Geometric Problems with Imprecise Input Points

Maarten Löffler   Marc van Kreveld

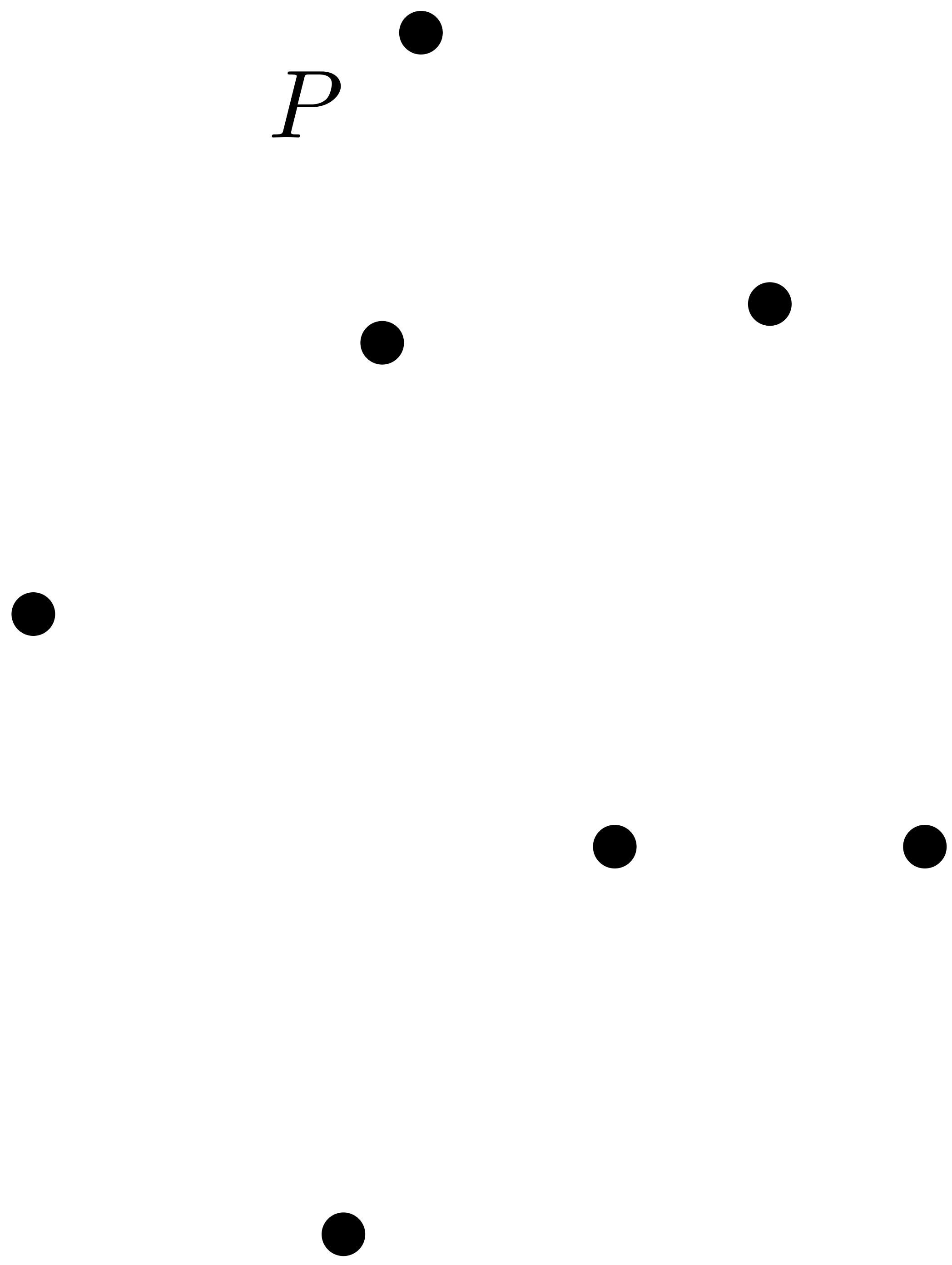
Center for Geometry, Imaging
and Virtual Environments

Utrecht University

Overview

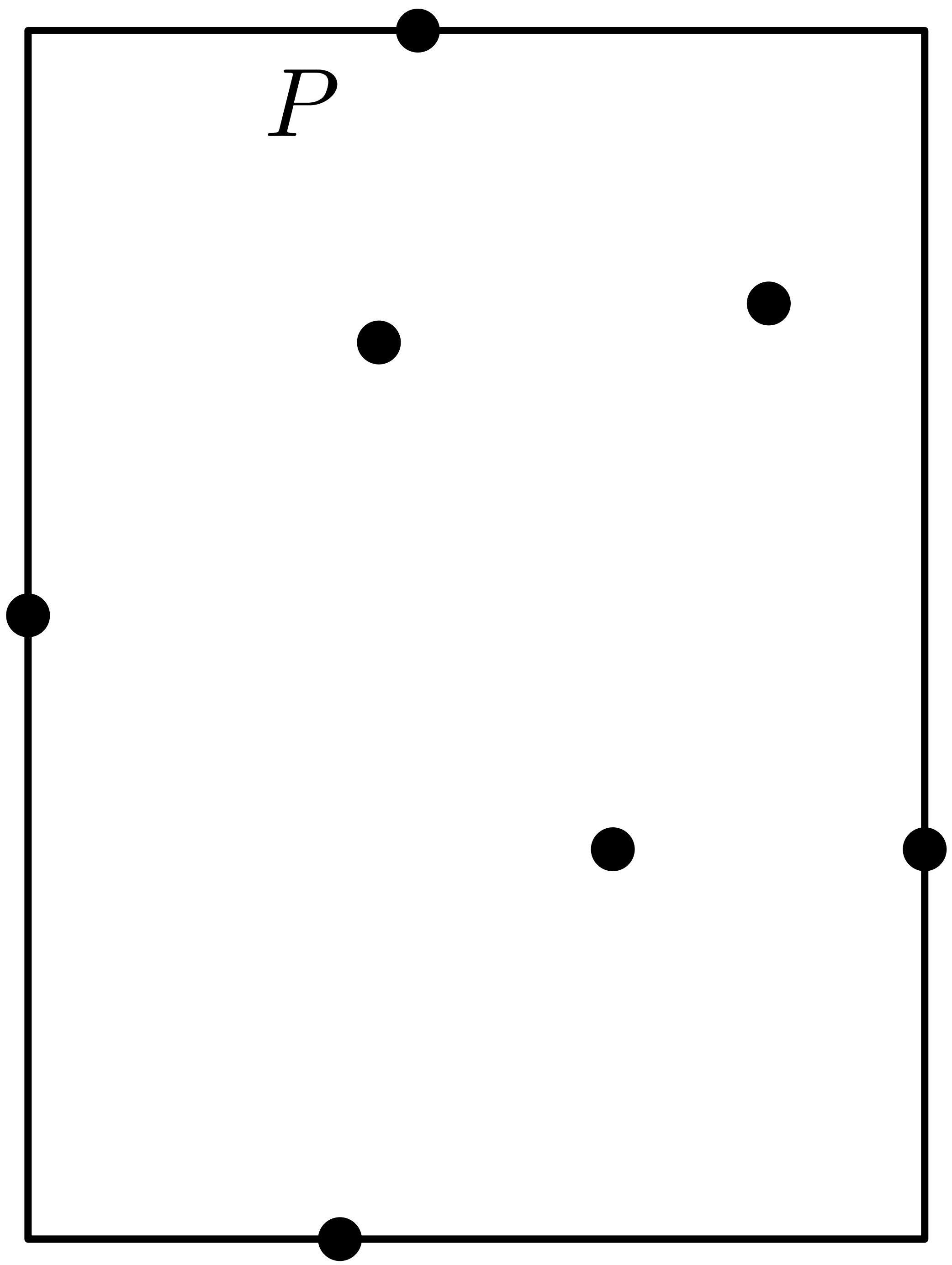
- Introduction
  - Geometric problems
  - Imprecise input points
- Overview of problems and results
- Algorithms
  - Largest diameter of squares
  - Smallest diameter of squares
- Concluding remarks

Geometric Structures on Point Sets



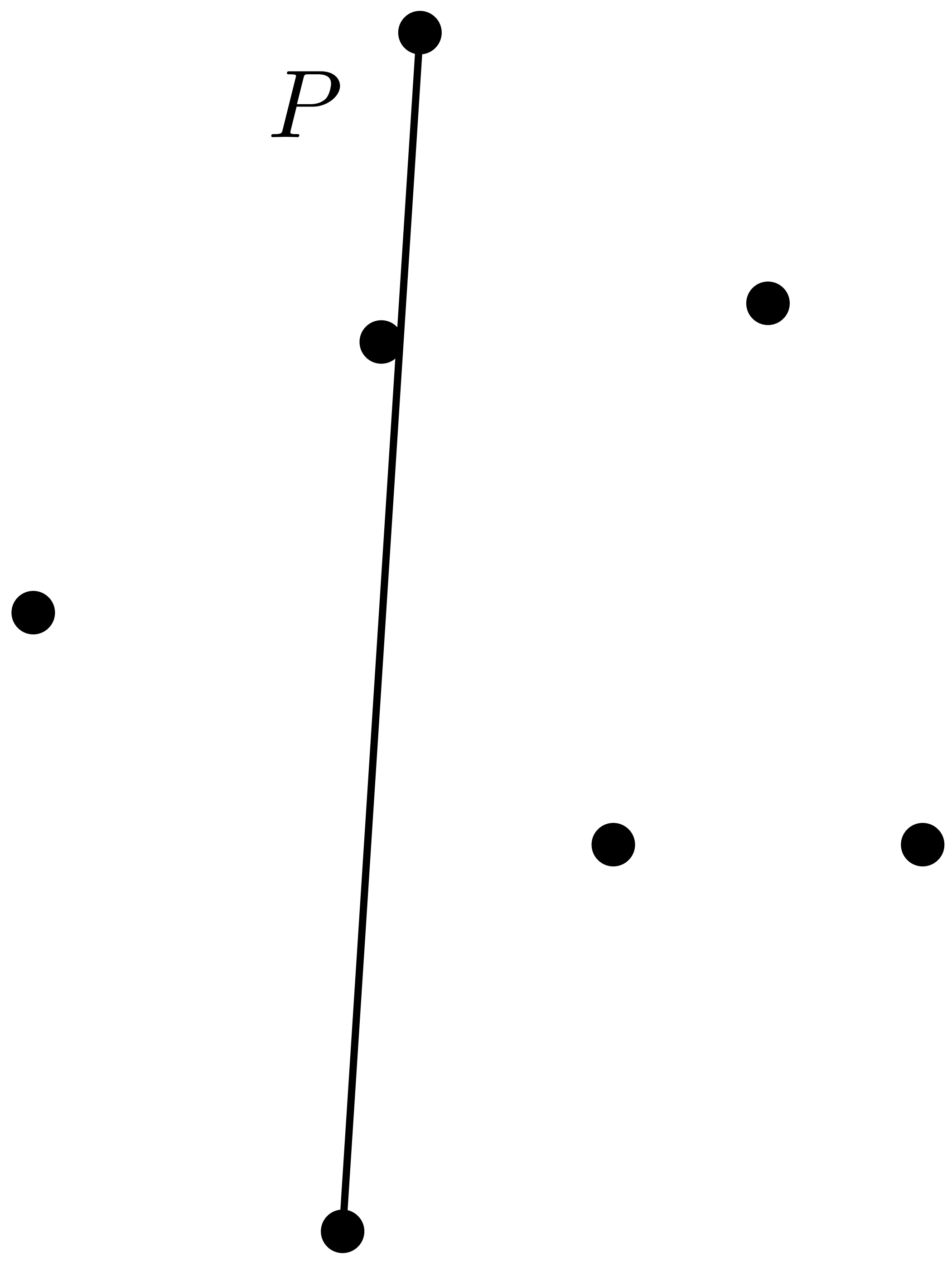
- Given a set P of n points in the plane
- Geometric structures:

Geometric Structures on Point Sets



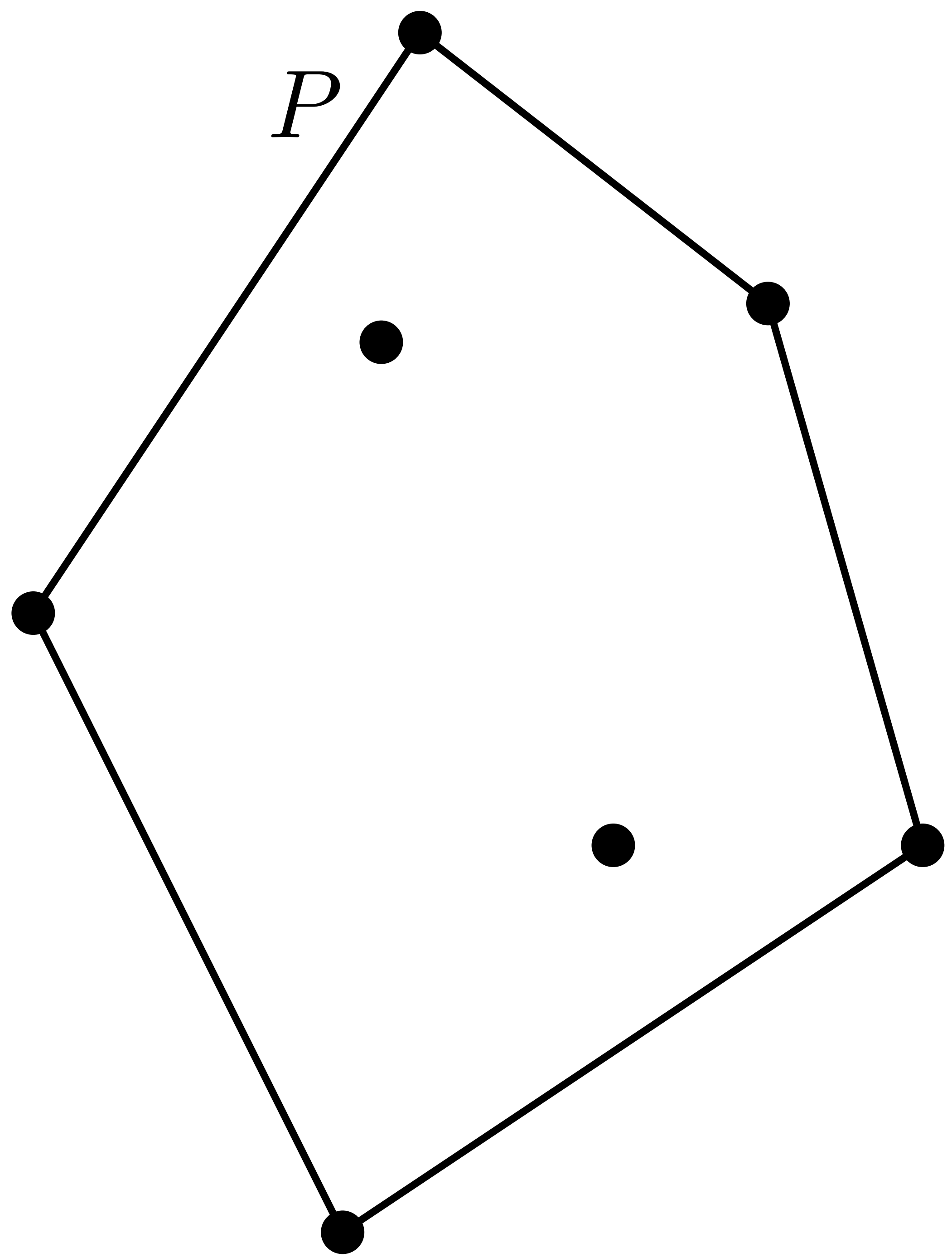
- Given a set P of n points in the plane
- Geometric structures:
  - Bounding box

Geometric Structures on Point Sets



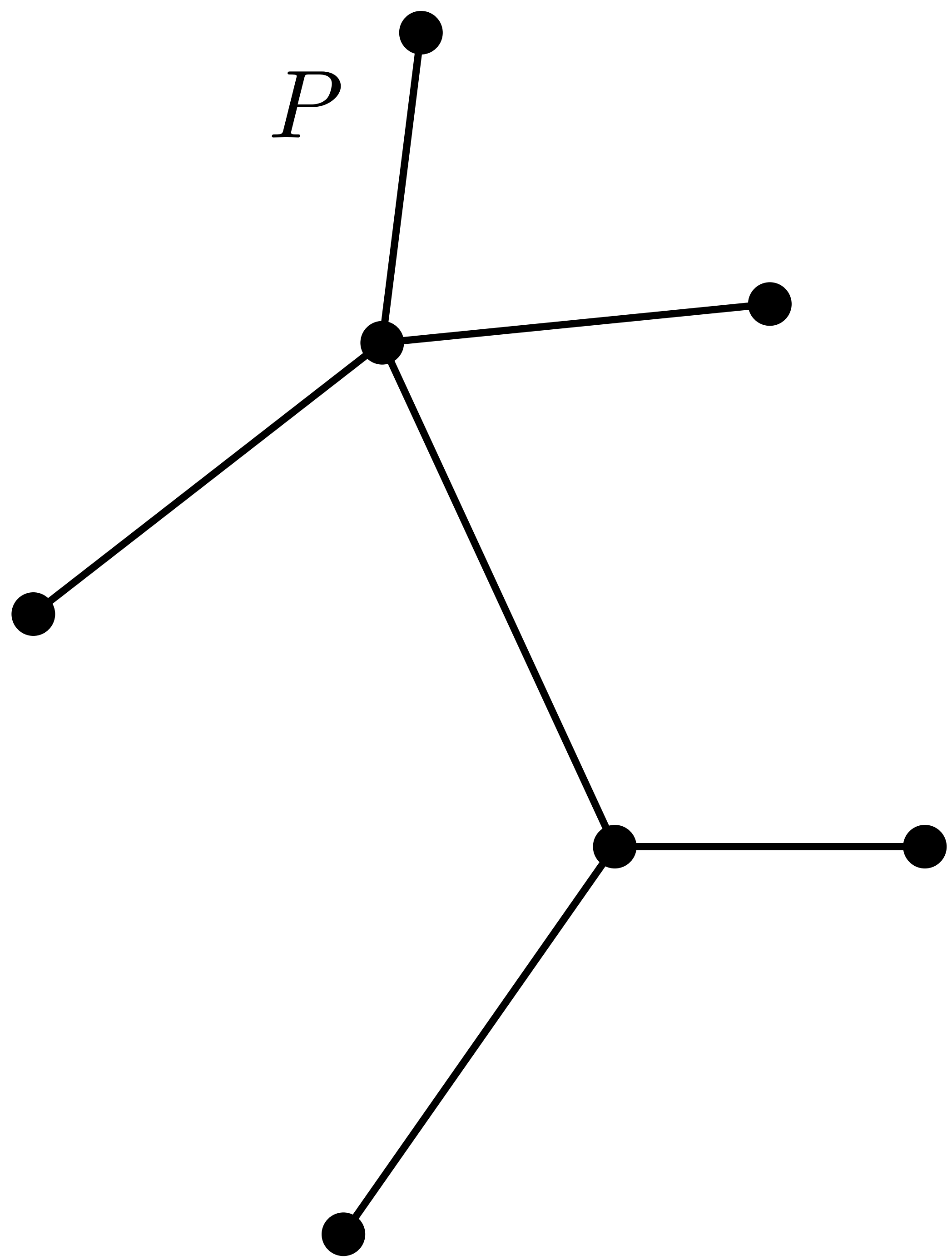
- Given a set P of n points in the plane
- Geometric structures:
  - Bounding box
  - Diameter

Geometric Structures on Point Sets



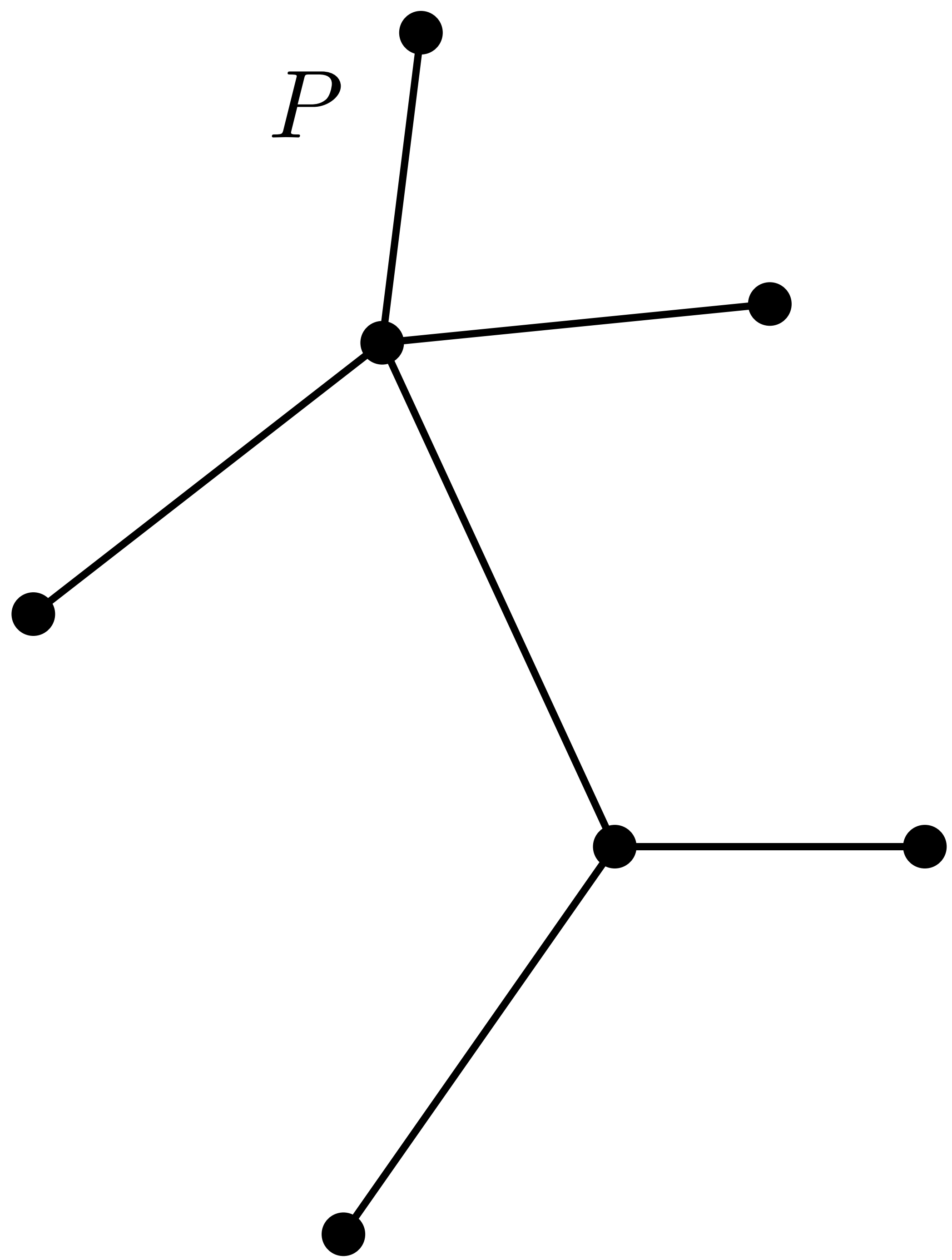
- Given a set P of n points in the plane
- Geometric structures:
  - Bounding box
  - Diameter
  - Convex hull

Geometric Structures on Point Sets



- Given a set P of n points in the plane
- Geometric structures:
  - Bounding box
  - Diameter
  - Convex hull
  - Minimum spanning tree

Geometric Structures on Point Sets

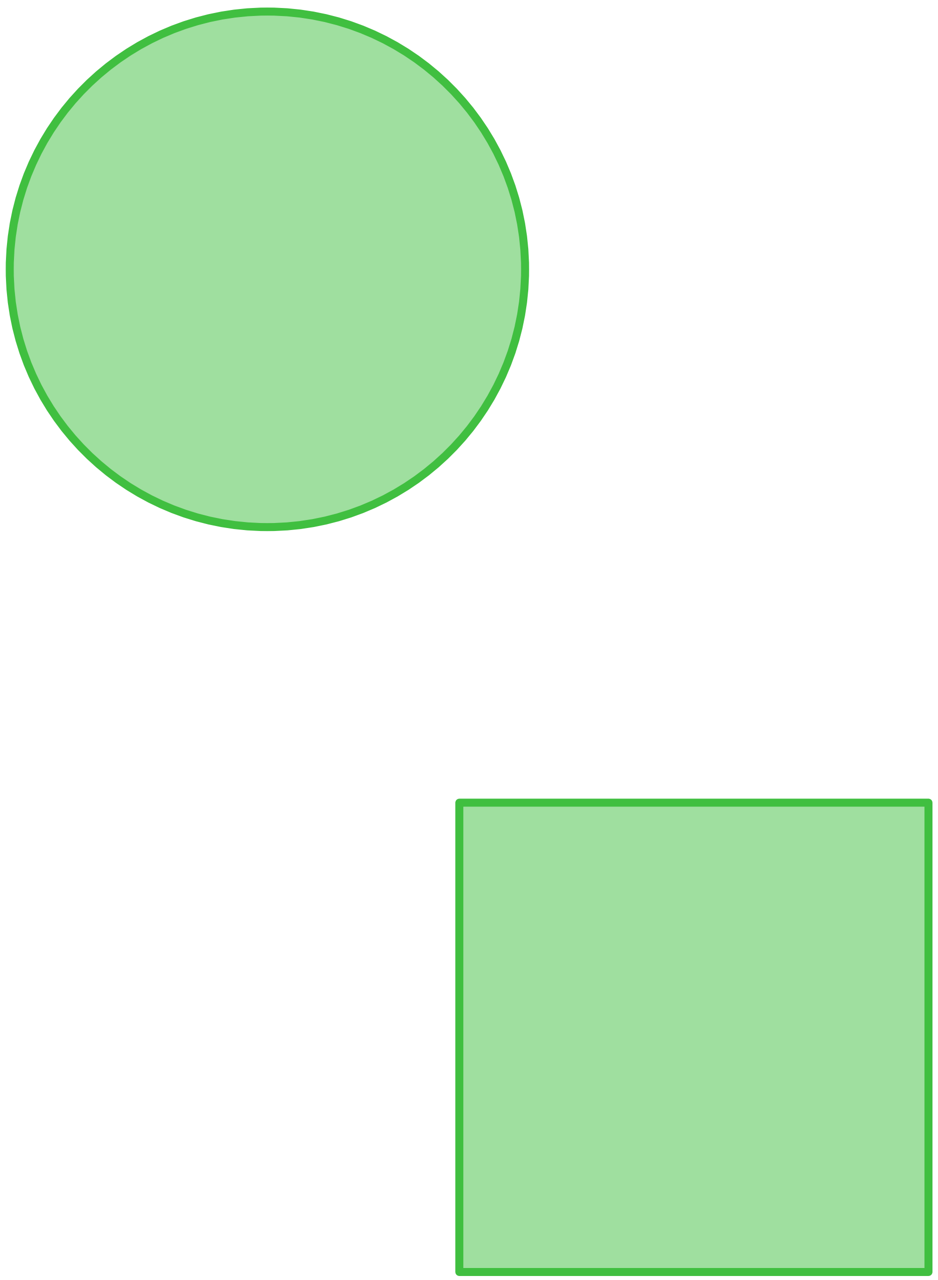


- Given a set P of n points in the plane
- Geometric structures:
  - Bounding box
  - Diameter
  - Convex hull
  - Minimum spanning tree
- Many others
- Optimal algorithms are known

