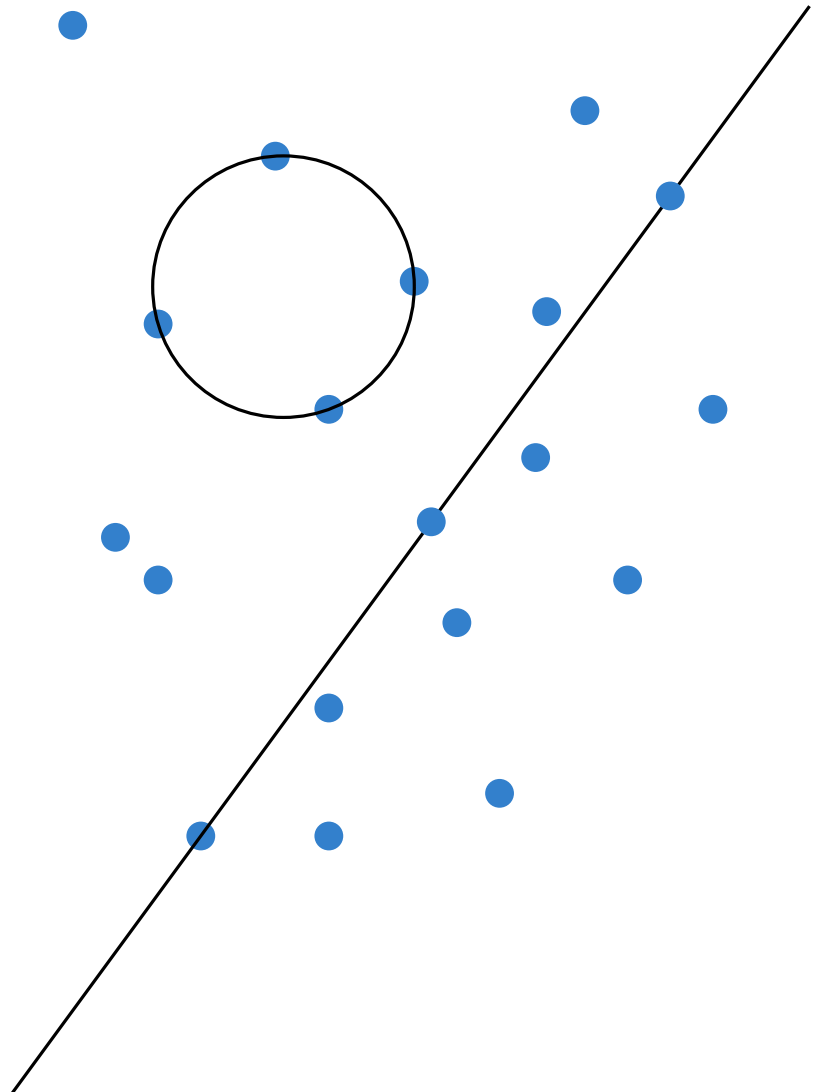
Implementing Geometric Algorithms

Implementing Geometric Algorithms

Difficulties:

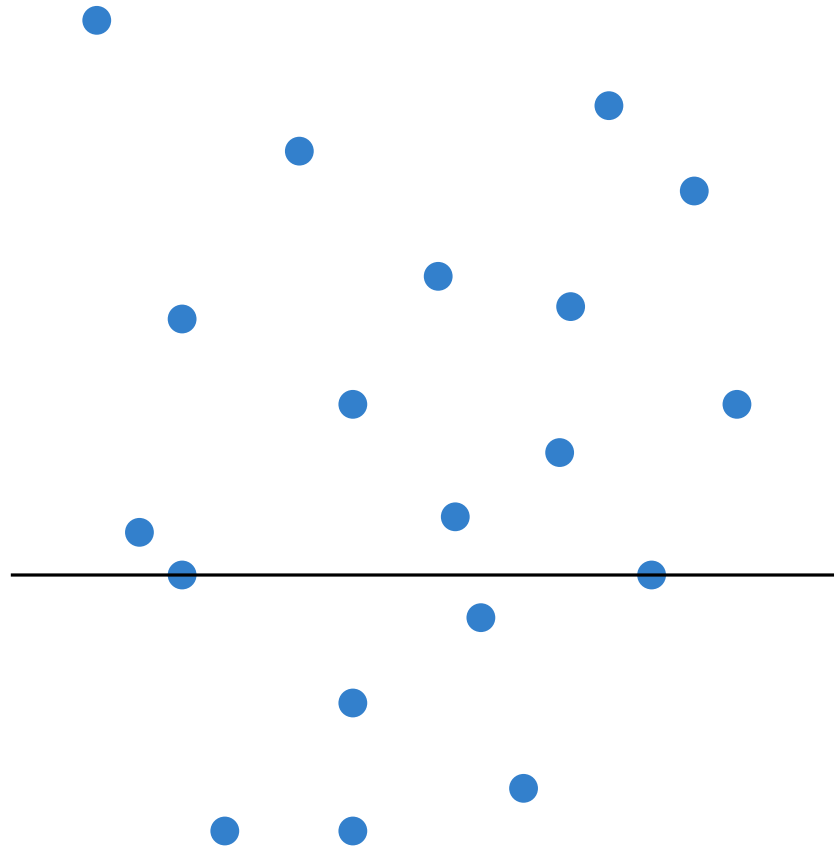
- Dealing with Degenerate Input



Implementing Geometric Algorithms

Difficulties:

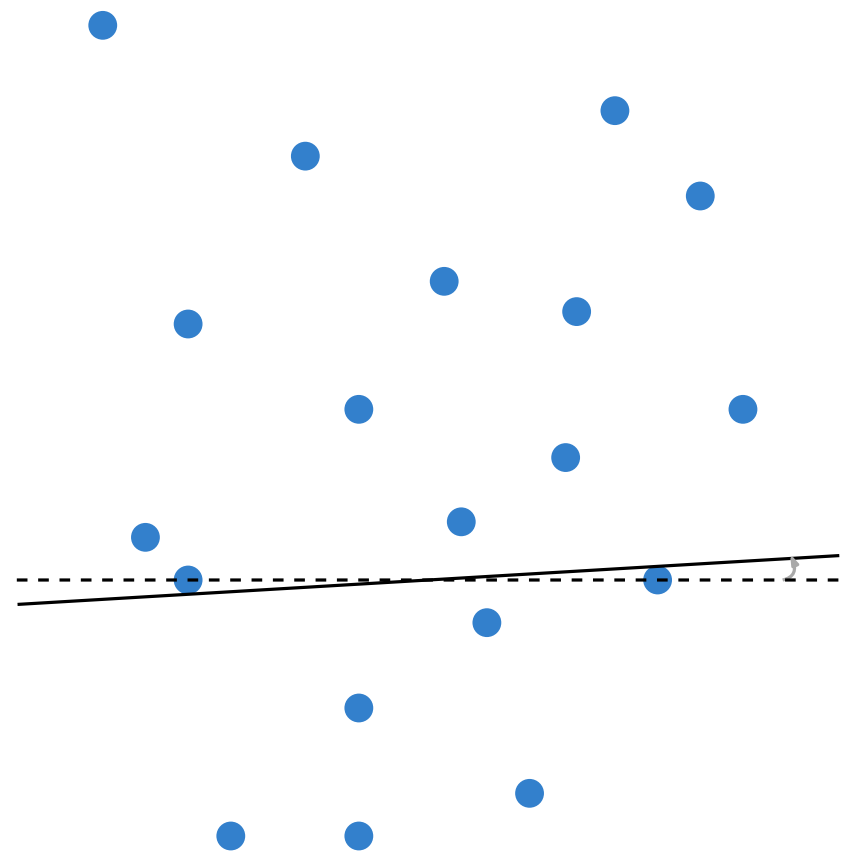
- Dealing with Degenerate Input



Implementing Geometric Algorithms

Difficulties:

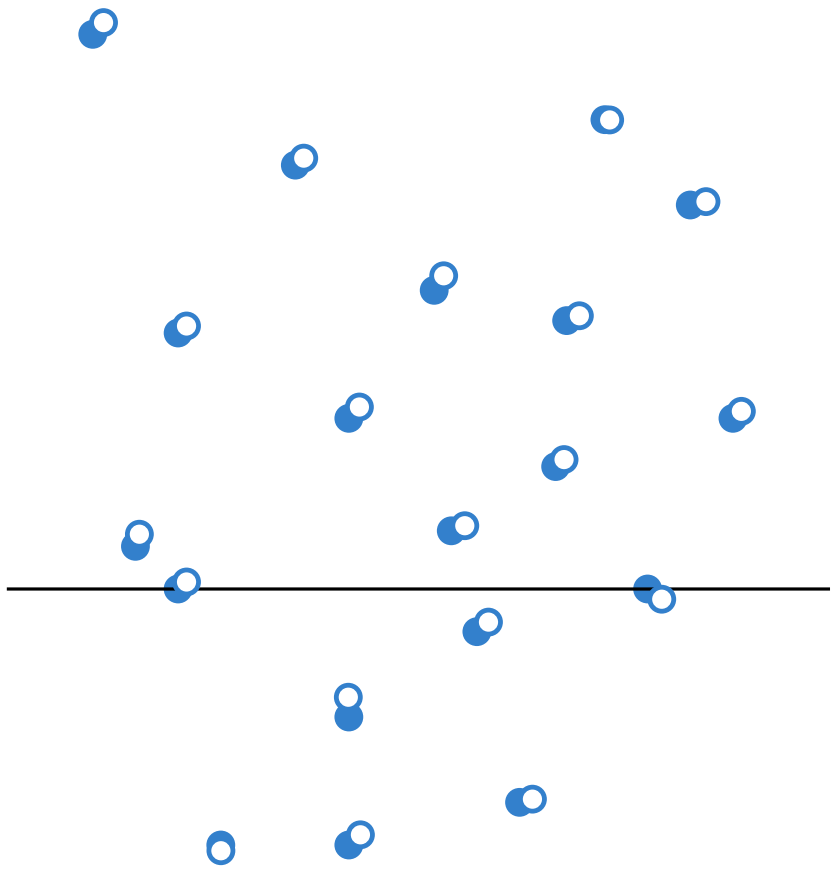
- Dealing with Degenerate Input
  - Adapt the algorithm



Implementing Geometric Algorithms

Difficulties:

- Dealing with Degenerate Input
  - Adapt the algorithm
  - Symbolic pertubation



Implementing Geometric Algorithms

Difficulties:

- Representing real numbers

Implementing Geometric Algorithms

Difficulties:

- Representing real numbers
  - Use Float or Double?

Implementing Geometric Algorithms

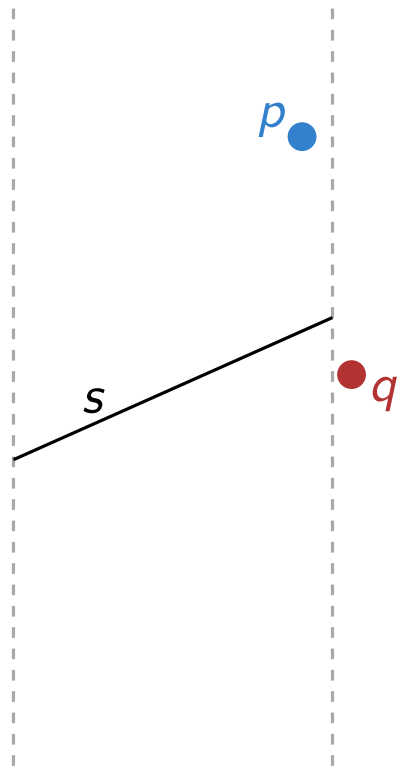
Difficulties:

- Representing real numbers
  - Use Float or Double?
    - Not all $x \in \mathbb{R}$ representable

Implementing Geometric Algorithms

Difficulties:

- Representing real numbers
  - Use Float or Double?
    - Not all $x \in \mathbb{R}$ representable
    - Errors accumulate

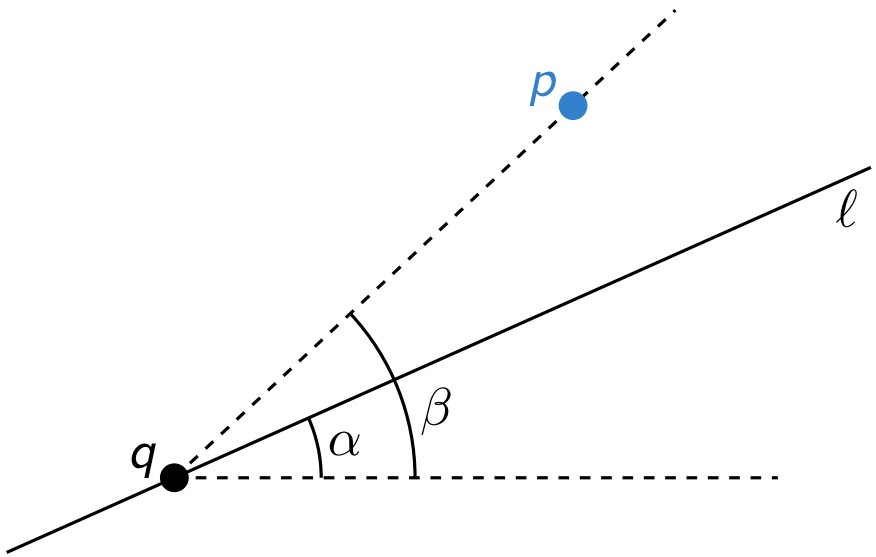


Implementing Geometric Algorithms

Difficulties:

- Representing real numbers
  - Use Float or Double?
    - Not all $x \in \mathbb{R}$ representable
    - Errors accumulate
  - Use better predicates?

isLeftOfLine(p, ℓ) :

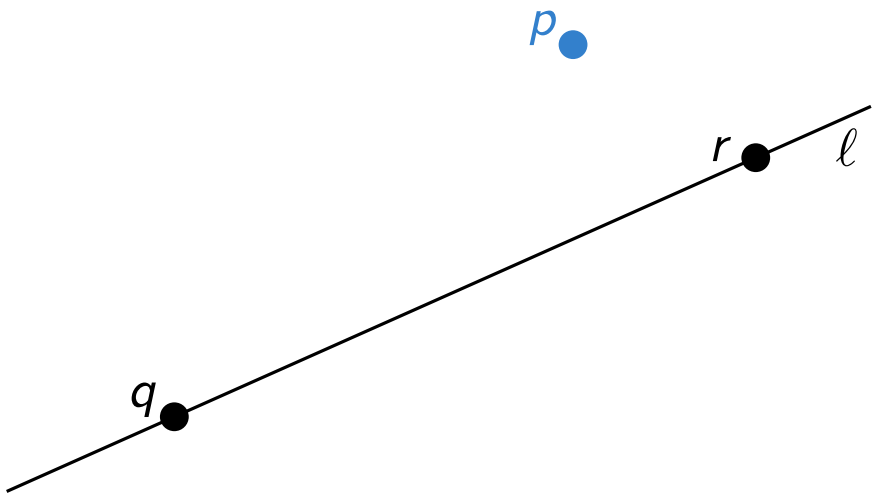


Implementing Geometric Algorithms

Difficulties:

- Representing real numbers
  - Use Float or Double?
    - Not all $x \in \mathbb{R}$ representable
    - Errors accumulate
  - Use better predicates?

