Computational Geometry

Tutorial #4 — Voronoi diagrams and enclosing disks

Institut für Betriebssysteme
und Rechnerverbund
Algorithmik

Peter Kramer | November 28th, 2024

Voronoi diagrams

Higher order

Farthest point

Peter Kramer | November 28th, 2024

Voronoi diagrams
Refresh

Voronoi diagrams
Refresh

A Voronoi diagram $\mathrm{Vor}(P)$ partitions a metric space based on which element of the discrete point set P is closest.

Voronoi diagrams
Refresh

A Voronoi diagram $\mathrm{Vor}(P)$ partitions a metric space based on which element of the discrete point set P is closest.

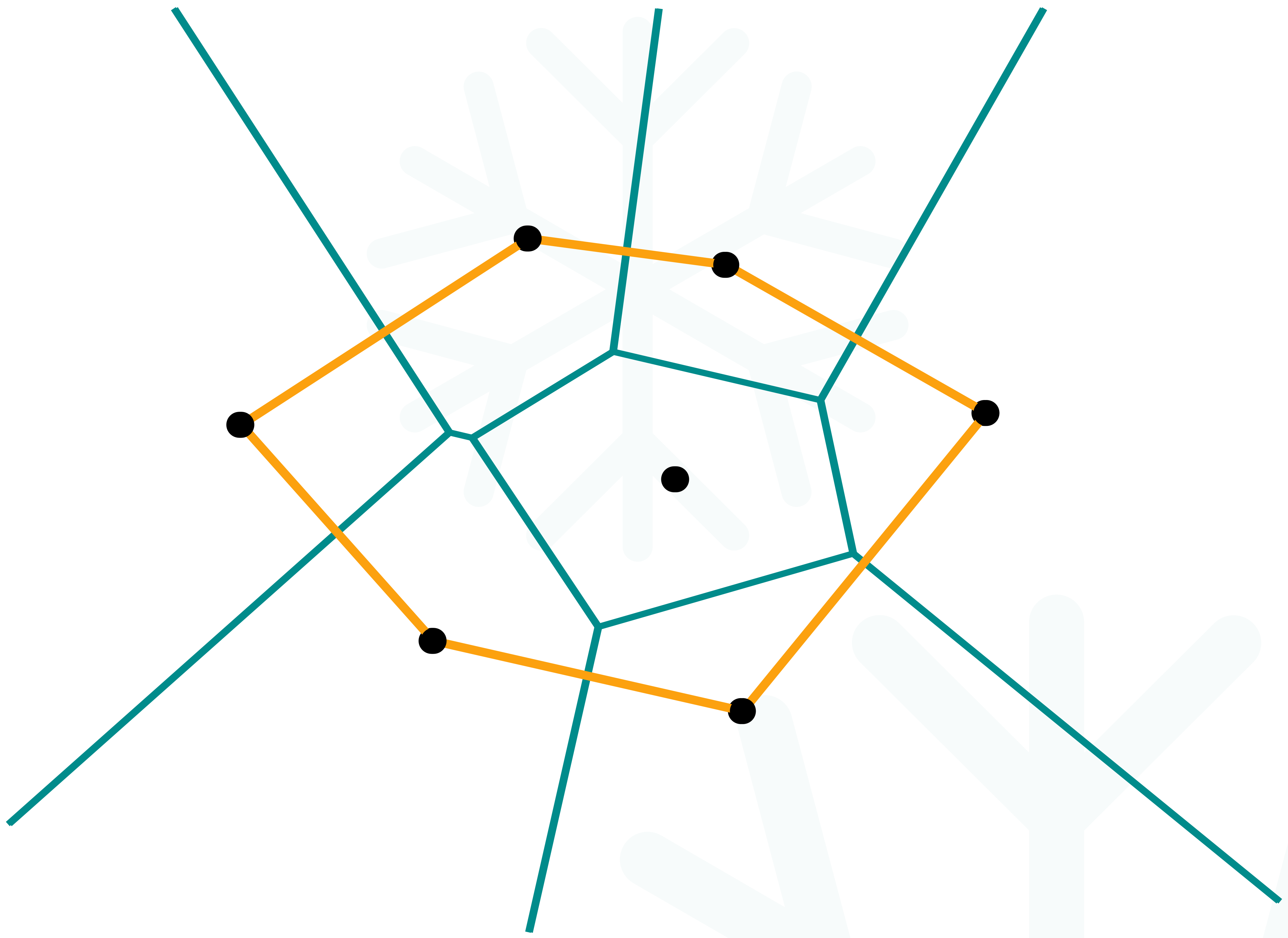
How do the unbounded faces relate to the convex hull $\mathrm{conv}(P)$*?*

Voronoi diagrams

Refresh

A Voronoi diagram $\mathrm{Vor}(P)$ partitions a metric space based on which element of the discrete point set P is closest.

How do the unbounded faces relate to the convex hull $\mathrm{conv}(P)$*?*



Voronoi diagrams
Higher order

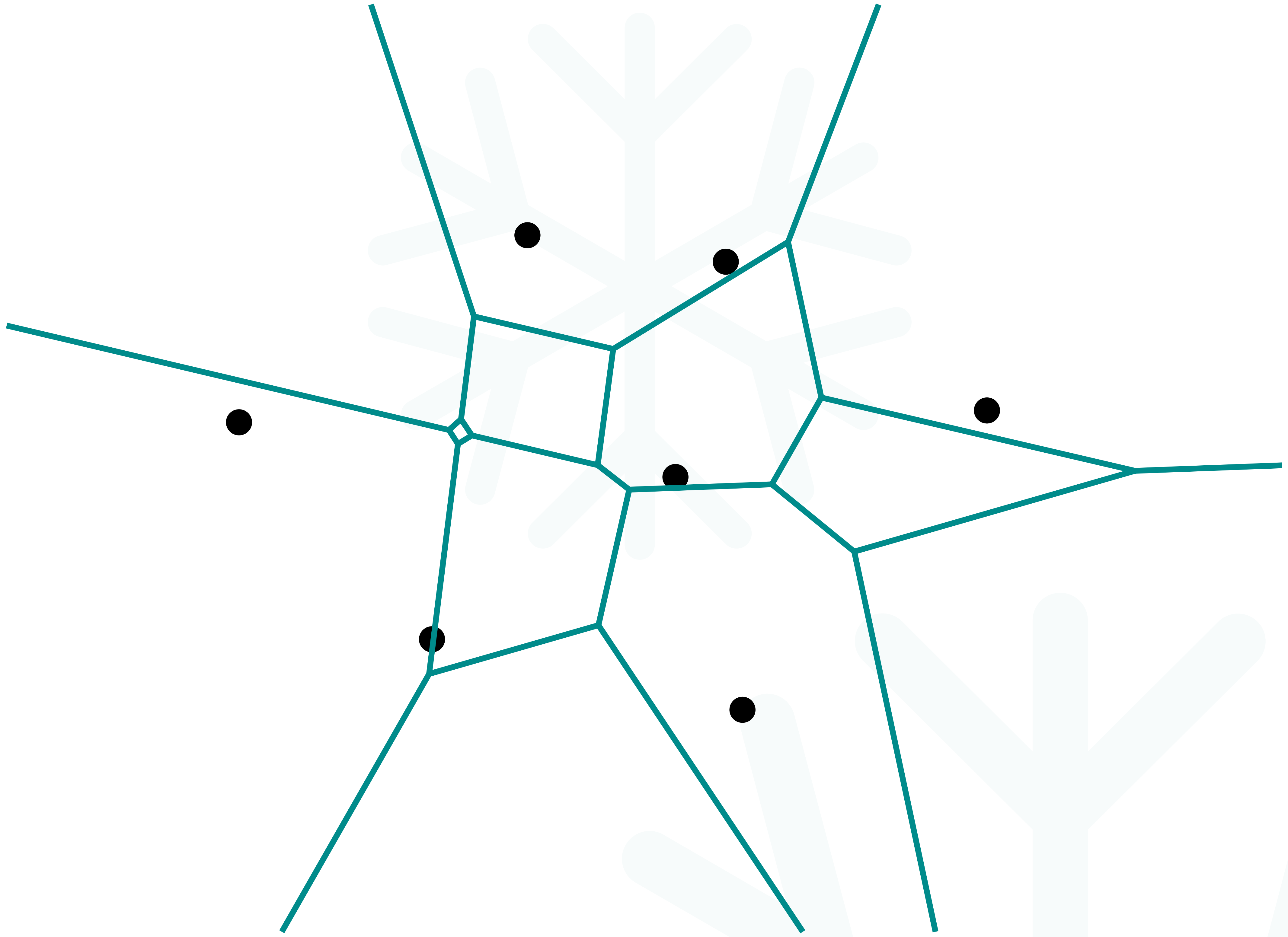
A Voronoi diagram $\mathrm{Vor}(P)$ partitions a metric space based on which element of the discrete point set P is closest.

*What if we wanted to divide based on which **two** points are closest?*

Voronoi diagrams
Higher order

A Voronoi diagram $\mathrm{Vor}(P)$ partitions a metric space based on which element of the discrete point set P is closest.

*What if we wanted to divide based on which **two** points are closest?*

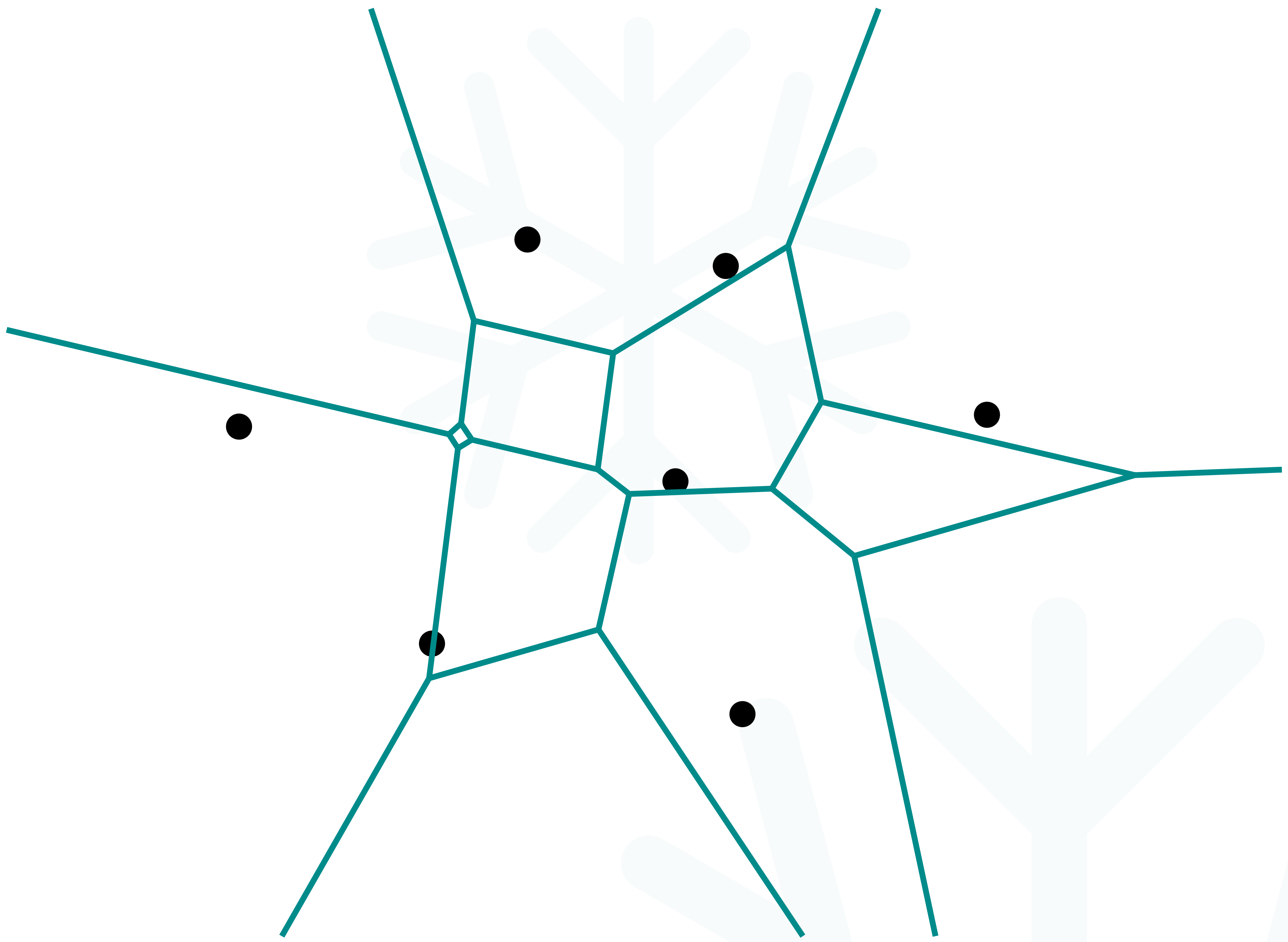


Voronoi diagrams
Higher order

An ith order Voronoi diagram of P divides a metric space based on **which i points** of a discrete set P are closest.