Imprecision

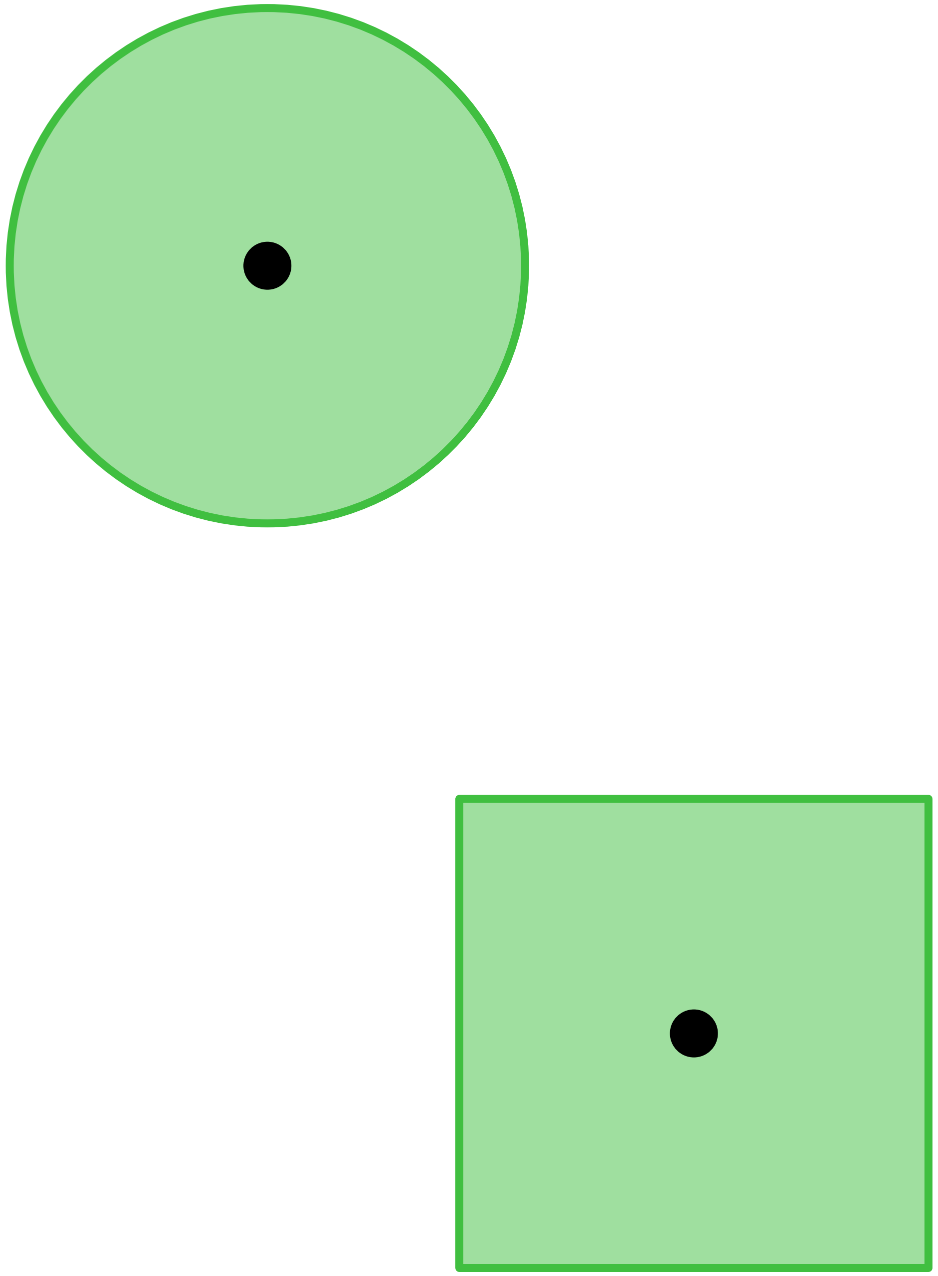
- Traditional algorithms assume exact input
- In practice, input data is often not exact
  - Measured from the real world
  - Stored with limited precision
  - Computed by inexact algorithms
- Output of traditional algorithms is unreliable
- Given exact description of input imprecision, we can exactly predict output imprecision

Imprecise Points



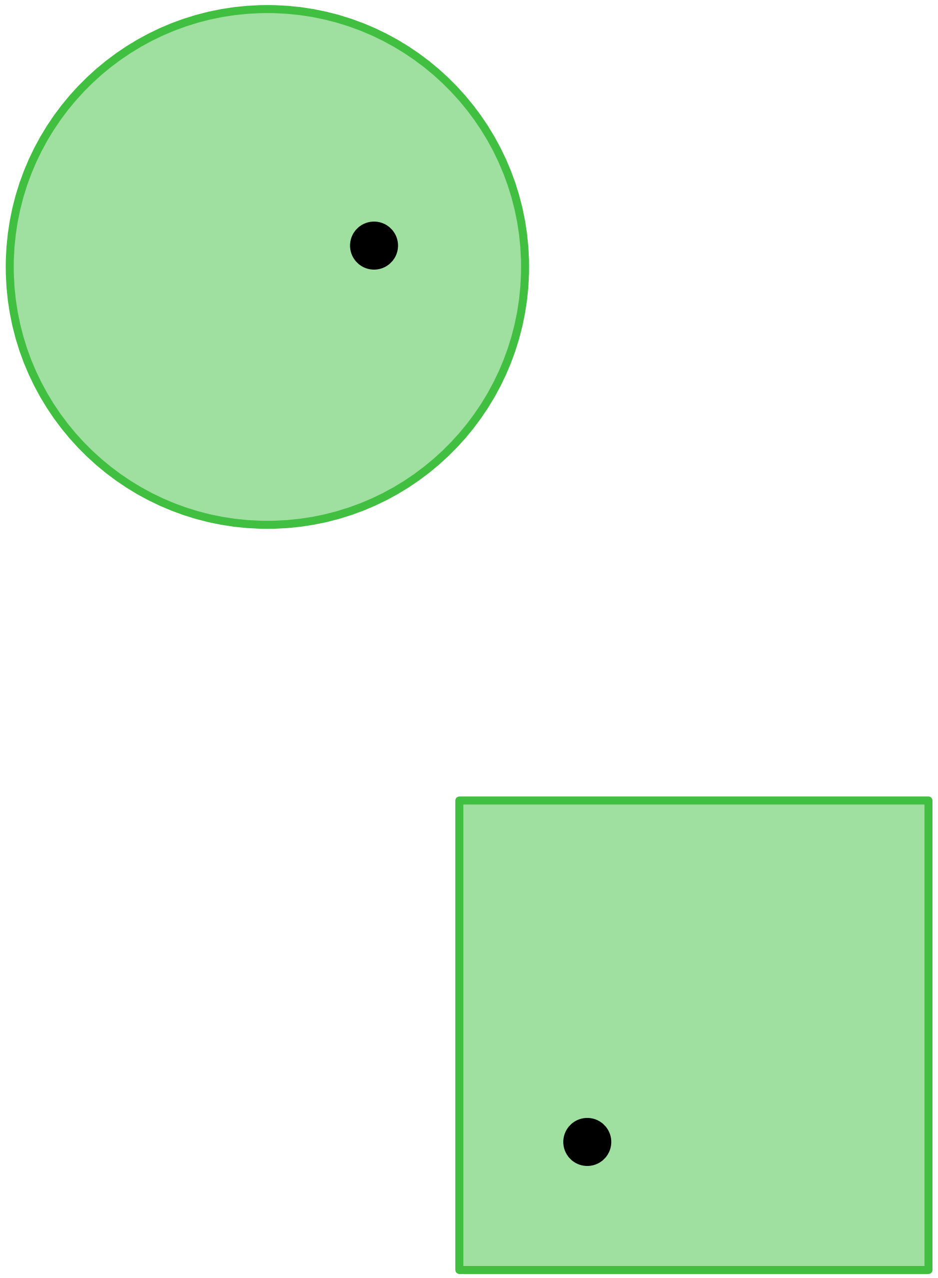
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Imprecise Points



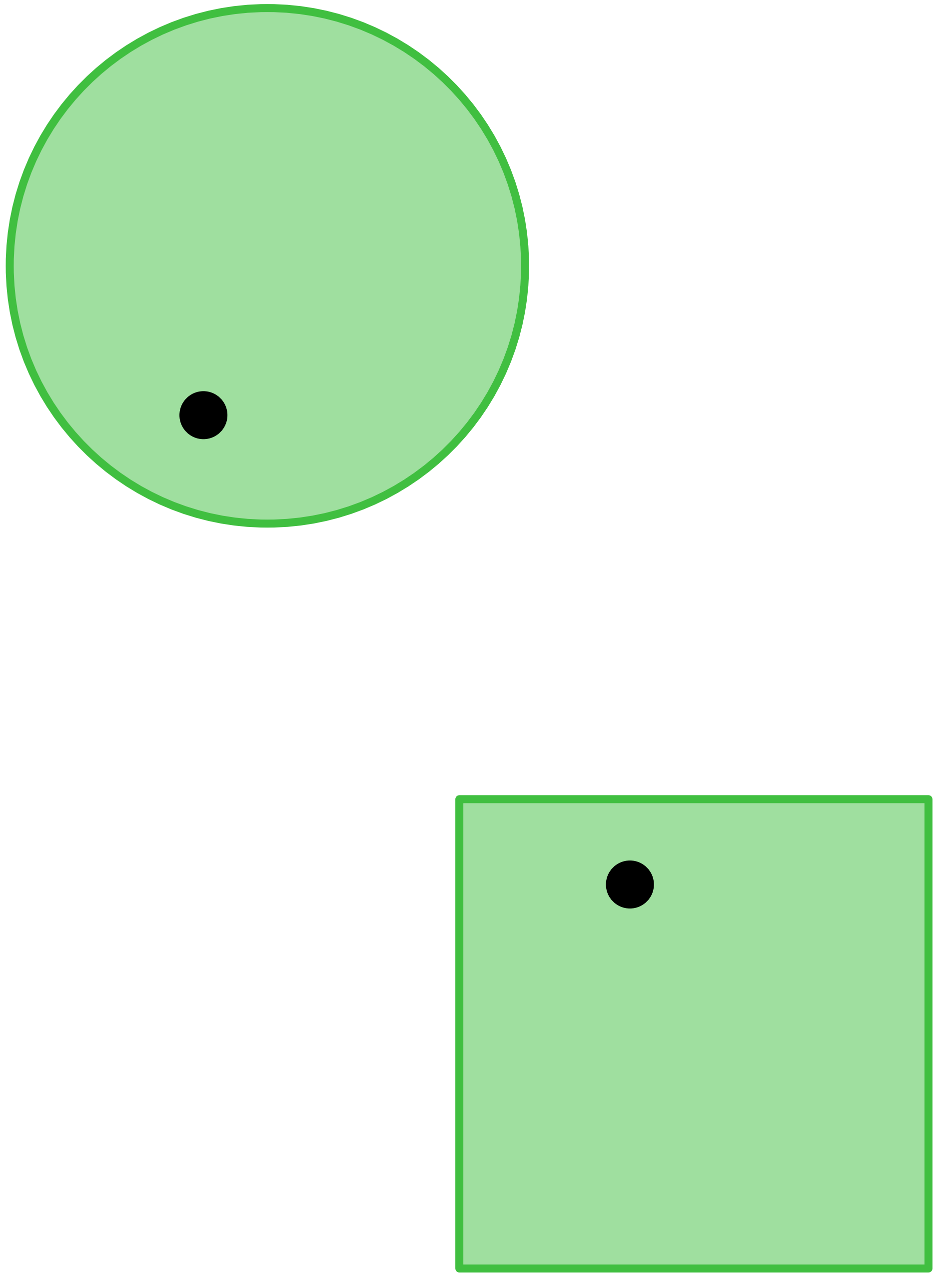
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Imprecise Points



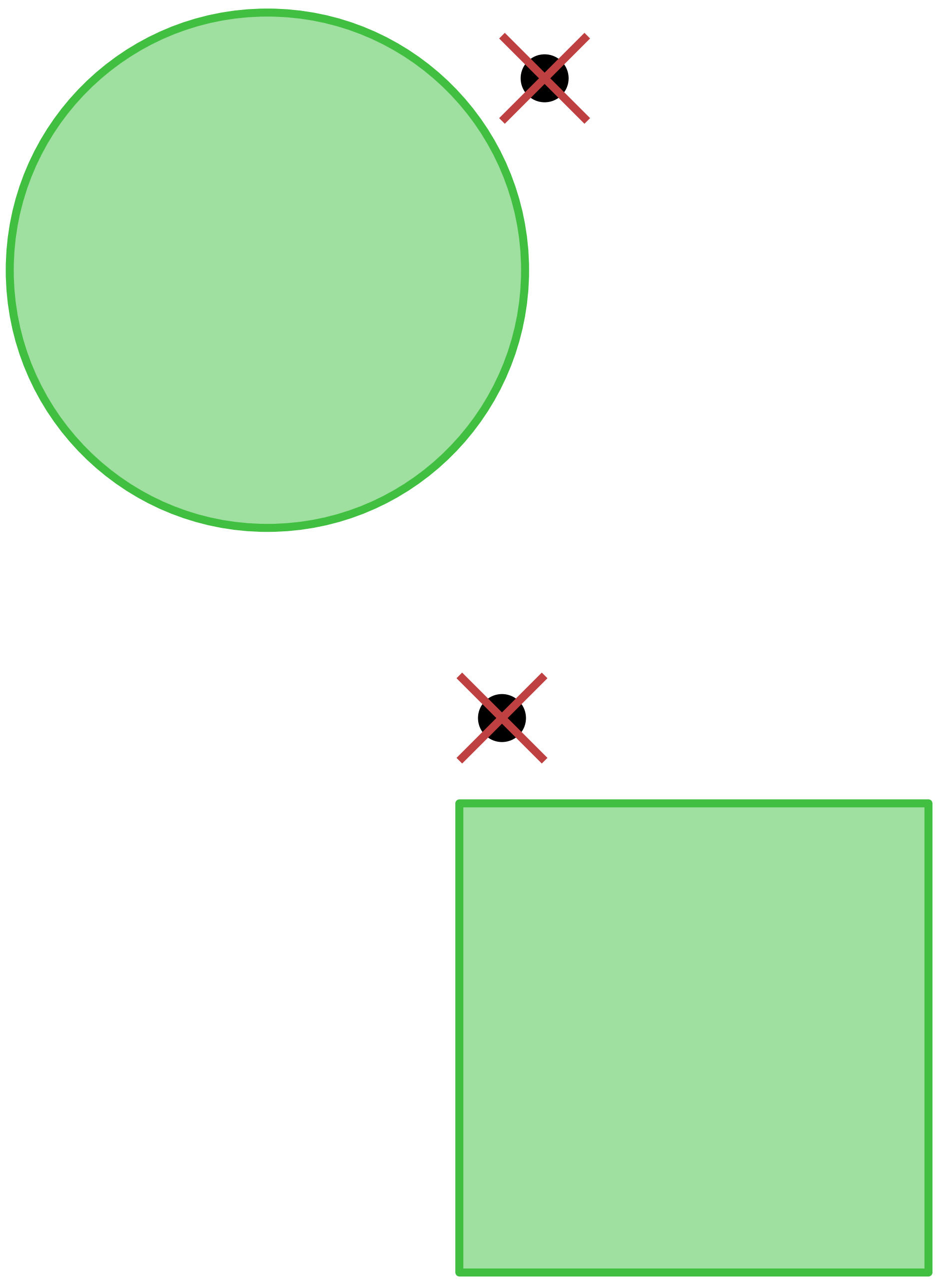
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Imprecise Points



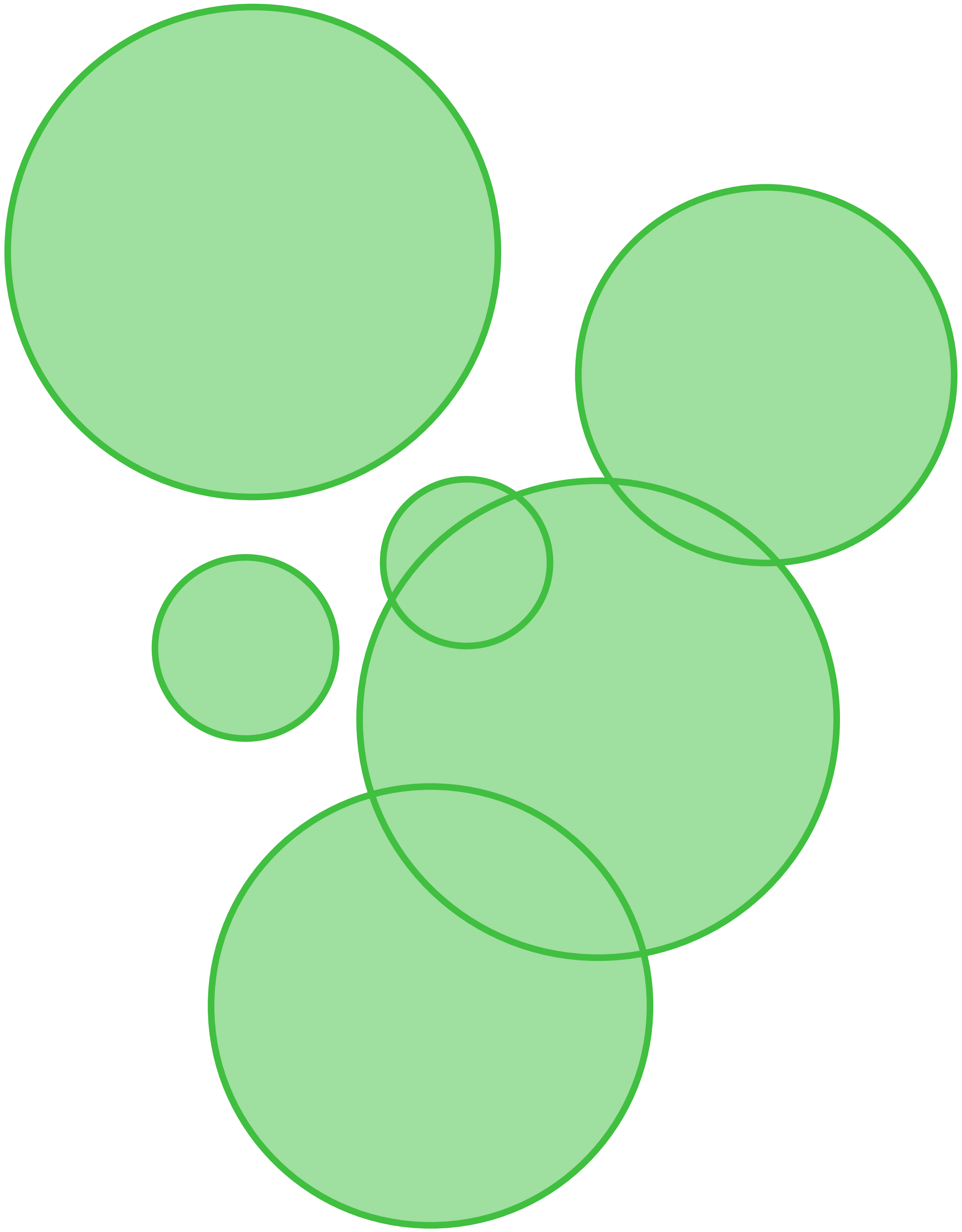
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Imprecise Points



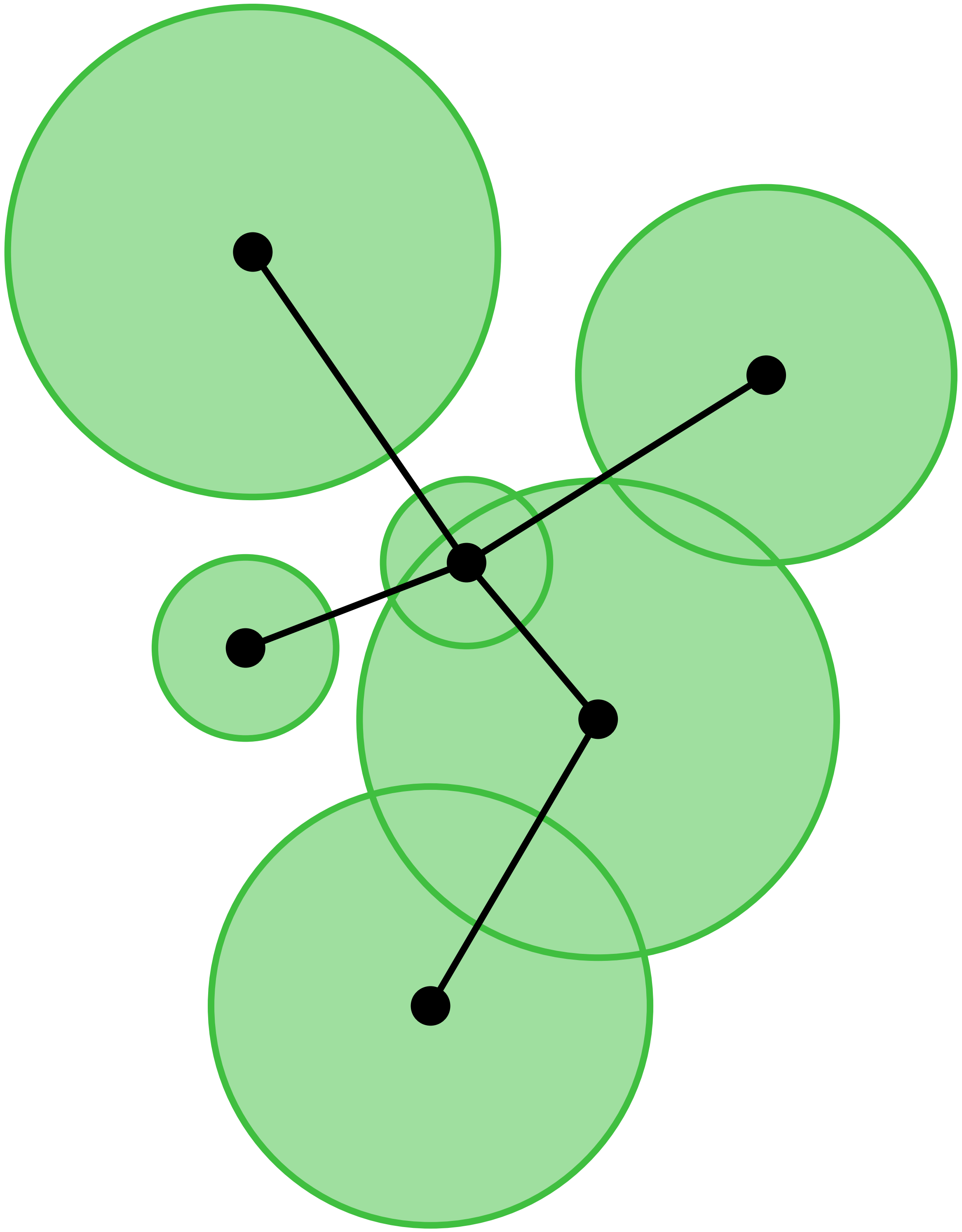
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Imprecision in Geometric Structures



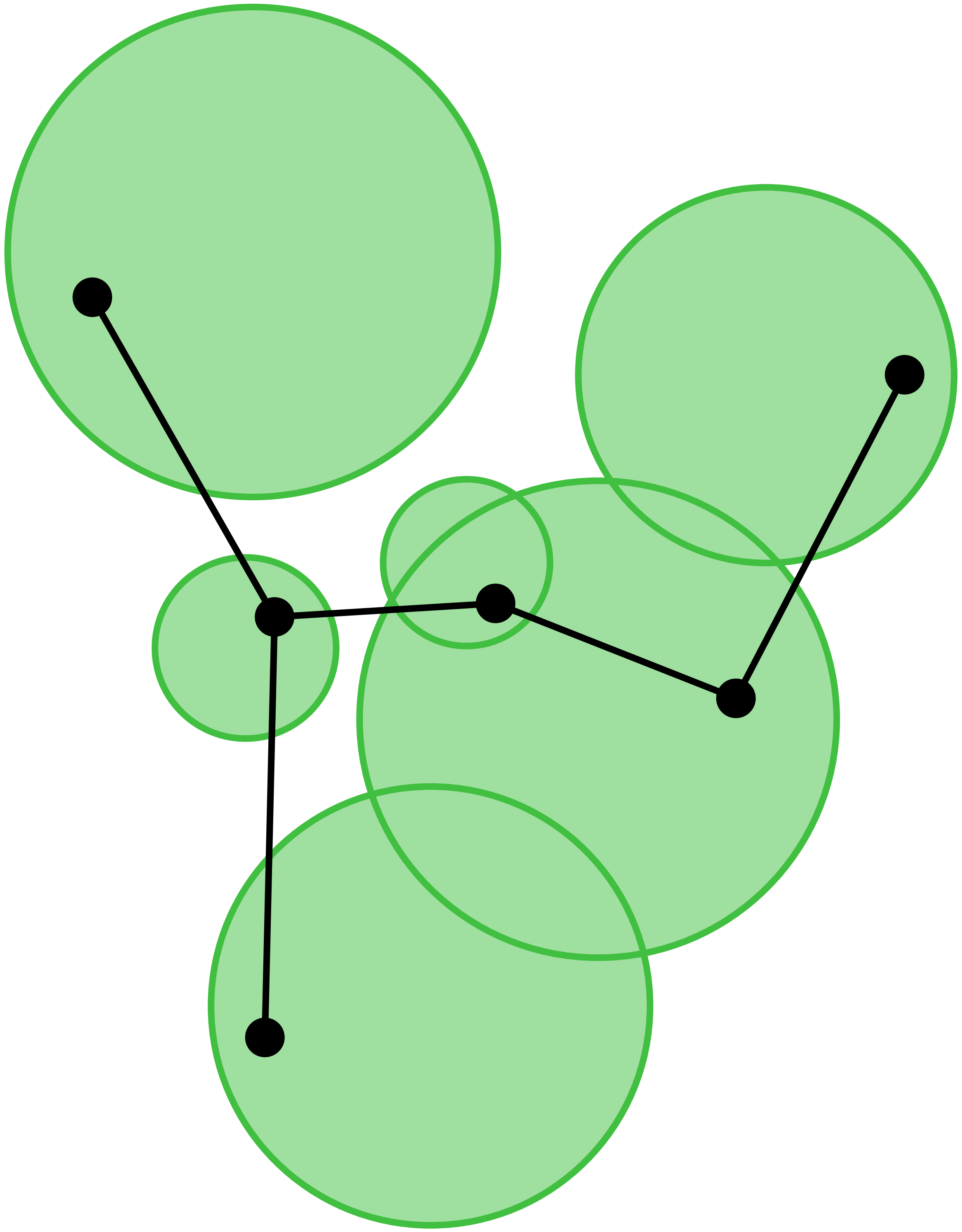
- Given a set $\mathcal{L}$ of n imprecise points
- Consider the same geometric structures
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Imprecision in Geometric Structures



