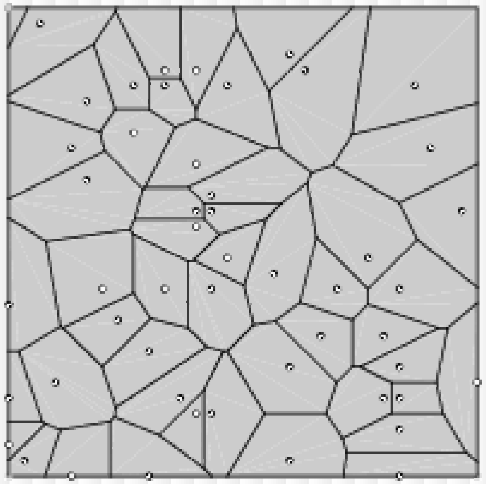
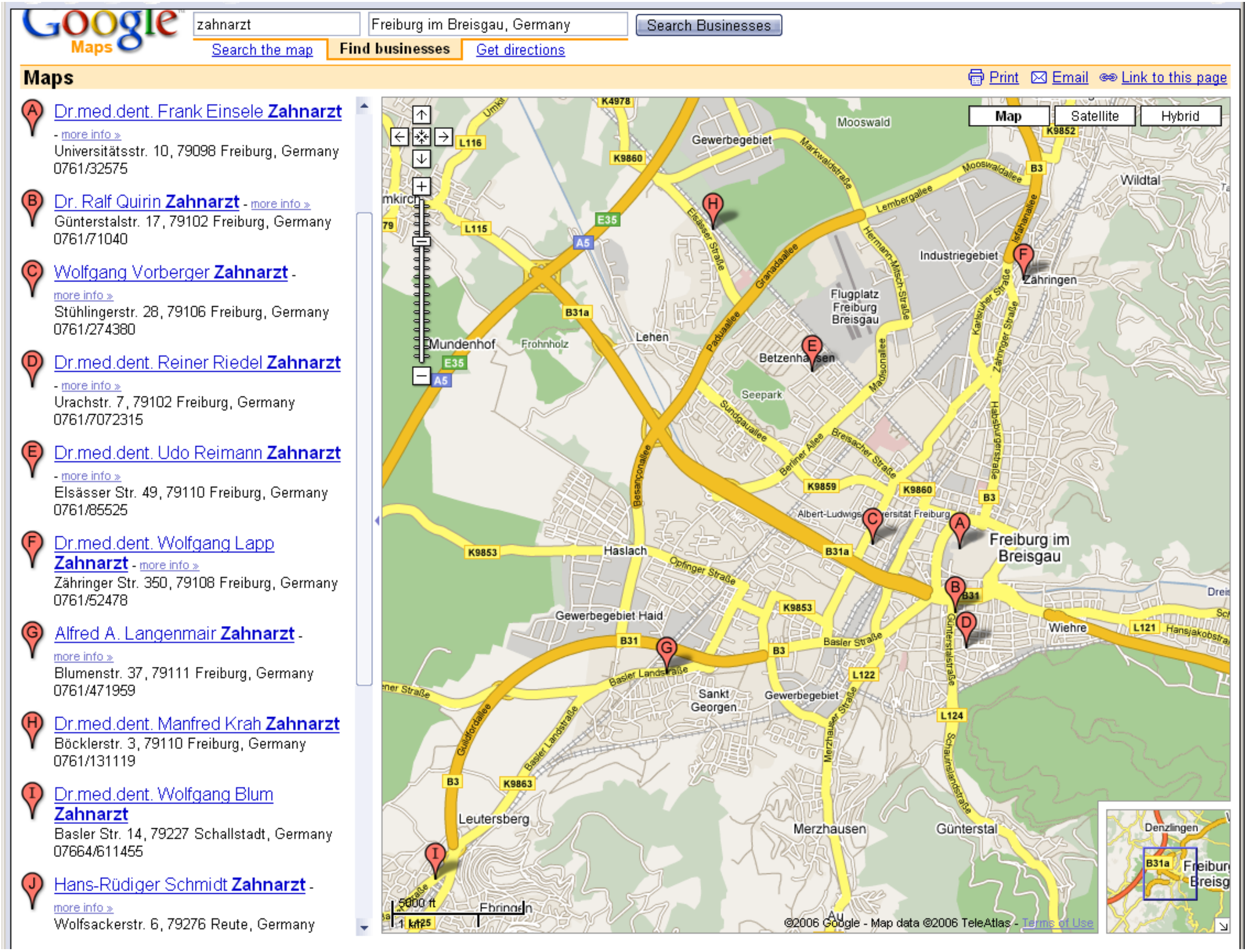
# Voronoi Diagram



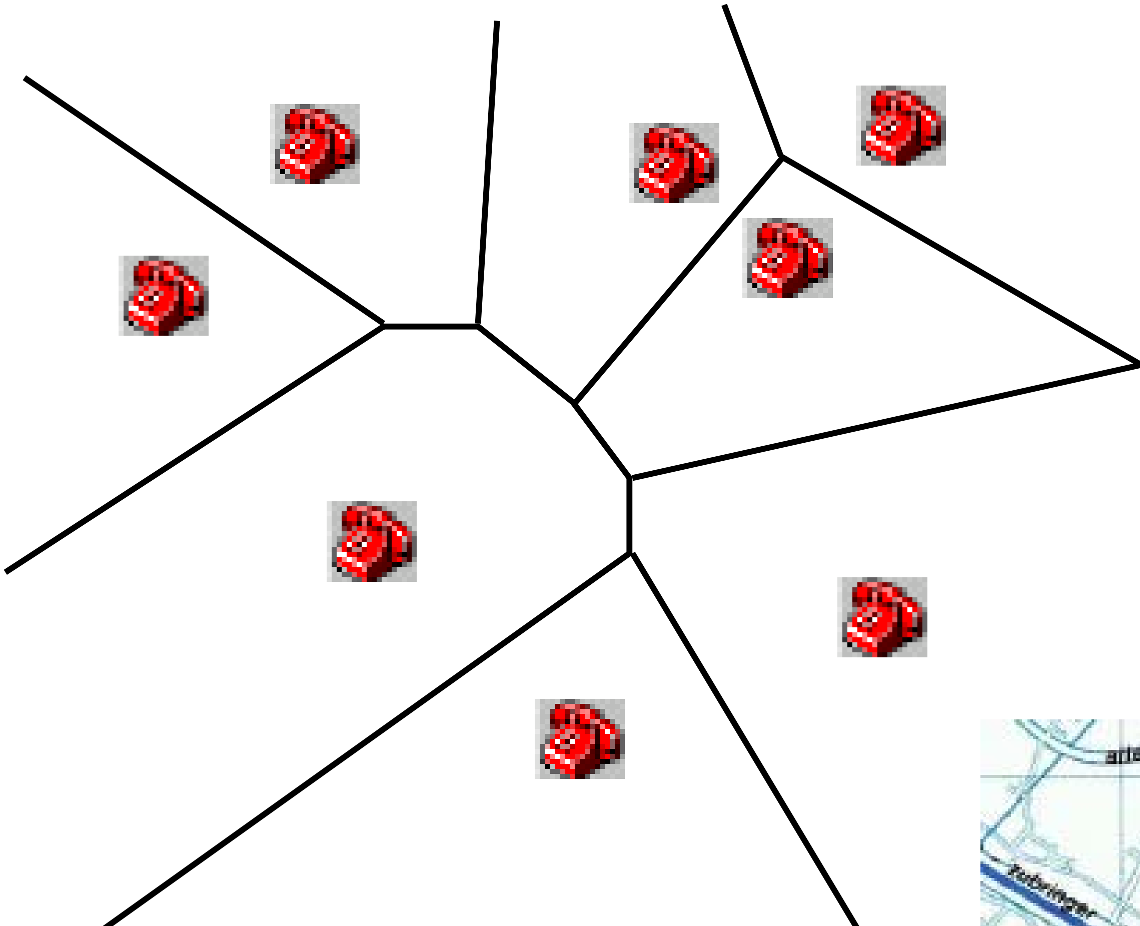
Subhas C. Nandy
Advanced Computing and Microelectronics Unit
Indian Statistical Institute
Kolkata 700108




Viewpoint 1: Locate the nearest dentist.

Viewpoint 2: Find the ‘service area’ of potential customers for each dentist.

# Voronoi Diagram



# Formal Definition

$P \rightarrow$ A set of *n* distinct points in the plane.

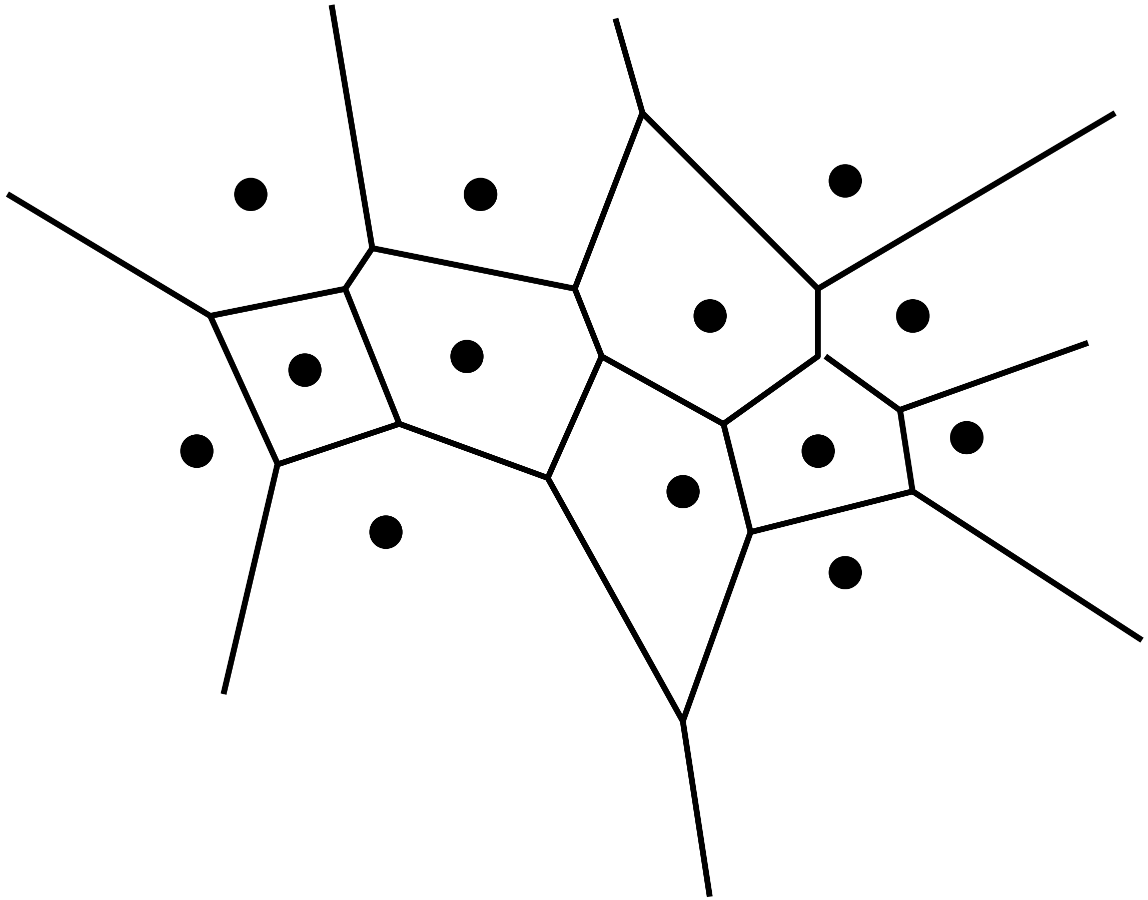
VD(*P*) $\rightarrow$ a subdivision of the plane into *n* cells such that

- each cell contains exactly one site,
- if a point *q* lies in a cell containing $p_i$ then
  $d(q, p_i) < d(q, p_j)$ for all $p_i \in P$, $j \neq i$.

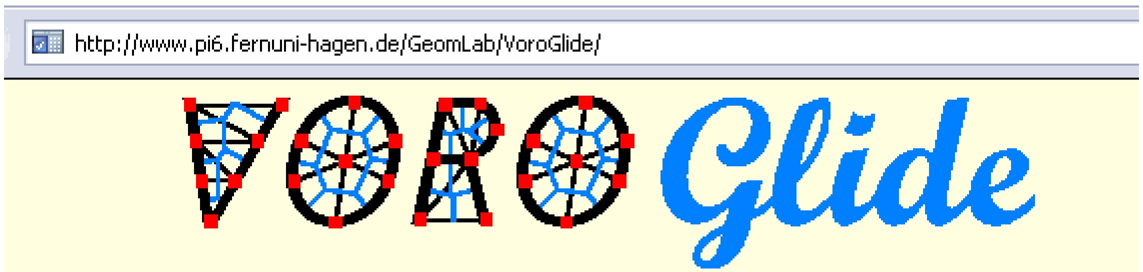
# Computing the Voronoi Diagram

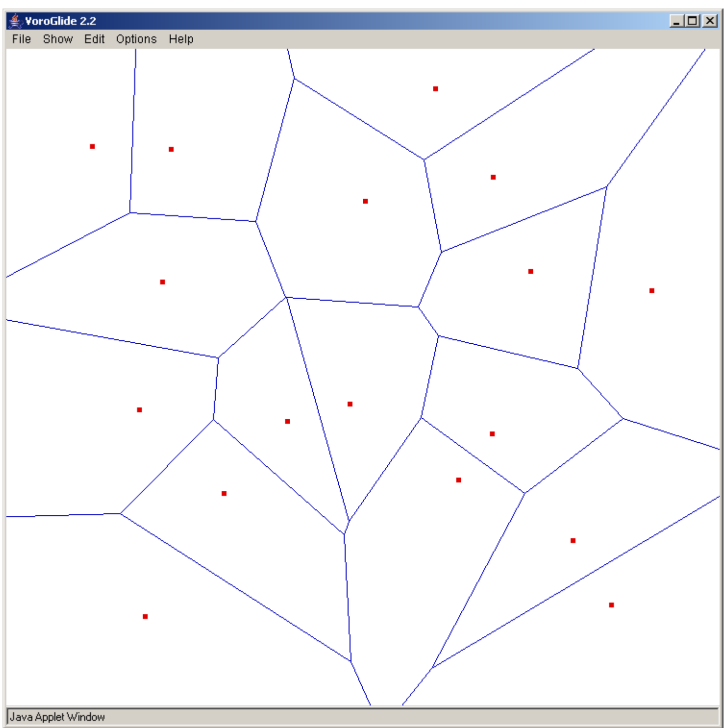
Input: A set of points (sites)

Output: A partitioning of the plane into regions of equal nearest neighbors



# Voronoi Diagram Animations




**Java applet animation of the Voronoi Diagram by:**

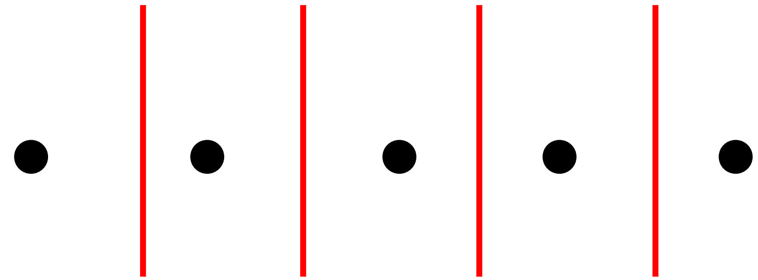
Christian Icking, Rolf Klein, Peter Köllner, Lihong Ma (FernUniversität Hagen)

# Characteristics of the Voronoi Diagram

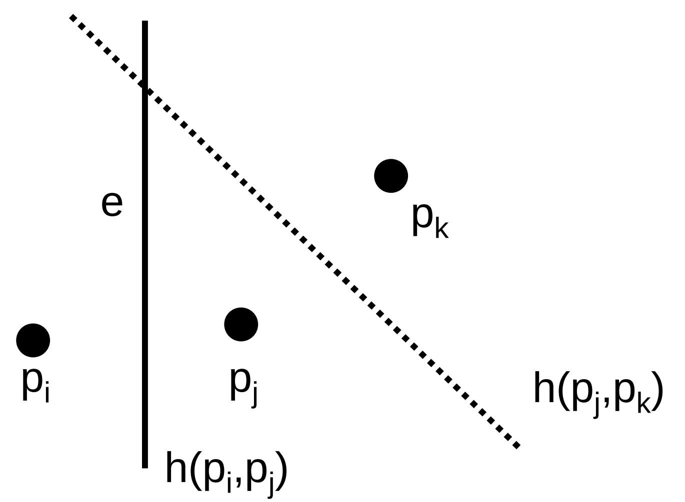
(1) Voronoi regions (cells) are bounded by line segments.

Special case :

Collinear points



Theorem : Let $P$ be a set of $n$ points (sites) in the plane.
If all the sites are collinear, then *Vor(P)* consist of $n-1$ parallel lines and $n$ cells.
Otherwise, *Vor(P)* is a connected graph and its edges are either line segments or half-lines.




If $p_i$, $p_j$ are not collinear with $p_k$, then $h(p_i, p_j)$ and $h(p_j, p_k)$ can not be parallel!

# Characteristics of the Voronoi Diagram

Assumption: No 4 points are co-circular.

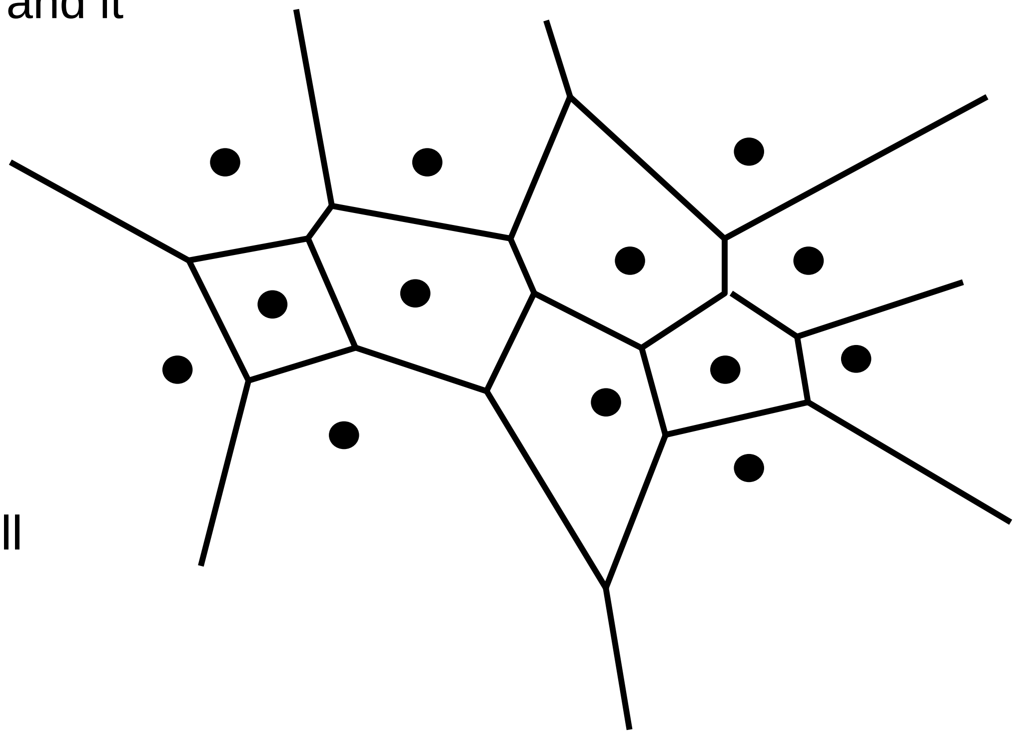
Each vertex (corner) of *VD(P)* has degree 3

The circle through the three points defines a vertex of the Voronoi diagram, and it does not contain any other point

The locus of the center of a largest empty circles passing through only a pair of points $p_i, p_j \in P$ defines an edge

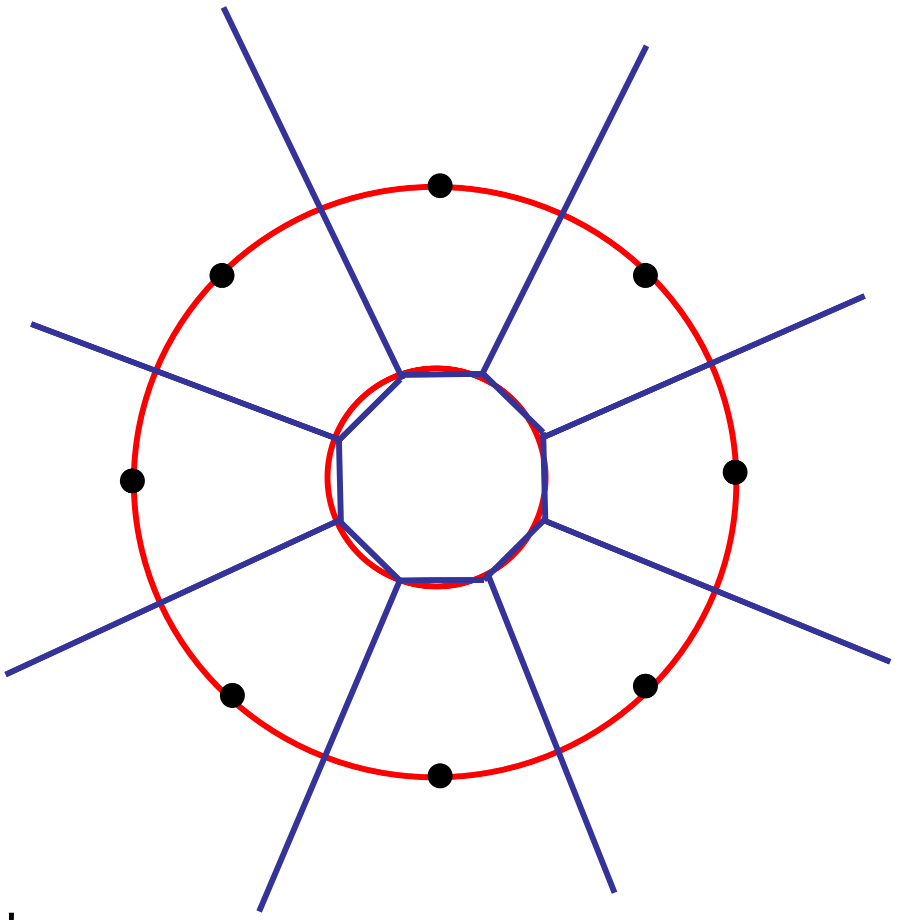
The locus of the center of largest empty circles passing through only one points in *P* defines a cell

The Voronoi region of a point is unbounded iff the point is a vertex of the convex hull of the point set.



