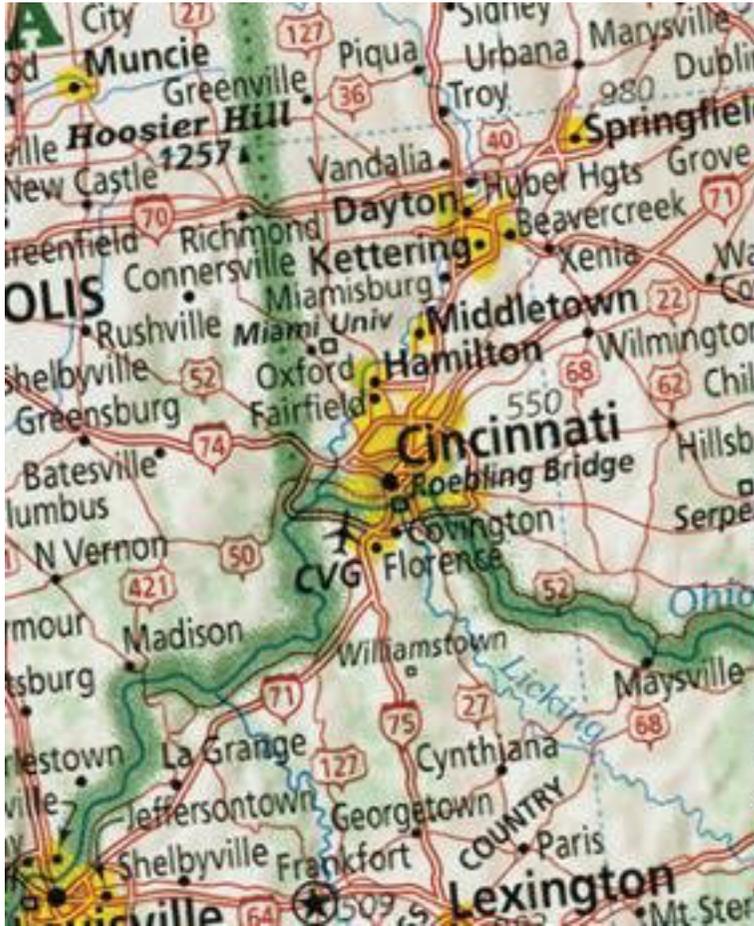


Algorithmen für Geographische Informationssysteme

9. Vorlesung Boundary Labeling

Alexander Wolff

Kartenbeschriftungen



Von Franz Schmelhaus. CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=100211969>

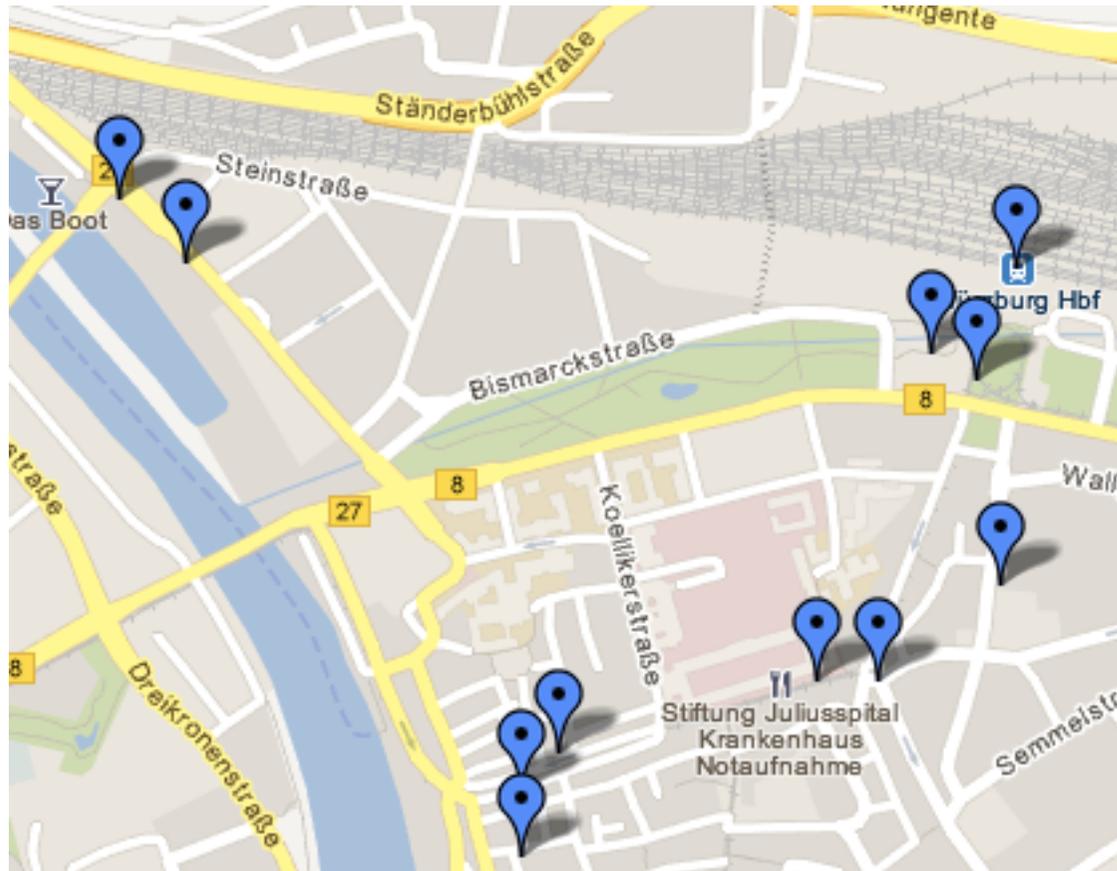


Eduard Imhof (Schiers 1895–1996 Erlenbach)

“Poor, sloppy, amateurish type placement is irresponsible; it spoils even the best image and impedes reading.”

[Eduard Imhof '75]

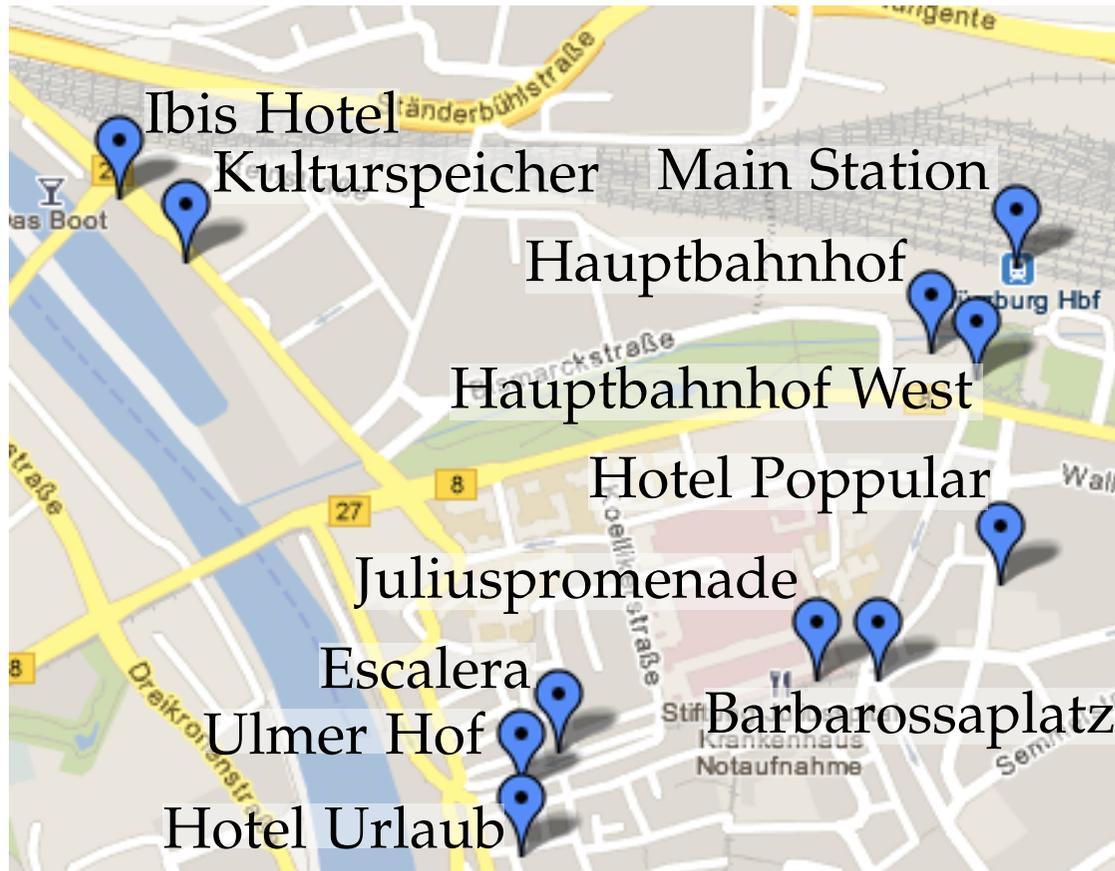
Randbeschriftungen



Randbeschriftungen



Randbeschriftungen



Nachteile klassischer Kartenbeschriftungen

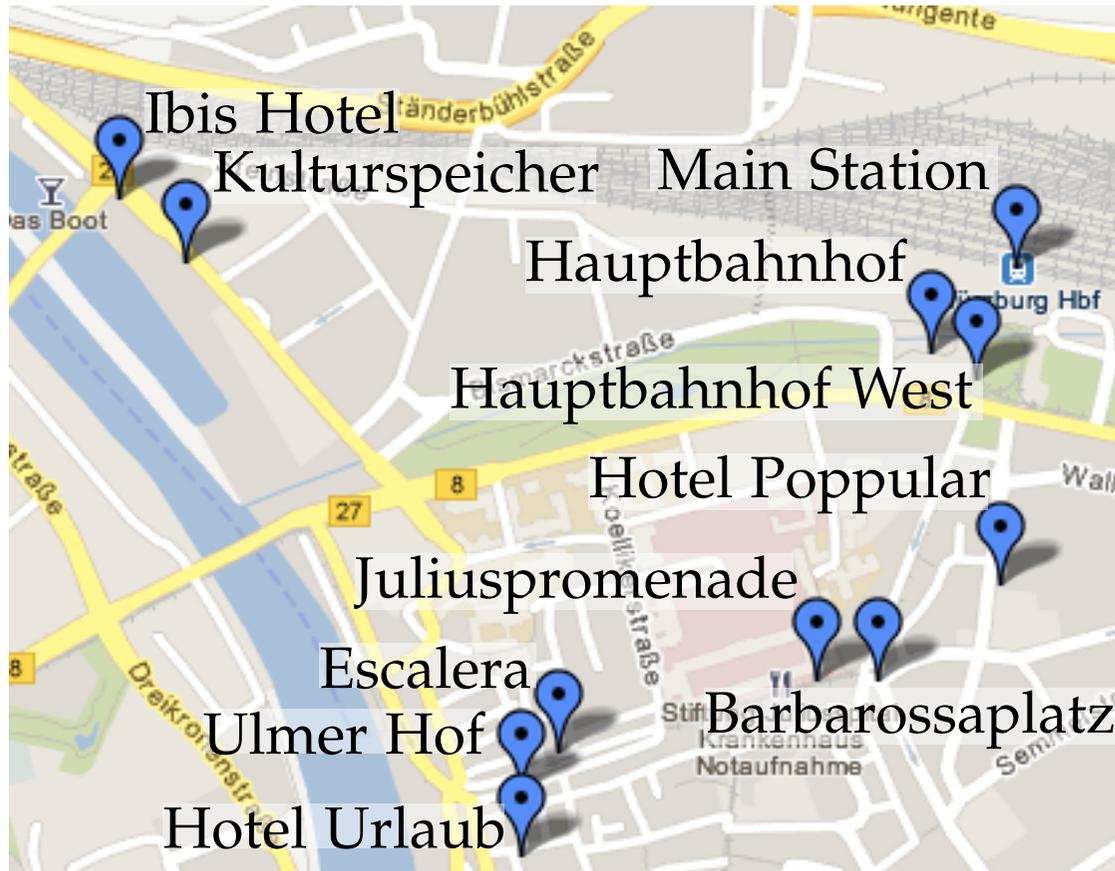
Randbeschriftungen



Nachteile klassischer Kartenbeschriftungen

- funktioniert nur gut für dünne Punktmengen

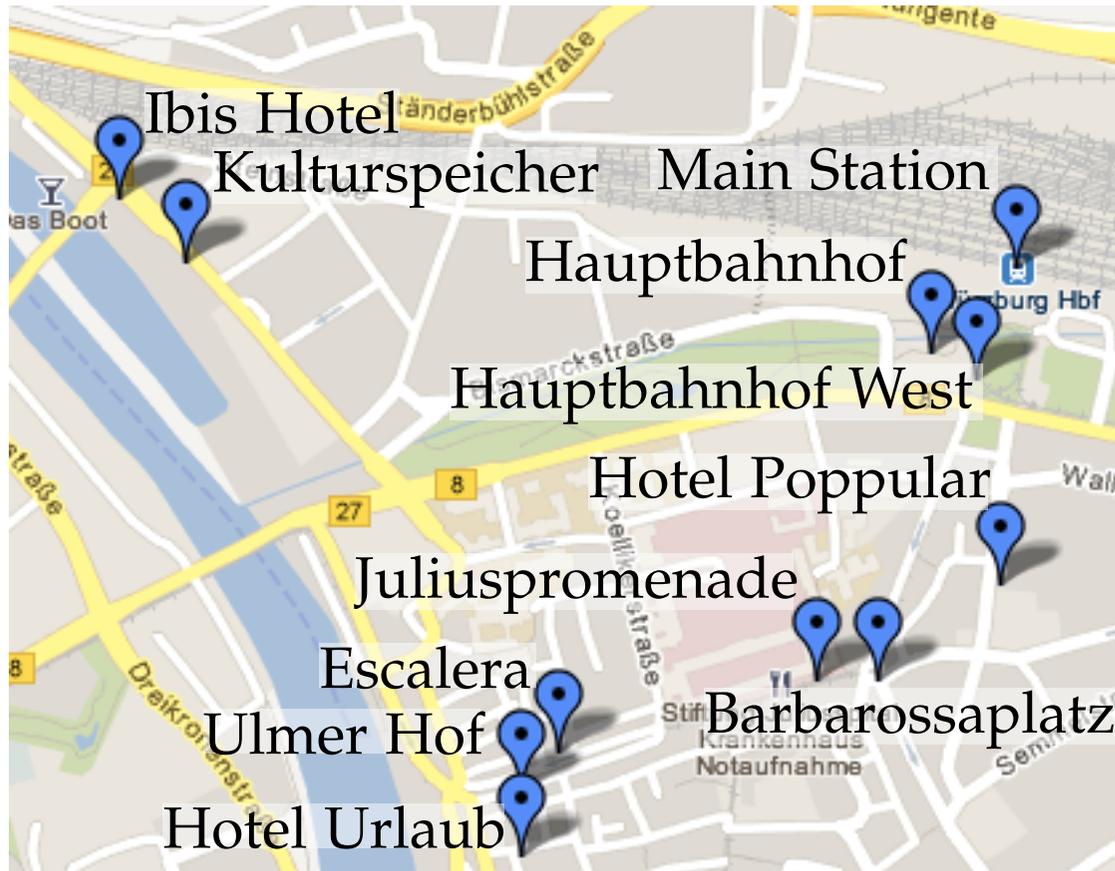
Randbeschriftungen



Nachteile klassischer Kartenbeschriftungen

- funktioniert nur gut für dünne Punktmengen
- Beschriftung überdecken die Karte

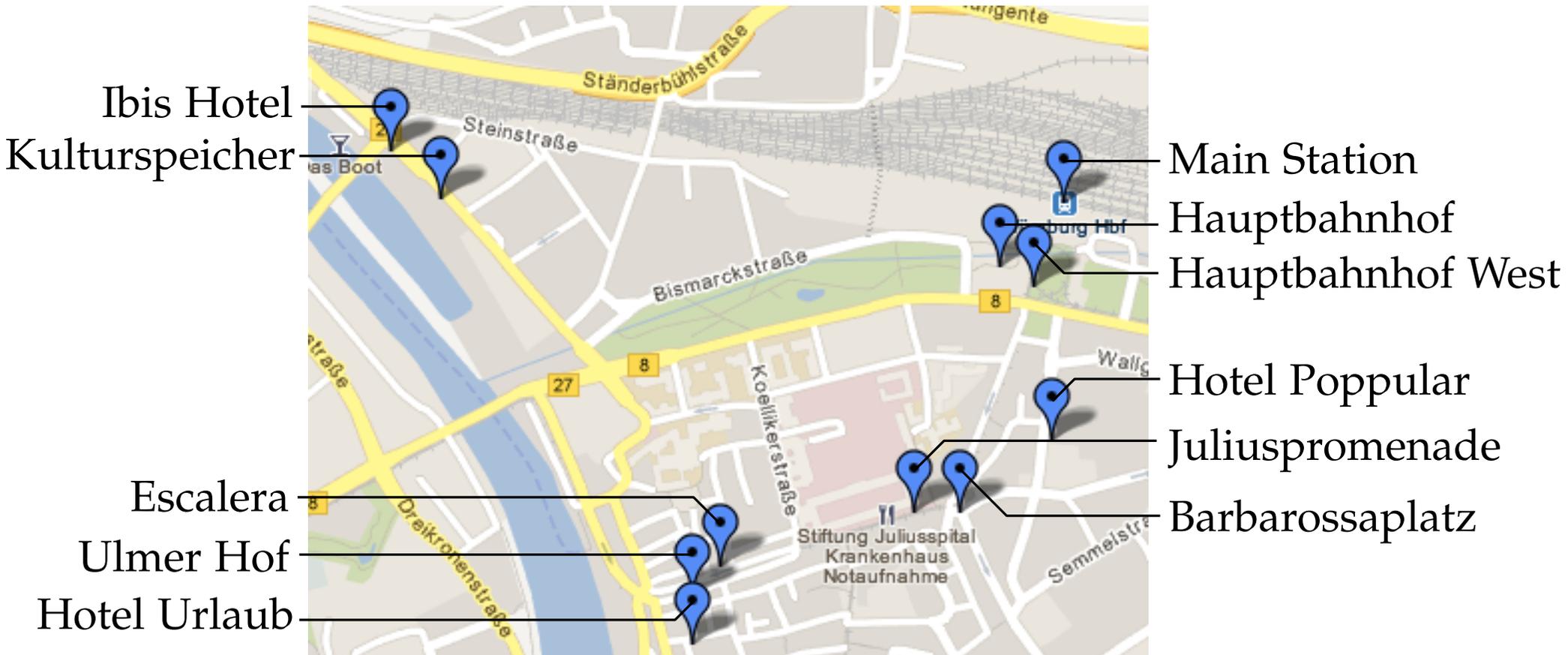
Randbeschriftungen



Nachteile klassischer Kartenbeschriftungen

- funktioniert nur gut für dünne Punktmengen
- Beschriftung überdecken die Karte
- Nicht alle Beschriftungen können angezeigt werden

