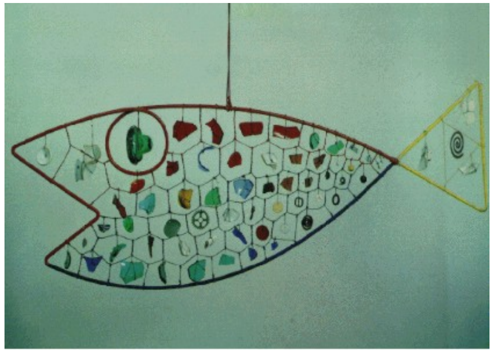
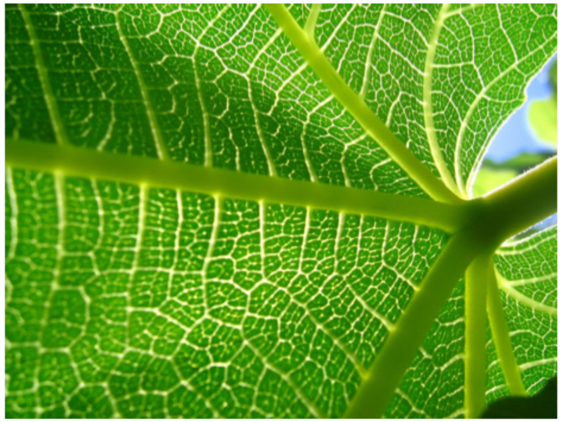
Post Office Problem

Let us assume that people simply get their goods at the nearest site. The trading area for a given site consists of all those points for which that site is closer than any other site !

The induced subdivision is called the Voronoi diagram of the set of sites.

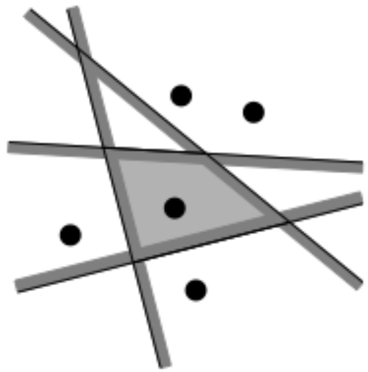
Voronoi Diagrams
A universal structure !





Voronoi Cell

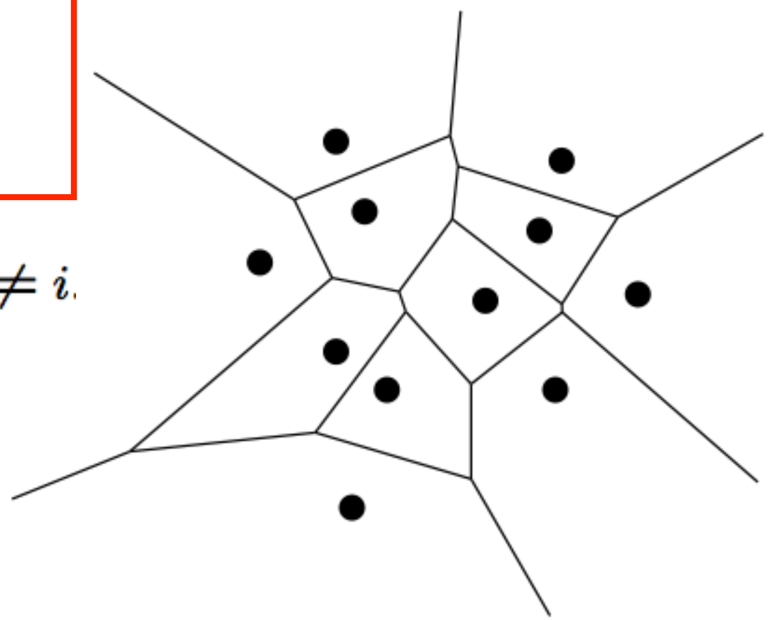
$$\mathcal{V}(\mathbf{p}_i) = \bigcap_{1 \le j \le N, j \ne i} h(\mathbf{p}_i, \mathbf{p}_j)$$



$$\mathrm{dist}(\mathbf{q}, \mathbf{p}_i) < \mathrm{dist}(\mathbf{q}, \mathbf{p}_j) \quad \forall \mathbf{p}_j \in P, \quad j \neq i.$$

Thus, the Voronoi cell is the intersection of $n-1$ half-planes.

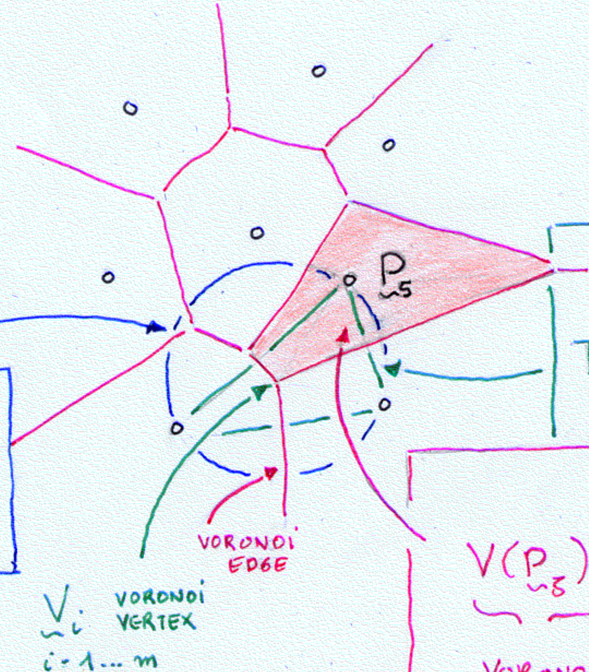
Hence, it is a possibly unbounded open convex polygonal region bounded by at most $n-1$ vertices and at most $n-1$ edges.



VORONOI REGIONS

$$P = \underbrace{\{ \underset{\sim}{p}_1, \ldots\ldots \underset{\sim}{p}_n \}}_{\text{SET OF POINTS}}$$

SITE



$C_i \triangleq C(\underset{\sim}{v}_i)$

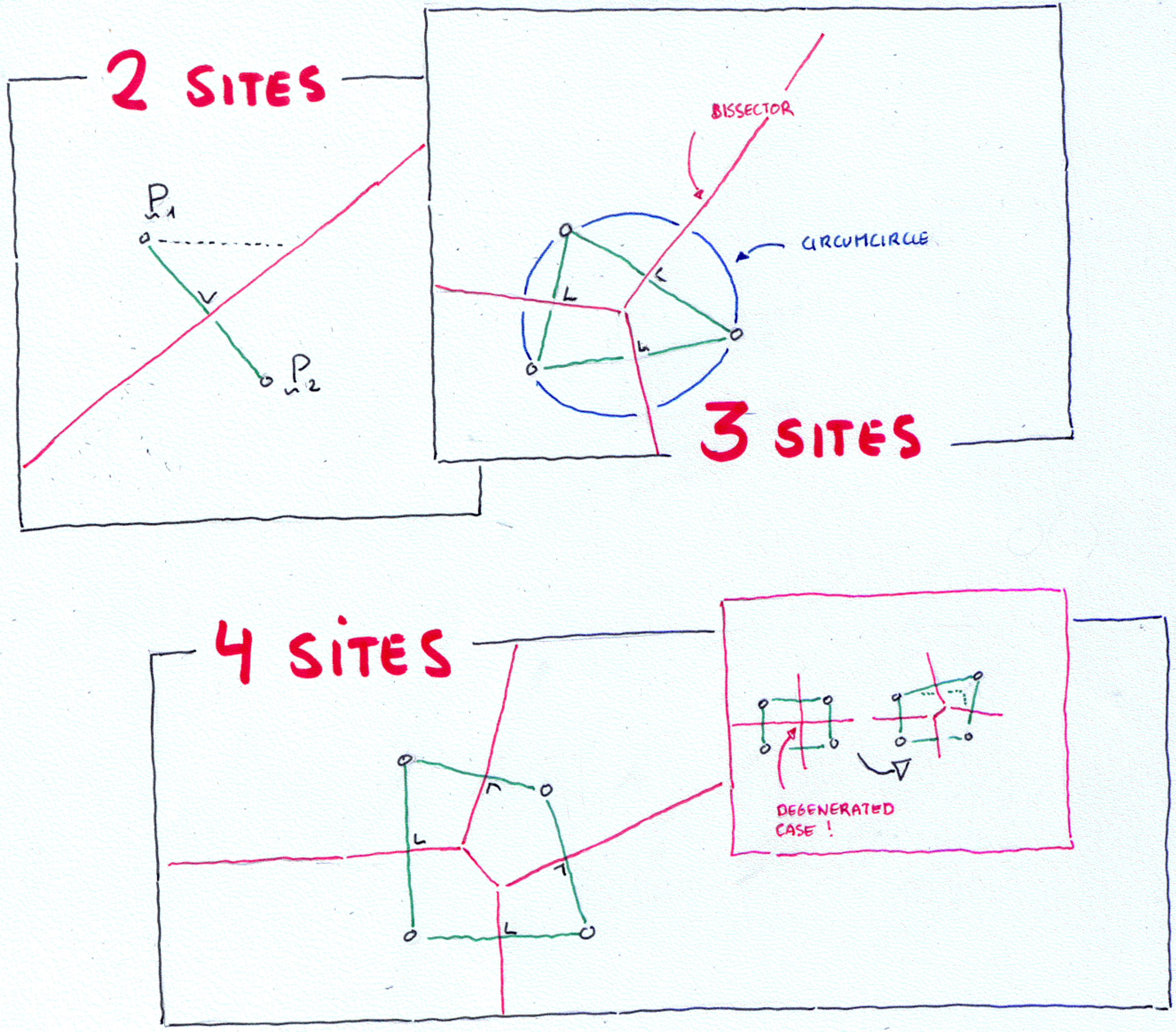
CIRCUMCIRCLE ASSOCIATED TO $\underset{\sim}{v}_i$

