

# Vorlesung Algorithmische Geometrie

## Konvexe Hülle im $\mathbb{R}^3$

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

Andreas Gemsa  
26.06.2012



# Prüfung!

# Prüfung!

## Termine:

- 20. Juli
- 27. September

## Termine:

- 20. Juli
- 27. September

## Ansprechpartner:

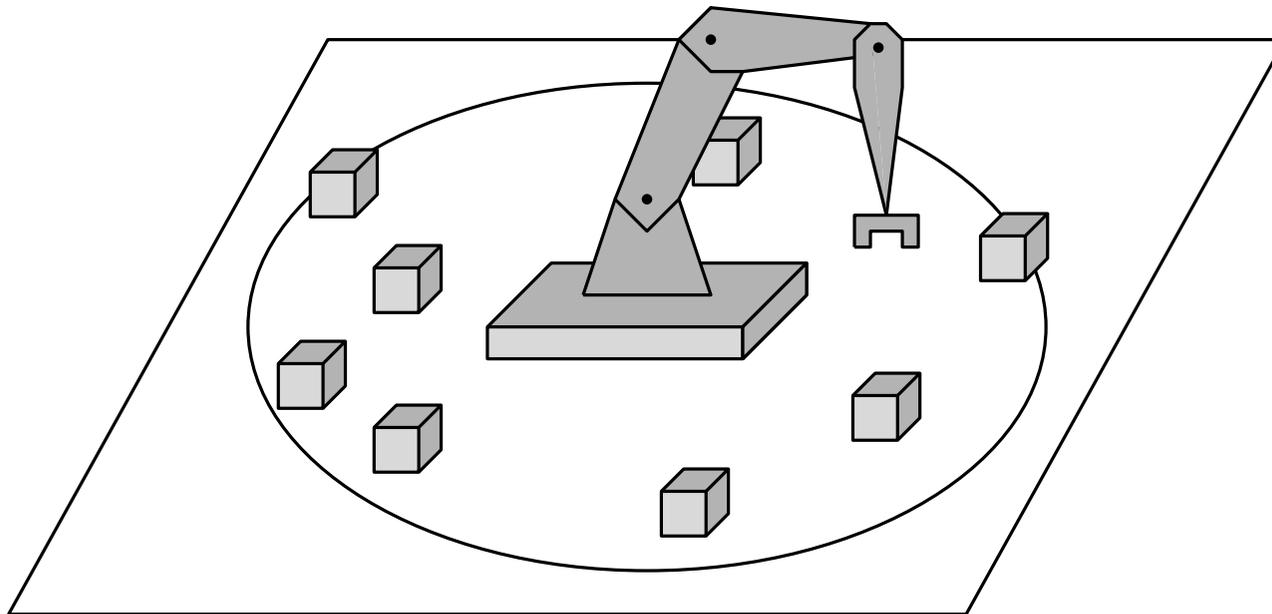
- Lilian Beckert
- [lilian.beckert@kit.edu](mailto:lilian.beckert@kit.edu)
- Raum 321

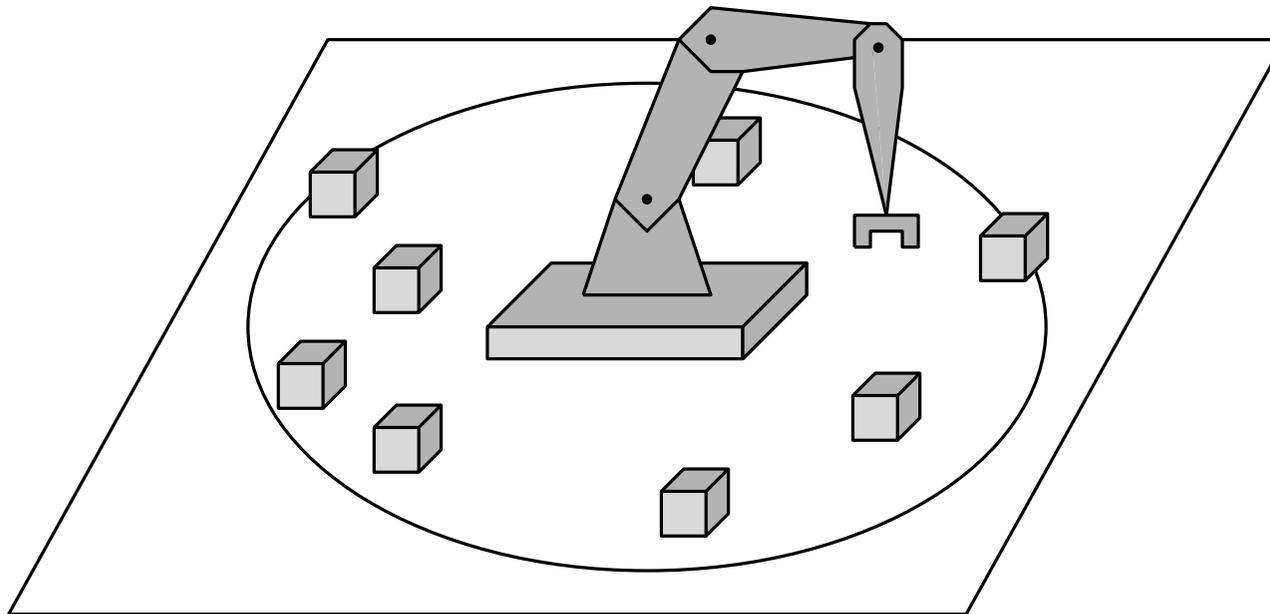
Alle Infos auch auf der Homepage

Smallest Enclosing Disk

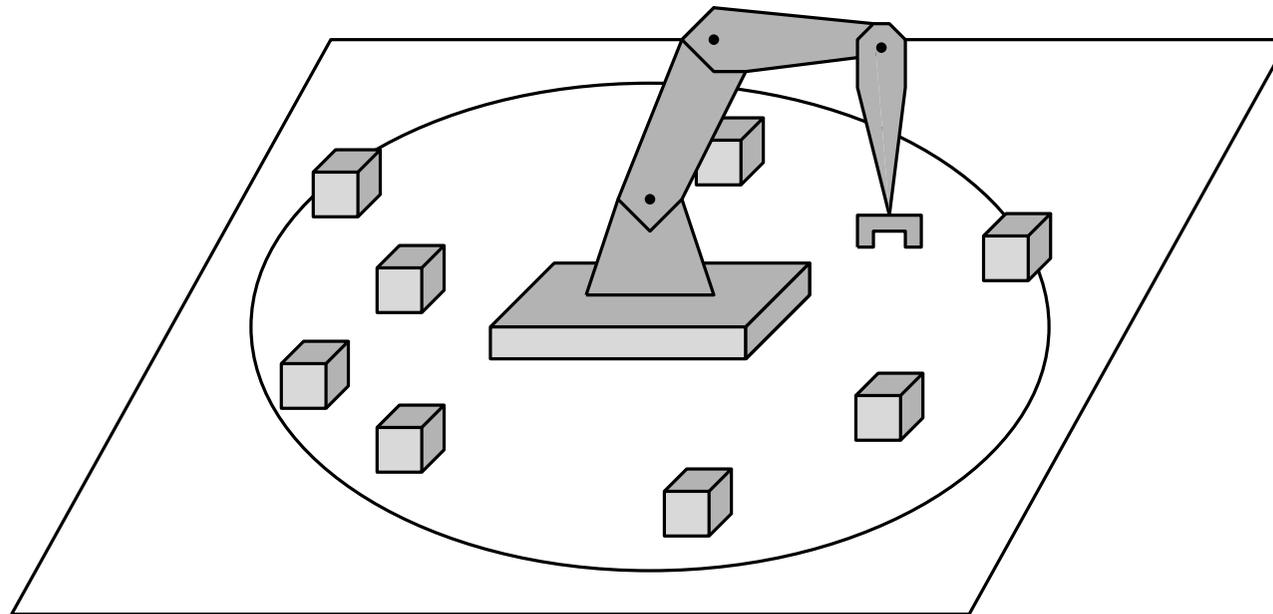
Konvexe Hülle im  $\mathbb{R}^3$

# Problem





**Frage:** Wie findet man eine gute Position für den Roboter?



**Frage:** Wie findet man eine gute Position für den Roboter?

*Smallest Enclosing Disc*

# Smallest Enclosing Disk (SED)

**Def.:** Sei  $P$  eine  $n$ -elementige Menge von Punkten im  $\mathbb{R}^2$ .  
Die *smallest enclosing disc* ist für  $P$  ist der Kreis der alle Punkte aus  $P$  enthält und minimalen Radius hat.

# Smallest Enclosing Disk (SED)

**Def.:** Sei  $P$  eine  $n$ -elementige Menge von Punkten im  $\mathbb{R}^2$ . Die *smallest enclosing disc* ist für  $P$  ist der Kreis der alle Punkte aus  $P$  enthält und minimalen Radius hat.

**Lösung:** Ähnlicher Ansatz wie bei linearer Programmierung.

# Smallest Enclosing Disk (SED)

**Def.:** Sei  $P$  eine  $n$ -elementige Menge von Punkten im  $\mathbb{R}^2$ . Die *smallest enclosing disc* ist für  $P$  ist der Kreis der alle Punkte aus  $P$  enthält und minimalen Radius hat.

**Lösung:** Ähnlicher Ansatz wie bei linearer Programmierung.

- randomisiert
- inkrementell

**Def.:** Sei  $P$  eine  $n$ -elementige Menge von Punkten im  $\mathbb{R}^2$ . Die *smallest enclosing disc* ist für  $P$  ist der Kreis der alle Punkte aus  $P$  enthält und minimalen Radius hat.

**Lösung:** Ähnlicher Ansatz wie bei linearer Programmierung.

- randomisiert
  - inkrementell
- 

**Idee:**

- Erzeuge zufällige Permutation von  $P \rightarrow \{p_1, \dots, p_n\}$ .
- Im  $i$ -ten Schritt:
  - Füge zur Menge  $P_i := \{p_1, \dots, p_i\}$  den Punkt  $p_{i+1}$  hinzu.
  - Finde neue smallest enclosing disk  $D_i$ .

**Def.:** Sei  $P$  eine  $n$ -elementige Menge von Punkten im  $\mathbb{R}^2$ . Die *smallest enclosing disc* ist für  $P$  ist der Kreis der alle Punkte aus  $P$  enthält und minimalen Radius hat.

**Lösung:** Ähnlicher Ansatz wie bei linearer Programmierung.

- randomisiert
  - inkrementell
- 

**Idee:**

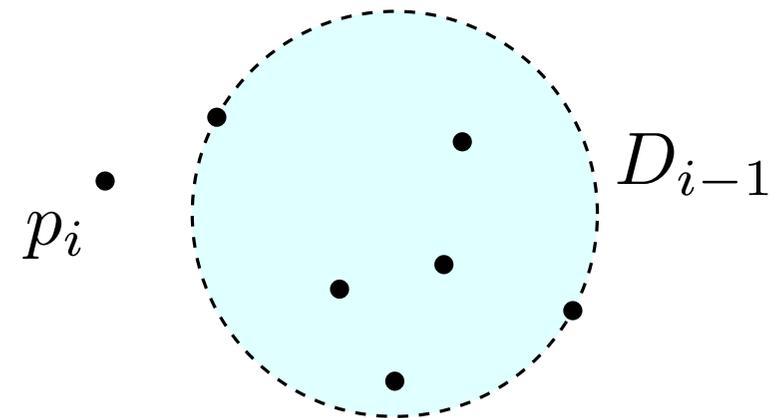
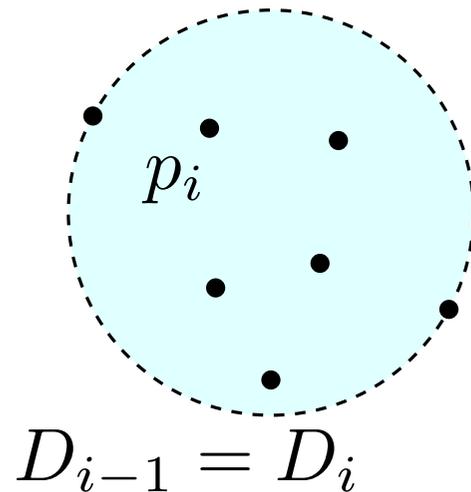
- Erzeuge zufällige Permutation von  $P \rightarrow \{p_1, \dots, p_n\}$ .
- Im  $i$ -ten Schritt:
  - Füge zur Menge  $P_i := \{p_1, \dots, p_i\}$  den Punkt  $p_{i+1}$  hinzu.
  - Finde neue smallest enclosing disk  $D_i$ .

# Smallest Enclosing Disk

**Lm:** Sei  $2 < i < n$ , und sei  $P_i$  und  $D_i$  wie gerade dann gilt:

(i) Wenn  $p_i \in D_{i-1}$ , dann gilt  $D_i = D_{i-1}$

(ii) Wenn  $p_i \notin D_{i-1}$ , dann liegt  $p_i$  auf dem Rand von  $D_i$

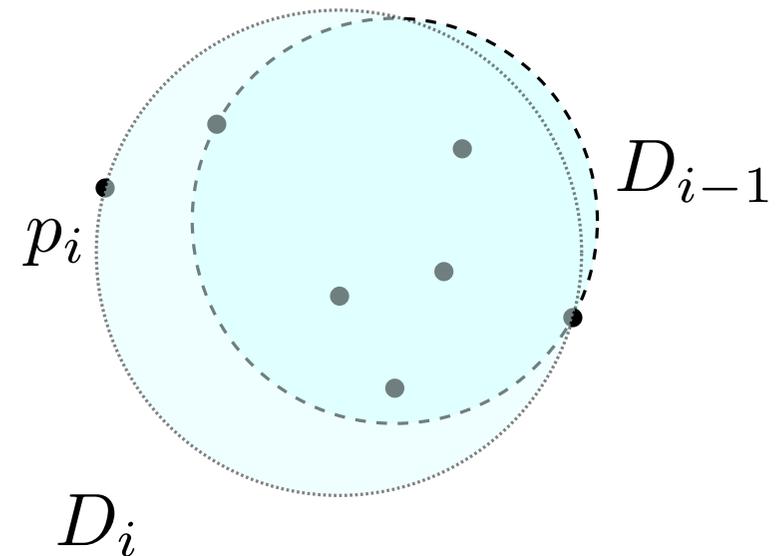
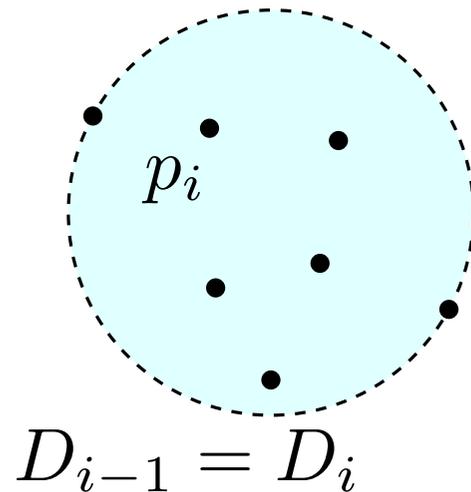


# Smallest Enclosing Disk

**Lm:** Sei  $2 < i < n$ , und sei  $P_i$  und  $D_i$  wie gerade dann gilt:

(i) Wenn  $p_i \in D_{i-1}$ , dann gilt  $D_i = D_{i-1}$

(ii) Wenn  $p_i \notin D_{i-1}$ , dann liegt  $p_i$  auf dem Rand von  $D_i$



# Smallest Enclosing Disk SED

MINIDISC( $P \subset \mathbb{R}^2$ )

$P = \text{RANDOMPERMUTATION}(P)$

Sei  $D_2$  die smallest enclosing disc von  $\{p_1, p_2\}$

**for**  $i \leftarrow 3$  bis  $n$  **do**

**if**  $p_i \in D_{i-1}$  **then**  $D_i = D_{i-1}$   
    **else**  $D_i = \text{MINIDISCWITHPOINT}(\{p_1, \dots, p_{i-1}\}, p_i)$

**return**  $D_i$

# Smallest Enclosing Disk SED

MINIDISC( $P \subset \mathbb{R}^2$ )

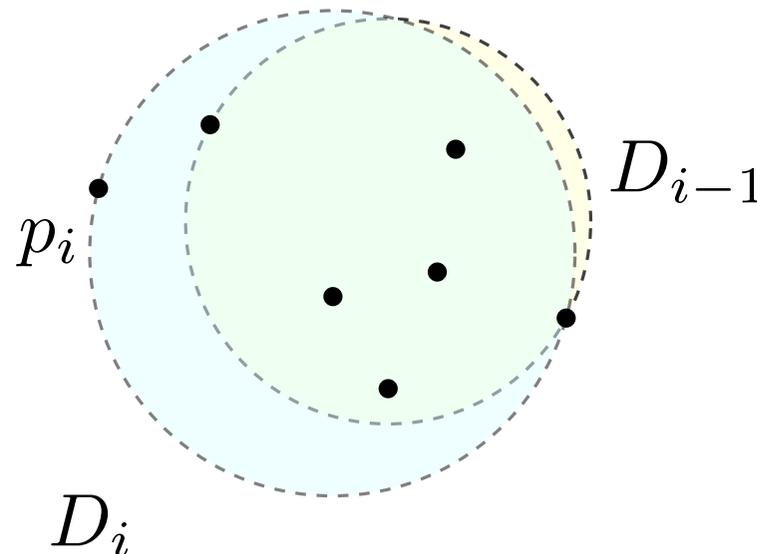
$P = \text{RANDOMPERMUTATION}(P)$

Sei  $D_2$  die smallest enclosing disc von  $\{p_1, p_2\}$

**for**  $i \leftarrow 3$  bis  $n$  **do**

**if**  $p_i \in D_{i-1}$  **then**  $D_i = D_{i-1}$   
    **else**  $D_i = \text{MINIDISCWITHPOINT}(\{p_1, \dots, p_{i-1}\}, p_i)$

**return**  $D_i$



# Smallest Enclosing Disk SED

MINIDISC( $P \subset \mathbb{R}^2$ )

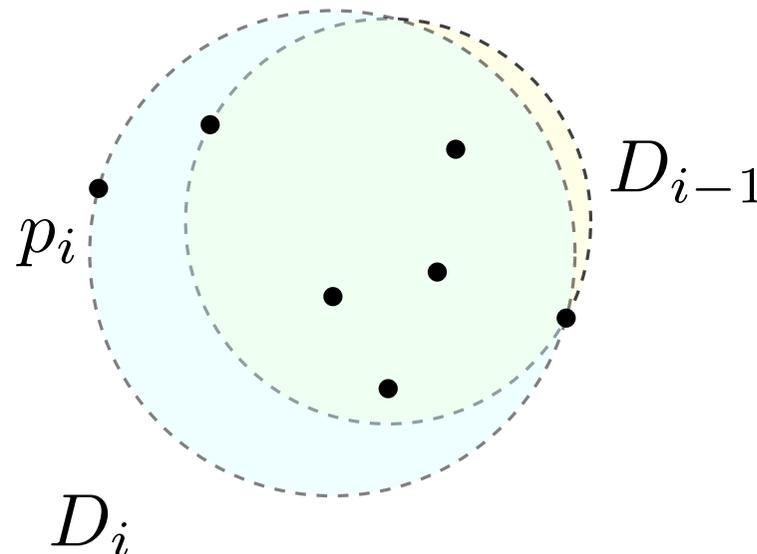
$P = \text{RANDOMPERMUTATION}(P)$

Sei  $D_2$  die smallest enclosing disc von  $\{p_1, p_2\}$

**for**  $i \leftarrow 3$  bis  $n$  **do**

**if**  $p_i \in D_{i-1}$  **then**  $D_i = D_{i-1}$   
    **else**  $D_i = \text{MINIDISCWITHPOINT}(\{p_1, \dots, p_{i-1}\}, p_i)$

**return**  $D_i$



# Smallest Enclosing Disk SED

MINIDISCWITHPOINT( $\{p_1, \dots, p_{i-1}\}, q$ )

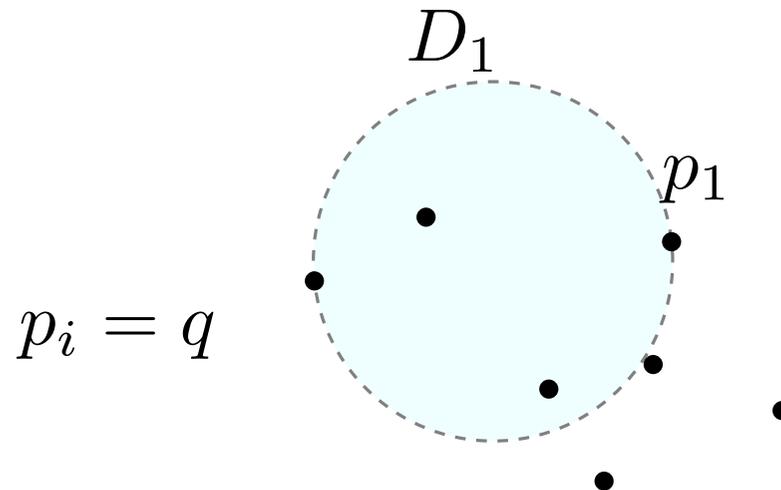
$P = \text{RANDOMPERMUTATION}(\{p_1, \dots, p_{i-1}\})$

Sei  $D_1$  die smallest disc die  $p_1$  und  $q$  auf dem Rand hat

**for**  $j \leftarrow 2$  bis  $i - 1$  **do**

**if**  $p_j \in D_{j-1}$  **then**  $D_j = D_{j-1}$   
    **else**  $D_j = \text{MINIDISCWITH2POINTS}(\{p_1, \dots, p_{j-1}\}, p_j, q)$

**return**  $D_j$



# Smallest Enclosing Disk SED

MINIDISCWITHPOINT( $\{p_1, \dots, p_{i-1}\}, q$ )

$P = \text{RANDOMPERMUTATION}(\{p_1, \dots, p_{i-1}\})$

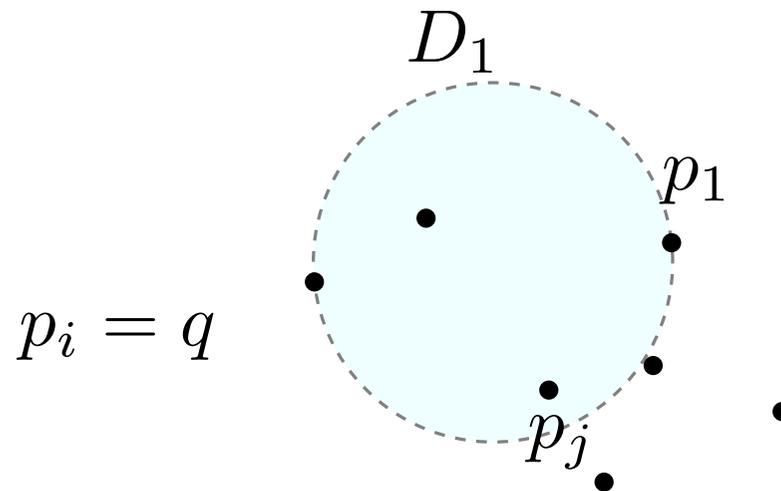
Sei  $D_1$  die smallest disc die  $p_1$  und  $q$  auf dem Rand hat

**for**  $j \leftarrow 2$  bis  $i - 1$  **do**

**if**  $p_j \in D_{j-1}$  **then**  $D_j = D_{j-1}$

**else**  $D_j = \text{MINIDISCWITH2POINTS}(\{p_1, \dots, p_{j-1}\}, p_j, q)$

**return**  $D_j$



# Smallest Enclosing Disk SED

MINIDISCWITH2POINTS( $\{p_1, \dots, p_{j-1}\}, q_1, q_2$ )

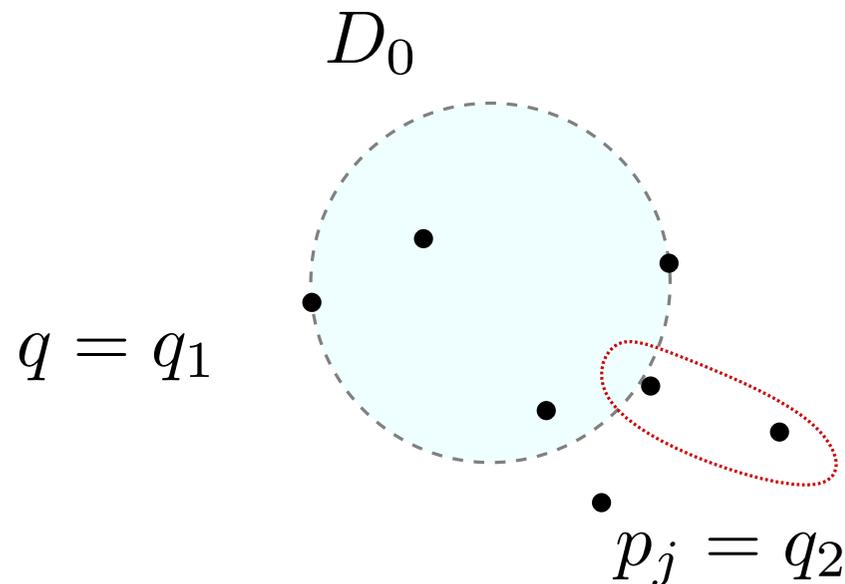
Sei  $D_0$  die smallest disc bei der  $q_1$  und  $q_2$  auf dem Rand liegen

**for**  $k \leftarrow 1$  to  $j - 1$  **do**

**if**  $p_k \in D_{k-1}$  **then**  $D_k = D_{k-1}$

**else**  $D_k =$  disc bei der  $q_1, q_2$  und  $p_k$  auf dem Rand liegen

**return**  $D_j$



# Smallest Enclosing Disk SED

MINIDISCWITH2POINTS( $\{p_1, \dots, p_{j-1}\}, q_1, q_2$ )

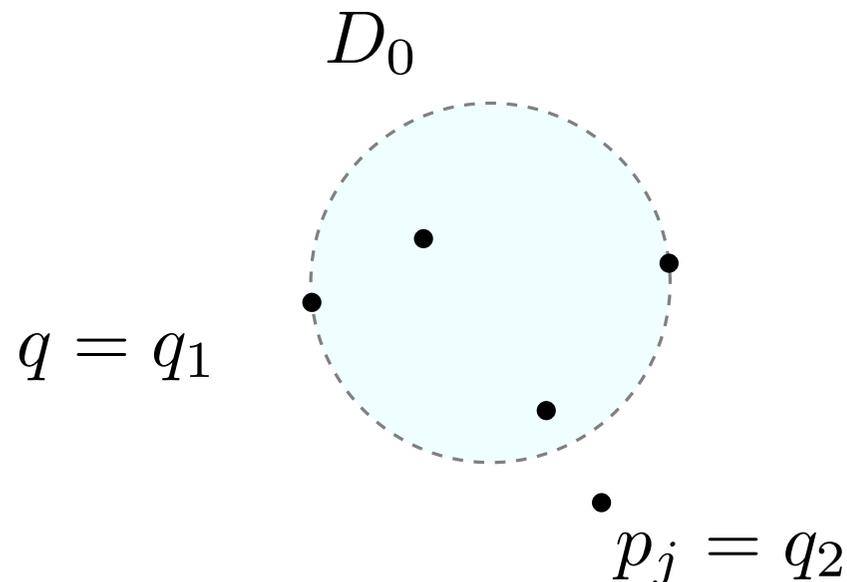
Sei  $D_0$  die smallest disc bei der  $q_1$  und  $q_2$  auf dem Rand liegen

**for**  $k \leftarrow 1$  to  $j - 1$  **do**

**if**  $p_k \in D_{k-1}$  **then**  $D_k = D_{k-1}$

**else**  $D_k =$  disc bei der  $q_1, q_2$  und  $p_k$  auf dem Rand liegen

**return**  $D_j$



# Smallest Enclosing Disk SED

MINIDISCWITH2POINTS( $\{p_1, \dots, p_{j-1}\}, q_1, q_2$ )

