CSCI 4560/6560 Computational Geometry

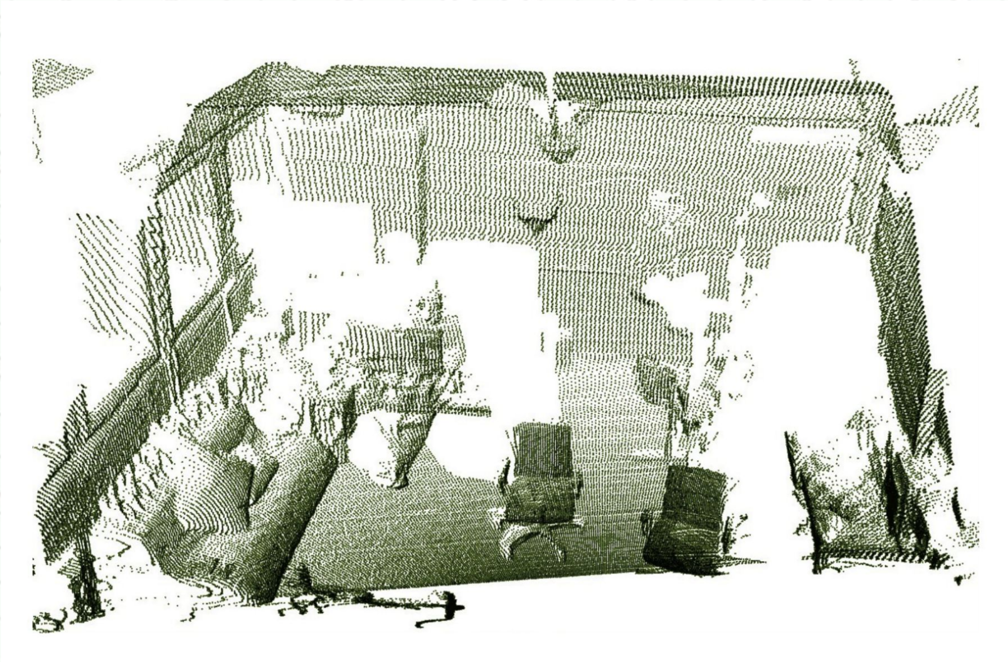
https://www.cs.rpi.edu/~cutler/classes/computationalgeometry/F23/

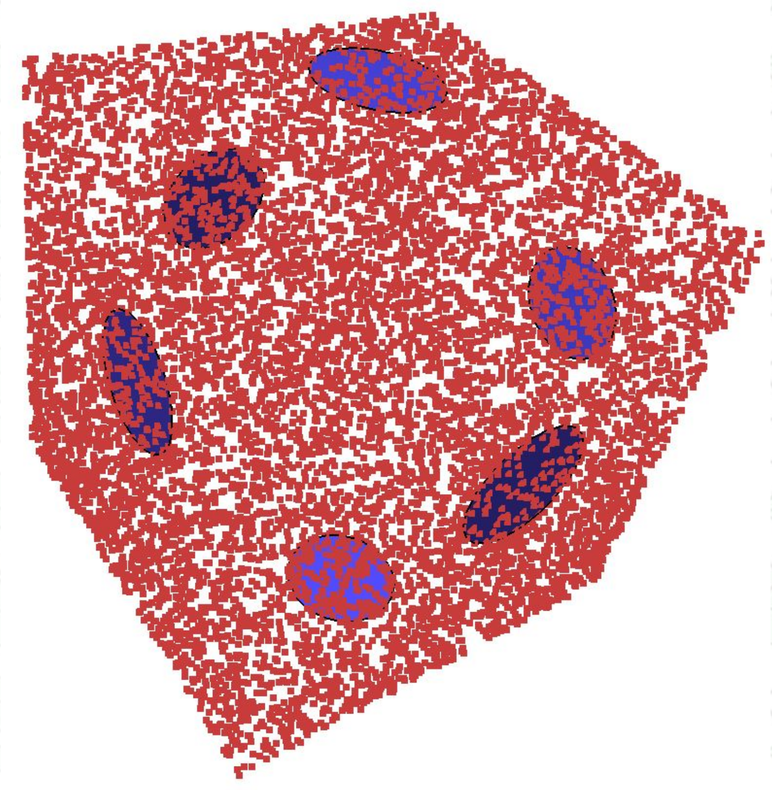
Lecture 10: Voronoi Diagrams, Part 1

Outline for Today

- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

Homework 5 - CGAL Programming Task



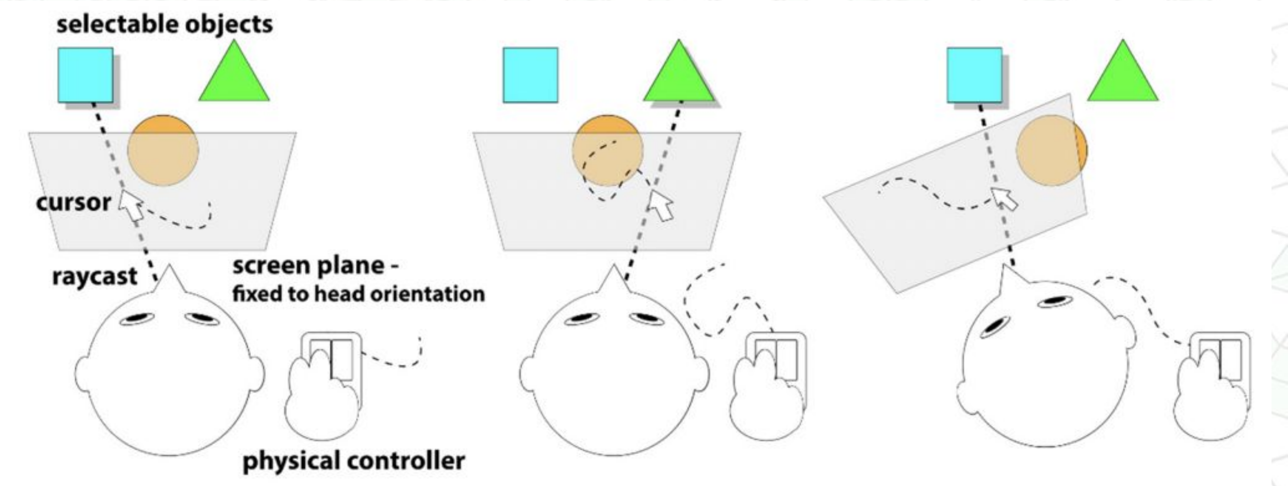


Outline for Today

- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

Graphics/VR Application: What is "Picking"?

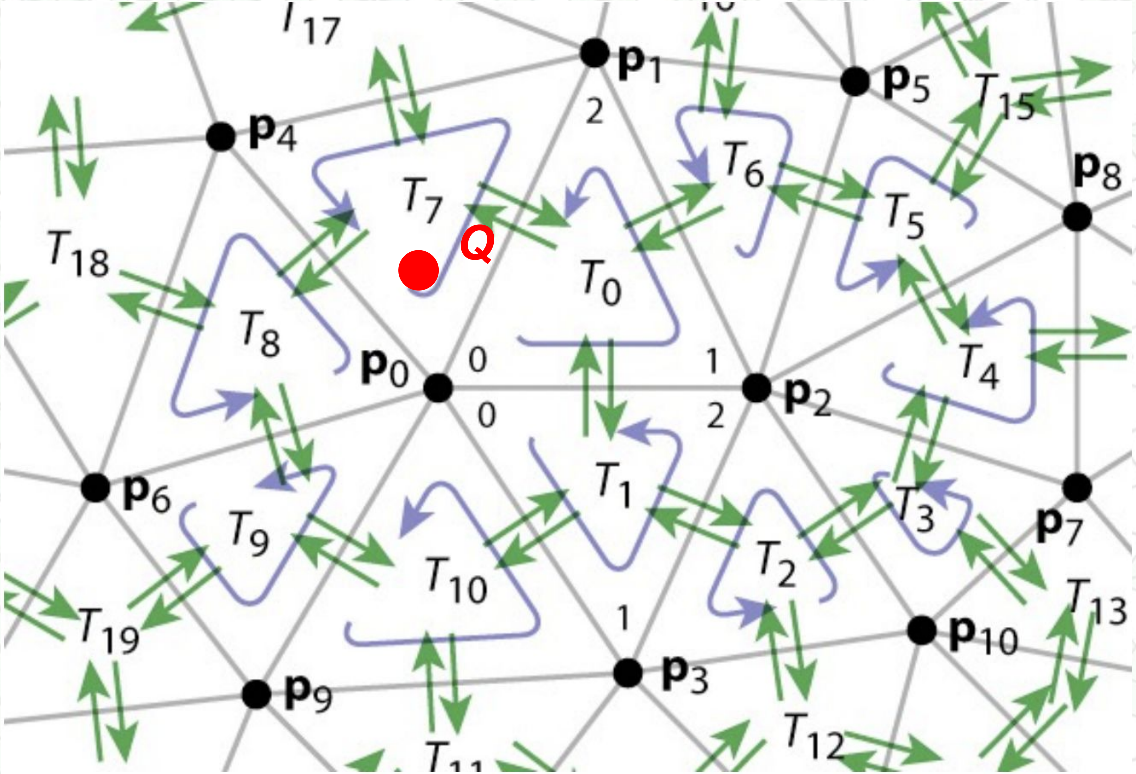
- Get the (3D) world coordinates of a (2D) mouse click
- Identify which object was selected and
  the point on the object closest to the click
- *Do we as users take this for granted??*
  - *What are the performance bottlenecks?*
  - *What are the usability concerns?*



https://www.csit.carleton.ca/~rteather/pdfs/GI_2018_EZCursorVR.pdf

Brute Force Picking Algorithm

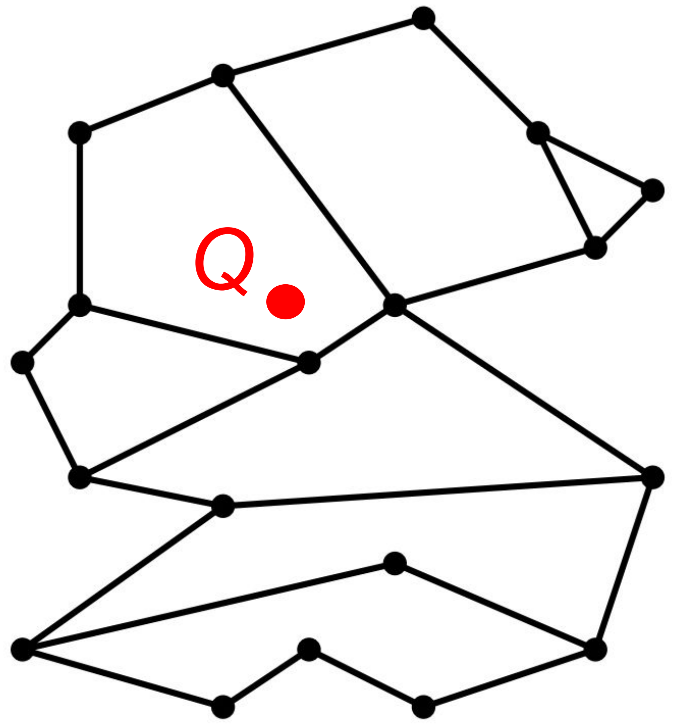
- Given a planar subdivision
  - *E.g., a collection of non-overlapping triangles (or polygons) that cover the plane*
- And a query point Q
- Which triangle/polygon is Q inside of?
  - E.g., T_7



Steve Marschner
http://www.cs.cornell.edu/courses/cs4620

Point Location in Planar Subdivision

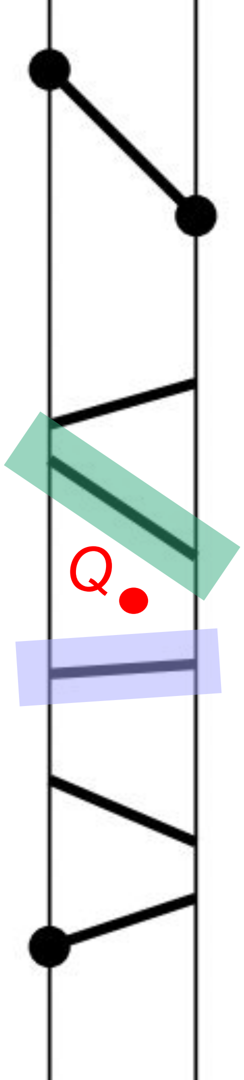
- Given *v* vertices, *n* edges, and *f* polygonal faces
- Which polygonal region contains the query point Q?