$T(\underset{\sim}{v}_i) \triangleq T_i$

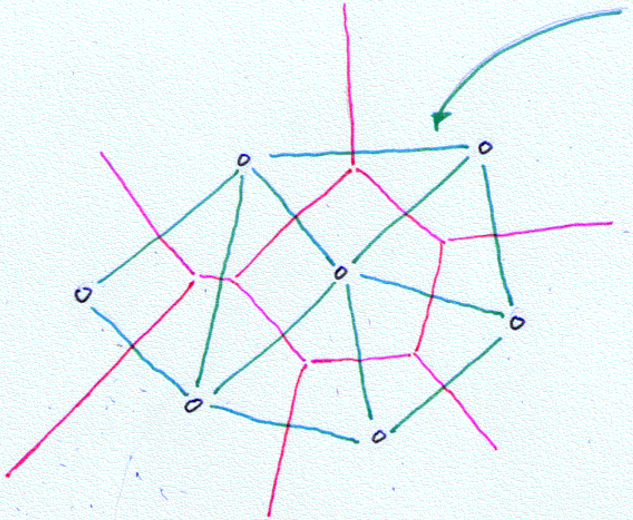
"DELAUNAY" TRIANGLE ASSOCIATED TO $\underset{\sim}{v}_i$

$$\underbrace{V(\underset{\sim}{p}_s)}_{\text{VORONOI CELL}} \triangleq \{ \underset{\sim}{x} \text{ SUCH THAT } |\underset{\sim}{p}_s - \underset{\sim}{x}| \leq |\underset{\sim}{p}_j - \underset{\sim}{x}| \quad \forall j \neq s \}$$

$$\underbrace{G_{\text{VORONOI}}(P)}_{\text{VORONOI DIAGRAM}} \triangleq \underbrace{\bigcup_j \left(V(\underset{\sim}{p}_i) \cap V(\underset{\sim}{p}_j) \right)}_{\text{SET OF POINTS THAT HAVE MORE THAN ONE NEAREST NEIGHBOR!}}$$


2 SITES
P_1
P_2
BISSECTOR
CIRCUMCIRCLE
3 SITES
4 SITES
DEGENERATED CASE !

DUAL GRAPH OF VORONOI = DELAUNAY TRIANGULATION *



EULER'S FORMULA
$3f = 2e$

$3n-6$ DELAUNAY EDGES
$2n-4$ DELAUNAY FACES
INCLUDING THE "OUTSIDE" FACE

n VORONOI REGIONS / SITES

n DELAUNAY NODES

VORONOI EDGES

DELAUNAY EDGES

m VORONOI VERTICES

m DELAUNAY FACES

$O(n)$

$O(n)$

* IN FACT,

IT IS A TRIANGULATION, ONLY IF WE ASSUME, FOR SIMPLICITY, NO 4 COCIRCULAR POINTS

FROM A PRACTICAL POINT OF VIEW, JUST DIVIDE QUADRILATERALS INTO 2 TRIANGLES

DELAUNAY'S PROPERTIES

$G_{DELAUNAY}(P)$ IS A TRIANGULATION IF NO 4 POINTS ARE COCIRCULAR

$\partial G_{DELAUNAY}(P) = \partial H(P)$

CONVEX HULL

INTERIOR OF EACH T_i AND C_i CONTAINS NO SITES

IF P_i IS A NEAREST NEIGHBOR OF P_j THEN (P_i, P_j) IS A DELAUNAY EDGE

$V(P_i)$ IS CONVEX

$V(P_i)$ IS UNBOUNDED ONLY IF $P_i \in H(P)$

INTERIOR OF CONVEX HULL

(P_i, P_j) DELAUNAY EDGE <=> $\exists$ AN EMPTY CIRCLE THROUGH P_i P_j

V_i IS THE CENTER OF C_i

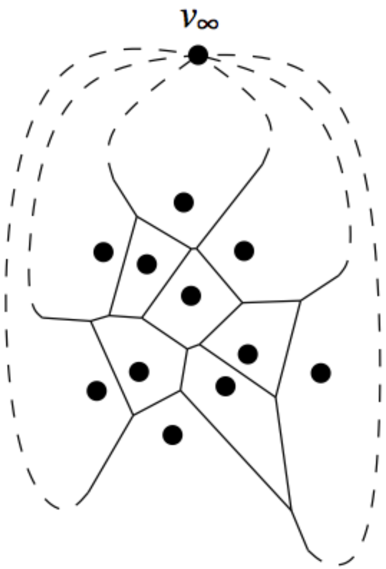
VORONOI'S PROPERTIES

2D DELAUNAY FACES ◁— ▷ VORONOI VERTICES 0D

1D DELAUNAY EDGES ◁— ▷ VORONOI EDGES 1D

0D DELAUNAY VERTICES ◁— ▷ VORONOI CELLS 2D

Complexity



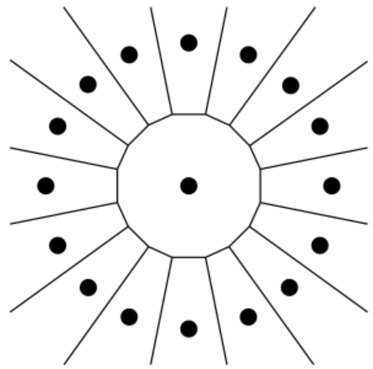
$$(n_v+1)-n_e+n=2.$$

$$2n_e \geqslant 3(n_v+1)$$

Theorem 7.3 *For $n \geqslant 3$, the number of vertices in the Voronoi diagram of a set of n point sites in the plane is at most $2n-5$ and the number of edges is at most $3n-6$.*

Since there are n sites and each Voronoi cell has at most n − 1 vertices and edges,

the complexity is at most quadratic.

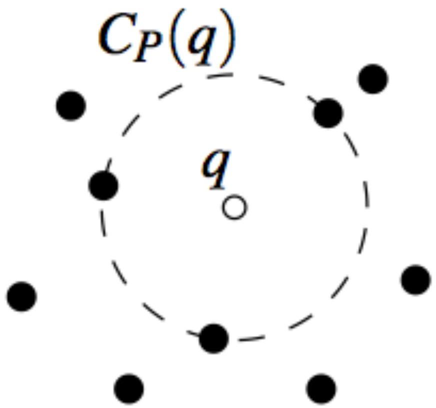


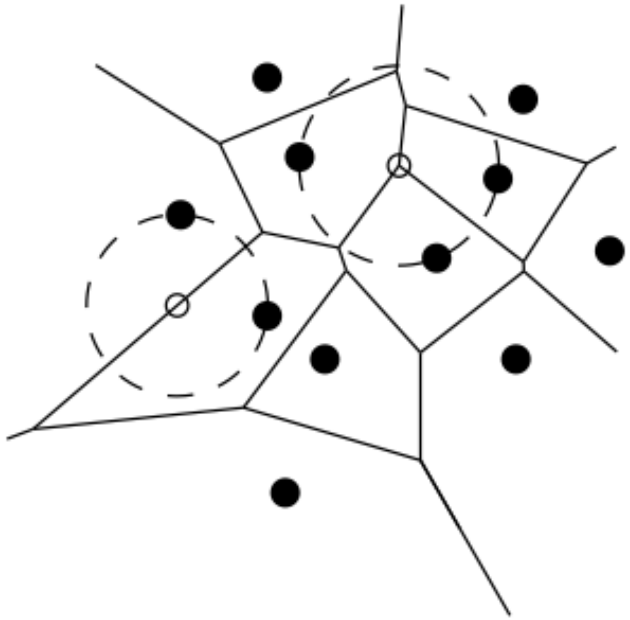
Largest empty circle

Theorem 7.4 *For the Voronoi diagram* $\mathrm{Vor}(P)$ *of a set of points* P *the following holds:*

(i) *A point* q *is a vertex of* $\mathrm{Vor}(P)$ *if and only if its largest empty circle* $C_P(q)$ *contains three or more sites on its boundary.*

(ii) *The bisector between sites* p_i *and* p_j *defines an edge of* $\mathrm{Vor}(P)$ *if and only if there is a point* q *on the bisector such that* $C_P(q)$ *contains both* p_i *and* p_j *on its boundary but no other site.*





INCREMENTAL CONSTRUCTION : WATSON ALGORITHM $\mathcal{O}(n^2)$

$$P = \{ \underbrace{P_{n_1}\ P_{n_2} \cdots\cdots P_{n_k}}_{P^k}\ P_{n_{k+1}} \cdots P_{n_n} \}$$

$$\underbrace{P_{n_1} \cdots P_{n_{k+1}}}_{P^{k+1}}$$

$$\boxed{G^k \triangleq G_{DELAUNAY}(P^k)}$$

$$\boxed{G^{k+1}}$$

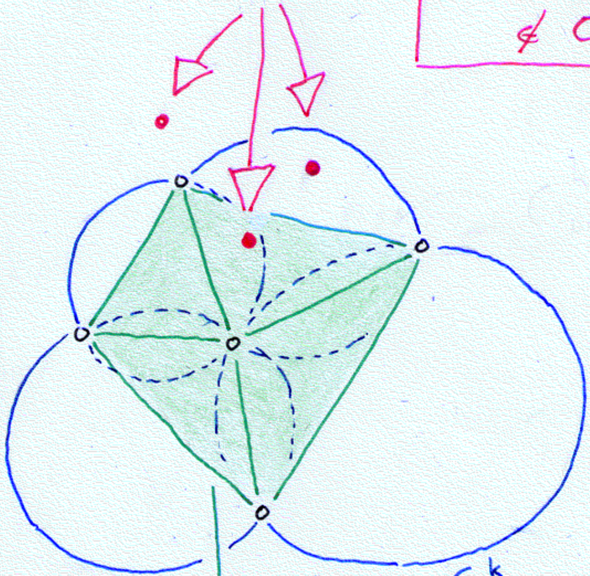
How to obtain G^{k+1} from G^k ?

