

Computational Geometry

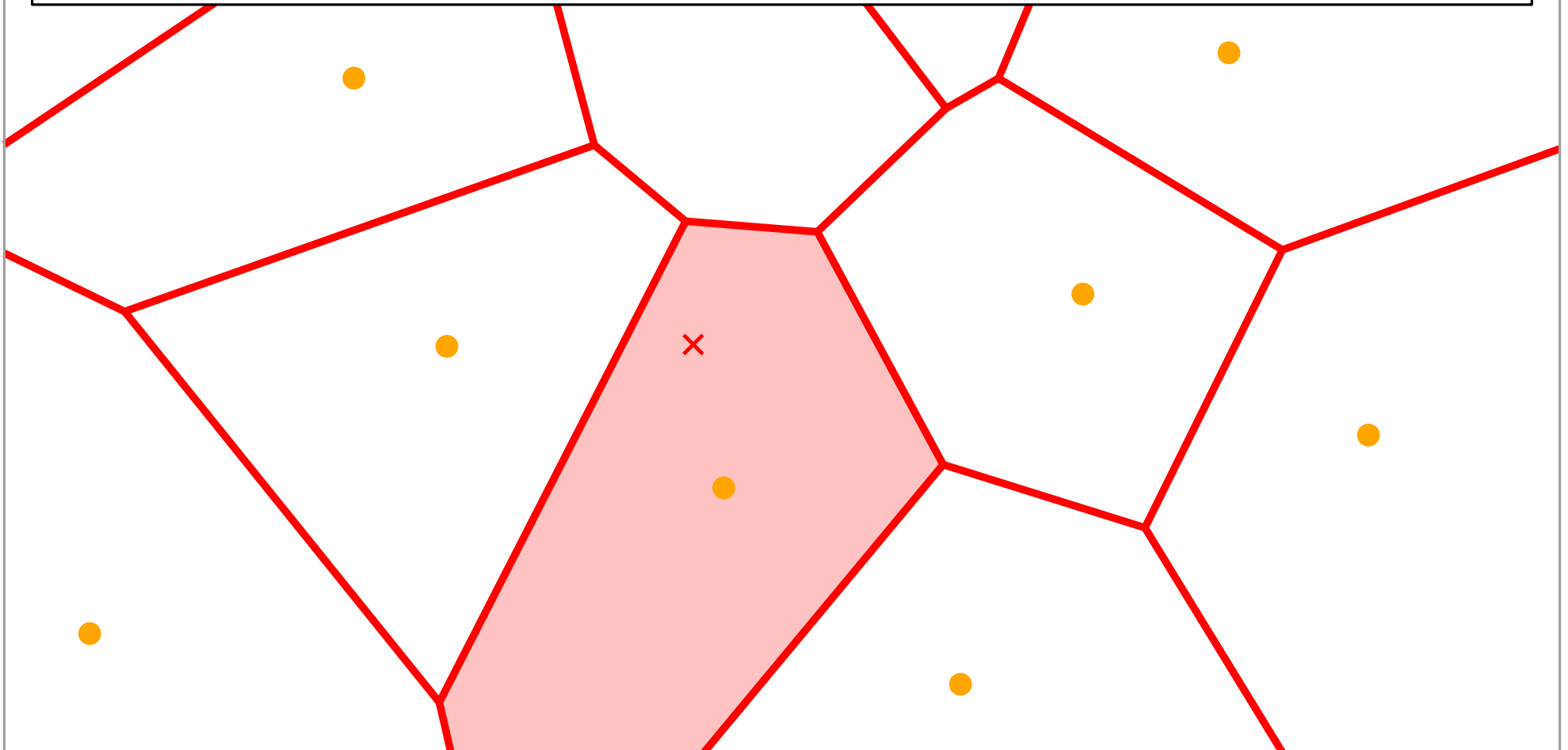
Winter term 2016/17

The Post-Office Problem

Lecture #7

The Post-Office Problem

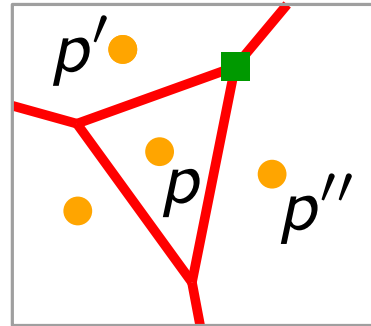
- Tasks:** 1) Define Voronoi cells, edges and vertices!
2) Are Voronoi cells convex?