Here: Second order Voronoi diagram.

How can we derive this?



Voronoi diagrams

Higher order

An ith order Voronoi diagram of P divides a metric space based on **which i points** of a discrete set P are closest.

Here: Second order Voronoi diagram.

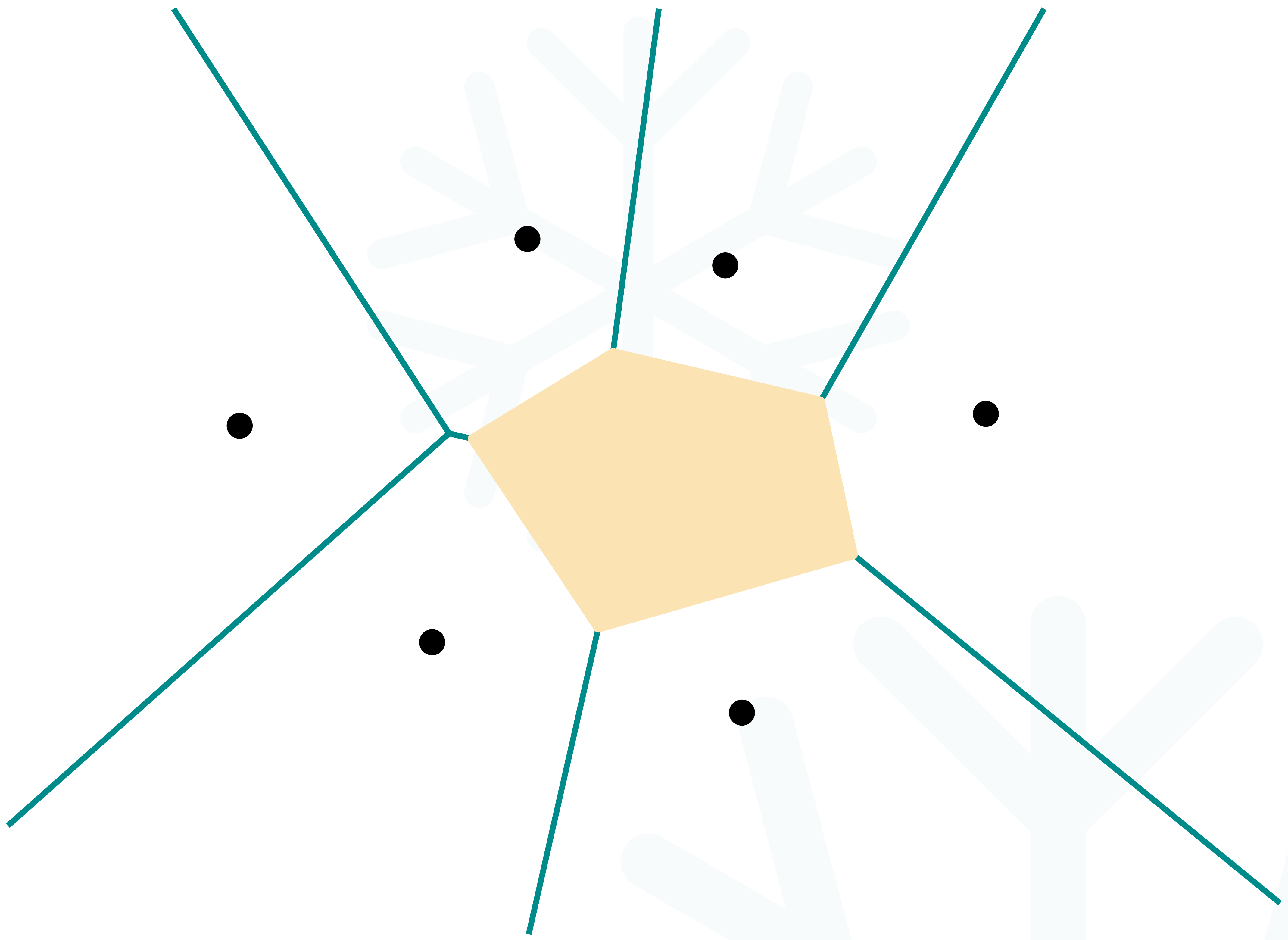
How can we derive this?

Voronoi diagrams
Higher order

An ith order Voronoi diagram of P divides a metric space based on **which i points** of a discrete set P are closest.

Here: Second order Voronoi diagram.

How can we derive this?



Voronoi diagrams

Higher order

An ith order Voronoi diagram of P divides a metric space based on **which i points** of a discrete set P are closest.

Here: Second order Voronoi diagram.

How can we derive this?

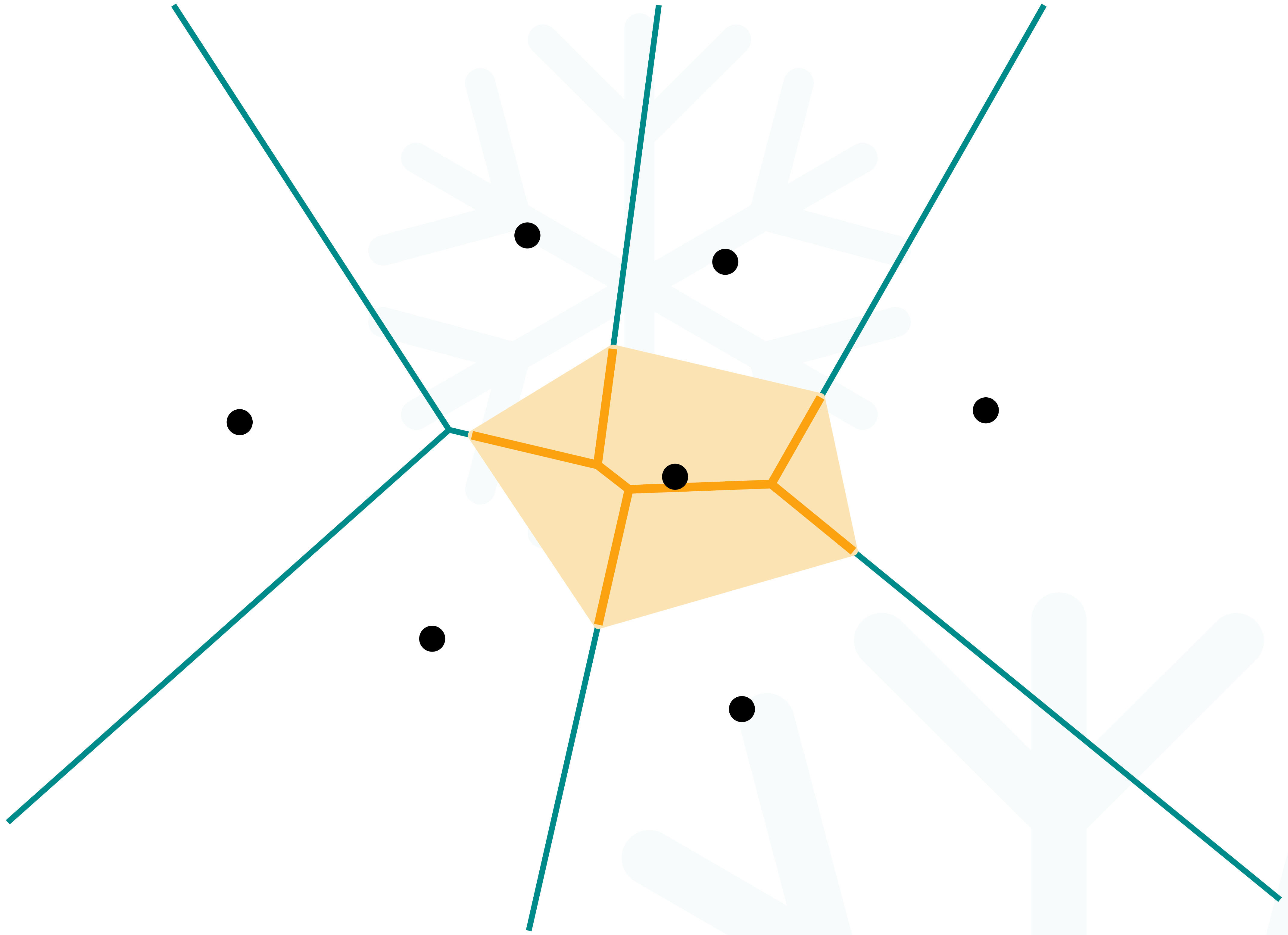
Voronoi diagrams

Higher order

An ith order Voronoi diagram of P divides a metric space based on **which i points** of a discrete set P are closest.

Here: Second order Voronoi diagram.

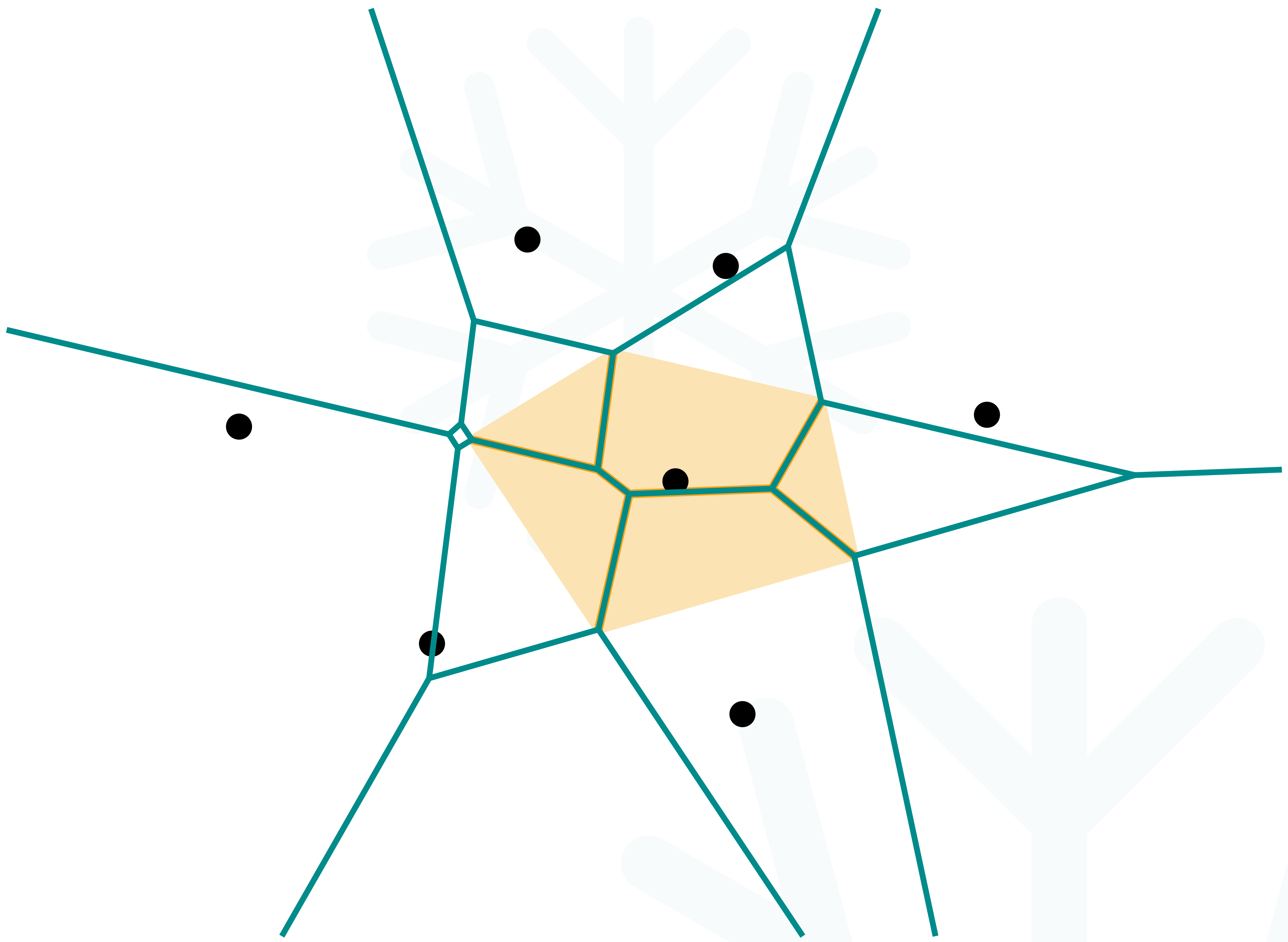
How can we derive this?



