CS60064 Spring 2022

Computational Geometry

Instructors

Bhargab B. Bhattacharya (BBB)
Partha Bhowmick (PB)

Lecture 01
05 January 2022

Indian Institute of Technology Kharagpur
Computer Science and Engineering

Class Schedule

Wednesday: 10:00 – 10:55

Thursday: 09:00 – 09:55

Friday: 11:00 – 11:55

Can we merge the classes scheduled on Thursday and Friday into a two-hour slot on Friday (11:00 – 12:50)?

Teaching Assistants

Faraaz Rahaman Mallick (faraazrm@gmail.com)
Sashank Bonda (sashank729@gmail.com)

Course Page

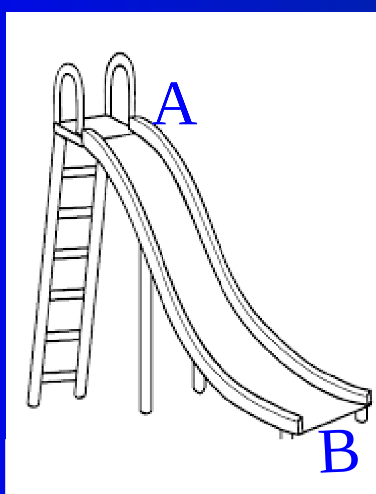
https://moodlecse.iitkgp.ac.in/moodle/login/index.php

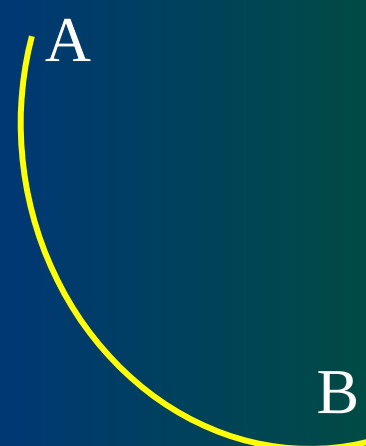
Moodle Student Registration Key for the Course:
STUBBPB22

What is the path of fastest descent of an object
from a point A to point B?

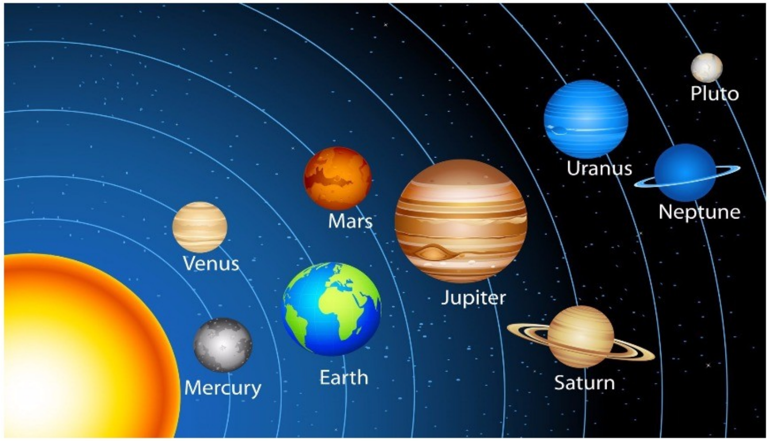
Brachistochrone: A cycloid upside-down. This is the path down which a particle will travel in the *shortest time*. This problem was first proposed by John Bernouilli in 1696, soon after the birth of calculus.

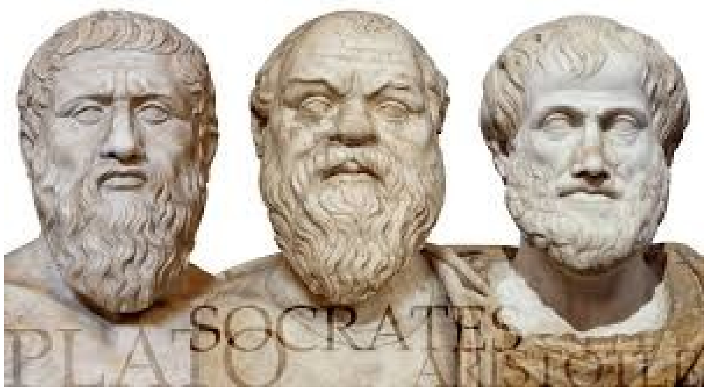
Newton solved it in a single night using calculus!





Geometry Everywhere



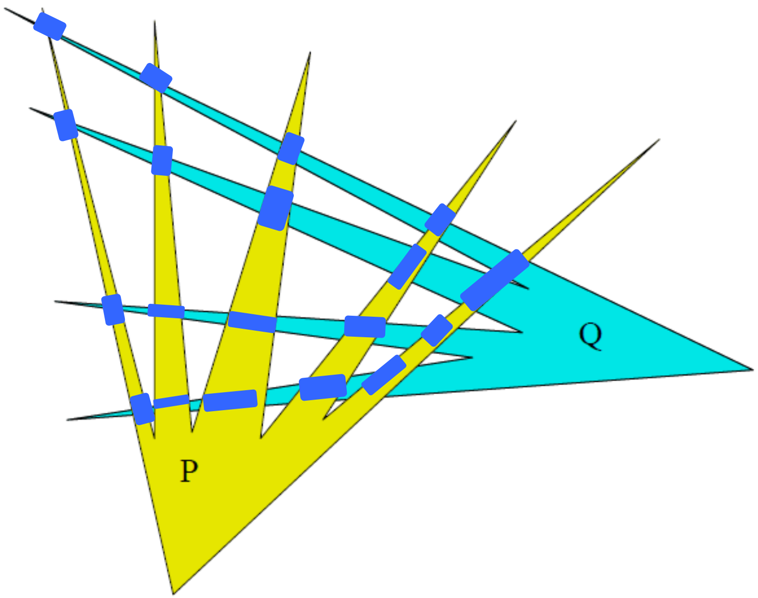


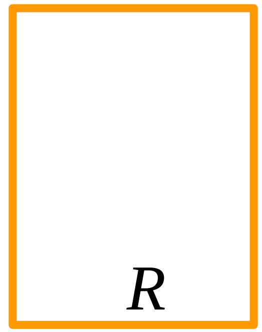
God forever geometrizes
Plato (Circa 424 – 348 BC)

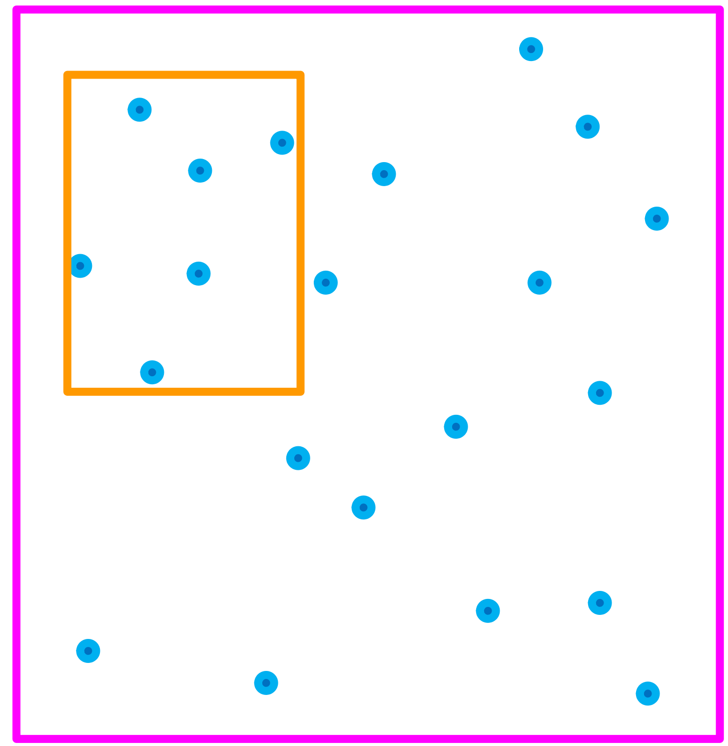
Warm-Up Examples

Why Geometric problems are so special?

Given two polygons P and Q,
determine $P \quad Q$







CG = Geometric properties +
data structures + algorithms + coding

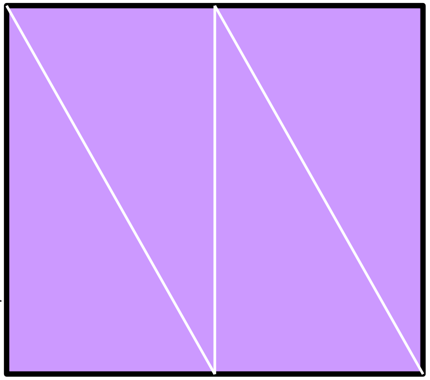
Place R such that it encloses
maximum number of points

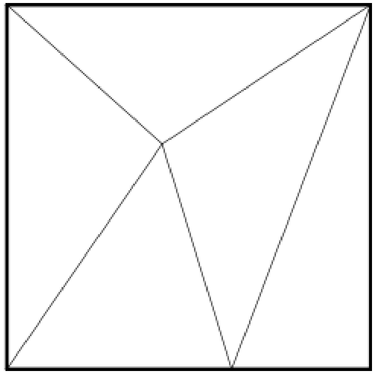
Why pizzas are supplied in square boxes, round in shape, and eaten in triangles?

A square-box can be constructed by folding a single sheet of card board, with certain cuts!

A round-box cannot be constructed by folding a single sheet

Monsky's theorem (1970): it is not possible to dissect a square into an odd number of triangles of equal area





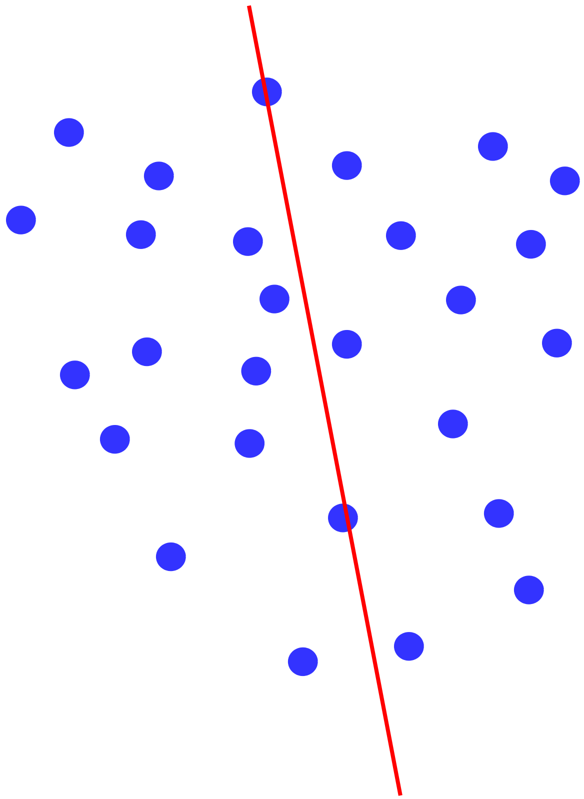
Proof: Self-study

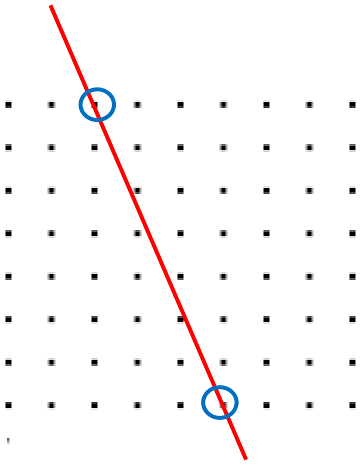
Why pizzas are supplied in square boxes, round in shape, and eaten in triangles?