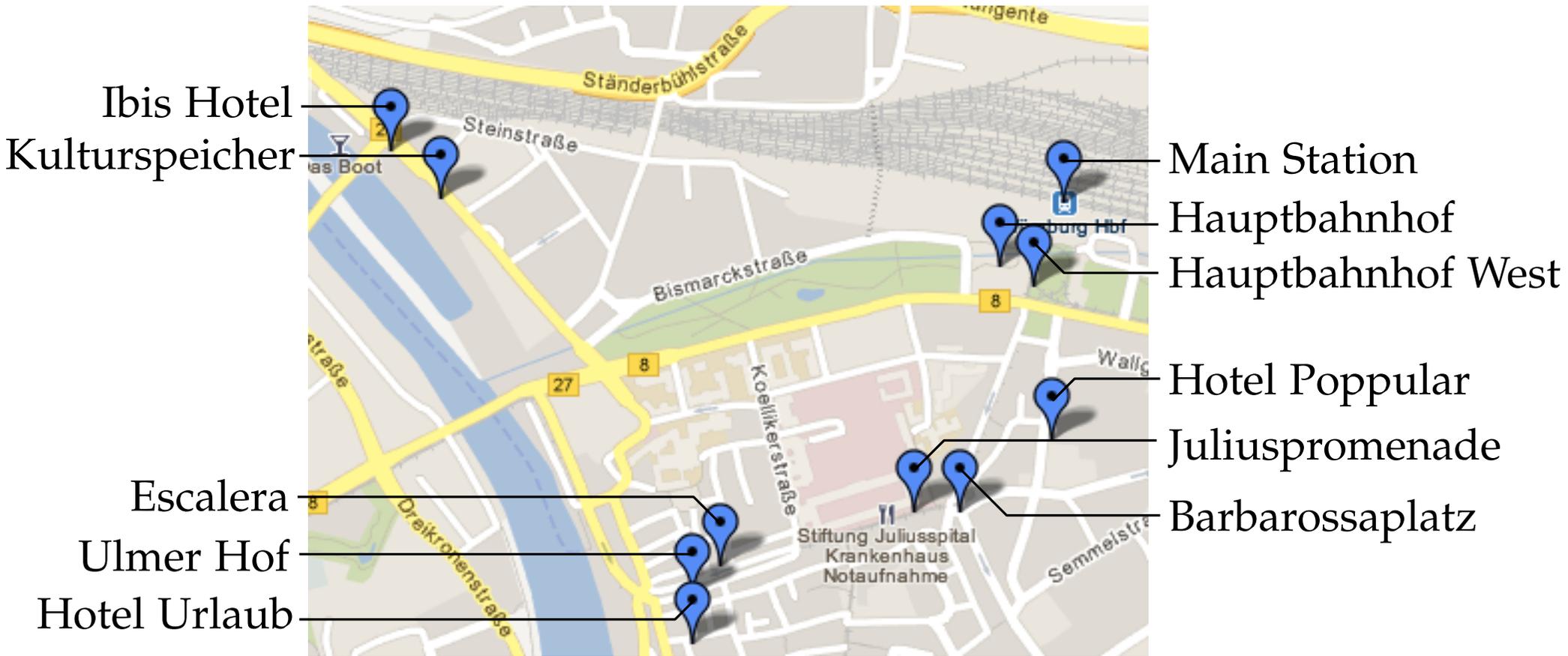
Randbeschriftungen



Nachteile klassischer Kartenbeschriftungen

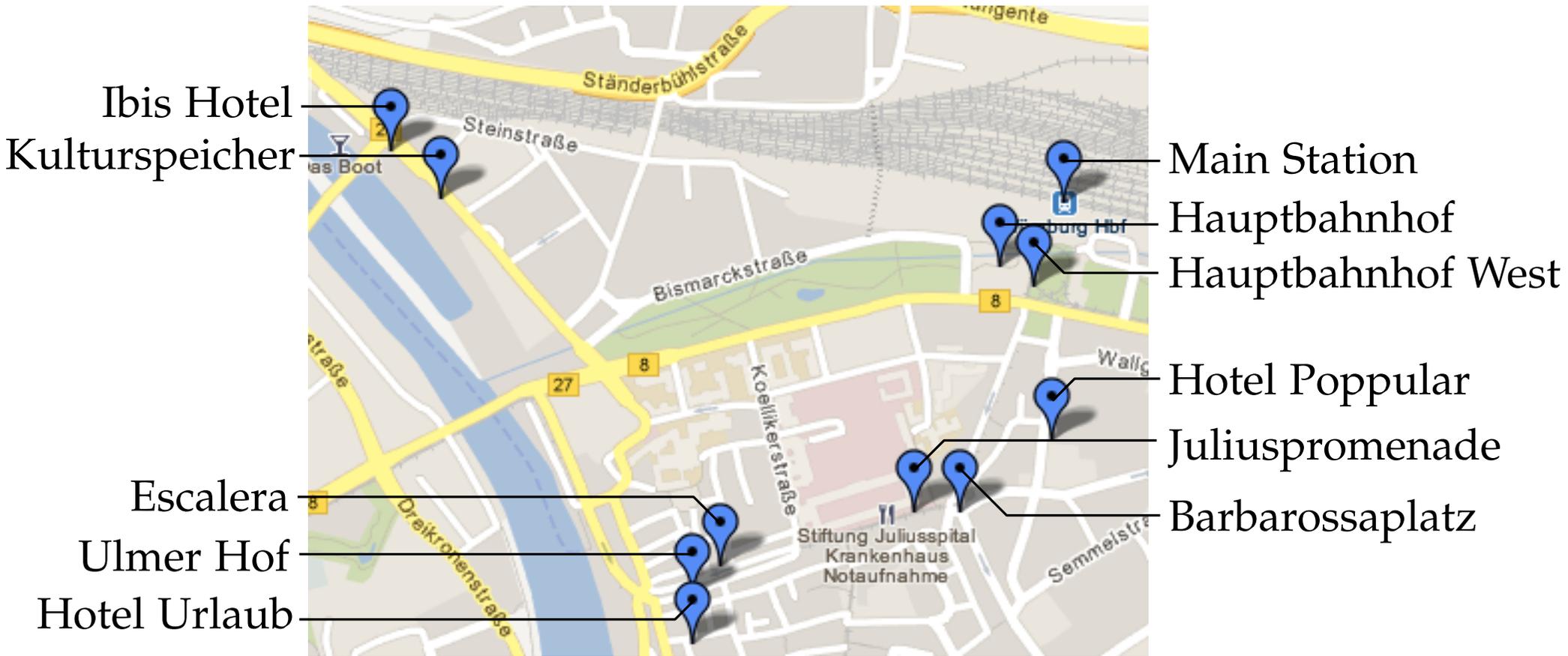
- funktioniert nur gut für dünne Punktmengen
- Beschriftung überdecken die Karte
- Nicht alle Beschriftungen können angezeigt werden

Randbeschriftungen



Vorteile von Randbeschriftungen

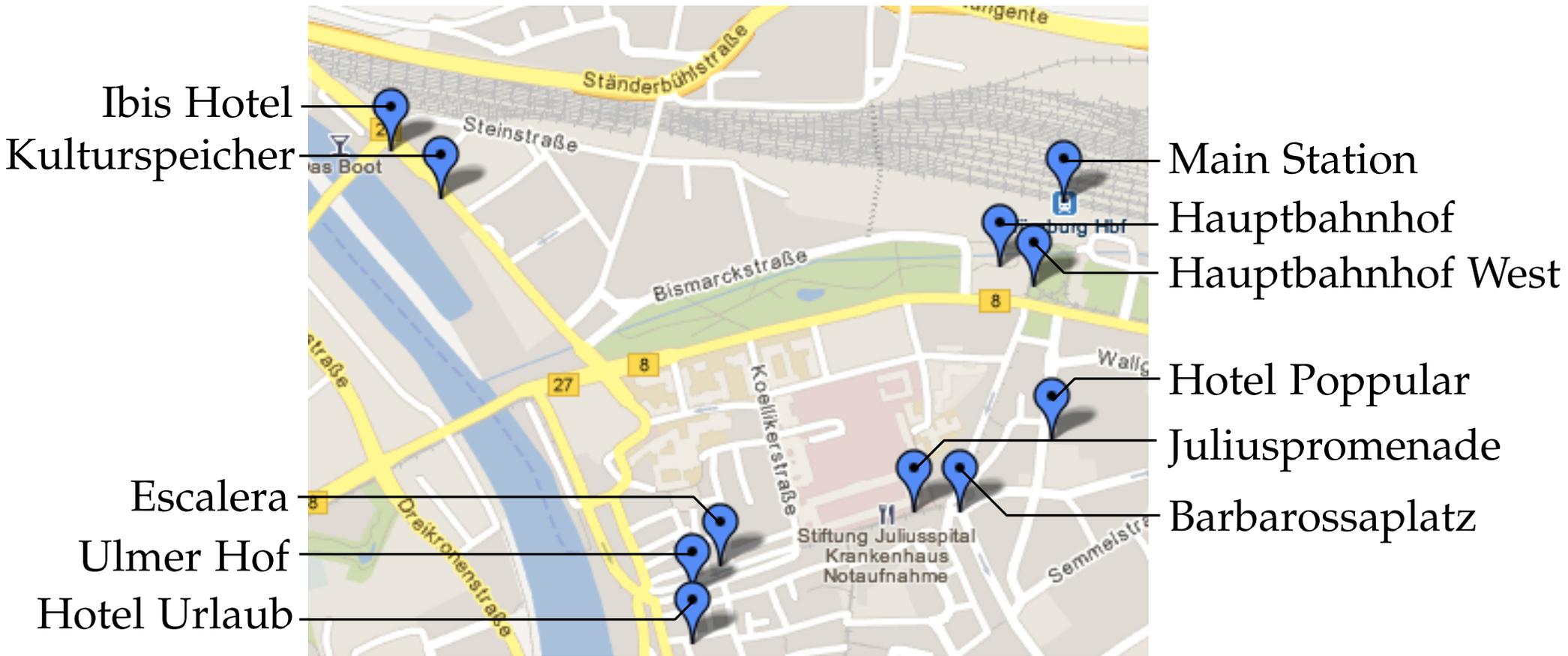
Randbeschriftungen



Vorteile von Randbeschriftungen

- Funktioniert gut für dichte Punktmengen

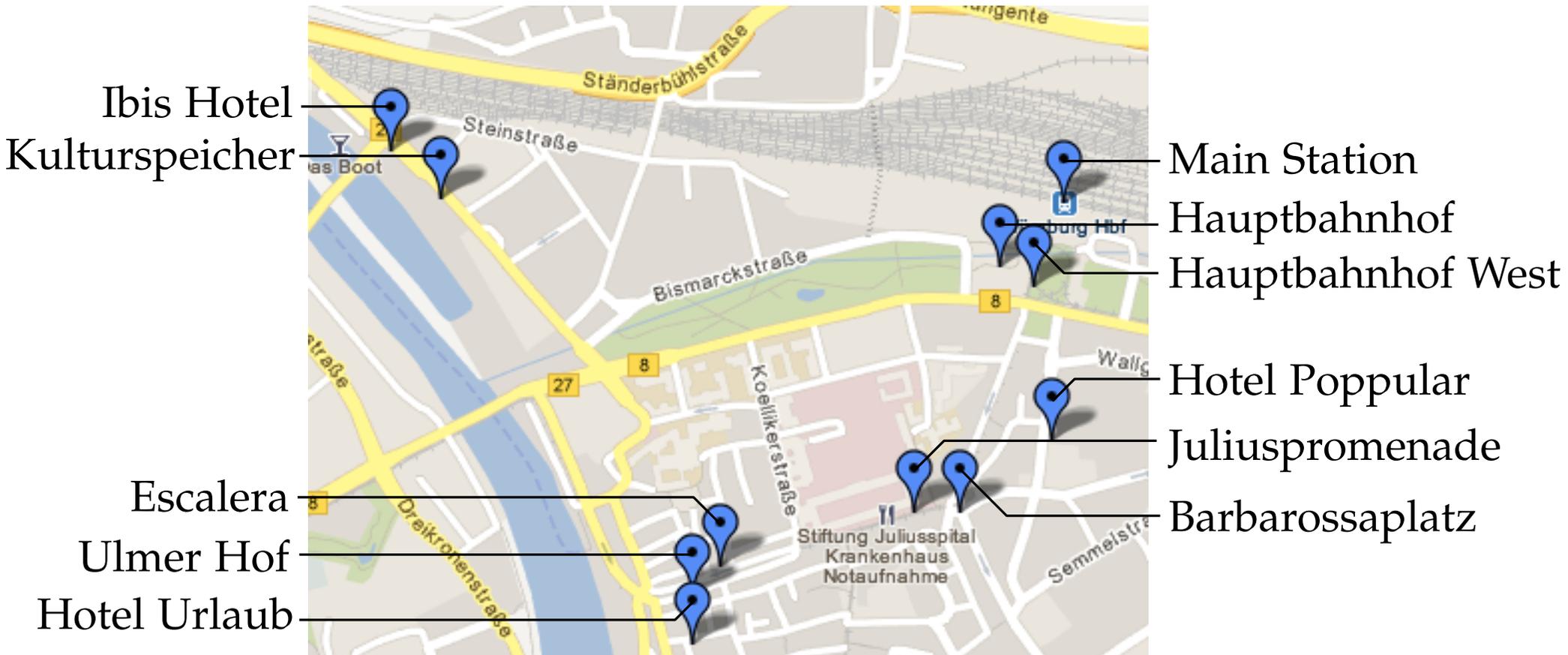
Randbeschriftungen



Vorteile von Randbeschriftungen

- Funktioniert gut für dichte Punktmengen
- Karte wird weniger verdeckt

Randbeschriftungen



Vorteile von Randbeschriftungen

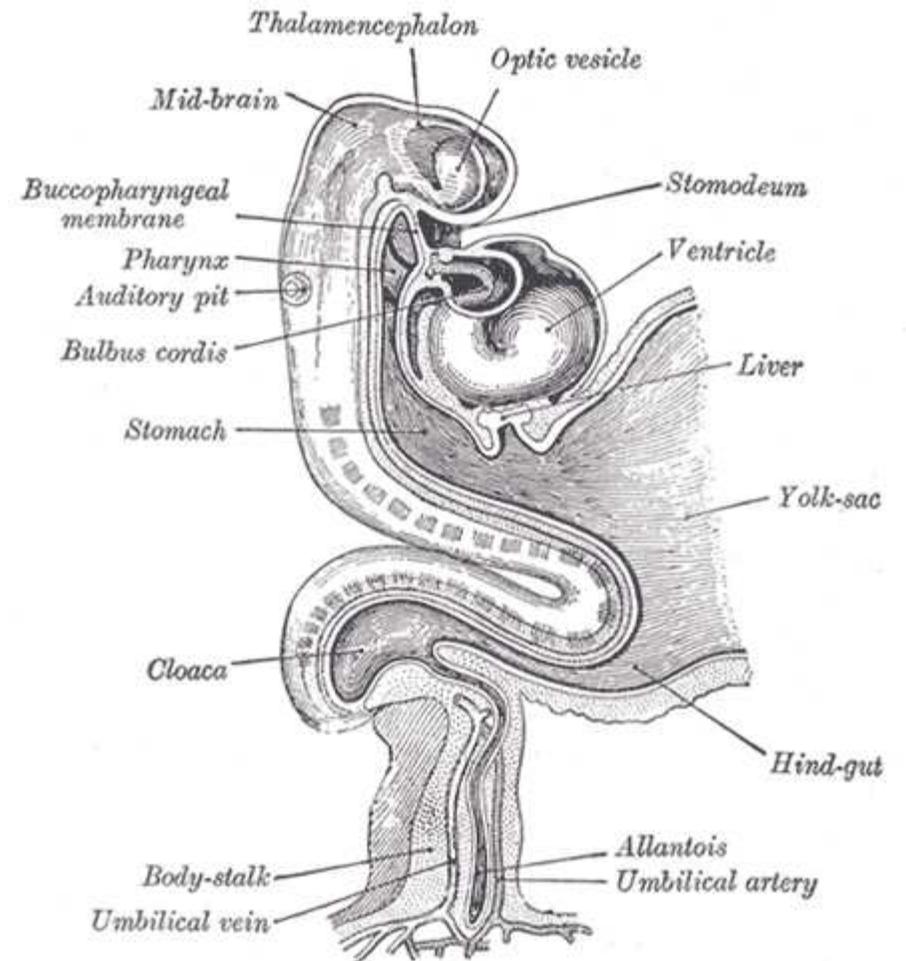
- Funktioniert gut für dichte Punktmengen
- Karte wird weniger verdeckt
- Alle Punkte können beschriftet werden

Randbeschriftungen



©DW-TV

Randbeschriftungen

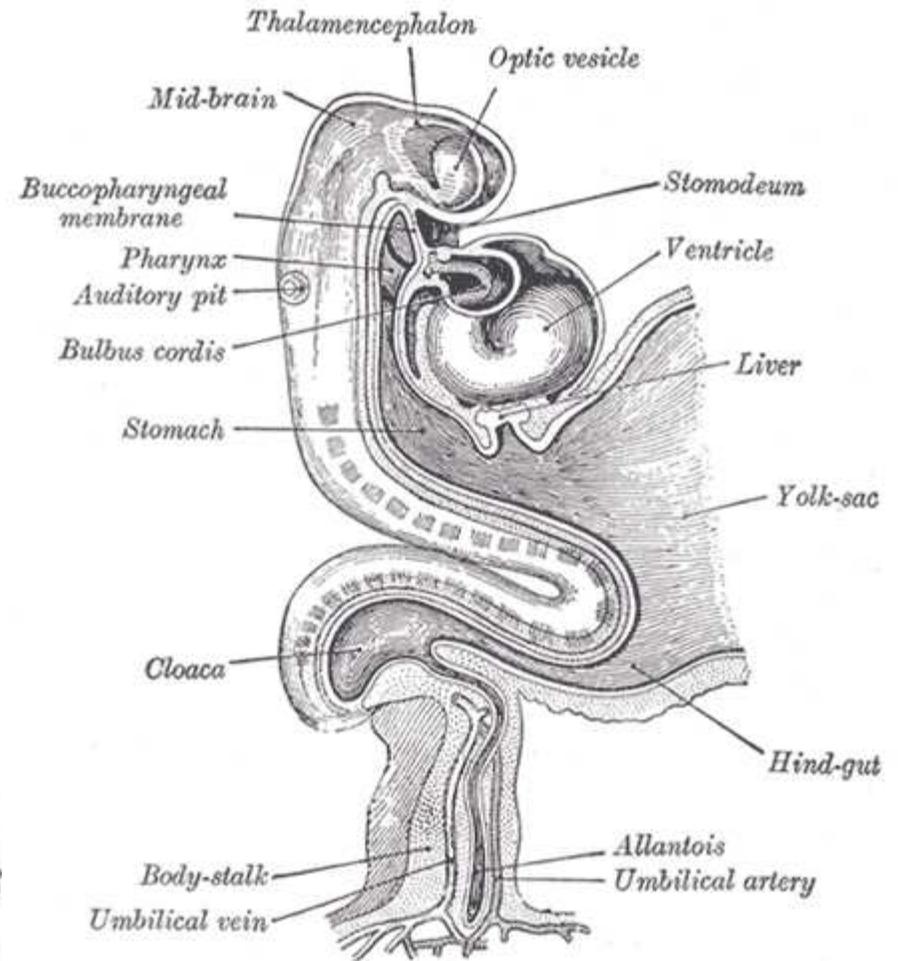
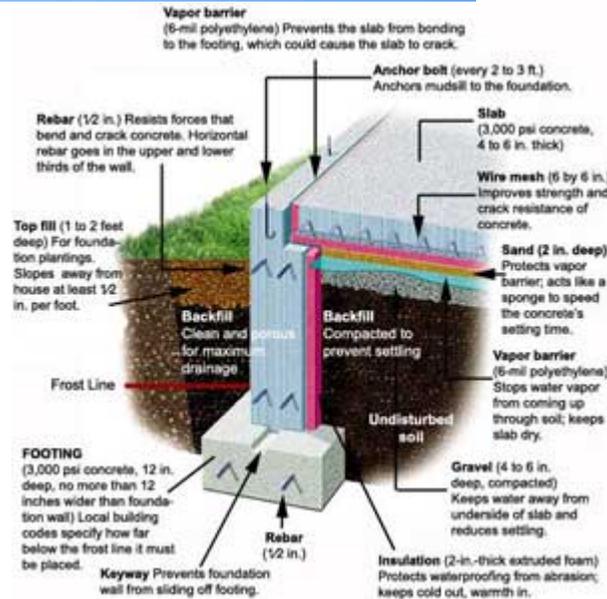


Henry Vandyke Carter, via Wikimedia Commons

Randbeschriftungen



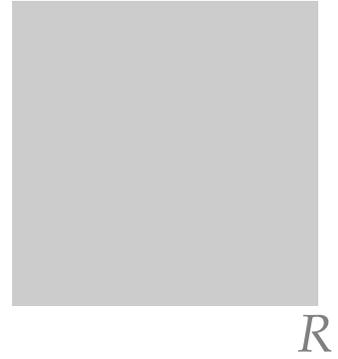
©DW-TV



Henry Vandyke Carter, via Wikimedia Commons

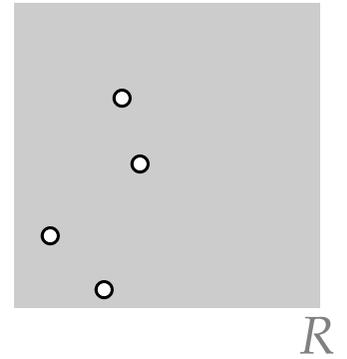
Problembeschreibung

Geg.: • Ein umschließendes Rechteck R



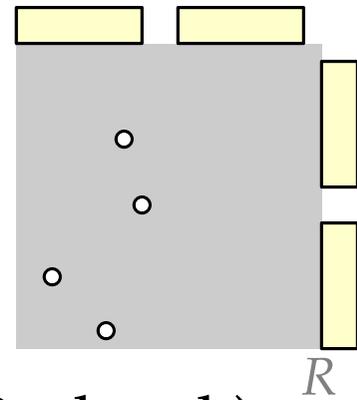
Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage



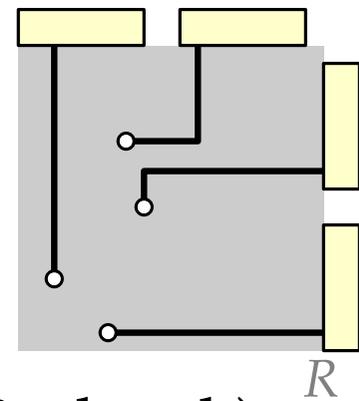
Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



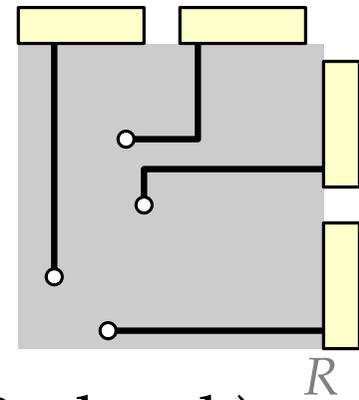
Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)
- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R , zulässige Verbindungen von Punkten zu Beschriftungen, so dass Beschriftungsqualität optimiert wird.



