Using orientation tests to solve basic problems on polygons

Vera Sacristán

Computational Geometry
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Intersection test line - polygon

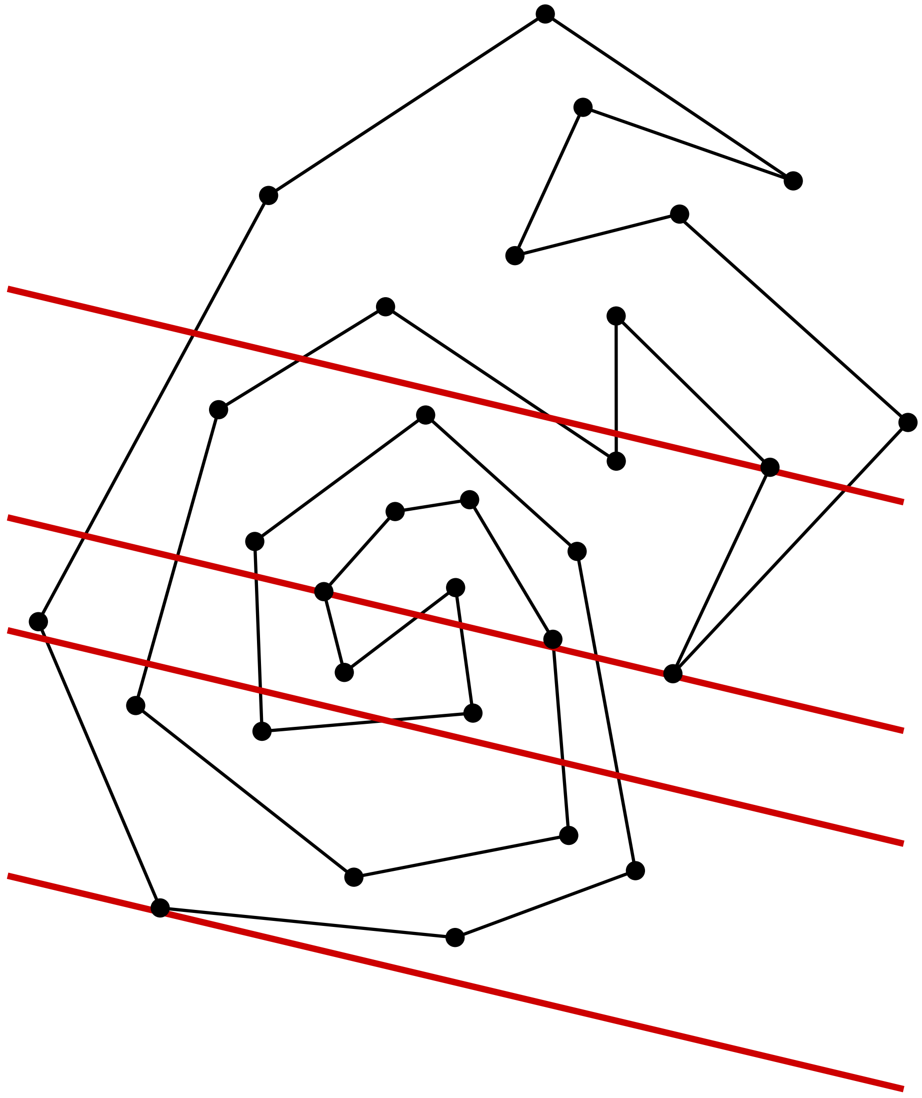
Input:

ℓ: a line (through p and q)

P: a polygon (with vertices $p_1, p_2, \ldots, p_m$)

Yes/No they intersect.

