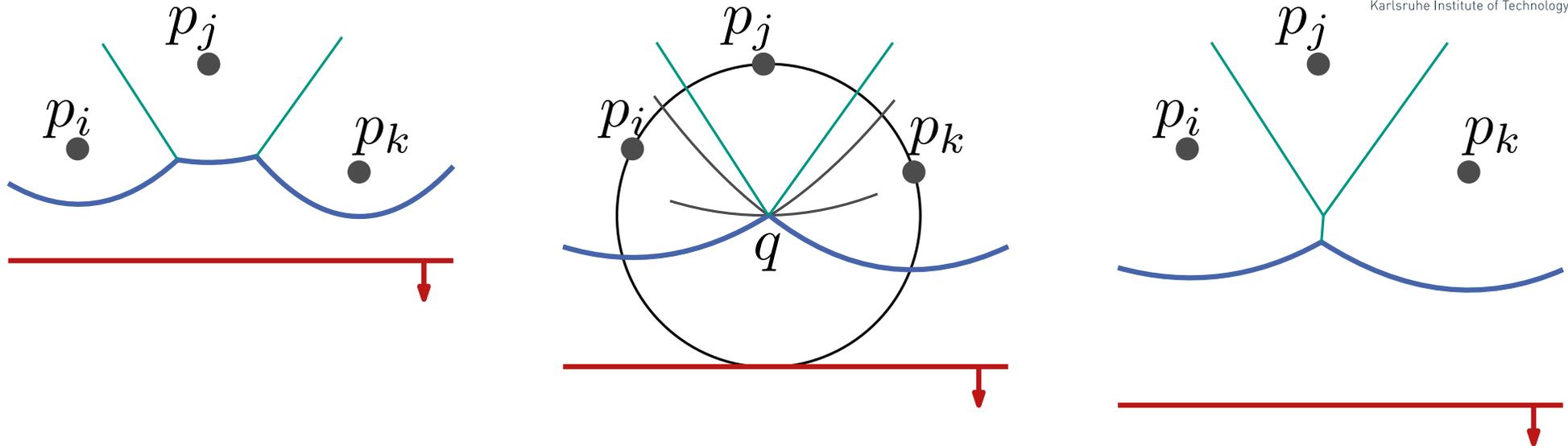


- verschwindet ein Bogen so laufen $f_{p_i}^\ell, f_{p_j}^\ell, f_{p_k}^\ell$ durch einen gemeinsamen Punkt q
- Kreis $C_P(q)$ geht durch p_i, p_j, p_k und berührt ℓ
 $\Rightarrow q$ ist Voronoi-Knoten

Def.: Der unterste Punkt des Kreises durch drei Punkte mit konsekutiven Bögen auf β definiert ein **Kreis-Event**.

Lemma: Bögen von β werden nur durch Kreis-Events entfernt.

Kreis-Events



- verschwindet ein Bogen so laufen $f_{p_i}^\ell, f_{p_j}^\ell, f_{p_k}^\ell$ durch einen gemeinsamen Punkt q
- Kreis $C_P(q)$ geht durch p_i, p_j, p_k und berührt ℓ
 $\Rightarrow q$ ist Voronoi-Knoten

Def.: Der unterste Punkt des Kreises durch drei Punkte mit konsekutiven Bögen auf β definiert ein **Kreis-Event**.

Lemma: Bögen von β werden nur durch Kreis-Events entfernt.

Lemma: Für jeden Voronoi-Knoten gibt es ein Kreis-Event.