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 6

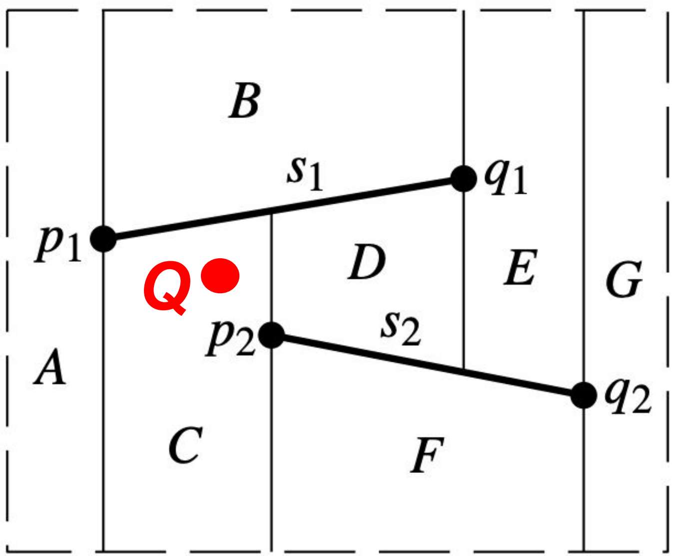
Analysis: Running Time

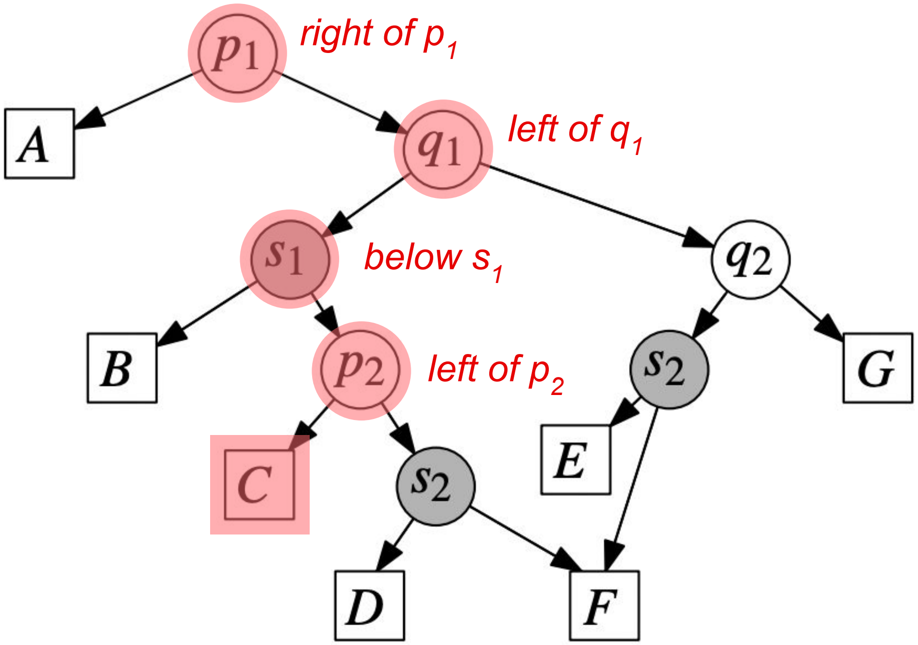
- Algorithm Preprocess
  - Sort slabs left to right → *O(n log n)*
  - Within each slab, sort trapezoids from top to bottom → *O(n log n)*
- Point Location Algorithm Overall: → *O(log n)*
  - Binary search to locate the correct slab between two points
    - Left vertical $_x$ < Q_x < right vertical $_x$ → *O(log n)*
  - Binary search to locate correct trapezoid
    - Q is below the upper segment and above the lower segment → *O(log n)*



Directed Acyclic Graph (DAG)

- Intermediate notes are vertices (vertical lines) and line segments
- The leaves are the trapezoidal regions (map back to original polygons)



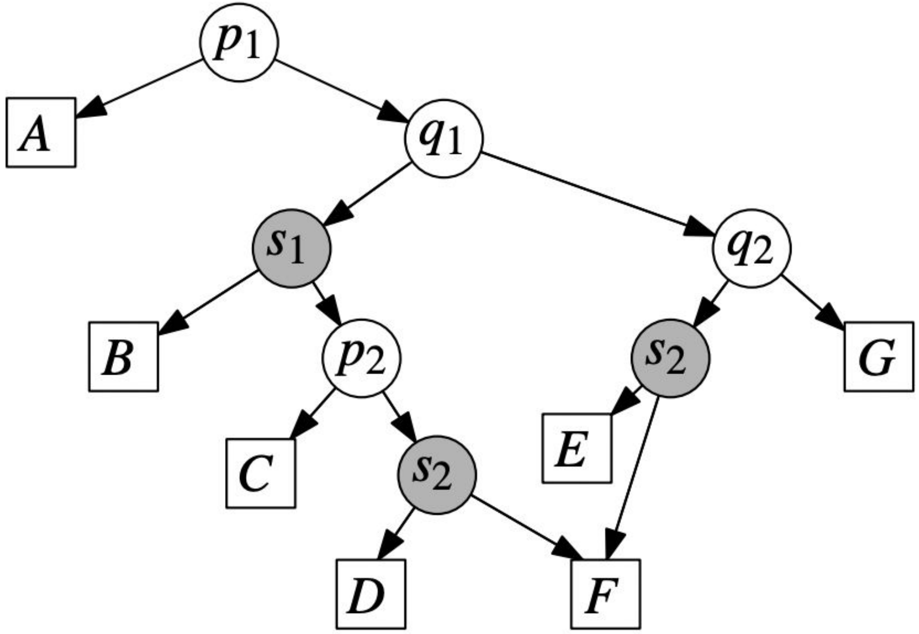


Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 6

Analysis: Directed Acyclic Graph (DAG)

Size of the DAG?

- # of leaves = # of trapezoids
  → *O(n)*
- # of intermediate nodes
  = # of vertices + # of line segments
  → *O(n)*
- Height of DAG
  → *O(log n) best case*
  → *O(n) worst case*
- Use **Randomized Incremental Construction** to achieve height
  → *O(log n) expected case!*



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 6

Outline for Today

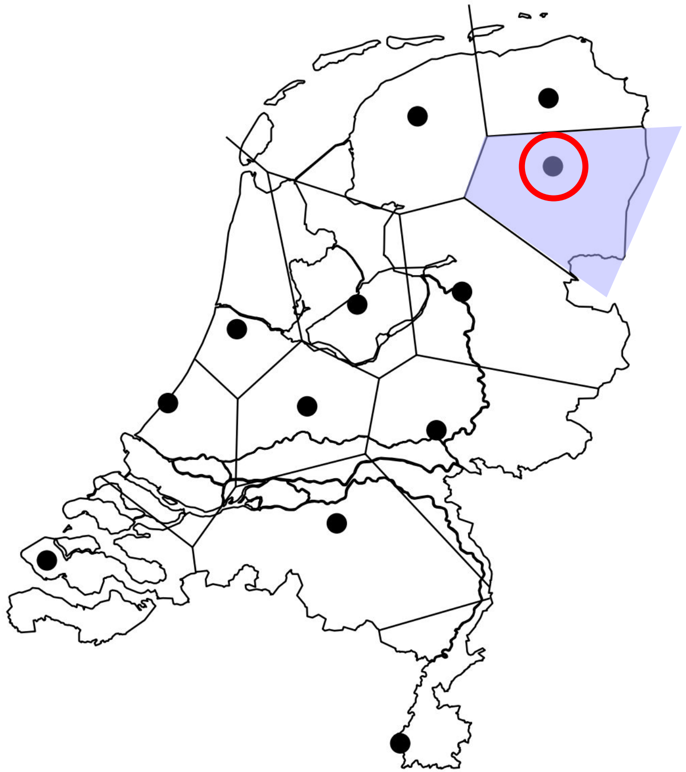
- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

Voronoi Diagram - Social Geography

actually these are the capitals of each province in the Netherlands

- There are a bunch of grocery stores spread across a large city.
- You're planning to open another grocery store at a specific location.
- How many customers can you expect at the new store location?
  Customers will choose the new store if it is closer to their home than their current store.

- a.k.a. The "Post Office Problem"



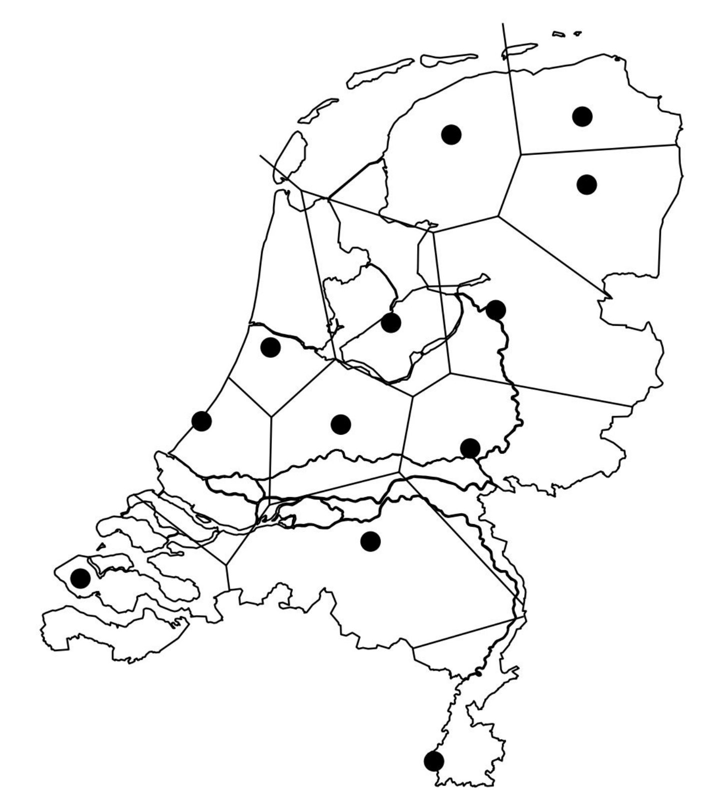
Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Voronoi Diagram - Social Geography

actually these are the capitals of each province in the Netherlands

Assumptions:

- The price of a good is the same at every site.
- The cost of acquiring the good is the price + transportation to the site.
- The cost of transportation is the Euclidean Distance * fixed price per unit distance.
- Consumers try to minimize cost of acquiring the good.