Gaussian curvature of a point in a surface: product of curvatures of the most convex and most concave paths

Courtesy: Aatish Bhatia, The WIRE, 2017

A Non-Sense (?) Problem





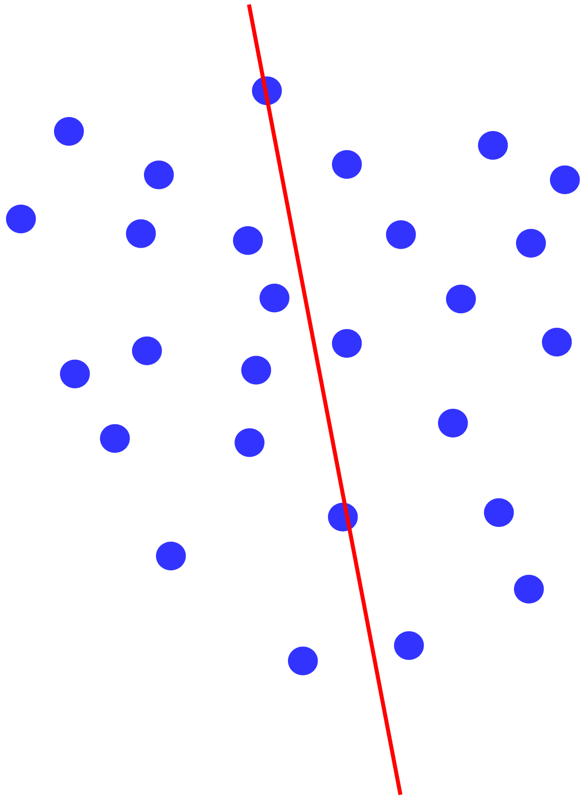
The Sylvester-Gallai Theorem

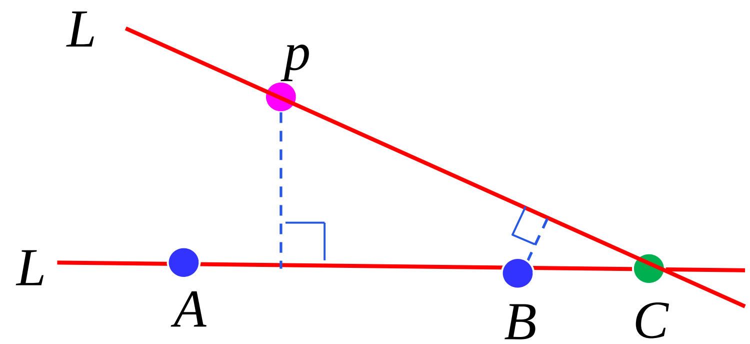
- J. J. Sylvester posed in 1893
- Remained unsolved for 50 years!
- T. Gallai first proved it in 1944

Claim: Given a finite set P of n points in the 2D plane, either all the points are collinear, or there exists at least one line passing through *exactly* two points in P.

Sylvester-Gallai Theorem: Kelly’s Proof

After 40 years, in 1986, Kelly provided a short and elegant proof!





Proof:

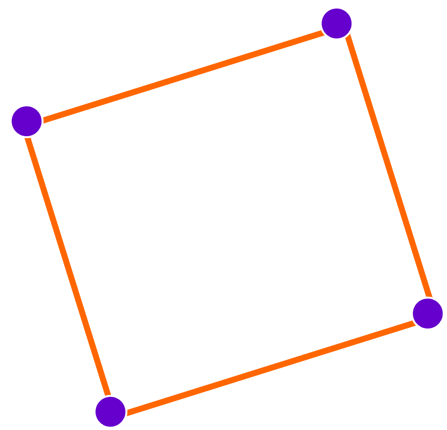
- Let L be a line passing through two points A, B and p P but L, such that *dist*(p, L) is minimum among all such (L, p) tuple;
- Assume L also passes through a third point C P; (L , B) s.t. *dist*(B, L) < *dist*(p, L), contradiction!

Claim: Given a finite set P of n points in the 2D plane, either all the points are collinear, or there exists at least one line passing through *exactly* two points in P. Such a line is called an “ordinary line”.

Toeplitz conjecture
(the inscribed square problem or the square-peg problem):

Question: Does every Jordan curve admit an inscribed square? That is, does every simple closed curve in the plane contain four vertices of a square?

In 1911, Otto Toeplitz posed the question, however, it still remains unsolved



CS60064 Spring 2022

Computational Geometry

Instructors

Bhargab B. Bhattacharya (BBB)

Partha Bhowmick (PB)

Lecture 02

06 January 2022

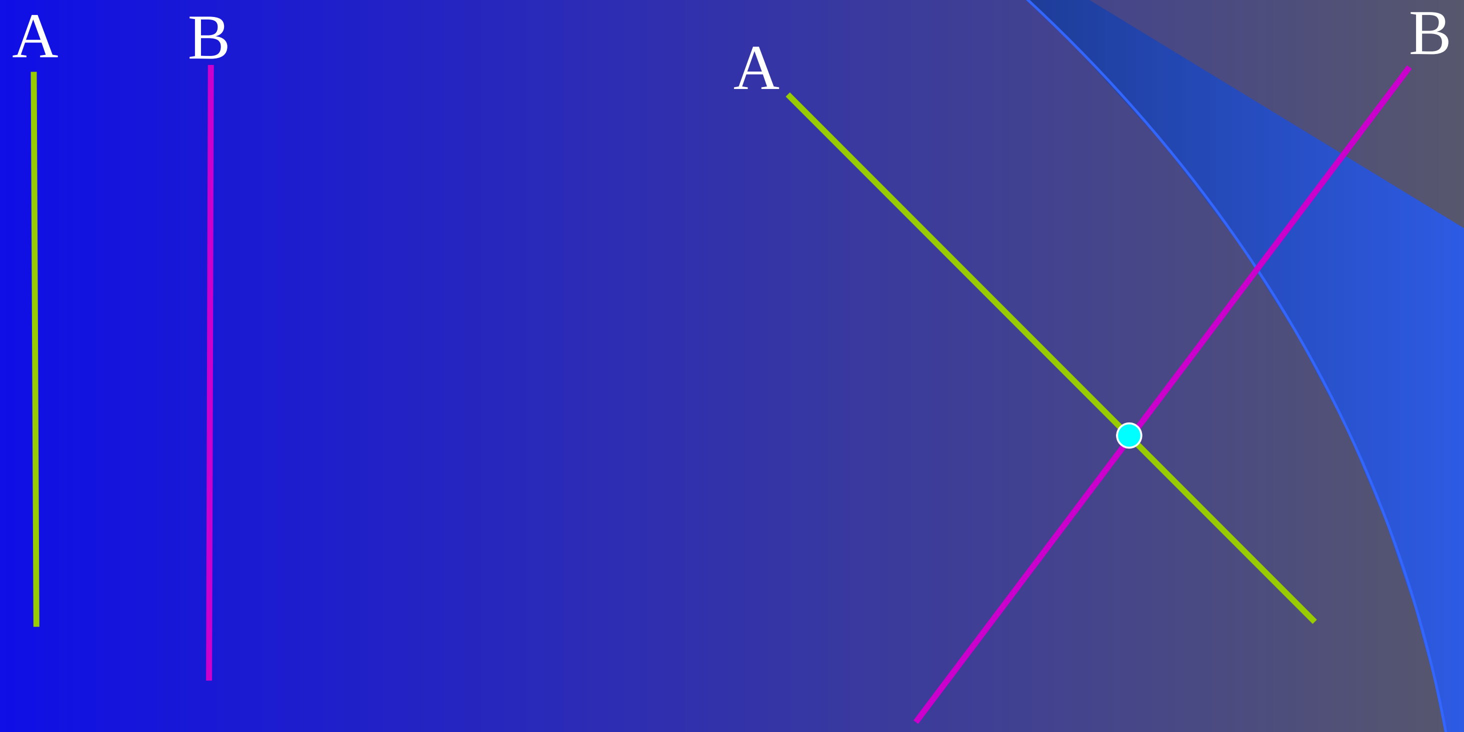
Indian Institute of Technology Kharagpur

Computer Science and Engineering

Warm-Up Examples (contd..)

Euclidean Straight Line

Euclidean: Two straight lines are either parallel or intersect at a unique point

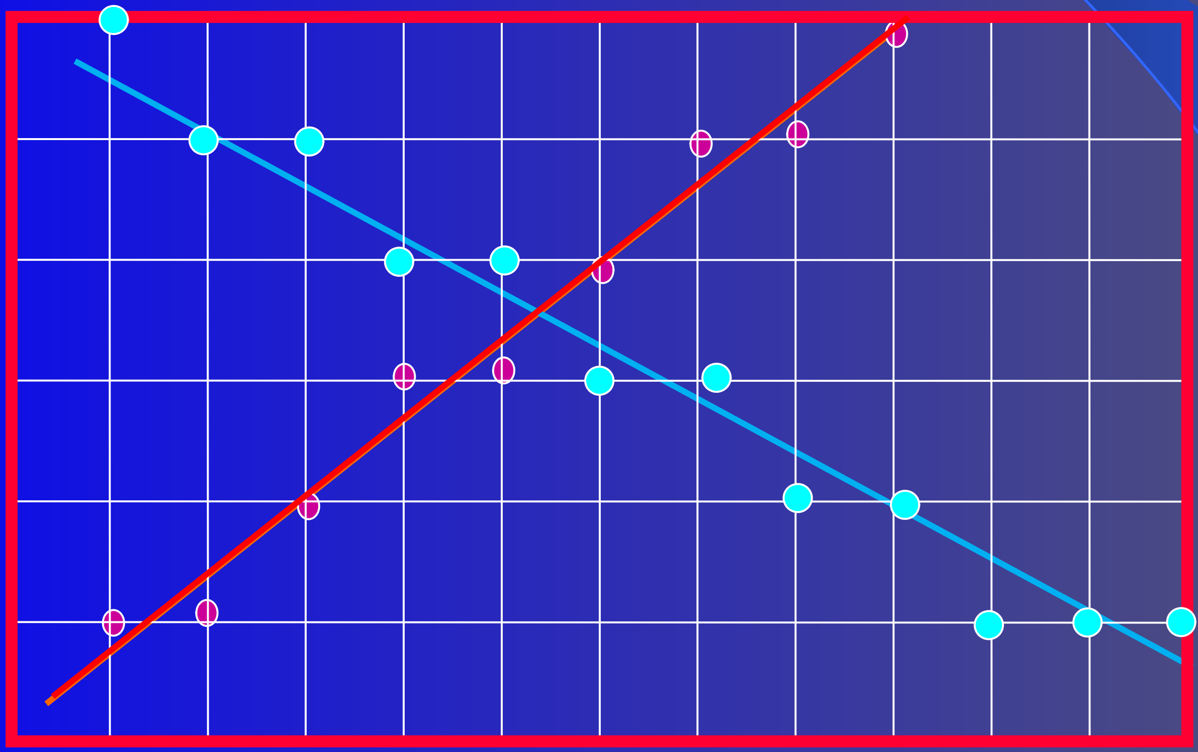


Grid (Digital) Geometry: Dilemma of Intersection

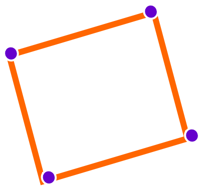
two intersecting Euclidean straight lines
the corresponding object sets are disjoint!

Take an image of this object
using a camera

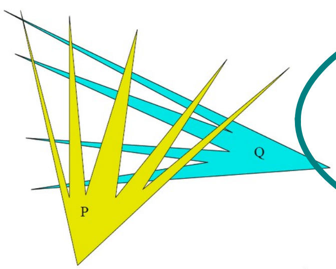
A straight line now becomes
a set of grid points



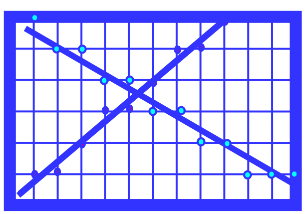
Computational geometry and related fields



- Discrete and combinatorial geometry: study of existential and extremal problems

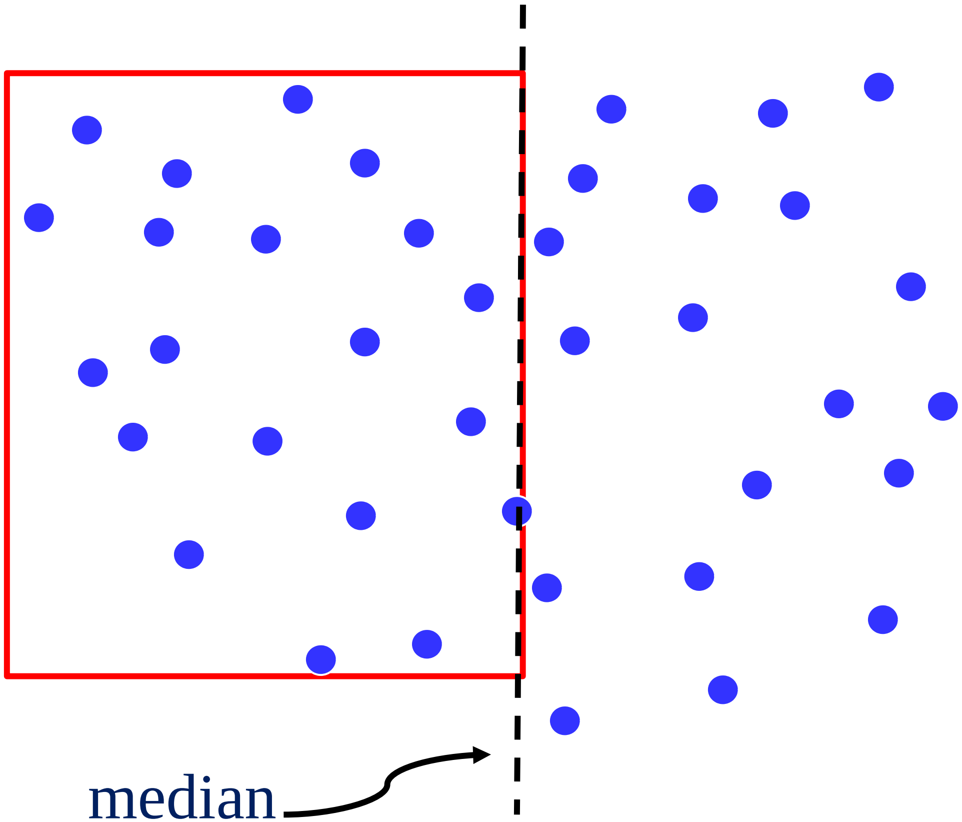


- Computational geometry: mostly in Euclidean space; *algorithmic* solutions to geometric problems, geometric *data structures;* applications to robotics, GIS, scientific computing, facility location, computational biology, graphics



