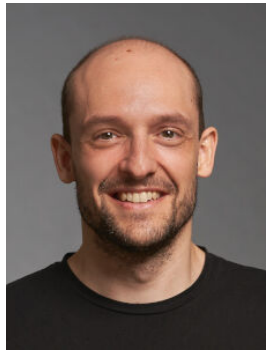


Computational Geometry

Exercise 1: Orga, Sheet 1, Max Distance

Elly, Wendy

Introductions

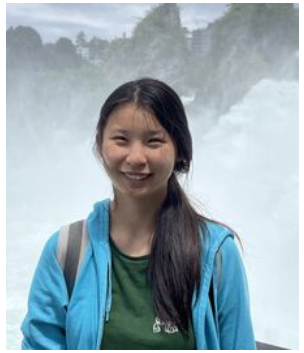


Thomas

Lecture



Elly



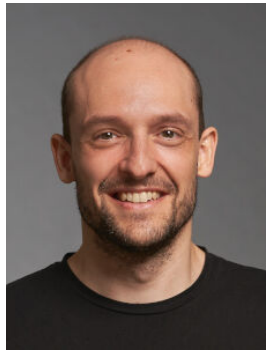
Wendy

Exercise

About you?

- name
- favorite lecture so far

Introductions

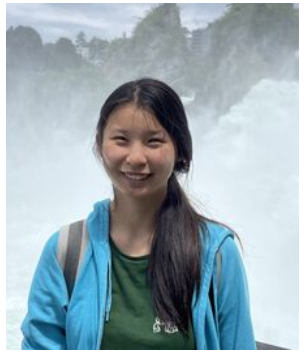


Thomas

Lecture



Elly



Wendy

Exercise

About you?

- name
- favorite lecture so far

Exercise Session

- two slots that are the same: 2pm, 3.45pm
- discuss last exercise sheet
- get hints for next exercise sheet
- work on other problems, get different perspectives

More Orga Stuff

Exercise Sheets

- new sheet every two weeks (Thu – Thu)
- submission via ILIAS (not handwritten, pls) unless your handwriting is really good
- work in pairs or groups of three!
- coding submissions (maybe sometimes)

More Orga Stuff

Exercise Sheets

- new sheet every two weeks (Thu – Thu)
- submission via ILIAS (not handwritten, pls) unless your handwriting is really good
- work in pairs or groups of three!
- coding submissions (maybe sometimes)

Points System

0%: did not write anything

25%: had ideas, explained where you are stuck

50%: basic idea is correct (and explained)

75%: small/few details missing

100%: correct and well-written

More Orga Stuff

Exercise Sheets

- new sheet every two weeks (Thu – Thu)
- submission via ILIAS (not handwritten, pls) unless your handwriting is really good
- work in pairs or groups of three!
- coding submissions (maybe sometimes)

Points System

0%: did not write anything

25%: had ideas, explained where you are stuck

50%: basic idea is correct (and explained)

75%: small/few details missing

100%: correct and well-written

Goals

- at least $\frac{1}{2}$ of total points, $\frac{1}{4}$ of every sheet
- apply stuff learned in lectures
- think about problems
- practice writing solutions

More Orga Stuff

Exercise Sheets

- new sheet every two weeks (Thu – Thu)
- submission via ILIAS (not handwritten, pls)
  unless your handwriting is really good
- work in pairs or groups of three!
- coding submissions (maybe sometimes)

Talk to each other, ask for hints on Discord, ask us in person!

send `!help join` to the scale bot

Points System

0%: did not write anything

25%: had ideas, explained where you are stuck

50%: basic idea is correct (and explained)

75%: small/few details missing

100%: correct and well-written

Goals

- at least $\frac{1}{2}$ of total points, $\frac{1}{4}$ of every sheet
- apply stuff learned in lectures
- think about problems
- practice writing solutions

How to: Assignment Sheet

- show correctness [and argue running time]

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)*(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

Let P be the intersection of lines g and h.

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode
- be concise and precise, no AI bullshitting

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)*(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

The **intersection of two lines** is the point where the two lines cross each other. To calculate it, we need to find the coordinates of the point that lies on **both lines simultaneously**.

There are several ways to do this, depending on how the lines are written. We'll start with the most common method, explain every step in detail, and then explore other methods and special cases.

1. Understanding what an intersection is

Imagine two straight lines drawn on a coordinate plane.

Let P be the intersection of lines g and h.

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode
- be concise and precise, no AI bullshitting

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

The **intersection of two lines** is the point where the two lines cross each other. To calculate it, we need to find the coordinates of the point that lies on **both lines simultaneously**.

There are several ways to do this, depending on how the lines are given. We'll start with the most common method, explain every step in detail, and then explore other methods and special cases.

1. Understanding what an intersection is

Imagine two straight lines drawn on a coordinate plane.

Let P be the intersection of lines g and h.

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode
- be concise and precise, no AI bullshitting
- choose proper abstraction level

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)*(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

The **intersection of two lines** is the point where the two lines cross each other. To calculate it, we need to find the coordinates of the point that lies on **both lines simultaneously**.

There are several ways to do this, depending on how the lines are given. We'll start with the most common method, explain every step in detail, and then explore other methods and special cases.

1. Understanding what an intersection is

Imagine two straight lines drawn on a coordinate plane.

Let P be the intersection of lines g and h.

Given two points of each line, first determine formulas $tx + y$ for lines. Then solve the equation system for x and y.

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode
- be concise and precise, no AI bullshitting
- choose proper abstraction level

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)*(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

The **intersection of two lines** is the point where the two lines cross each other. To calculate it, we need to find the coordinates of the point that lies on **both lines simultaneously**.

There are several ways to do this, depending on how the lines are given. We'll start with the most common method, explain every step in detail, and then explore other methods and special cases.

1. Understanding what an intersection is

Imagine two straight lines drawn on a coordinate plane.

Let P be the intersection of lines g and h.

Given two points of each line, first determine formulas $tx + y$ for lines. Then solve the equation system for x and y.

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode
- be concise and precise, no AI bullshitting
- choose proper abstraction level
- drawings are (often) helpful

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)*(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

The **intersection of two lines** is the point where the two lines cross each other. To calculate it, we need to find the coordinates of the point that lies on **both lines simultaneously**.

There are several ways to do this, depending on how the lines are given. We'll start with the most common method, explain every step in detail, and then explore other methods and special cases.

1. Understanding what an intersection is

Imagine two straight lines drawn on a coordinate plane.

Let P be the intersection of lines g and h.

Given two points of each line, first determine formulas $tx + y$ for lines. Then solve the equation system for x and y.

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode
- be concise and precise, no AI bullshitting
- choose proper abstraction level
- drawings are (often) helpful

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

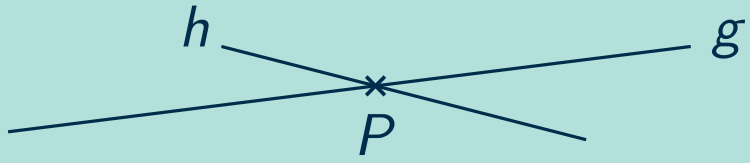
```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)*(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

The **intersection of two lines** is the point where the two lines cross each other. To calculate it, we need to find the coordinates of the point that lies on **both lines simultaneously**.

There are several ways to do this, depending on how the lines are given. We'll start with the most common method, explain every step in detail, and then explore other methods and special cases.

1. Understanding what an intersection is

Imagine two straight lines drawn on a coordinate plane.

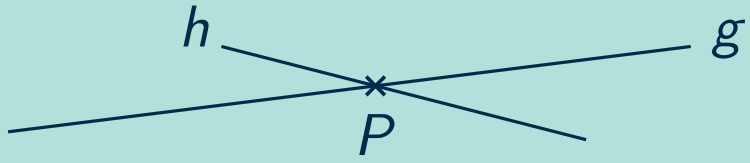


Let P be the intersection of lines g and h.

Given two points of each line, first determine formulas $tx + y$ for lines. Then solve the equation system for x and y.

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode
- be concise and precise, no AI bullshitting
- choose proper abstraction level
- drawings are (often) helpful
- consider edge cases



Let P be the intersection of lines g and h.

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)*(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

The **intersection of two lines** is the point where the two lines cross each other. To calculate it, we need to find the coordinates of the point that lies on **both lines simultaneously**.

There are several ways to do this, depending on how the lines are given. We'll start with the most common method, explain every step in detail, and then explore other methods and special cases.

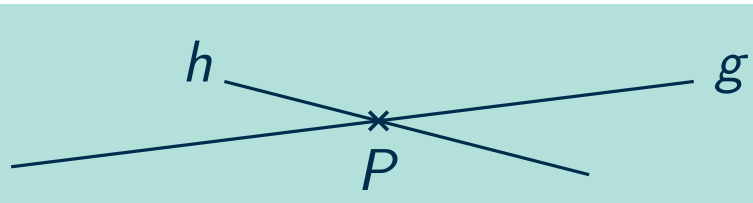
1. Understanding what an intersection is

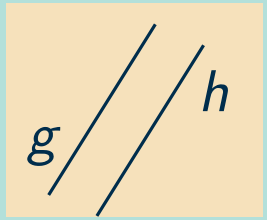
Imagine two straight lines drawn on a coordinate plane.

Given two points of each line, first determine formulas $tx + y$ for lines. Then solve the equation system for x and y.

How to: Assignment Sheet

- show correctness [and argue running time]
- important: convey (all of) the main idea
- explanations $>$ mathematical formulas
- natural language $>>$ pseudocode
- be concise and precise, no AI bullshitting
- choose proper abstraction level
- drawings are (often) helpful
- consider edge cases





Let P be the intersection of lines g and h.

$$P_x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$P_y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