If they do, the edges of P intersecting ℓ

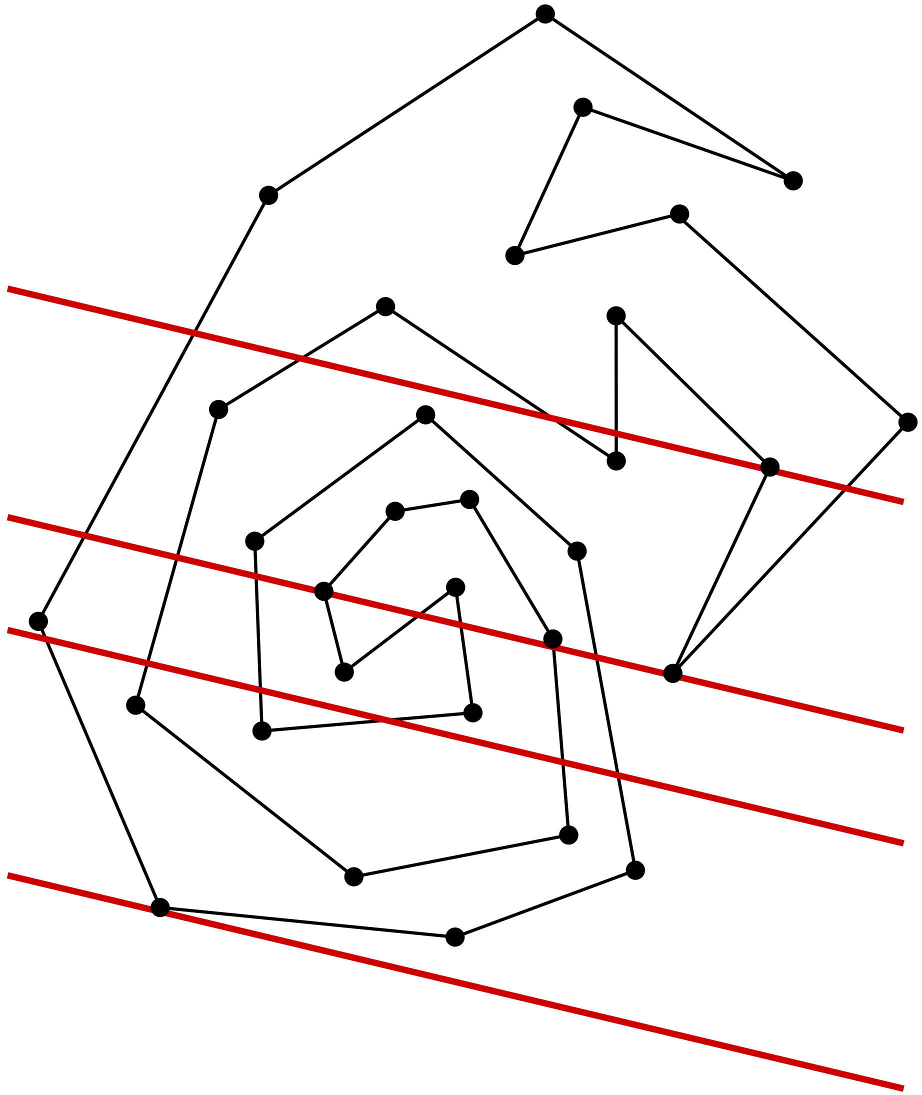


Intersection test line - polygon

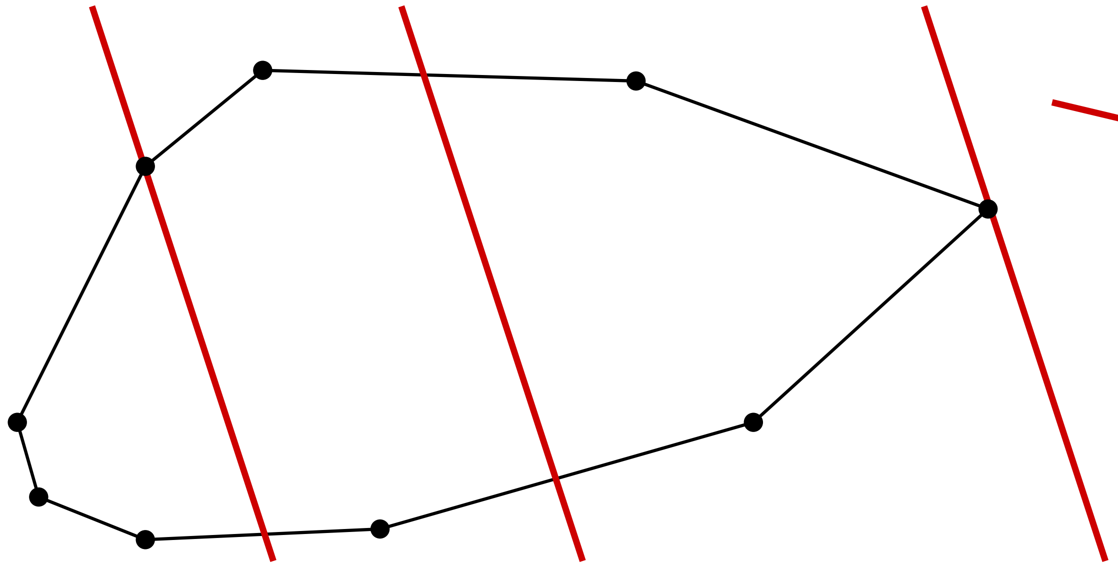
Input:

ℓ: a line (through p and q)
P: a polygon (with vertices $p_1, p_2, \ldots, p_m$)

Yes/No they intersect.
If they do, the edges of P intersecting ℓ



What if the polygon is convex?



Point in polygon test

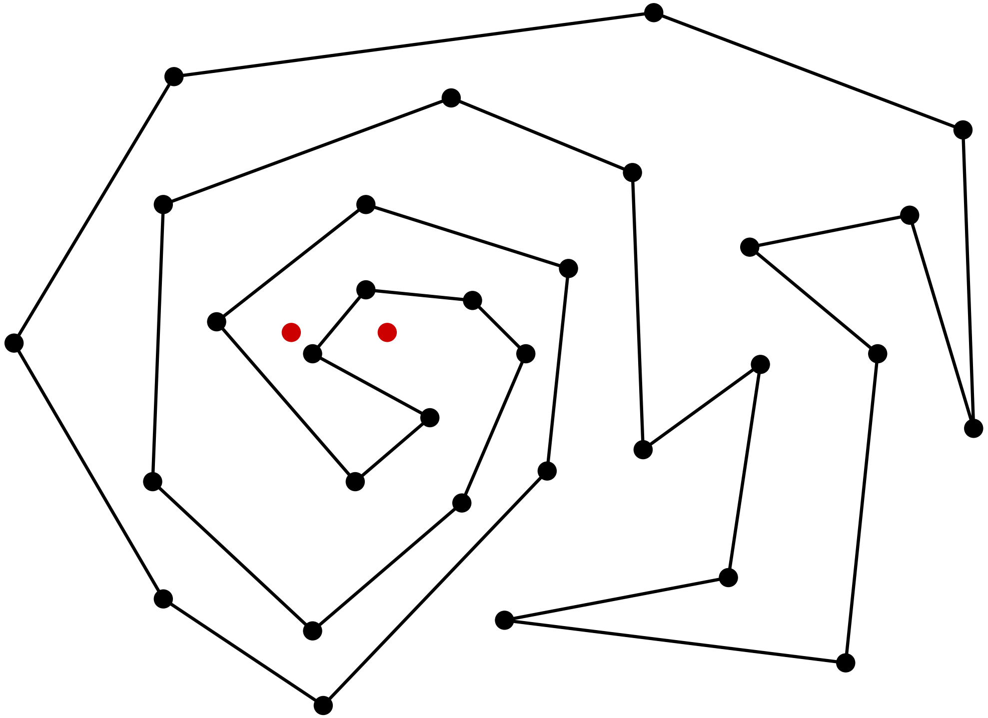
Input:

A polygon $p_1, p_2, \ldots, p_n$

A query point q

Output:

Yes/No $q \in P$



Point in polygon test

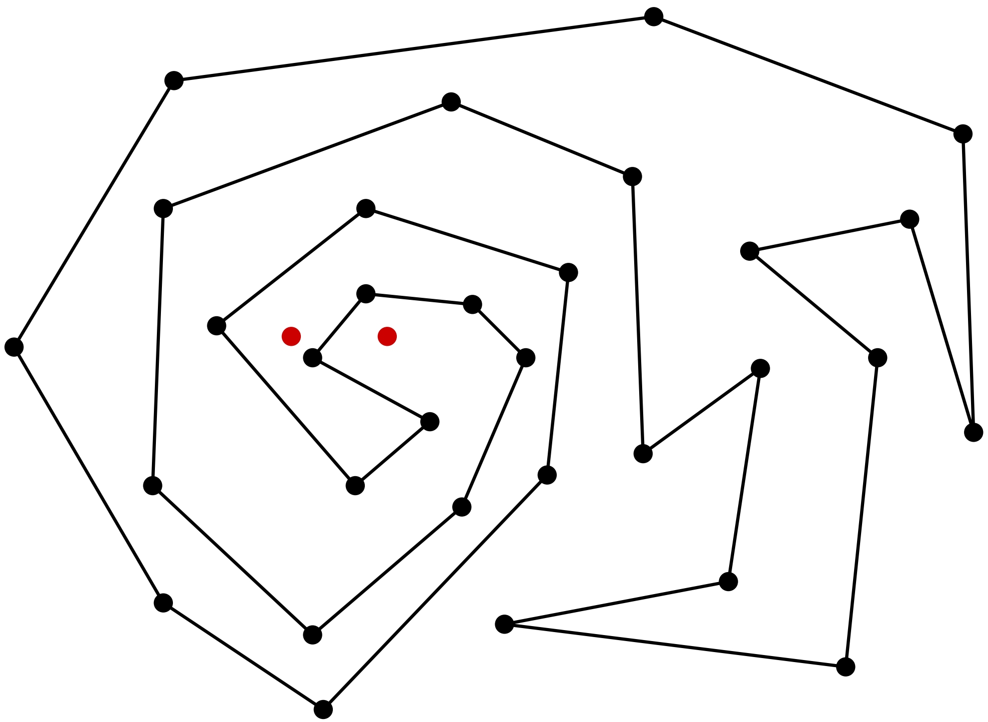
Input:

A polygon $p_1, p_2, \ldots, p_n$

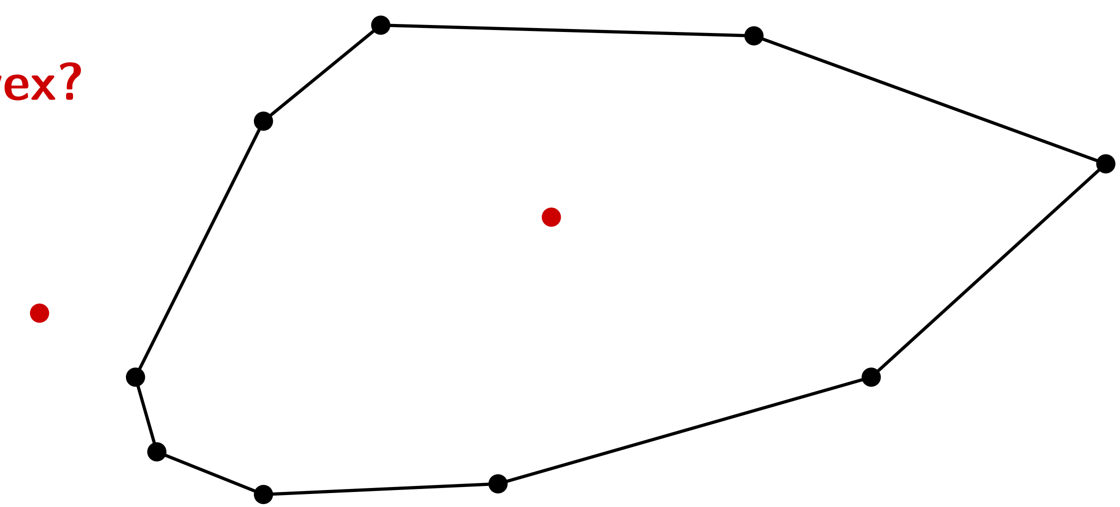
A query point q

Output:

Yes/No $q \in P$



What if the polygon is convex?