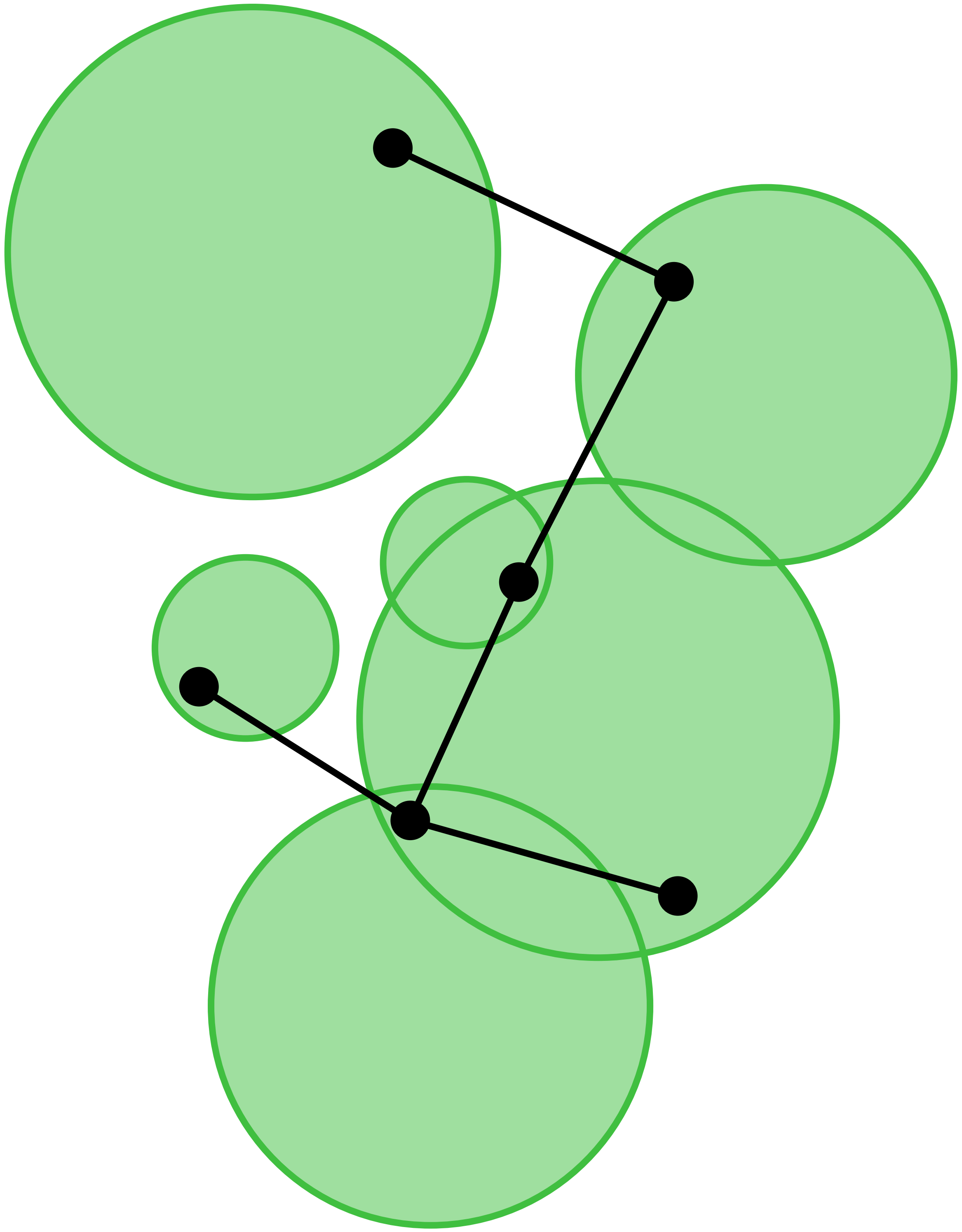
- Given a set $\mathcal{L}$ of n imprecise points
- Consider the same geometric structures
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Imprecision in Geometric Structures



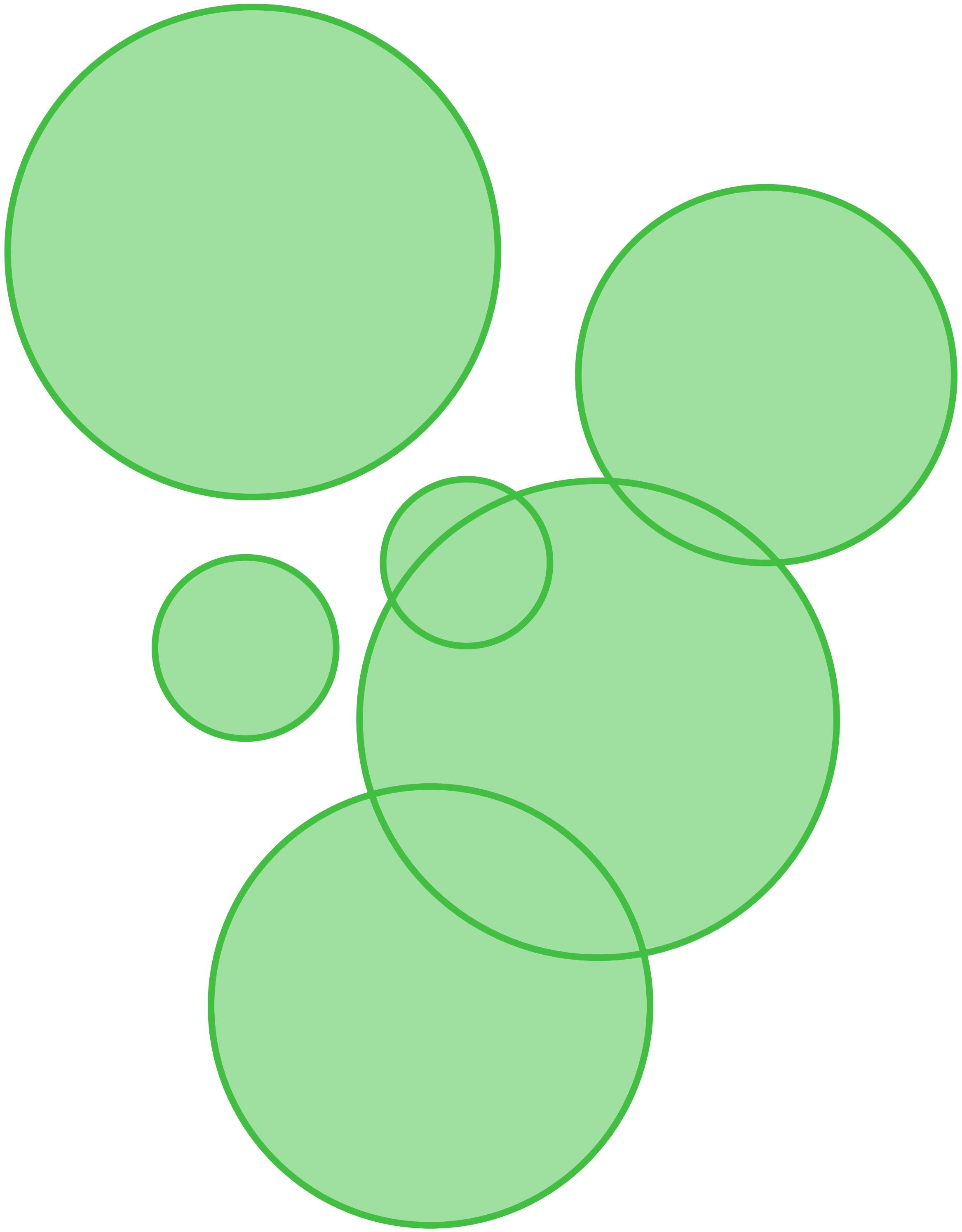
- Given a set $\mathcal{L}$ of n imprecise points
- Consider the same geometric structures
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Imprecision in Geometric Structures



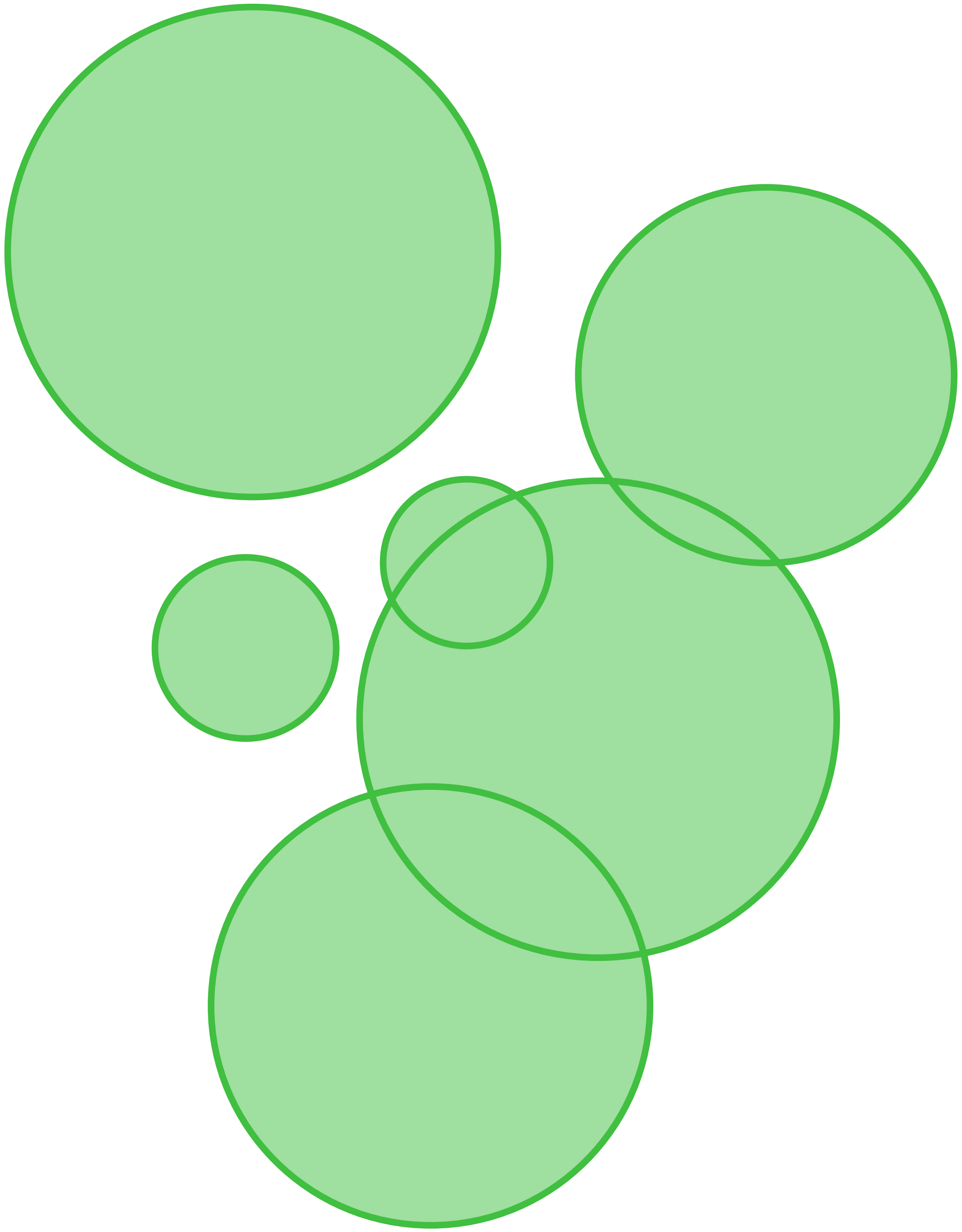
- Given a set $\mathcal{L}$ of n imprecise points
- Consider the same geometric structures
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Bounds on Measures



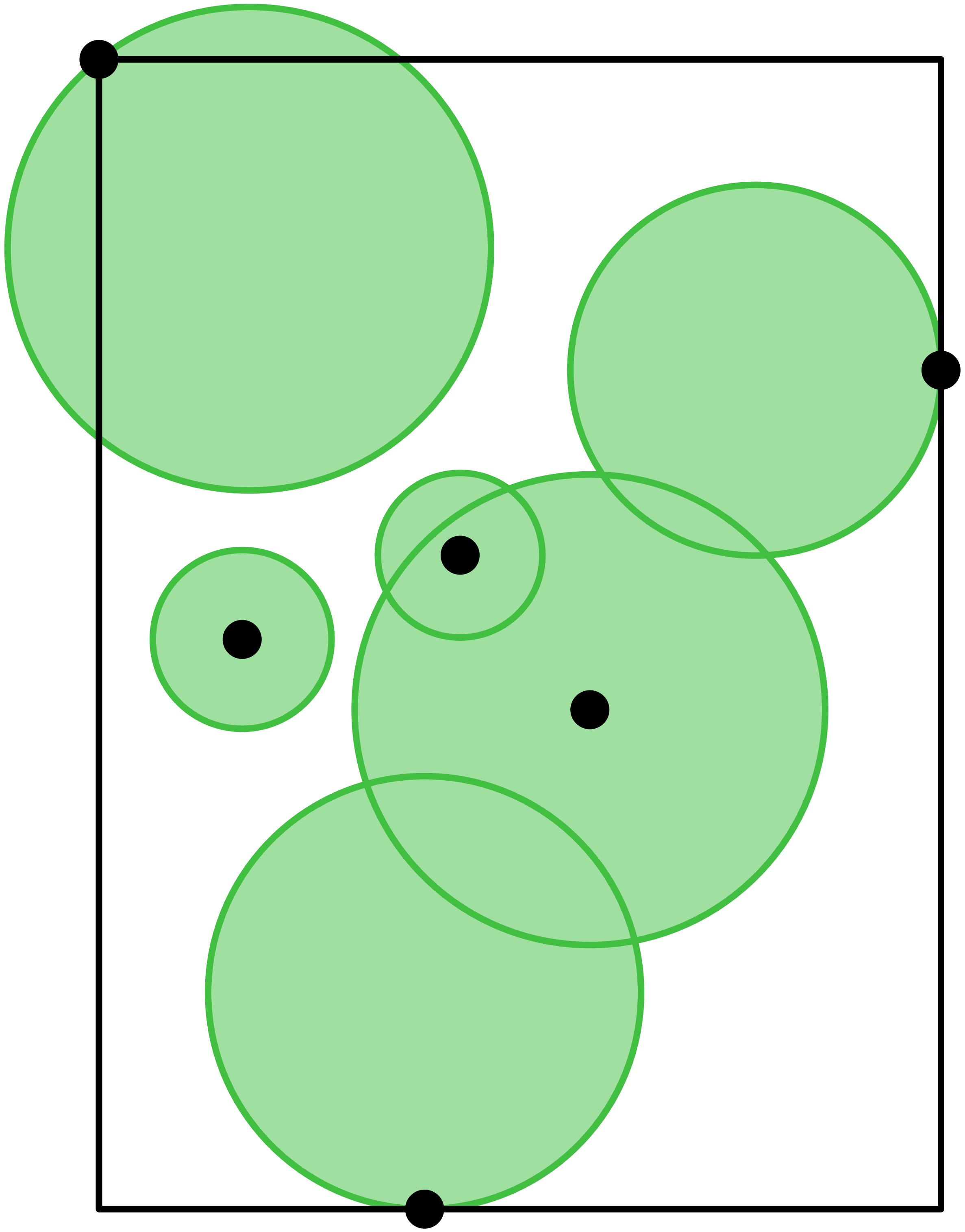
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \rightarrow \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision

Bounds on Measures



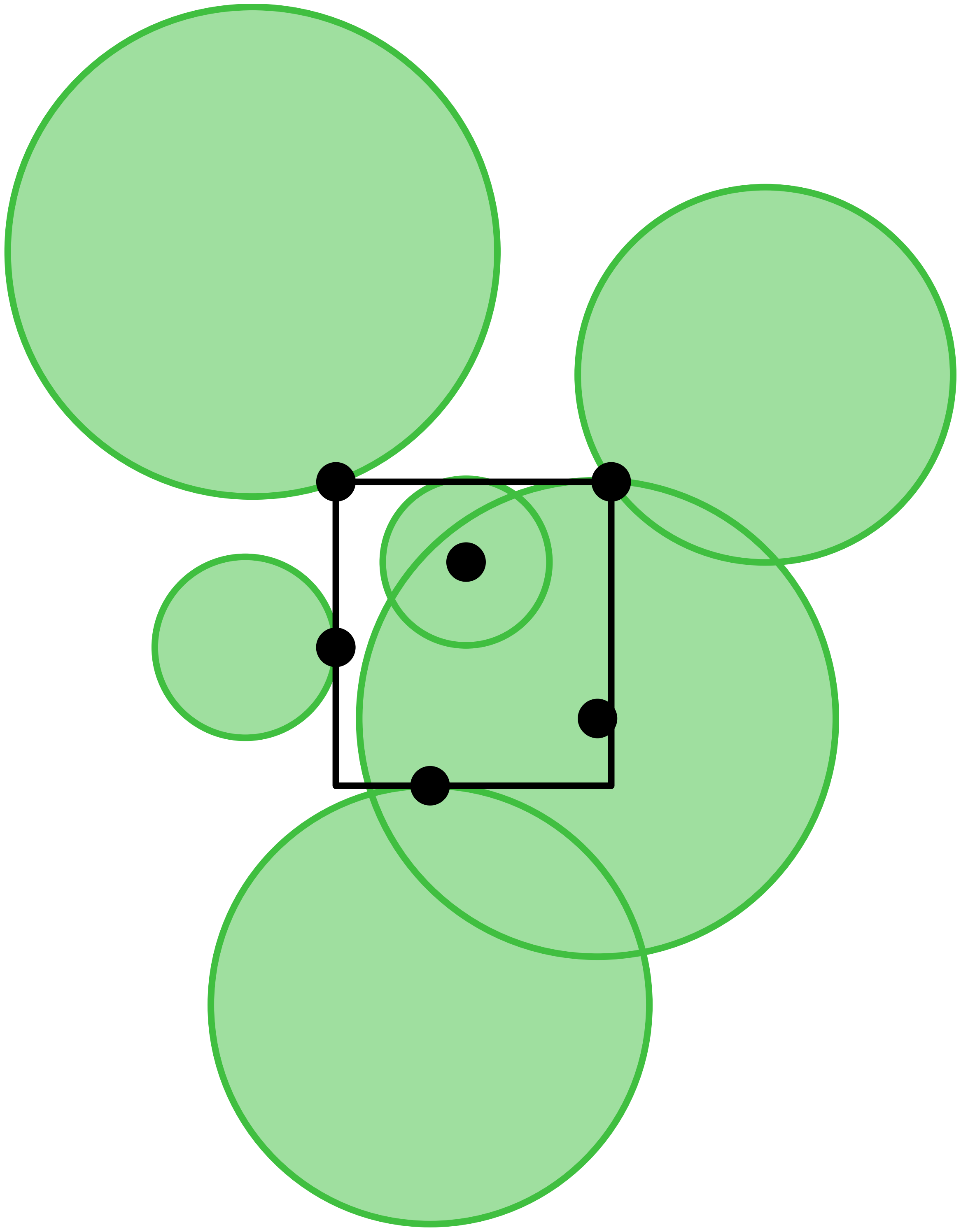
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \rightarrow \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area

Bounds on Measures



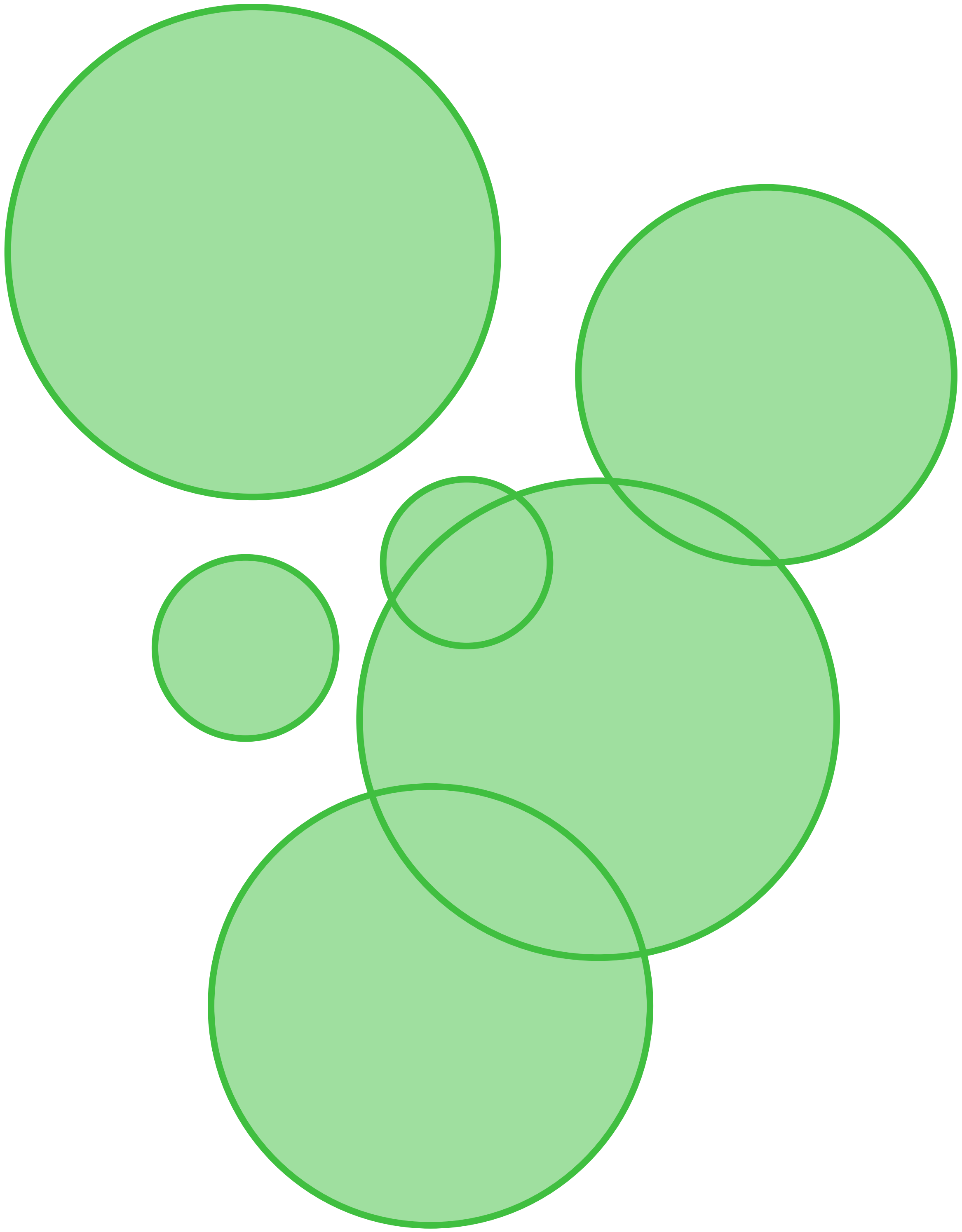
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \to \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area

Bounds on Measures



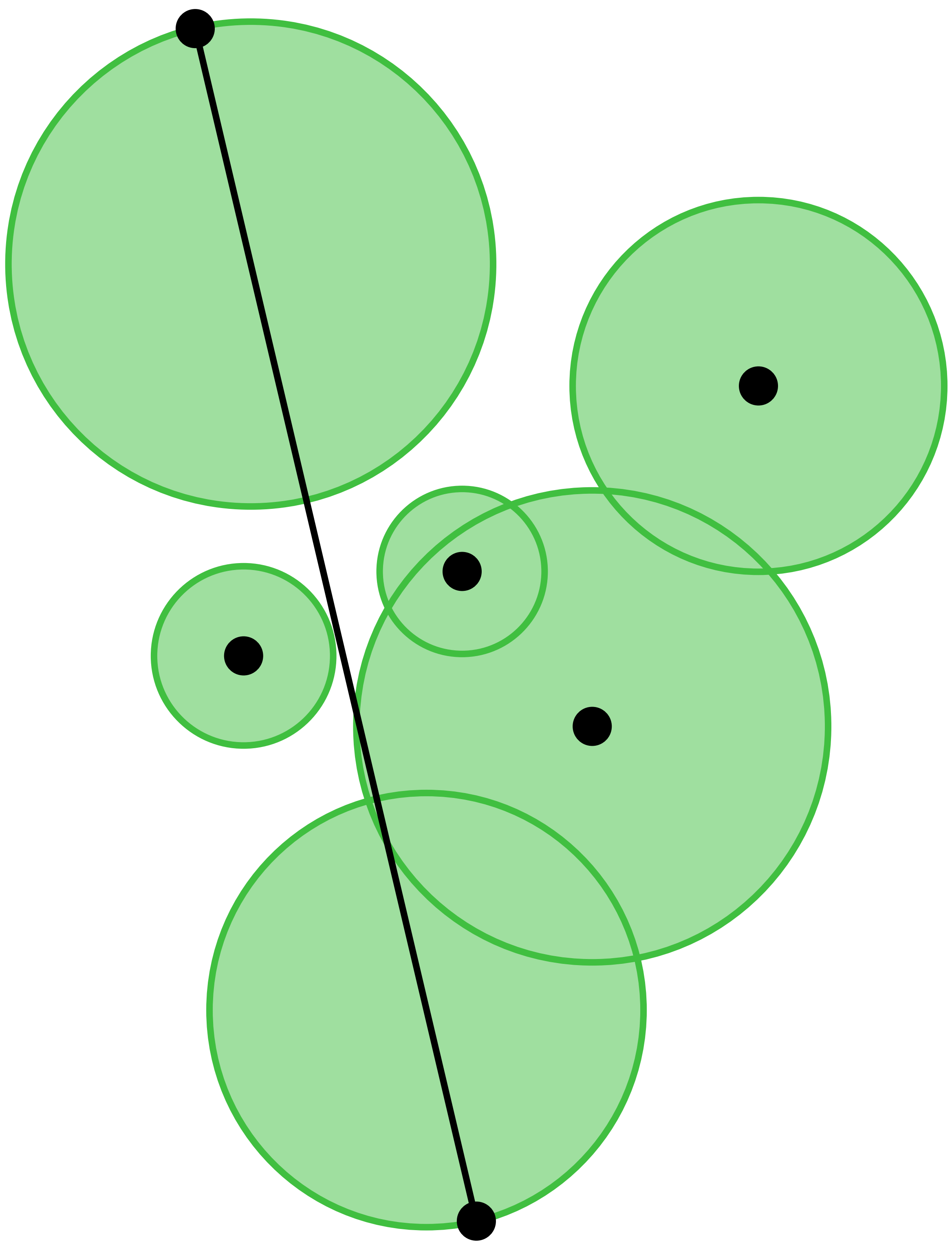
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \to \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area