```
FUNCTION Calculate(A, B, C, D)
    denom ← (A.x-B.x)*(C.y-D.y) - (A.y-B.y)*(C.x-D.x)
    IF ABS(denom) < EPSILON THEN RETURN NONE
    t ← ((A.x-C.x)*(C.y-D.y) - (A.y-C.y)*(C.x-D.x)) / denom
    RETURN Point(A.x + t*(B.x-A.x), A.y + t*(B.y-A.y))
END FUNCTION
```

The **intersection of two lines** is the point where the two lines cross each other. To calculate it, we need to find the coordinates of the point that lies on **both lines simultaneously**.

There are several ways to do this, depending on how the lines are given. We'll start with the most common method, explain every step in detail, and then explore other methods and special cases.

1. Understanding what an intersection is

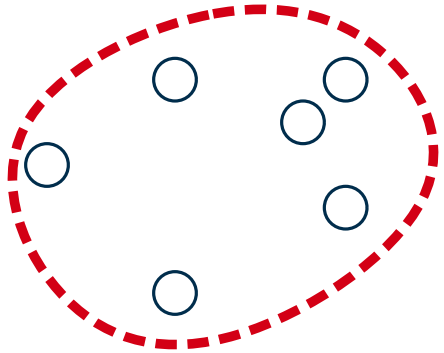
Imagine two straight lines drawn on a coordinate plane.

Given two points of each line, first determine formulas $tx + y$ for lines. Then solve the equation system for x and y.

Assignment Sheet 1

Mysterious Meteorites

(a)

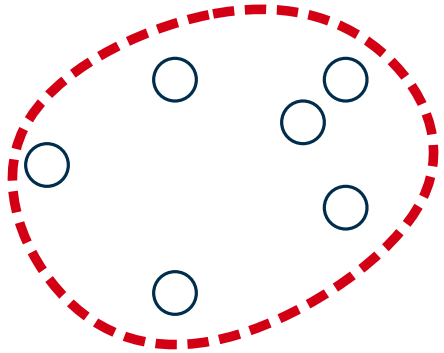


Q: min barrier tape?

Assignment Sheet 1

Mysterious Meteorites

(a)



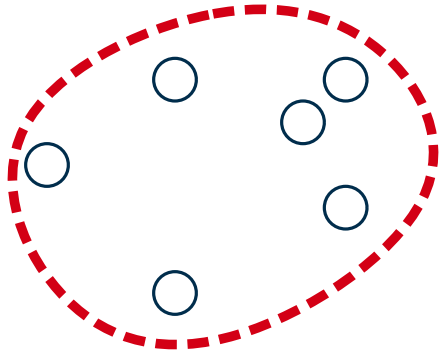
explain high-level

Q: min barrier tape?

Assignment Sheet 1

Mysterious Meteorites

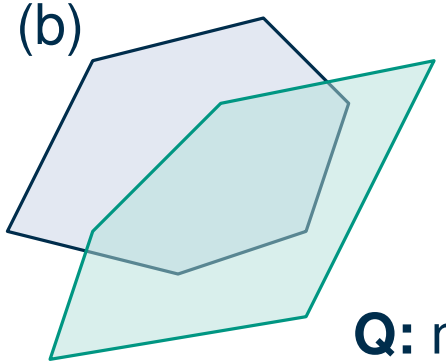
(a)



explain high-level

Q: min barrier tape?

(b)

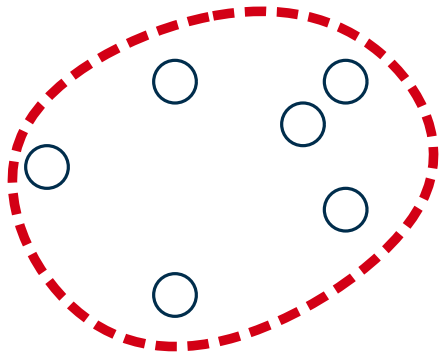


Q: max # corners?