Problembeschreibung

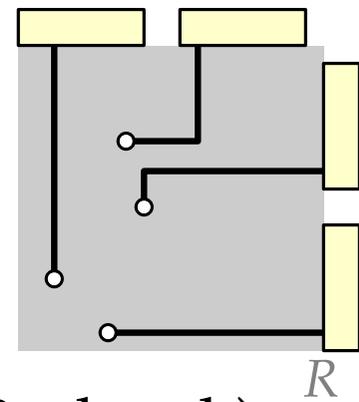
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R , zulässige Verbindungen von Punkten zu Beschriftungen, so dass Beschriftungsqualität optimiert wird.

Problembeschreibung

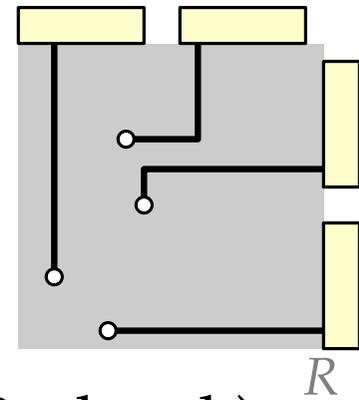
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R , zulässige Verbindungen von Punkten zu Beschriftungen, so dass Beschriftungsqualität optimiert wird.

Problembeschreibung

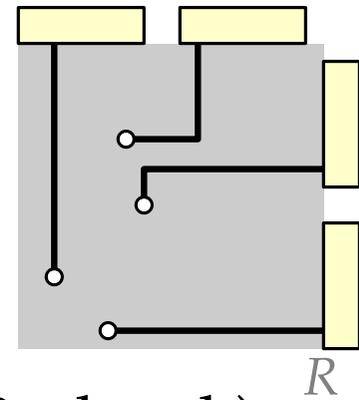
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
leader → zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

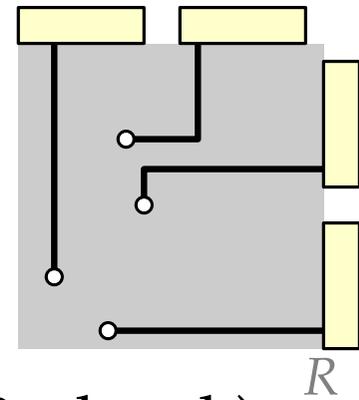


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

leader →

Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

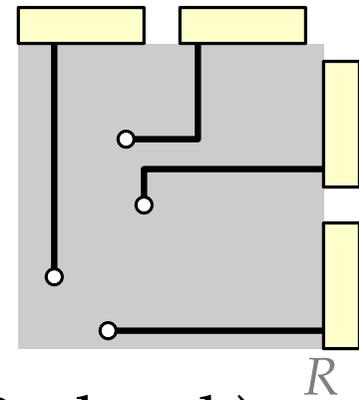


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
leader → zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Zulässige Platzierung der Beschriftungen

Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



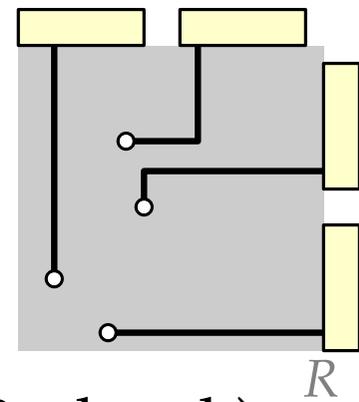
- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Zulässige Platzierung der Beschriftungen

- disjunkt

Problembeschreibung

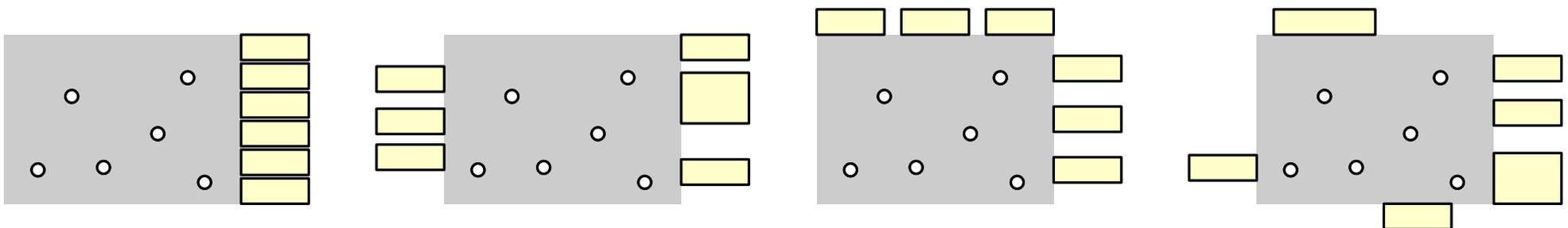
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

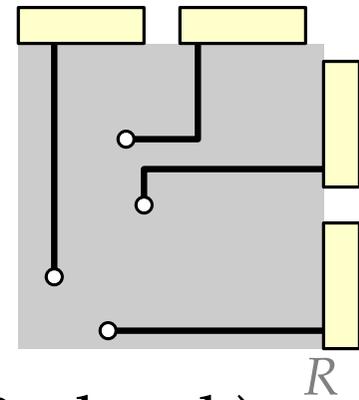
Zulässige Platzierung der Beschriftungen

- disjunkt



Problembeschreibung

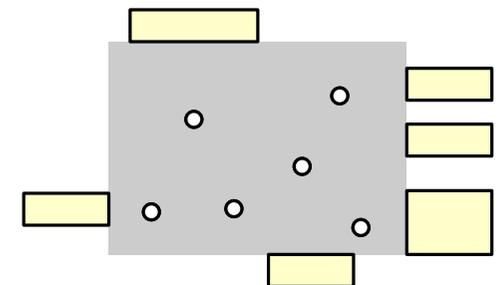
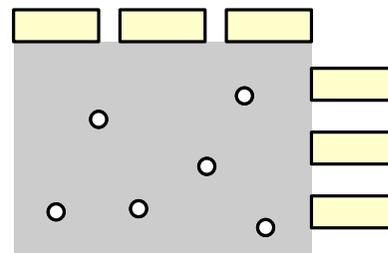
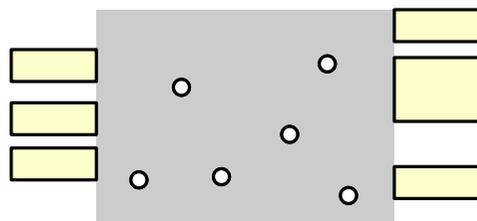
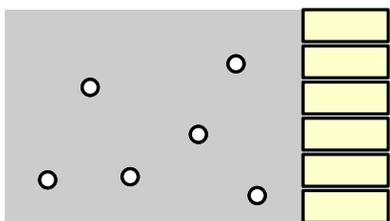
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

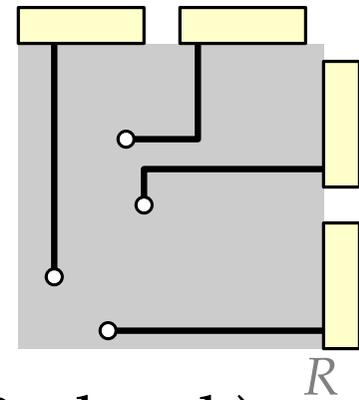
Zulässige Platzierung der Beschriftungen

- disjunkt
- Welche Seiten?



Problembeschreibung

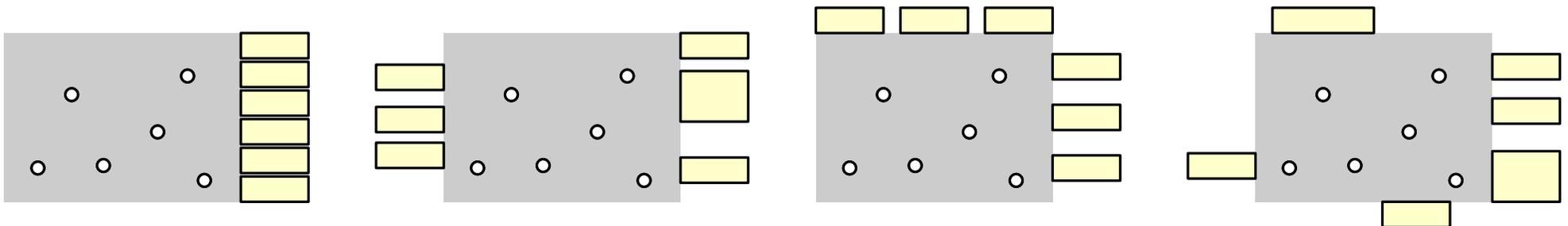
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
leader → zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

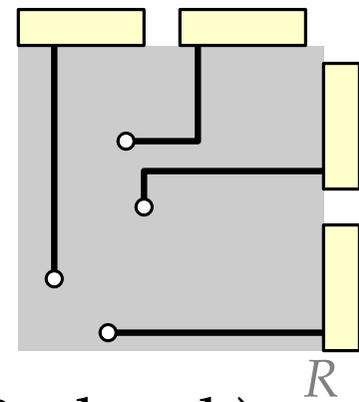
Zulässige Platzierung der Beschriftungen

- disjunkt
- Welche Seiten?
- Uniforme Beschriftungen?



Problembeschreibung

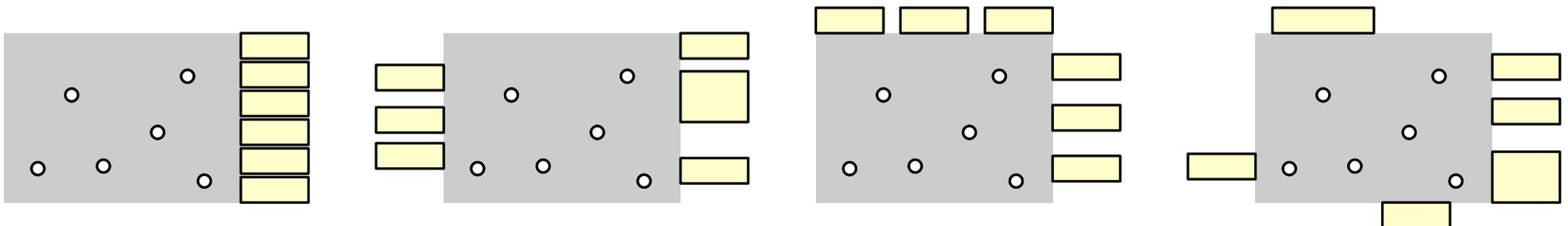
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

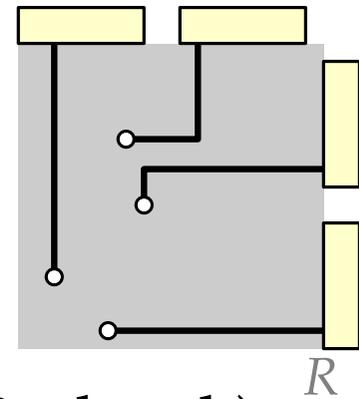
Zulässige Platzierung der Beschriftungen

- disjunkt
- Welche Seiten?
- Uniforme Beschriftungen?
- Fixe Positionen der Beschriftungen?



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

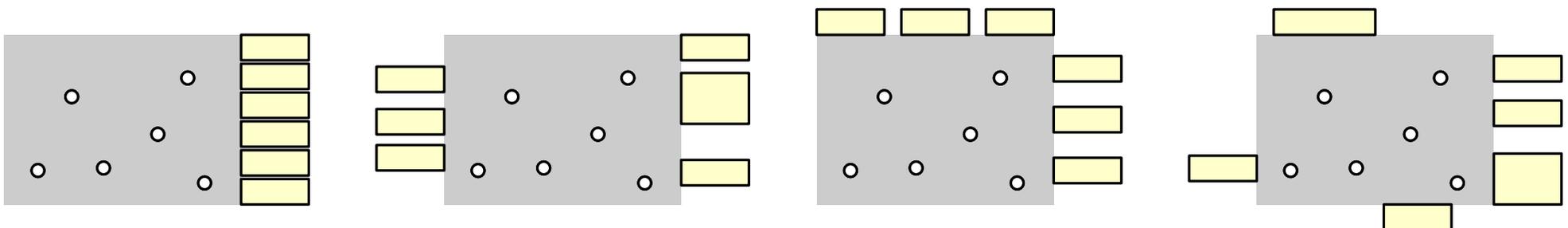


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Zulässige Platzierung der Beschriftungen

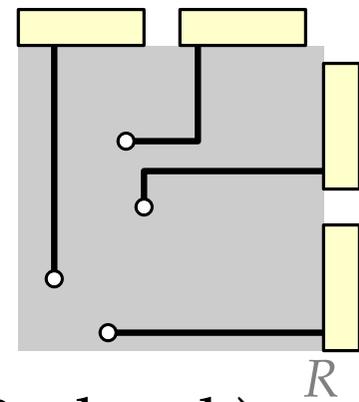
- disjunkt
- Welche Seiten?
- Uniforme Beschriftungen?
- Fixe Positionen der Beschriftungen?

1 Seite, uniform, fix



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

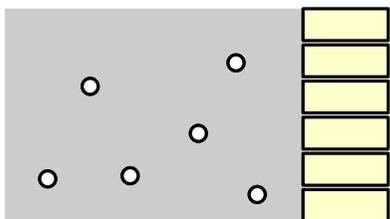


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

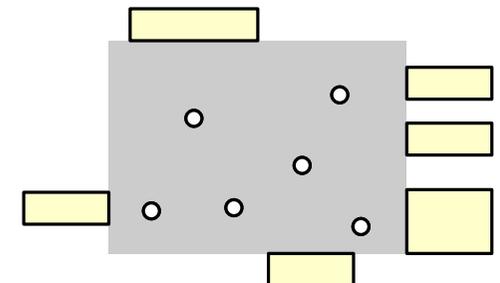
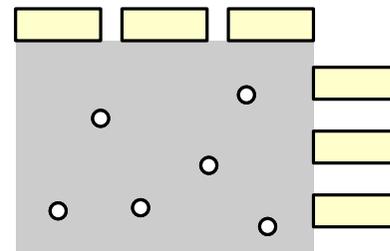
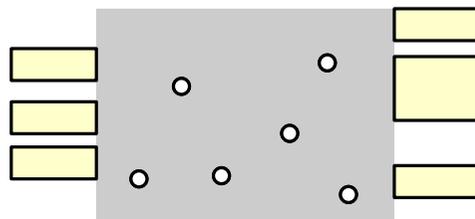
Zulässige Platzierung der Beschriftungen

- disjunkt
- Welche Seiten?
- Uniforme Beschriftungen?
- Fixe Positionen der Beschriftungen?

1 Seite, uniform, fix

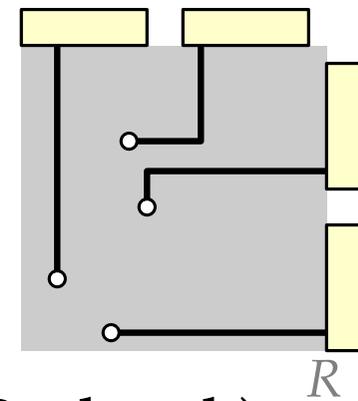


2 Seiten (gegenüber),
nicht uniform, nicht fix



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

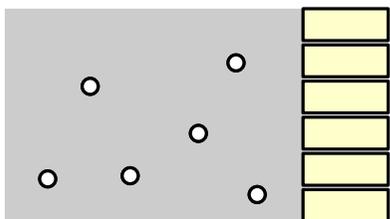


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

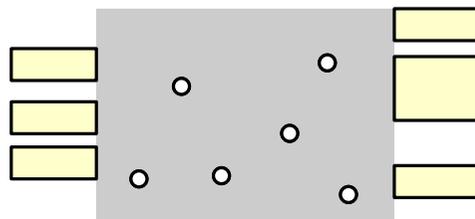
Zulässige Platzierung der Beschriftungen

- disjunkt
- Welche Seiten?
- Uniforme Beschriftungen?
- Fixe Positionen der Beschriftungen?

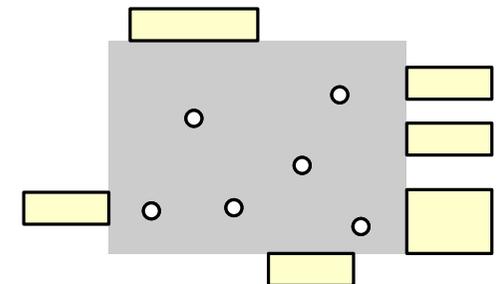
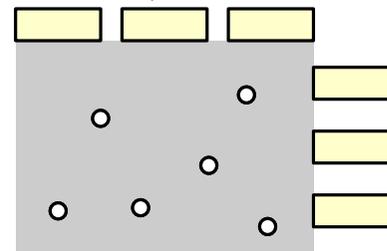
1 Seite, uniform, fix



2 Seiten (gegenüber),
nicht uniform, nicht fix

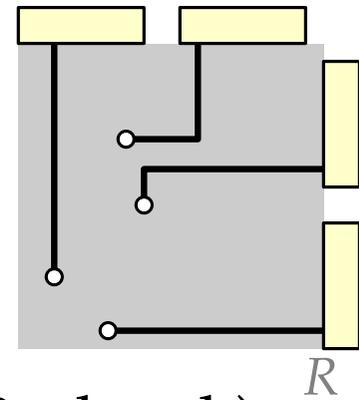


2 Seiten (benachbart),
uniform, fix



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

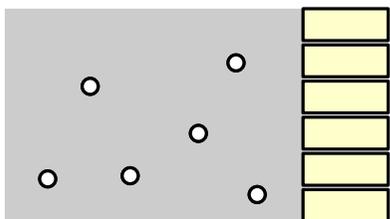


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

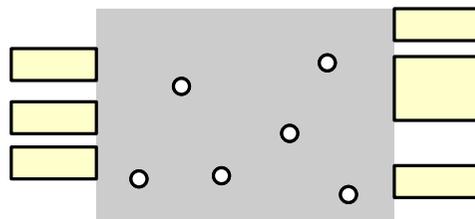
Zulässige Platzierung der Beschriftungen

- disjunkt
- Welche Seiten?
- Uniforme Beschriftungen?
- Fixe Positionen der Beschriftungen?