Bounds on Measures



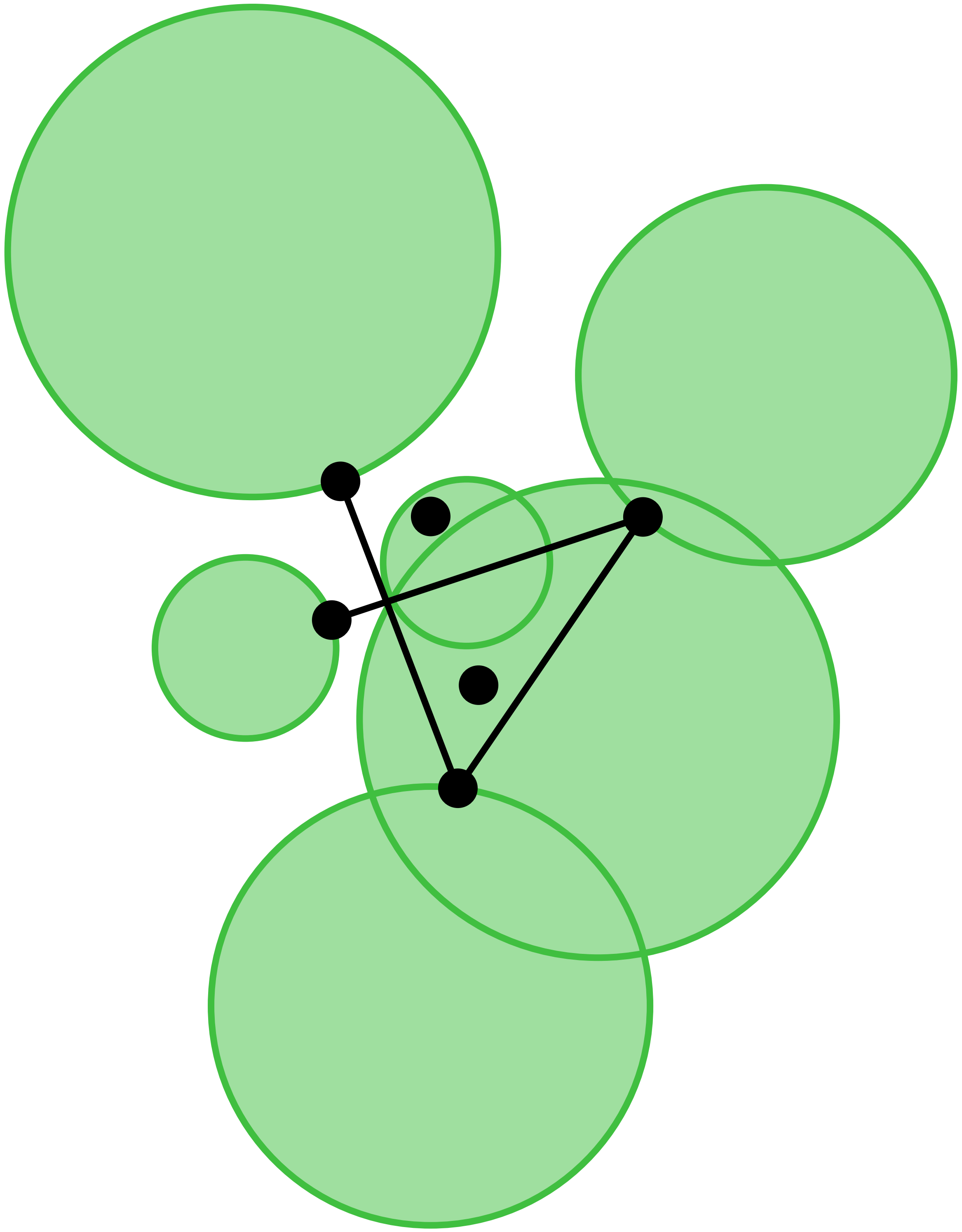
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \rightarrow \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area
  - Diameter

Bounds on Measures



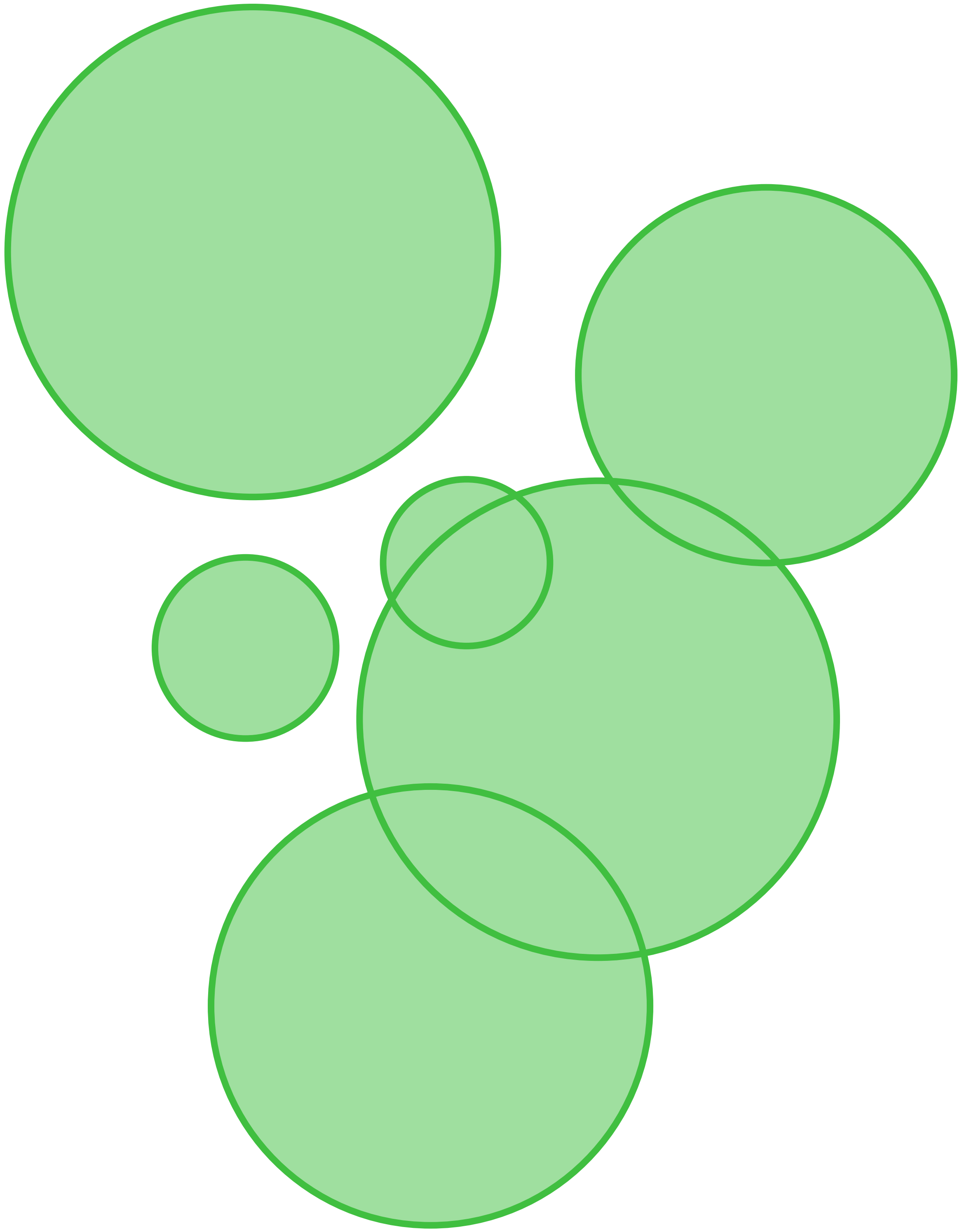
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \rightarrow \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area
  - Diameter

Bounds on Measures



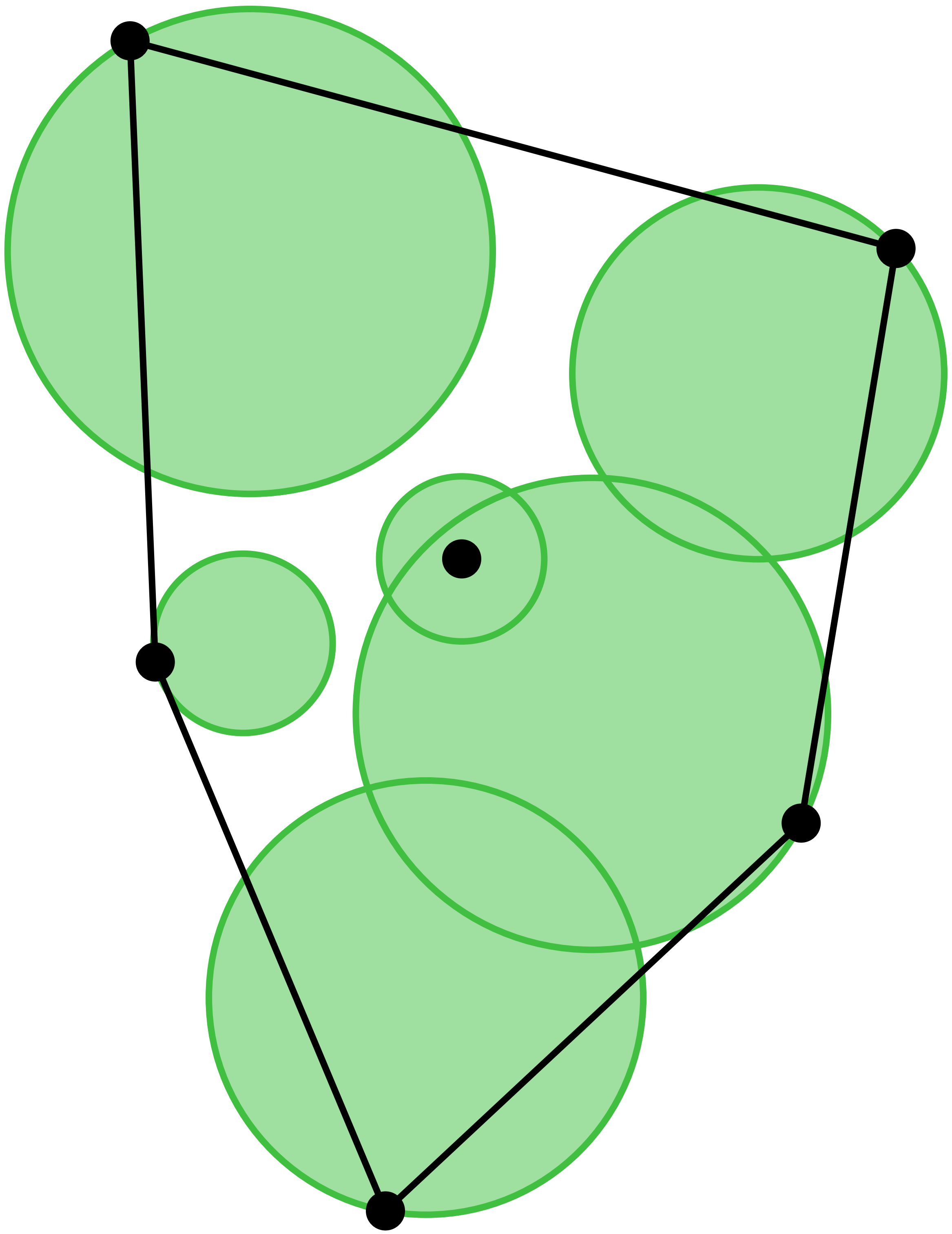
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \rightarrow \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area
  - Diameter

Bounds on Measures



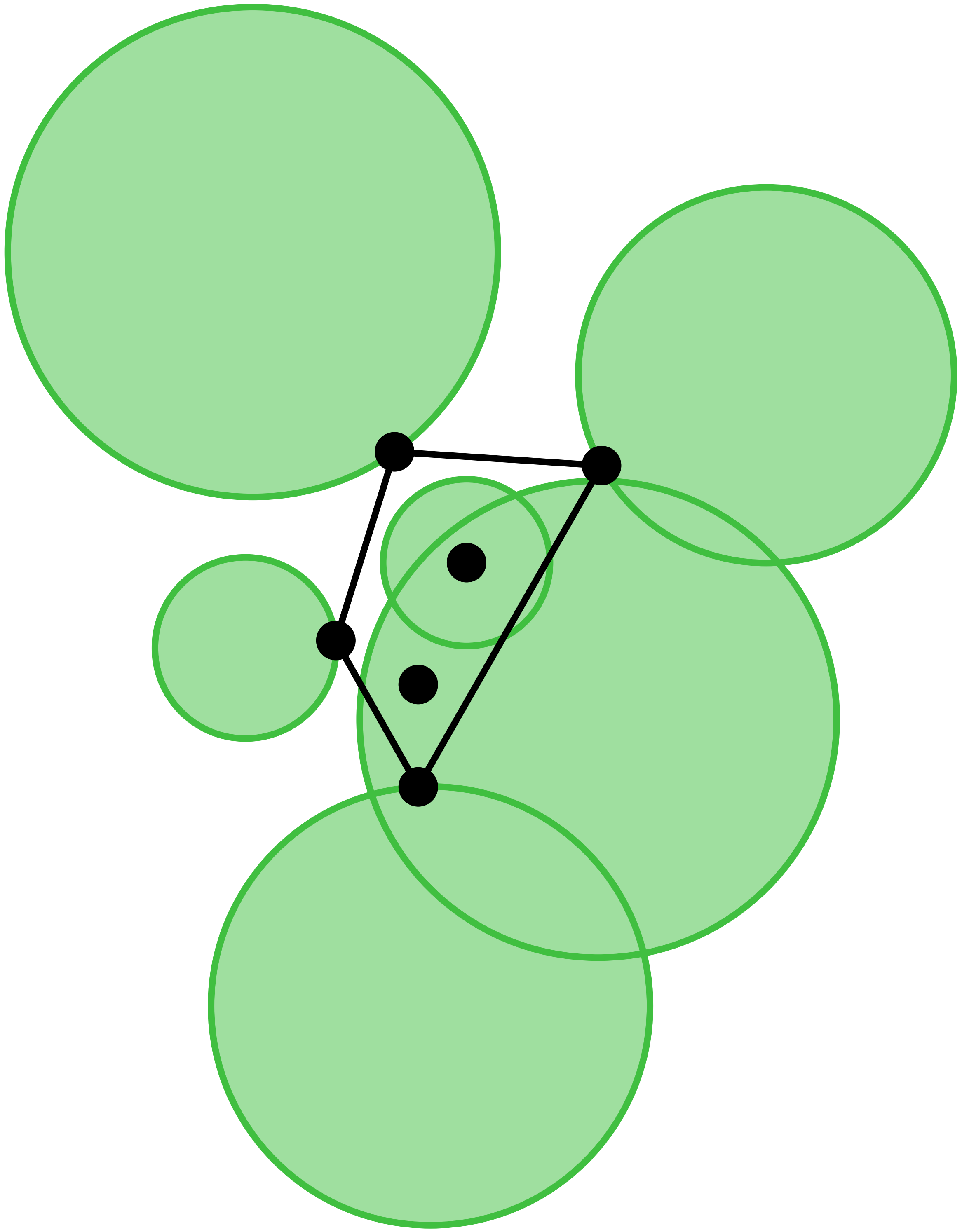
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \to \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area
  - Diameter
  - Convex hull perimeter

Bounds on Measures



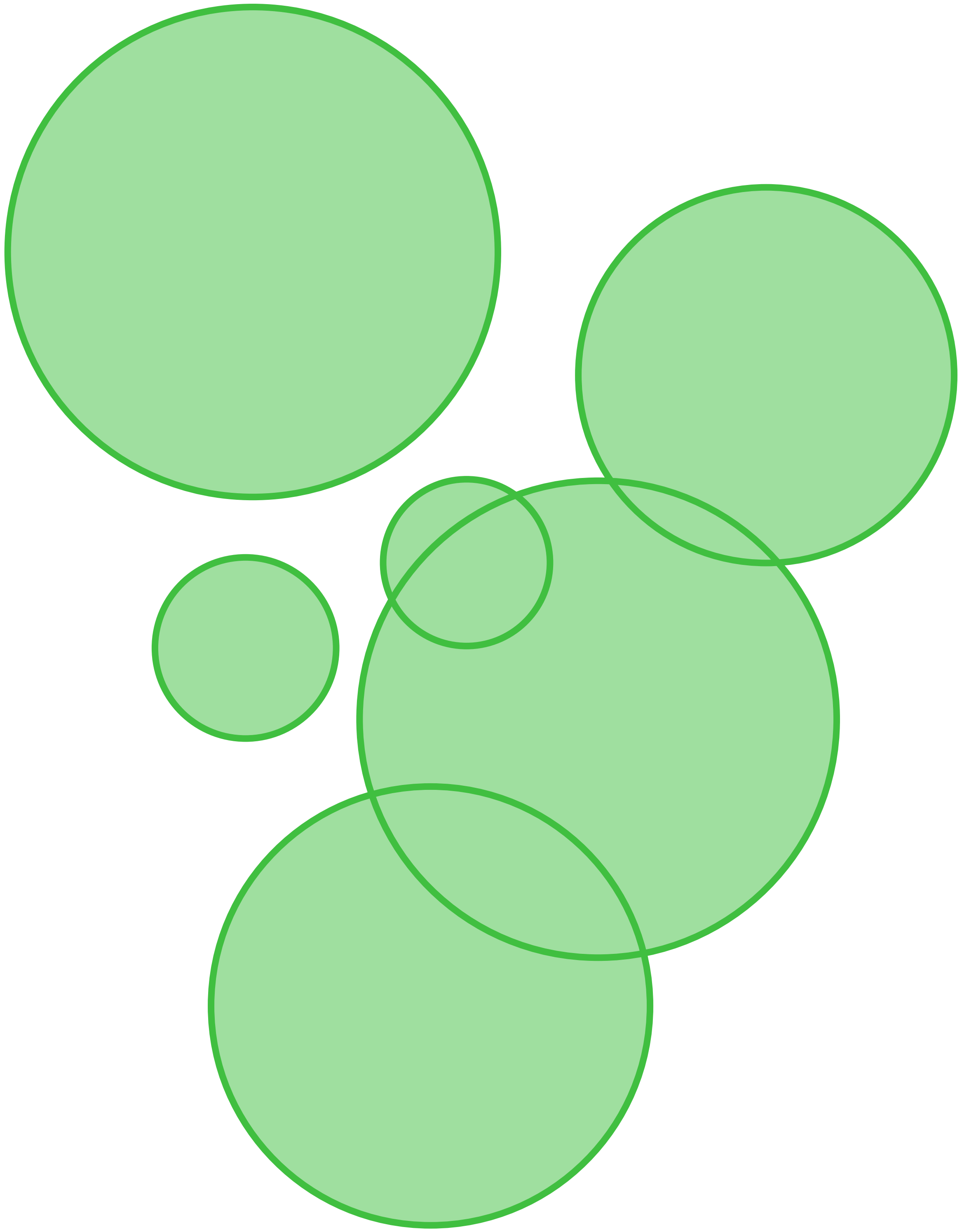
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \rightarrow \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area
  - Diameter
  - Convex hull perimeter

Bounds on Measures



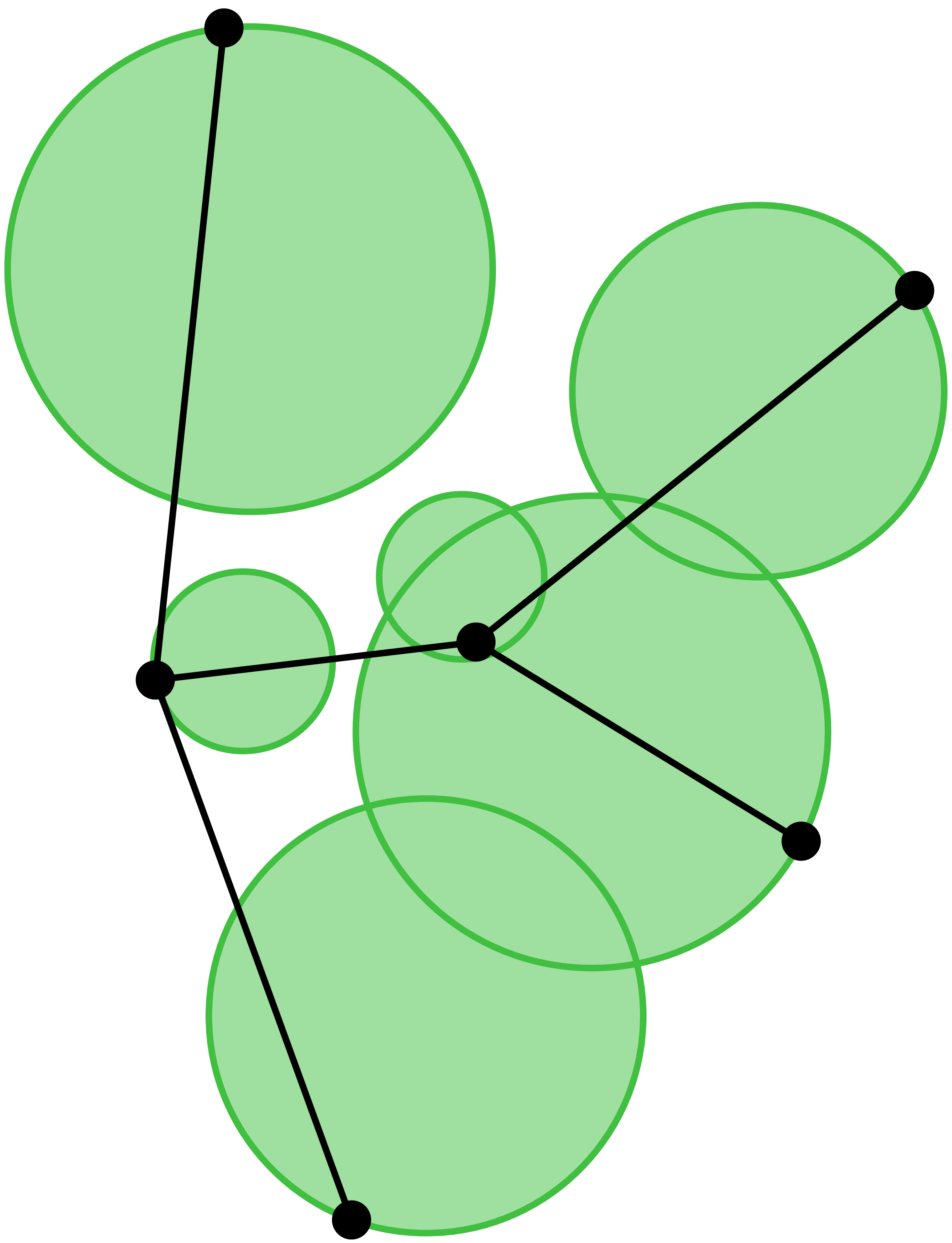
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \rightarrow \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area
  - Diameter
  - Convex hull perimeter

Bounds on Measures



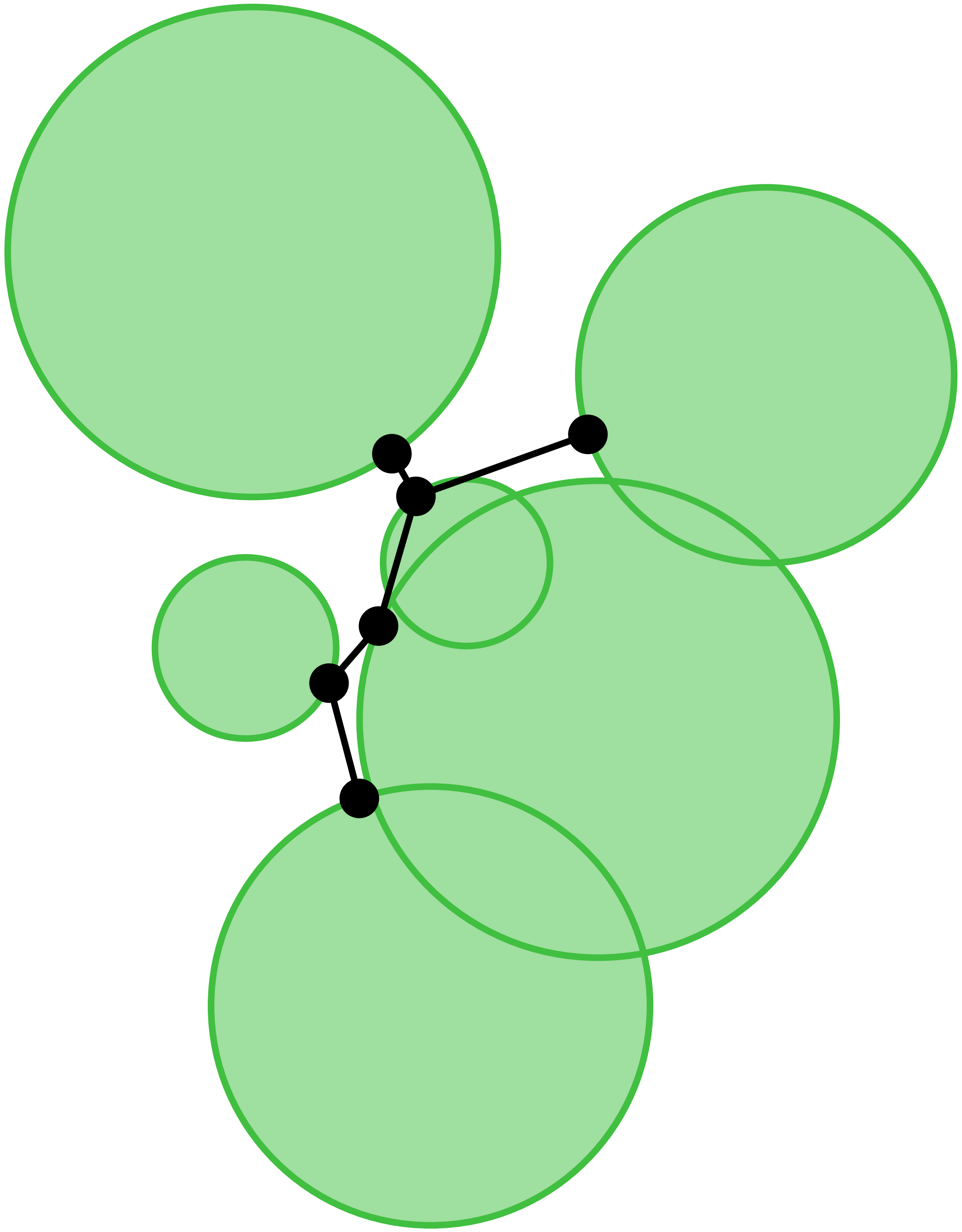
- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \to \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area
  - Diameter
  - Convex hull perimeter
  - MST weight

