IMPA 2013 Postdoc Course
Convex Hulls and Point Location

Marc Pouget
VEGAS research team,
INRIA Nancy, France

Who am I?

Nancy is 1:30 hours from Paris

My research team at INRIA:

- Non-linear computational geometry
- 3d visibility and line geometry

People: L. Dupont, S. Lazard, S. Petitjean, X. Goaoc, G. Moroz

My favorite topics:

- Approximation of curvatures on discrete surfaces
- Topology and geometry of curves and surfaces with algebraic/numerical tools

Who am I?

Nancy is 1:30 hours from Paris

My research team at INRIA:

- Non-linear computational geometry
- 3d visibility and line geometry

People: L. Dupont, S. Lazard, S. Petitjean, X. Goaoc, G. Moroz

My favorite topics:

- Approximation of curvatures on discrete surfaces
- Topology and geometry of curves and surfaces with algebraic/numerical tools

Want to join our team? INRIA Internship/Postdoc

REFERENCES

Algorithmic Geometry. J.D. Boissonnat and M. Yvinec.

Computational Geometry: Algorithm and Applications. M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf.

Thanks to:
J.D. Boissonnat
O. Devillers
M. Yvinec
X. Goaoc
...

CONVEX HULLS

- Define the problem
- Algorithms in 2d and 3d
- Combinatorics of polytopes
- Algorithms in nd
- Robustness issues
- Application to Voronoi/Delaunay

Transversal topics:

- CGAL (Computational Geometry Algorithm Library)
- Output sensitive analysis
- Randomized algorithm
- Expected complexity

POINT LOCATION

- Define the problem
- Naive slab decomposition
- Trapezoidal maps
- Hierarchy and walks in triangulations

Also:

- Sweep algorithm for segment intersections
- CGAL

INTRODUCTION TO COMPUTATIONAL GEOMETRY

CONVEX HULL: DEFINITION

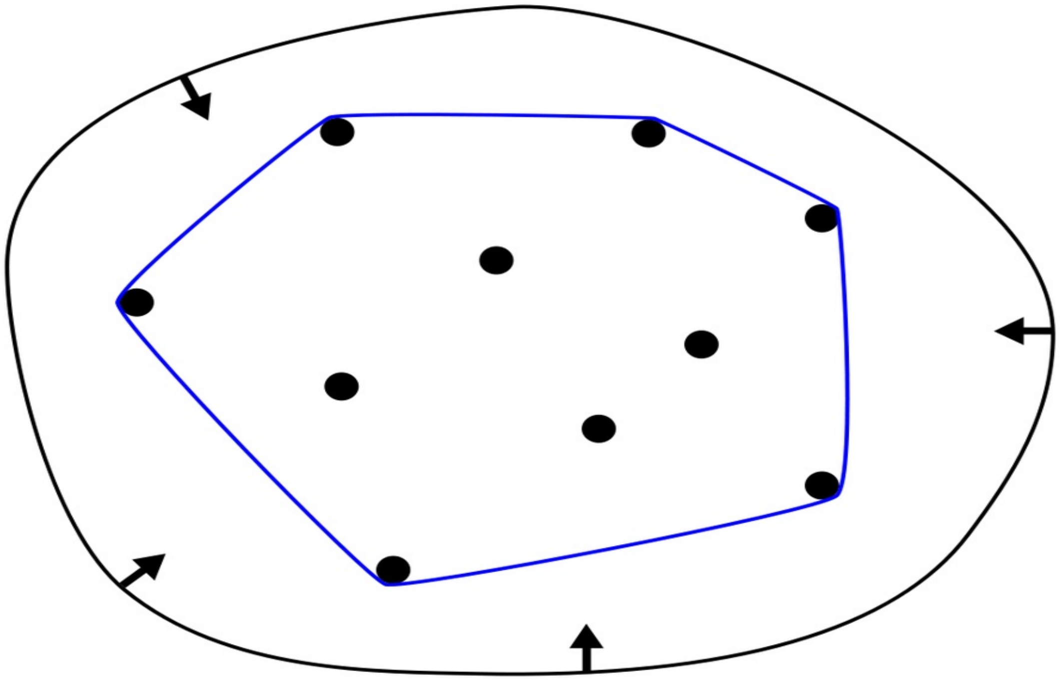
Convex set K: for any 2 points p, q of K, segment $[pq] \subset K$