Assignment Sheet 1

Mysterious Meteorites

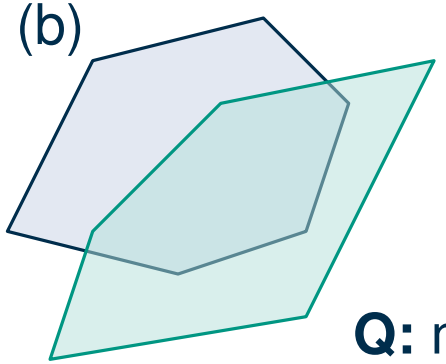
(a)



explain high-level

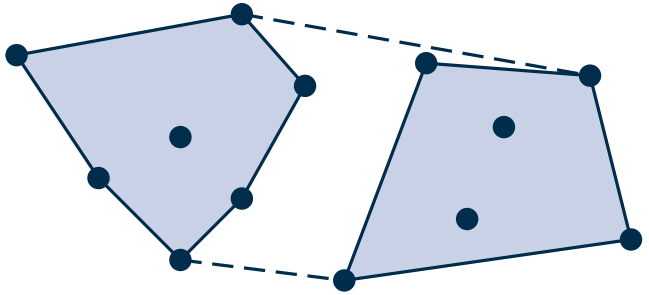
Q: min barrier tape?

(b)



Q: max # corners?

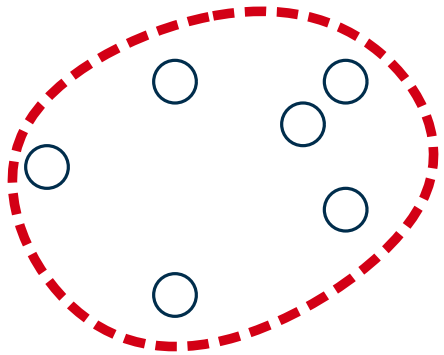
Convex Hull via Divide and Conquer



Assignment Sheet 1

Mysterious Meteorites

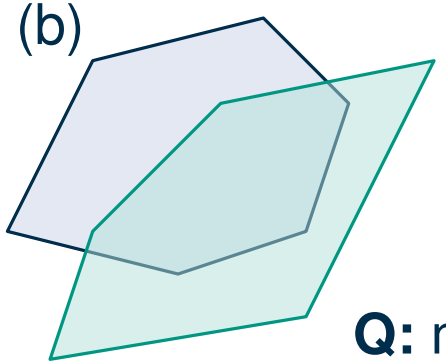
(a)



explain high-level

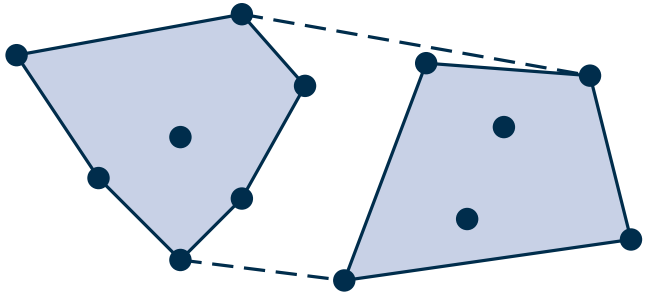
Q: min barrier tape?

(b)



Q: max # corners?

Convex Hull via Divide and Conquer

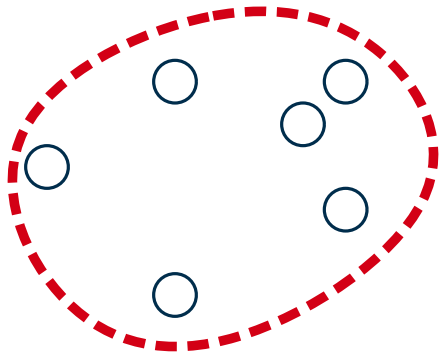


How does the convex hull of two polygons look like?

Assignment Sheet 1

Mysterious Meteorites

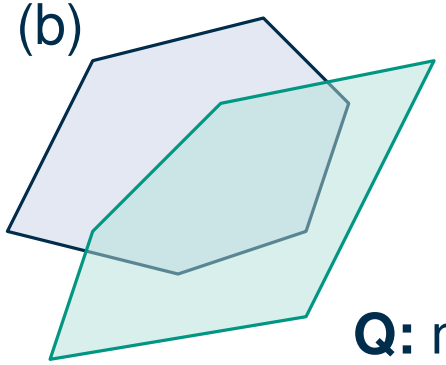
(a)



explain high-level

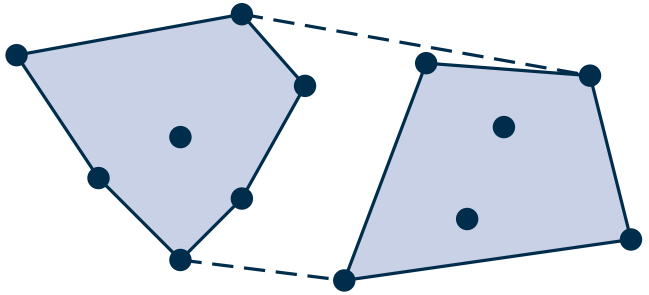
Q: min barrier tape?

(b)



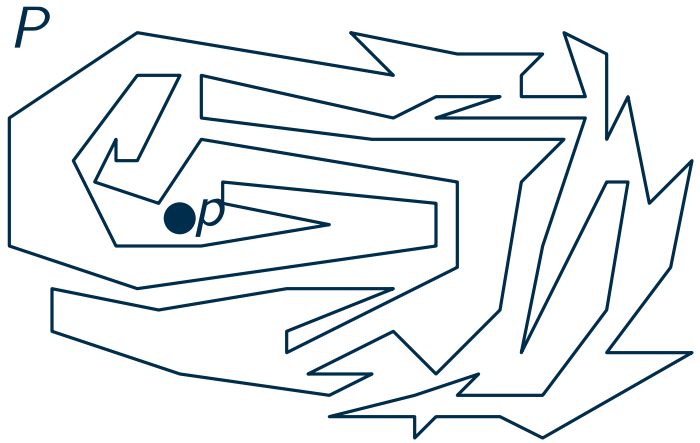
Q: max # corners?

Convex Hull via Divide and Conquer



How does the convex hull of two polygons look like?

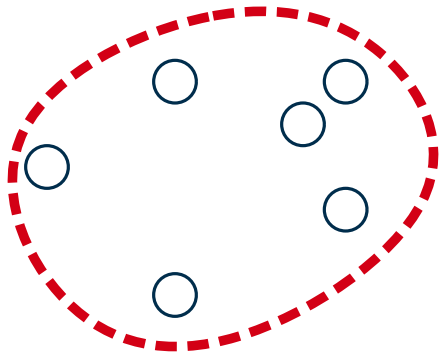
Aerial Photography



Assignment Sheet 1

Mysterious Meteorites

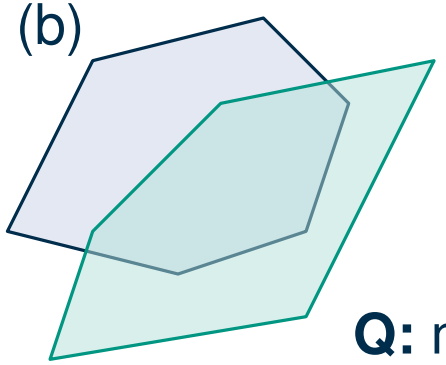
(a)



explain high-level

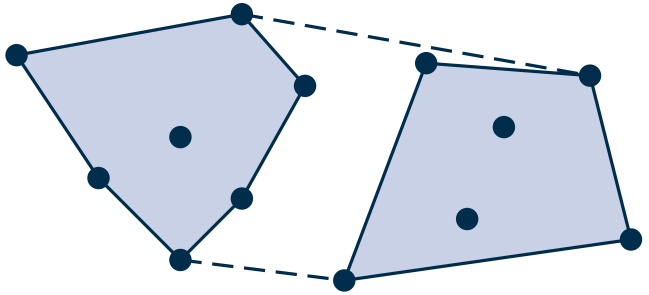
Q: min barrier tape?

(b)



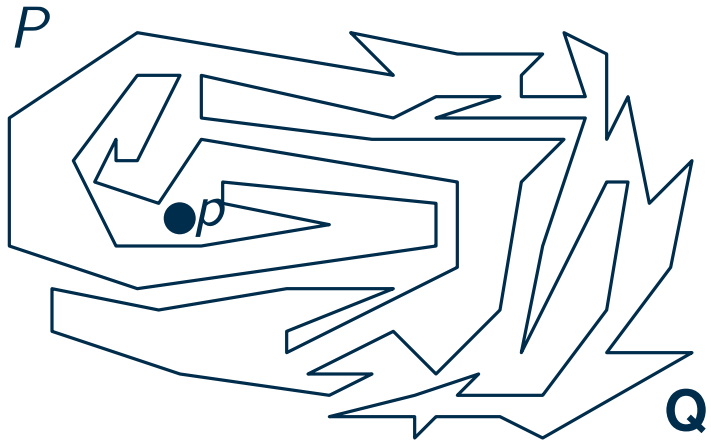
Q: max # corners?

Convex Hull via Divide and Conquer



How does the convex hull of two polygons look like?

Aerial Photography

