Übung Algorithmische Geometrie
Sichtbarkeitsgraph

LEHRSTUHL FÜR ALGORITHMIK I · INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Andreas Gemsa

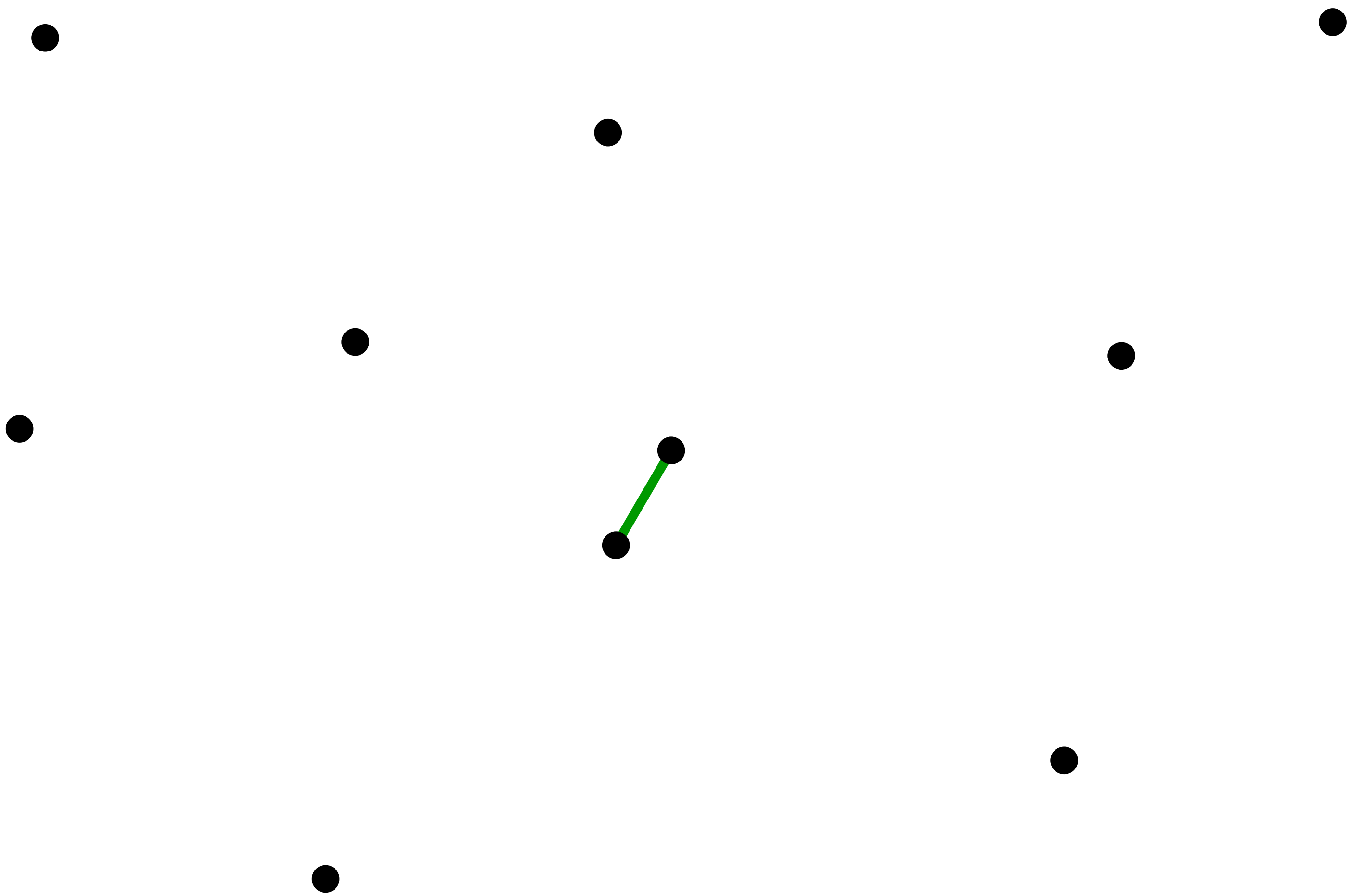
19.07.2012

Ablauf

Nachtrag

Sichtbarkeitsgraph

WSPD



Bedingung: Für jedes $q, p \in P$ existiert genau ein Index ℓ, so dass $q \in A_\ell$ und $p \in B_\ell$ für $\{A_\ell, B_\ell\}$ in $\mathcal{W}$.

Ablauf

Nachtrag

Sichtbarkeitsgraph

Ablauf

Nachtrag

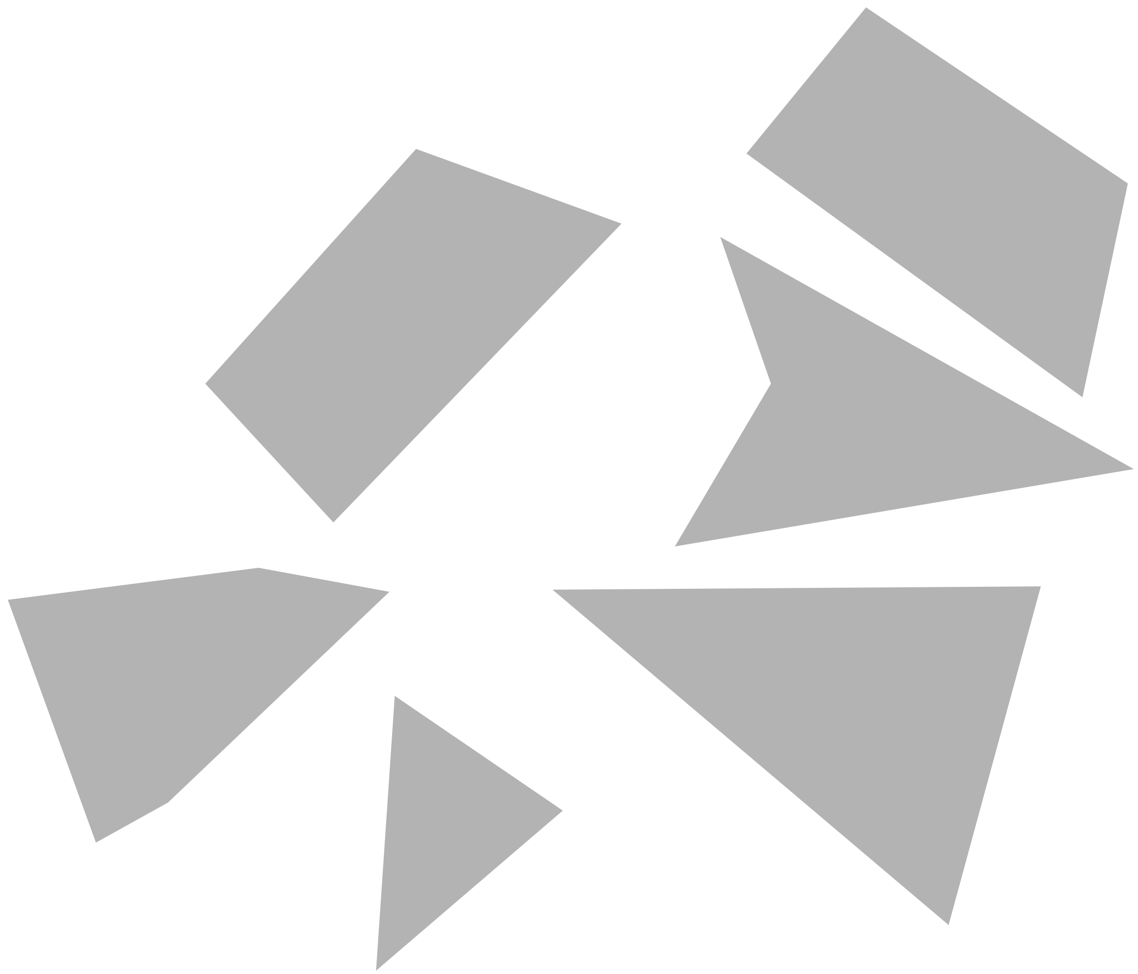
Sichtbarkeitsgraph

Aufgabe 1

Gegeben: Menge S von Polygonen mit insgesamt n Knoten.

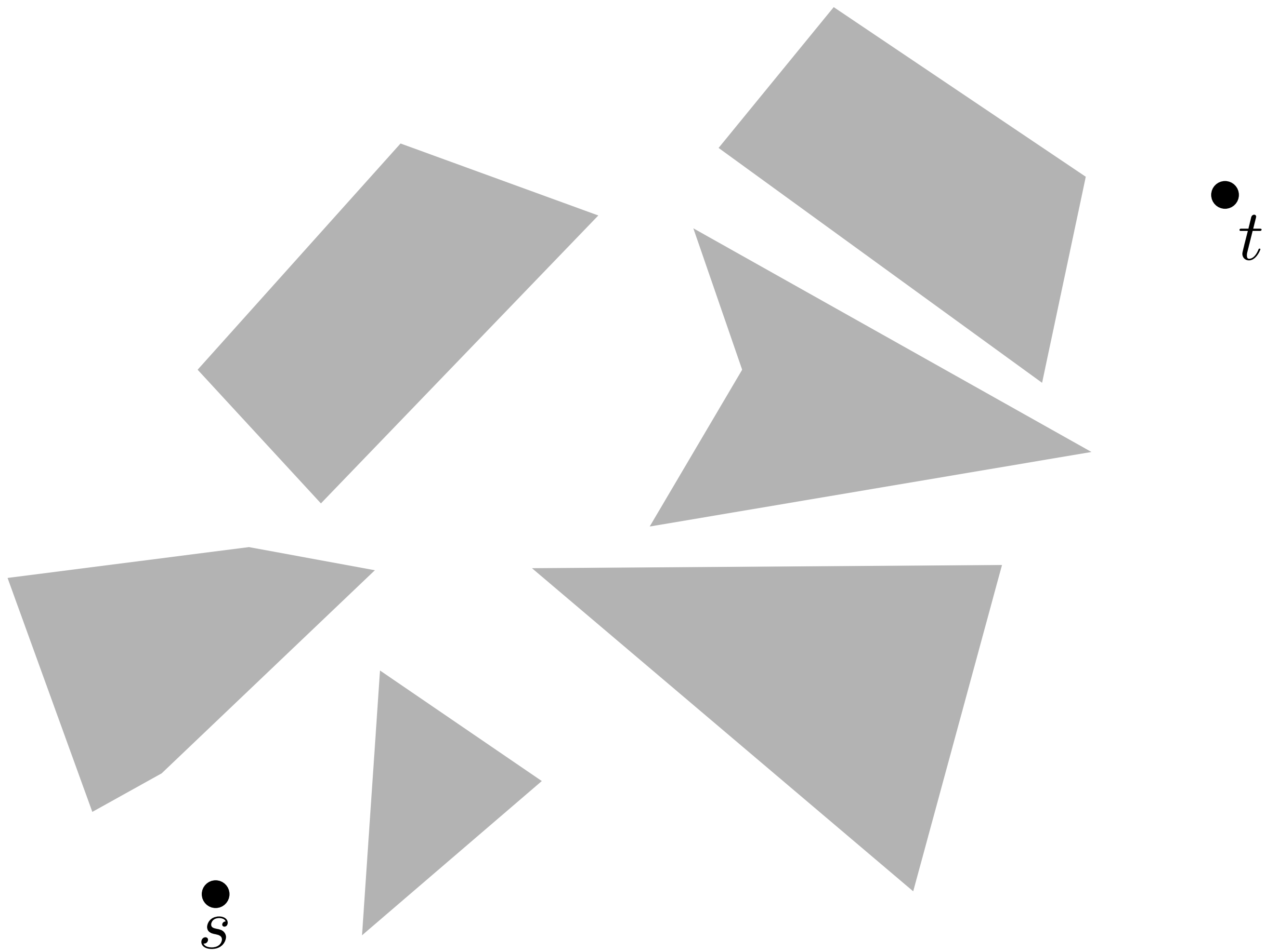
Aufgabe 1

Gegeben: Menge S von Polygonen mit insgesamt n Knoten.



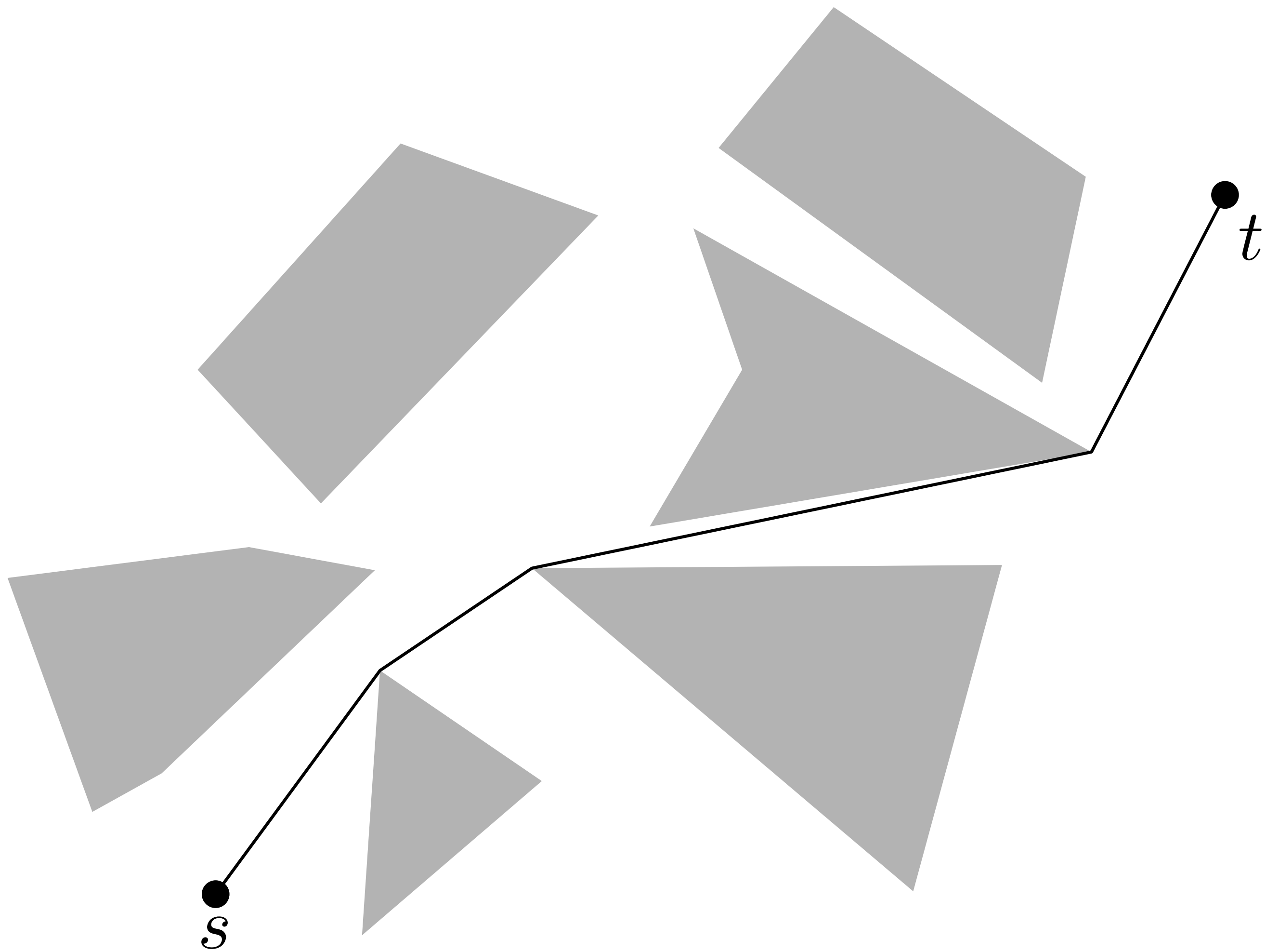
Aufgabe 1

Gegeben: Menge S von Polygonen mit insgesamt n Knoten.



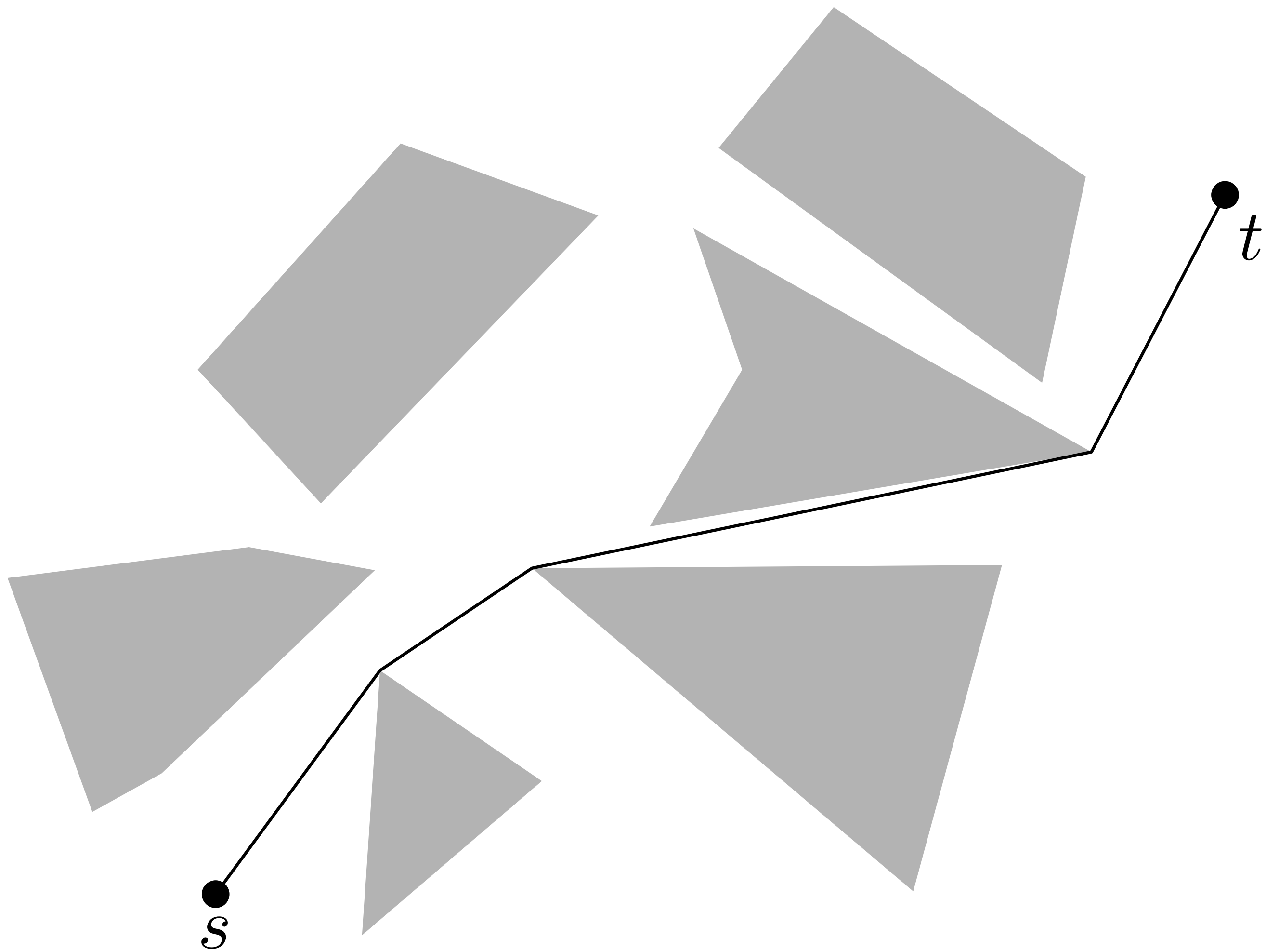
Aufgabe 1

Gegeben: Menge S von Polygonen mit insgesamt n Knoten.



Aufgabe 1

Gegeben: Menge S von Polygonen mit insgesamt n Knoten.

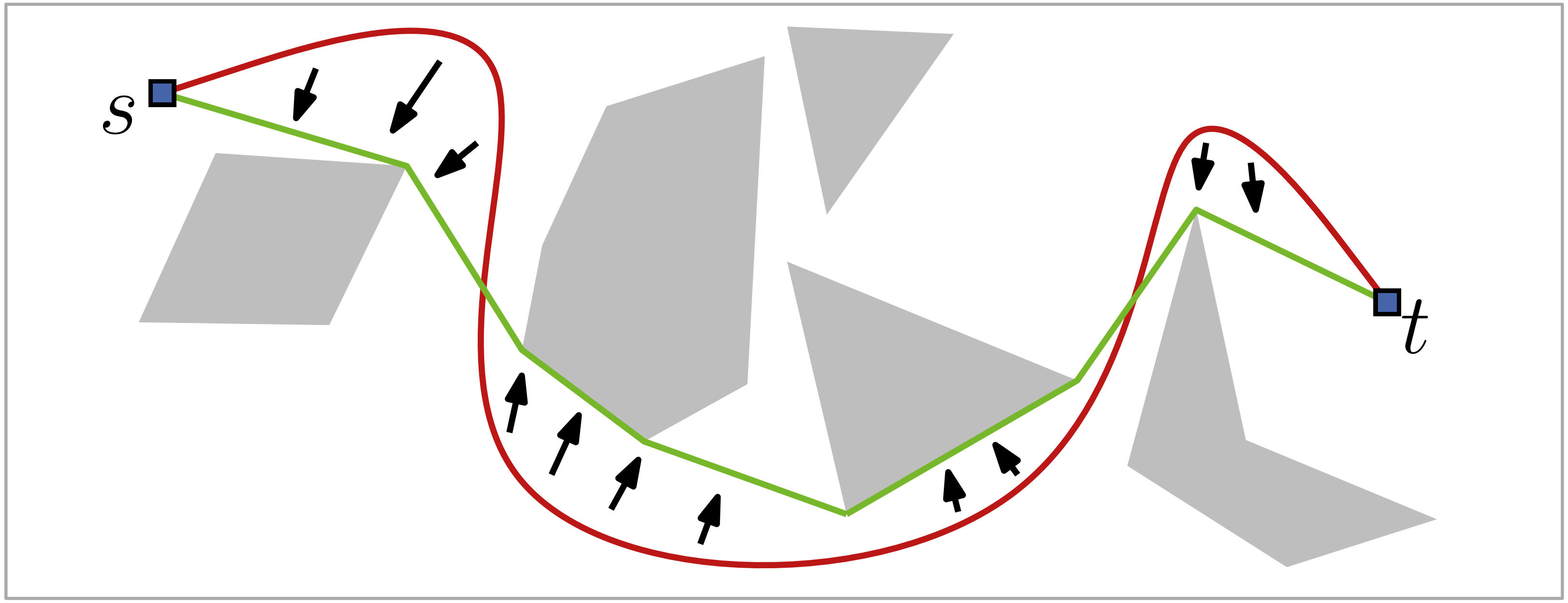


a) Zeige, dass #Segmente eines kürzesten Wegs in $O(n)$ ist.