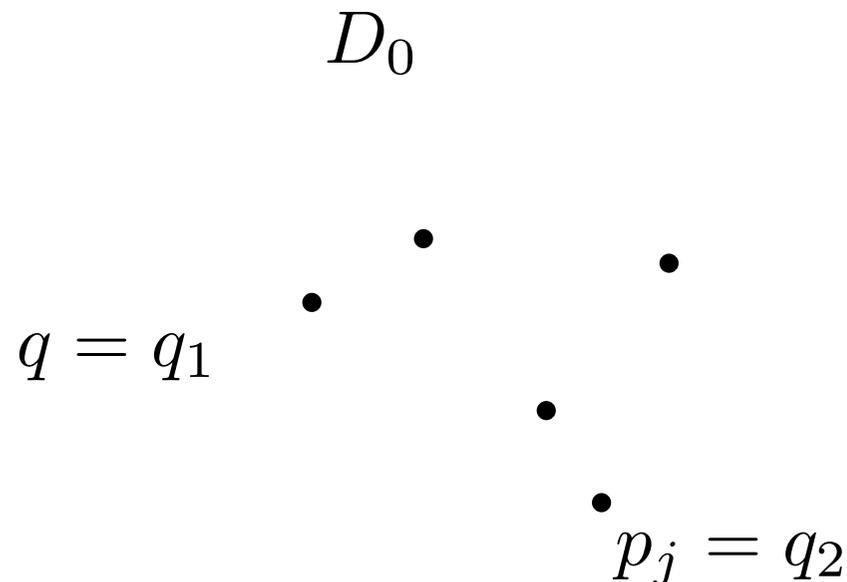
Sei  $D_0$  die smallest disc bei der  $q_1$  und  $q_2$  auf dem Rand liegen

**for**  $k \leftarrow 1$  to  $j - 1$  **do**

**if**  $p_k \in D_{k-1}$  **then**  $D_k = D_{k-1}$

**else**  $D_k =$  disc bei der  $q_1, q_2$  und  $p_k$  auf dem Rand liegen

**return**  $D_j$



# Smallest Enclosing Disk SED

MINIDISCWITH2POINTS( $\{p_1, \dots, p_{j-1}\}, q_1, q_2$ )

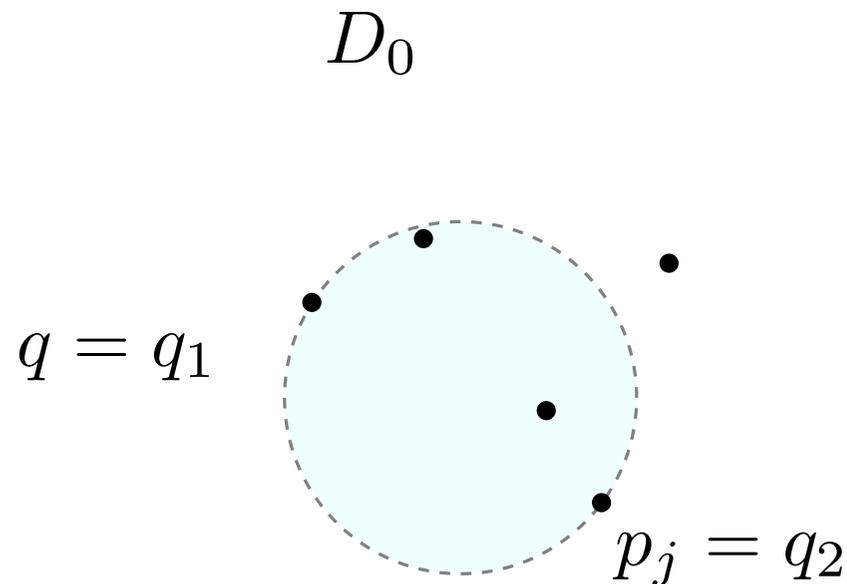
Sei  $D_0$  die smallest disc bei der  $q_1$  und  $q_2$  auf dem Rand liegen

**for**  $k \leftarrow 1$  to  $j - 1$  **do**

**if**  $p_k \in D_{k-1}$  **then**  $D_k = D_{k-1}$

**else**  $D_k =$  disc bei der  $q_1, q_2$  und  $p_k$  auf dem Rand liegen

**return**  $D_j$



# Smallest Enclosing Disk SED

MINIDISCWITH2POINTS( $\{p_1, \dots, p_{j-1}\}, q_1, q_2$ )

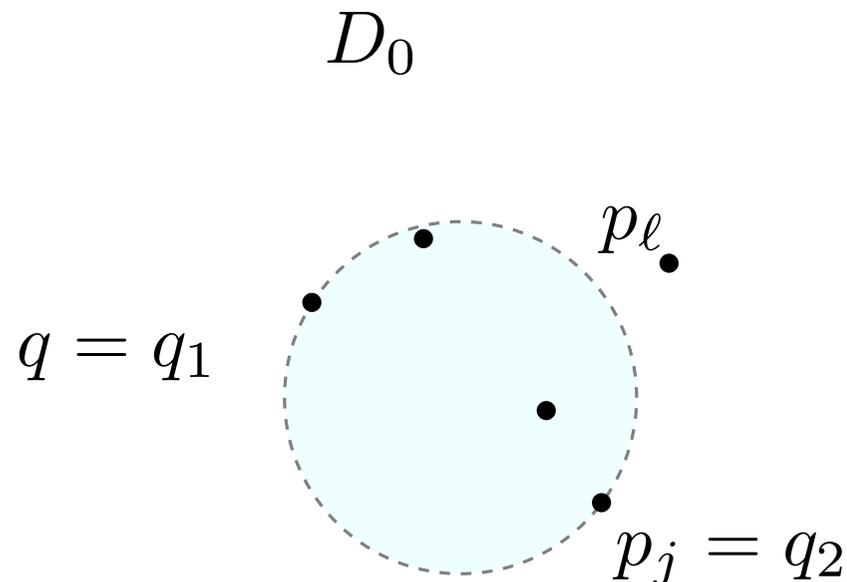
Sei  $D_0$  die smallest disc bei der  $q_1$  und  $q_2$  auf dem Rand liegen

**for**  $k \leftarrow 1$  to  $j - 1$  **do**

**if**  $p_k \in D_{k-1}$  **then**  $D_k = D_{k-1}$

**else**  $D_k =$  disc bei der  $q_1, q_2$  und  $p_k$  auf dem Rand liegen

**return**  $D_j$



# Smallest Enclosing Disk SED

MINIDISCWITH2POINTS( $\{p_1, \dots, p_{j-1}\}, q_1, q_2$ )

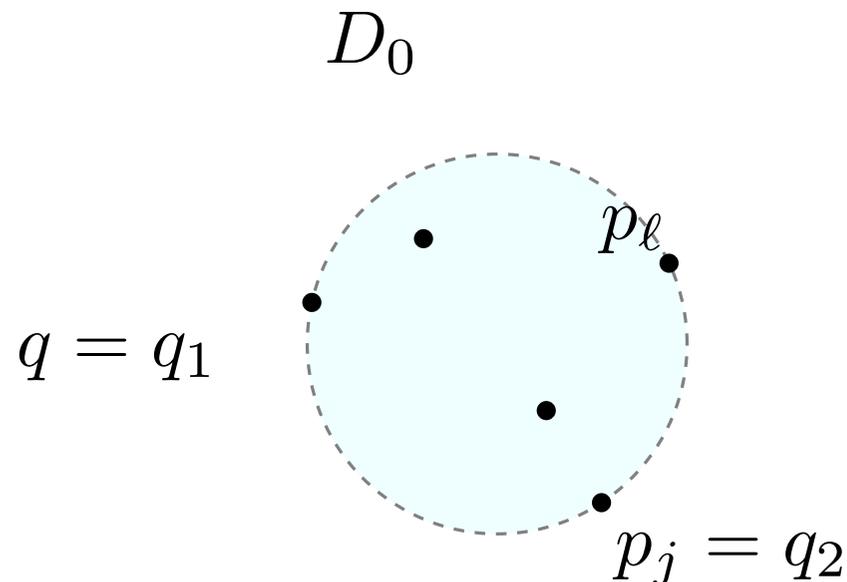
Sei  $D_0$  die smallest disc bei der  $q_1$  und  $q_2$  auf dem Rand liegen

**for**  $k \leftarrow 1$  to  $j - 1$  **do**

**if**  $p_k \in D_{k-1}$  **then**  $D_k = D_{k-1}$

**else**  $D_k =$  disc bei der  $q_1, q_2$  und  $p_k$  auf dem Rand liegen

**return**  $D_j$



**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

MINIDISC

MINIDISCWITHPOINT

MINIDISCWITH2POINTS

**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

MINIDISC

MINIDISCWITHPOINT

MINIDISCWITH2POINTS

---

MINIDISC( $P \subset \mathbb{R}^2$ )

$P = \text{RANDOMPERMUTATION}(P)$

Sei  $D_2$  die smallest enclosing disc von  $\{p_1, p_2\}$

**for**  $i \leftarrow 3$  bis  $n$  **do**

**if**  $p_i \in D_{i-1}$  **then**  $D_i = D_{i-1}$   
    **else**  $D_i = \text{MINIDISCWITHPOINT}(\{p_1, \dots, p_{i-1}\}, p_i)$

**return**  $D_i$

**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

MINIDISC

MINIDISCWITHPOINT

MINIDISCWITH2POINTS

---

MINIDISC( $P \subset \mathbb{R}^2$ )

$P = \text{RANDOMPERMUTATION}(P)$

Sei  $D_2$  die smallest enclosing disc von  $\{p_1, p_2\}$

**for**  $i \leftarrow 3$  bis  $n$  **do**

**if**  $p_i \in D_{i-1}$  **then**  $D_i = D_{i-1}$   
    **else**  $D_i = \text{MINIDISCWITHPOINT}(\{p_1, \dots, p_{i-1}\}, p_i)$

**return**  $D_i$

$\mathcal{O}(n)$

**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

MINIDISC

MINIDISCWITHPOINT

MINIDISCWITH2POINTS

---

MINIDISCWITHPOINT( $\{p_1, \dots, p_{i-1}\}, q$ )

$P = \text{RANDOMPERMUTATION}(\{p_1, \dots, p_{i-1}\})$

Sei  $D_1$  die smallest disc die  $p_1$  und  $q$  auf dem Rand hat

**for**  $j \leftarrow 2$  bis  $i - 1$  **do**

**if**  $p_j \in D_{j-1}$  **then**  $D_j = D_{j-1}$

**else**  $D_j = \text{MINIDISCWITH2POINTS}(\{p_1, \dots, p_{j-1}\}, p_j, q)$

**return**  $D_j$

**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

MINIDISC

MINIDISCWITHPOINT

MINIDISCWITH2POINTS

---

```
MINIDISCWITHPOINT( $\{p_1, \dots, p_{i-1}\}, q$ )  $\mathcal{O}(i)$   
  P = RANDOMPERMUTATION( $\{p_1, \dots, p_{i-1}\}$ )  
  Sei  $D_1$  die smallest disc die  $p_1$  und  $q$  auf dem Rand hat  
  for  $j \leftarrow 2$  bis  $i - 1$  do  
    if  $p_j \in D_{j-1}$  then  $D_j = D_{j-1}$   
    else  $D_j = \text{MINIDISCWITH2POINTS}(\{p_1, \dots, p_{j-1}\}, p_j, q)$   
  return  $D_j$ 
```

**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

MINIDISC

MINIDISCWITHPOINT

MINIDISCWITH2POINTS

---

MINIDISCWITH2POINTS( $\{p_1, \dots, p_{j-1}\}, q_1, q_2$ )

Sei  $D_0$  die smallest disc bei der  $q_1$  und  $q_2$  auf dem Rand liegen

**for**  $k \leftarrow 1$  to  $j - 1$  **do**

**if**  $p_k \in D_{k-1}$  **then**  $D_k = D_{k-1}$

**else**  $D_k =$  disc bei der  $q_1, q_2$  und  $p_k$  auf dem Rand liegen

**return**  $D_j$

**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

MINIDISC

MINIDISCWITHPOINT

MINIDISCWITH2POINTS

---

MINIDISCWITH2POINTS( $\{p_1, \dots, p_{j-1}\}, q_1, q_2$ )  $\mathcal{O}(j)$

Sei  $D_0$  die smallest disc bei der  $q_1$  und  $q_2$  auf dem Rand liegen

**for**  $k \leftarrow 1$  to  $j - 1$  **do**

**if**  $p_k \in D_{k-1}$  **then**  $D_k = D_{k-1}$

**else**  $D_k =$  disc bei der  $q_1, q_2$  und  $p_k$  auf dem Rand liegen

**return**  $D_j$

**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

MINIDISC

MINIDISCWITHPOINT

MINIDISCWITH2POINTS

---

**Erkenntnis:** Linearer Aufwand für jede Funktion.

**Theorem:** Die smallest enclosing disc für  $n$  Punkte in der Ebene kann in erwarteter  $\mathcal{O}(n)$  berechnet werden.

MINIDISC

MINIDISCWITHPOINT

MINIDISCWITH2POINTS

---

**Erkenntnis:** Linearer Aufwand für jede Funktion.

**Frage:** Wie wahrscheinlich ist es, dass MINIDISCWITHPOINT  
MINIDISCWITH2POINTS aufruft?

# Analyse (i)

**Frage:** Wie wahrscheinlich ist es, dass `MINIDISCWITHPOINT`  
`MINIDISCWITH2POINTS` aufruft?

```
MINIDISCWITHPOINT( $\{p_1, \dots, p_{i-1}\}, q$ )
```

```
...
```

```
for  $j \leftarrow 2$  bis  $i - 1$  do
```

```
  if  $p_j \in D_{j-1}$  then  $D_j = D_{j-1}$ 
```

```
  else  $D_j = \text{MINIDISCWITH2POINTS}(\{p_1, \dots, p_{j-1}\}, p_j, q)$ 
```

```
return  $D_j$ 
```

**Frage:** Wie wahrscheinlich ist es, dass `MINIDISCWITHPOINT`  
`MINIDISCWITH2POINTS` aufruft?

```
MINIDISCWITHPOINT( $\{p_1, \dots, p_{i-1}\}, q$ )
```

```
...
```

```
for  $j \leftarrow 2$  bis  $i - 1$  do
```

```
  if  $p_j \in D_{j-1}$  then  $D_j = D_{j-1}$ 
```

```
  else  $D_j = \text{MINIDISCWITH2POINTS}(\{p_1, \dots, p_{j-1}\}, p_j, q)$ 
```

```
return  $D_j$ 
```

## Rückwärtsanalyse

**Frage:** Wie wahrscheinlich ist es, dass `MINIDISCWITHPOINT` `MINIDISCWITH2POINTS` aufruft?

```
MINIDISCWITHPOINT( $\{p_1, \dots, p_{i-1}\}, q$ )
```

```
...
```

```
for  $j \leftarrow 2$  bis  $i - 1$  do
```

```
  if  $p_j \in D_{j-1}$  then  $D_j = D_{j-1}$ 
```

```
  else  $D_j = \text{MINIDISCWITH2POINTS}(\{p_1, \dots, p_{j-1}\}, p_j, q)$ 
```

```
return  $D_j$ 
```

## Rückwärtsanalyse

**Idee:** Fixiere  $P_i := \{p_1, \dots, p_i\}$ . Sei  $D_i$  die smallest enclosing disc für  $P_i$ . Außerdem liege  $q$  auf dem Rand von  $D_i$ .

**Frage:** Wie wahrscheinlich ist es, dass `MINIDISCWITHPOINT` `MINIDISCWITH2POINTS` aufruft?

```
MINIDISCWITHPOINT( $\{p_1, \dots, p_{i-1}\}, q$ )
```

```
...
```

```
for  $j \leftarrow 2$  bis  $i - 1$  do
```

```
  if  $p_j \in D_{j-1}$  then  $D_j = D_{j-1}$ 
```

```
  else  $D_j = \text{MINIDISCWITH2POINTS}(\{p_1, \dots, p_{j-1}\}, p_j, q)$ 
```

```
return  $D_j$ 
```

## Rückwärtsanalyse

**Idee:** Fixiere  $P_i := \{p_1, \dots, p_i\}$ . Sei  $D_i$  die smallest enclosing disc für  $P_i$ . Außerdem liege  $q$  auf dem Rand von  $D_i$ .

- Entferne nun ein  $p_j \in P_i$ .
- Wann ändert sich die smallest enclosing disc?

## Rückwärtsanalyse

**Idee:** Fixiere  $P_i := \{p_1, \dots, p_i\}$ . Sei  $D_i$  die smallest enclosing disc für  $P_i$ . Außerdem liege  $q$  auf dem Rand von  $D_i$ .

- Entferne nun ein  $p_j \in P_i$ .
- Wann ändert sich die smallest enclosing disc?

## Rückwärtsanalyse

**Idee:** Fixiere  $P_i := \{p_1, \dots, p_i\}$ . Sei  $D_i$  die smallest enclosing disc für  $P_i$ . Außerdem liege  $q$  auf dem Rand von  $D_i$ .

- Entferne nun ein  $p_j \in P_i$ .
- Wann ändert sich die smallest enclosing disc?

**Erkenntnis:** Höchstens 2 der  $i$  Knoten dafür verantwortlich.

## Rückwärtsanalyse

**Idee:** Fixiere  $P_i := \{p_1, \dots, p_i\}$ . Sei  $D_i$  die smallest enclosing disc für  $P_i$ . Außerdem liege  $q$  auf dem Rand von  $D_i$ .

- Entferne nun ein  $p_j \in P_i$ .
- Wann ändert sich die smallest enclosing disc?

**Erkenntnis:** Höchstens 2 der  $i$  Knoten dafür verantwortlich.

$$\mathcal{O}(i) + \sum_{j=2}^n \left( \mathcal{O}(j) \frac{2}{j} \right) = \mathcal{O}(i)$$

## Rückwärtsanalyse

**Idee:** Fixiere  $P_i := \{p_1, \dots, p_i\}$ . Sei  $D_i$  die smallest enclosing disc für  $P_i$ . Außerdem liege  $q$  auf dem Rand von  $D_i$ .

- Entferne nun ein  $p_j \in P_i$ .
- Wann ändert sich die smallest enclosing disc?

**Erkenntnis:** Höchstens 2 der  $i$  Knoten dafür verantwortlich.

$$\mathcal{O}(i) + \sum_{j=2}^n \left( \mathcal{O}(j) \frac{2}{j} \right) = \mathcal{O}(i)$$

Erwarteter Aufwand für `MINIDISCWITHPOINT`:  $\mathcal{O}(i)$

# Analyse (iii)

**Frage:** Wie wahrscheinlich ist es, dass `MINIDISC`  
`MINIDISCWITHPOINT` aufruft?

# Analyse (iii)

**Frage:** Wie wahrscheinlich ist es, dass `MINIDISC`  
`MINIDISCWITHPOINT` aufruft?

Gleiche Idee liefert gleiches Ergebnis

# Analyse (iii)

**Frage:** Wie wahrscheinlich ist es, dass `MINIDISC`  
`MINIDISCWITHPOINT` aufruft?

Gleiche Idee liefert gleiches Ergebnis

<code>MINIDISC</code>	Erwartet: $\mathcal{O}(n)$
<code>MINIDISCWITHPOINT</code>	Erwartet: $\mathcal{O}(j)$
<code>MINIDISCWITH2POINTS</code>	$\mathcal{O}(i)$

# Analyse (iii)

**Frage:** Wie wahrscheinlich ist es, dass `MINIDISC`  
`MINIDISCWITHPOINT` aufruft?

Gleiche Idee liefert gleiches Ergebnis

<code>MINIDISC</code>	Erwartet: $\mathcal{O}(n)$
<code>MINIDISCWITHPOINT</code>	Erwartet: $\mathcal{O}(j)$
<code>MINIDISCWITH2POINTS</code>	$\mathcal{O}(i)$

**Thm:** Die SED für  $n$  Punkte in der Ebene kann  
in erwartet  $\mathcal{O}(n)$  Zeit berechnet werden.



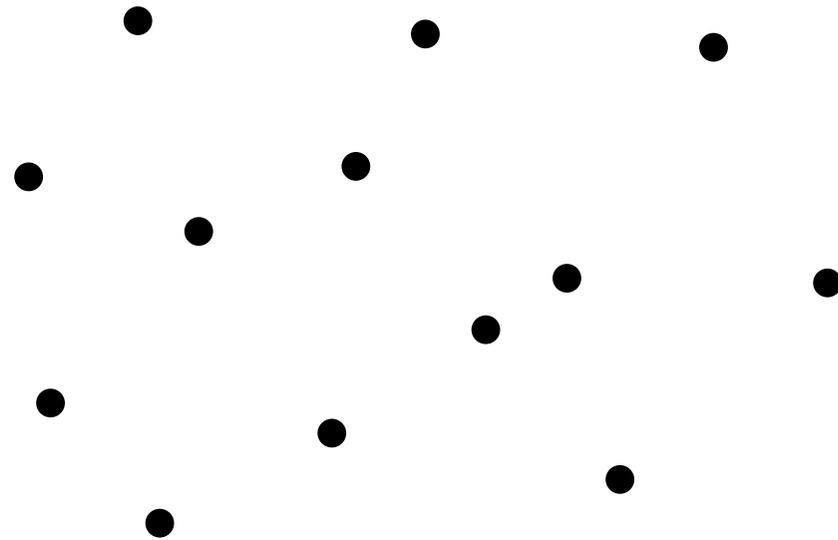
Smallest Enclosing Disk

Konvexe Hülle im  $\mathbb{R}^3$

# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^2$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

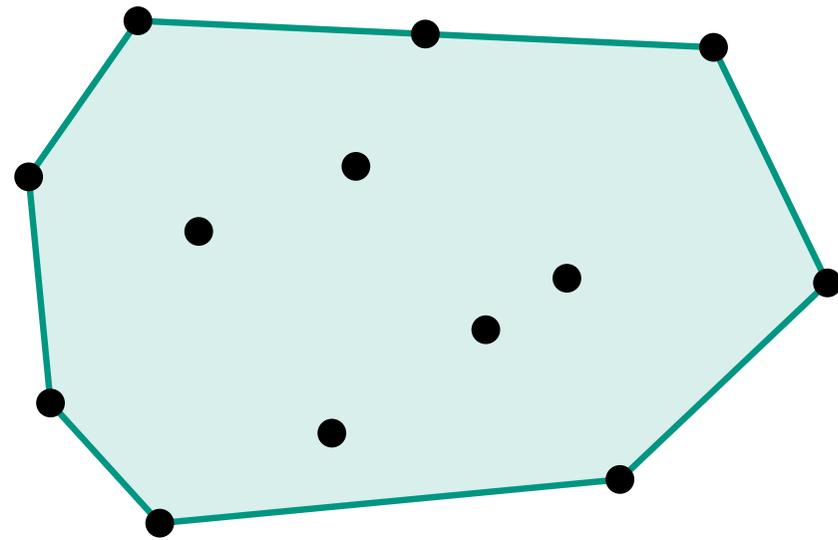
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^2$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

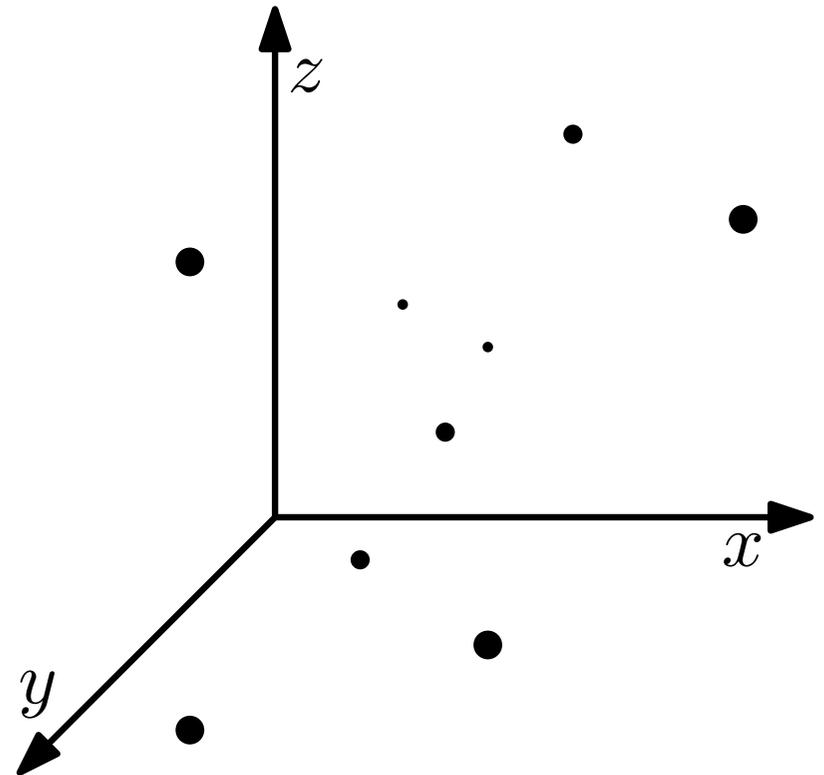
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

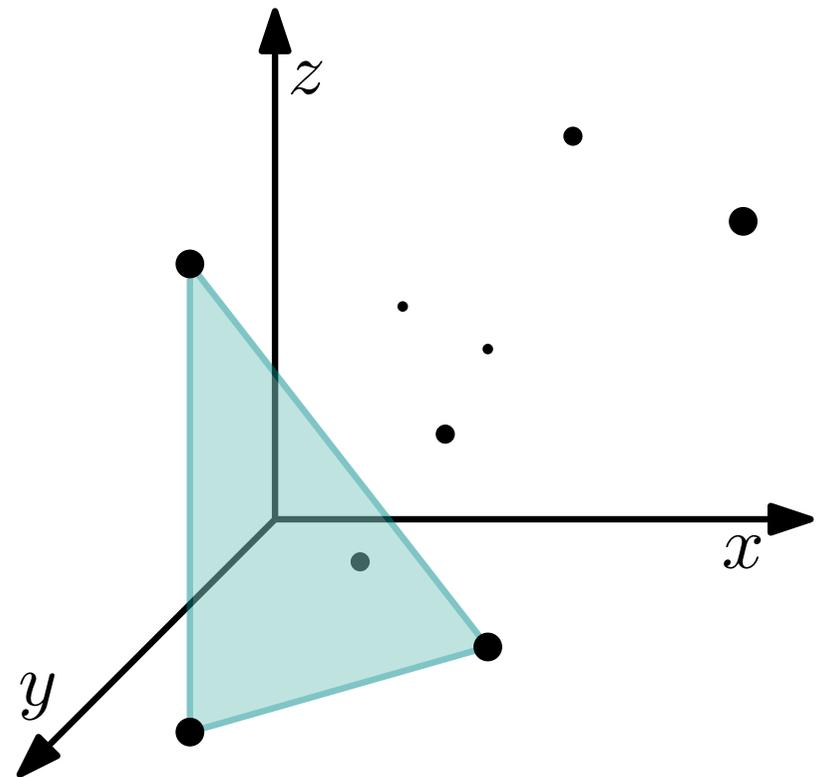
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

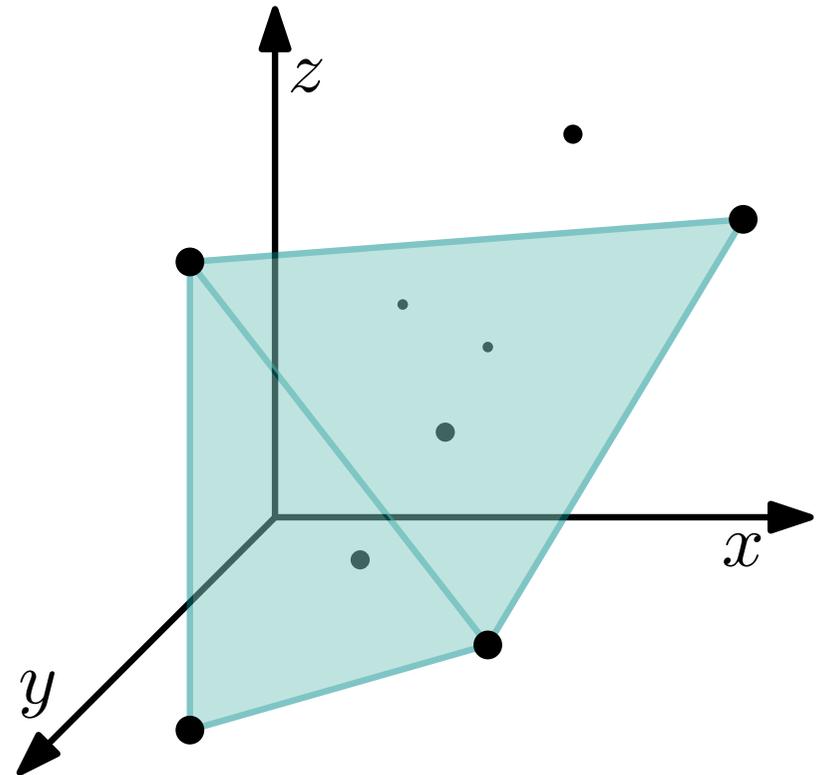
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