# Degenerate Case with no bounded cells!

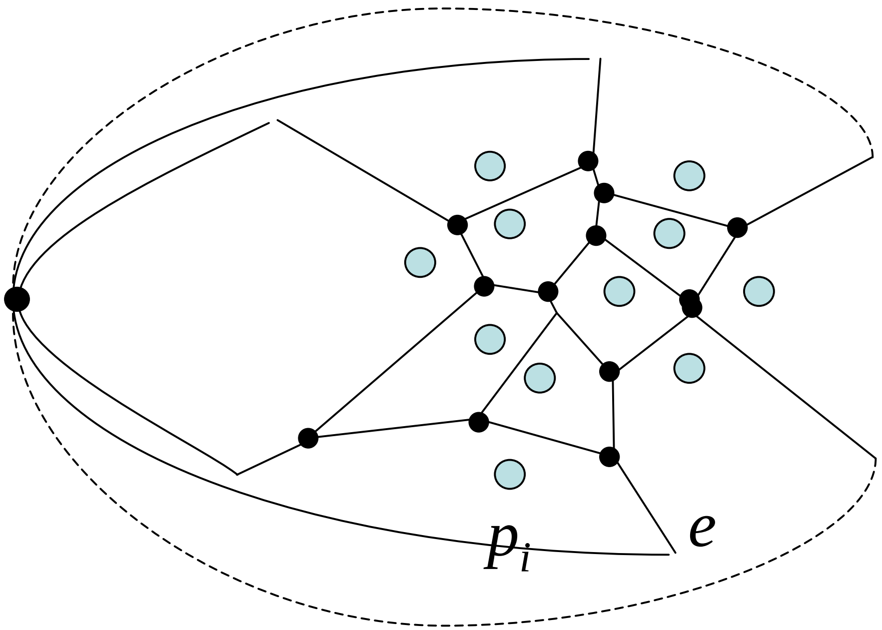
Size of the Voronoi Diagram:



*V(p)* can have O(*n*) vertices!

# Combinatorial Complexity of Voronoi Diagram

Theorem: The number of vertices in the Voronoi diagram of a set of *n* points in the plane is at most *2n-5* and the number of edges is at most *3n-6.*




Proof:

1. Connect all Half-lines with fictitious point $\infty$
2. 2. Apply Euler`s formula:
   $v - e + f = 2$

For *VD(P)* + $\infty$ *:*

$v$ = number of vertices of *VD(P)* + *1*
$e$ = number of edges of *VD(P)*
$f$ = number of sites of *VD(P)* = *n*

# Proof (Continued)

Each edge in *VD(P)* + $\infty$ has exactly two vertices and each vertex of *VD(P)* + $\infty$ has at least a degree of 3:

$\Rightarrow$ sum of the degrees of all vertices of *Vor(P)* + $\infty$

$= 2 \cdot (\# \text{ edges of } VD(P))$

$\geq 3 \cdot (\# \text{ vertices of } VD(P) + 1)$

Number of vertices of *VD(P)* = $v_p$

Number of edges of *VD(P)* = $e_p$

We can apply: $(v_p + 1) - e_p + n = 2$

$$2\,e_p \geq 3\,(v_p + 1)$$

$$2\,e_p \geq 3\,(2 + e_p - n)$$

$$= 6 + 3e_p - 3n$$

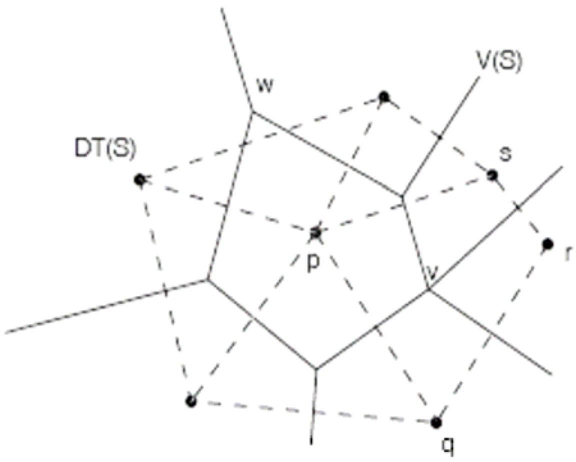
$$3n - 6 \geq e_p$$

# Voronoi Diagram and Delaunay Tessellation

**Delaunay triangulation DT(S):**

A tessellation obtained by connecting a pair of points $p.q \in S$ with a line segment if a circle $C$ exists that passes through $p$ and $q$ and does not contain any other site of $S$ in its interior or boundary.

The edges of DT($S$) are called *Delaunay edges*.

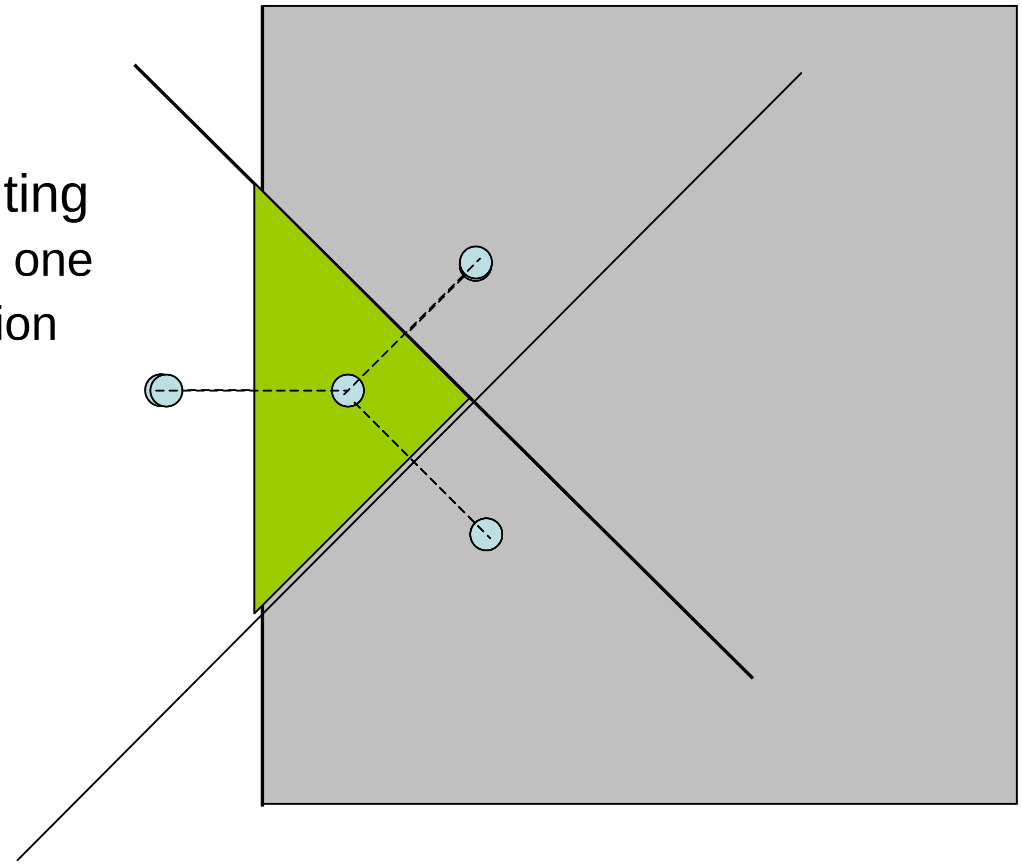



1. **Two points in S are joined by a Delaunay edge if their Voronoi regions are adjacent.**
2. **If no four points of S are cocircular then DT(S) – the dual of the Voronoi diagram V(S) – is a triangulation of S DT(S) is called the Delaunay triangulation.**
3. **Three points of S give rise to a Delaunay triangle if their circumcircle does not contain a point of S in its interior.**

# Construction of Voronoi Diagram
# A simple algorithm

Given an algorithm for computing the intersection of halfplanes, one can construct the Voronoi region of each point separately.

This needs $O(n^2 \log n)$ time



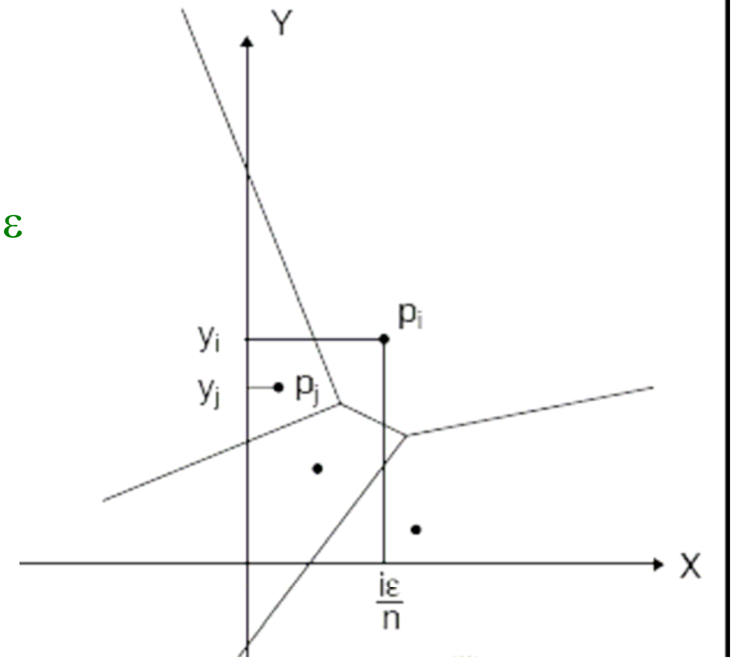
# Lower bound proof

Time Complexity for Computing Voronoi Diagram is $\Omega(n \log n)$

Proof: Using reduction from $\varepsilon$-closeness

Suppose $y_1, y_2, \ldots, y_n$ be $n$ real numbers
Does there exists $i \neq j$ *such that* $|y_i - y_j| \leq \varepsilon$

Define points $p_i = (i\varepsilon/n, y_i)$, $i = 1, 2, \ldots n$



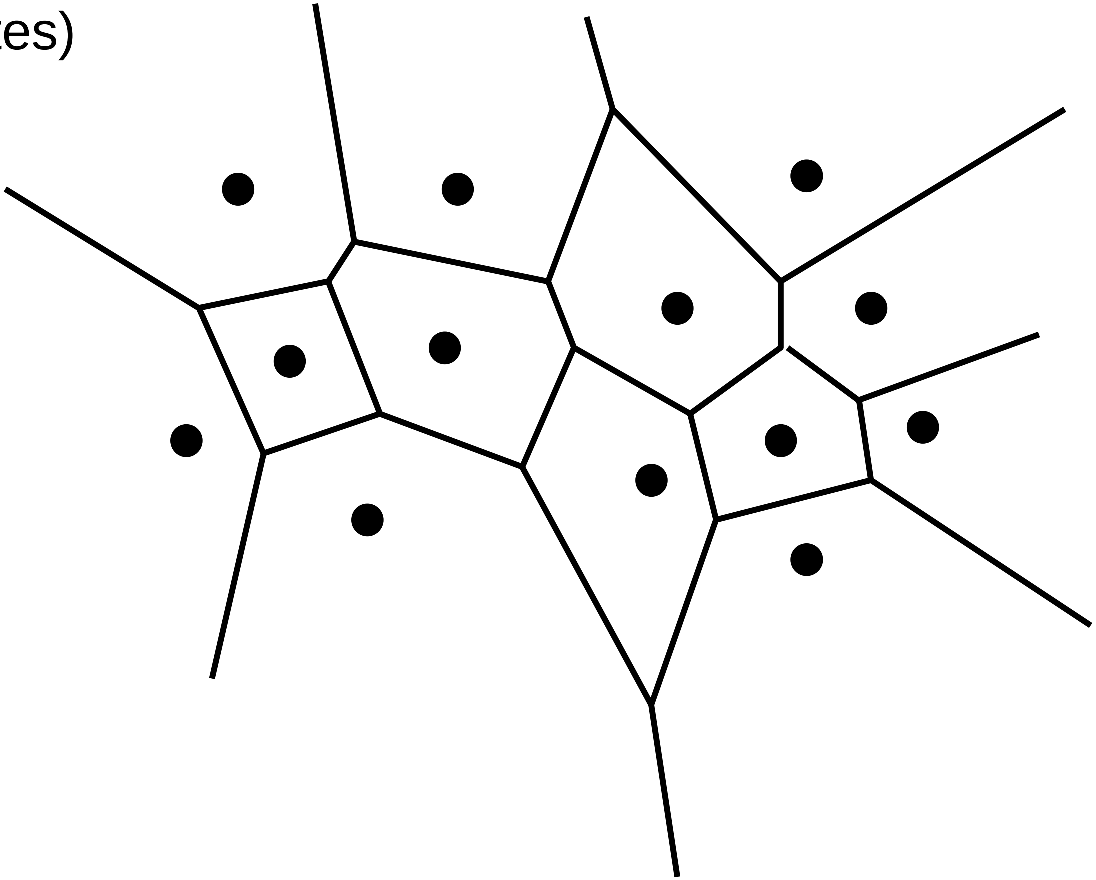

1. Compute the Voronoi Diagram
2. In $O(n)$ time, it can be checked that every Voronoi region is intersected by the y-axis in bottom-up order.
3. If for each $p_i$, its projection onto y-axis lies in its Voronoi region, then the order of $y_i$'s in decreasing order is available. Next check the desired condition in $O(n)$ time.
4. Otherwise there exists a $p_i$ whose projection falls in the Voronoi region of some $p_j$ In such a case $|y_i - y_j| < \varepsilon$ holds since

$$|y_i - y_j| \leq dist((0,y_i), p_j) < dist((0,y_i), p_i) \leq \varepsilon$$

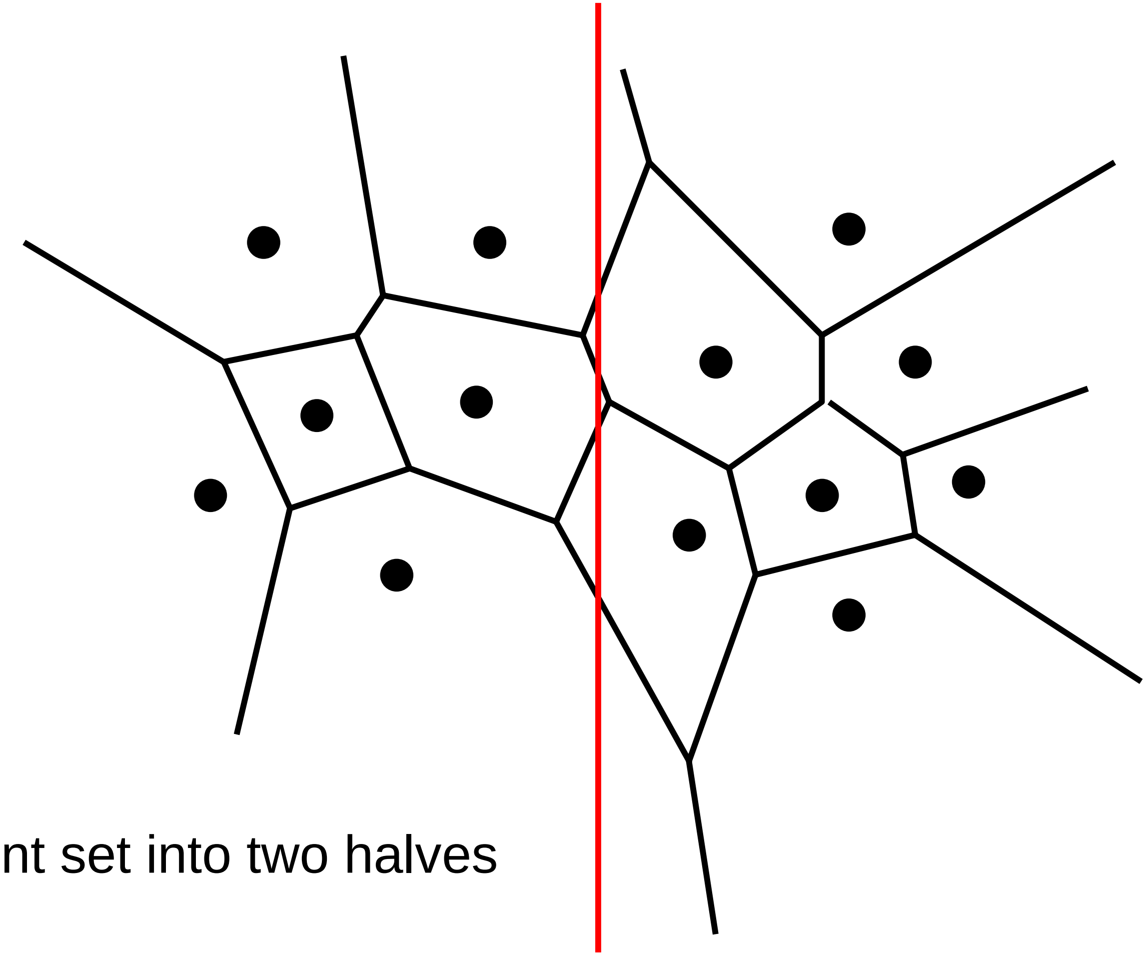
# Construction of Voronoi Diagram using divide and conquer

Input: A set of points (sites)

Output: A partitioning of the plane into regions of equal nearest neighbors.