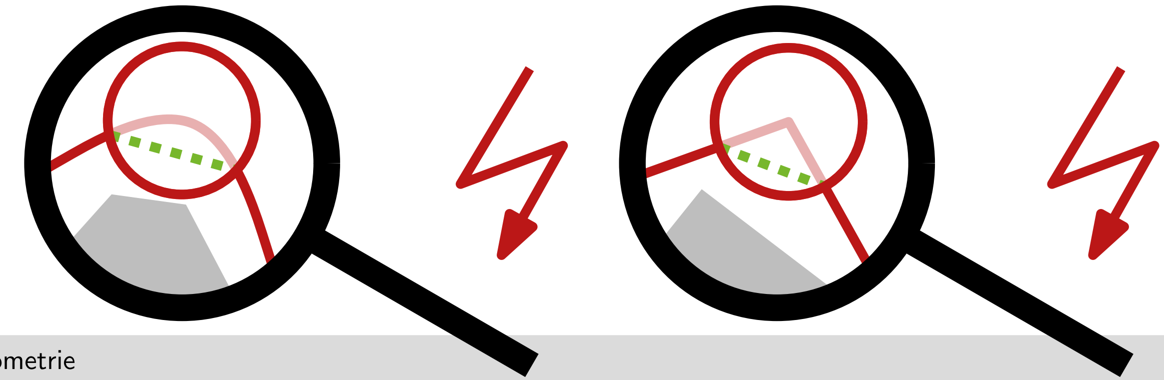
b) Beispiel für $\Theta(n)$

Aufgabe 1

Lemma 1: Für eine Menge S von disjunkten Polygonen in $\mathbb{R}^2$ und zwei Punkte s und t außerhalb S ist jeder kürzeste st-Weg in $\mathbb{R}^2 \setminus \bigcup S$ ein Polygonzug dessen innere Knoten Knoten von S sind.

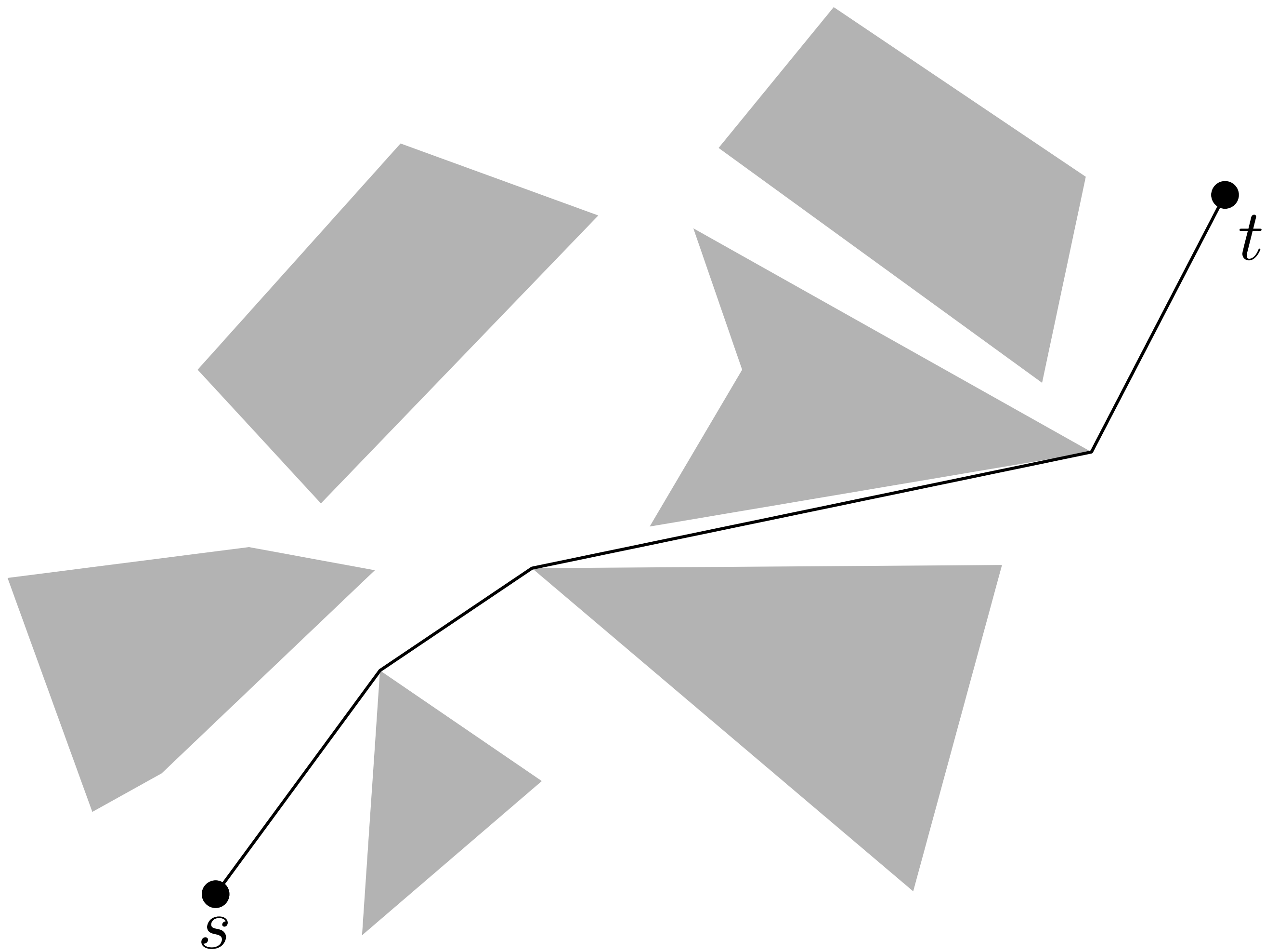


Beweisskizze:



Aufgabe 1

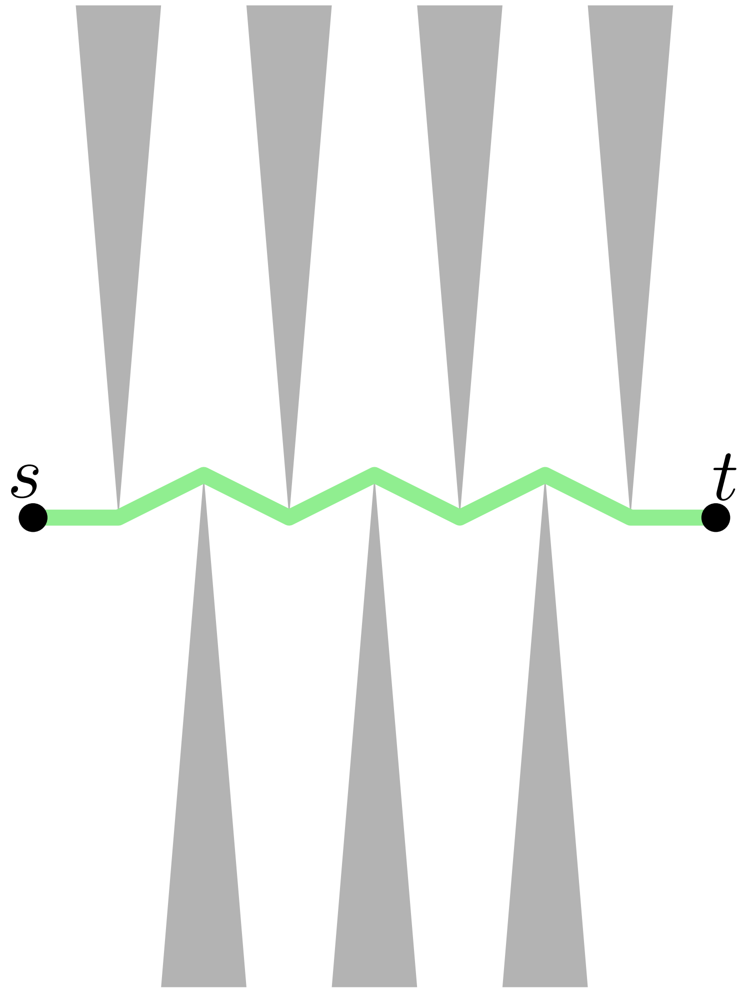
Gegeben: Menge S von Polygonen mit insgesamt n Knoten.



a) Zeige, dass #Segmente eines kürzesten Wegs in $O(n)$ ist.

b) Beispiel für $\Theta(n)$

Aufgabe 1



b) Beispiel für $\Theta(n)$

Aufgabe 2

$\textsc{VisibilityGraph}(S)$

Input: Menge disjunkter Polygone S

Output: Sichtbarkeitsgraph $G_{\mathsf{vis}}(S)$

$E \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

  $W \leftarrow \textsc{VisibleVertices}(v, S)$

  $E \leftarrow E \cup \{vw \mid w \in W\}$

return E

Aufgabe 2

$\textsc{VisibilityGraph}(S)$

Input: Menge disjunkter Polygone S

Output: Sichtbarkeitsgraph $G_{\mathsf{vis}}(S)$

$E \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

  $W \leftarrow \textsc{VisibleVertices}(v, S)$

  $E \leftarrow E \cup \{vw \mid w \in W\}$

return E

Beobachtung: $\textsc{VisibleVertices}$ wird n mal aufgerufen

Aufgabe 2

$\textsc{VisibilityGraph}(S)$

Input: Menge disjunkter Polygone S

Output: Sichtbarkeitsgraph $G_{\text{vis}}(S)$

```
E ← ∅
foreach v ∈ V(S) do
    W ← VisibleVertices(v, S)
    E ← E ∪ {vw | w ∈ W}
return E
```

Beobachtung: $\textsc{VisibleVertices}$ wird n mal aufgerufen
$\textsc{VisibleVertices}$ in $O(n \log n)$

Aufgabe 2

$\textsc{VisibilityGraph}(S)$

Input: Menge disjunkter Polygone S

Output: Sichtbarkeitsgraph $G_{\mathsf{vis}}(S)$

$E \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

  $W \leftarrow \textsc{VisibleVertices}(v, S)$

  $E \leftarrow E \cup \{vw \mid w \in W\}$

return E

Beobachtung: $\textsc{VisibleVertices}$ wird n mal aufgerufen

$\textsc{VisibleVertices}$ in $O(n \log n)$

Gesamtlaufzeit: $O(n^2 \log n)$

Aufgabe 2

$\textsc{VisibilityGraph}(S)$

Input: Menge disjunkter Polygone S
Output: Sichtbarkeitsgraph $G_{\text{vis}}(S)$

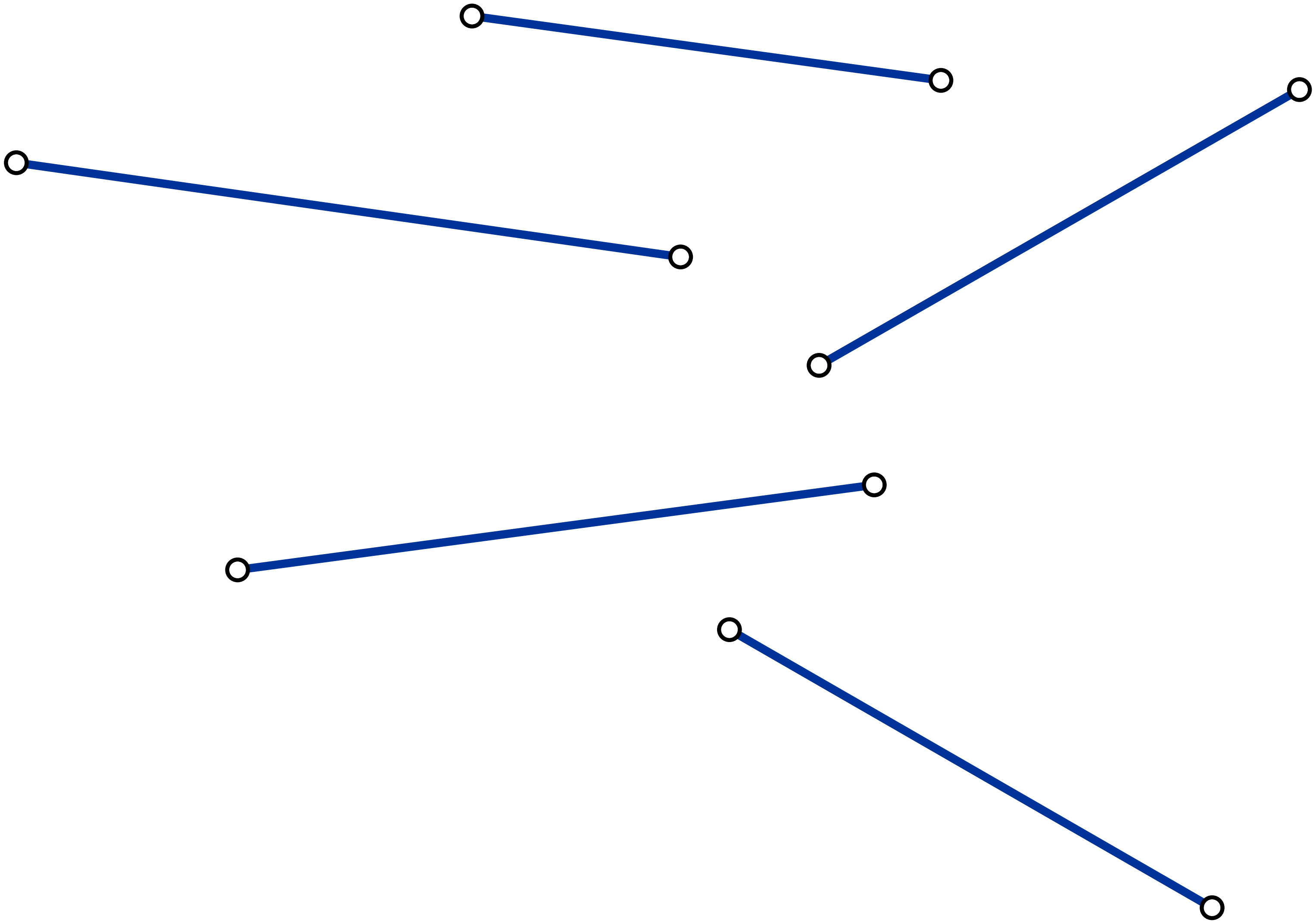
$E \leftarrow \emptyset$
foreach $v \in V(S)$ **do**
  $W \leftarrow \textsc{VisibleVertices}(v, S)$
  $E \leftarrow E \cup \{vw \mid w \in W\}$
return E

Beobachtung: $\textsc{VisibleVertices}$ wird n mal aufgerufen
$\textsc{VisibleVertices}$ in $O(n \log n)$
Gesamtlaufzeit: $O(n^2 \log n)$