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Euclidean Distance

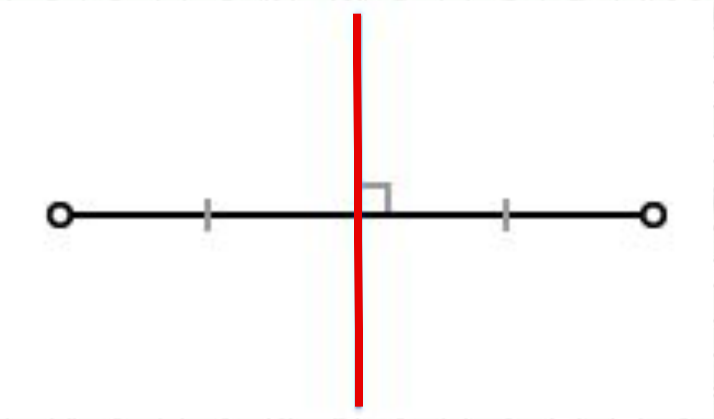
- **Euclidean Distance** - straight-line distance between two points
- *NOT* **Manhattan Distance** - only travel along axis aligned roads
- *NOT* **Geodesic Distance** - shortest distance between two cities on the globe – via a “great circle”

Outline for Today

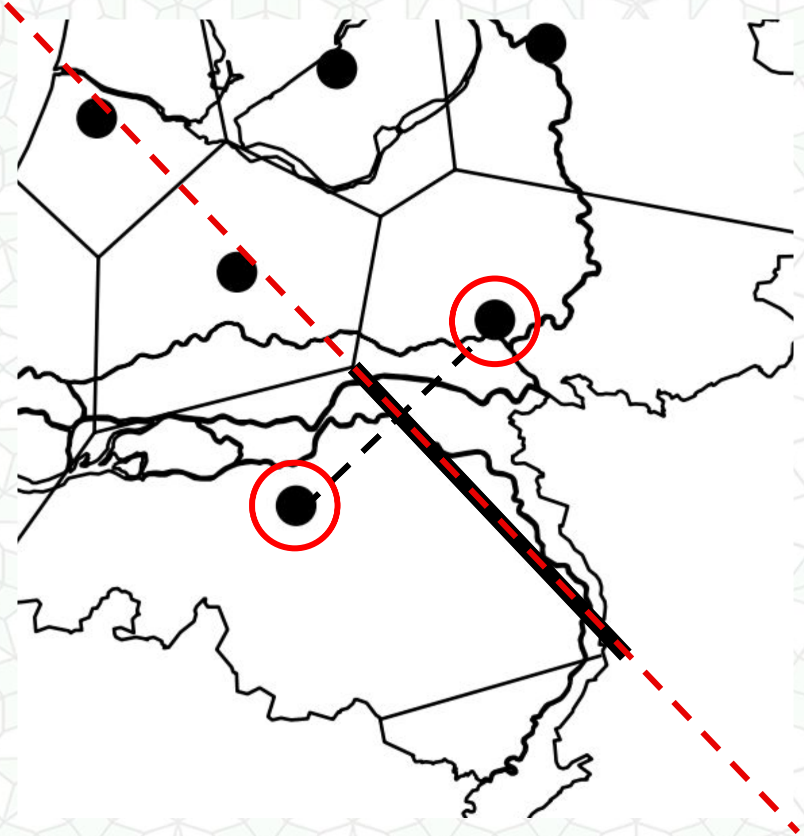
- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

Observation: Perpendicular Bisector

- Points on the edge between two Voronoi cells are equidistant from two Voronoi sites.
- Edges of Voronoi cells are perpendicular bisectors of two Voronoi sites.



https://euclidea.fandom.com/wiki/Perpendicular_Bisector

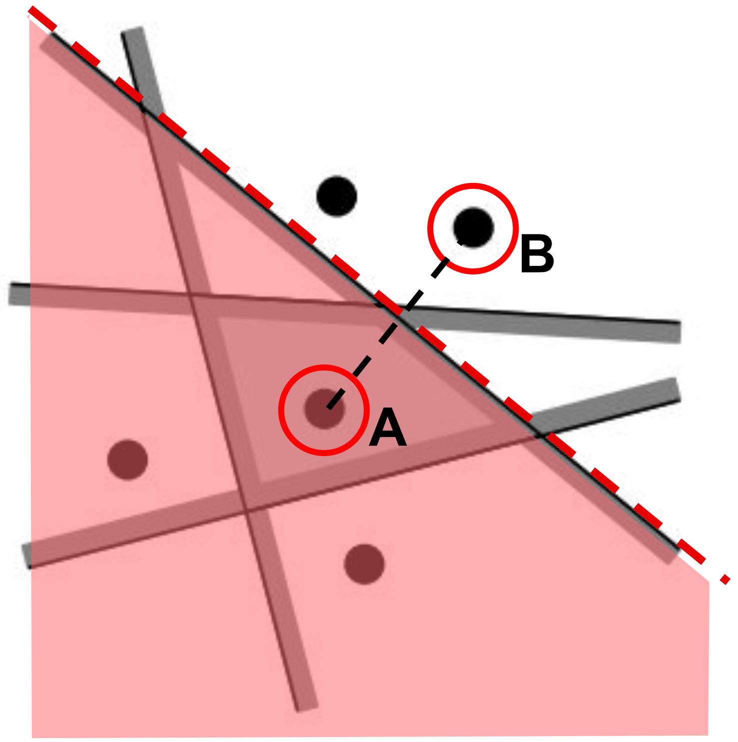


Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Observation: Intersection of Half Spaces

- All points that lie on one side of the perpendicular bisector,
- Are the half-space of points that will chose site A over site B because site A is closer than site B.

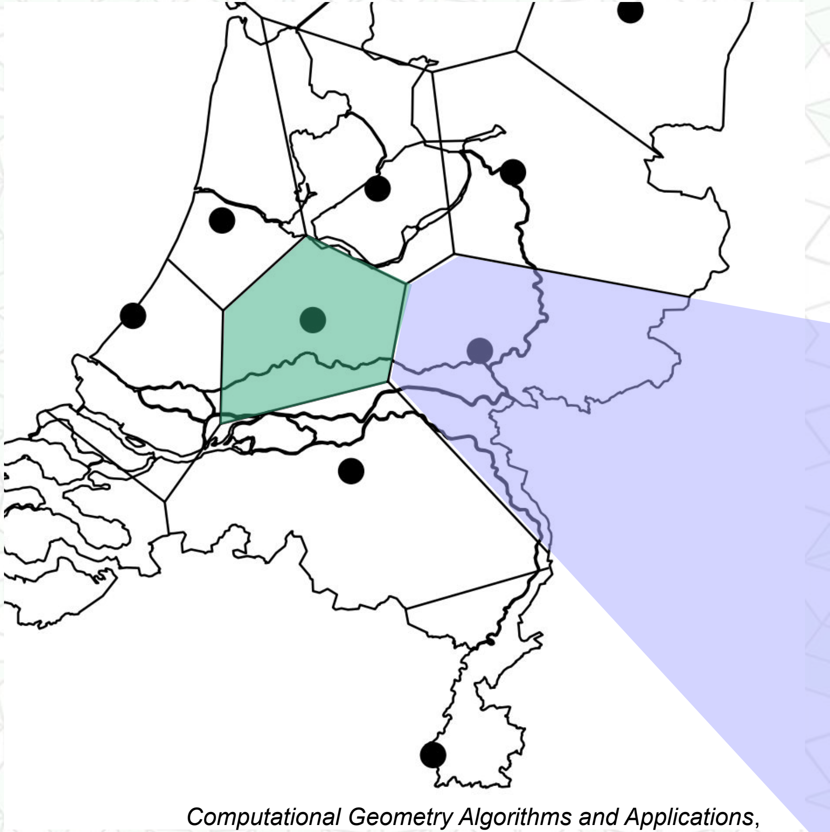
- This suggests a brute force construction algorithm…



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Observation: Voronoi Cells are Convex

- Because a Voronoi cell is the intersection of half-spaces…
- A Voronoi Cell must be convex
- *Note: Some Voronoi Cells are unbounded*



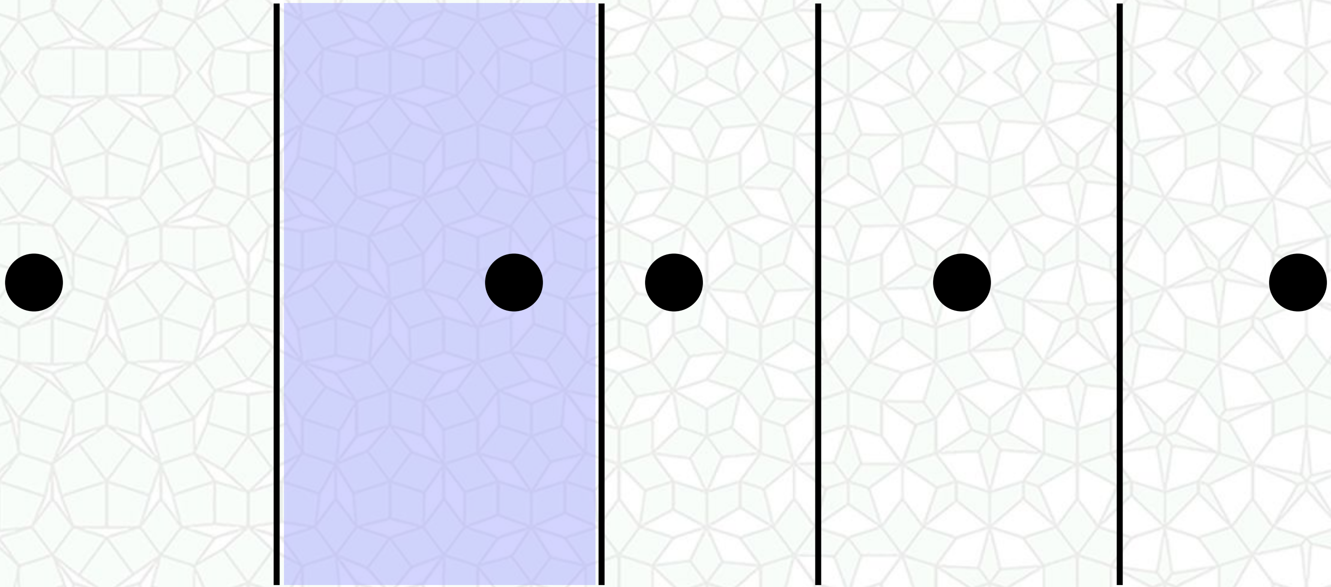
Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

What about Collinear Voronoi Sites?

- If all Voronoi sites are collinear – a degenerate configuration…

What about Collinear Voronoi Sites?

- If all Voronoi sites are collinear – a degenerate configuration…
- All Voronoi edges will be parallel lines (unbounded in both directions)
- All Voronoi cells will be unbounded areas

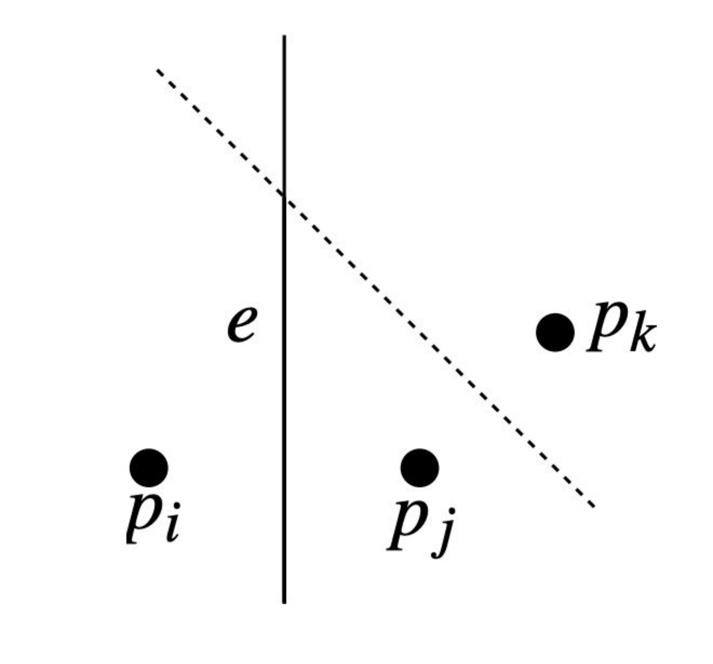


What about Collinear Voronoi Sites?