Bounds on Measures



- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \rightarrow \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area
  - Diameter
  - Convex hull perimeter
  - MST weight

Bounds on Measures



- Measure function $\mu : \mathcal{F}(\mathbb{R}^2) \rightarrow \mathbb{R}$
- Largest and smallest possible values of μ
- Output imprecision
- For example
  - Bounding box area
  - Diameter
  - Convex hull perimeter
  - MST weight

Results

- Bounding box
  - Largest area or perimeter, squares or discs $O(n)$
  - Smallest area or perimeter, squares $O(n)$
  - Smallest area or perimeter, discs $O(n^2)$
- Smallest enclosing circle
  - Largest or smallest radius, squares or discs $O(n)$
- Convex hull
  - Largest area, disjoint squares $O(n^7)$
  - Smallest area, squares $O(n^2)$
  - Largest perimeter, disjoint squares $O(n^{10})$
  - Smallest perimeter, squares $O(n \log n)$

Results

- Diameter
  - Largest diameter, squares or discs $O(n \log n)$
  - Smallest diameter, squares $O(n \log n)$
  - Smallest diameter, discs $O(n^{c\varepsilon^{-1}})$
- Width
  - Smallest width, squares or discs $O(n \log n)$
  - Largest width, line segments NP-hard
  - Largest width, squares or discs ?
- Closest pair
  - Smallest distance, squares or discs $O(n \log n)$
  - Largest distance, squares or discs NP-hard

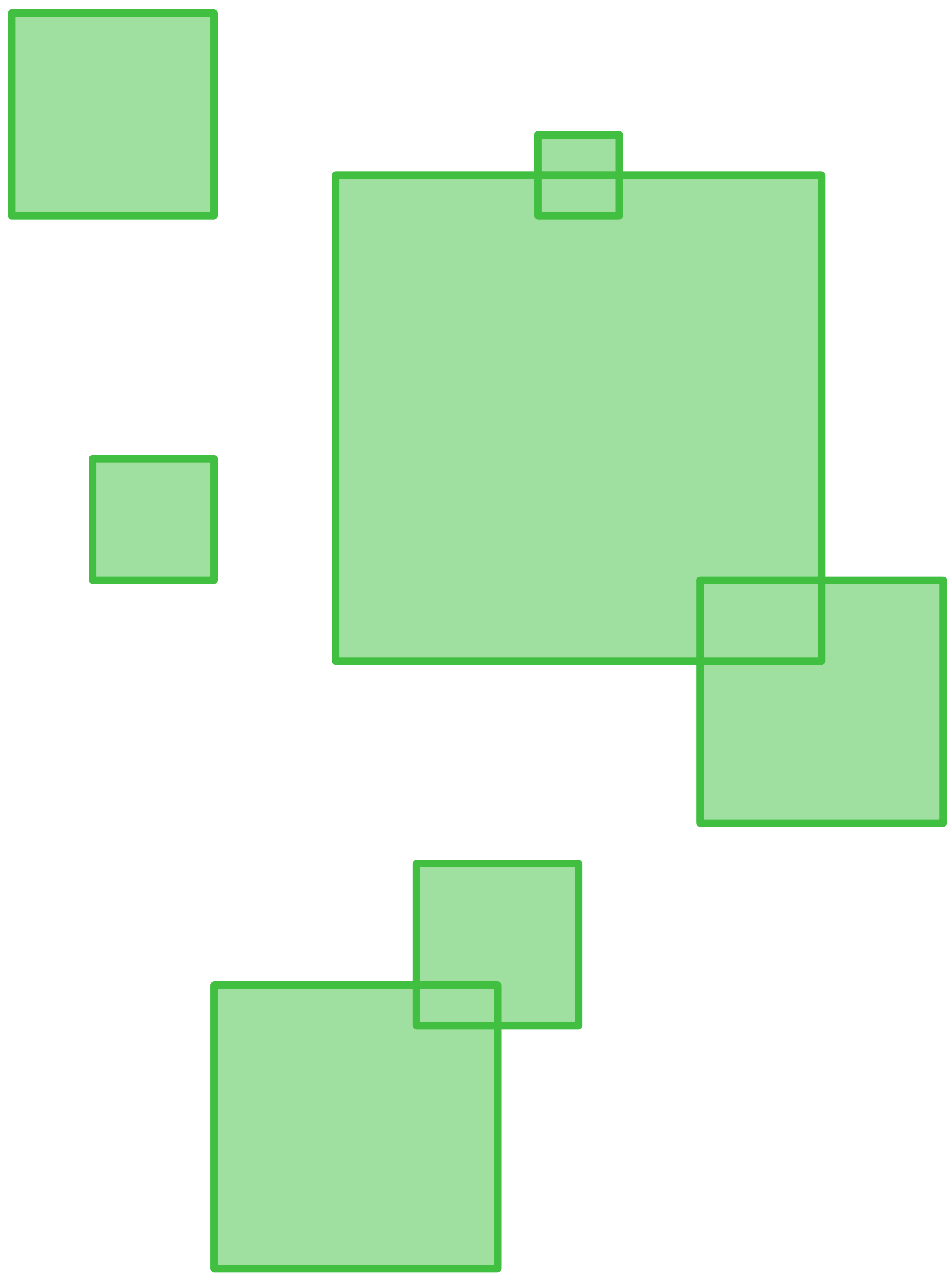
Results

- Minimum spanning tree
  - Smallest weight, squares or discs — NP-hard
  - Largest weight, squares or discs — ?
- Tours
  - Shortest tour, sequence of squares — $O(n)$
  - Longest tour, sequence of squares — $O(n)$
  - Existence of simple tour, any sequence — NP-hard

Diameter

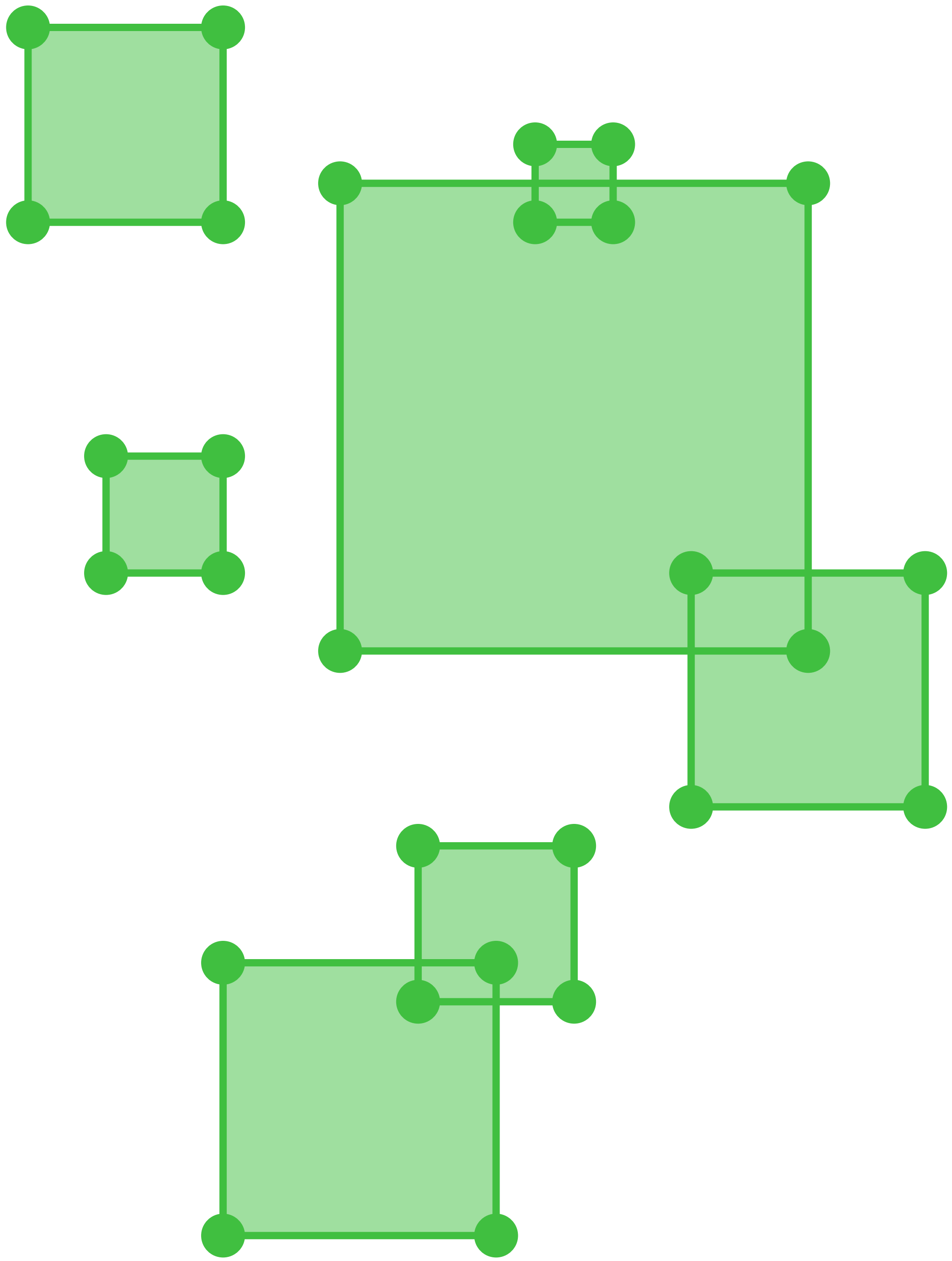
- Diameter of imprecise points, square model
- Largest diameter
  - Place two points as far away as possible
  - Relatively easy
  - Optimal $O(n \log n)$ algorithm
- Smallest diameter
  - Place all points as close together as possible
  - Much harder
  - $O(n^2)$ algorithm
  - Optimal $O(n \log n)$ algorithm

Largest Diameter



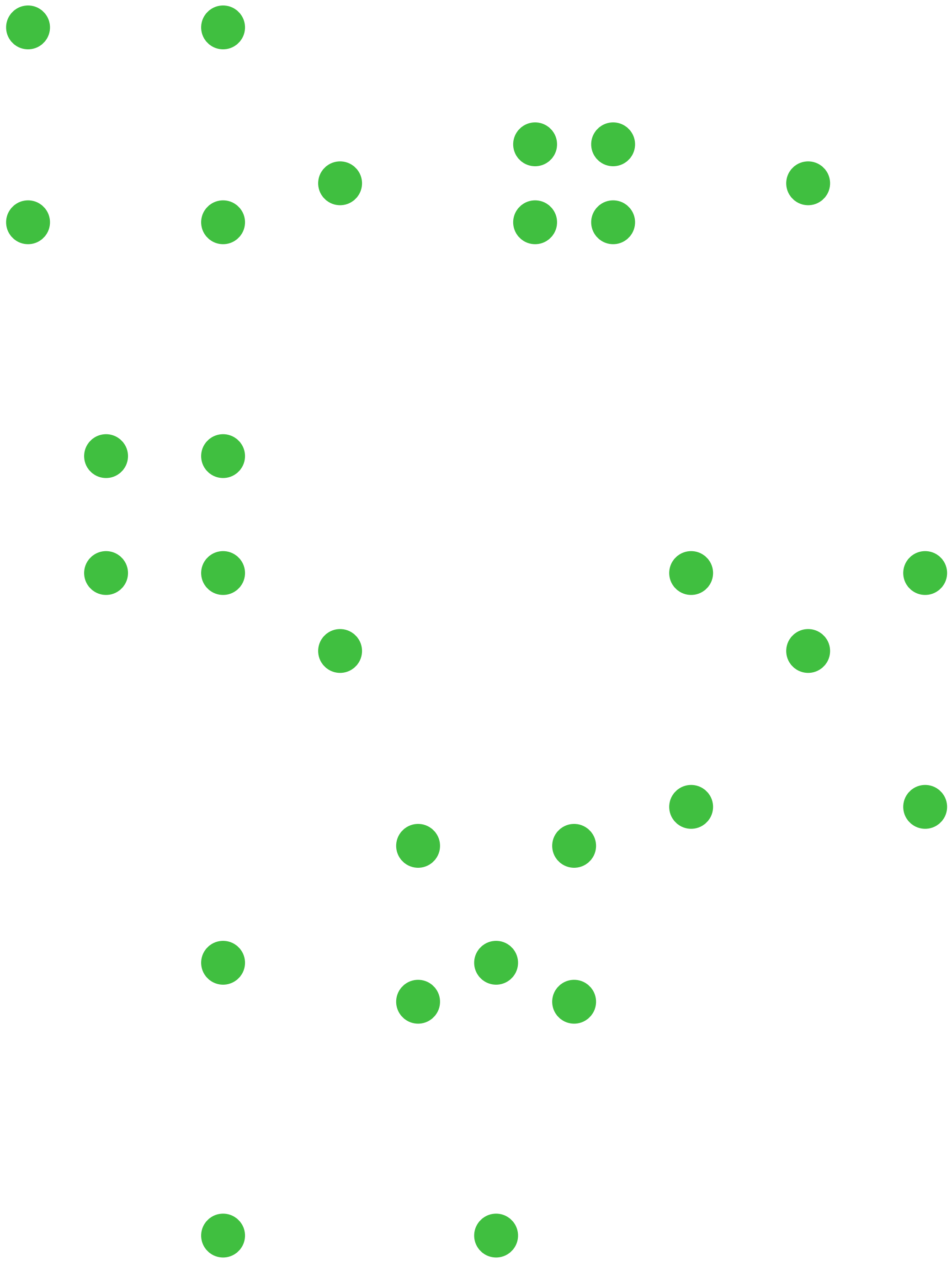
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



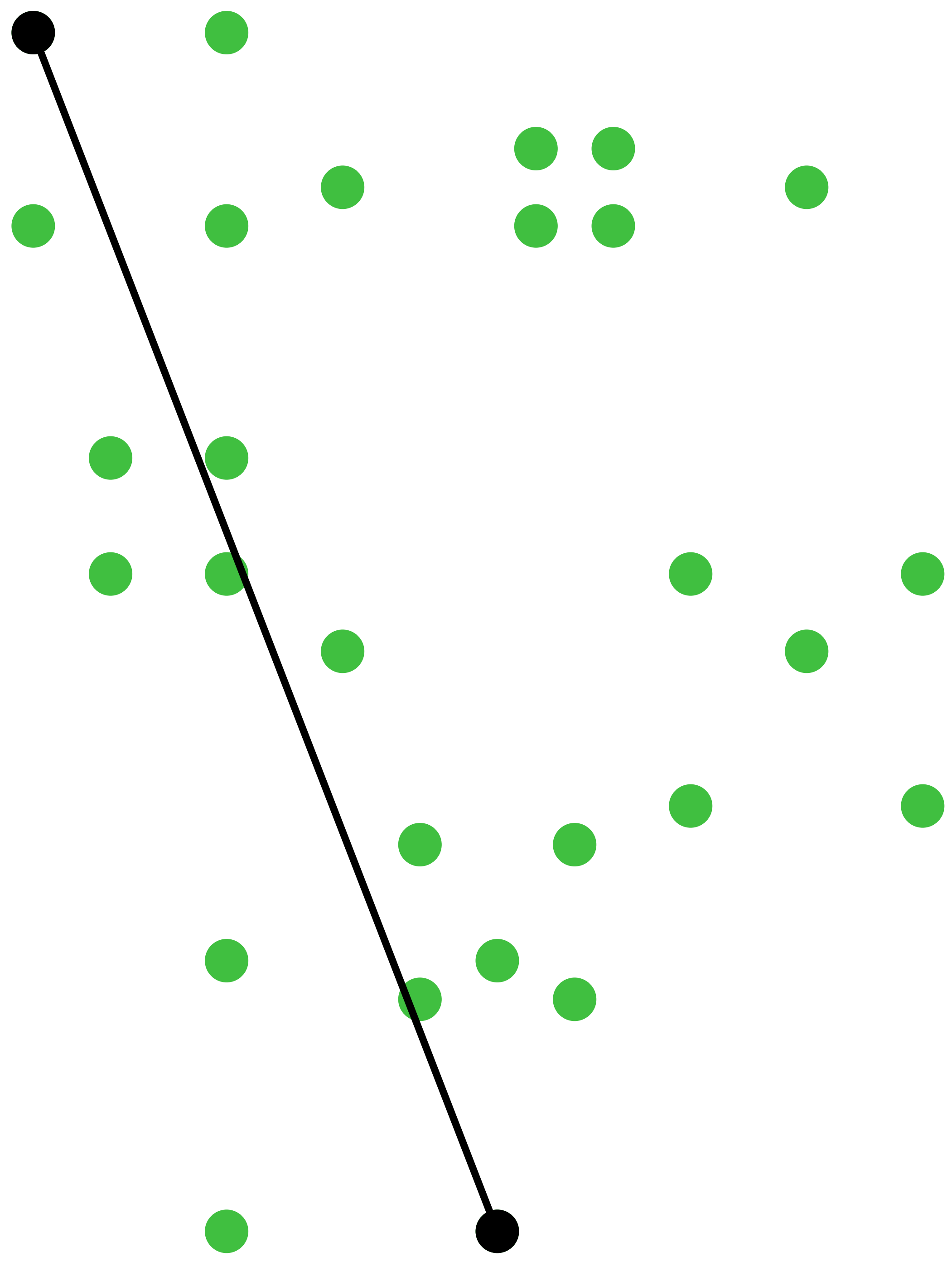
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



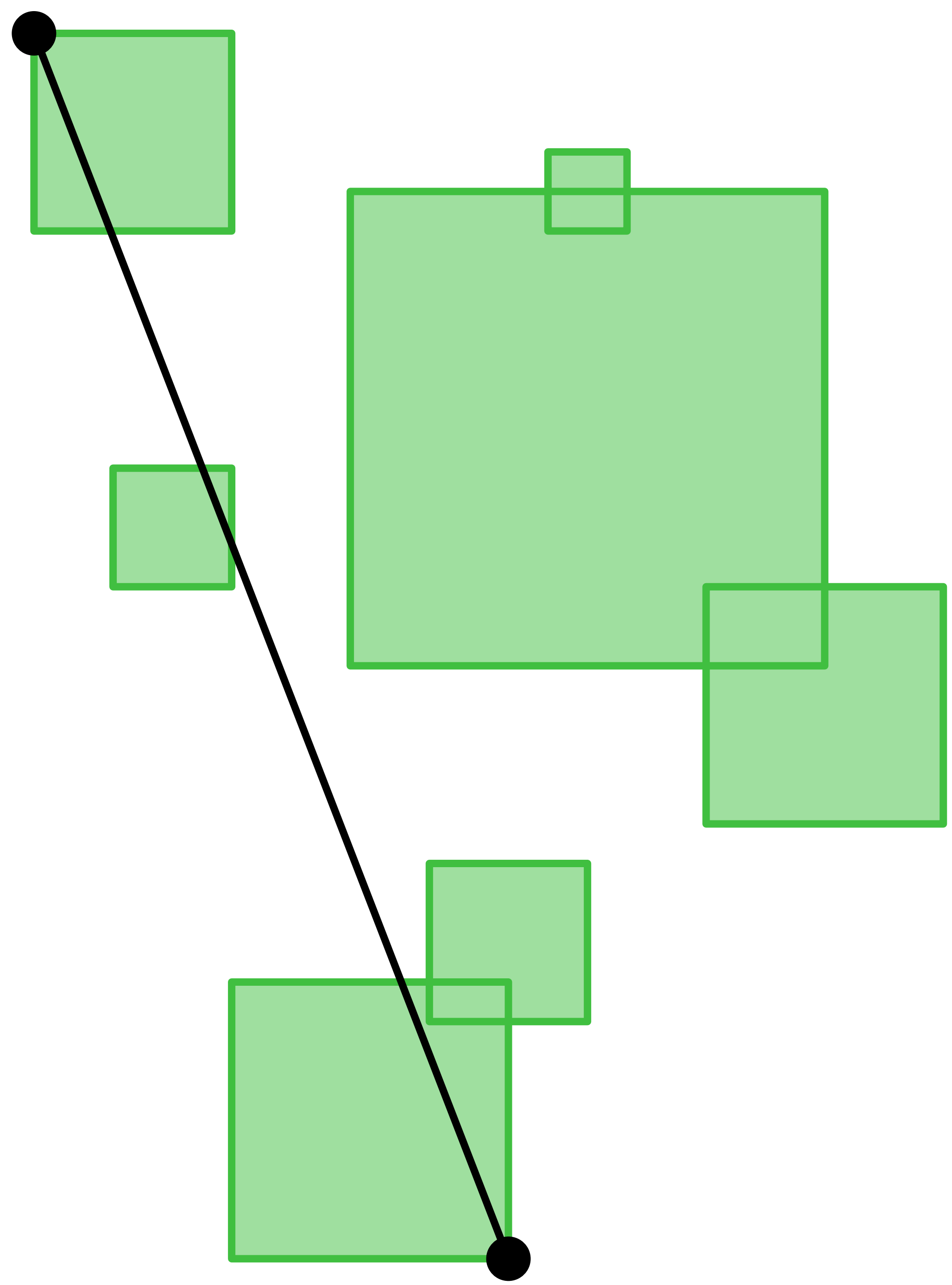
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



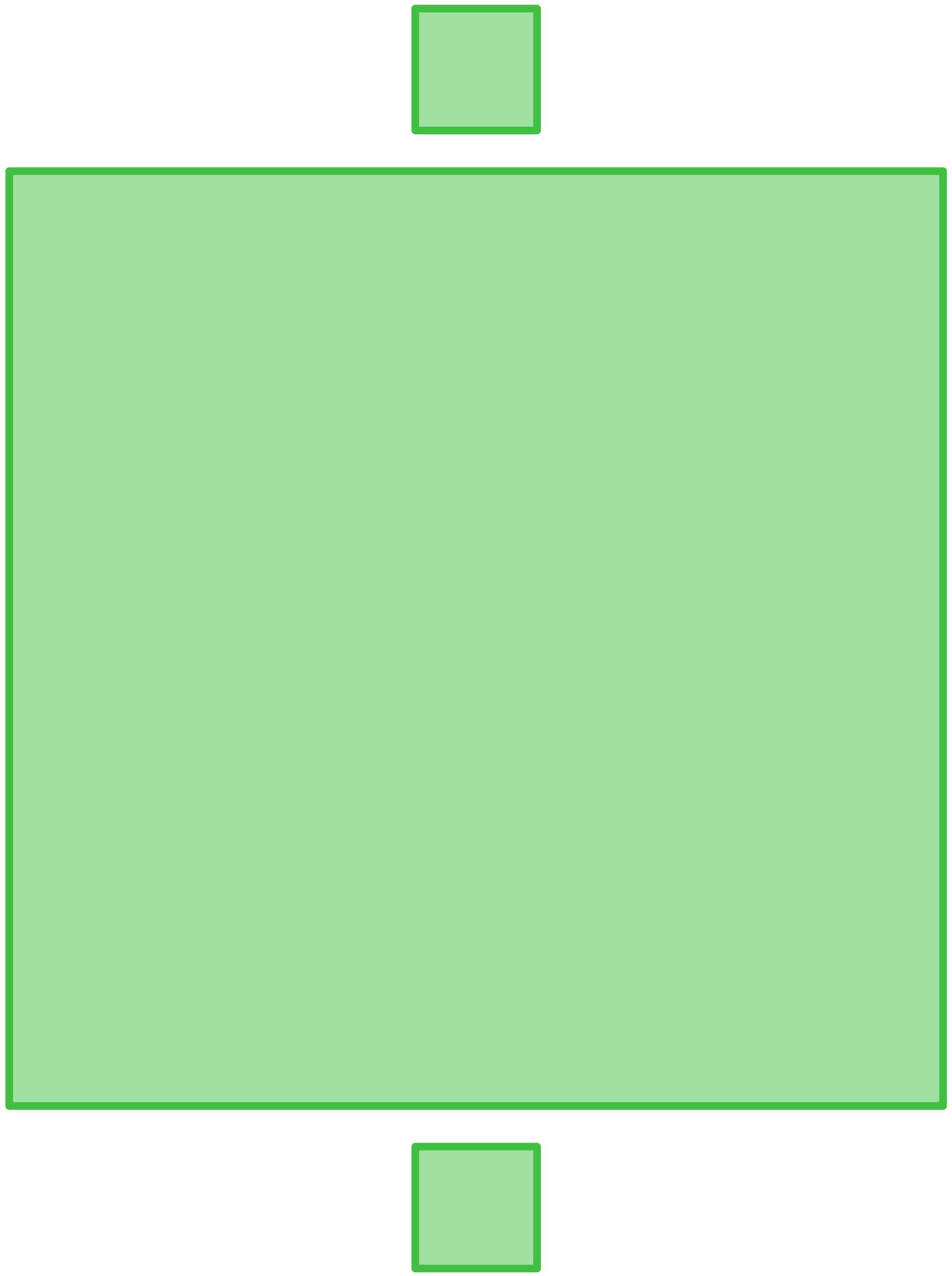
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



