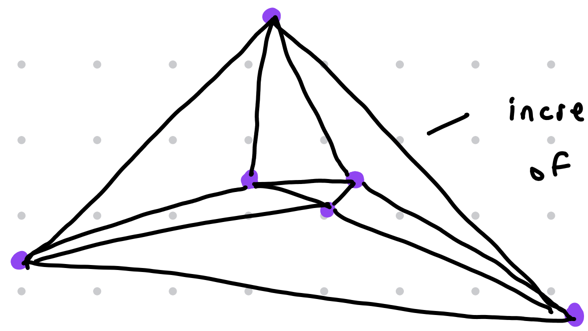
16 January 2025

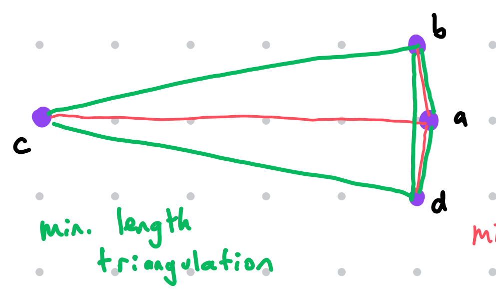
Bowyer-Watson Algorithm



incremental construction of the Delaunay triangulation

Minimum-Length Triangulation

- Finding it is NP-hard   not $O(n^c)$
- Min-length triangulation does <u>not</u> always contain the min. spanning tree



lengths:
$|bd| < |ac| < |cb| = |cd|$

Let S be a set of points, called "sites" in the plane.

For each site p, the **VORONOI REGION** $Vor(p)$ is the set of points that are at least as close to p as to any other site.

$$Vor(p) = \{ x \in \mathbb{R}^2 \mid \underbrace{\|x-p\|}_{\text{distance}} \leq \|x-q\| \text{ for all sites } q \in S \}$$

The **VORONOI DIAGRAM** $Vor(S)$ is the collection of boundaries of all Voronoi regions.

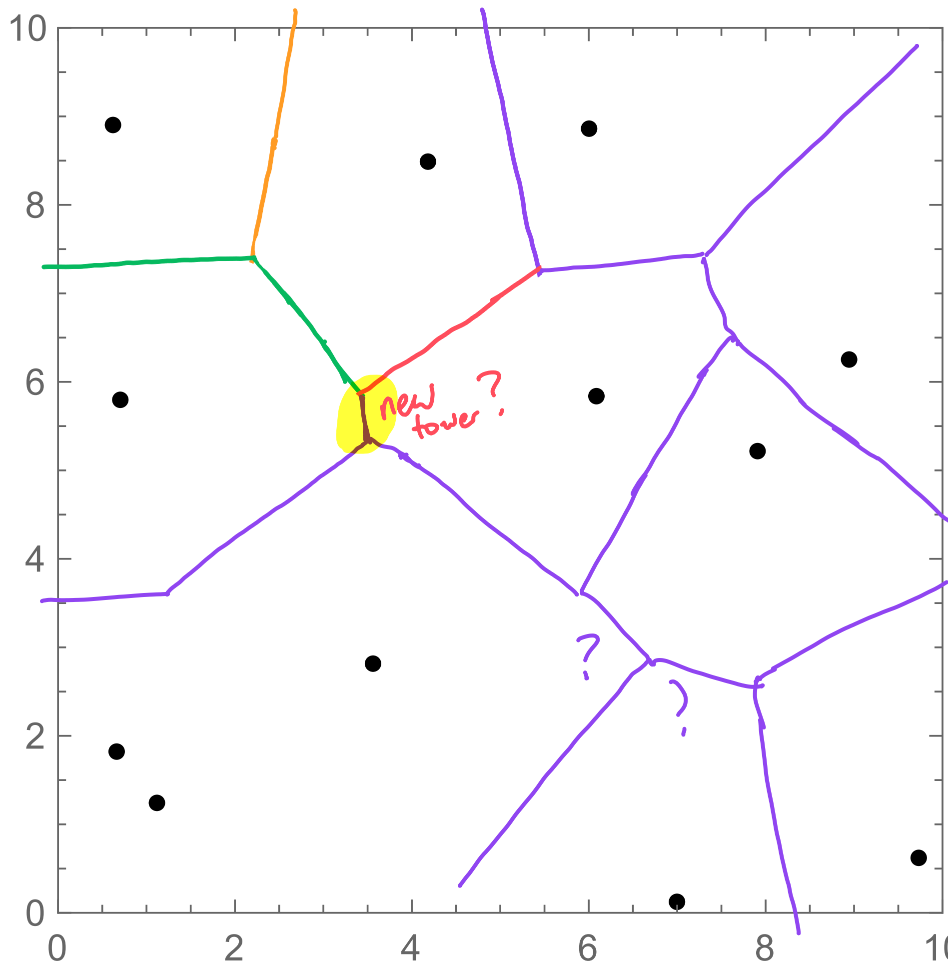
That is, the Voronoi diagram consists of all **VORONOI EDGES** and **VORONOI VERTICES** that partition the plane into Voronoi regions.