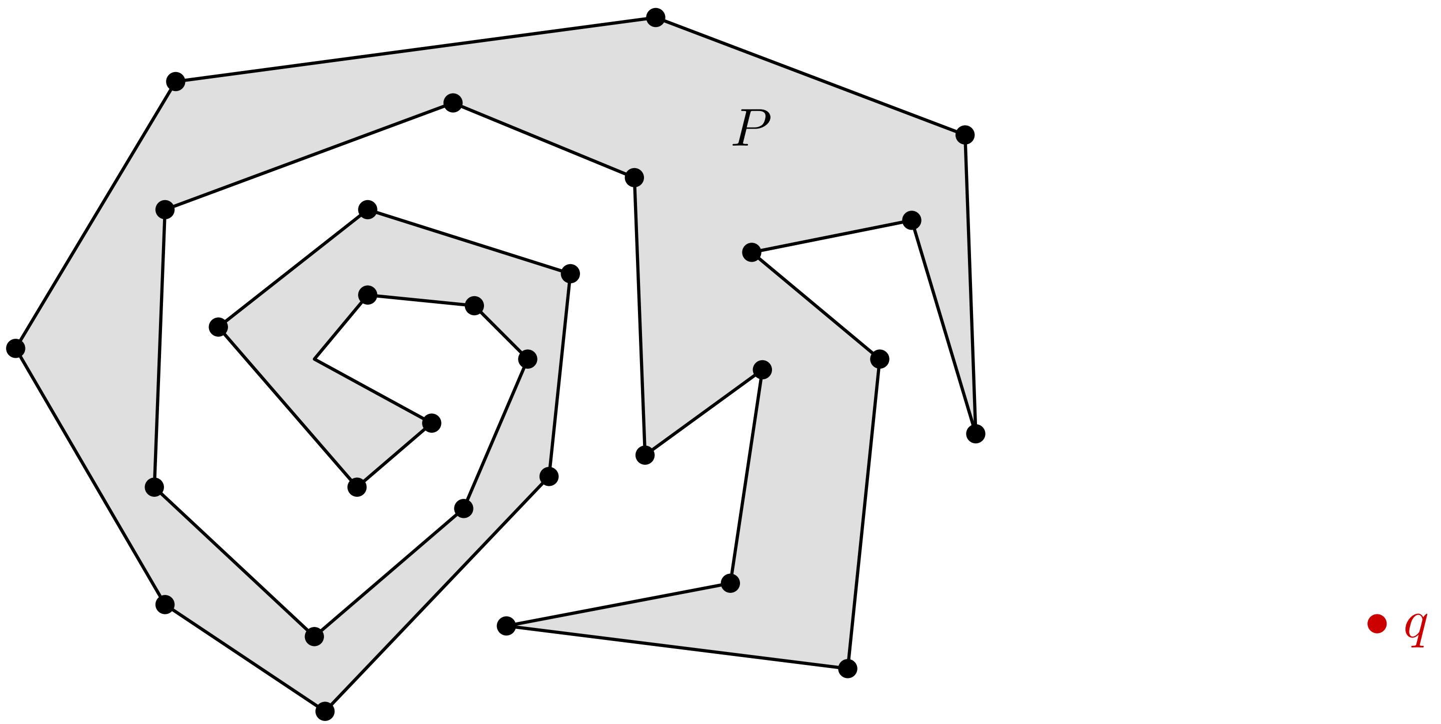
Supporting lines point - polygon

Input:

A polygon P with vertices $p_1, p_2, \ldots, p_n$
A point q not belonging to the convex hull of P

Output:

Lines through q and P that leave all of P to one side



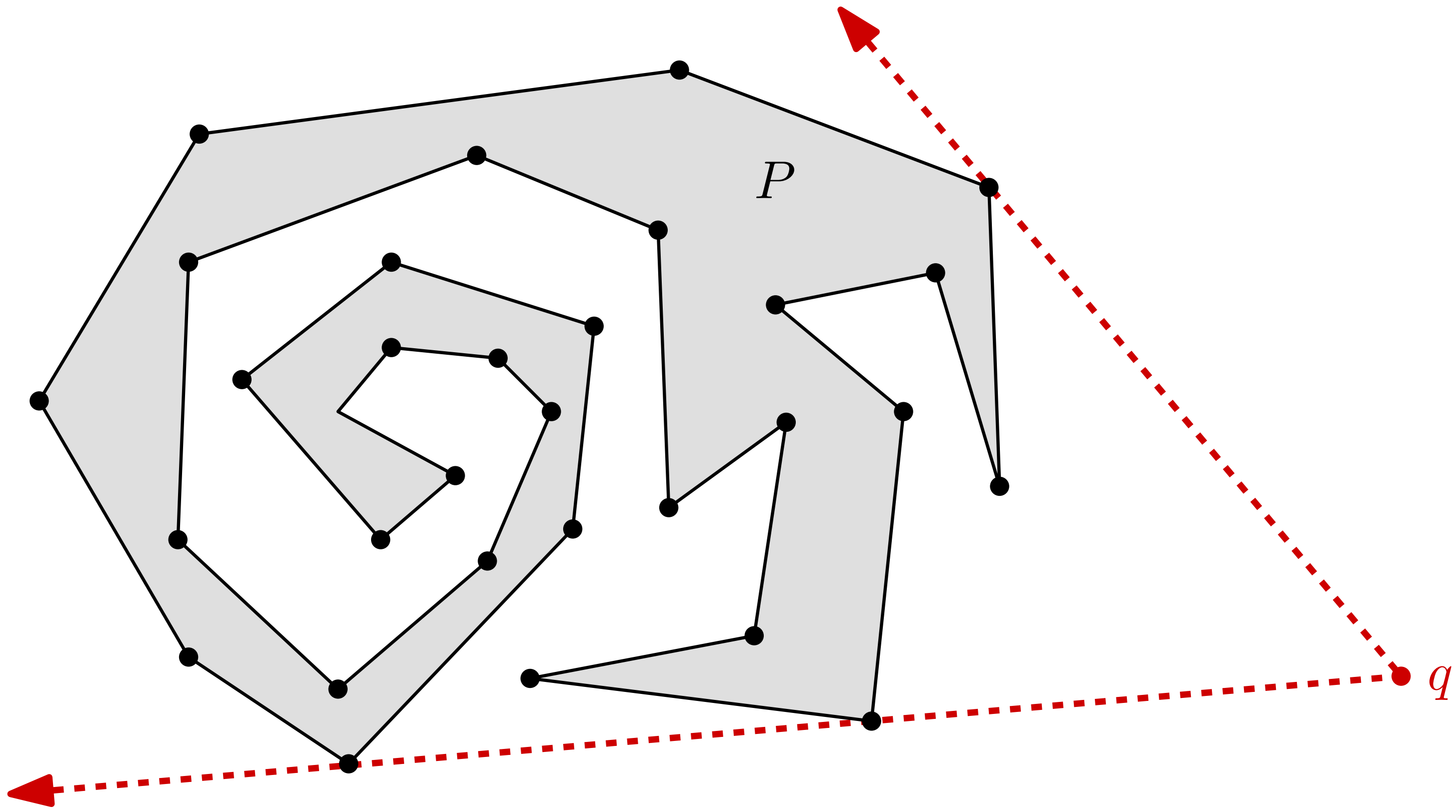
Supporting lines point - polygon

Input:

A polygon P with vertices $p_1, p_2, \ldots, p_n$
A point q not belonging to the convex hull of P

Output:

Lines through q and P that leave all of P to one side



Supporting lines point - polygon

Input:

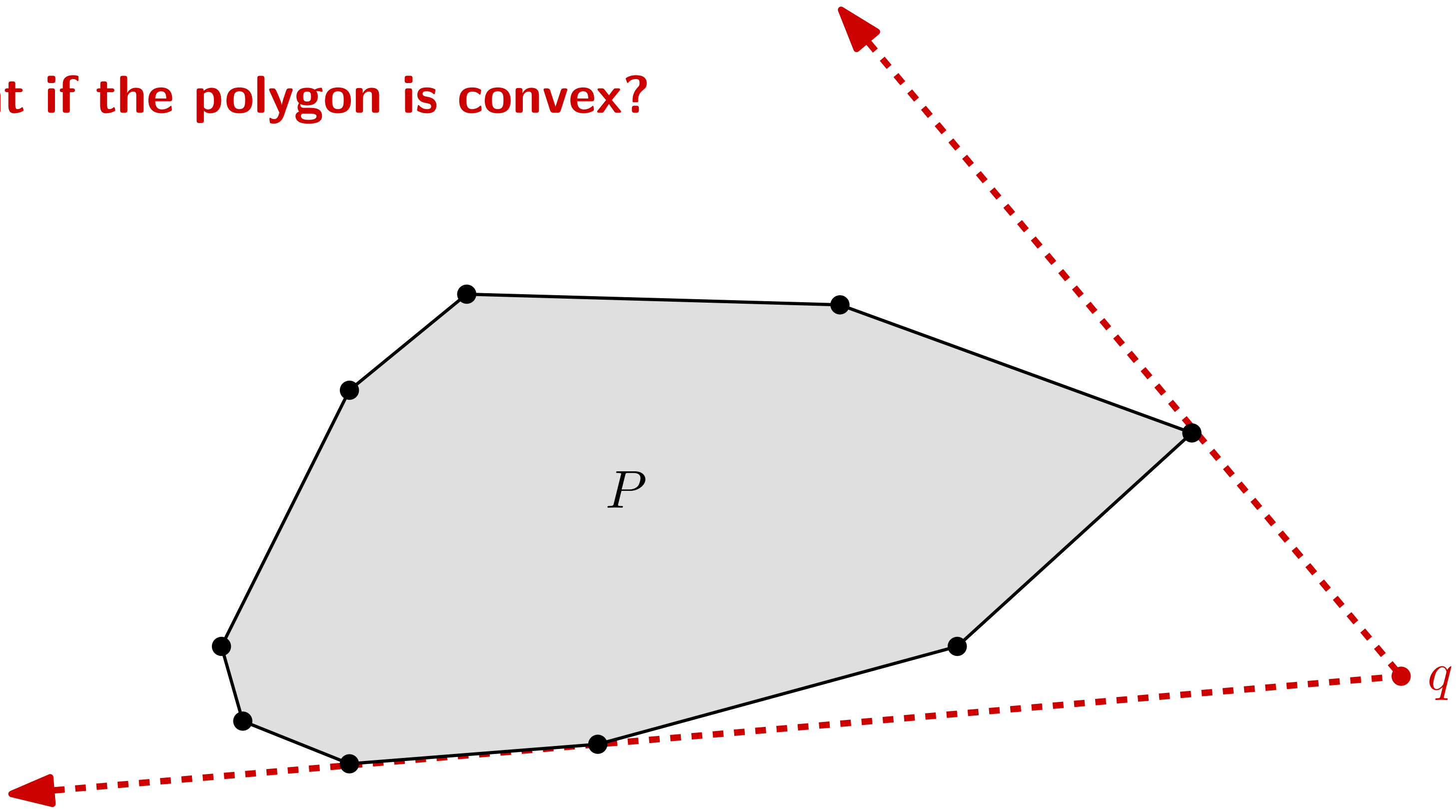
A polygon P with vertices $p_1, p_2, \ldots, p_n$

A point q not belonging to the convex hull of P

Output:

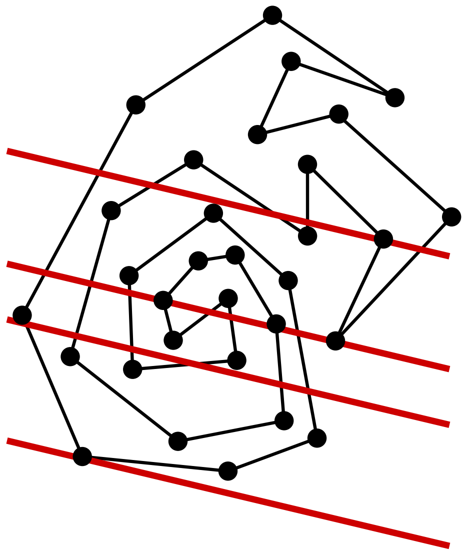
Lines through q and P that leave all of P to one side

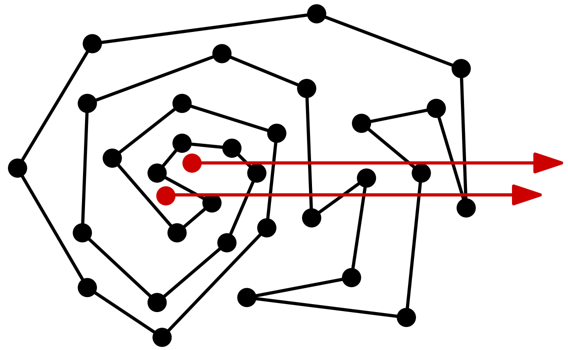
What if the polygon is convex?

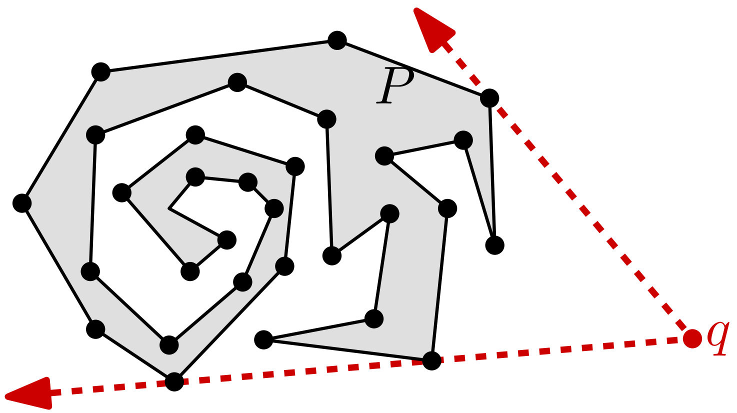


How did we prove the correctness of our solutions?

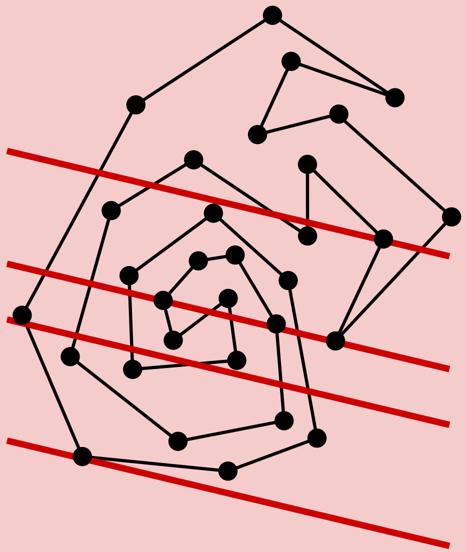
How did we prove the correctness of our solutions?