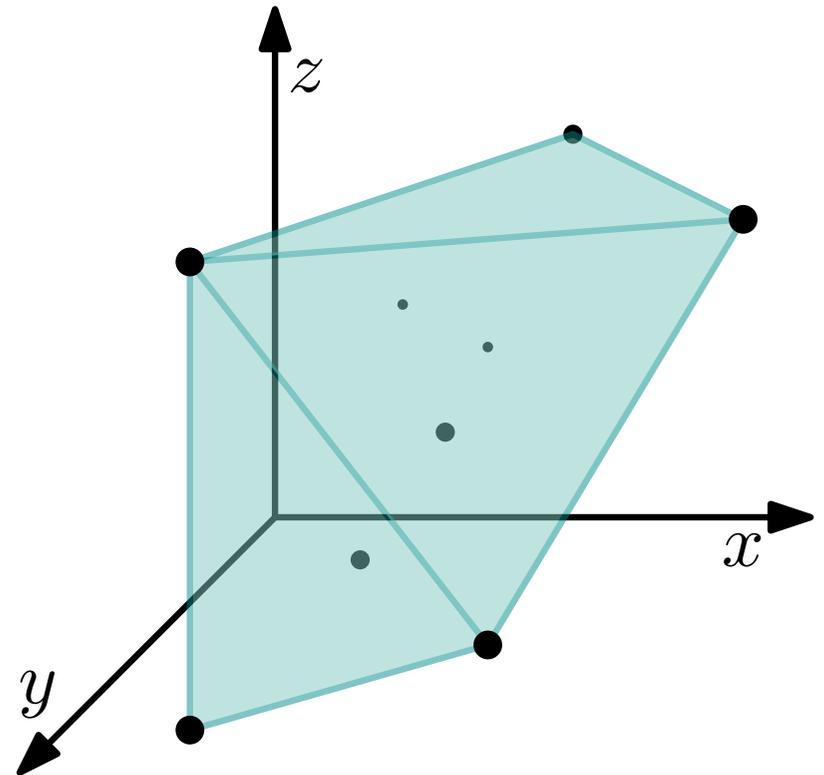
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

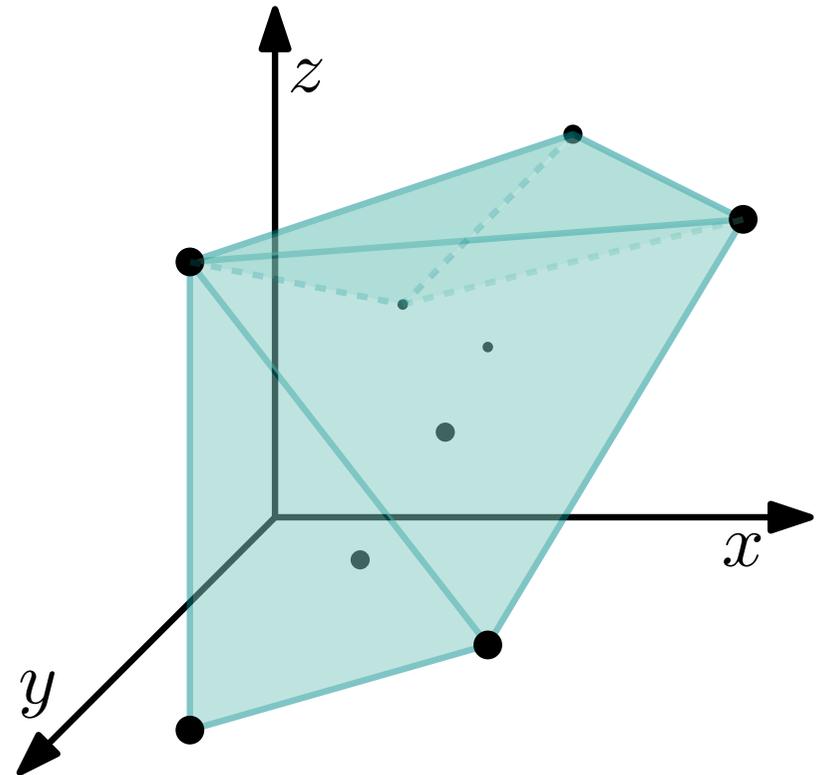
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

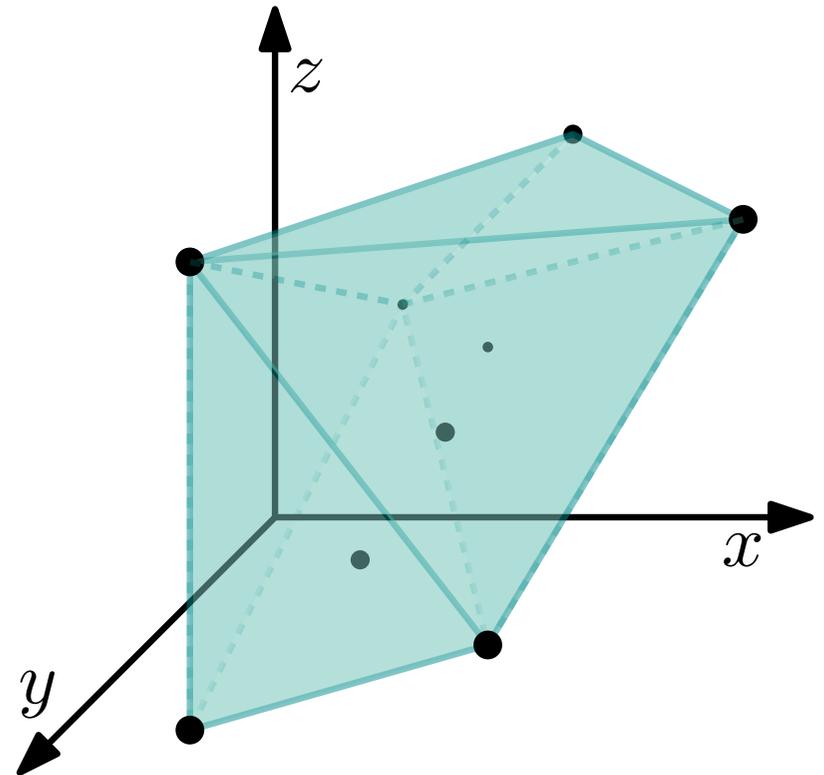
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

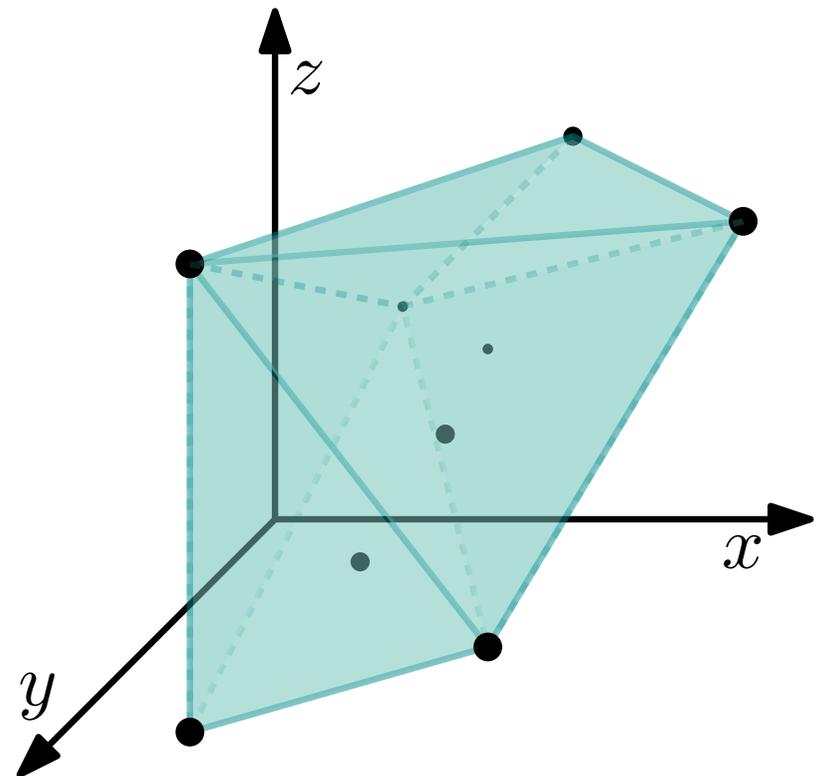
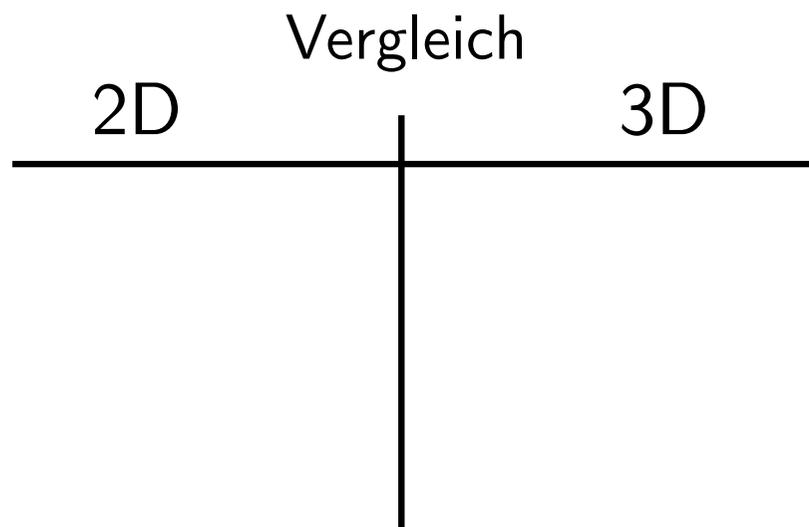
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

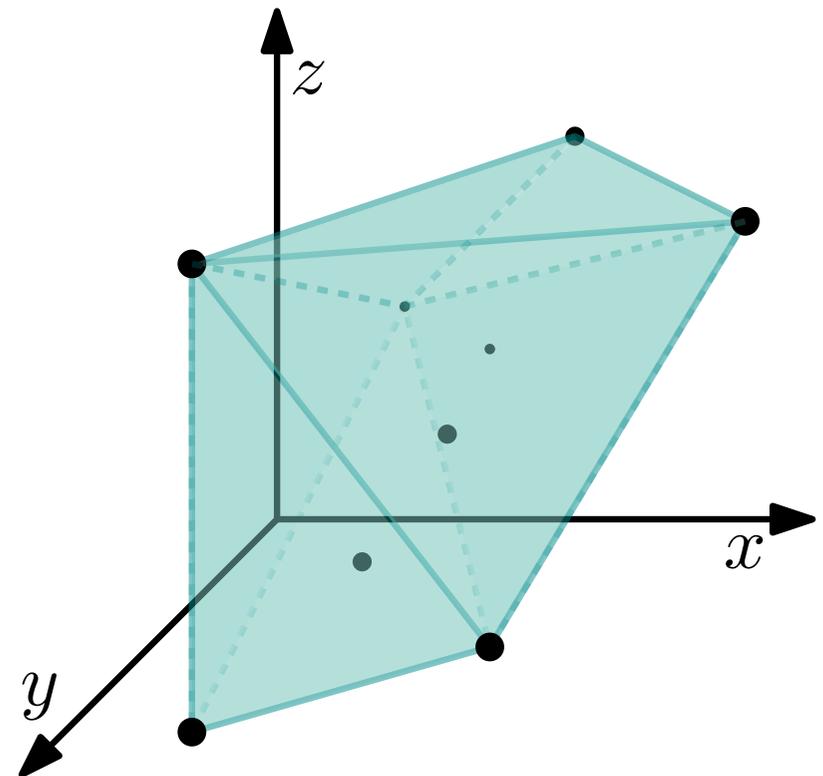
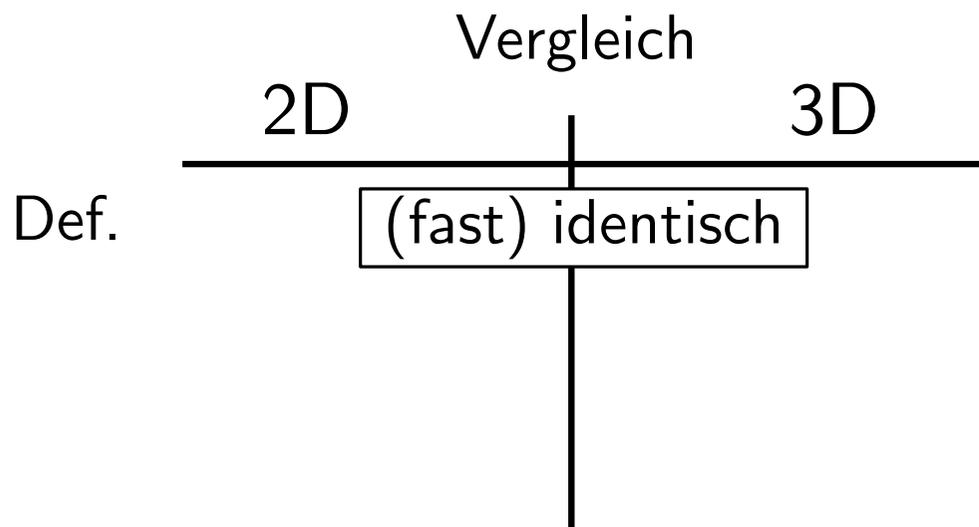
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

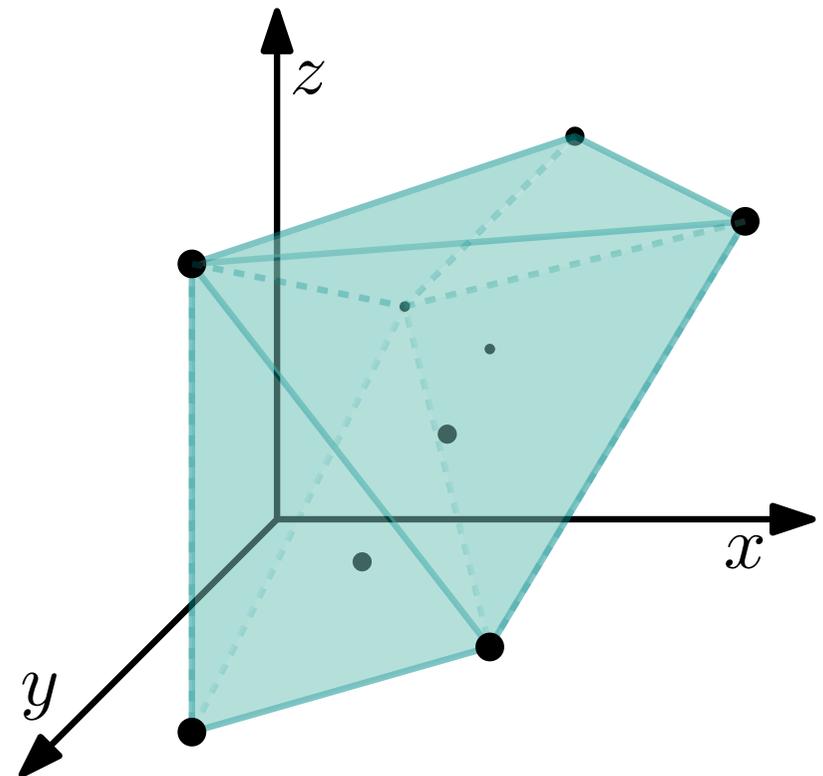
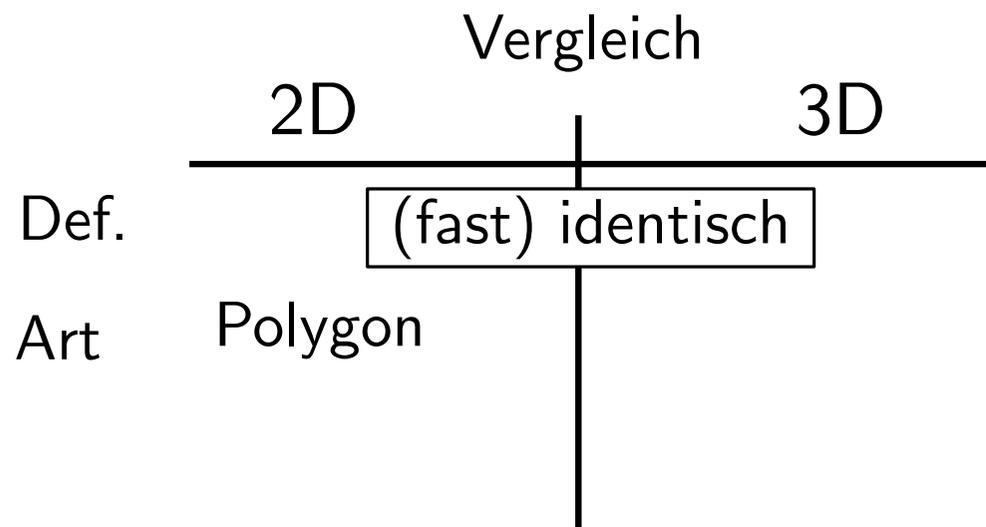
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

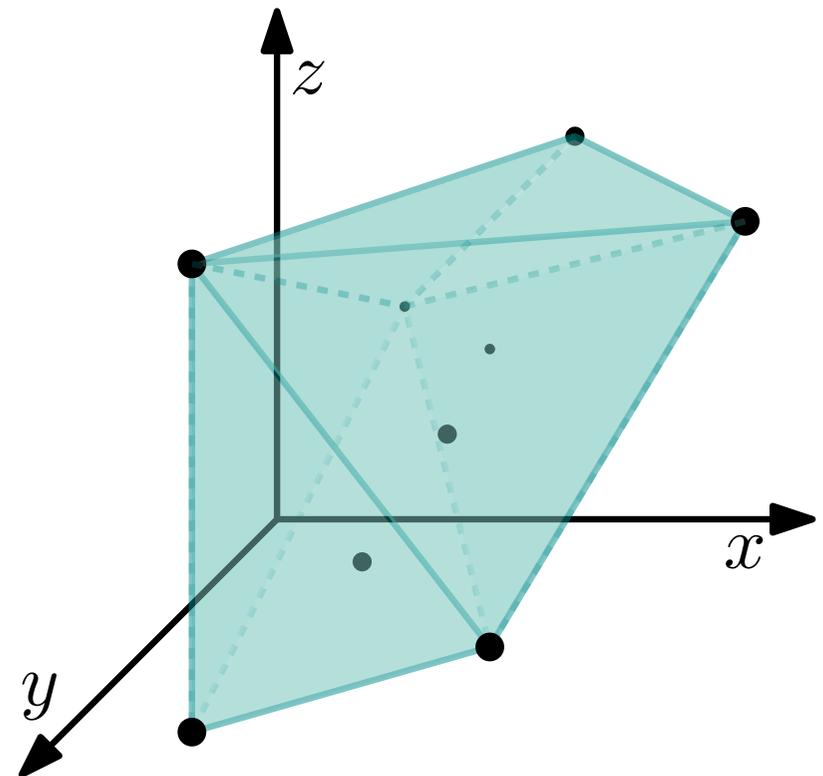
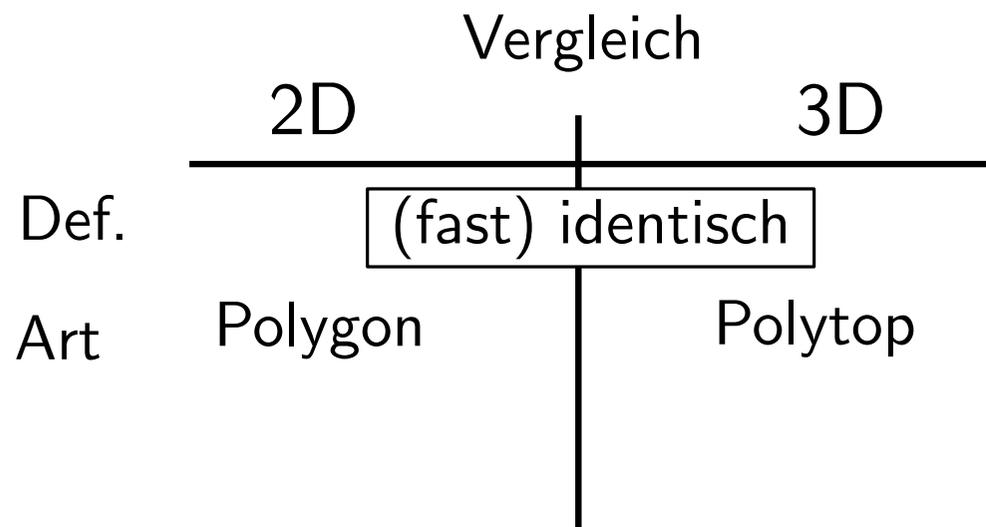
Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.



# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.

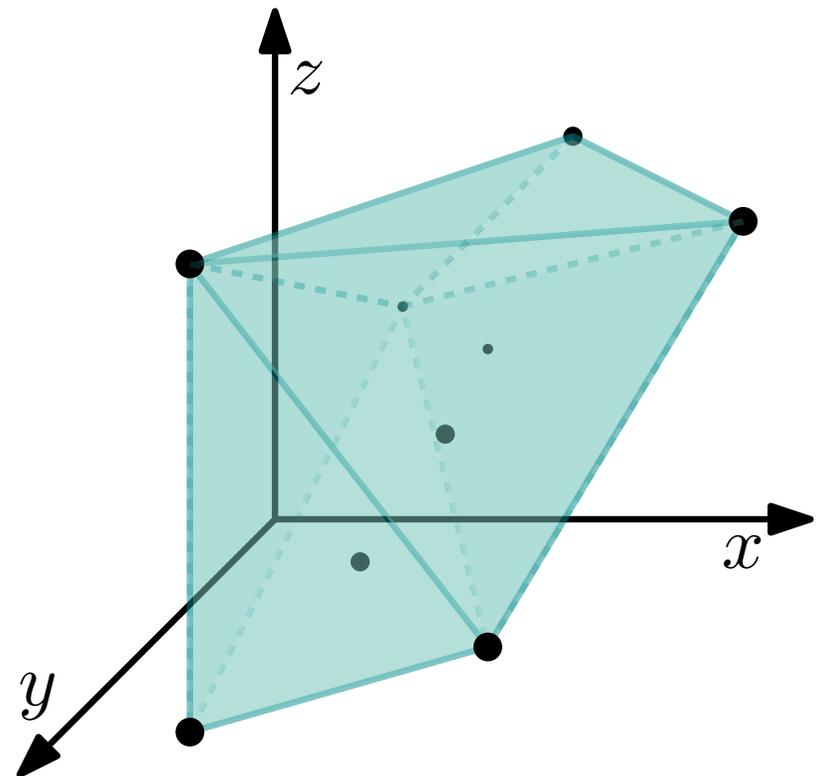


# Die konvexe Hülle

**Def:** Eine Menge  $S \subseteq \mathbb{R}^3$  heißt **konvex**, wenn für je zwei Punkte  $p, q \in S$  auch gilt  $\overline{pq} \in S$ .

Die **konvexe Hülle**  $CH(S)$  von  $S$  ist die kleinste konvexe Menge, die  $S$  enthält.

	Vergleich	
	2D	3D
Def.	(fast) identisch	
Art	Polygon	Polytop
Kompl.	$\mathcal{O}(n)$ Kanten	?



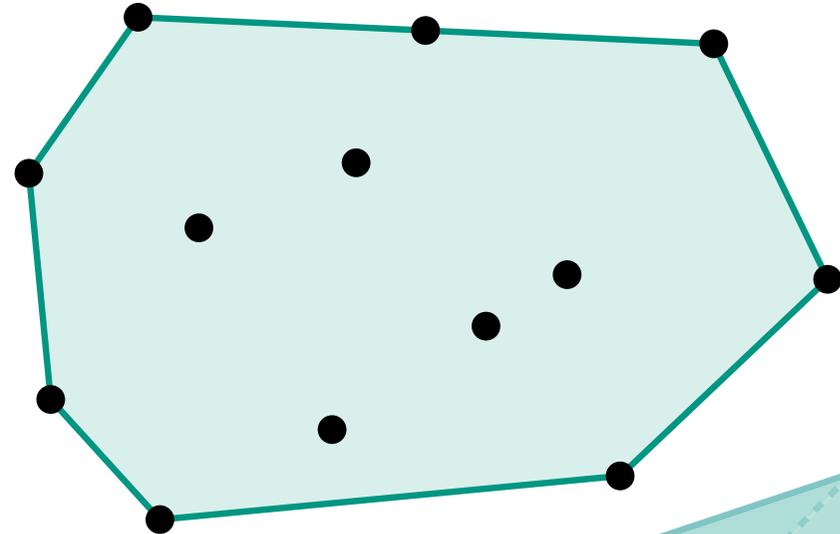
# Komplexität konvexer Hüllen

Dimension	Komplexität
1	?
2	$\mathcal{O}(n)$
3	?

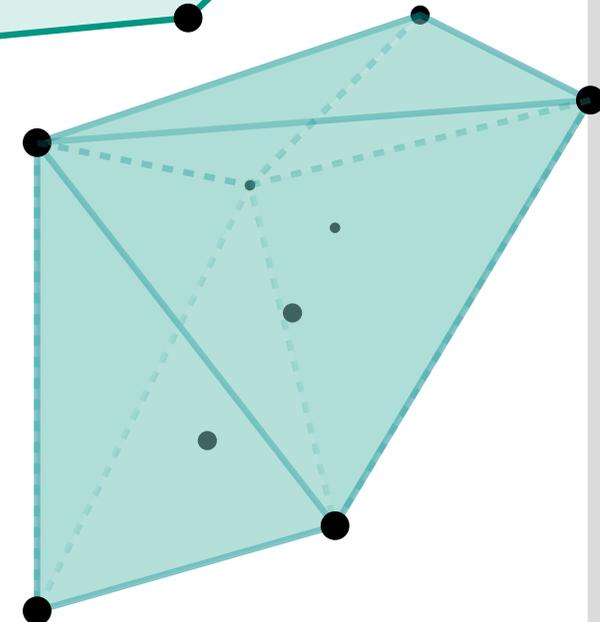
1D



2D



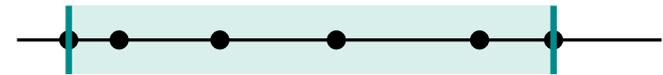
3D



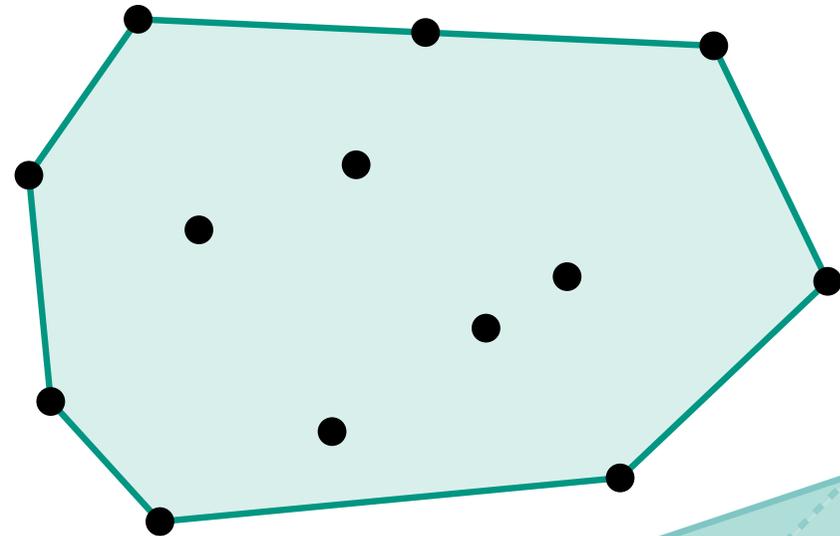
# Komplexität konvexer Hüllen

Dimension	Komplexität
1	?
2	$\mathcal{O}(n)$
3	?