3 POSSIBLE CASES

$p^{k+1} \in G^k$

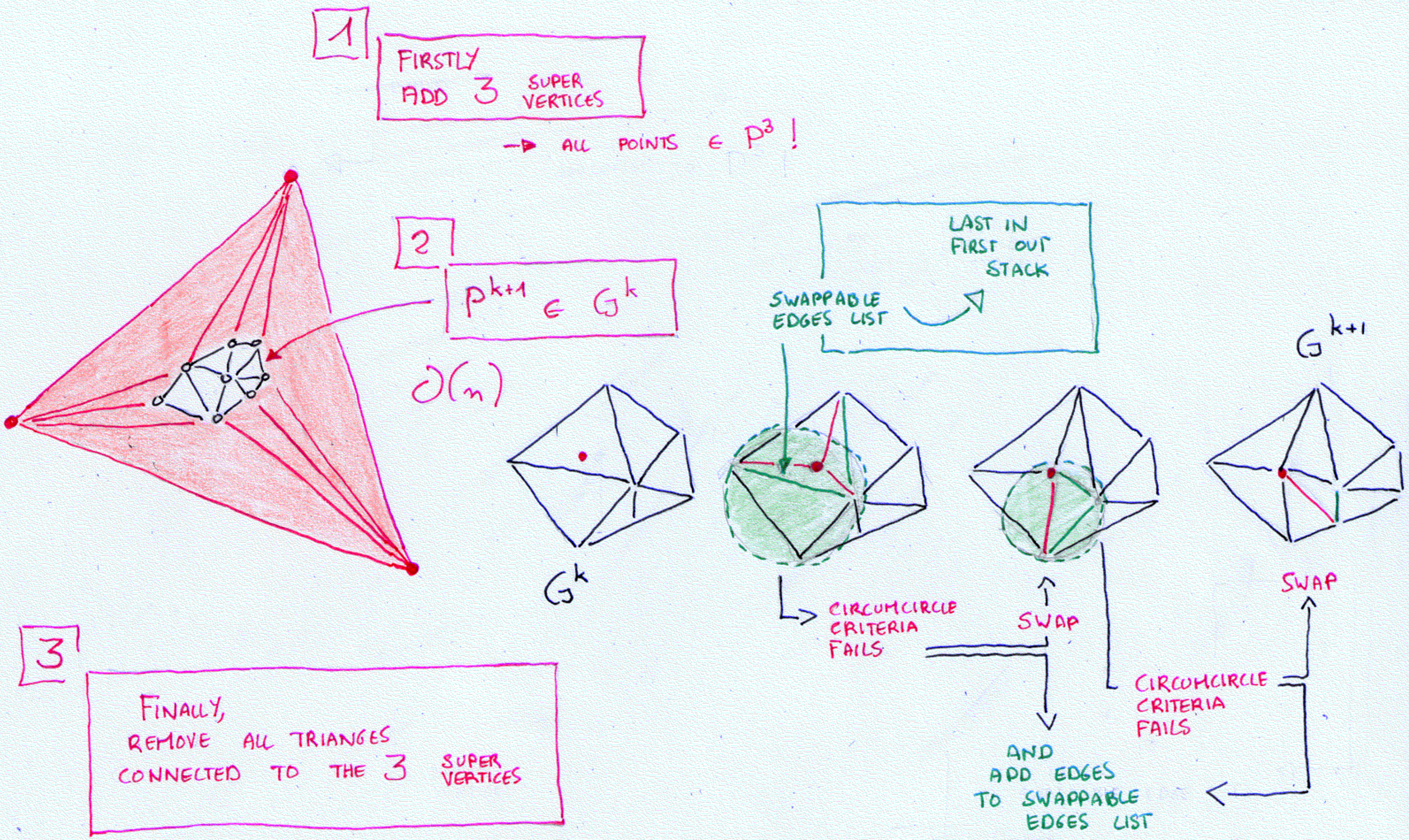
$\notin G^k$ BUT $\in C^k$

$\notin C^k$



$$\overbrace{C(P^k)}^{C^k} = \bigcup_{j=1}^{m} C_j$$

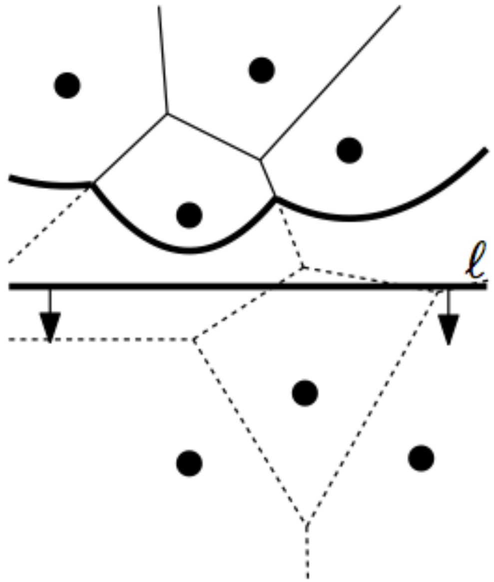
$$\underbrace{G_{DELAUNAY}(P^k)}_{G_k}$$



WATSON PROCEDURE

Fortune Algorithm

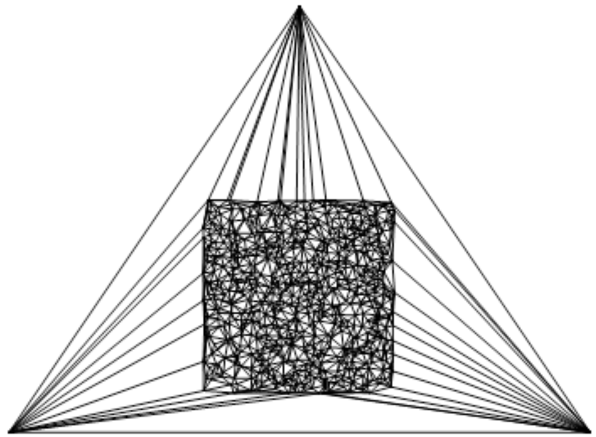
$O(n \log n)$



Let us use a sweep line !

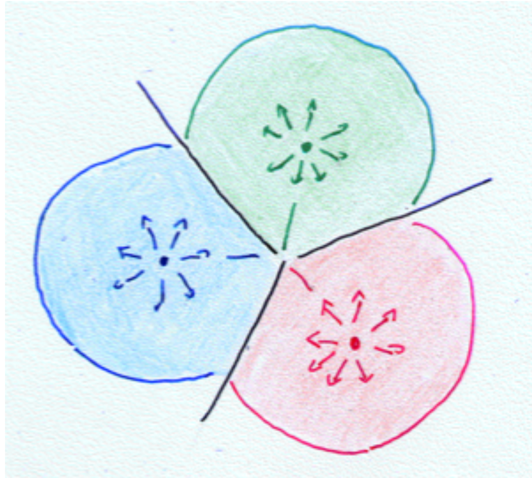
But, it is not so easy, because the part of Vor(P) above depends not only on the sites that lie above the line but also on sites below.