- If some Voronoi sites are *not* collinear…

- Every perpendicular bisector will intersect with at least one other perpendicular bisector

- Every Voronoi edge will be bounded on one or both directions



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Voronoi Diagram has Linear Complexity

- Given *n* Voronoi sites
- The diagram will have *n* Voronoi cells

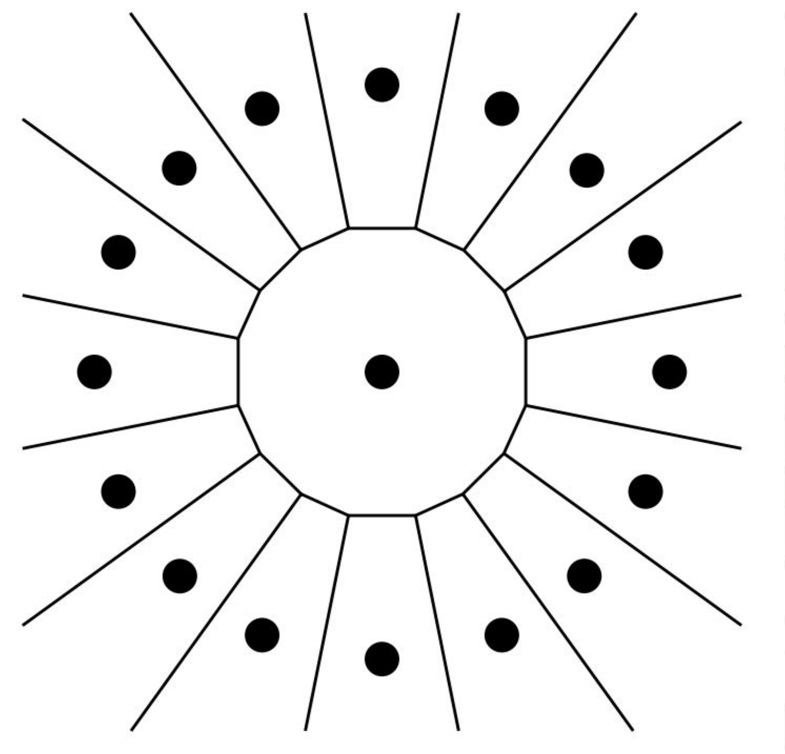
cells = faces

- A single Voronoi cell may have many edges – as many as *n-1*

- But by Euler's formula $F + V = E + 2$

sites ≠ vertices

  - # of vertices ≤ *2n-5*
  - # of edges ≤ *3n-6*



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Voronoi Diagram has Linear Complexity

- For *n* Voronoi sites = *n* cells/faces = F
- To apply Euler's formula,

  $F + V = E + 2$

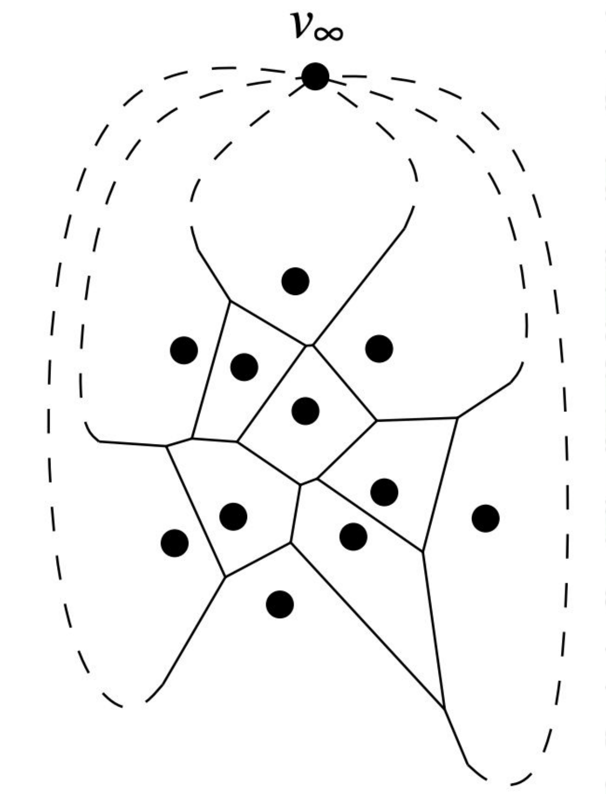
  we V_∞ to connecting all unbounded edges

- Σ all *vertex degrees* = $2 * E \leq 3 * (V+1)$
  - Every edge touches 2 vertices
  - Vertex degree = # edges that touch vertex
  - Minimum vertex degree = 3

- Substitute & algebra…

  # of vertices = V ≤ *2n-5*

  # of edges = E ≤ *3n-6*



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Outline for Today

- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

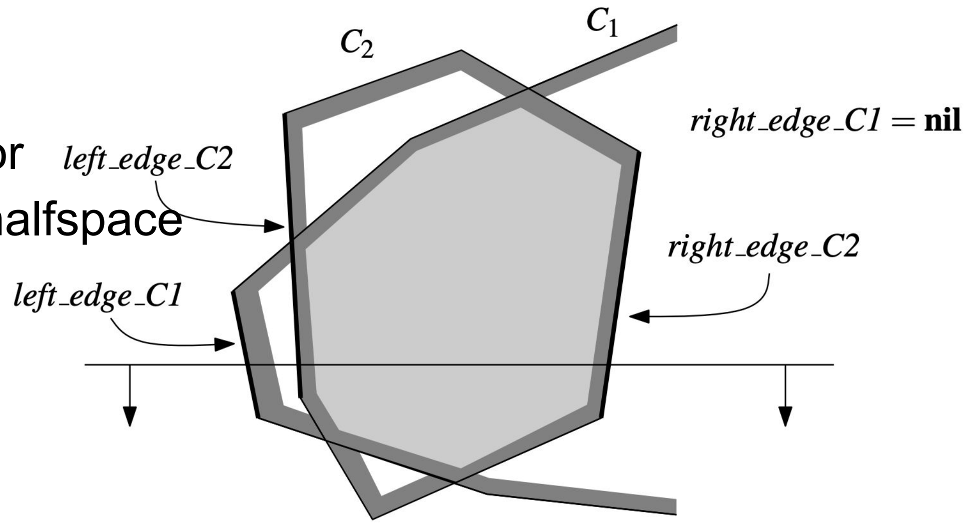
Brute Force Voronoi Algorithm - Analysis

- Given *n* Voronoi sites

Brute Force Voronoi Algorithm - Analysis

- Given *n* Voronoi sites
- For every point:
    - For every other point:
      Construct perpendicular bisector between two points, defines a halfspace
    - Intersect *n-1* half spaces
      Divide & conquer recursion, polygonal Sweep Line overlay

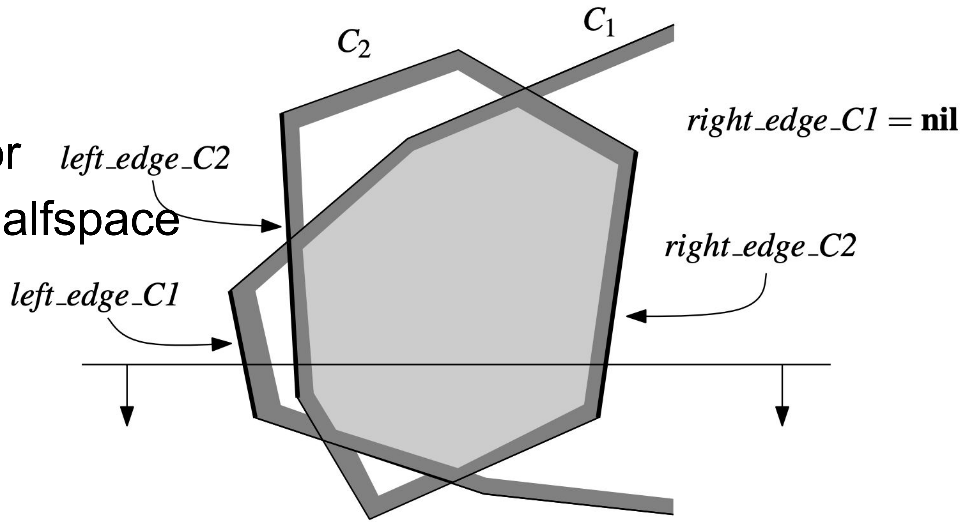
    - This is a Voronoi Cell
- Overall:



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 4

Brute Force Voronoi Algorithm - Analysis

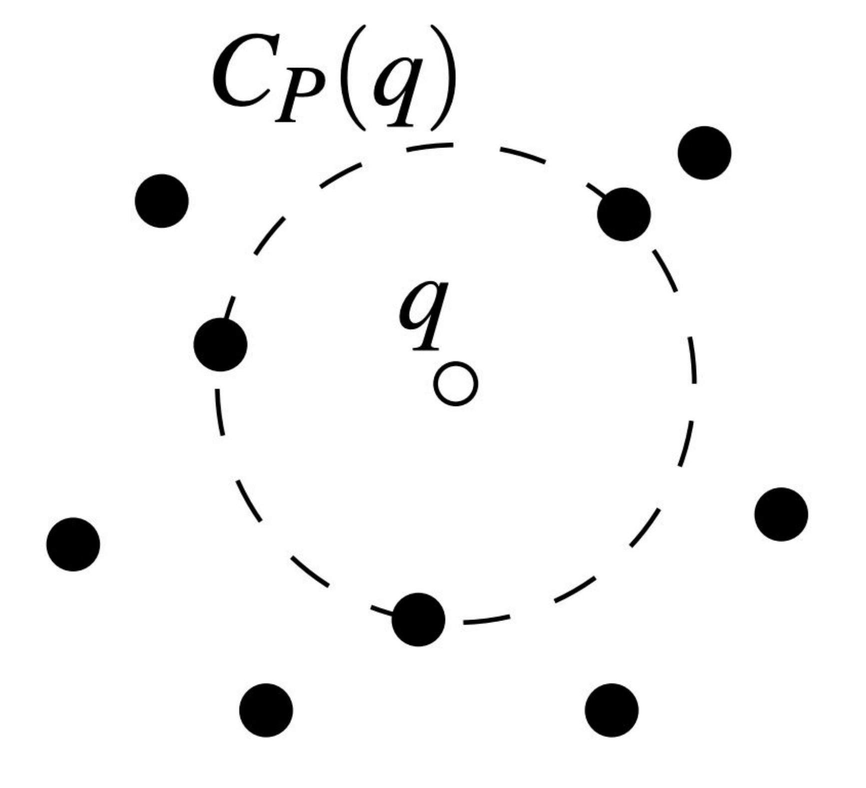
- Given *n* Voronoi sites
- For every point: *O(n)*
  - For every other point: *O(n)*
    Construct perpendicular bisector between two points, defines a halfspace
  - Intersect *n-1* half spaces
    Divide & conquer recursion, polygonal Sweep Line overlay
    → *O(n log n)*
  - This is a Voronoi Cell
- Overall: → $O(n^2 \log n)$



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 4

Voronoi Vertices

- A Voronoi vertex, *q*, is the center of a circle that passes through 3 (or more) Voronoi sites and does not contain any other sites inside of it.

