Voronoi diagrams and Delaunay triangulations

Lecture 9, CS 4235
12 march 2004

Antoine Vigneron

`antoine@comp.nus.edu.sg`

National University of Singapore

Outline

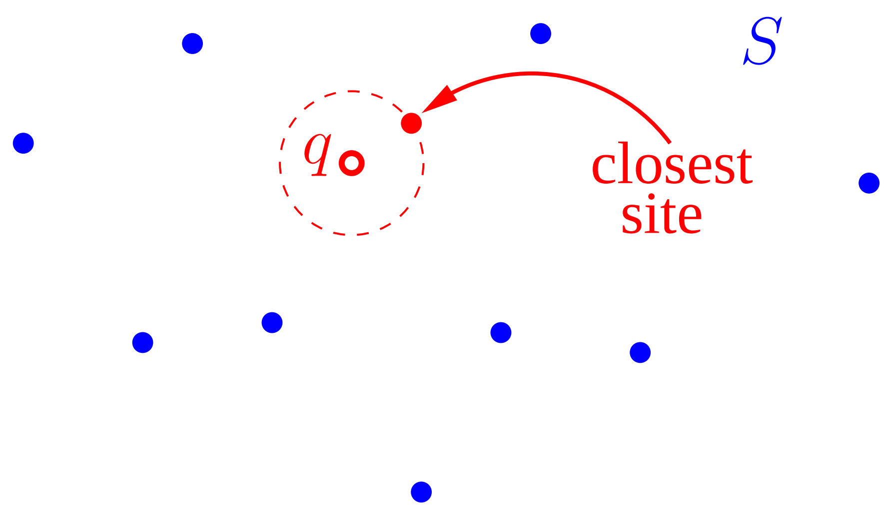
- today: geometry, no algorithm
- two problems
  - proximity, mesh generation
  - surprisingly, they are closely related
- geometric notions
  - Voronoi diagram, Delaunay triangulation
  - planar graph duality
- references
  - D. Mount Lecture 16 (except algorithm) and 17
  - textbook chapter 7 (online!) and 9
  - demo (J. Snoeyink) at:
    http://www.cs.ubc.ca/spider/snoeyink/demos/crust/home.html

Voronoi diagrams

A Voronoi diagram

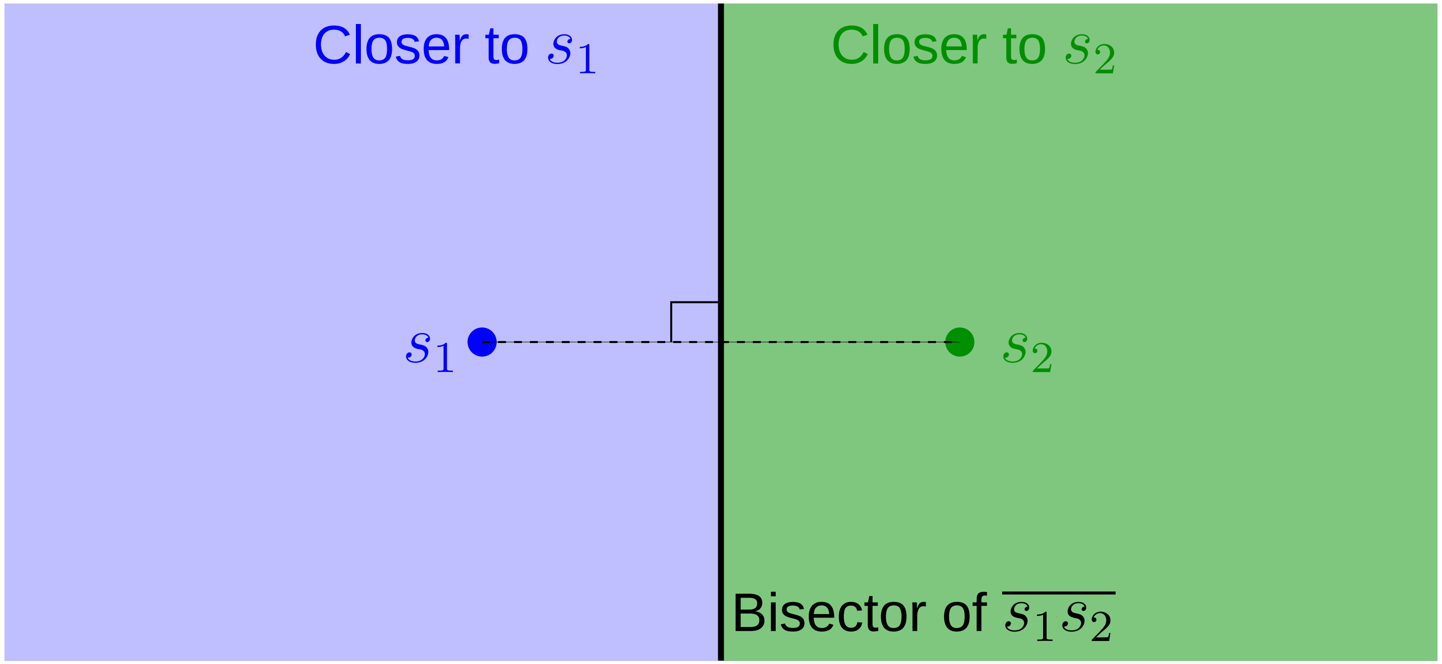
Proximity

- what is it?
- a dataset S of n points (called *sites*) in $\mathbb{R}^2$
- let $S = \{s_1, s_2 \dots s_n\}$
- query point q, find closest site to q (=*proximity queries*)



Problem

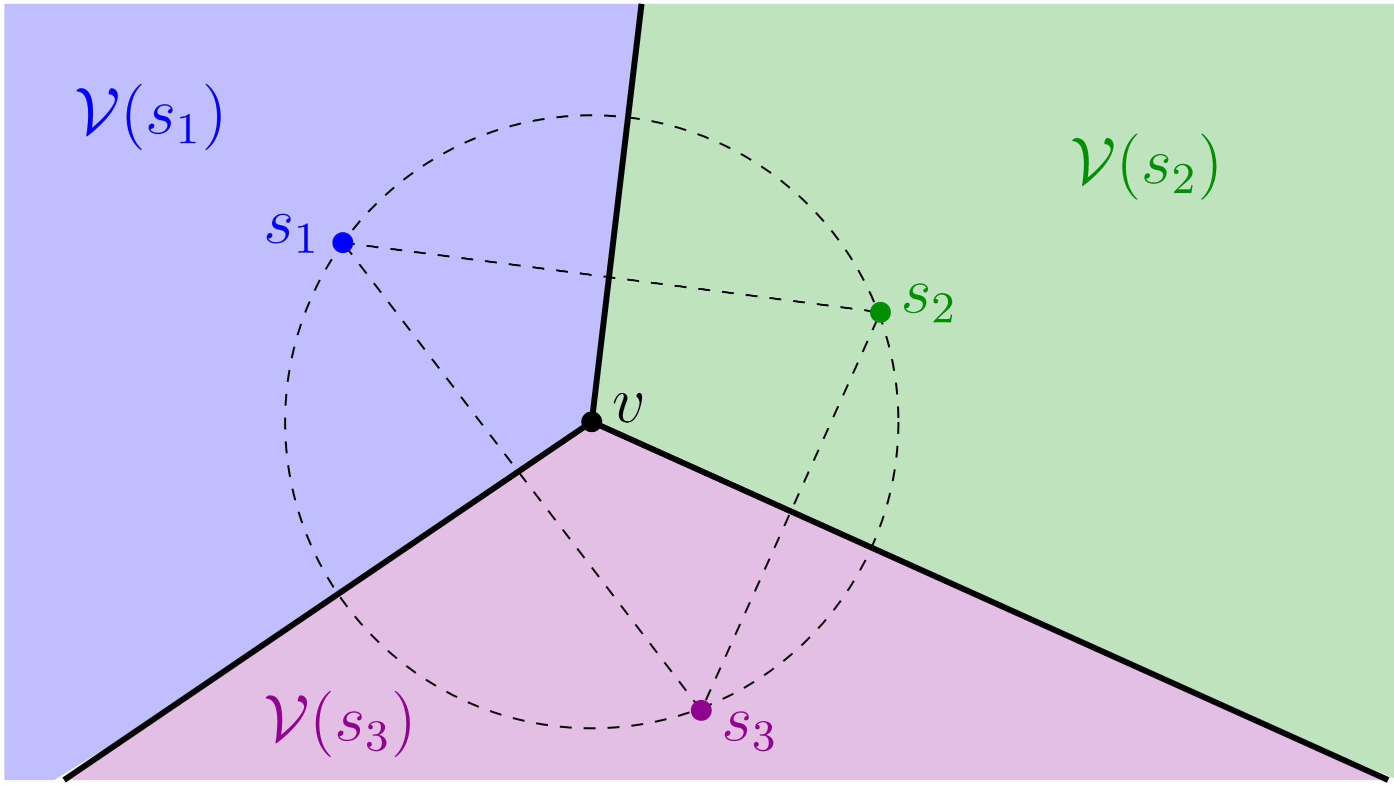
- how to address proximity?
- draw a diagram
- example with $|S| = 2$:



- this is the *Voronoi diagram* of $S = \{s_1, s_2\}$

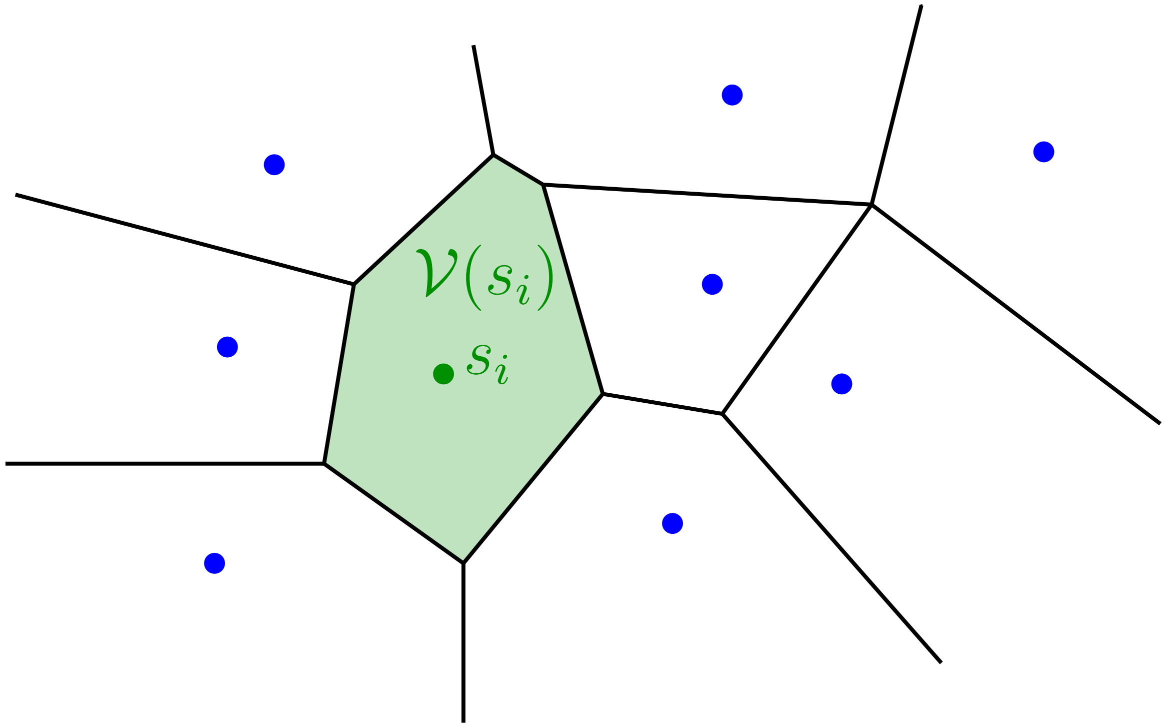
Example with $|S| = 3$

- Voronoi diagram of $S = \{s_1, s_2, s_3\}$



- v: a *Voronoi vertex*. Center of the circumcircle of triangle $s_1 s_2 s_3$
- $\mathcal{V}(s_i)$: *Voronoi cell* of s_i

Example

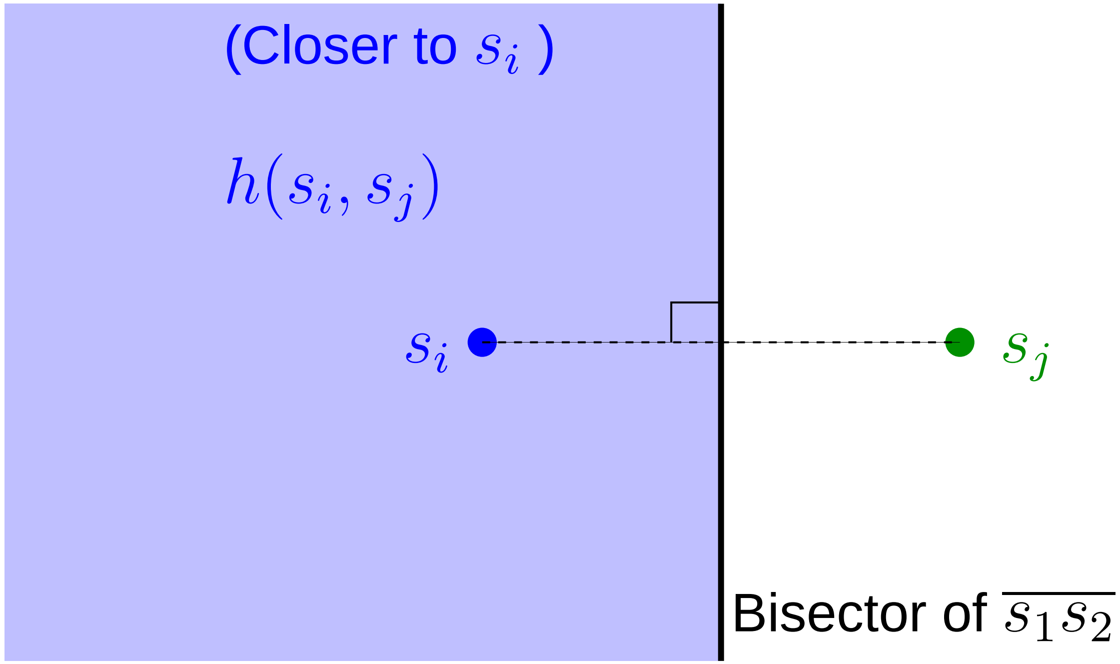


- $\mathcal{V}(s_i) = \{x \in \mathbb{R}^2 \mid \ \forall j \neq i, \ |s_i x| < |s_j x|\}$

Half–plane $h(s_i, s_j)$

- we denote

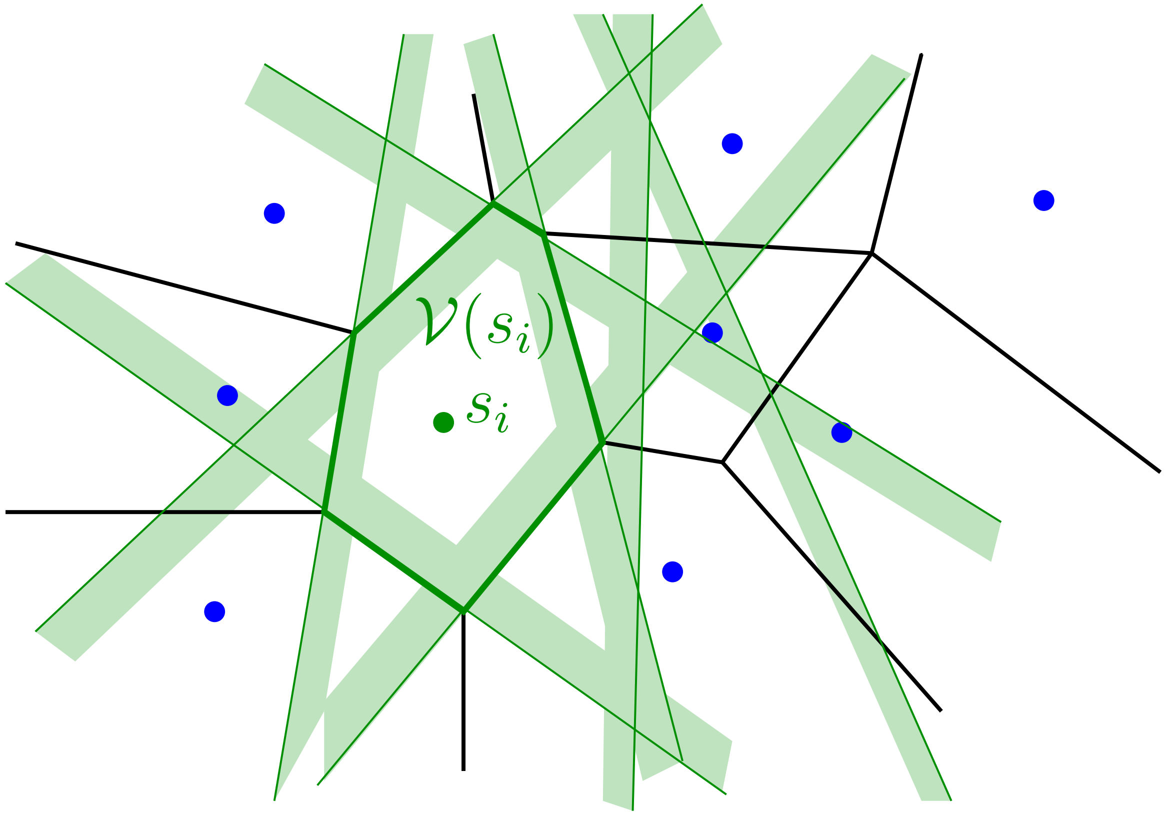
$$h(s_i, s_j) = \left\{ x \in \mathbb{R}^2 \middle| |s_i x| < |s_j x| \right\}$$