- Digital geometry: properties of pixels and algorithms in discrete lattice space
   applications to image analysis, computer graphics, computer vision, computer art

Warming Up: Starting with a Simple Puzzle of CG…

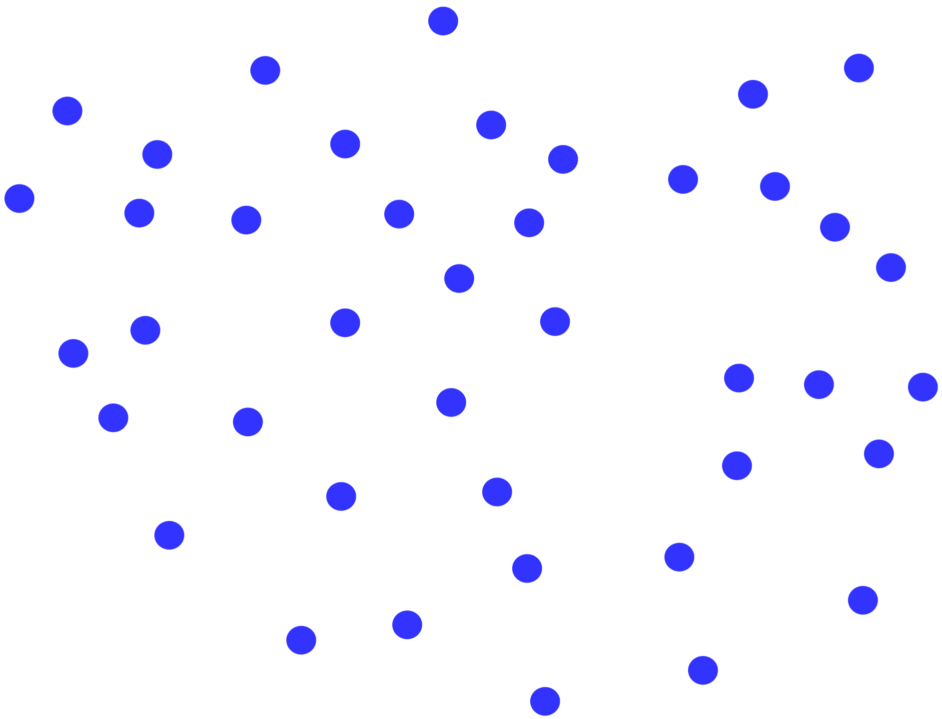


$O(n)$

Given n points on the 2D plane where n is an odd number ≥ 5, find a rectangle touching a point such that it encloses exactly half of the remaining points. Assume no two points have the same x-coordinate.

Making the puzzle little more difficult

$O(n^4)$?
Does it always exist?



Halving-circle problem

Given n points on the 2D plane where n is an odd number ≥ 5, find a circle passing through three points such that it encloses exactly half of the remaining points. Assume no four points are concyclic.

Jaime Rangel-Mondragon, "Problems on Circles II: Halving a Set of Points"
http://demonstrations.wolfram.com/ProblemsOnCirclesIIHalvingASetOfPoints/; Wolfram Demonstrations Project
Published: March 7 2011

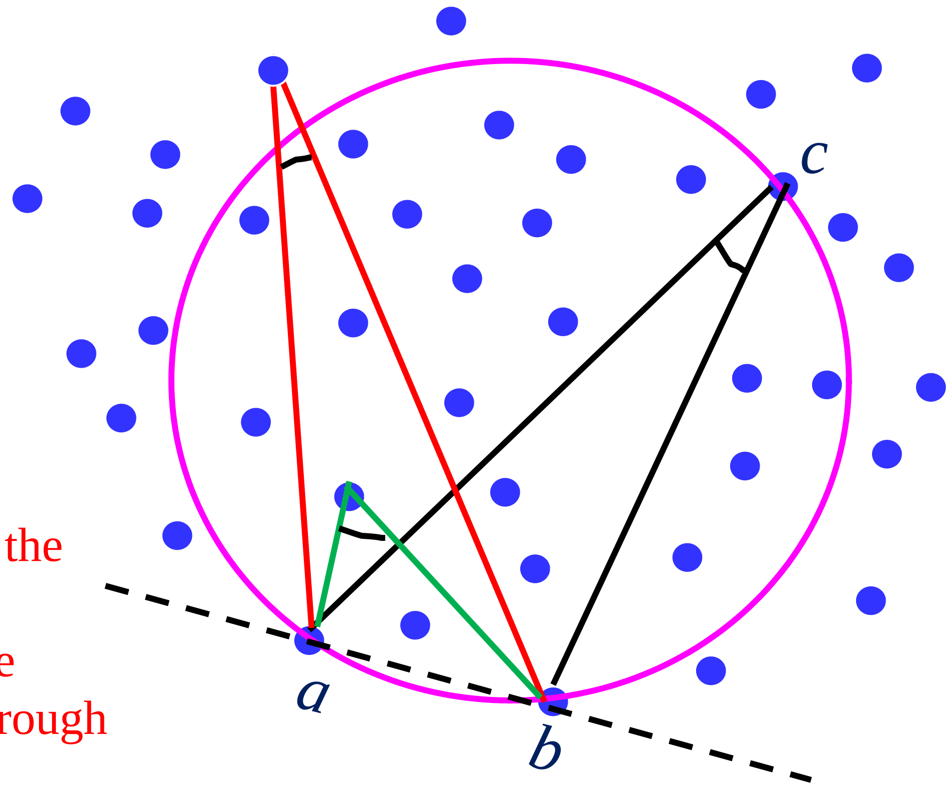
Making the puzzle little more difficult

Halving-circle problem

$O(n)$?

no four points
are concyclic

< <



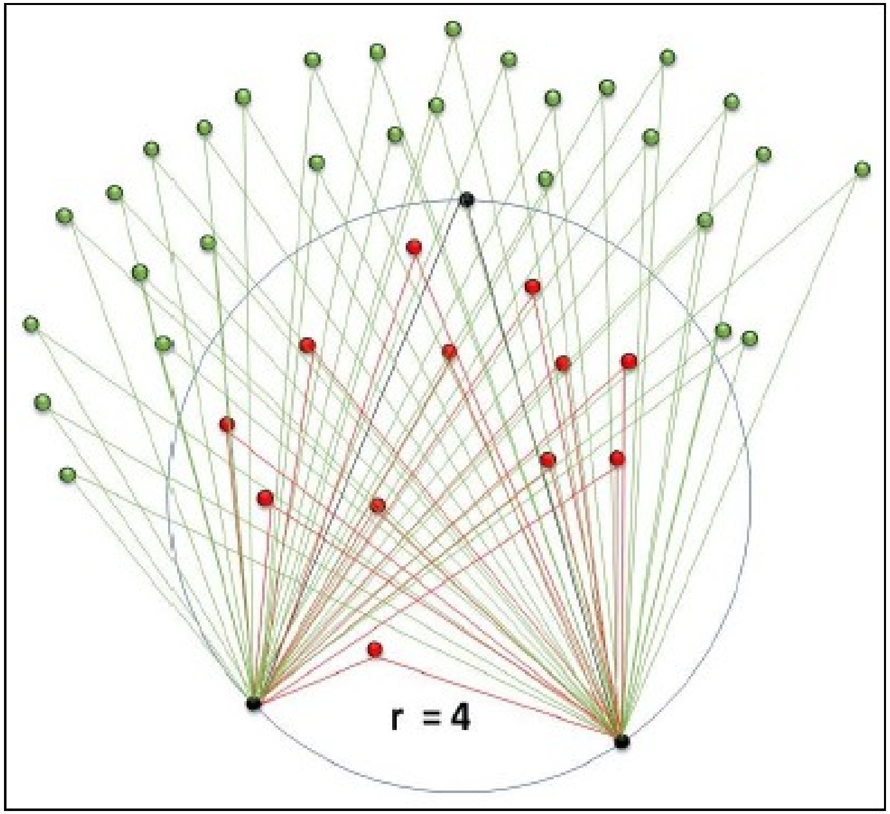
b is the lowest point; a is the point that makes the least angle clock-wise, with the horizontal line passing through b. Both points can be identified in $O(n)$ time.

Claim: The angle subtended by the line segment ab to each of the n points must be distinct; Let c is the point where the subtended angle is the median. Separating circle will pass through a, b, c

Problem

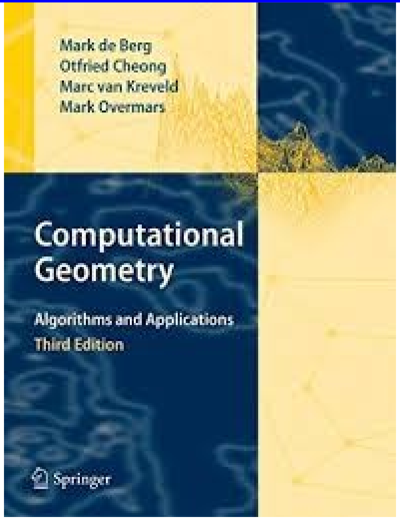
Generate instances of *n* random points ($n = 10r + 3$, r: integer > 1) on the plane,

Construct a separating circle that encloses 30% of points in the interior



Textbook and Notes

M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars: C*omputational Geometry: Algorithms and Applications* (3rd edition), Springer Verlag, 2008 (commonly known as 4A-Book as there are four authors)



David M. Mount: CMSC 754 *Computational Geometry Lecture Notes*, Department of Computer Science,
University of Maryland 2002
(
http://www.cs.umd.edu/~mount/754/Lects/754lects.pdf)

References

S. L. Devadoss and J. O’Rourke: *Discrete and Computational Geometry*, Princeton University Press, 2011.

F. Preparata and M. Shamos: *Computational Geometry: An Introduction (3rd edition)*, Springer Verlag,1993.

J. O'Rourke: *Computational Geometry in C (2nd edition),* Cambridge Univ, Press, 1998.

J. O’Rourke, *Art Gallery Theorems and Algorithms*, Oxford Univ. Press, 1987.

K. Mulmuley: *Computational Geometry: An Introduction Through Randomized Algorithms,* Prentice Hall, 1994.

S. Ghosh: *Visibility Algorithms in the Plane,* Cambridge University Press, 2007.

Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth (Ed.): *Handbook of Discrete and Computational Geometry*, Third Edition, CRC Press, 2017.

Jiri Matousek: *Lectures on Discrete Geometry*, Springer, 2002.

Erik D. Demaine and Joseph O'Rourke: *Geometric Folding Algorithms*: *Linkages, Origami, Polyhedra,* Cambridge University Press, 2007.

Micha Sharir and Pankaj Agarwal: *Davenport Schinzel Sequences and Their Geometric Applications*, Cambridge University Press, 2010.

Godfried T. Toussaint, *The Geometry of Musical Rhythm*, CRC Press, 2013.

Marvin Minsky and Seymour Papert: *Perceptrons – An Introduction to Computational Geometry*, MIT Press, 1969.