Q: Wie $\textsc{VisibiltyGraph}$ in $O(n^2)$?
Hinweis: Nutze Dualität

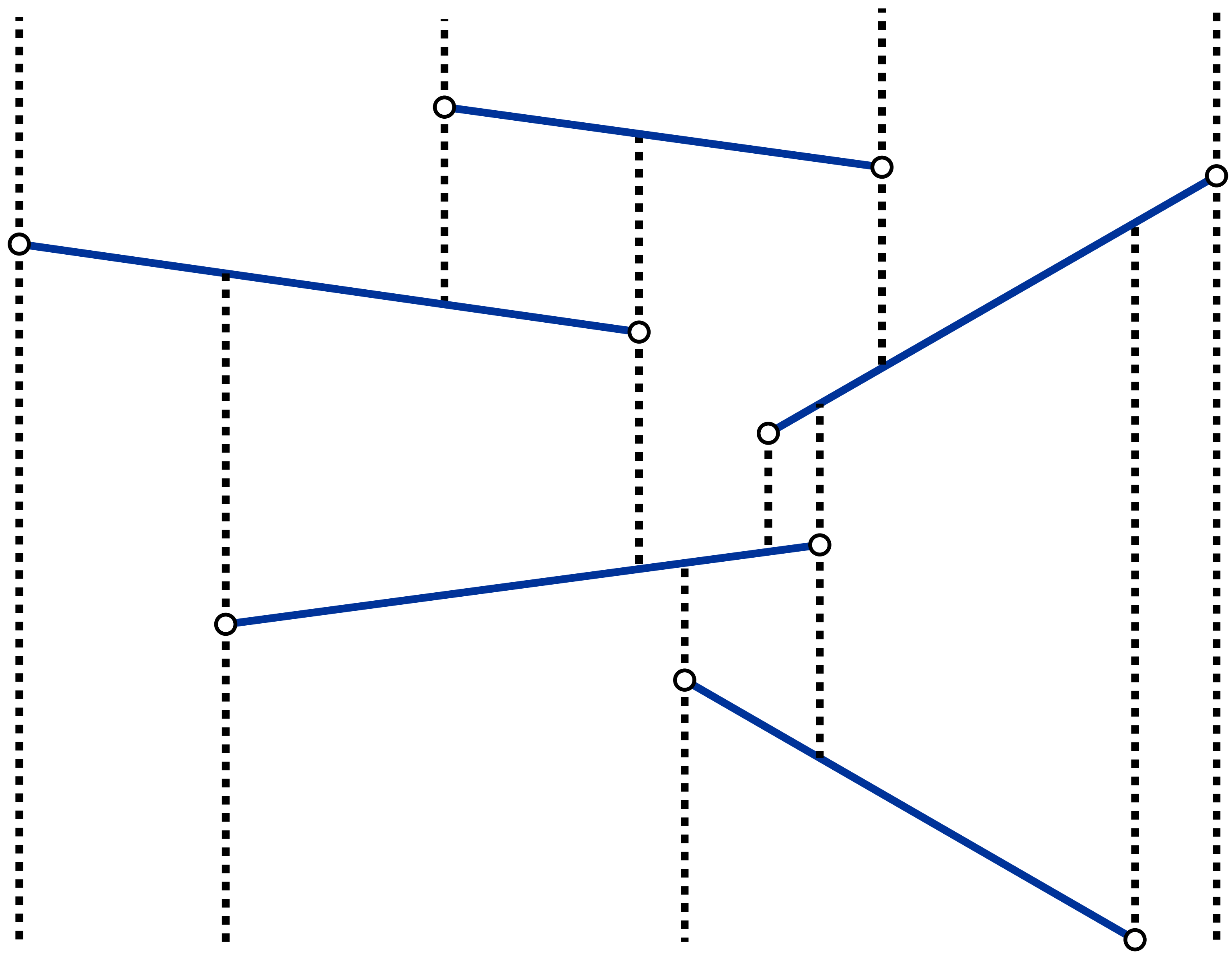
Aufgabe 2

Trapezzerlegung



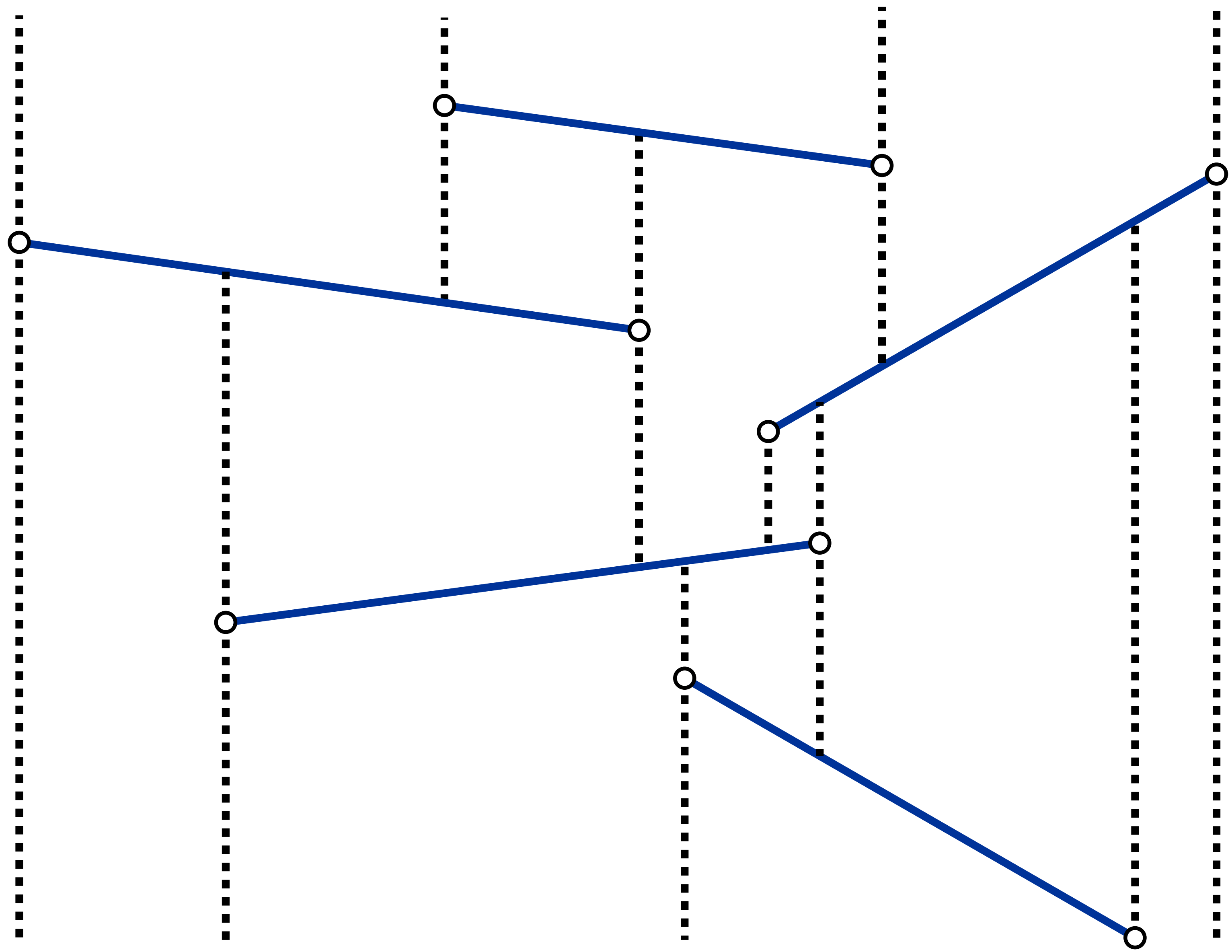
Aufgabe 2

Trapezzerlegung



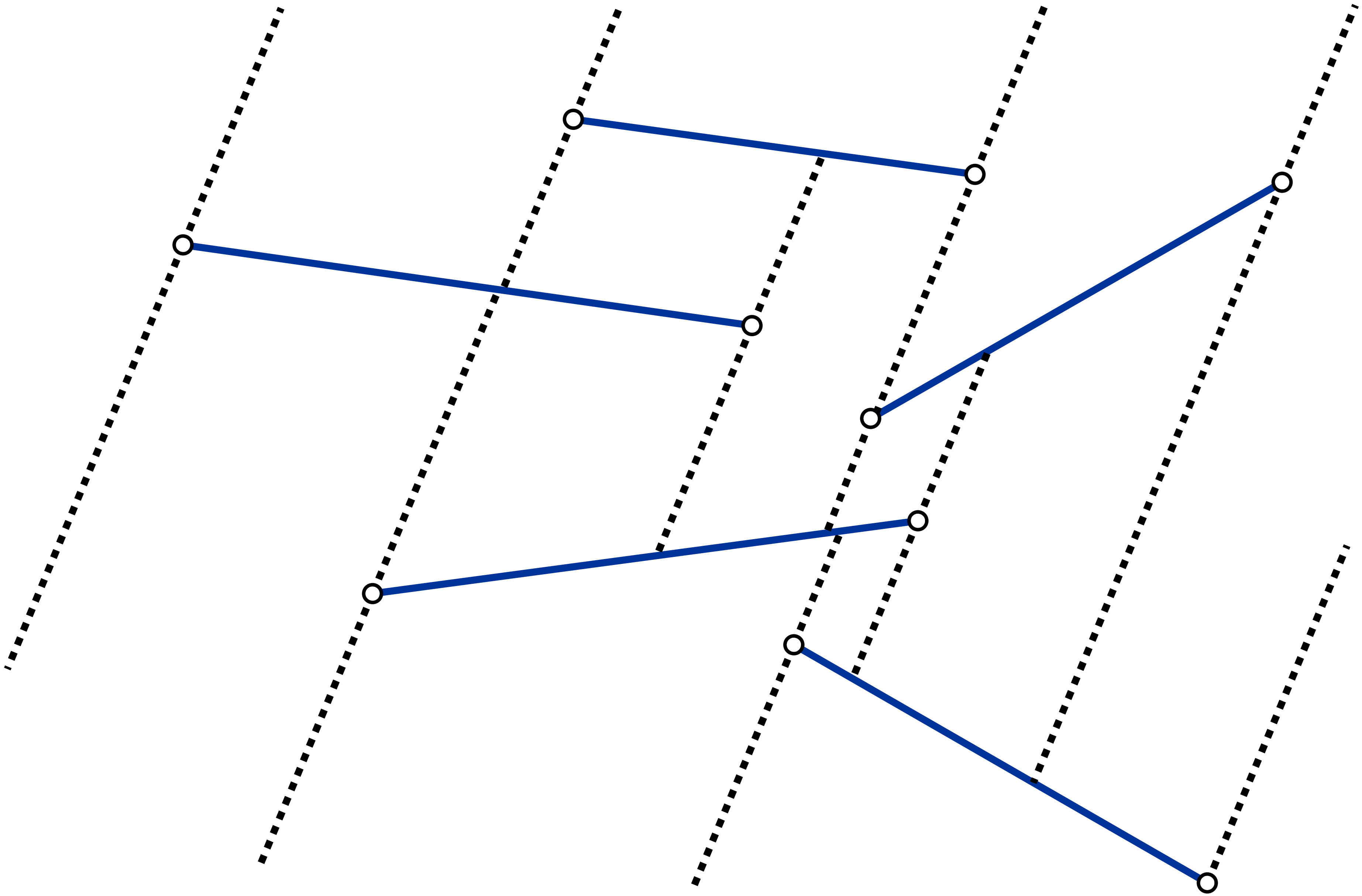
Aufgabe 2

Trapezzerlegung $\rightsquigarrow$ Sichtbarkeit Punkte zu Segmente



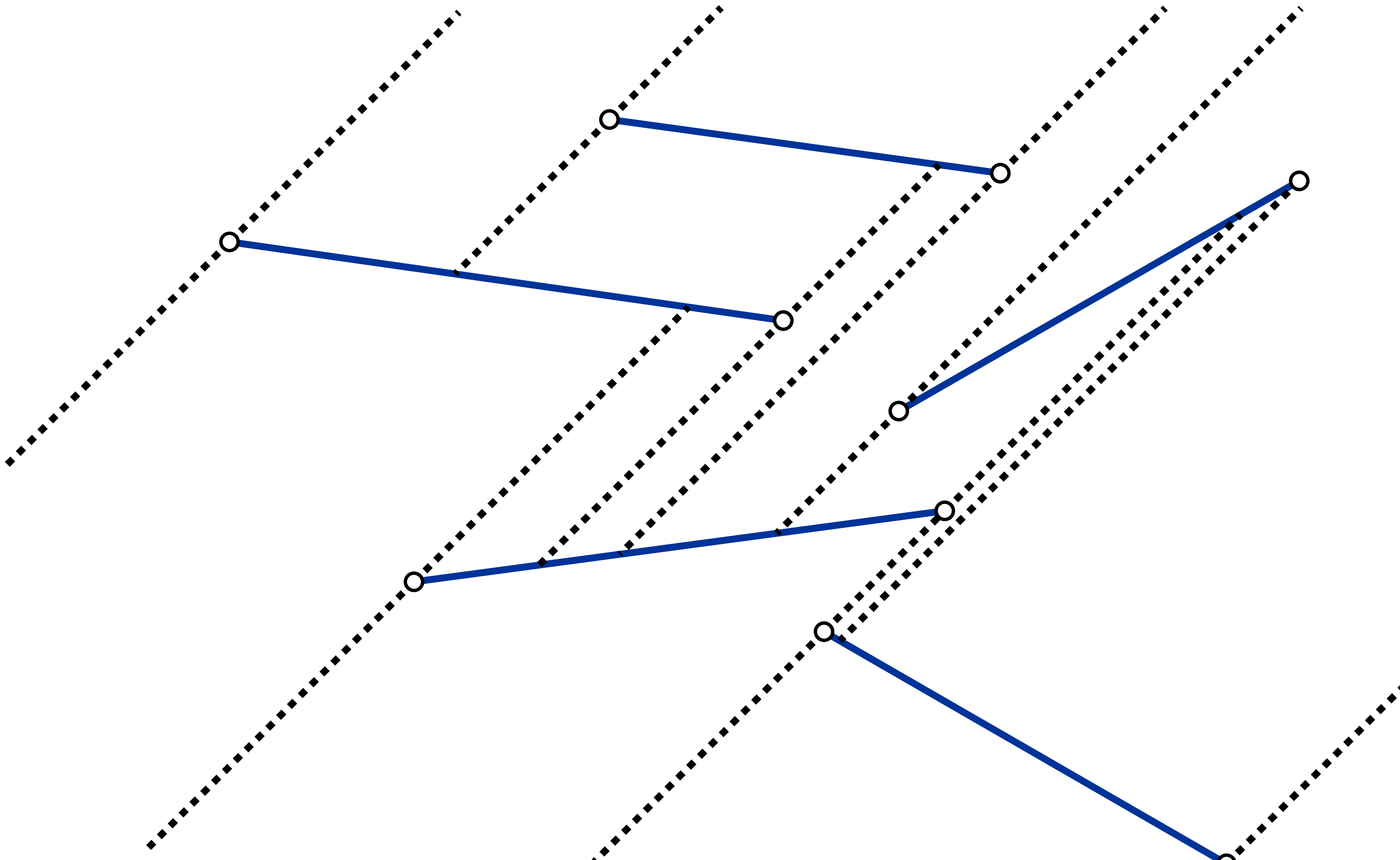
Aufgabe 2

Trapezzerlegung $\rightsquigarrow$ Sichtbarkeit Punkte zu Segmente



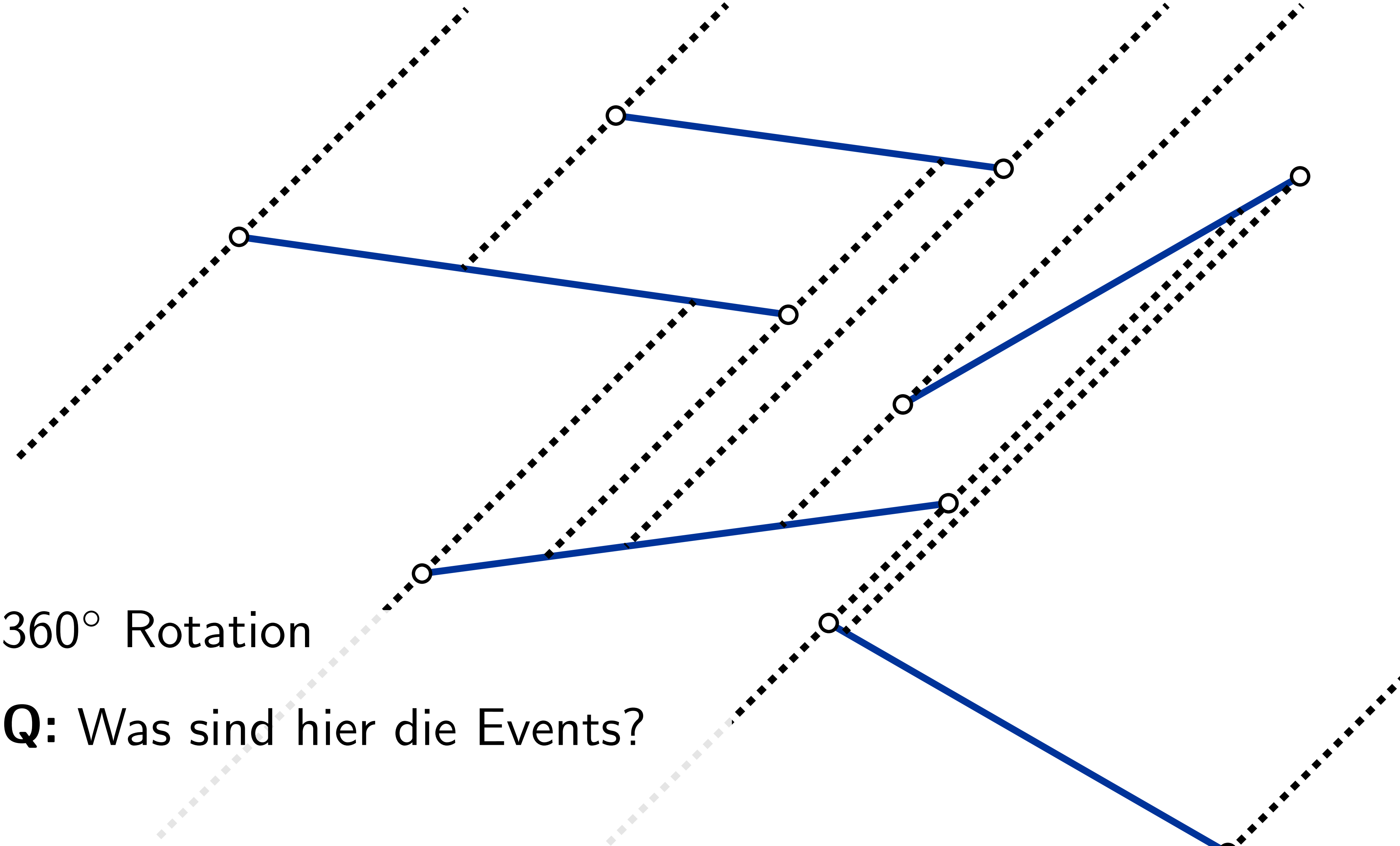
Aufgabe 2

Trapezzerlegung $\rightsquigarrow$ Sichtbarkeit Punkte zu Segmente



Aufgabe 2

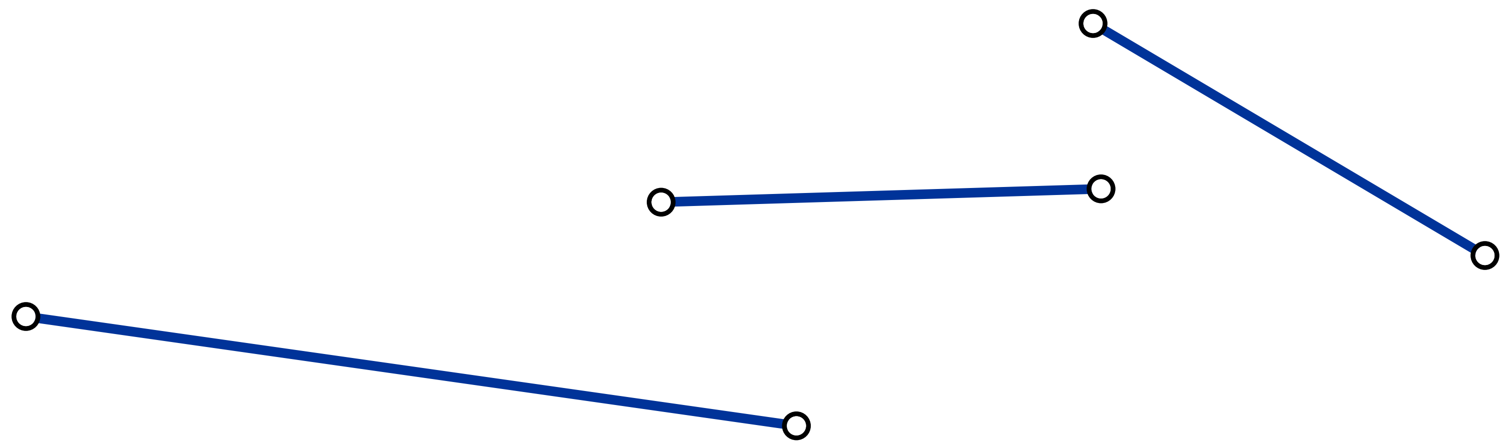
Trapezzerlegung $\rightsquigarrow$ Sichtbarkeit Punkte zu Segmente



360° Rotation

Q: Was sind hier die Events?

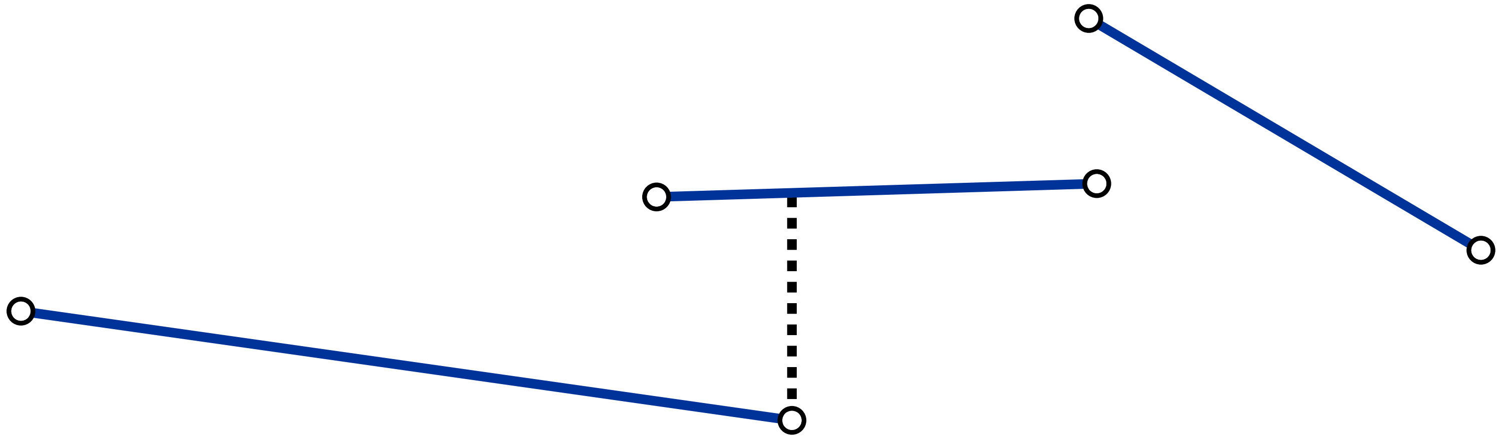
Aufgabe 2



360° Rotation

Q: Was sind hier die Events?

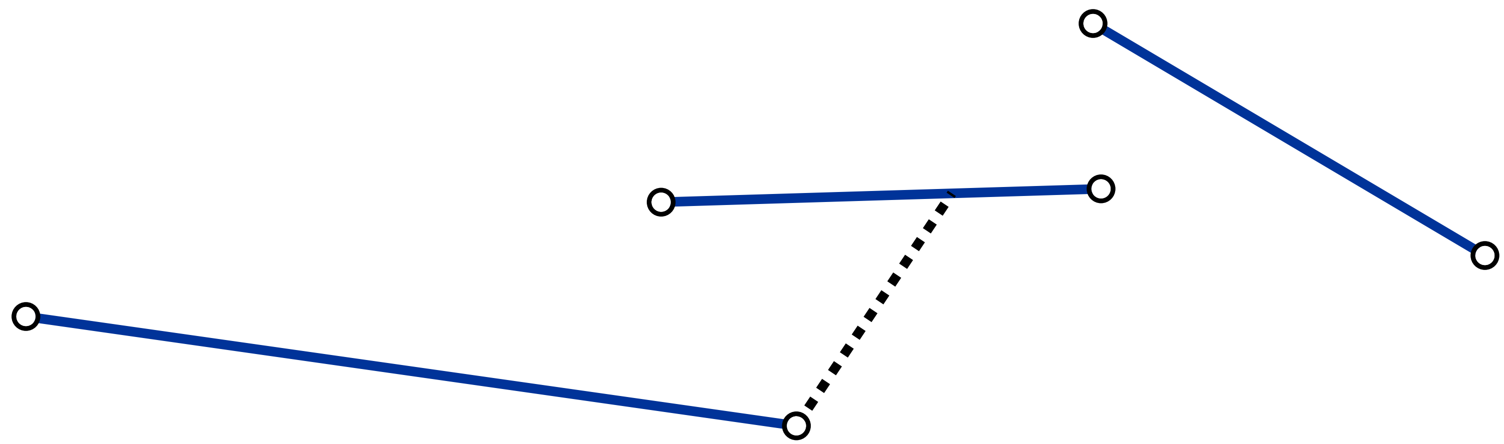
Aufgabe 2



360° Rotation

Q: Was sind hier die Events?

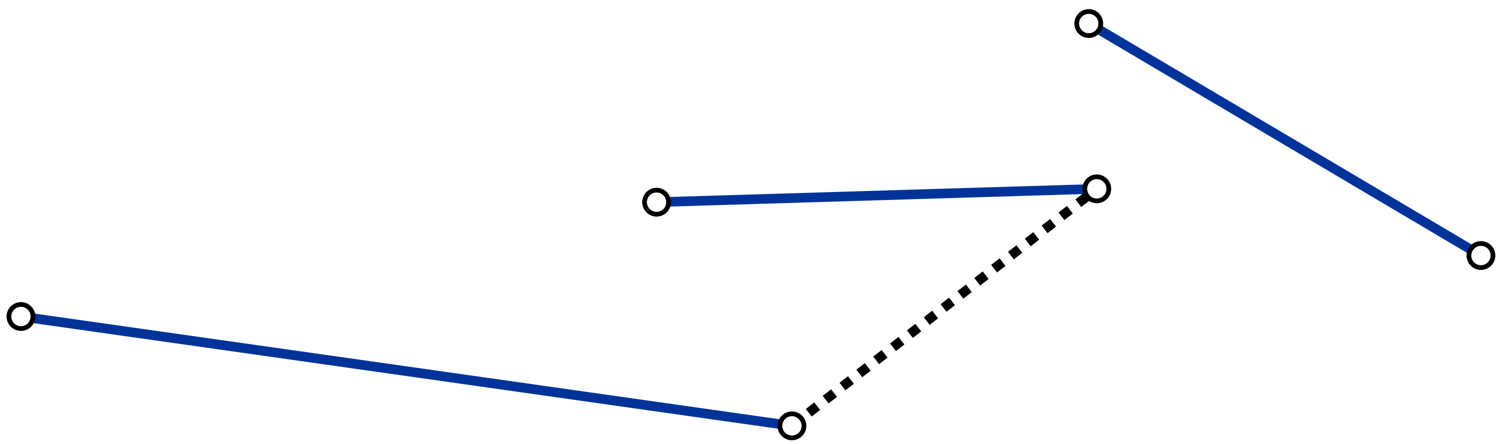
Aufgabe 2



360° Rotation

Q: Was sind hier die Events?

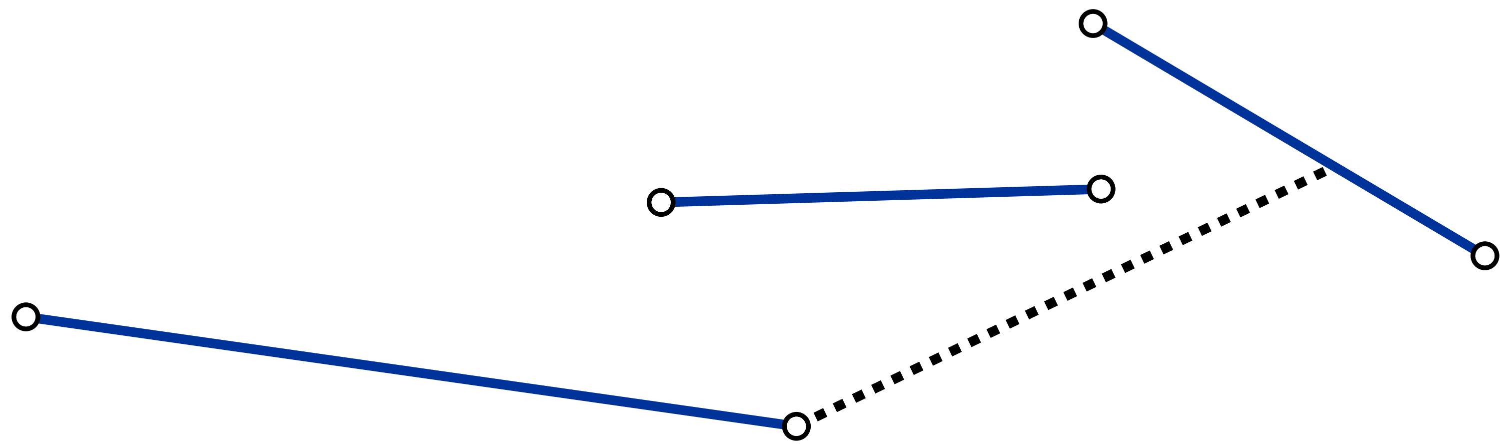
Aufgabe 2



360° Rotation

Q: Was sind hier die Events?