# Divide and conquer: Divide Step

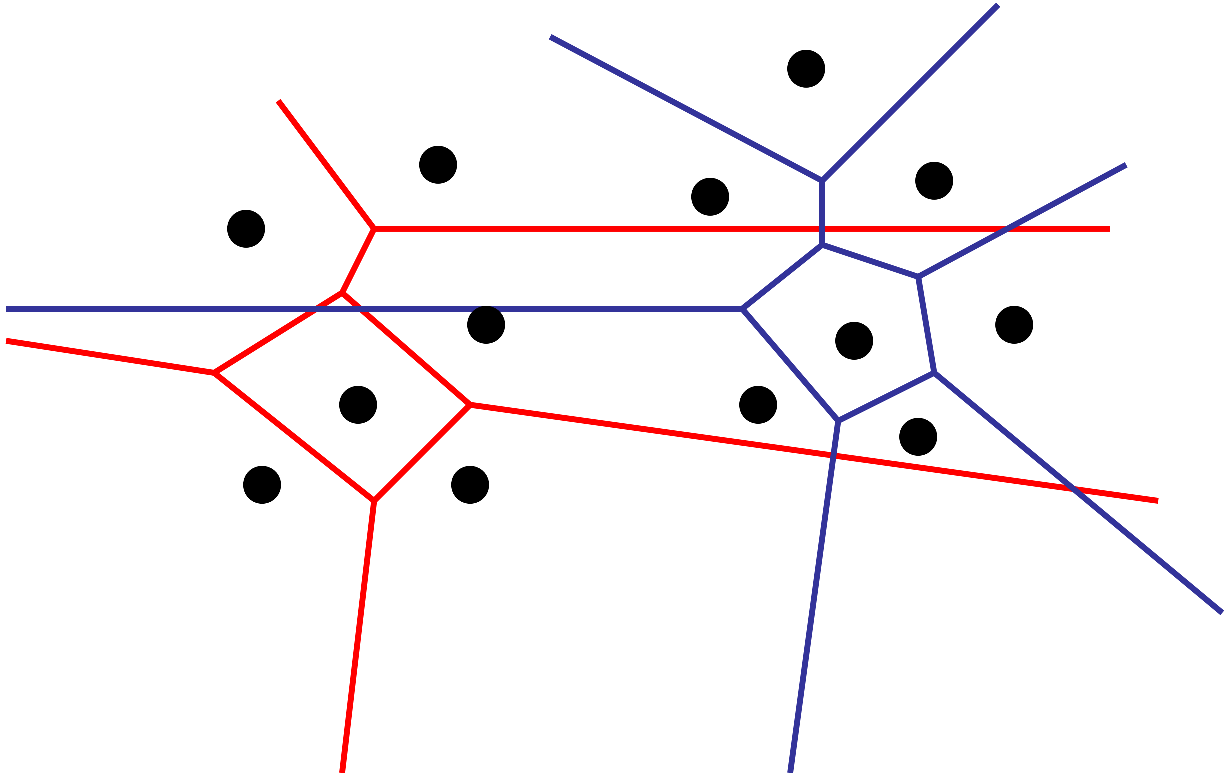


Divide: Divide the point set into two halves

# Divide and Conquer: Conquer Step

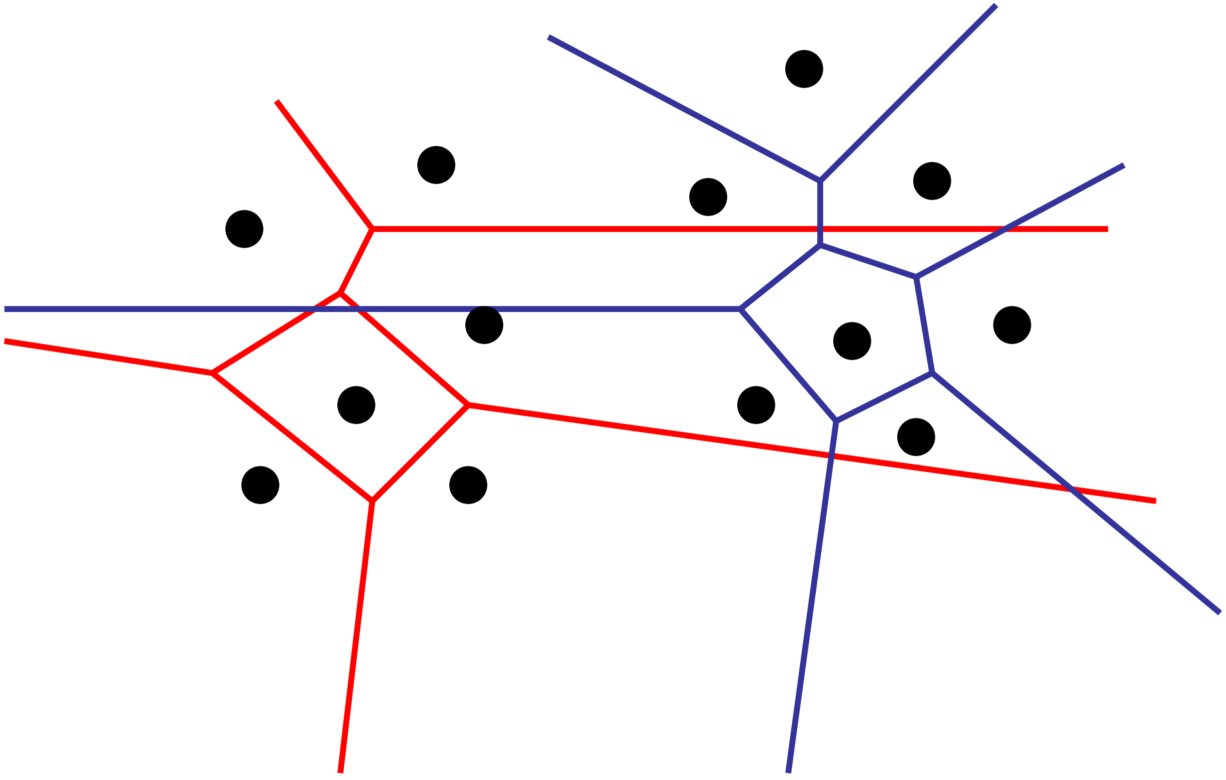
Conquer: Recursively compute the Voronoi diagrams for the smaller point sets.

Abort condition: Voronoi diagram of a single point is the entire plane.



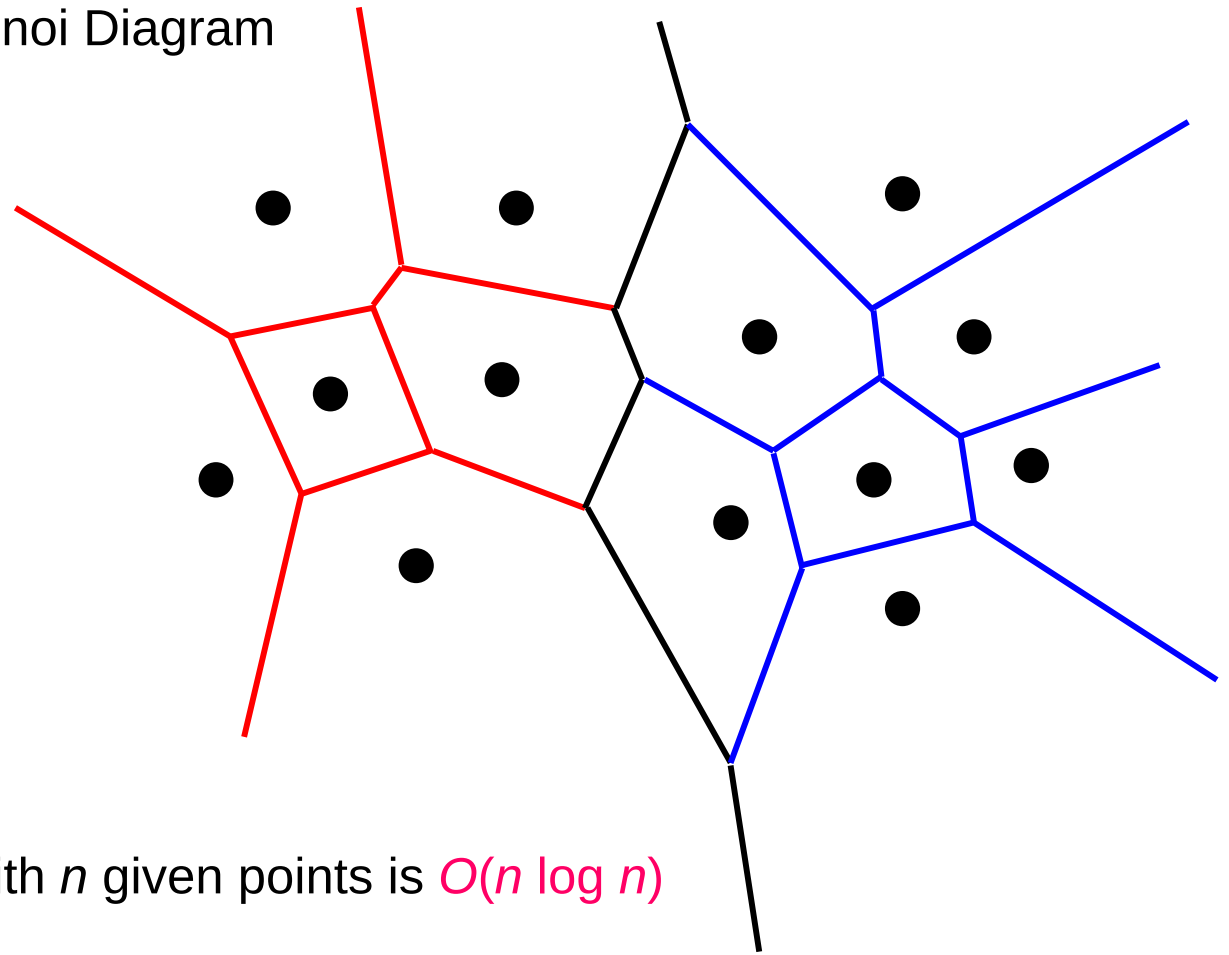
# Divide and Conquer: Merge

Merge the diagrams by a (monotone) sequence of edges



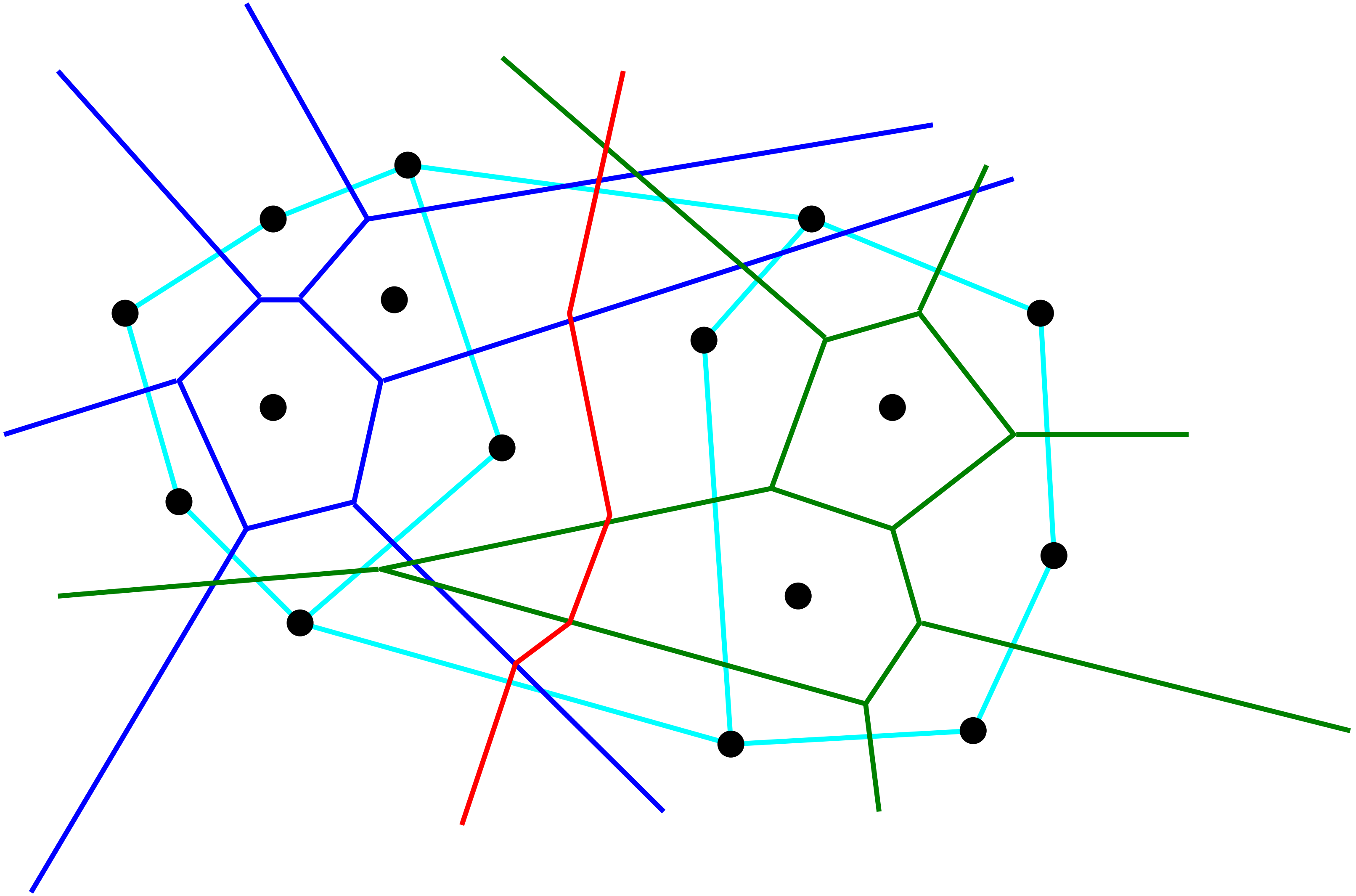
# The Result

The finished Voronoi Diagram



Running time: With *n* given points is $O(n \log n)$

# Example



# Fortune’s line sweep algorithm

It is an incremental construction

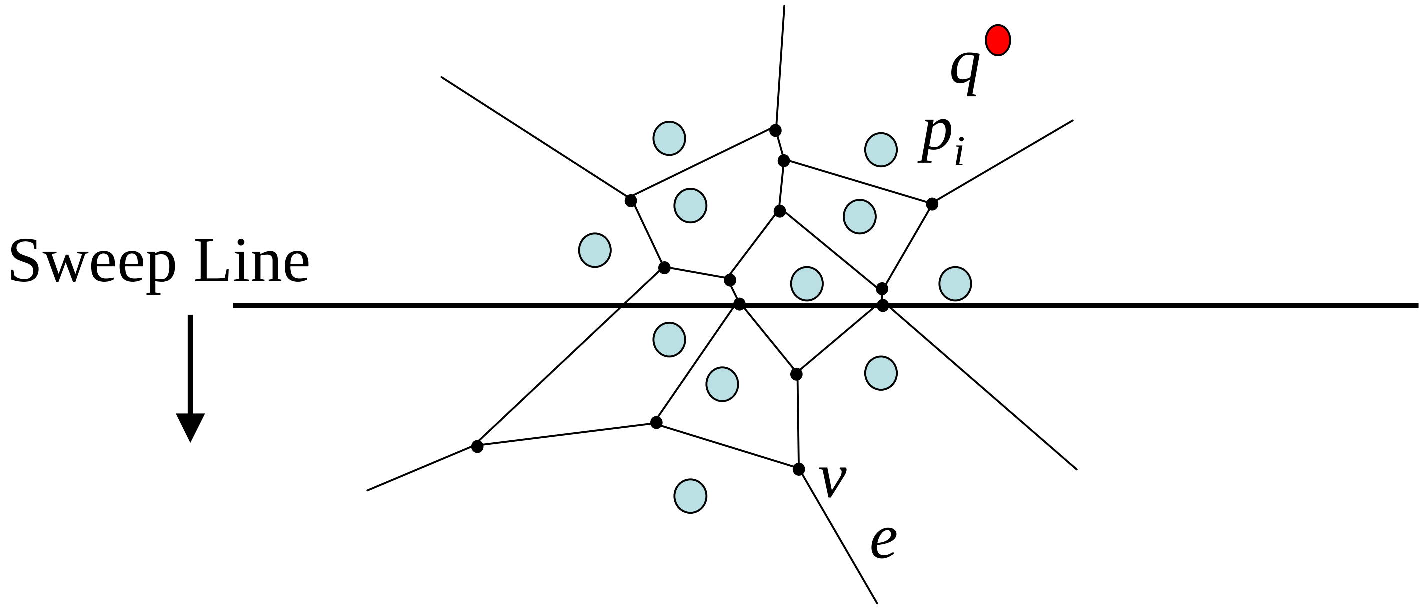
A horizontal line is swept among the sites from top to bottom

It maintains portion of Voronoi diagram which does not change due to the appearance of new sites below sweep line;

It keeps track of incremental changes of the Voronoi diagram that is caused for the appearance of each site on the sweep line.

# Construction of Voronoi diagram

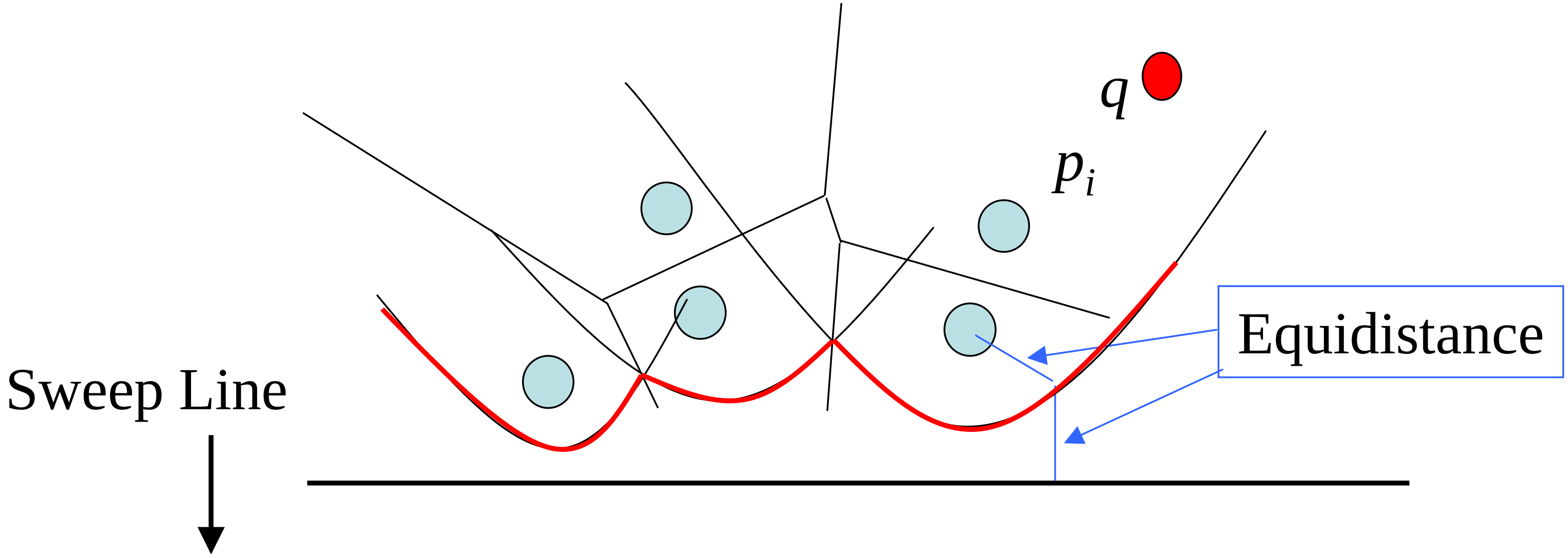
What is the invariant we are looking for?




It maintains a representation of the locus of the point *q* that are at the same distance from some site $p_i$ *above* the sweep line and the line itself.

# Construction of Voronoi diagram (contd.)

Which points are closer to a site above the sweep line than to the sweep line itself?