The Voronoi diagram

Let P be a set of points in the plane and let $p, p', p'' \in P$.

[Voronoi diagram]



$\text{Vor}(P)$ $\begin{cases} \rightarrow \text{subdivision of } \mathbb{R}^2 \\ \rightarrow \text{geometric graph} \end{cases}$

[Voronoi cell]

$$\begin{aligned} \mathcal{V}(\{p\}) = \mathcal{V}(p) &= \{x \in \mathbb{R}^2 : |xp| < |xq| \text{ for all } q \in P \setminus \{p\}\} \\ &= \bigcap_{q \neq p} h(p, q) \end{aligned}$$

[Voronoi edge]

$$\begin{aligned} \mathcal{V}(\{p, p'\}) &= \{x : |xp| = |xp'| \text{ and } |xp| < |xq| \ \forall q \neq p, p'\} \\ &= \text{rel-int}(\partial\mathcal{V}(p) \cap \partial\mathcal{V}(p')), \text{ i.e., w/o the endpoints} \end{aligned}$$

[Voronoi vertex]

$$\mathcal{V}(\{p, p', p''\}) = \partial\mathcal{V}(p) \cap \partial\mathcal{V}(p') \cap \partial\mathcal{V}(p'')$$

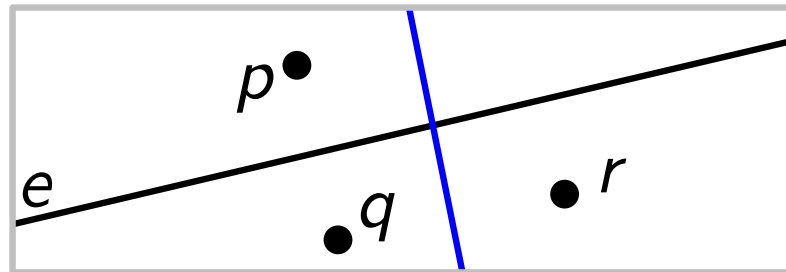
Overall Shape of $\text{Vor}(P)$

Theorem. Let $P \subset \mathbb{R}^2$ be a set of n pts (called *sites*).
If all sites are collinear, $\text{Vor}(P)$ consists of $n - 1$ parallel lines. Otherwise, $\text{Vor}(P)$ is connected and its edges are line segments or half-lines.

Proof.

Assume that P is not collinear.

- Assume that $\text{Vor}(P)$ contains an edge e that is a full line, say, $e = b(p, q)$.



Let $r \in P$ be not collinear with p and q .

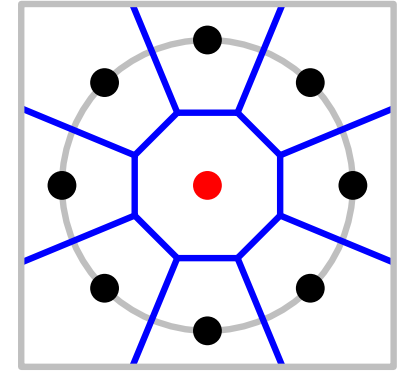
Then $b(q, r)$ is not parallel to e .

$\Rightarrow e \cap h(r, q)$ is closer to r than to p or q .

$\Rightarrow e$ is bounded on at least one side. □

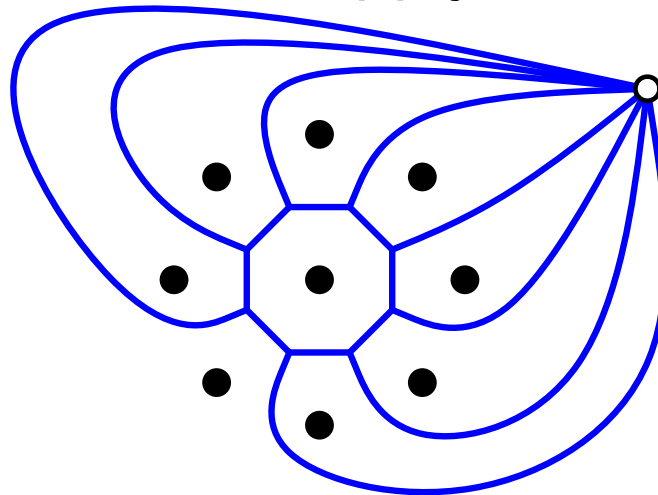
Complexity

Task: Construct a set P of point sites such that $\text{Vor}(P)$ has a cell of linear complexity!