NOTE: Voronoi regions are always convex!

Voroni Diagrams

MATH 261 Computational Geometry

1. Suppose that following map shows the locations of cell towers, and that cell phones always connect to the closest cell tower. Partition the diagram into regions, one per tower, indicating the locations from which phones will connect to that tower.

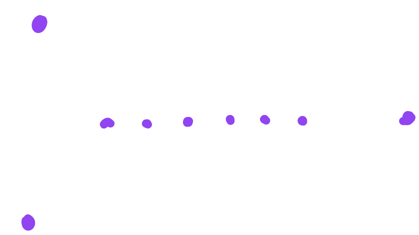


2. Now suppose you are a consultant helping the cell phone company decide where to place a new cell tower. Where would you position the new tower? Defend your choice.

3. If the cell phone company builds a new tower on the location you recommended, how does your diagram change?

4. For each $n \geq 3$, find a point set S with n sites such that $\mathrm{Vor}(S)$ has the maximum possible number of vertices. What is this maximum number of vertices?

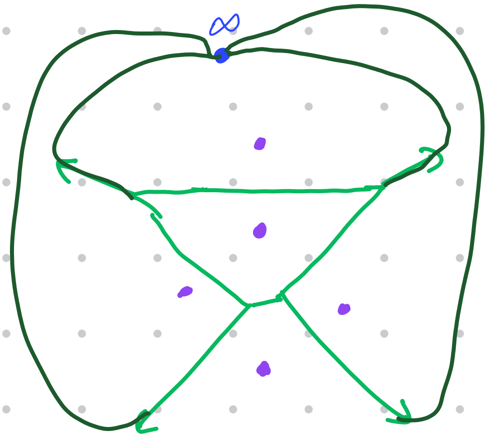
n sites	max Voronoi vertices
3	1
4	3
5	5
6	7
7	9
$\vdots$	$\vdots$
n	$2n-5$



5. For each $n \geq 3$, find a point set S with n sites such that $\mathrm{Vor}(S)$ has the maximum possible number of edges. What is this maximum number of edges?

assume not all sites collinear

THEOREM: Let S be a point set with $n \geq 3$ sites. Then $\text{Vor}(S)$ has at most $2n-5$ Voronoi vertices and at most $3n-6$ Voronoi edges.



proof: Transform Voronoi diagram into a graph by adding a new vertex "infinity"

This graph has:

$F = n$ faces

If e is the number of Voronoi edges, then graph also has $E = e$ edges

If v is the number of Voronoi vertices, then graph has $V = v+1$.

By Euler's formula: $V - E + F = 2 \Rightarrow v+1 = 2-n+e$

$(v+1) - e + n = 2 \Rightarrow (v+1)+n-2 = e$

Summing up degrees of all vertices of the graph counts each edge twice. Since each vertex has degree at least 3,

total degree: $2e \geq 3(v+1)$

Substitute: $(v+1)+n-2 = e$:

$$2((v+1)+n-2) \geq 3(v+1)$$
$$2v+2+2n-4 \geq 3v+3$$
$$\boxed{2n-5 \geq v}$$

Substitute: $v+1 = 2-n+e$:

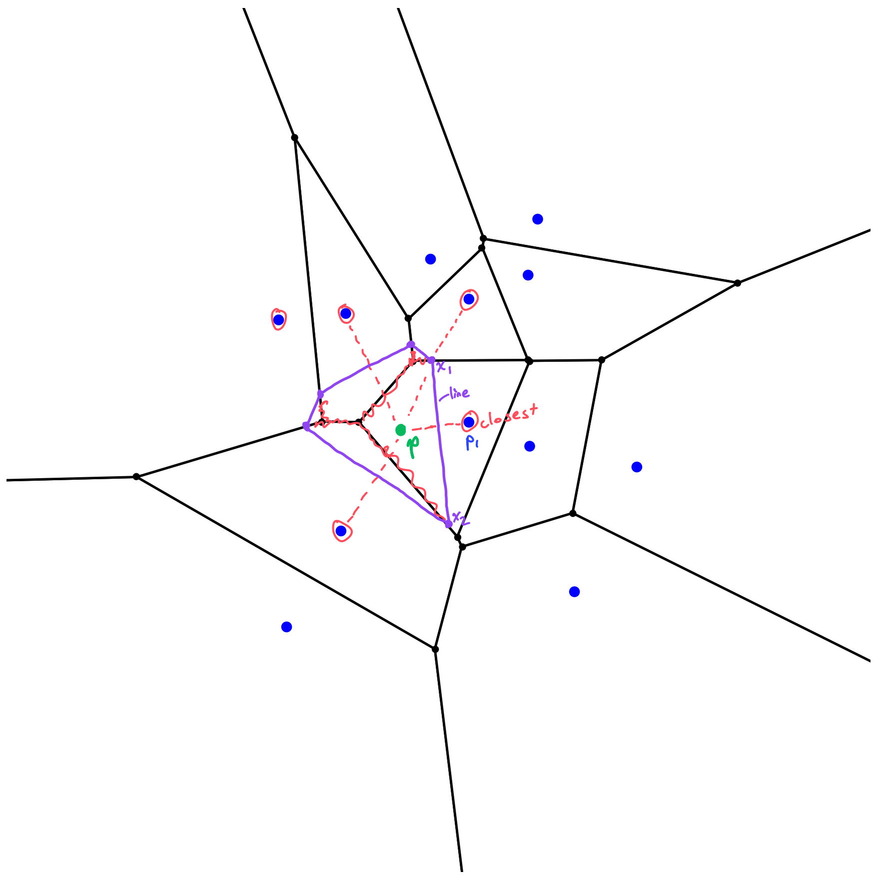
$$2e \geq 3(2-n+e)$$
$$2e \geq 6-3n+3e$$
$$\boxed{3n-6 \geq e}$$