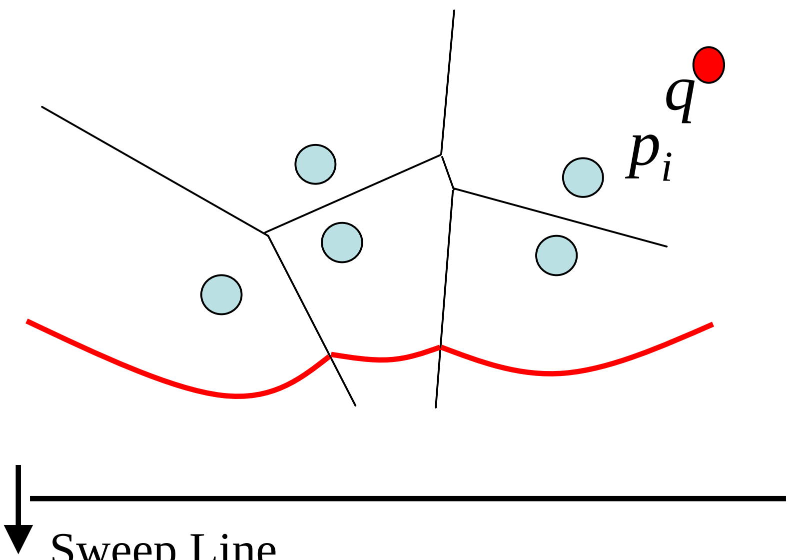



The set of parabolic arcs form a beach-line that bounds the locus of all such points

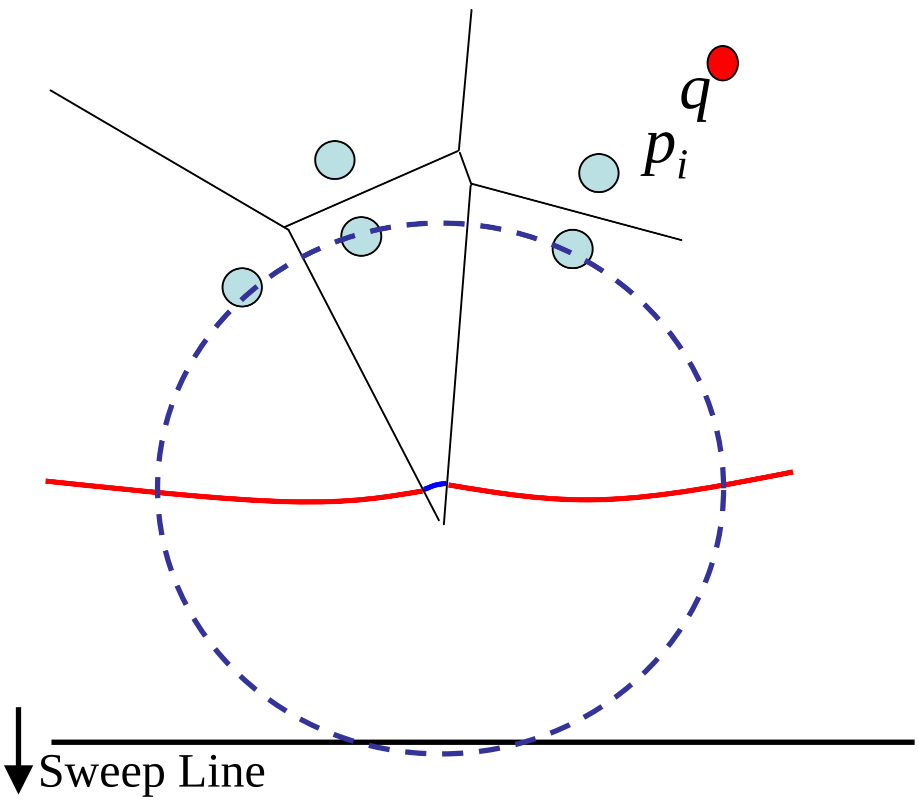
Break points trace out Voronoi edges

# Construction of Voronoi diagram (contd.)

Arcs flatten out as sweep line moves down

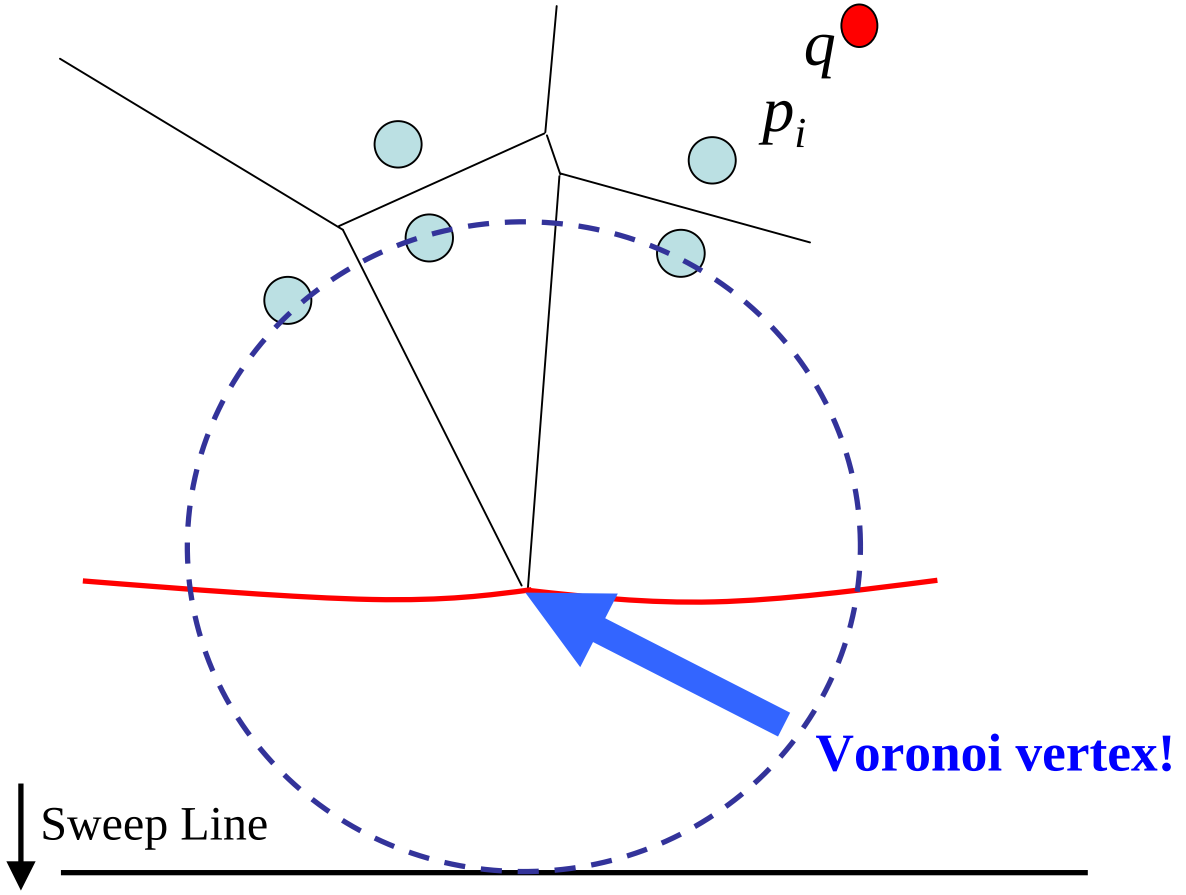



Eventually, the middle arc disappears

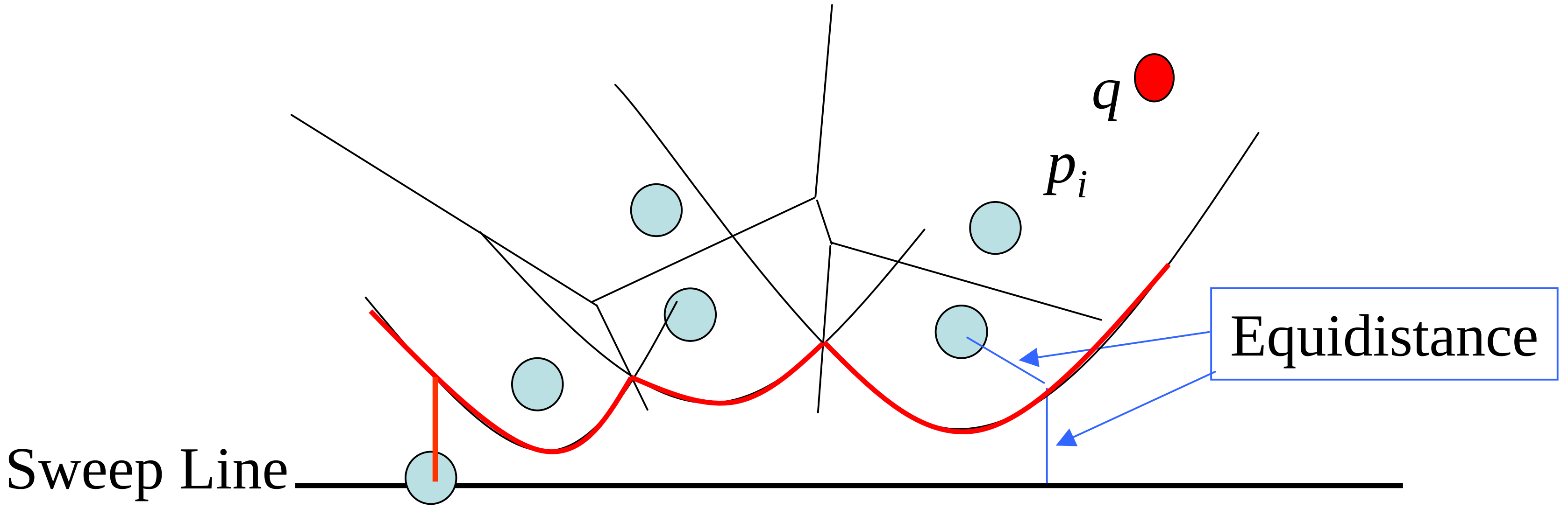


# Construction of Voronoi diagram (contd.)

Thus, we have detected a circle that contains no site in *P* and touches 3 or more sites.



# Construction of Voronoi diagram (contd.)




When a new site appears on the sweep line,
a new arc appears on the beach line

# Beach Line properties

- Voronoi edges are traced by the break points as the sweep line moves down.

  Emergence of a new break point (due to the formation of a new arc or a fusion of two existing break points) identifies a new edge

- Voronoi vertices are identified when two break points meet (fuse).

  Decimation of an old arc identifies new vertex

# Data Structures

Current state of the Voronoi diagram

Doubly linked list (*D*) containing half-edges, edges, vertices and cell records

Current state of the beach line (*T*)

Keeps track of break points, and the arcs currently on beach line

Current state of the sweep line (Event queue)

Priority queue on decreasing y-coordinate

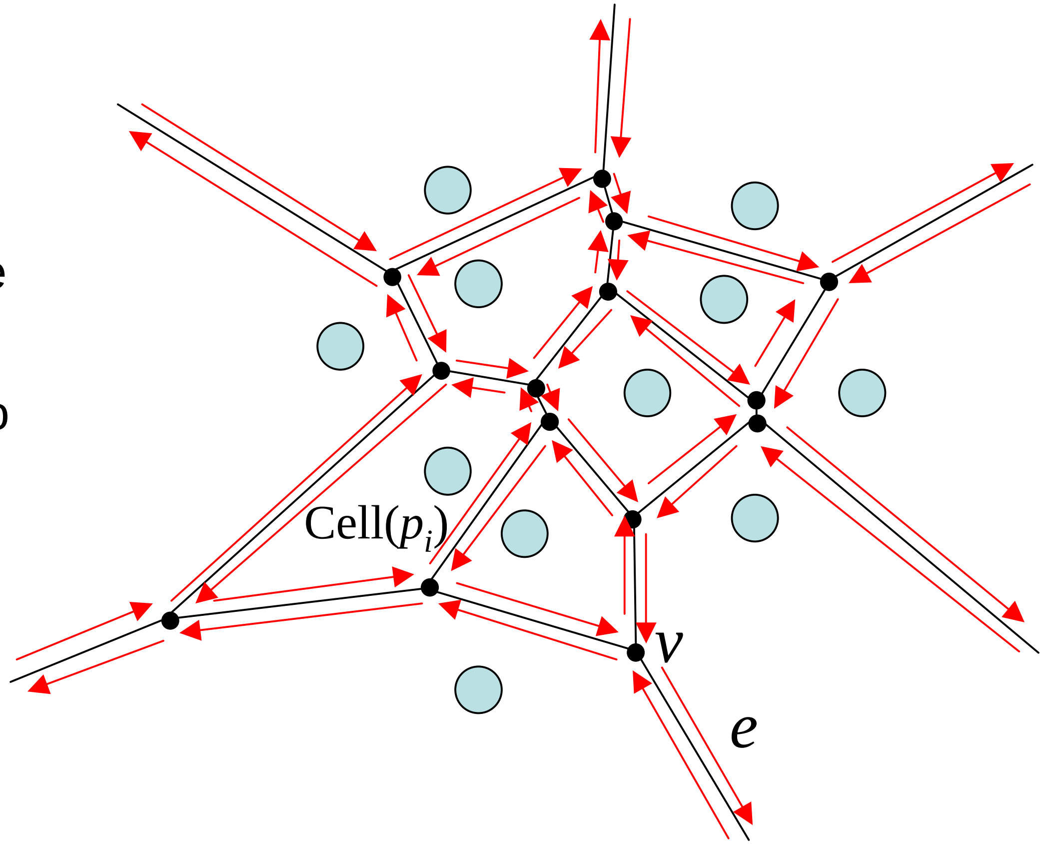
# Doubly-linked list (D)

A simple data structure that allows an algorithm to traverse a Voronoi diagram’s vertices, edges and cells

Consider edges as a pair of uni-directional half-edges

A chain of counter-clockwise half-edges forms a cell

Define a half-edge’s “twin” to be its opposite half-edge of the same Voronoi edge



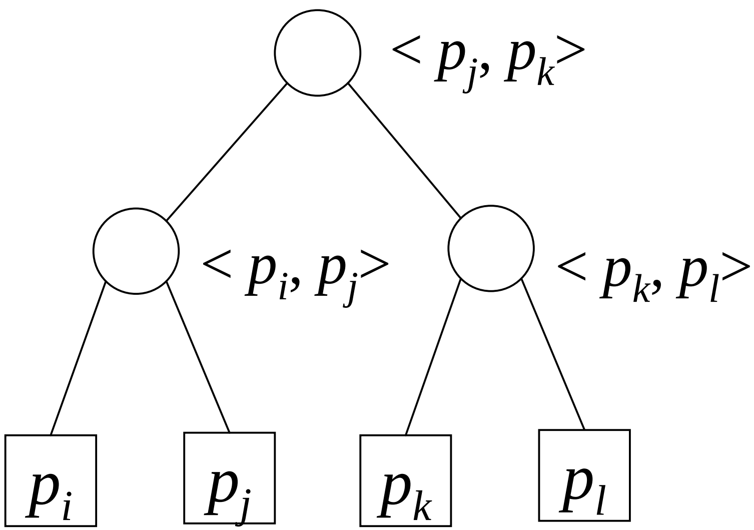
# Beach Line Data Structure ($T$)

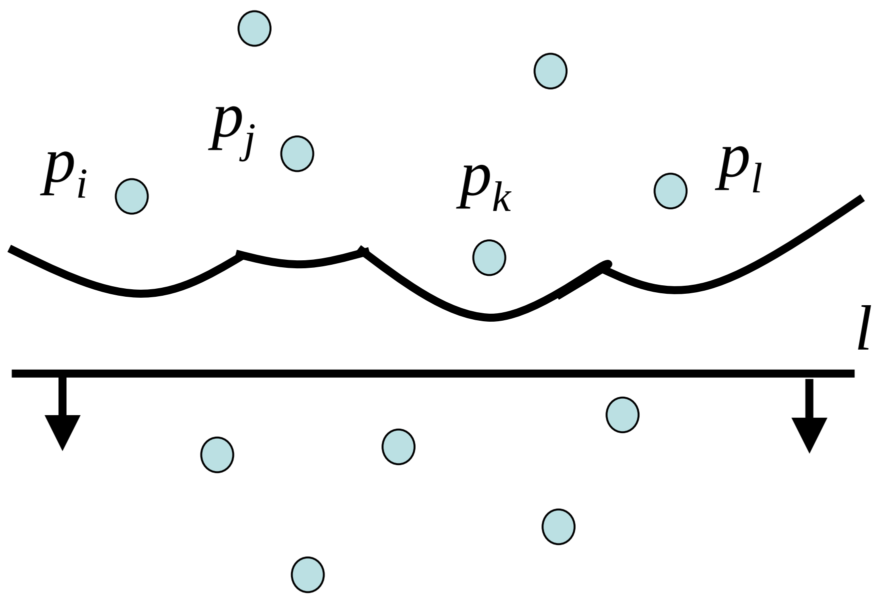
It is a balanced binary search tree

Internal nodes represent break points between two arcs

Leaf nodes represent arcs, each arc in turn is represented by the site that has generated it

- It also contains a pointer to a potential circle event



# Event Queue (*Q*)

Consists of

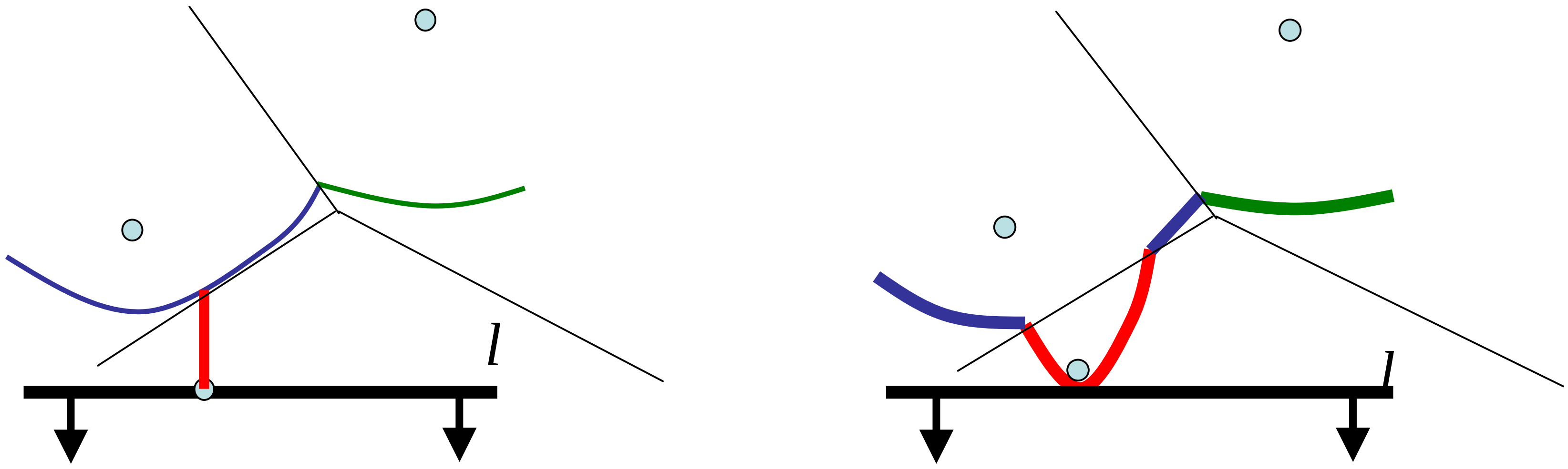
Site Events (when the sweep line encounters a new site point)

Circle Events (when the sweep line encounters the *bottom* of an empty circle touching 3 or more sites).