$P = p_1, ..., p_n$ set of n points

$CH(P)$: Convex Hull of $P =$

1. smallest (for the volume) convex containing P
2. set of all convex combinations of points in P
3. intersection of all half-spaces containing P

$CH(P)$ is a polytope with vertices in P



What does mean computing $CH(P)$?

• 1st answer: give the vertices of the polytope $CH(P)$
i.e. a subset of points in P

• 2nd answer: give a combinatorial description of the polytope $CH(P)$

What does mean computing $CH(P)$?

• 1st answer: give the vertices of the polytope $CH(P)$ i.e. a subset of points in P

• 2nd answer: give a combinatorial description of the polytope $CH(P)$

  1D: find the smallest and largest points
  2D: store CH vertices in circular order
  3D: store a planar (spherical) graph with CH points as vertices

What does mean computing $CH(P)$?

• 1st answer: give the vertices of the polytope $CH(P)$
i.e. a subset of points in P

• 2nd answer: give a combinatorial description of the polytope $CH(P)$

> 1D: find the smallest and largest points
> 2D: store CH vertices in circular order
> 3D: store a planar (spherical) graph with CH points as vertices

• Up to 3D: the complexity of a polytope (nb of edges, faces) is linear wrt the number of its vertices.

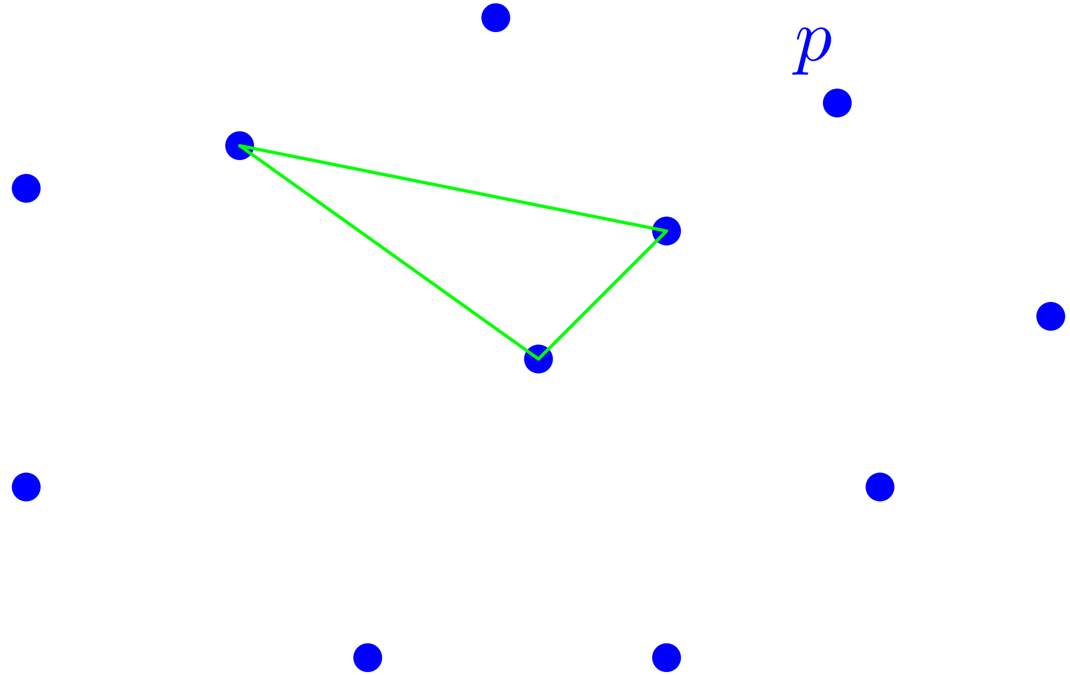
• Higher D: the complexity of a polytope (faces of all dimensions) is much higher