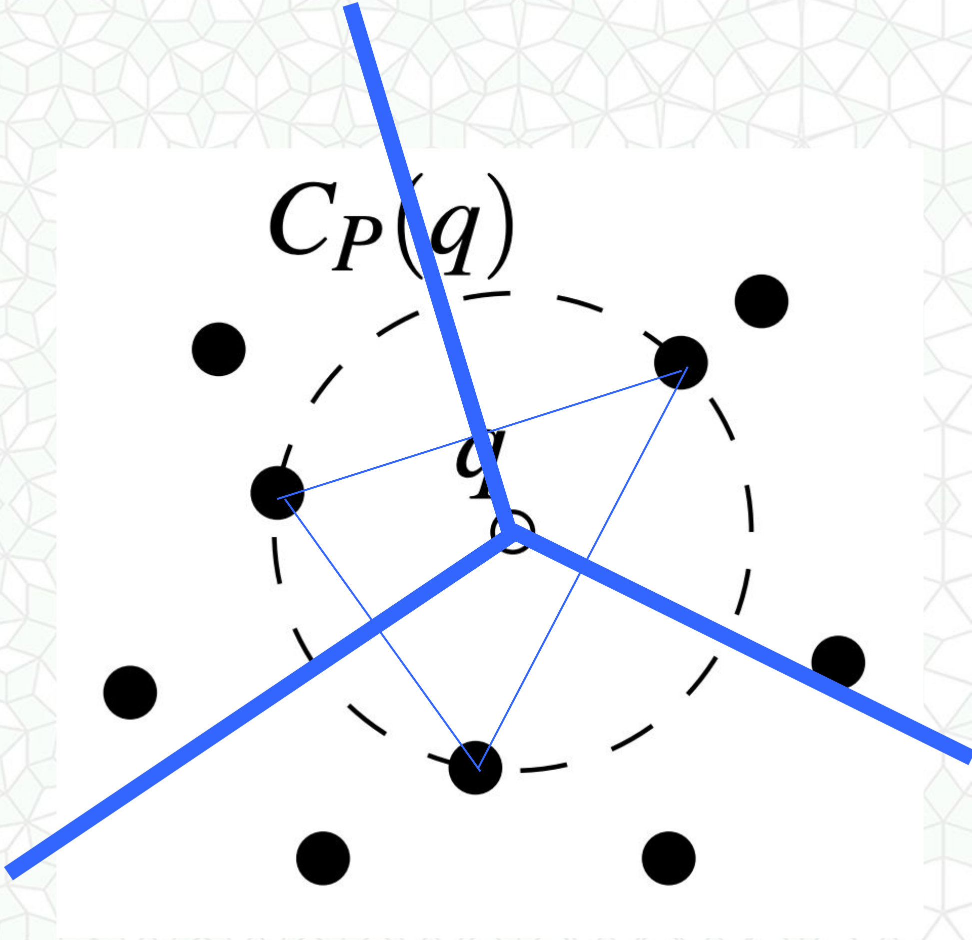


Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Voronoi Vertices

- A Voronoi vertex, *q*, is the center of a circle that passes through 3 (or more) Voronoi sites and does not contain any other sites inside of it.

- It is also the intersection of 3 (or more) perpendicular bisectors of those Voronoi sites.

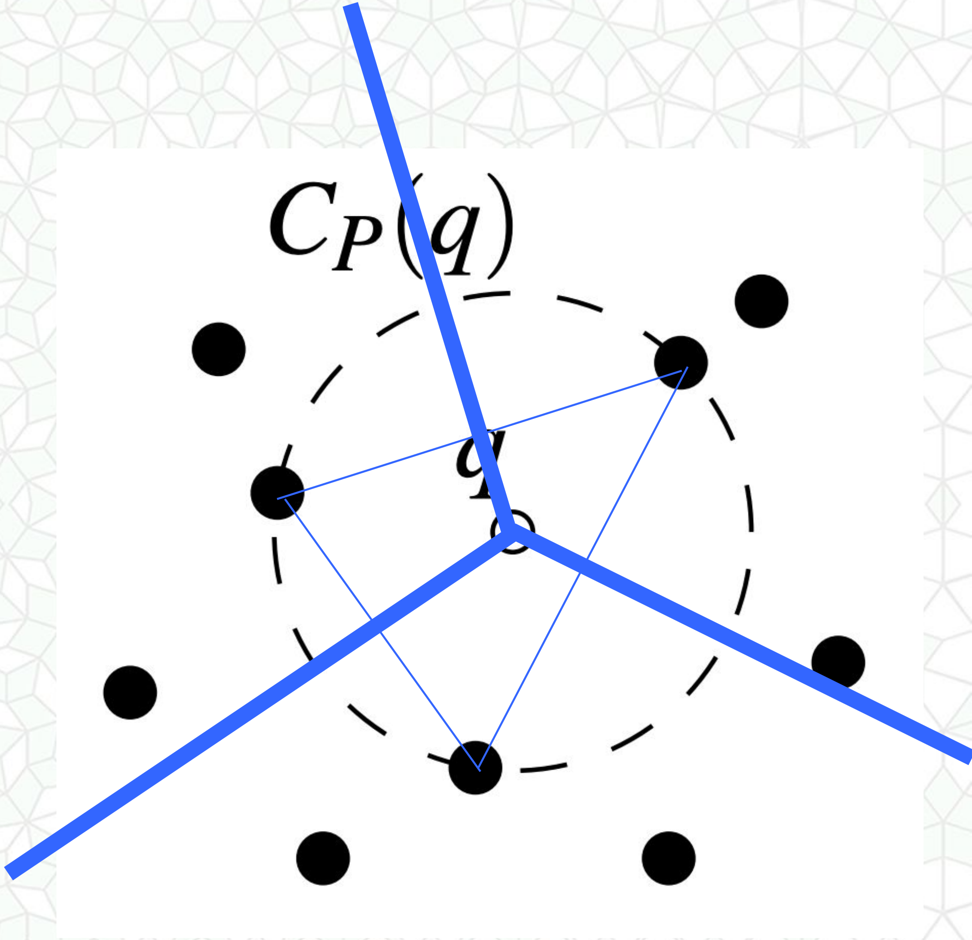


Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Voronoi Vertices

- Alternate Brute Force Construction algorithm:

- For all triples of Voronoi sites
  - If they are not collinear
  - Construct the circle

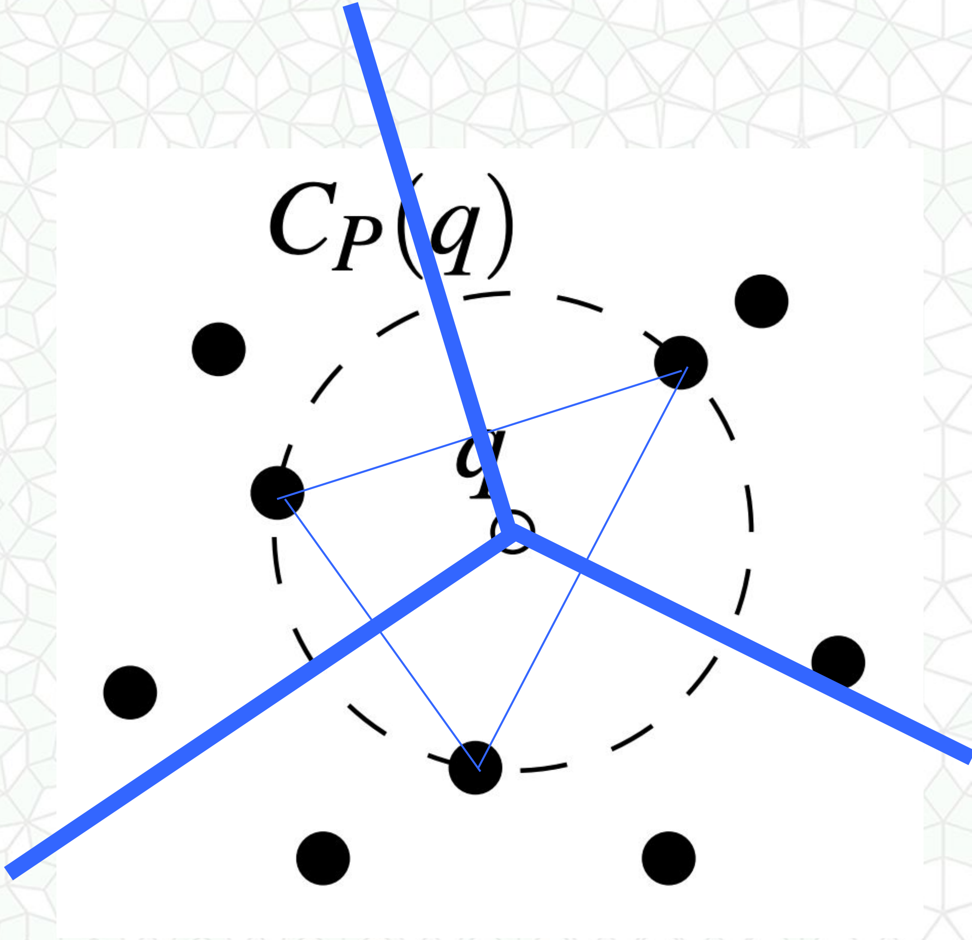


Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Voronoi Vertices

- Alternate Brute Force Construction algorithm:

- For all triples of Voronoi sites
  - If they are not collinear
  - Construct the circle
  - Check to see if any other Voronoi site lies within the circle
  - If not, keep the circle center as a Voronoi Vertex
- Overall:

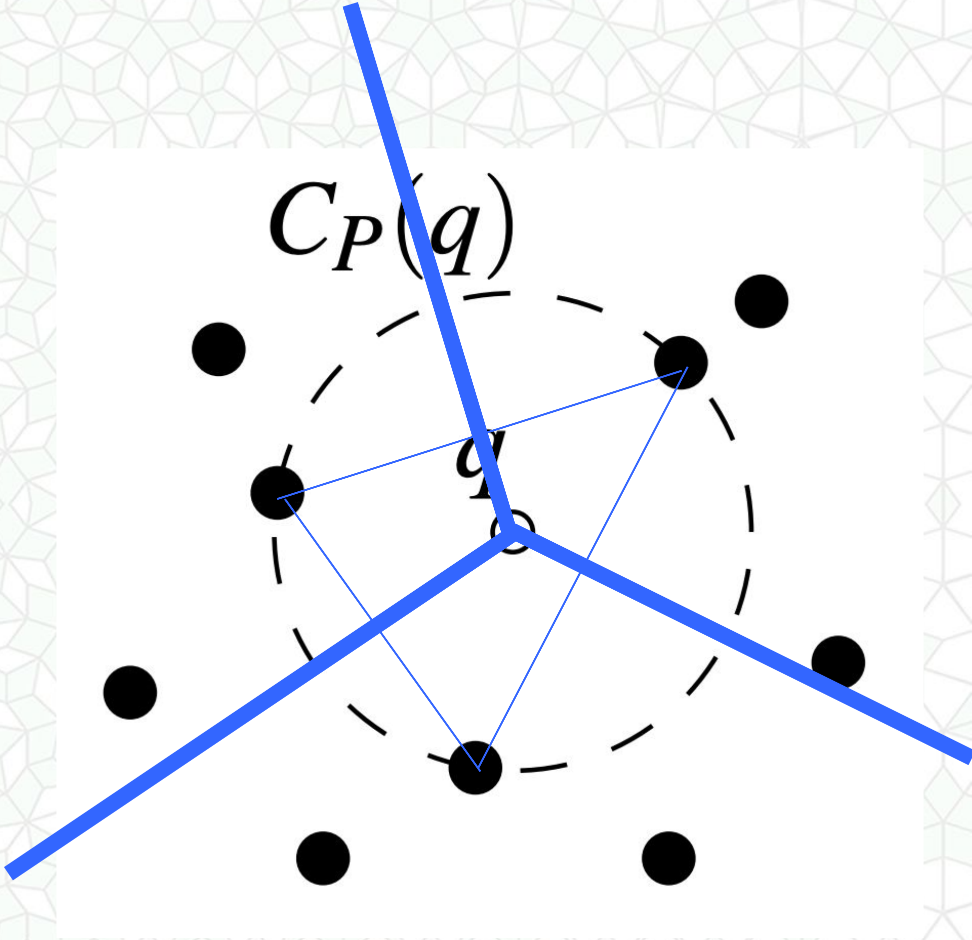


Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Voronoi Vertices

- Alternate Brute Force Construction algorithm:

- For all triples of Voronoi sites $O(n^3)$
  - If they are not collinear
  - Construct the circle
  - Check to see if any other Voronoi site lies within the circle $O(n)$
  - If not, keep the circle center as a Voronoi Vertex
- Overall: $\rightarrow O(n^4)$