It is prioritized with respect to the decreasing order of the y-coordinate of the events

# Site Event

A new arc appears when a new site appears

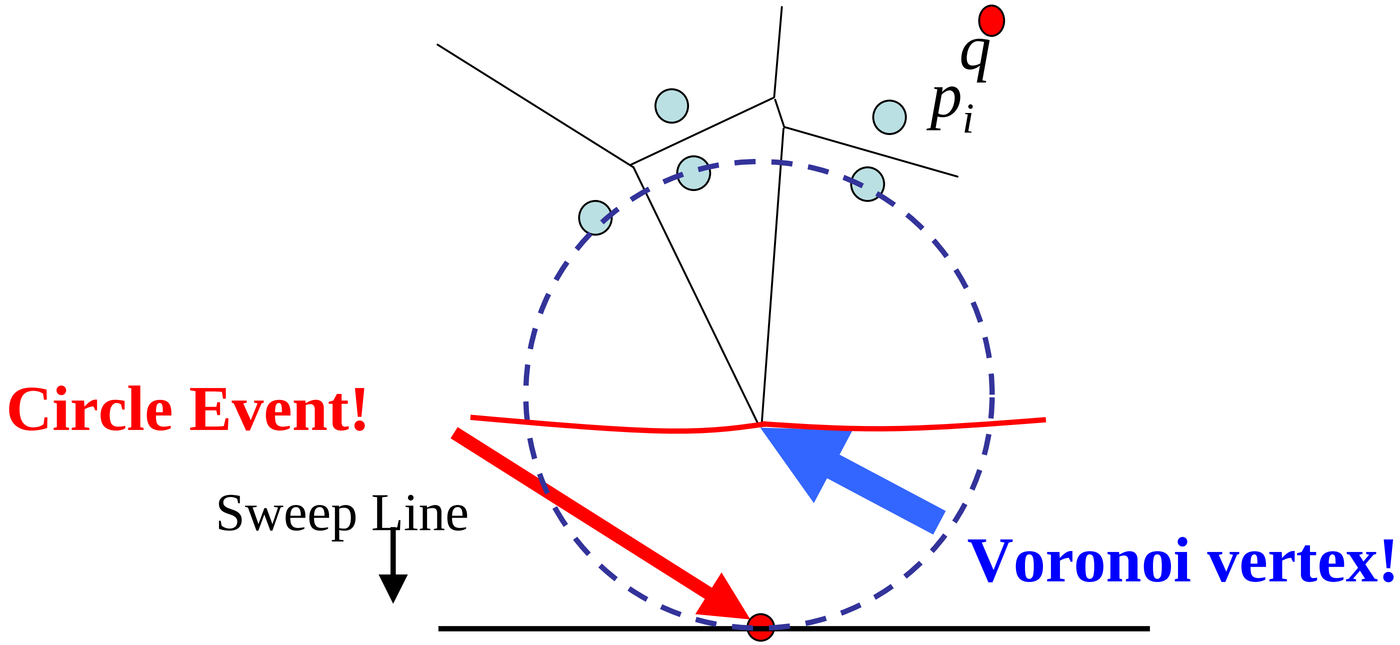



Original arc above the new site is broken into two

⇒ Number of arcs on beach line is $O(n)$

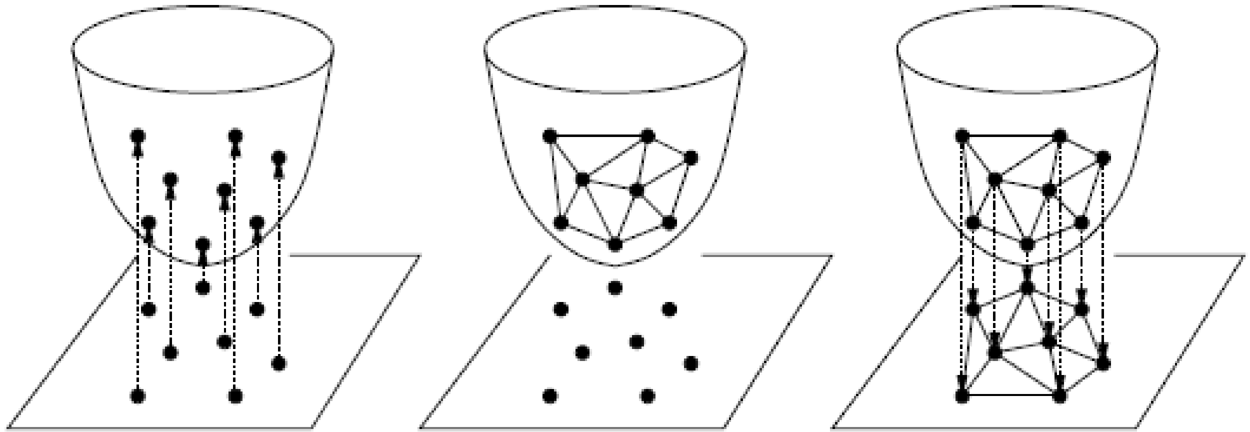
# Circle Event

An arc disappears whenever an empty circle touches three or more sites and is tangent to the sweep line.




Sweep line helps determine that the circle is indeed empty.

# Voronoi diagram: A different Formulation

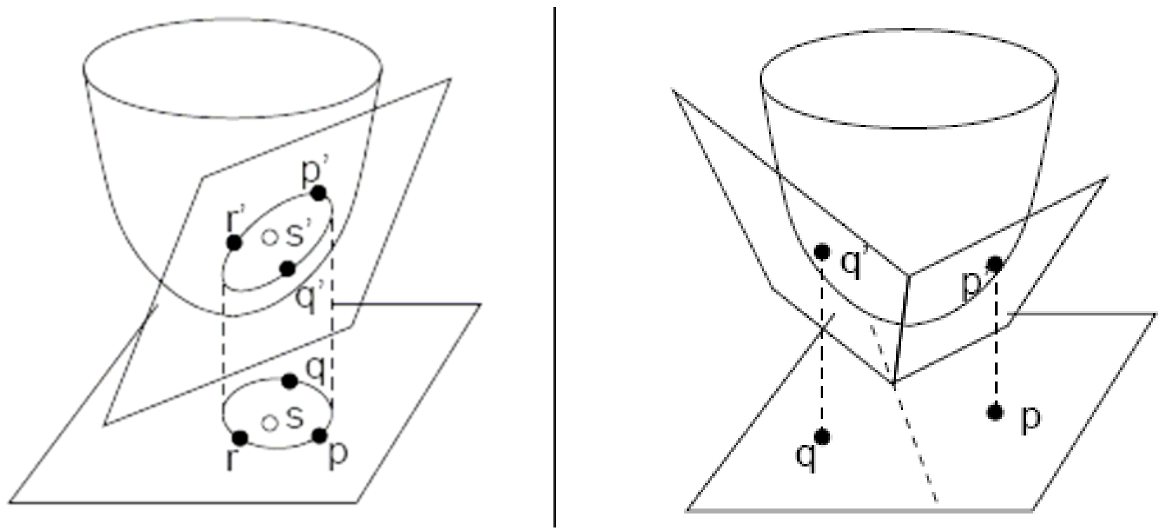



1. Project each point $p_i$ on the surface of a unit paraboloid
2. Compute the lower convex hull of the projected points.

Result: Given $S = \{p_i | i=1, 2, \ldots n\}$ in the plane (no 4 points co-circular) and given 3 points $p$, $q$, $r \in S$, the triangle $\Delta$ pqr is a triangle of Delauney triangulation if $\Delta\, p'q'r'$ is a face of the lower convex hull of the projected points $S'$

Conclusion: The projection of this convex hull gives the Delauney Triangulation of the point set.

# Voronoi diagram: A different Formulation




1. Project each point $p_i$ on the surface of a unit paraboloid
2. Draw tangent planes of the paraboloid at every projected point.
3. Compute the upper envelope of these planes.

Result: The projection of this upper envelope gives the Voronoi diagram of the point set.

# Voronoi diagram in Laguerre geometry

Define the distance of two points $p = (x_1, y_1, z_1)$ and $q = (x_2, y_2, z_2)$ in $\mathrm{R}^3$ is

$$D^2(p,q) = (x_1 - x_2)^2 + (y_1 - y_2)^2 - (z_1 - z_2)^2$$

In Laguerre geometry

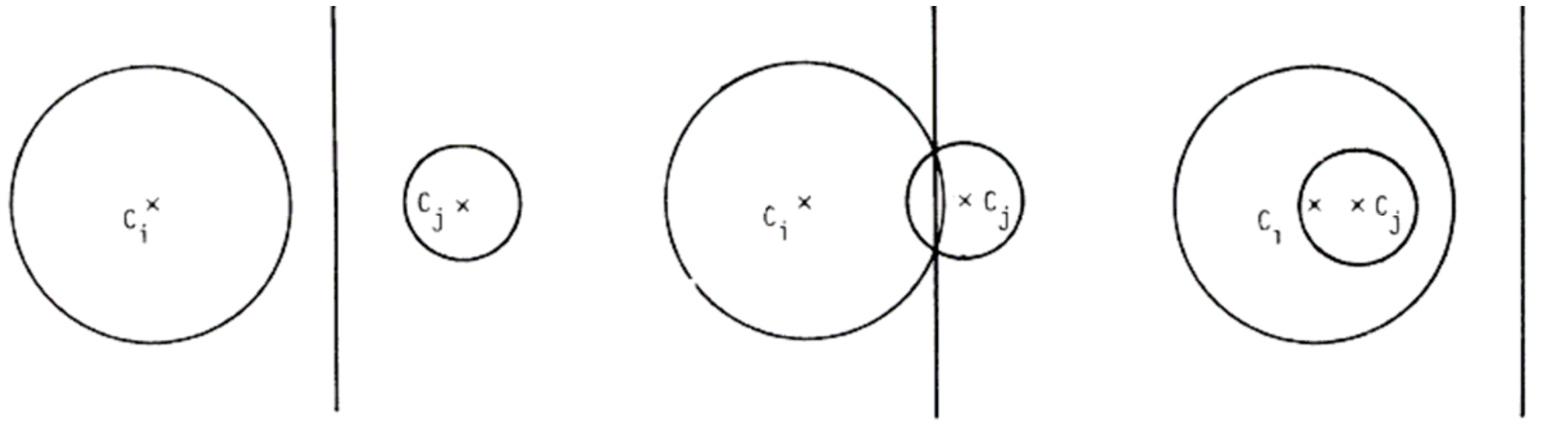
A point $(x, y, z)$ is mapped to a circle in the Euclidean plane with center $(x, y)$ and radius $|z|$

The distance between a pair of points in $\mathrm{R}^3$ corresponds to the length of the common tangent of the corresponding two circles

The distance of a point $p = (x, y)$ from a circle $C_i(Q_i, r_i)$ with center $Q_i = (x_i, y_i)$ and radius $r_i$
= length of the tangent segment of the circle $C_i(Q_i, r_i)$ from point $p = (x, y)$
$= D_L^2(C_i,p) = (x_i - x)^2 + (y_i - y)^2 - r_i^2$

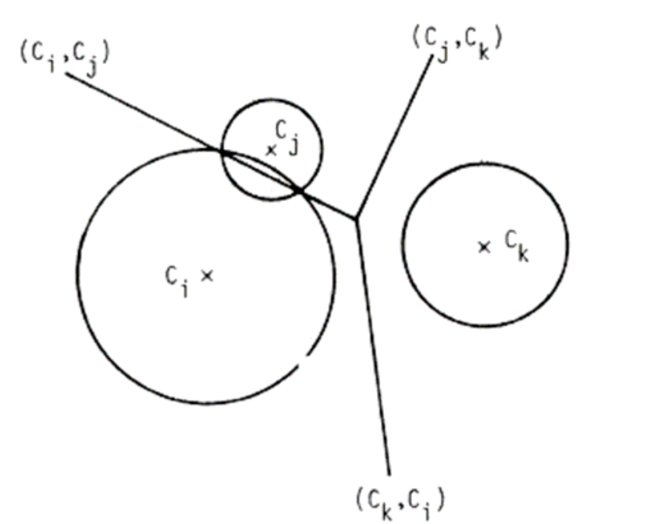
$D_L^2(C_i,p)$ is negative, zero or positive depending on whether p lies inside, on or outside $C_i$

# Voronoi diagram in Laguerre geometry




*Radical axis:* Locus of the points equidistant from two circles $C_i$ and $C_j$.

*Radical center:* If the centers of three circles are not collinear, then the radical axes of ($C_i$ and $C_j$), ($C_j$ and $C_k$) and ($C_i$ and $C_k$) meet at a point.



# Voronoi diagram in Laguerre geometry

*Voronoi Polygon:* Suppose *n* circles $C_i(Q_i, r_i)$ are given in the plane.
Distance of $C_i$ and a point *p* is defined by $D_L(C_i,p)$,
Then the Voronoi polygon $V(C_i)$ for circle $C_i$ is

$$V(C_i) = \cap \{p \in R^2 \mid D_L^2(C_i,p) \leq D_L^2(C_j,p)\}$$

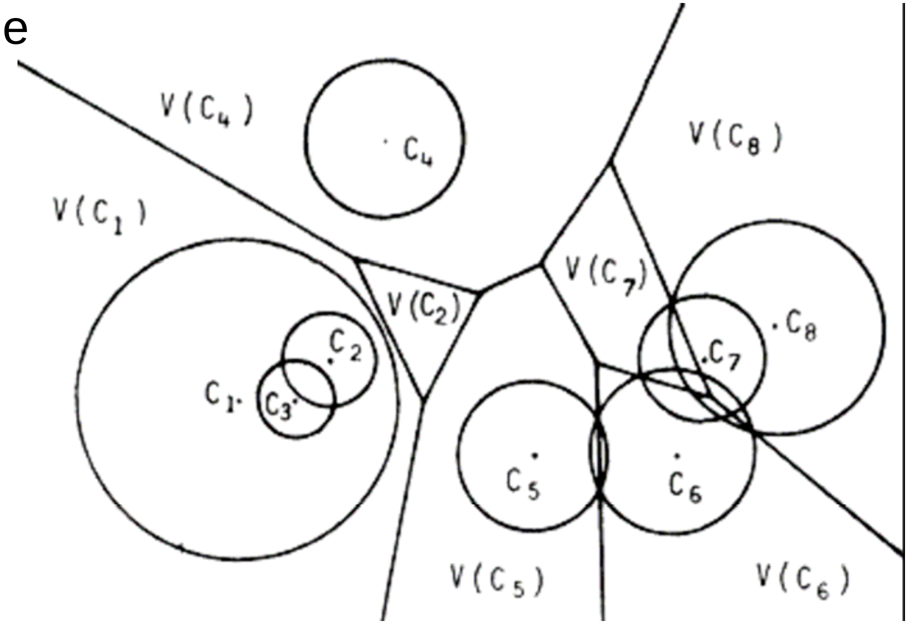
Voronoi polygons partition the whole plane

$V(C_i)$ is always convex

$V(C_i)$ may be empty if $C_i$ is contained in the union of other circles

A circle whose Voronoi polygon is non-empty is called *substantial circle*

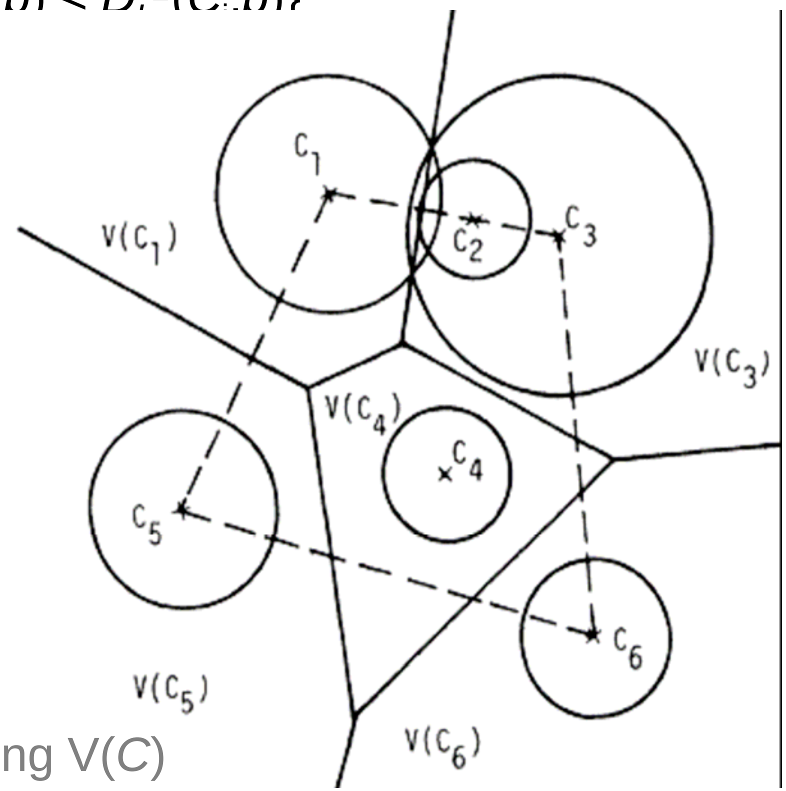
A circle whose Voronoi polygon is empty is called *trivial circle* ($C_3$ is a trivial circle)



# Voronoi diagram in Laguerre geometry

*Voronoi Polygon:* Suppose $n$ circles $C_i(Q_i, r_i)$ are given in the plane.
Distance of $C_i$ and a point $p$ is defined by $D_L(C_i,p)$,
Then the Voronoi polygon $V(C_i)$ for circle $C_i$ is

$$V(C_i) = \cap \{p \in R^2 \mid D_L^2(C_i,p) < D_L^2(C_j,p)\}$$




A circle that intersects its Voronoi polygon is said to be proper; otherwise it is improper.

A trivial circle is necessarily improper

If $V(C_i)$ is non-empty and unbounded then the center of $C_i$ is at a corner of the convex hull of the centers of $C_i$'s.

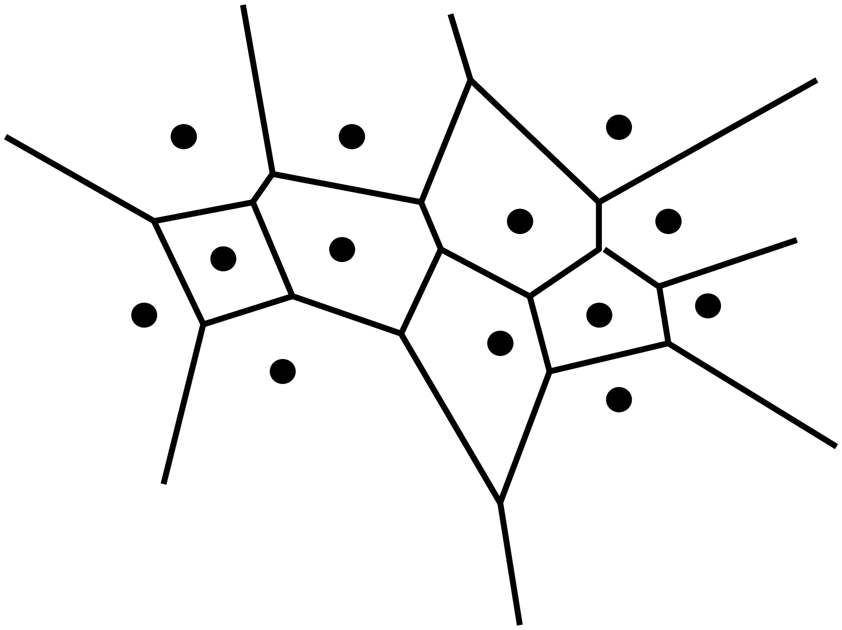
A divide and conquer method for constructing $V(C)$ is described by Imai, Iri and Murota, 1985.

# Use of Voronoi Diagram

## Search for nearest neighbour

Input: A fixed (static) set *P* of *n* points in the plane, and a query point *p*

Output: Nearest neighbour of *p* in *P*



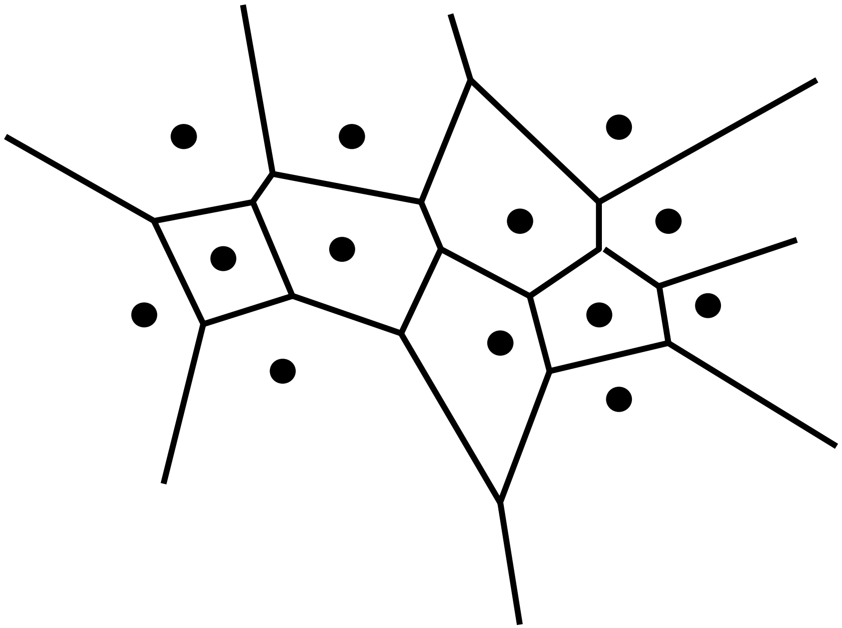
## Solution

- Construct the Voronoi diagram for *P* in time $O(n \log n)$
- Solve the point location problem in O(log n) time.

# Use of Voronoi Diagram (contd.)

Closest pair of points:
Inspect all the edges list of *Vor(P)* and determine the minimally separated pair

Largest empty circle:
Each Voronoi vertex represents the center of a maximal empty circle. Find one having maximum radius.