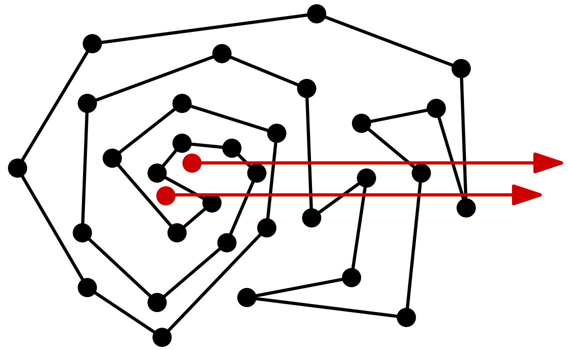


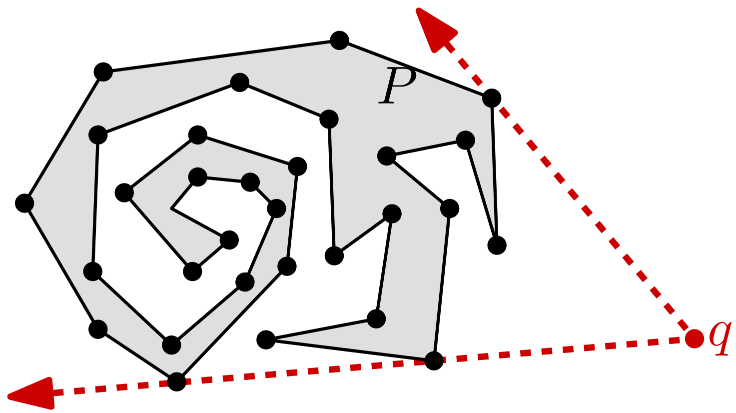


How did we prove the correctness of our solutions?

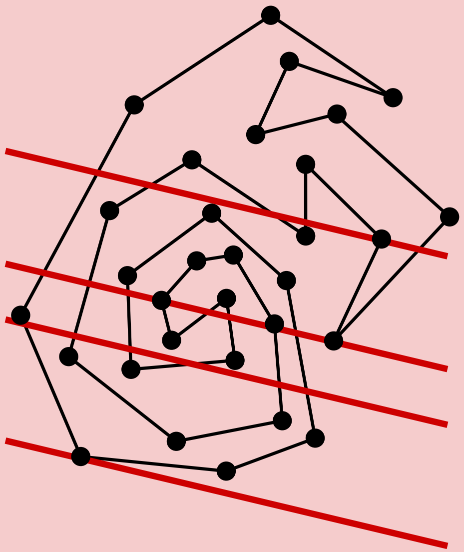


Geometric property:
No particular one.





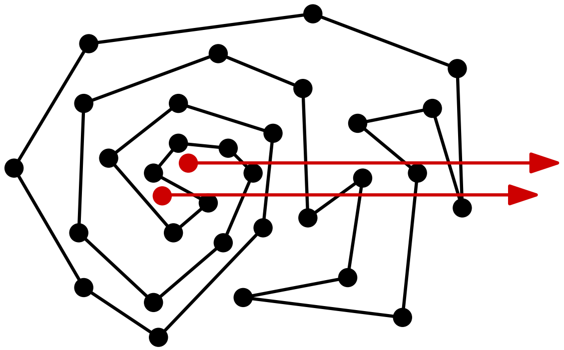
How did we prove the correctness of our solutions?

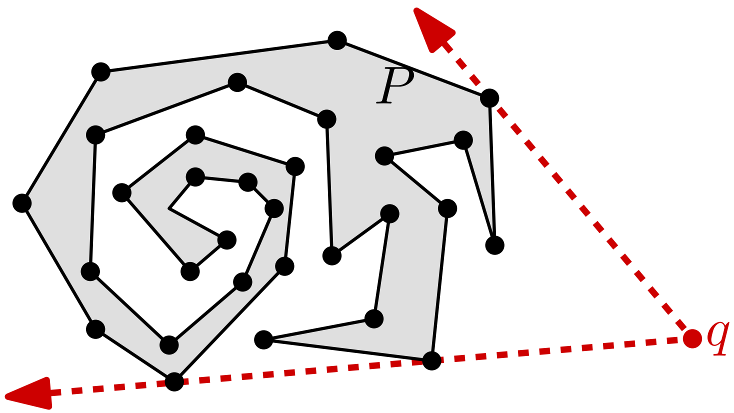


Geometric property:
No particular one.

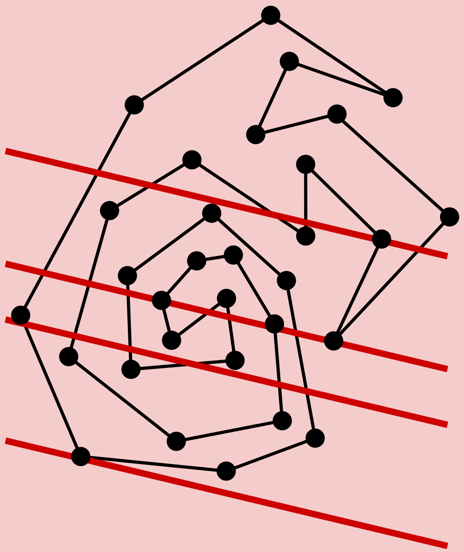
Brute-force solution

$O(n)$ time
$O(n)$ space





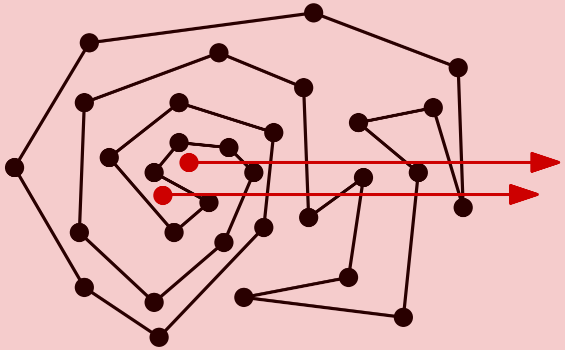
How did we prove the correctness of our solutions?



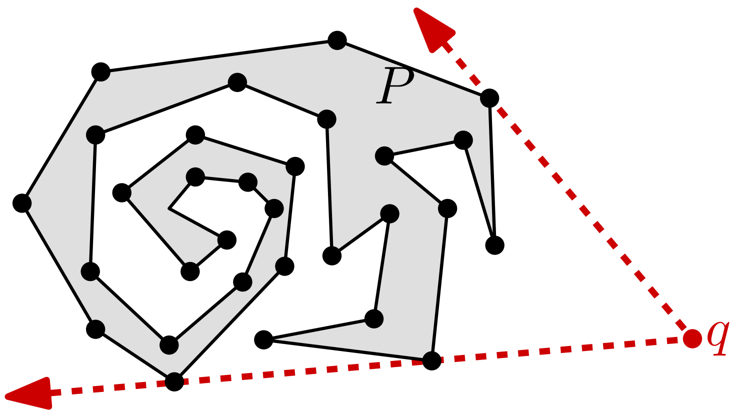
Geometric property:
No particular one.

Brute-force solution

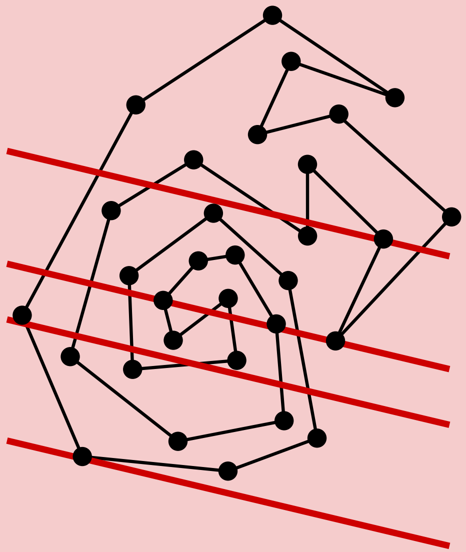
$O(n)$ time
$O(n)$ space



Geometric property:
$p \in P \Leftrightarrow$ The number of intersections of ∂P and any halfline with origin at p is odd.