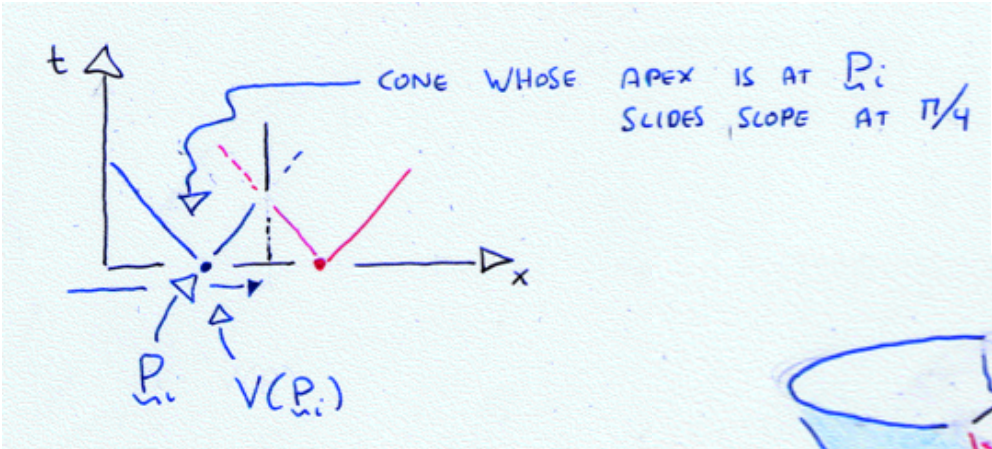
$O(n^2)$

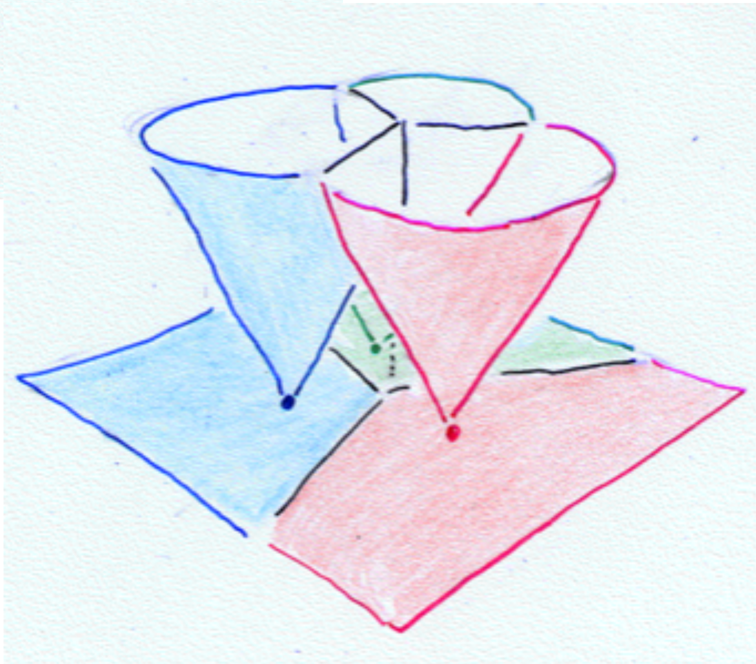


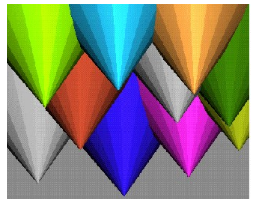
Watson Algorthm

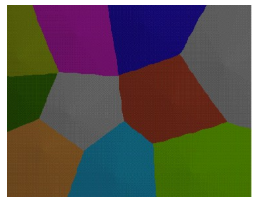
Crystal seeds growth and Voronoi diagrams



