

Algorithmische Geometrie

Übung 1



Orga

- Namensrunde

Orga

- Namensrunde

Übung

- Übung: alle zwei Wochen (zwischen Ausgabe und Abgabe von ÜBs)

Orga

- Namensrunde

Übung

- Übung: alle zwei Wochen (zwischen Ausgabe und Abgabe von ÜBs)
- Inhalte: altes ÜB besprechen, Tipps für neues ÜB, anderes

Orga

- Namensrunde

Übung

- Übung: alle zwei Wochen (zwischen Ausgabe und Abgabe von ÜBs)
- Inhalte: altes ÜB besprechen, Tipps für neues ÜB, anderes

Übungsblätter

Orga

- Namensrunde

Übung

- Übung: alle zwei Wochen (zwischen Ausgabe und Abgabe von ÜBs)
- Inhalte: altes ÜB besprechen, Tipps für neues ÜB, anderes

Übungsblätter

- alle zwei Wochen, Abgabe bei algogeom_abgaben@lists.kit.edu

Orga

- Namensrunde

Übung

- Übung: alle zwei Wochen (zwischen Ausgabe und Abgabe von ÜBs)
- Inhalte: altes ÜB besprechen, Tipps für neues ÜB, anderes

Übungsblätter

- alle zwei Wochen, Abgabe bei algogeom_abgaben@lists.kit.edu
- gerne in Zweiergruppen!

Orga

- Namensrunde

Übung

- Übung: alle zwei Wochen (zwischen Ausgabe und Abgabe von ÜBs)
- Inhalte: altes ÜB besprechen, Tipps für neues ÜB, anderes

Übungsblätter

- alle zwei Wochen, Abgabe bei algogeom_abgaben@lists.kit.edu
- gerne in Zweiergruppen!
- Voraussetzung für mündliche Prüfung: regelmäßige Abgabe

ÜB: Allgemeine Tipps

- Korrektheit + Laufzeit zeigen

ÜB: Allgemeine Tipps

- Korrektheit + Laufzeit zeigen
- Sonderfälle betrachten

ÜB: Allgemeine Tipps

- Korrektheit + Laufzeit zeigen
- Sonderfälle betrachten

Abstraktionsniveau

- Idee soll verständlich sein

ÜB: Allgemeine Tipps

- Korrektheit + Laufzeit zeigen
- Sonderfälle betrachten

Abstraktionsniveau

- Idee soll verständlich sein
- Erklärung > Formeln

ÜB: Allgemeine Tipps

- Korrektheit + Laufzeit zeigen
- Sonderfälle betrachten

Abstraktionsniveau

- Idee soll verständlich sein
- Erklärung > Formeln
- Details abstrahieren, wenn klar ist, dass sie funktionieren

ÜB: Allgemeine Tipps

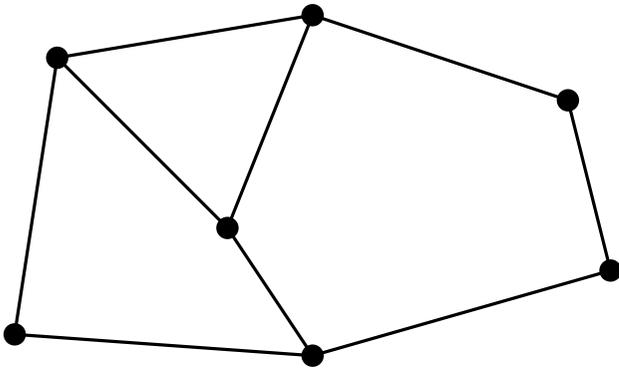
- Korrektheit + Laufzeit zeigen
- Sonderfälle betrachten

Abstraktionsniveau

- Idee soll verständlich sein
- Erklärung > Formeln
- Details abstrahieren, wenn klar ist, dass sie funktionieren
- Skizzen sind hilfreich

Doppelt-verkettete Kantenliste

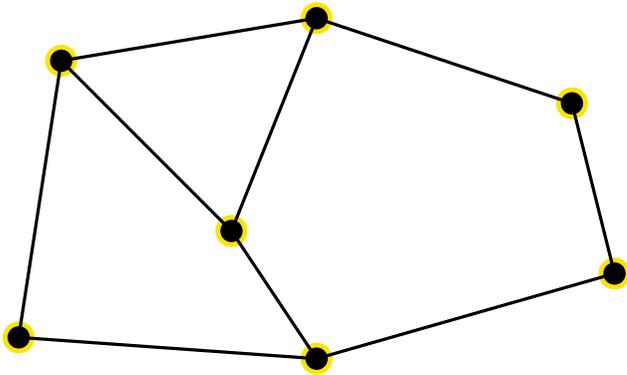
Bestandteile eines geometrischen Graphen



Doppelt-verkettete Kantenliste

Bestandteile eines geometrischen Graphen

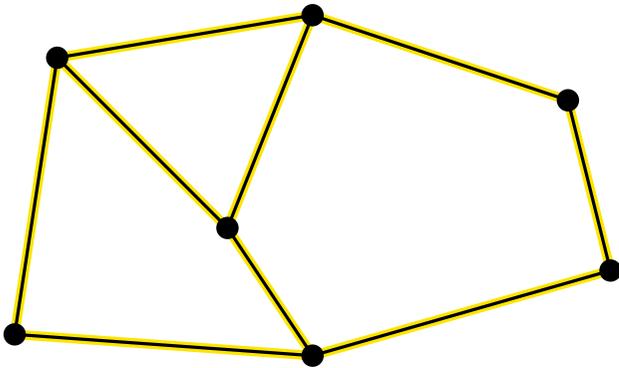
- Knoten mit Koordinaten



Doppelt-verkettete Kantenliste

Bestandteile eines geometrischen Graphen

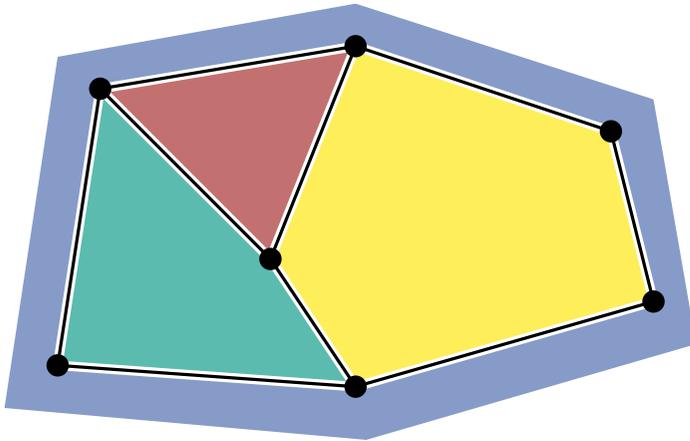
- Knoten mit Koordinaten
- Kanten



Doppelt-verkettete Kantenliste

Bestandteile eines geometrischen Graphen

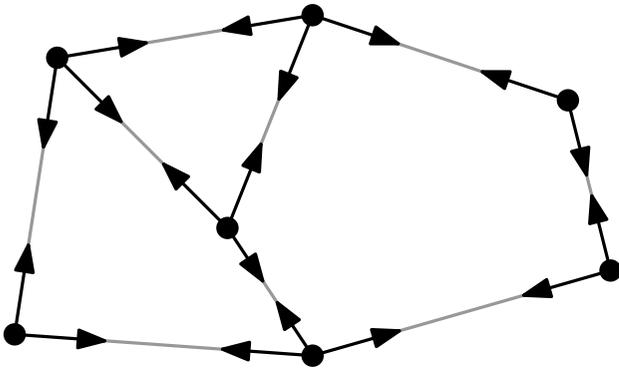
- Knoten mit Koordinaten
- Kanten
- Facetten



Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



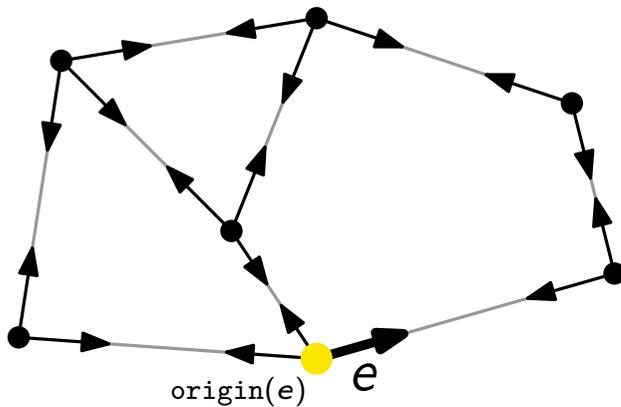
Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt

Für jede „Halbkante“ e

- zugehöriger Knoten: `origin(e)`



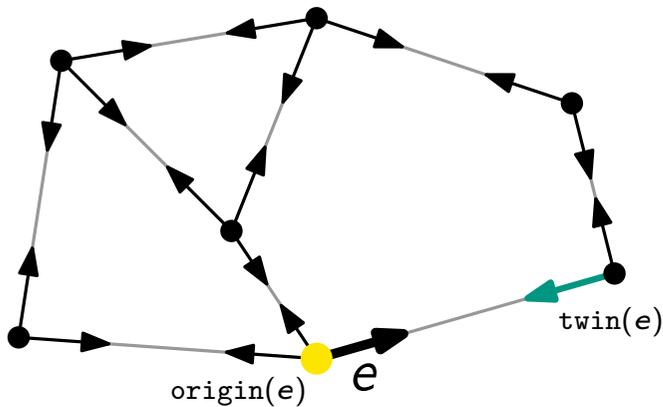
Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt

Für jede „Halbkante“ e

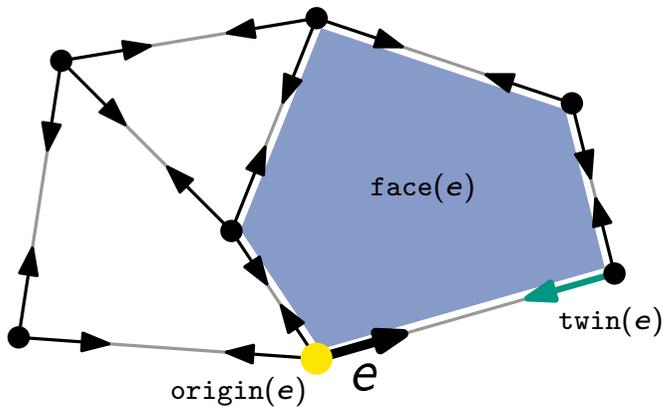
- zugehöriger Knoten: `origin(e)`
- Kante am anderen Endpunkt: `twin(e)`



Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



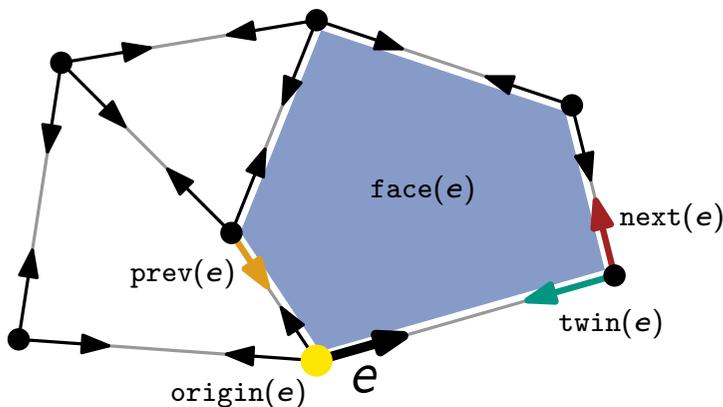
Für jede „Halbkante“ e

- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$

Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



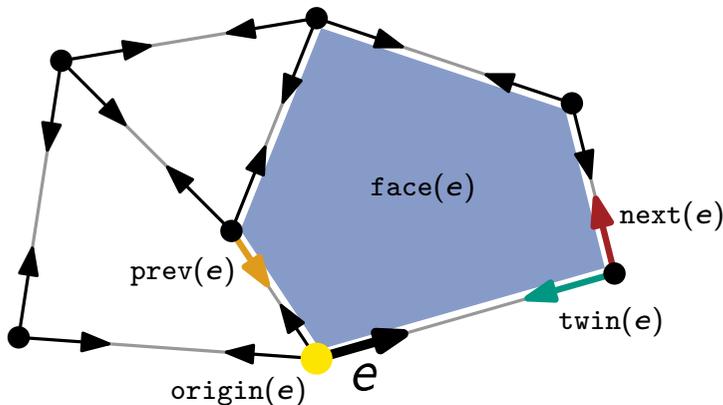
Für jede „Halbkante“ e

- zugehöriger Knoten: $\text{origin}(e)$
- Kante am anderen Endpunkt: $\text{twin}(e)$
- inzidente Facette (links): $\text{face}(e)$
- nächste/vorherige Kante dieser Facette: $\text{next}(e)$, $\text{prev}(e)$

Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



Für jede „Halbkante“ e

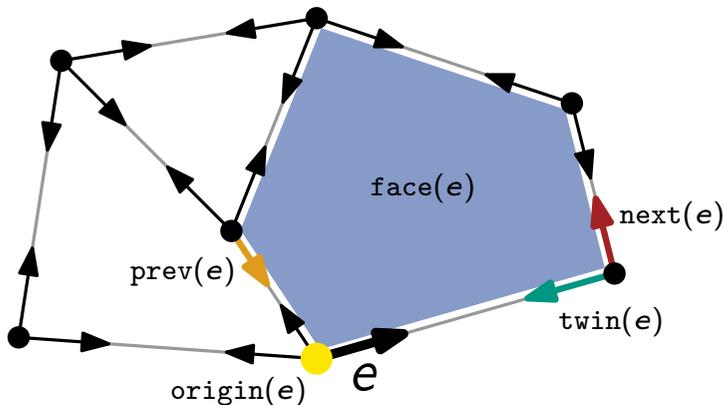
- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$
- nächste/vorherige Kante dieser Facette: $next(e), prev(e)$

- für Knoten v / Facette f eine angrenzende Kante $edge(v)$ / $edge(f)$

Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



Für jede „Halbkante“ e

- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$
- nächste/vorherige Kante dieser Facette: $next(e)$, $prev(e)$

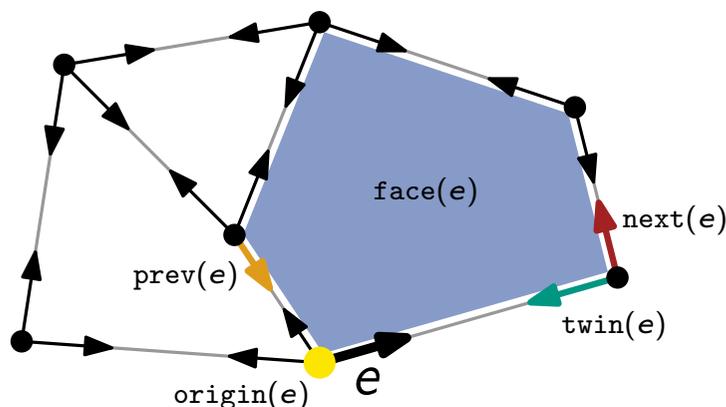
- für Knoten v / Facette f eine angrenzende Kante $edge(v)$ / $edge(f)$

Wie findet man alle inzidenten Kanten von v ?

Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



Für jede „Halbkante“ e

- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$
- nächste/vorherige Kante dieser Facette: $next(e)$, $prev(e)$

- für Knoten v / Facette f eine angrenzende Kante $edge(v)$ / $edge(f)$

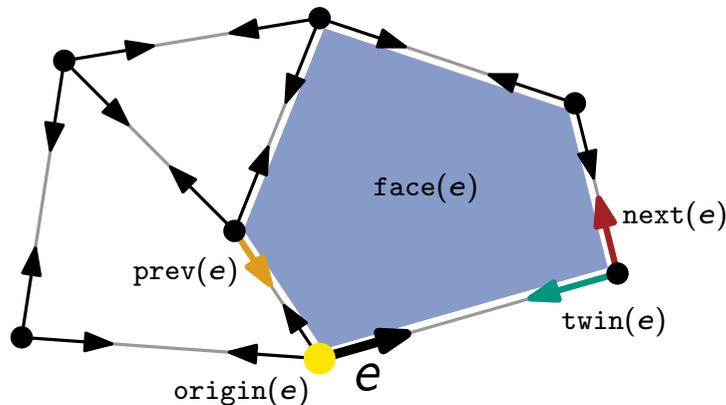
Wie findet man alle inzidenten Kanten von v ?

- im Uhrzeigersinn nächste Kante nach e um v :

Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



Für jede „Halbkante“ e

- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$
- nächste/vorherige Kante dieser Facette: $next(e)$, $prev(e)$

- für Knoten v / Facette f eine angrenzende Kante $edge(v)$ / $edge(f)$

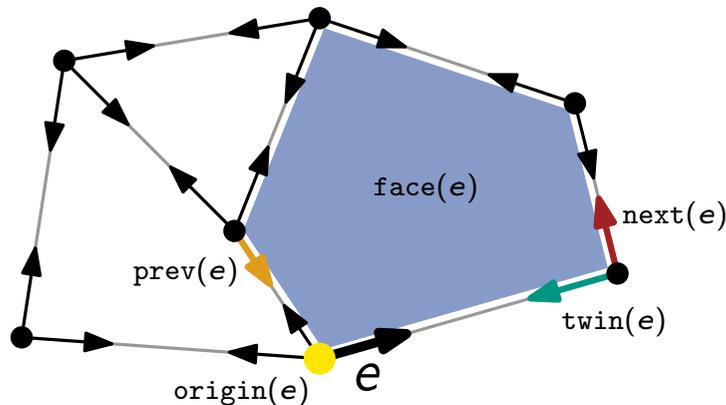
Wie findet man alle inzidenten Kanten von v ?

- im Uhrzeigersinn nächste Kante nach e um v : $next(twin(e))$

Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



Für jede „Halbkante“ e

- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$
- nächste/vorherige Kante dieser Facette: $next(e)$, $prev(e)$

- für Knoten v / Facette f eine angrenzende Kante $edge(v)$ / $edge(f)$

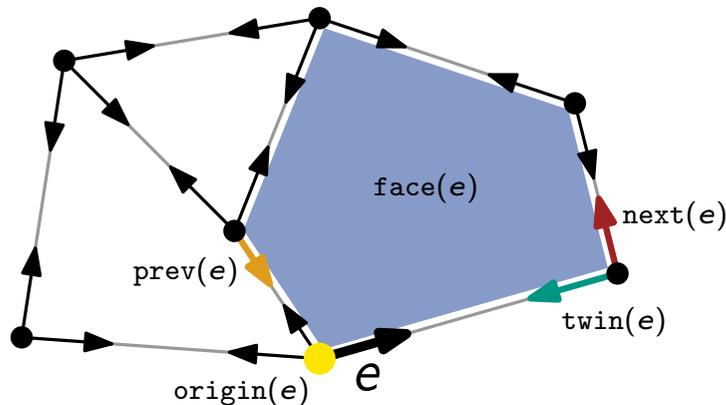
Wie findet man alle inzidenten Kanten von v ?

- im Uhrzeigersinn nächste Kante nach e um v : $next(twin(e))$
- gegen den Uhrzeigersinn nächste Kante nach e um v :

Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



Für jede „Halbkante“ e

- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$
- nächste/vorherige Kante dieser Facette: $next(e)$, $prev(e)$

- für Knoten v / Facette f eine angrenzende Kante $edge(v)$ / $edge(f)$

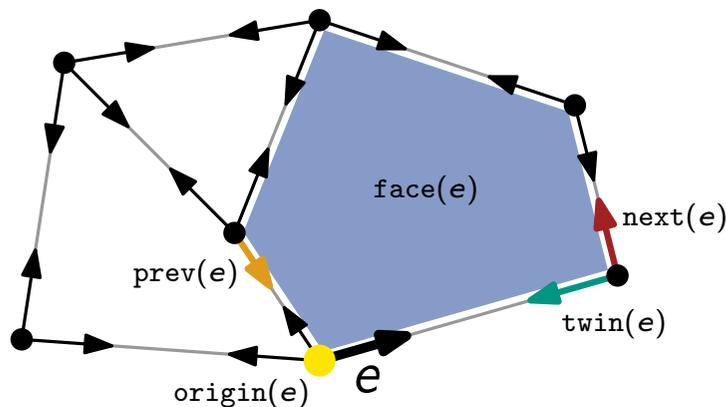
Wie findet man alle inzidenten Kanten von v ?

- im Uhrzeigersinn nächste Kante nach e um v : $next(twin(e))$
- gegen den Uhrzeigersinn nächste Kante nach e um v : $twin(prev(e))$

Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



Für jede „Halbkante“ e

- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$
- nächste/vorherige Kante dieser Facette: $next(e)$, $prev(e)$

- für Knoten v / Facette f eine angrenzende Kante $edge(v)$ / $edge(f)$

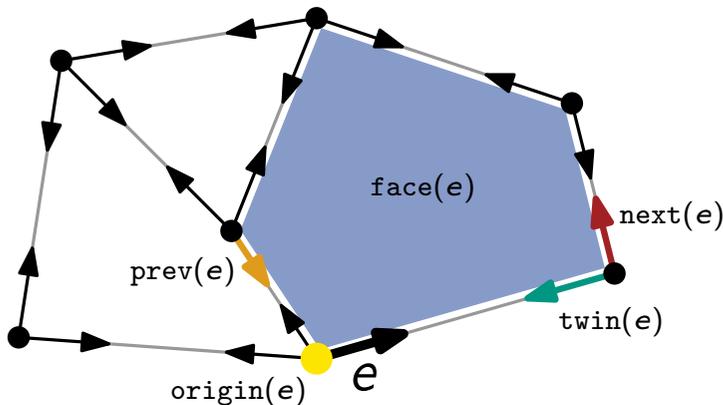
Wie findet man alle inzidenten Kanten von v ?

Wie findet man alle adjazenten Facetten von f ?

Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt

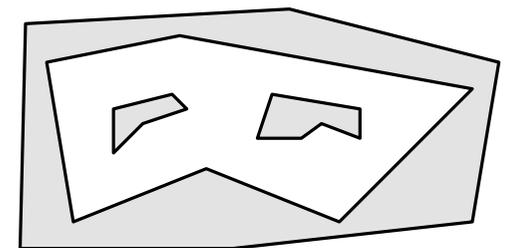


Für jede „Halbkante“ e

- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$
- nächste/vorherige Kante dieser Facette: $next(e)$, $prev(e)$

- für Knoten v / Facette f eine angrenzende Kante $edge(v)$ / $edge(f)$

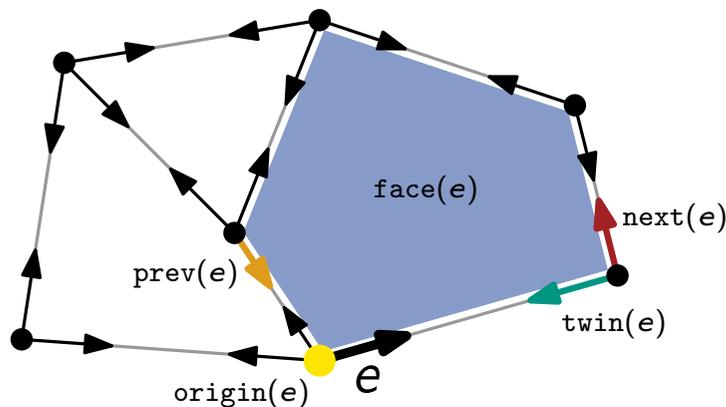
Mehrere Zusammenhangskomponenten



Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



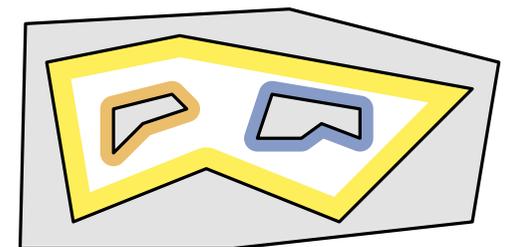
Für jede „Halbkante“ e

- zugehöriger Knoten: $origin(e)$
- Kante am anderen Endpunkt: $twin(e)$
- inzidente Facette (links): $face(e)$
- nächste/vorherige Kante dieser Facette: $next(e)$, $prev(e)$