NAIVE ALGORITHM in 2D

- Extreme points

For each point $p \in P$

Check whether p is in the interior of a triangle formed by 3 other points in P

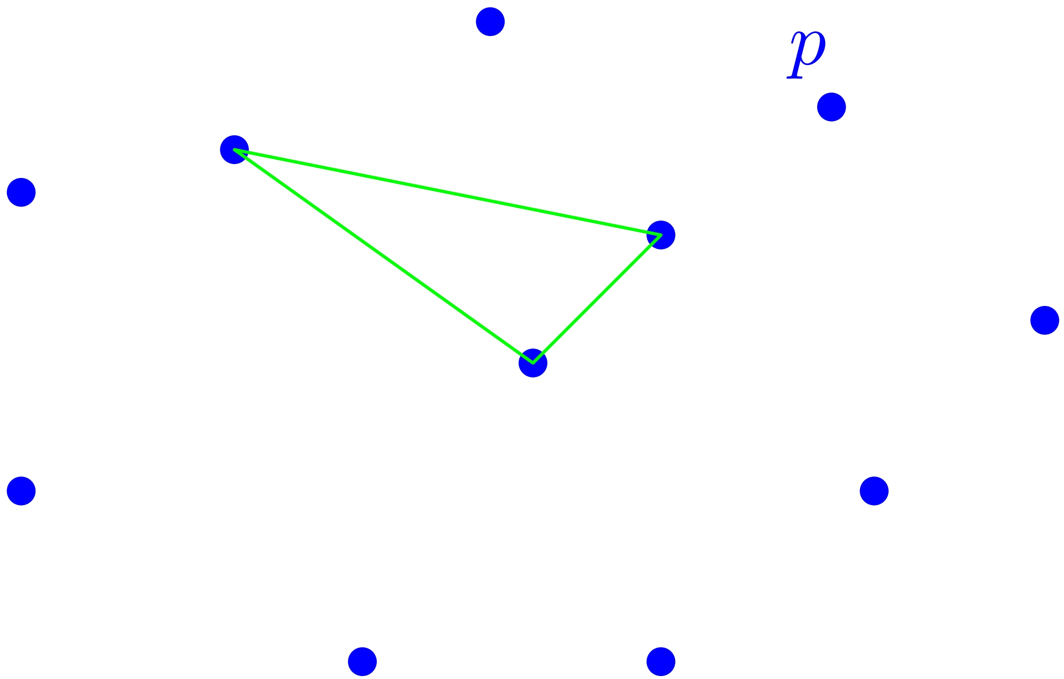


NAIVE ALGORITHM in 2D

- Extreme points

For each point $p \in P$

Check whether p is in the interior of a triangle formed by 3 other points in P



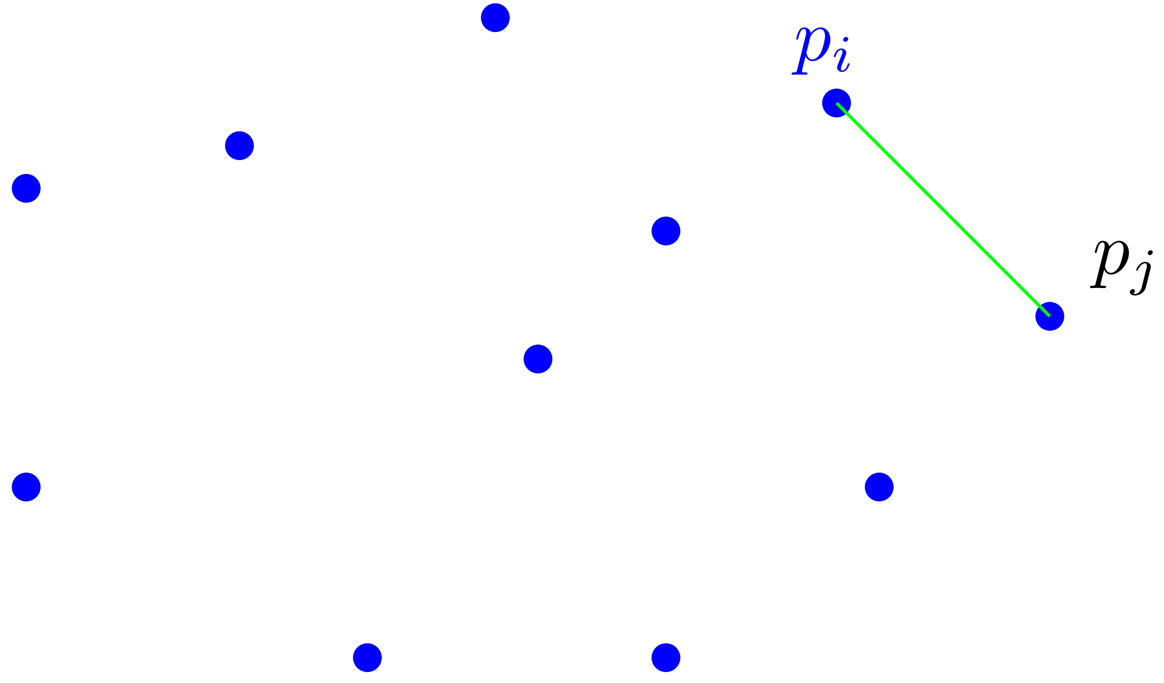
Complexity: $O(n^4)$

NAIVE ALGORITHM in 2D