Web Resources

- NPTEL Video Course on Computational Geometry by *Sandeep Sen,* Department of Computer Science & Engineering, IIT Delhi, and *Pankaj Agarwal*, Dept. of CS, Duke University, USA
  (http://www.nptelvideos.in/2012/11/computational-geometry.html)
- Lecture Slides on Computational Geometry by Martin Held, University of Salzburg, Austria (https://www.cosy.sbg.ac.at/~held)
- Godfried T. Toussaint's page: http://cgm.cs.mcgill.ca/~godfried/teaching.html
- CGAL Computational Geometry Library (https://www.cgal.org/)
- K. Mehlhorn and St. Näher: *The LEDA Platform of Combinatorial and Geometric Computing*, Cambridge University Press, 1999 (https://people.mpi-inf.mpg.de/~mehlhorn/LEDAbook.html).
- Computational Geometry (Wolfram Research) (http://mathworld.wolfram.com/topics/ComputationalGeometry.html)
- David Eppstein's Geometry in Action (https://www.ics.uci.edu/~eppstein/geom.html)
- David Eppstein's Geometry Junkyard (https://www.ics.uci.edu/~eppstein/junkyard/)

Journals and Conferences

- CGTA: Computational Geometry: Theory and Applications
- DCG: Discrete and Computational Geometry
- IJCGA: International Journal of Computational Geometry
- SoCG: Symposium on Computational Geometry
- CCCG: Canadian Conference on Computational Geometry

Grading Policy

- Homework/Quiz/Term papers – 50%
- Two Tests – 50% (25% credit each)

For doing Homework/Term Papers, the students are requested to form disjoint teams, each team comprising *two students.* A student may work individually as well, if so desired. In order to form teams, please coordinate with TA's who will share a google doc file soon.

Computational Geometry

- Fascinating subject that deals with algorithmic issues encountered in geometric problems
- History of geometry?
- Pythagoras (*b*. 570 BC), Euclid (*b*. 325 BC), Archimedes (*b*. 287 BC), da Vinci (*b*. 1452) Descarte (*b*. 1596), Fermat (*b*. 1607), Gauss (*b*. 1777), Lobachevsky (*b*. 1792), Riemann (*b*. 1826), Minkowski (*b*. 1864), Erdős (*b*. 1913)
- In India:

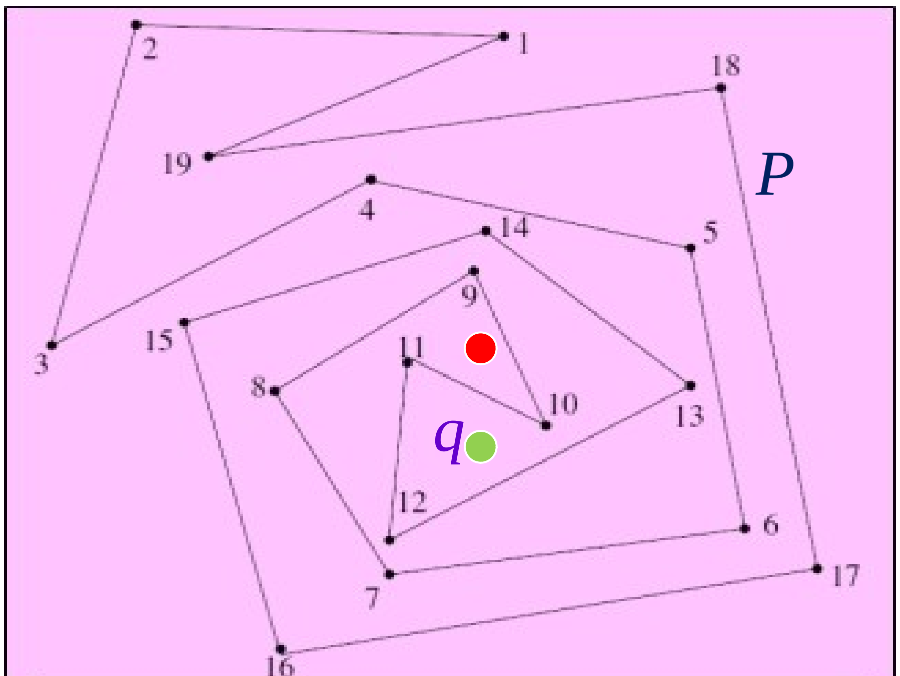
  Aryabhatta (*b*. 476), Brahmagupta (*b*. 598), Bhaskara (*b*. 600), Bhaskaracharya (*b*. 1114), famous for Lilavati

What is Computational Geometry?

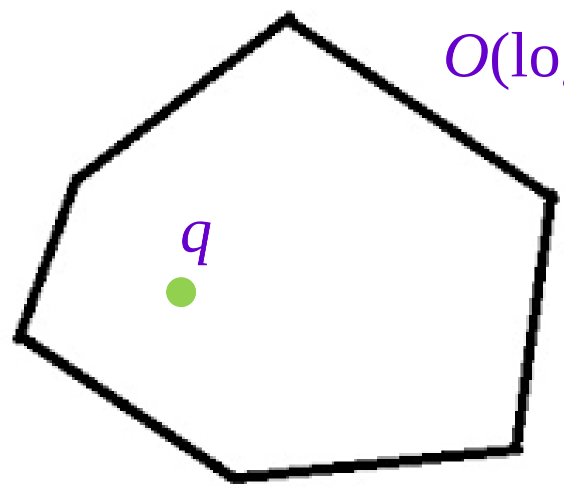
The algorithmic and mathematical study of efficient methods to solve geometric problems, mostly in the Euclidean space□

Question: Given a polygon P on the 2D plane with n sides, and a query point q, does P include q?

Representation? Choice of origin and axes?



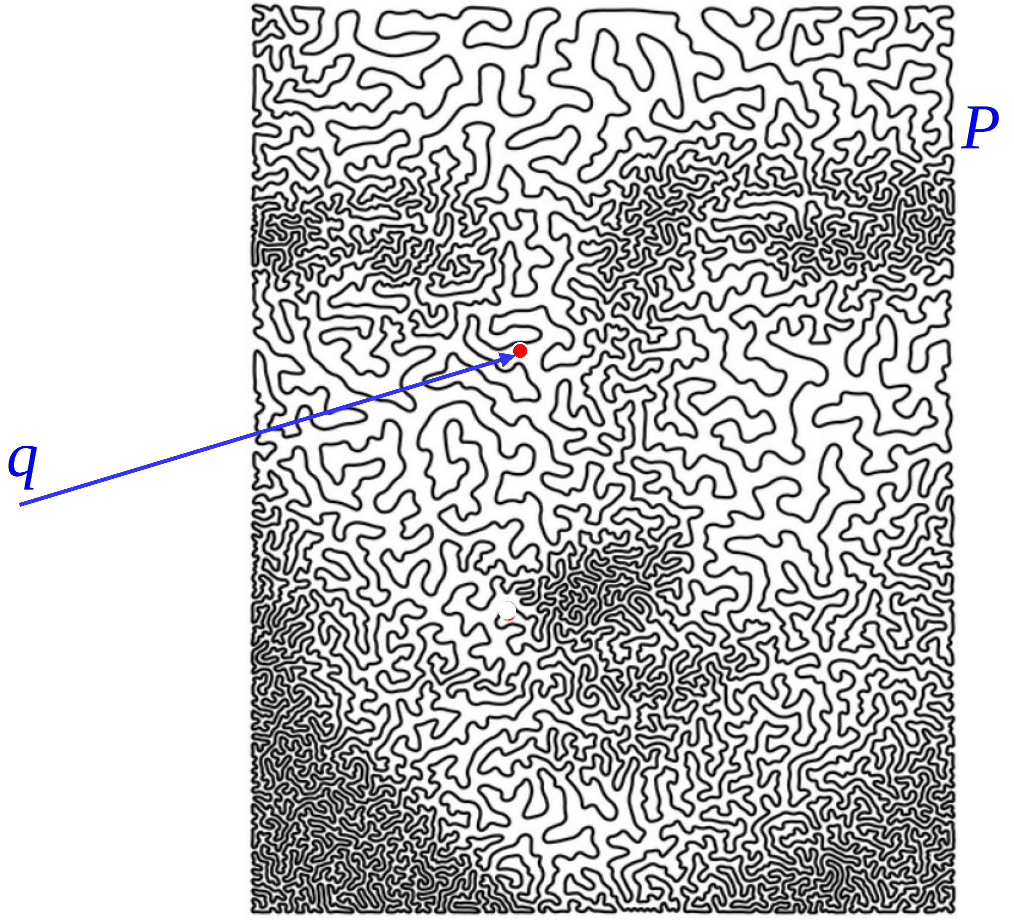
$O(n)$



$O(\log n)$

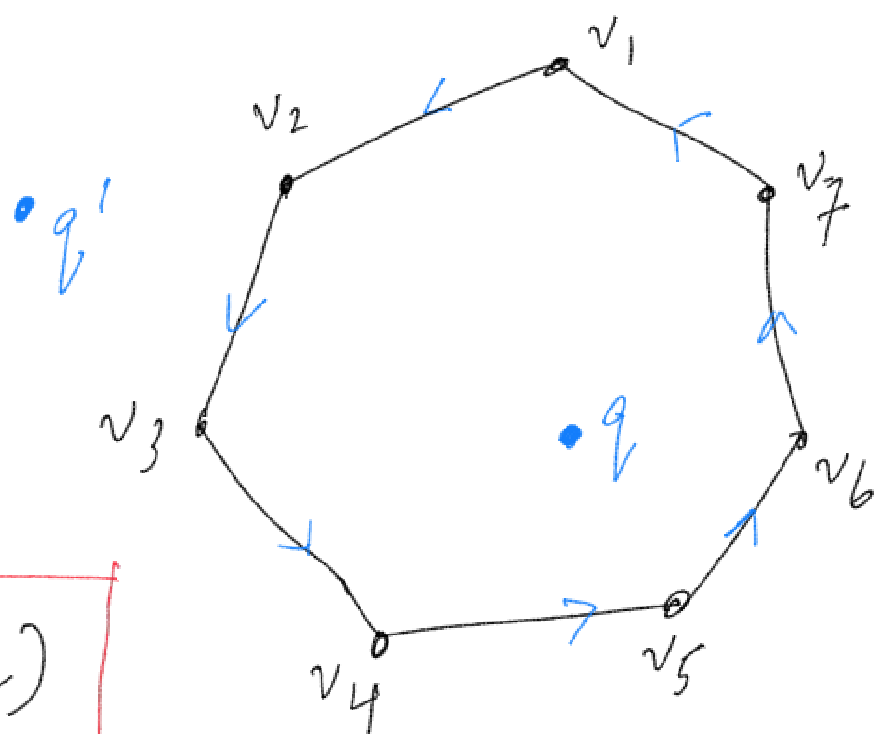
Convex polygon: all internal angles ≤

Question: Given a polygon P on the 2D plane with n sides, and a query point q, does P include q?



Courtesy: Gabriel Robins

Given a convex polygon P and a query point q, does P include q?



$\boxed{O(n)}$

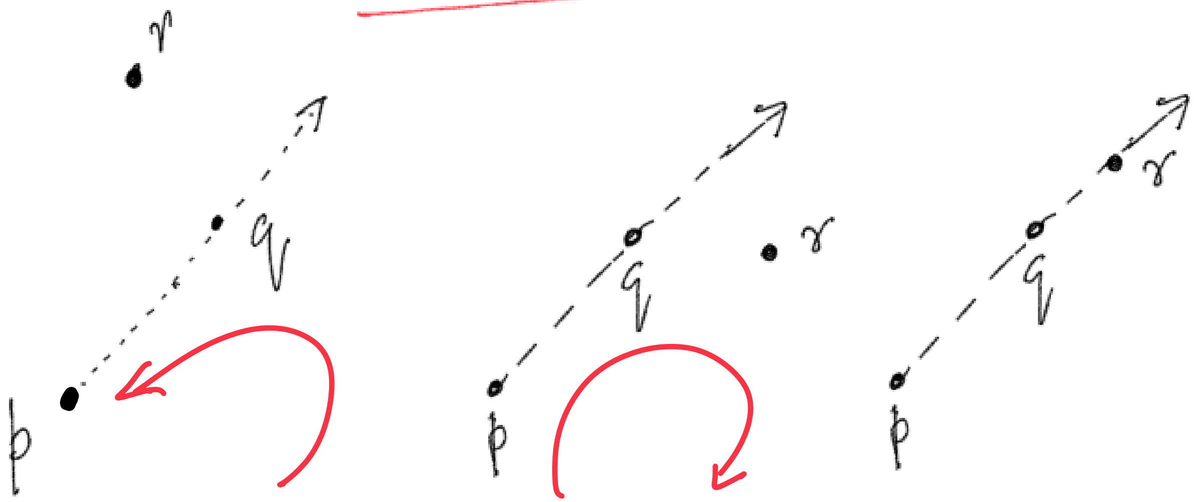
P is described as an ordered sequence of vertices $v_1, v_2, \ldots, v_n$

check orientation of $(v_i, v_{i+1}, q), \forall i$

all CCW $\Rightarrow$ q is included in P.