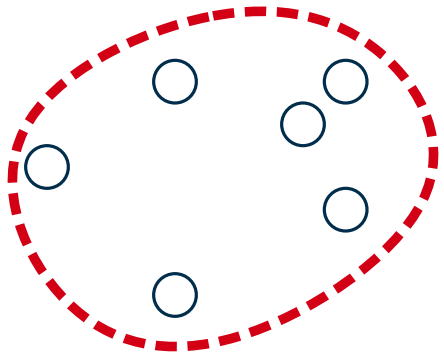


Q: Is p in P?

Assignment Sheet 1

Mysterious Meteorites

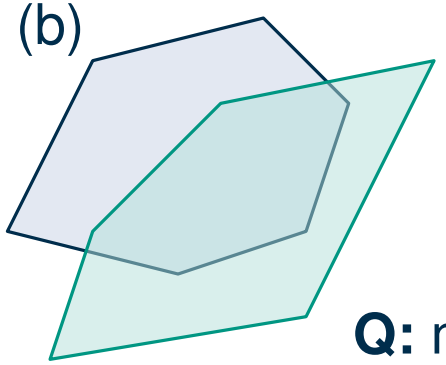
(a)



explain high-level

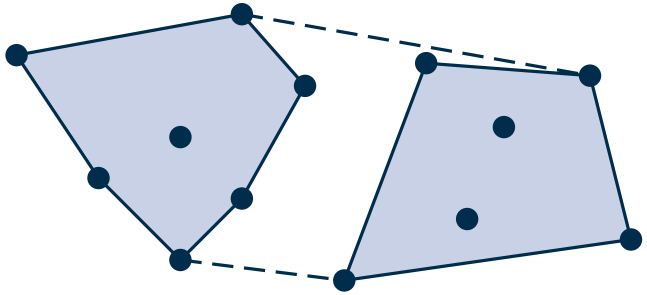
Q: min barrier tape?

(b)



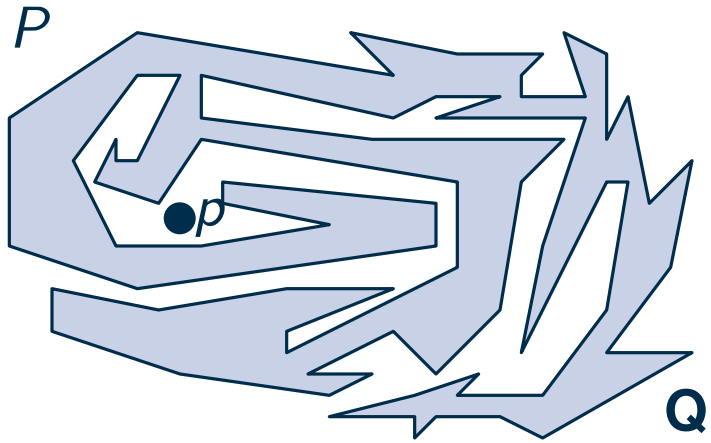
Q: max # corners?

Convex Hull via Divide and Conquer



How does the convex hull of two polygons look like?

Aerial Photography



edge cases?

Q: Is p in P?



What? Volunteer for NWERC

Details? Programming Contest

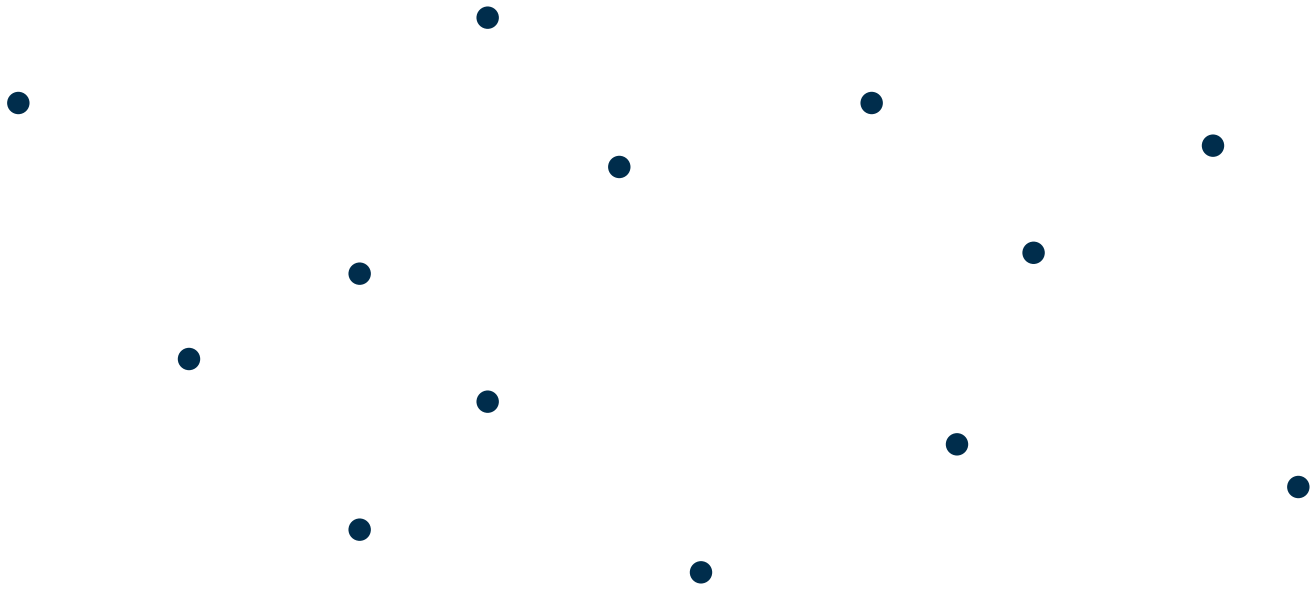
Why? 🥳 + Goodies, Shirt

Register? Scan QR-Code

https://volunteers.cpulm.org/nwerc-2026/

Maximum Distance

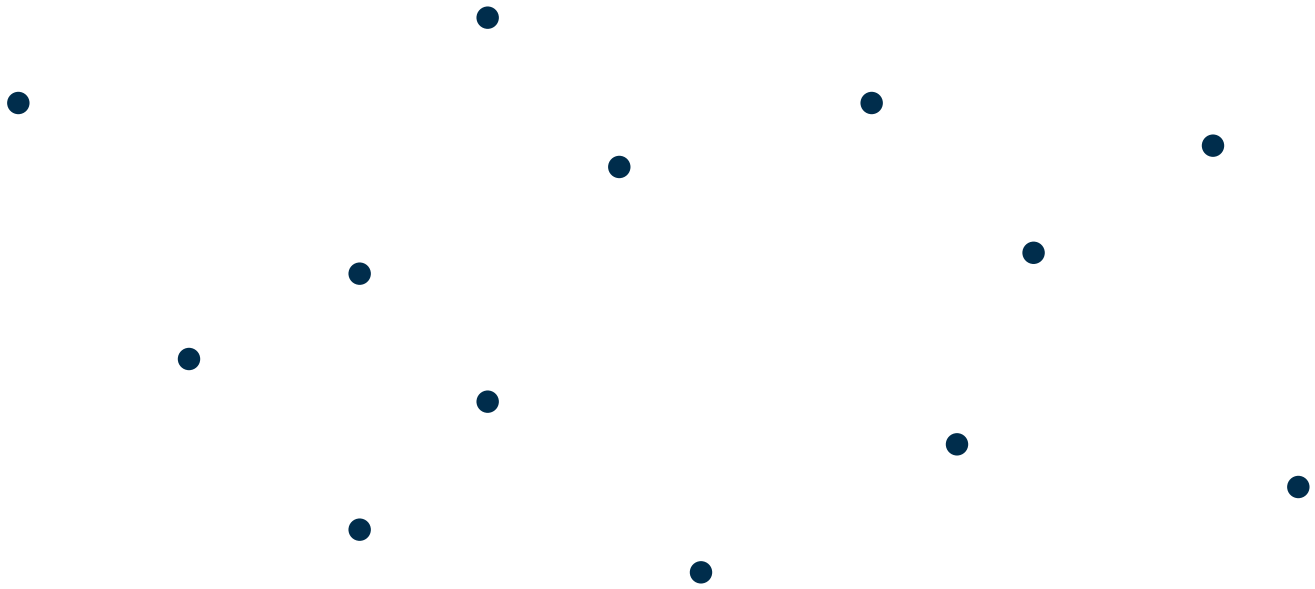
Given: set of points



Maximum Distance

Given: set of points

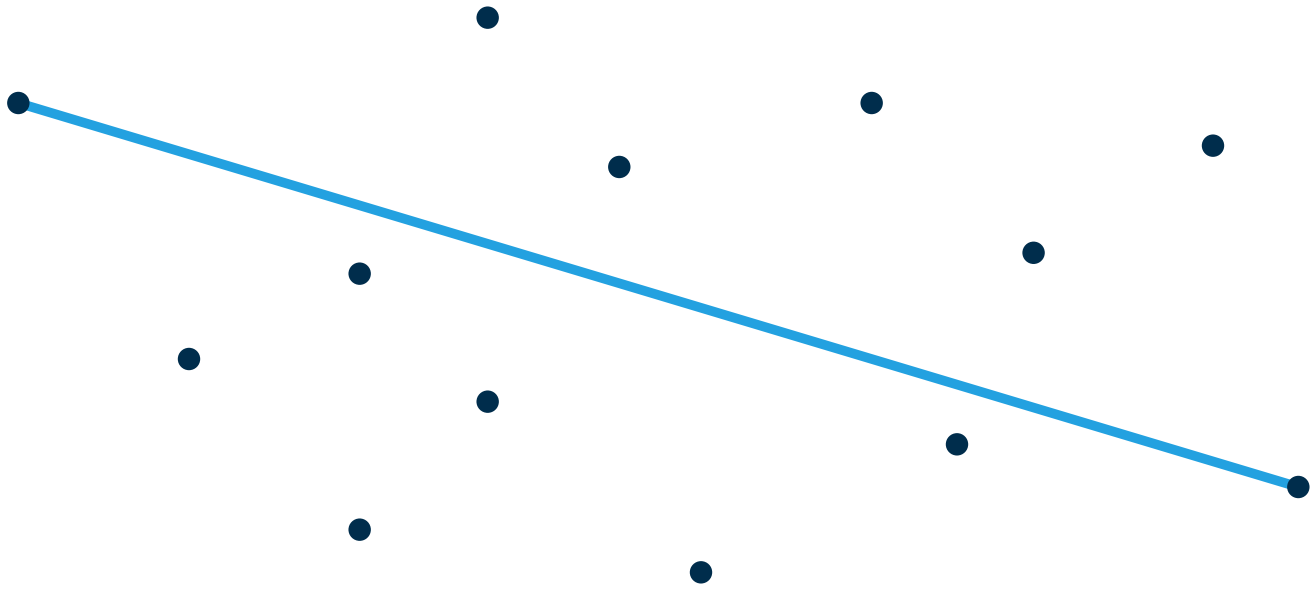
Task: find two points with maximum distance



Maximum Distance

Given: set of points

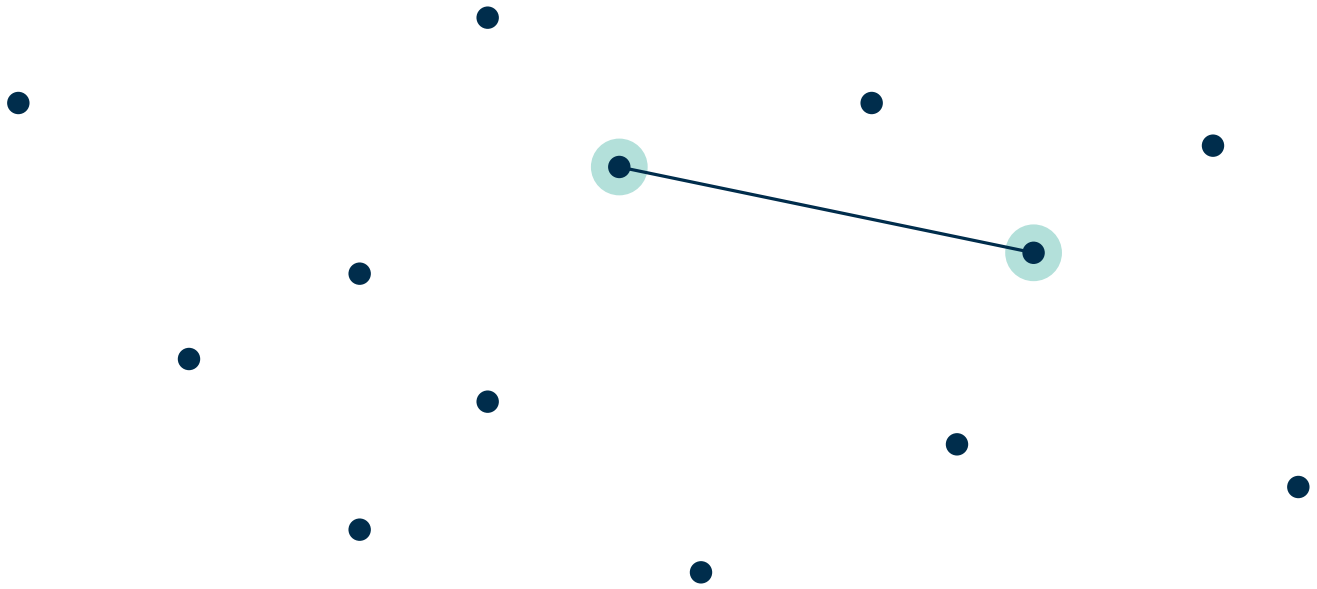
Task: find two points with maximum distance



Maximum Distance

Given: set of points

Task: find two points with maximum distance

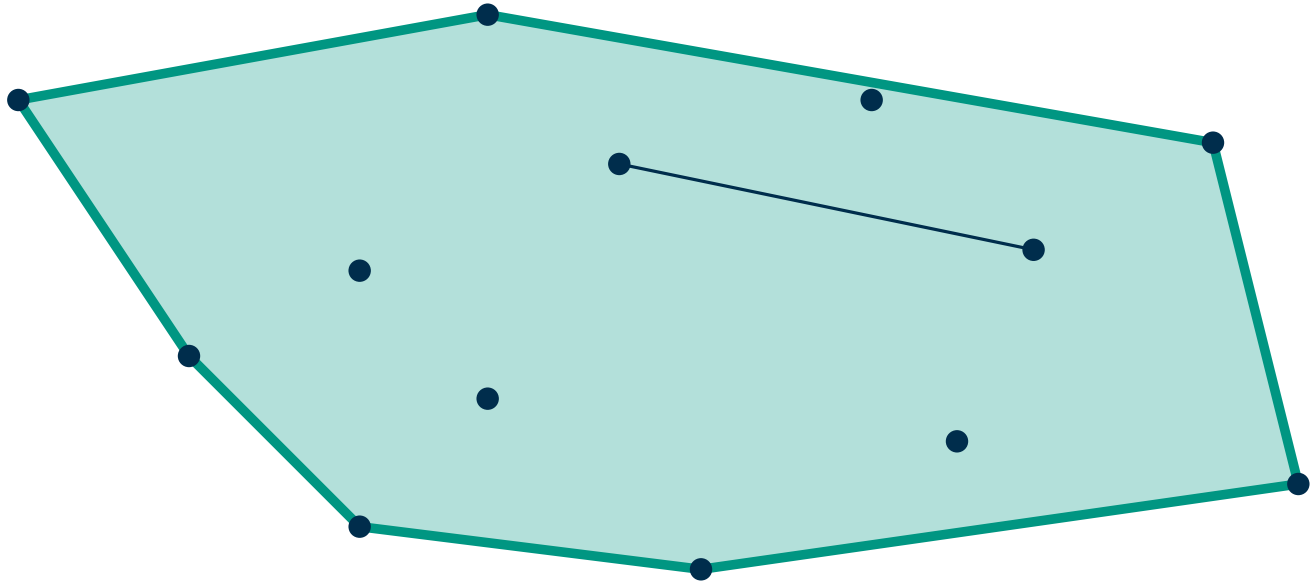


- Which points are candidates?

Maximum Distance

Given: set of points

Task: find two points with maximum distance

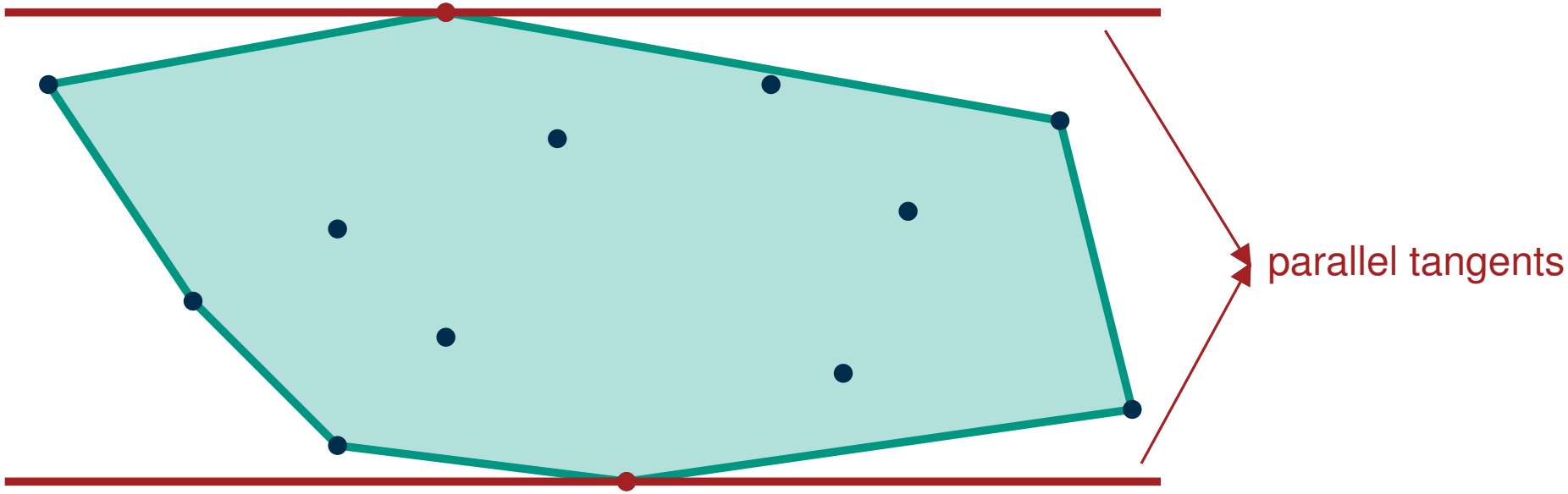


- Which points are candidates?

Maximum Distance

Given: set of points