- Extreme edges

For each edge $p_i p_j$

Check whether all other points are on one side

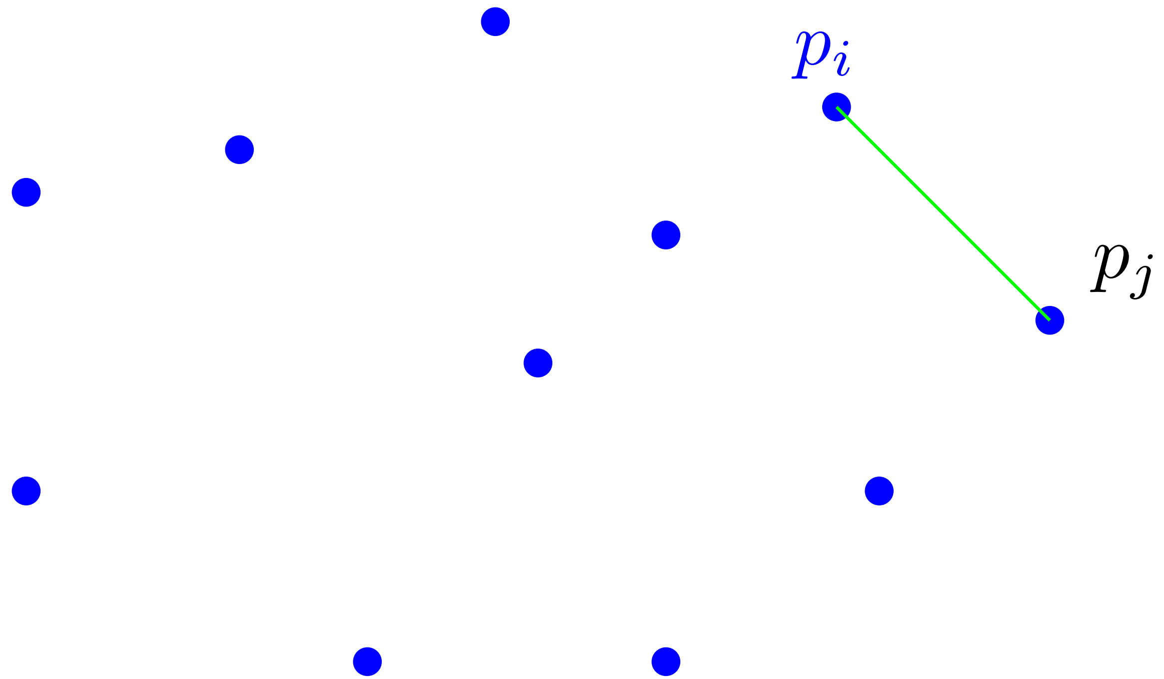


NAIVE ALGORITHM in 2D

- Extreme edges

For each edge $p_i p_j$

Check whether all other points are on one side

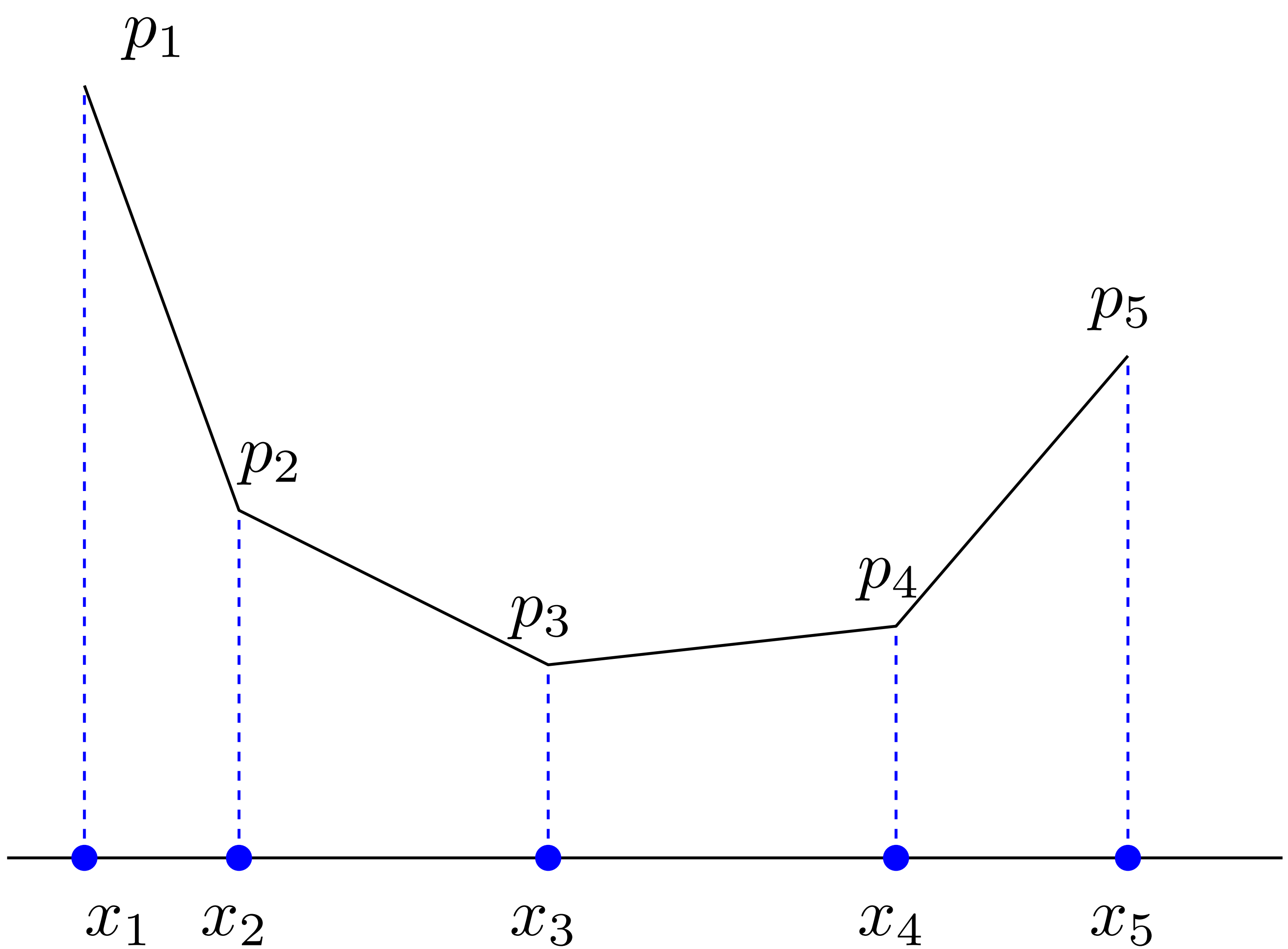


Complexity: $O(n^3)$

LOWER BOUND

Convex hull is at least as hard as sorting

$x_i \to p_i = (x_i, x_i^2)$