Orientation Test

Three points on the plane
p, q, r



Travel from p to q ($\overrightarrow{pq}$) and check whether the point r lies at the left (CCW), right (CW) of $\overrightarrow{pq}$ or collinear.

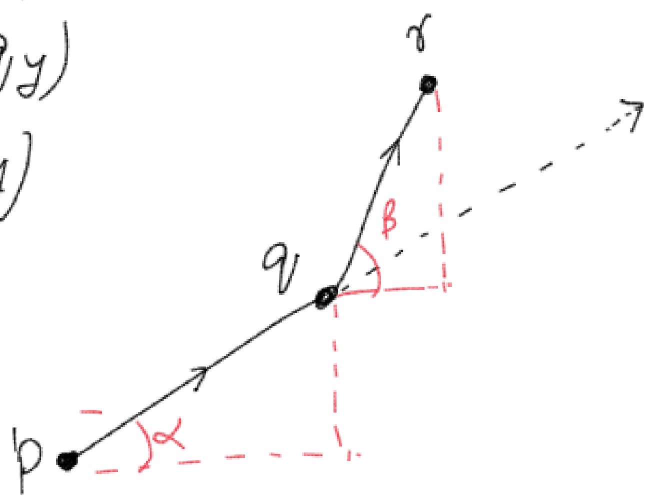
Ref: David Mount, Lecture Notes

Orientation Test

$p \rightarrow (p_x, p_y)$

$q \rightarrow (q_x, q_y)$

$r \rightarrow (r_x, r_y)$



$p, q, r \rightarrow$ left turn (CCW)

if $\alpha < \beta$

$p, q, r \rightarrow$ right turn (CW)

if $\alpha > \beta$

---

p, q, r collinear

if $\alpha = \beta$

$$\tan\alpha = \frac{q_y - p_y}{q_x - p_x}$$

$$\tan\beta = \frac{r_y - q_y}{r_x - q_x}$$

$$\tan\alpha - \tan\beta = \frac{q_y - p_y}{q_x - p_x} - \frac{r_y - q_y}{r_x - q_x}$$

$$\alpha \gtreqless \beta \iff \tan\alpha \gtreqless \tan\beta$$

Check whether $(\tan\alpha - \tan\beta) \lesseqgtr 0$

i.e., $(q_y - p_y)(r_x - q_x) - (q_x - p_x)(r_y - q_y) \lesseqgtr 0$

$$\boxed{\text{i.e., check } \; -\det\begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix} \begin{array}{l} < 0 \implies \text{ccw} \\ = 0 \implies \text{Collinear} \\ > 0 \implies \text{cw} \end{array}}$$

CS60064 Spring 2022

Computational Geometry

Instructors

Bhargab B. Bhattacharya (BBB)

Partha Bhowmick (PB)

Lecture 03

07 January 2022

Indian Institute of Technology Kharagpur

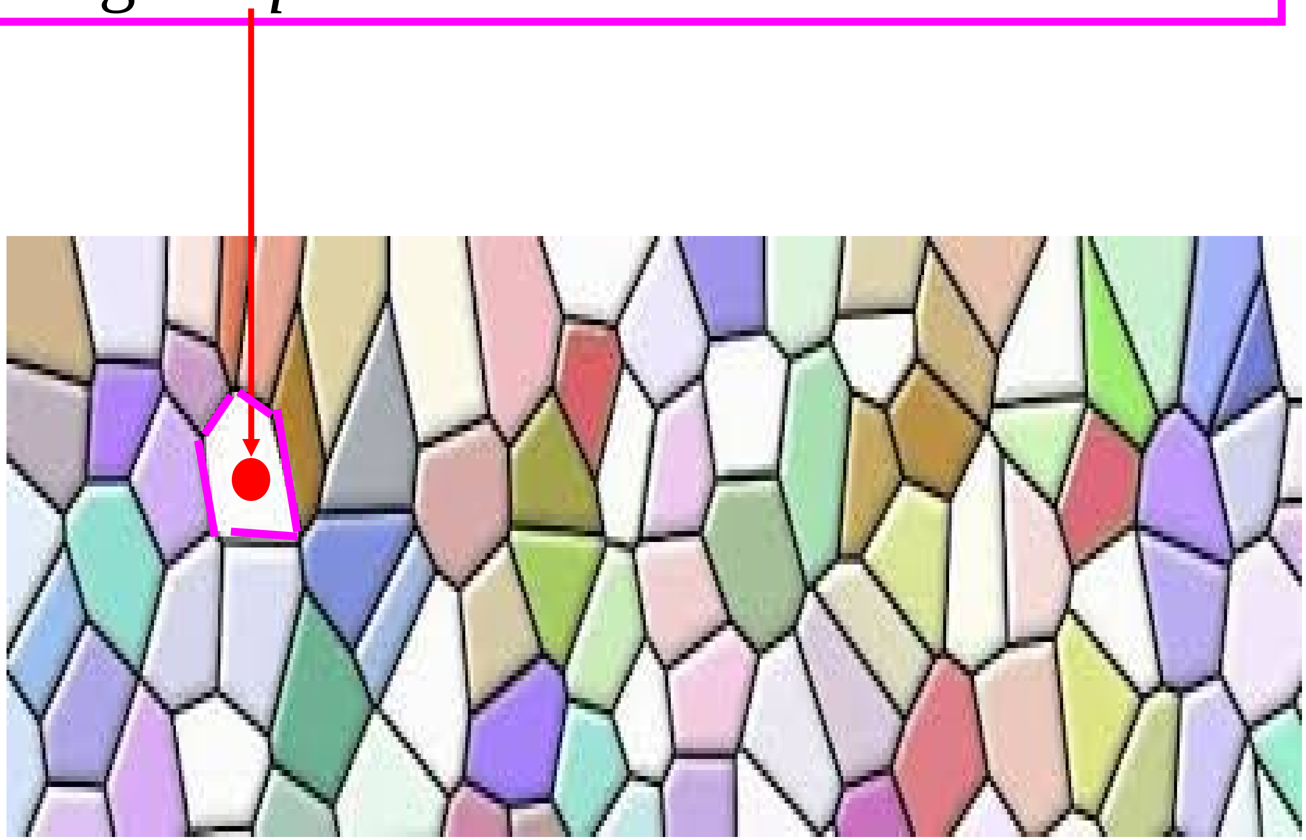
Computer Science and Engineering

Point Inclusion in Polygons Orientation Test (contd..)

A More General Version: Point Location, Map Navigation

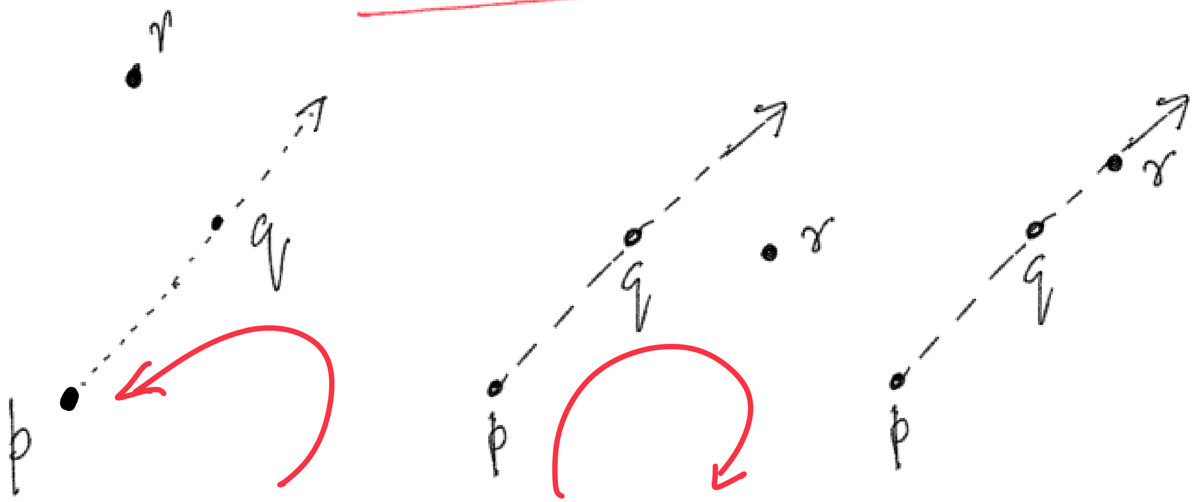
Question: Given a map on a 2D plane and a query point q, determine is which region q is contained

How does a mouse-click work?



Recap: # Orientation Test

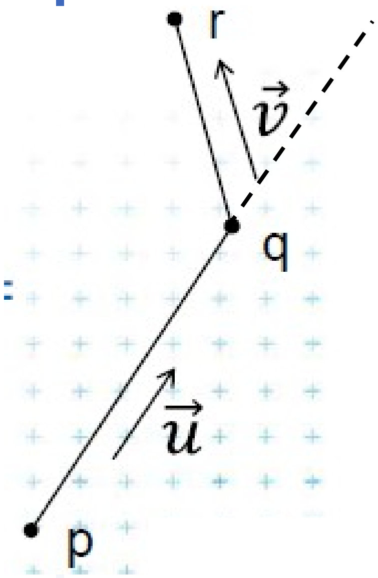
Three points on the plane p, q, r

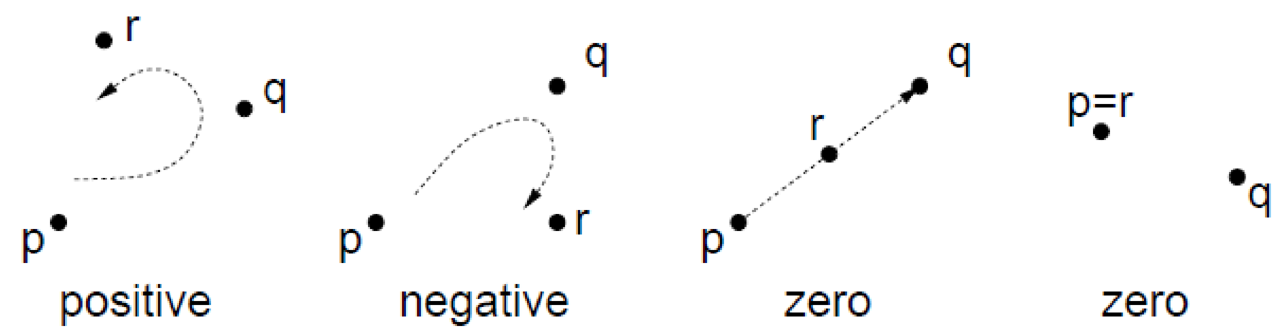


Travel from p to q ($\overrightarrow{pq}$) and check whether the point r lies at the left (CCW), right (CW) of $\overrightarrow{pq}$ or collinear.

Orientation Test

Orientation test can be done in $O(1)$ time!