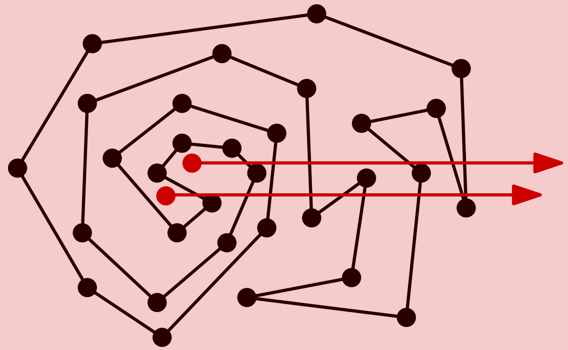
How did we prove the correctness of our solutions?



Geometric property:
No particular one.

Brute-force solution

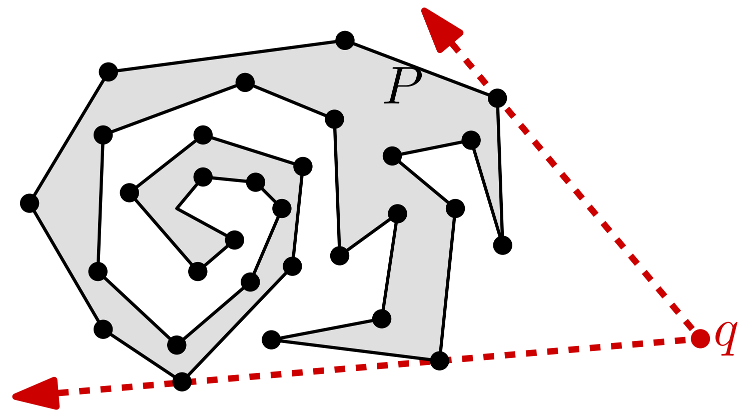
$O(n)$ time
$O(n)$ space



Geometric property:
$p \in P \Leftrightarrow$ The number of intersections of ∂P and any halfline with origin at p is odd.

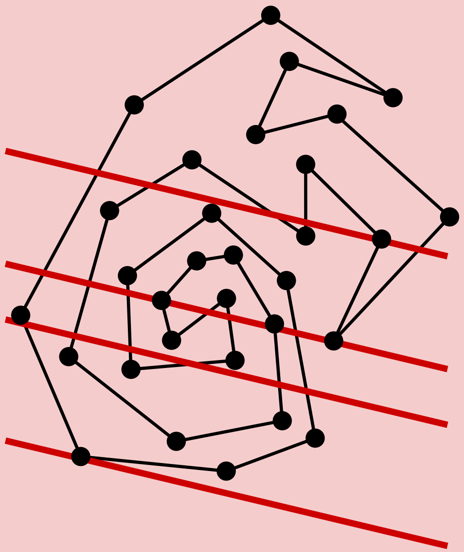
Brute-force solution

$O(n)$ time
$O(n)$ space



USING ORIENTATION TESTS ON POLYGONS

How did we prove the correctness of our solutions?

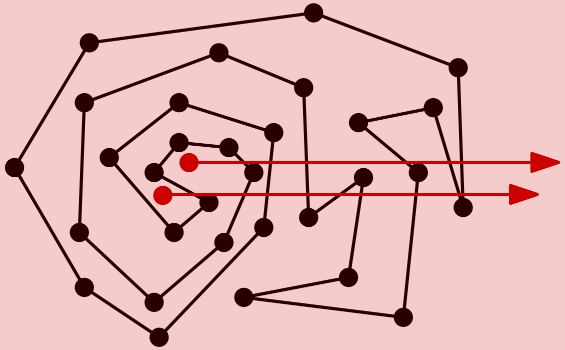


Geometric property:
No particular one.

⬇

Brute-force solution

$O(n)$ time
$O(n)$ space

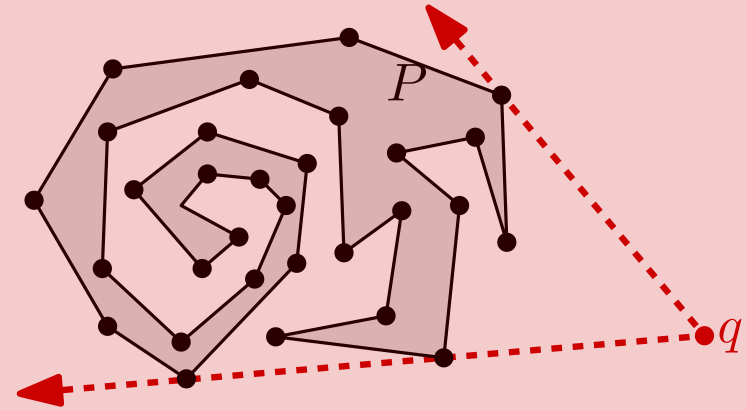


Geometric property:
$p \in P \Leftrightarrow$ The number of intersections of ∂P and any halfline with origin at p is odd.

⬇

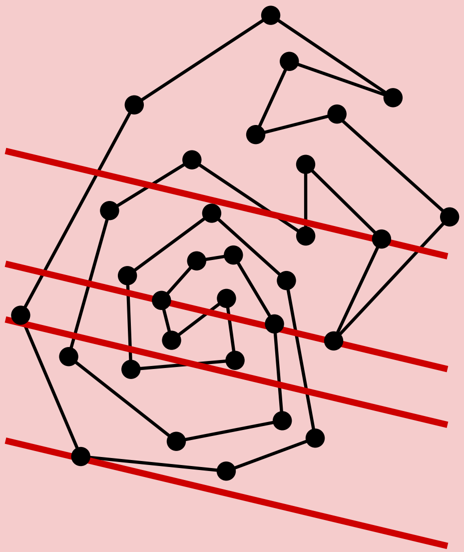
Brute-force solution

$O(n)$ time
$O(n)$ space



Geometric property:
The solutions are the angularly extreme vertices of P as seen from p.

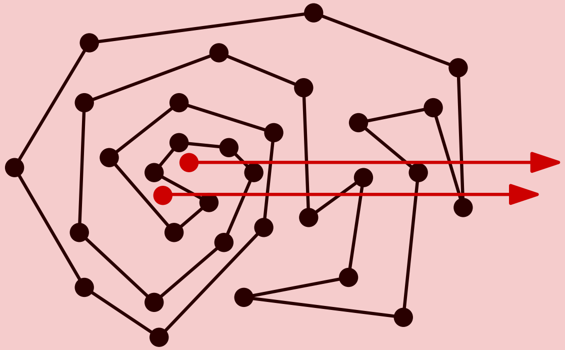
How did we prove the correctness of our solutions?



Geometric property:
No particular one.

Brute-force solution

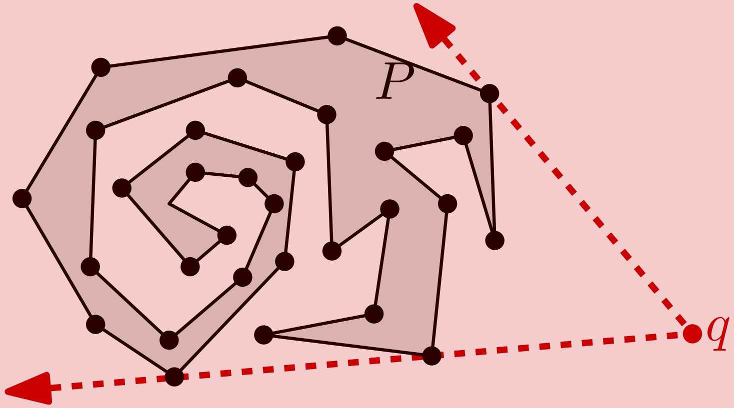
$O(n)$ time
$O(n)$ space



Geometric property:
$p \in P \Leftrightarrow$ The number of intersections of ∂P and any halfline with origin at p is odd.