Voronoi diagrams

Higher order

An ith order Voronoi diagram of $Vor(P, i)$ divides a metric space based on **which i points** of the discrete set P are closest.



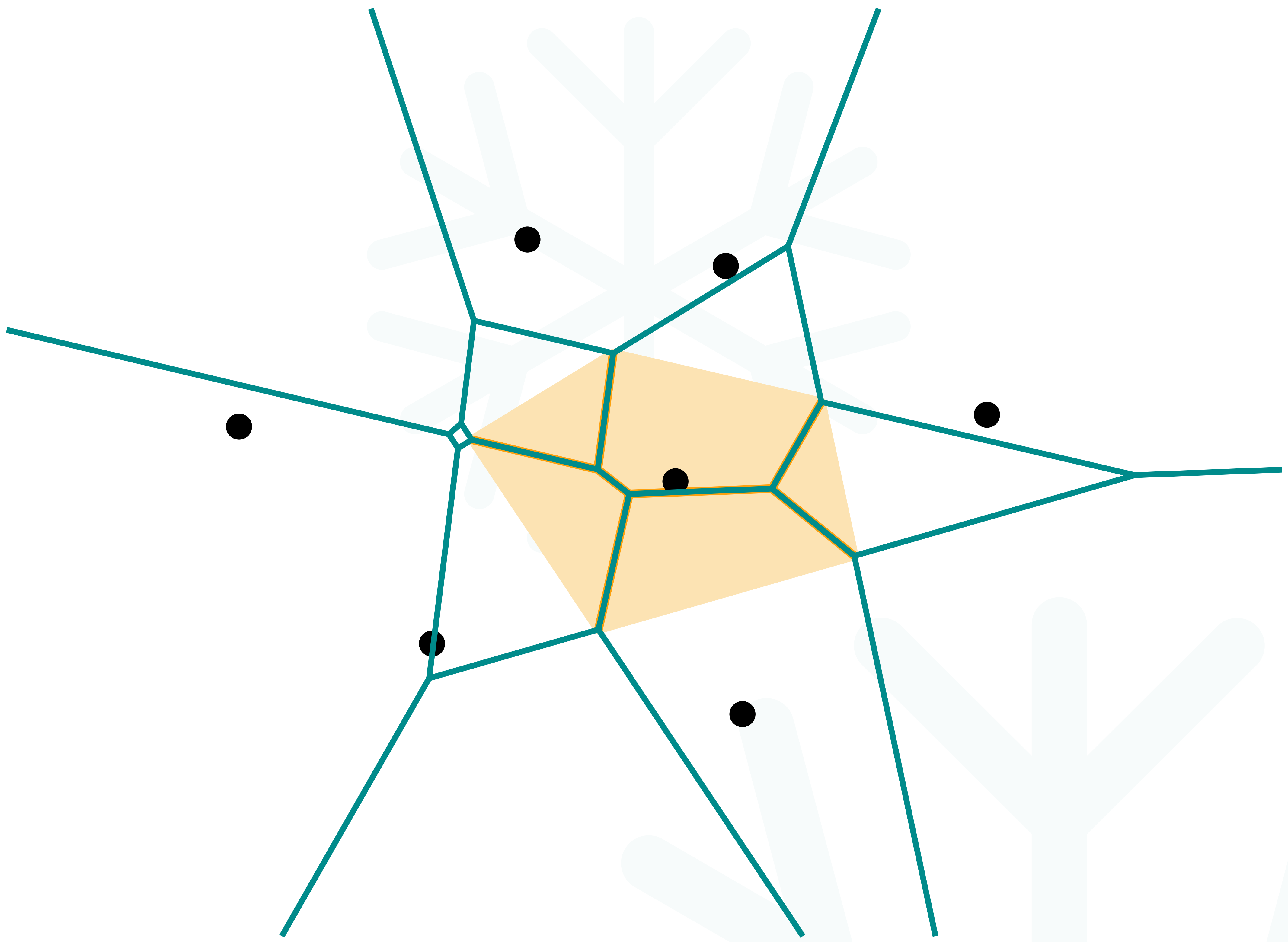
```
Basic idea for Vor(P, i+1):
   For region R in Vor(P, i) do:
      Let P_R = sites in P that define R
      R_i+1 = Vor(P\P_R, i) ∩ R
      Replace R by R_i+1
```

Voronoi diagrams

Higher order

An ith order Voronoi diagram of $Vor(P, i)$ divides a metric space based on **which** i **points** of the discrete set P are closest.

```
Basic idea for Vor(P, i+1):
  For region R in Vor(P, i) do:
    Let P_R = sites in P that define R
    R_i+1 = Vor(P\P_R, i) ∩ R
    Replace R by R_i+1
```



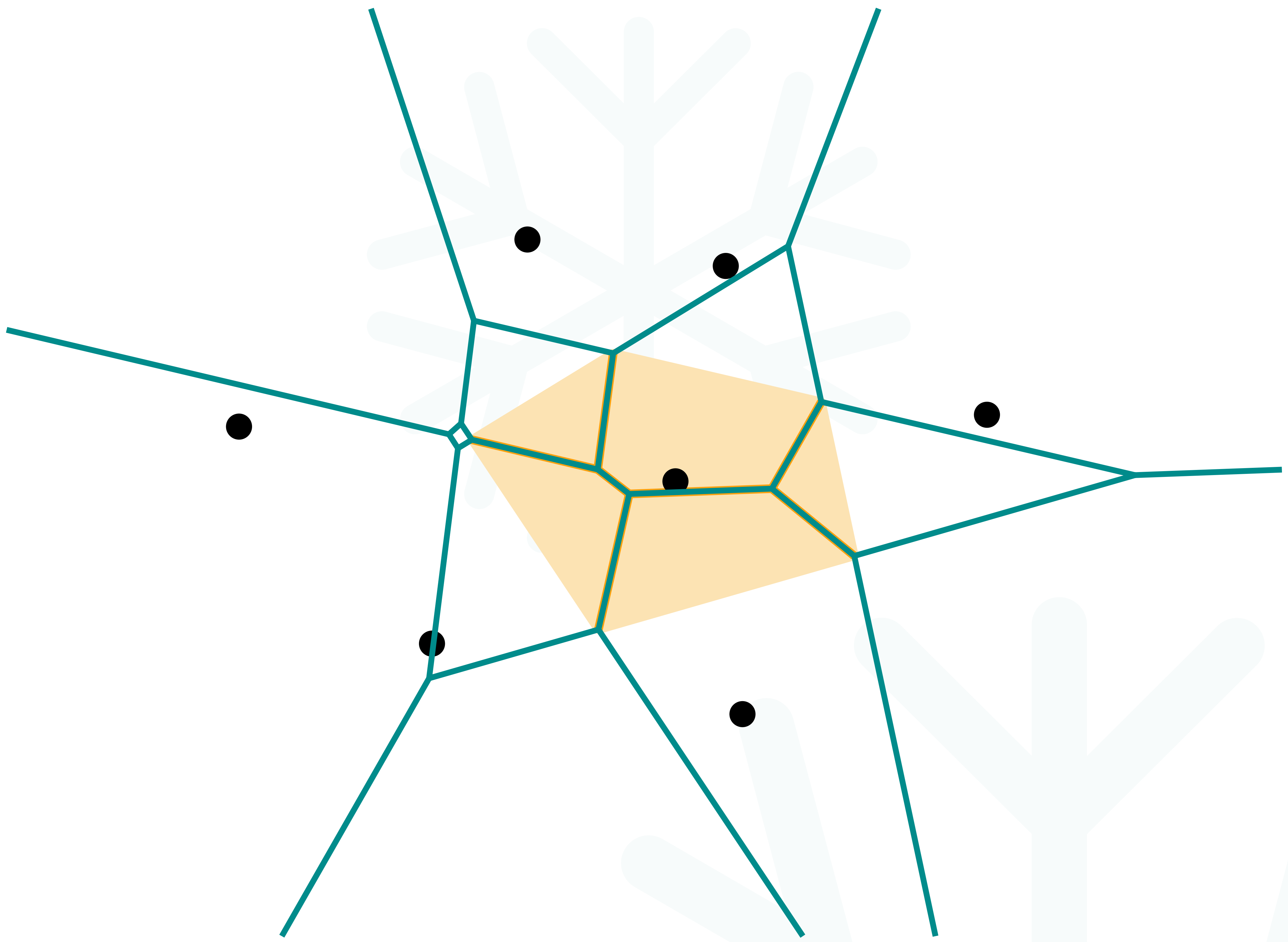
Voronoi diagrams
Higher order

An ith order Voronoi diagram of $Vor(P, i)$ divides a metric space based on **which i points** of the discrete set P are closest.

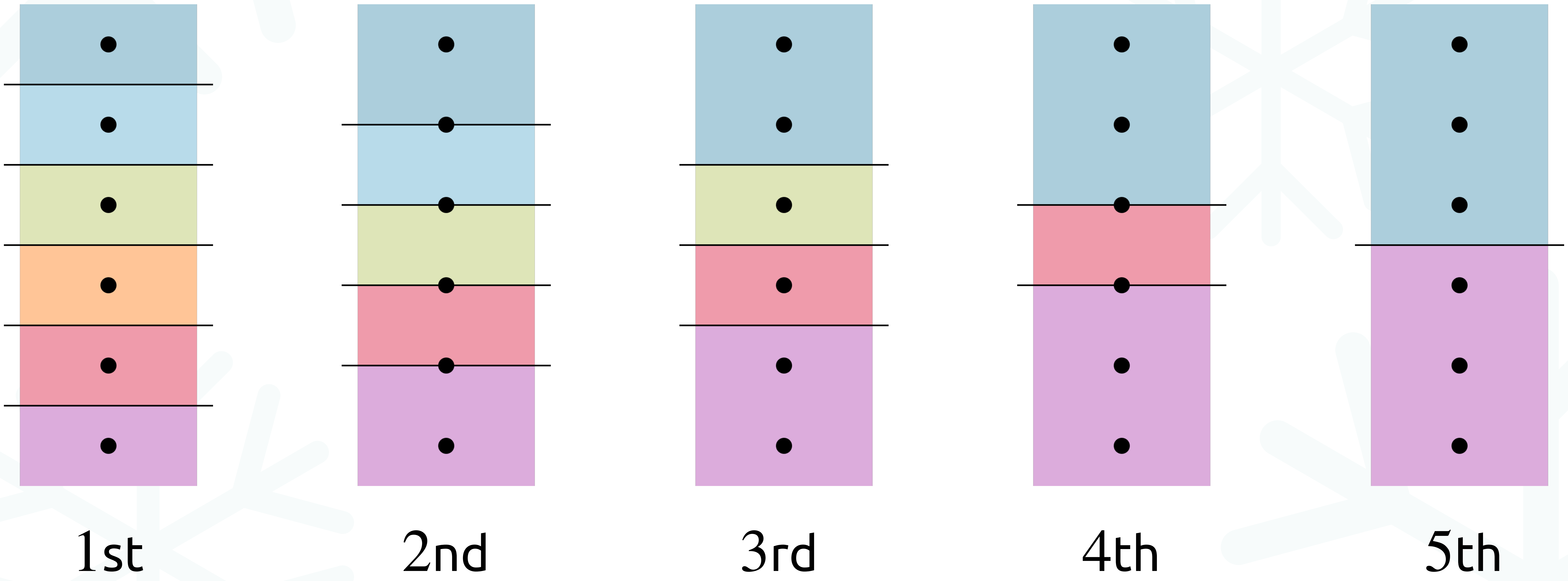
Using better methods:
Theorem E4.1 (Chan and Tsakalidis, 2015):
The ith order Voronoi diagram of n points in the plane can be computed in $\mathcal{O}(n \log n + ni \log i)$.

Theorem E4.2 (Chan et al, 2023):

[… or] in $\mathcal{O}(n \log n + ni)$ expected time.