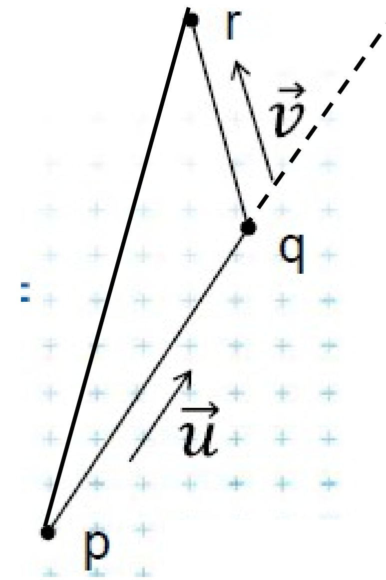
$$\text{orientation}(p,q,r) = \text{sign}\left(\det\begin{bmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{bmatrix}\right)$$

Orientation < 0, for clockwise (CW) p, q, r

$= 0$, when p, q, r are collinear, or coincide

> 0, for counter-clockwise (CCW) p, q, r

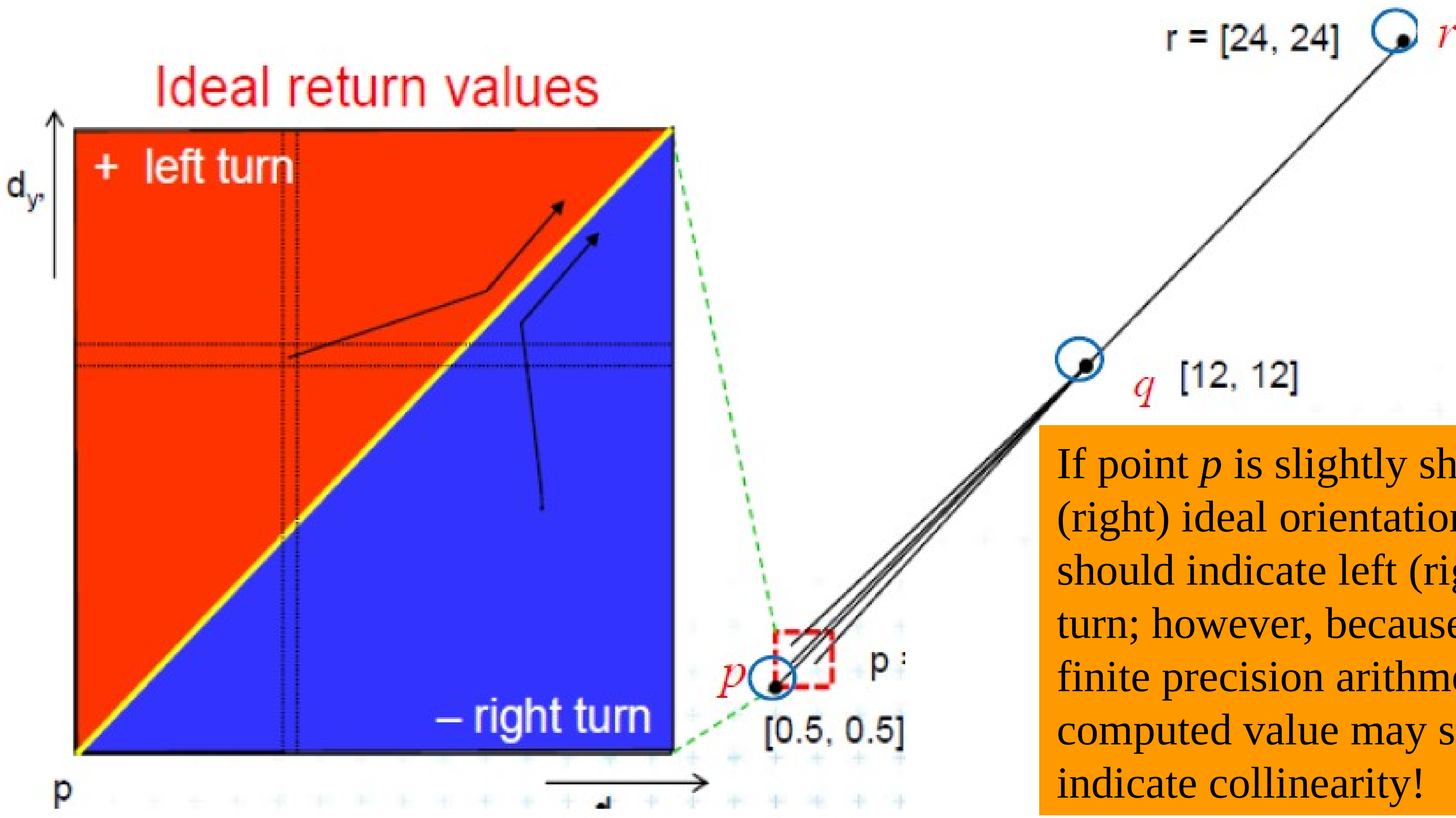
Area of ∘(p, q, r)



Signed area of ∘(p, q, r) = (1/2) $\left(\det \begin{bmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{bmatrix}\right)$

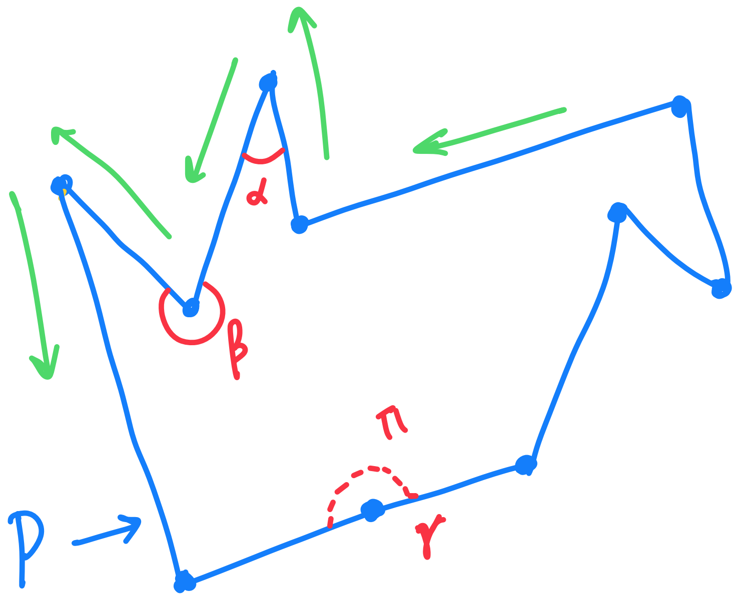
Effect of Finite-Precision Arithmetic

- orientation(p,q,r) = $\mathrm{sign}((p_x - r_x)(q_y - r_y) - (p_y - r_y)(q_x - r_x))$



If point *p* is slightly shifted up (right) ideal orientation test should indicate left (right) turn; however, because of finite precision arithmetic, computed value may still indicate collinearity!

Simple polygons: Convex and reflex angles



internal angles of P

α: strictly convex $< \pi$

β: strictly concave $> \pi$
(also called reflex)

γ: assume convex $= \pi$

(We often consider collinearity as degenerate cases)

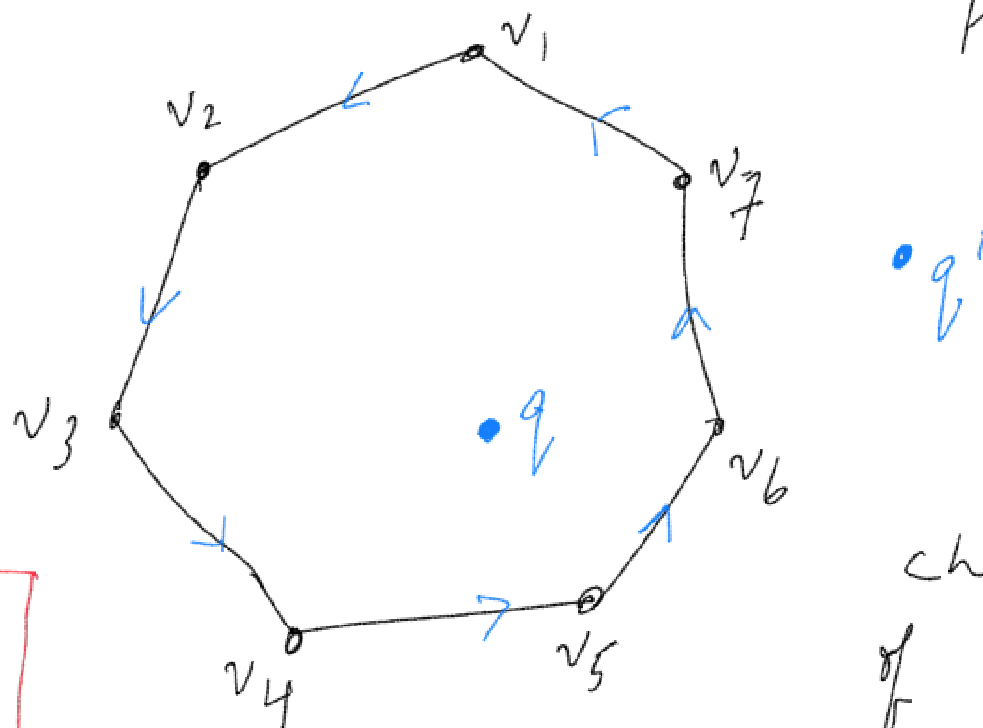
Walk along δP CCW

Left turn $\Rightarrow$ strictly convex

Right turn $\Rightarrow$ strictly reflex

No turn $\Rightarrow$ π

Given a convex polygon P and a query point q, does P include q?



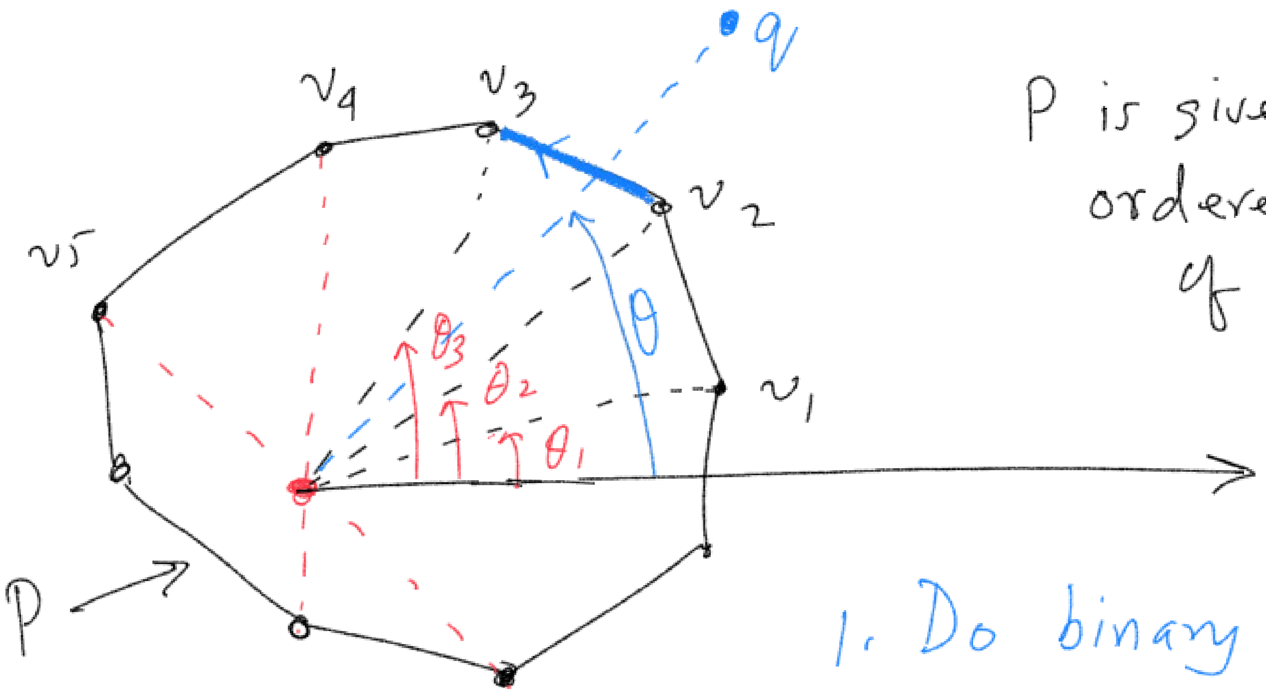
P is described as an ordered sequence of vertices $v_1, v_2, \ldots, v_n$

check orientation of $(v_i, v_{i+1}, q), \forall i$

all CCW $\Rightarrow$ q is included in P.

$\boxed{O(n)}$

Can we do better?



P is given as an ordered sequence of vertices

1. Do binary search over θ

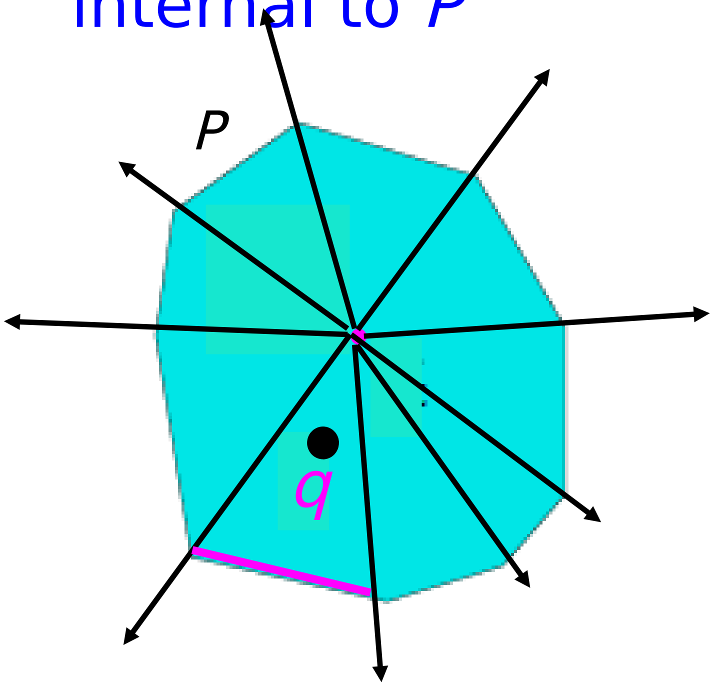
2. Perform only one orientation test.