Voronoi cell

- it follows that

$$\mathcal{V}(s_i) = \bigcap_{j \neq i} h(s_i, s_j)$$



Properties

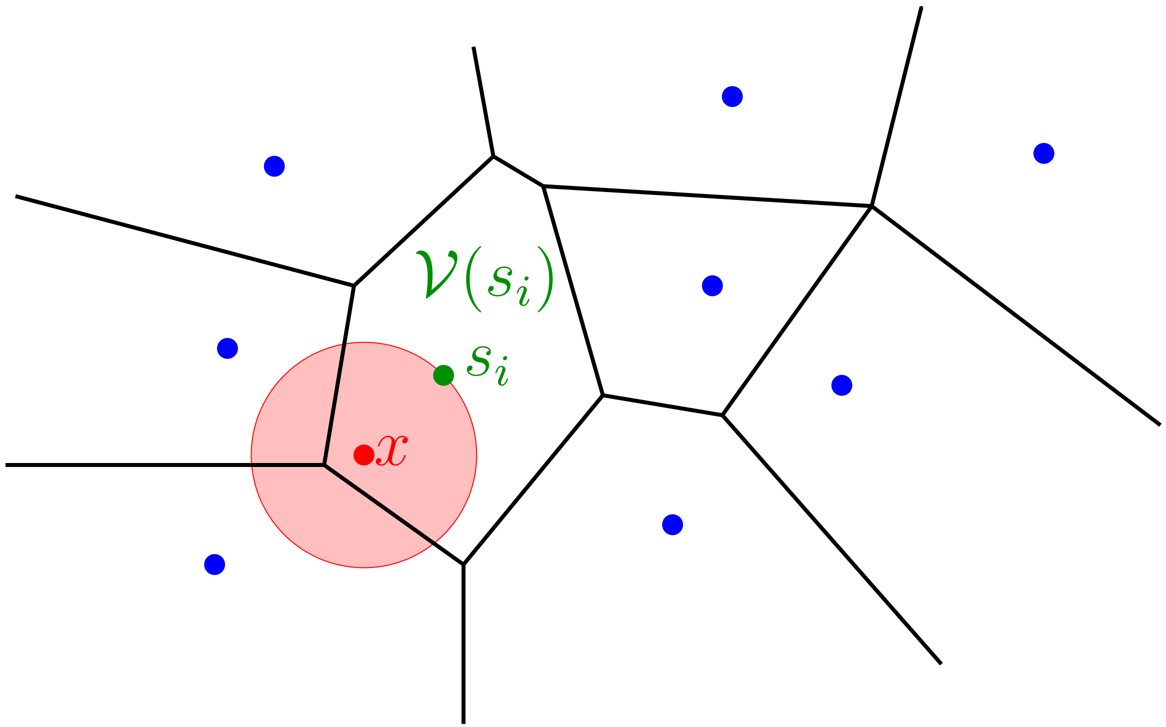
- the Voronoi diagram of S is *not* a planar straight line graph
  - reason: it has infinite edges
  - to fix this problem, we can restrict our attention to the portion of the Voronoi diagram that is within a large bounding box
- all the cells are convex, hence connected
- so the Voronoi diagram has n faces, one for each site
- it has $O(n)$ edges and vertices
  - non trivial; we can use the fact that vertices have degree at least 3+double counting+Euler's relation

Algorithmic consequences

- $\mathcal{V}(s_i)$ is an intersection of n half–planes
- so it can be computed in $O(n \log n)$ time (cf linear programming)
- we can compute the Voronoi diagram of S in $O(n^2 \log n)$ time
- we associate it with a point location data structure
- so we can answer proximity queries in:
  - $O(n^2 \log n)$ preprocessing time
  - expected $O(n)$ space usage
  - expected $O(\log n)$ query time
- next lecture: preprocessing time down to expected $\Theta(n \log n)$

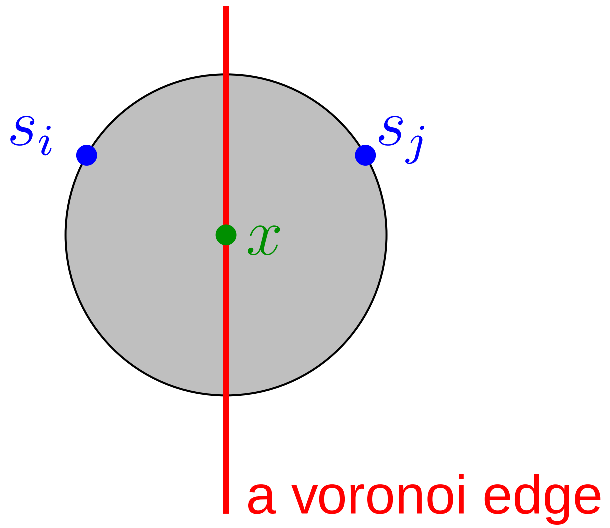
Voronoi cell 2

- $\forall x \in \mathcal{V}(s_i)$ the disk through s_i centered at x contains no other site than s_i



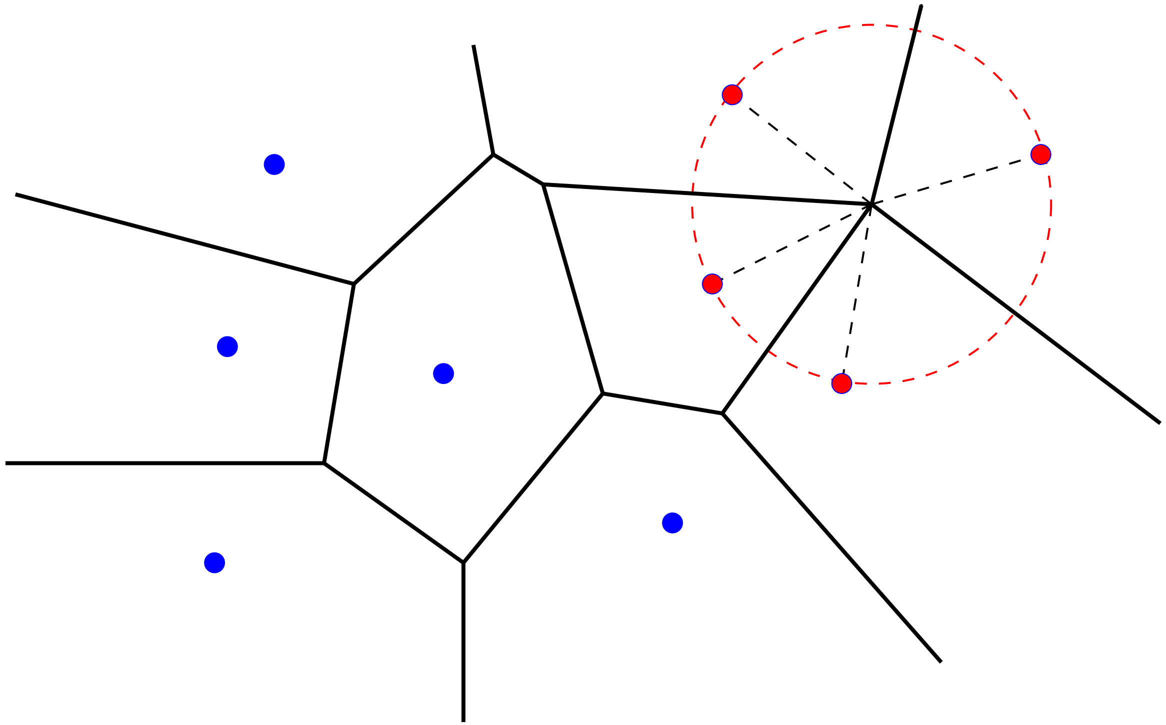
Voronoi edges

- a *Voronoi edge* is an edge of the Voronoi diagram
- a point x on a Voronoi edge is equidistant to two nearest sites s_i and s_j
- hence the circle centered at x through s_i and s_j contains no site in its interior