Task: find two points with maximum distance

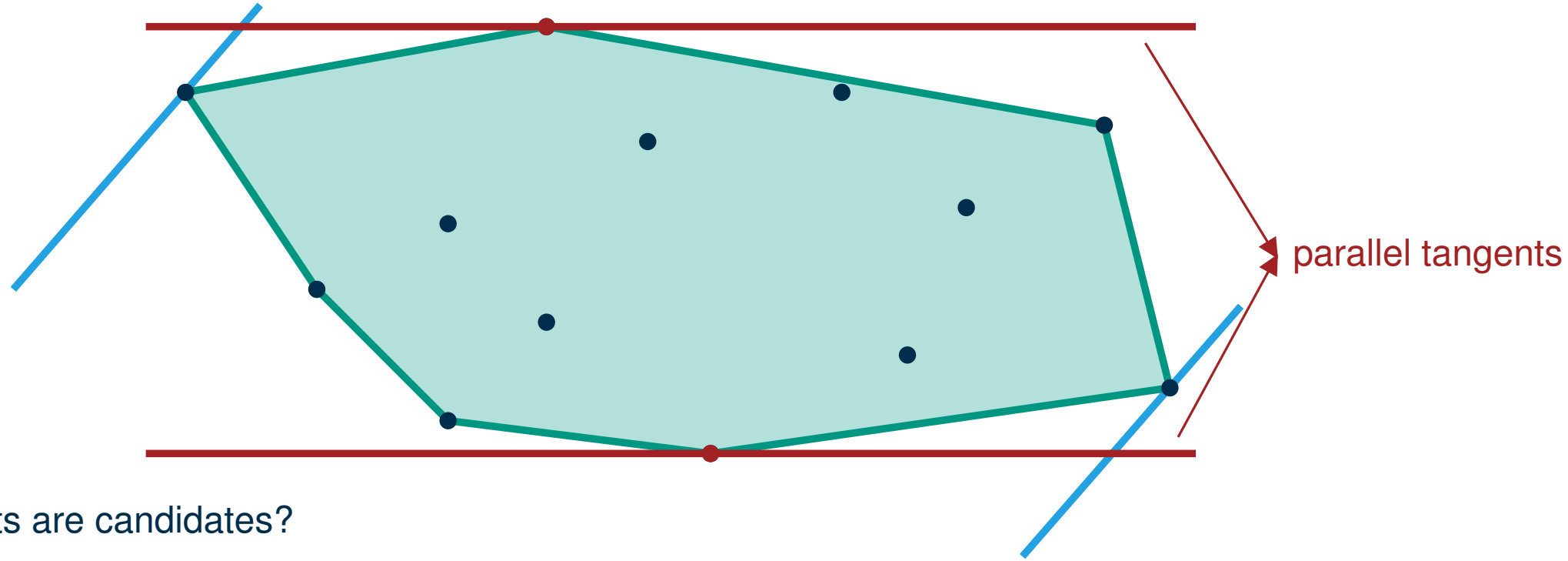


- Which points are candidates?
- Show: points with maximum distance are *antipodal*

Maximum Distance

Given: set of points

Task: find two points with maximum distance

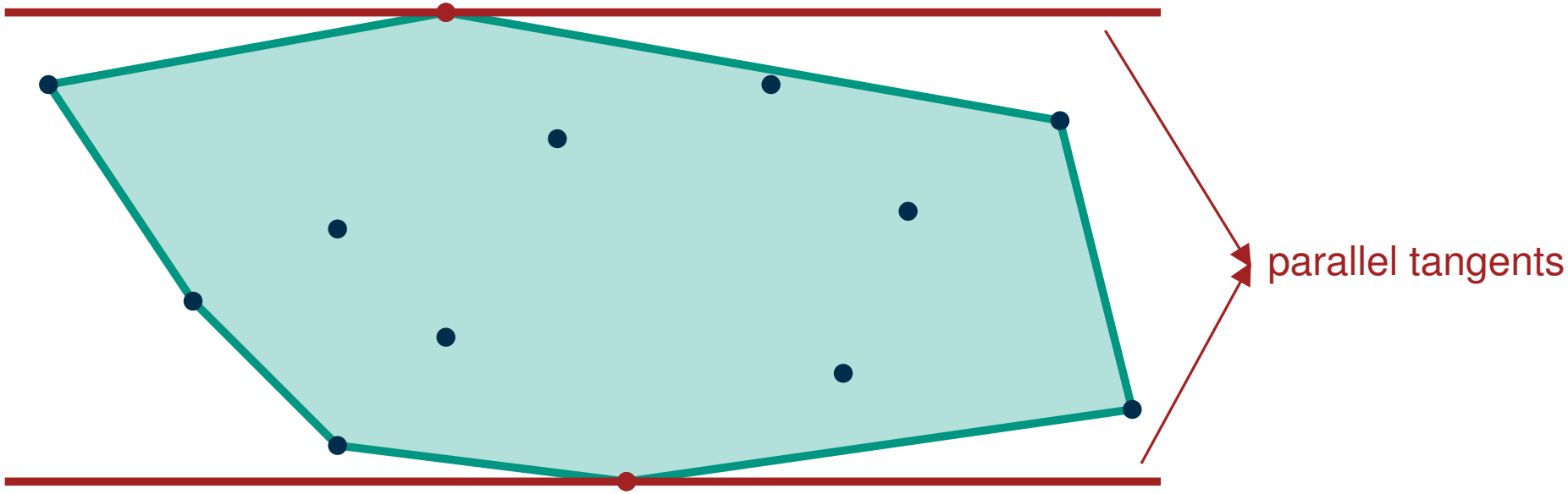


- Which points are candidates?
- Show: points with maximum distance are *antipodal*

Maximum Distance

Given: set of points

Task: find two points with maximum distance

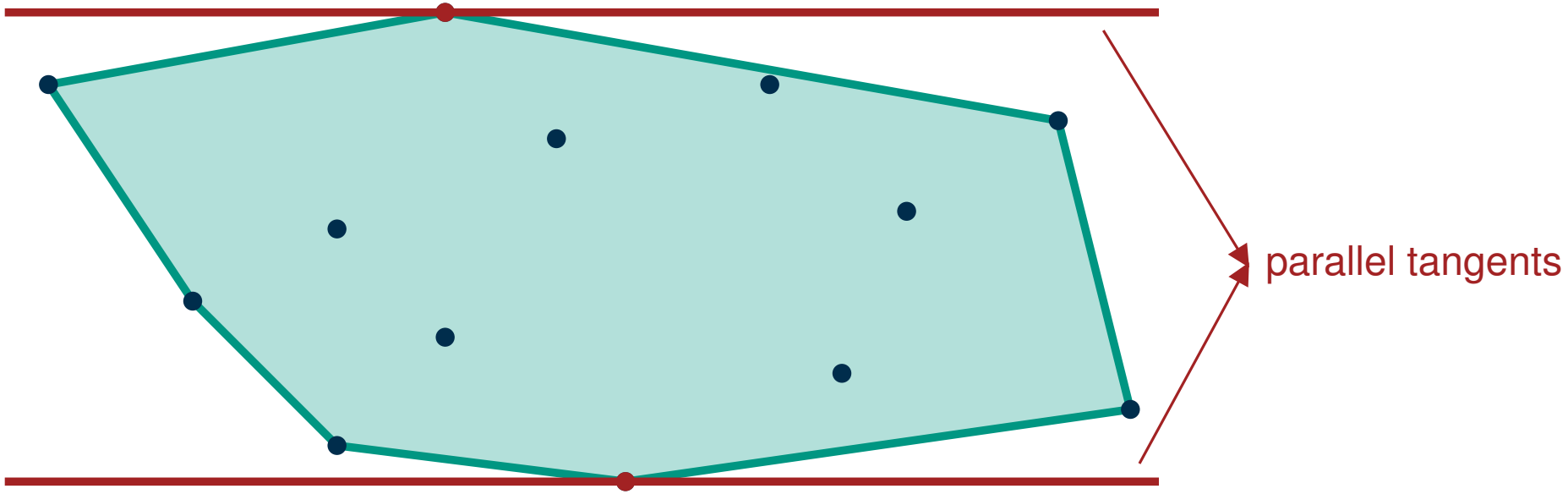


- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?

Maximum Distance

Given: set of points

Task: find two points with maximum distance



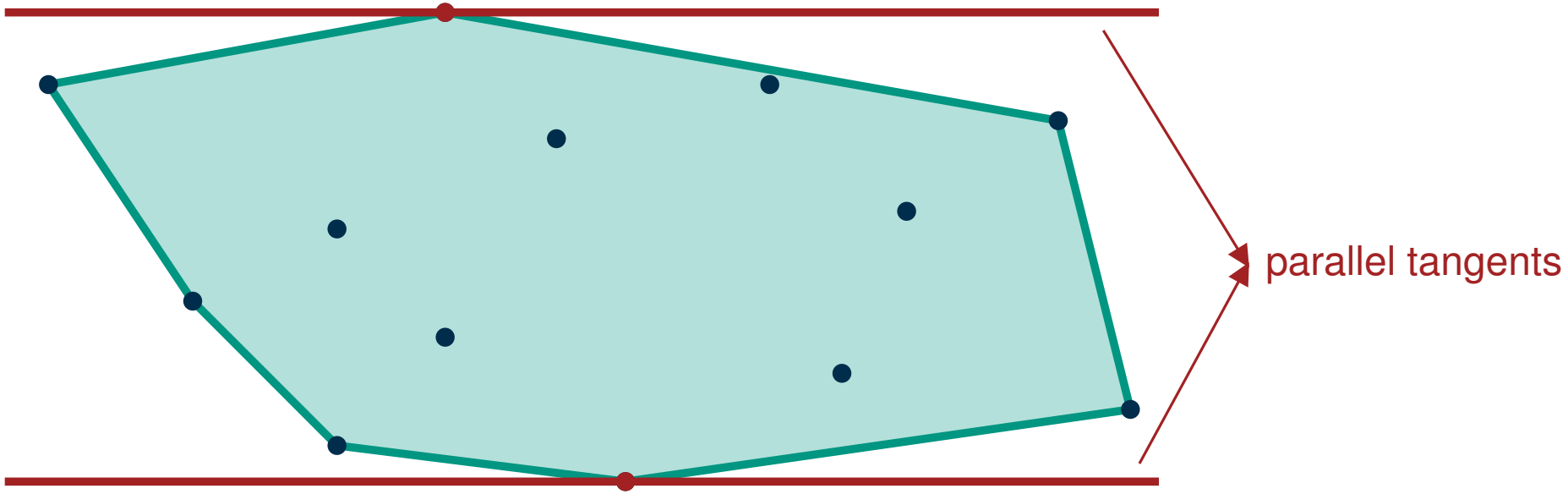
show: not on convex hull $\Rightarrow$ extend line between points

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?

Maximum Distance

Given: set of points

Task: find two points with maximum distance



- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?

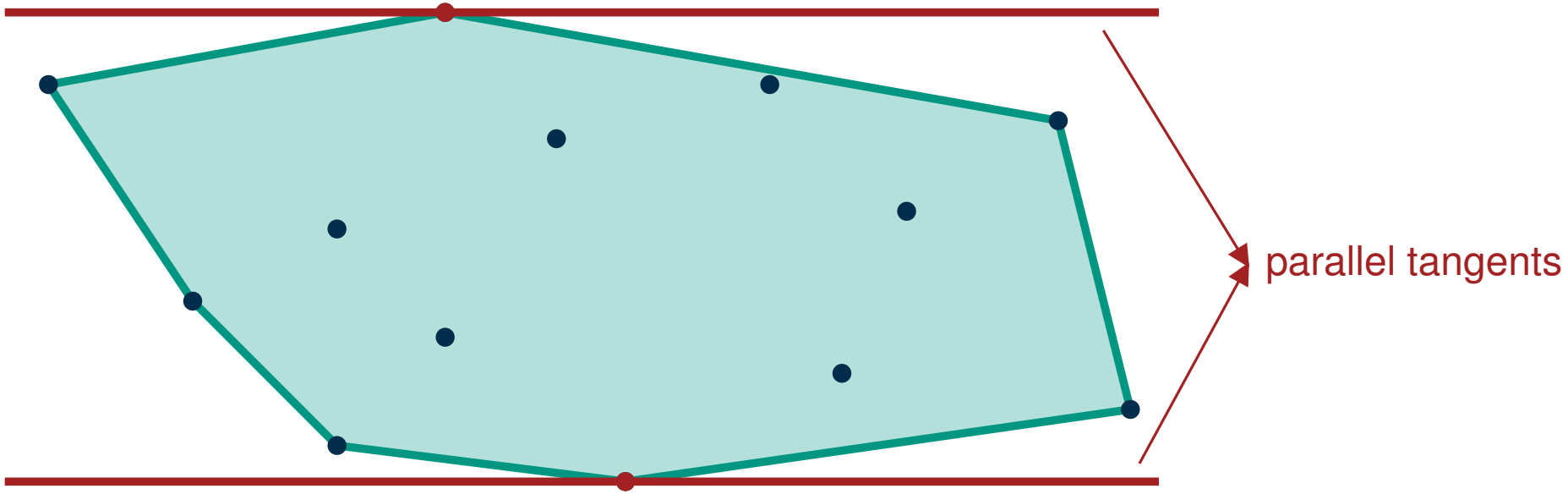
show: not on convex hull $\Rightarrow$ extend line between points

show: not antipodal $\Rightarrow$ extend line between points

Maximum Distance

Given: set of points

Task: find two points with maximum distance



- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?

show: not on convex hull $\Rightarrow$ extend line between points