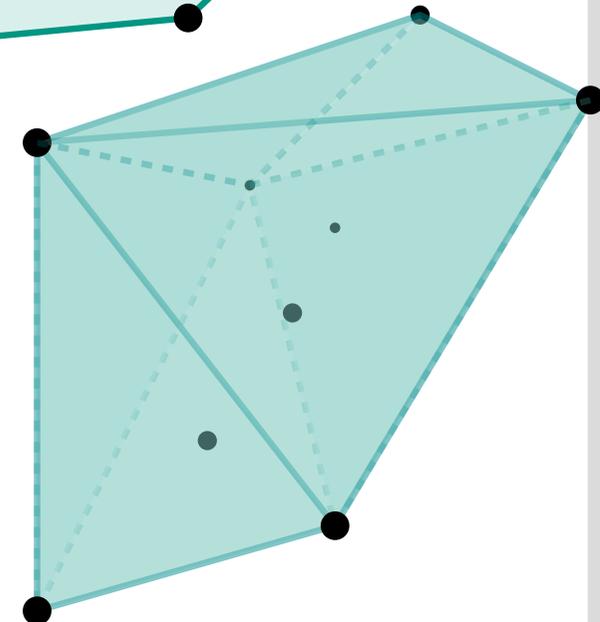
1D



2D



3D



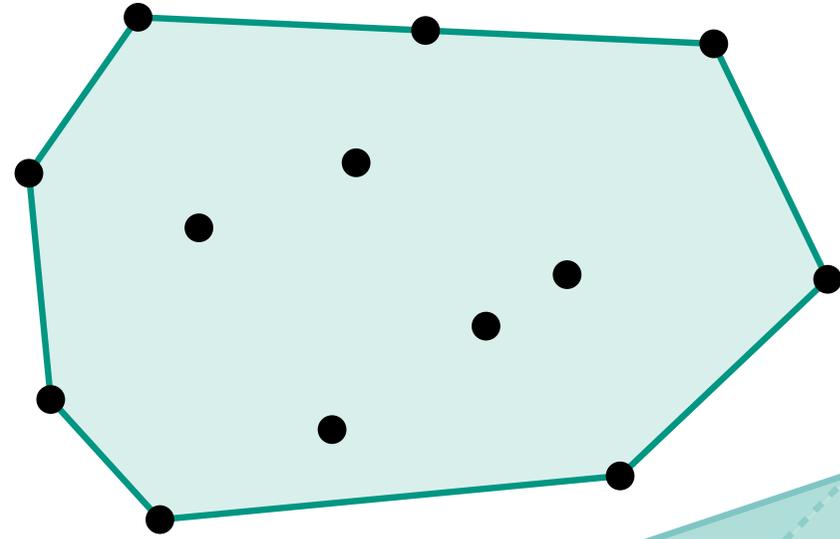
# Komplexität konvexer Hüllen

Dimension	Komplexität
1	$\mathcal{O}(1)$
2	$\mathcal{O}(n)$
3	?

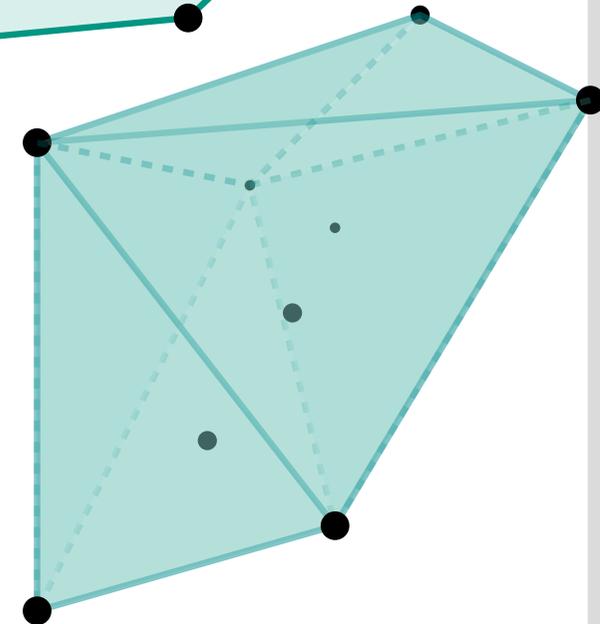
1D



2D



3D



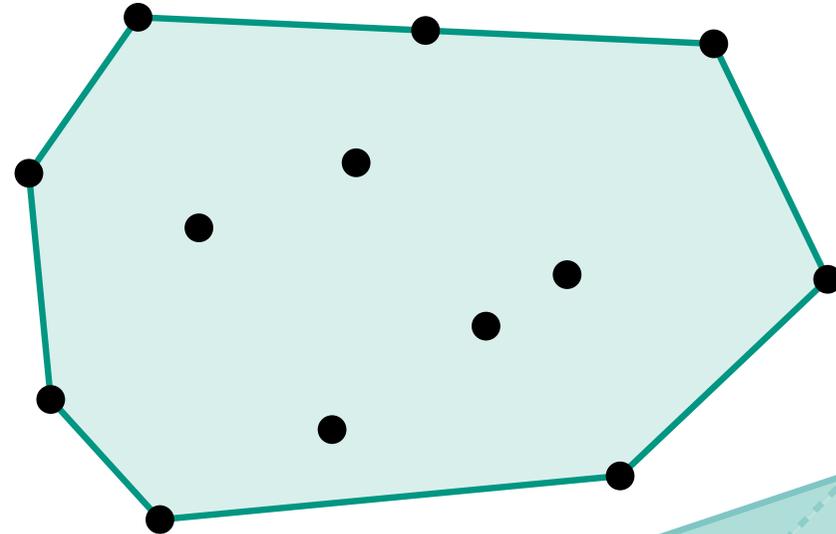
# Komplexität konvexer Hüllen

Dimension	Komplexität
1	$\mathcal{O}(1)$
2	$\mathcal{O}(n)$
3	?

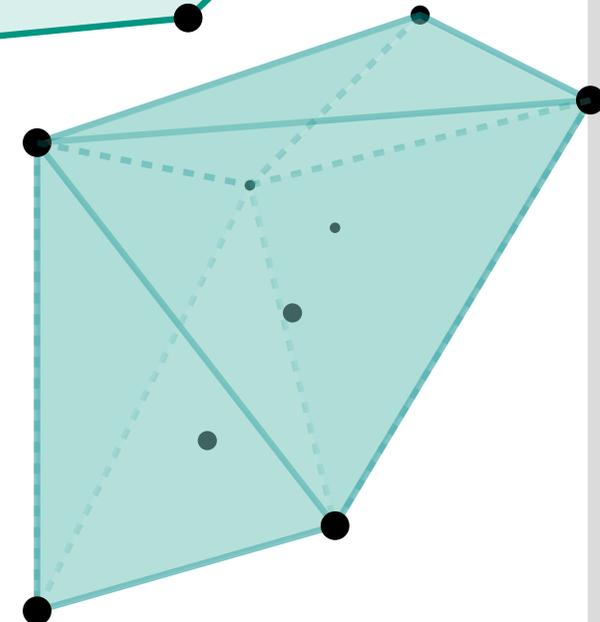
1D



2D



3D



## Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten.  
Dann ist  $\#$ Kanten in  $\mathcal{P}$  höchstens  $3n - 6$   
und  $\#$ Facetten höchstens  $2n - 4$ .

# Komplexität konvexer Hüllen in 3D

## Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist  $\#$ Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und  $\#$ Facetten höchstens  $2n - 4$ .

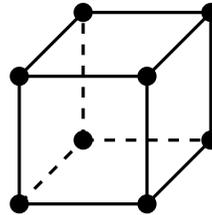
## Beweis:

# Komplexität konvexer Hüllen in 3D

## Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist  $\#$ Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und  $\#$ Facetten höchstens  $2n - 4$ .

## Beweis:

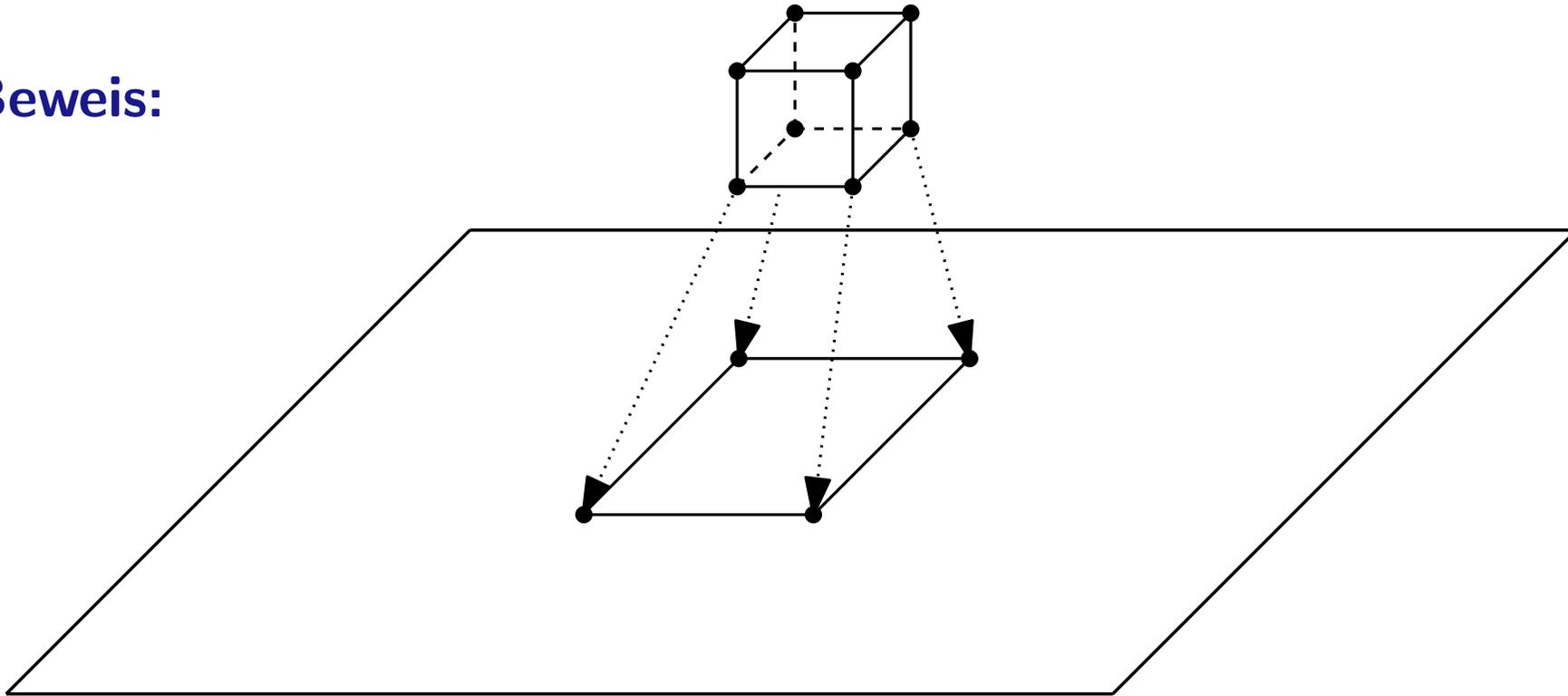


# Komplexität konvexer Hüllen in 3D

## Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist  $\#$ Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und  $\#$ Facetten höchstens  $2n - 4$ .

## Beweis:

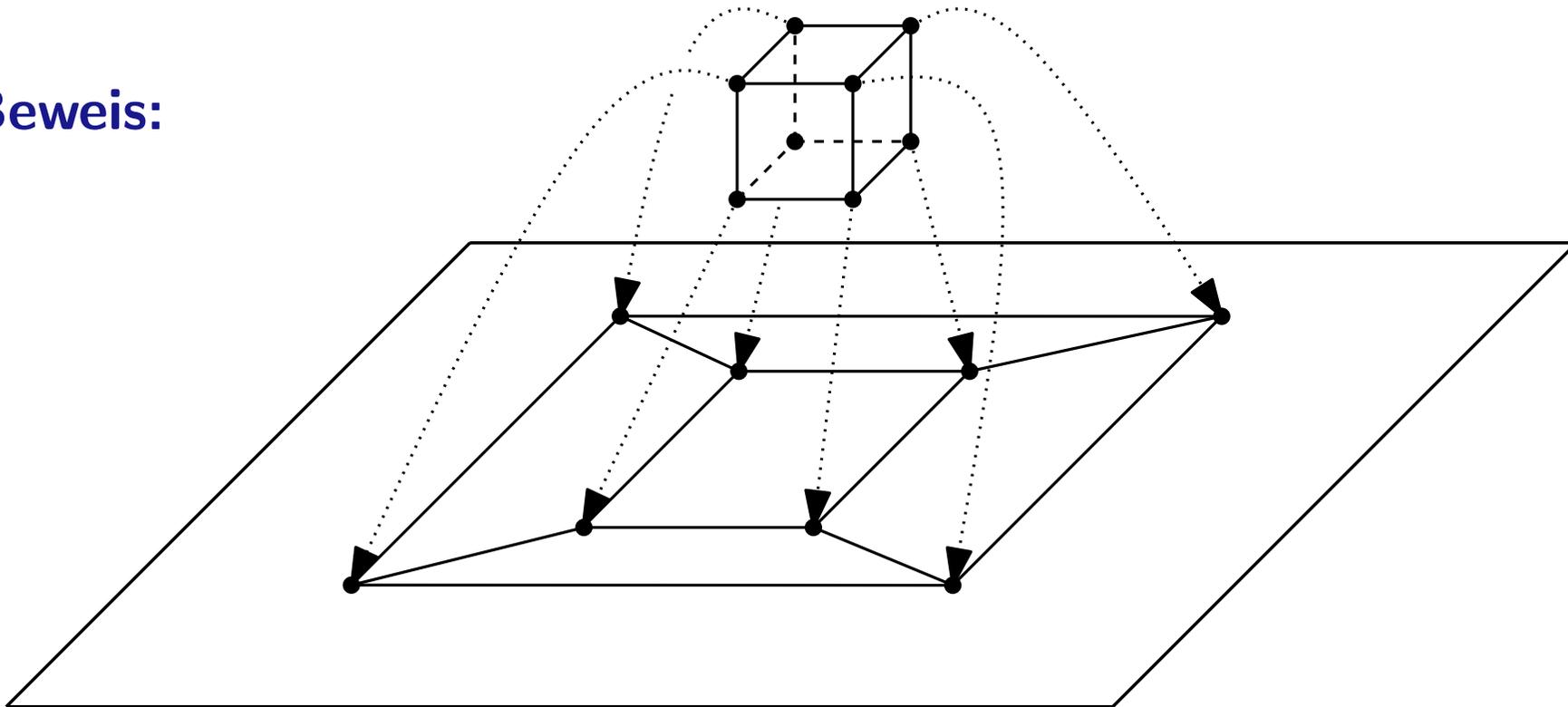


# Komplexität konvexer Hüllen in 3D

## Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist  $\#$ Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und  $\#$ Facetten höchstens  $2n - 4$ .

## Beweis:

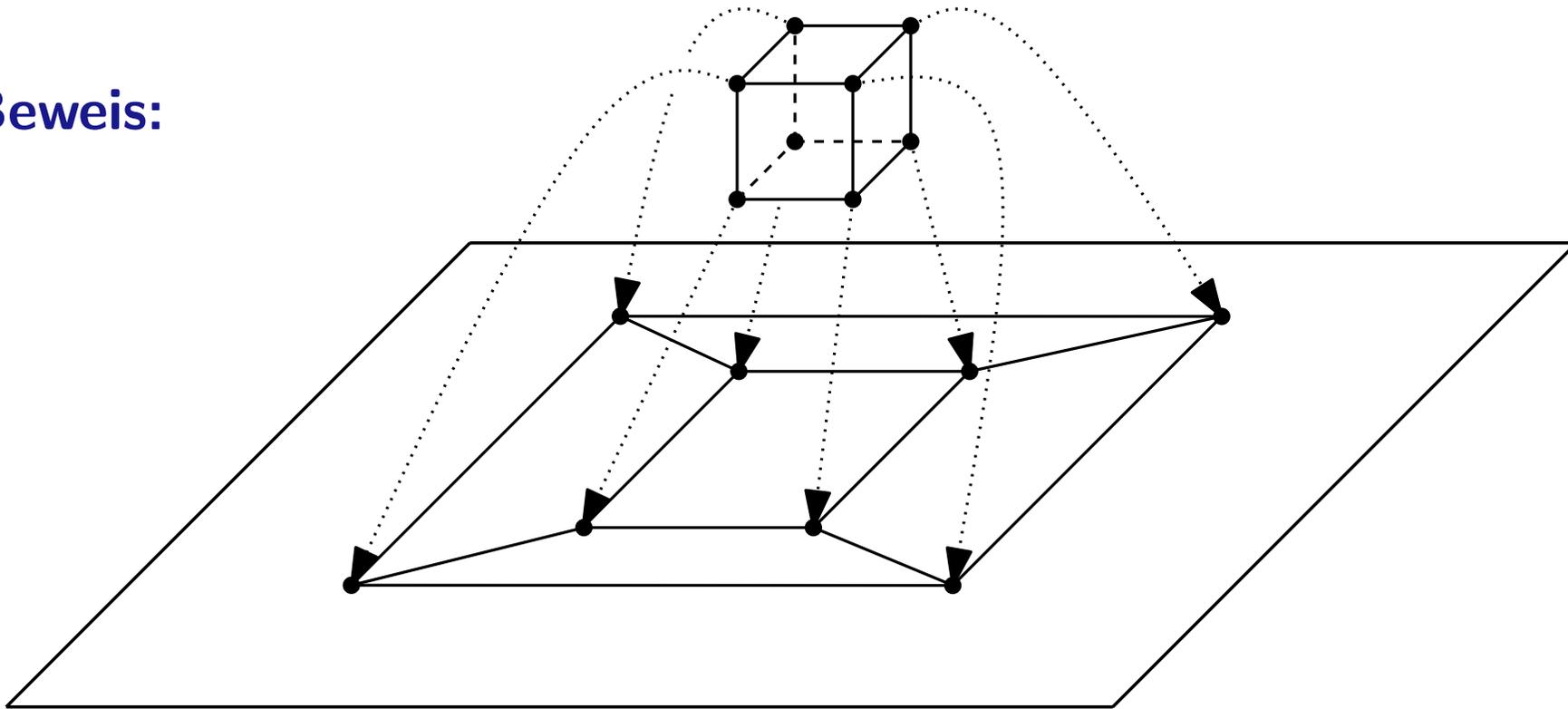


# Komplexität konvexer Hüllen in 3D

## Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist  $\#$ Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und  $\#$ Facetten höchstens  $2n - 4$ .

## Beweis:



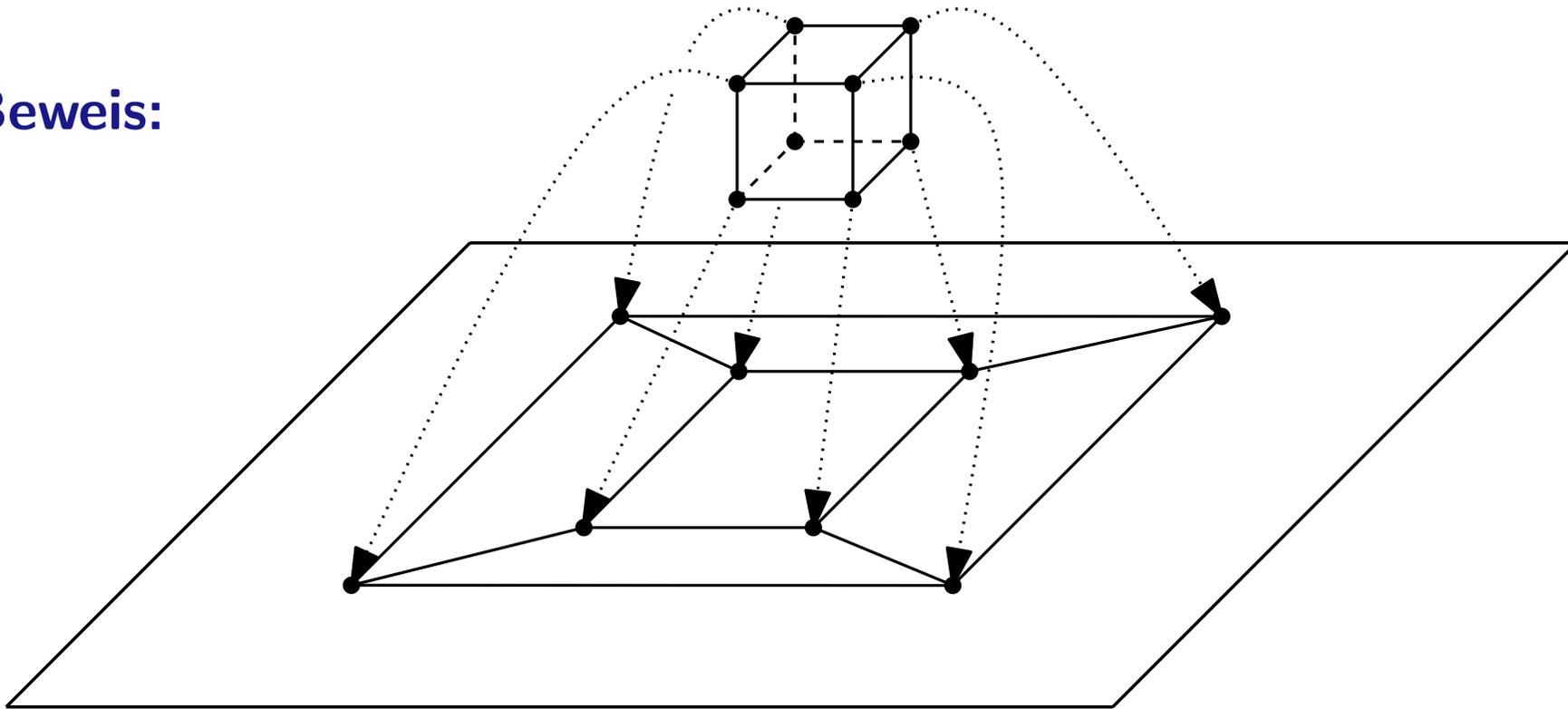
s. Tafel

# Komplexität konvexer Hüllen in 3D

## Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist  $\#$ Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und  $\#$ Facetten höchstens  $2n - 4$ .

## Beweis:



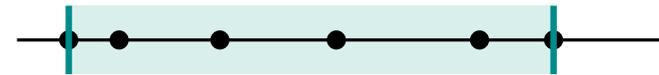
s. Tafel



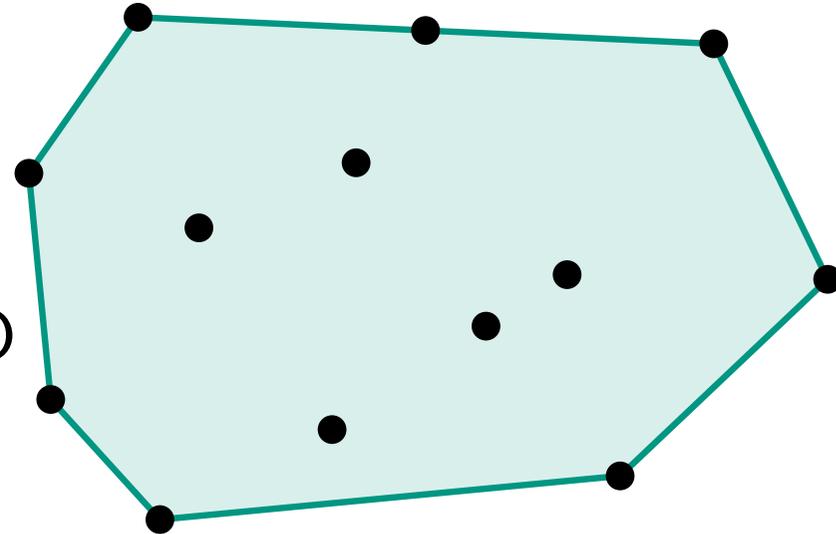
# Komplexität konvexer Hüllen

Dimension	Komplexität
1	$\mathcal{O}(1)$
2	$\mathcal{O}(n)$
3	?

1D



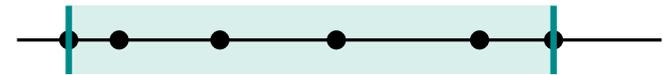
2D



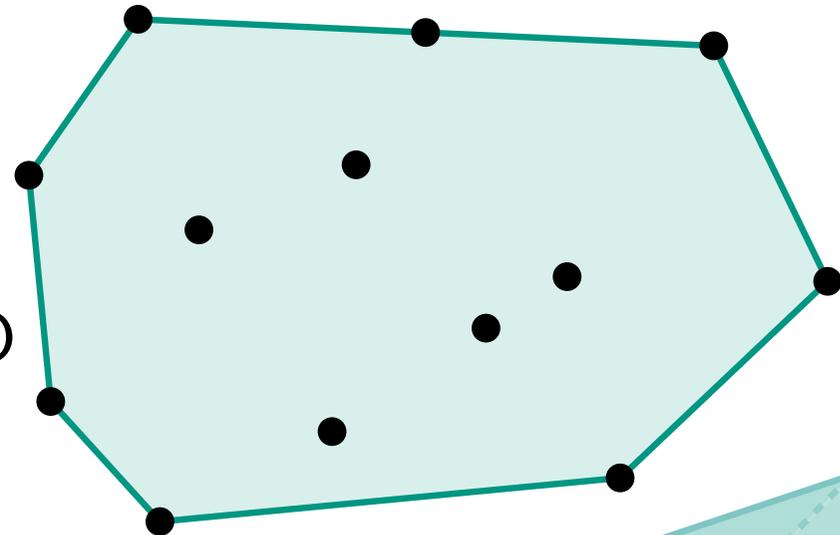
# Komplexität konvexer Hüllen

Dimension	Komplexität
1	$\mathcal{O}(1)$
2	$\mathcal{O}(n)$
3	$\mathcal{O}(n)$

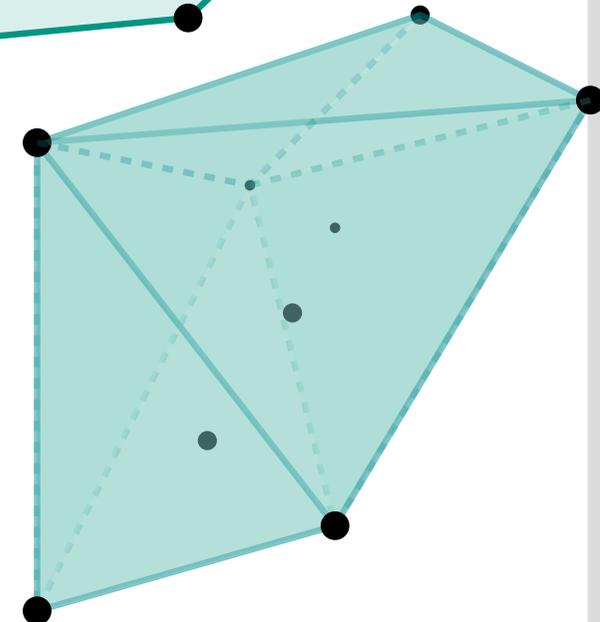
1D



2D



3D



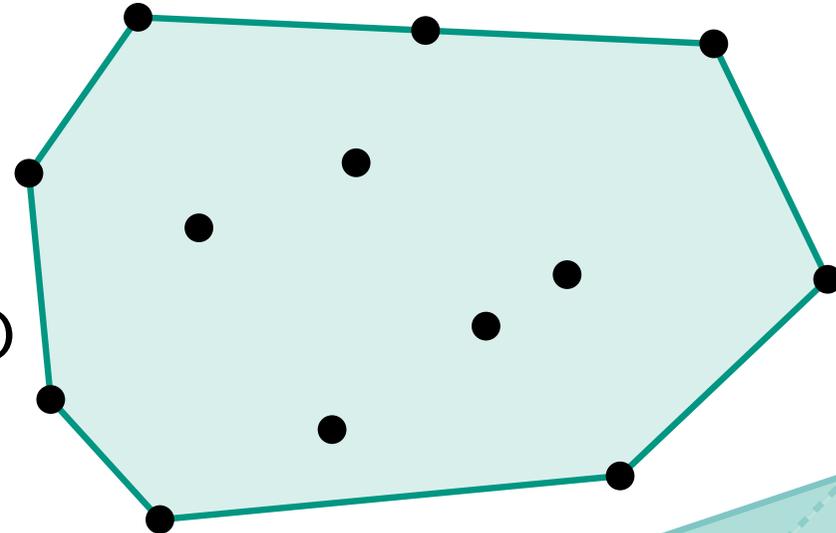
# Komplexität konvexer Hüllen

Dimension	Komplexität
1	$\mathcal{O}(1)$
2	$\mathcal{O}(n)$
3	$\mathcal{O}(n)$
$d$	?