Degenerate case: Collinearity



Voronoi diagrams

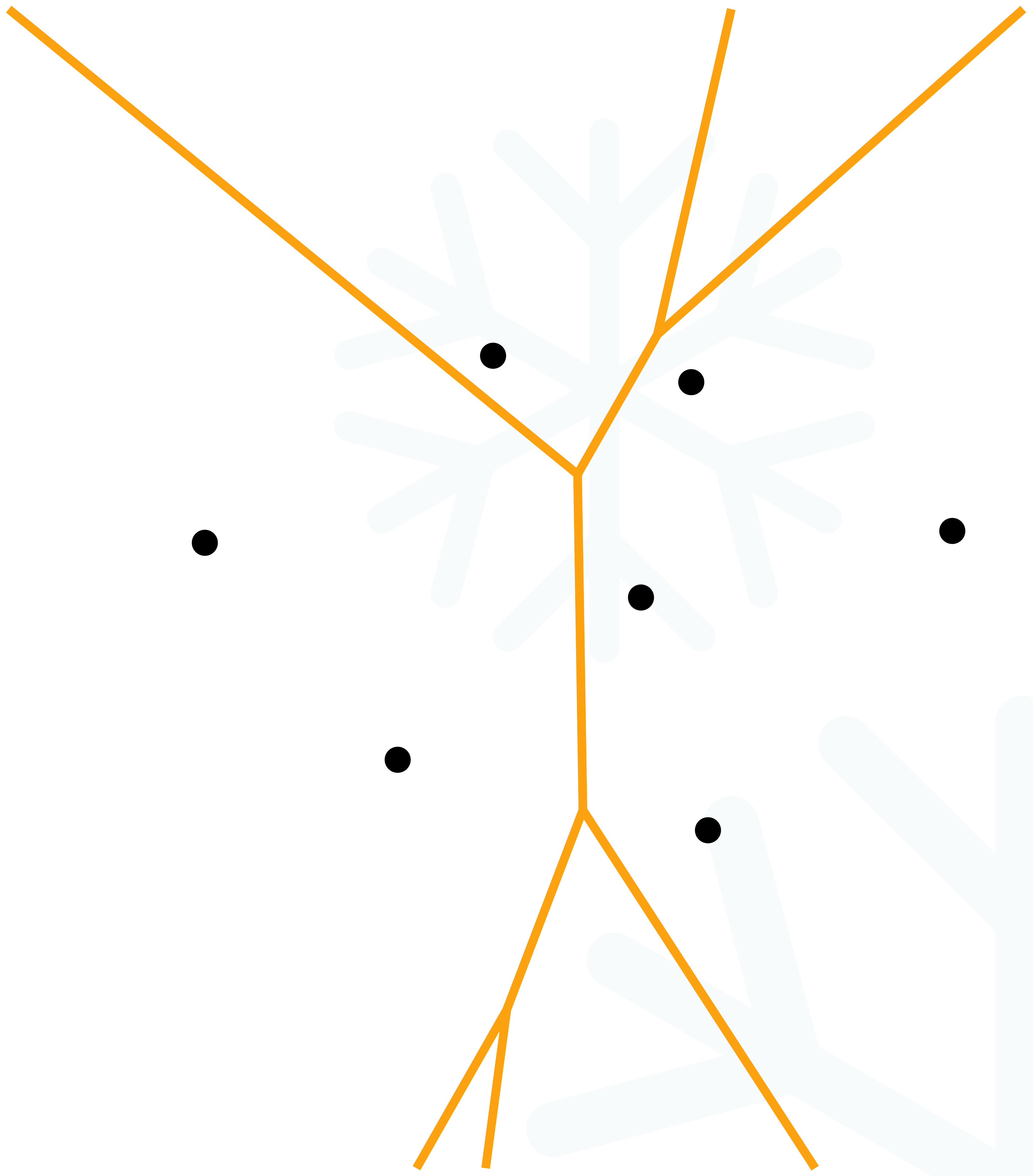
Farthest point

Peter Kramer | November 28th, 2024

Voronoi diagrams
Farthest point

An $(n-1)$th order Voronoi diagram divides a metric space based on which element of a discrete point set P is **farthest**.

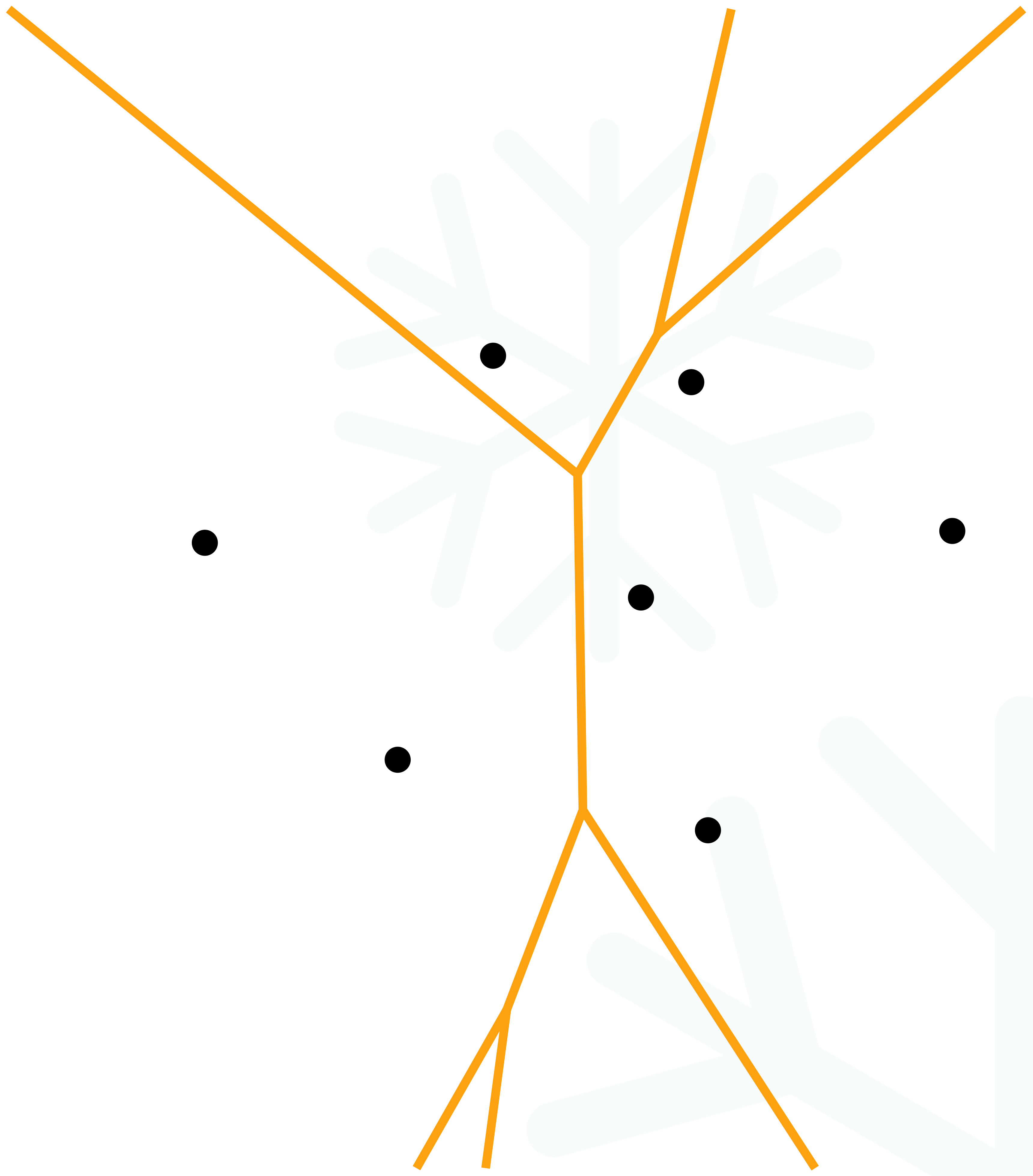
What can you say about this diagram?
How many regions?
What's the graph topology?



Voronoi diagrams
Farthest point

An $(n-1)$th order Voronoi diagram divides a metric space based on which element of a discrete point set P is **farthest**.

Can you think of some relation to the convex hull $\mathrm{conv}(P)$?

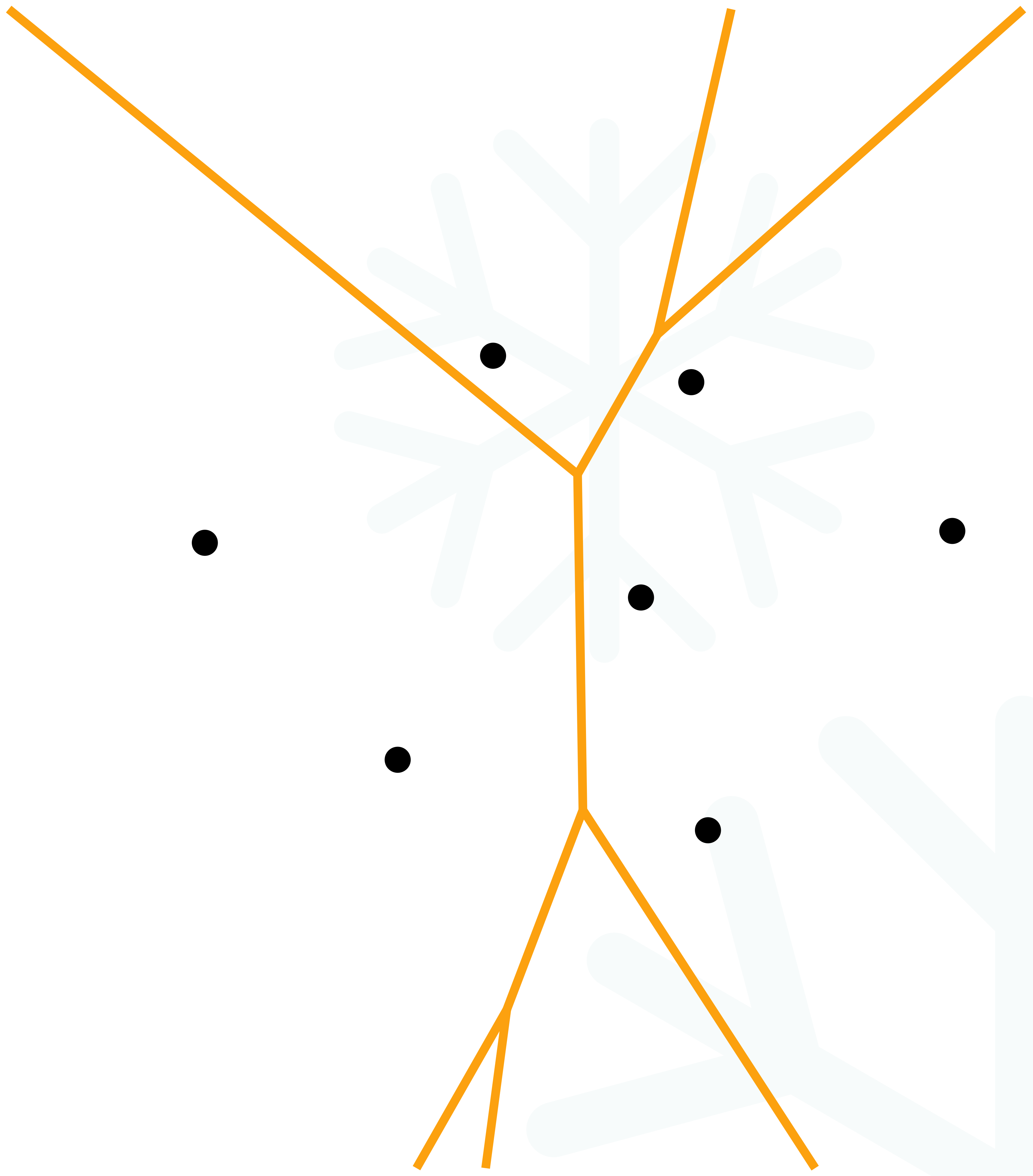


Voronoi diagrams

Farthest point

An $(n-1)$th order Voronoi diagram divides a metric space based on which element of a discrete point set P is **farthest**.

All cells are unbounded, i.e., the dual graph is a tree. A point $p \in P$ has a non-empty Voronoi region exactly if it lies on the boundary of the convex hull $\operatorname{conv}(P)$.