# Base station placement problem

Problem: Place *k* base stations of same power in a convex region

Method:

Initial Configuration:

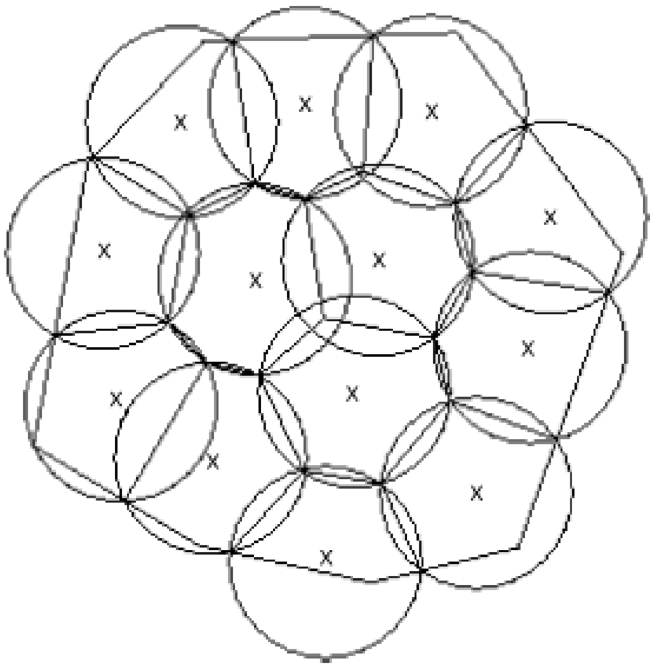
Randomly distribute k points inside the region

Iterative Step:

1. Compute the Voronoi diagram
2. Compute the minimum enclosing circle of each Voronoi polygon
3. Move each point to the center of its corresponding circle.

Termination Condition:

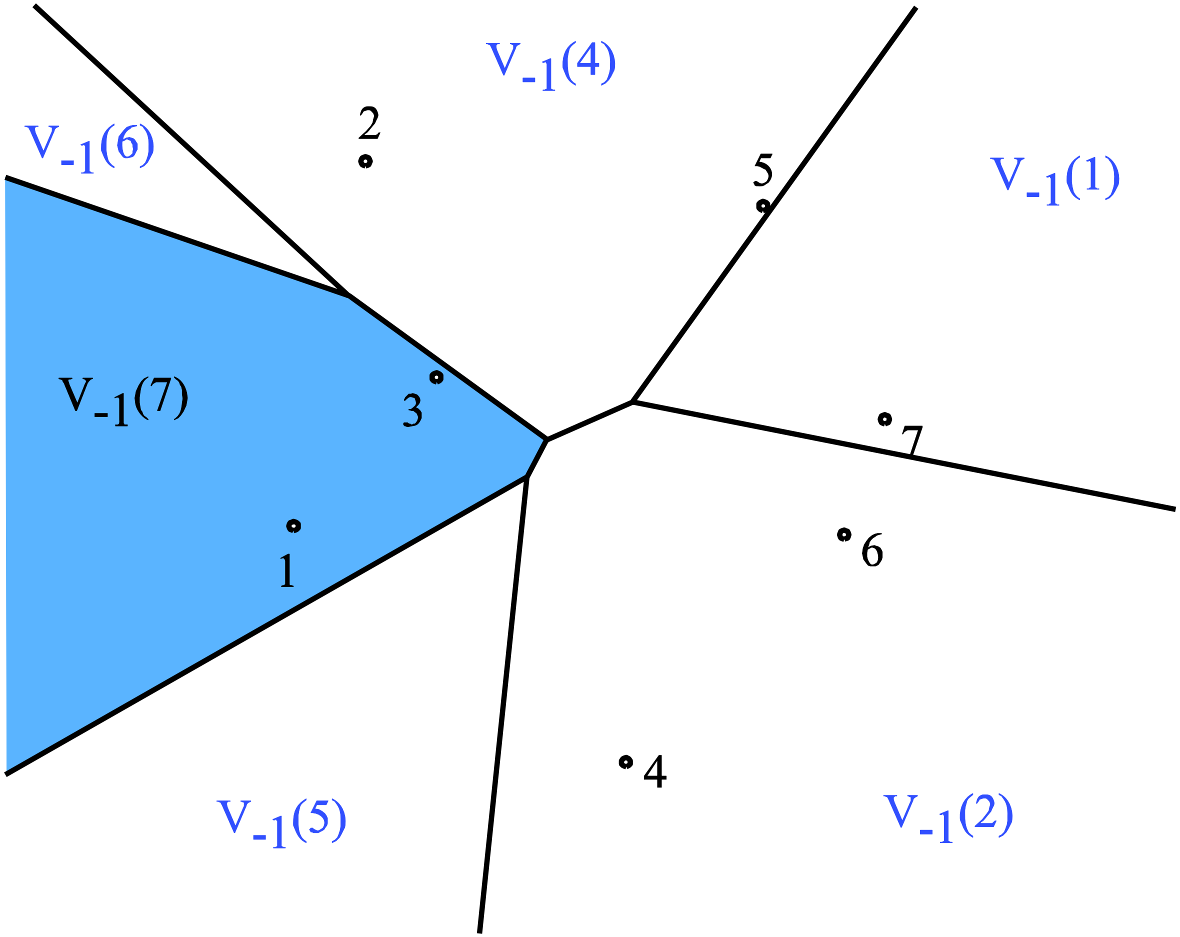
The radius of each circle is almost same.



# Furthest Point Voronoi Diagram

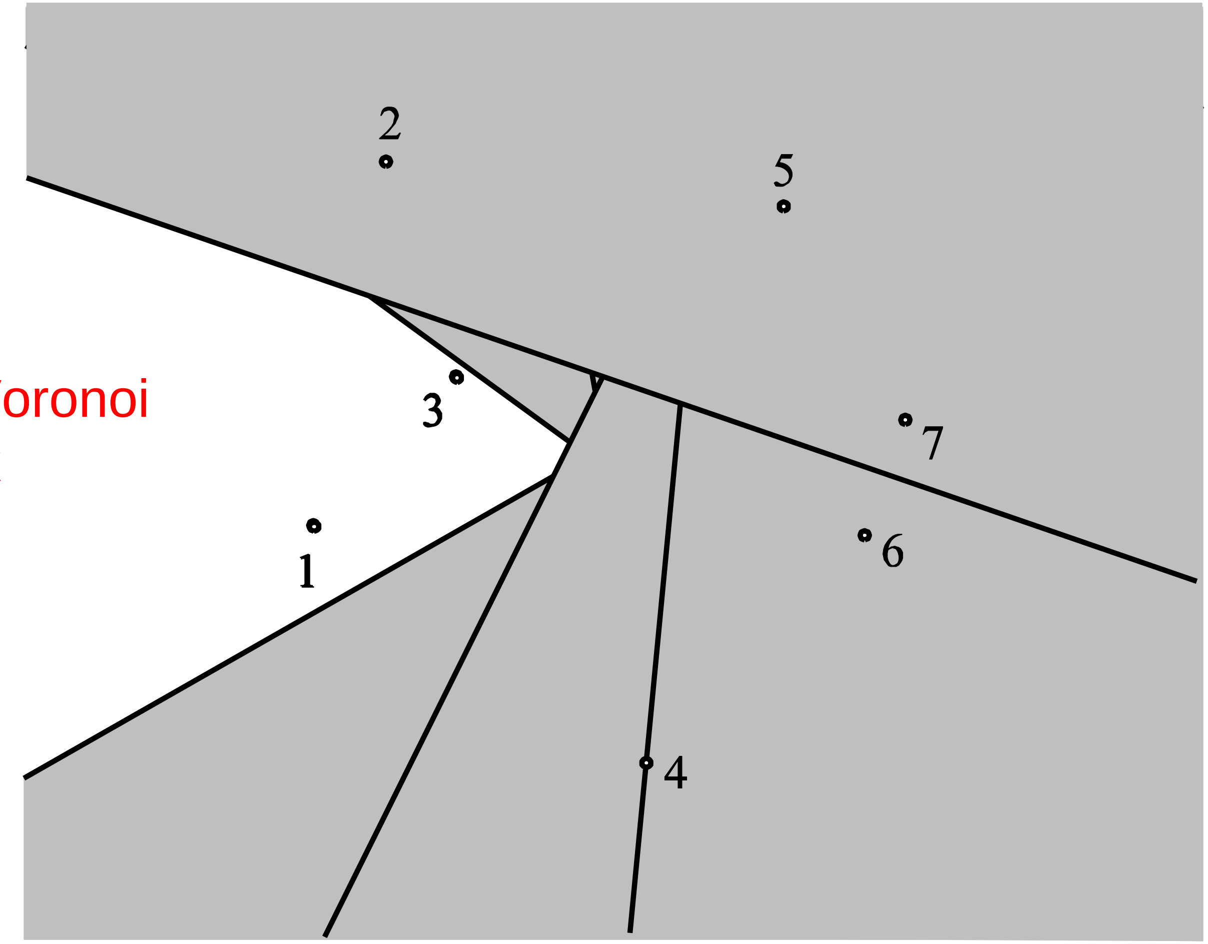
$V_{-1}(p_i)$ : the set of point of the plane farther from $p_i$ than from any other site

Vor-1(P) : the partition of the plane formed by the farthest point Voronoi regions, their edges, and vertices



# Furthest Point Voronoi Region

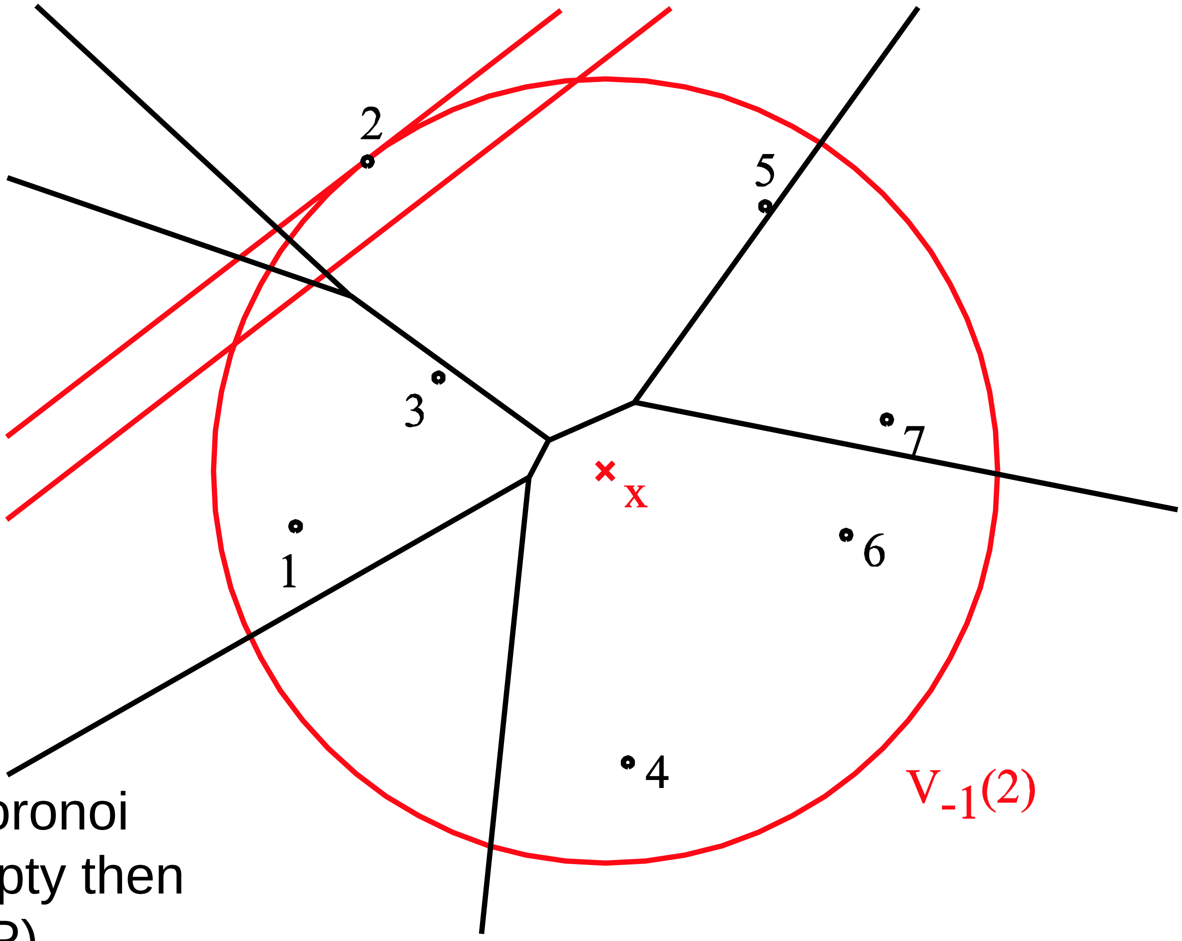
Construction of $V_{-1}(7)$




Property

The farthest point Voronoi regions are convex

# Furthest Point Voronoi Region




Property

If the farthest point Voronoi region of $p_i$ is non empty then $p_i$ is a vertex of conv(P)

# Furthest Point Voronoi Region

Property

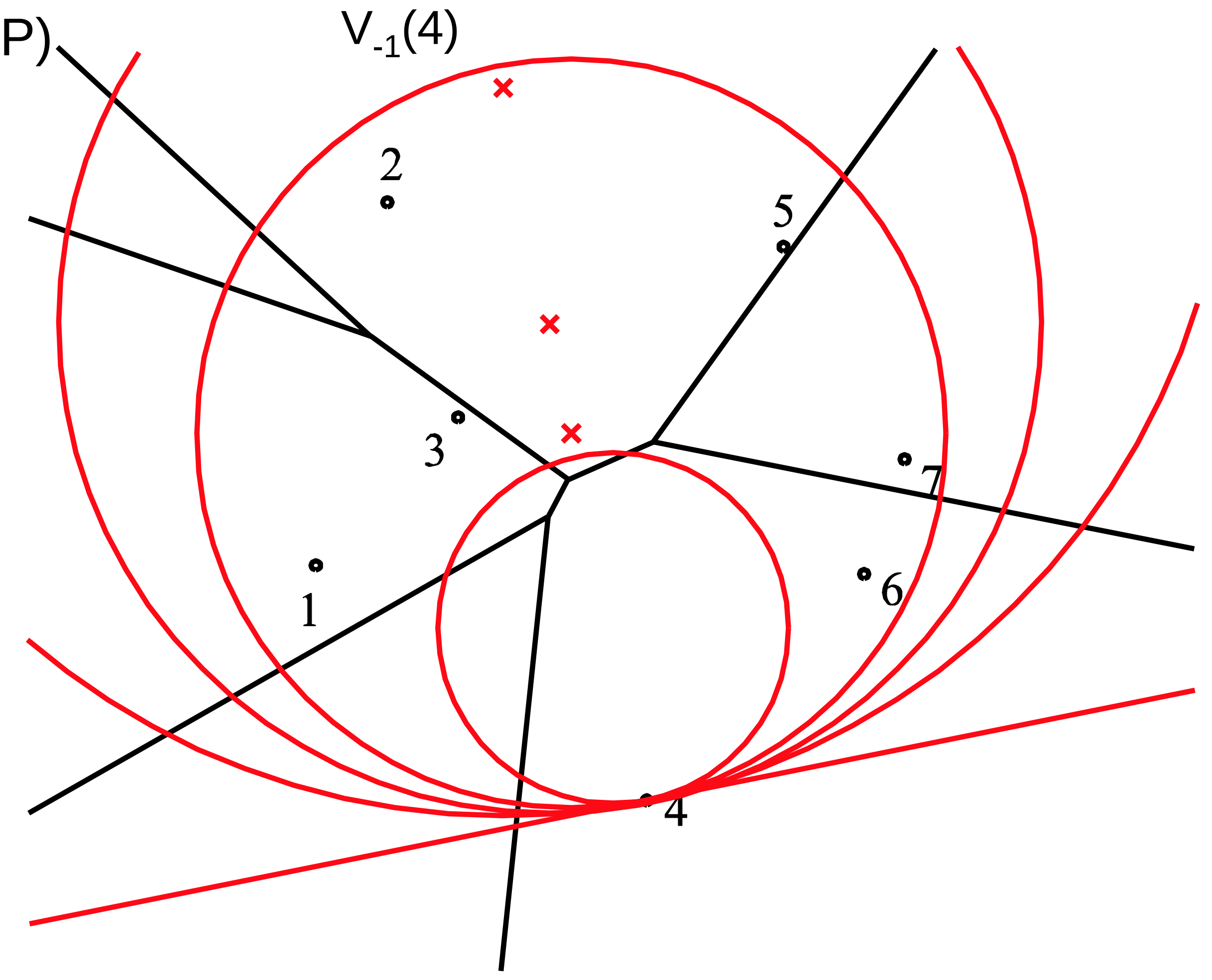
- If $p_i$ is a vertex of conv(P) then the farthest point Voronoi region of $p_i$ is non empty

Property

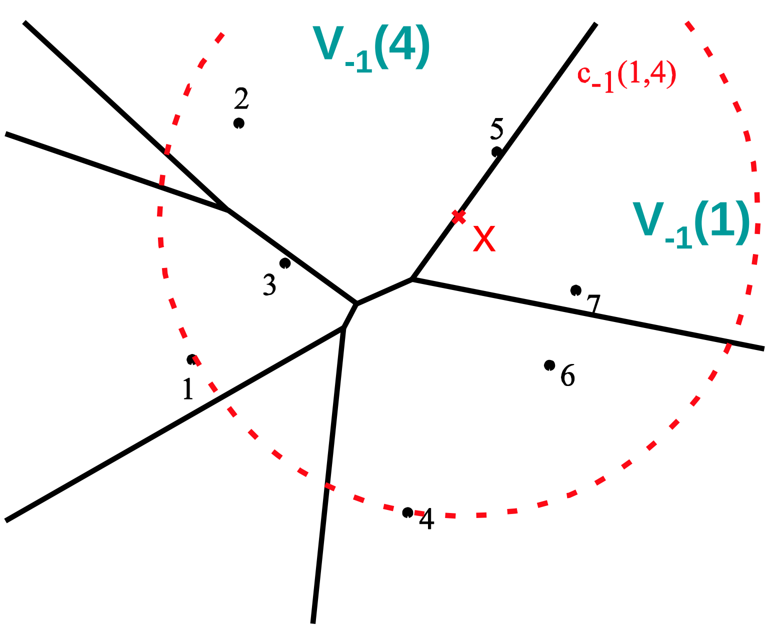
- The farthest point Voronoi regions are unbounded