- für Knoten v / Facette f eine angrenzende Kante $edge(v)$ / $edge(f)$

Mehrere Zusammenhangskomponenten

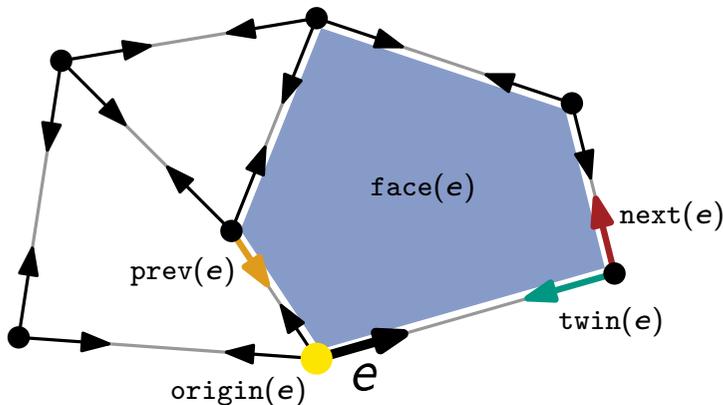
- zunächst: Struktur für jede Komponenten getrennt



Doppelt-verkettete Kantenliste

Doppelt-verkettete Kantenliste

- jede Kante hat 2 inzidente Knoten → speichere jede Kante doppelt



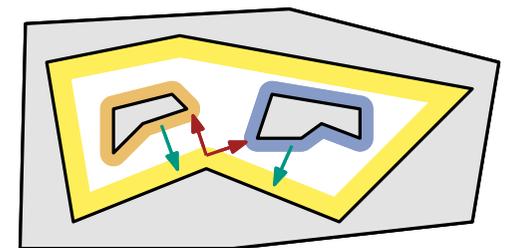
Für jede „Halbkante“ e

- zugehöriger Knoten: `origin(e)`
- Kante am anderen Endpunkt: `twin(e)`
- inzidente Facette (links): `face(e)`
- nächste/vorherige Kante dieser Facette: `next(e)`, `prev(e)`

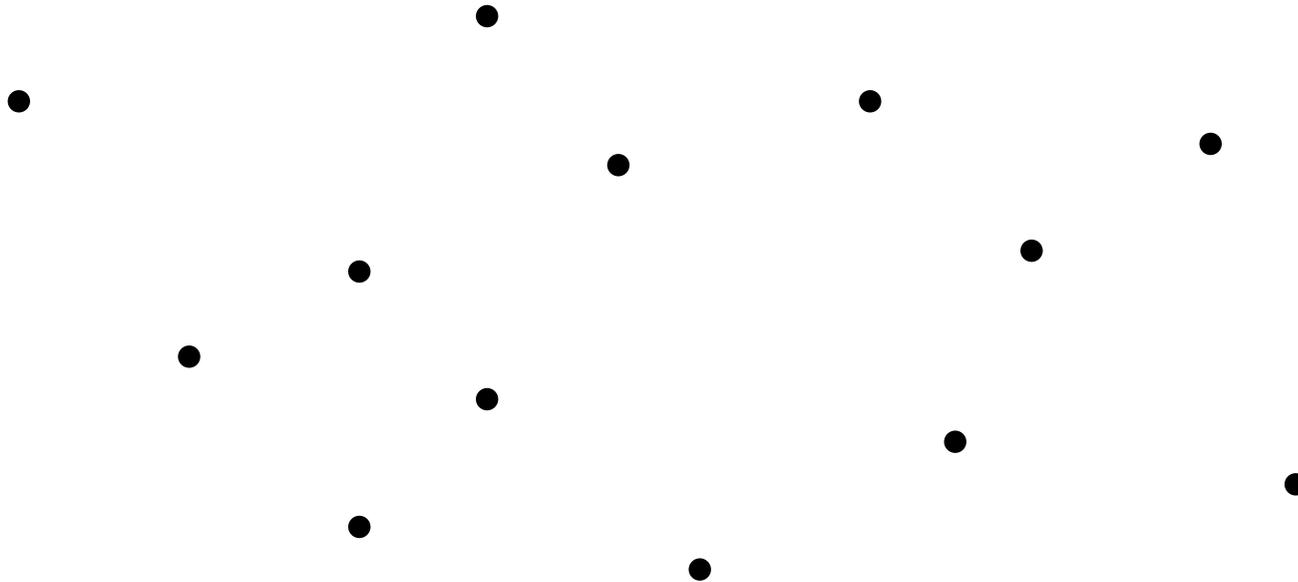
- für Knoten v / Facette f eine angrenzende Kante `edge(v)` / `edge(f)`

Mehrere Zusammenhangskomponenten

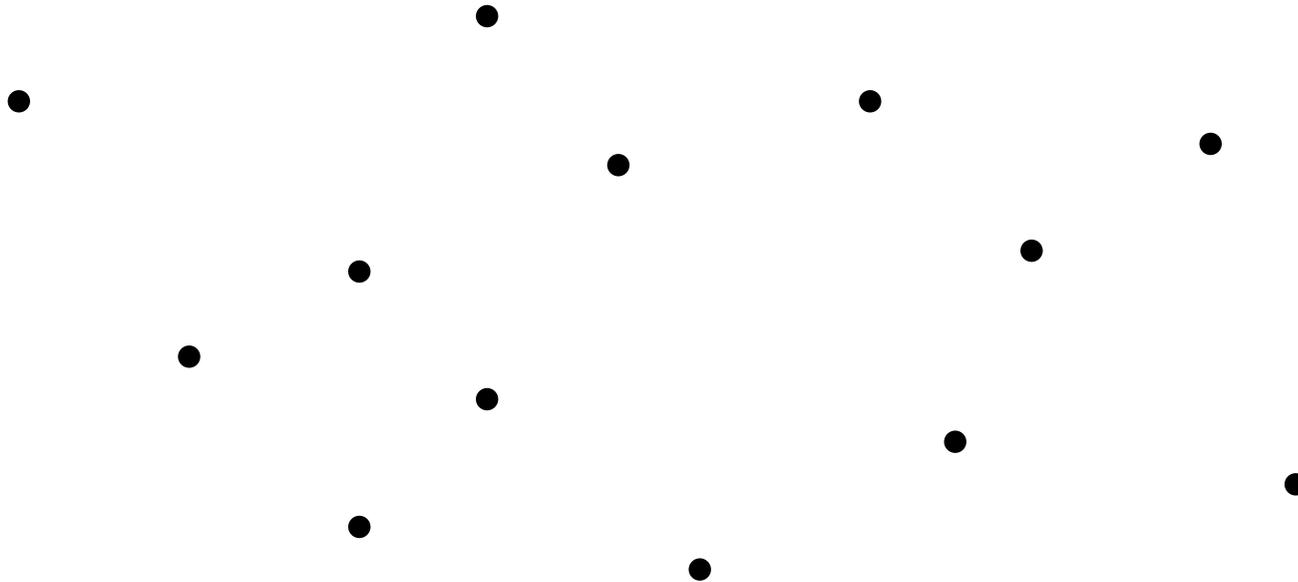
- zunächst: Struktur für jede Komponenten getrennt
- innere Facette f : Liste `children(f)` von Kindfacetten
- äußere Facette f : Elternfacette `parent(f)`



Maximale Distanz

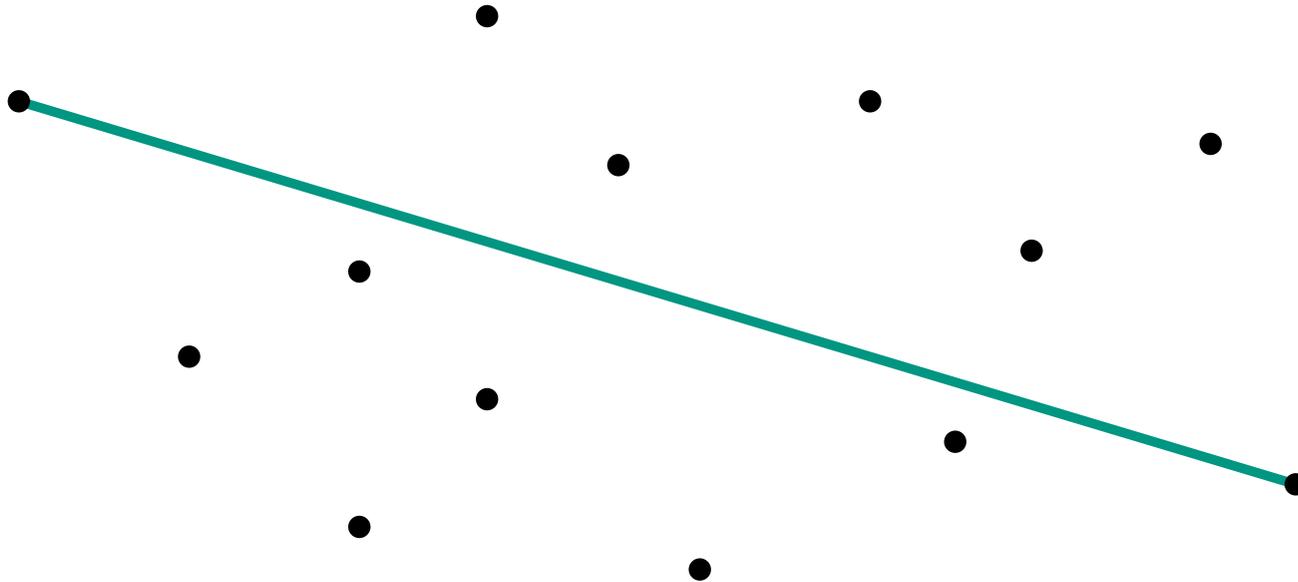


Maximale Distanz



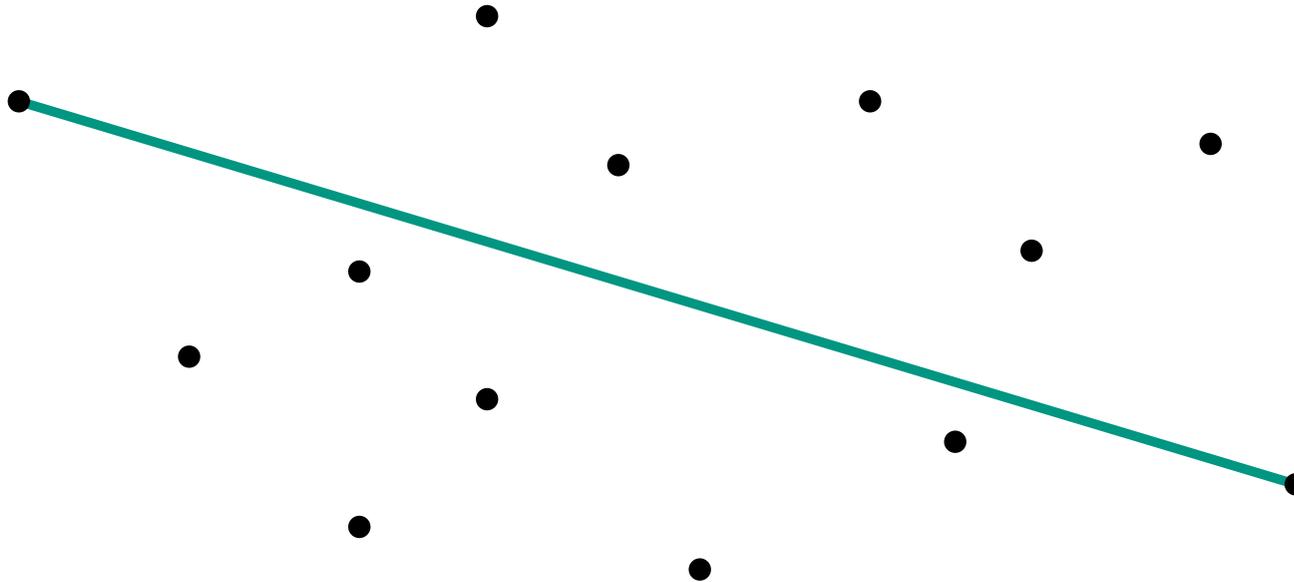
Gesucht: zwei Punkte mit maximaler Distanz zueinander

Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

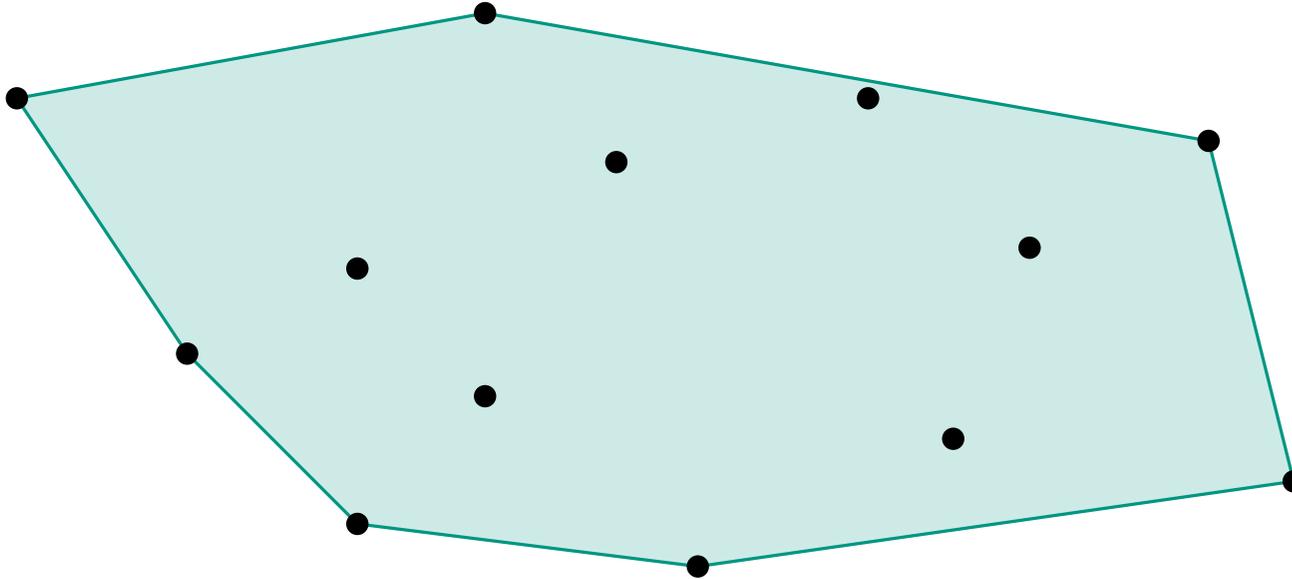
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?

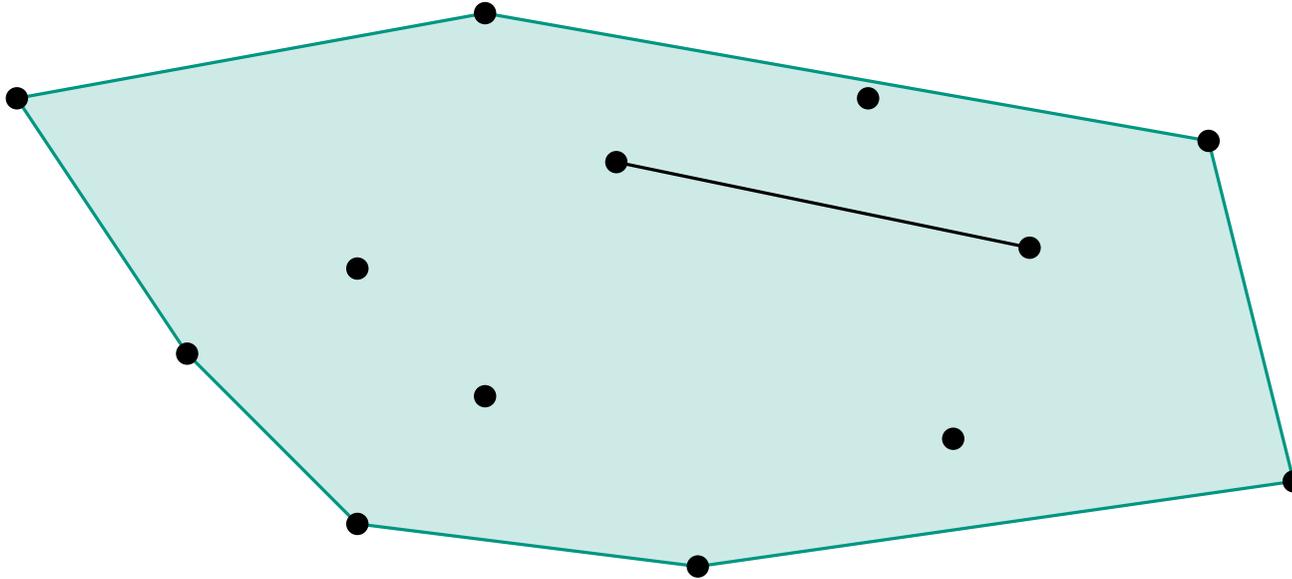
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?

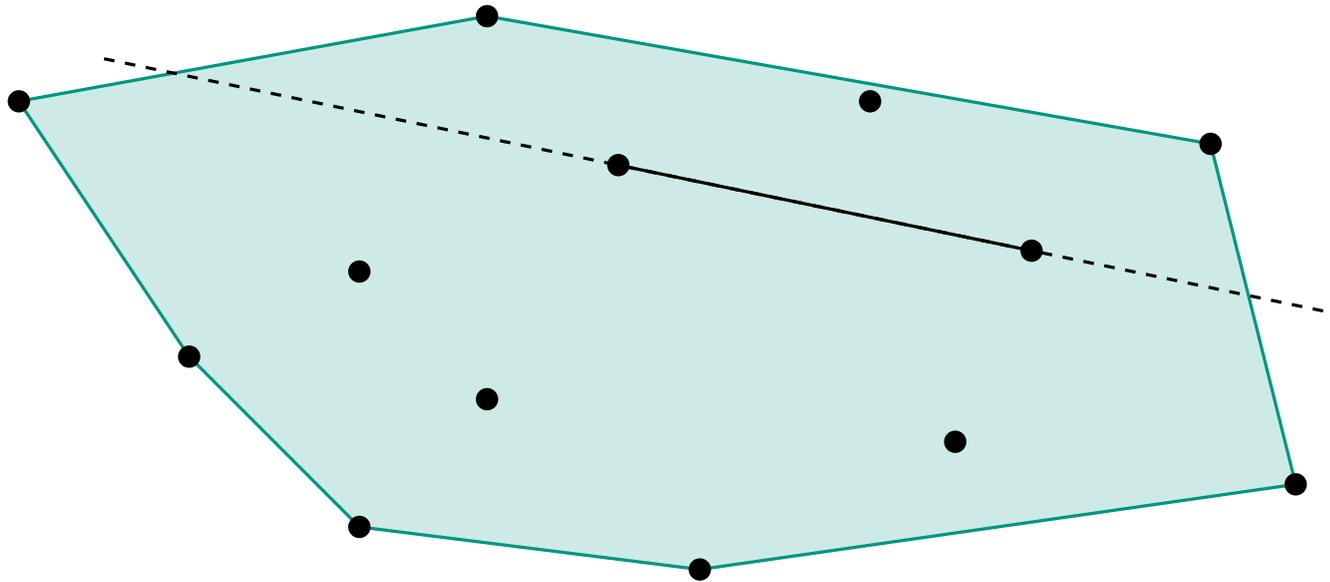
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?

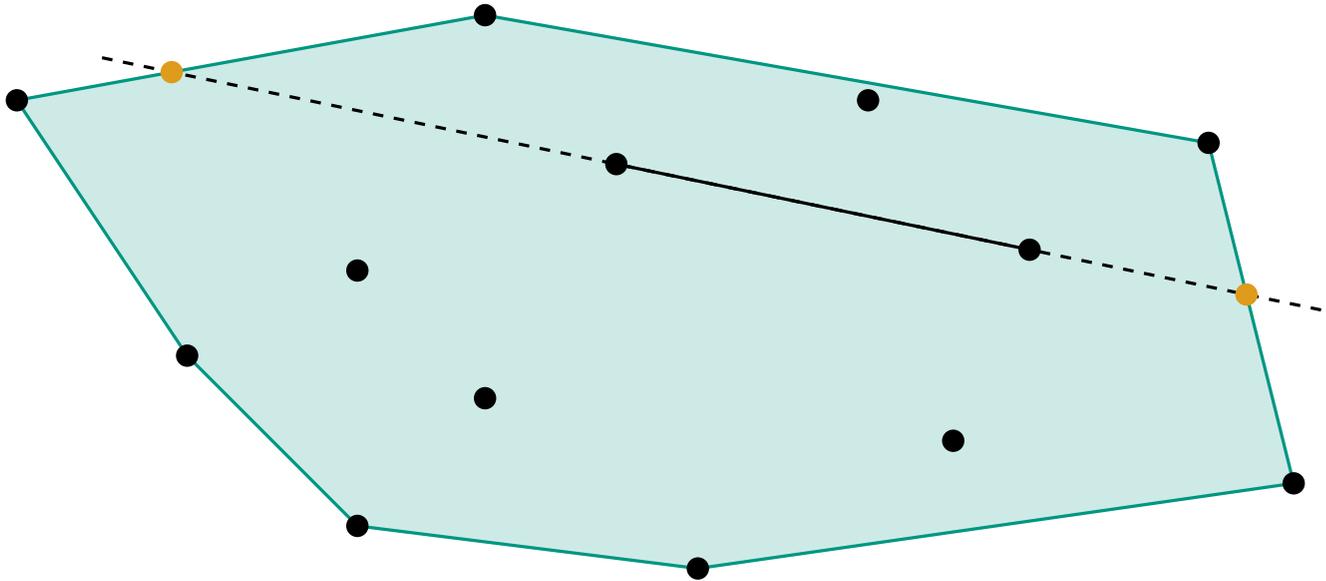
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?

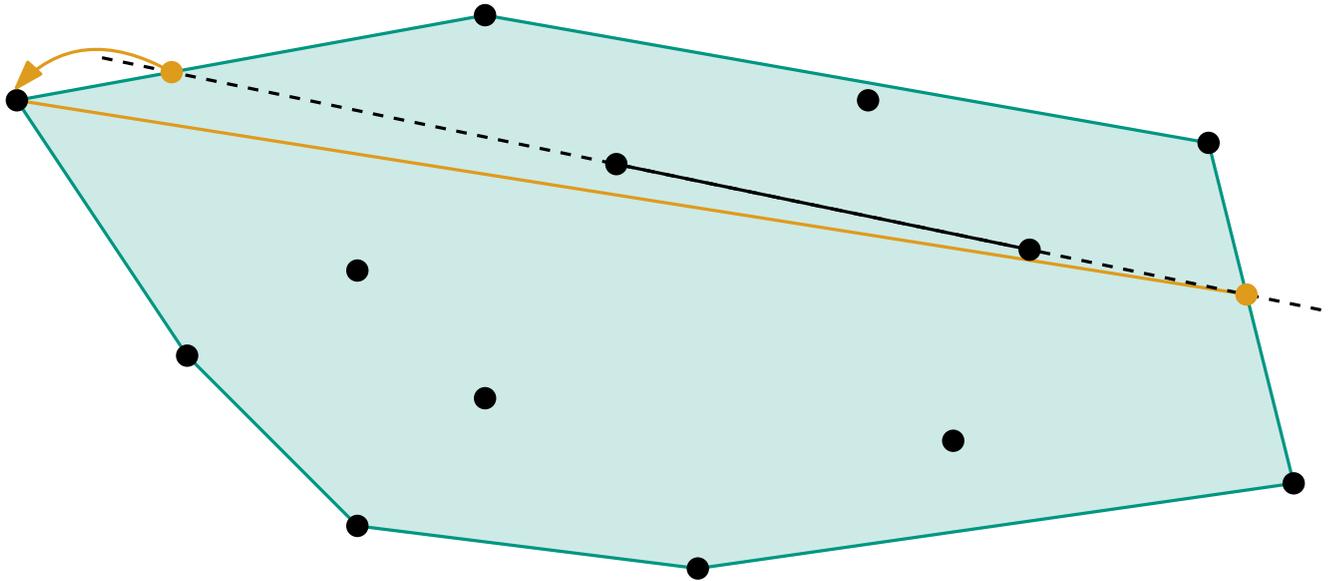
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?