Brute-force solution

$O(n)$ time
$O(n)$ space

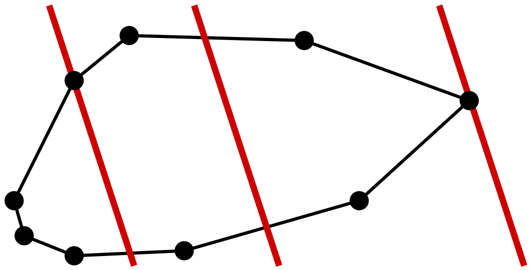


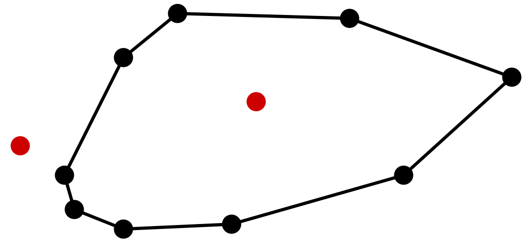
Geometric property:
The solutions are the angularly extreme vertices of P as seen from p.

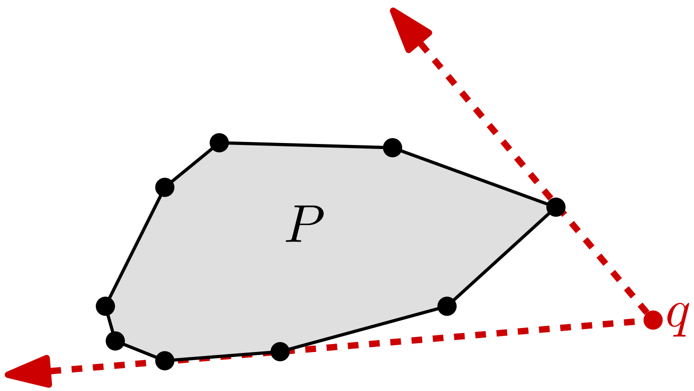
Use a max/min algorithm

$O(n)$ time
$O(n)$ space

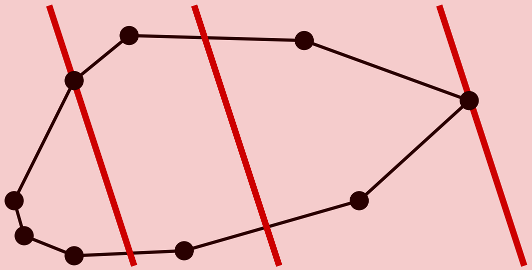
How did we prove the correctness of our solutions?



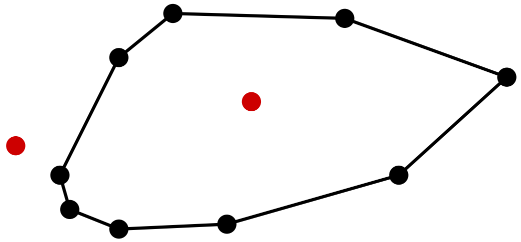


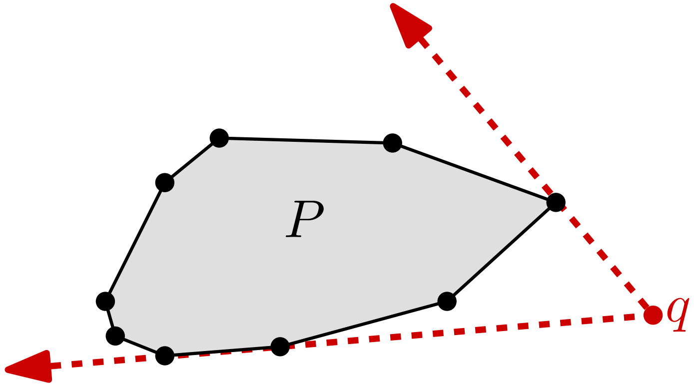


How did we prove the correctness of our solutions?

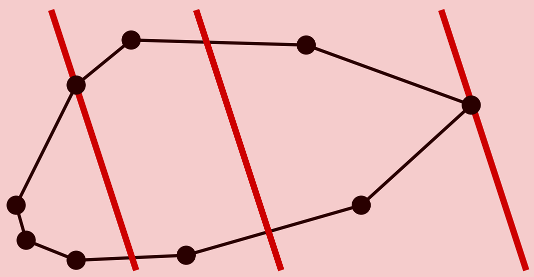


Geometric property: Distance to line is unimodal along each chain of ∂P.





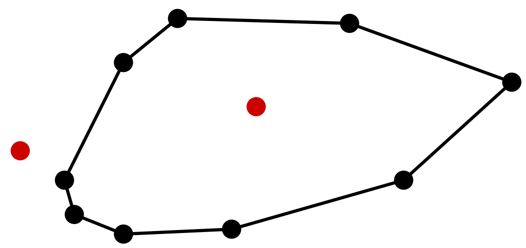
How did we prove the correctness of our solutions?

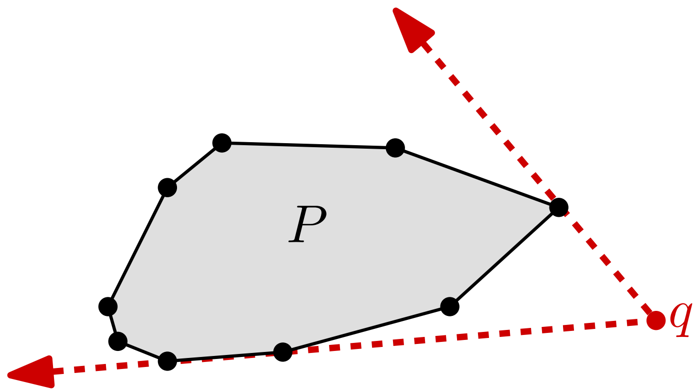


Geometric property: Distance to line is unimodal along each chain of ∂P.

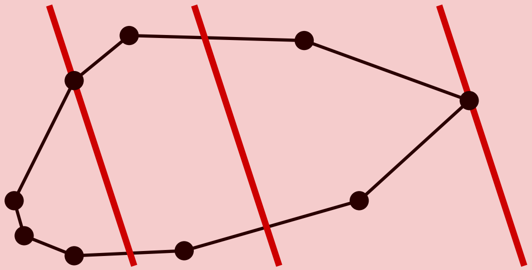
Binary search solution

$O(\log n)$ time
$O(n)$ space
(after preprocess)





How did we prove the correctness of our solutions?

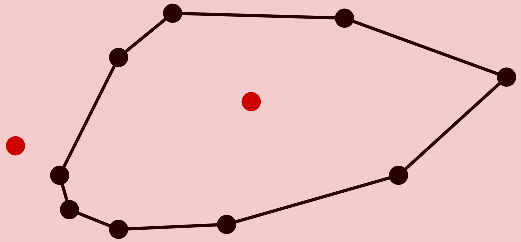


Geometric property: Distance to line is unimodal along each chain of ∂P.