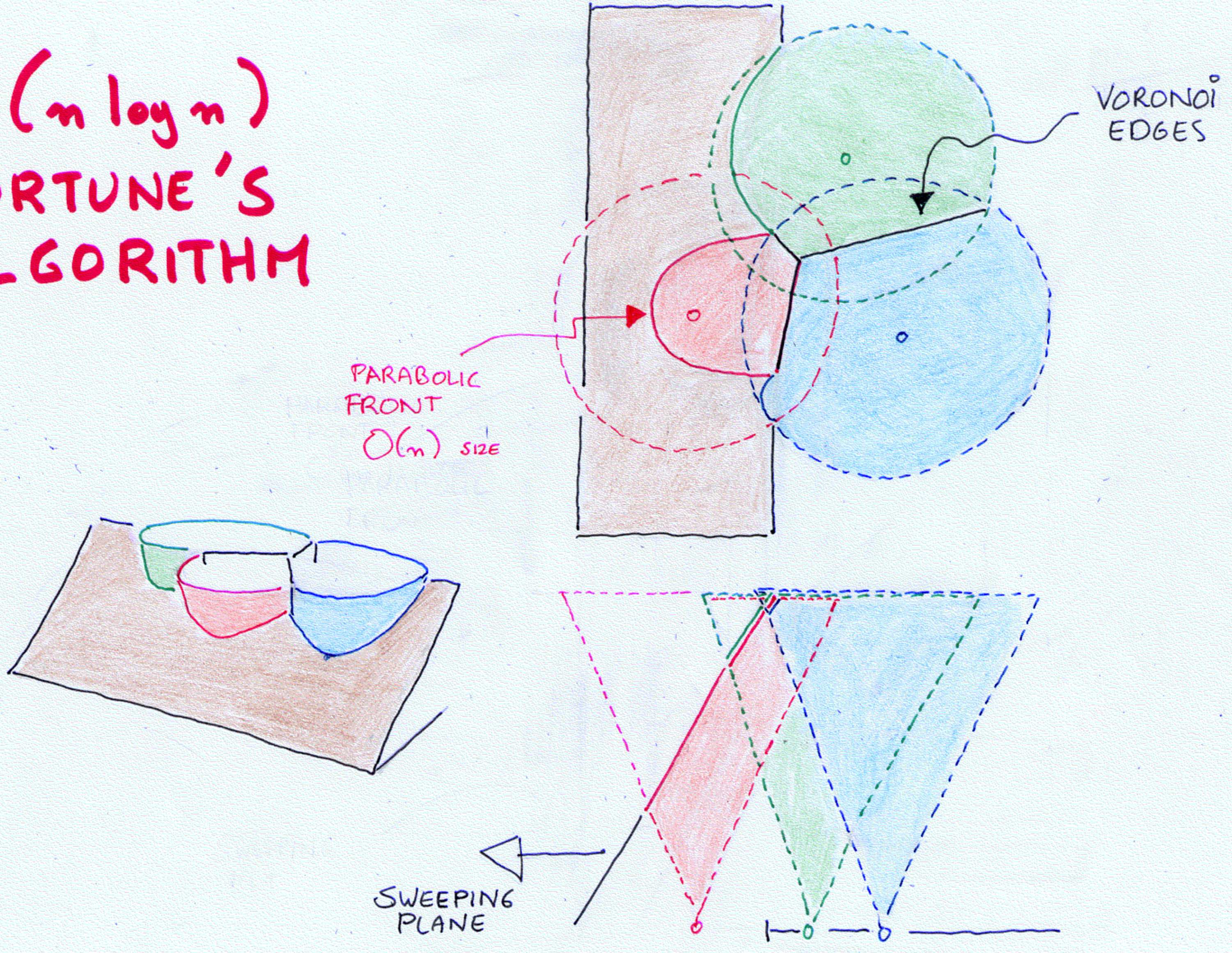




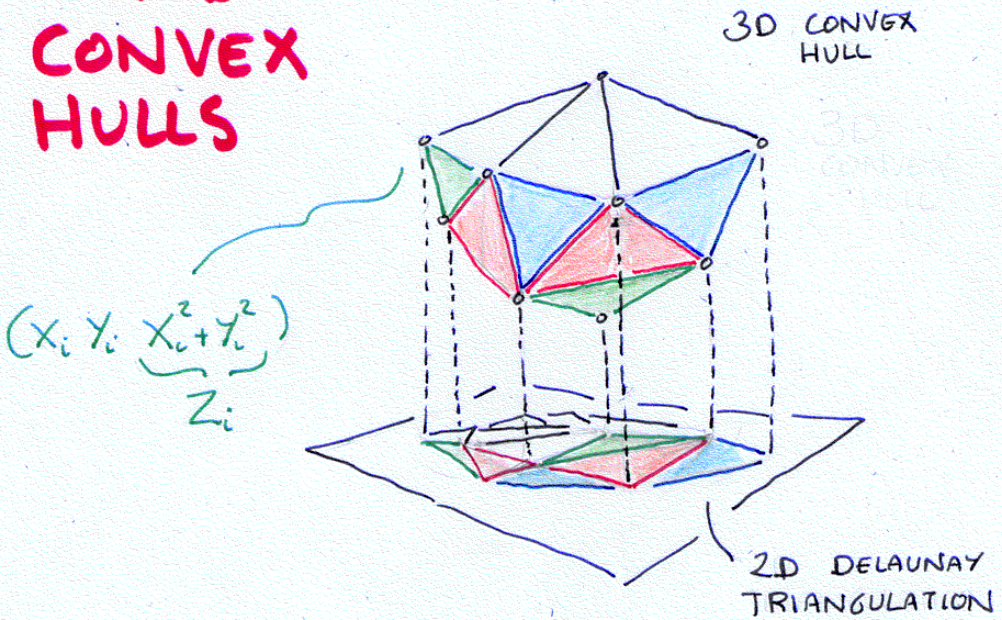


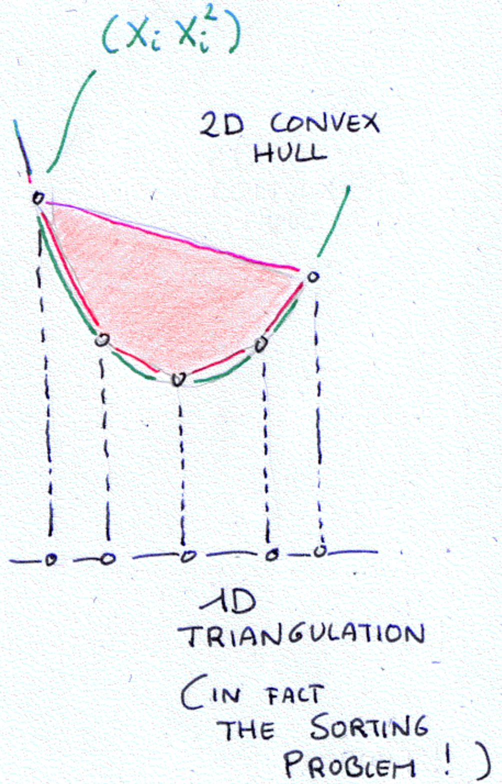


$O(n \log n)$
FORTUNE'S ALGORITHM

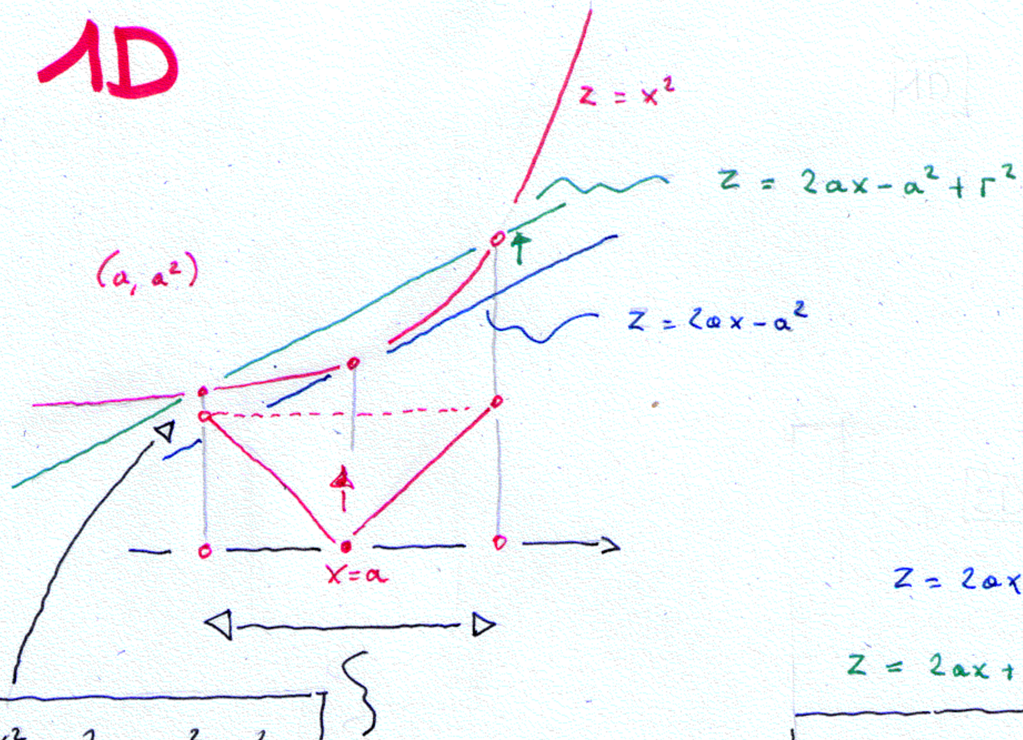


Connection Between Delaunay Triangulation and Convex Hulls





1D



$$x^2 = 2ax - a^2 + r^2$$
$$\Downarrow$$
$$(x-a)^2 = r^2$$

$$z = x^2 + y^2$$

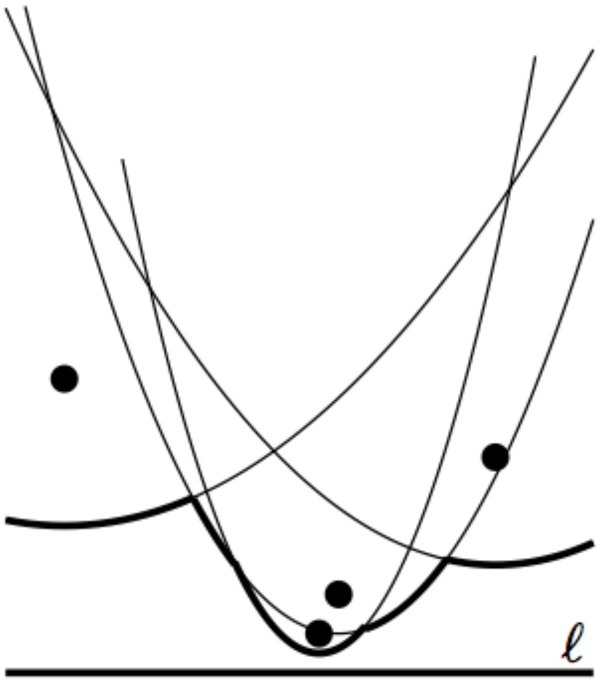
$$z = 2ax + 2by - (a^2 + b^2)$$

$$z = 2ax + 2by - (a^2 + b^2) + r^2$$

$$x^2 + y^2 = 2ax + 2by - (a^2 + b^2) + r^2$$
$$\Downarrow$$
$$(x-a)^2 + (y-b)^2 = r^2$$

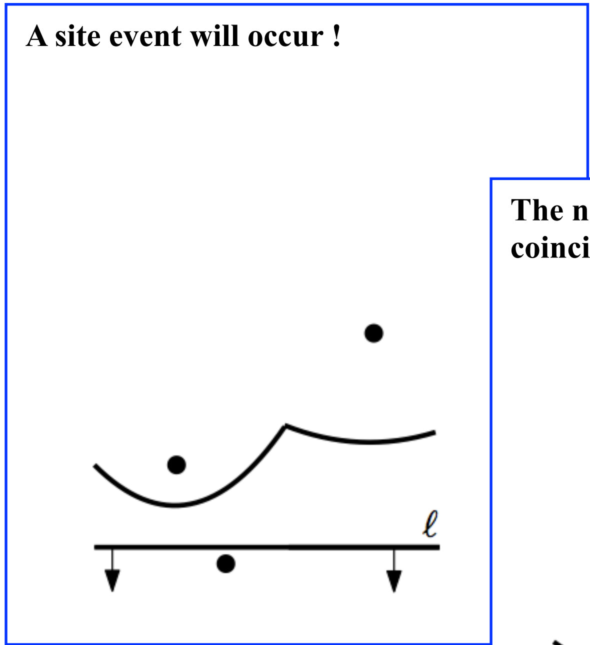
2D

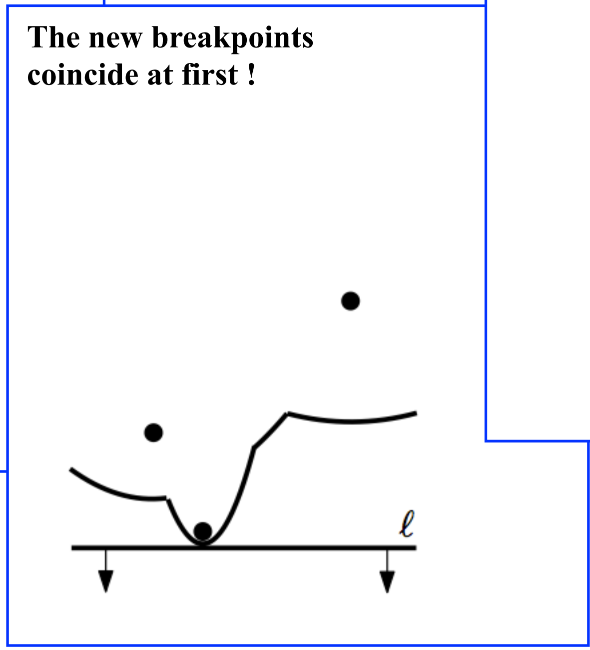
The beach line !

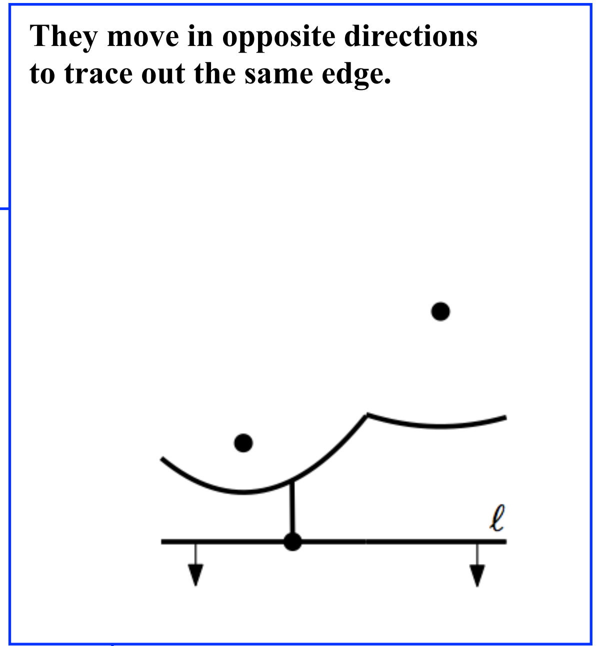


The locus of points that arec loser to some site than to the line is bounded by a parabola.