isLeftOfLine(p, ℓ) :



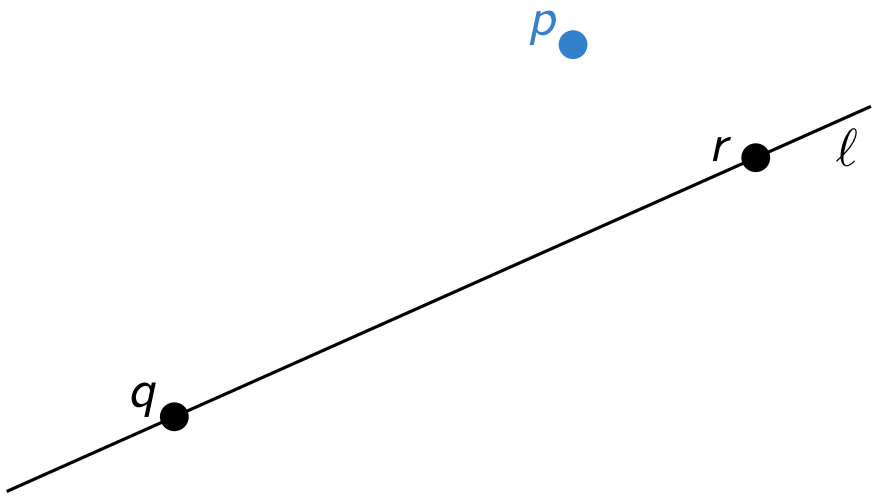
$$\begin{vmatrix} q_x & q_y & 1 \\ r_x & r_y & 1 \\ p_x & p_y & 1 \end{vmatrix} < 0$$

Implementing Geometric Algorithms

Difficulties:

- Representing real numbers
  - Use Float or Double?
    - Not all $x \in \mathbb{R}$ representable
    - Errors accumulate
  - Use better predicates?

isLeftOfLine(p, ℓ) :



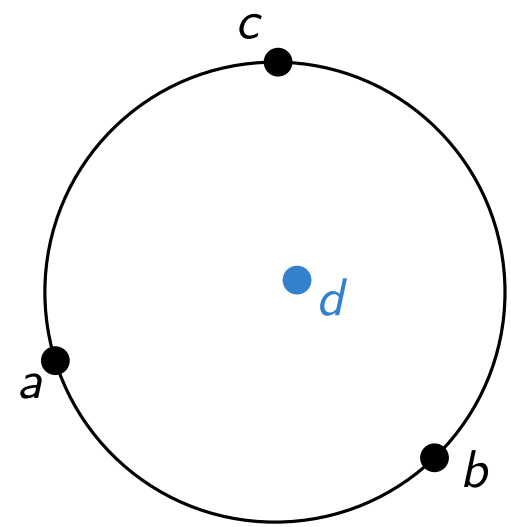
$$\begin{vmatrix} q_x - p_x & q_y - p_y \\ r_x - p_x & r_y - p_y \end{vmatrix} < 0$$

Implementing Geometric Algorithms

Difficulties:

- Representing real numbers
  - Use Float or Double?
    - Not all $x \in \mathbb{R}$ representable
    - Errors accumulate
  - Use better predicates?

$InCircle(a, b, c, d)$:



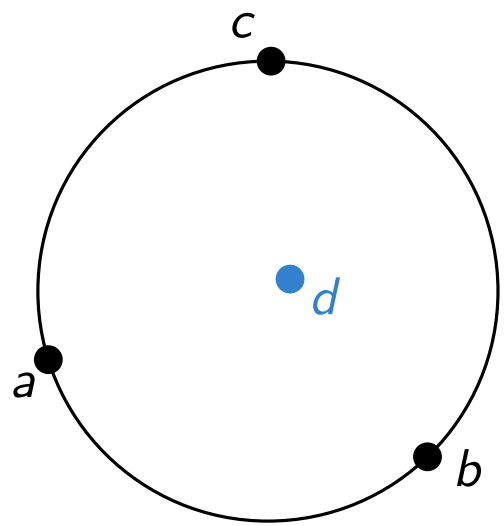
$$\begin{vmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{vmatrix} < 0$$