General position assumption

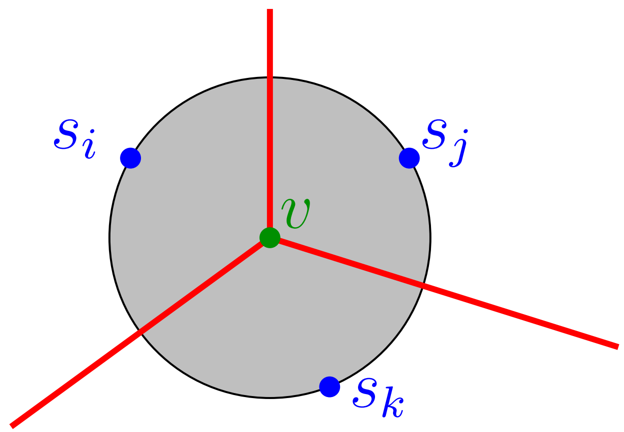
- general position assumption:
  - no four sites are cocircular



- a degenerate case: 4 sites lie on the same circle

Voronoi vertices

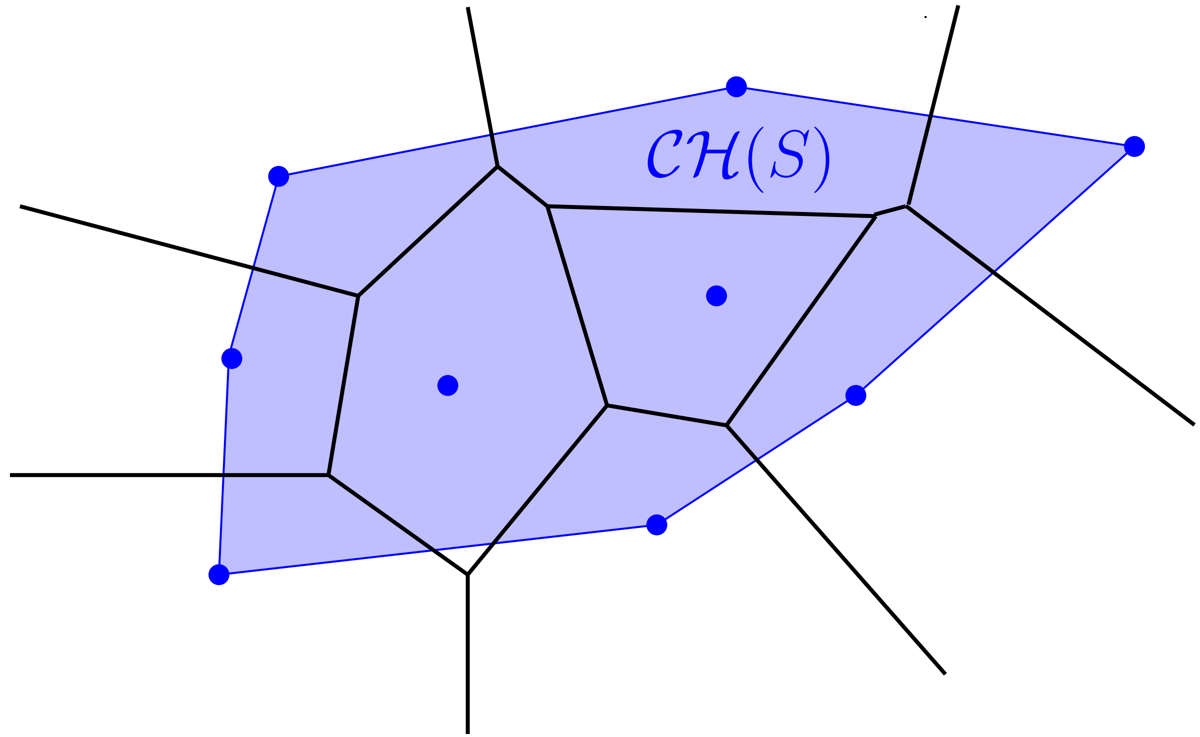
- a Voronoi vertex v is equidistant to three nearest sites s_i, s_j and s_k
- hence the circle centered at v through s_i, s_j and s_k contains no site in its interior



- by our general position assumption, each Voronoi vertex has degree 3 (= adjacent to three edges)

Voronoi cells

- if $\mathcal{V}(s_i)$ is bounded, then it is a convex polygon
- $\mathcal{V}(s_i)$ is unbounded iff s_i is a vertex of $\mathcal{CH}(S)$



Consequence

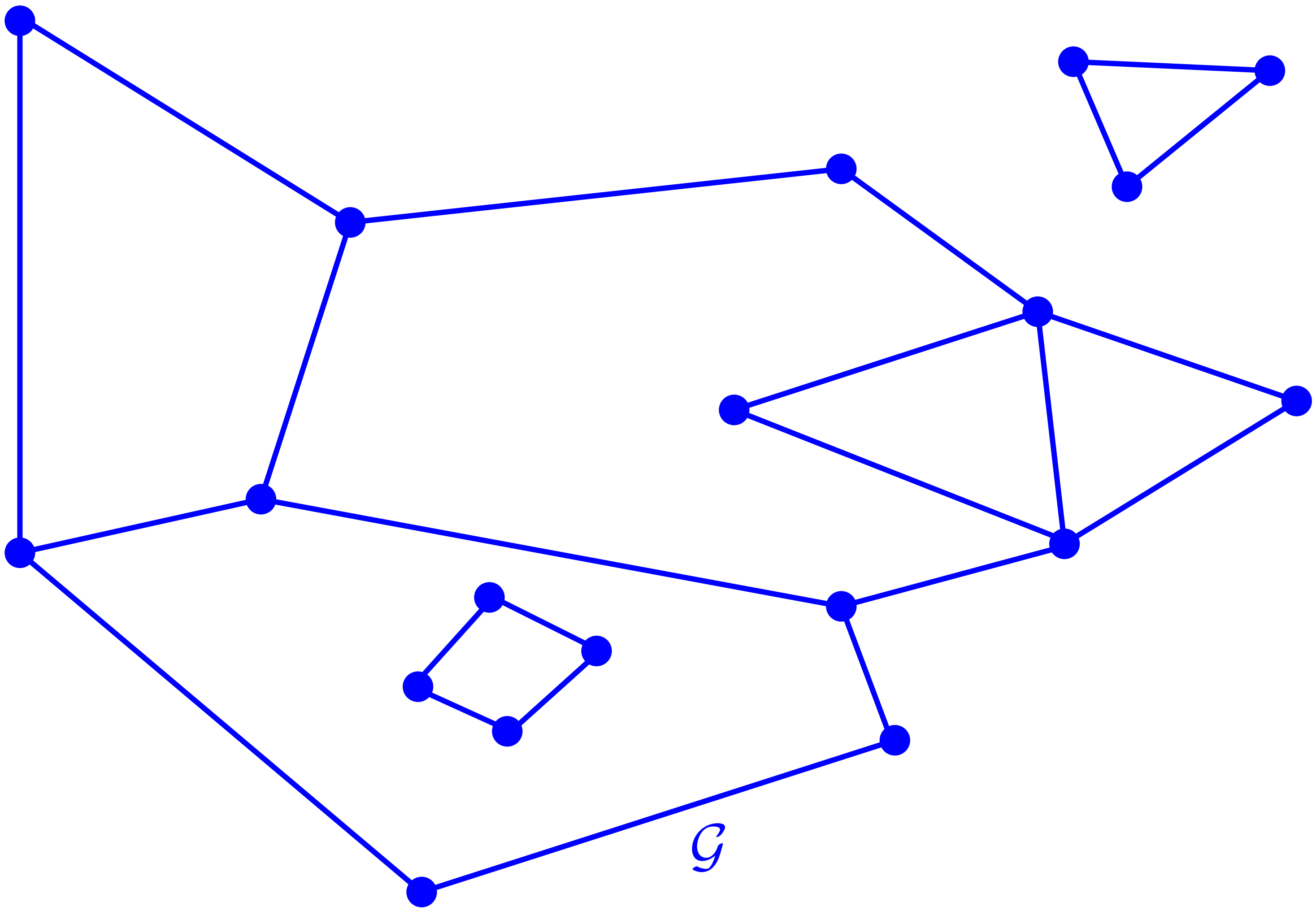
- knowing the Voronoi diagram, we can compute the convex hull in $O(n)$ time
- so computing a Voronoi diagram takes $\Omega(n \log n)$ time
- next lecture: an optimal $O(n \log n)$ time randomized algorithm
- there is also a deterministic $O(n \log n)$ time algorithm
  - plane–sweep algorithm
  - see Lecture 16 of D. Mount or Lecture 7 in textbook
  - you do not need to read it for CS4235