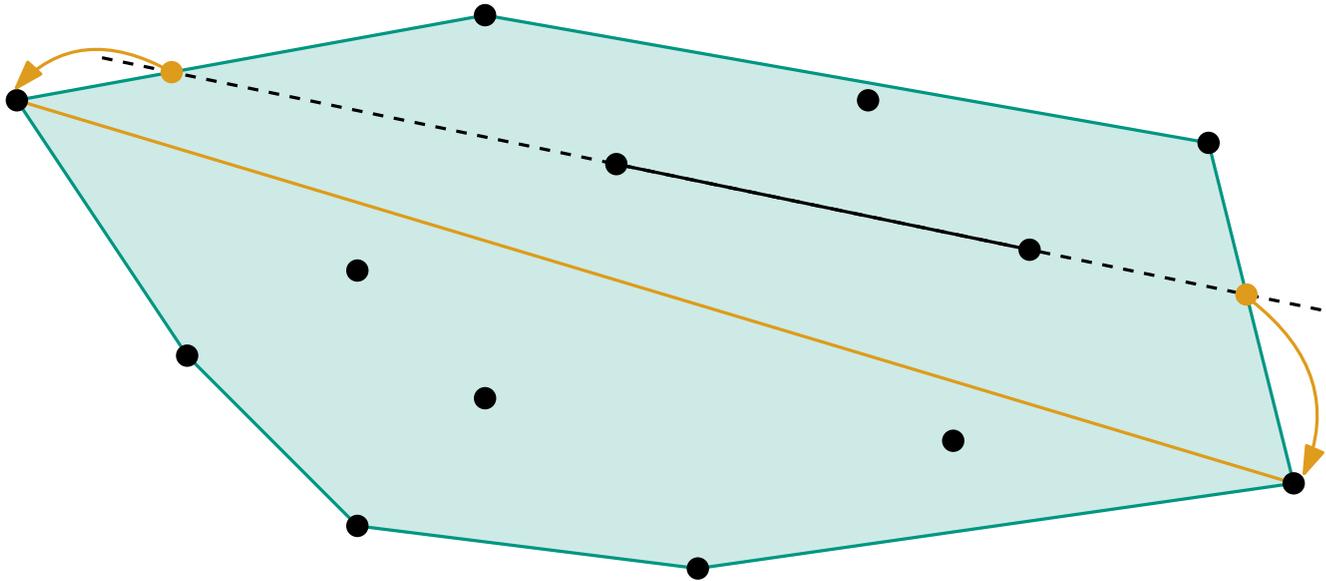
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktpaare sind Kandidaten? Warum?

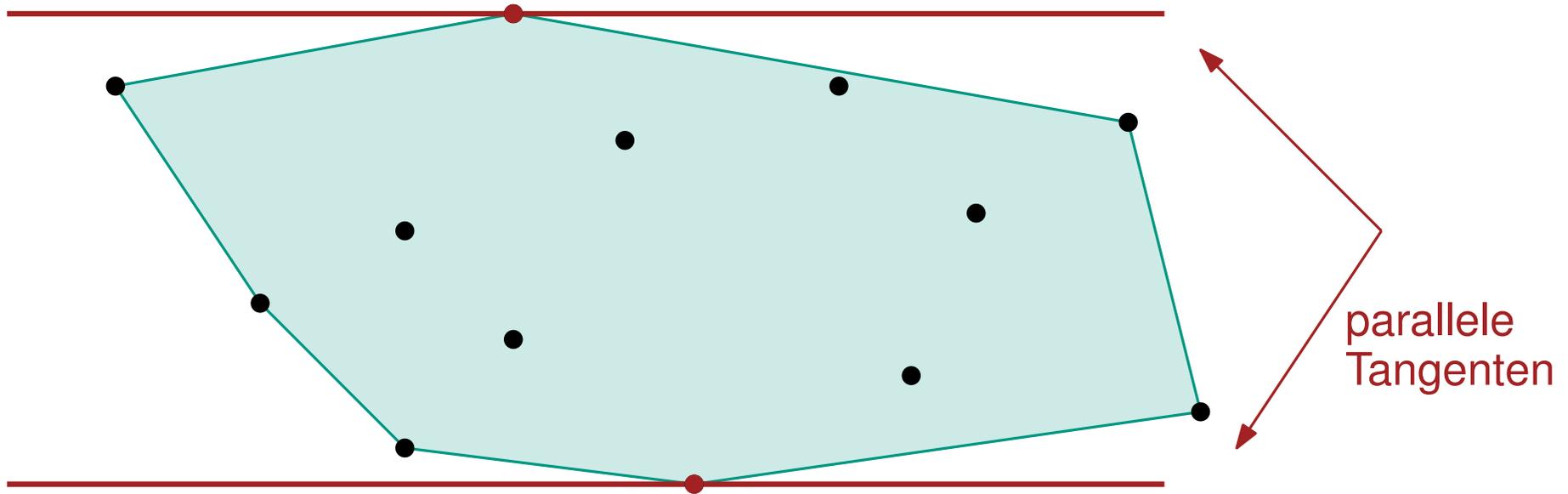
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?

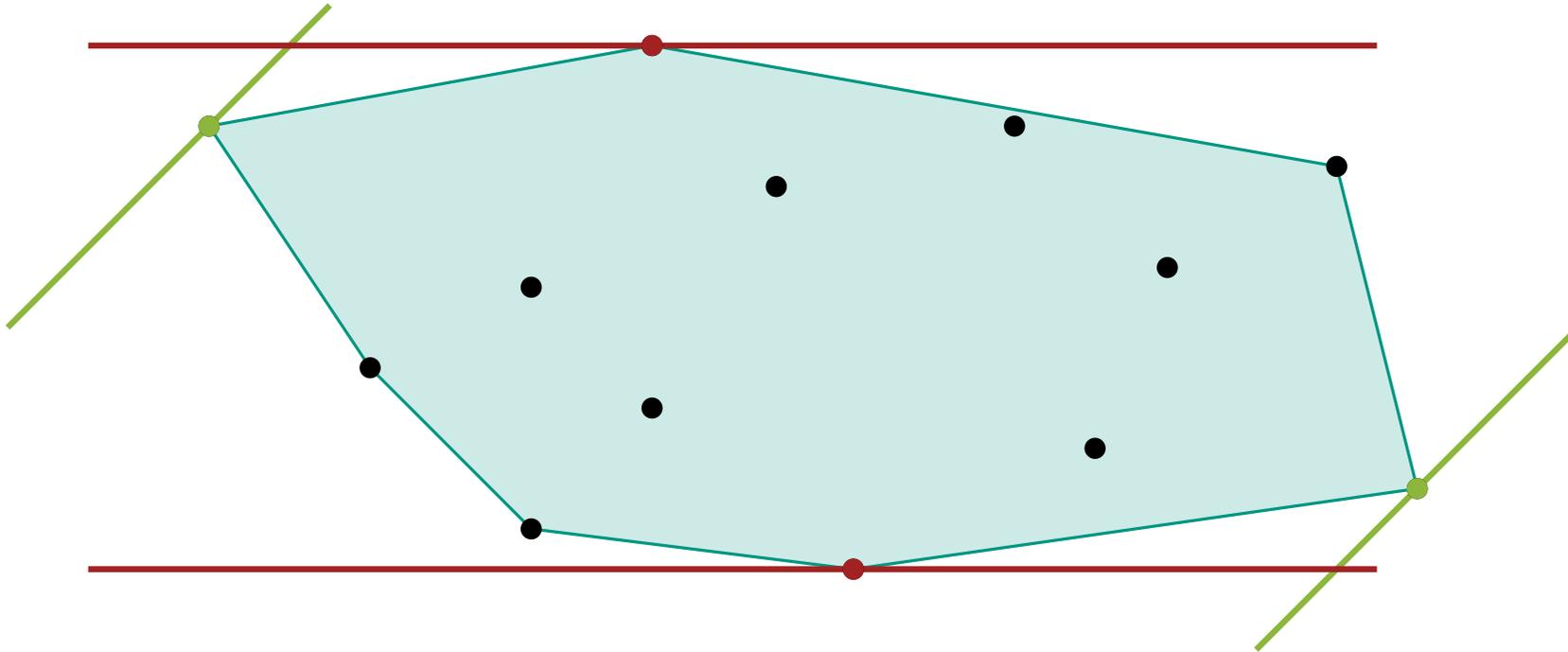
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal

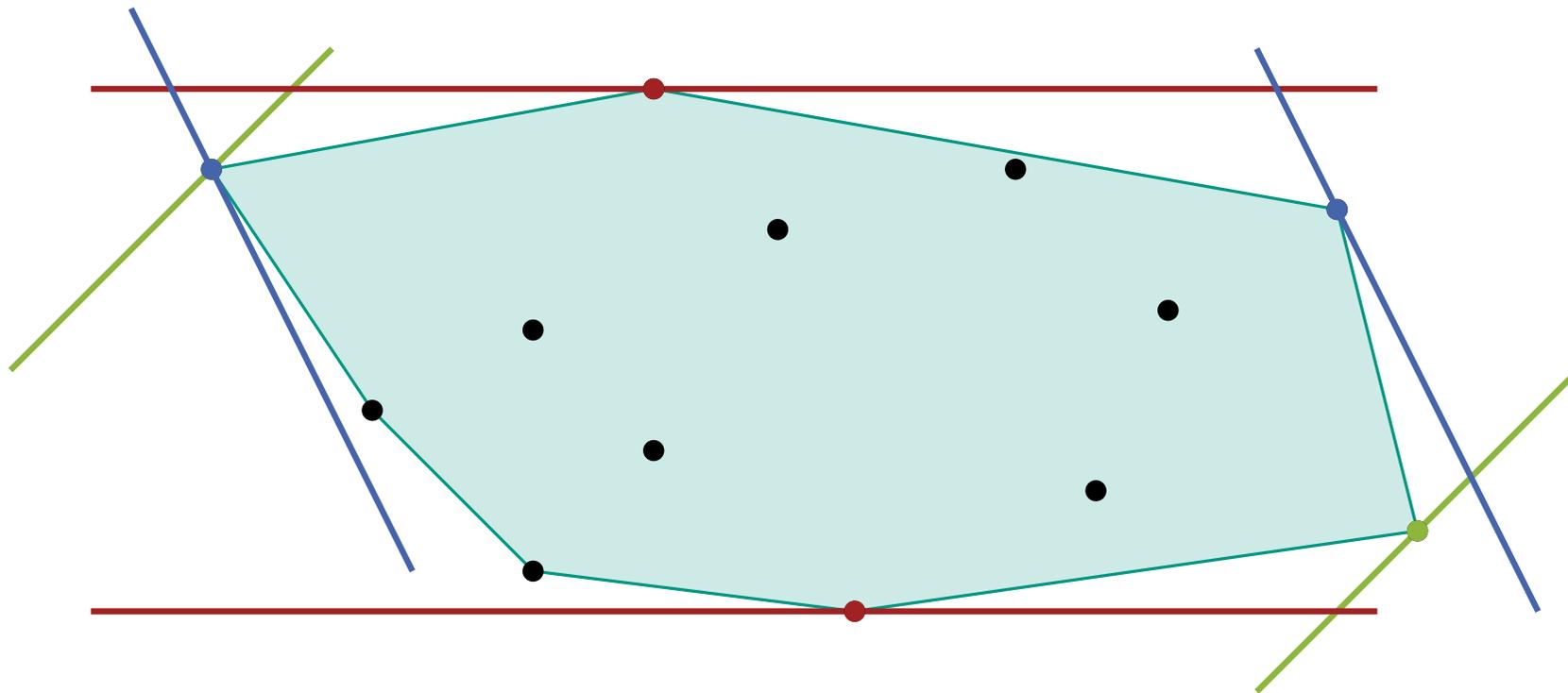
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktpaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal

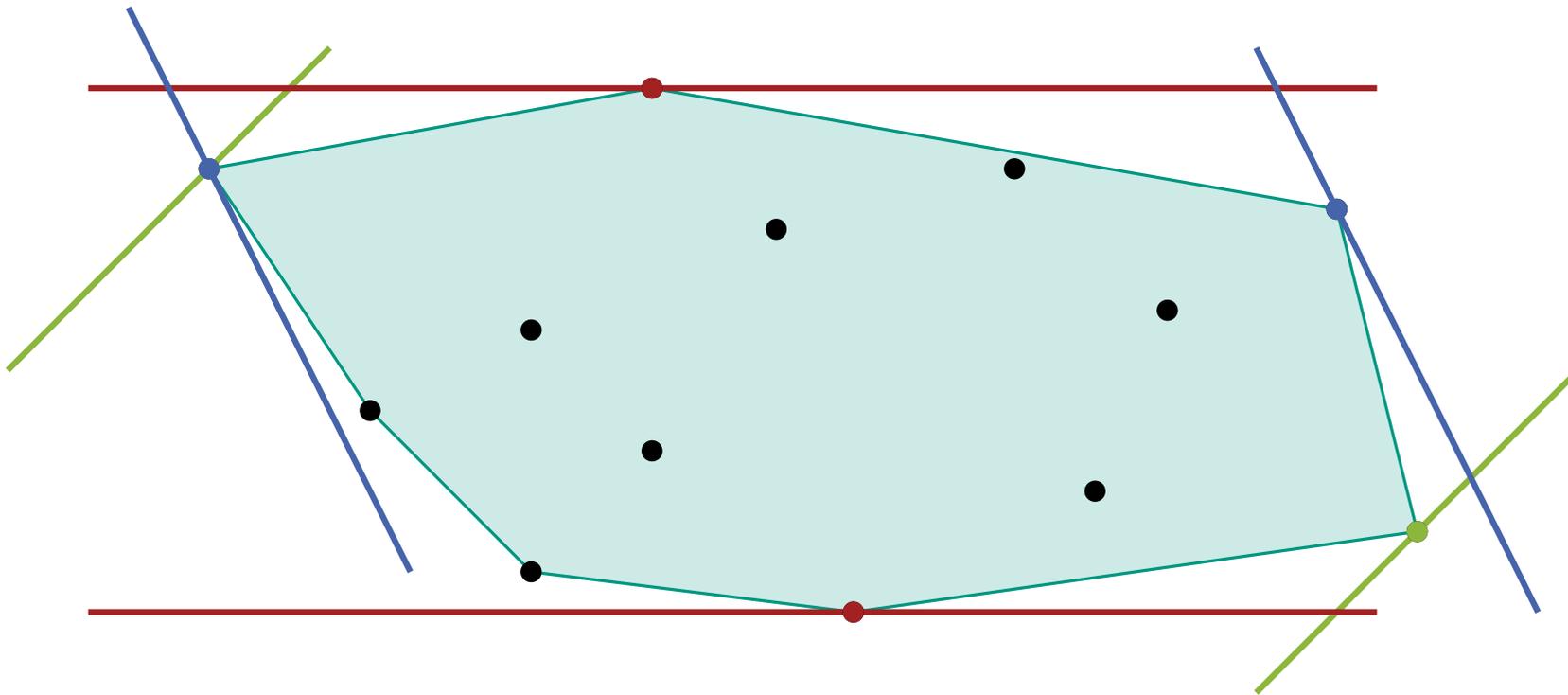
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal

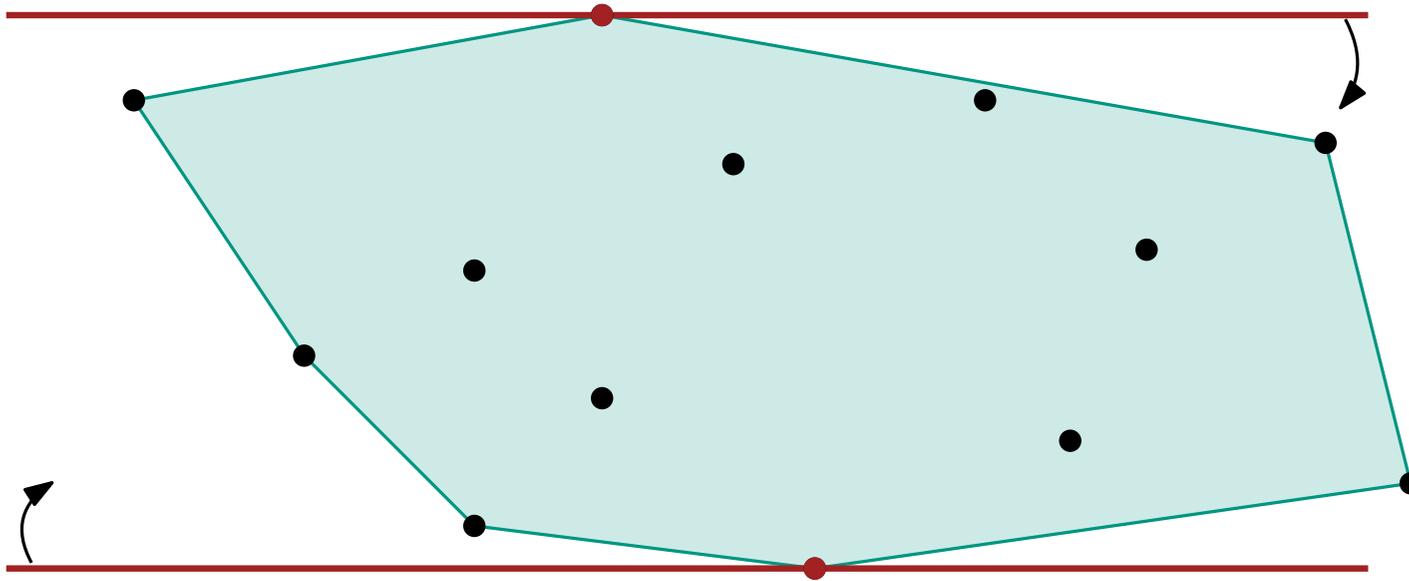
Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Maximale Distanz

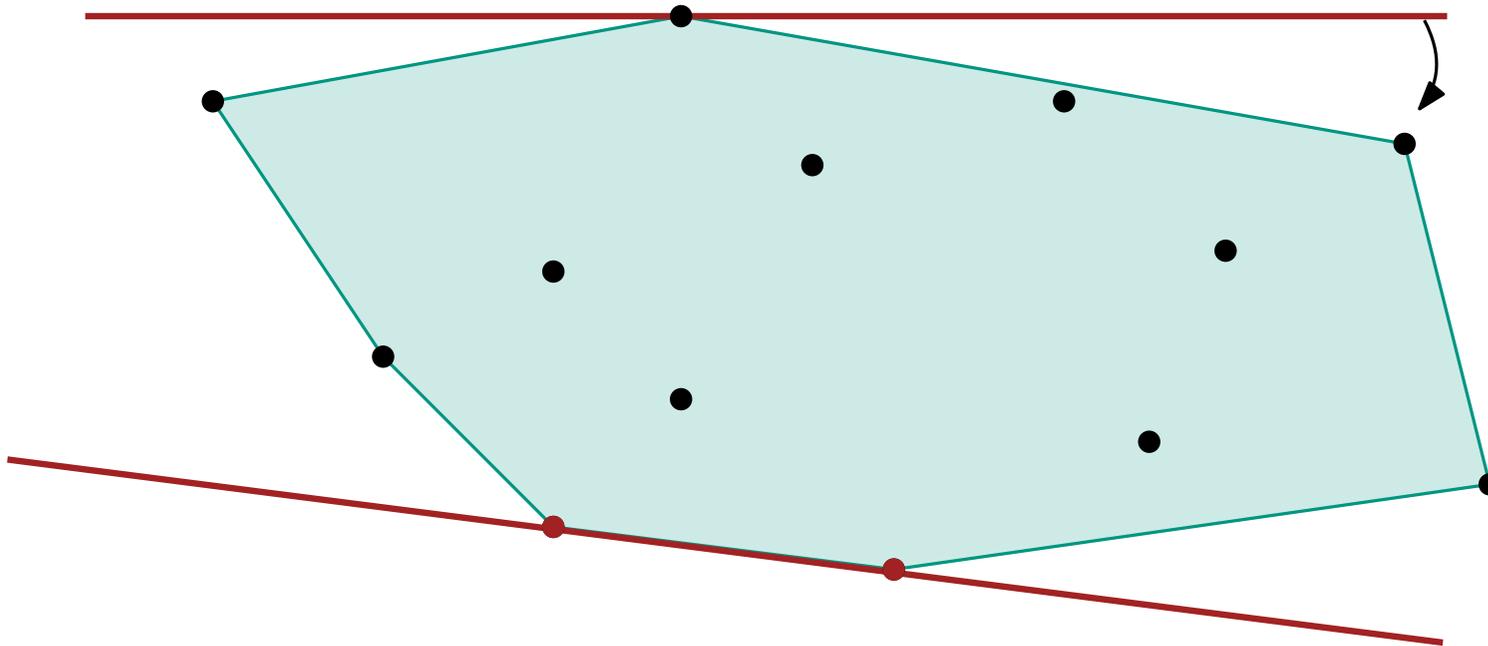


Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

Maximale Distanz

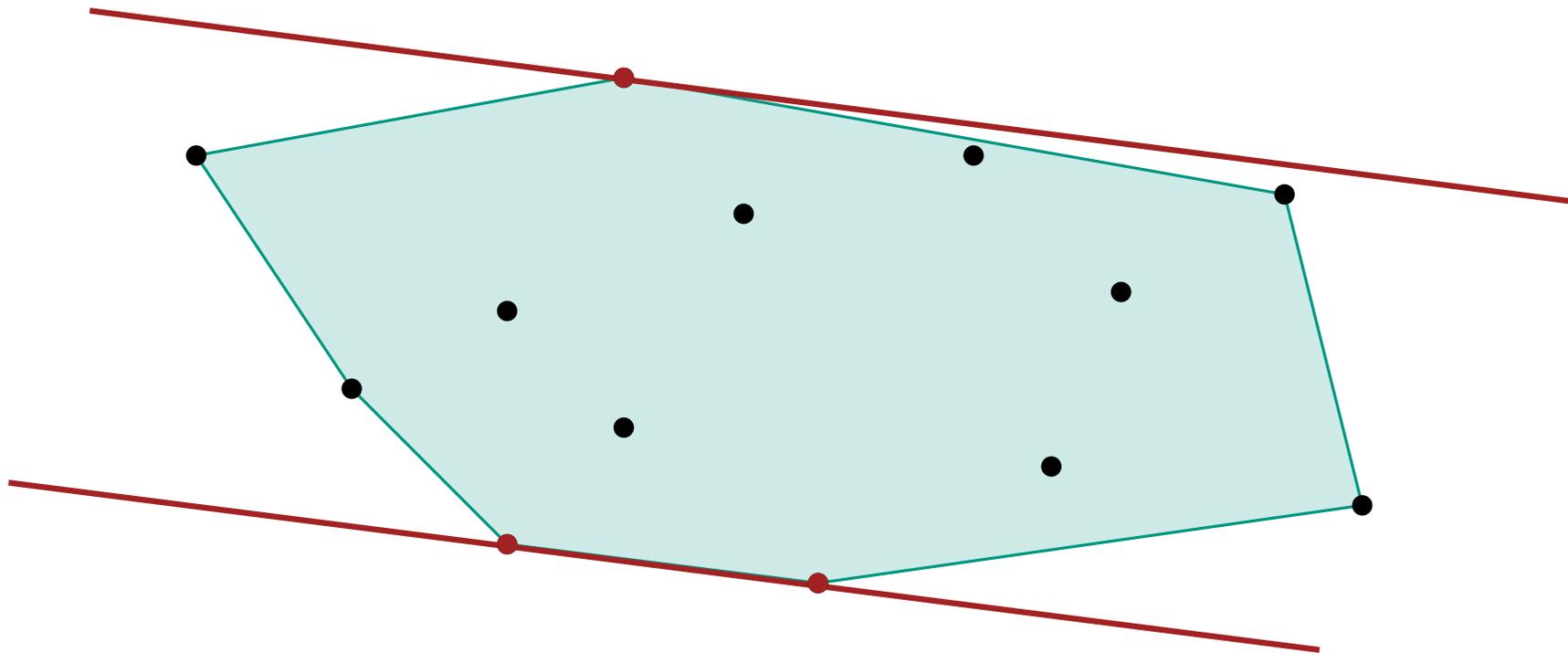


Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktpaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