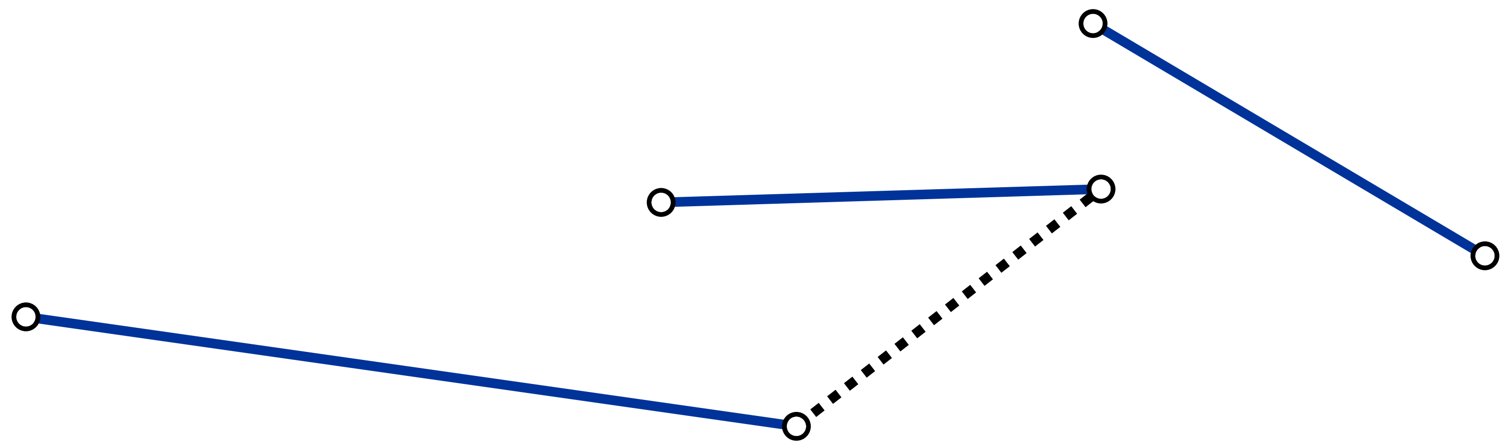
Aufgabe 2



360° Rotation

Q: Was sind hier die Events?

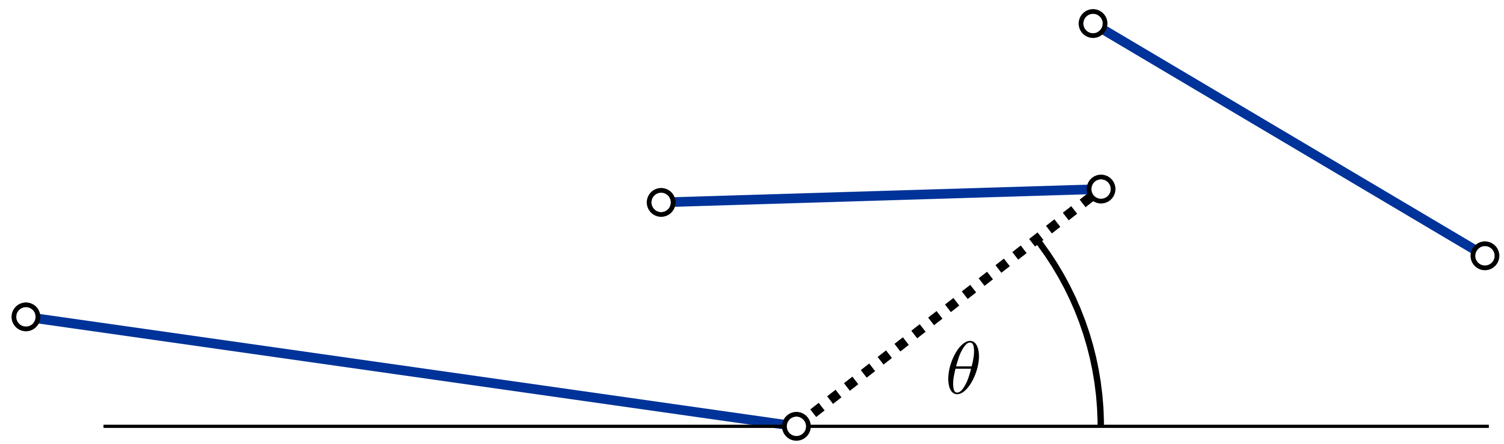
Aufgabe 2



360° Rotation

Q: Was sind hier die Events? **A:** Strahl trifft Knoten

Aufgabe 2

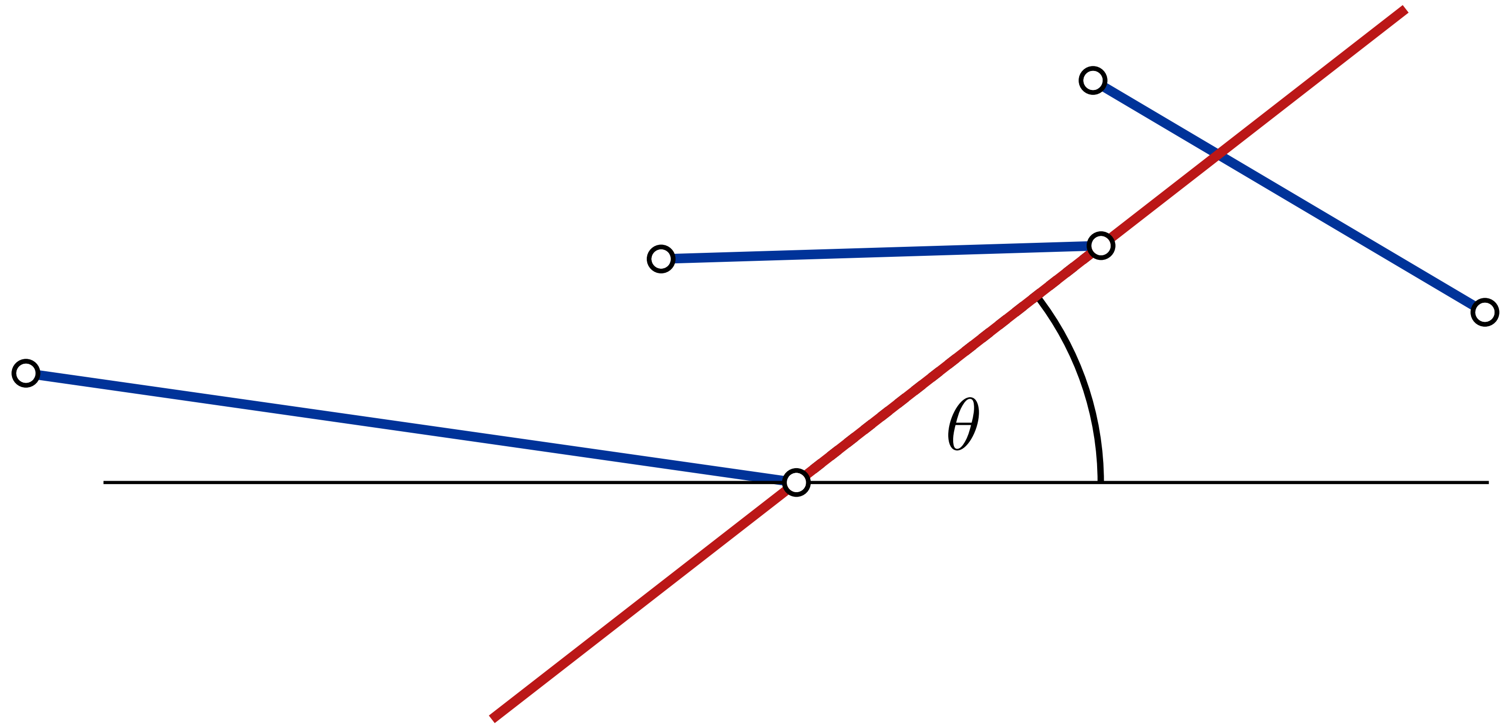


360° Rotation

Q: Was sind hier die Events?

A: Strahl trifft Knoten

Aufgabe 2

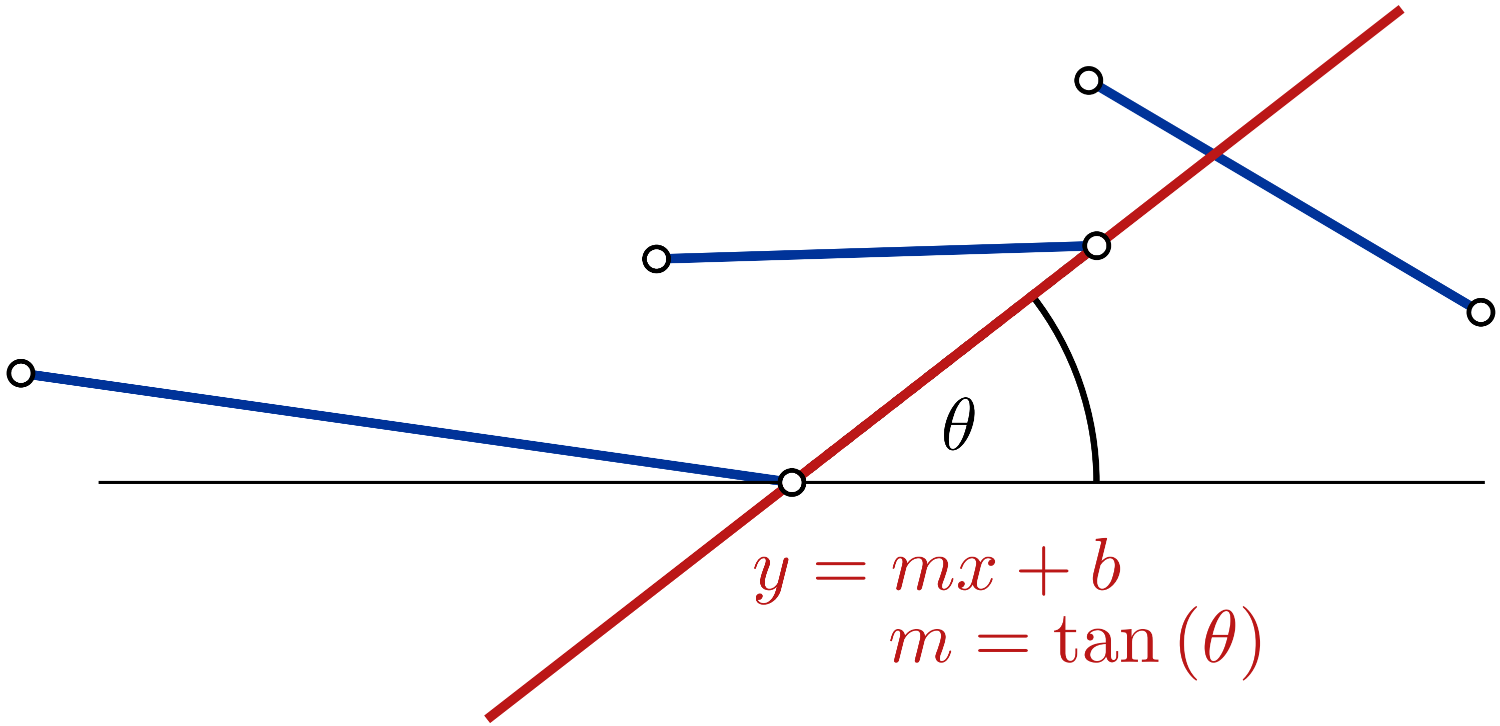


360° Rotation

Q: Was sind hier die Events?

A: Strahl trifft Knoten

Aufgabe 2



360° Rotation

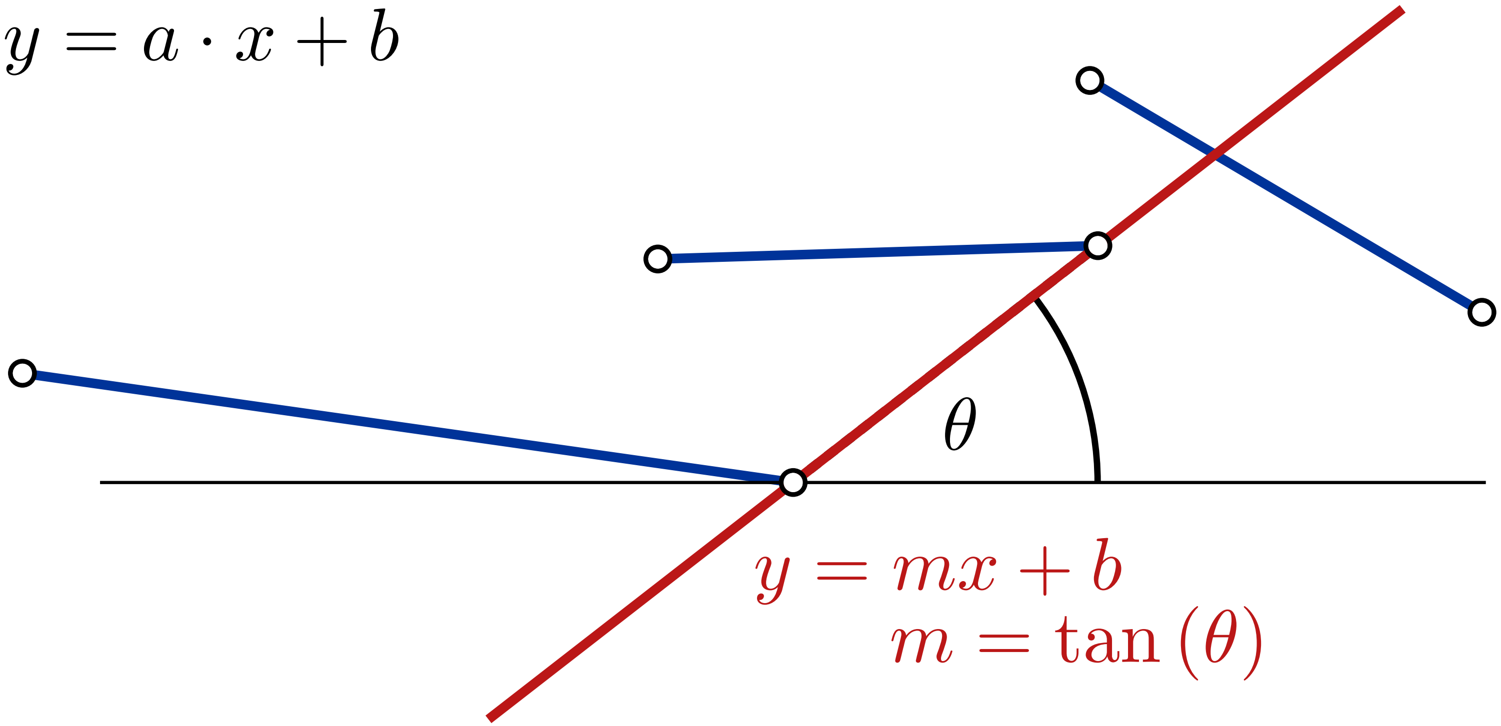
Q: Was sind hier die Events?

A: Strahl trifft Knoten

Aufgabe 2

Dualität:

$(a, b) \leftrightsquigarrow y = a \cdot x + b$



360° Rotation

Q: Was sind hier die Events?

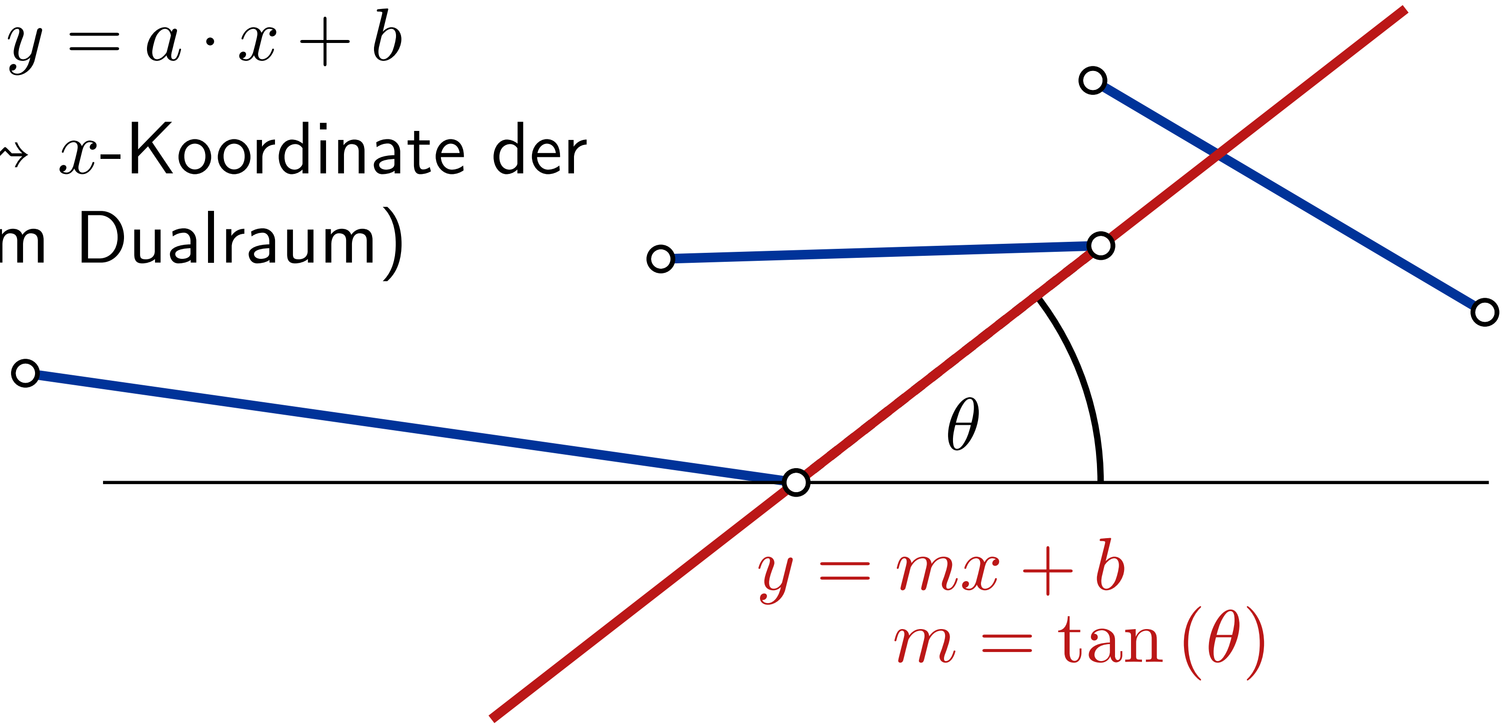
A: Strahl trifft Knoten

Aufgabe 2

Dualität:

$(a, b) \leftrightsquigarrow y = a \cdot x + b$

Events $\leftrightsquigarrow$ x-Koordinate der Punkte (im Dualraum)



360° Rotation

Q: Was sind hier die Events?