Further remarks

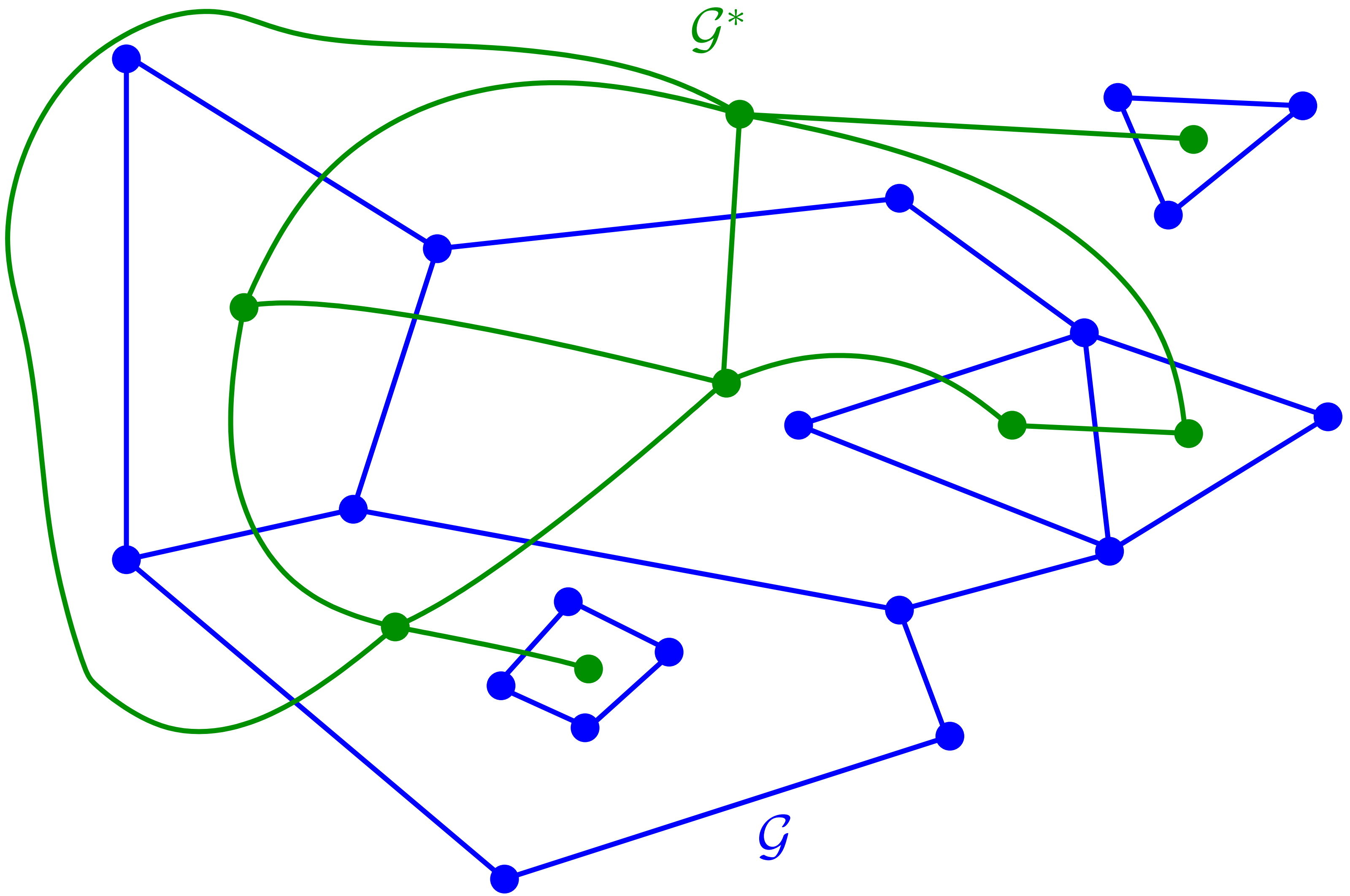
- sites need not be points: one can define in the same way the Voronoi diagram of a set of line segments, or any shape
- we can also use different distance functions
- the Voronoi diagram can also be defined in $\mathbb{R}^d$ for any d
  - but it has size $\Theta\left(n^{\lceil d/2 \rceil}\right)$
  - so it is only useful when d is small (say, at most 4)
  - active research line: approximate Voronoi diagrams with smaller size

Dual of a planar graph

Example



Example

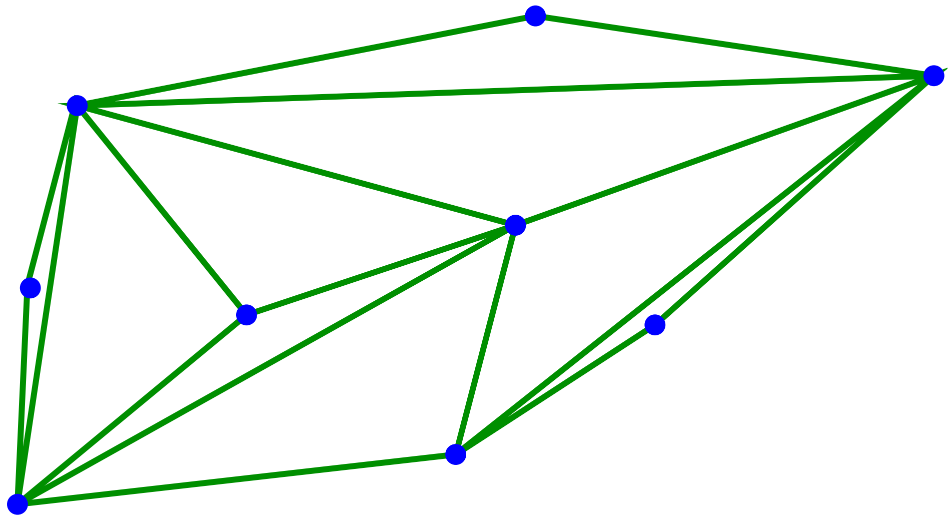


Definition

- let $\mathcal{G}$ be a graph
- dual graph $\mathcal{G}^*$
  - each face f of $\mathcal{G}$ corresponds to a vertex f^* of $\mathcal{G}^*$
  - (f^*, g^*) is an edge of $\mathcal{G}^*$ iff f and g are adjacent in $\mathcal{G}$
- property: the dual of a planar graph is planar too
- What is the dual of a Voronoi diagram?