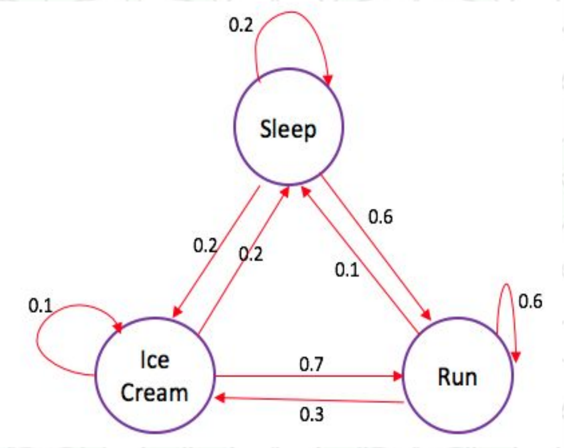
Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Outline for Today

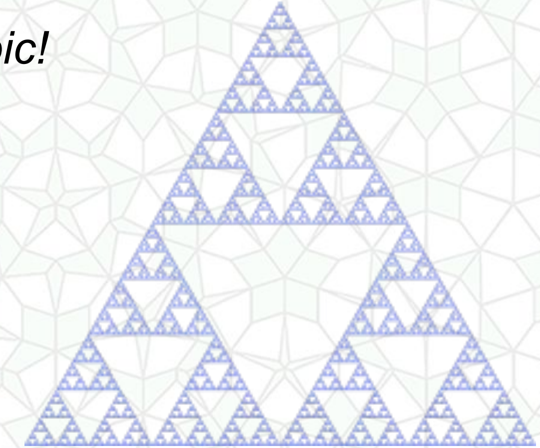
- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

A Little History - Voronoi Diagram

- Ukrainian Mathematician Georgy Voronoy (1868-1908)
  - PhD advisor, Andrey Markov (1856-1922): Markov chains & Markov processes
  - PhD student, Wacław Sierpiński (1882-1969): Sierpiński Triangle (& other fractals)
  - PhD student, Boris Delaunay (1890-1980): Delaunay Triangulation – *foreshadowing a future topic!*
- a.k.a. “Dirichlet Tessellation” (only studied 2D & 3D?)
  - German Mathematician Peter Gustav Lejeune Dirichlet (1805-1859)
- a.k.a. “Thiessen Polygon”
  - Meteorologist Alfred H. Thiessen (1872-1956), *born in Troy, NY*! (PhD from Cornell)



https://www.datacamp.com/community/tutorials/markov-chains-python-tutorial



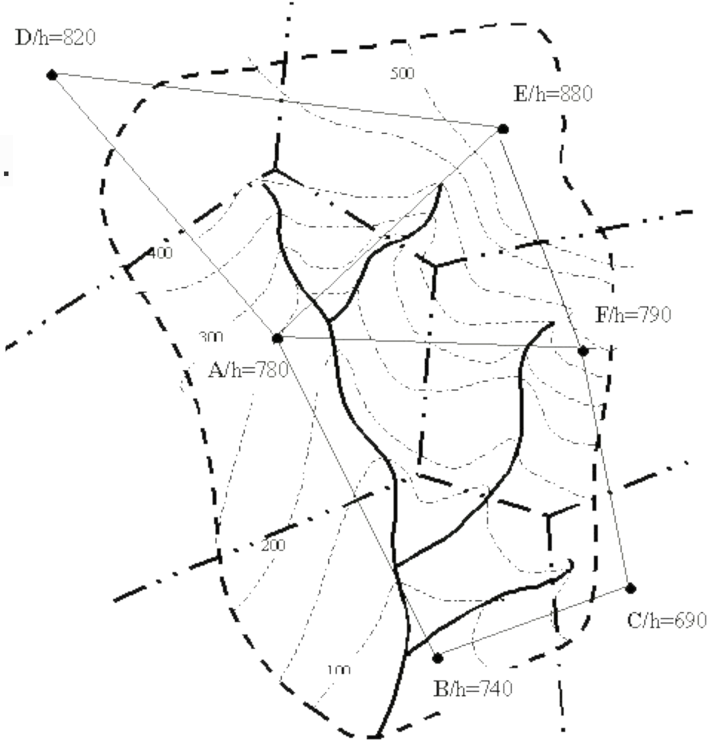
https://en.wikipedia.org/wiki/Sierpi%C5%84ski_triangle#/media/File:Sierpinski_triangle.svg

Thiessen Method - Computing Areal Precipitation

- Rainfall measurements at any station can be applied halfway to the next station in any direction.
- Rainfall is equal to observed rainfall at closest gauge.
- The weight of each rain gauge is computed by the area of the Thiessen Polygon.
- If the amount for any station is missing, the polygon must be changed.
- The Thiessen method is unable to consider orographic differences in rainfall distributions.

From: Encyclopedia of Earth Science, A. H. Schumann

Definition of **Orographic**: relating to mountains, e.g., resulting from the effects of mountains in forcing moist air to rise.



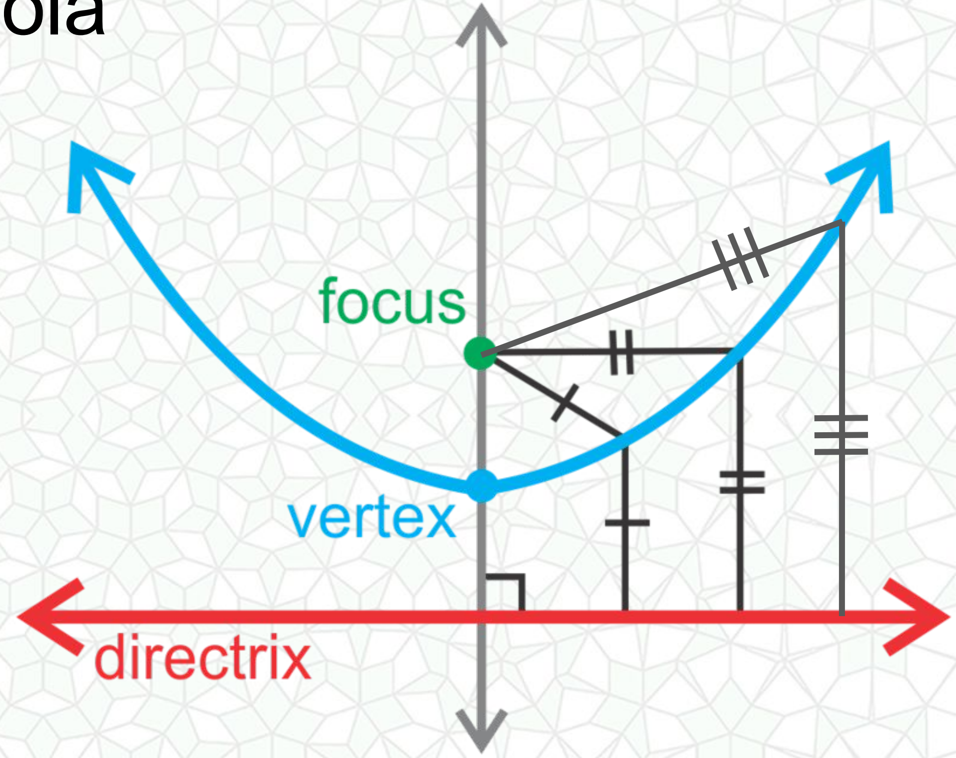
https://echo2.epfl.ch/VICAIRE/mod_1a/chapt_3/sol_exercices.htm

Outline for Today

- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

How to Graph a Parabola

All points on
a parabola
are equidistant
from
a point,
the focus
and
a line,
the directrix



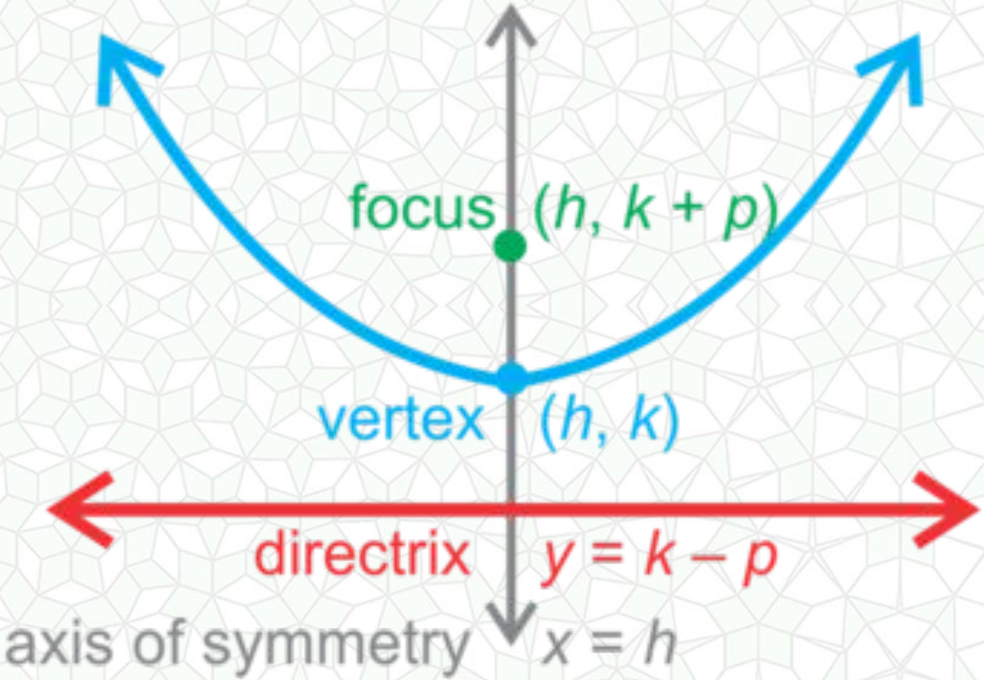
https://www.ck12.org/book/ck-12-algebra-ii-with-trigonometry-concepts/section/10.1/

How to Graph a Parabola

- Typical parabola equation:
  $y = ax^2 + bx + c$

- Rewrite as:
  $(x-h)^2 = 4p(y-k)$

- *h* gives you the vertical axis of symmetry
- *p* & *k* gives you the focus & directrix



https://www.ck12.org/book/ck-12-algebra-ii-with-trigonometry-concepts/section/10.2/

Outline for Today

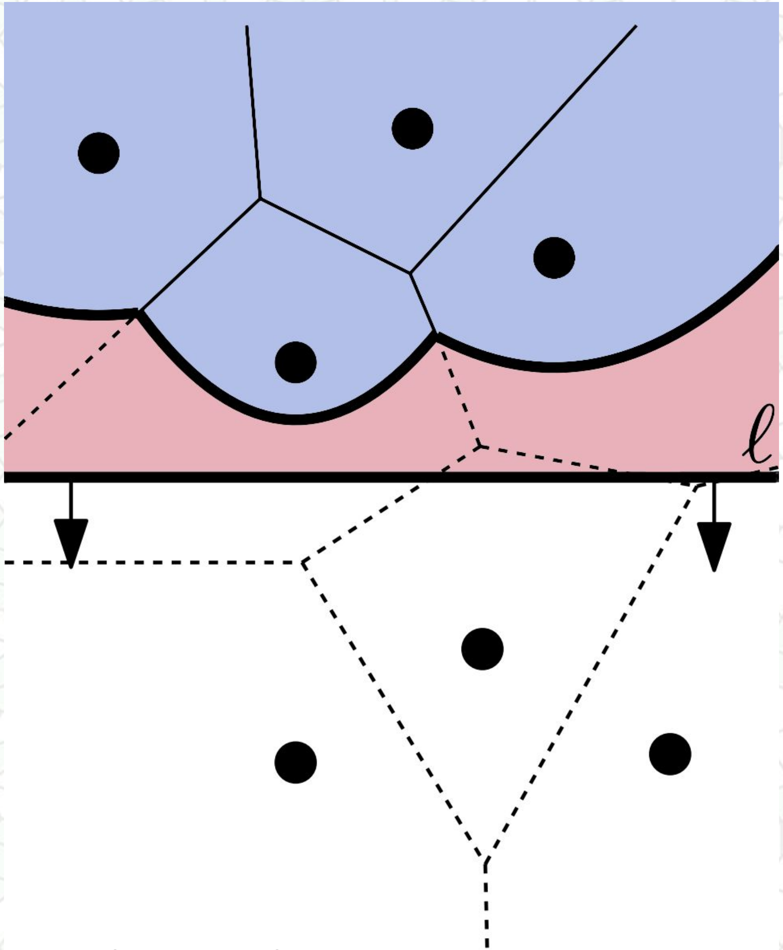
- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