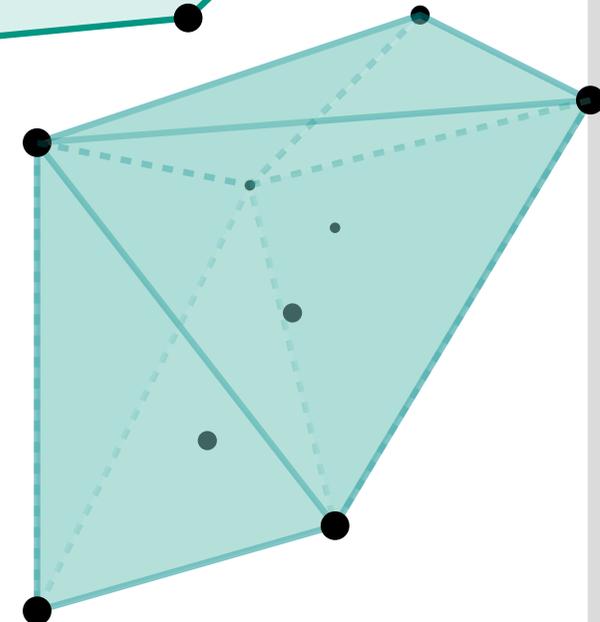
1D



2D



3D



# Komplexität konvexer Hüllen

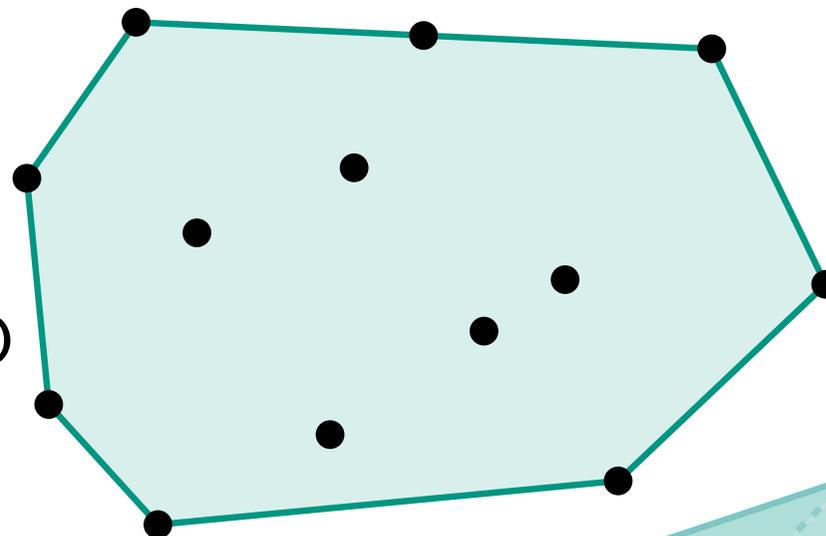
Dimension	Komplexität
1	$\mathcal{O}(1)$
2	$\mathcal{O}(n)$
3	$\mathcal{O}(n)$
$d$	$\Theta(n^{\lfloor d/2 \rfloor})$

Upper Bound Theorem

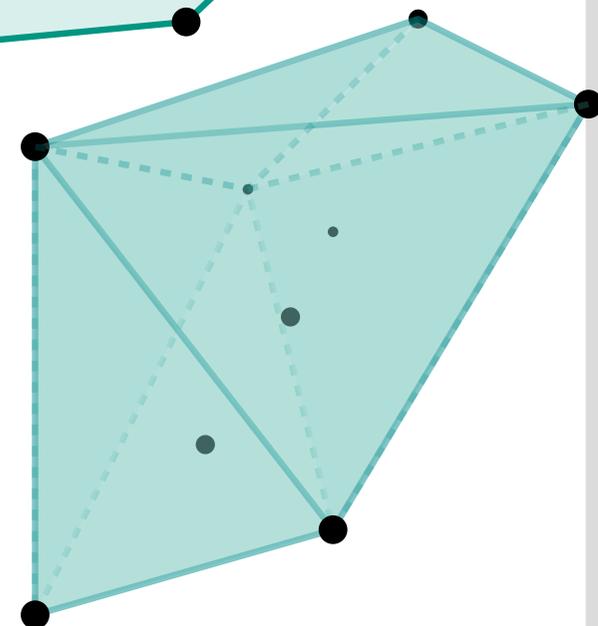
1D



2D



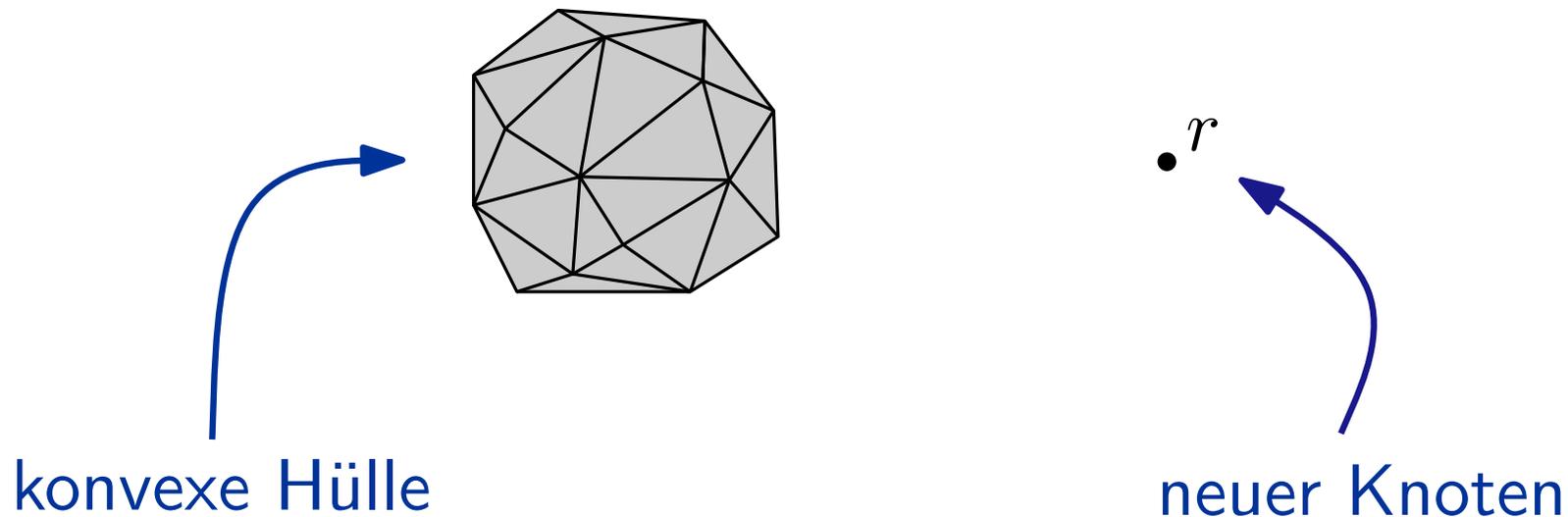
3D



# Erster Schritt

## Angenommen:

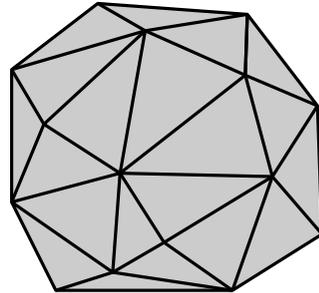
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



# Erster Schritt

## Angenommen:

Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?

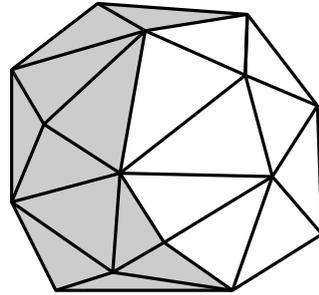


•  $r$

# Erster Schritt

## Angenommen:

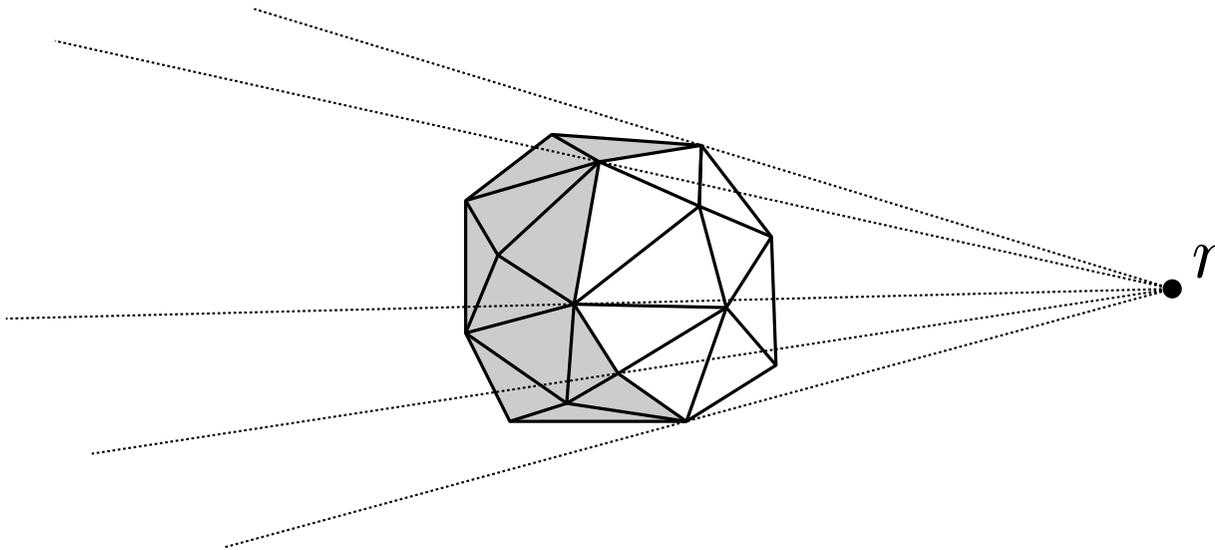
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



# Erster Schritt

## Angenommen:

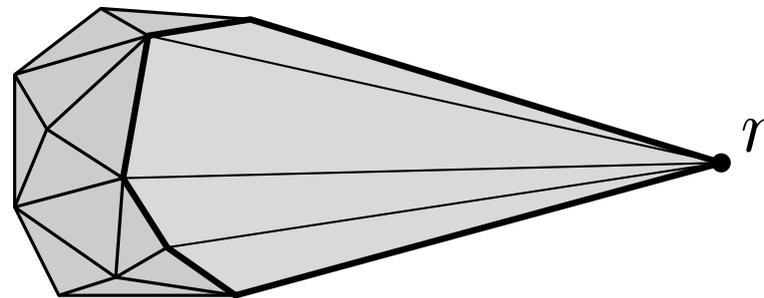
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



# Erster Schritt

## Angenommen:

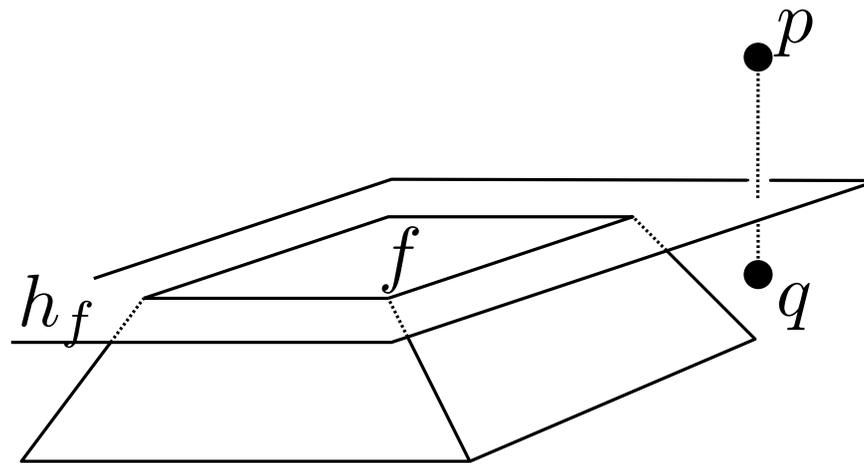
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



# Erster Schritt

## Angenommen:

Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?

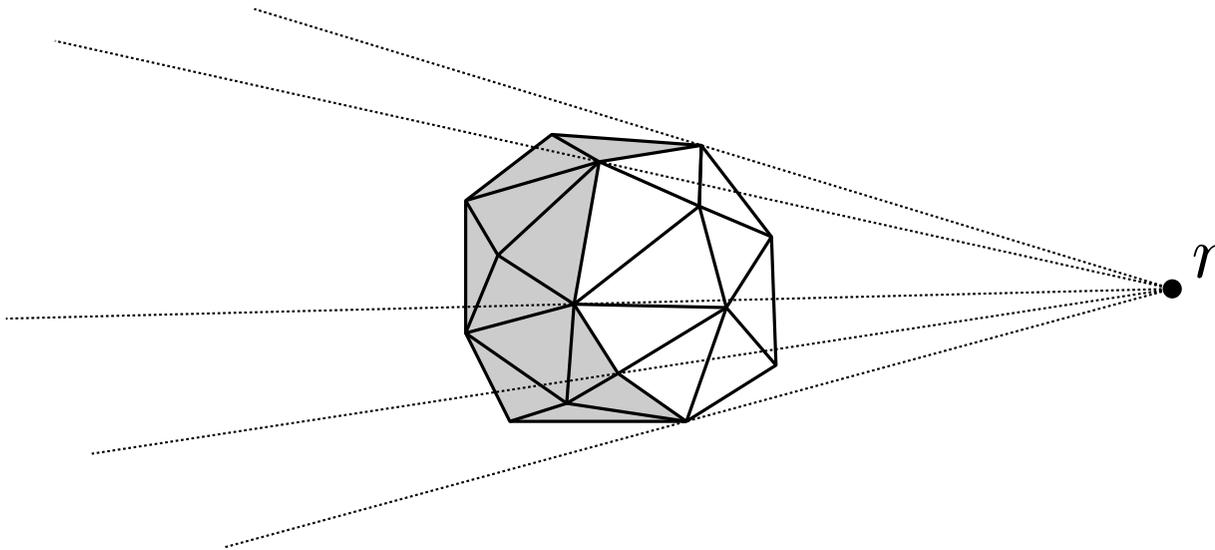


Facette  $f$  sichtbar von  $p$  aber nicht von  $q$  aus

# Erster Schritt

## Angenommen:

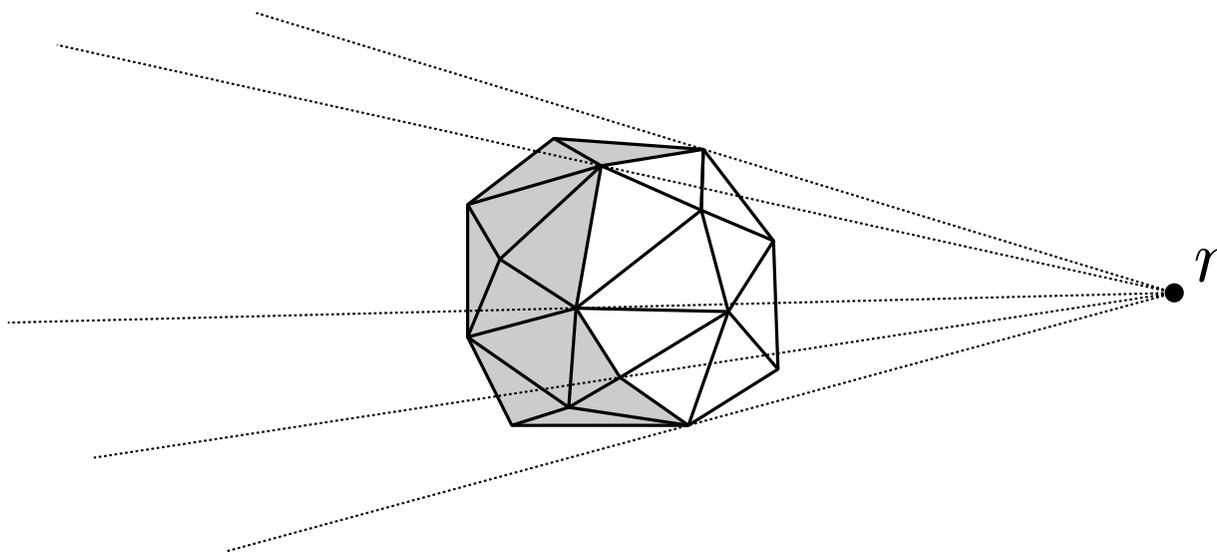
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



# Erster Schritt

## Angenommen:

Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



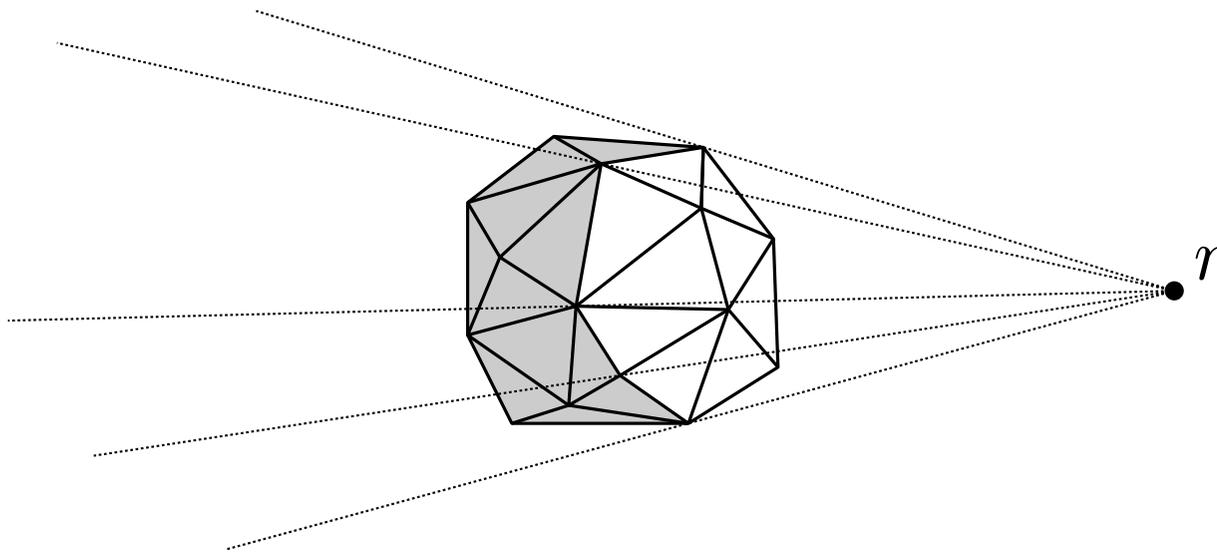
## Aufgabe:

Überlege einen  $\mathcal{O}(n^2)$  Algorithmus der die konvexe Hülle berechnet.

# Erster Schritt

## Angenommen:

Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?

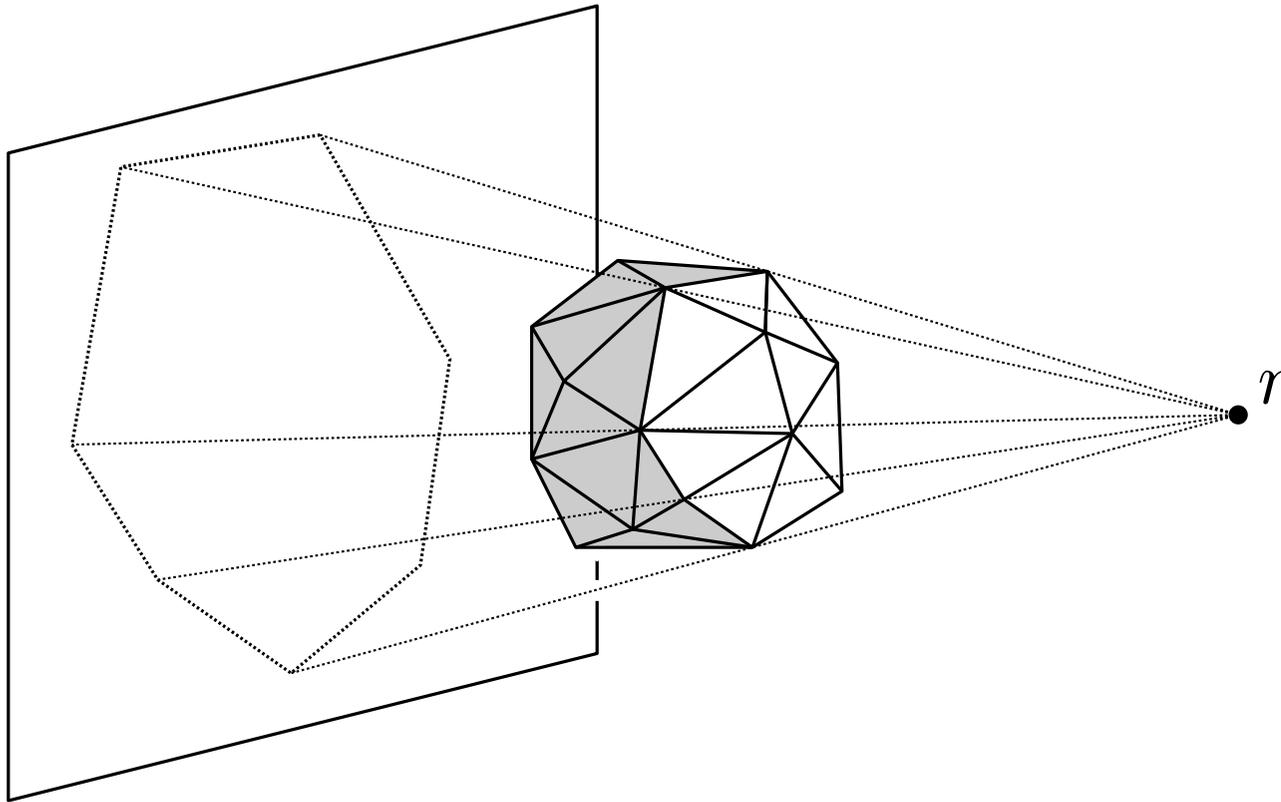


Geht's schneller?

# Erster Schritt

## Angenommen:

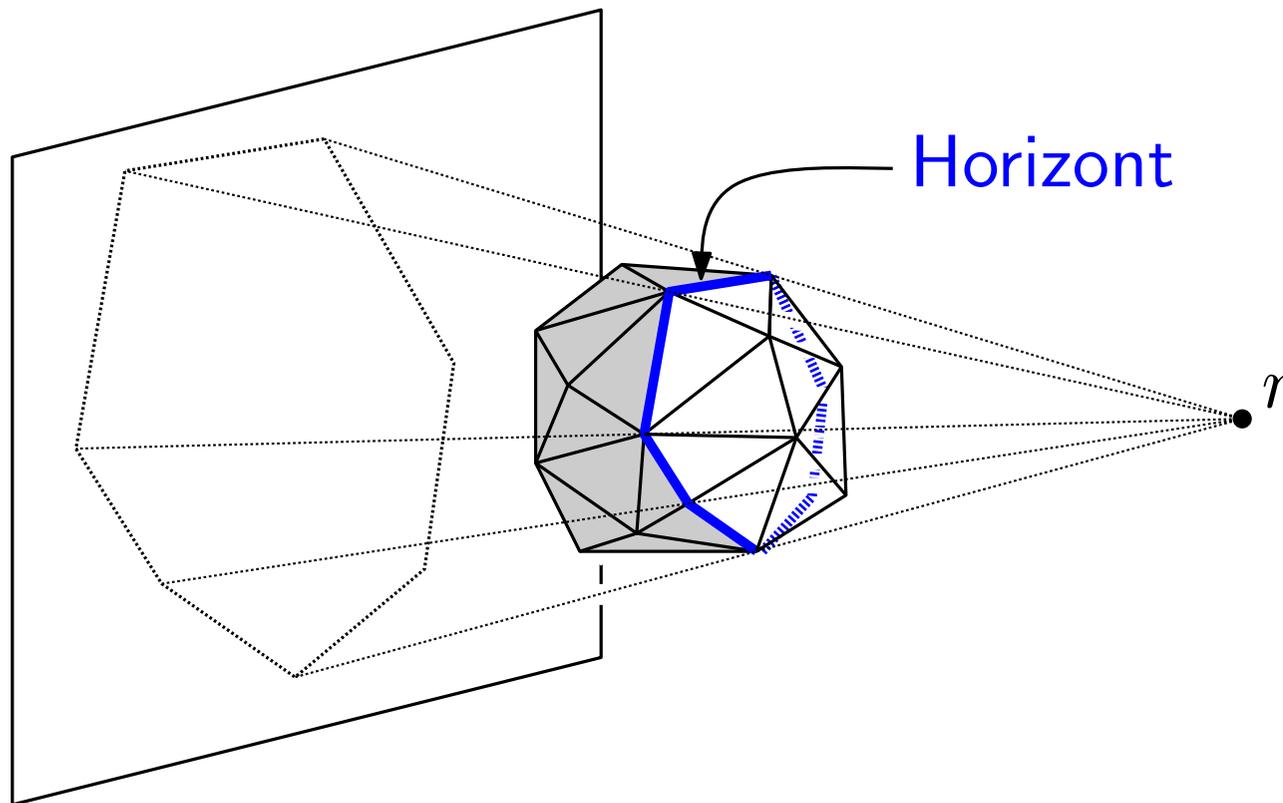
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



# Erster Schritt

## Angenommen:

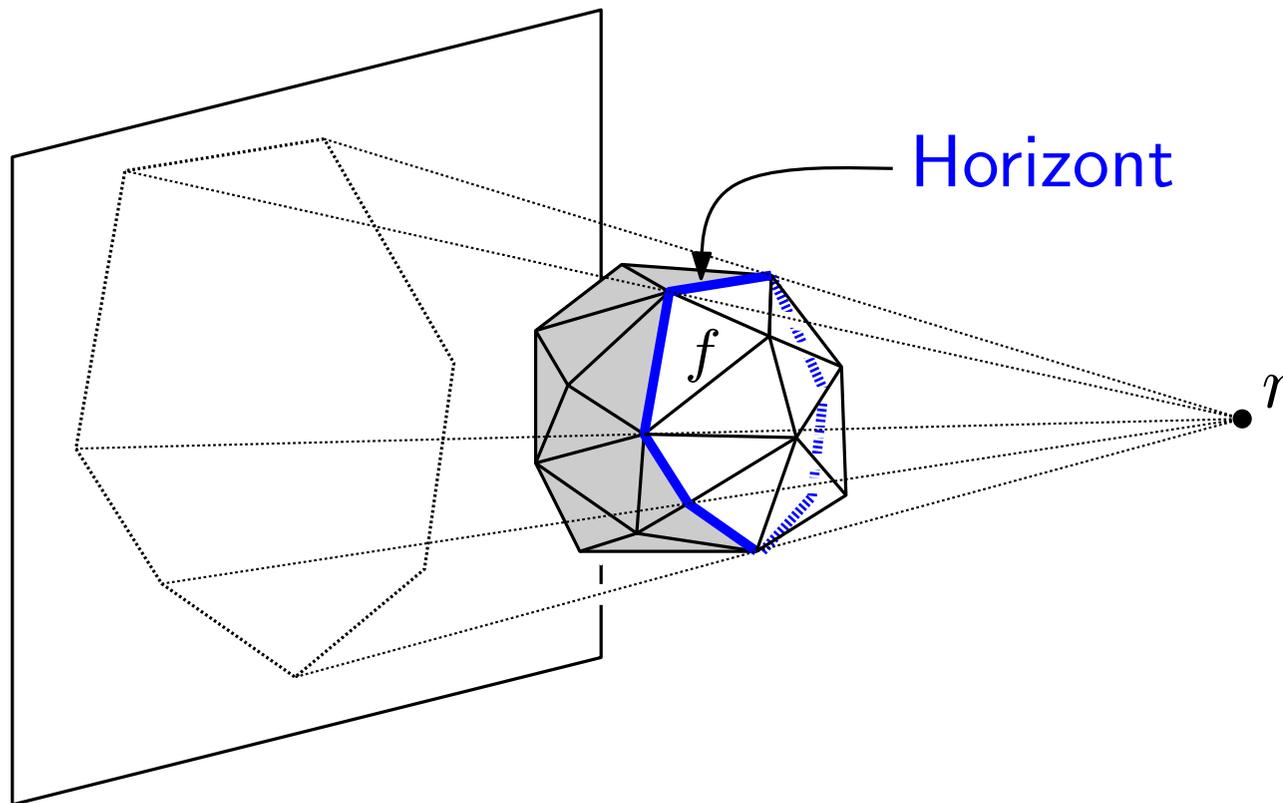
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