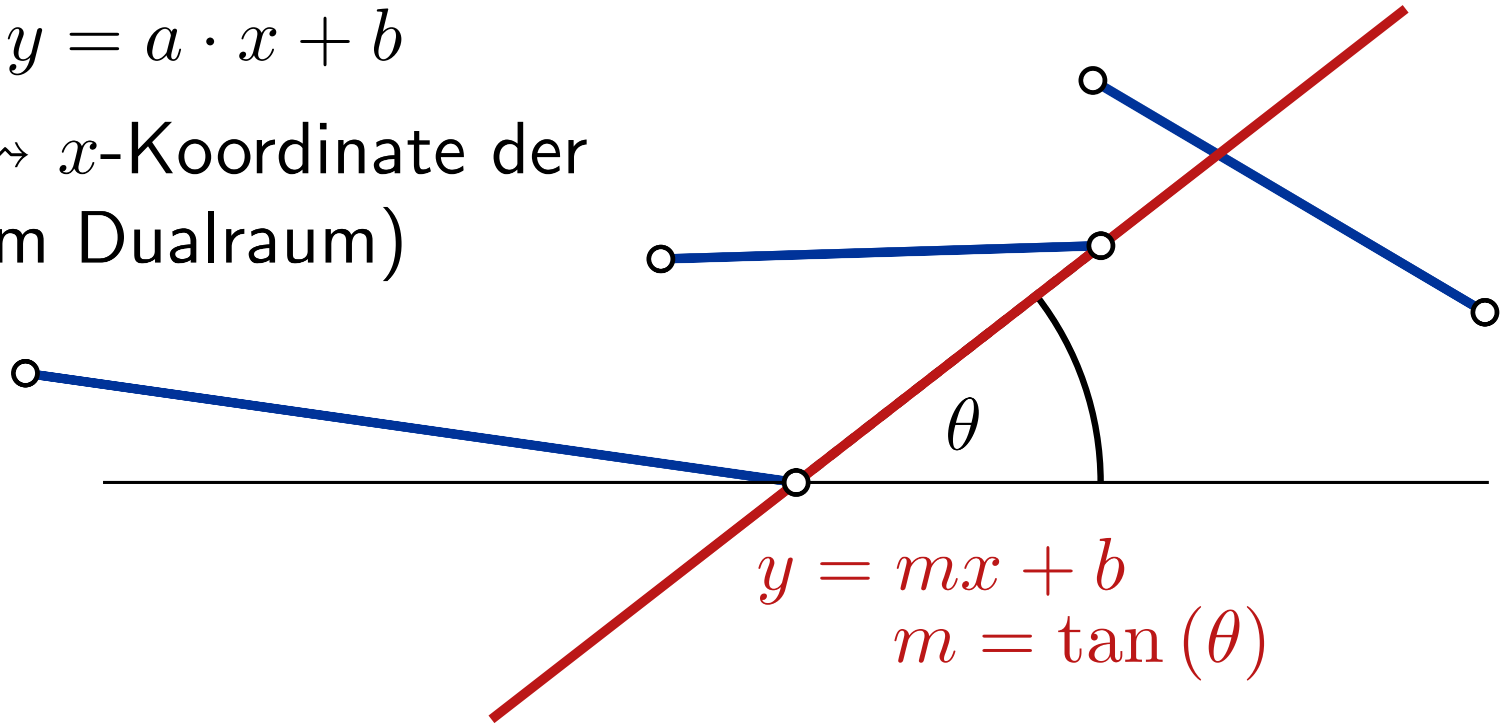
A: Strahl trifft Knoten

Aufgabe 2

Dualität:

$(a, b) \leftrightsquigarrow y = a \cdot x + b$

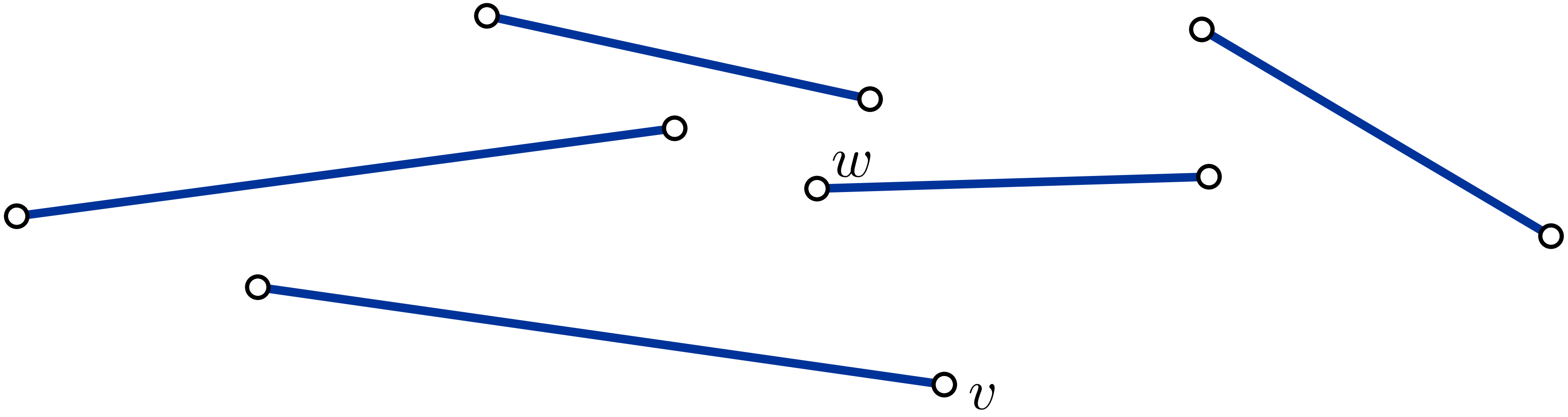
Events $\leftrightsquigarrow$ x-Koordinate der Punkte (im Dualraum)



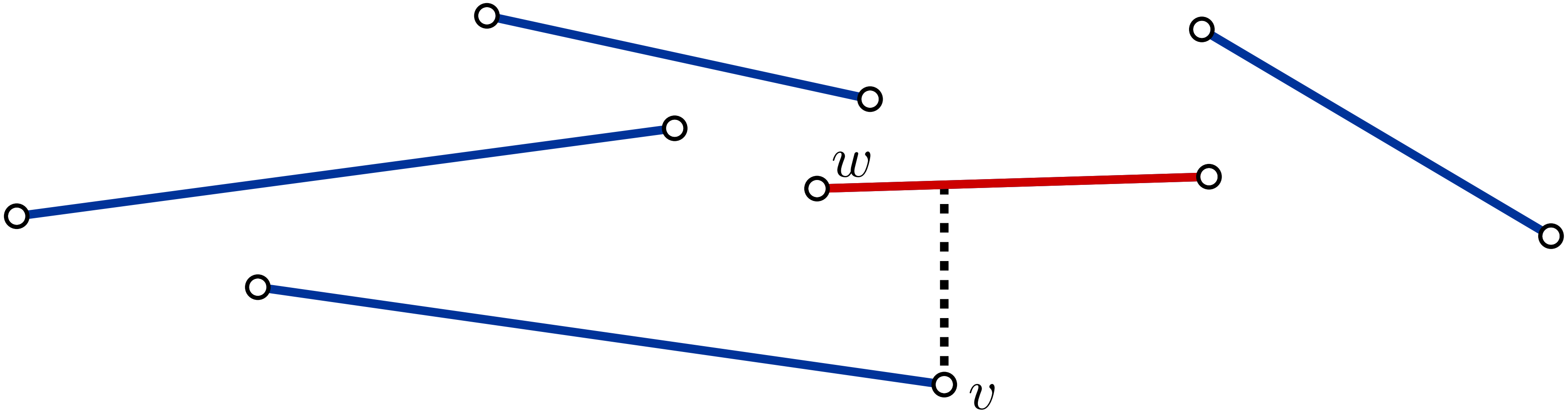
Vorgehen:

- Wandle alle Knoten der Polygone in Geraden um (Dualität)
- Sweepe von links nach rechts durch das Geradenarrangement
- Beim Sweep merke die Schnittpunkte der Geraden
- Sweep liefert Events sortiert nach Winkel in $O(n^2)$

Aufgabe 2



Aufgabe 2

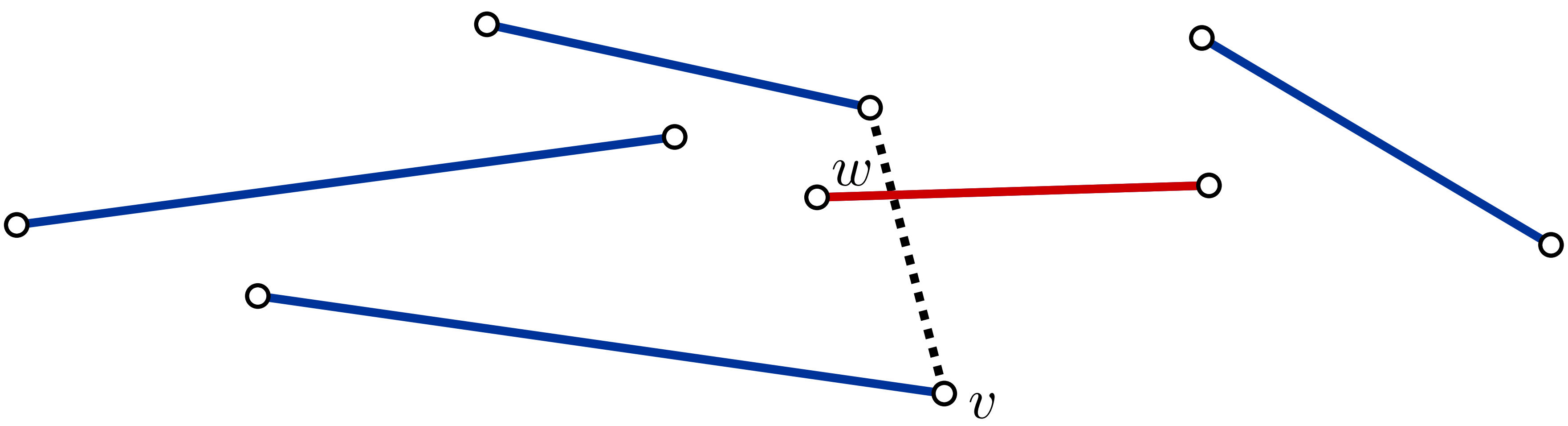


- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar

Aufgabe 2

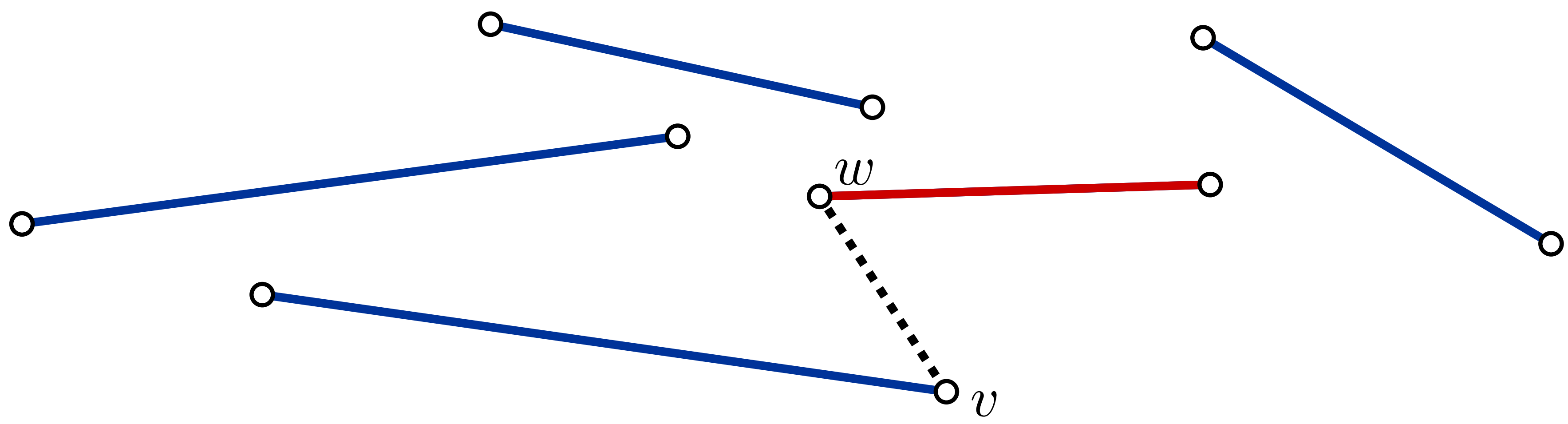


- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar

Aufgabe 2

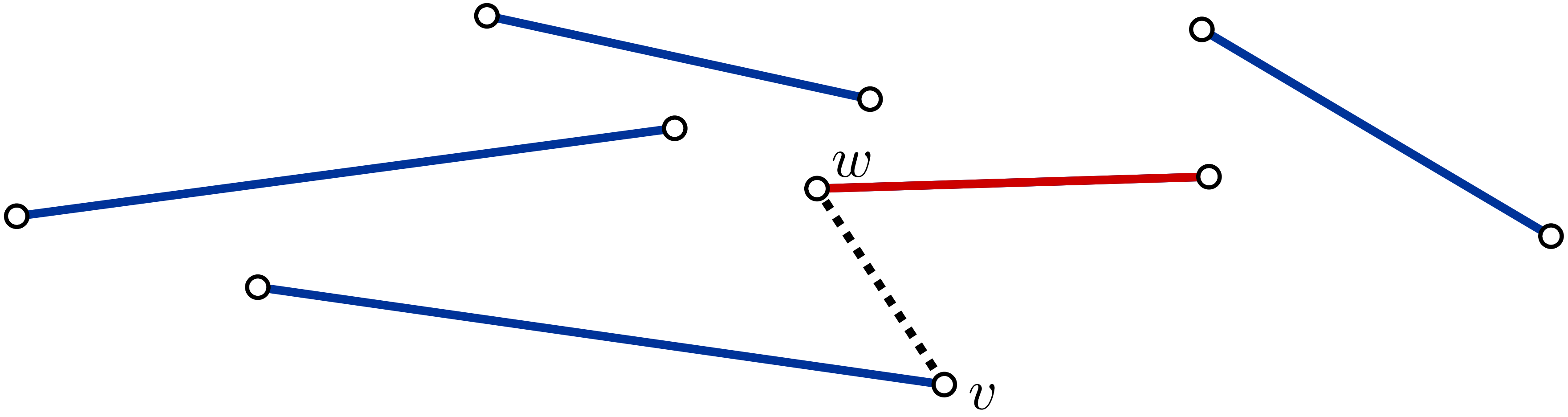


- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar
- Verlassen eines Segments

Aufgabe 2



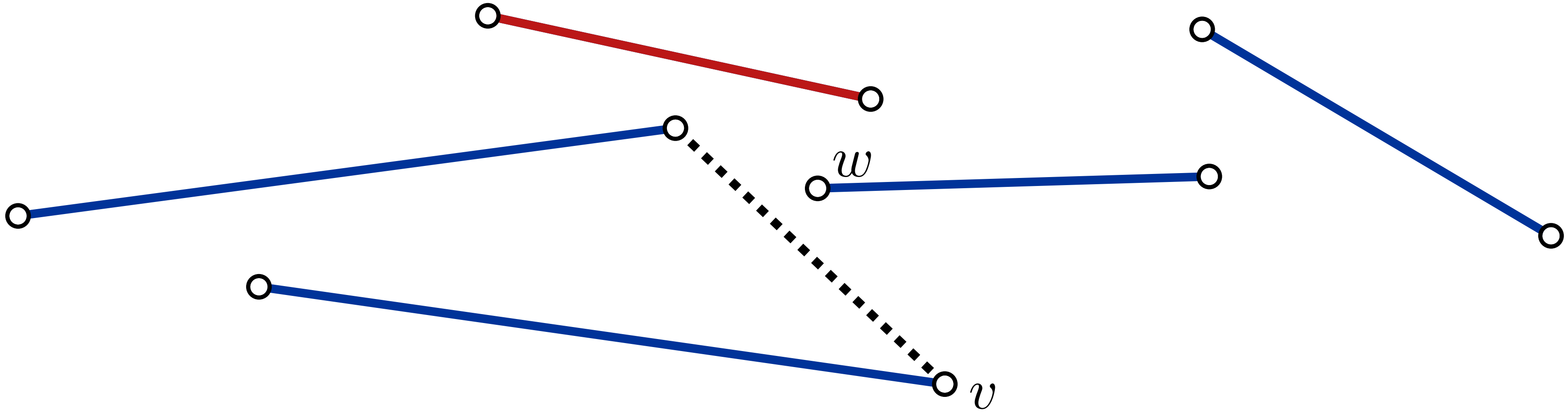
- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar
- Verlassen eines Segments

Wie $f(v)$ updaten?

Aufgabe 2

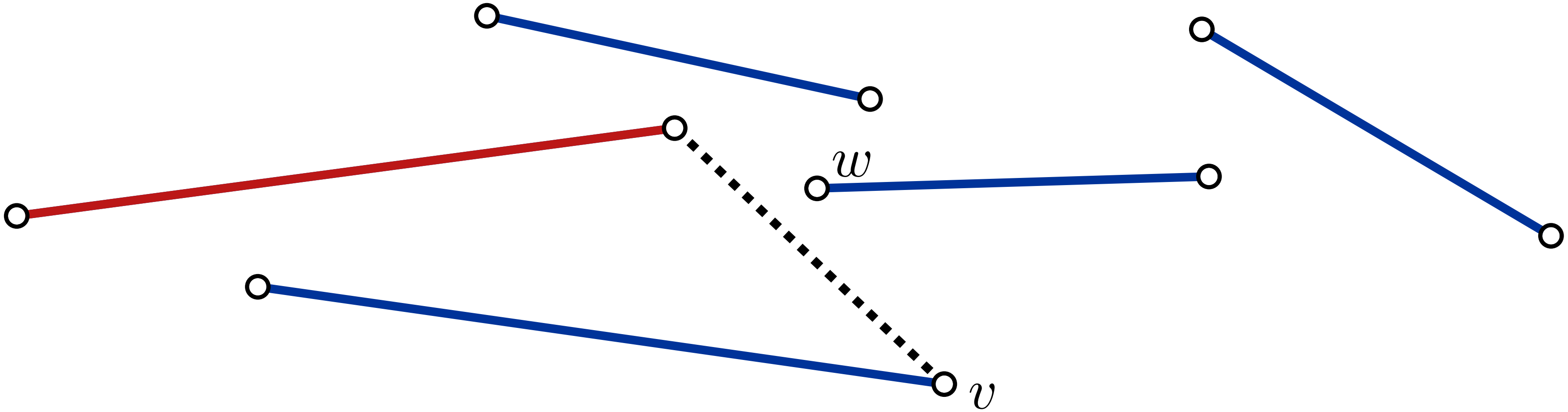


- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar
- Verlassen eines Segments
- Betreten eines Segments

Aufgabe 2

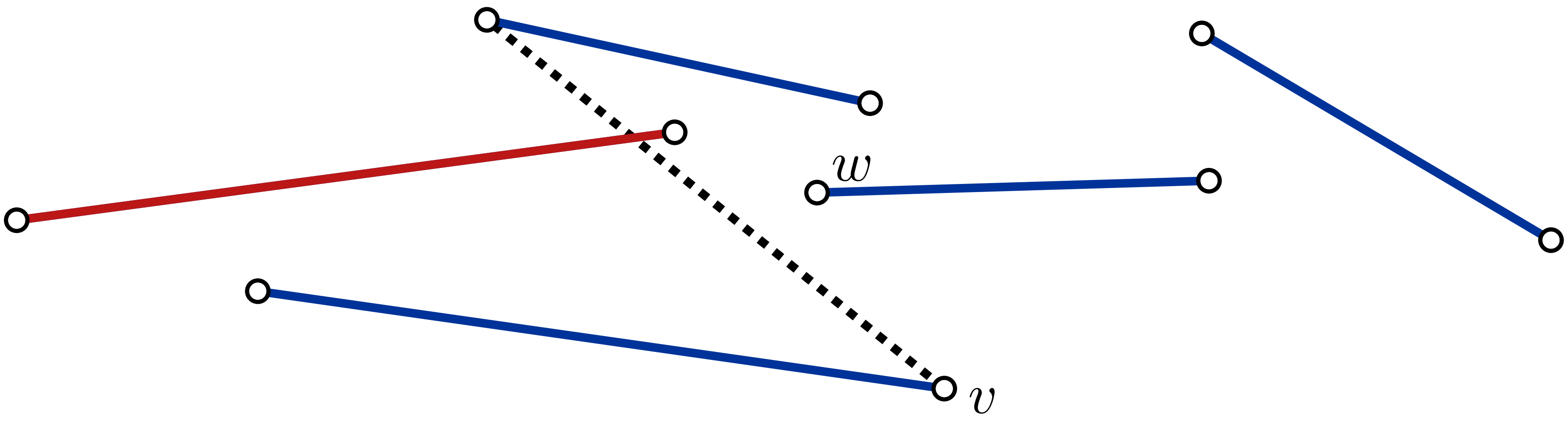


- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar
- Verlassen eines Segments
- Betreten eines Segments

Aufgabe 2

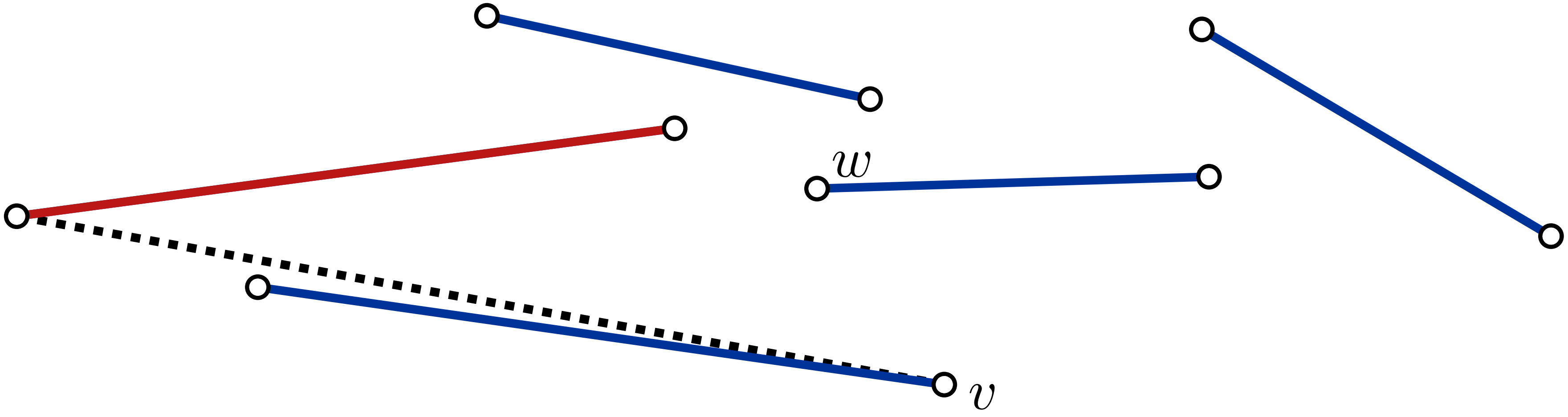


- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar
- Verlassen eines Segments
- Betreten eines Segments

Aufgabe 2

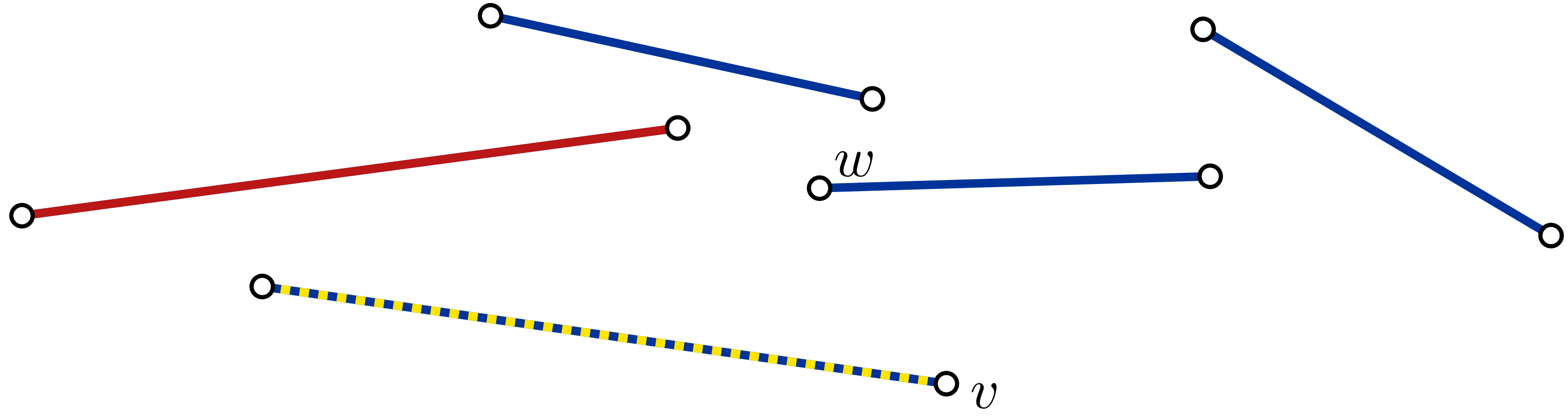


- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar
- Verlassen eines Segments
- Betreten eines Segments