“Fortune’s Algorithm”

- Can we do better than $O(n^2 \log n)$ construction runtime? *Yes!*
- “A sweepline algorithm for Voronoi diagrams”, Steven Fortune, SoCG 1986

- Sort the Voronoi sites vertically
- Move sweep line from top to bottom

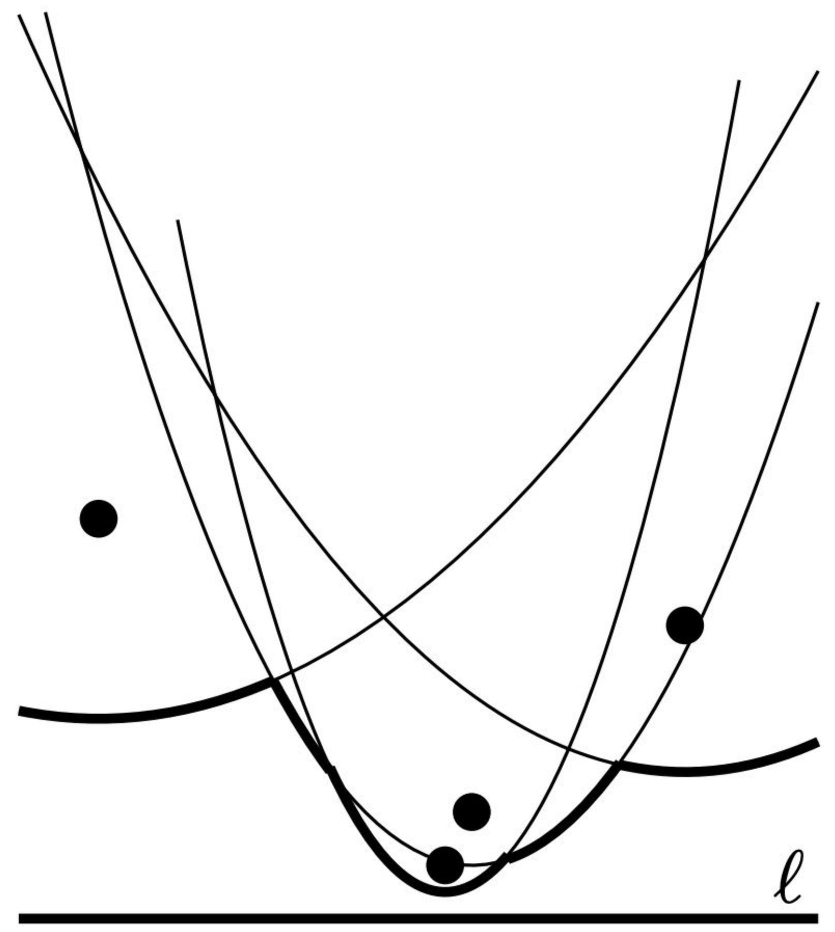
- Region above the “beach line” is known
- Region between the beach line and the sweep line may be impacted by Voronoi sites below the sweep line!



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

“Beach Line”

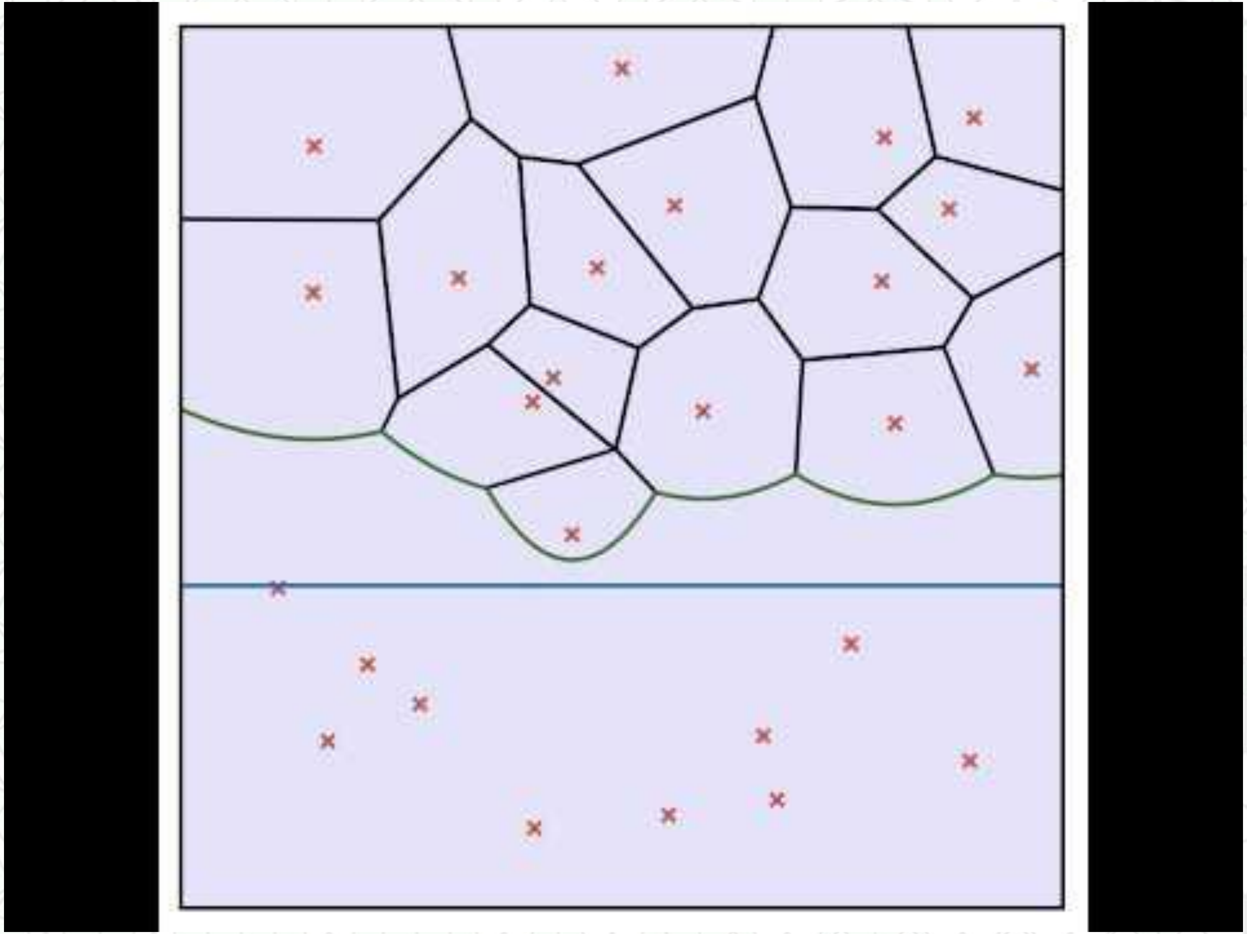
- Sequence of parabolic segments.
- Each Voronoi site above the sweep line creates a parabola. *site = focus, sweep line = directrix*
- Beach line is the lowest point of all of the parabolas for every x value.
- Beach line is x-monotone – every vertical line intersects it exactly once.
- Breakpoints – intersection of two parabolic ars – trace edges between two of Voronoi cells.



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Sweep line algorithm - Voronoi tessellation

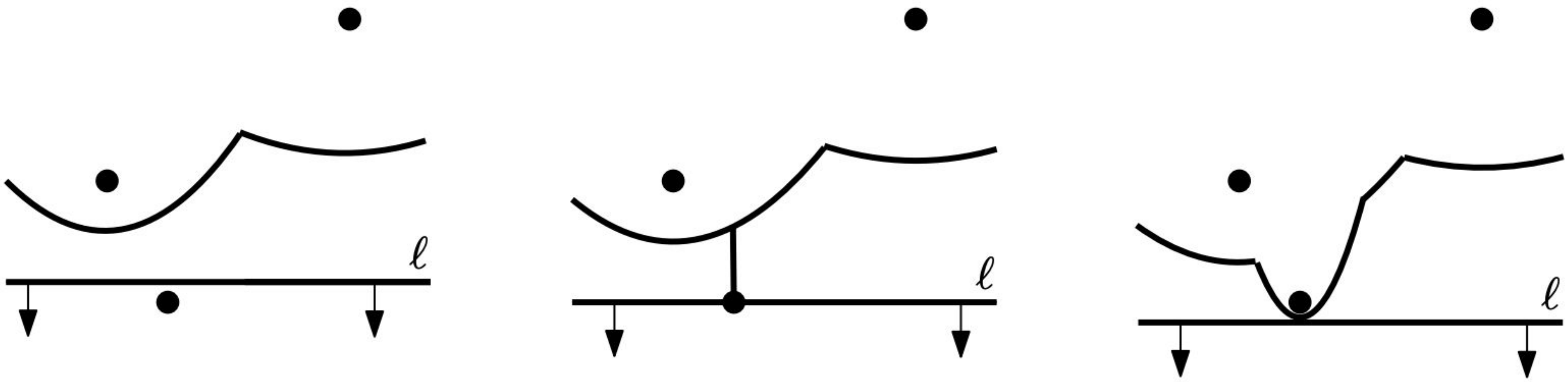
Kevin Schaal



https://www.youtube.com/watch?v=k2P9yWSMaXE

Event: New Parabola Appears

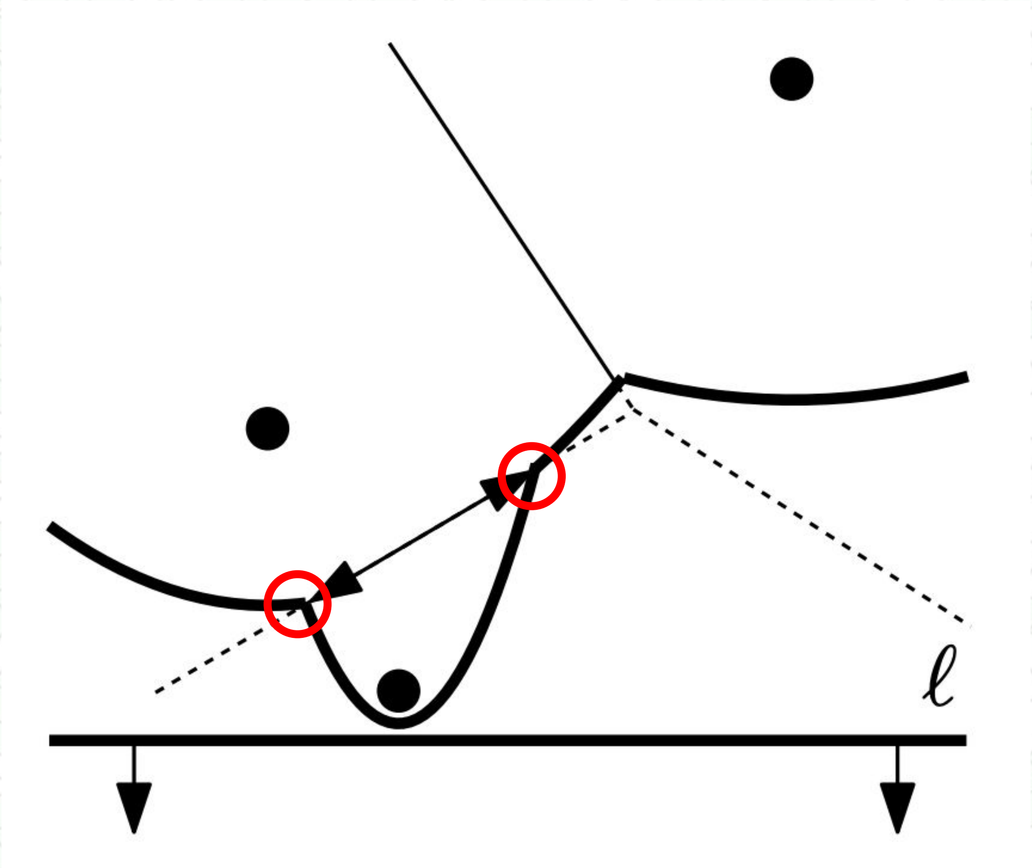
- Only happens when the sweep line reaches the next Voronoi Site
- Initially the parabola is degenerate, with width = 0