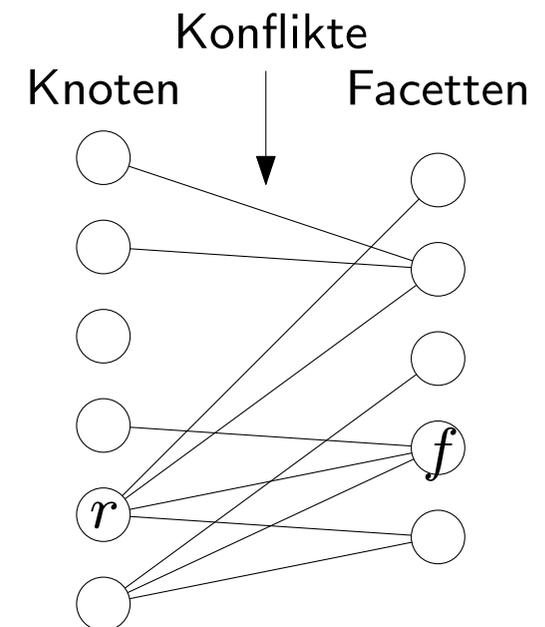
# Erster Schritt

## Angenommen:

Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



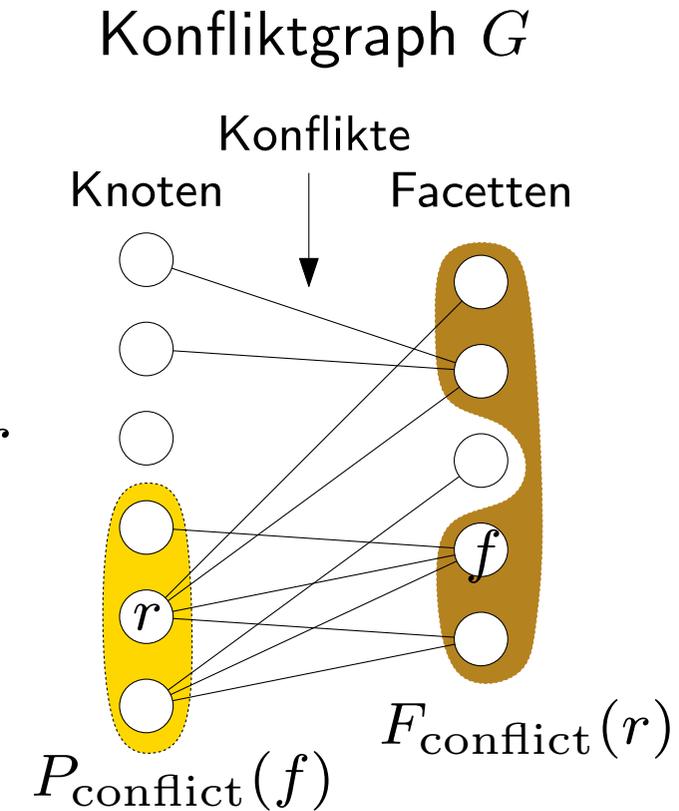
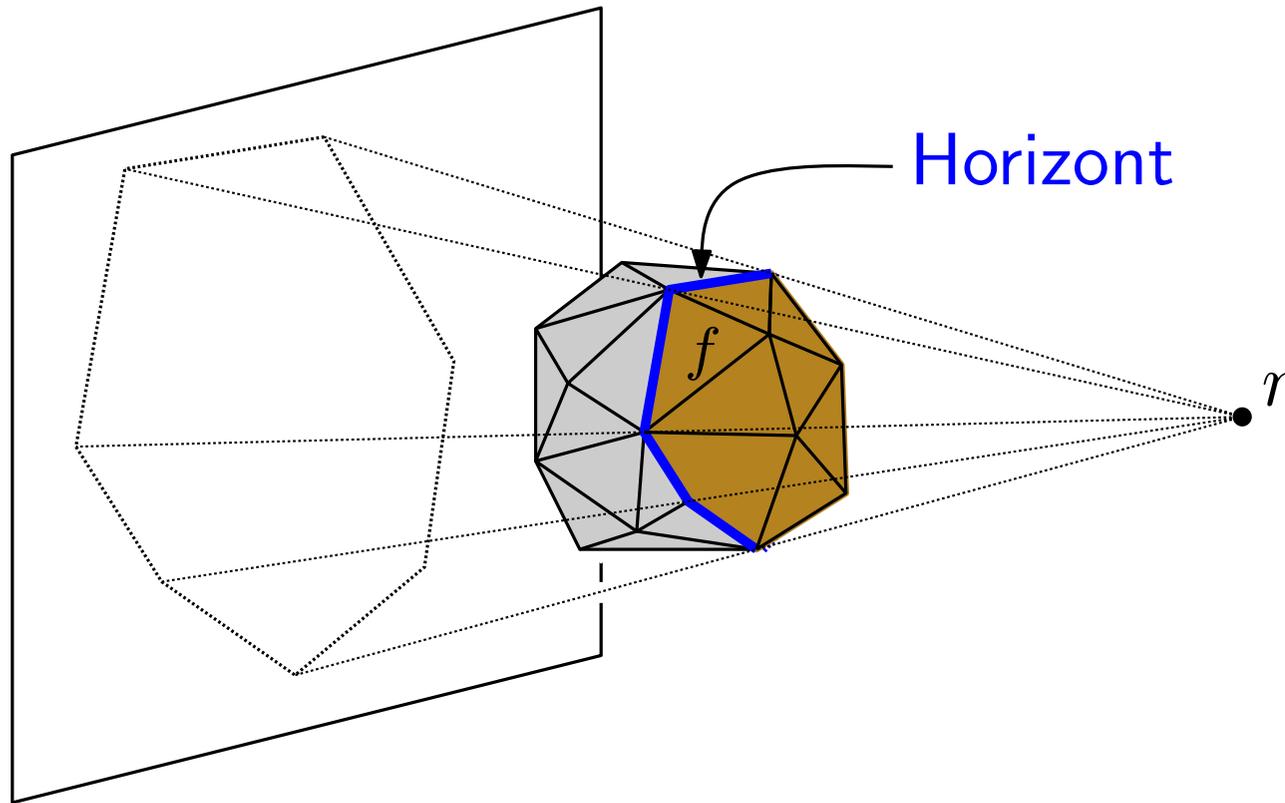
## Konfliktgraph $G$



# Erster Schritt

## Angenommen:

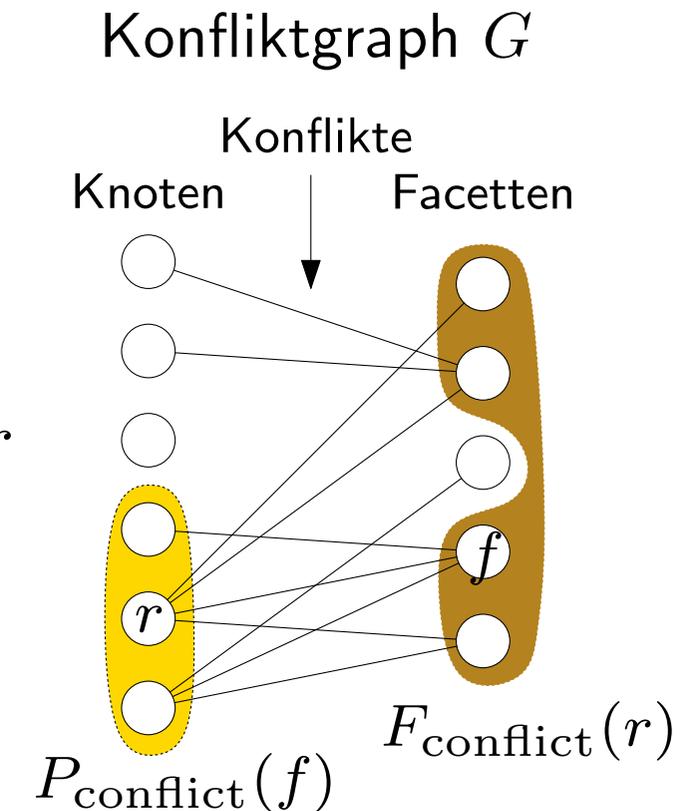
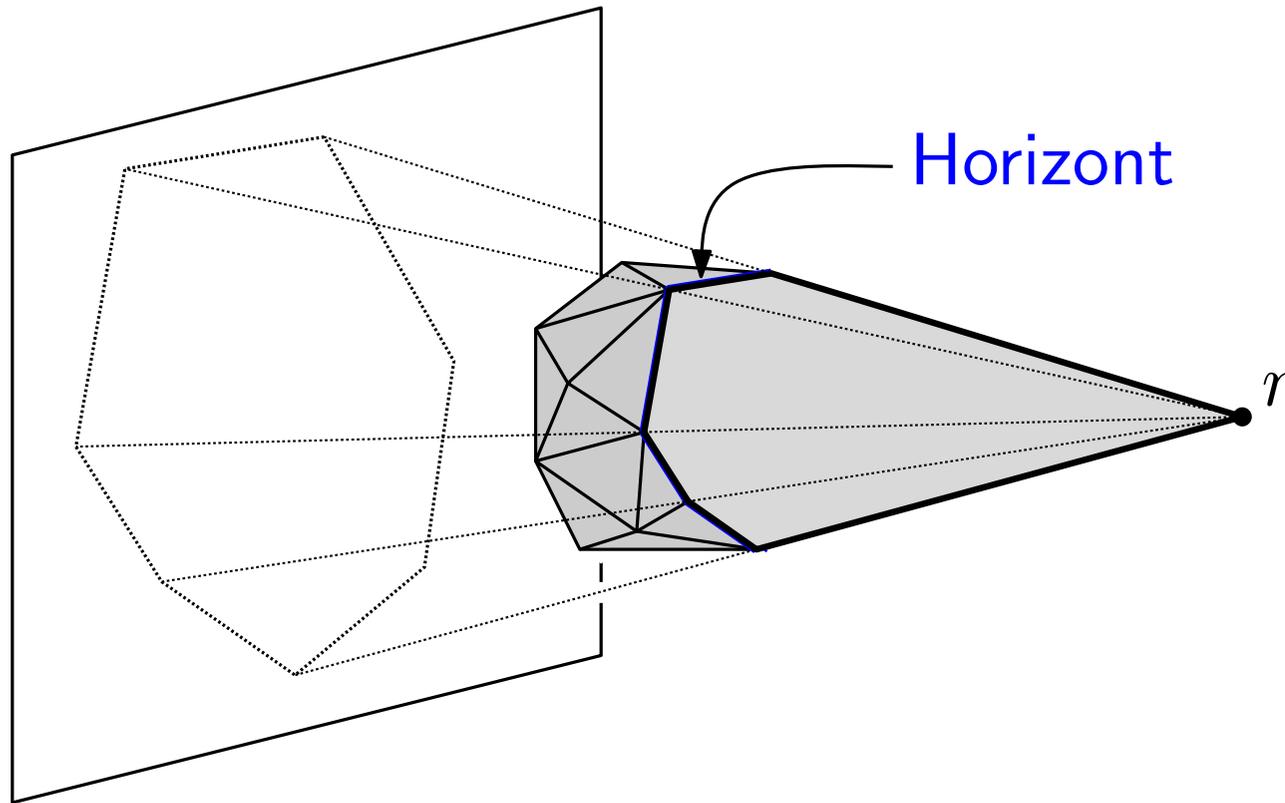
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



# Erster Schritt

## Angenommen:

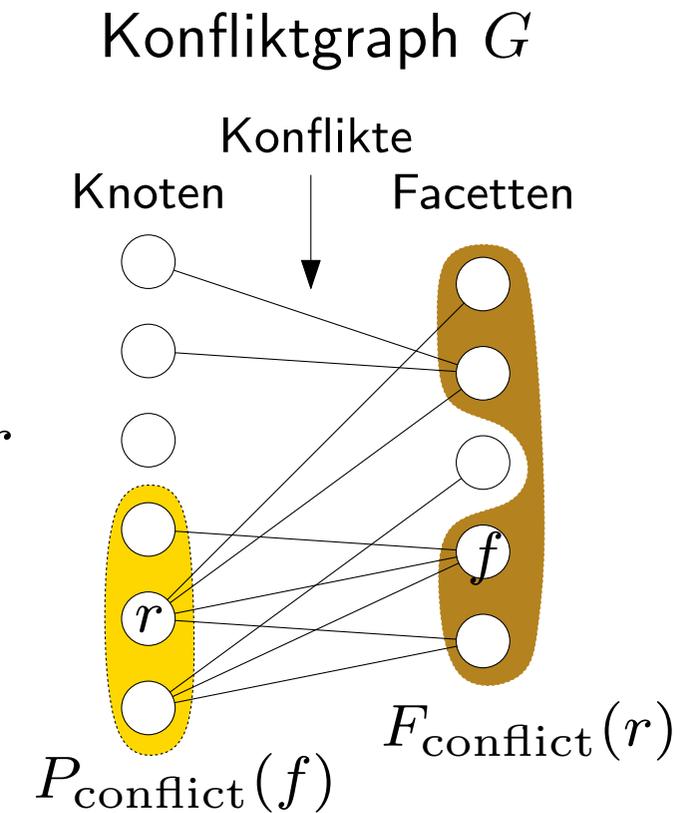
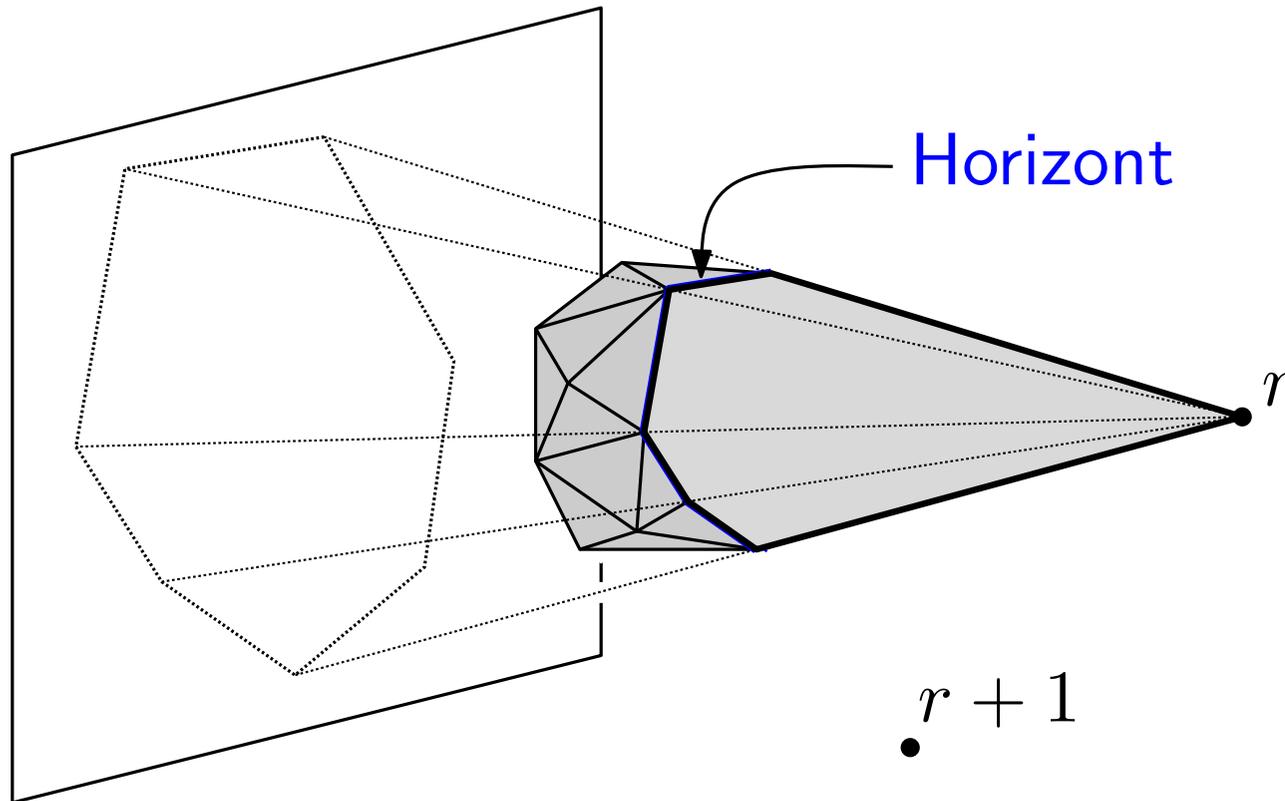
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



# Erster Schritt

## Angenommen:

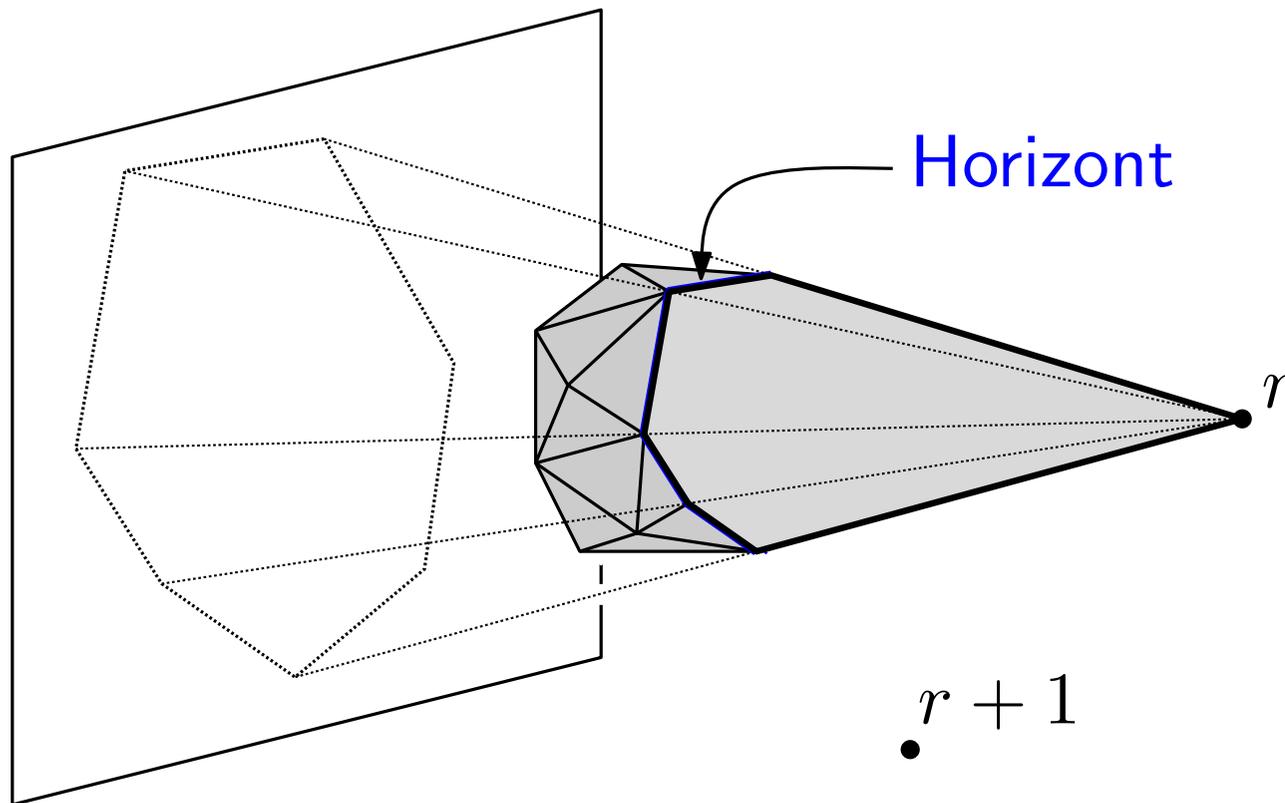
Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?



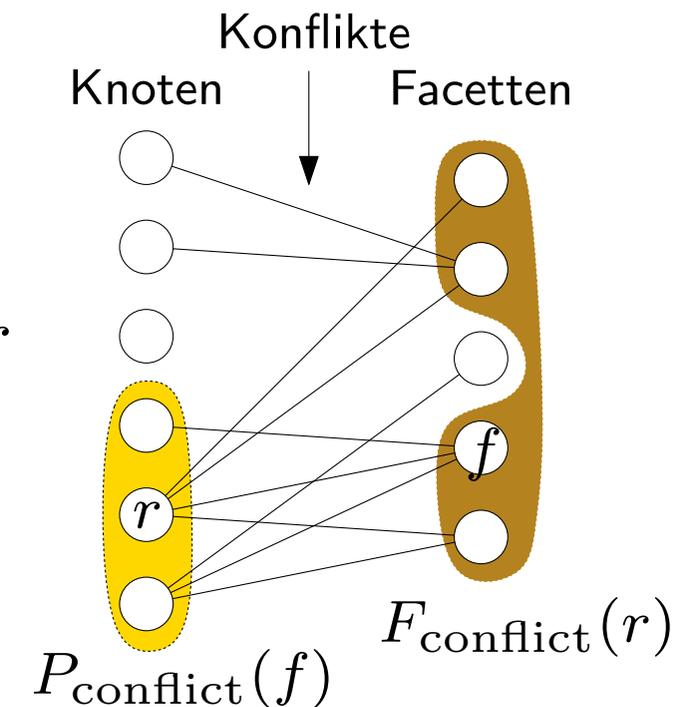
# Erster Schritt

## Angenommen:

Konvexe Hülle von  $r - 1$  Knoten gegeben. Wie erweitern zu einer konvexen Hülle für  $r$  Knoten?

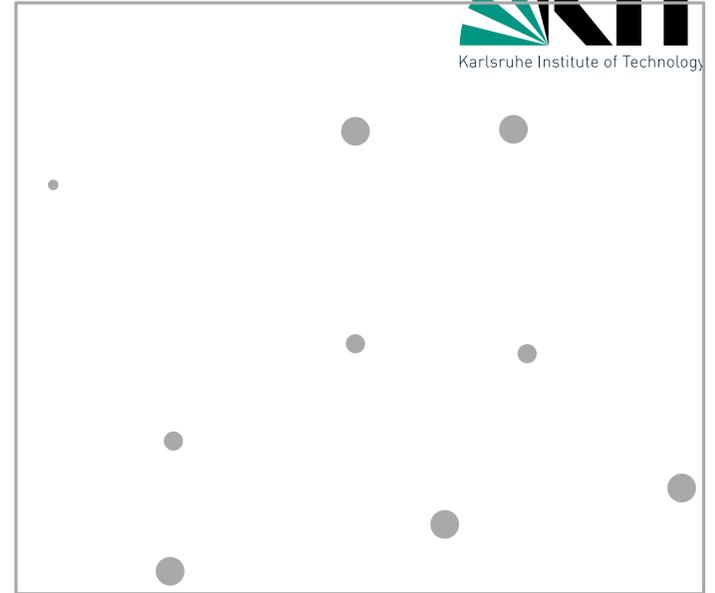


## Konfliktgraph $G$



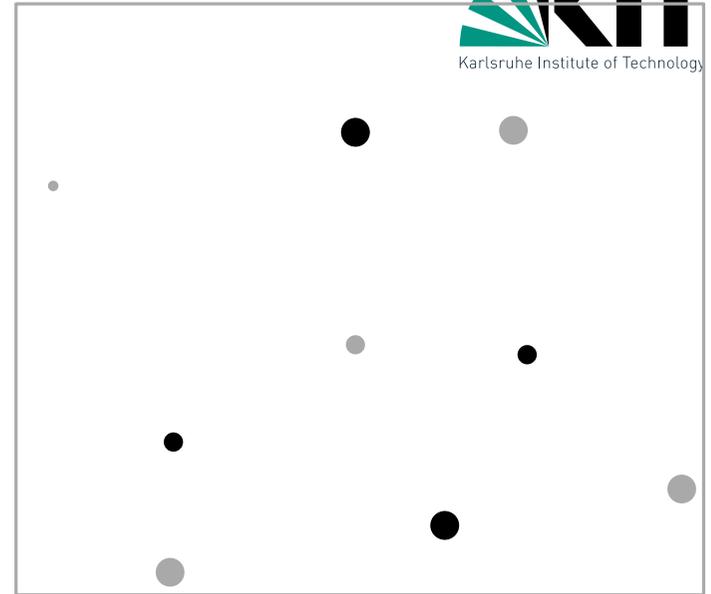
Wie bestimmt man neue Konflikte?