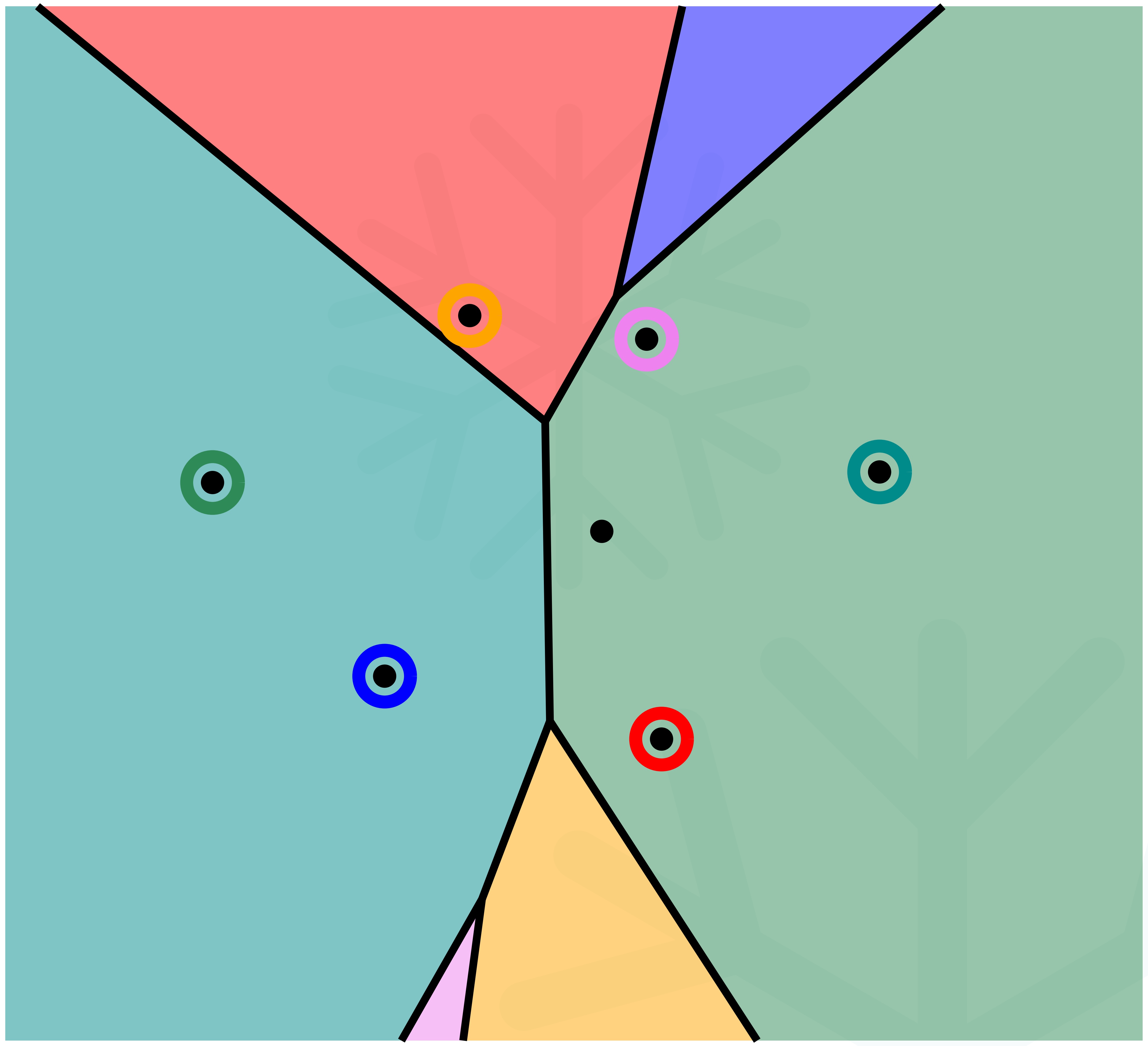


Voronoi diagrams

Farthest point

An $(n-1)$th order Voronoi diagram divides a metric space based on which element of a discrete point set P is **farthest**.

All cells are unbounded, i.e., the dual graph is a tree. A point $p \in P$ has a non-empty Voronoi region exactly if it lies on the boundary of the convex hull $\mathrm{conv}(P)$.



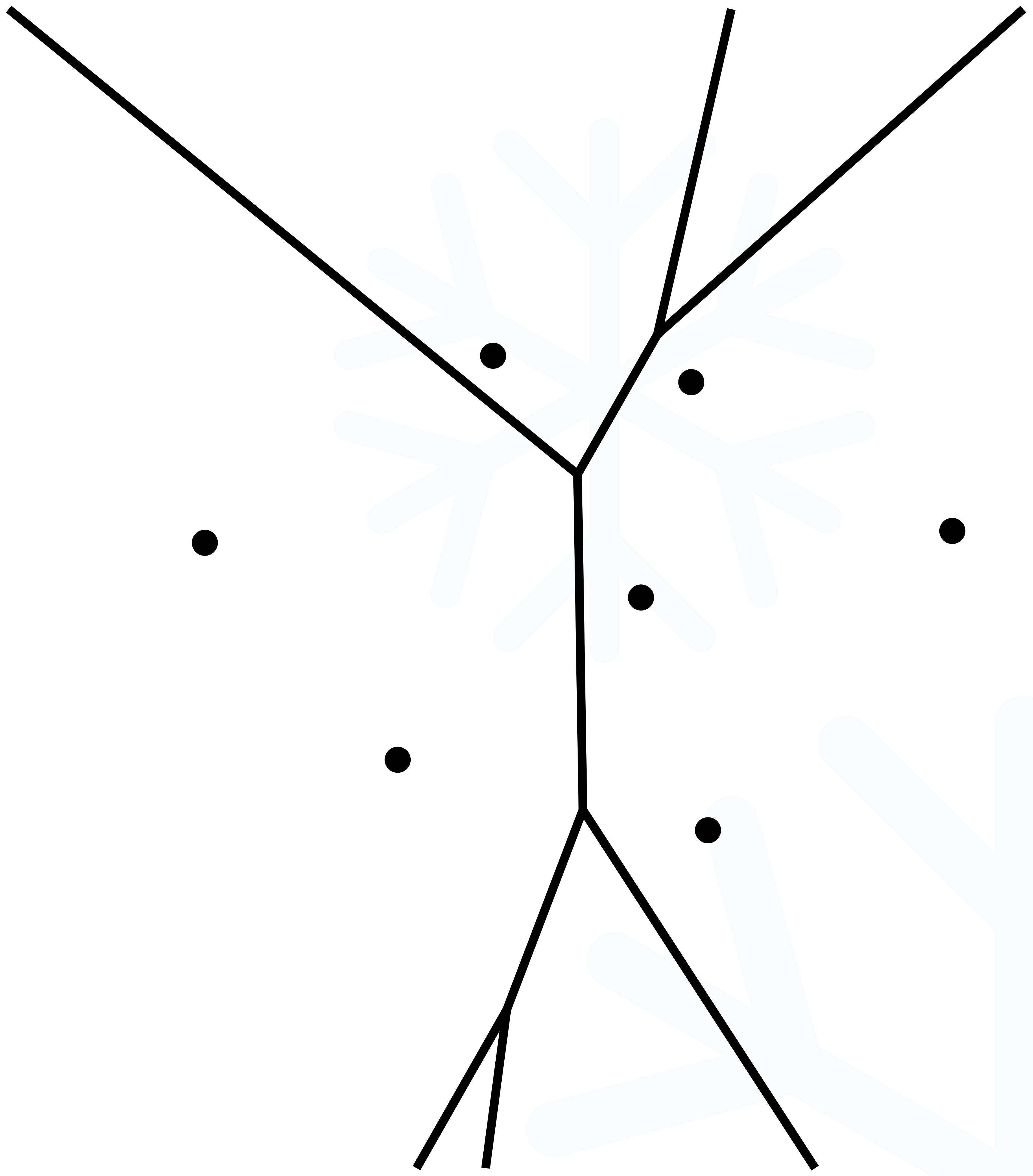
Voronoi diagrams

Farthest point

An $(n-1)$th order Voronoi diagram ***(farthest-point Voronoi diagram)*** divides a metric space based on which element of a discrete point set P is **farthest**.

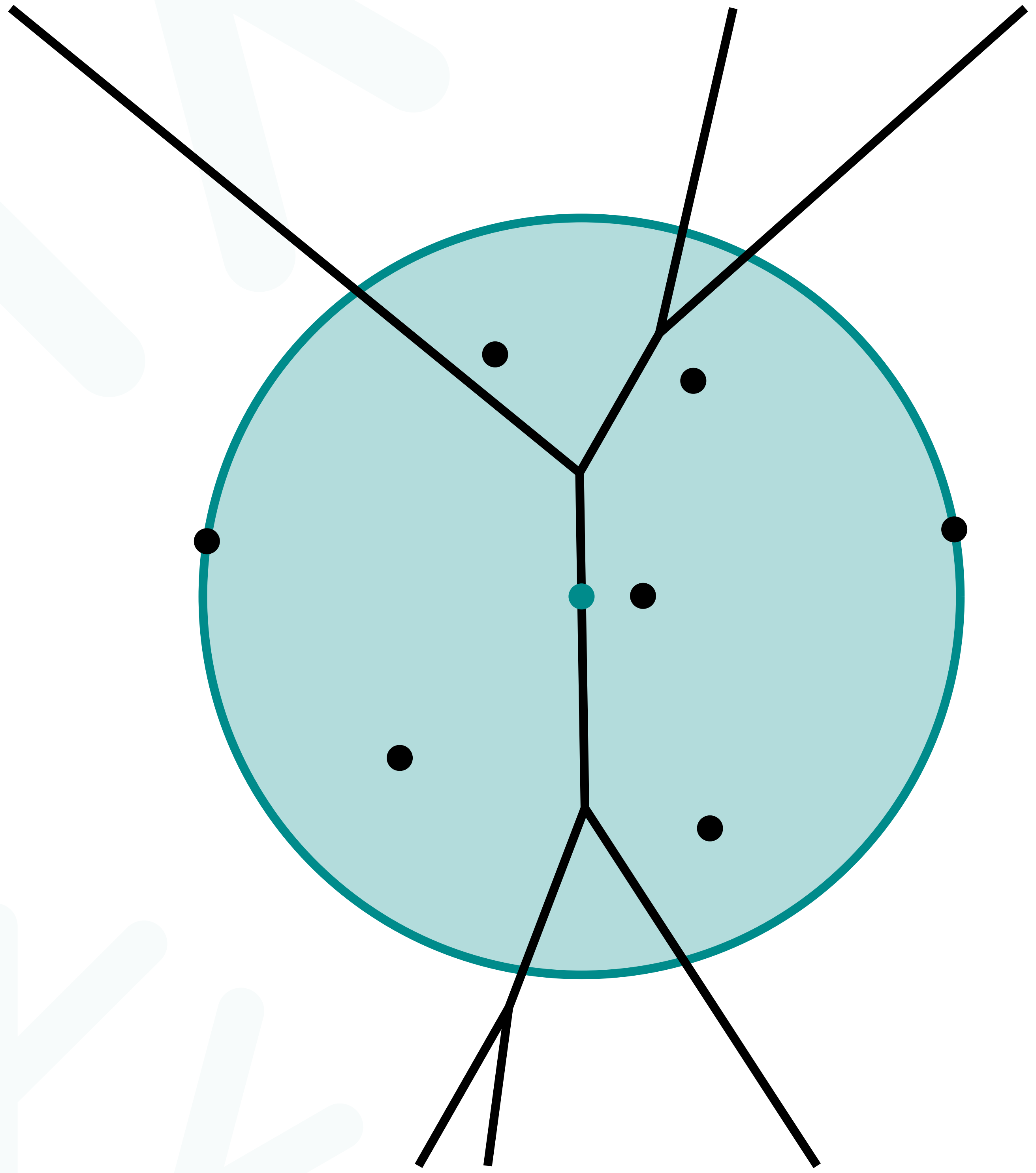
For first order, we had the empty circumcircle property *(what's this?)*.

Does a similar property hold for every vertex and edge of the farthest point Voronoi diagram?