Implementing Geometric Algorithms

Difficulties:

- Representing real numbers
  - Use Float or Double?
    - Not all $x \in \mathbb{R}$ representable
    - Errors accumulate
  - Use better predicates?
    - Does not really avoid the problem:
      https://youtu.be/3DLfkWWw_Tg?t=237

$InCircle(a, b, c, d)$:



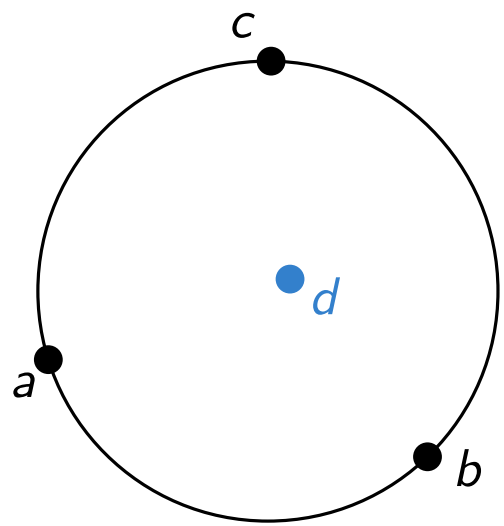
$$\begin{vmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{vmatrix} < 0$$

Implementing Geometric Algorithms

Difficulties:

- Representing real numbers
  - Use Float or Double?
    - Not all $x \in \mathbb{R}$ representable
    - Errors accumulate
  - Use better predicates?
    - Does not really avoid the problem:
      https://youtu.be/3DLfkWWw_Tg?t=237
  - Use fixed precision numbers
  - Use arbitrary precision numbers

$InCircle(a, b, c, d)$:



$$\begin{vmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{vmatrix} < 0$$

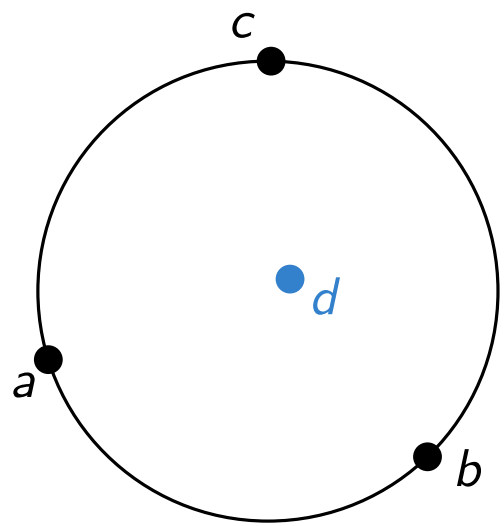
Implementing Geometric Algorithms

Difficulties:

- Representing real numbers
  - Use Float or Double?
    - Not all $x \in \mathbb{R}$ representable
    - Errors accumulate
  - Use better predicates?
    - Does not really avoid the problem:
      https://youtu.be/3DLfkWWw_Tg?t=237
  - Use fixed precision numbers
  - Use arbitrary precision numbers

$\Longrightarrow$ Use CGAL, LEDA, etc.

$InCircle(a, b, c, d)$:

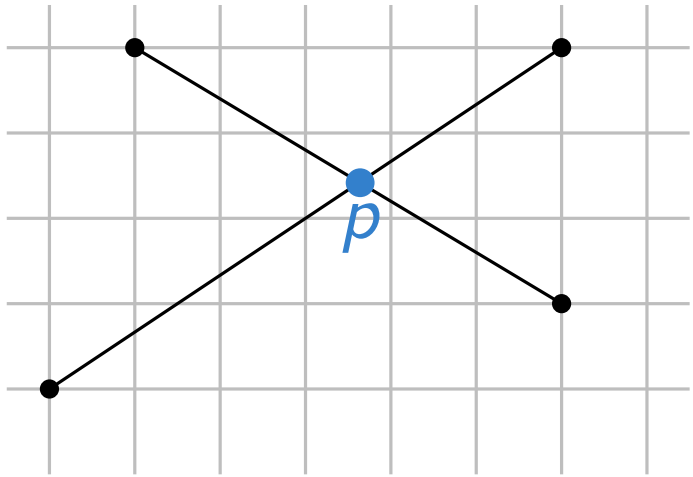


$$\begin{vmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{vmatrix} < 0$$

Implementing Geometric Algorithms

Difficulties:

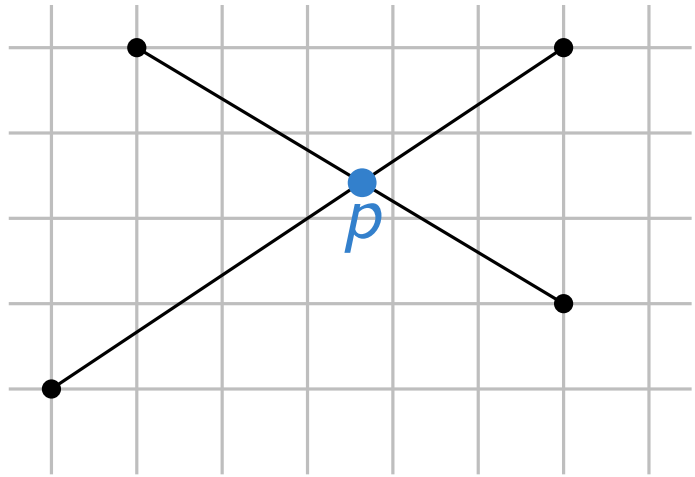
- Creating real numbers
  - Creating new geometry increases complexity



Implementing Geometric Algorithms

Difficulties:

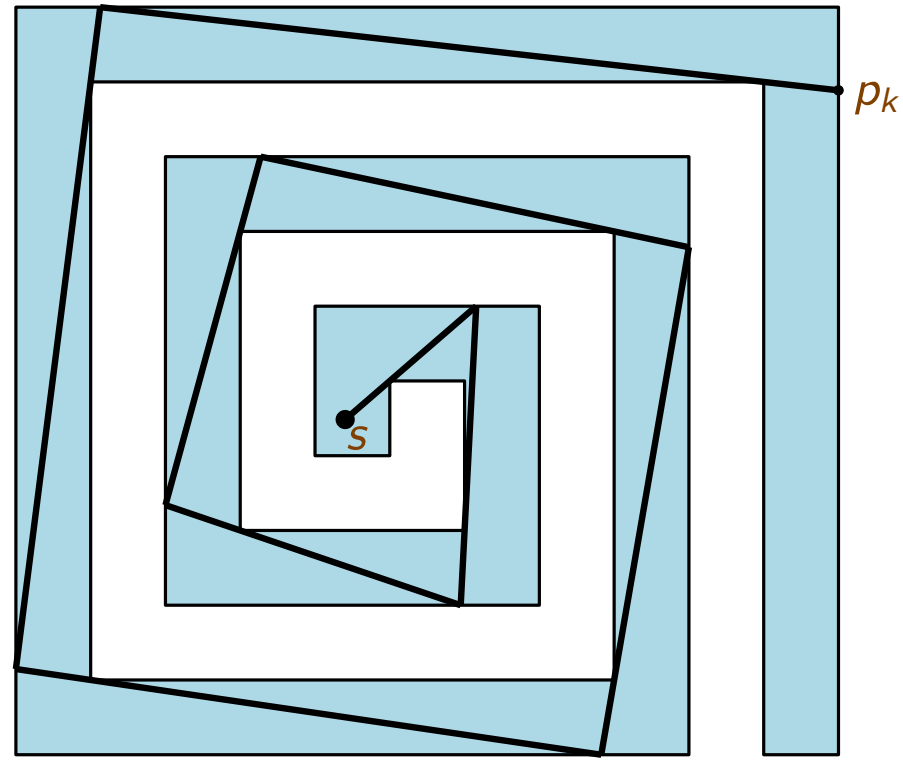
- Creating real numbers
  - Creating new geometry increases complexity
  - Theory: primitives in $O(1)$ time
    Practice: not so much