$3D\text{CONVEXHULL}(P \subset \mathbb{R}^3)$



$3D\text{CONVEXHULL}(P \subset \mathbb{R}^3)$

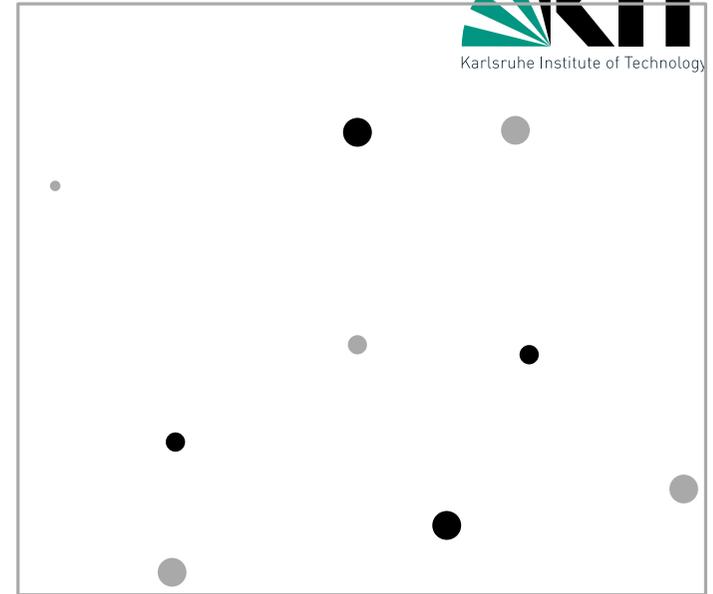
wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

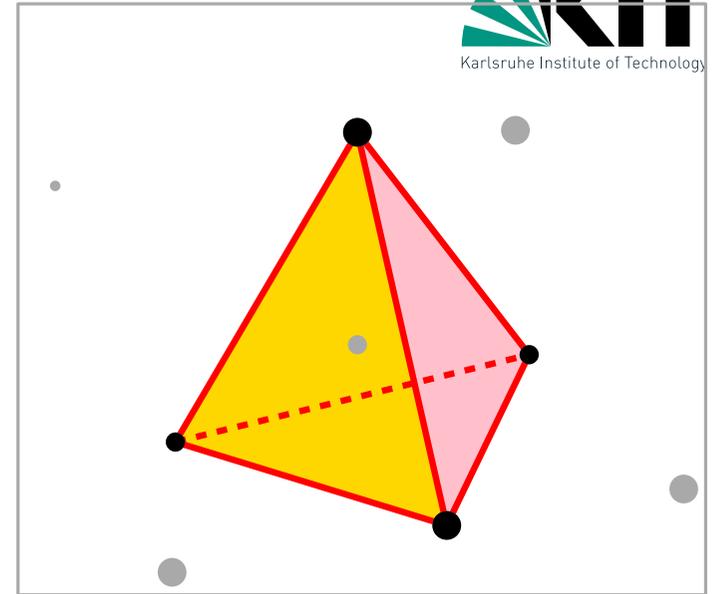
$C \leftarrow \text{CH}(P')$ ;



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

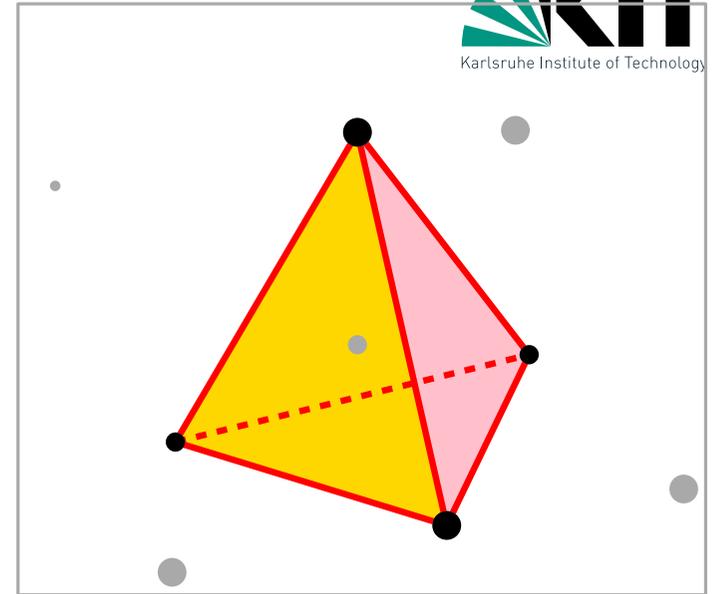
$C \leftarrow \text{CH}(P')$ ;



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

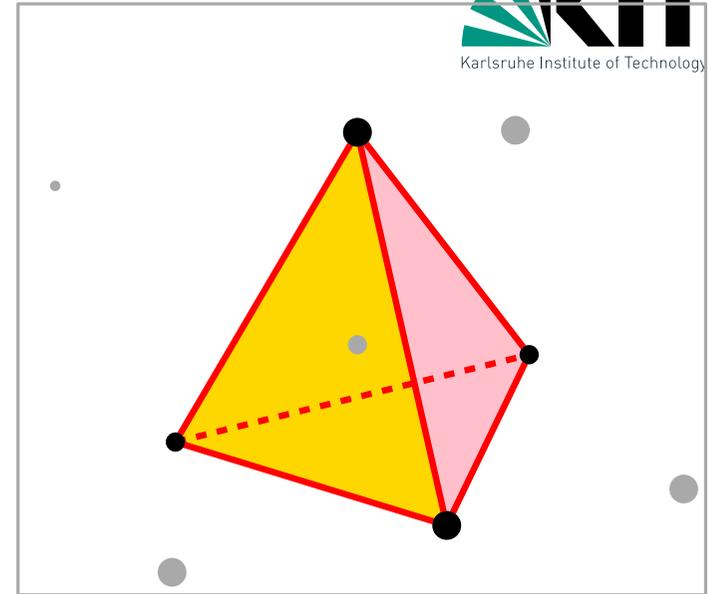


3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

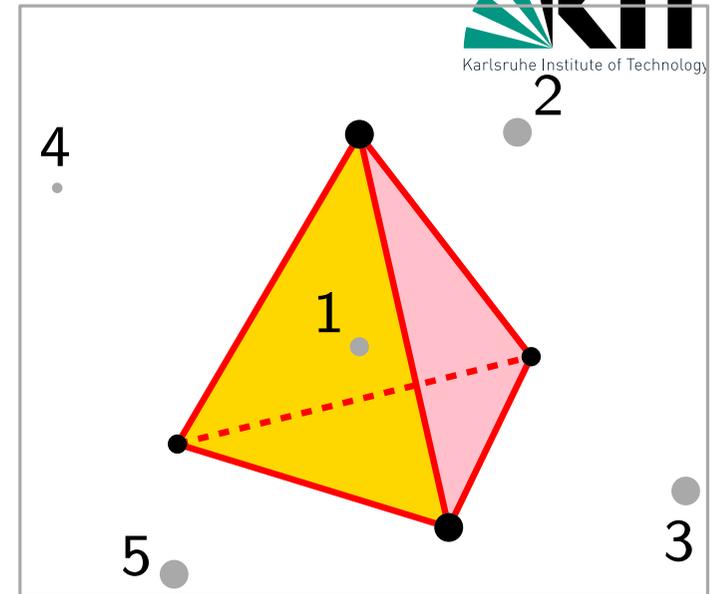


$3D\text{CONVEXHULL}(P \subset \mathbb{R}^3)$

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$



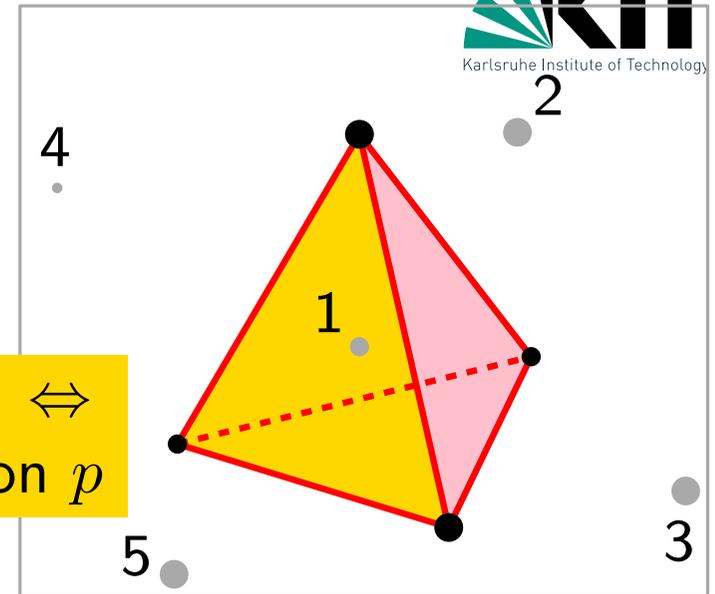
3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initializiere Konfliktgraph  $G$ :  $(p, f)$  Kante  $\Leftrightarrow$   
 $f$  sichtbar von  $p$



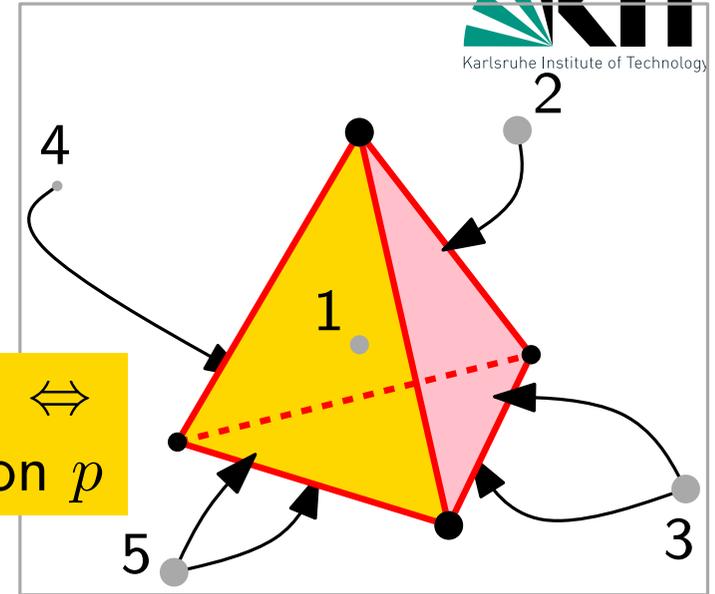
3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$ :  $(p, f)$  Kante  $\Leftrightarrow$   
 $f$  sichtbar von  $p$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

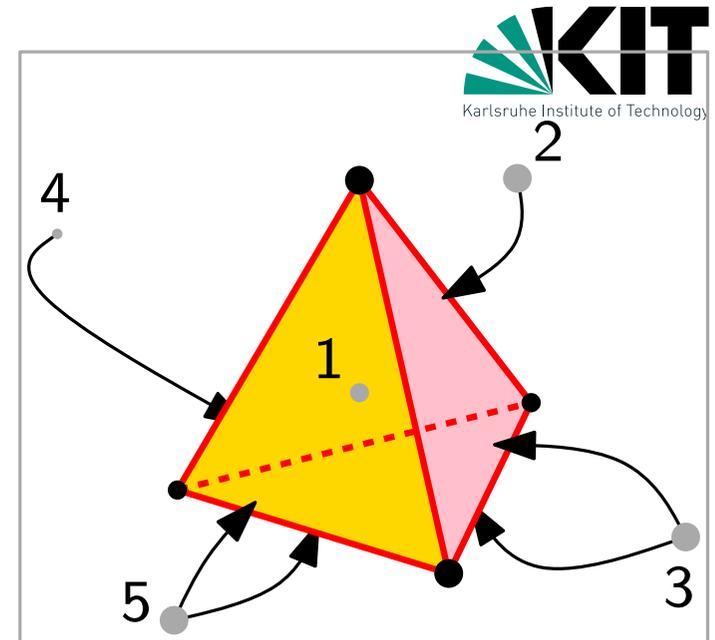
$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

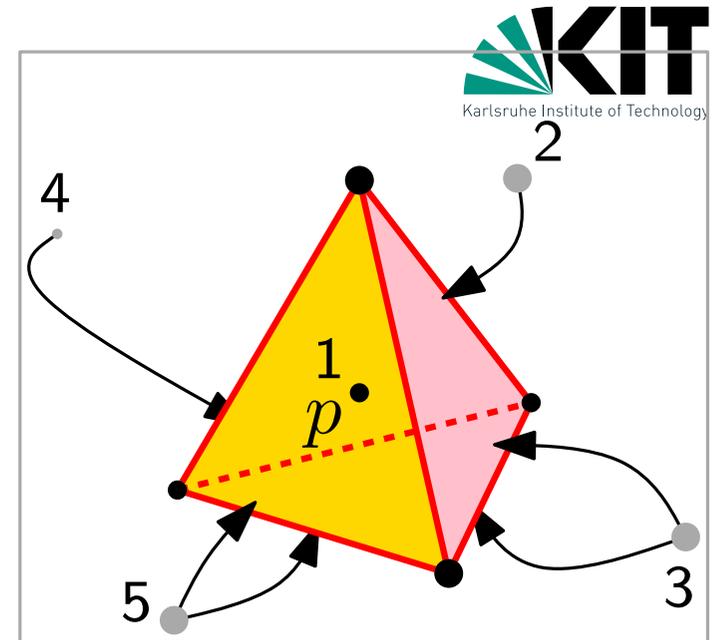
$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

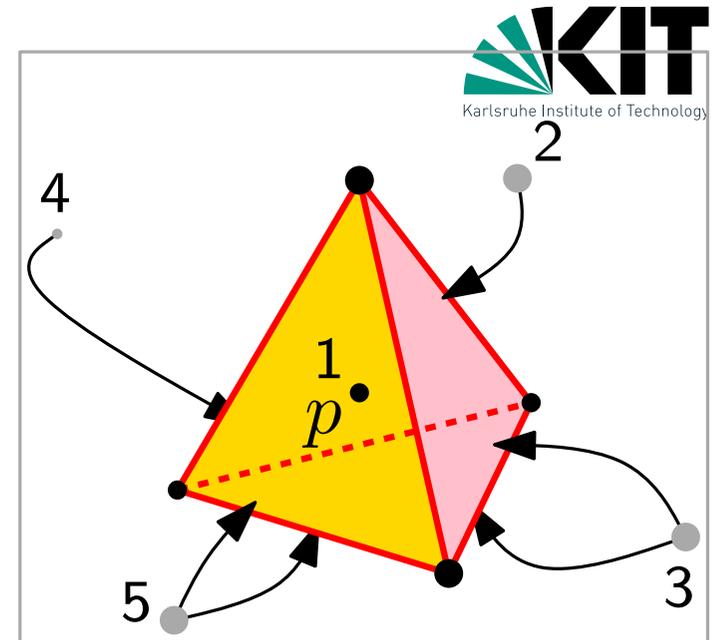
$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**       //



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

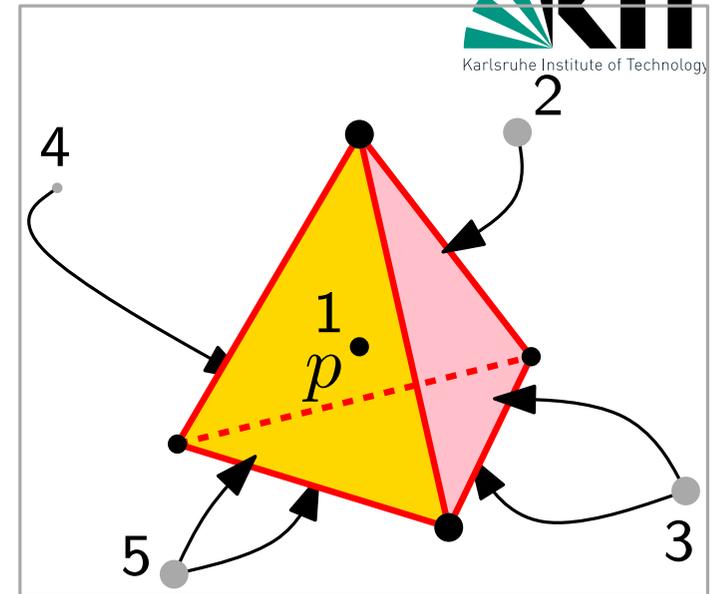
initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**

//  $p$  liegt außerhalb von  $C$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

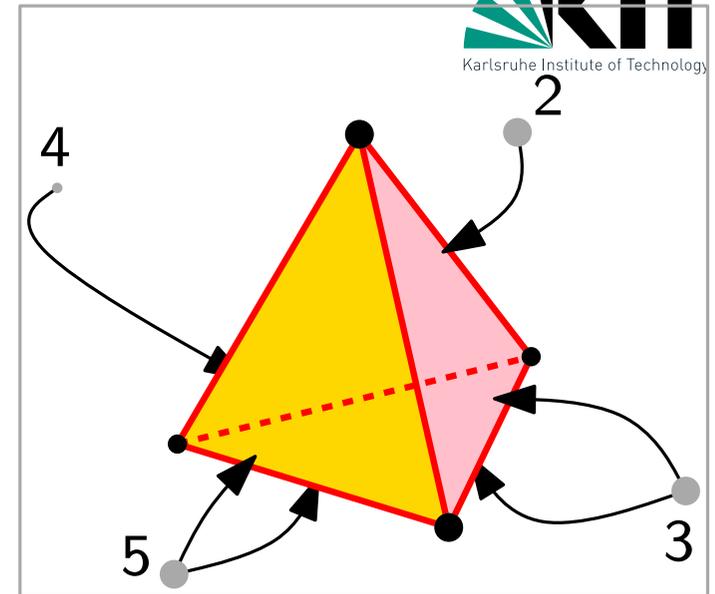
initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**

//  $p$  liegt außerhalb von  $C$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

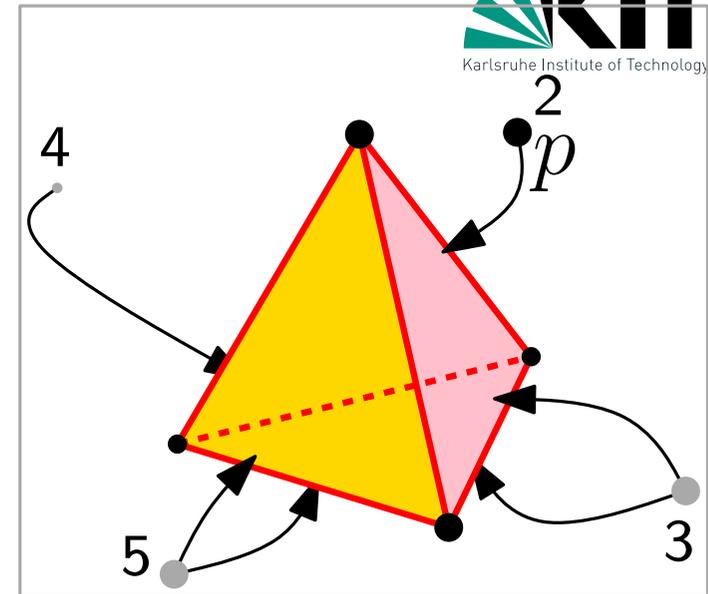
initializiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**

//  $p$  liegt außerhalb von  $C$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

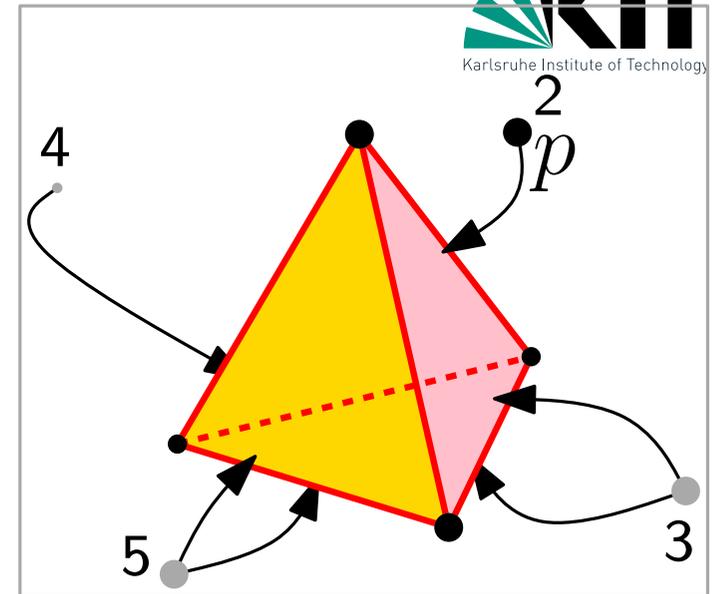
initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then** //  $p$  liegt außerhalb von  $C$

Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

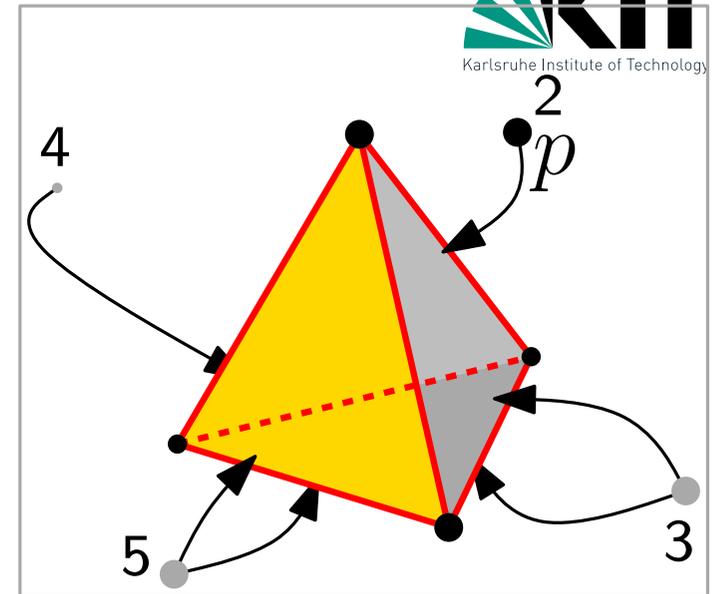
initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then** //  $p$  liegt außerhalb von  $C$

Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

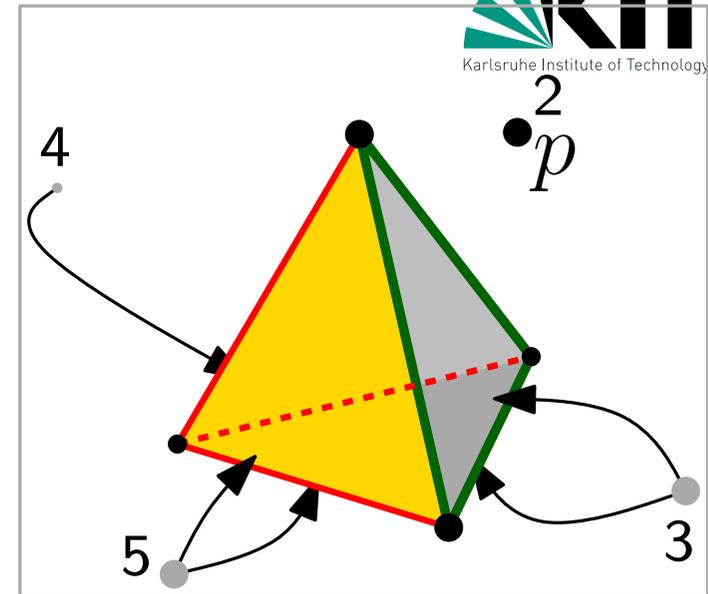
**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then** //  $p$  liegt außerhalb von  $C$

Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

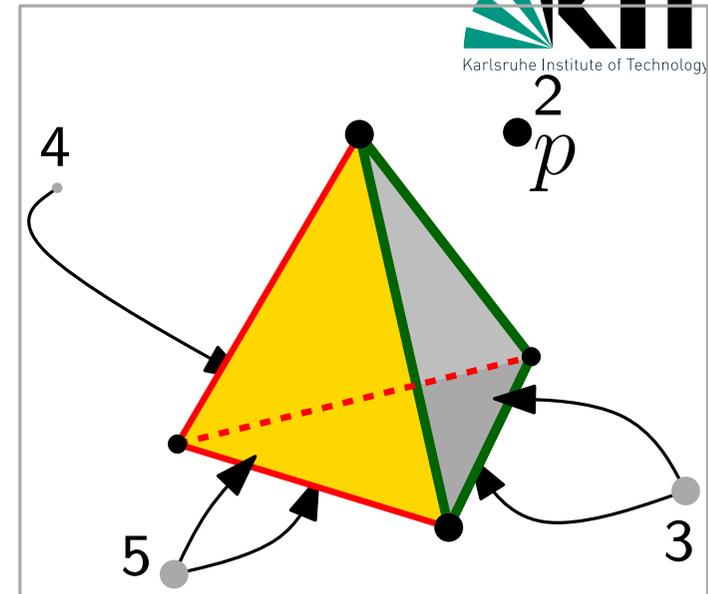
$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then** //  $p$  liegt außerhalb von  $C$

Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

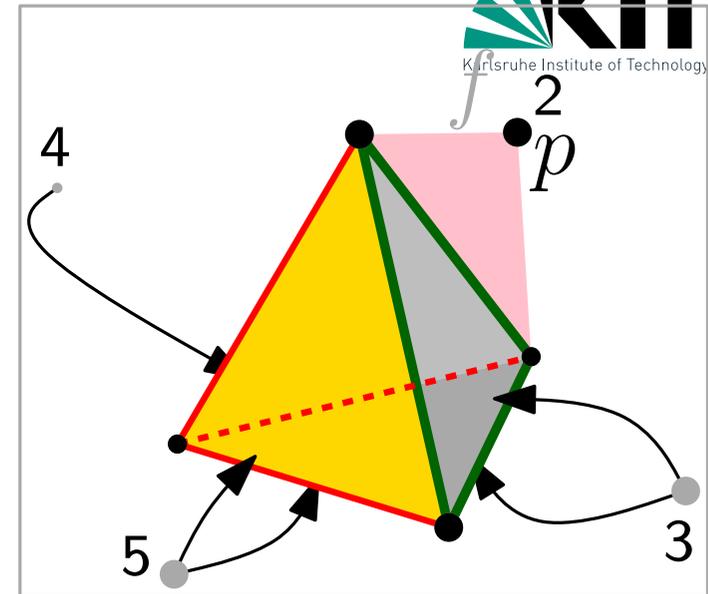
**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**           //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then** //  $p$  liegt außerhalb von  $C$

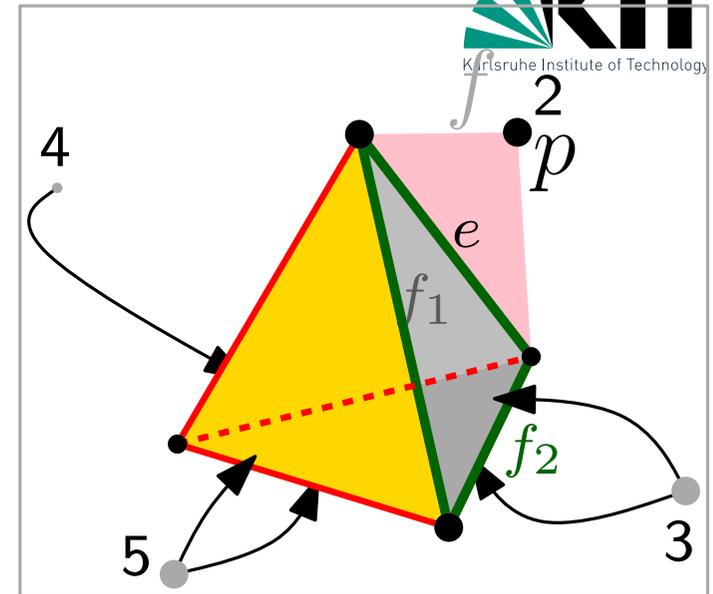
Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then** //  $p$  liegt außerhalb von  $C$

Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

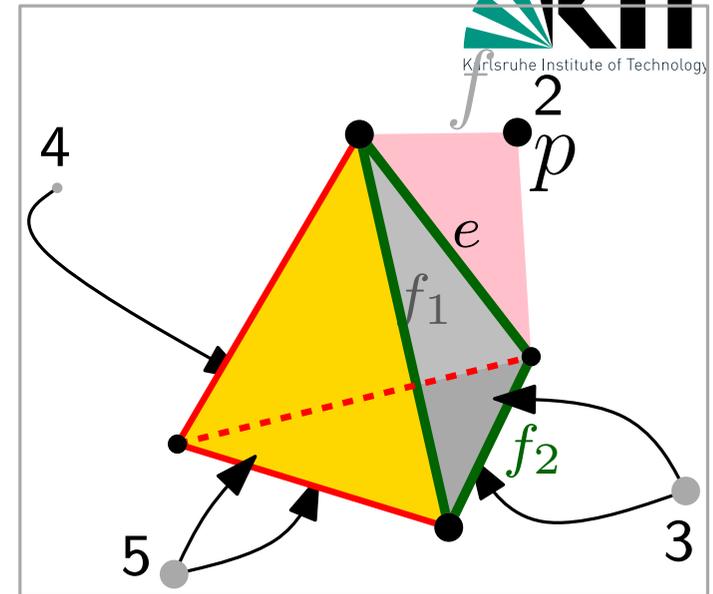
$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$



### 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then** //  $p$  liegt außerhalb von  $C$

Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

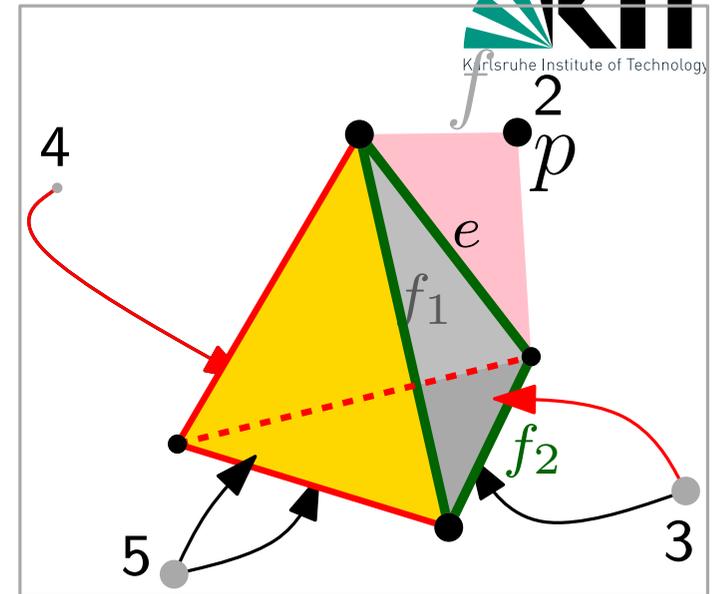
$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$



### 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**      //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

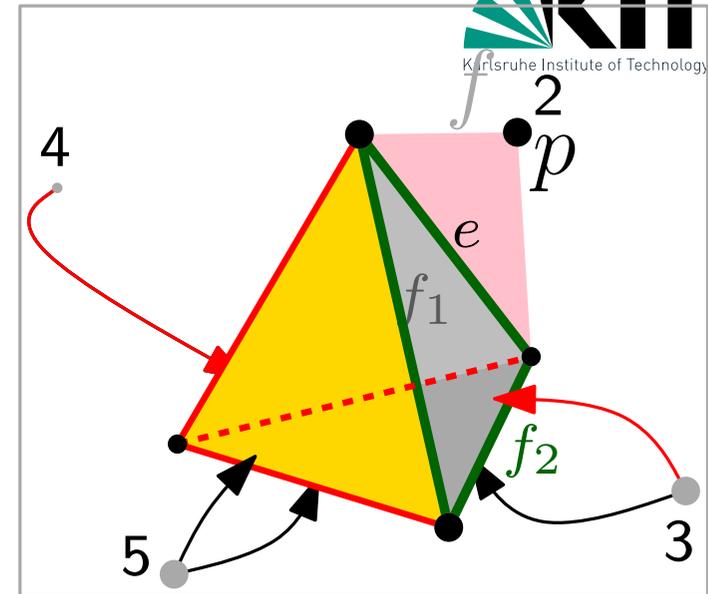
$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

                └ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$



### 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**           //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