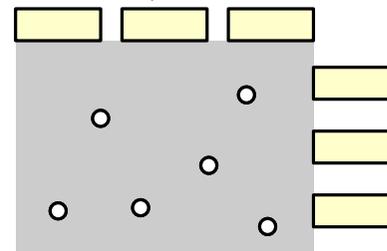
1 Seite, uniform, fix



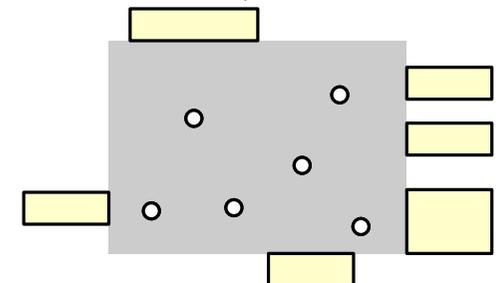
2 Seiten (gegenüber),
nicht uniform, nicht fix



2 Seiten (benachbart),
uniform, fix

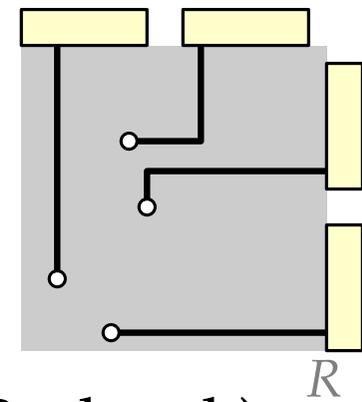


4 Seiten, nicht
uniform, nicht fix



Problembeschreibung

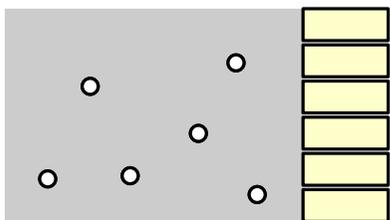
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



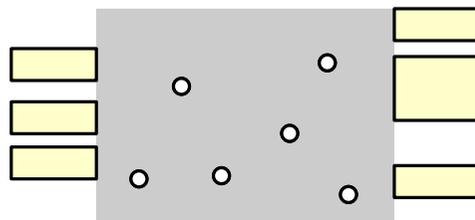
- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Zulässige Verbindungen

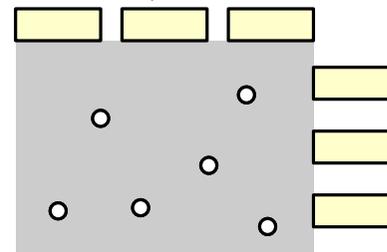
1 Seite, uniform, fix



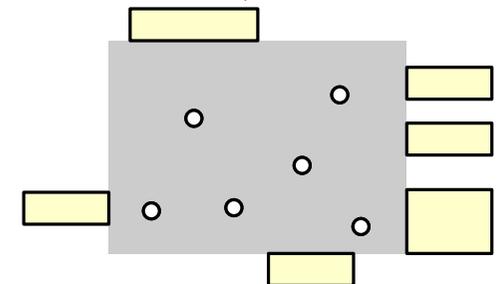
2 Seiten (gegenüber),
nicht uniform, nicht fix



2 Seiten (benachbart),
uniform, fix

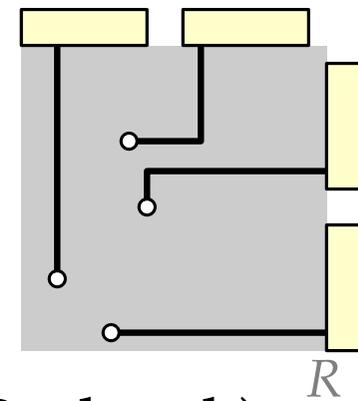


4 Seiten, nicht
uniform, nicht fix



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

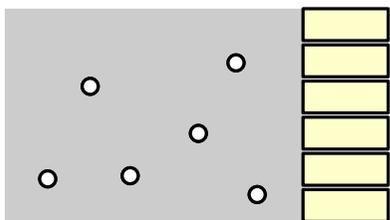


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

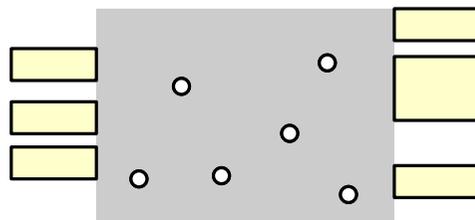
Zulässige Verbindungen

- kreuzungsfrei

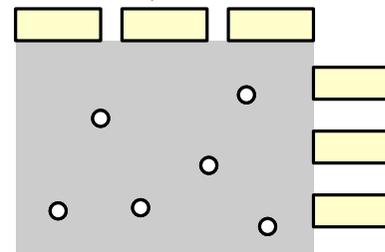
1 Seite, uniform, fix



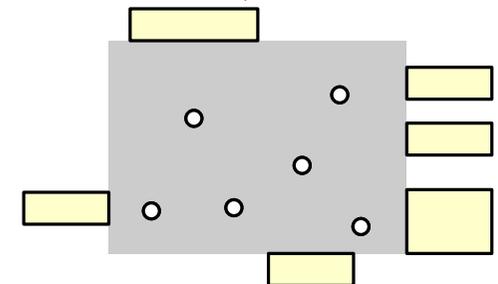
2 Seiten (gegenüber),
nicht uniform, nicht fix



2 Seiten (benachbart),
uniform, fix

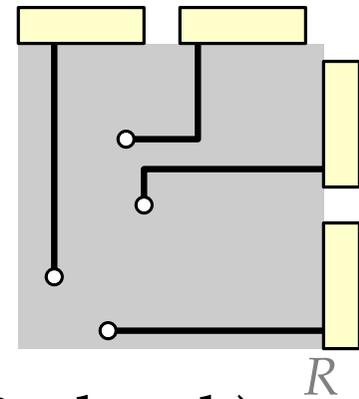


4 Seiten, nicht
uniform, nicht fix



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

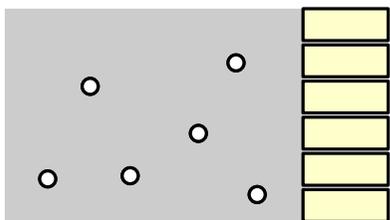


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

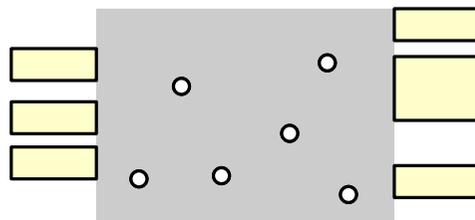
Zulässige Verbindungen

- kreuzungsfrei
- Form? (parallel, orthogonal, diagonal, segment, ...)

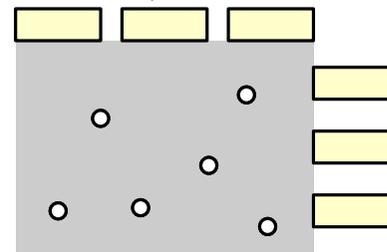
1 Seite, uniform, fix



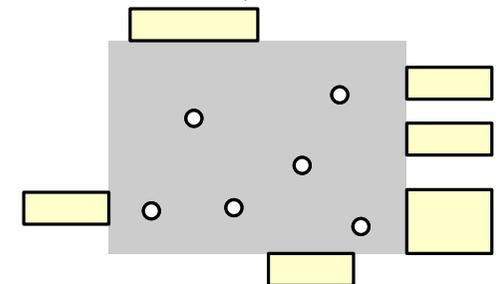
2 Seiten (gegenüber),
nicht uniform, nicht fix



2 Seiten (benachbart),
uniform, fix

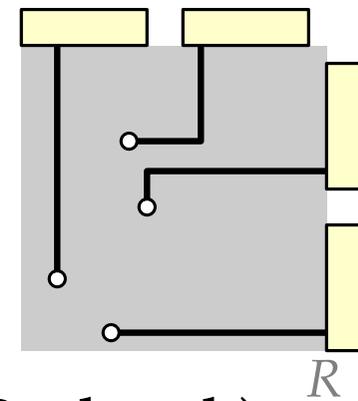


4 Seiten, nicht
uniform, nicht fix



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

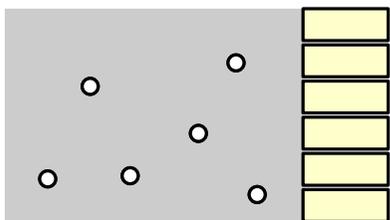


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

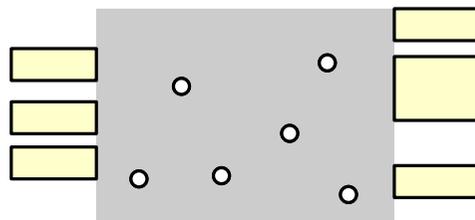
Zulässige Verbindungen

- kreuzungsfrei
- Form? (parallel, orthogonal, diagonal, segment, ...)
- Fixe Andockpunkte der Verbindungen?

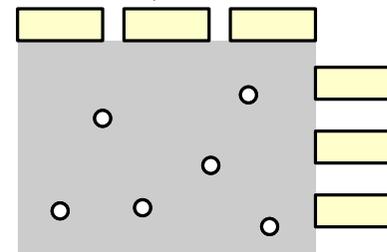
1 Seite, uniform, fix



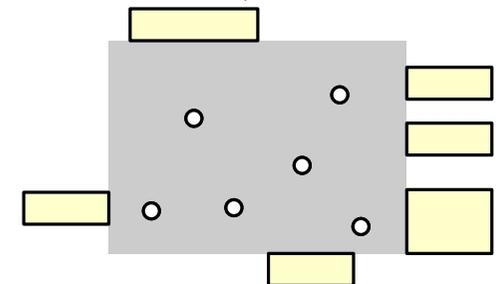
2 Seiten (gegenüber),
nicht uniform, nicht fix



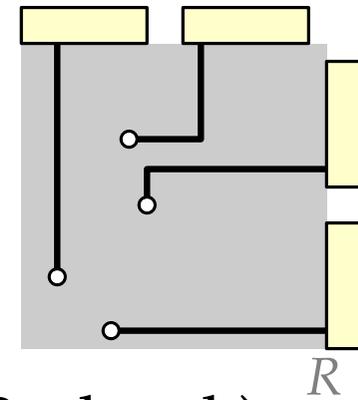
2 Seiten (benachbart),
uniform, fix



4 Seiten, nicht
uniform, nicht fix



Problembeschreibung



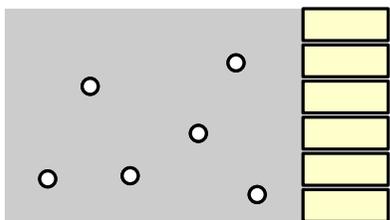
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

Ges.: Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

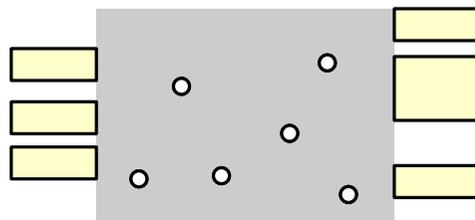
Zulässige Verbindungen

- kreuzungsfrei
- Form? (parallel, orthogonal, diagonal, segment, ...)
- Fixe Andockpunkte der Verbindungen?

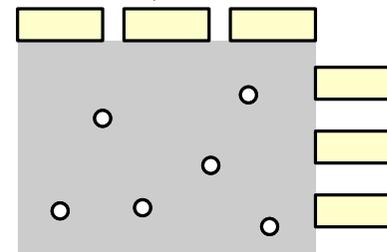
1 Seite, uniform, fix



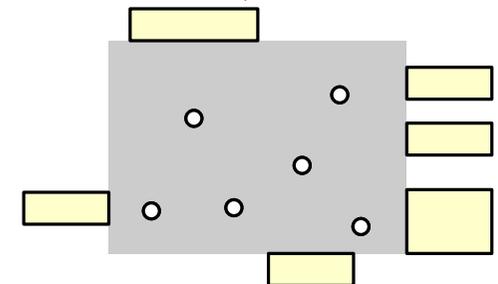
2 Seiten (gegenüber),
nicht uniform, nicht fix



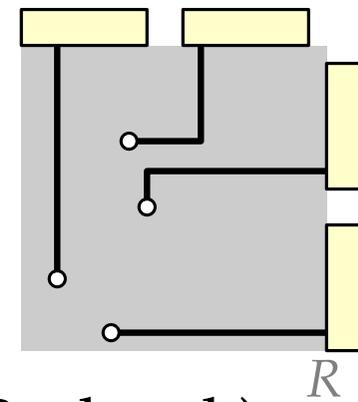
2 Seiten (benachbart),
uniform, fix



4 Seiten, nicht
uniform, nicht fix



Problembeschreibung



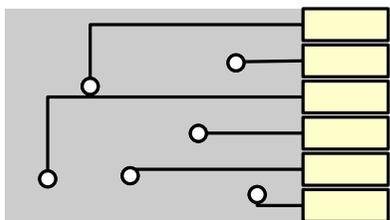
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

Ges.: Zulässige Platzierung der Beschriftungen am Rand von R ,
leader → zulässige Verbindungen von Punkten zu Beschriftungen,
 so dass **Beschriftungsqualität** optimiert wird.

Zulässige Verbindungen

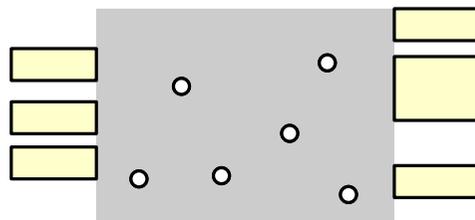
- kreuzungsfrei
- Form? (**p**arallel, **o**rthogonal, **d**iagonal, **s**egment, ...)
- Fixe Andockpunkte der Verbindungen? **ports**

1 Seite, uniform, fix

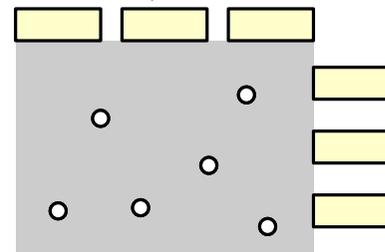


po, fix

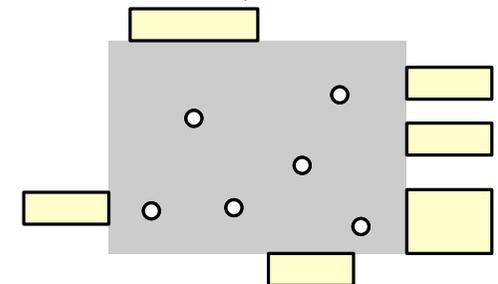
2 Seiten (gegenüber),
nicht uniform, nicht fix



2 Seiten (benachbart),
uniform, fix

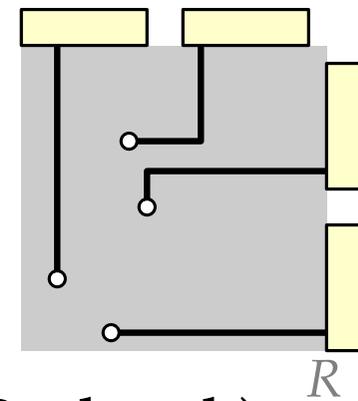


4 Seiten, nicht
uniform, nicht fix



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

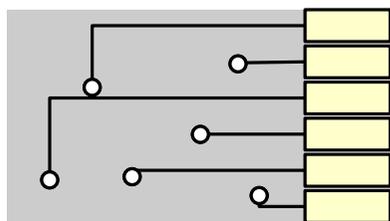


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Zulässige Verbindungen

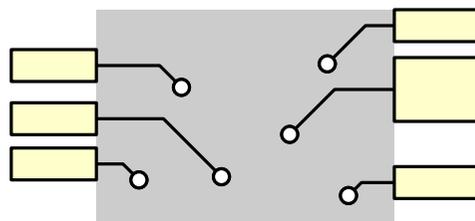
- kreuzungsfrei
- Form? (parallel, orthogonal, diagonal, segment, ...)
- Fixe Andockpunkte der Verbindungen?

1 Seite, uniform, fix



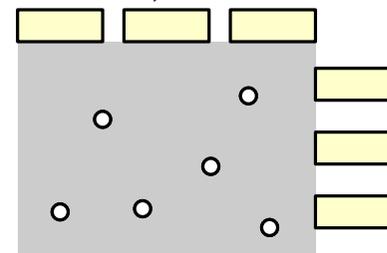
po, fix

2 Seiten (gegenüber),
nicht uniform, nicht fix

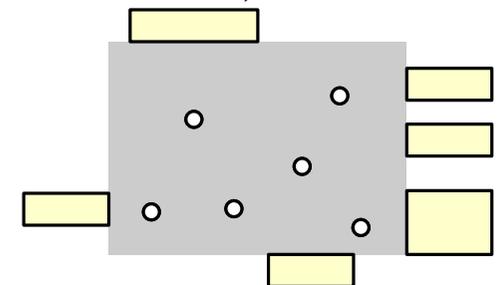


do, frei

2 Seiten (benachbart),
uniform, fix

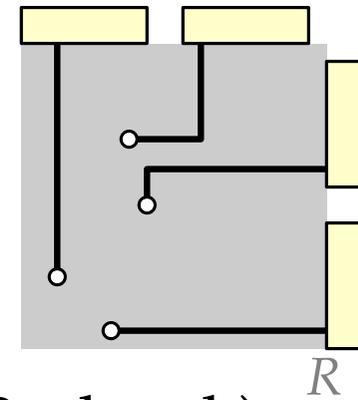


4 Seiten, nicht
uniform, nicht fix



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

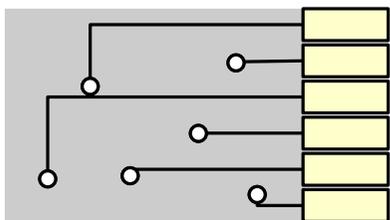


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
leader → zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Zulässige Verbindungen

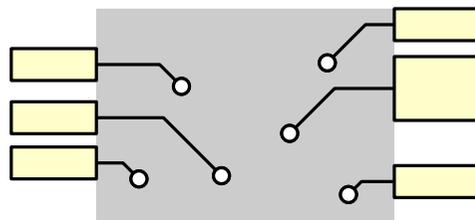
- kreuzungsfrei
- Form? (parallel, orthogonal, diagonal, segment, ...)
- Fixe Andockpunkte der Verbindungen? ports

1 Seite, uniform, fix



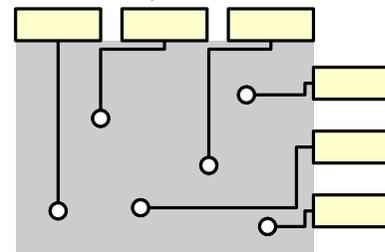
po, fix

2 Seiten (gegenüber),
nicht uniform, nicht fix



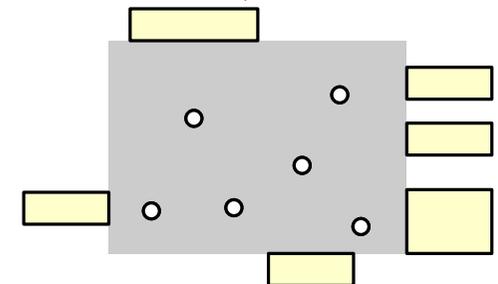
do, frei

2 Seiten (benachbart),
uniform, fix



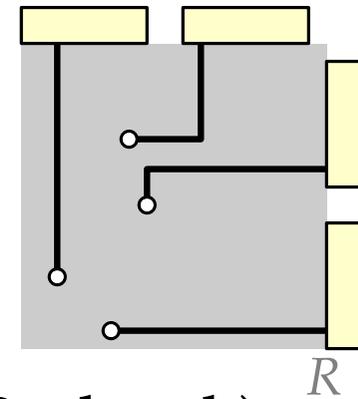
opo, fix

4 Seiten, nicht
uniform, nicht fix



Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

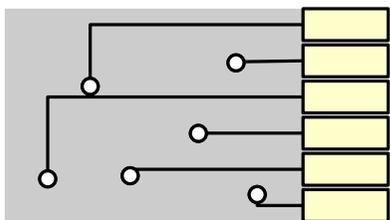


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
leader → zulässige Verbindungen von Punkten zu Beschriftungen,
 so dass **Beschriftungsqualität** optimiert wird.