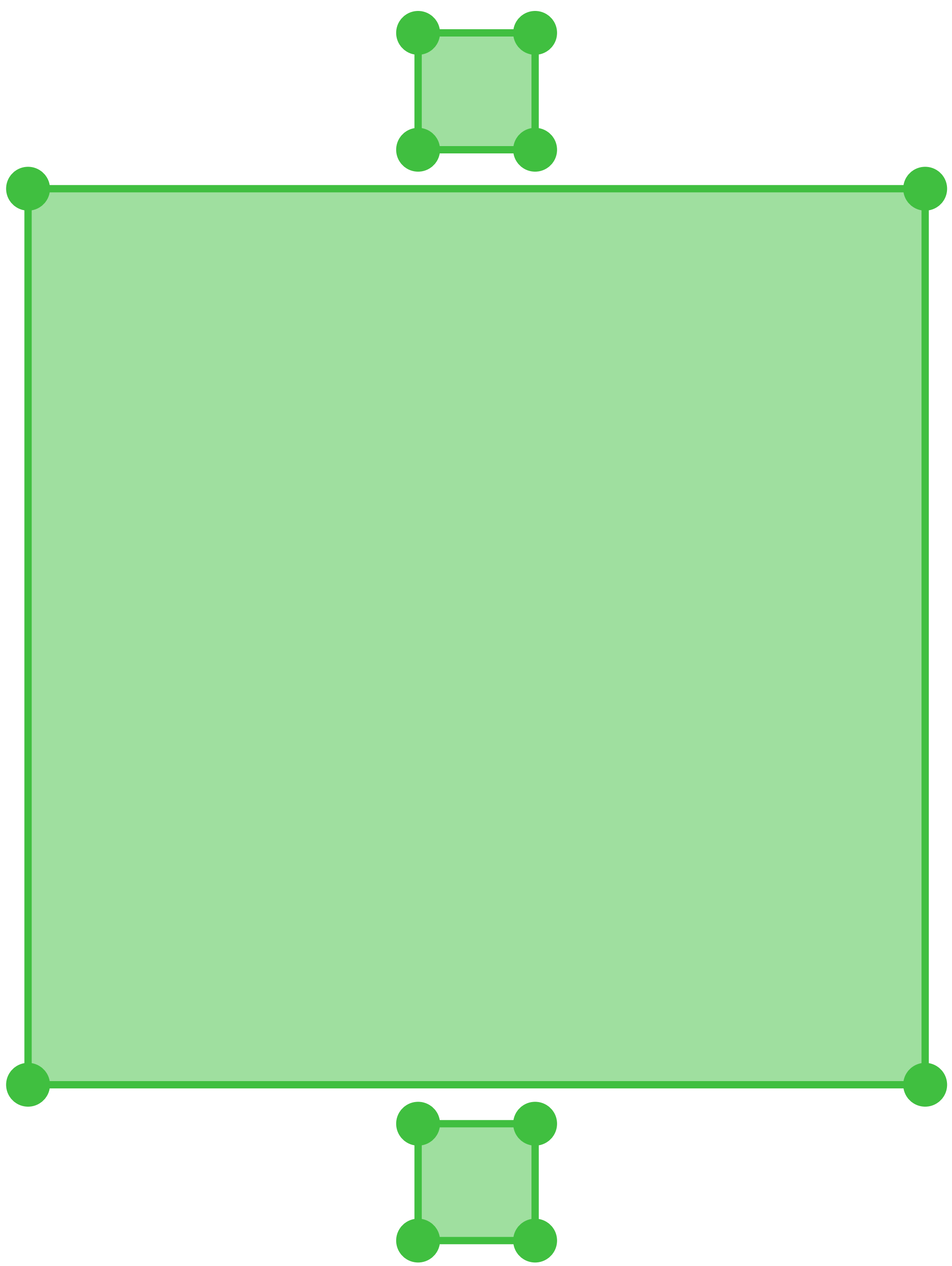
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



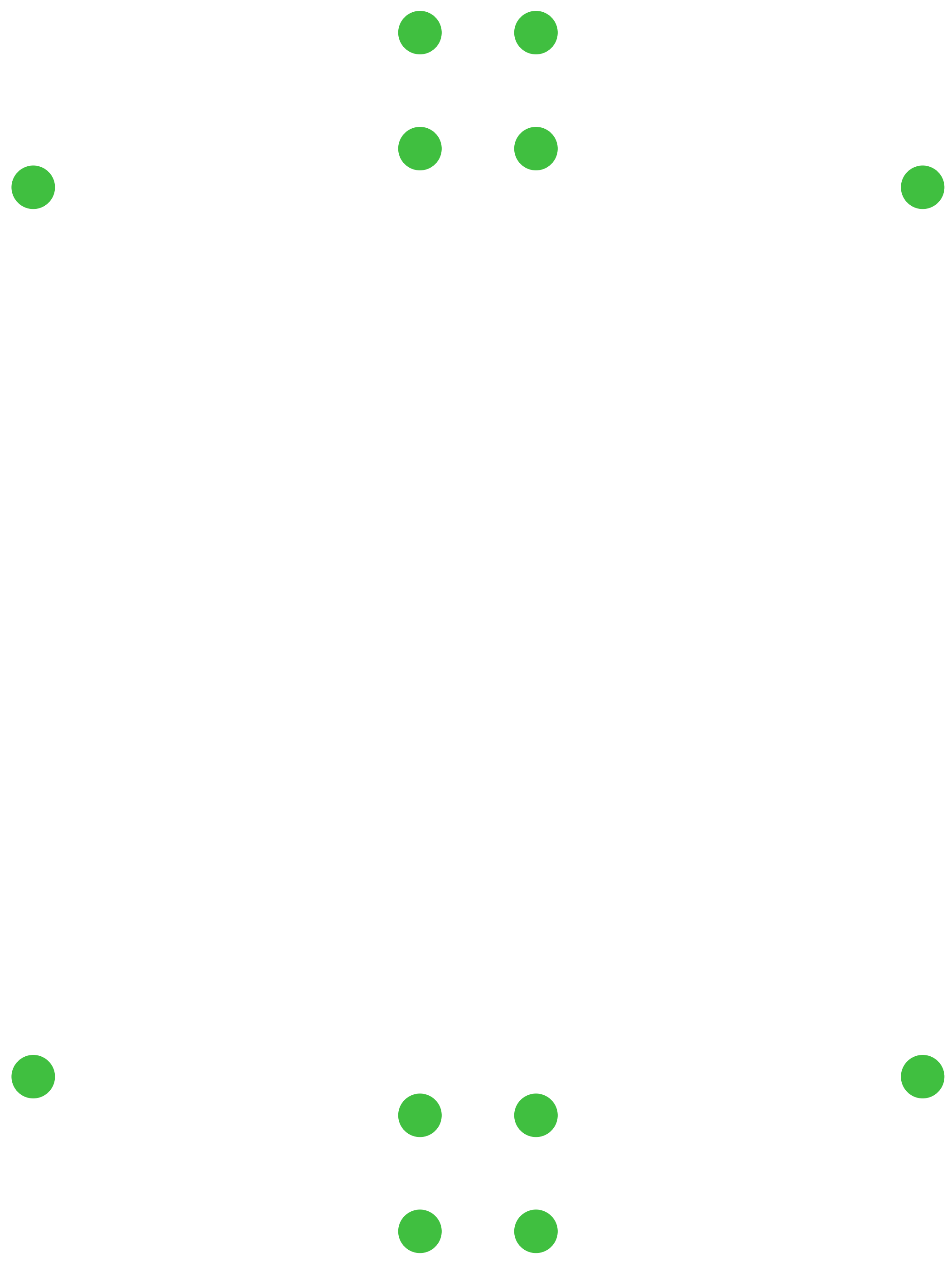
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



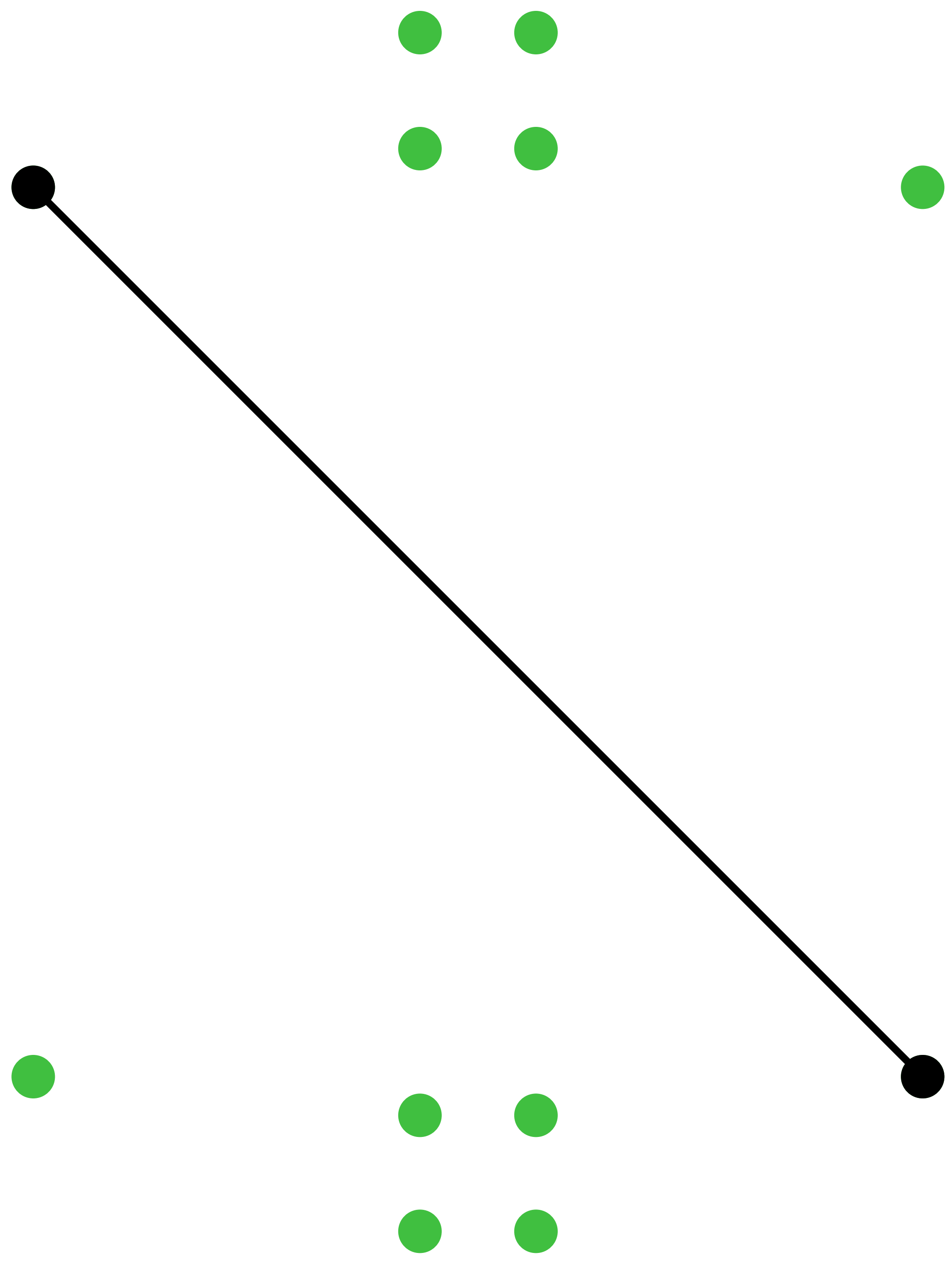
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



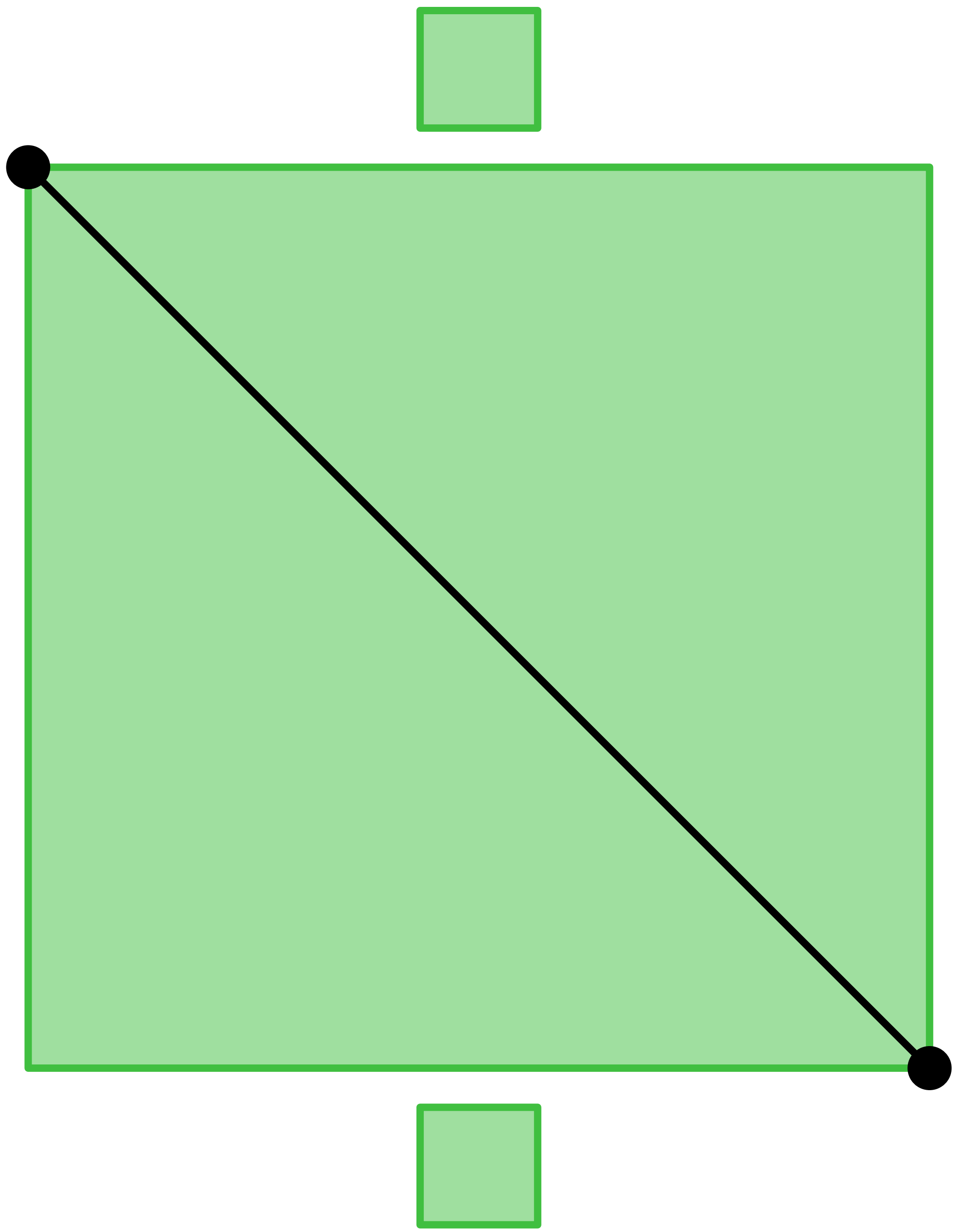
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



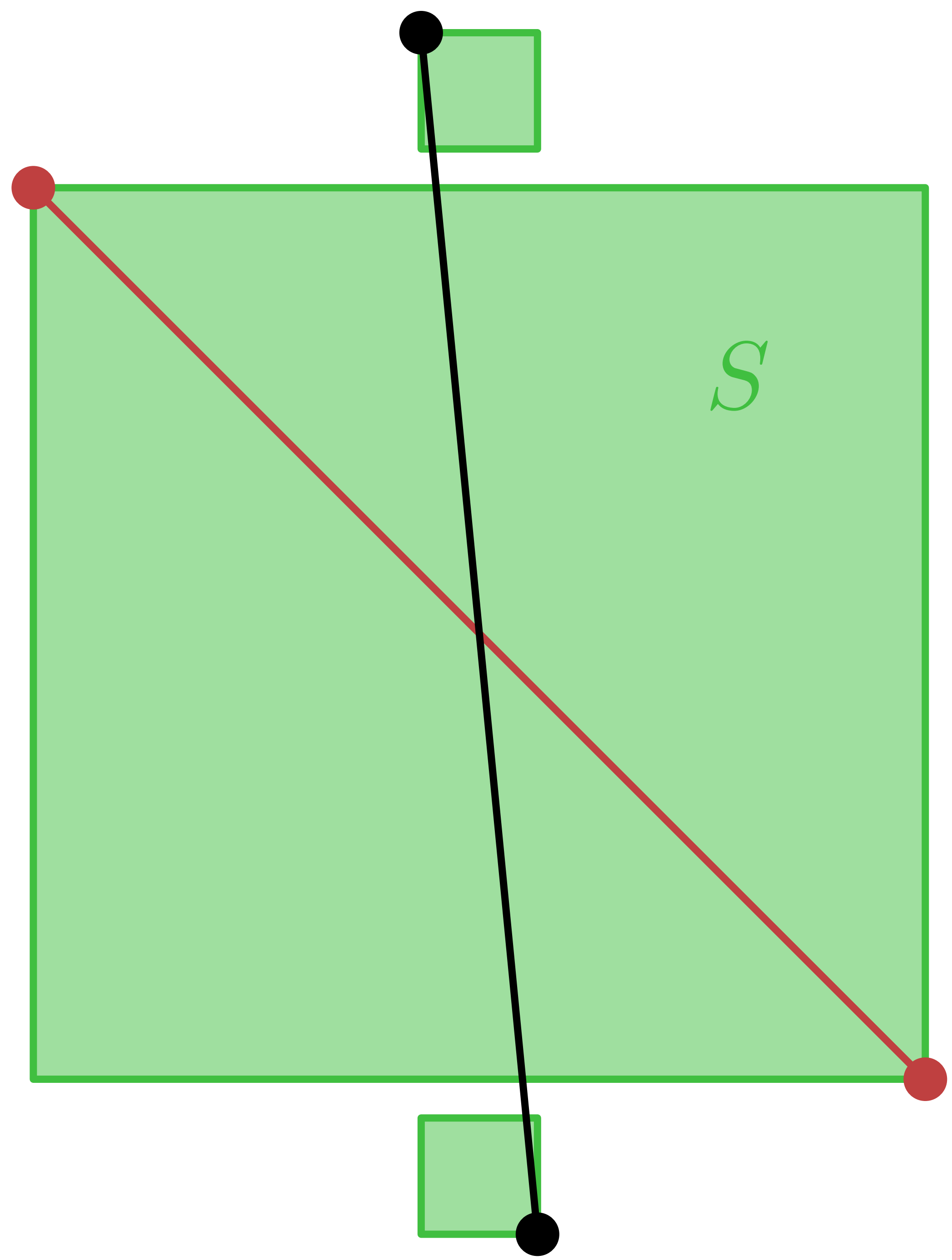
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



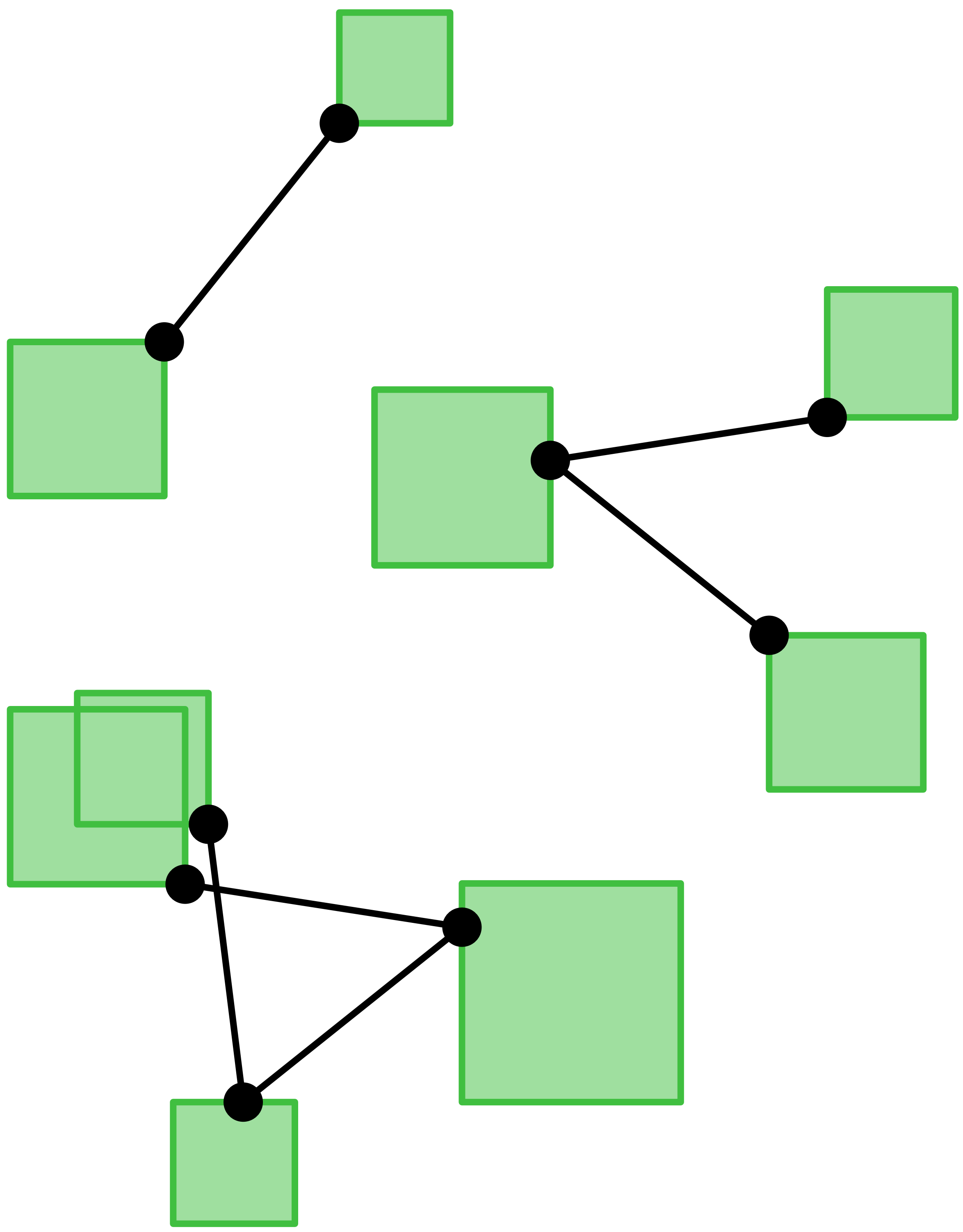
- Compute the diameter of all corners
- If the computed points belong to different regions, we are happy
- Otherwise, they are diagonally opposite corners of one big square S

Largest Diameter



- Diameter determined by two corners of S
- Two possible cases
- Either S contributes
  - Try all corner pairs
  - Only a linear number
- Or it does not
  - Compute diameter of remaining corners
  - Points cannot belong to one square again

Smallest Diameter

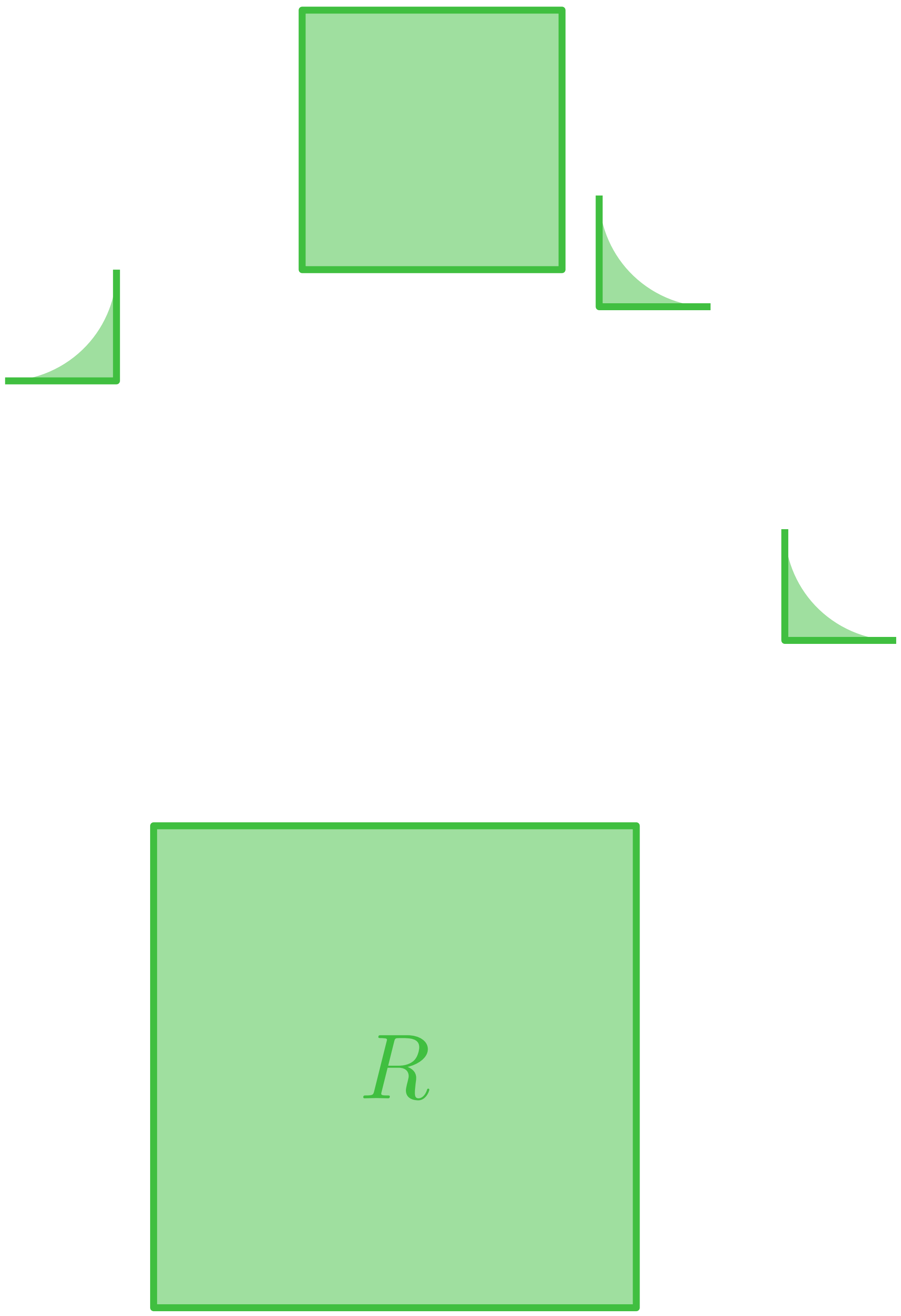


- Diameter could be determined by more pairs simultaneously
- The pairs form a *star*
- Bends only occur at axis-extreme squares
- Reflection angles must be less than 90°
- Consequence: at most two bends