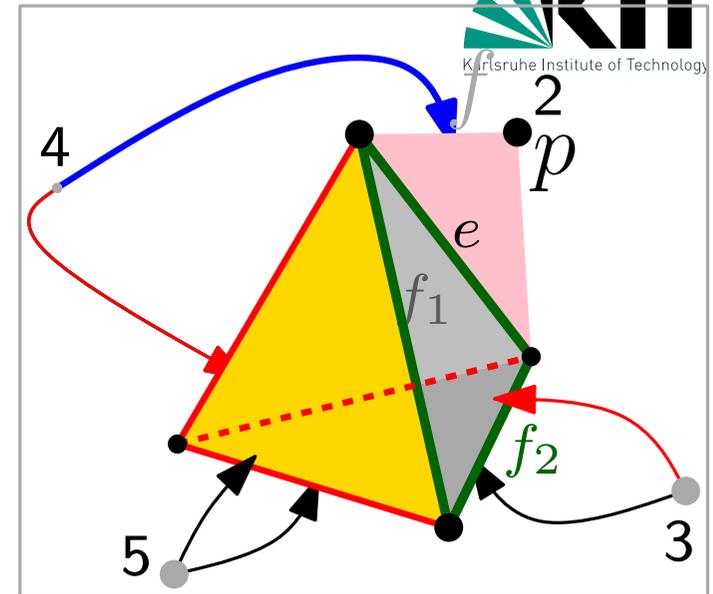
$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

                └ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

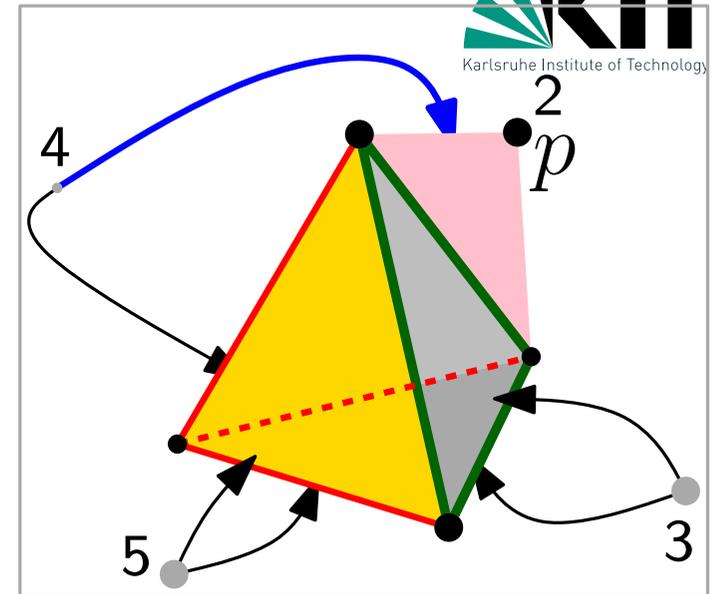
**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**      //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

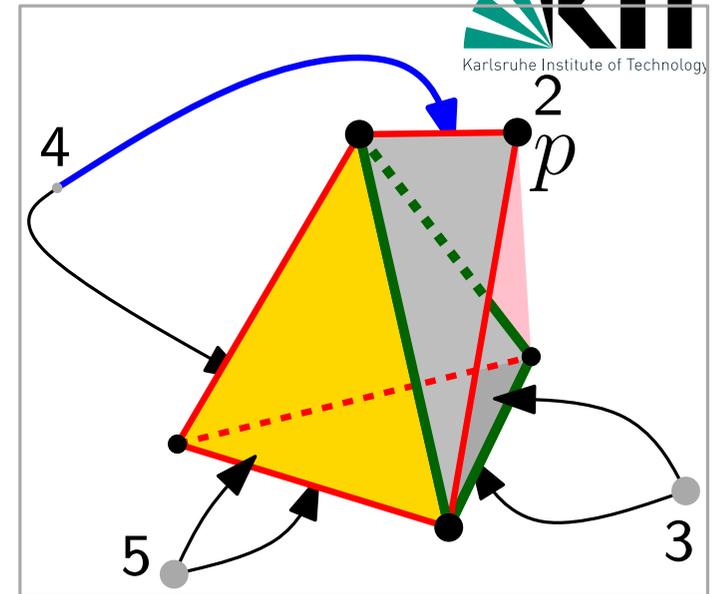
**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**           //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

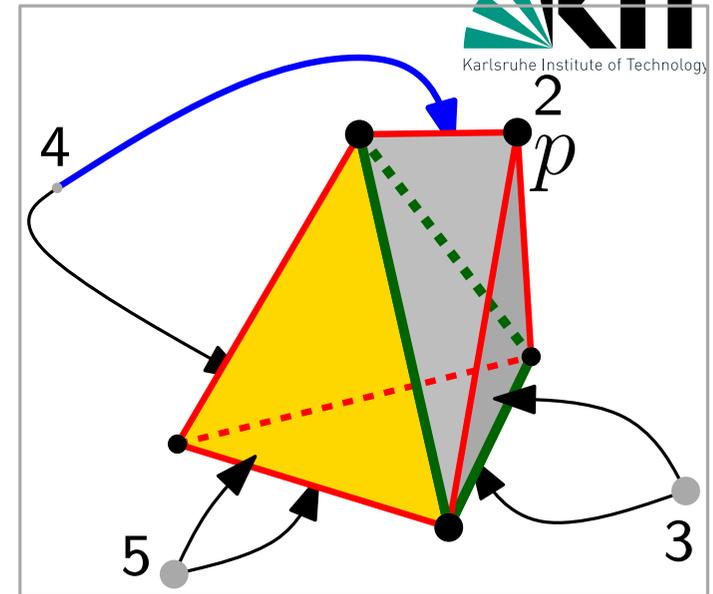
**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**           //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**           //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

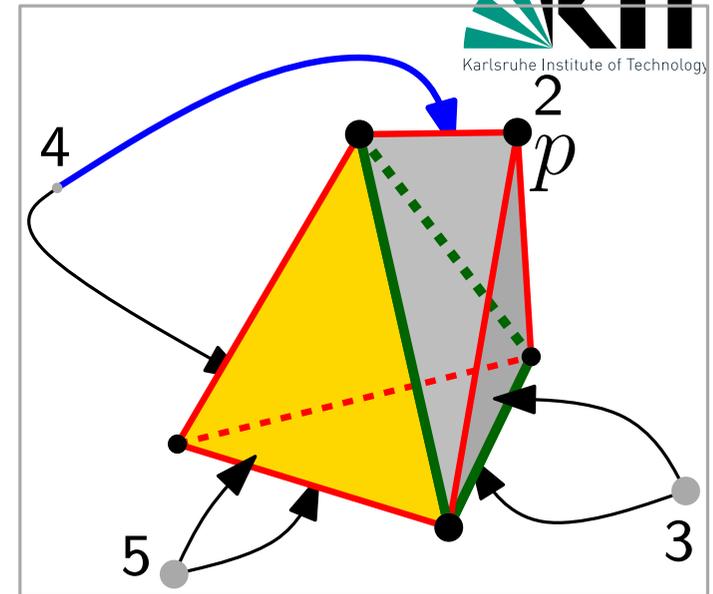
$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

                └ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$



3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**           //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

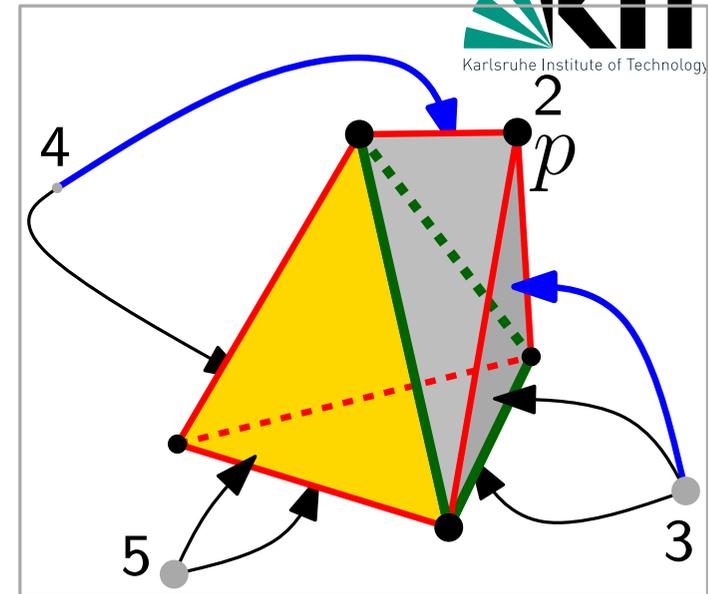
$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

                └ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$



# 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**      //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

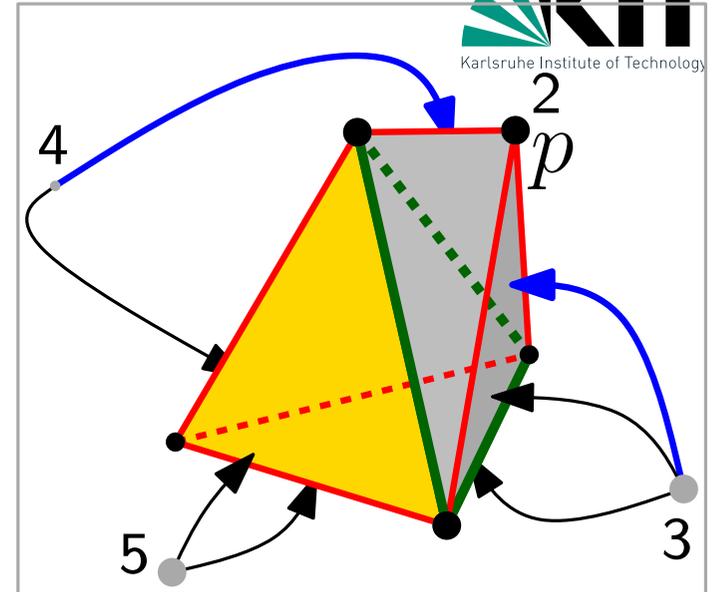
$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

                └ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$



# 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**      //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

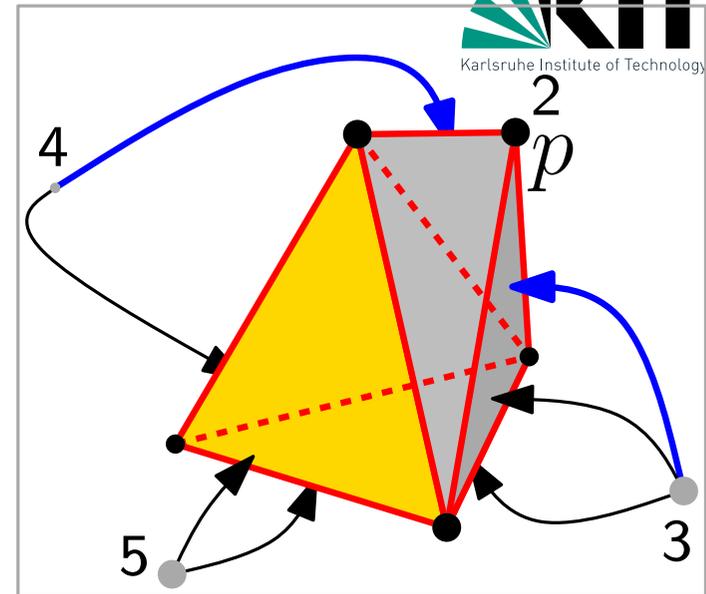
$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

                └ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$



# 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**           //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

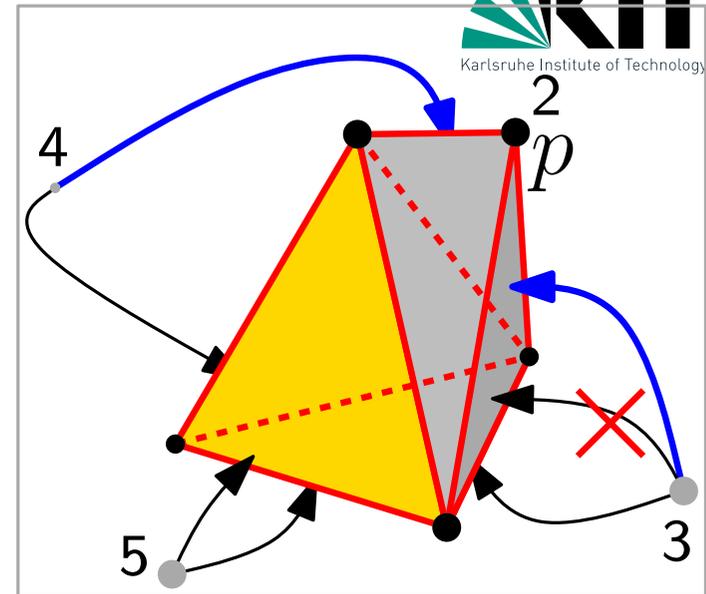
$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

                └ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$



# 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**           //  $p$  liegt außerhalb von  $C$

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

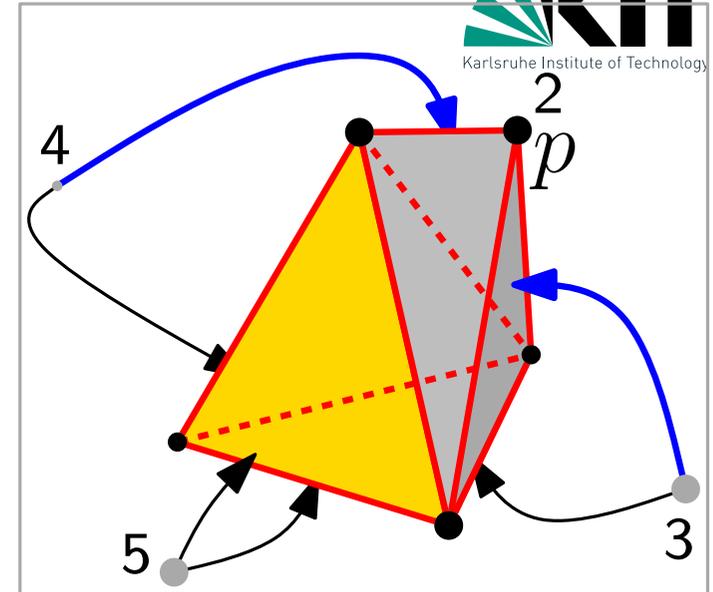
$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

                └ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$



### 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then** //  $p$  liegt außerhalb von  $C$

Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

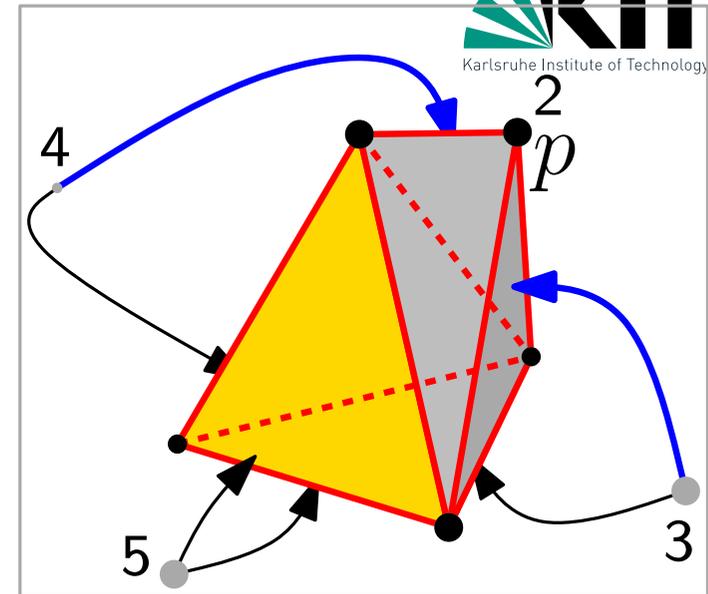
$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

└ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$



Begrenze #Facetten, die der Algorithmus erzeugt.

## Rückwärtsanalyse

**Idee:**

Fixiere  $P_i := \{p_1, \dots, p_i\}$ .

- Entferne nun ein  $p_j \in P_i$ .
- Wie viele Facetten werden entfernt?

Begrenze #Facetten, die der Algorithmus erzeugt.

## Rückwärtsanalyse

**Idee:**

Fixiere  $P_i := \{p_1, \dots, p_i\}$ .

- Entferne nun ein  $p_j \in P_i$ .
- Wie viele Facetten werden entfernt?

Begrenze #Facetten, die der Algorithmus erzeugt.

**Lemma:** Die erwartete Anzahl an erzeugten Facetten ist höchstens  $6n - 20$ .

## Rückwärtsanalyse

**Idee:** Fixiere  $P_i := \{p_1, \dots, p_i\}$ .

- Entferne nun ein  $p_j \in P_i$ .
- Wie viele Facetten werden entfernt?

Begrenze #Facetten, die der Algorithmus erzeugt.

**Lemma:** Die erwartete Anzahl an erzeugten Facetten ist höchstens  $6n - 20$ .

## Rückwärtsanalyse

**Idee:** Fixiere  $P_i := \{p_1, \dots, p_i\}$ .

- Entferne nun ein  $p_j \in P_i$ .
- Wie viele Facetten werden entfernt?

---

**Erinnerung:**

**Thm:**

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist #Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und #Facetten höchstens  $2n - 4$ .

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

---

## Erinnerung:

### Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist #Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und #Facetten höchstens  $2n - 4$ .

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

In Schritt  $r$ : ■ Nach Thm insgesamt  $\leq 3r - 6$  Kanten

---

## Erinnerung:

### Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist #Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und #Facetten höchstens  $2n - 4$ .

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

In Schritt  $r$ : ■ Nach Thm insgesamt  $\leq 3r - 6$  Kanten

■ 
$$\sum_i^r \deg(p_i) \leq 6r - 12$$

---

## Erinnerung:

### Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist #Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und #Facetten höchstens  $2n - 4$ .

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

In Schritt  $r$ : ■ Nach Thm insgesamt  $\leq 3r - 6$  Kanten

■ 
$$\sum_i^r \deg(p_i) \leq 6r - 12$$

■ durchschnittl.  $\deg(p_i) \leq 6 - 12/r$

---

## Erinnerung:

### Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist #Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und #Facetten höchstens  $2n - 4$ .

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

In Schritt  $r$ : ■ Nach Thm insgesamt  $\leq 3r - 6$  Kanten

■ 
$$\sum_i^r \deg(p_i) \leq 6r - 12$$

■ durchschnittl.  $\deg(p_i) \leq 6 - 12/r$

■ Erwartungswert  $\deg(p_i) \leq 6 - 12/r$  ?

---

## Erinnerung:

### Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist #Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und #Facetten höchstens  $2n - 4$ .

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

In Schritt  $r$ : ■ Nach Thm insgesamt  $\leq 3r - 6$  Kanten

■ 
$$\sum_i^r \deg(p_i) \leq 6r - 12$$

■ durchschnittl.  $\deg(p_i) \leq 6 - 12/r$

■ Erwartungswert  $\deg(p_i) \leq 6 - 12/r$  ?

Die ersten 4 Punkte werden *nicht* zufällig gewählt.

---

## Erinnerung:

### Thm:

Sei  $\mathcal{P}$  ein konvexes Polytop mit  $n$  Knoten. Dann ist #Kanten in  $\mathcal{P}$  höchstens  $3n - 6$  und #Facetten höchstens  $2n - 4$ .

# Analyse

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

# Analyse

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$E[\deg(p_r, CH(P_r))]$$

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$E[\deg(p_r, CH(P_r))] = \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i))$$

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$E[\deg(p_r, CH(P_r))] = \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i))$$

Die ersten 4 Knoten haben mind. Grad 12

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$\begin{aligned} E[\deg(p_r, CH(P_r))] &= \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i)) \\ &\leq \frac{1}{r-4} \left( \sum_{i=1}^r [\deg(p_i, CH(P_i))] - 12 \right) \end{aligned}$$

Die ersten 4 Knoten haben mind. Grad 12

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$\begin{aligned} E[\deg(p_r, CH(P_r))] &= \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i)) \\ &\leq \frac{1}{r-4} \left( \sum_{i=1}^r [\deg(p_i, CH(P_i))] - 12 \right) \end{aligned}$$

Die ersten 4 Knoten haben mind. Grad 12

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$\begin{aligned} E[\deg(p_r, CH(P_r))] &= \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i)) \\ &\leq \frac{1}{r-4} \left( \sum_{i=1}^r [\deg(p_i, CH(P_i))] - 12 \right) \\ &= \frac{6r-12-12}{r-4} = 6 \end{aligned}$$

Die ersten 4 Knoten haben mind. Grad 12

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$\begin{aligned} E[\deg(p_r, CH(P_r))] &= \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i)) \\ &\leq \frac{1}{r-4} \left( \sum_{i=1}^r [\deg(p_i, CH(P_i))] - 12 \right) \\ &= \frac{6r-12-12}{r-4} = 6 \end{aligned}$$

Die ersten 4 Knoten haben mind. Grad 12

**Lemma:** Die erwartete Anzahl an erzeugten Facetten ist höchstens  $6n - 20$ .

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$\begin{aligned} E[\deg(p_r, CH(P_r))] &= \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i)) \\ &\leq \frac{1}{r-4} \left( \sum_{i=1}^r [\deg(p_i, CH(P_i))] - 12 \right) \\ &= \frac{6r-12-12}{r-4} = 6 \end{aligned}$$

Die ersten 4 Knoten haben mind. Grad 12

**Lemma:** Die erwartete Anzahl an erzeugten Facetten ist höchstens  $6n - 20$ .

$$4 + \sum_{r=5}^n E[\deg(p_r, CH(P_r))]$$

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$\begin{aligned} E[\deg(p_r, CH(P_r))] &= \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i)) \\ &\leq \frac{1}{r-4} \left( \sum_{i=1}^r [\deg(p_i, CH(P_i))] - 12 \right) \\ &= \frac{6r-12-12}{r-4} = 6 \end{aligned}$$

Die ersten 4 Knoten haben mind. Grad 12

**Lemma:** Die erwartete Anzahl an erzeugten Facetten ist höchstens  $6n - 20$ .

$$4 + \sum_{r=5}^n E[\deg(p_r, CH(P_r))] \leq 4 + (n - 4) \cdot 6$$

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$\begin{aligned} E[\deg(p_r, CH(P_r))] &= \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i)) \\ &\leq \frac{1}{r-4} \left( \sum_{i=1}^r [\deg(p_i, CH(P_i))] - 12 \right) \\ &= \frac{6r-12-12}{r-4} = 6 \end{aligned}$$

Die ersten 4 Knoten haben mind. Grad 12

**Lemma:** Die erwartete Anzahl an erzeugten Facetten ist höchstens  $6n - 20$ .

$$4 + \sum_{r=5}^n E[\deg(p_r, CH(P_r))] \leq 4 + (n - 4) \cdot 6 = 6n - 20$$

Begrenze #Facetten durch den Grad  $\deg(p_r)$  eines Knotens  $p_r$ .

$$\begin{aligned} E[\deg(p_r, CH(P_r))] &= \frac{1}{r-4} \sum_{i=5}^r \deg(p_i, CH(P_i)) \\ &\leq \frac{1}{r-4} \left( \sum_{i=1}^r [\deg(p_i, CH(P_i))] - 12 \right) \\ &= \frac{6r-12-12}{r-4} = 6 \end{aligned}$$

Die ersten 4 Knoten haben mind. Grad 12

**Lemma:** Die erwartete Anzahl an erzeugten Facetten ist höchstens  $6n - 20$ .

$$4 + \sum_{r=5}^n E[\deg(p_r, CH(P_r))] \leq 4 + (n - 4) \cdot 6 = 6n - 20$$



**Lemma:** Die erwartete Anzahl an erzeugten Facetten ist höchstens  $6n - 20$ .

**Thm:** Die konvexe Hülle für eine Menge von  $n$  Punkten im  $\mathbb{R}^3$  kann in erwarteter  $O(n \log n)$  Zeit berechnet werden.

**Lemma:** Die erwartete Anzahl an erzeugten Facetten ist höchstens  $6n - 20$ .

# 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**

Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

└ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$

lösche Knoten  $\{p\} \cup F_{\text{conflict}}(p)$  von  $G$

# 3DCONVEXHULL( $P \subset \mathbb{R}^3$ )

wähle  $P' \subseteq P$  (4 nicht koplan. Punkte)

$C \leftarrow \text{CH}(P')$ ;  $P \leftarrow P \setminus P'$

$P = \text{RANDOMPERMUTATION}(P)$

initialisiere Konfliktgraph  $G$

**while**  $P \neq \emptyset$  **do**

$p \leftarrow P.\text{remove\_first}()$

**if**  $F_{\text{conflict}}(p) \neq \emptyset$  **then**

        Lösche alle Facetten  $F_{\text{conflict}}(p)$  aus  $C$

$\mathcal{L} \leftarrow$  Horizont-Kanten die von  $p$  aus sichtbar sind

**foreach**  $e \in \mathcal{L}$  **do**

$f \leftarrow$  erzg. Facette $_C(e, p)$ ; erzg. Knoten für  $f$  in  $G$

$(f_1, f_2) \leftarrow$  vorher\_inzident $_C(e)$

$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$

**foreach**  $p \in P(e)$  **do**

                └ **if**  $f$  sichtbar von  $p$  **then** add\_edge( $p, f$ ) zu  $G$

        lösche Knoten  $\{p\} \cup F_{\text{conflict}}(p)$  von  $G$

**Thm:** Die konvexe Hülle für eine Menge von  $n$  Punkten im  $\mathbb{R}^3$  kann in erwarteter  $O(n \log n)$  Zeit berechnet werden.

$$O(n)$$

$$p_r: O(|F_c(p_r)|)$$

$$\text{Gesamt: } E\left[\sum_{r=5}^n |F_c(p_r)|\right] = O(n)$$

$$O(n)$$

Kapitel 9, 11 in [BCK08]

$$O(n \log n)$$

$$O(n)$$

# Diskussion

## Optimale Laufzeit?

## Optimale Laufzeit?

Divide-and-Conquer Algorithmus mit Laufzeit  $\mathcal{O}(n \log n)$

## Optimale Laufzeit?

Divide-and-Conquer Algorithmus mit Laufzeit  $\mathcal{O}(n \log n)$

Was ist mit Dimensionen  $> 3$  ?

## Optimale Laufzeit?

Divide-and-Conquer Algorithmus mit Laufzeit  $\mathcal{O}(n \log n)$

## Was ist mit Dimensionen $> 3$ ?

Durch das *Upper Bound Theorem* ist bekannt, dass die Komplexität der konvexen Hülle einer Punktmenge bestehend aus  $n$  Punkten im  $d$ -dimensionalen Raum in  $\Theta(n^{\lfloor d/2 \rfloor})$  ist. Der hier vorgestellte Algorithmus kann für höhere Dimensionen angepasst werden.

Der beste Ausgabe-Sensitive Algorithmus für die Berechnung konvexer Hüllen im  $\mathbb{R}^d$  hat eine Laufzeit von:

## Optimale Laufzeit?

Divide-and-Conquer Algorithmus mit Laufzeit  $\mathcal{O}(n \log n)$

## Was ist mit Dimensionen $> 3$ ?

Durch das *Upper Bound Theorem* ist bekannt, dass die Komplexität der konvexen Hülle einer Punktmenge bestehend aus  $n$  Punkten im  $d$ -dimensionalen Raum in  $\Theta(n^{\lfloor d/2 \rfloor})$  ist. Der hier vorgestellte Algorithmus kann für höhere Dimensionen angepasst werden.

Der beste Ausgabe-Sensitive Algorithmus für die Berechnung konvexer Hüllen im  $\mathbb{R}^d$  hat eine Laufzeit von:

$$\mathcal{O}(n \log k + (nk)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{\mathcal{O}(1)} n)$$

## Optimale Laufzeit?

Divide-and-Conquer Algorithmus mit Laufzeit  $\mathcal{O}(n \log n)$

## Was ist mit Dimensionen $> 3$ ?

Durch das *Upper Bound Theorem* ist bekannt, dass die Komplexität der konvexen Hülle einer Punktmenge bestehend aus  $n$  Punkten im  $d$ -dimensionalen Raum in  $\Theta(n^{\lfloor d/2 \rfloor})$  ist. Der hier vorgestellte Algorithmus kann für höhere Dimensionen angepasst werden.

Der beste Ausgabe-Sensitive Algorithmus für die Berechnung konvexer Hüllen im  $\mathbb{R}^d$  hat eine Laufzeit von:

$$\mathcal{O}(n \log k + (nk)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{\mathcal{O}(1)} n)$$

# Danke!