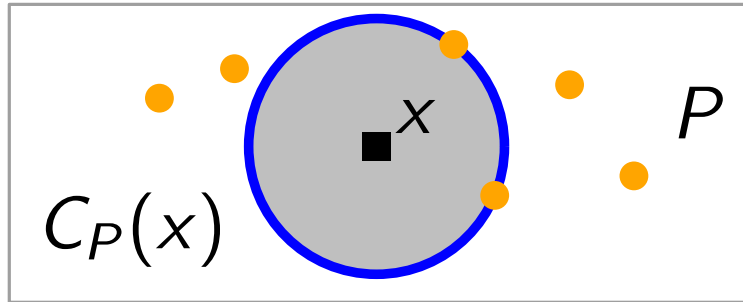
Theorem. Given a set $P \subset \mathbb{R}^2$ of n sites, $\text{Vor}(P)$ consists of at most $2n - 5$ vertices and $3n - 6$ edges.

Proof. *Problem:* unbounded edges!
 \Rightarrow can't apply Euler directly, but...



Characterization of Voronoi vtc and edges

$C_P(x) :=$ largest circle centered at x w/o sites in its interior



- Theorem:**
- (i) x Voronoi vtx $\Leftrightarrow |C_P(x) \cap P| \geq 3$
 - (ii) $b(p, p')$ contains a Voronoi edge $\Leftrightarrow \exists x \in b(p, p') : C_P(x) \cap P = \{p, p'\}$

Computation

Brute force: For each $p \in P$, compute $\mathcal{V}(p) = \underbrace{\bigcap_{p' \neq p} h(p, p')}_{O(n \log n) \text{ time}}$.

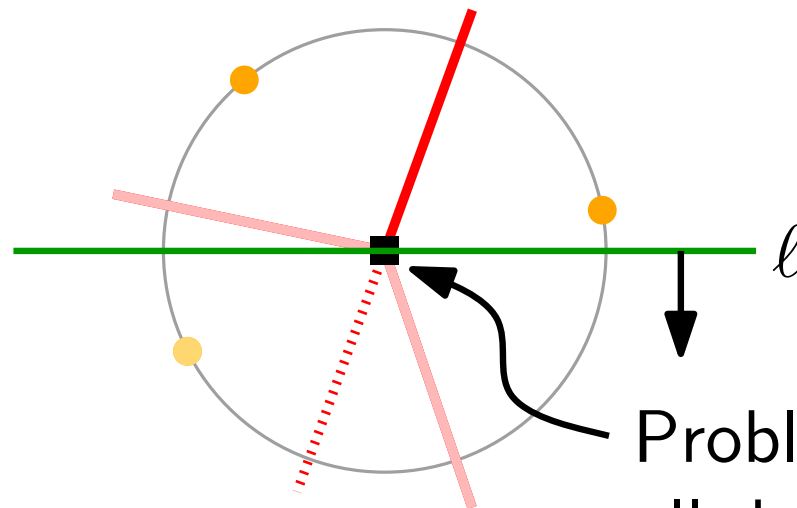
[Chapter 4]

$O(n \log n)$ time

in total: $O(n^2 \log n)$ time

– but the complexity of $\text{Vor}(P)$ is *linear*!