Aufgabe 2

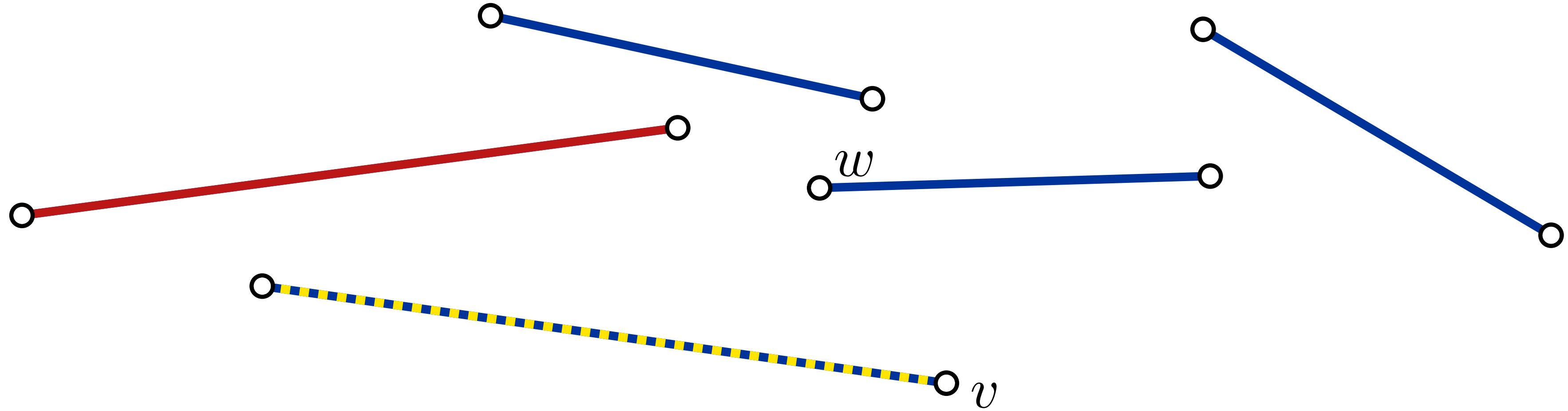


- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar
- Verlassen eines Segments
- Betreten eines Segments
- Gleiches Segment

Aufgabe 2



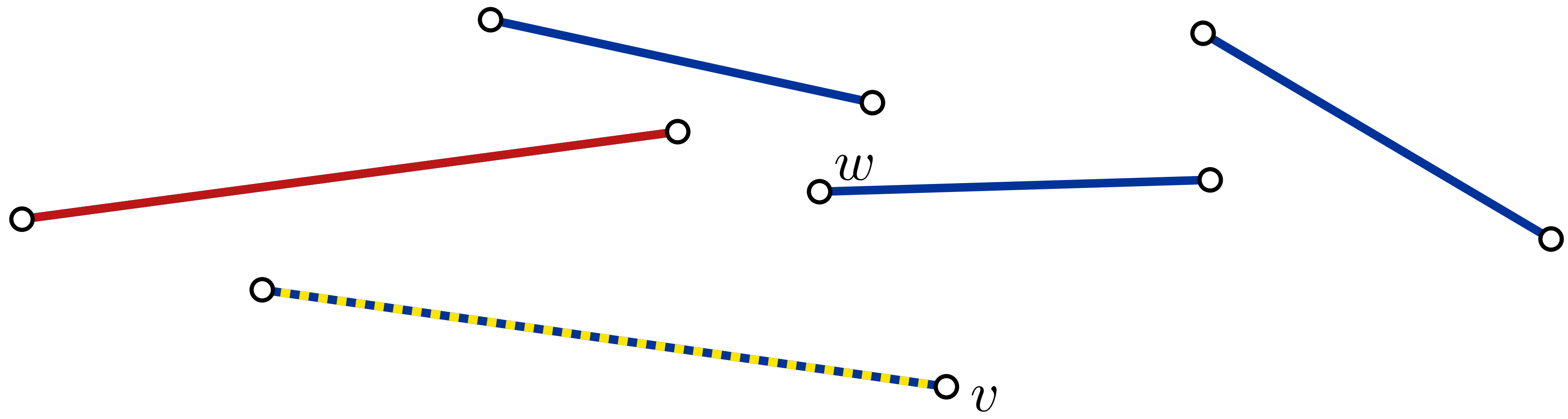
- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

Fälle:

- Unsichtbar
- Verlassen eines Segments
- Betreten eines Segments
- Gleiches Segment

$\left.\right\}$ Füge Kante zum Sichtbarkeitsgraph hinzu

Aufgabe 2



- Bestimme Segment $f(v)$ das als erstes vom Strahl getroffen wird.

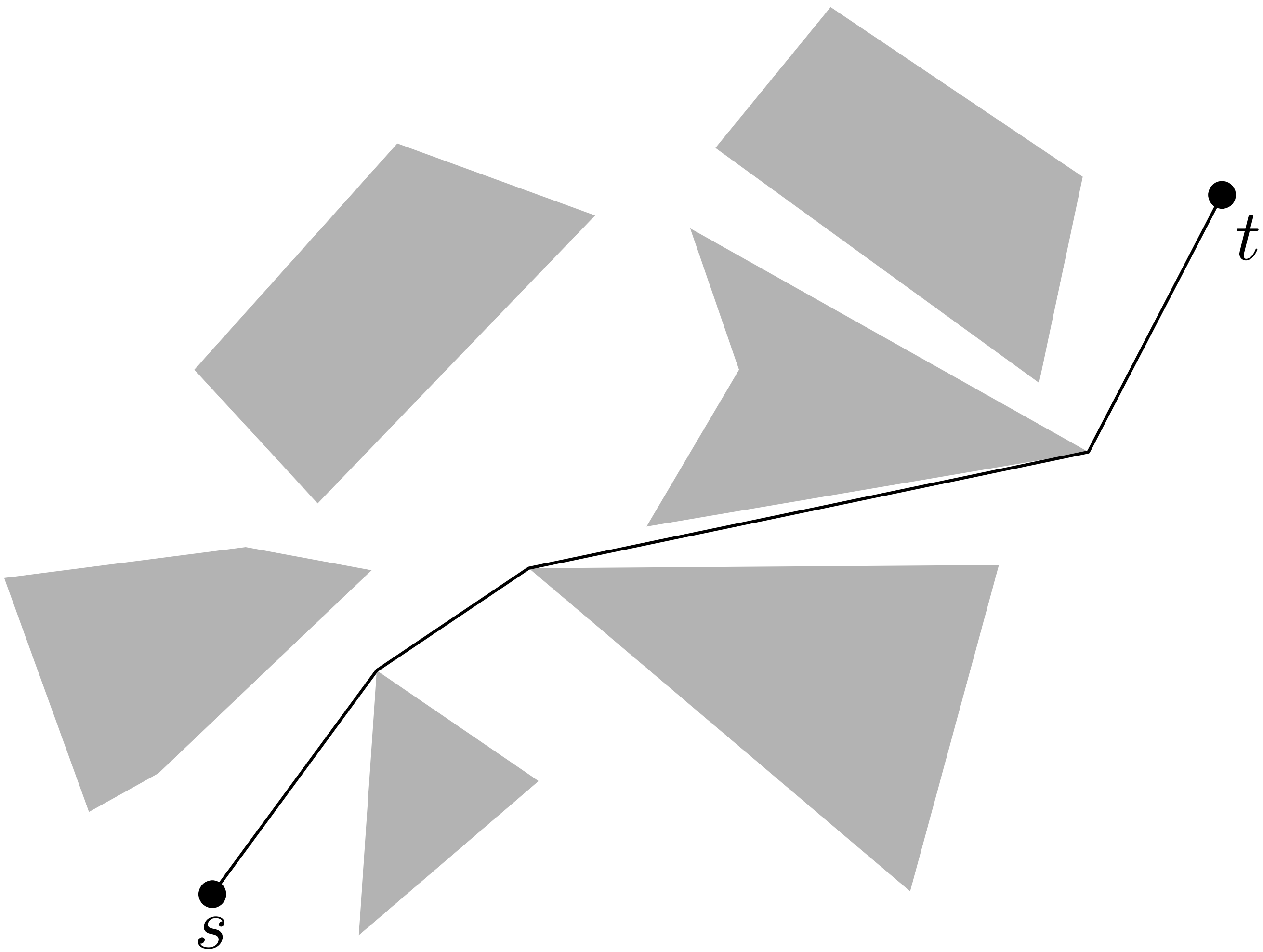
Fälle:

- Unsichtbar $O(1)$
- Verlassen eines Segments $O(1)$
- Betreten eines Segments $O(1)$
- Gleiches Segment $O(1)$

Füge Kante zum Sichtbarkeitsgraph hinzu

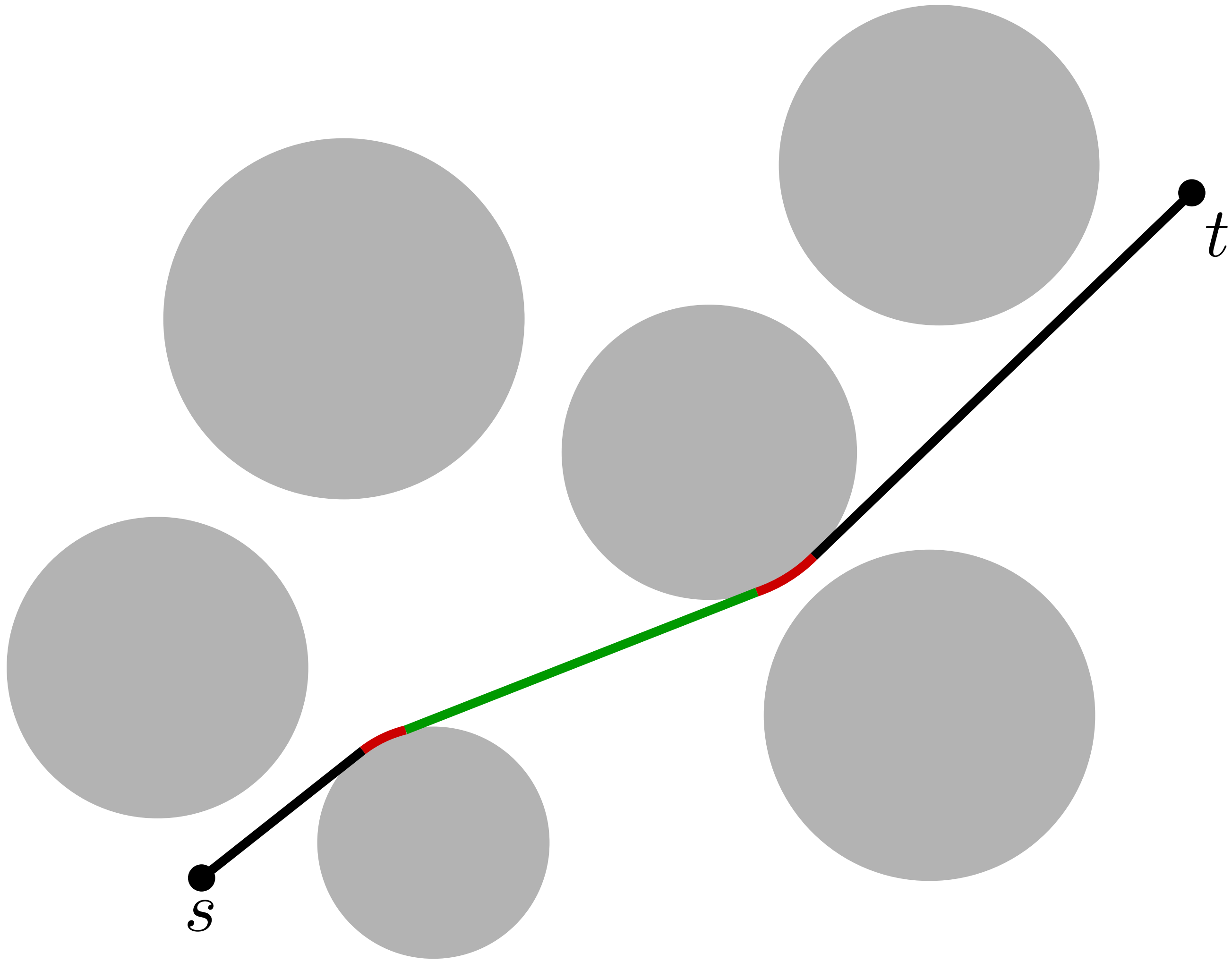
Aufgabe 3

Gegeben: Menge S von Polygonen mit insgesamt n Knoten.



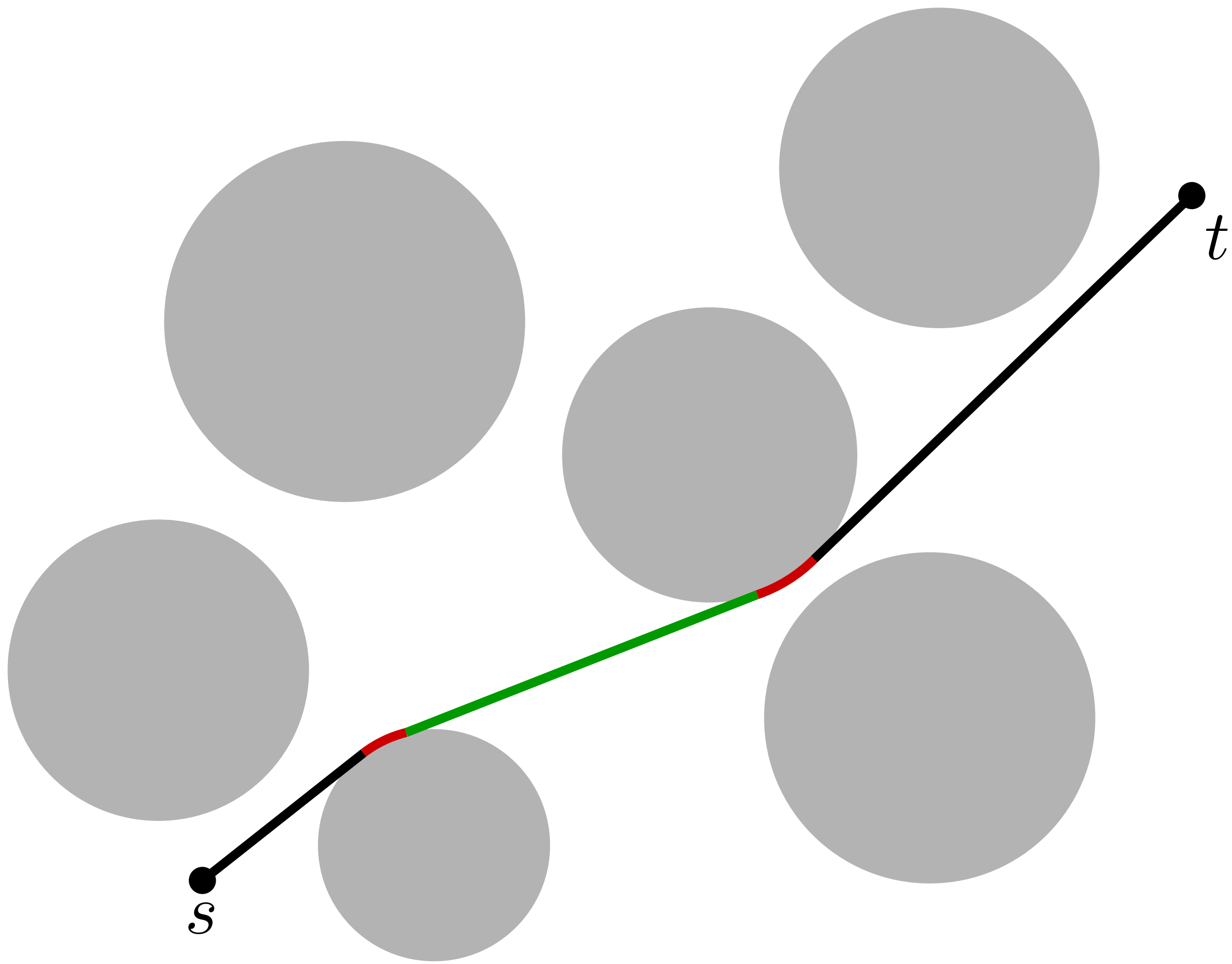
Aufgabe 3

Gegeben: Menge S von n Scheiben



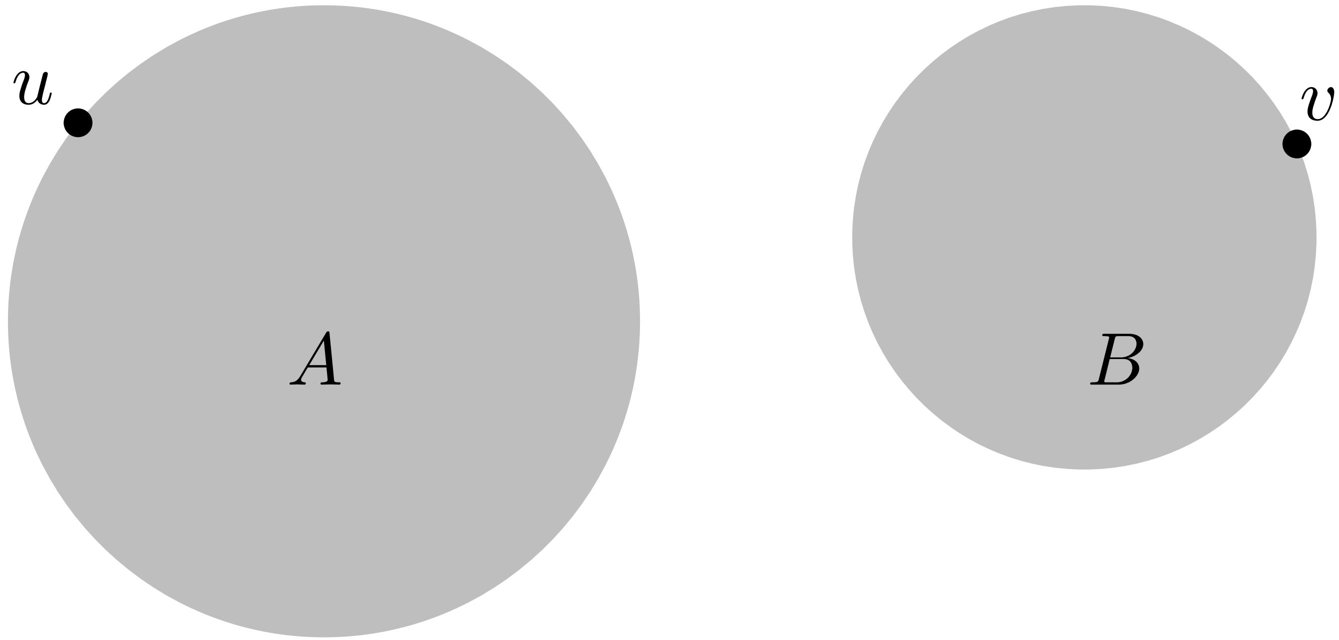
Aufgabe 3

Gegeben: Menge S von n Scheiben



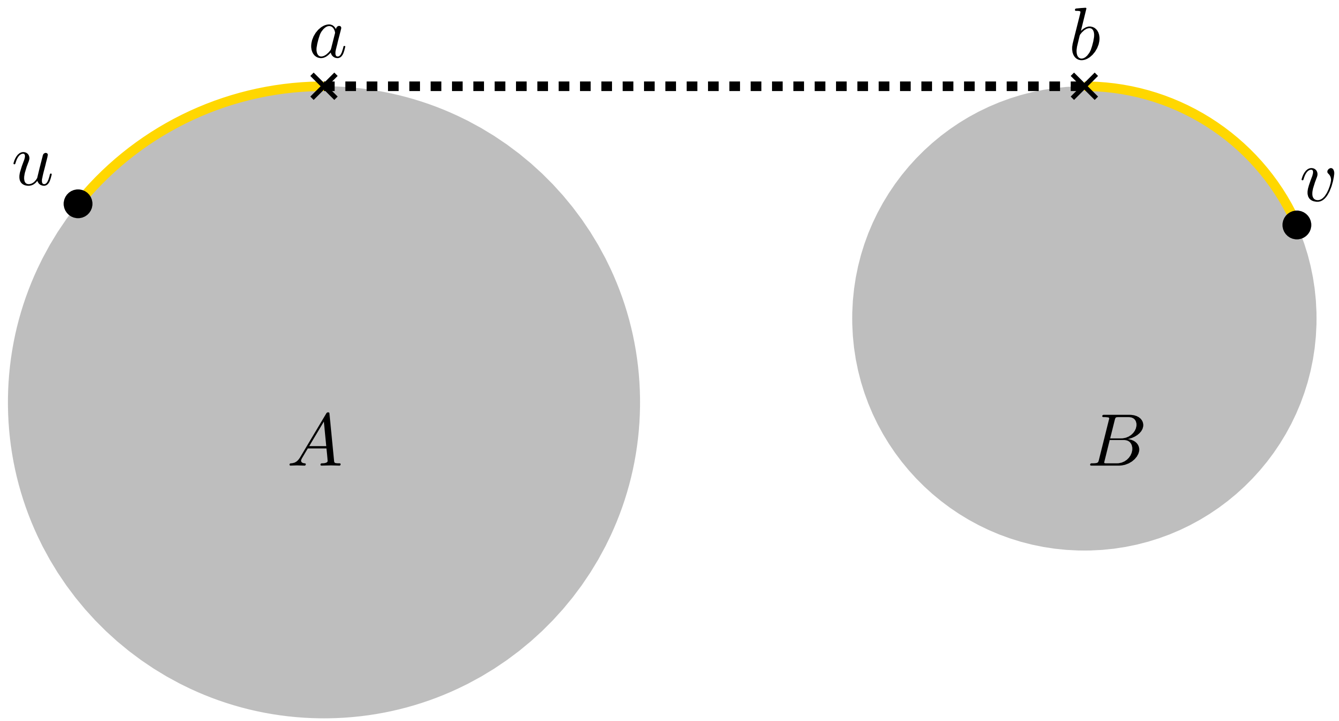
Tangente s/t – Kreis
Tangente Kreis – Kreis
Kreissegment