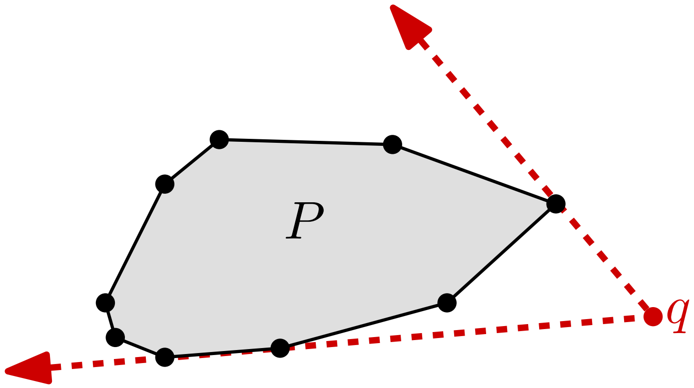
⬇

Binary search solution

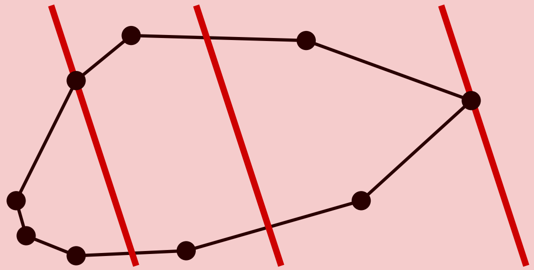
$O(\log n)$ time
$O(n)$ space
(after preprocess)



Geometric property: Segments connecting two vertices decompose P into two convex subpolygons.



How did we prove the correctness of our solutions?

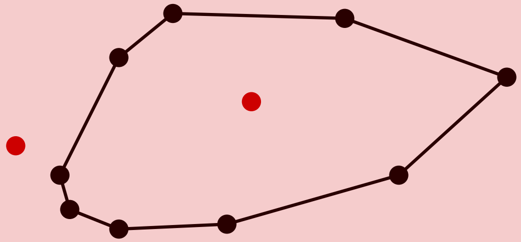


Geometric property: Distance to line is unimodal along each chain of ∂P.

⬇

Binary search solution

$O(\log n)$ time
$O(n)$ space
(after preprocess)

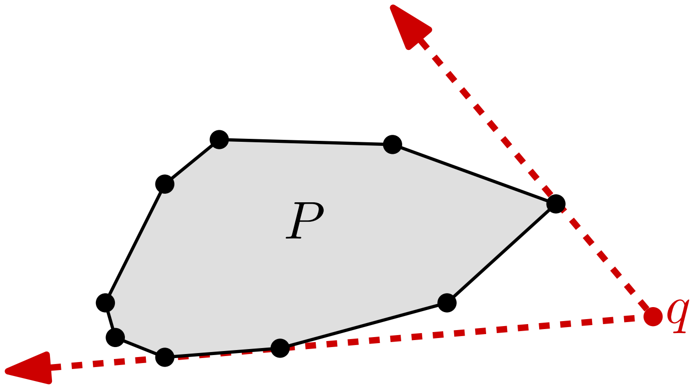


Geometric property: Segments connecting two vertices decompose P into two convex subpolygons.

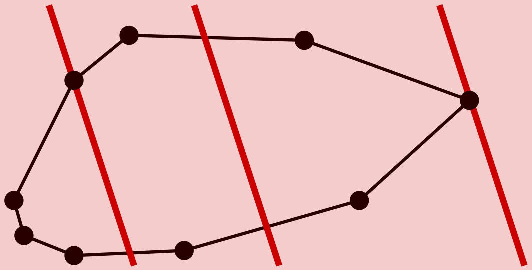
⬇

Binary search solution

$O(\log n)$ time
$O(n)$ space
(after preprocess)



How did we prove the correctness of our solutions?

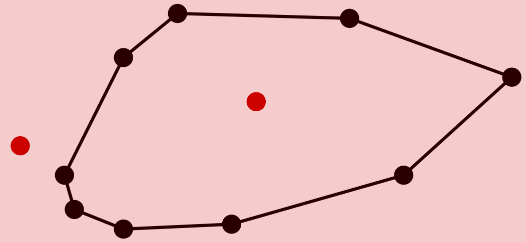


Geometric property: Distance to line is unimodal along each chain of ∂P.

⬇

Binary search solution

$O(\log n)$ time
$O(n)$ space
(after preprocess)

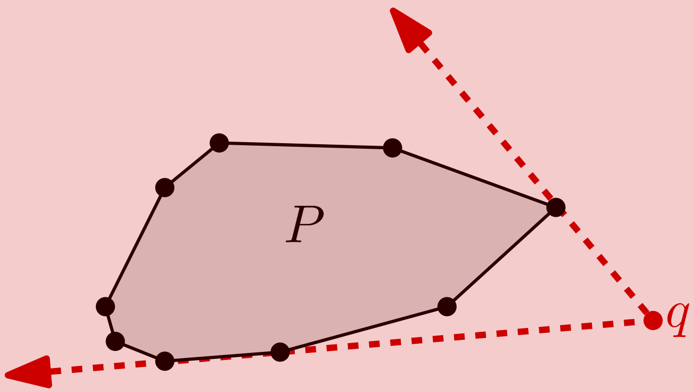


Geometric property: Segments connecting two vertices decompose P into two convex subpolygons.

⬇

Binary search solution

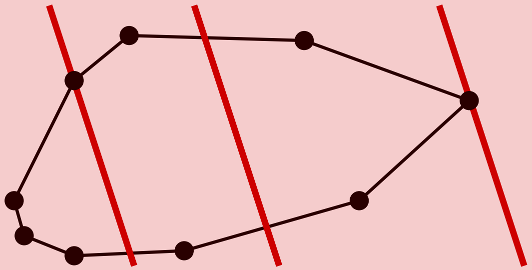
$O(\log n)$ time
$O(n)$ space
(after preprocess)



Geometric property: Angle wrt q is unimodal along ∂P.

USING ORIENTATION TESTS ON POLYGONS

How did we prove the correctness of our solutions?

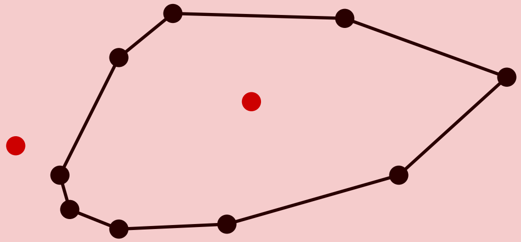


Geometric property: Distance to line is unimodal along each chain of ∂P.

⬇

Binary search solution

$O(\log n)$ time
$O(n)$ space
(after preprocess)

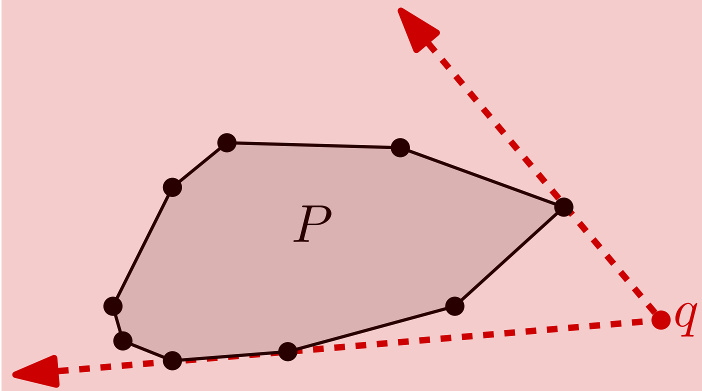


Geometric property: Segments connecting two vertices decompose P into two convex subpolygons.

⬇

Binary search solution

$O(\log n)$ time
$O(n)$ space
(after preprocess)



Geometric property: Angle wrt q is unimodal along ∂P.

⬇

Binary search solution

$O(\log n)$ time
$O(n)$ space
(after preprocess)

FURTHER READING

J. O'Rourke
Computational Geometry in C
Cambridge University Press, 1994 (2nd ed. 1998), pp. 17-35.

F. P. Preparata and M. I. Shamos
Computational Geometry: An Introduction
Springer-Verlag, 1985, pp. 36-45.