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 7

New Voronoi Edge

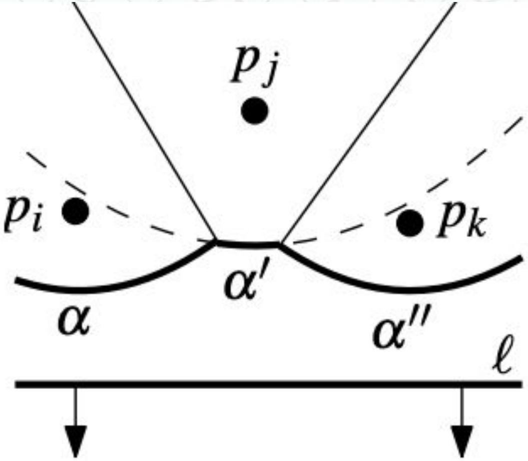
- New Parabolic Arc adds two new breakpoints
- New **Voronoi edge** is traced between those breakpoints

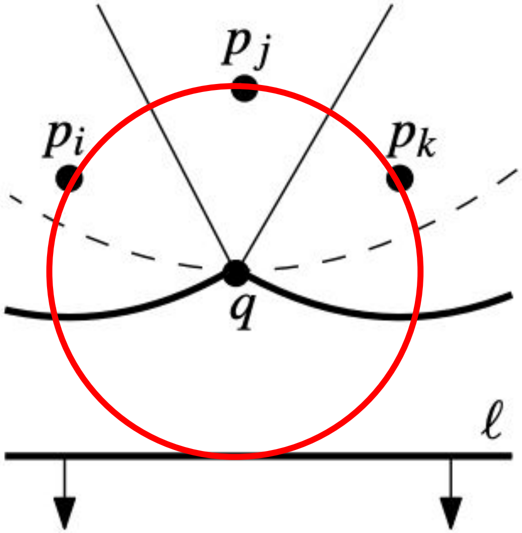


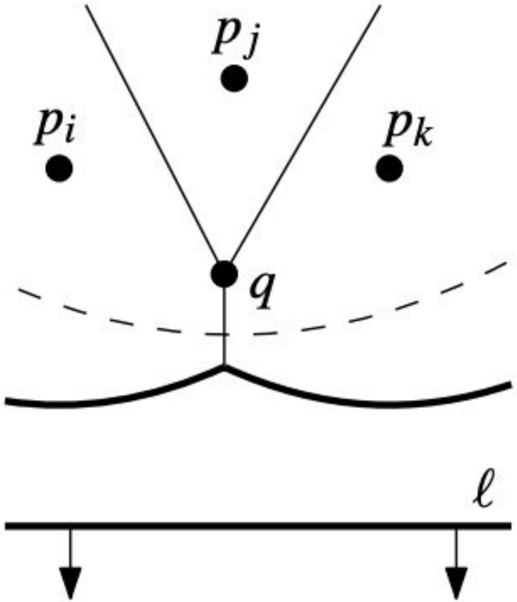
Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Event: Parabolic Arc is Absorbed

- Arc disappears when beach line reaches point q, a **Voronoi vertex,** center of circle that passes through 3 Voronoi sites / parabola foci







Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 7

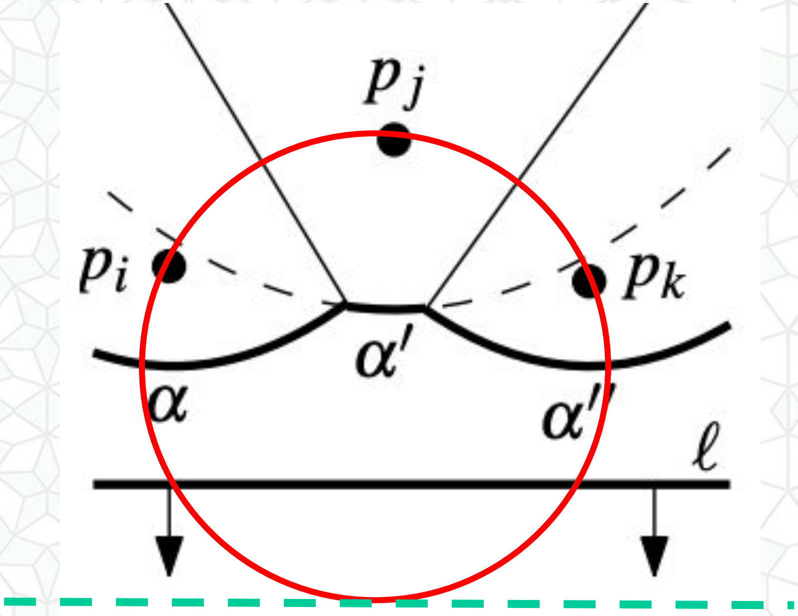
Outline for Today

- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

Sweep Line Algorithm

- For *n* Voronoi sites
- New Arc Events: Sort Voronoi sites vertically
- Keep a horizontal sorted ordering of the parabolic arcs on the current beachline.
- (Potential) Arc Absorption Events: For each triple of neighboring arcs α, α’, α’’ on the beachline, compute the **circle**, and **tangent sweep line**

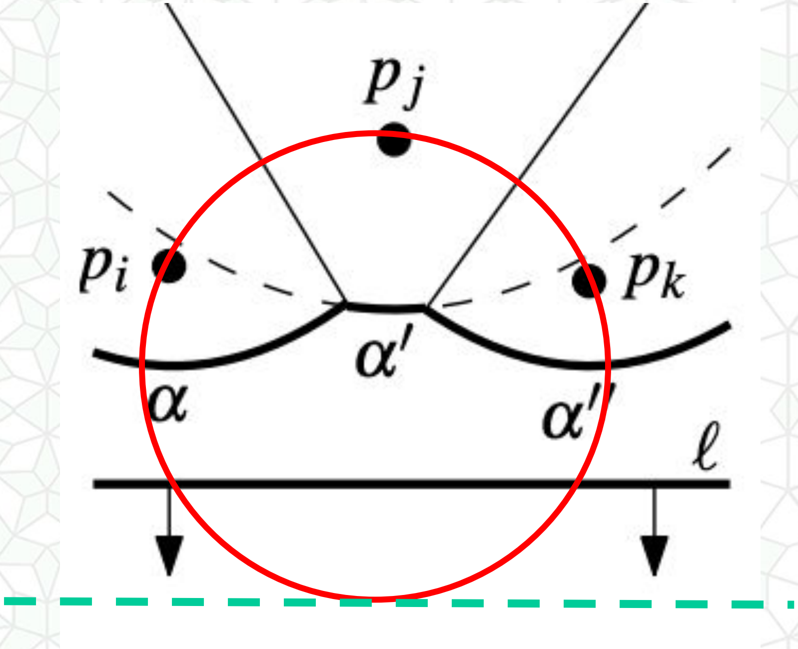
- Move sweep line to the next event…
- Overall:



Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Sweep Line Algorithm

- For n Voronoi sites
- New Arc Events: Sort Voronoi sites vertically → $O(n \log n)$
- Keep a horizontal sorted ordering of the parabolic arcs on the current beachline. $2n$ arcs maximum
- (Potential) Arc Absorption Events: For each triple of neighboring arcs α, α', α'' on the beachline, compute the **circle**, and **tangent sweep line** → $O(n)$ Voronoi vertices
- Move sweep line to the next event…
- Overall: → $O(n \log n)$

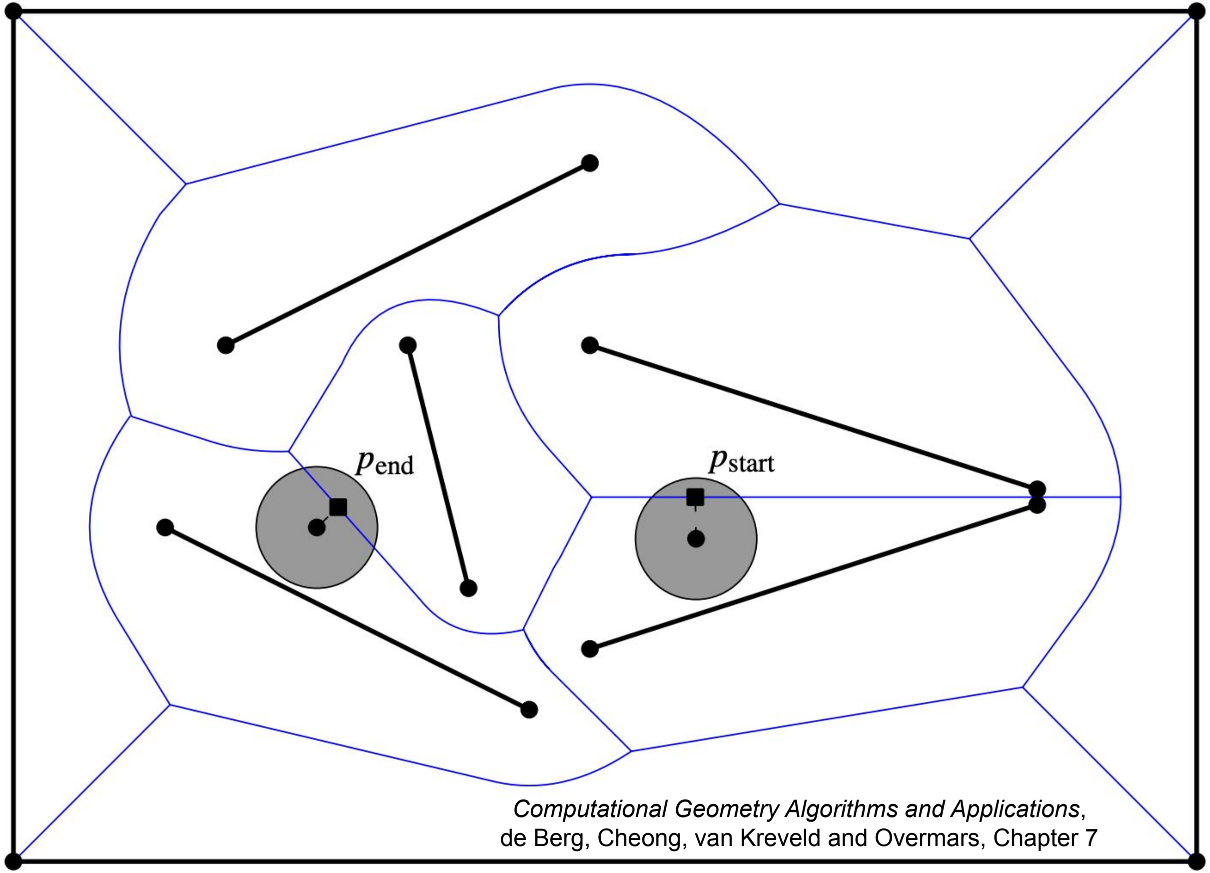


Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7

Outline for Today

- Homework 5 Questions?
- Last Time: Point Location & Trapezoidal Maps
- Motivating Application: Social Geography
- Observations about Voronoi Diagrams
- Brute Force Constructions & Analysis & Complexity
- A History of the Names Voronoi/Dirichlet/Thiessen
- How to Graph A Parabola
- Sweep Line Algorithm to Construct a Voronoi Diagram
- Analysis of Sweep Line Algorithm
- Next Time: More Voronoi Diagrams!

Next Time: Voronoi Diagram of Line Segments!



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 7