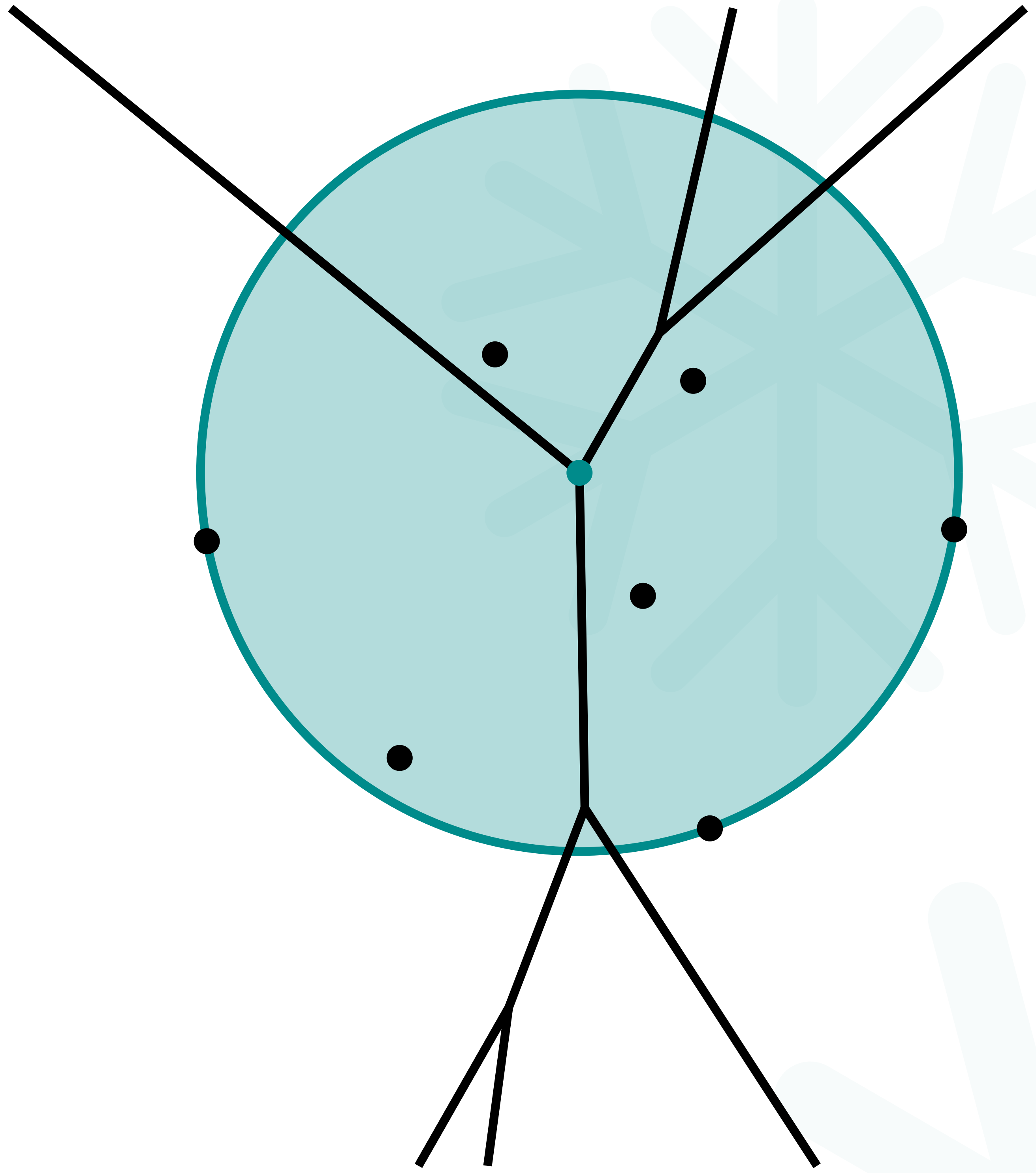


Voronoi diagrams

Farthest point



Edges are equidistant to **two** sites, closer to all other.

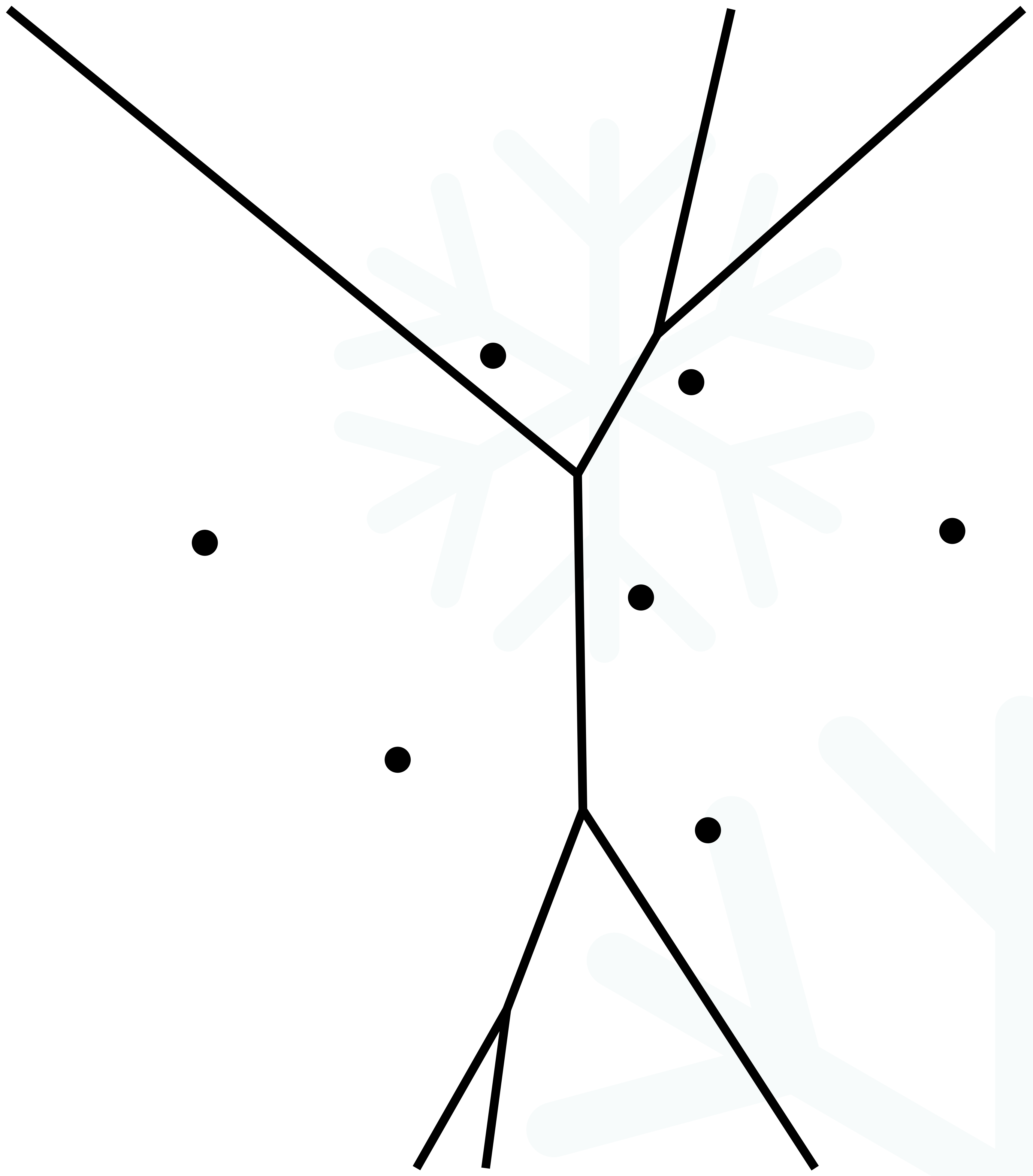


Vertices are equidistant to **three** sites, closer to all others.

Voronoi diagrams
Farthest point

An $(n-1)$th order Voronoi diagram divides a metric space based on which element of a discrete point set P is **farthest**.

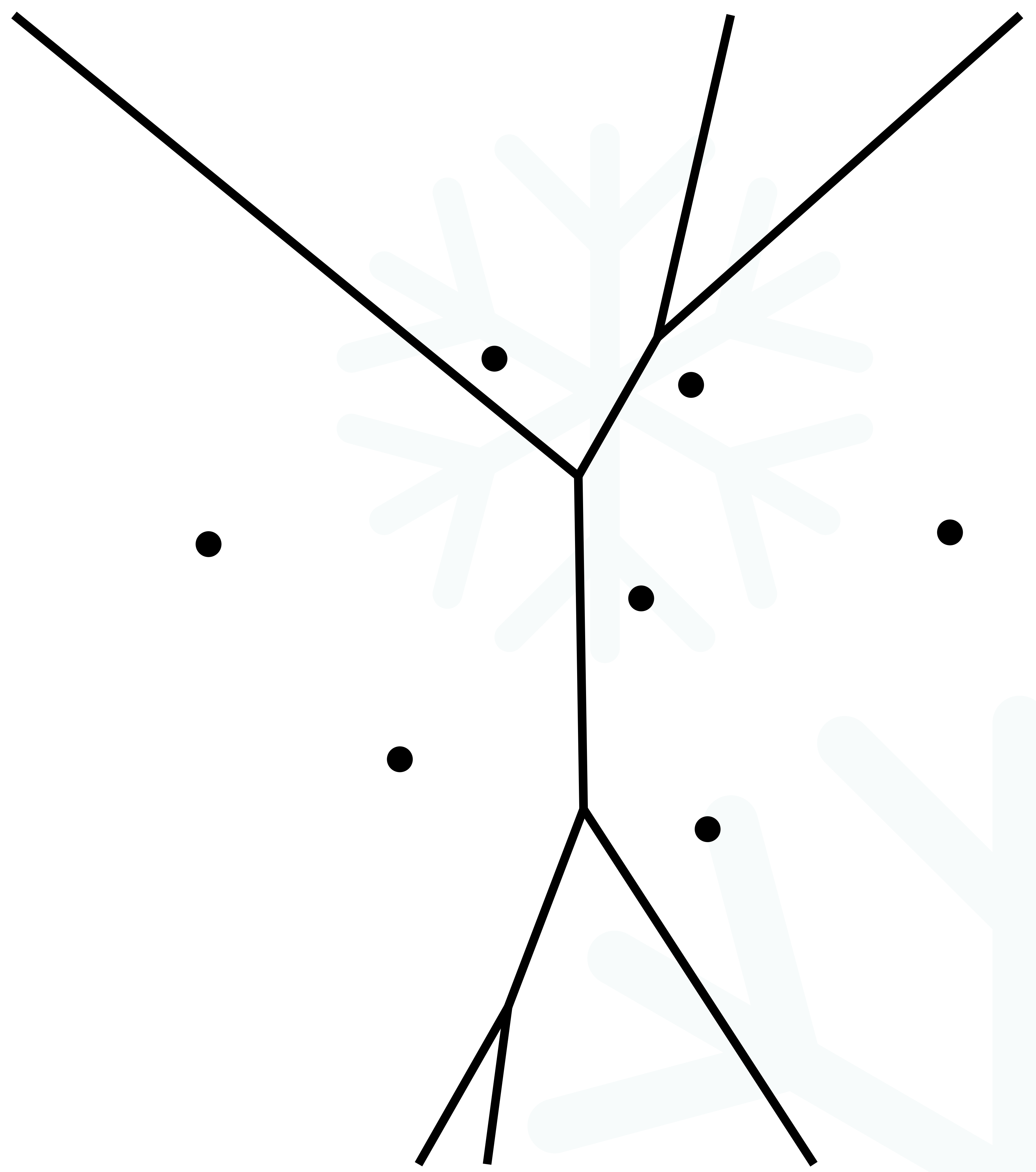
Theorem E4.2 (Cheong et al., 2011):
The farthest point Voronoi diagram of n points in the plane can be computed in $\mathcal{O}(n \log^3 n)$ time.



Voronoi diagrams
Farthest point

An $(n-1)$th order Voronoi diagram divides a metric space based on which element of a discrete point set P is **farthest**.

Using a DCEL, this graph structure can be stored such that the corresponding sites to each face, vertex, and edge can be accessed in $\mathcal{O}(1)$ time!



Homework Sheet #3

Peter Kramer | November 28th, 2024

Computational Geometry – Sheet 3 **Winter 2024/2025**
Prof. Dr. Sándor P. Fekete **Due** 12.12.2024
Peter Kramer **Discussion** 19.12.2024

Please submit your handwritten answers in pairs, using the box in front of `IZ338` *before the exercise timeslot on the due date above. Make sure to include your* full names, matriculation numbers, *and the* programmes *that you are enrolled in. In accordance with the* `guidelines` *of the TU Braunschweig, using AI tools to solve any part of the exercises is* ***not permitted***.

Exercise 1 (Safehouse Problem). **(5+15 points)**

You've successfully pulled off your biggest heist yet – you've stolen the golden Leibniz cookie! However, agents of `baked goods manufacturing co.` *are now on your tail, and all roads out of the city have been locked down in search for the famous symbol of delicacy: You cannot leave. You decide that your best bet is to find a safehouse in town as far as possible from all cookie outles, hide the cookie there, and lie low.*

Given an axis-aligned rectangle R in the Euclidean plane $\mathbb{R}^2$ that defines the city limits and the locations of nearby (both inside and outside the city) cookie outlets $c_1, c_2, \ldots, c_n$, you need a location inside R that maximizes the distance to the closest cookie outlet, see Fig. 1. (Hint: Start by thinking about a geometric suitable structure from the lecture to discretize the problem.)

a) Identify a *(finite!)* set of candidate locations in R to choose from. Argue why an optimal solution is contained in this set.

b) Design an $\mathcal{O}(n \log n)$ time algorithm based on your candidates and prove its correctness.