$O(n^2)$ Algorithm

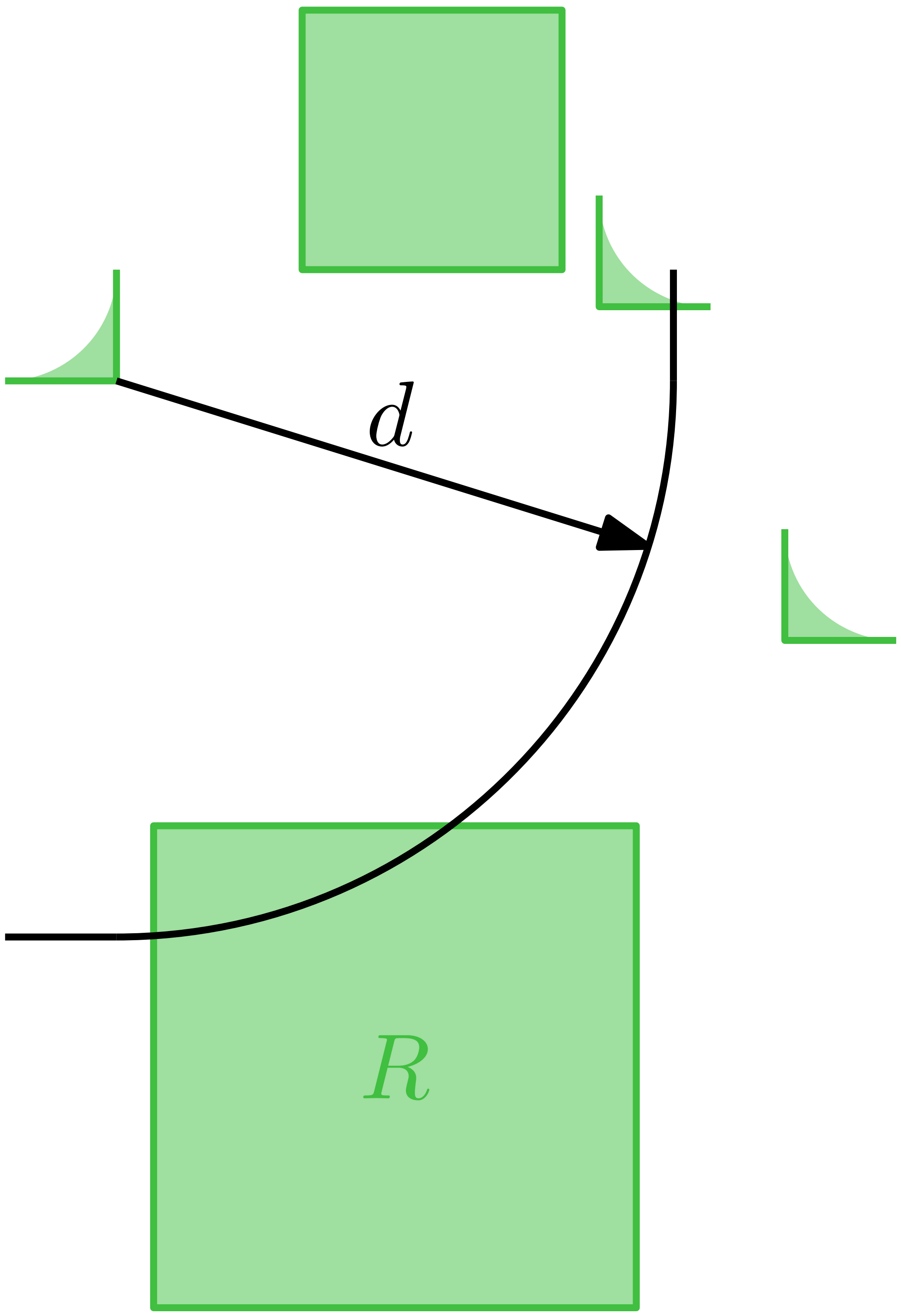
- The optimal star has at most two bends
- Compute the star for every subset of:
  - Two axis-extreme squares
  - And two other squares
- There are $O(n^2)$ stars to be computed
- The largest among these is the optimal star
- We now know the optimal diameter d and the star that defines it
- We still need to place points in all regions

$O(n^2)$ Algorithm



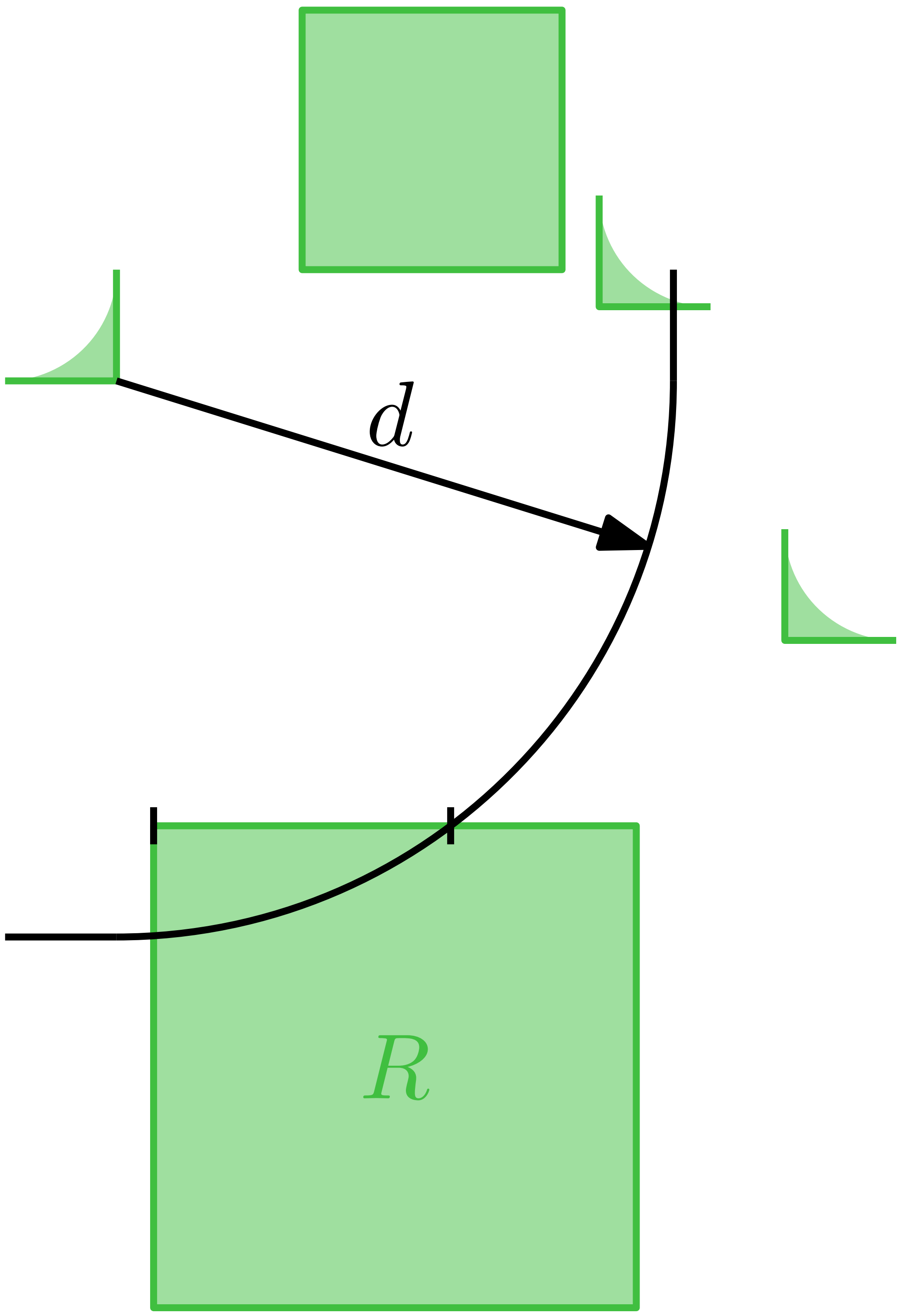
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



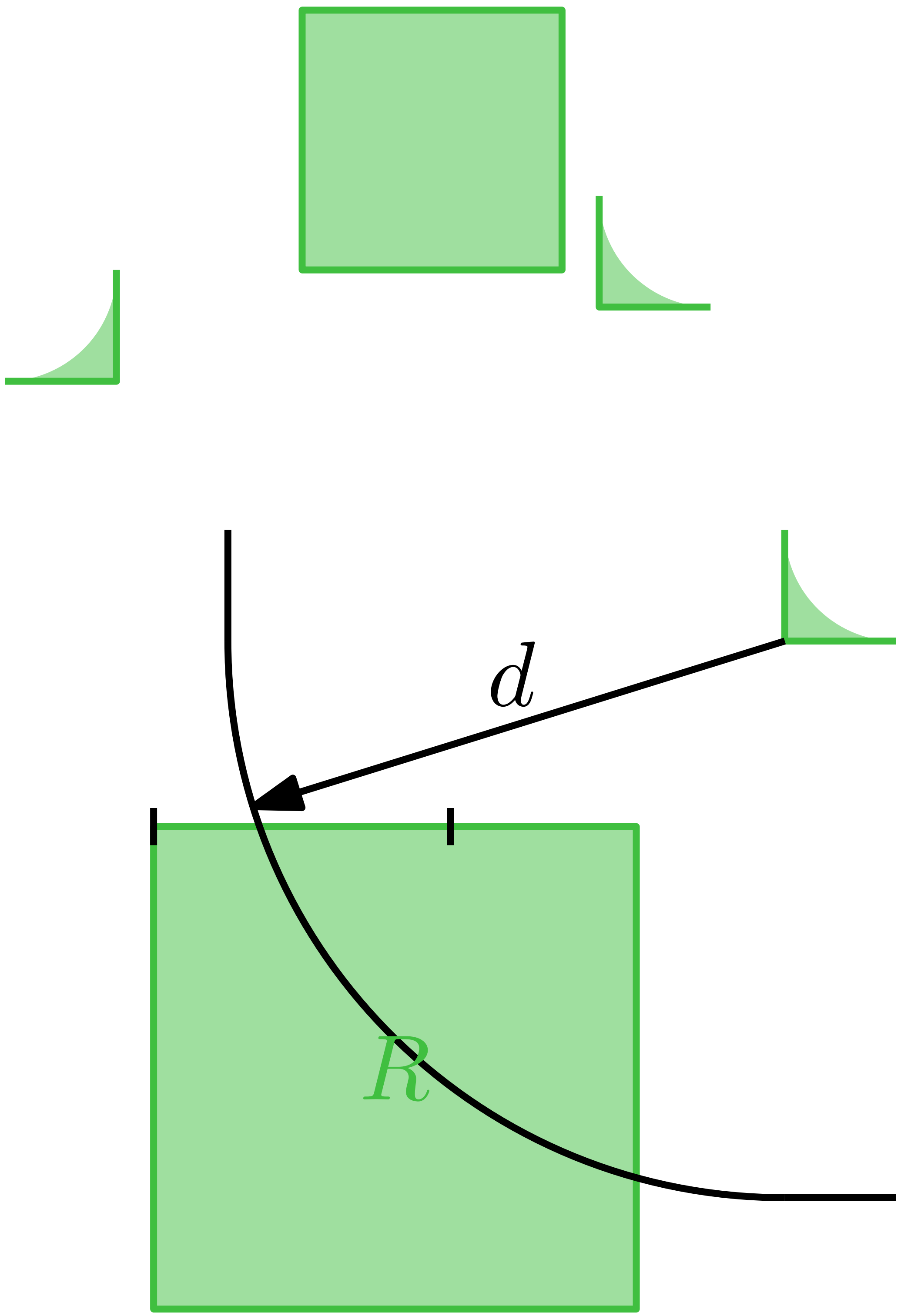
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



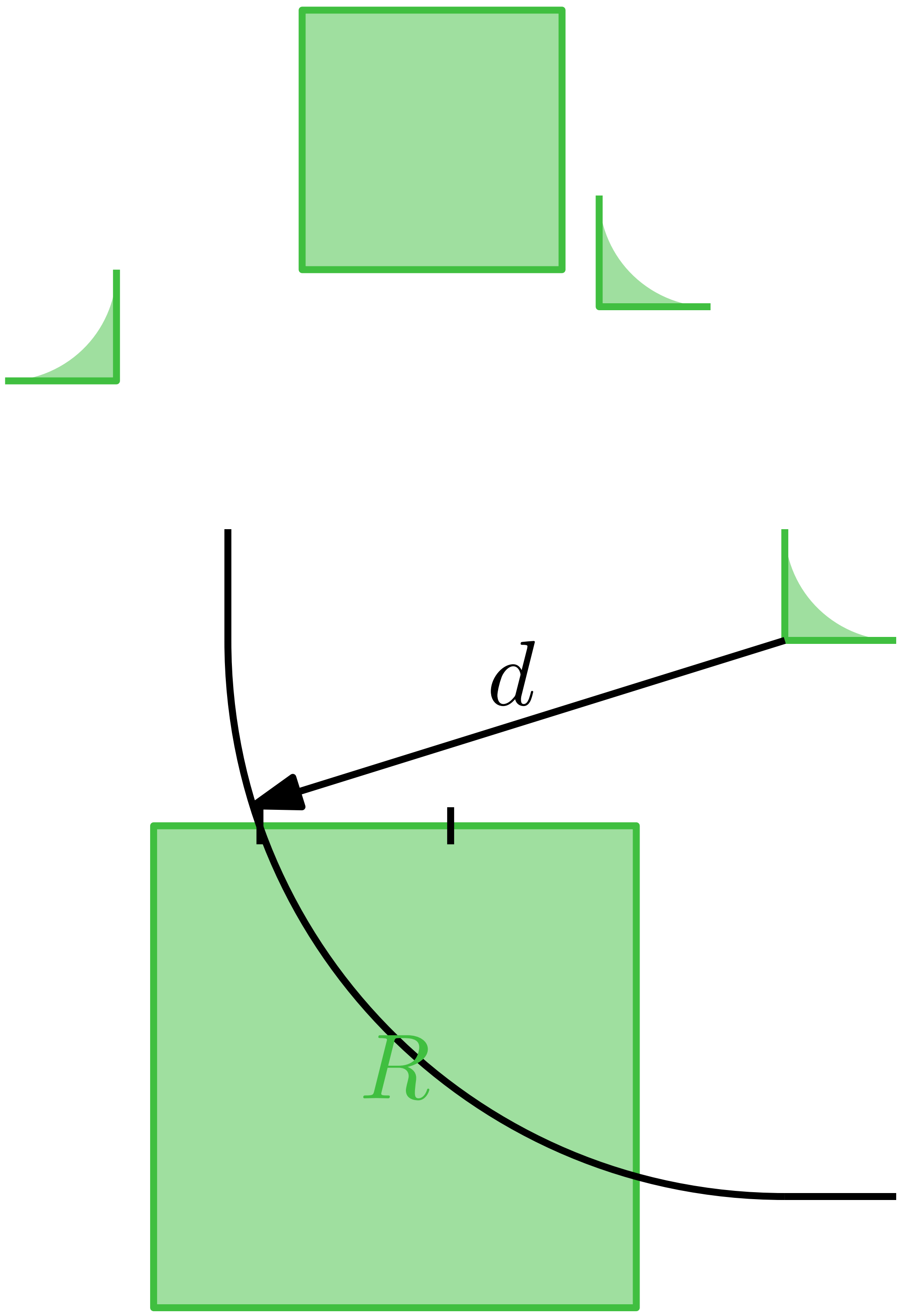
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



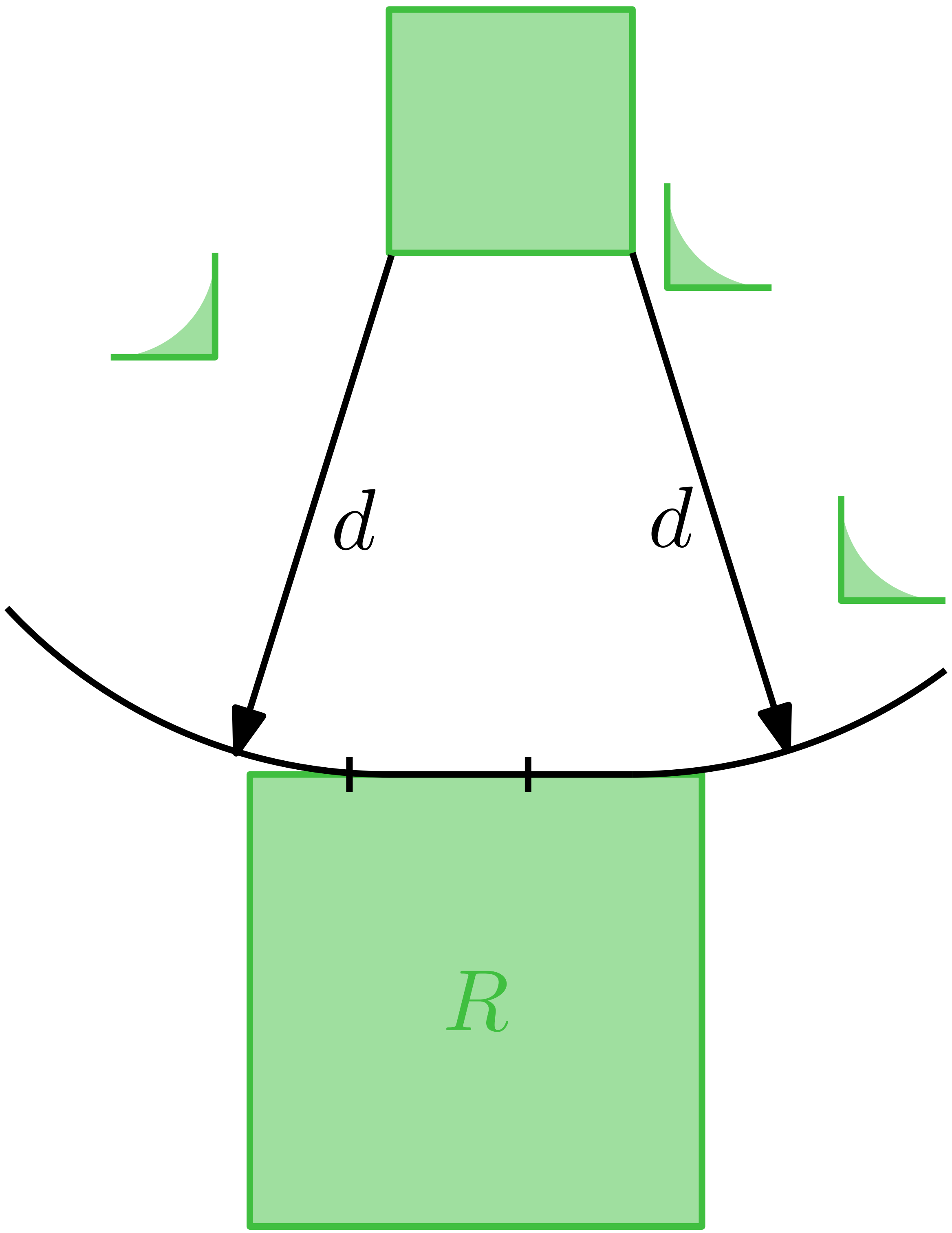
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



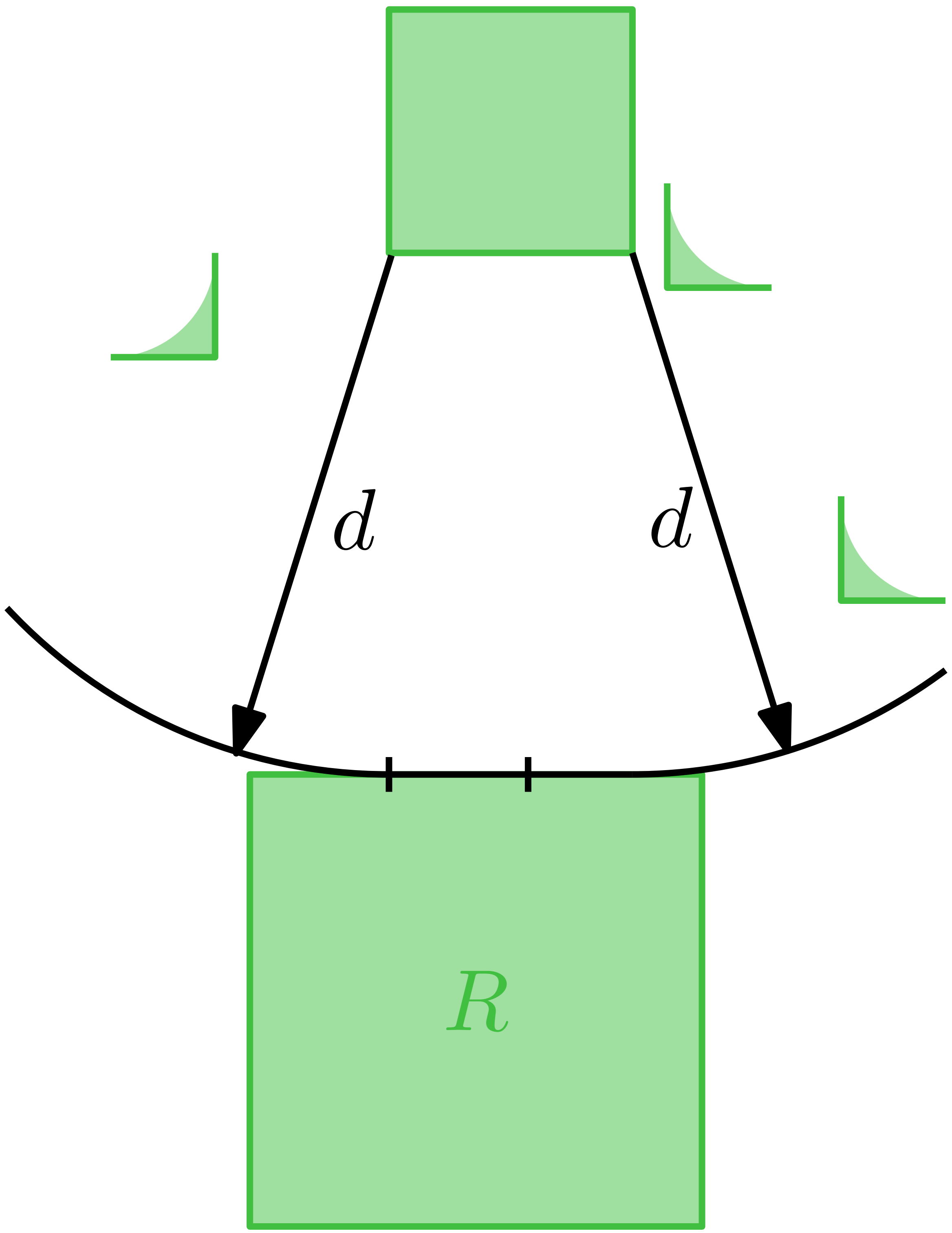
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



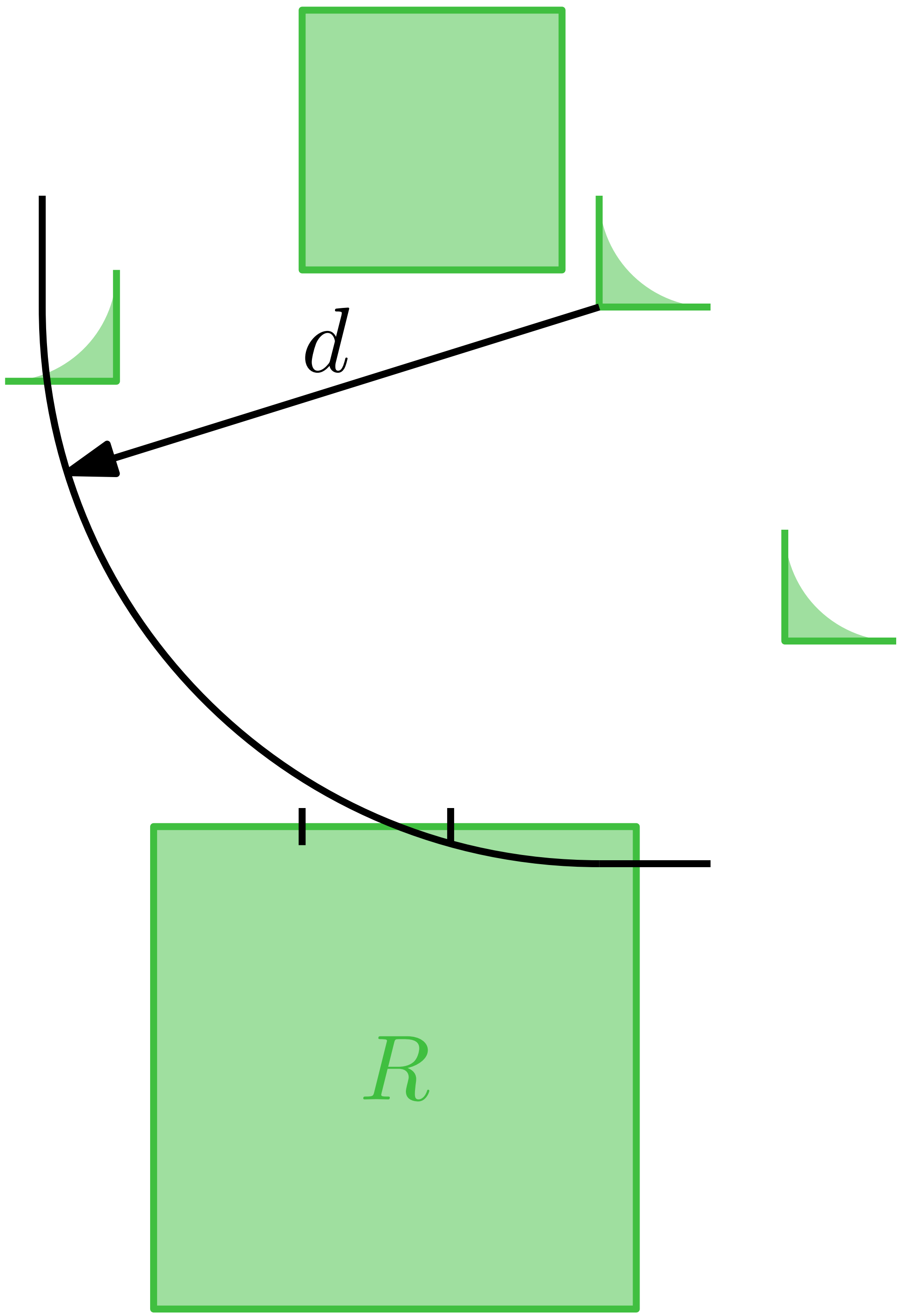
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