show: not antipodal $\Rightarrow$ extend line between points

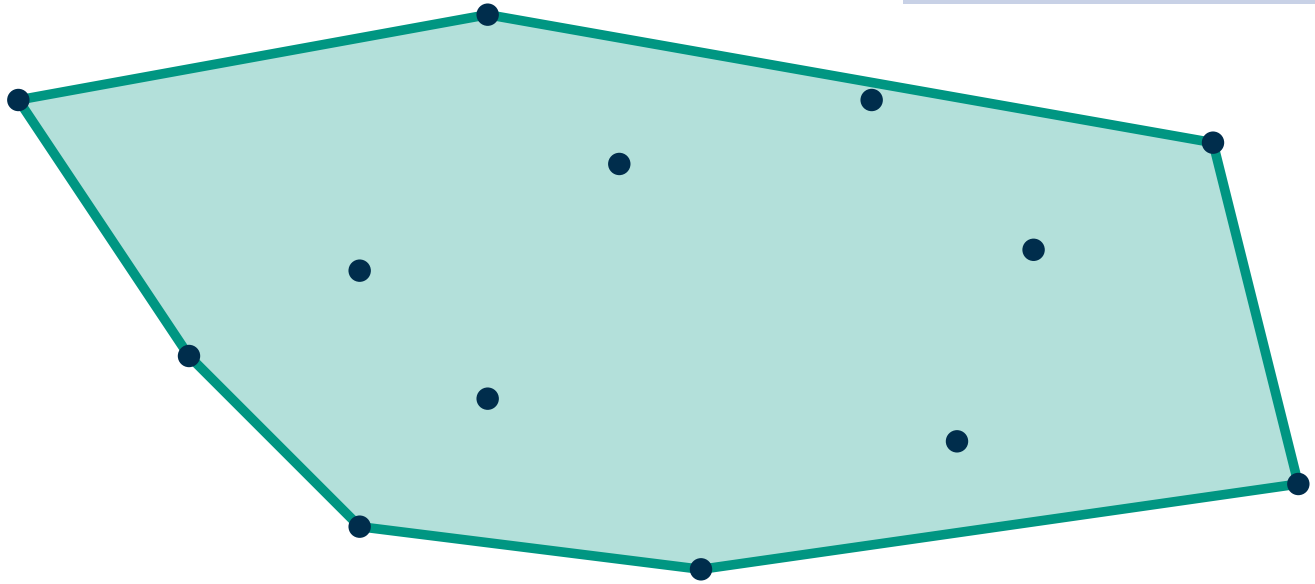
Start with antipodal pair, how do we get the next?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?

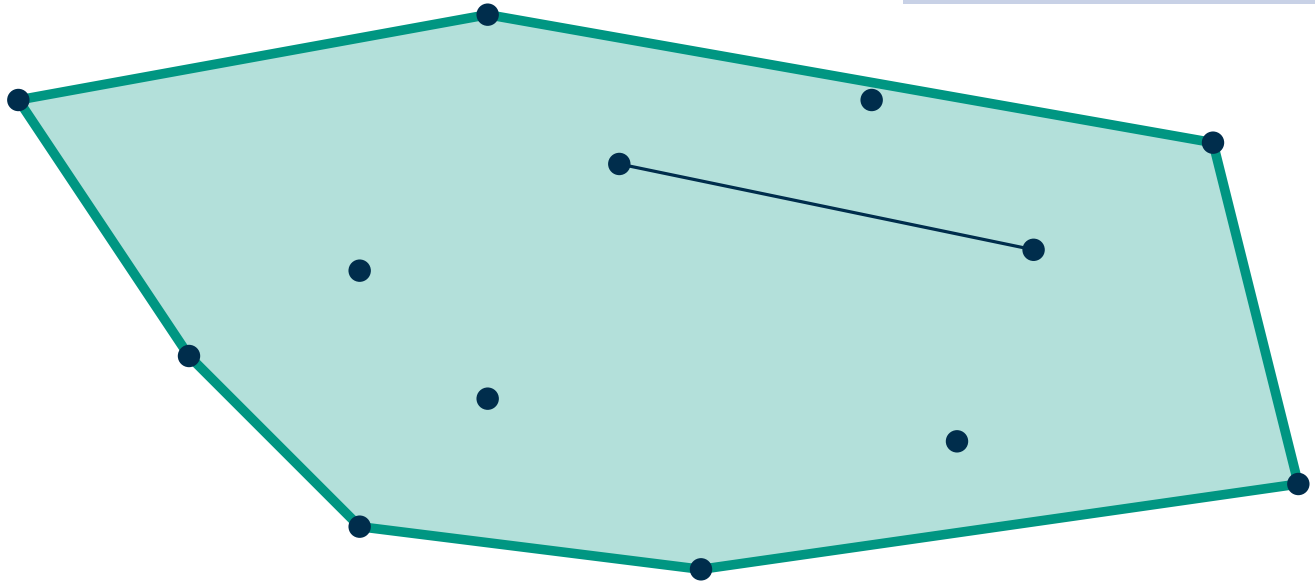


Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?

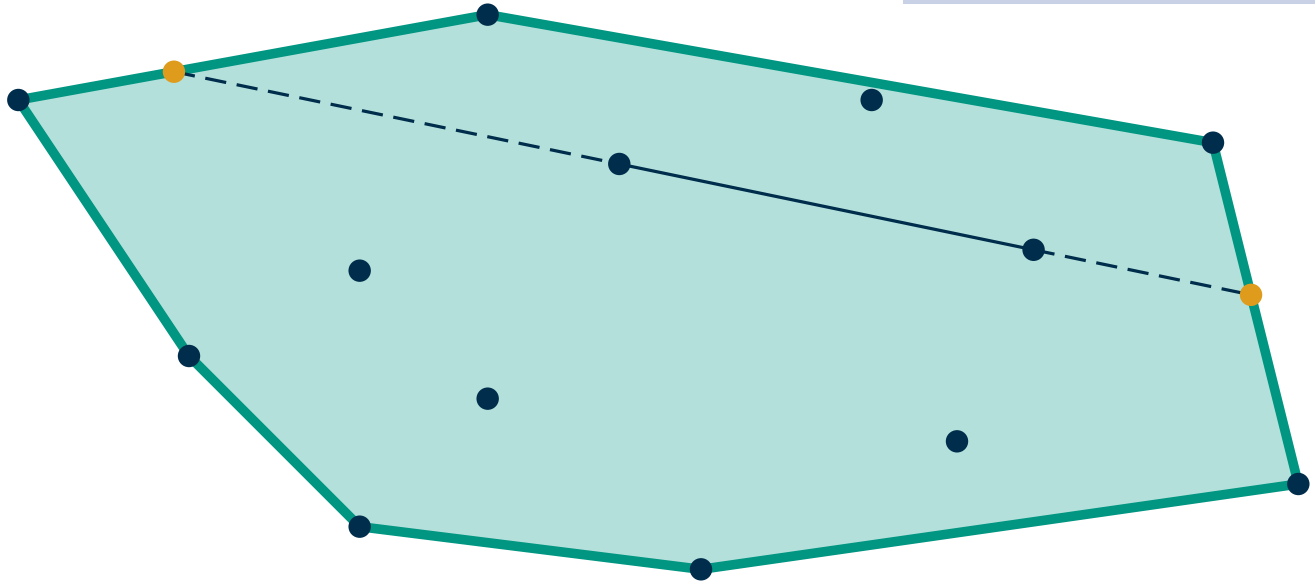


Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



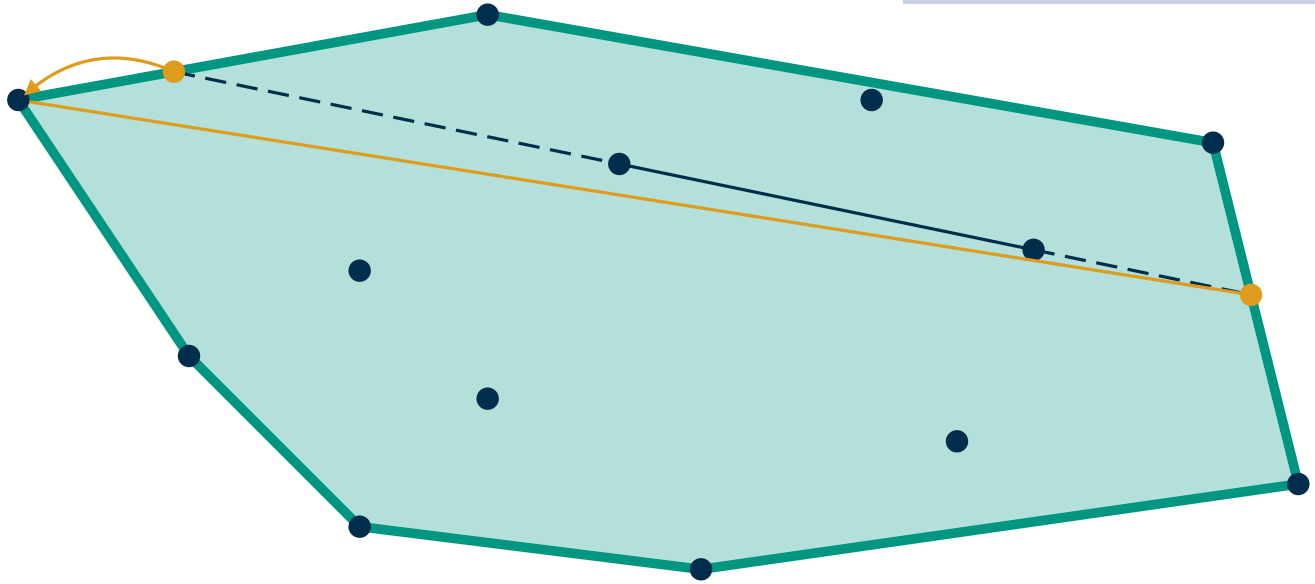
- line intersects convex hull somewhere

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



- line intersects convex hull somewhere $\Rightarrow$ extend line

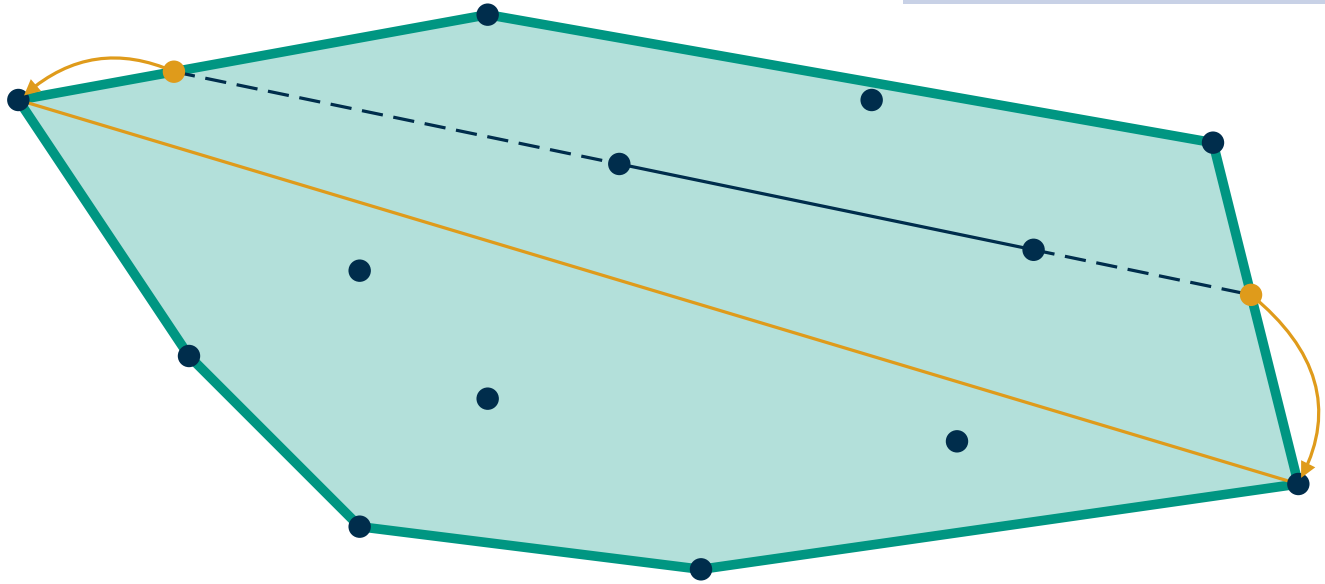
Why?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



- line intersects convex hull somewhere $\Rightarrow$ extend line

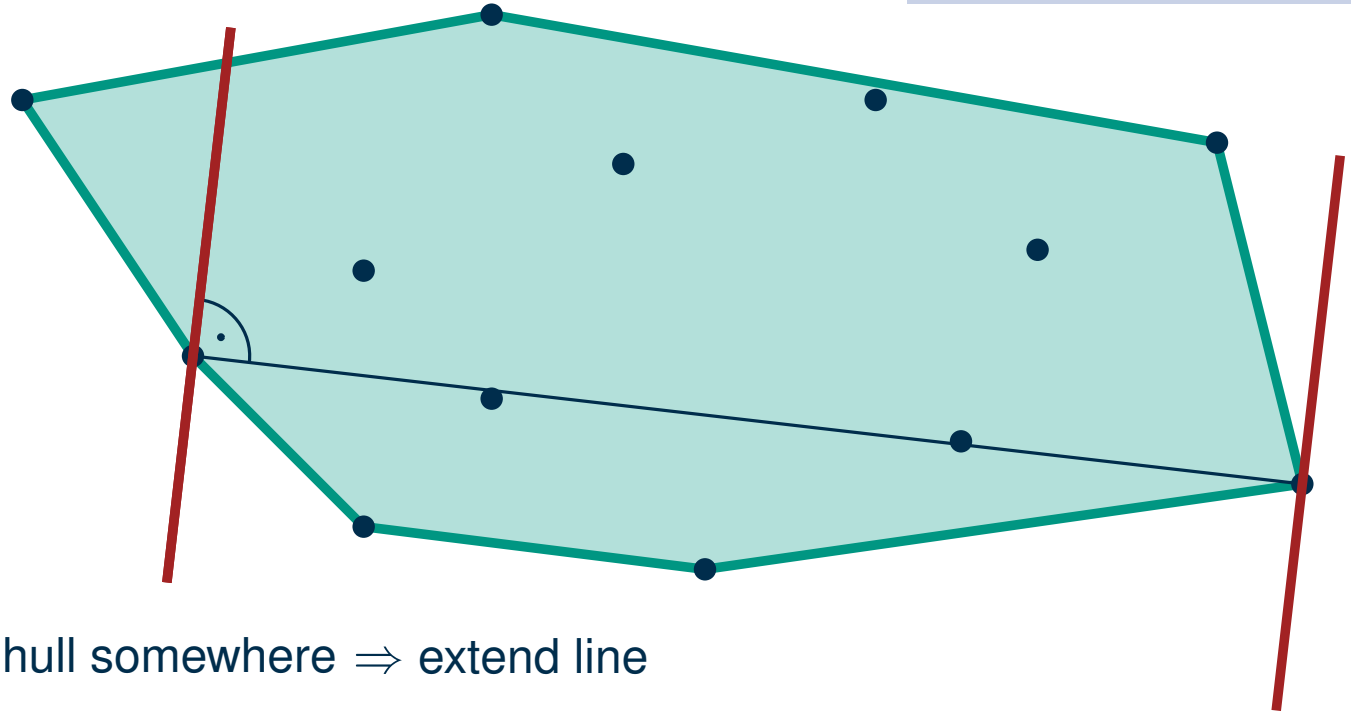
Why?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



- line intersects convex hull somewhere $\Rightarrow$ extend line
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere

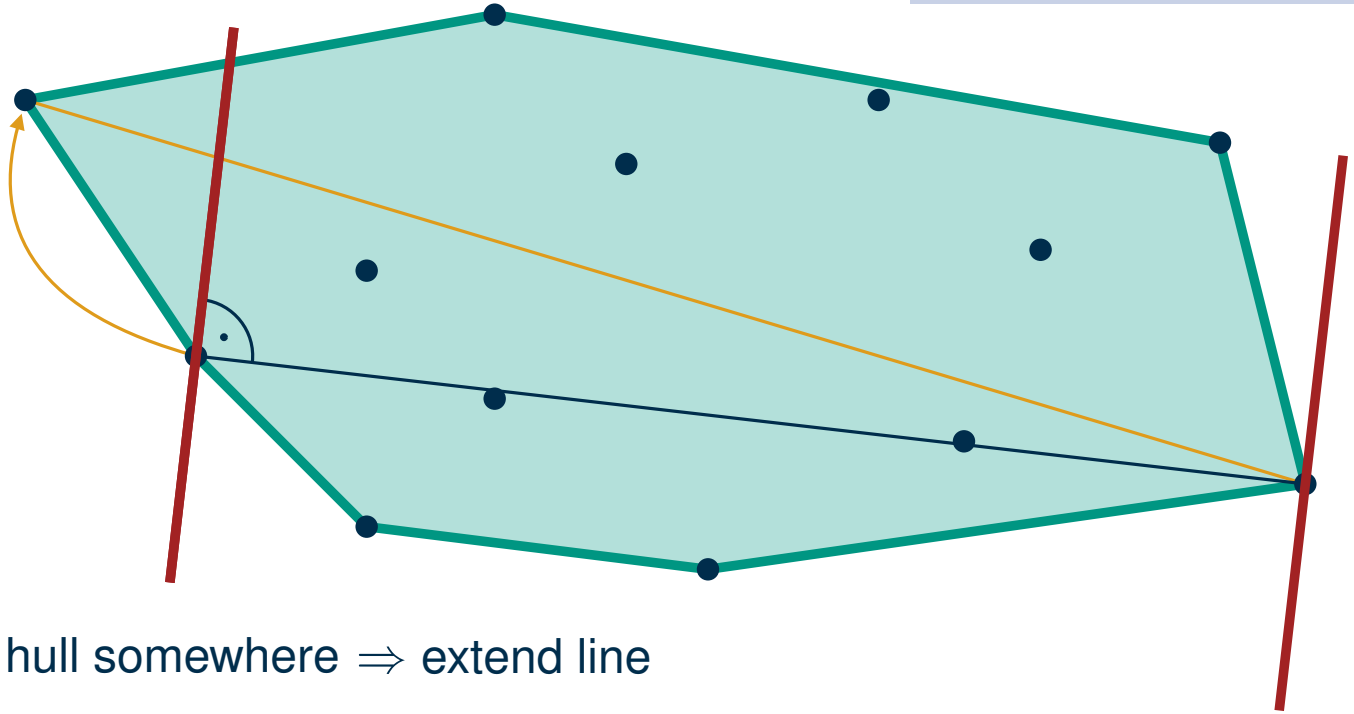
Why?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



- line intersects convex hull somewhere $\Rightarrow$ extend line
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away

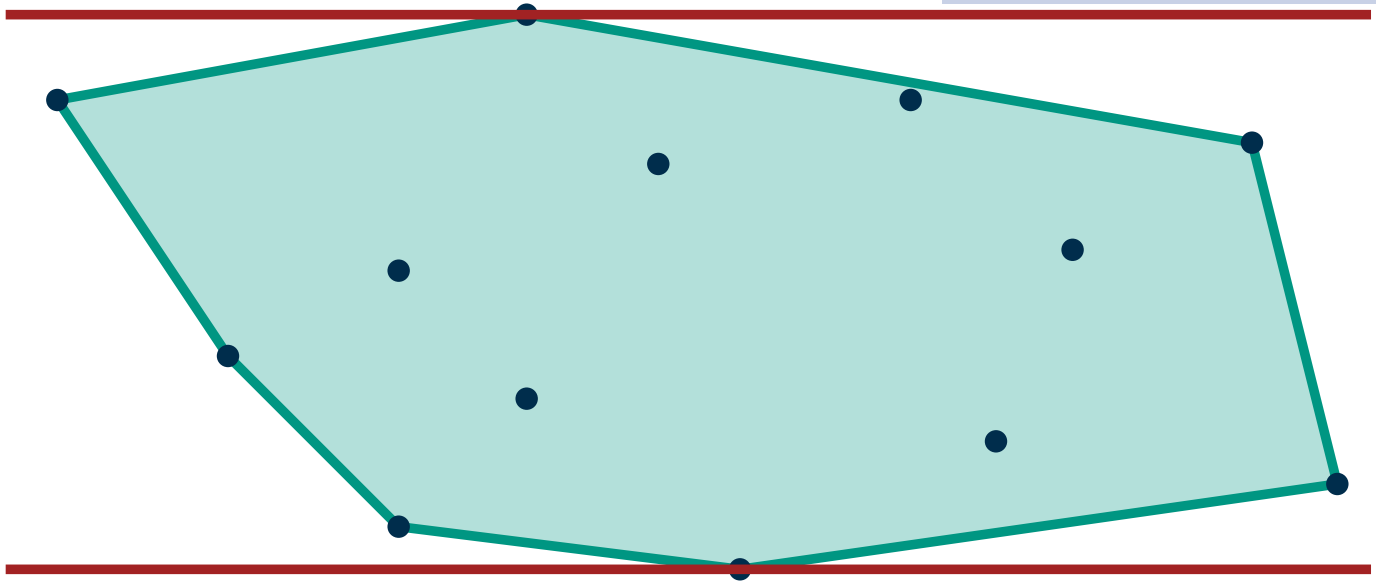
Why?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



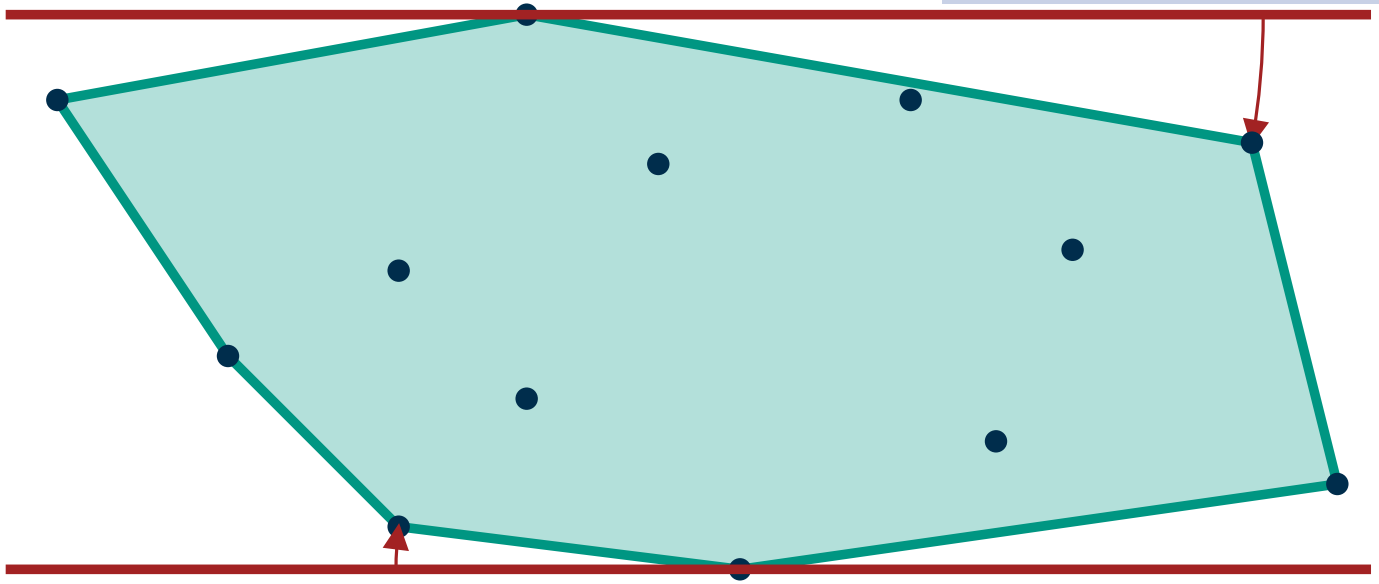