Maximale Distanz

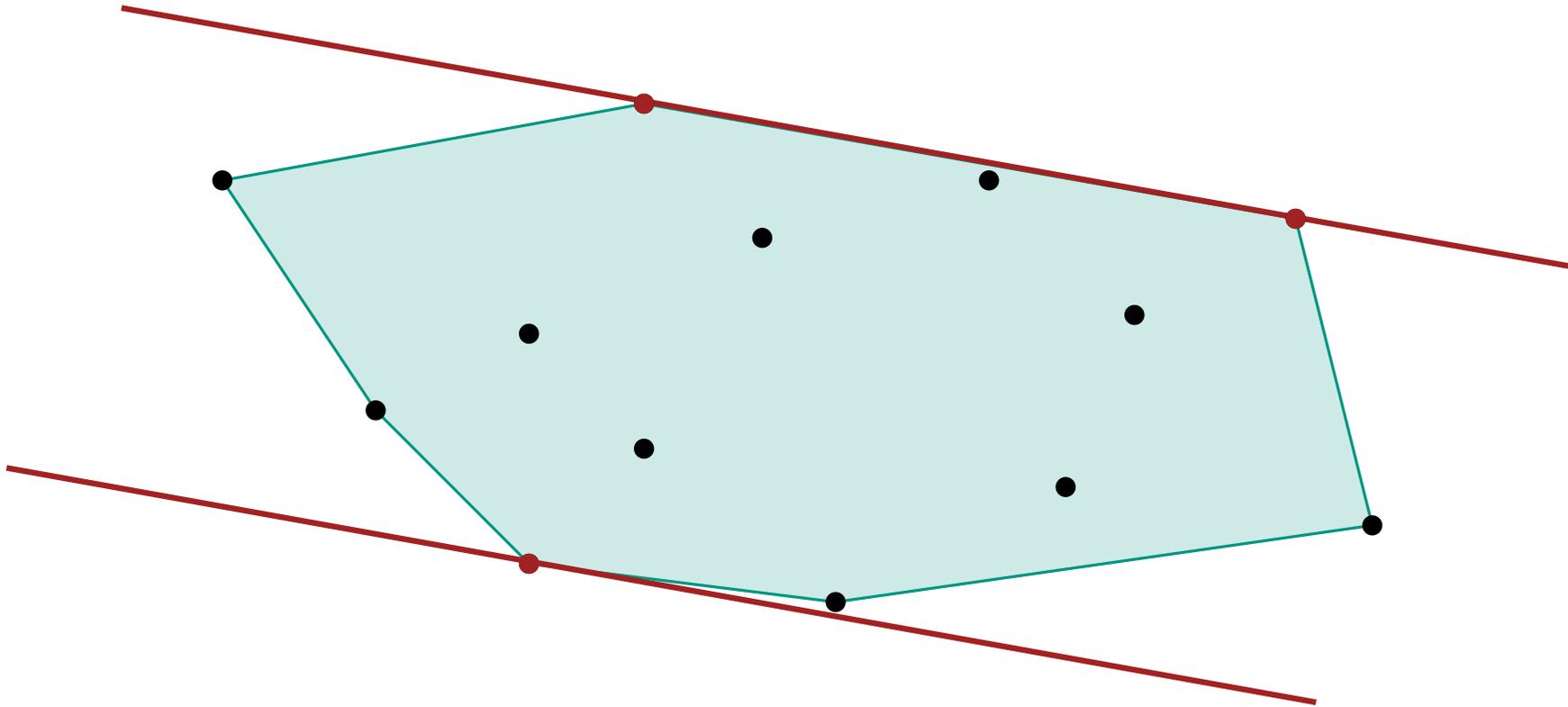


Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

Maximale Distanz

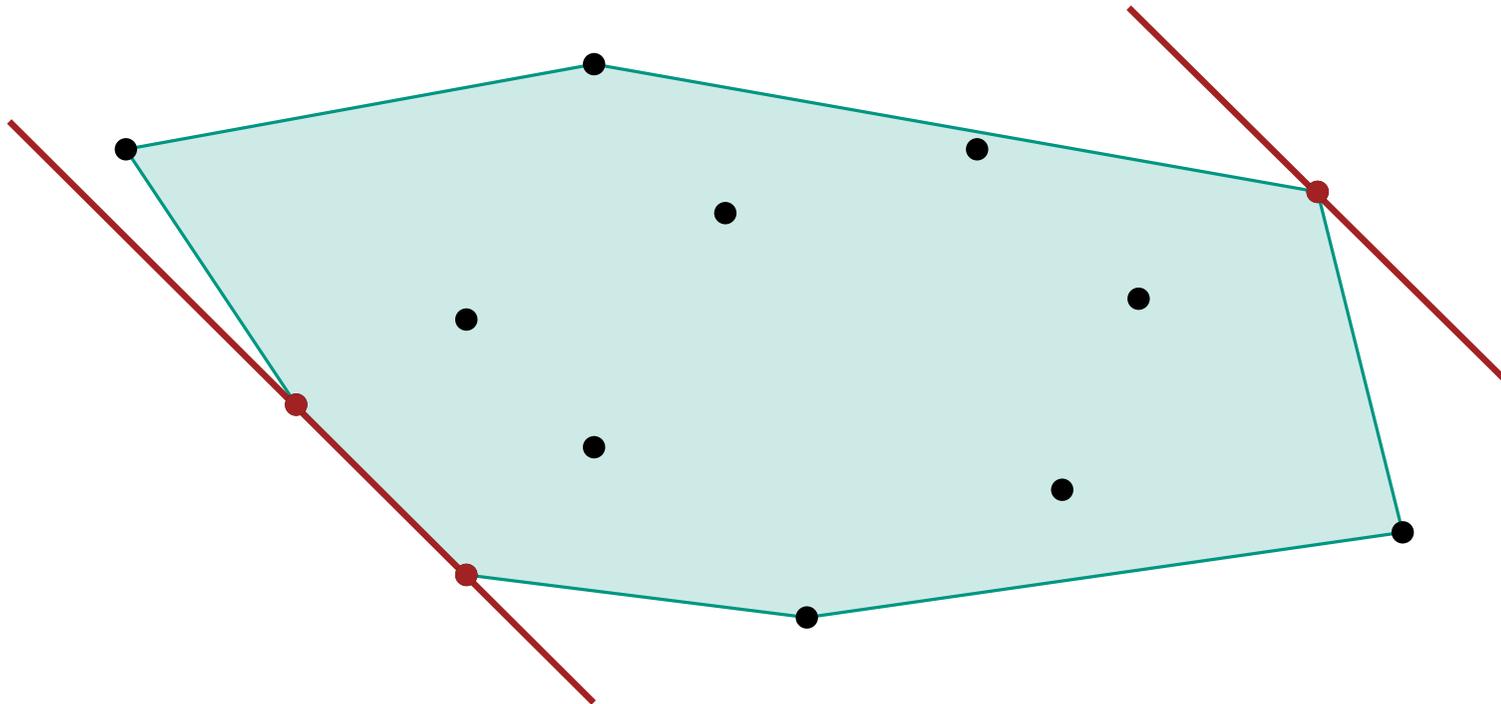


Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

Maximale Distanz

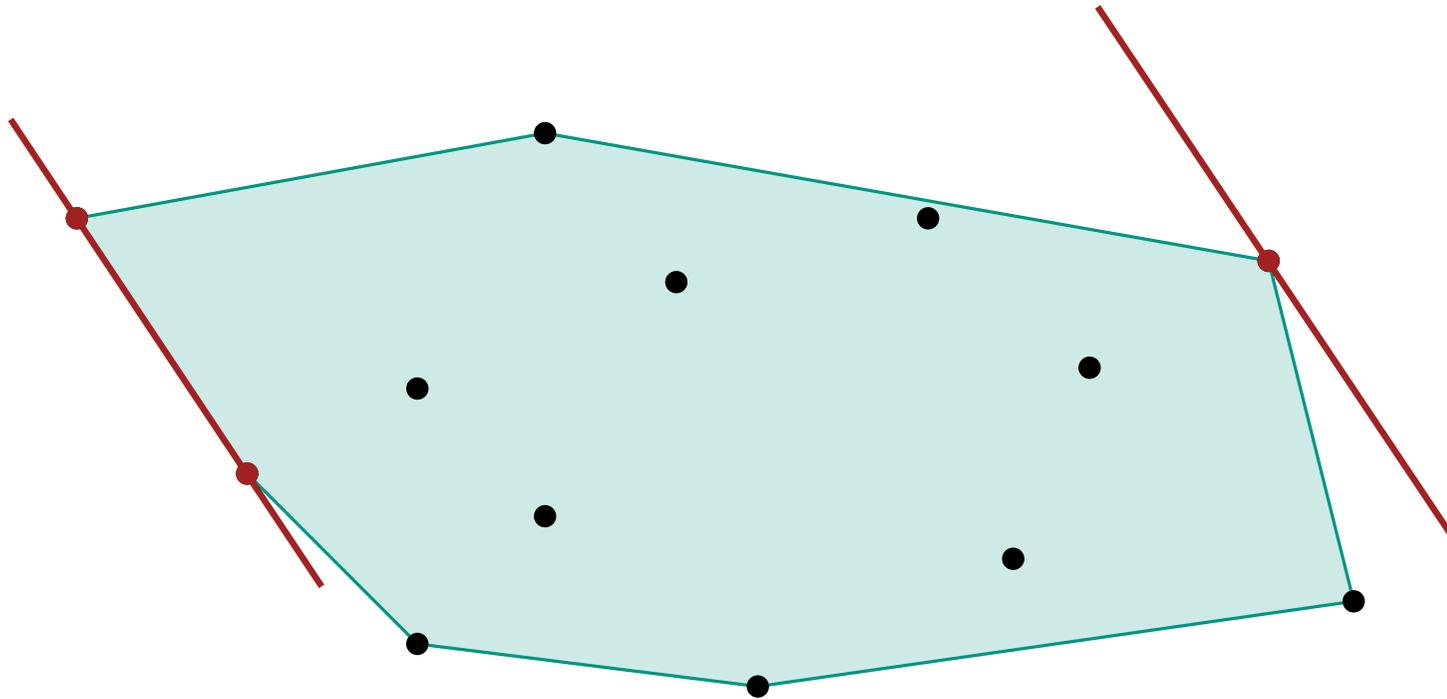


Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

Maximale Distanz

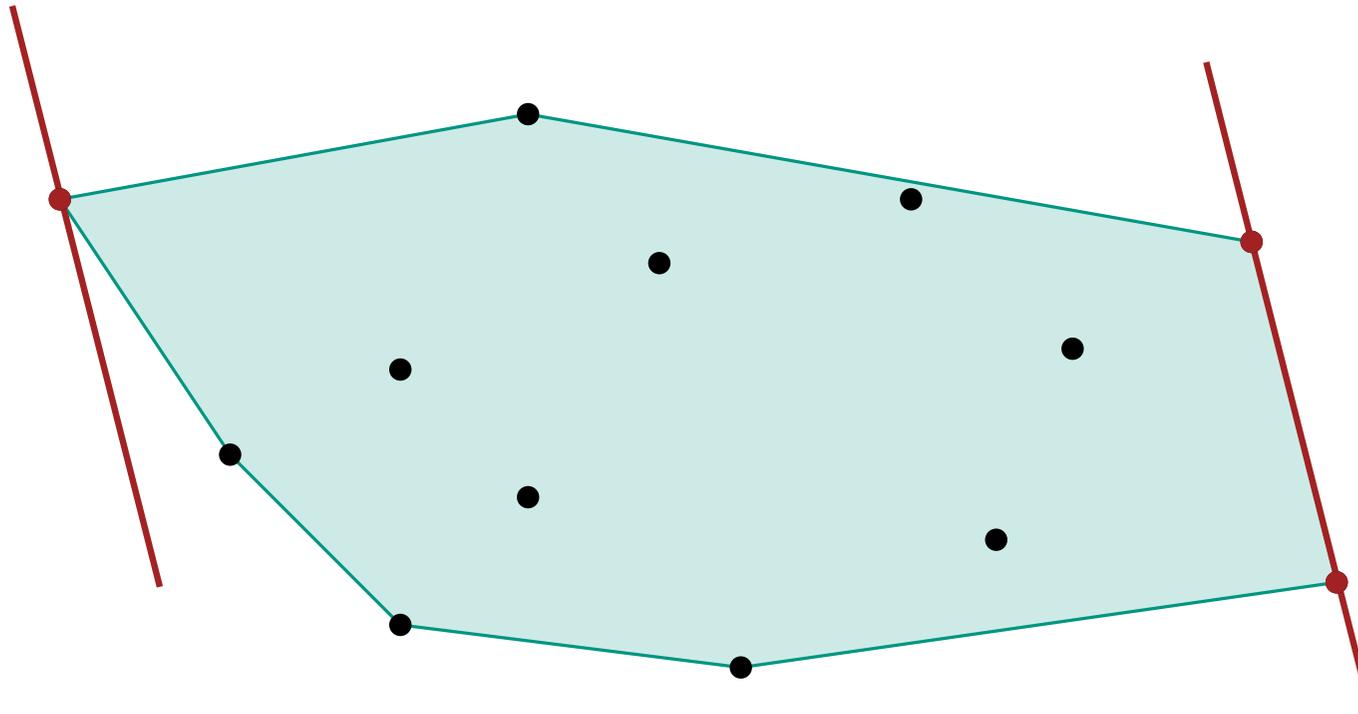


Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

Maximale Distanz

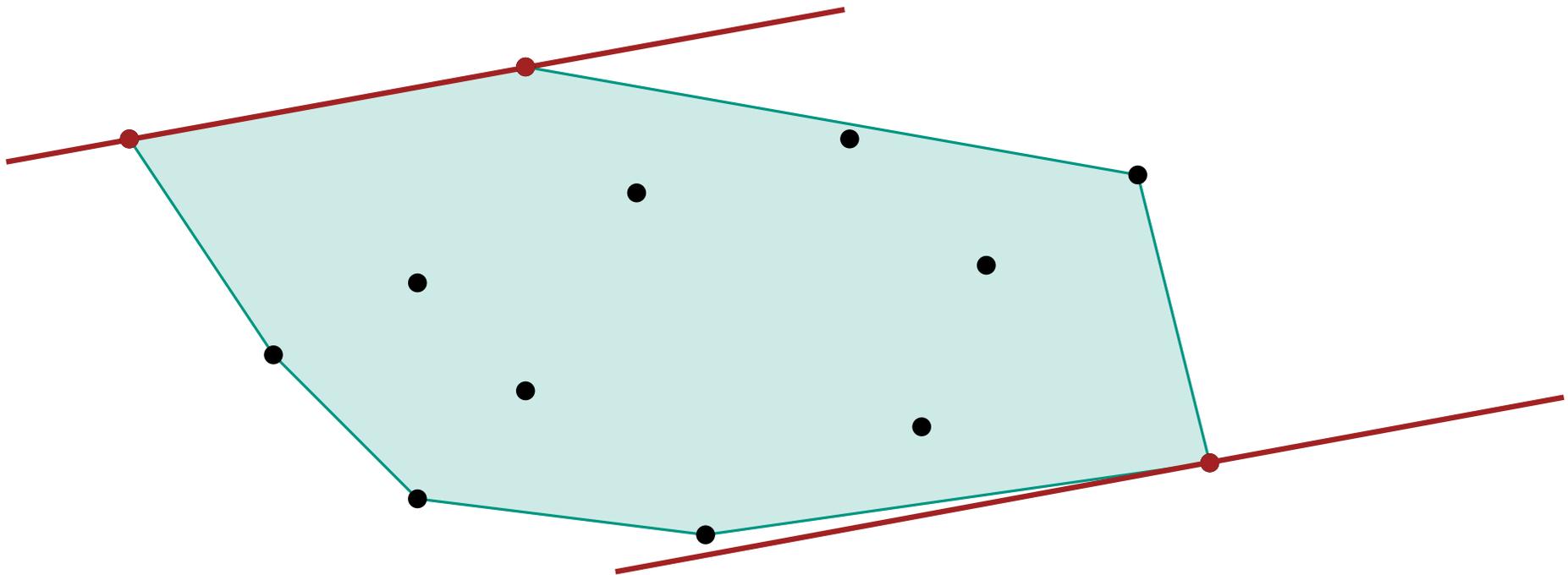


Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

Maximale Distanz

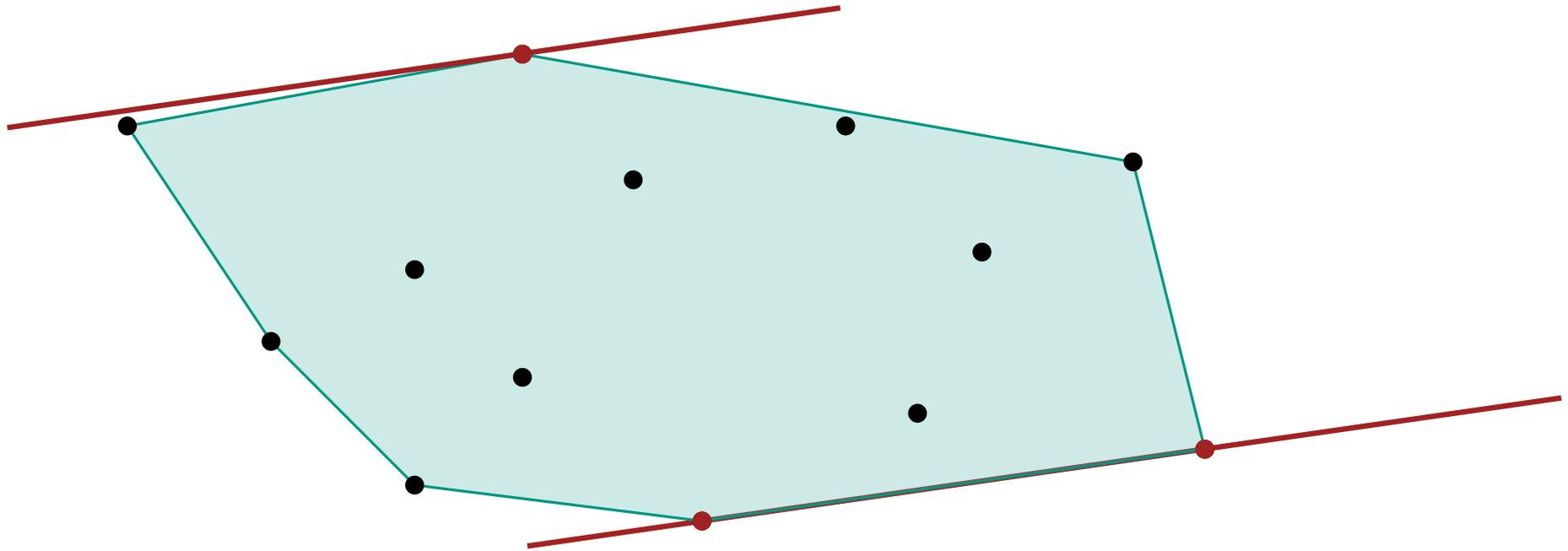


Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktpaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

Maximale Distanz

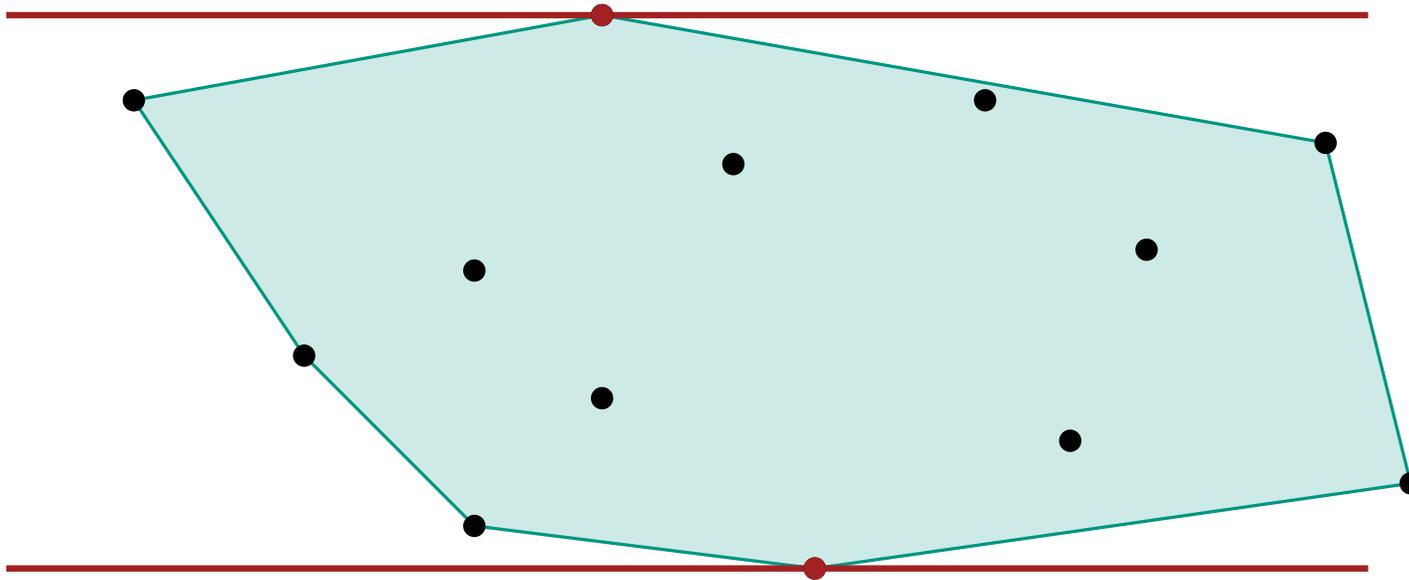


Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

Maximale Distanz