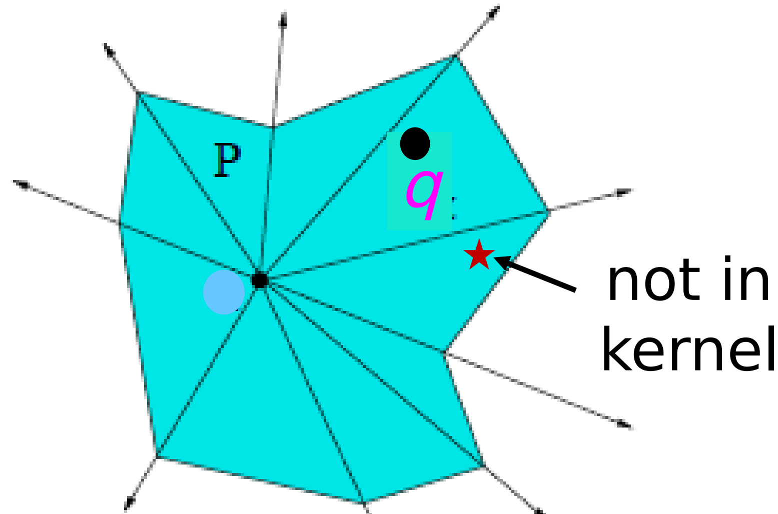
$O(\log n)$

Test for point inclusion

Assuming that the ordered sequence of vertices of *P* is given in terms of *polar coordinates* with respect to an interior point

Problem: Given a convex polygon *P* with *n* vertices and a query point *q*, determine whether or not *q* is internal to *P*



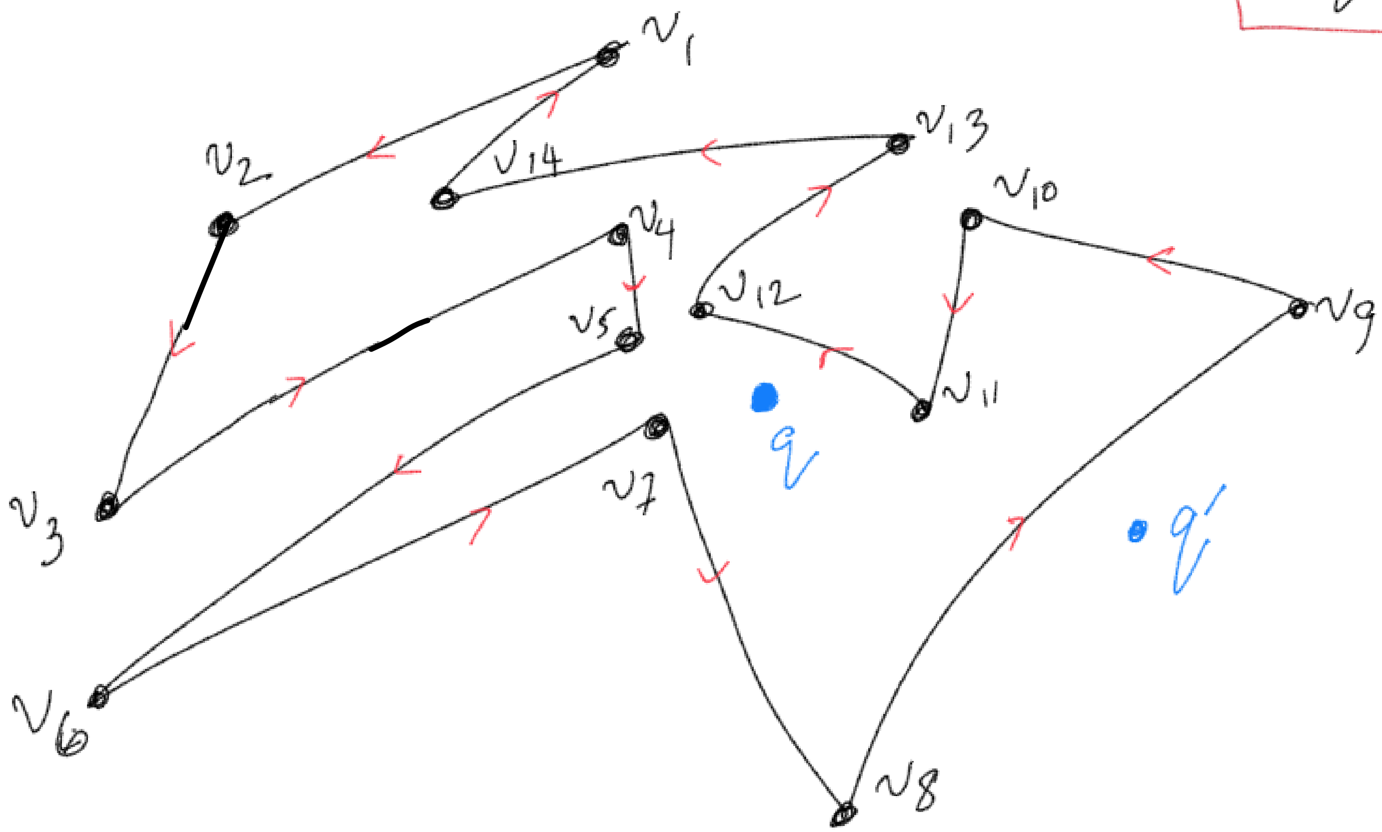


The same technique holds for star-shaped polygons as well, provided the kernel is known

Assume that vertices of *P* have been provided as an ordered sequence of polar coordinates;
The entire plane can be thought as if divided into wedges;
Given a query point *q*, the corresponding wedge can be located by binary search;
Perform a single orientation test *w.r.t.* the corresponding edge of P;

$O(\log n)$ query time, $O(n)$ space

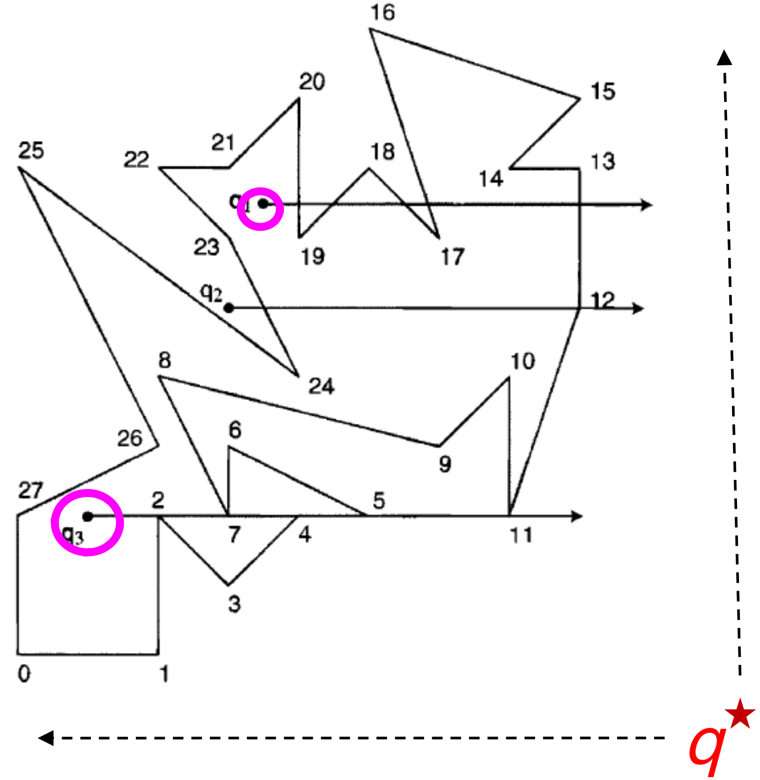
Given a polygon P, and a point q, check whether P includes q



Unidirectional orientation test done as in the convex case, fails here, e.g., while checking orientation (v_{10}, v_{11}, q), (v_{11}, v_{12}, q)

Ray Shooting

J. O'Rourke: *Computational Geometry in C (2nd edition)*, Cambridge Univ. Press, 1998.



Shoot a ray from *q* and count # intersections with the edges of *P*

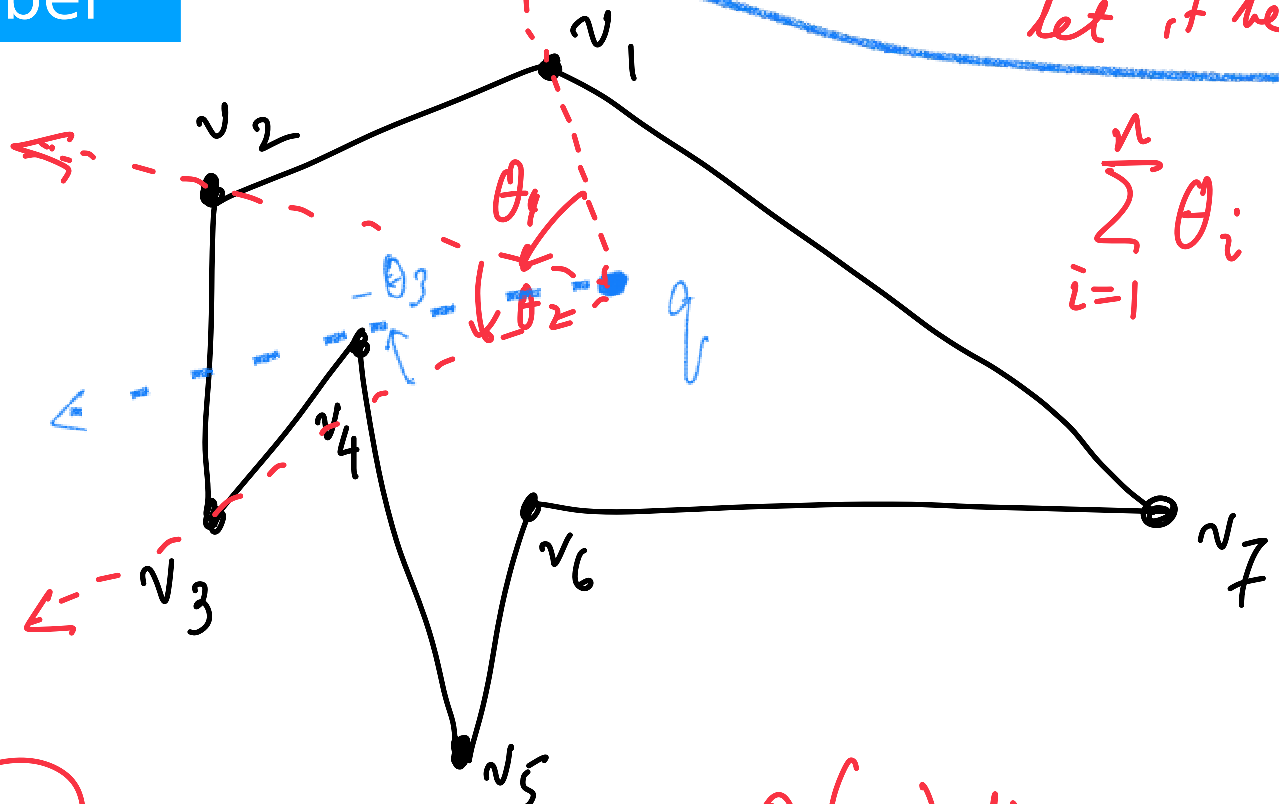
= odd   *q* is in the interior of polygon *P*

= even   *q* is in the exterior of polygon *P*

Beware of degenerate cases!

Winding number

start at a convex vertex, let it be v_1

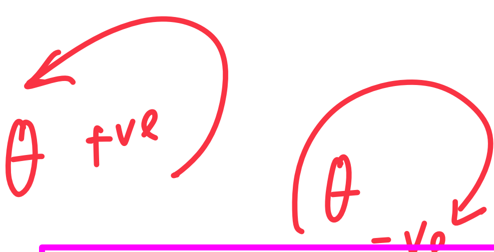


$$\sum_{i=1}^{n} \theta_i = 2\pi$$

if q is included in P.

otherwise $= 0$!