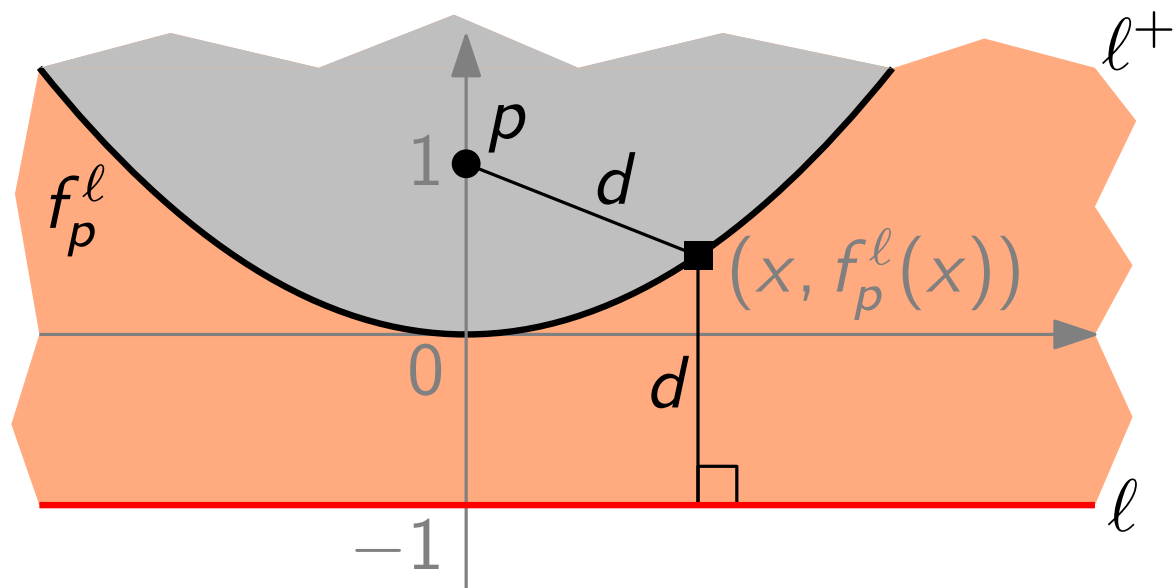
Sweep?



Problem: We don't know all defining sites yet :(

Sweep?

Which part of the plane above ℓ is fixed by what we've seen?



Solution:

f_p^l is the parabola with focus p and directrix ℓ .

Task: Compute f_p^l for $p = (0, 1)$ and $\ell: y = -1$!

Definition. beachline $\beta \equiv$ lower envelope of $(f_p^l)_{p \in P \cap \ell^+}$

Observation. β is x -monotone. 

The beachline β

Question: What does β have to do with $\text{Vor}(P)$?

Answer: “Breakpoints” of β trace out the Voronoi edges!

Lemma. New arcs on β only appear through *site events*, that is, whenever ℓ hits a new site.

Corollary. β consists of at most $2n - 1$ arcs.

Definition. *Circle event:* ℓ reaches lowest pt of a circle through three sites above ℓ whose arcs are consecutive on β .

Lemma. Arcs disappear from β only at circle events.

Lemma. The Voronoi vtc correspond 1:1 to circle events.

Fortune's Sweep

```
VoronoiDiagram( $P \subset \mathbb{R}^2$ )
```

```
 $Q \leftarrow$  new PriorityQueue( $P$ ) // site events sorted acc.  $y$ -coord.
```

```
 $\mathcal{T} \leftarrow$  new BalancedBinarySearchTree() // sweep status ( $\beta$ )
```

```
 $\mathcal{D} \leftarrow$  new DCEL() // to-be Vor( $P$ )
```

```
while not  $Q.empty()$  do
```

```
     $p \leftarrow Q.ExtractMax()$ 
```

```
    if  $p$  site event then
```

```
        | HandleSiteEvent( $p$ )
```

```
    else
```

```
        |  $\alpha \leftarrow$  arc on  $\beta$  that will disappear
```

```
        | HandleCircleEvent( $\alpha$ )
```

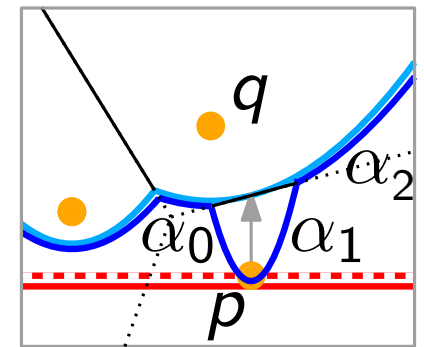
```
treat remaining internal nodes of  $\mathcal{T}$  ( $\equiv$  unbnd. edges of Vor( $P$ ))
```

```
return  $\mathcal{D}$ 
```