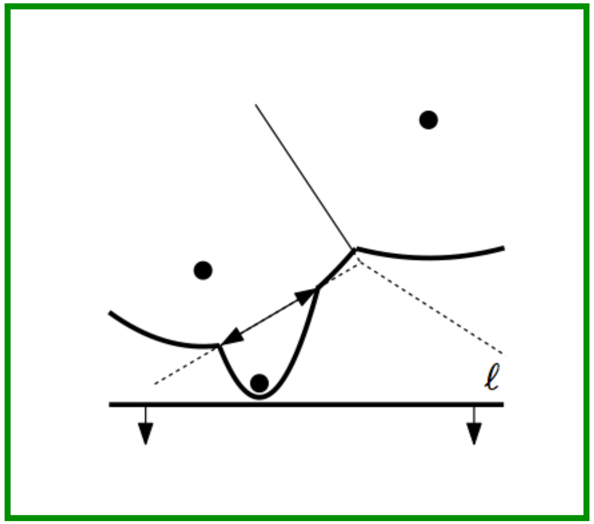
Hence, the locus of points that are closer to any site above than to the line is bounded by parabolic arcs



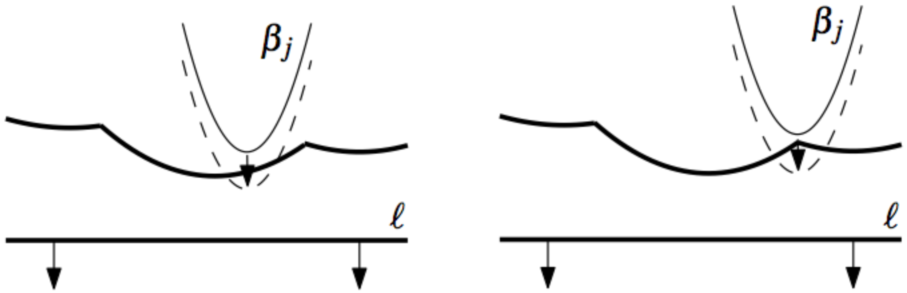




Maintaining
the beach line
as the sweep line moves



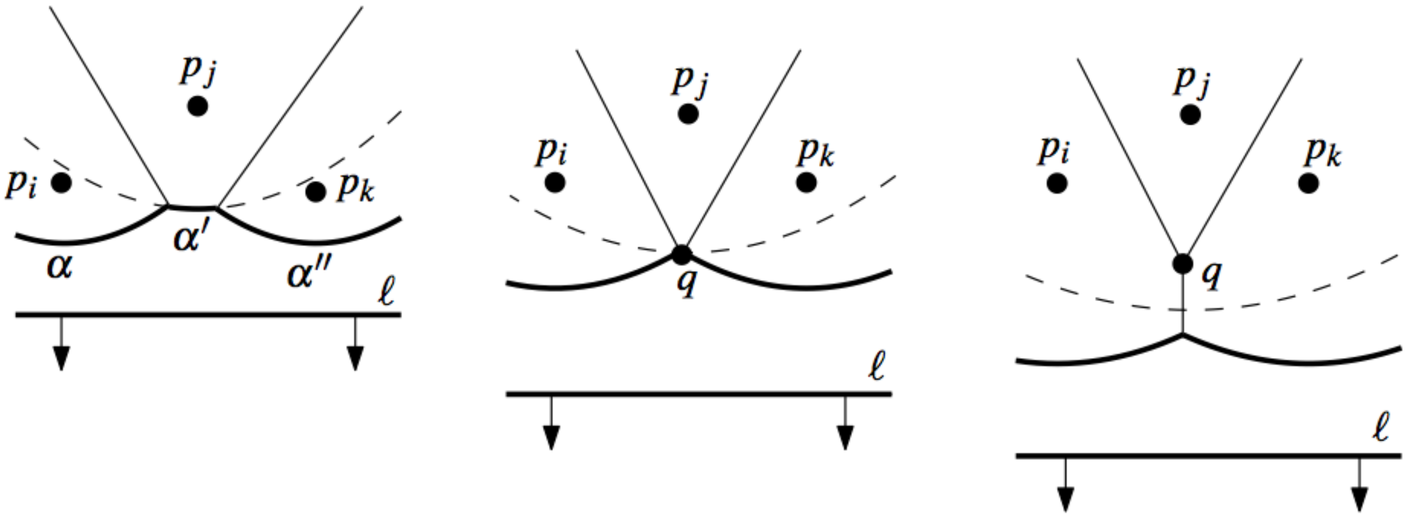
What happens at a site event ?



Lemma 7.6 *The only way in which a new arc can appear on the beach line is through a site event.*

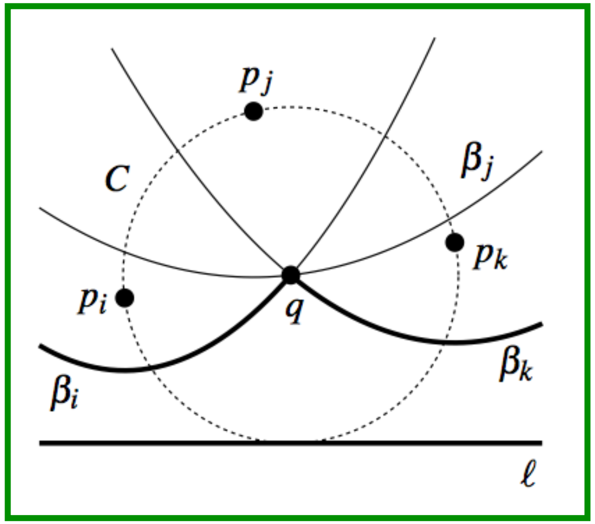
So now we understand what happens at a site event:
a new arc appears on the beach line, and a new edge of the Voronoi diagram starts to be traced out.

Is it possible that a new arc appears on the beach line in any other way ?
The answer is no !



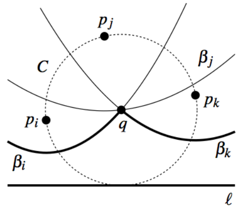
Lemma 7.7 *The only way in which an existing arc can disappear from the beach line is through a circle event.*

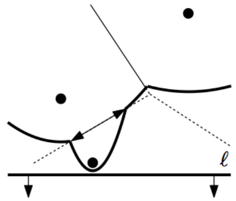
An arc disappears fromt the beach line



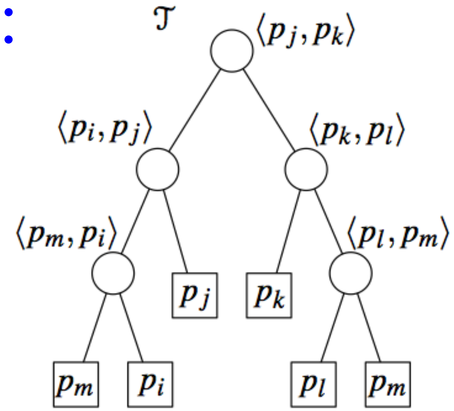
Data Structures

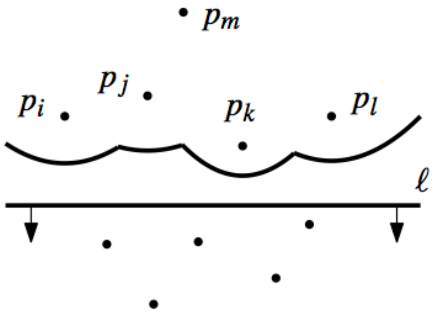
Events :
a priority
queue



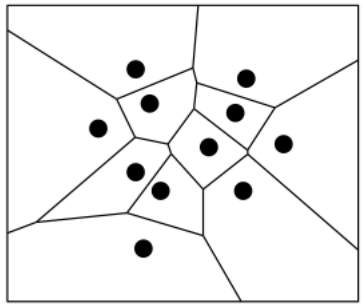


Beach line :
a binary
search
tree

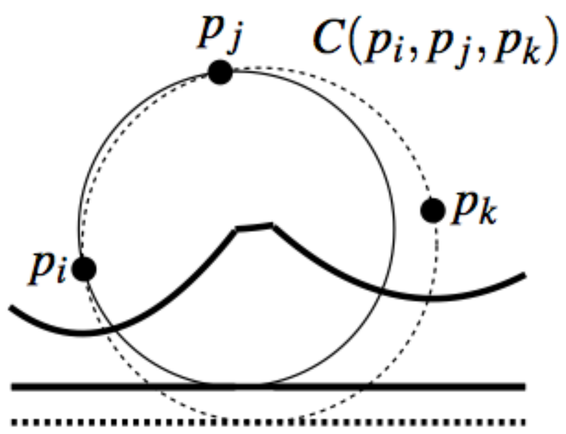




Voronoi diagram :
a doubly-connected
edge list



Designing Fortune's algorithm



All the site events are known in advance, but the circle events are not.

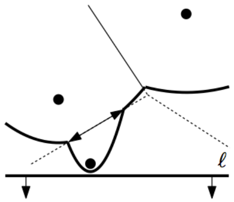
This brings us to one final issue that we must discuss, namely the detection of circle events.