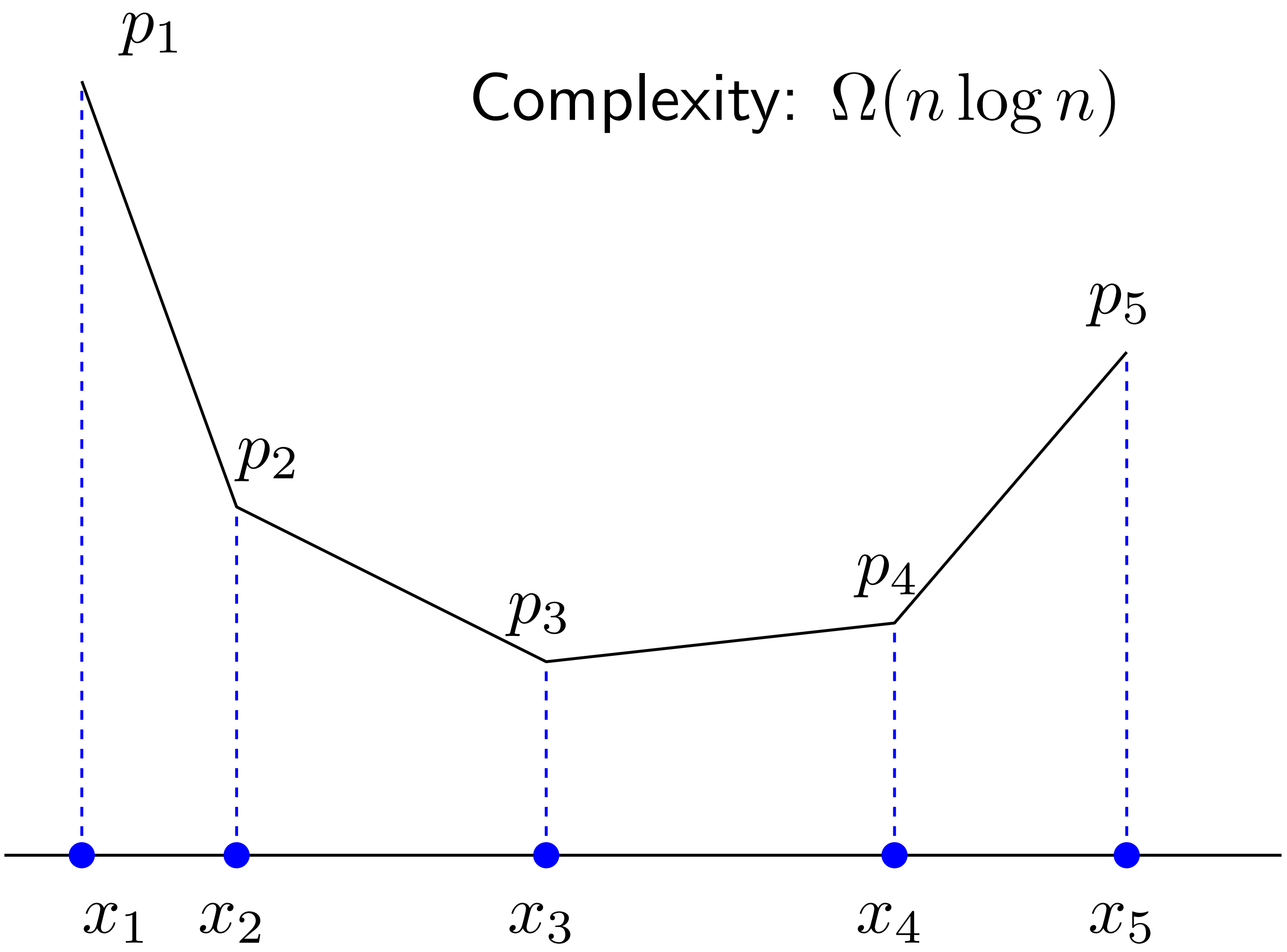


LOWER BOUND

Convex hull is at least as hard as sorting

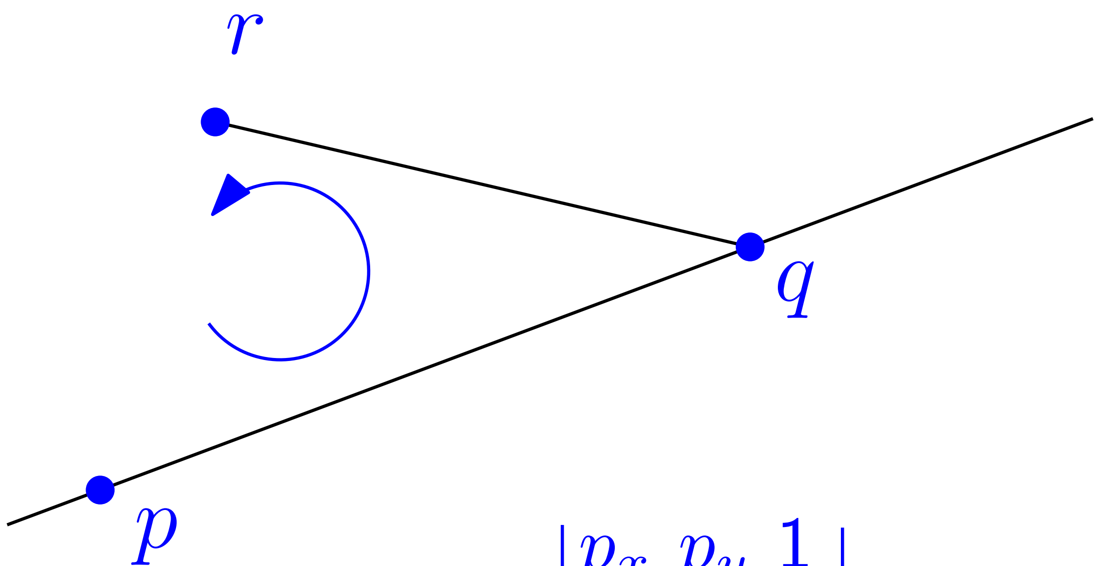
$x_i \to p_i = (x_i, x_i^2)$

Complexity: $\Omega(n \log n)$



ROBUSTNESS ISSUES

Orientation of 2D points

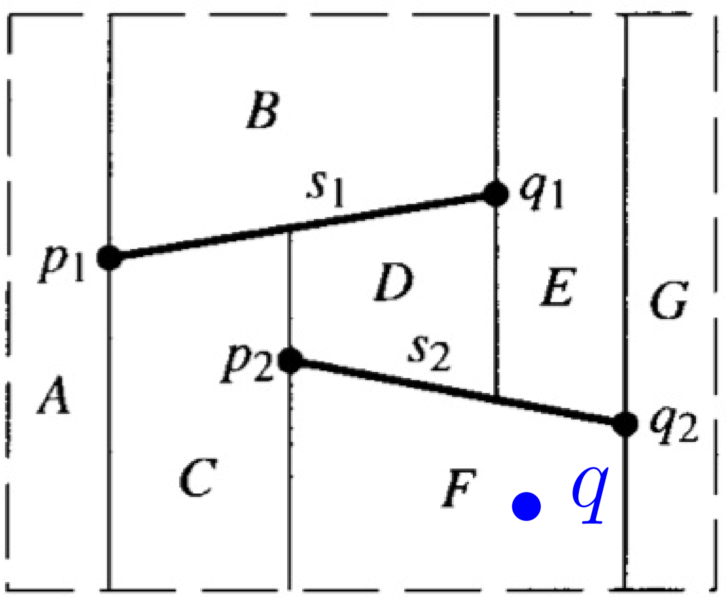


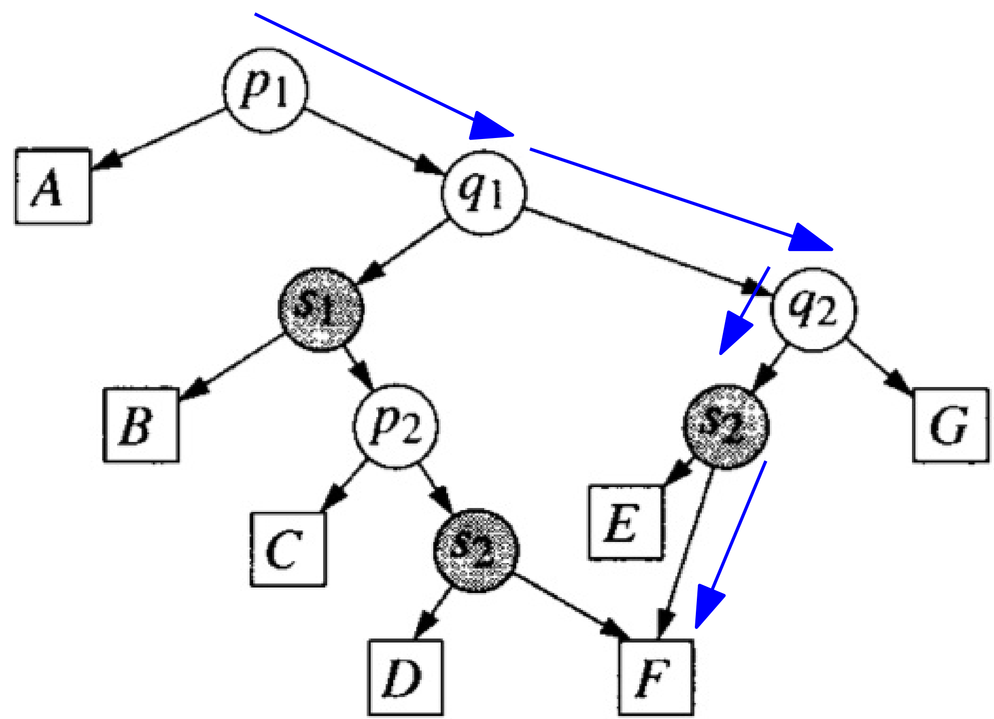
$$Orientation(p,q,r) = sign(\det \begin{vmatrix} p_x & p_y & 1 \\ q_x & q_y & 1 \\ r_x & r_y & 1 \end{vmatrix})$$