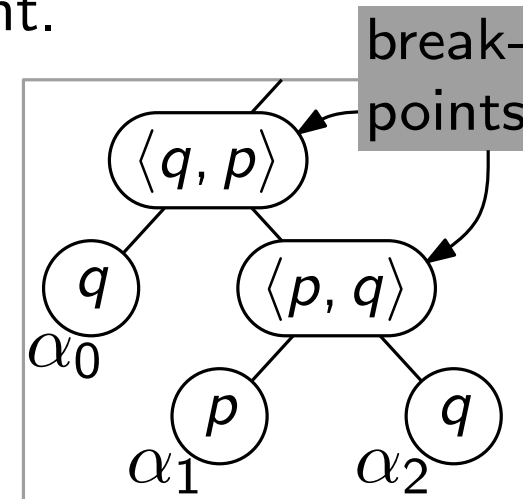
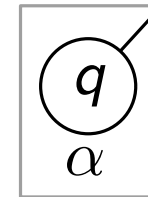
Handling Events

HandleSiteEvent(point p)

- Search in \mathcal{T} for the arc α vertically above p .
If α has pointer to circle event in \mathcal{Q} , delete this event.
- Split α into α_0 and α_2 .
Let α_1 be the new arc of p .
- Add Vor-edges $\langle q, p \rangle$ and $\langle p, q \rangle$ to DCEL.
- Check $\langle \cdot, \alpha_0, \alpha_1 \rangle$ and $\langle \alpha_1, \alpha_2, \cdot \rangle$ for circle events.

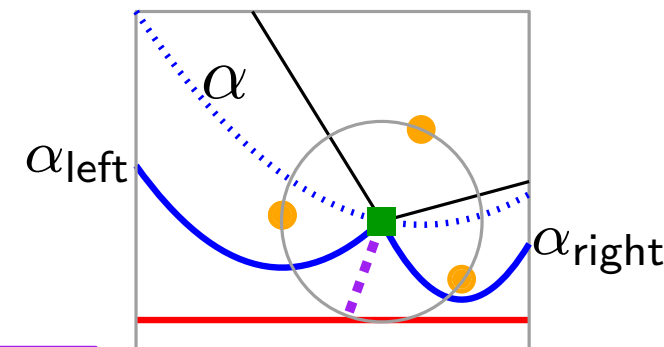


In \mathcal{T} :



HandleCircleEvent(arc α)

- $\mathcal{T}.\text{delete}(\alpha)$; update breakpts
- Delete all circle events involving α from \mathcal{Q} .
- Add Vor-vtx $\alpha_{\text{left}} \cap \alpha_{\text{right}}$ and Vor-edge $\langle \alpha_{\text{left}}, \alpha_{\text{right}} \rangle$ to DCEL.
- Check $\langle \cdot, \alpha_{\text{left}}, \alpha_{\text{right}} \rangle$ and $\langle \alpha_{\text{left}}, \alpha_{\text{right}}, \cdot \rangle$ for circle events.



Running time? $O(\log n)$ per event...

Running Time?

VoronoiDiagram($P \subset \mathbb{R}^2$)

$Q \leftarrow$ new PriorityQueue(P) // site events sorted acc. y -coord.

$\mathcal{T} \leftarrow$ new BalancedBinarySearchTree() // sweep status (β)

$\mathcal{D} \leftarrow$ new DCEL() // to-be Vor(P)

while not $Q.empty()$ **do**

$p \leftarrow Q.ExtractMax()$

if p site event **then**

 | **HandleSiteEvent**(p) exactly n such events

else

 | $\alpha \leftarrow$ arc on β that will disappear

 | **HandleCircleEvent**(α) at most $2n - 5$ such events

treat remaining internal nodes of \mathcal{T} (\equiv unbnd. edges of Vor(P))

return \mathcal{D}

Summary

Theorem. Given a set P of n pts in the plane, Fortune's sweep computes $\text{Vor}(P)$ in $O(n \log n)$ time and $O(n)$ space.



Steven Fortune
Bell Labs

Steven Fortune. A sweepline algorithm for Voronoi diagrams.
Proc. 2nd Annual ACM Symposium on Computational Geometry.
Yorktown Heights, NY, pp. 313–322. 1986.