Gesucht: zwei Punkte mit maximaler Distanz zueinander

- Welche Punktepaare sind Kandidaten? Warum?
- Die zwei gesuchten Punkte sind antipodal zueinander. Warum?
Tipp: zeige maximale Distanz \Rightarrow antipodal
- Wie generiert man alle antipodalen Punkte in $O(n)$?
Tipp: beginne mit antipodalem Paar, finde nächstes Paar

Rotating Caliper

ÜB Tipps

Aufgabe 1

ÜB Tipps

Aufgabe 1

a) grobe Beschreibung + Begründung reicht, xhairkr Uhryyr

ÜB Tipps

Aufgabe 1

- a) grobe Beschreibung + Begründung reicht, xhairkr Uhryyr
- b) ausprobieren

ÜB Tipps

Aufgabe 1

- a) grobe Beschreibung + Begründung reicht, xhairkr Uhryyr
- b) ausprobieren

Aufgabe 2

- a) Woraus besteht die konvexe Hülle von zwei Polygonen?

ÜB Tipps

Aufgabe 1

- a) grobe Beschreibung + Begründung reicht, xhairkr Uhryyr
- b) ausprobieren

Aufgabe 2

- a) Woraus besteht die konvexe Hülle von zwei Polygonen?

Aufgabe 3

Ynfre fpuvpxra