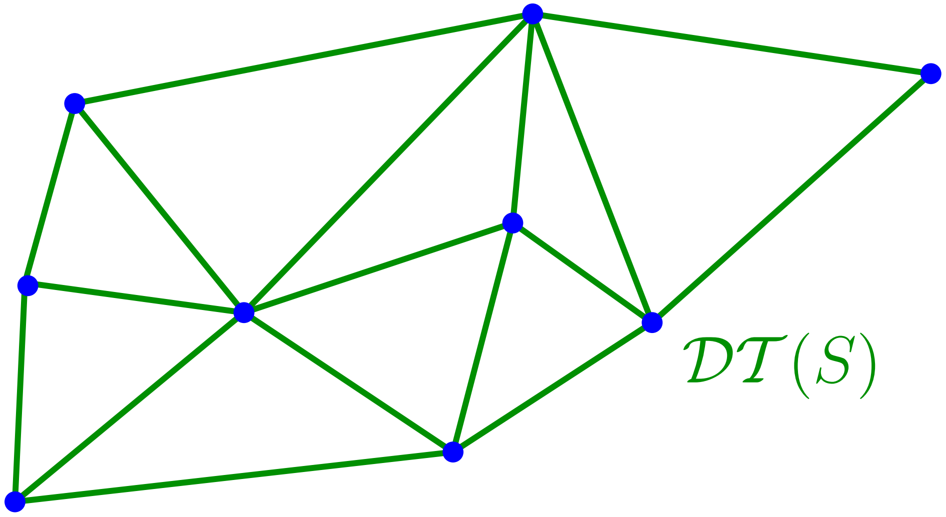
Delaunay triangulation

Triangulation of a set of points



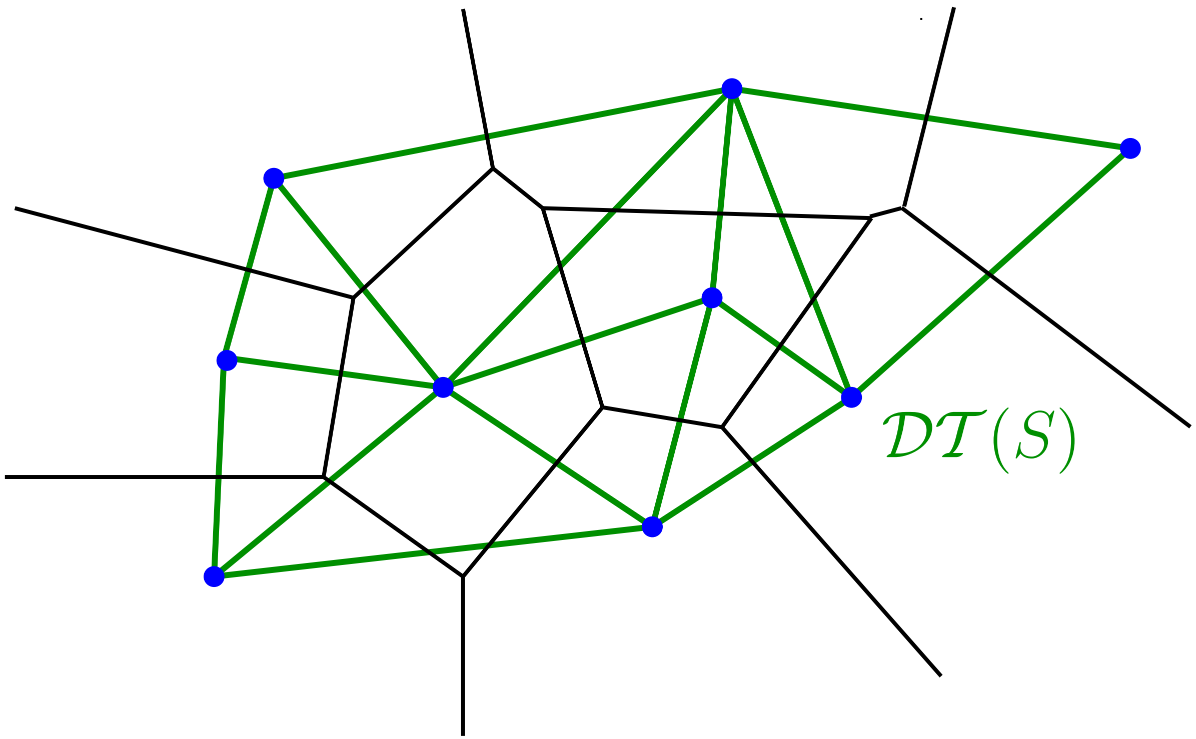
- we are given a set S of n points in $\mathbb{R}^2$
- we want to find a planar map with set of vertices S, where $\mathcal{CH}(S)$ is partitioned into triangles
- this is called a *triangulation* of S

The Delaunay triangulation



- the *Delaunay triangulation* of the same set
- looks nicer
- has many interesting properties

Definition



Definition

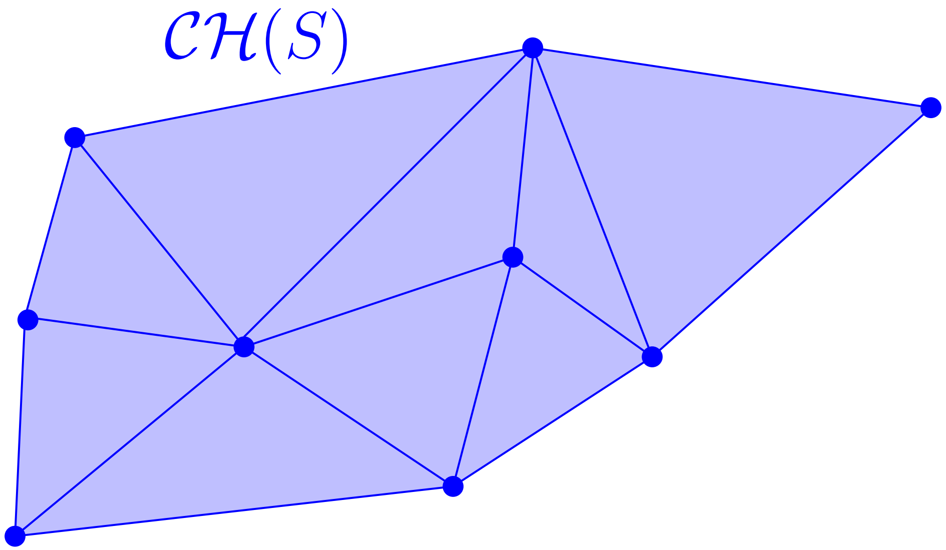
- let S be a set of n points in $\mathbb{R}^2$
- S is in general position: no 4 points are cocircular
- the *Delaunay triangulation* $\mathcal{DT}(S)$ of S is the embedding of the dual graph of the Voronoi diagram of S where
  - the vertices are the sites: $\forall i,\ \mathcal{V}(s_i)^* = s_i$
  - the edges of $\mathcal{DT}(S)$ are straight line segments

Remarks

- $\mathcal{DT}(S)$: is it well defined?
- we need to prove that
  - edges do not intersect (= it is a PSLG)
    - left as an exercise
  - faces are triangles
    - the number of edges in a face of $\mathcal{DT}(S)$ is the degree of the corresponding Voronoi vertex
    - general position assumption implies that Voronoi vertices have degree 3

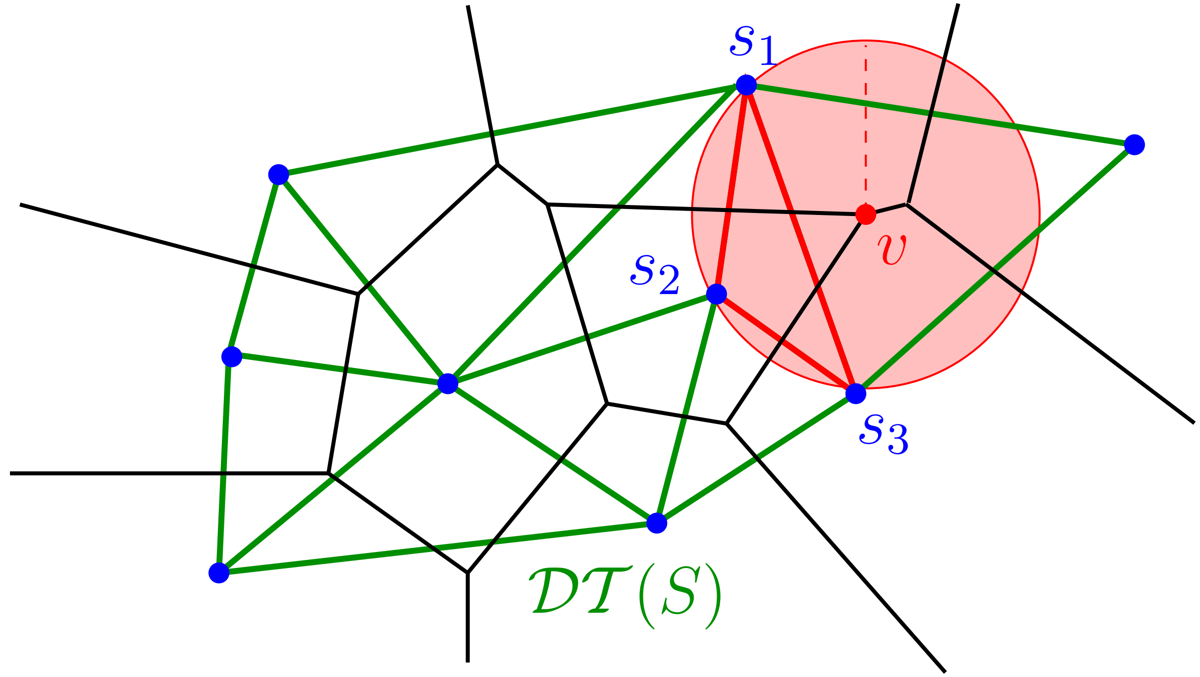
Convex hull

- the convex hull of S is the complement of the unbounded face of $\mathcal{DT}(S)$



Circumcircle property

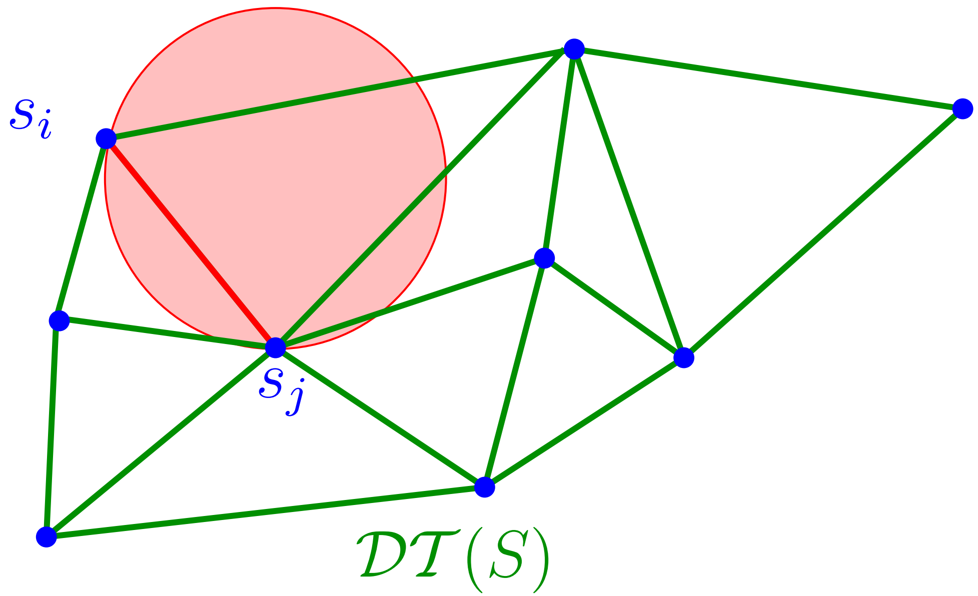
- the circumcircle of any triangle in $\mathcal{DT}(S)$ is empty (contains no site in its interior)