Voronoi diagram on n sites has at most $3n-6$ edges

Voronoi Algorithms

MATH 261 Computational Geometry

1. Suppose we have an existing Voronoi diagram, constructed from a set of sites $\{p_1, p_2, \ldots, p_k\}$. We want to add new site p, and to update the diagram to include the region $\mathrm{Vor}(p)$.



(a) How does the diagram change when site p is added?

(b) Describe an algorithm for updating the Voronoi diagram to include the new site.

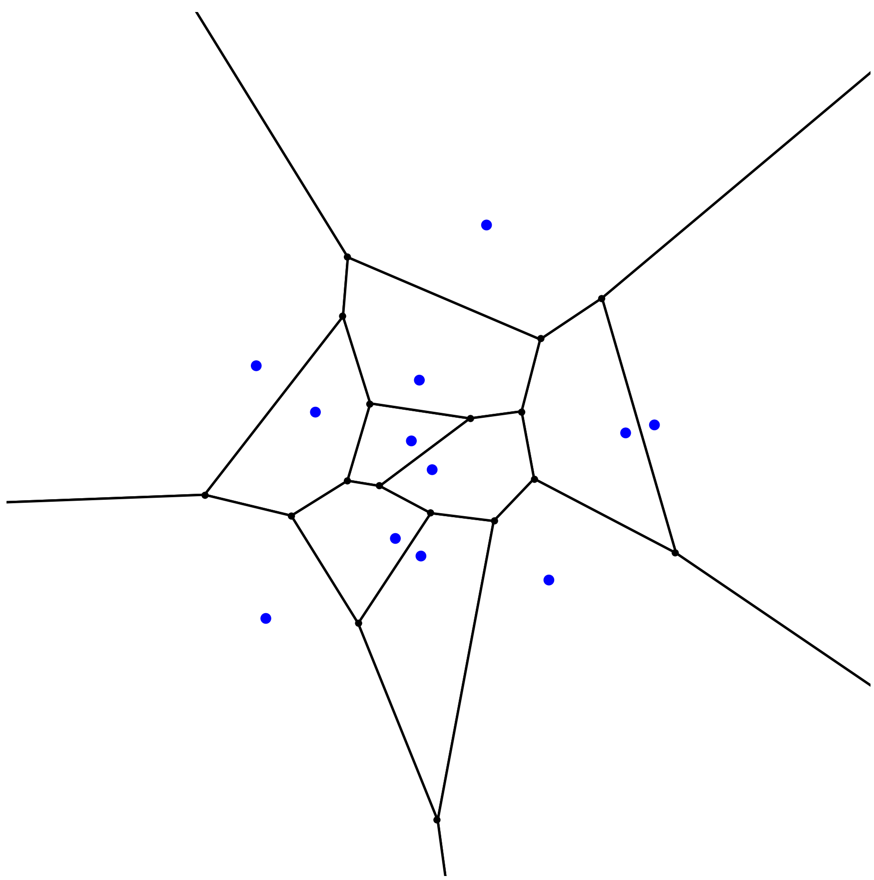
(c) What point location and line intersection operations does your algorithm require?

(d) What is the computational complexity of your algorithm?

2. How would your algorithm handle the following special cases?

(a) The new site is outside of the convex hull of the existing sites.

(b) The new site is on a Voronoi edge or Voronoi vertex.



INCREMENTAL VORONOI ALGORITHM

Given an existing Voronoi diagram (of k sites) and a new site p:

$O(k)$

(1) Find the Voronoi region that contains p. Say this is $Vor(p_1)$.
compute distances to existing sites
$O(k)$

(2) Construct the line that is the perpendicular bisector of $\overline{pp_1}$. $O(1)$
Let x_1 and x_2 be points of intersection between this line and edges $Vor(p_1)$. $O(k)$

(3) Continue from x_1, constructing edges of $Vor(p)$ one by one until we reach x_2. $O(k)$, since the total num. of edges is linear in k

(4) Remove the edges and verticies that are inside of $Vor(p)$. $O(k)$ edges and vertices to remove

Repeat for n sites: $O(n^2)$ overall

DATA STRUCTURE:

Common data structure for planar graphs:
Doubly Connected Edge List (DCEL)