Aufgabe 3



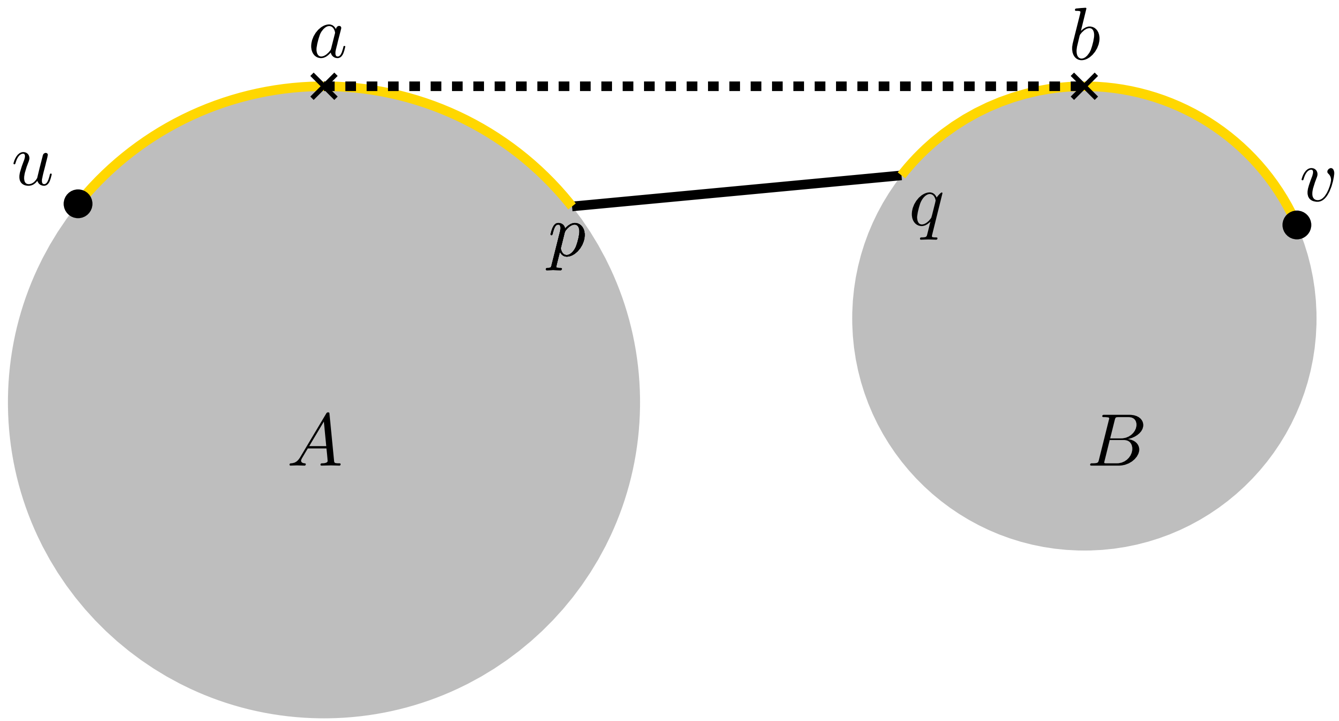
Tangente Kreis – Kreis

Aufgabe 3



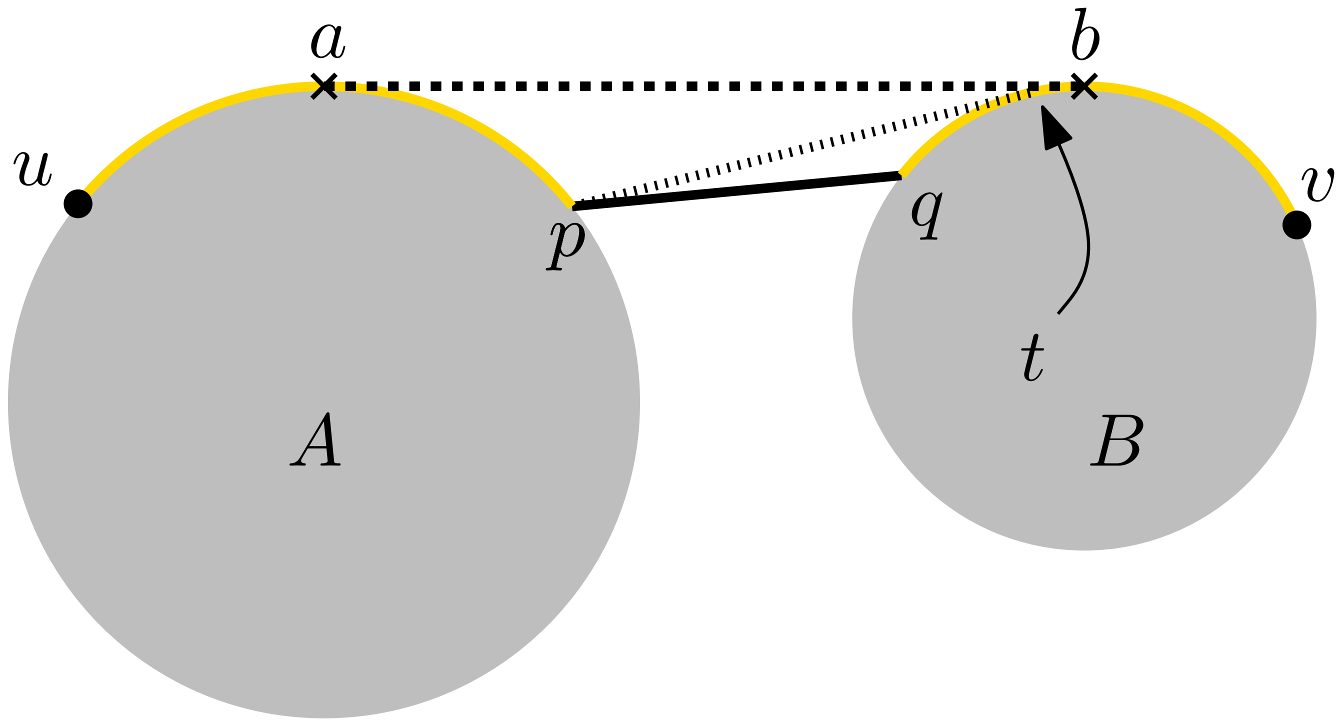
Tangente Kreis – Kreis

Aufgabe 3



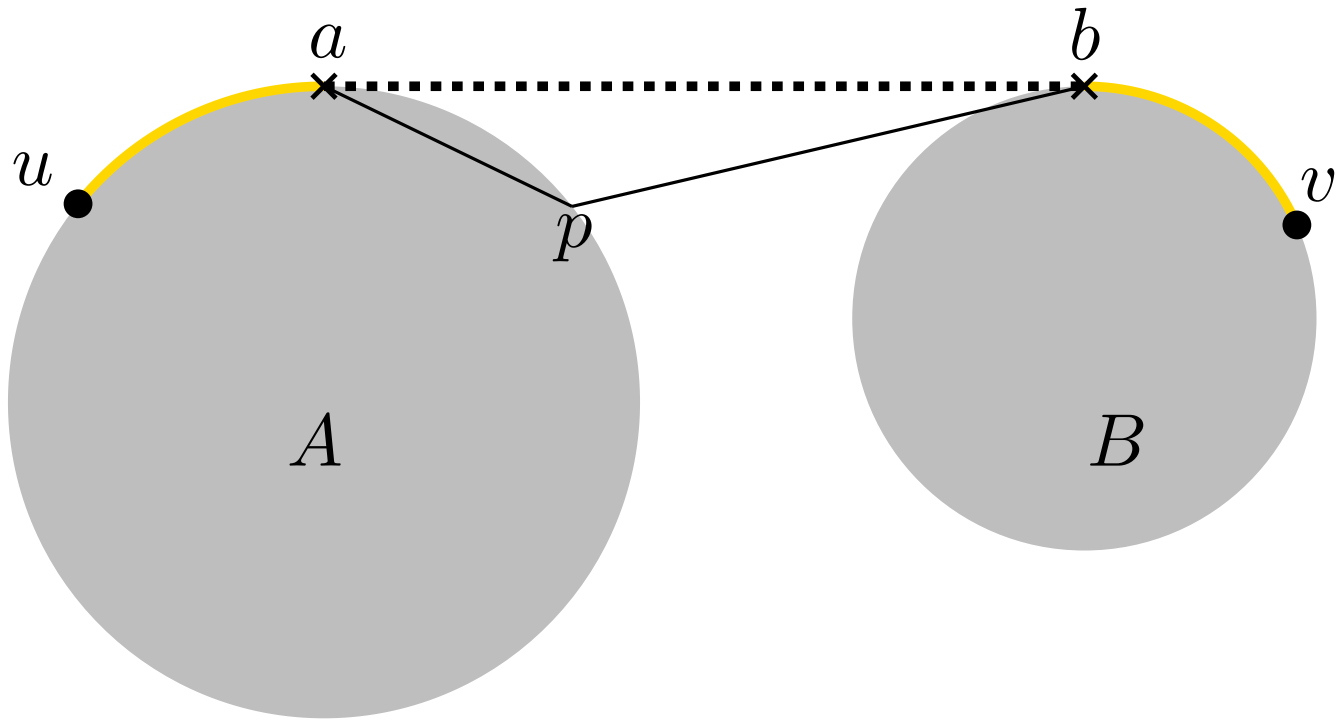
Tangente Kreis – Kreis

Aufgabe 3



Tangente Kreis – Kreis

Aufgabe 3



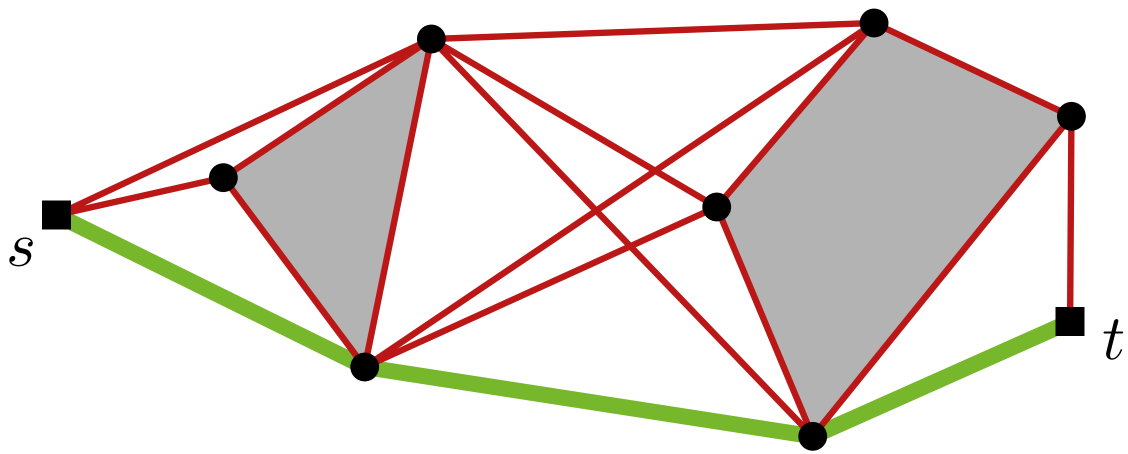
Tangente Kreis – Kreis

Sichtbarkeitsgraph

VL

Gegeben sei eine Menge S disjunkter offener Polygone...

...mit Knotenmenge $V(S)$.



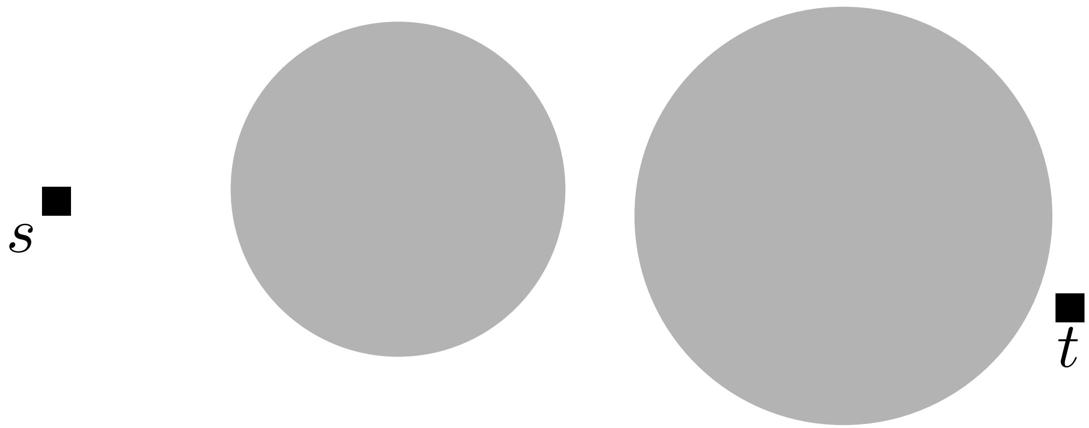
Def.: Dann ist $G_{\text{vis}}(S) = (V(S), E_{\text{vis}}(S))$ der **Sichtbarkeitsgraph** von S mit $E_{\text{vis}}(S) = \{uv \mid u, v \in V(S)$ und u *sieht* $v\}$ und $w(uv) = |uv|$.
Dabei gilt u **sieht** $v :\Leftrightarrow \overline{uv} \cap \bigcup S = \{u, v\}$

Definiere $S^\star = S \cup \{s, t\}$ und $G_{\text{vis}}(S^\star)$ analog.

Lemma 1 $\Rightarrow$ Der kürzeste st-Weg, der die Hindernisse in S vermeidet, entspricht einem kürzesten Weg in $G_{\text{vis}}(S^\star)$.

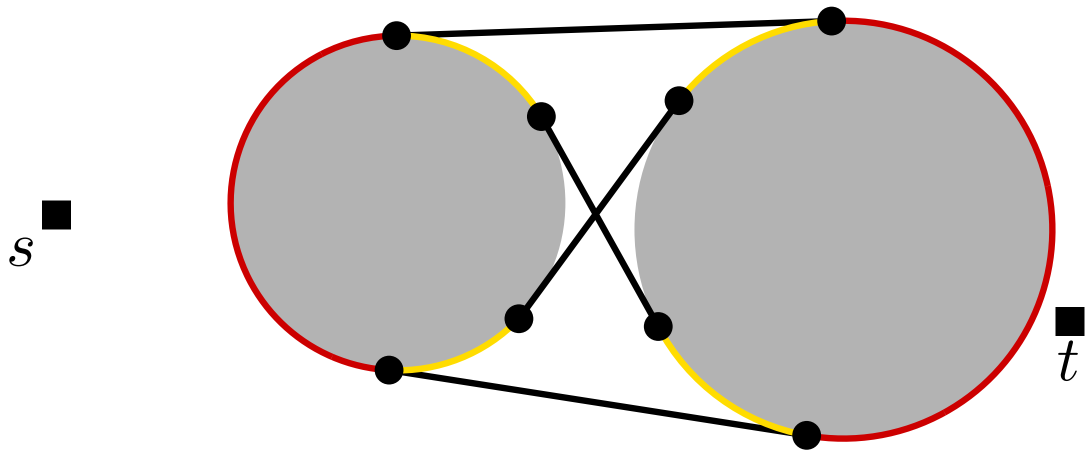
Aufgabe 3 – Sichtbarkeitsgraph

Gegeben sei eine Menge S disjunkter offener Scheiben.



Aufgabe 3 – Sichtbarkeitsgraph

Gegeben sei eine Menge S disjunkter offener Scheiben.



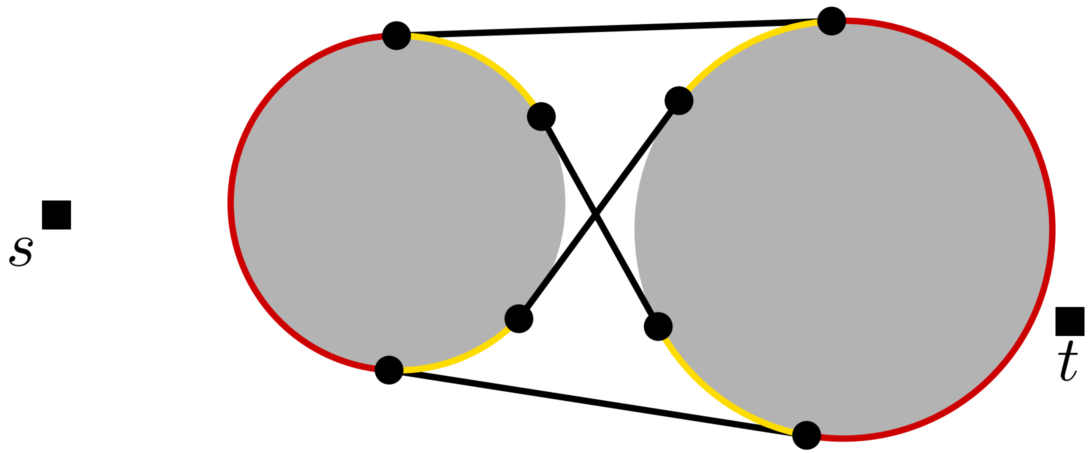
Tangentenknoten: $V(S)$

$E_{\text{vis}}(S) = \{uv \mid u, v \in V(S)$ und u *sieht* $v\}$ und $w(uv) = |uv|$.

$E_{\text{arc}}(S) = \{$Kreisbogen $uv \mid u, v \in$
$V(S)$ und u, v auf gleicher Scheibe 'hintereinander' $\}$
und $w(uv) = |uv| \cdot \alpha$.

Aufgabe 3 – Sichtbarkeitsgraph

Gegeben sei eine Menge S disjunkter offener Scheiben.



Tangentenknoten: $V(S)$

$E_{\text{vis}}(S) = \{uv \mid u, v \in V(S)$ und u *sieht* $v\}$ und $w(uv) = |uv|$.

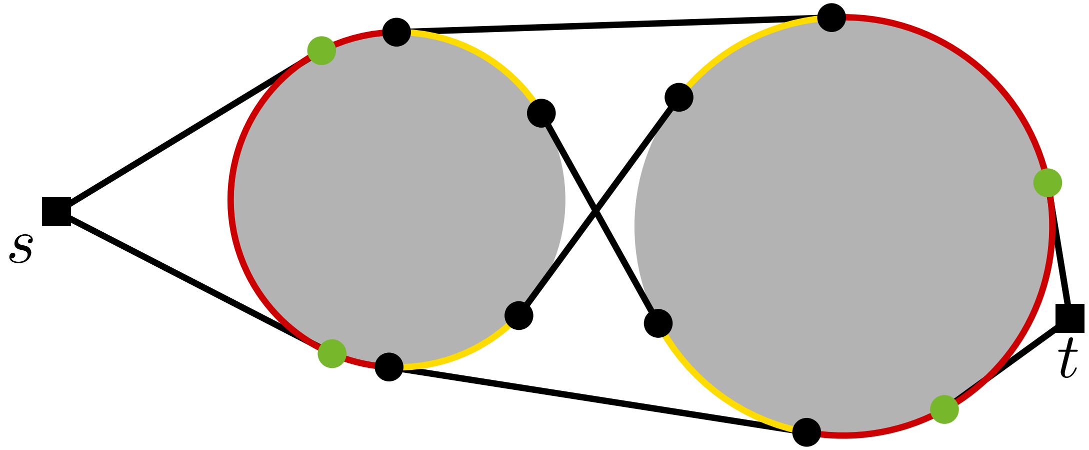
$E_{\text{arc}}(S) = \{$Kreisbogen $uv \mid u, v \in$
$V(S)$ und u, v auf gleicher Scheibe 'hintereinander' $\}$
und $w(uv) = |uv| \cdot \alpha$.

Def.: Dann ist $G_{\text{vis}}(S) = (V(S), E_{\text{vis}}(S) \cup E_{\text{arc}})$ der **Sichtbarkeitsgraph** von S.

Definiere $S^\star = S \cup \{s, t\}$ und $G_{\text{vis}}(S^\star)$ analog.

Aufgabe 3 – Sichtbarkeitsgraph

Gegeben sei eine Menge S disjunkter offener Scheiben.



Tangentenknoten: $V(S)$

$E_{\text{vis}}(S) = \{uv \mid u, v \in V(S)$ und u *sieht* $v\}$ und $w(uv) = |uv|$.

$E_{\text{arc}}(S) = \{$Kreisbogen $uv \mid u, v \in$
$V(S)$ und u, v auf gleicher Scheibe 'hintereinander' $\}$
und $w(uv) = |uv| \cdot \alpha$.

Def.: Dann ist $G_{\text{vis}}(S) = (V(S), E_{\text{vis}}(S) \cup E_{\text{arc}})$ der **Sichtbarkeitsgraph** von S.

Definiere $S^\star = S \cup \{s, t\}$ und $G_{\text{vis}}(S^\star)$ analog.