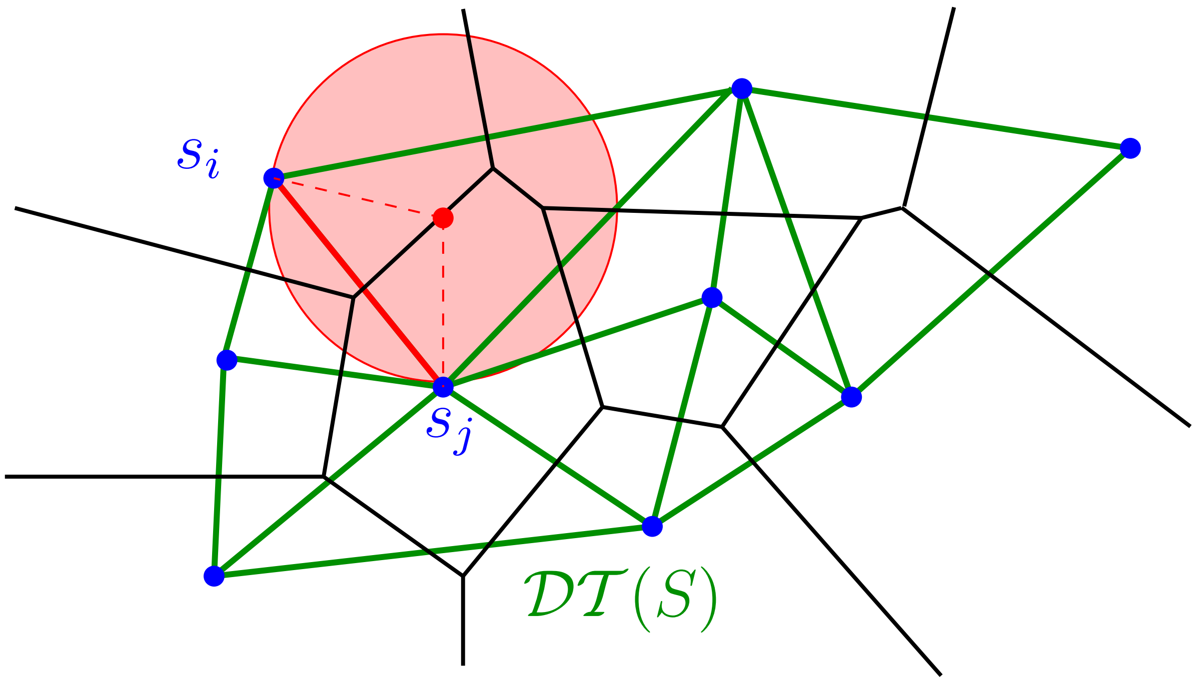
- proof: let $s_1 s_2 s_3$ be a triangle in $\mathcal{DT}(S)$, let v be the corresponding Voronoi vertex. Property of Voronoi vertices: the circle centered at v through $s_1 s_2 s_3$ is empty

Empty circle property

- $\overline{s_i s_j}$ is an edge of $\mathcal{DT}(S)$ iff there is an empty circle through s_i and s_j

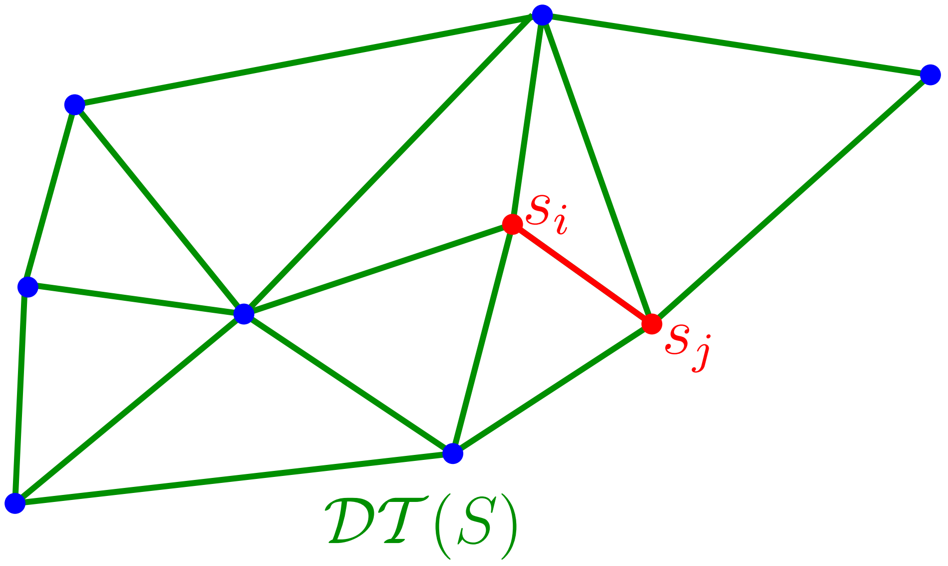


Proof (Empty circle property)

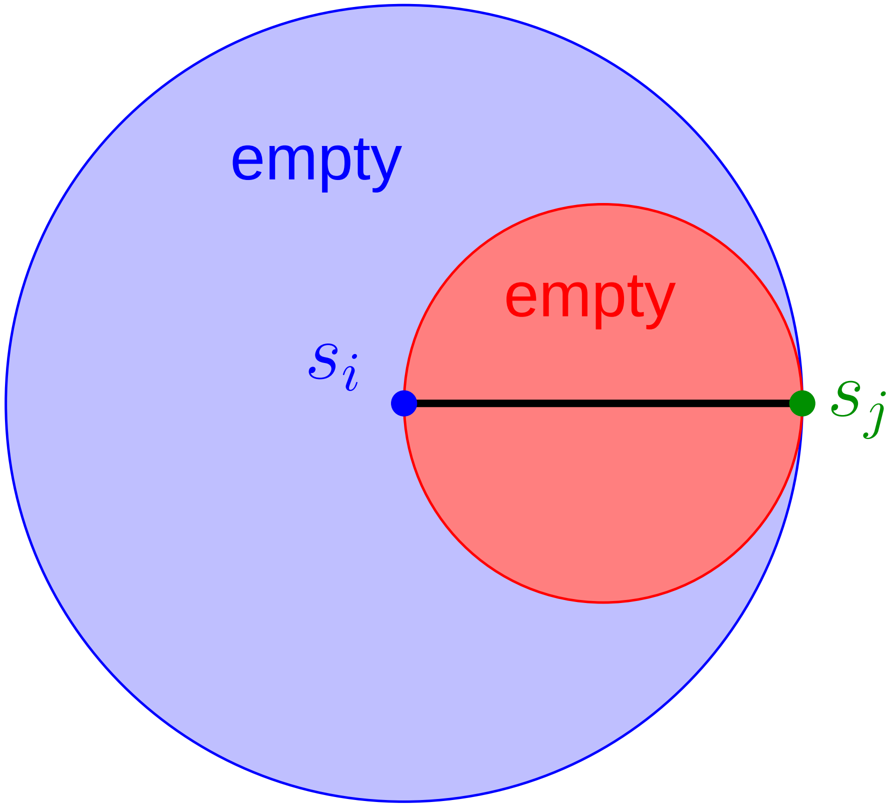


Closest pair property

- the closest two sites $s_i s_j$ are connected by an edge of $\mathcal{DT}(S)$



Proof (closest pair property)

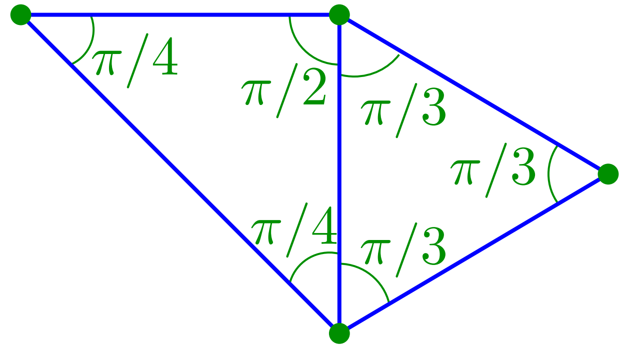


Euclidean minimum spanning tree

- Euclidean graph
  - set of vertices= S
  - for all $i \neq j$ there is an edge between s_i and s_j with weight $|s_i s_j|$
- Euclidean Minimum Spanning Tree: minimum spanning tree of the euclidean graph
- Property: the EMST is a subgraph of $\mathcal{DT}(S)$
- Corollary: it can be computed in $O(n \log n)$ time
- see D. Mount's notes pages 75–76