$$= sign((q_x - p_x)(r_y - p_y) - (q_y - p_y)(r_x - p_x))$$

Predicate for CH: decide the sign of a polynomial of degree 2 wrt the coordinates of the input points

TRAPEZOIDAL MAP & SEARCH STRUCTURE





TRAPEZOIDAL MAP & SEARCH STRUCTURE

Algorithm TRAPEZOIDALMAP(S)
Input. A set S of n non-crossing line segments.
Output. The trapezoidal map $\mathcal{T}(S)$ and a search structure $\mathcal{D}$ for $\mathcal{T}(S)$ in a bounding box.
1. Determine a bounding box R that contains all segments of S, and initialize the trapezoidal map structure $\mathcal{T}$ and search structure $\mathcal{D}$ for it.
2. Compute a random permutation $s_1, s_2, \ldots, s_n$ of the elements of S.

3. **for** $i \leftarrow 1$ **to** n
4. **do** Find the set $\Delta_0, \Delta_1, \ldots, \Delta_k$ of trapezoids in $\mathcal{T}$ properly intersected by s_i.
5. Remove $\Delta_0, \Delta_1, \ldots, \Delta_k$ from $\mathcal{T}$ and replace them by the new trapezoids that appear because of the insertion of s_i.
6. Remove the leaves for $\Delta_0, \Delta_1, \ldots, \Delta_k$ from $\mathcal{D}$, and create leaves for the new trapezoids. Link the new leaves to the existing inner nodes by adding some new inner nodes, as explained below.