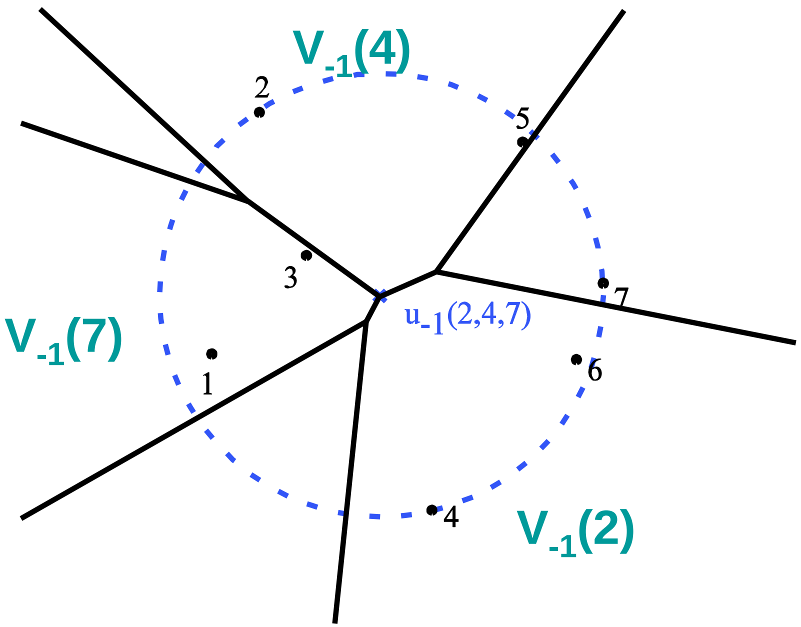
Corollary

- The farthest point Voronoi edges and vertices form a tree



# Farthest point Voronoi edges and vertices

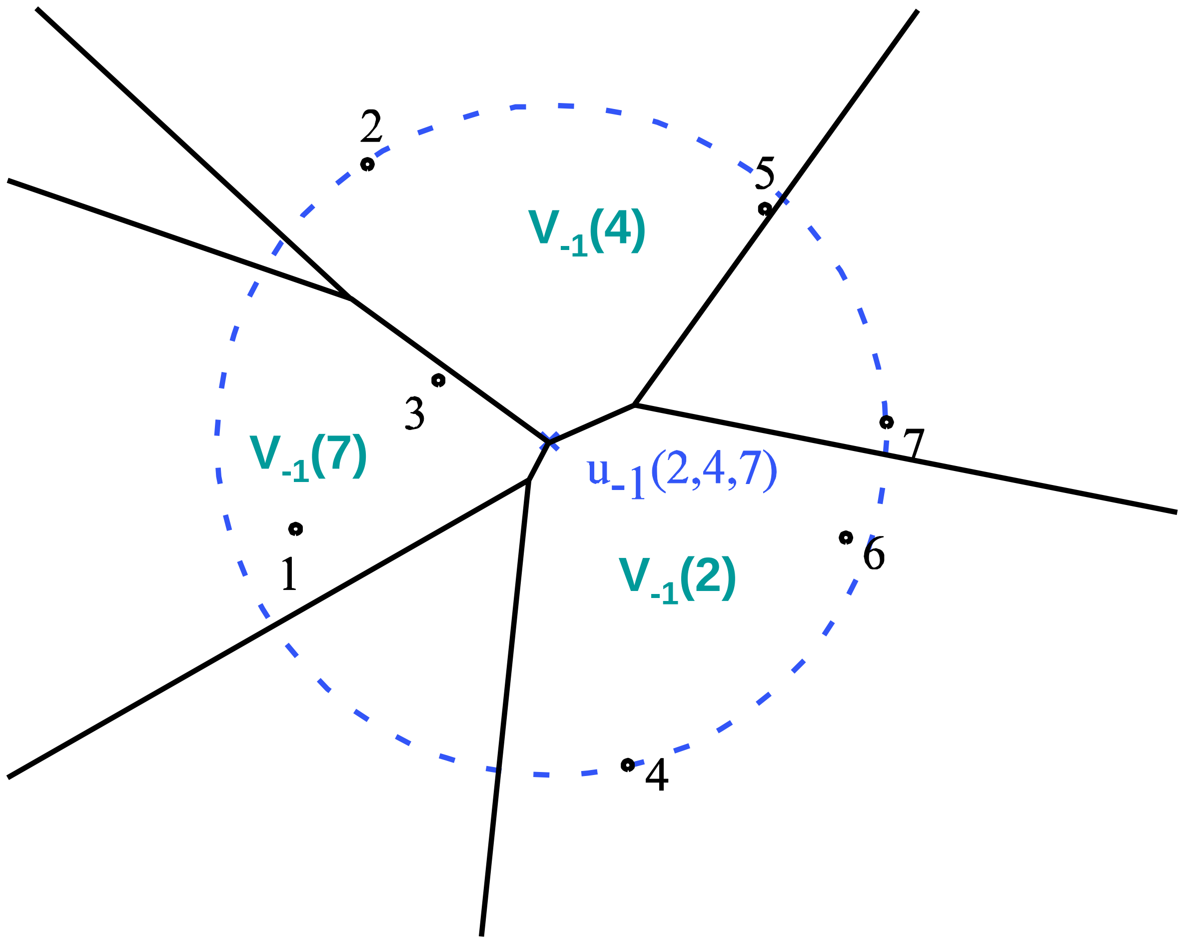



edge : set of points equidistant from 2 sites and closer to all the other sites

vertex : point equidistant from at least 3 sites and closer to all the other sites

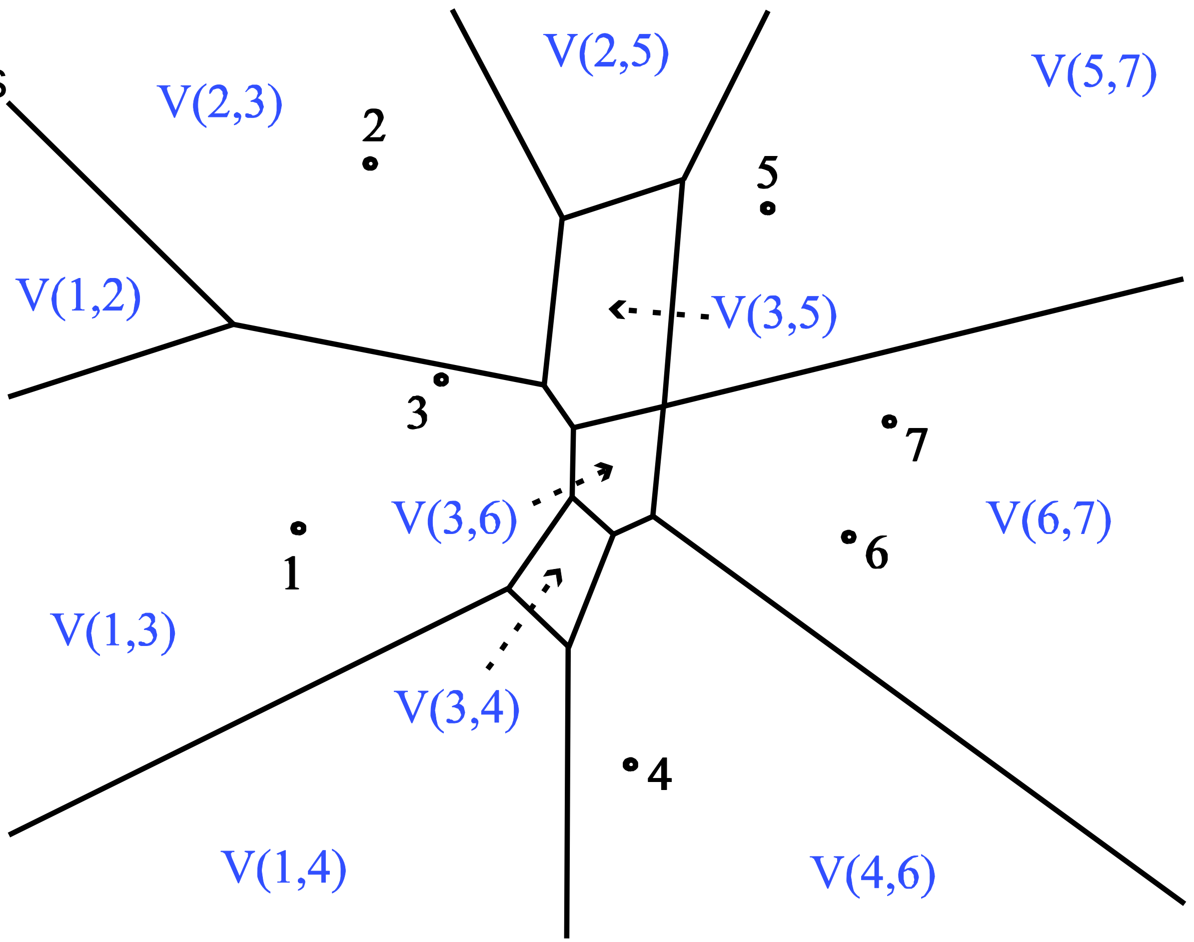
# Application: Smallest enclosing circle



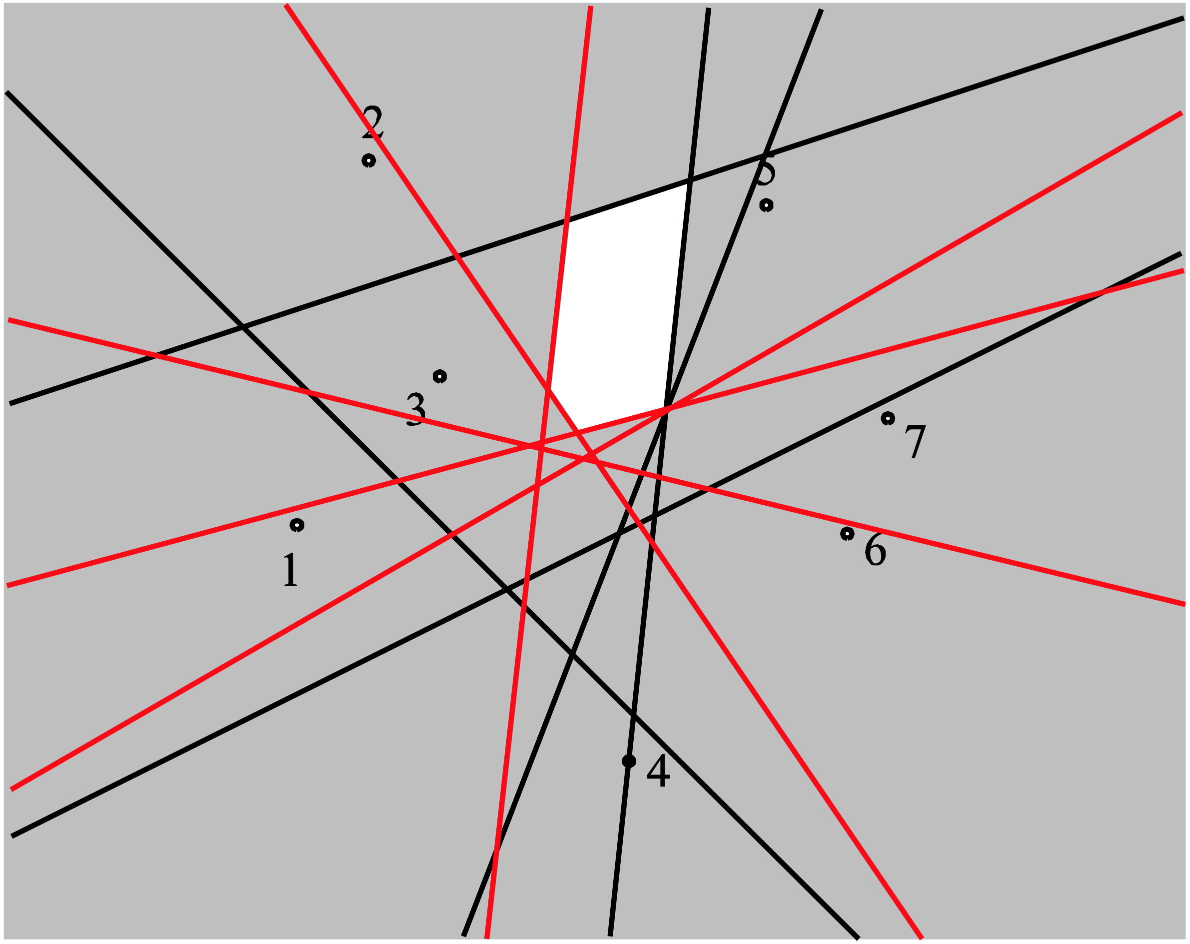
# Order-2 Voronoi diagram

$V(p_i,p_j)$ : the set of points of the plane closer to each of $p_i$ and $p_j$ than to any other site

Property
The order-2 Voronoi regions are convex



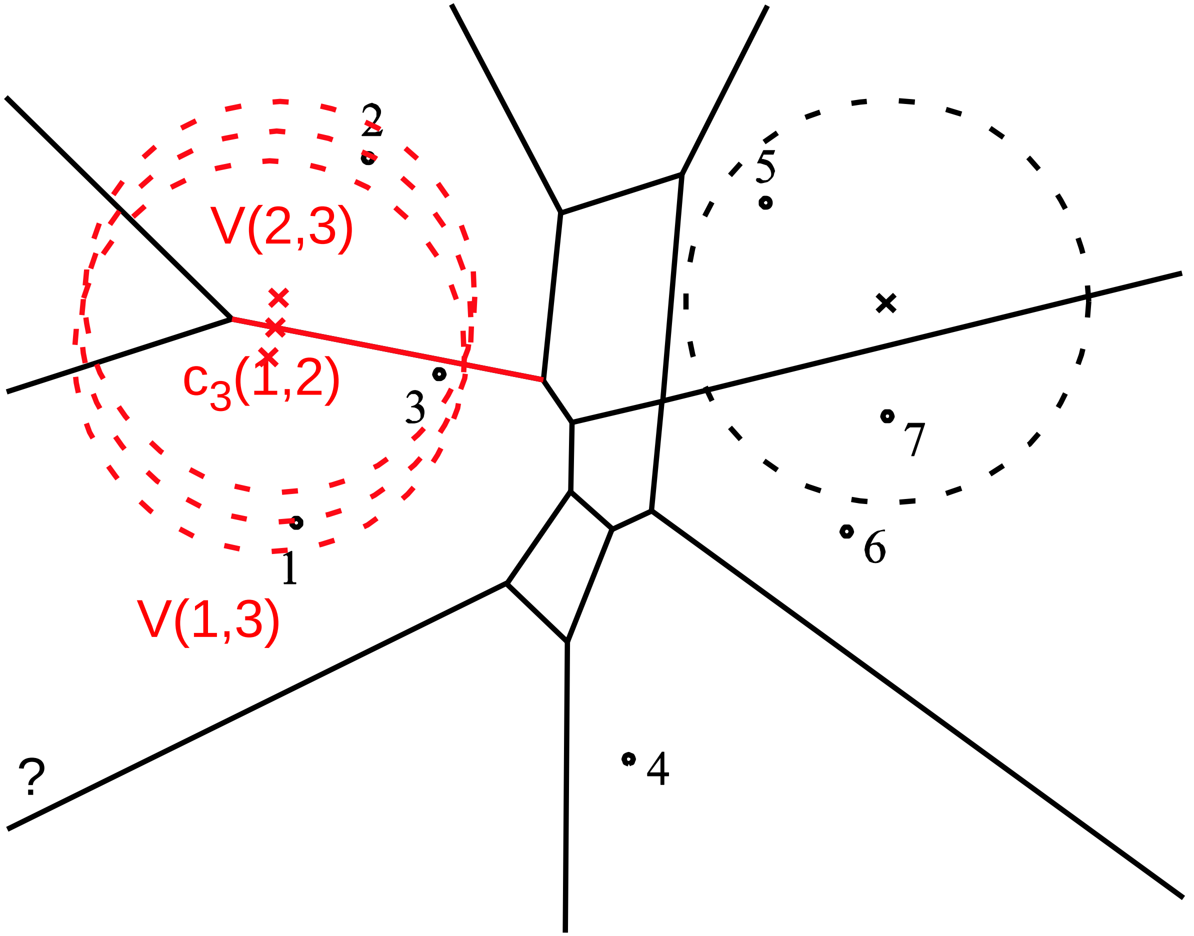
# Construction of V(3,5)



# Order-2 Voronoi edges

edge : set of centers of circles passing through 2 sites s and t and containing 1 site p

=> $c_p(s,t)$




Question

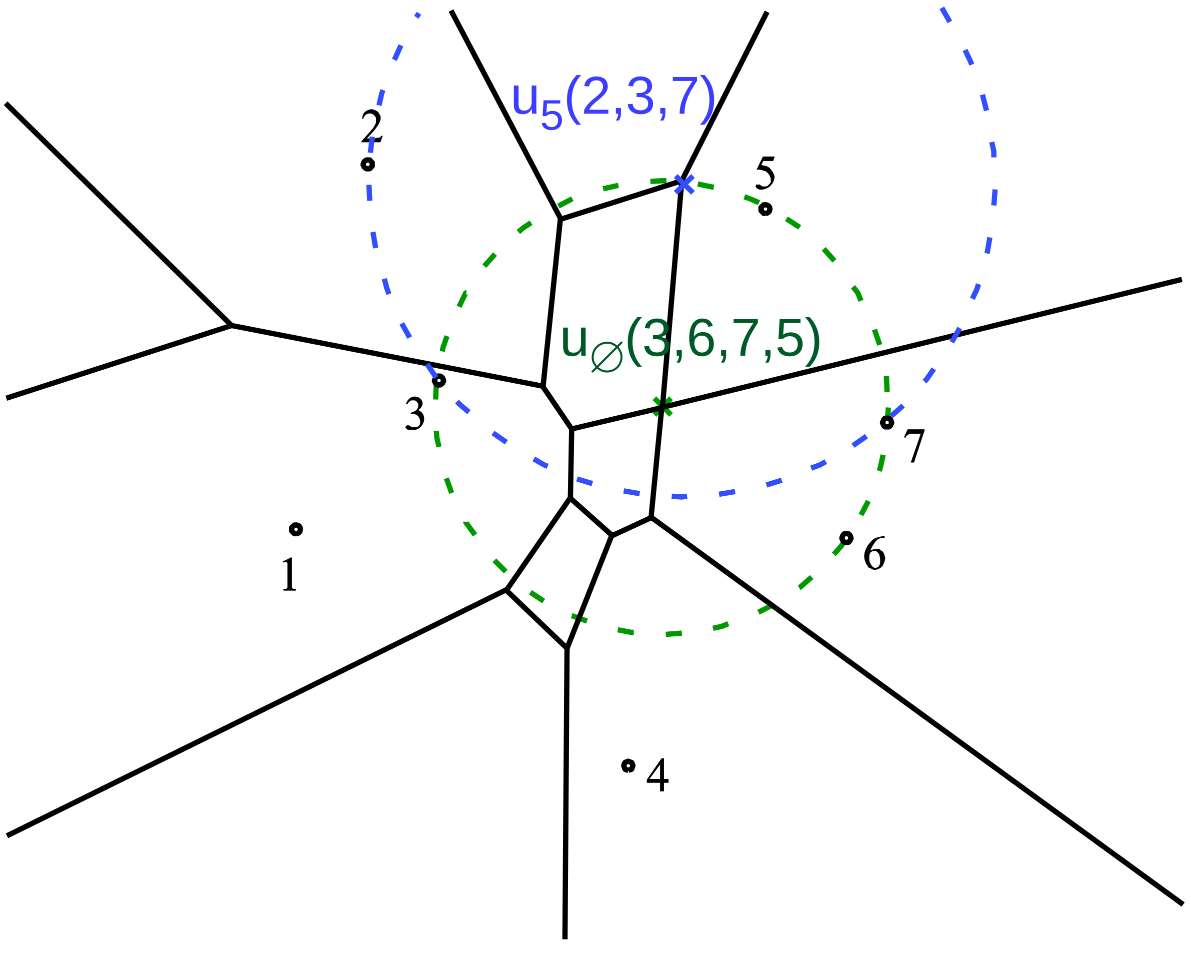
Which are the regions on both sides of $c_p(s,t)$ ?