Angle sequence

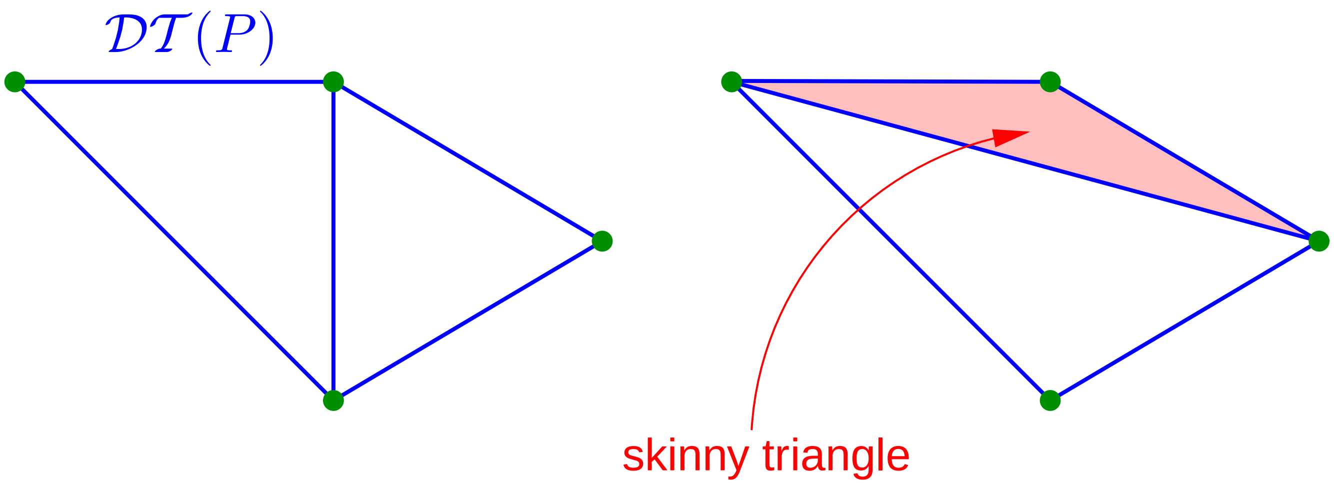
- let $\mathcal{T}$ be a triangulation of S
- angle sequence $\Theta(\mathcal{T})$: sequence of all the angles of the triangle of $\mathcal{T}$ in non–decreasing order
- example



- $\Theta(\mathcal{T}) = (\pi/4, \pi/4, \pi/3, \pi/3, \pi/3, \pi/2)$
- comparison: let $\mathcal{T}$ and $\mathcal{T}'$ be two triangulations of S
- we compare $\Theta(\mathcal{T})$ and $\Theta(\mathcal{T}')$ using lexicographic order
- example: $(1, 1, 3, 4, 5) < (1, 2, 4, 4, 4)$

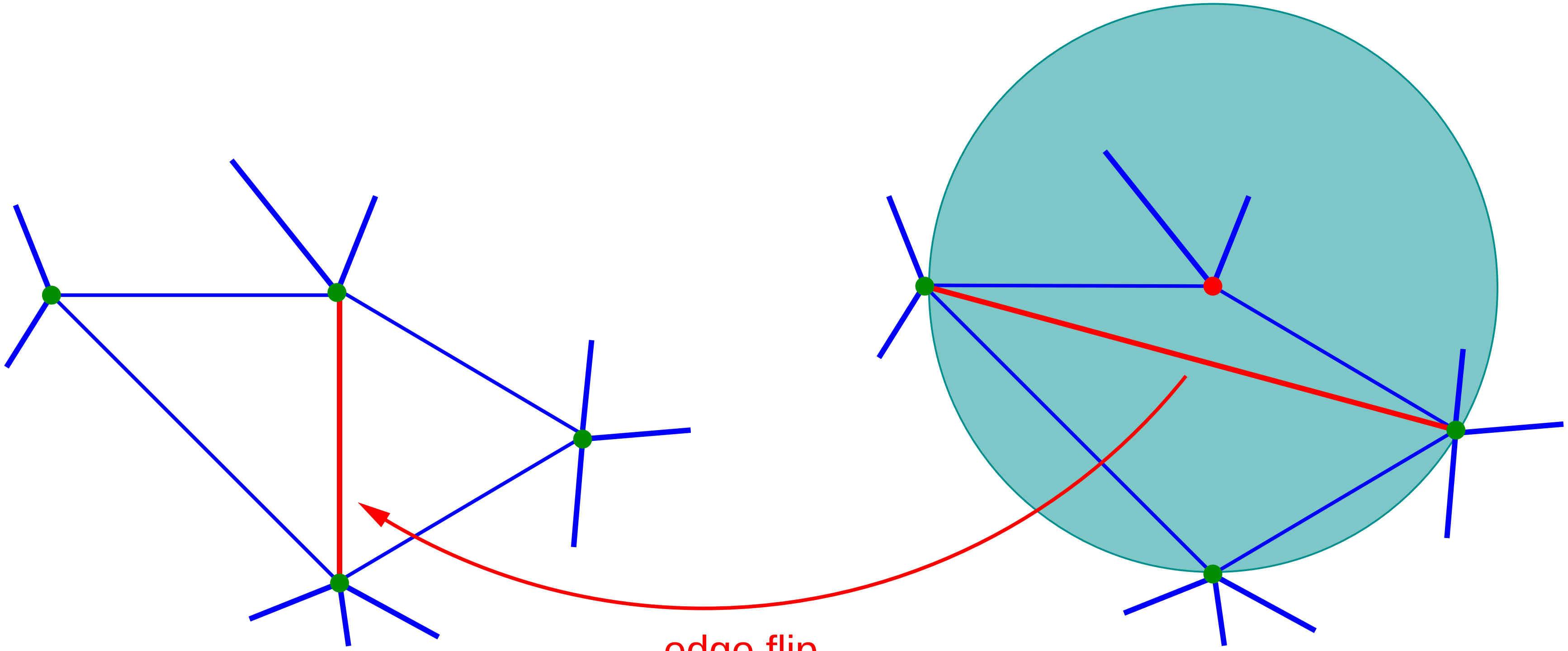
Optimality of $\mathcal{DT}(S)$

- Theorem: the angle sequence of $\mathcal{DT}(S)$ is maximal among all triangulations of S
- in other words: the Delaunay triangulation maximizes the minimum angle
- intuition: avoids skinny triangles



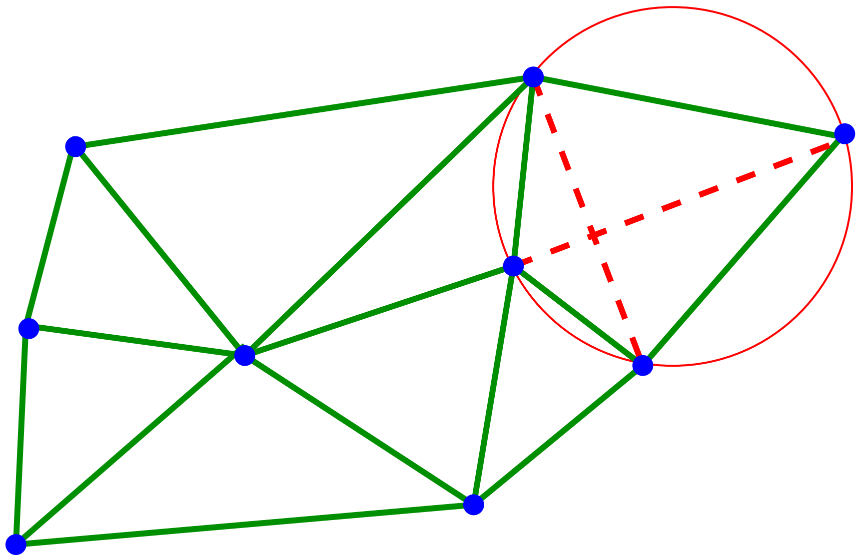
Proof

- idea:
  - flip edges to ensure the circumcircle property
  - it decreases the angle sequence



Degenerate cases

- several possible Delaunay triangulations
  - equally good in terms of angle sequence
- example:



- two possibilities

Applications

- generating good meshes
  http://www.cs.berkeley.edu/~jrs/mesh/
  - skinny triangles are bad in numerical analysis
- statistics: natural neighbor interpolation
- textbook example: height interpolation
- shape reconstruction
- ...

Conclusion

- next lecture: an $O(n \log n)$ time RIC of the Delaunay triangulation
- from the Delaunay triangulation, we can obtain the Voronoi diagram in $O(n)$ time (how?)
- the converse is also true, so these two problems are equivalent in an algorithmic point of view
- there are $O(n \log n)$ time deterministic algorithms
  - divide and conquer
  - plane sweep (D. Mount lecture 16)
  - reduction to 3D convex hull (D. Mount lecture 28)
  - not presented in CS 4235
  - in practice, RIC is used