1. Initialize
   - Event queue Q ← all site events
   - Binary search tree T ← ∅
   - Doubly linked list D ← ∅
2. While Q not ∅,
   - Remove event (e) from Q with largest y-coordinate
     - HandleEvent(e, T, D)

Handling Site Events

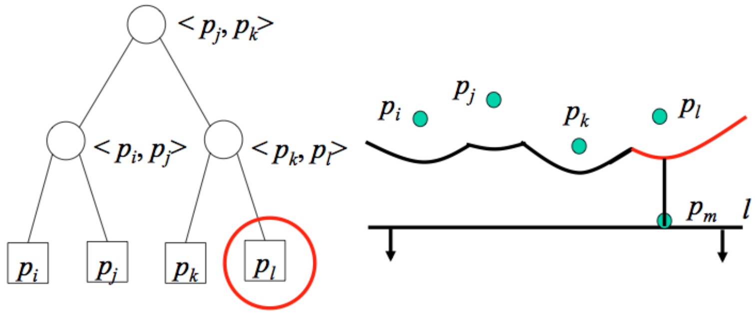
1. Locate the existing arc (if any) that is above the new site
2. Break the arc by replacing the leaf node with a sub tree representing the new arc and its break points
3. Add two half-edge records in the doubly linked list
4. Check for potential circle event(s), add them to event queue if they exist

A site event



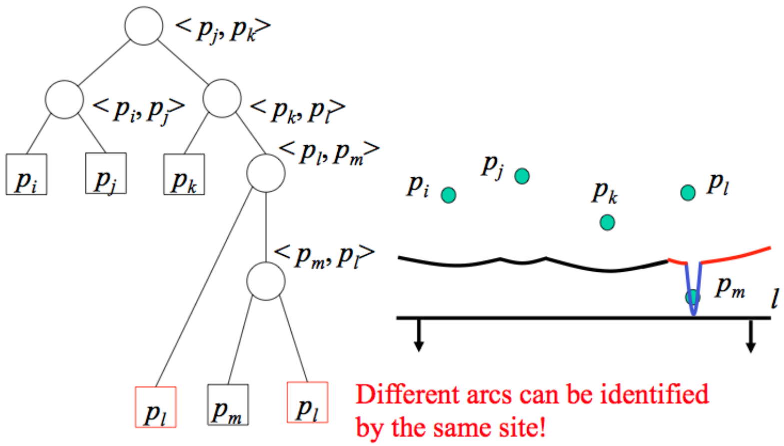
Locate the arc above

- The x coordinate of the new site is used for the binary search
- The x coordinate of each breakpoint along the root to leaf path is computed on the fly



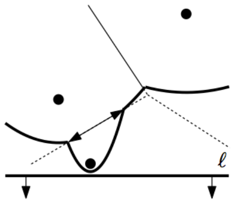
Break the arc

Corresponding leaf replaced by a new sub-tree

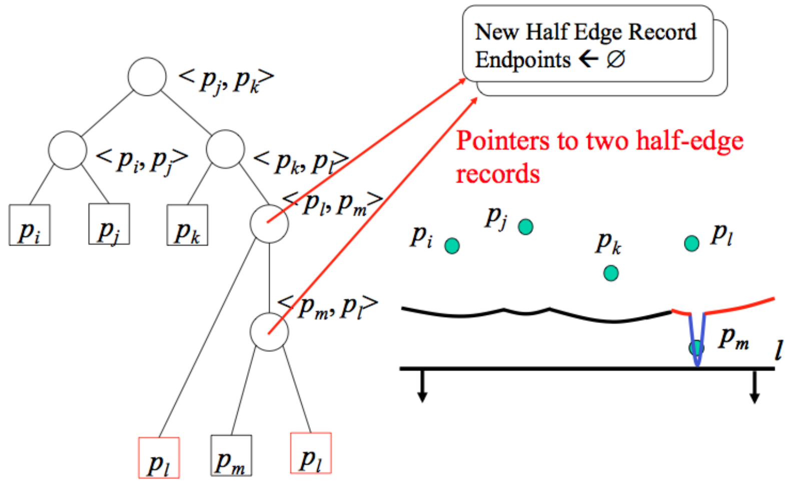


Different arcs can be identified by the same site!

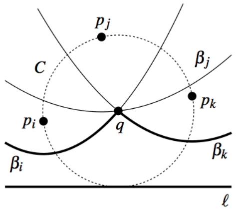
A new site



Add a new edge

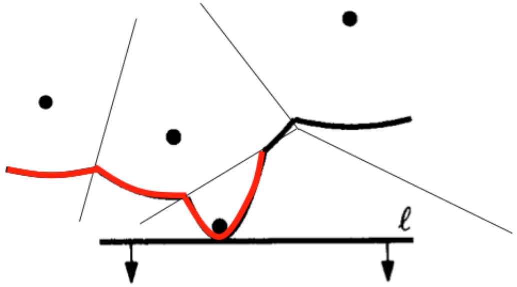


A circle event



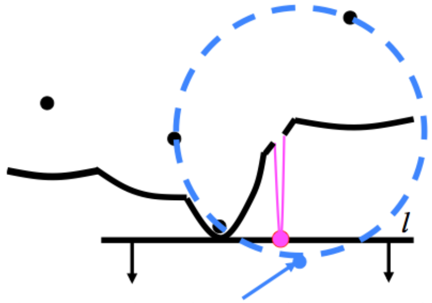
Checking for potential circle events

- Scan for triple of consecutive arcs and determine if breakpoints converge
  - Triples with new arc in the middle do not have break points that converge



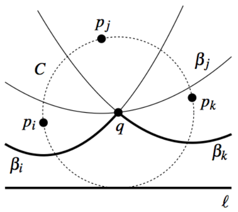
Converging breakpoints may not always yield a circle event

- Appearance of a new site before the circle event makes the potential circle non-empty



(The original circle event becomes a *false alarm*)

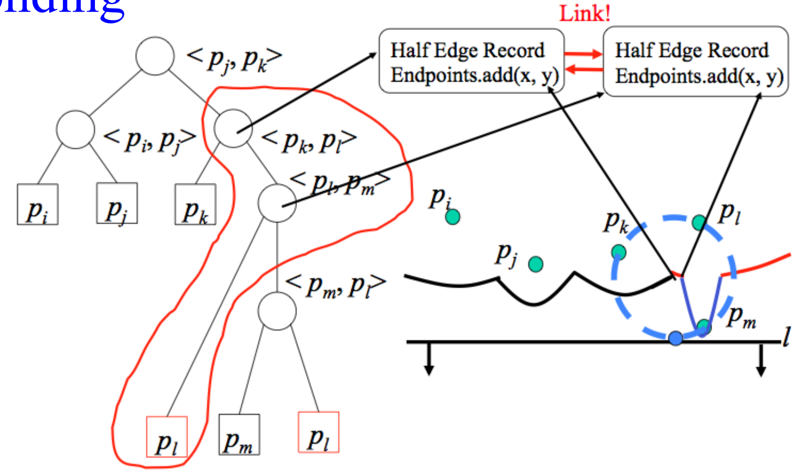
A circle event



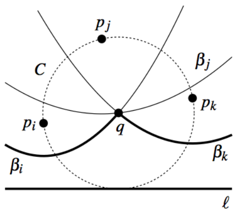
Handling circle events

1. Locate the leaf representing the existing arc that is above the new site
   - Delete the potential circle event in the event queue
2. Break the arc by replacing the leaf node with a sub tree representing the new arc and break points
3. Add a new edge record in the doubly linked list
4. Check for potential circle event(s), add them to queue if they exist
   - Store in the corresponding leaf of T a pointer to the new circle event in the queue