=> V(p,s) and V(p,t)

# Order-2 Voronoi vertices

vertex : center of a circle passing through at least 3 sites and containing either 1 or 0 site

=> $u_p(Q)$ or $u_\varnothing(Q)$



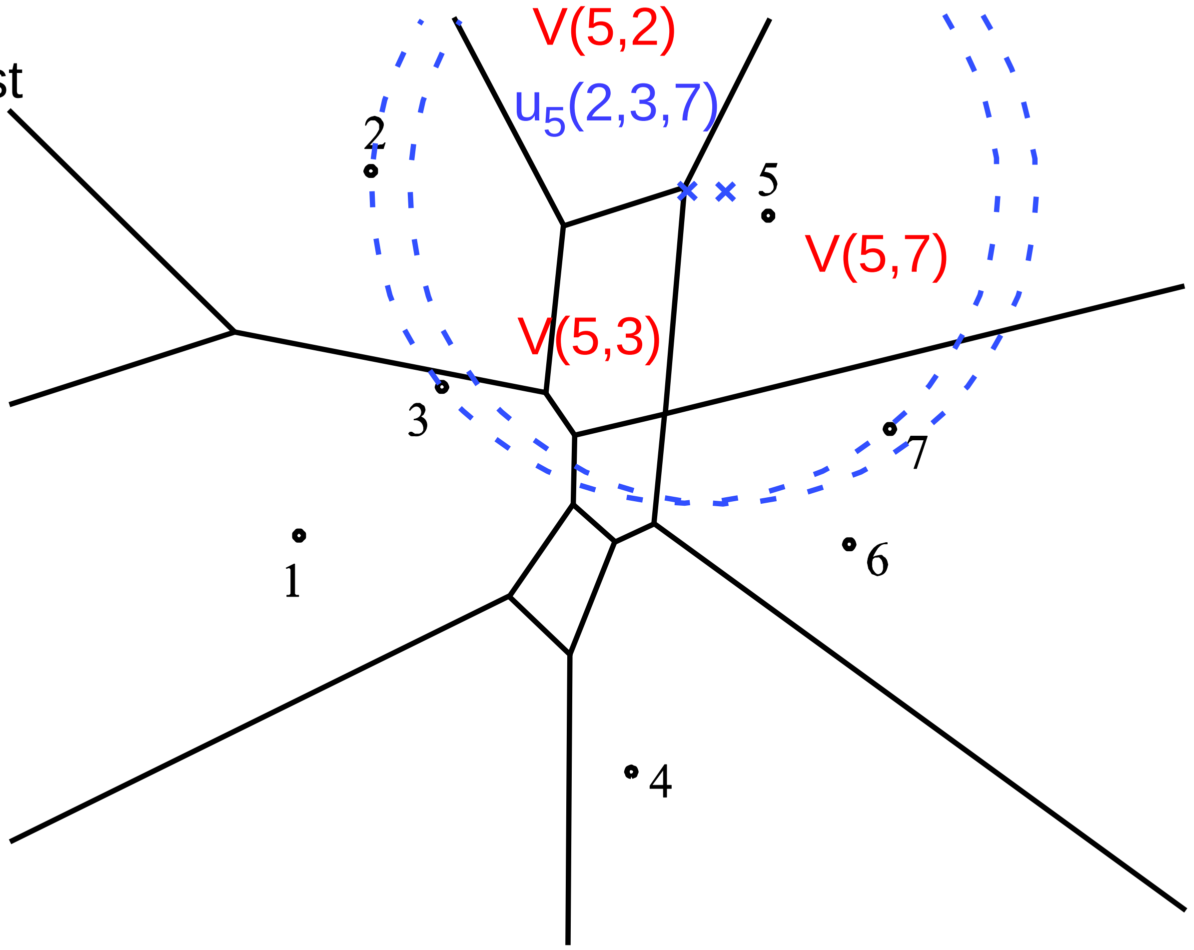
# Order-2 Voronoi vertices

vertex : center of a circle
passing through at least
3 sites and containing
either 1 or 0 site

=> $u_p(Q)$ or $u_\emptyset(Q)$

Question
Which are the regions
incident to $u_p(Q)$ ?

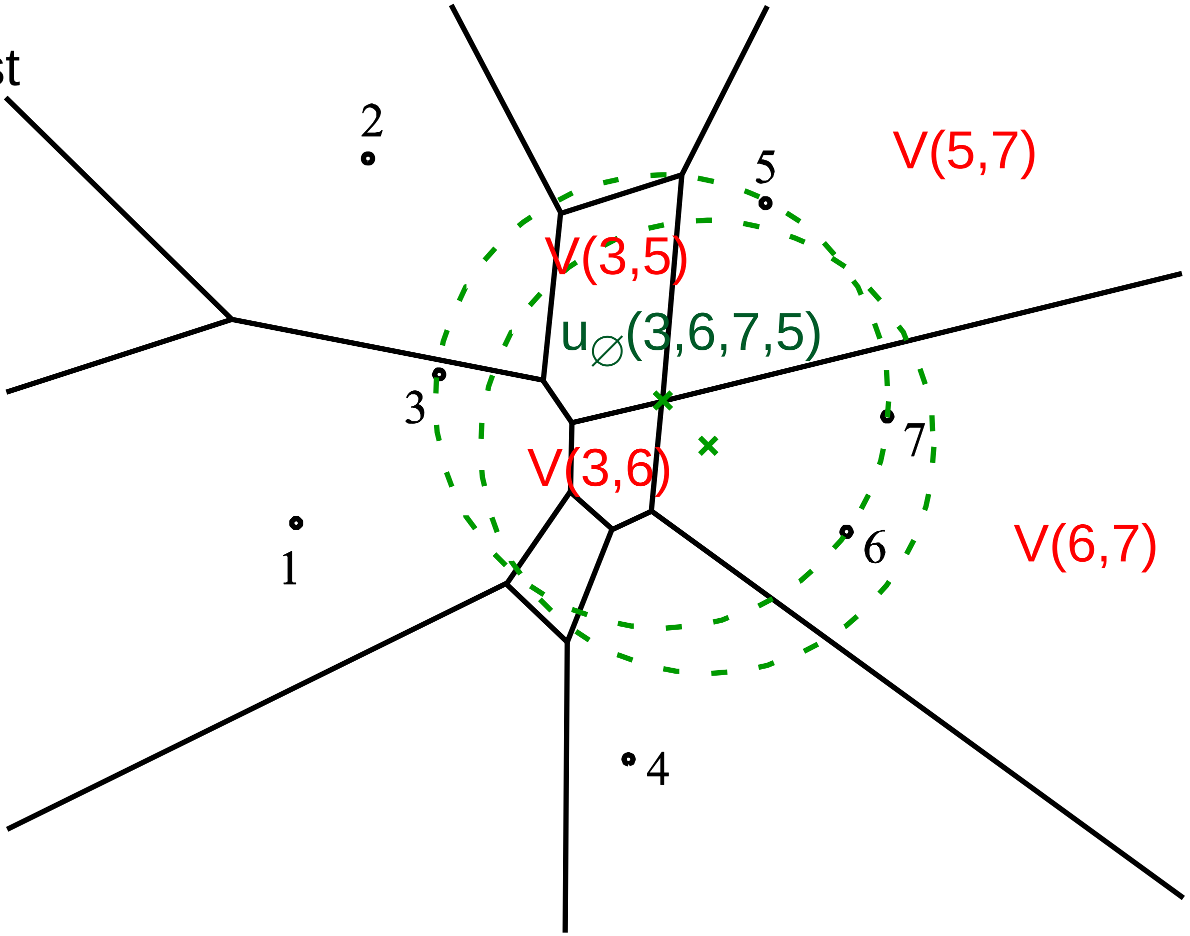
=> V(p,q) with q $\in$ Q



# Order-2 Voronoi vertices

vertex : center of a circle passing through at least 3 sites and containing either 1 or 0 site

=> $u_p(Q)$ or $u_\varnothing(Q)$




Question

Which are the regions incident to $u_p(Q)$ ?

=> V(p,q) with $q \in Q$

Question

Which are the regions incident to $u_\varnothing(Q)$ ?

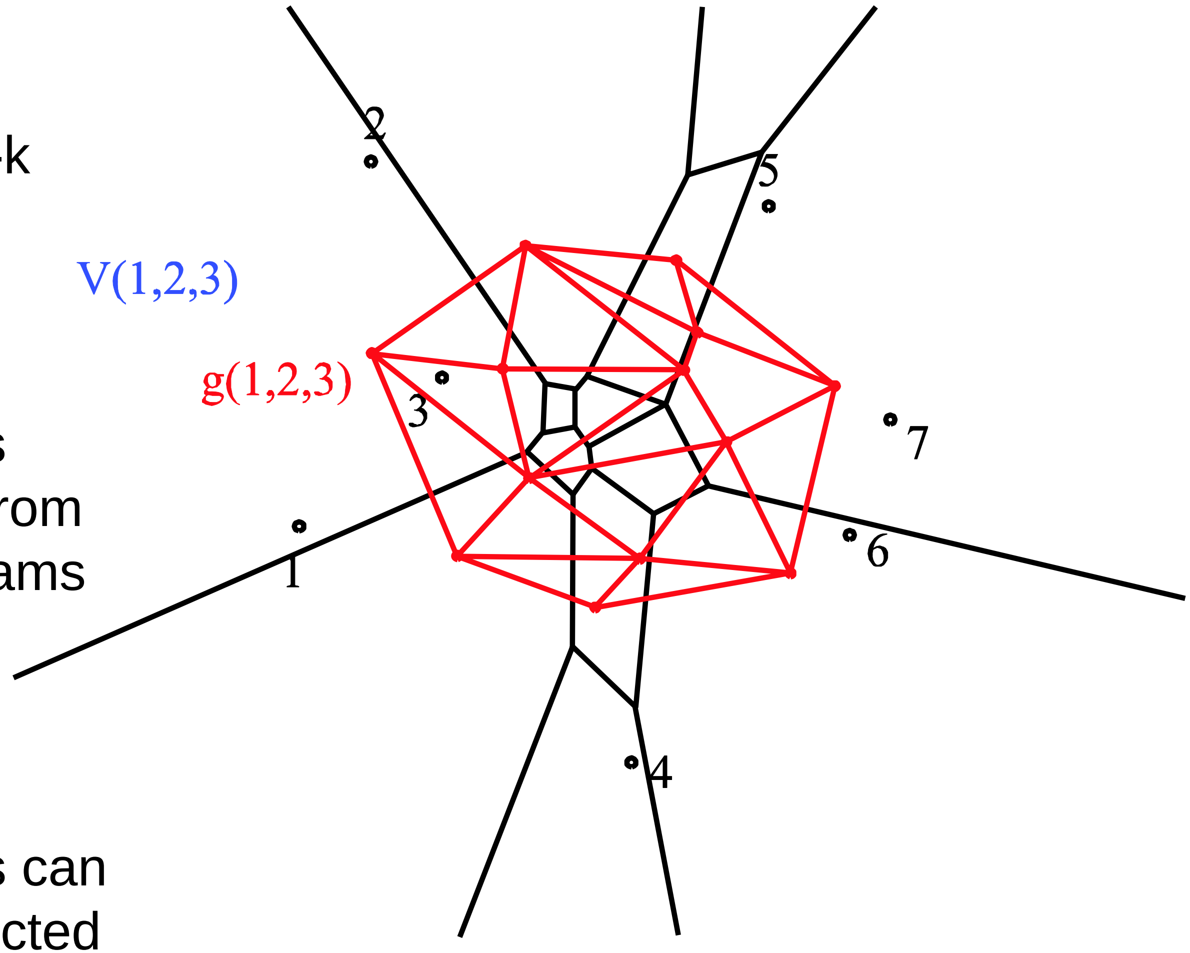
=> V(q,q') with q and q' consecutive on the circle circumscribed to Q

# Order-k Voronoi Diagram

Theorem
The size of the order-k diagrams is O(k(n-k))

Theorem
The order-k diagrams can be constructed from the order-(k-1) diagrams in O(k(n-k)) time

Corollary
The order-k diagrams can be iteratively constructed in $O(n \log n + k^2(n-k))$ time



# Voronoi diagram of weighted points

$S \rightarrow$ Set of points in 2D

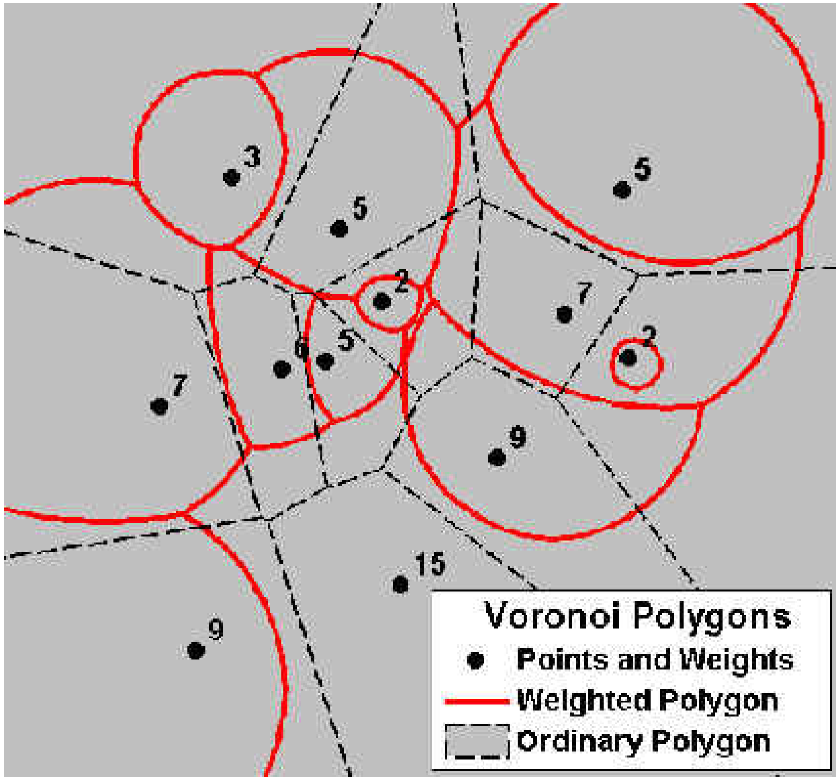
$w(p) \rightarrow$ weight attached with the point $p \in S$

$d_w(x,p) = d_e(x,p)/w(p) \rightarrow$ weighted distance of a point x from $p \in S$

Weighted Voronoi diagram
WVD($S$) $\rightarrow$ the subdivision of the plane such that

$region(p) = \{x \;\; d_w(x,p) \leq d_w(x,q) \; \forall \; q \in S$

If a point $x$ falls in *region(p)*, then $p$ is the weighted nearest neighbor of $x$.



# Voronoi diagram of weighted points

$S = \{p, q\}$ be two weighted points in 2D with $w(p) < w(q)$.

Then *dom(p,q)* = the region of influence of $p$ is the closed disk with
center at $(w^2(p)\, p - w^2(q)\, q)/(w^2(p) - w^2(q))$,
and radius $(w(p)\, w(q)\, d_e(p,q))/(w^2(p) - w^2(q))$

*dom(q,p)* = the region of influence of q is the closed complement of this disk.

For a set $S$ of more than 2 points $region(p) = \bigcap_{q \in S \setminus \{p\}} dom(p,q)$

Observations:

Let *p, q*, r be three weighted points. Then there are at most two points common to *sep(p,q), sep(q,r)* and *sep(p,r);*

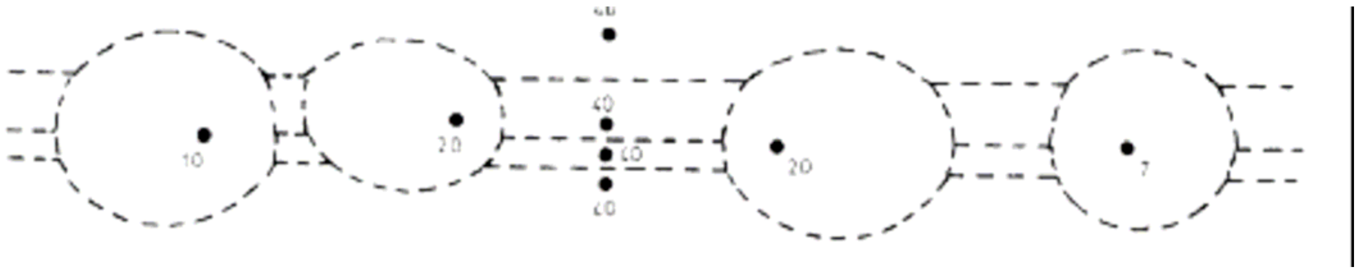
A point common to two of them is common to all of them.

region(p) may not always be connected.

region(p) may be empty for some point p.

# Weighted Voronoi diagram: Combinatorial Complexity

Let *S* denote the set of *n* weighted points in the plane. Then *WVD(S)* contains $\Omega(n^2)$ faces, edges and vertices



Let S be a set of n weighted points in the plane. Then a region may be bounded by O(n) edges.

Algorithm for constructing weighted Voronoi diagram:

See Aurenhammer and Edelsbrunner, Pattern Recognition, 1985

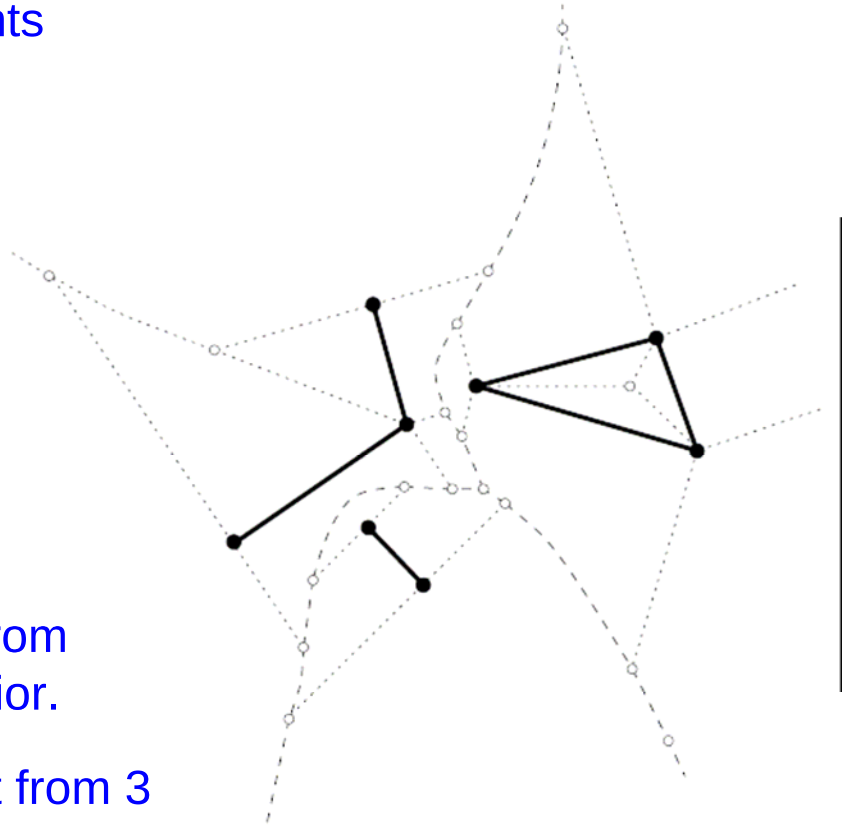
Application (in mobile communication):

Power of one base station is more than that of others. Now given the position of a mobile terminal where from it will get the service.

# Voronoi diagram for line segments

Input: A set of non-intersecting line segments

Output: Voronoi partition of the region



Voronoi edges: These are formed with line segments and/or parabolic arcs.

Straight line edges are part of either the perpendicular bisector of two segment end-points or the angular bisector or two segments.

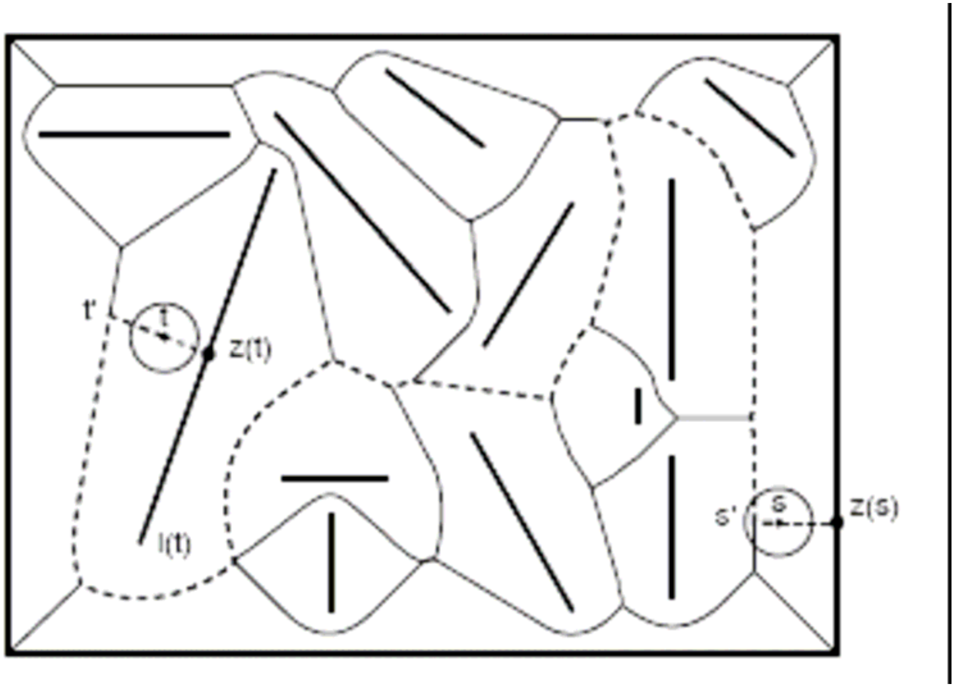
Curve edges consist of points equidistant from a segment end-point and a segment's interior.

Voronoi vertices: Each vertex is equidistant from 3 objects (segment end-points and segment interiors) These are of two types

Type 2: It's two objects are a segment and one of its end-points

Type 3: Its three objects are different.

# Voronoi diagram for line segments




Moving a disk from *s* to *t* in the presence of barriers