- line intersects convex hull somewhere $\Rightarrow$ extend line — Why?
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon — How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



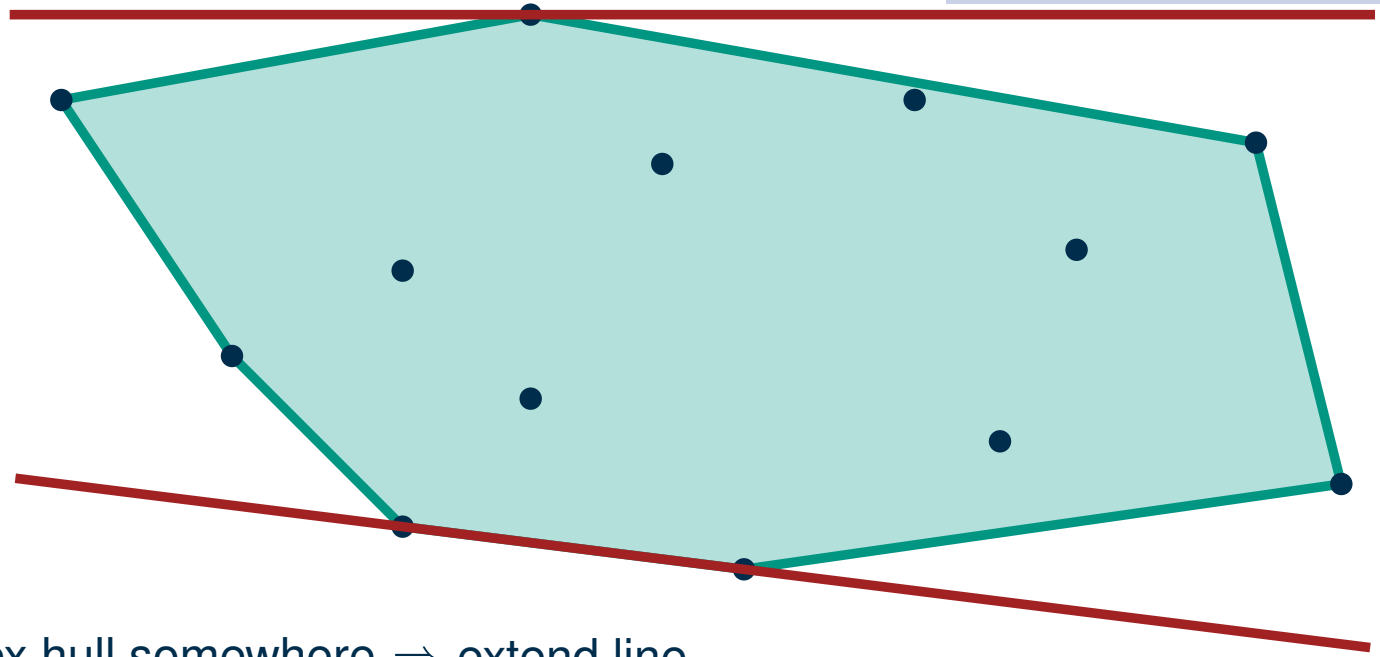
- line intersects convex hull somewhere $\Rightarrow$ extend line — Why?
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon — How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



- line intersects convex hull somewhere $\Rightarrow$ extend line

Why?

- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon

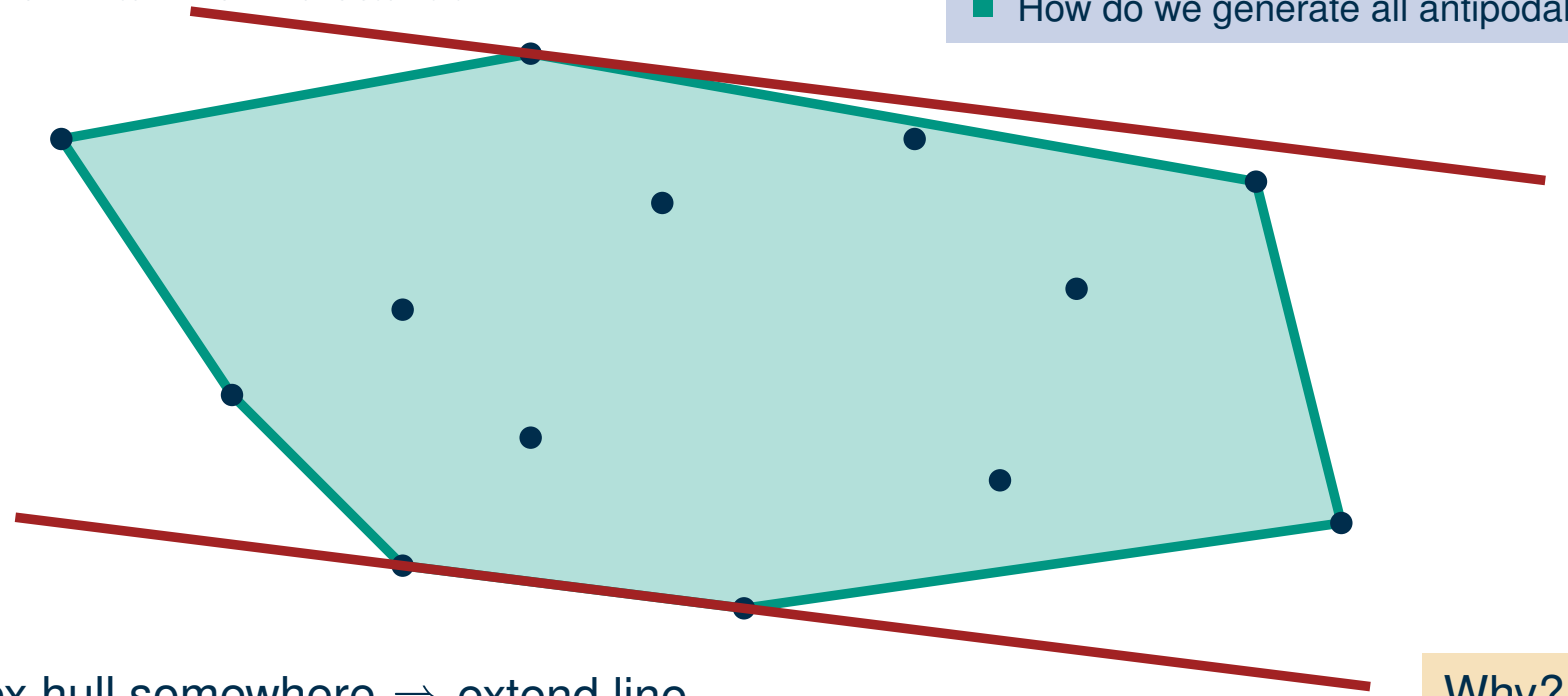
How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



- line intersects convex hull somewhere $\Rightarrow$ extend line

Why?

- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon

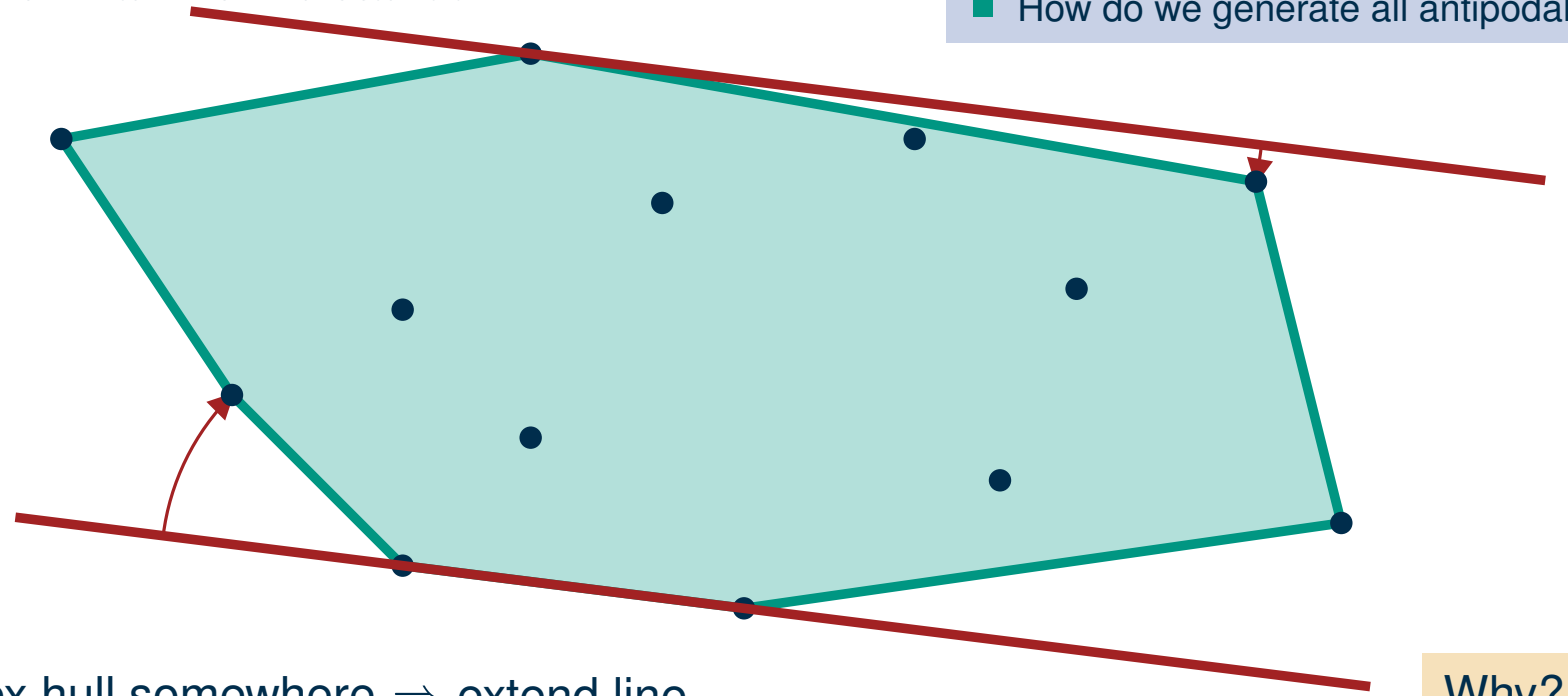
How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



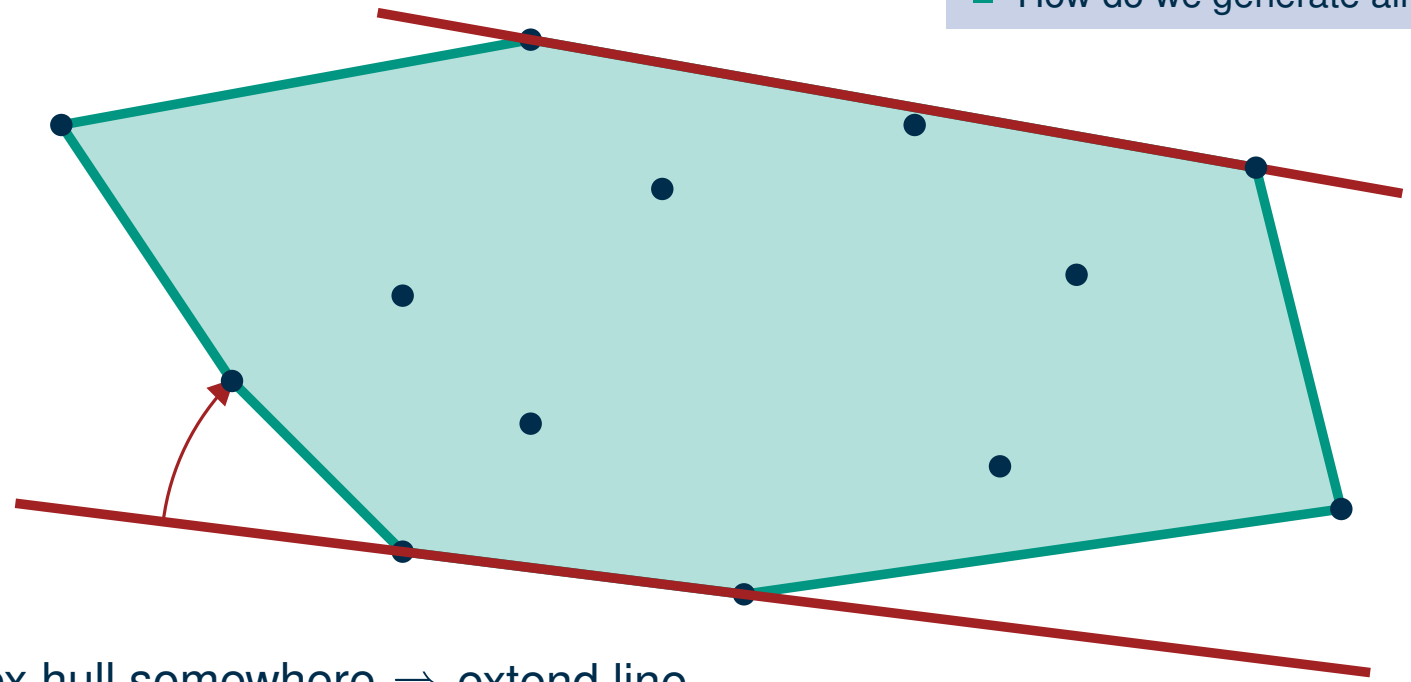
- line intersects convex hull somewhere $\Rightarrow$ extend line — Why?
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon — How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



- line intersects convex hull somewhere $\Rightarrow$ extend line

Why?

- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon

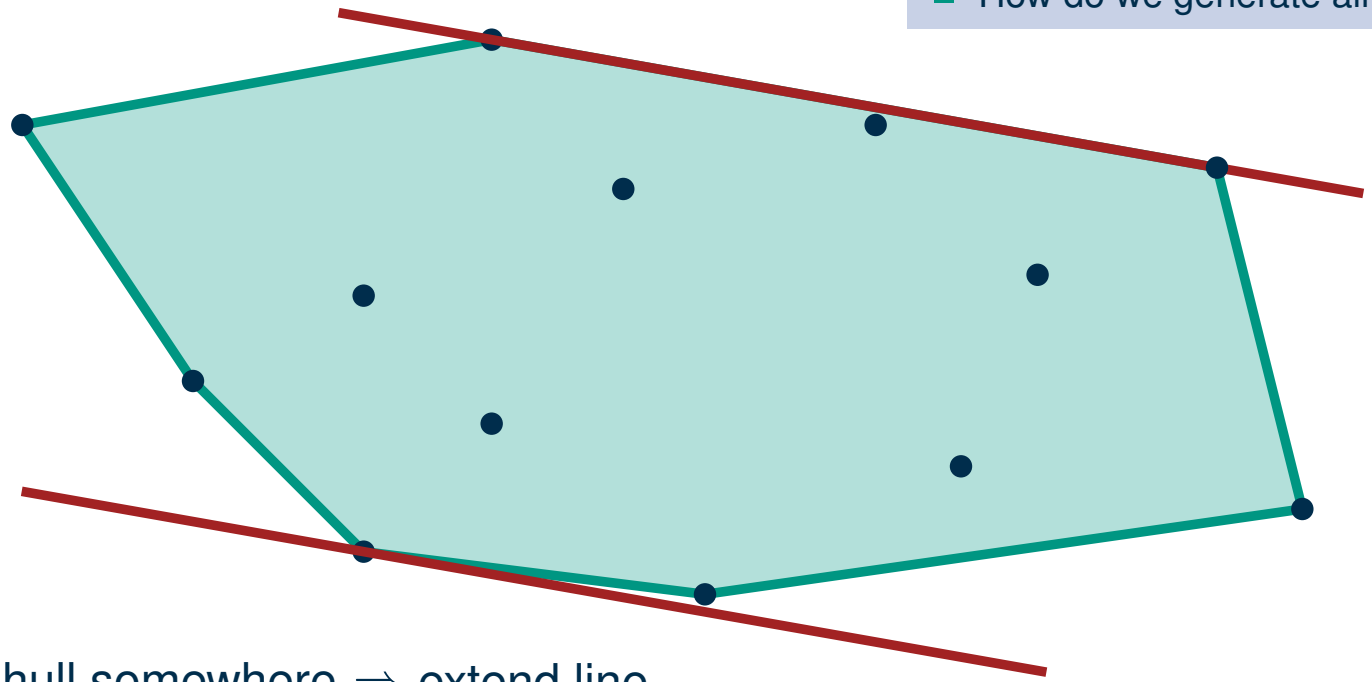
How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