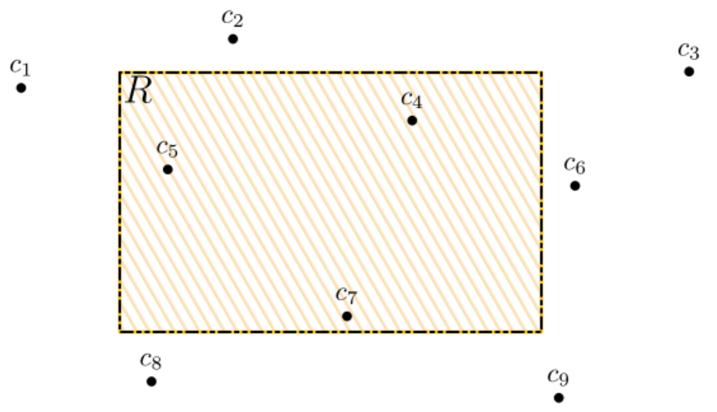


Figure 1: You require a location inside R that maximizes the distance to the nearest cookie outlet c_i.

Exercise 2 (Min enclosing disk). **(5+15 points)**

Let $\mathcal{P}$ be a set of n points in the Euclidean plane in *general position*, such that no four points in $\mathcal{P}$ are concyclic: No four points lie on a common circle. A *minimum enclosing disk* $md(\mathcal{P})$ is a disk with minimal radius that contains $\mathcal{P}$. Let $c \in \mathbb{R}^2$ and $r \in \mathbb{R}$ be its center and radius.

Using concepts from the lecture and the tutorial, design an algorithm that determines $md(\mathcal{P})$ in $\mathcal{O}(n^2)$ time. (Hint: Review Tutorial 4)

1/1

Computational Geometry – Sheet 3 **Winter 2024/2025**

Prof. Dr. Sándor P. Fekete **Due** 12.12.2024

Peter Kramer **Discussion** 19.12.2024

Please submit your handwritten answers in pairs, using the box in front of `IZ338` *before the exercise timeslot on the due date above. Make sure to include your full names, matriculation numbers, and the programmes that you are enrolled in. In accordance with the* `guidelines` *of the TU Braunschweig, using AI tools to solve any part of the exercises is* **not permitted**.

Exercise 1 (Safehouse Problem). **(5+15 points)**

You've successfully pulled off your biggest heist yet – you've stolen the golden Leibniz cookie! However, agents of `baked goods manufacturing co.` *are now on your tail, and all roads out of the city have been locked down in search for the famous symbol of delicacy: You cannot leave. You decide that your best bet is to find a safehouse in town as far as possible from all cookie outles, hide the cookie there, and lie low.*

Given an axis-aligned rectangle R in the Euclidean plane $\mathbb{R}^2$ that defines the city limits and the locations of nearby (both inside and outside the city) cookie outlets $c_1, c_2, \ldots, c_n$, you need a location inside R that maximizes the distance to the closest cookie outlet, see Fig. 1. (Hint: Start by thinking about a geometric suitable structure from the lecture to discretize the problem.)

a) Identify a *(finite!)* set of candidate locations in R to choose from. Argue why an optimal solution is contained in this set.

b) Design an $\mathcal{O}(n \log n)$ time algorithm based on your candidates and prove its correctness.

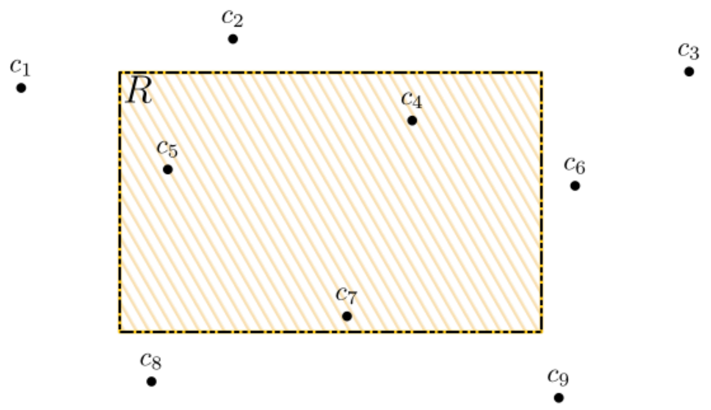


Figure 1: You require a location inside R that maximizes the distance to the nearest cookie outlet c_i.

Exercise 2 (Min enclosing disk). **(5+15 points)**

Let $\mathcal{P}$ be a set of n points in the Euclidean plane in *general position*, such that no four points in $\mathcal{P}$ are concyclic: No four points lie on a common circle. A *minimum enclosing disk* $md(\mathcal{P})$ is a disk with minimal radius that contains $\mathcal{P}$. Let $c \in \mathbb{R}^2$ and $r \in \mathbb{R}$ be its center and radius.

Using concepts from the lecture and the tutorial, design an algorithm that determines $md(\mathcal{P})$ in $\mathcal{O}(n^2)$ time. (Hint: Review Tutorial 4)

1/1

Exercise 1 (Safehouse Problem). **(5+15 points)**

You've successfully pulled off your biggest heist yet – you've stolen the golden Leibniz cookie! However, agents of `baked goods manufacturing co.` *are now on your tail, and all roads out of the city have been locked down in search for the famous symbol of delicacy: You cannot leave. You decide that your best bet is to find a safehouse in town as far as possible from all cookie outles, hide the cookie there, and lie low.*

Given an axis-aligned rectangle R in the Euclidean plane $\mathbb{R}^2$ that defines the city limits and the locations of nearby (both inside and outside the city) cookie outlets $c_1, c_2, \ldots, c_n$, you need a location inside R that maximizes the distance to the closest cookie outlet, see Fig. 1. (Hint: Start by thinking about a geometric suitable structure from the lecture to discretize the problem.)

a) Identify a *(finite!)* set of candidate locations in R to choose from. Argue why an optimal solution is contained in this set.

b) Design an $\mathcal{O}(n \log n)$ time algorithm based on your candidates and prove its correctness.

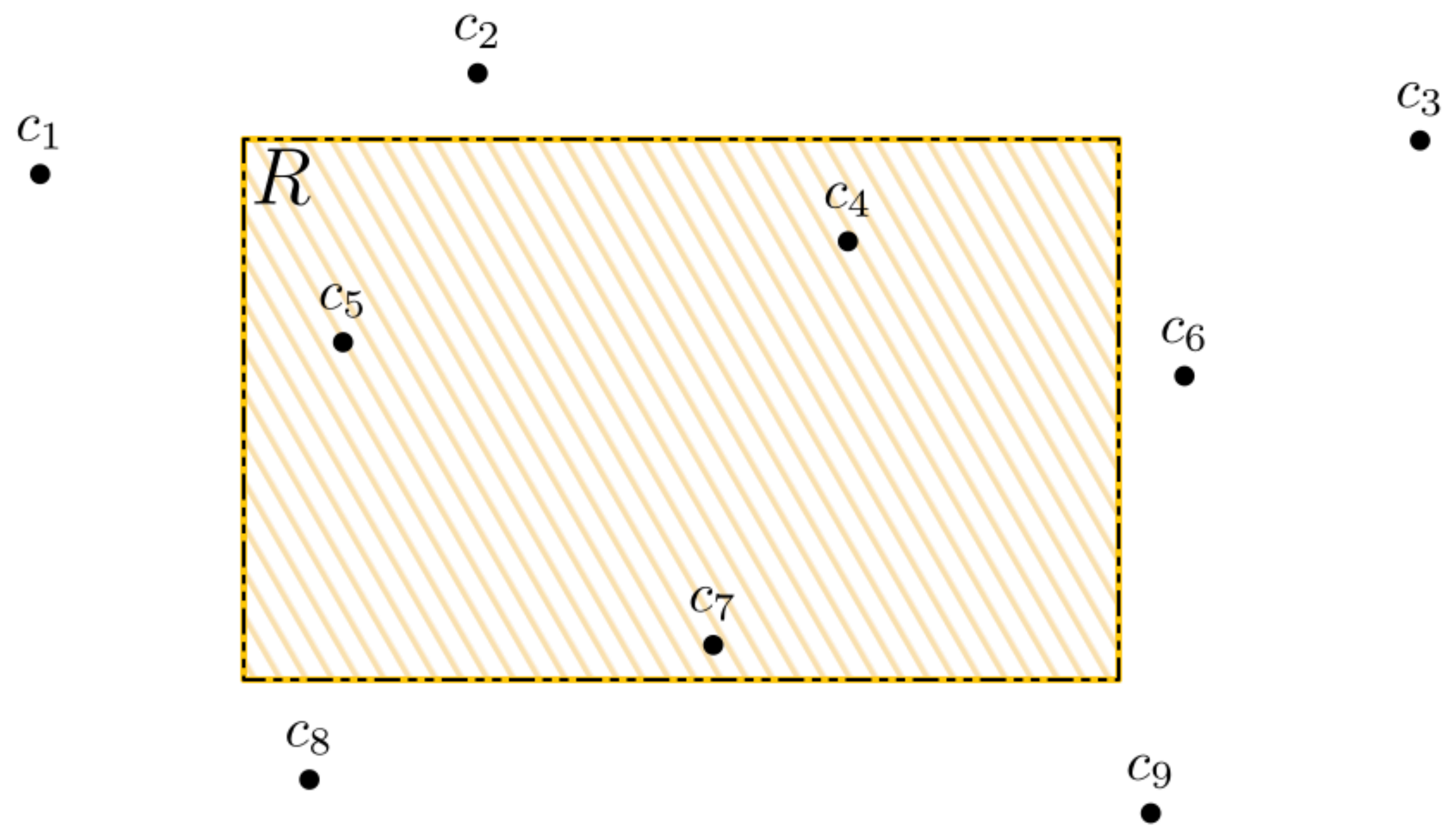


Figure 1: You require a location inside R that maximizes the distance to the nearest cookie outlet c_i.

Computational Geometry – Sheet 3 **Winter 2024/2025**
Prof. Dr. Sándor P. Fekete **Due** 12.12.2024
Peter Kramer **Discussion** 19.12.2024

Please submit your handwritten answers in pairs, using the box in front of `IZ338` *before the exercise timeslot on the due date above. Make sure to include your full names, matriculation numbers, and the programmes that you are enrolled in. In accordance with the* `guidelines` *of the TU Braunschweig, using AI tools to solve any part of the exercises is* **not permitted**.

Exercise 1 (Safehouse Problem). **(5+15 points)**

You've successfully pulled off your biggest heist yet – you've stolen the golden Leibniz cookie! However, agents of `baked goods manufacturing co.` *are now on your tail, and all roads out of the city have been locked down in search for the famous symbol of delicacy: You cannot leave. You decide that your best bet is to find a safehouse in town as far as possible from all cookie outles, hide the cookie there, and lie low.*

Given an axis-aligned rectangle R in the Euclidean plane $\mathbb{R}^2$ that defines the city limits and the locations of nearby (both inside and outside the city) cookie outlets $c_1, c_2, \ldots, c_n$, you need a location inside R that maximizes the distance to the closest cookie outlet, see Fig. 1. (Hint: Start by thinking about a geometric suitable structure from the lecture to discretize the problem.)

a) Identify a *(finite!)* set of candidate locations in R to choose from. Argue why an optimal solution is contained in this set.

b) Design an $\mathcal{O}(n \log n)$ time algorithm based on your candidates and prove its correctness.

1/1

Exercise 1 (Safehouse Problem). **(5+15 points)**

You've successfully pulled off your biggest heist yet – you've stolen the golden Leibniz cookie! However, agents of `baked goods manufacturing co.` *are now on your tail, and all roads out of the city have been locked down in search for the famous symbol of delicacy: You cannot leave. You decide that your best bet is to find a safehouse in town as far as possible from all cookie outles, hide the cookie there, and lie low.*

Given an axis-aligned rectangle R in the Euclidean plane $\mathbb{R}^2$ that defines the city limits and the locations of nearby (both inside and outside the city) cookie outlets $c_1, c_2, \ldots, c_n$, you need a location inside R that maximizes the distance to the closest cookie outlet, see Fig. 1. (Hint: Start by thinking about a geometric suitable structure from the lecture to discretize the problem.)

a) Identify a *(finite!)* set of candidate locations in R to choose from. Argue why an optimal solution is contained in this set.

b) Design an $\mathcal{O}(n \log n)$ time algorithm based on your candidates and prove its correctness.