Zulässige Verbindungen

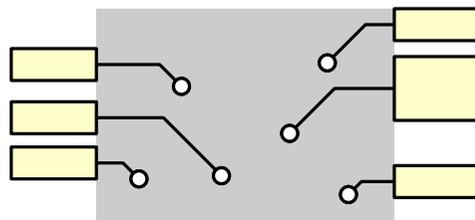
- kreuzungsfrei
- Form? (**p**arallel, **o**rthogonal, **d**iagonal, **s**egment, ...)
- Fixe Andockpunkte der Verbindungen? ports

1 Seite, uniform, fix



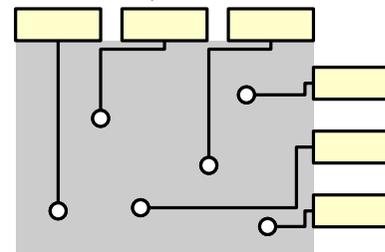
po, fix

2 Seiten (gegenüber),
nicht uniform, nicht fix



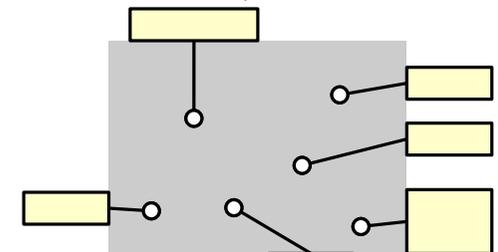
do, frei

2 Seiten (benachbart),
uniform, fix



opo, fix

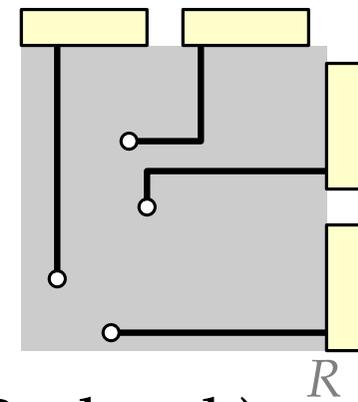
4 Seiten, nicht
uniform, nicht fix



s, frei

Problembeschreibung

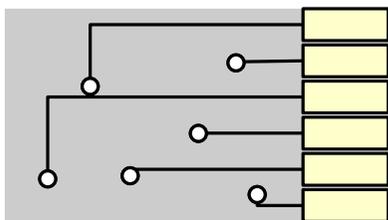
- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)



- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
leader → zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

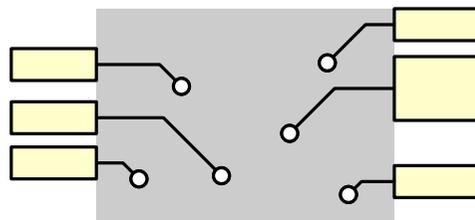
Beschriftungsqualität

1 Seite, uniform, fix



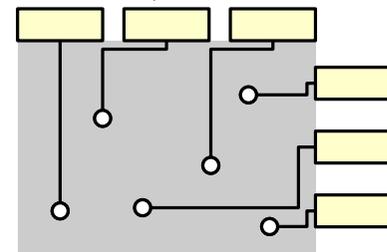
po, fix

2 Seiten (gegenüber),
nicht uniform, nicht fix



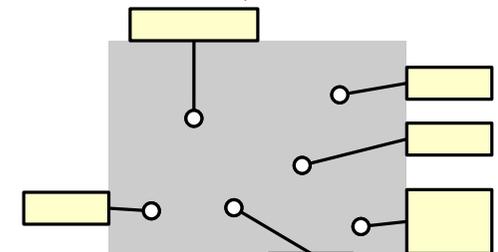
do, frei

2 Seiten (benachbart),
uniform, fix



opo, fix

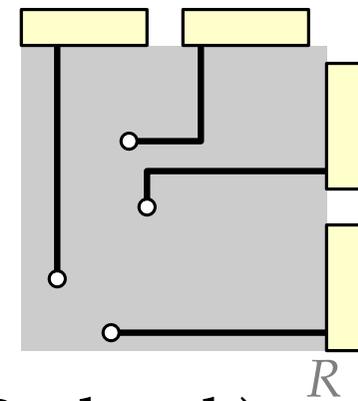
4 Seiten, nicht
uniform, nicht fix



s, frei

Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

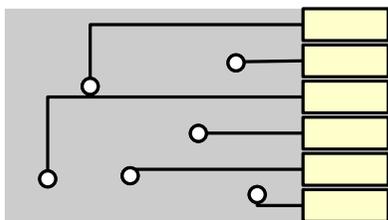


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Beschriftungsqualität

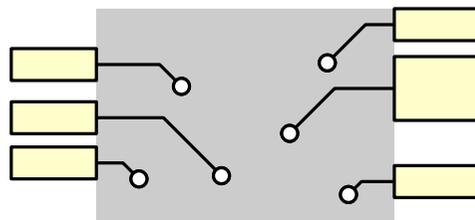
- Min. Länge

1 Seite, uniform, fix



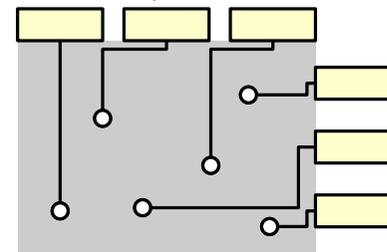
po, fix

2 Seiten (gegenüber),
nicht uniform, nicht fix



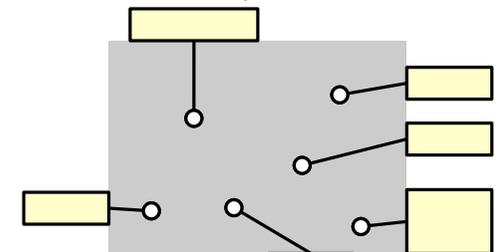
do, frei

2 Seiten (benachbart),
uniform, fix



opo, fix

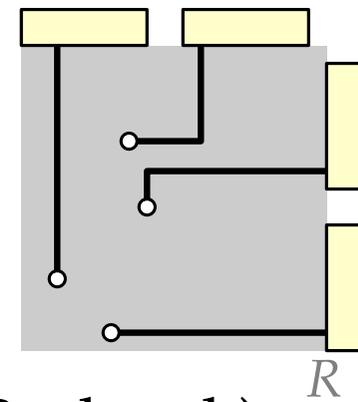
4 Seiten, nicht
uniform, nicht fix



s, frei

Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

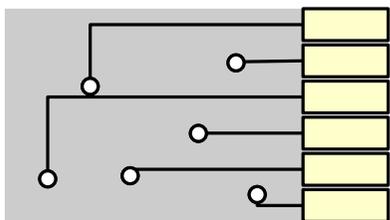


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Beschriftungsqualität

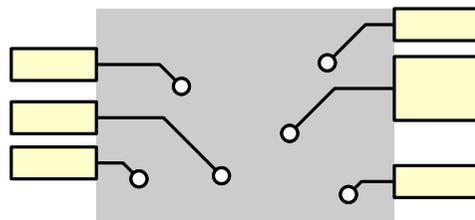
- Min. Länge
- Min. Knicke

1 Seite, uniform, fix



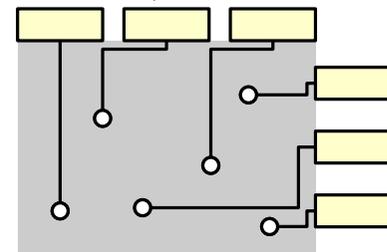
po, fix

2 Seiten (gegenüber),
nicht uniform, nicht fix



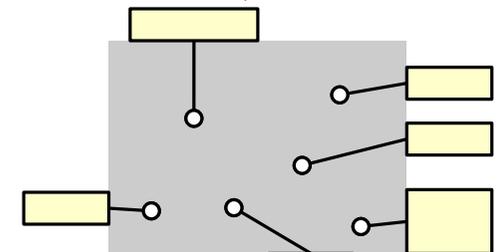
do, frei

2 Seiten (benachbart),
uniform, fix



opo, fix

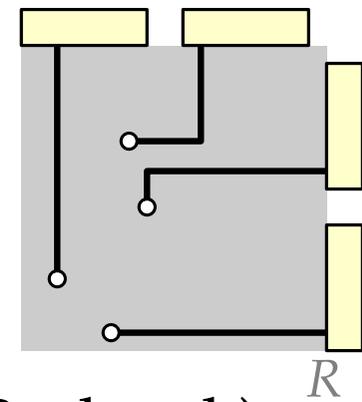
4 Seiten, nicht
uniform, nicht fix



s, frei

Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

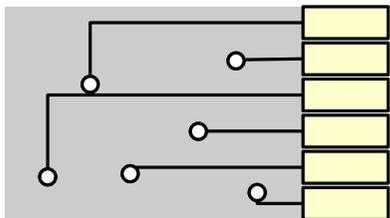


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Beschriftungsqualität

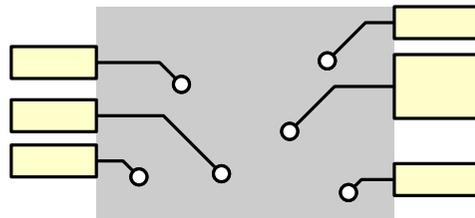
- Min. Länge
- Min. Knicke
- Beliebige Funktion

1 Seite, uniform, fix



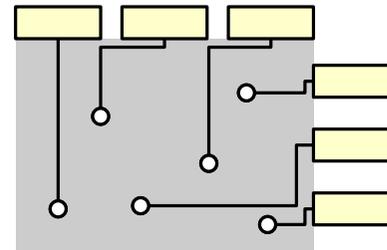
po, fix

2 Seiten (gegenüber),
nicht uniform, nicht fix



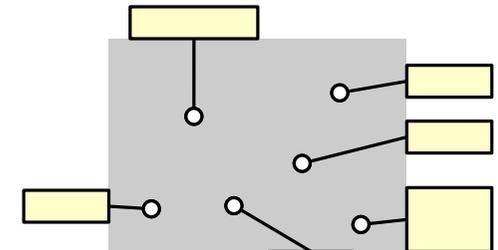
do, frei

2 Seiten (benachbart),
uniform, fix



opo, fix

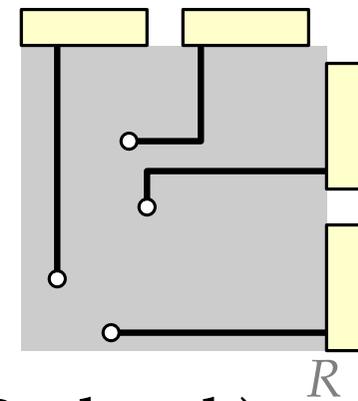
4 Seiten, nicht
uniform, nicht fix



s, frei

Problembeschreibung

- Geg.:**
- Ein umschließendes Rechteck R
 - n Punkte in R in allgemeiner Lage
 - Je Punkt eine Beschriftung (achsen-paralleles Rechteck)

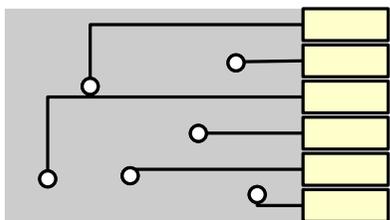


- Ges.:** Zulässige Platzierung der Beschriftungen am Rand von R ,
zulässige Verbindungen von Punkten zu Beschriftungen,
so dass Beschriftungsqualität optimiert wird.

Beschriftungsqualität

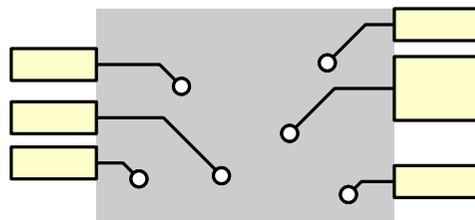
- Min. Länge
- Min. Knicke
- Beliebige Funktion

1 Seite, uniform, fix



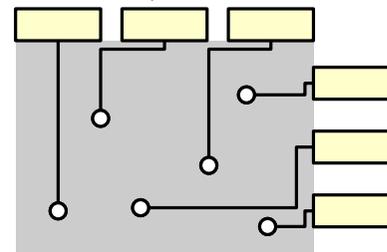
po, fix

2 Seiten (gegenüber),
nicht uniform, nicht fix



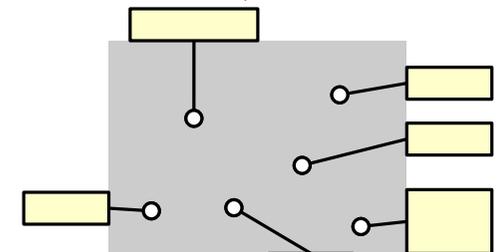
do, frei

2 Seiten (benachbart),
uniform, fix



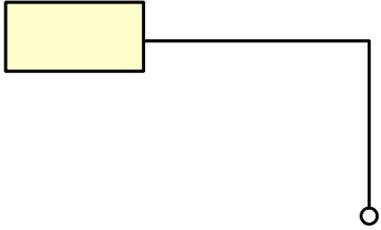
opo, fix

4 Seiten, nicht
uniform, nicht fix



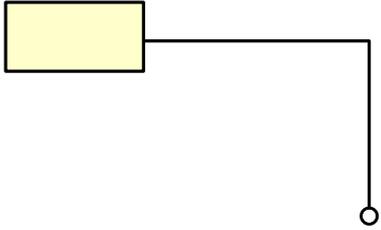
s, frei

po-Leader

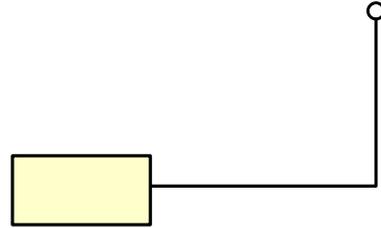


aufsteigender Leader

po-Leader

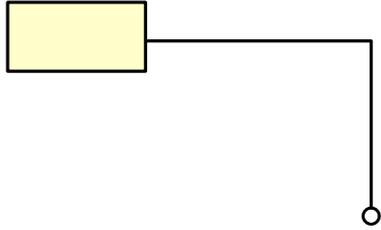


aufsteigender Leader

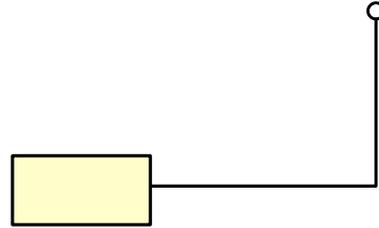


absteigender Leader

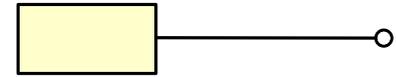
po-Leader



aufsteigender Leader

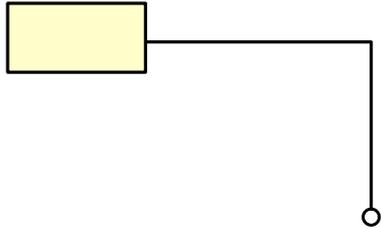


absteigender Leader

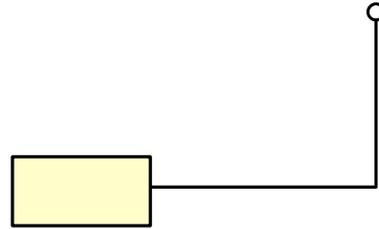


direkter Leader

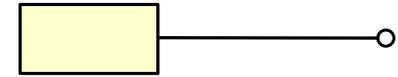
po-Leader



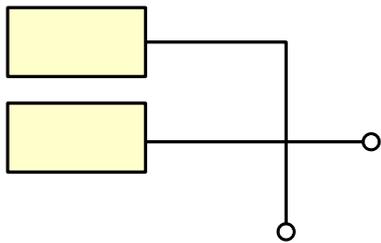
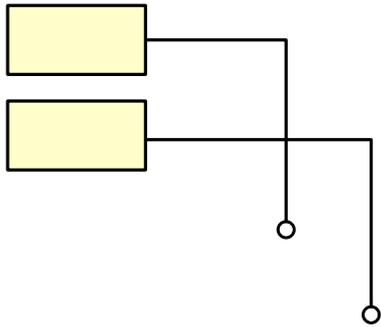
aufsteigender Leader



absteigender Leader

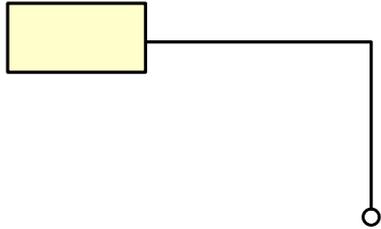


direkter Leader

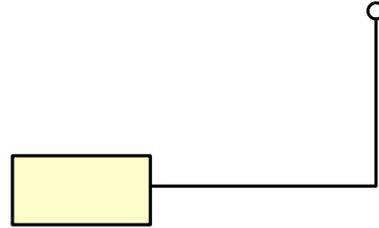


aufsteigende Kreuzung

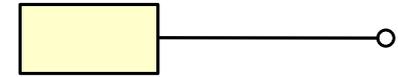
po-Leader



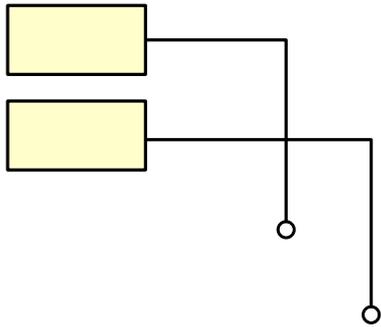
aufsteigender Leader



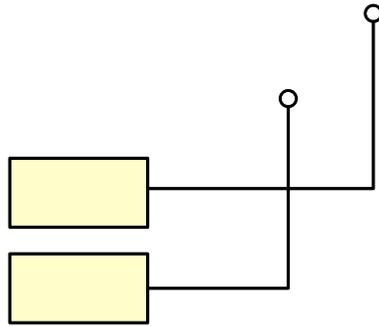
absteigender Leader



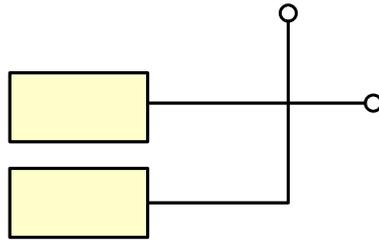
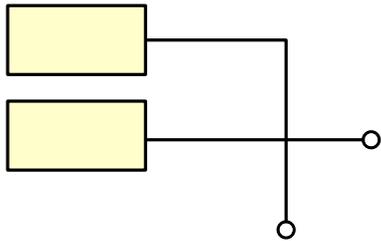
direkter Leader



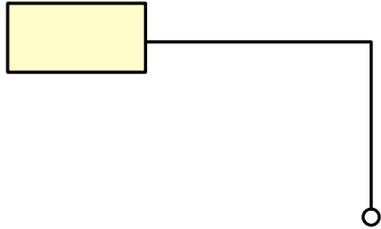
aufsteigende Kreuzung



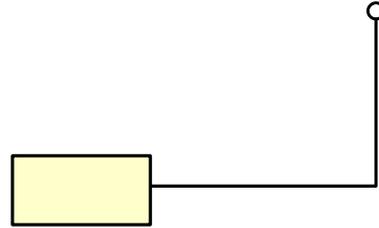
absteigende Kreuzung



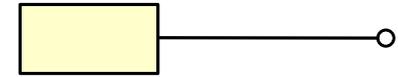
po-Leader



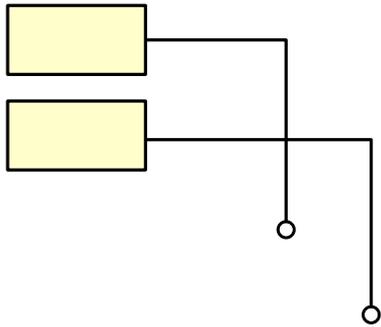
aufsteigender Leader



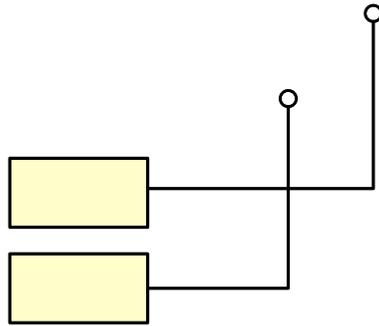
absteigender Leader



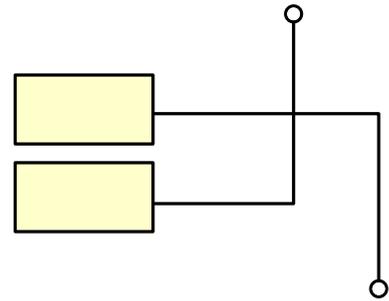
direkter Leader



aufsteigende Kreuzung



absteigende Kreuzung



gemischte Kreuzung

Neuverdrahtung

Geg.: Beschriftung minimaler Länge mit Kreuzungen.

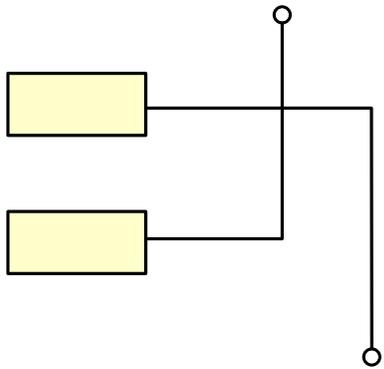
Ges.: Kreuzungsfreie Beschriftung gleicher Länge.

Neuverdrahtung

Geg.: Beschriftung minimaler Länge mit Kreuzungen.

Ges.: Kreuzungsfreie Beschriftung gleicher Länge.

Absteigende und aufsteigende Leader schneiden sich nicht.



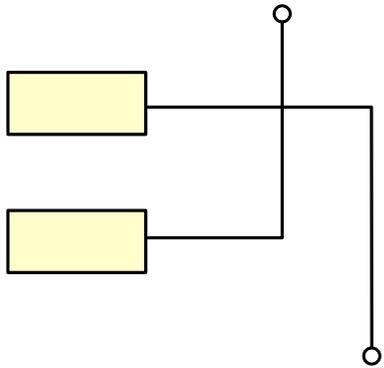
Neuverdrahtung

Geg.: Beschriftung minimaler Länge mit Kreuzungen.

Ges.: Kreuzungsfreie Beschriftung gleicher Länge.

Absteigende und aufsteigende Leader schneiden sich nicht.