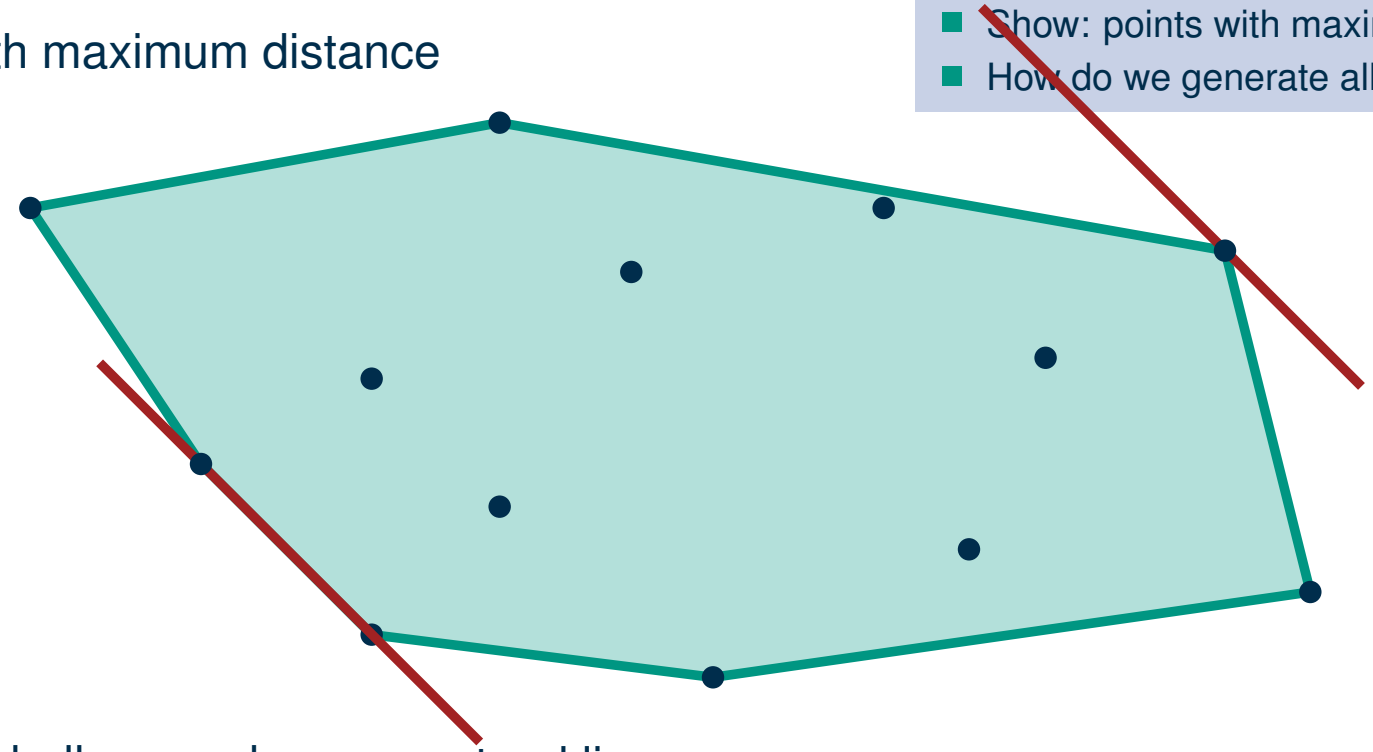
- line intersects convex hull somewhere $\Rightarrow$ extend line — Why?
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon — How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



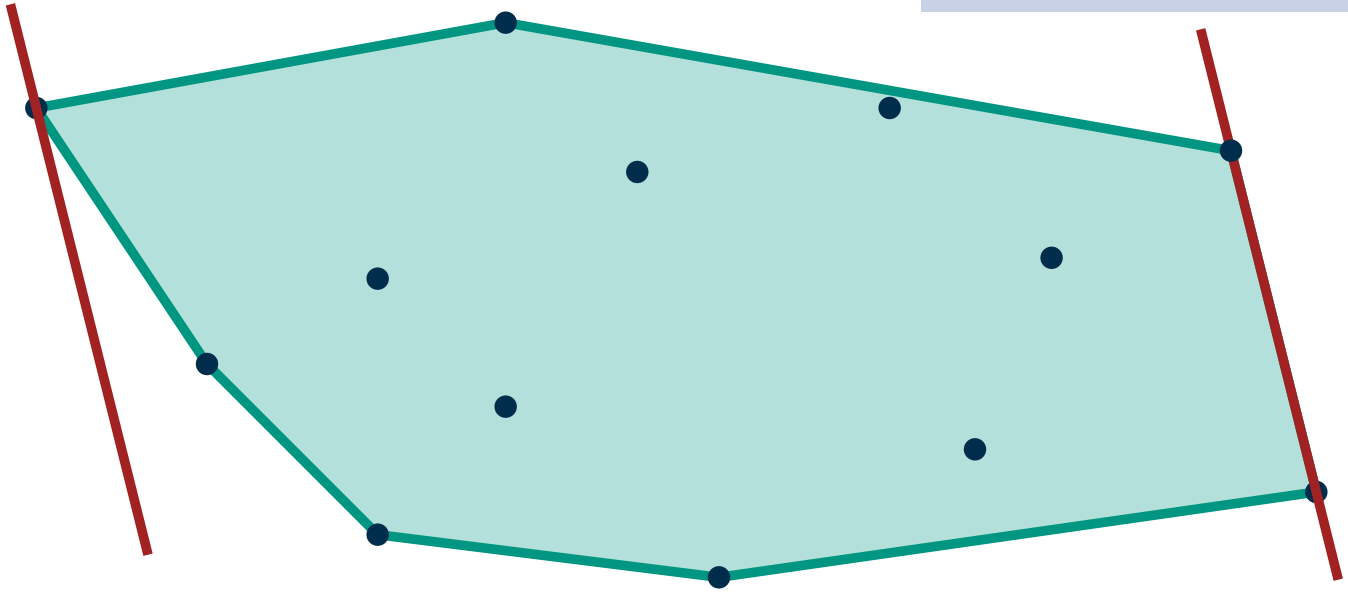
- line intersects convex hull somewhere $\Rightarrow$ extend line — Why?
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon — How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



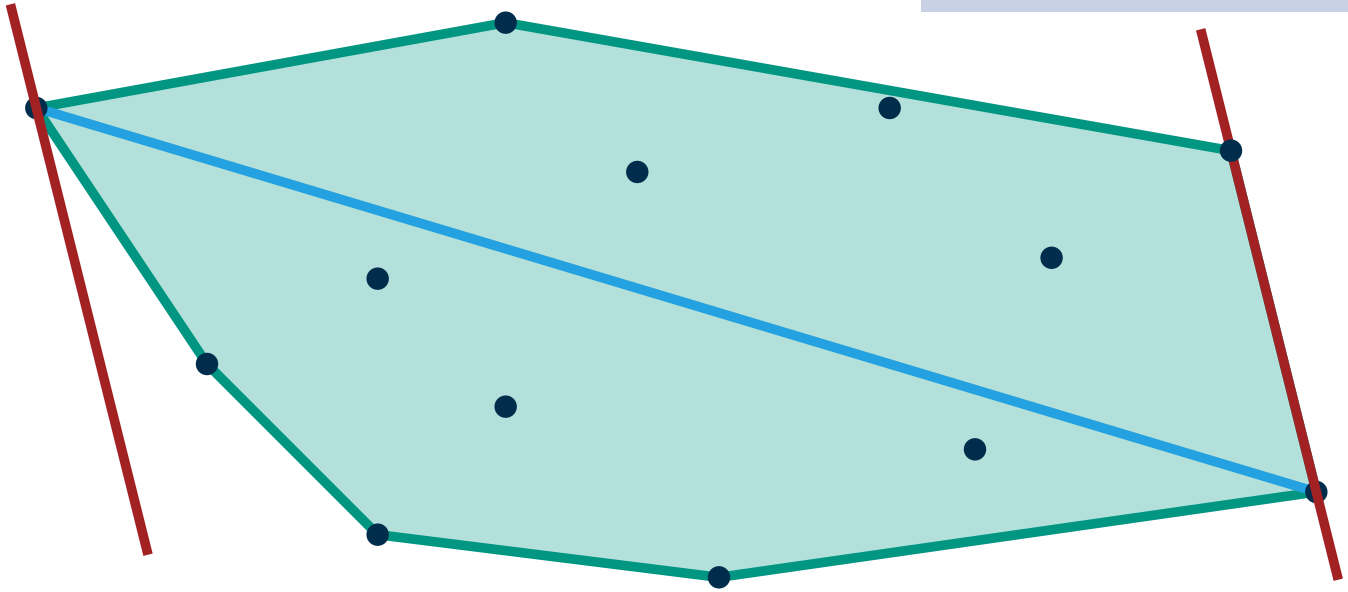
- line intersects convex hull somewhere $\Rightarrow$ extend line — Why?
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon — How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?



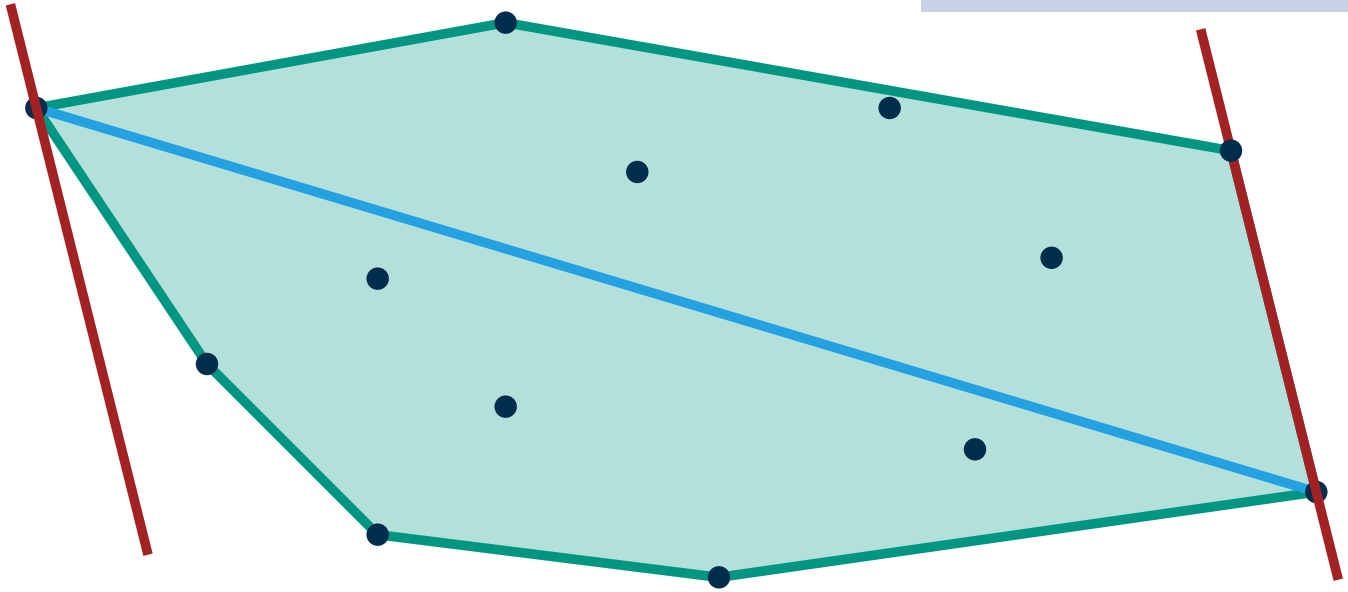
- line intersects convex hull somewhere $\Rightarrow$ extend line — Why?
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon — How?

Maximum Distance – Solution Sketch

Given: set of points

Task: find two points with maximum distance

- Which points are candidates?
- Show: points with maximum distance are *antipodal*
- How do we generate all antipodal points in $O(n)$?





"rotating calipers"

- line intersects convex hull somewhere $\Rightarrow$ extend line — Why?
- not antipodal $\Rightarrow$ orthogonal lines intersect polygon somewhere $\Rightarrow$ point that is further away
- rotate tangents around polygon — How?