$O(n)$ time



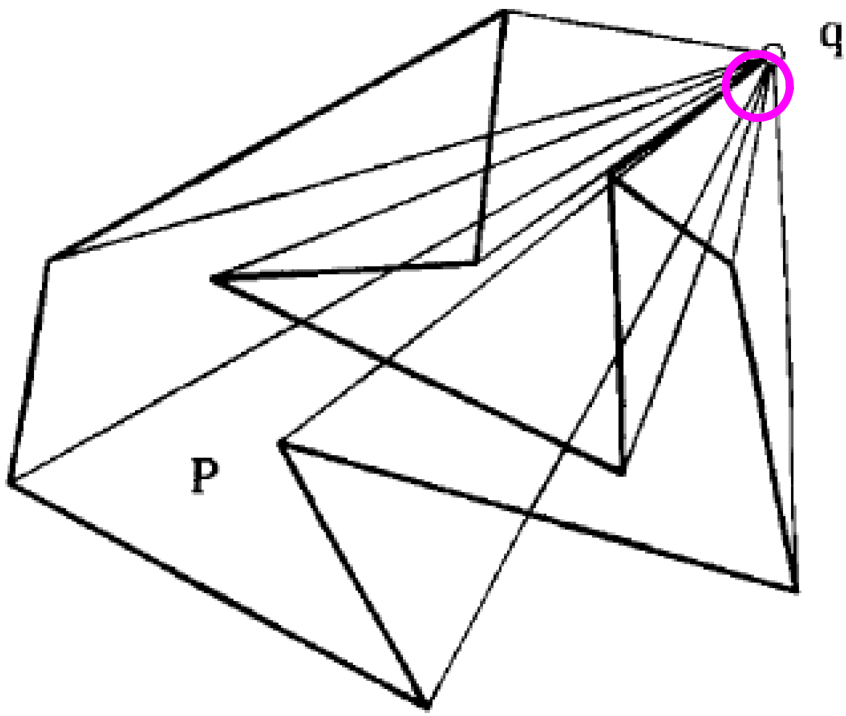
Compute total angular turn around q w.r.t. vertices of P

= 2π q is in the interior of polygon P

= 0 q is in the exterior of polygon P

Winding number

J. O'Rourke: *Computational Geometry in C (2nd edition)*, Cambridge Univ. Press, 1998.

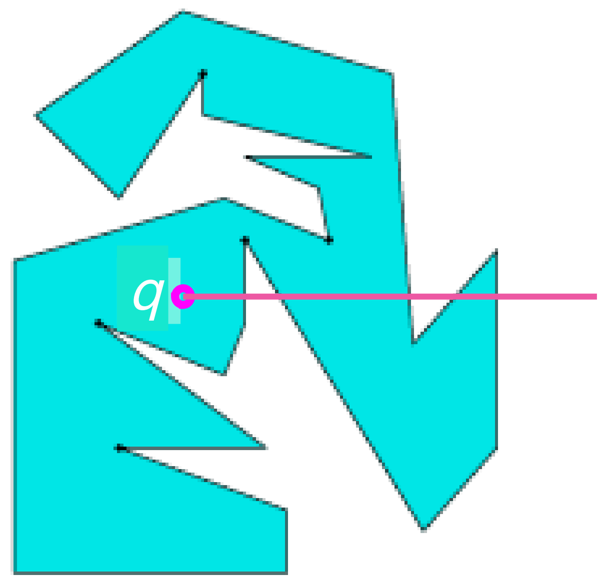


Compute total angular turn around q w.r.t. vertices of P

= 2 q is in the interior of polygon P

= 0 q is in the exterior of polygon P

Problem: Given a simple polygon P with n vertices and a query point q, determine whether or not q is internal to P

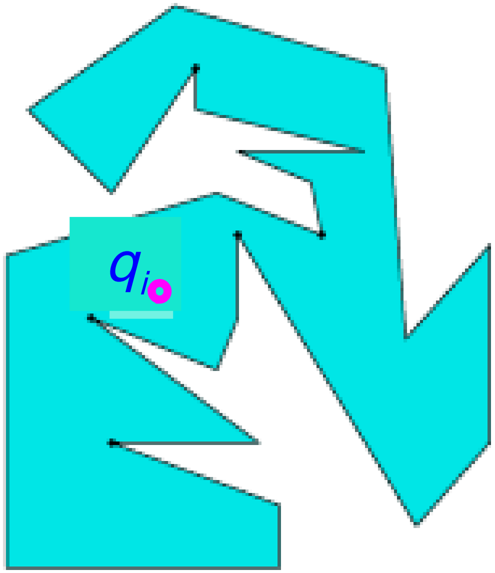


intersections = 5

1. *Ray shooting:* Count # intersections with edges, check parity
$O(n)$ time complexity (degenerate cases cause problems, but overall, the algorithm runs very fast)

2. *Winding number*: Compute algebraic sum of angular turns spanning all vertices; check whether 2 or 0; $O(n)$ time complexity (Note: trigonometric calculations make the algorithm very slow)

Problem (Multiple queries): Given a simple polygon P with n vertices and m query points q_1, q_1, ..., q_m, determine whether they are internal to P



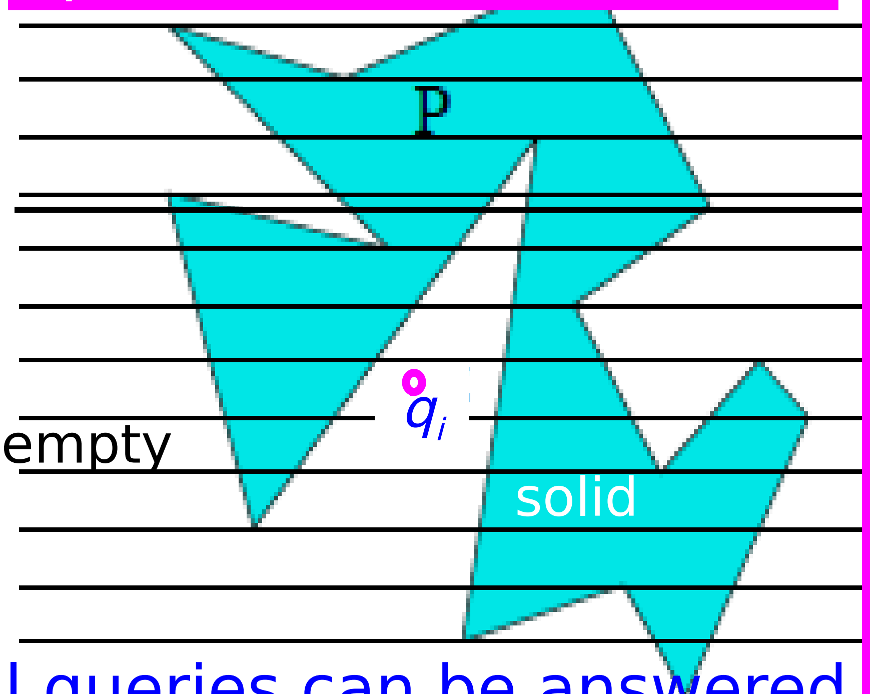
Ray shooting/winding number:
Multiple calls $O(nm)$ time,
$O(n)$ space complexity

Can we improve query-time complexity …?

… at the cost of some pre-processing work or space ..?

Problem (Multiple queries): Given a simple polygon P with n vertices and m query points q_1, q_1, ..., q_m, determine whether or not they are internal to P.

Naive: $O(nm)$ time, $O(n)$ space

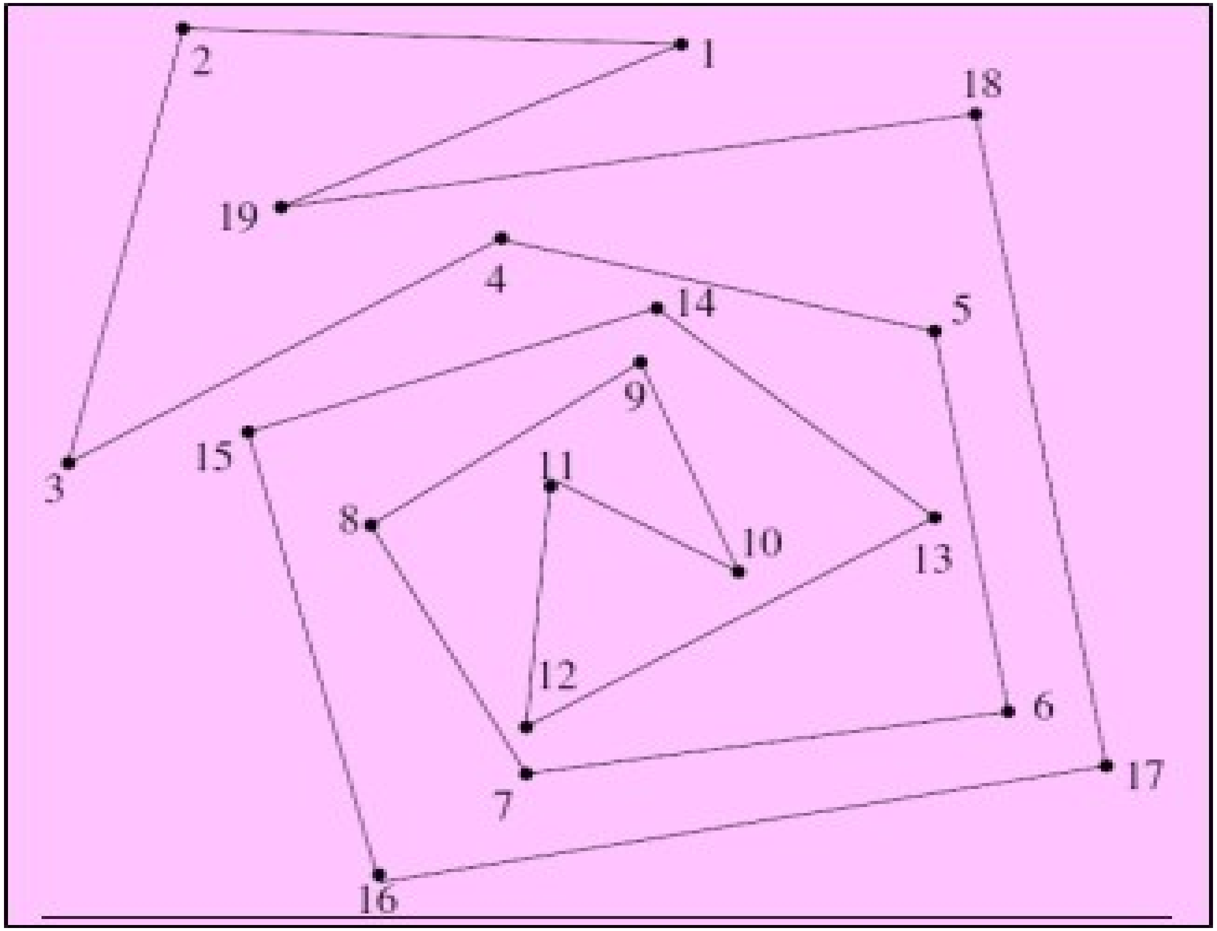


All queries can be answered in $O(m\log n)$ time using $O(n^2 \log n)$ preprocessing time and $O(n^2)$ space

Trapezoidal space partitioning

- Draw n horizontal lines through each vertex of P dividing the plane into $(n - 1 + 2) = n + 1$ horizontal strips called "slabs";
- Label trapezoids as *solid* or *empty*; store descriptions;
- Sort the slabs by y-coordinates as a *preprocessing* step;
- Perform a binary search to locate the slab that contains the query point q_i;
- Within this slab, perform a binary search to locate trapezoid that contains the query point q_i;

Beware of trapezoidation in spiral polygons …….; see how pre-processing complexities are determined



Homework Set - 01:

1. Given n points on 2D-plane, propose an algorithm to construct a simple polygon P with all the given points as vertices, and only those. Provide its proof of correctness and deduce its time complexity. (A simple polygon is one in which no two edges intersect each other excepting possibly at their endpoints.
2. (a) A convex polygon P is given as counter-clockwise ordered sequence of n vertices in general positions, whose locations are supplied as (x, y) co-ordinates on the x-y plane. Given a query point q, propose an algorithm to determine in $O(\log n)$ time and $O(n)$ space, including pre-processing, if any, whether or not P includes q.

   (b) Write a code to implement your algorithm. Construct a convex polygon with 30 vertices, and show your results for a few internal and external points.

Submit solutions via Moodle. **Due:** 23:55, January 21, 2022; Credit: 10%