Implementing Geometric Algorithms

Difficulties:

- Creating real numbers
  - Creating new geometry increases complexity
  - Theory: primitives in $O(1)$ time
    Practice: not so much
  - Even a problem in theory:

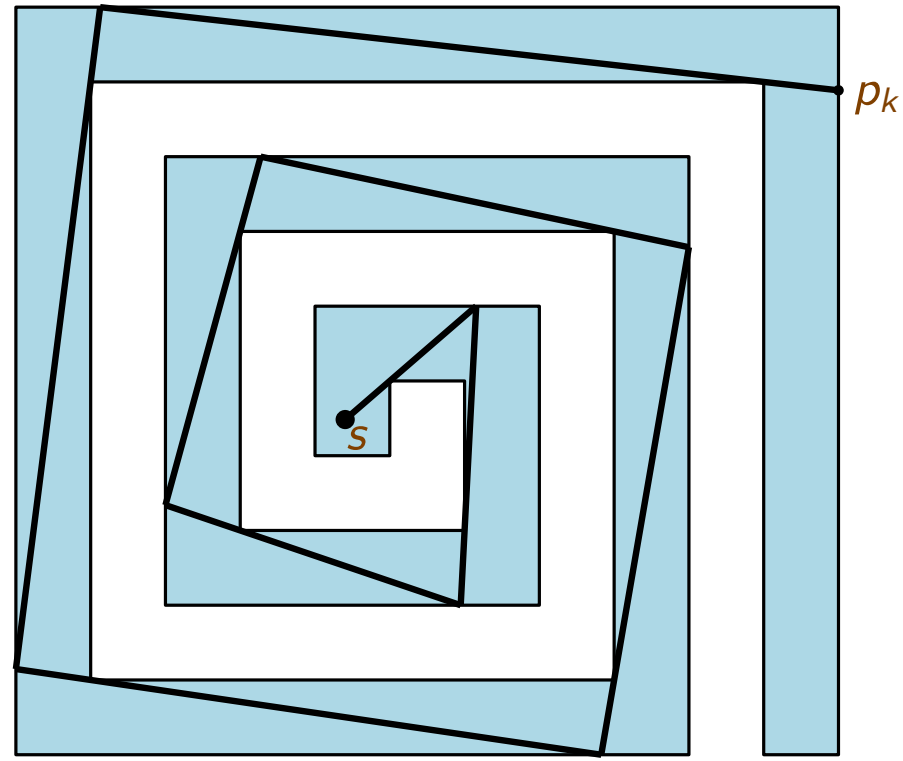


Lemma.
A MinLinkPath of length k between s and t in a simple polygon whose vertices, as well as s and t, have bit-complexity $\log n$, may contain vertices of bit-complexity $\Omega(k \log n)$.

Implementing Geometric Algorithms

Difficulties:

- Creating real numbers
  - Creating new geometry increases complexity
  - Theory: primitives in $O(1)$ time
    Practice: not so much
  - Even a problem in theory:



Lemma.
A MinLinkPath of length k between s and t in a simple polygon whose vertices, as well as s and t, have bit-complexity $\log n$, may contain vertices of bit-complexity $\Omega(k \log n)$.

$\Longrightarrow$ In Real RAM we can compute a MinLinkPath in $O(n \log n)$ time.
However, if we measure bit complexity it takes $O(n^2 \log n)$ time.

Implementing Geometric Algorithms

$\Longrightarrow$ Use CGAL, LEDA, etc.