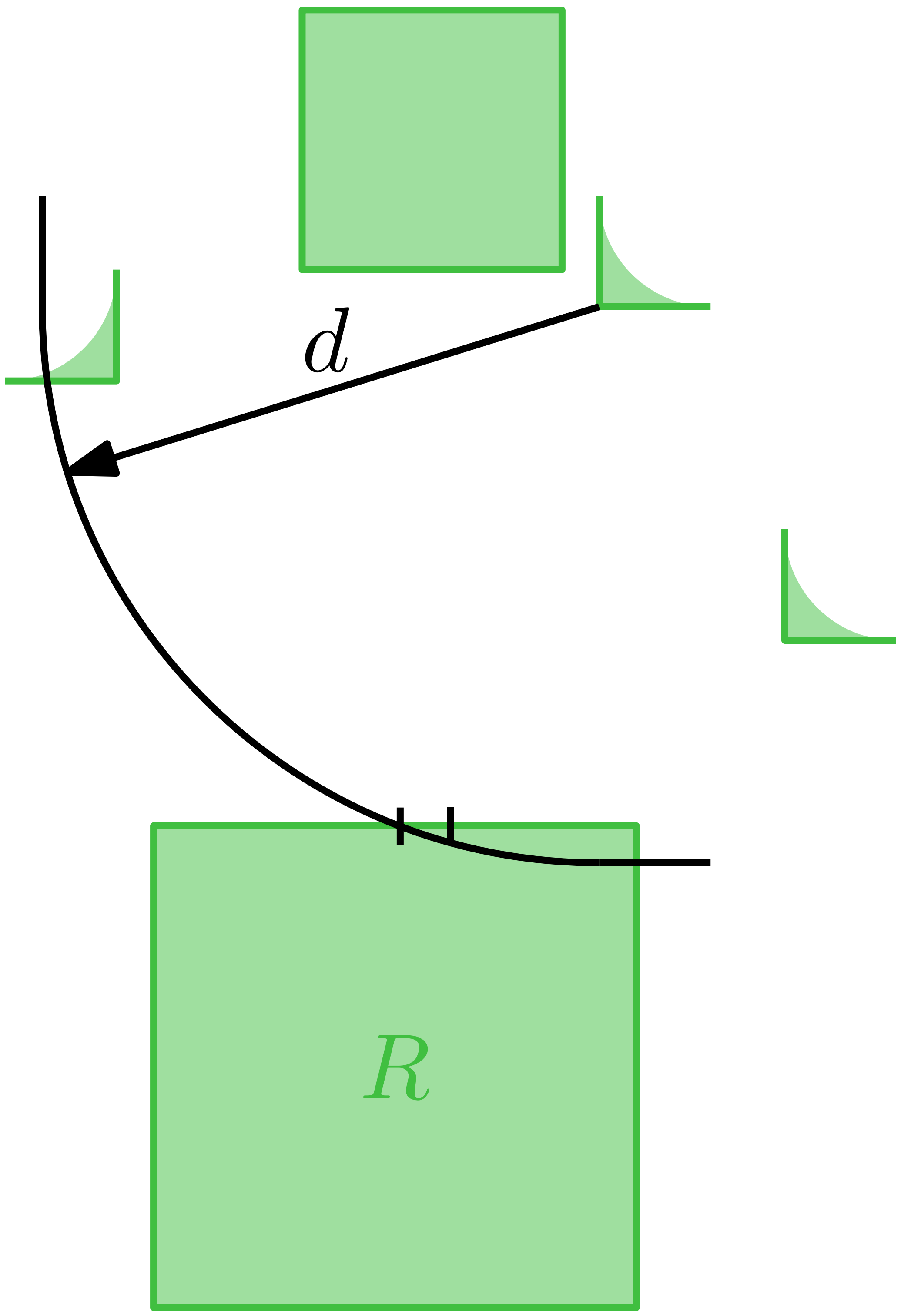
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



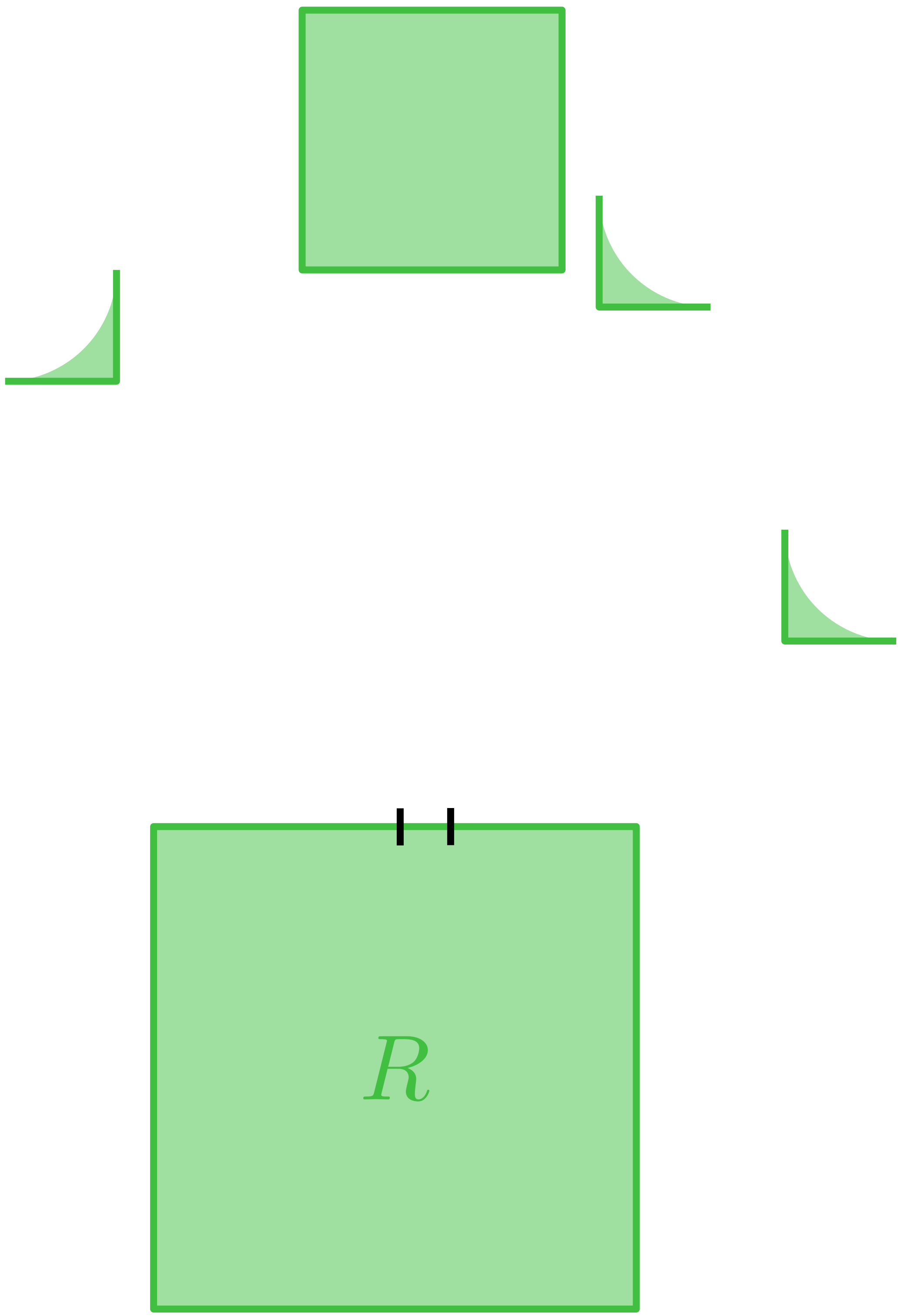
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



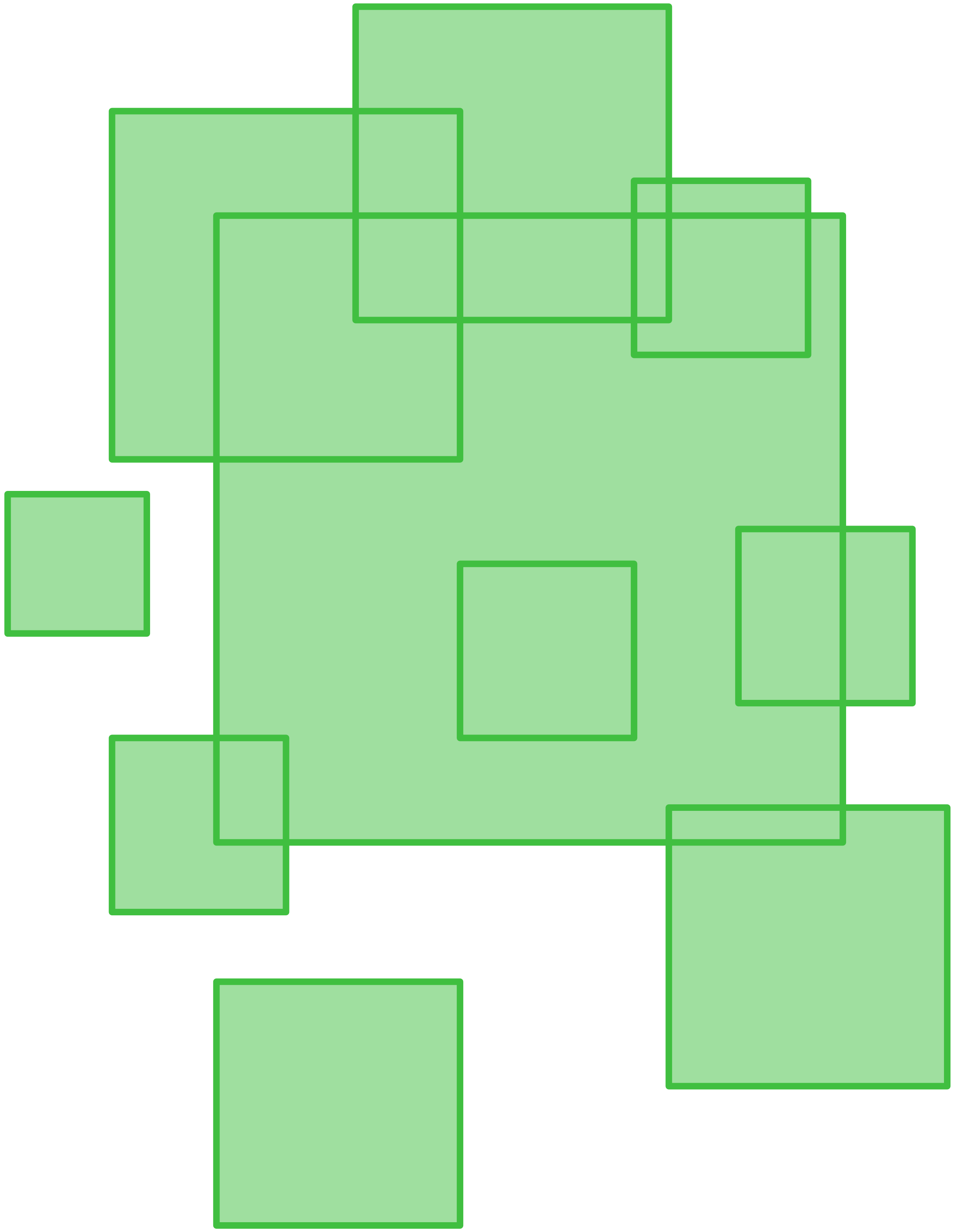
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



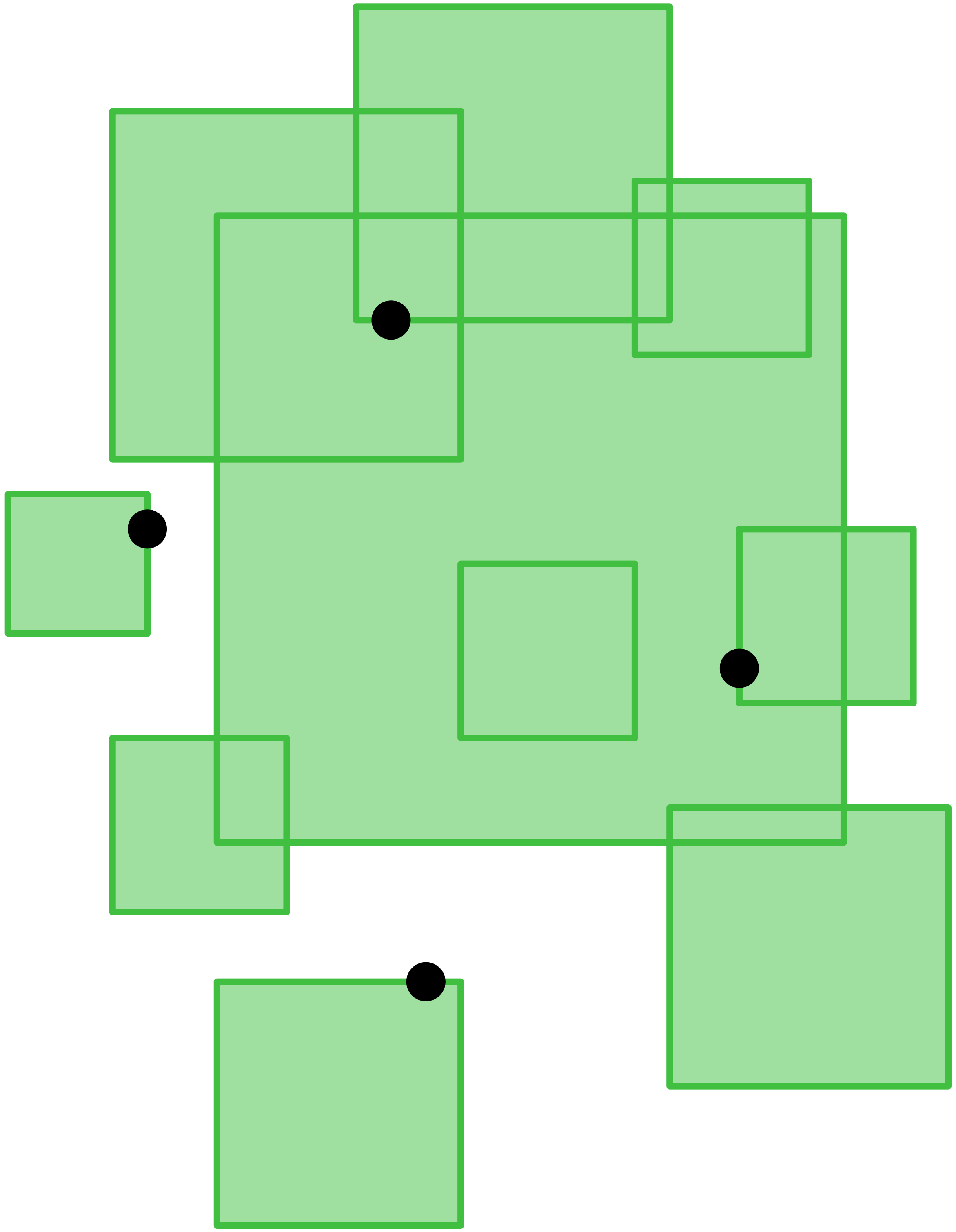
- We know d
- Let R be axis-extreme
- Valid placements in R are at most d away from any other square
- Place axis-extreme points validly and at most d away from each other
- This is sufficient

$O(n^2)$ Algorithm



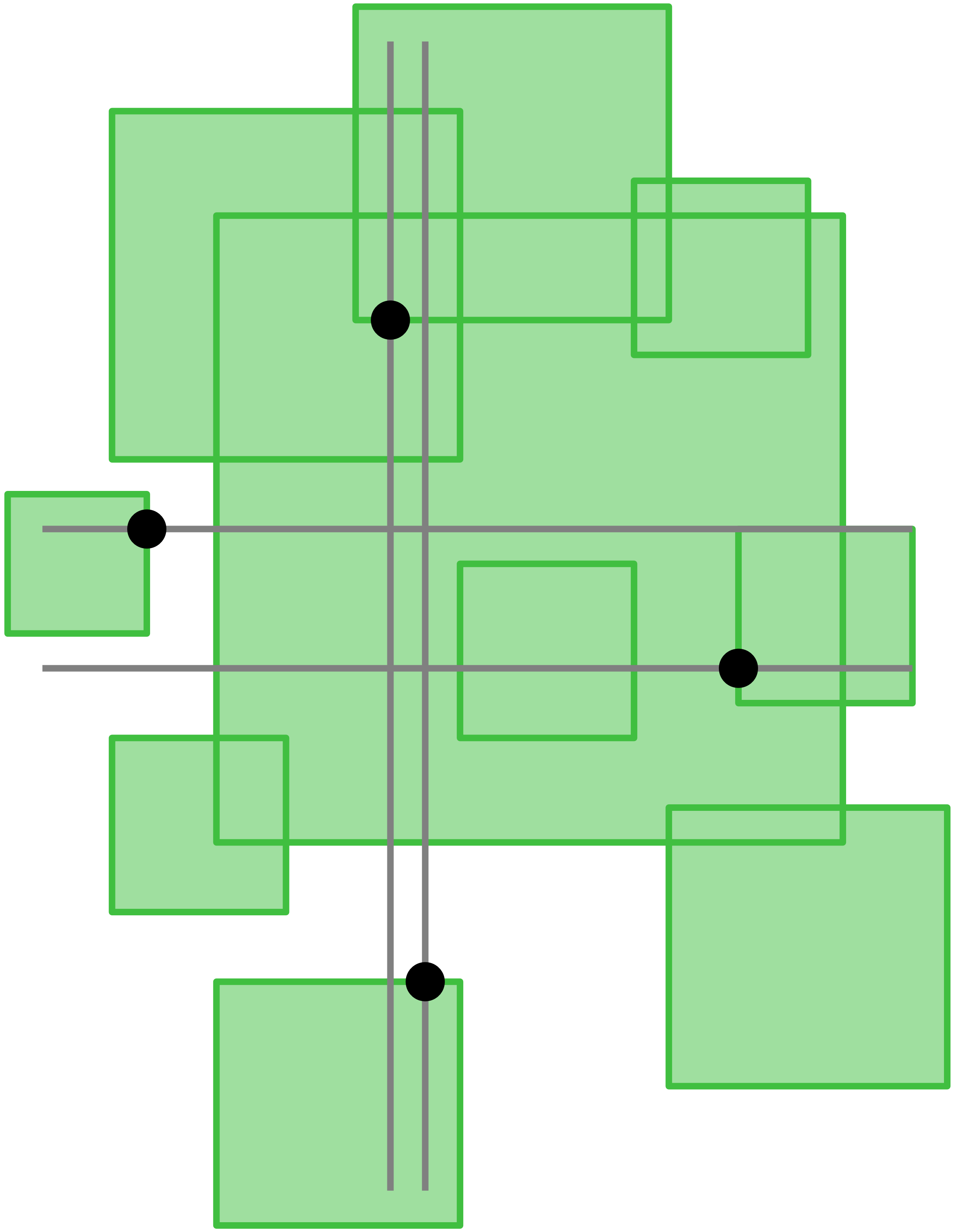
- We know the axis-extreme points
- Move rest of points towards the middle
- Optimal for given axis-extreme points
- Solution of d exists
- Therefore, the resulting point set must have diameter d

$O(n^2)$ Algorithm



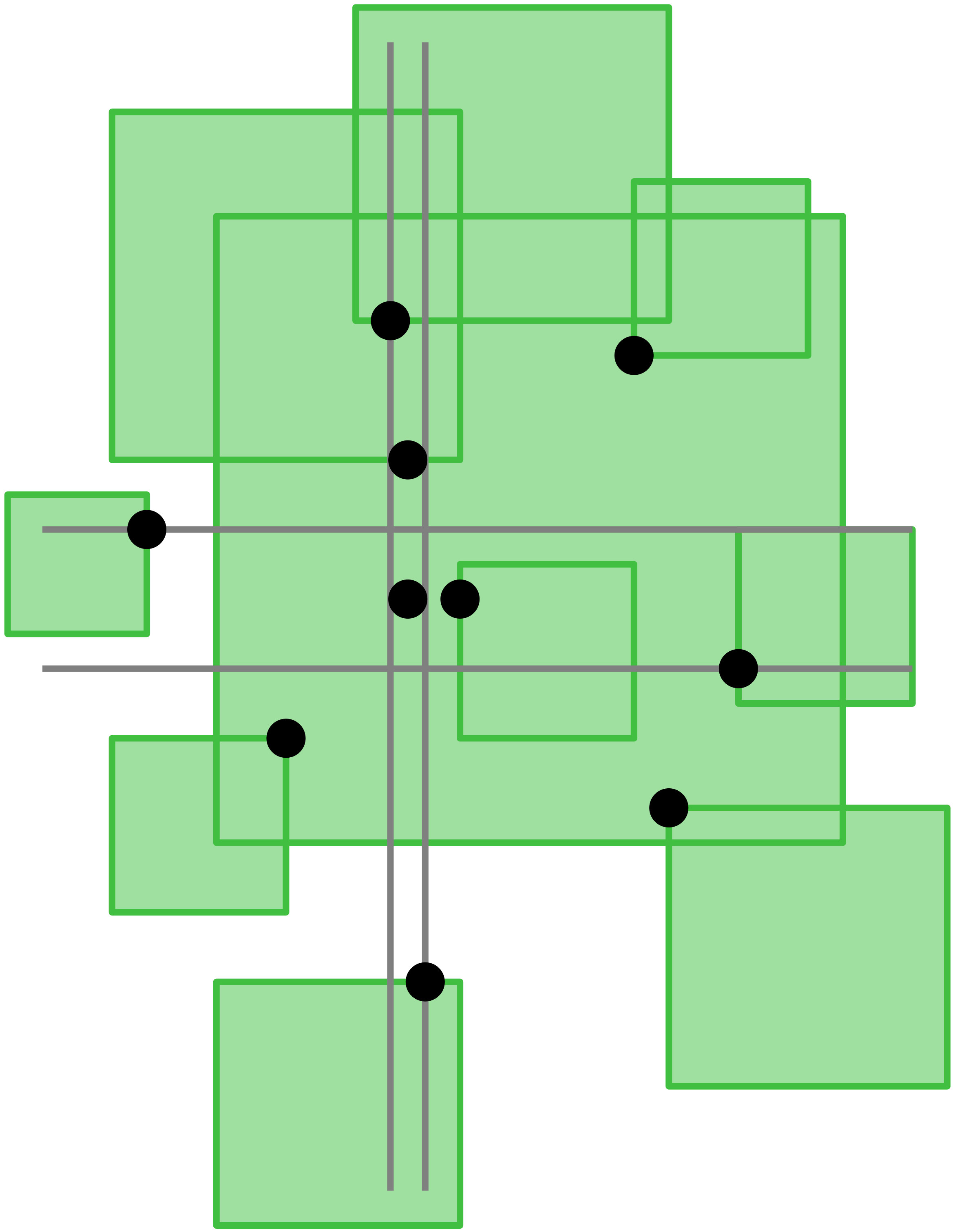
- We know the axis-extreme points
- Move rest of points towards the middle
- Optimal for given axis-extreme points
- Solution of d exists
- Therefore, the resulting point set must have diameter d

$O(n^2)$ Algorithm



- We know the axis-extreme points
- Move rest of points towards the middle
- Optimal for given axis-extreme points
- Solution of d exists
- Therefore, the resulting point set must have diameter d

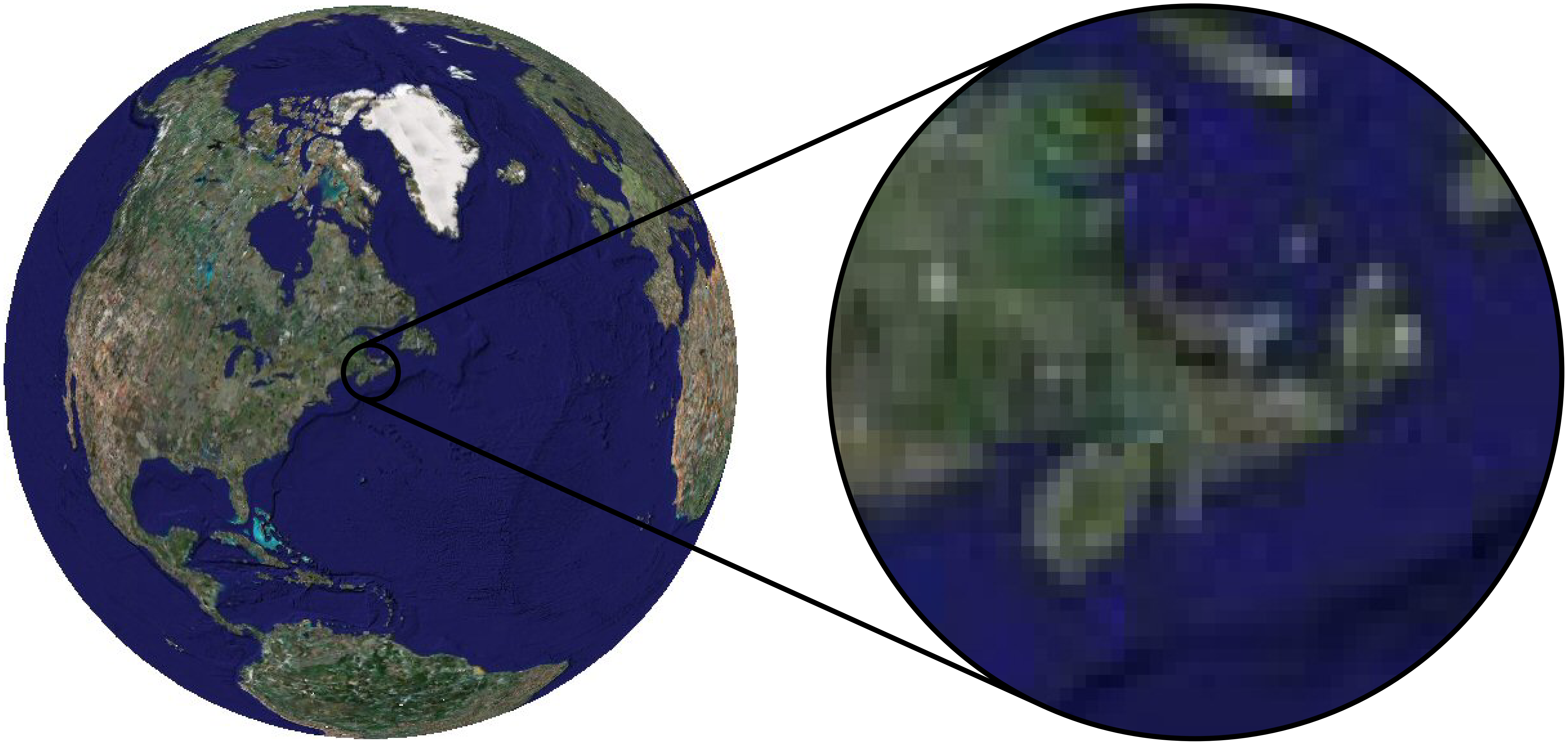
$O(n^2)$ Algorithm



- We know the axis-extreme points
- Move rest of points towards the middle
- Optimal for given axis-extreme points
- Solution of d exists
- Therefore, the resulting point set must have diameter d

Concluding Remarks

- Simple geometric problems become really interesting when input points are imprecise
- Largest and smallest diameter for squares can be computed efficiently
- Open problems
  - Largest width problem
  - Several variants of the convex hull
  - Third dimension?



Questions?