Kein Leader schneidet absteigenden und aufsteigenden Leader.



Neuverdrahtung

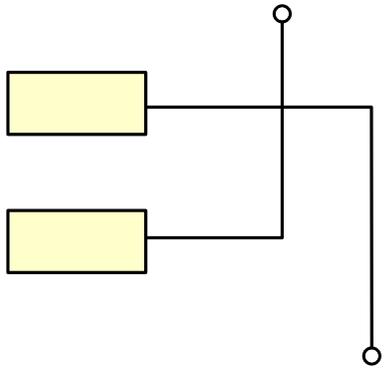
Geg.: Beschriftung minimaler Länge mit Kreuzungen.

Ges.: Kreuzungsfreie Beschriftung gleicher Länge.

Absteigende und aufsteigende Leader schneiden sich nicht.

Kein Leader schneidet absteigenden und aufsteigenden Leader.

⇒ Keine gemischten Kreuzungen.



Neuverdrahtung

Geg.: Beschriftung minimaler Länge mit Kreuzungen.

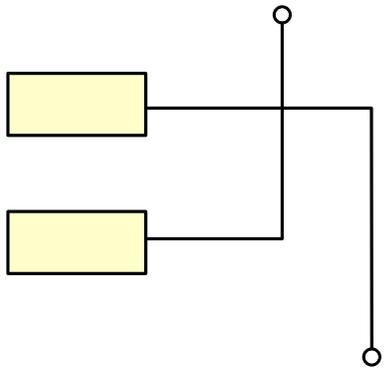
Ges.: Kreuzungsfreie Beschriftung gleicher Länge.

Absteigende und aufsteigende Leader schneiden sich nicht.

Kein Leader schneidet absteigenden und aufsteigenden Leader.

⇒ Keine gemischten Kreuzungen.

⇒ Absteigende und aufsteigende Kreuzungen disjunkt.



Neuverdrahtung

Geg.: Beschriftung minimaler Länge mit Kreuzungen.

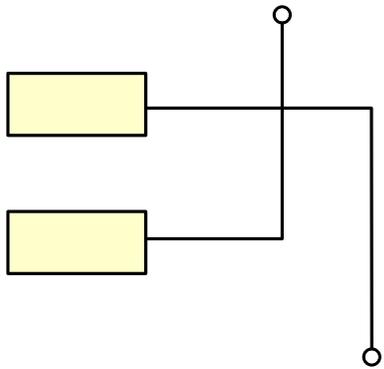
Ges.: Kreuzungsfreie Beschriftung gleicher Länge.

Absteigende und aufsteigende Leader schneiden sich nicht.

Kein Leader schneidet absteigenden und aufsteigenden Leader.

⇒ Keine gemischten Kreuzungen.

⇒ Absteigende und aufsteigende Kreuzungen disjunkt.



Übung!

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen ℓ_1, \dots, ℓ_n aufsteigend

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



Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen ℓ_1, \dots, ℓ_n aufsteigend

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

Sei $\lambda(p, \ell_i)$ Leader.



Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

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

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**



Entfernung aufsteigender Kreuzungen

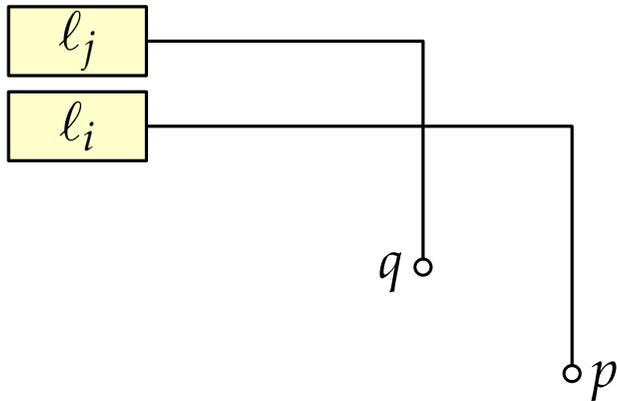
Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

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

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkester Kreuzung mit $\lambda(p, l_i)$.



Entfernung aufsteigender Kreuzungen

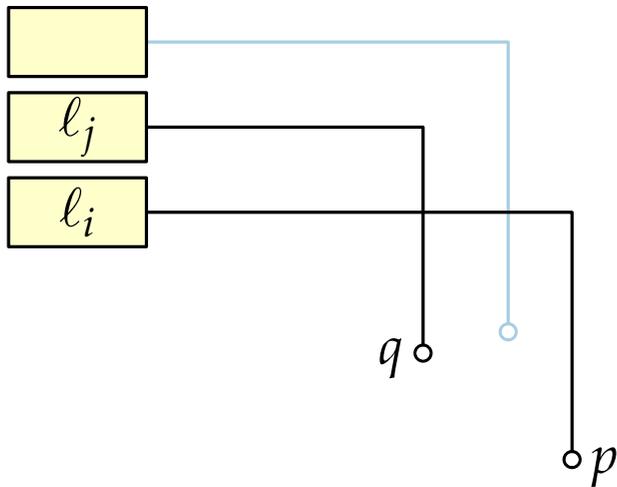
Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

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

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkester Kreuzung mit $\lambda(p, l_i)$.



Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

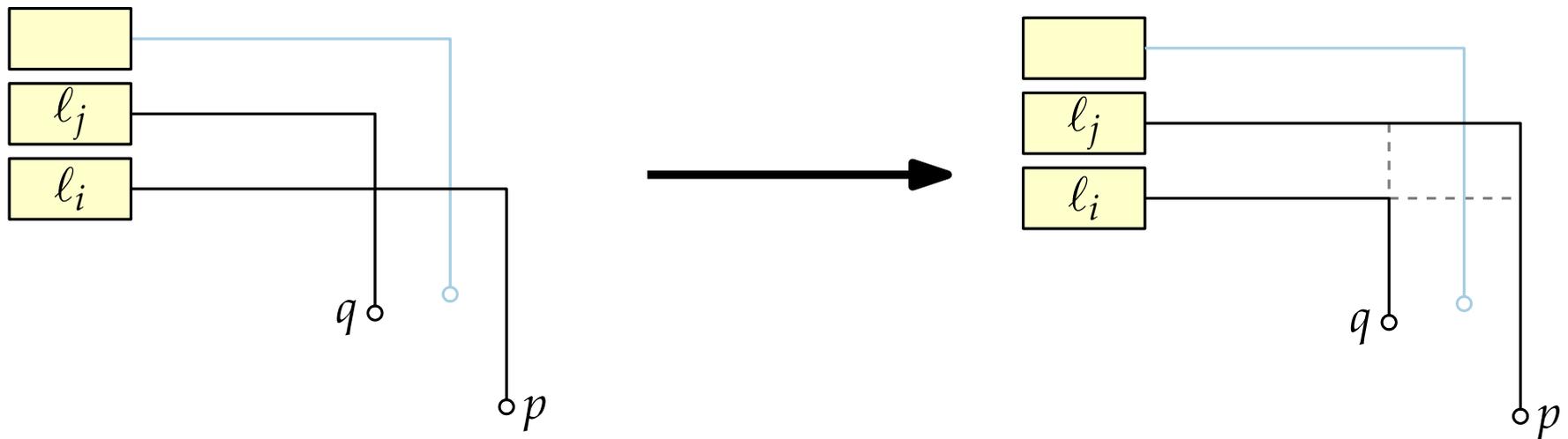
for $i = 1$ **to** n **do**

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkester Kreuzung mit $\lambda(p, l_i)$.

 Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

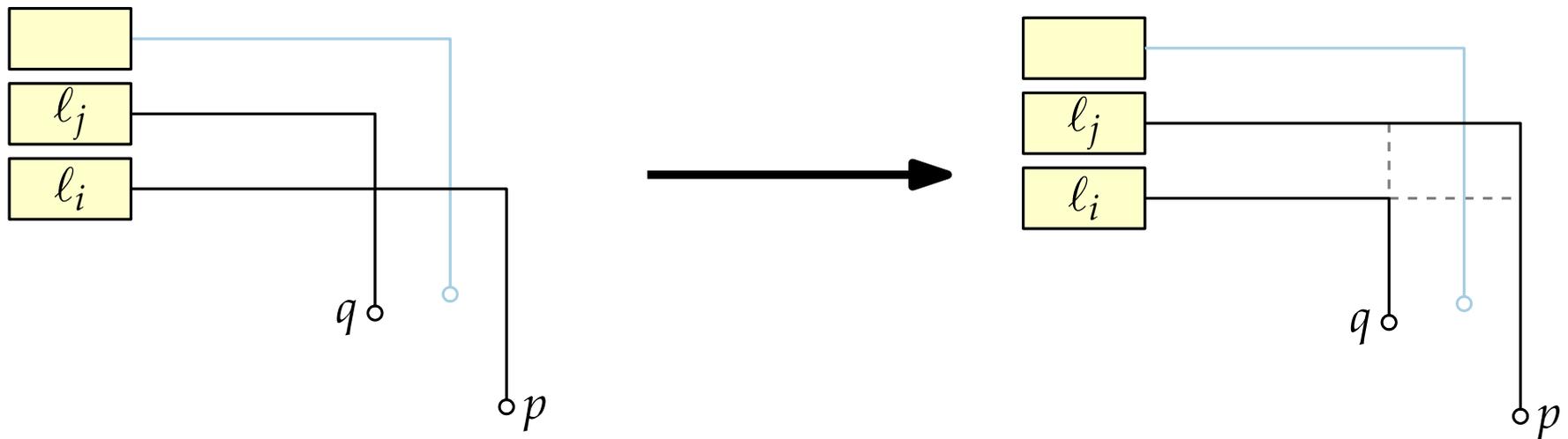
for $i = 1$ **to** n **do**

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

 Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i :

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

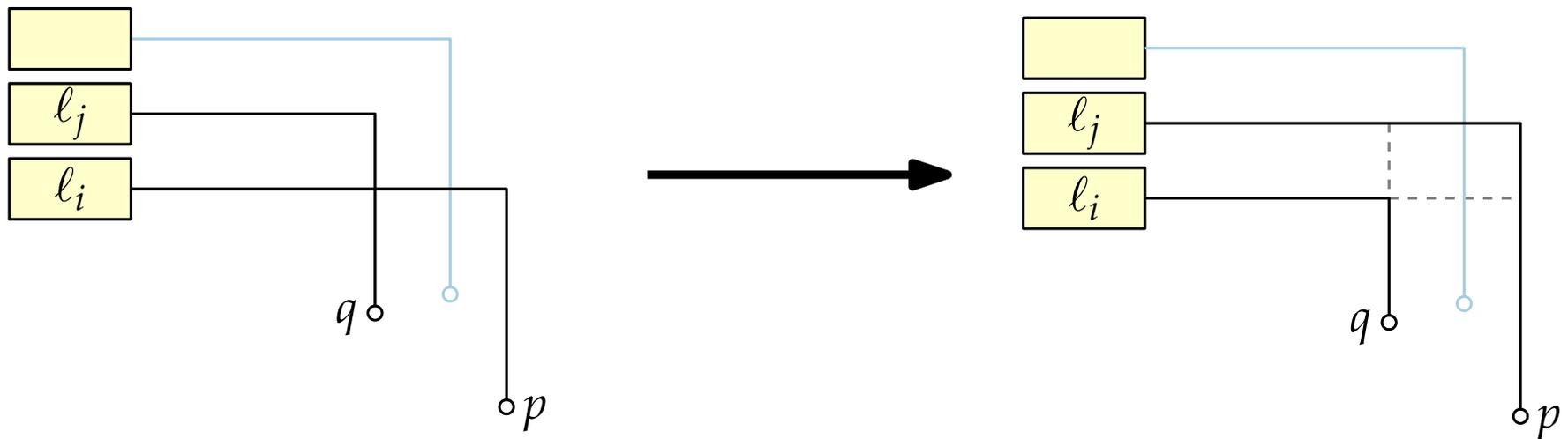
for $i = 1$ **to** n **do**

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

 Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i : 1. Gesamtlänge bleibt gleich

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

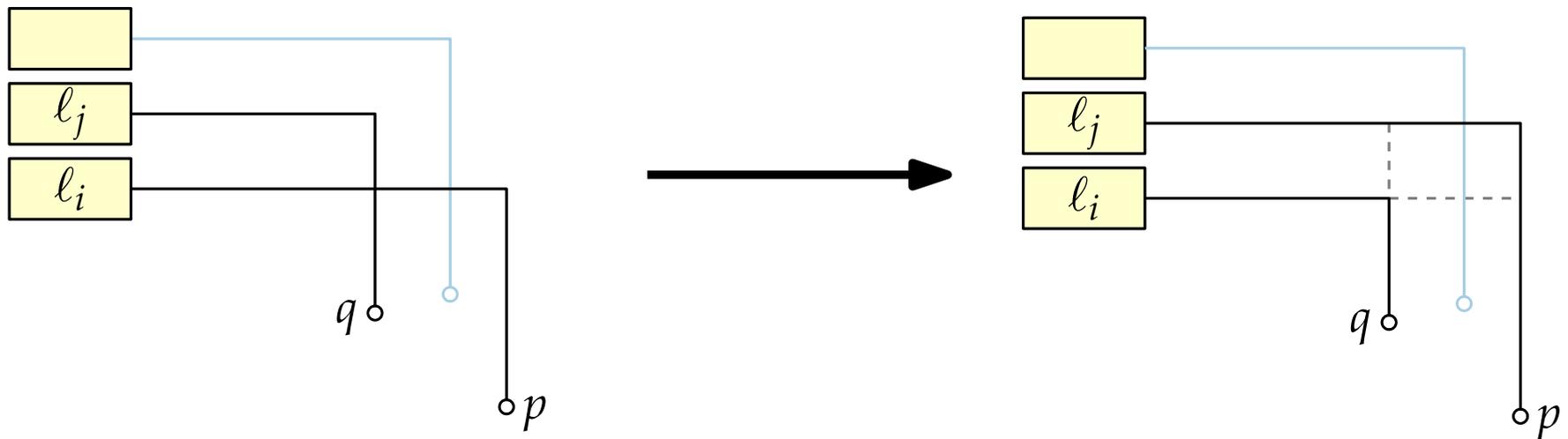
for $i = 1$ **to** n **do**

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

 Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i : 1. Gesamtlänge bleibt gleich

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

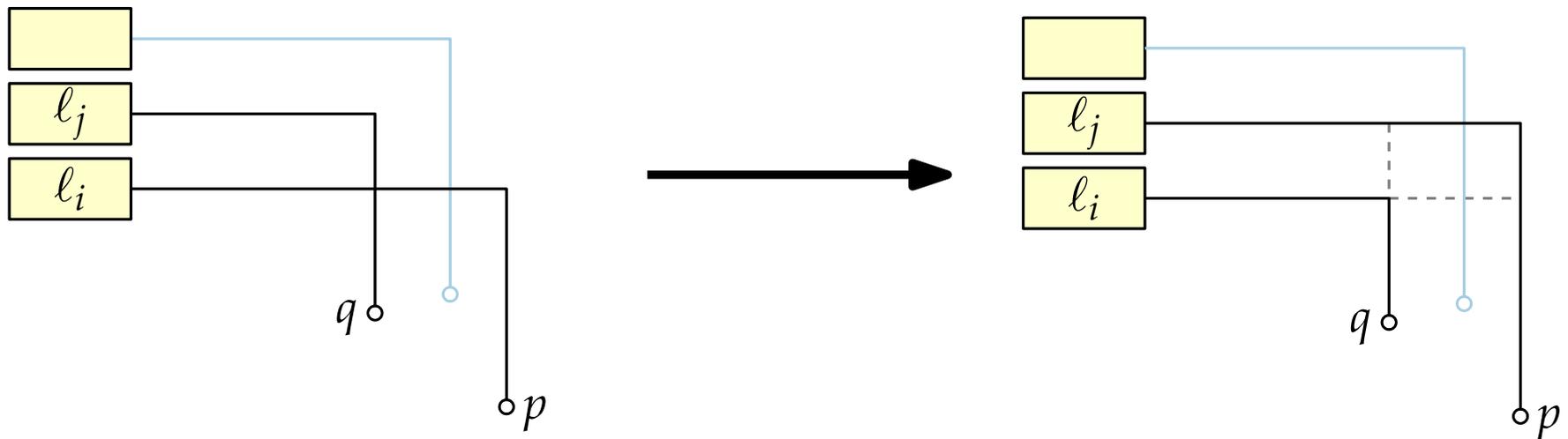
for $i = 1$ **to** n **do**

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

 Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i :
1. Gesamtlänge bleibt gleich
 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$
 3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

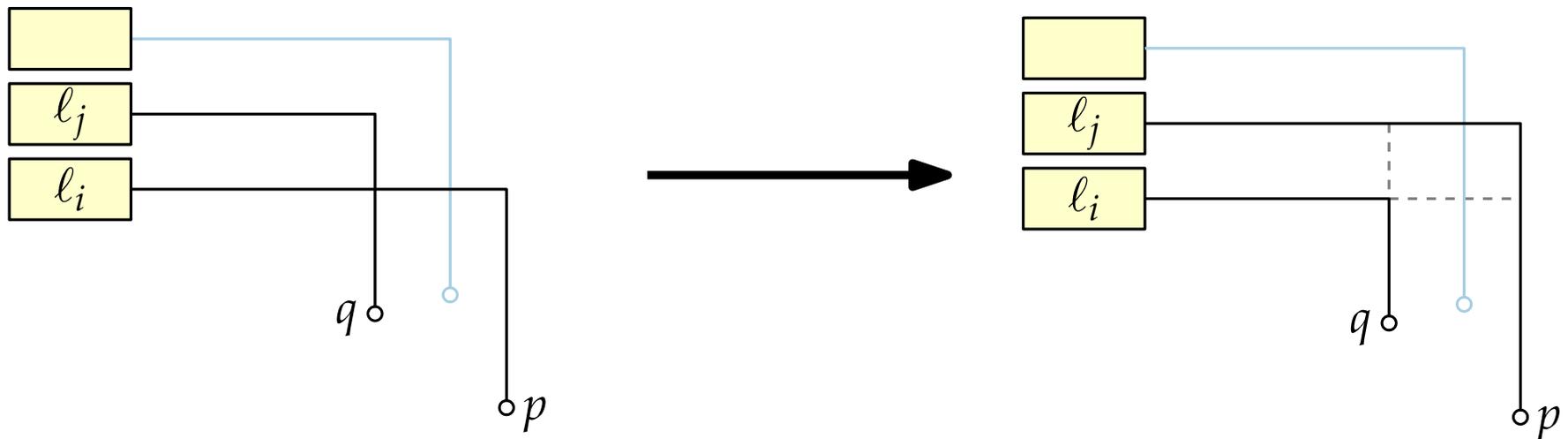
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i :

1. Gesamtlänge bleibt gleich

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

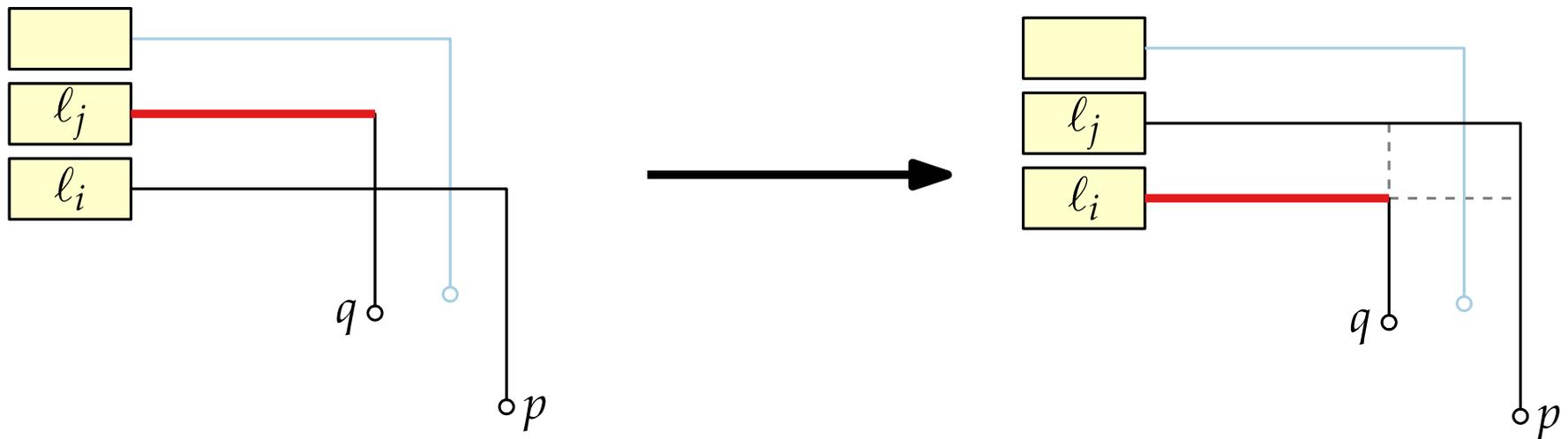
for $i = 1$ **to** n **do**

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkester Kreuzung mit $\lambda(p, l_i)$.

 Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i :

1. Gesamtlänge bleibt gleich

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

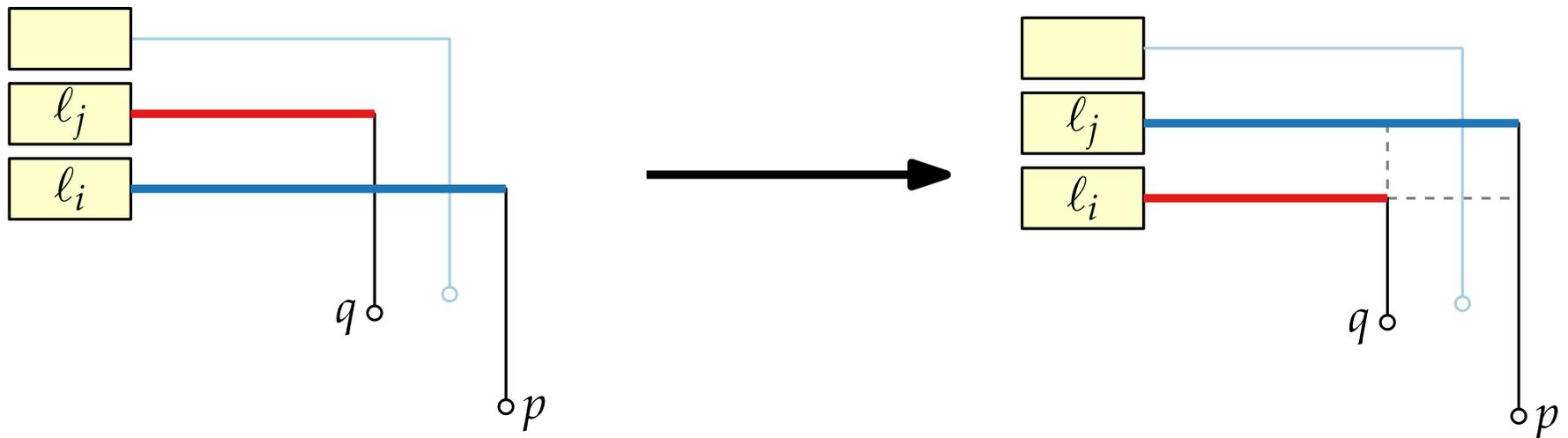
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i :

1. Gesamtlänge bleibt gleich

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

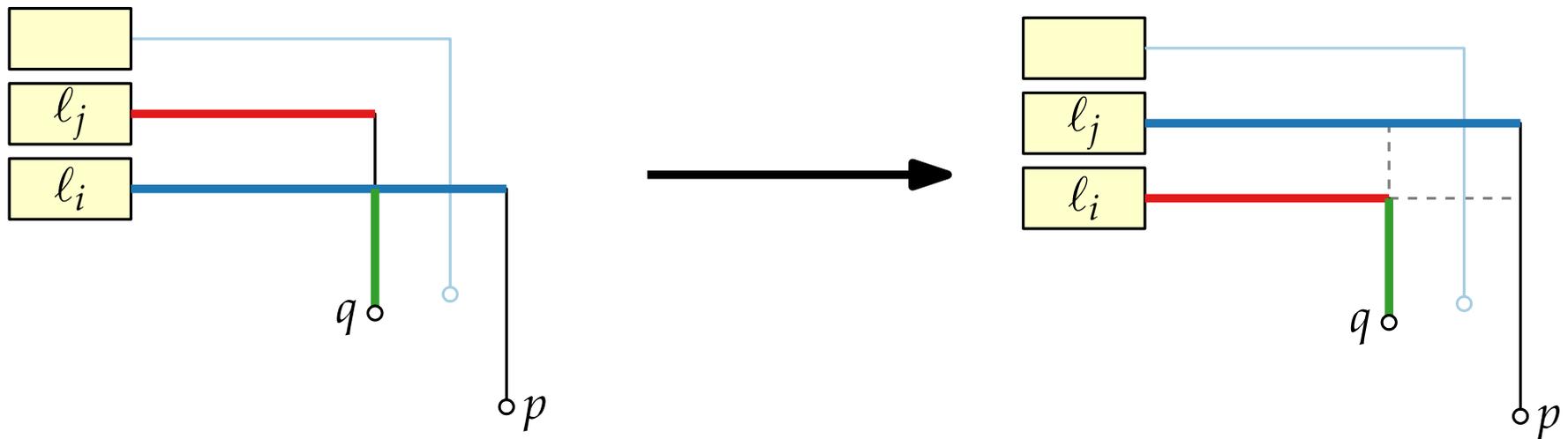
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkester Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i :

1. Gesamtlänge bleibt gleich

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

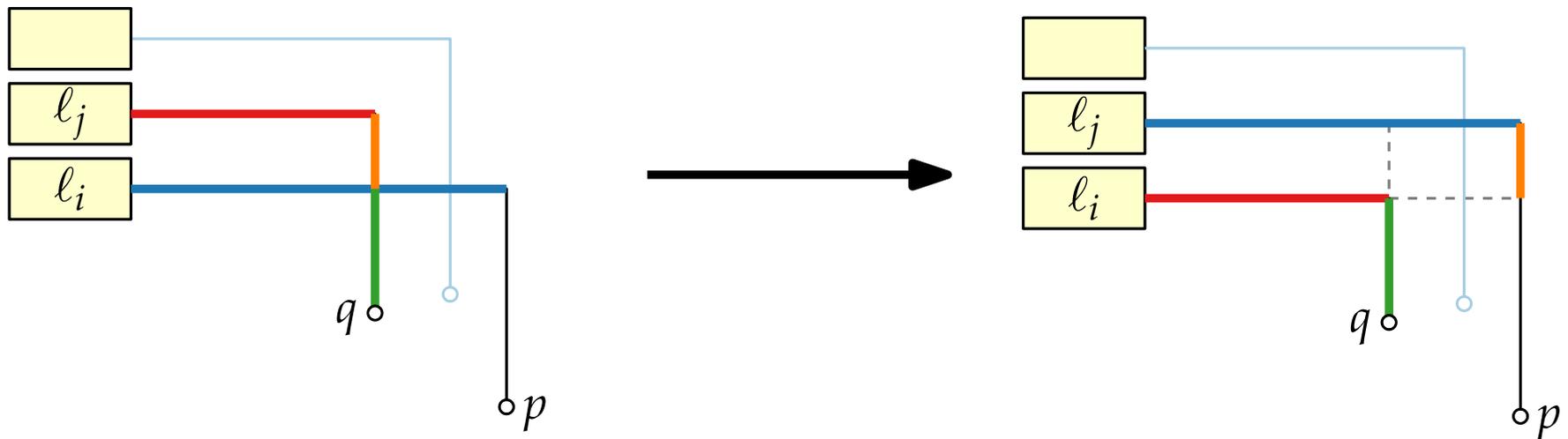
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i :

1. Gesamtlänge bleibt gleich

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

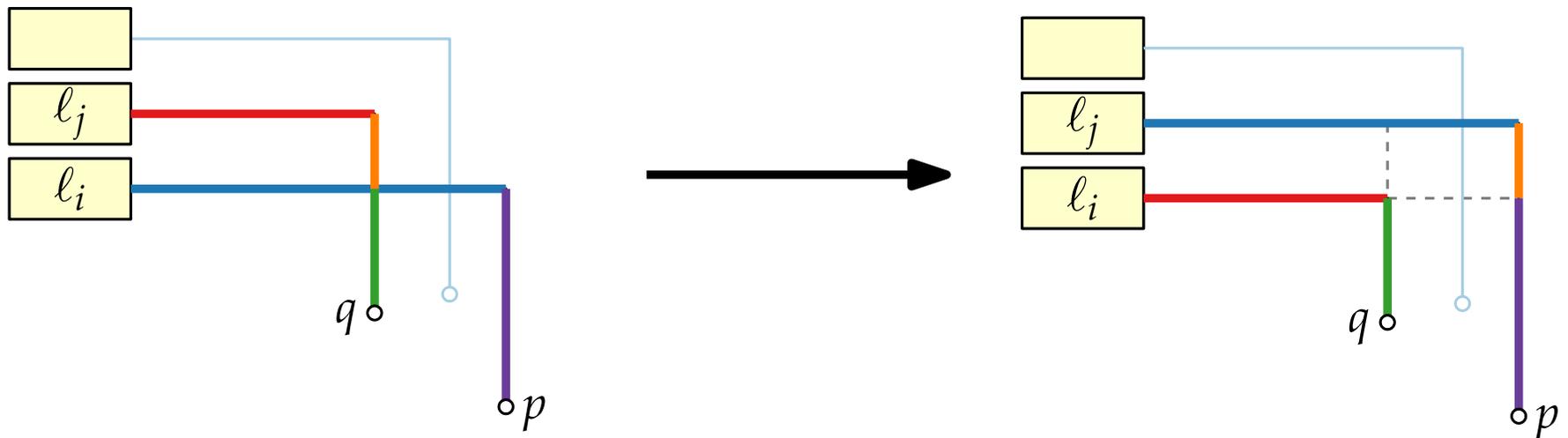
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkester Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i :

1. Gesamtlänge bleibt gleich

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

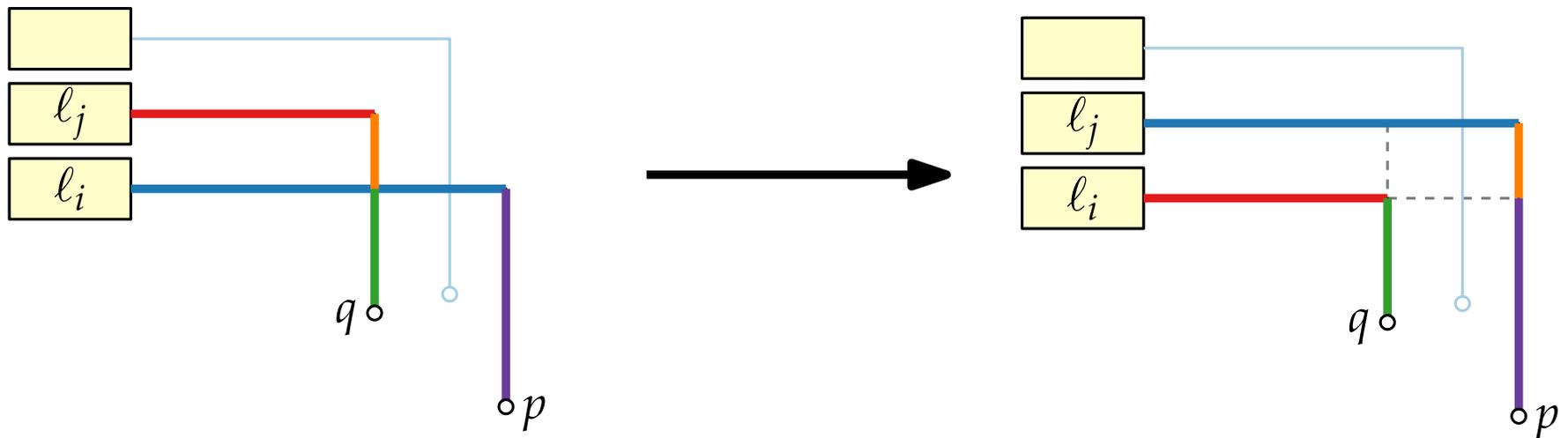
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i :

1. Gesamtlänge bleibt gleich

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen



Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

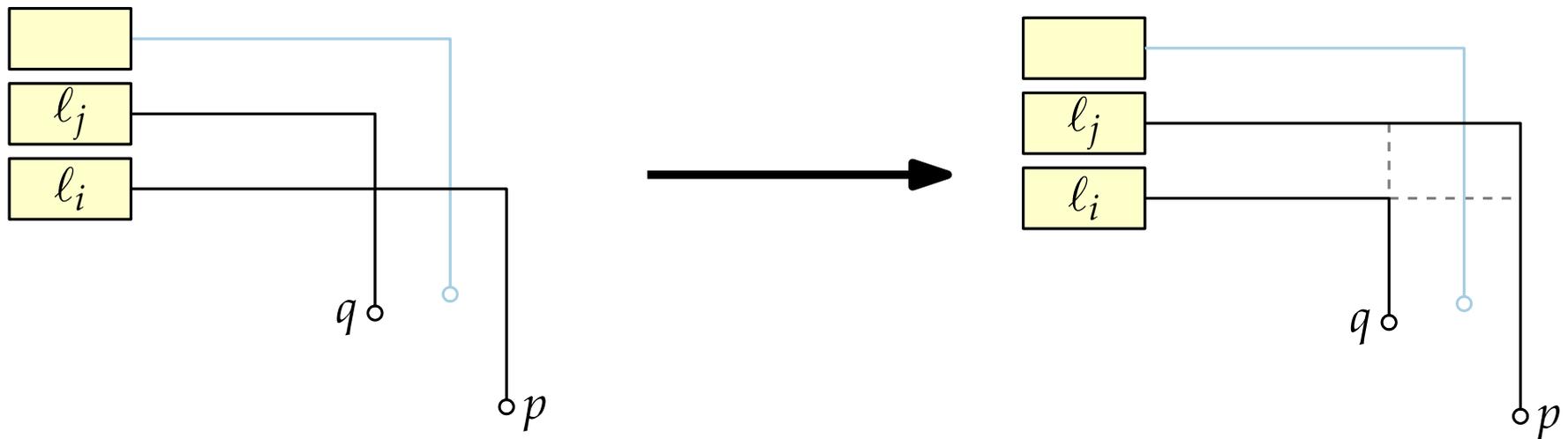
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i : 1. Gesamtlänge bleibt gleich

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen



Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

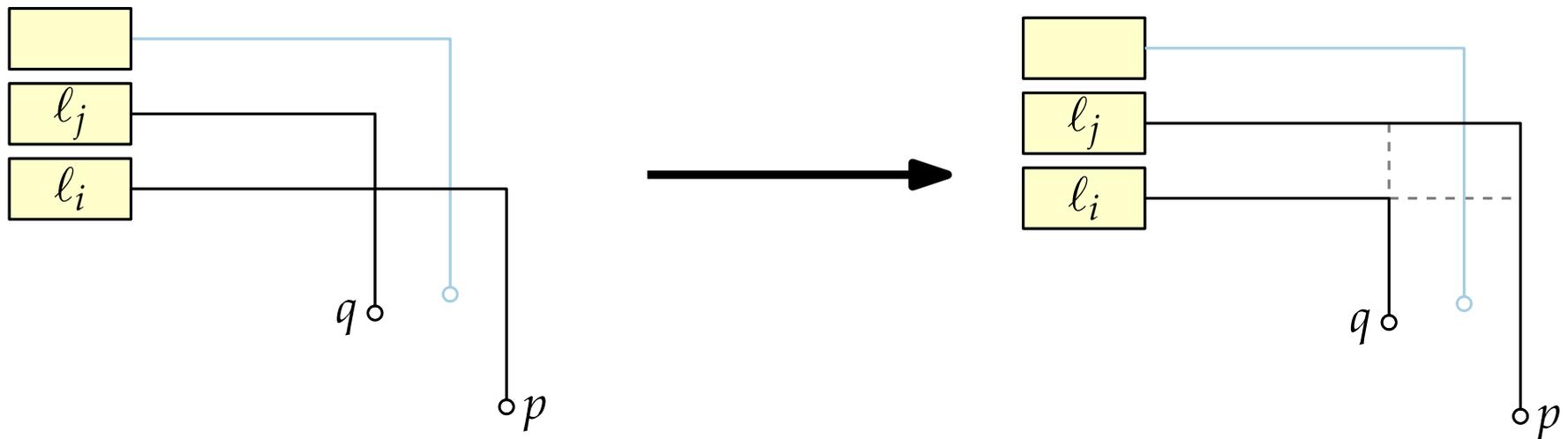
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i : 1. Gesamtlänge bleibt gleich

$j > i$ durch IA

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen



Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

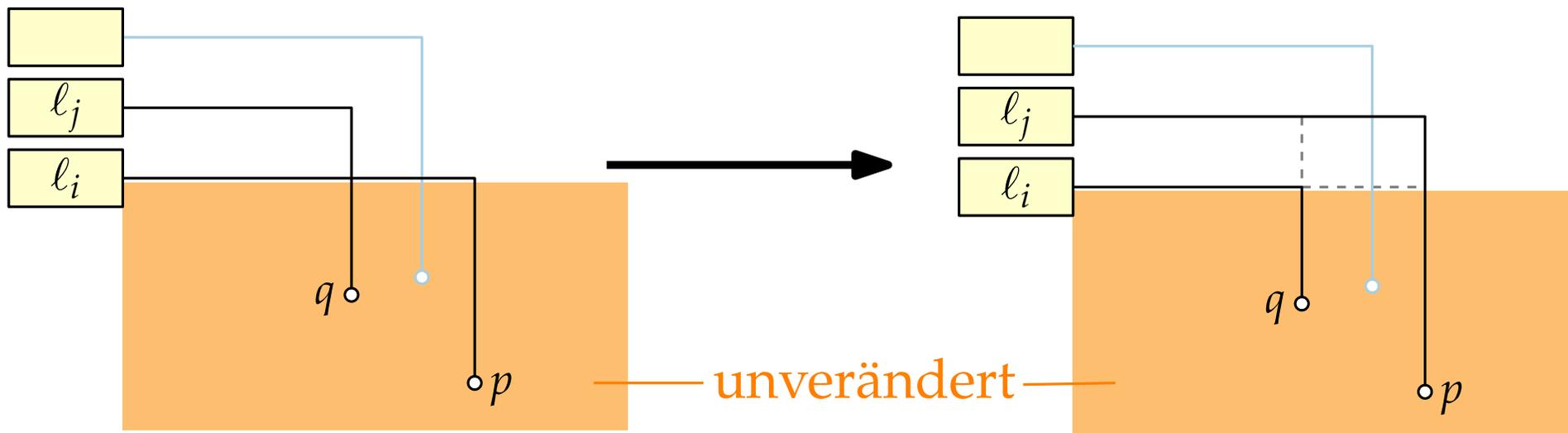
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Induktion über i : 1. Gesamtlänge bleibt gleich

$j > i$ durch IA

2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$

3. Keine absteigenden Kreuzungen



Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

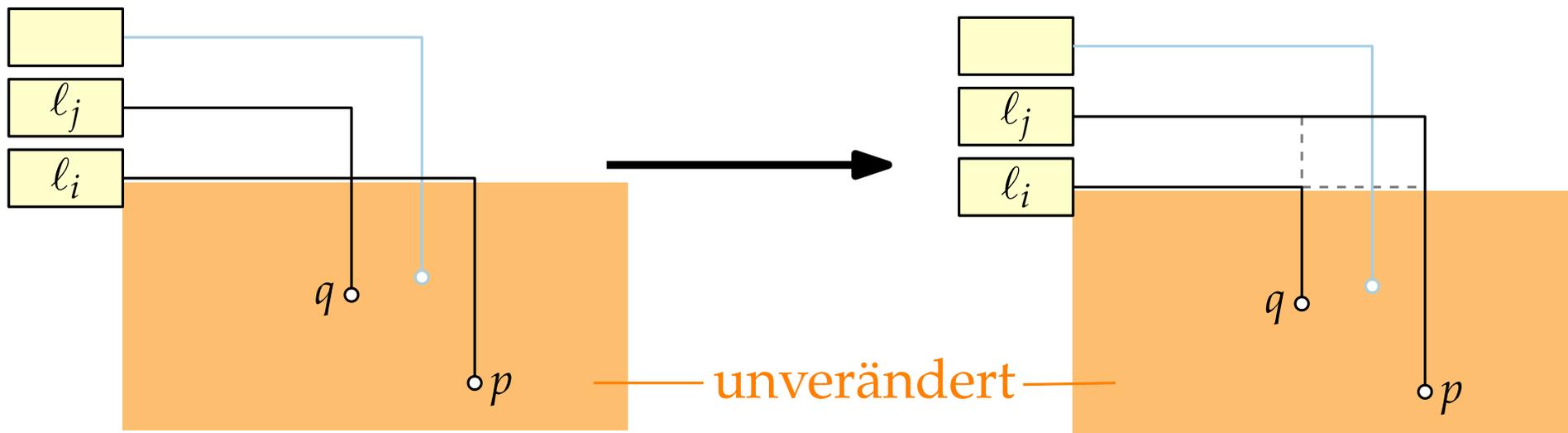
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i : 1. Gesamtlänge bleibt gleich ✓
- $j > i$ durch IA 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$ ✓
3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