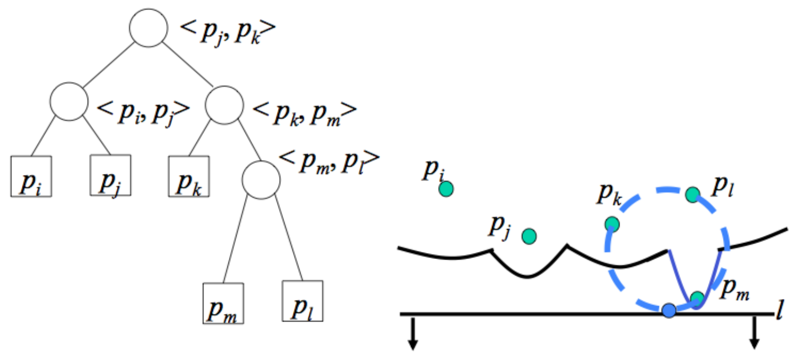
Add vertex to the corresponding edge



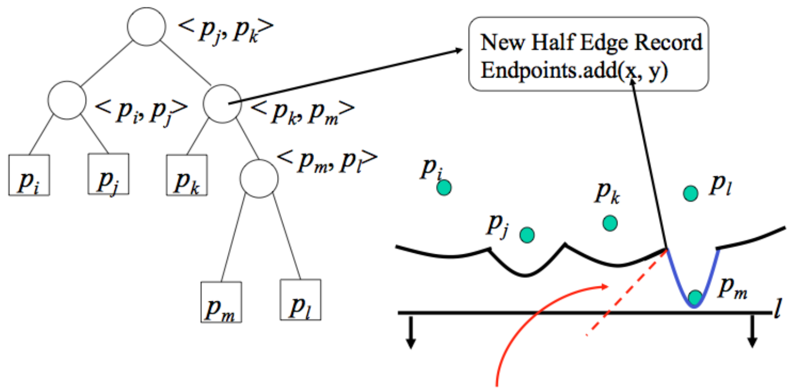
A circle event



Deleting the disappearing arc

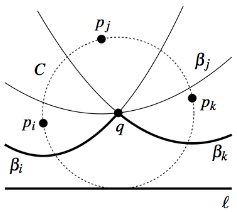


Creating a new edge

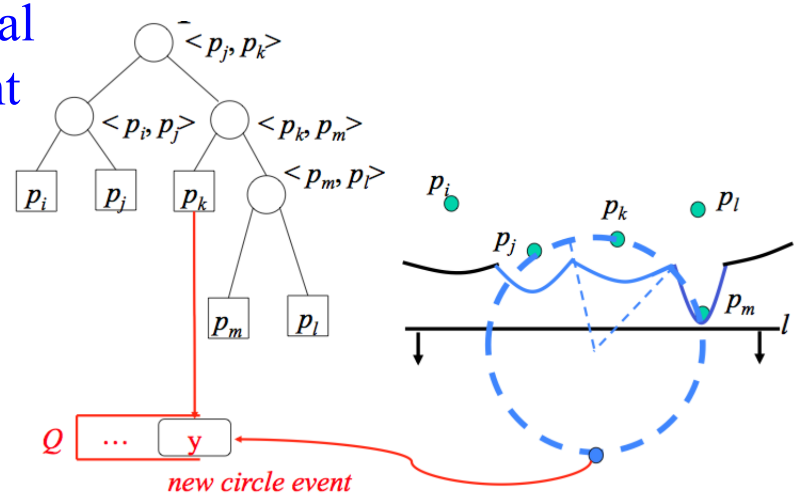


A new edge is traced out by the new break point $<p_k, p_m>$

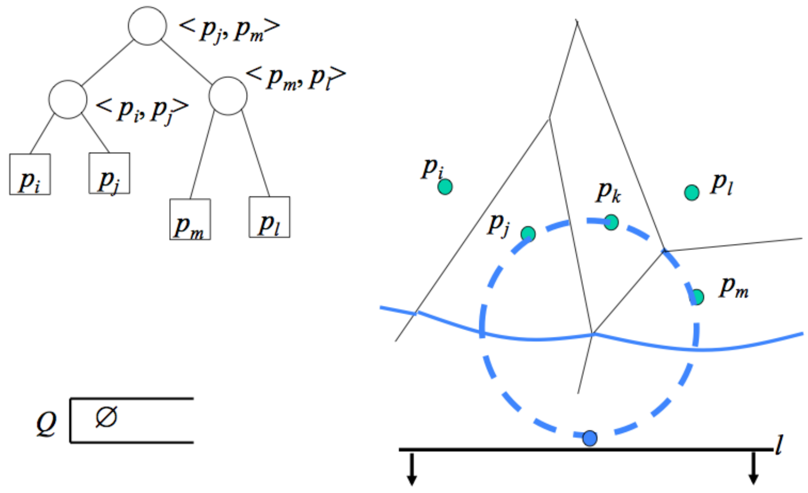
A circle event



Check the new triplet for potential circle event



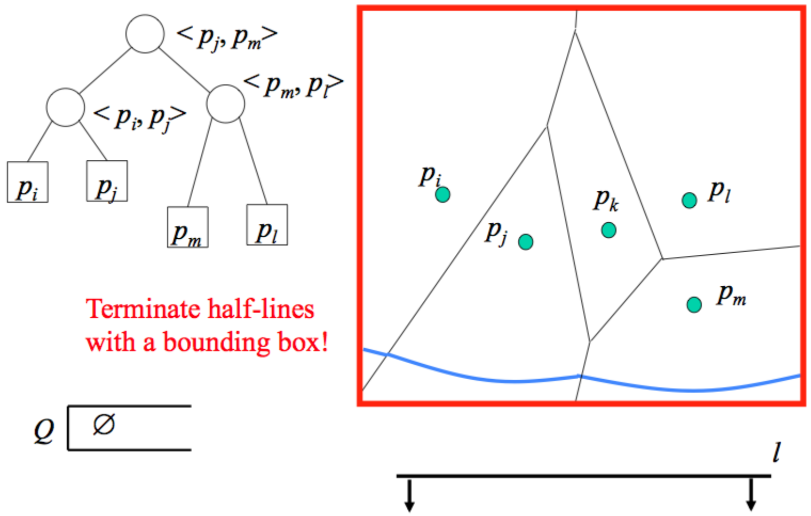
End of the story



Algorithm terminates when the queue is empty,
but the beach line and its break points continue to trace Voronoi edges.

Close those half-infinite edges with a bounding box

Incorporate
a bounding box



O(n log(n))

Handling a circle event

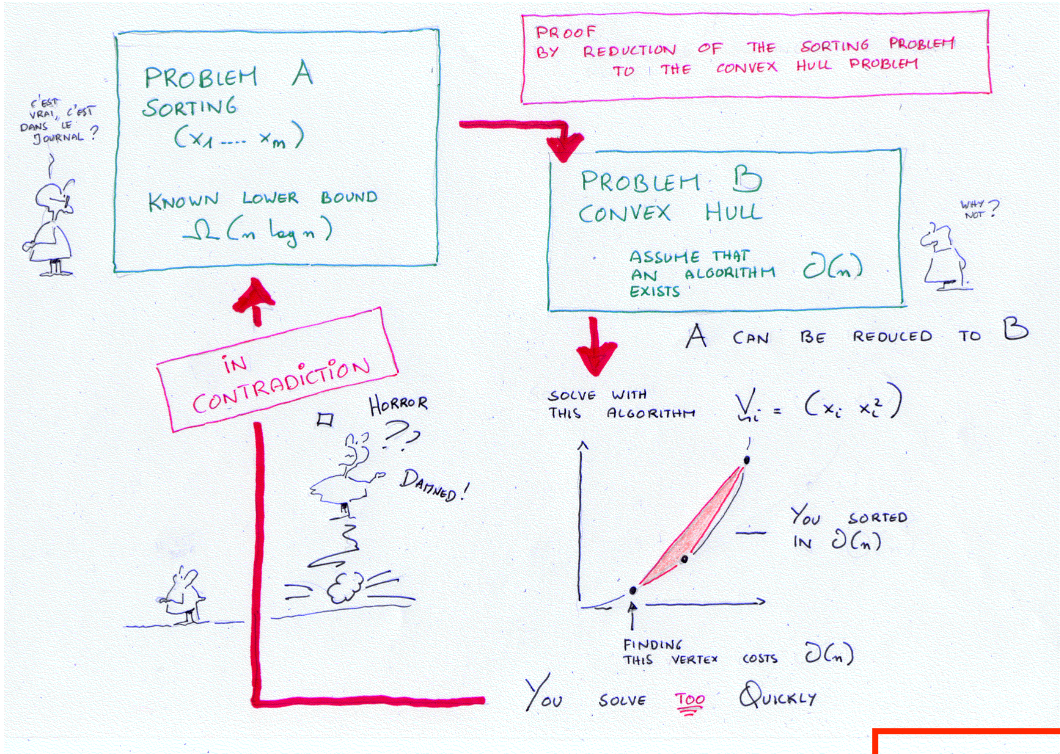
1. Delete from T the leaf node of the disappearing arc and its associated circle events in the event queue — O(log n)
2. Add vertex record in doubly link list — O(1)
3. Create new edge record in doubly link list — O(1)
4. Check the new triplets formed by the former neighboring arcs for potential circle events — O(1)

Handling a site event

1. Locate the leaf representing the existing arc that is above the new site — O(log n)
   - Delete the potential circle event in the event queue
2. Break the arc by replacing the leaf node with a sub tree representing the new arc and break points — O(1)
3. Add a new edge record in the link list — O(1)
4. Check for potential circle event(s), add them to queue if they exist — O(1)
   - Store in the corresponding leaf of T a pointer to the new circle event in the queue

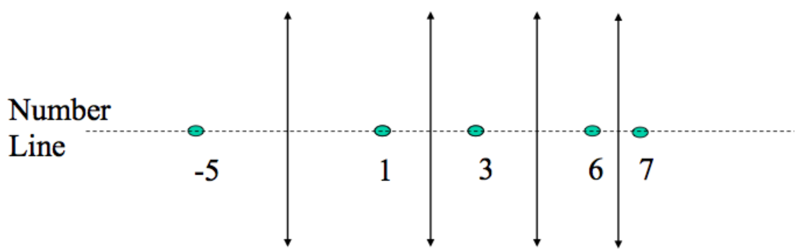
Each new site can generate at most two new arcs.
The beach line has at most 2n+1 arcs.
At most *O*(n) events in the queue list.

The total running time is *O*(n log(n))



Is Fortune algorithm optimal ?

- We can sort numbers using any algorithm that constructs a Voronoi diagram!



- Map input numbers to a position on the number line. The resulting Voronoi diagram is doubly linked list that forms a chain of unbounded cells in the left-to-right (sorted) order.

Exercice 7

7.5 Give an example where the parabola defined by some site p_i contributes more than one arc to the beach line. Can you give an example where it contributes a linear number of arcs?

7.6 Give an example of six sites such that the plane sweep algorithm encounters the six site events before any of the circle events. The sites should lie in general position: no three sites on a line and no four sites on a circle.

7.7 Do the breakpoints of the beach line always move downwards when the sweep line moves downwards? Prove this or give a counterexample.