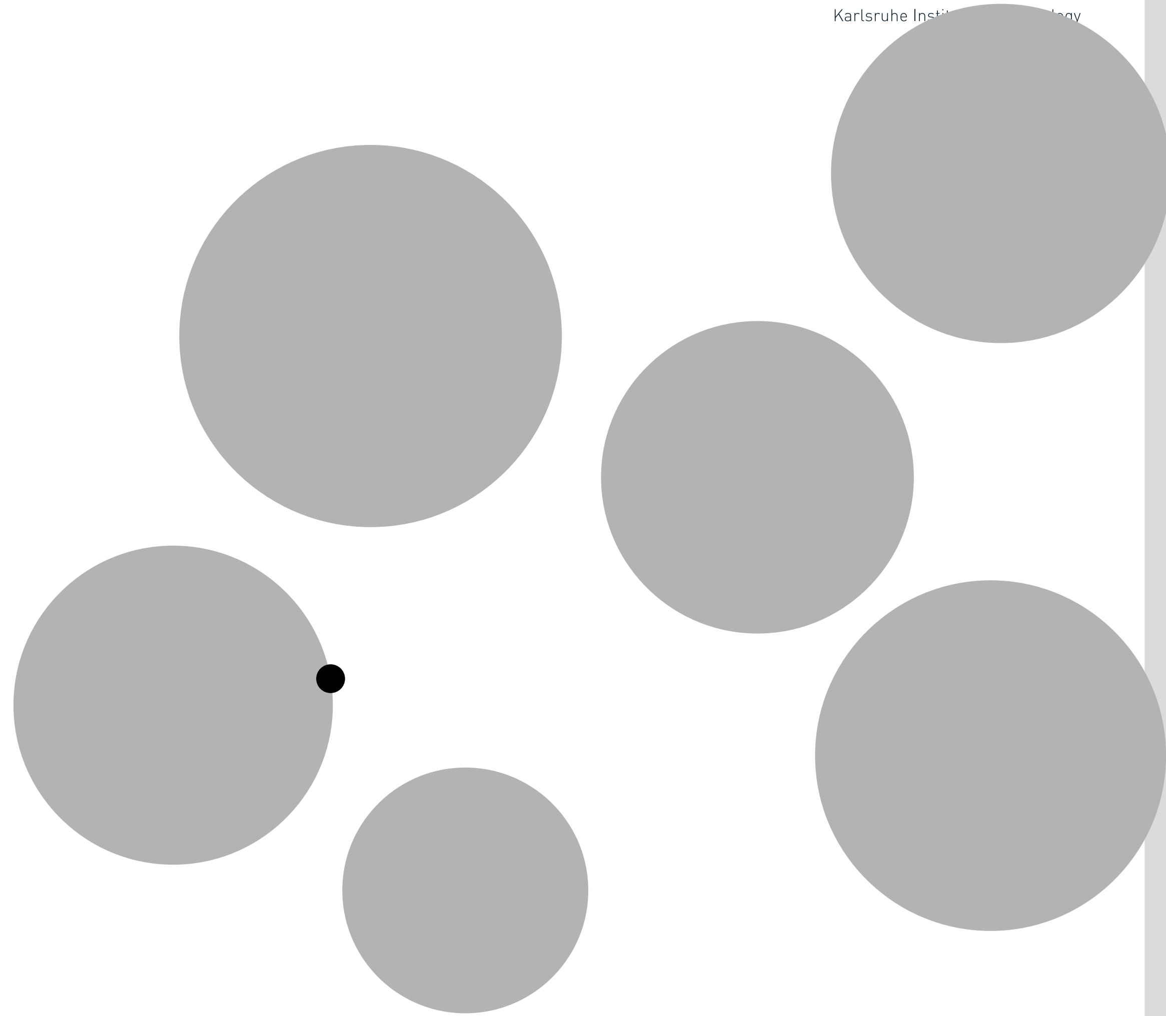
Aufgabe 3 – Sichtbarkeitsgraph berechnen

$\textsc{VisibilityGraph}(S)$

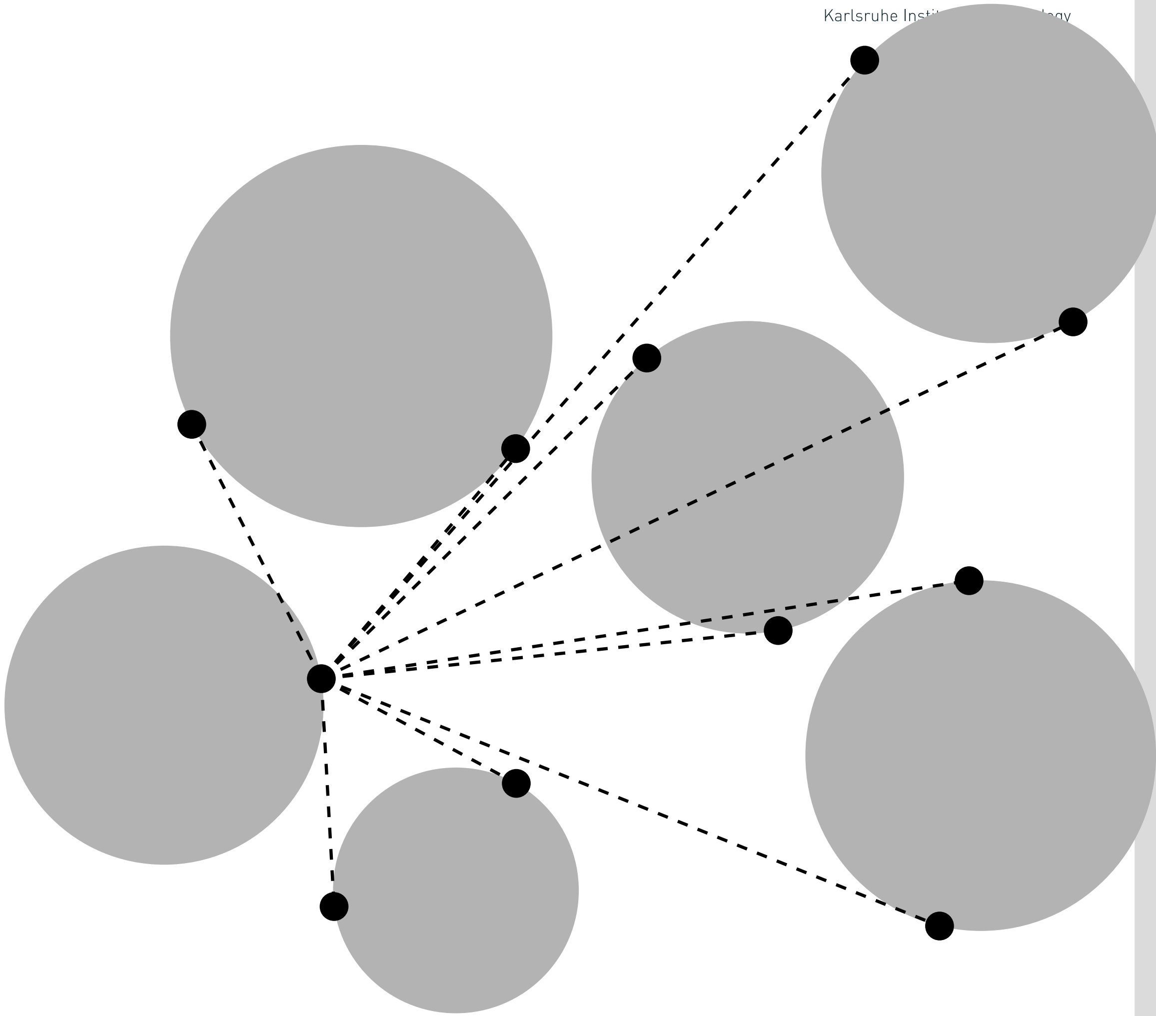
Input: Menge disjunkter Polygone S
Output: Sichtbarkeitsgraph $G_{\mathsf{vis}}(S)$

$E \leftarrow \emptyset$
foreach $v \in V(S)$ **do**
    $W \leftarrow \textsc{VisibleVertices}(v, S)$
    $E \leftarrow E \cup \{vw \mid w \in W\}$
    $E \leftarrow E \cup \{$ arc zum nächsten Nachbarn auf der Scheibe $\}$
return E

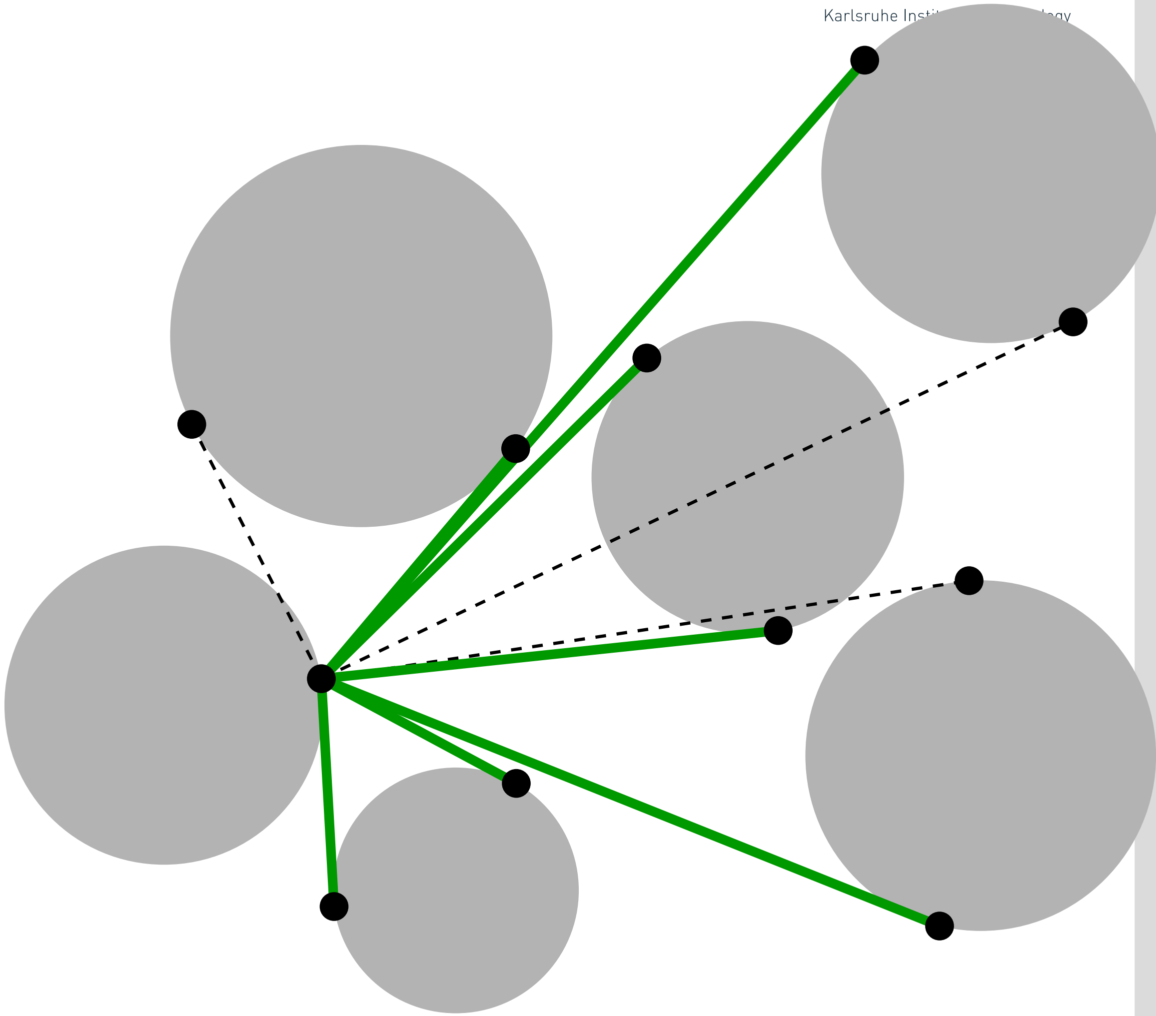
Aufgabe 3 – Sichtbare Knoten berechnen



Aufgabe 3 – Sichtbare Knoten berechnen



Aufgabe 3 – Sichtbare Knoten berechnen



Aufgabe 3 – Sichtbare Knoten berechnen

$\textsc{VisibleVertices}(p, S)$

$r \leftarrow \{p + k\binom{1}{0} \mid k \in \mathbb{R}_0^+\}$

$I \leftarrow \{e \in E(S) \mid e \cap r \neq \emptyset\}$

$\mathcal{T} \leftarrow$ balancedBinaryTree(I)

$w_1, \dots, w_n \leftarrow$ sortiere $V(S)$ im UZS

$W \leftarrow \emptyset$

for $i = 1$ **to** n **do**

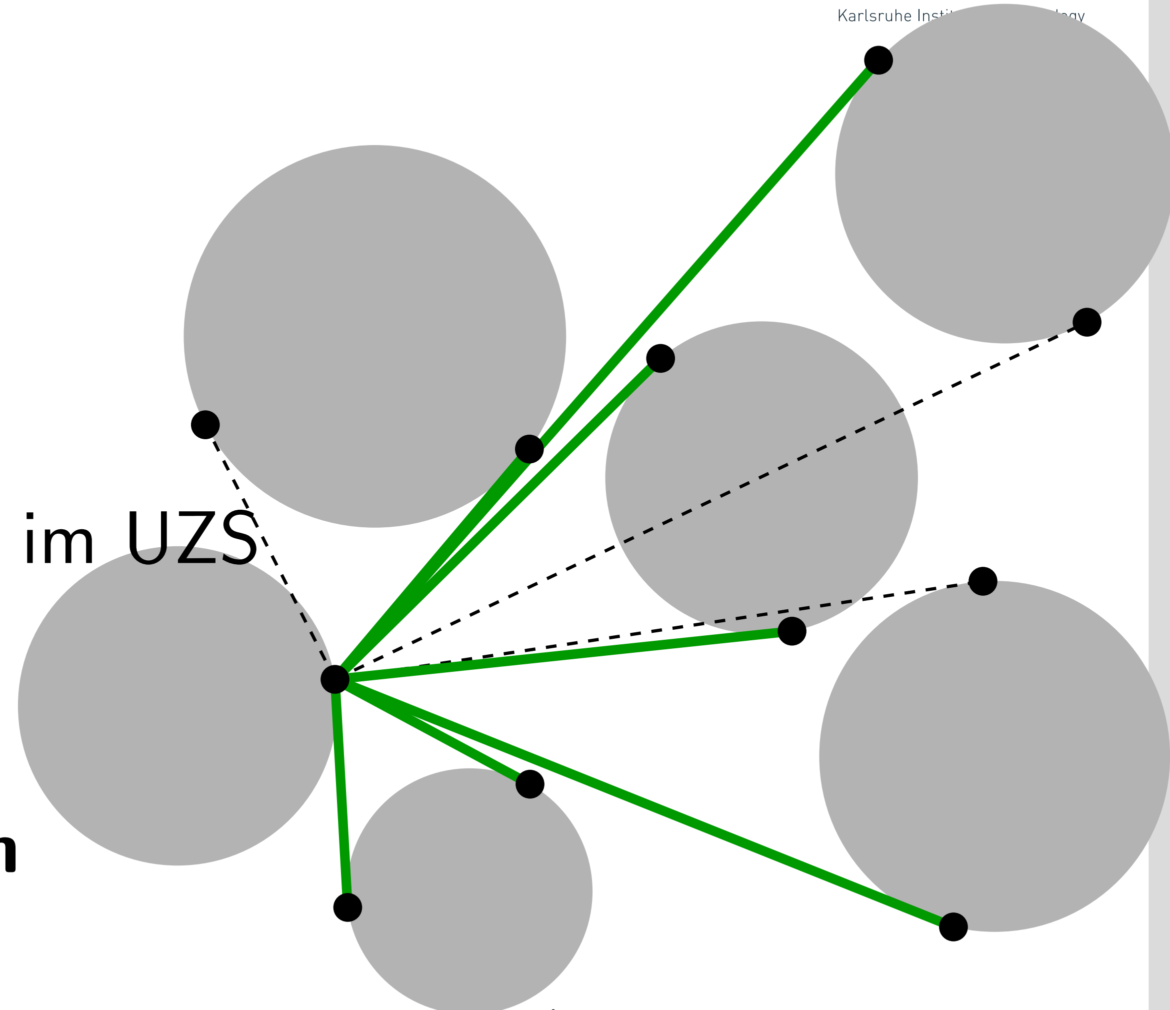
  if $\textsc{Visible}(p, w_i)$ **then**

    $W \leftarrow W \cup \{w_i\}$

  füge in $\mathcal{T}$ zu w_i inzidente Kante aus $\overrightarrow{pw_i}^+$ ein

  lösche aus $\mathcal{T}$ zu w_i inzidente Kanten aus $\overrightarrow{pw_i}^-$

return W



Aufgabe 3 – Shortest Path

$\textsc{ShortestPath}(S, s, t)$ $\boxed{n = |V(S)|, m = |E_{\mathsf{vis}}(S)|}$

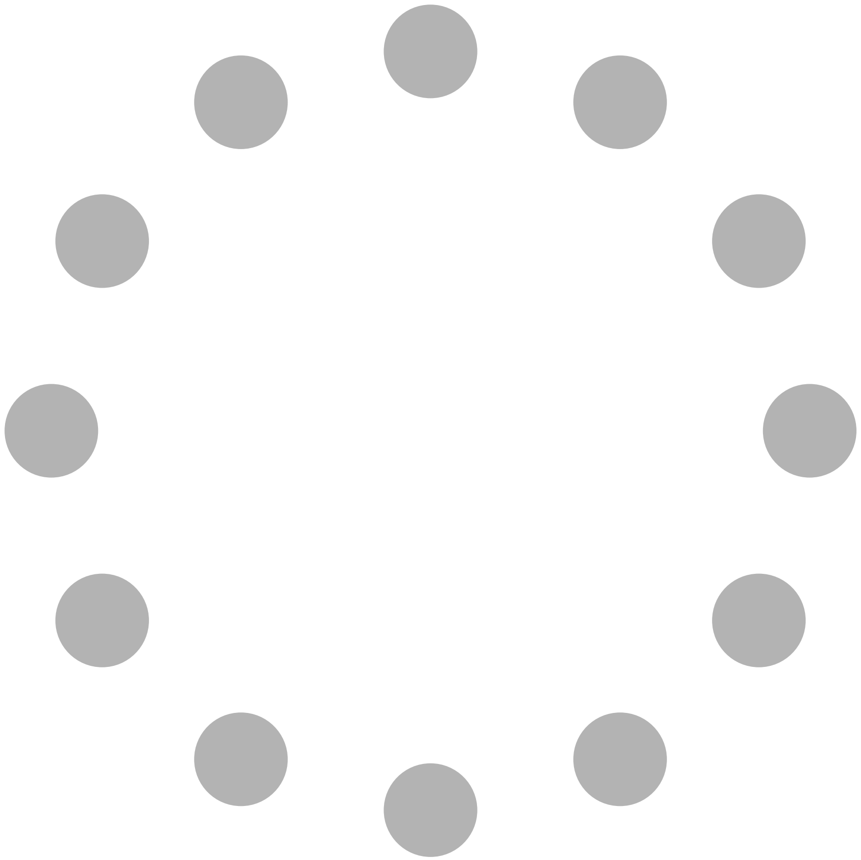
Input: Hindernismenge S, Punkte $s, t \in \mathbb{R}^2 \setminus \bigcup S$
Output: kürzester kollisionsfreier st-Weg in S

1 $G_{\mathsf{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{s, t\})$ $O(n^2 \log n)$
2 **foreach** $uv \in E_{\mathsf{vis}}(S)$ **do** $w(uv) \leftarrow |uv|$ $O(m)$
3 **return** $\textsc{Dijkstra}(G_{\mathsf{vis}}, w, s, t)$ $O(n \log n + m)$

$O(n^2 \log n)$

Aufgabe 3 – Worst Case

n Scheiben



Von jeder Scheibe mindestens eine Tangente an jede andere Scheibe

$\Omega(n)$ Tangentenknoten pro Scheibe $\Rightarrow$ $\Omega(n^2)$ Knoten insgesamt

$\Omega(n)$ Tangentenknoten pro Scheibe $\Rightarrow$ $\Omega(n^2)$ Kanten pro Scheibe $+$
$\Omega(n^2)$ Kanten von Scheiben zu Scheiben $\Rightarrow$ $\Omega(n^2)$ Kanten insgesamt

Aufgabe 3 – Shortest Path

$\textsc{ShortestPath}(S, s, t)$

$\boxed{n^2 = |V(S)|, m = |E|}$

Input: Hindernismenge S, Punkte $s, t \in \mathbb{R}^2 \setminus \bigcup S$

Output: kürzester kollisionsfreier st-Weg in S

$G_{\mathsf{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{s, t\})$ $\quad O(n^4 \log n^2)$

foreach $uv \in E_{\mathsf{vis}}(S)$ **do** $w(uv) \leftarrow |uv|$ $\quad O(m)$

return $\textsc{Dijkstra}(G_{\mathsf{vis}}, w, s, t)$ $\quad O(n^2 \log n^2 + m)$

$$O(n^4 \log n^2)$$

Aufgabe 3 – Shortest Path

$\textsc{ShortestPath}(S, s, t)$ $\boxed{n^2 = |V(S)|, m = |E|}$

Input: Hindernismenge S, Punkte $s, t \in \mathbb{R}^2 \setminus \bigcup S$

Output: kürzester kollisionsfreier st-Weg in S

1 $G_{\mathsf{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{s, t\})$ $O(n^4 \log n^2)$

2 **foreach** $uv \in E_{\mathsf{vis}}(S)$ **do** $w(uv) \leftarrow |uv|$ $O(m)$

3 **return** $\textsc{Dijkstra}(G_{\mathsf{vis}}, w, s, t)$ $O(n^2 \log n^2 + m)$

$O(n^4 \log n^2)$

Danke!

Viel Erfolg bei der Prüfung!

Praktikum Routenplanung

Praktikum zur Routenplanung

- im Wintersemester 2012/2013 (Master)
- Betreuung: Moritz Baum, Julian Dibbelt, Thomas Pajor
- 6 ECTS-Punkte
- Praktische Implementierung einiger Aspekte aus der Vorlesung
  - Beschleunigungstechniken
  - Alternativrouten
  - Public Transport
  - Hub-Labels (in Datenbanken)
  - ...
- Möglichkeit aktuelle Forschung im Bereich Routenplanung praktisch auszuprobieren

Details werden auf unserer Seite bekanntgegeben unter
`http://i11www.iti.kit.edu`

Vorlesung Graphvisualisierung

Algorithmen zur Visualisierung von Graphen

- Dozenten: Tamara Mchedlidze, Martin Nöllenburg, Ignaz Rutter
- 5 ECTS-Punkte
- Termin: Dienstags 9:45 Uhr und mittwochs 14 Uhr
- Themen (Auswahl):
  - Kräftebasierte Layouts
  - Knickminimierung, Winkelauflösung
  - Lagenlayouts, Baumlayouts, Gitterzeichnungen
  - Zeichnen von Metromaps
  - ...

Mehr Informationen werden auf unserer Seite bekanntgegeben unter
`http://i11www.iti.kit.edu`

Seminar Algorithmentechnik

Geometrische Algorithmen für Anwendungen in der Geovisualisierung

- Andreas Gemsa, Martin Nöllenburg, Ignaz Rutter
- Termin: Dienstags 15:45 Uhr
- 4 ECTS-Punkte
- Themen (Auswahl):
  - Schematische Karten
  - Kartogramme
  - Beschriftung von Karten
  - Dynamische Darstellung von Karten
  - Generalisierung
  - ...

Mehr Informationen werden auf unserer Seite bekanntgegeben unter
`http://i11www.iti.kit.edu`

Seminar Graph-Theorie/Data-Mining

Synergien aus Graph-Theorie und Data-Mining zur Analyse von Netzwerkdaten

- Proseminar und Seminar (Algorithmentechnik B)
- Zusammen mit dem IPD
- Methoden zur Analyse komplexer Netwzwerke
  z.B. soziale Netzwerke, Zitationsdatenbanken, Sensornetzwerke
- Schwerpunkte:
  - Maße zur Bewertung von Knoten und Subgraphen
  - Graph Clustering zur Gruppierung von ähnlichen Graph-Strukturen
  - Graph Outlier Mining zur Erkennung von Graph-Anomalien

Erster Termin: 16. Oktober 2012, 14 Uhr (SR 348, Geb. 50.34)

Institut für Theoretische Informatik
Lehrstuhl Algorithmik