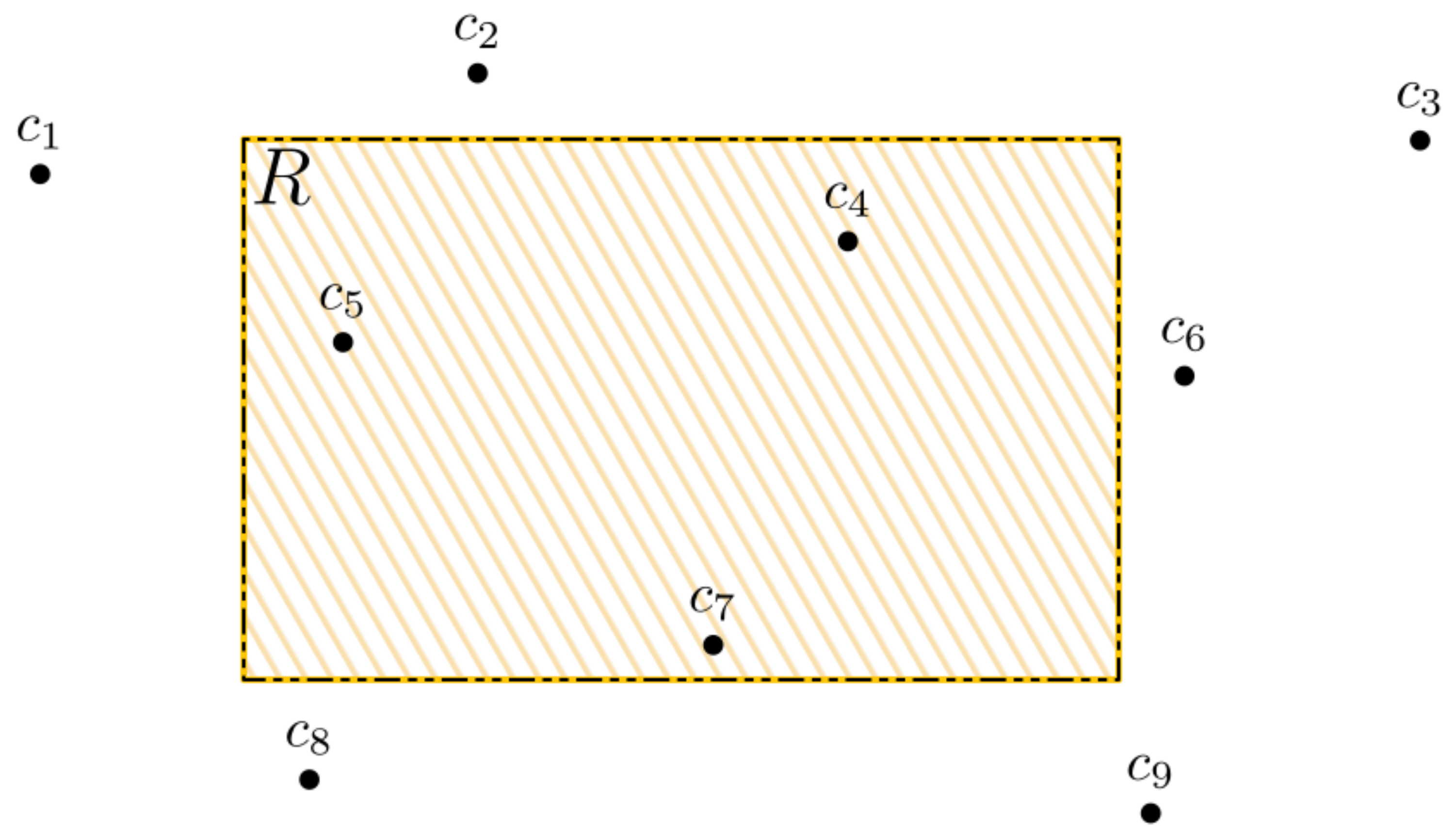


Figure 1: You require a location inside R that maximizes the distance to the nearest cookie outlet c_i.

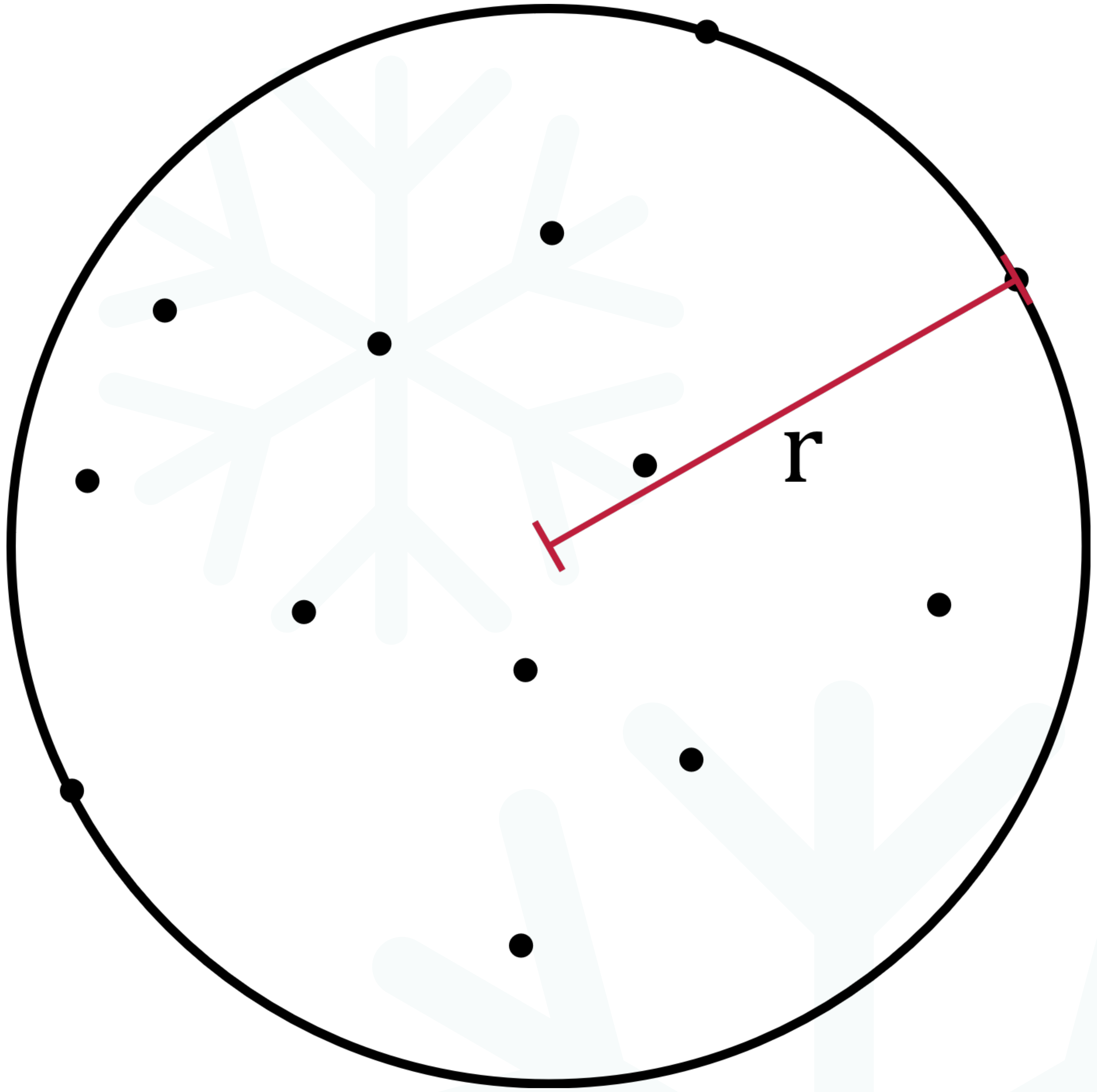
Exercise 2 (Minimal enclosing disk). **(5+15 points)**

Let $\mathcal{P}$ be a set of n points in the Euclidean plane in *general position*, such that no four points in $\mathcal{P}$ are concyclic: No four points lie on a common circle. A *minimal enclosing disk* $md(\mathcal{P})$ is a disk with minimal radius that contains $\mathcal{P}$. Let $c \in \mathbb{R}^2$ and $r \in \mathbb{R}$ be its center and radius.

Using concepts from the lecture and the tutorial, design an algorithm that determines $md(\mathcal{P})$ in less than $\mathcal{O}(n^2)$ time, or better. (Hint: Review Tutorial 4)

Given: Points $P := p_1, \ldots, p_n$ in the Euclidean plane, in general position (no four concyclic points).

Wanted: An enclosing disk $\mathrm{md}(P)$ of **minimal radius** r.



Can you think of a fast approximation method?
Which factor can you achieve?

Min enclosing disk

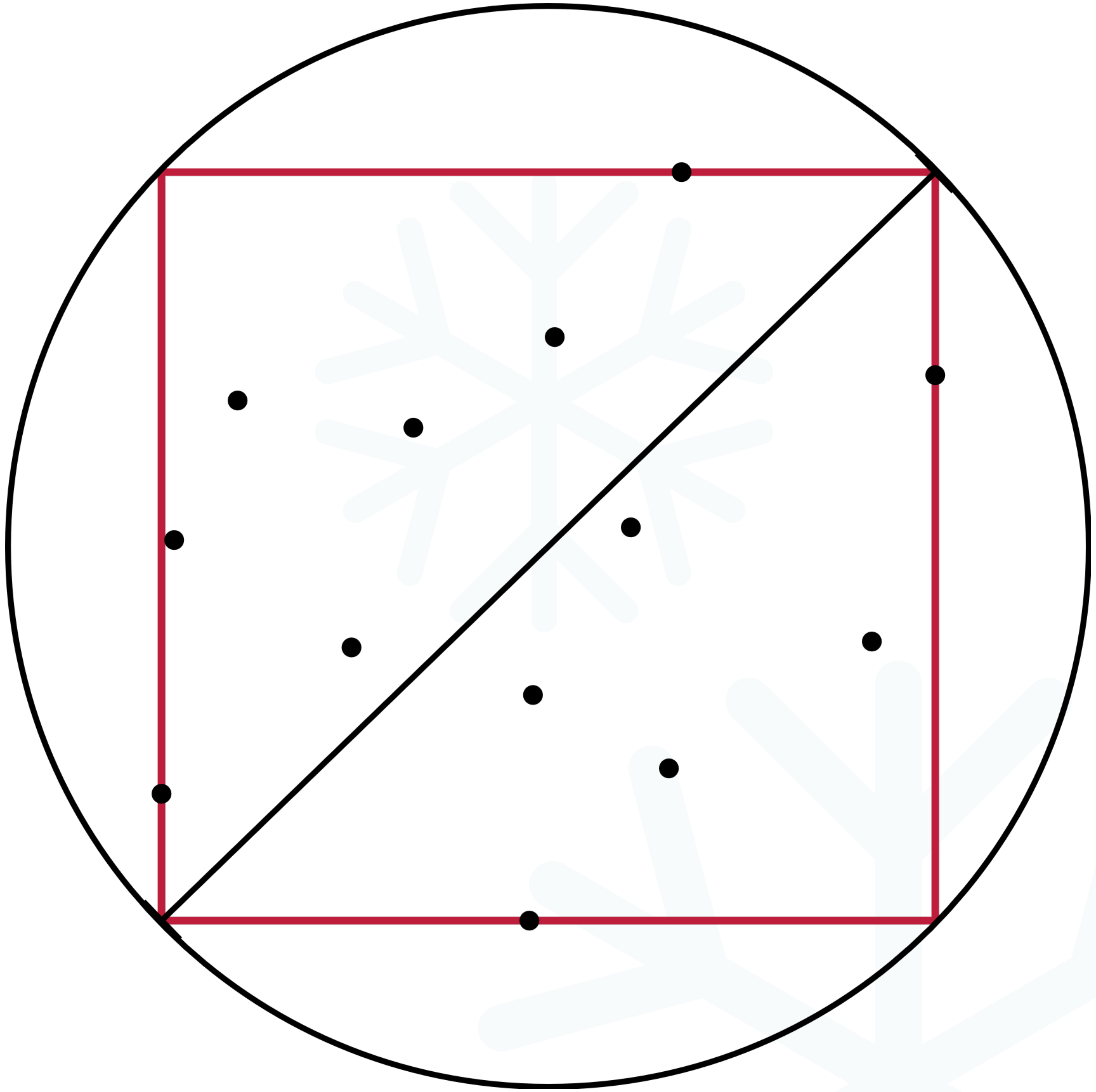
A $\sqrt{2}$-approximation

Given: Points $\mathscr{P} := p_1, \ldots, p_n$ in the plane, in general position.

Idea: Compute in $\mathcal{O}(n)$ an axis-aligned bounding box via $\min$ and $\max$ coordinates, use the smallest enclosing disk of those.

The diameter of this disk is at most $\sqrt{2}$ times larger than $\operatorname{diam}(\mathscr{P})$, which bounds the diameter of any enclosing disk from below.

Note: r is not necessarily equal to $\frac{1}{2}\operatorname{diam}(\mathscr{P})$.



Thank you

… and see you next year :)