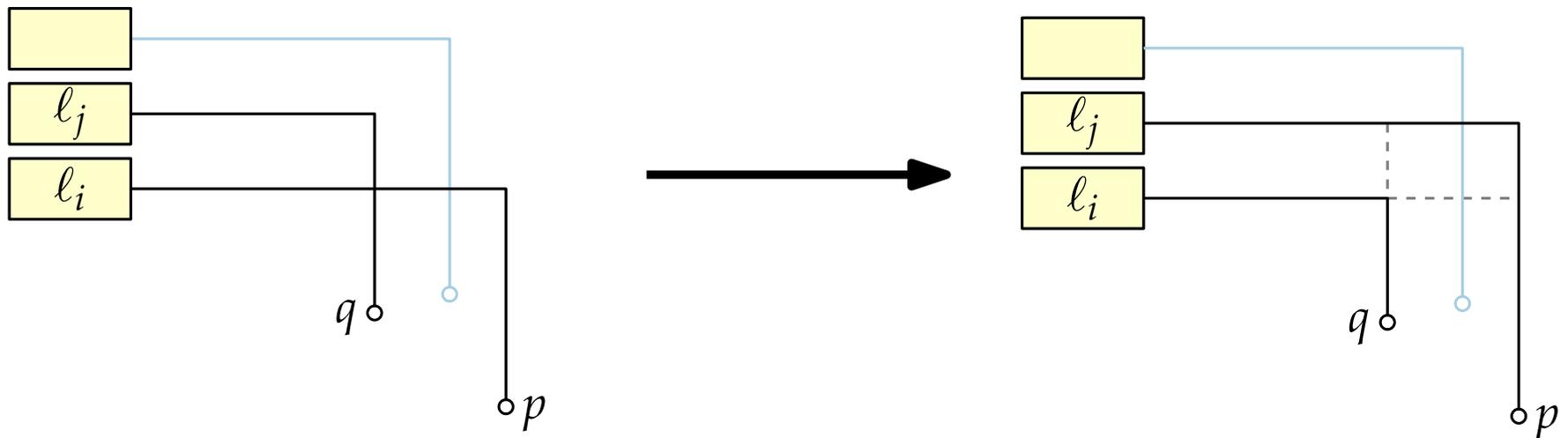
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i : 1. Gesamtlänge bleibt gleich ✓
 $j > i$ durch IA 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$ ✓
3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

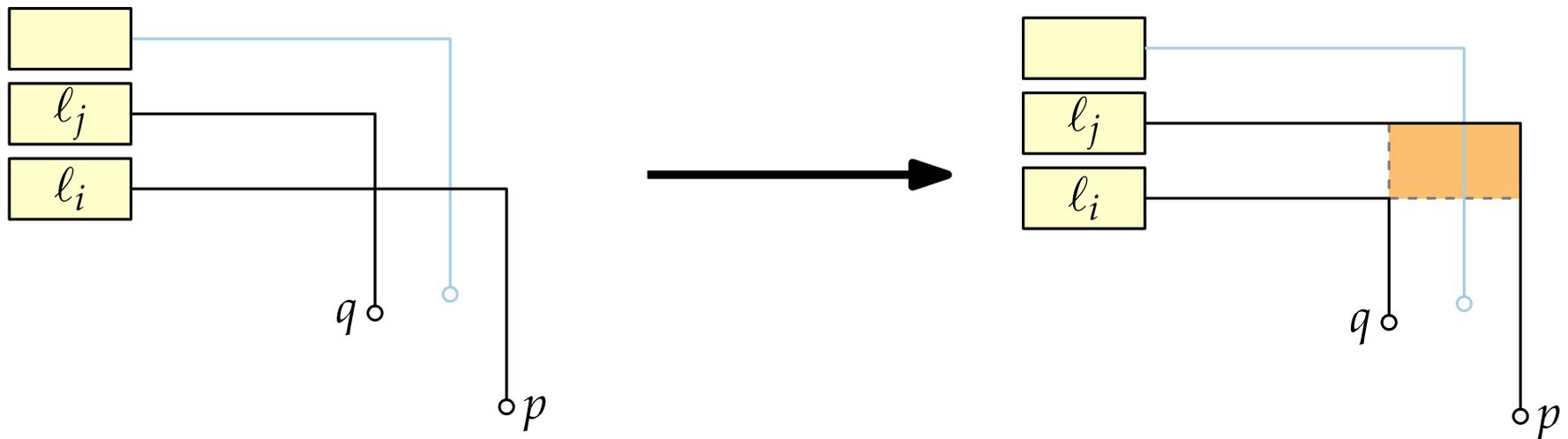
for $i = 1$ **to** n **do**

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

 Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i : $j > i$ durch IA
1. Gesamtlänge bleibt gleich ✓
 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$ ✓
 3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

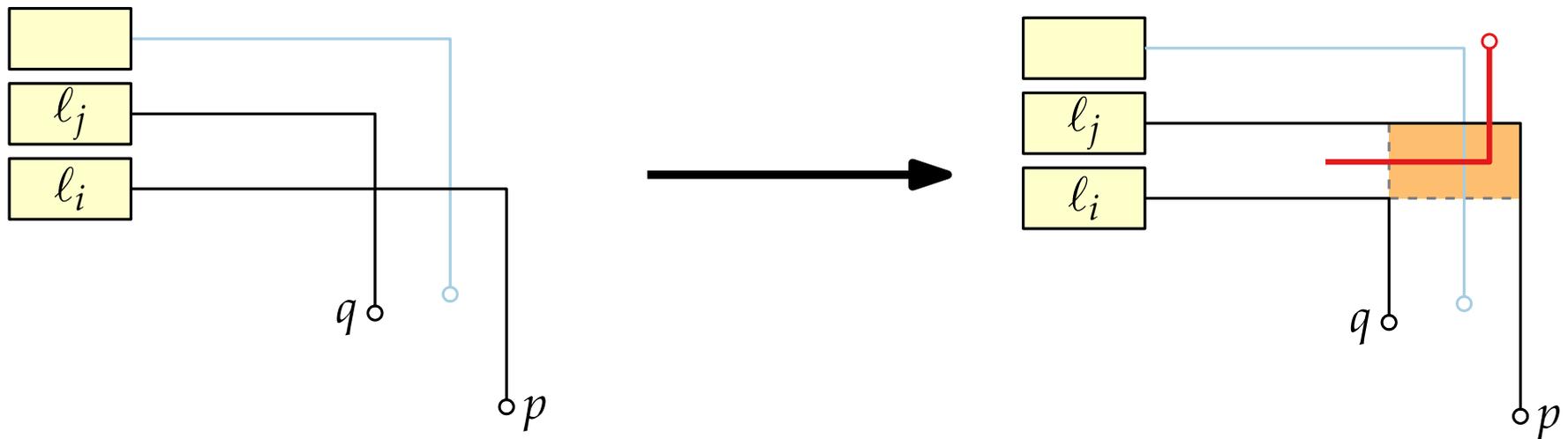
for $i = 1$ **to** n **do**

 Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

 Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

 Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i : $j > i$ durch IA
1. Gesamtlänge bleibt gleich ✓
 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$ ✓
 3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

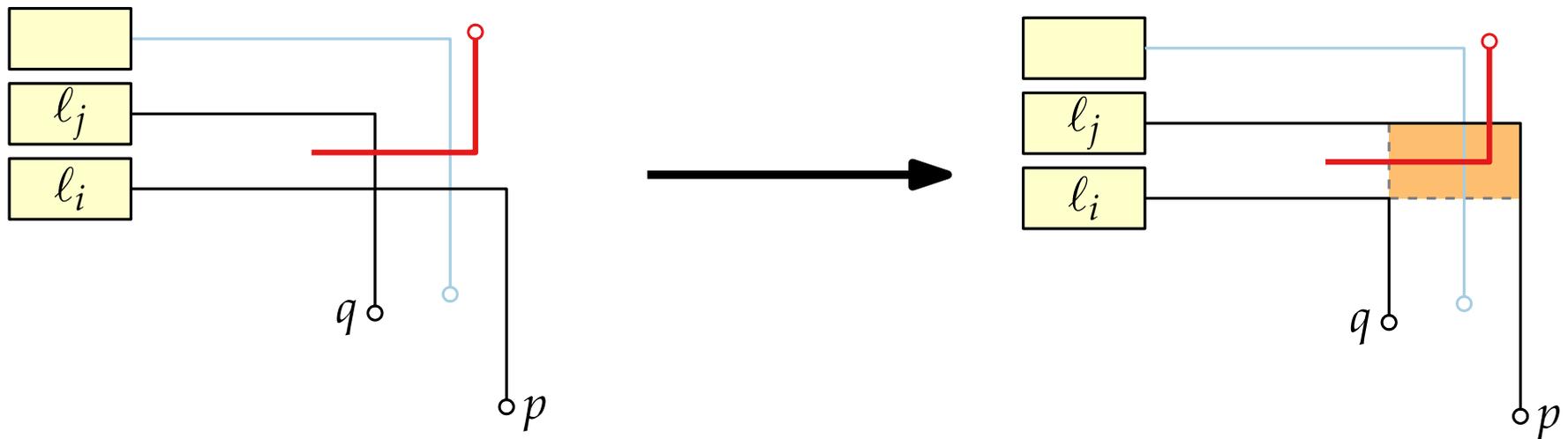
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i : $j > i$ durch IA
1. Gesamtlänge bleibt gleich ✓
 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$ ✓
 3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

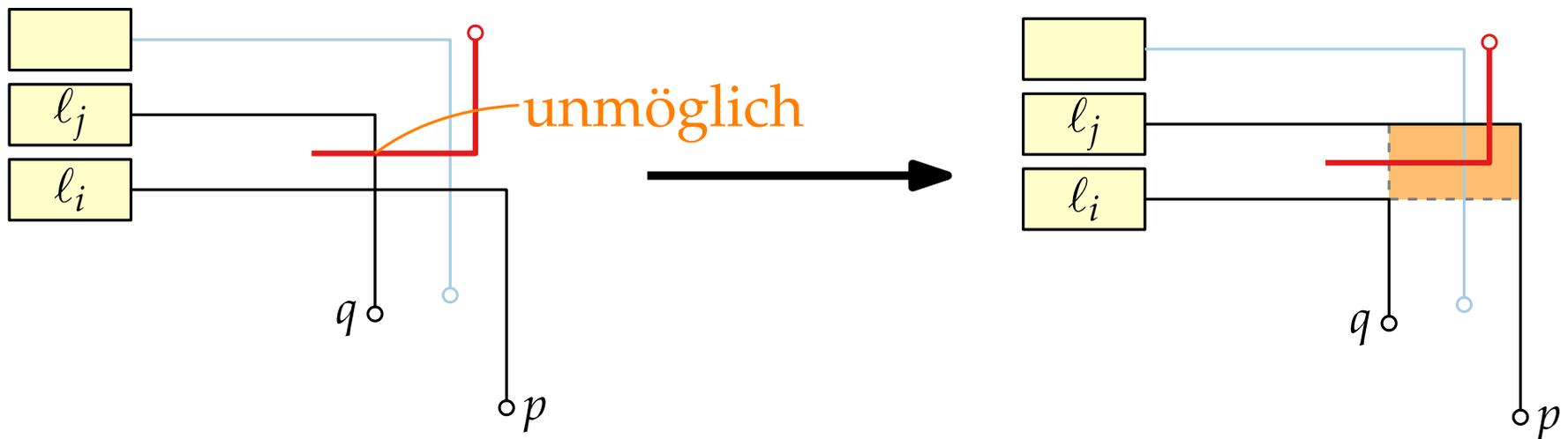
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i : $j > i$ durch IA
1. Gesamtlänge bleibt gleich ✓
 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$ ✓
 3. Keine absteigenden Kreuzungen

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

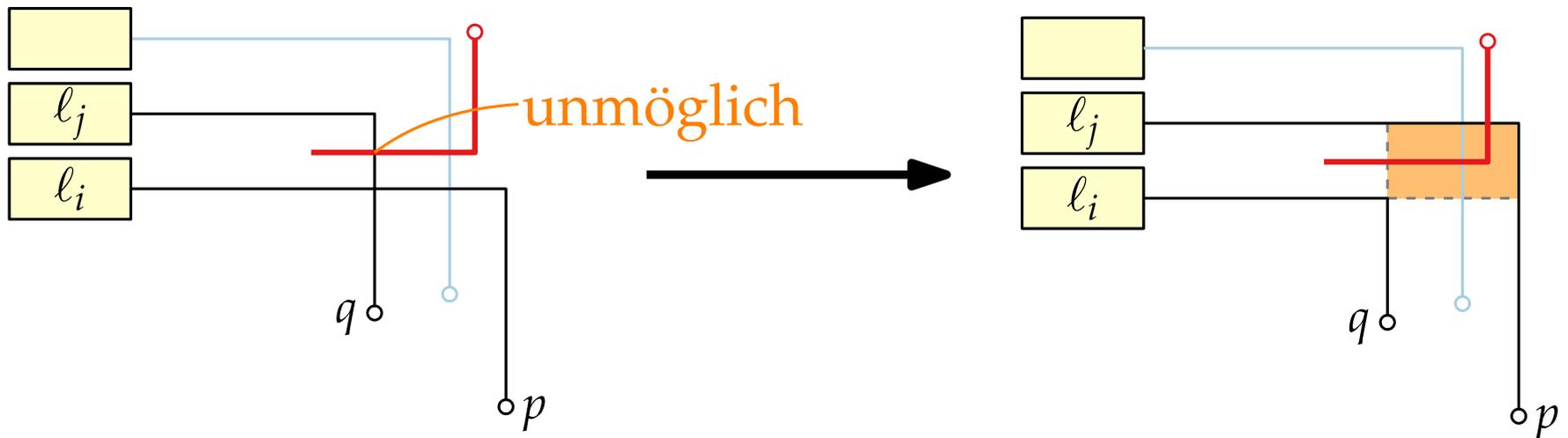
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i : $j > i$ durch IA
1. Gesamtlänge bleibt gleich ✓
 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$ ✓
 3. Keine absteigenden Kreuzungen ✓

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

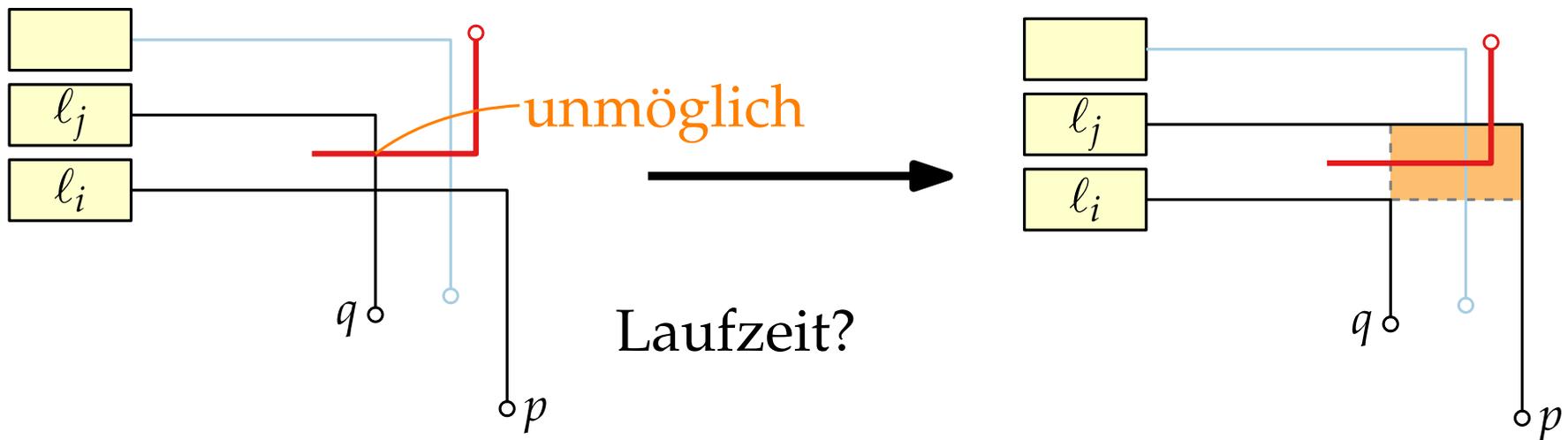
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i : $j > i$ durch IA
1. Gesamtlänge bleibt gleich ✓
 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$ ✓
 3. Keine absteigenden Kreuzungen ✓

Entfernung aufsteigender Kreuzungen

Sortiere Beschriftungen l_1, \dots, l_n aufsteigend

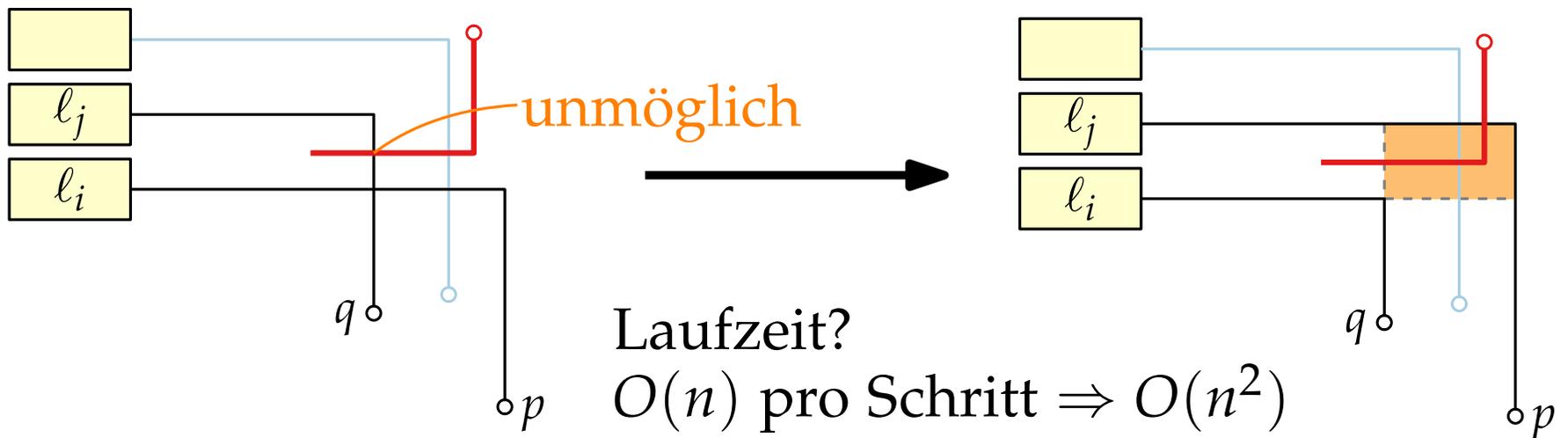
for $i = 1$ **to** n **do**

Sei $\lambda(p, l_i)$ Leader.

if $\lambda(p, l_i)$ aufsteigend **then**

Sei leader $\lambda(q, l_j)$ mit linkerster Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



- Induktion über i : $j > i$ durch IA
1. Gesamtlänge bleibt gleich ✓
 2. Alle Kreuzungen nur oberhalb $\lambda(q, l_i)$ ✓
 3. Keine absteigenden Kreuzungen ✓

Längenminimierung

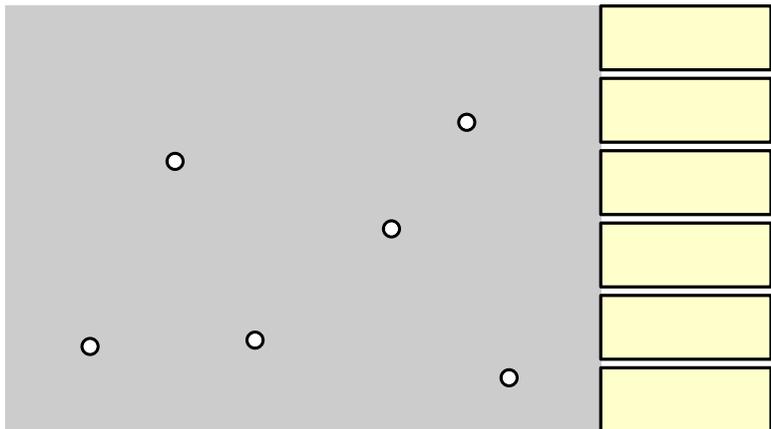
Lemma. Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.

Längenminimierung

- Lemma.** Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.
- Lemma.** Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.

Längenminimierung

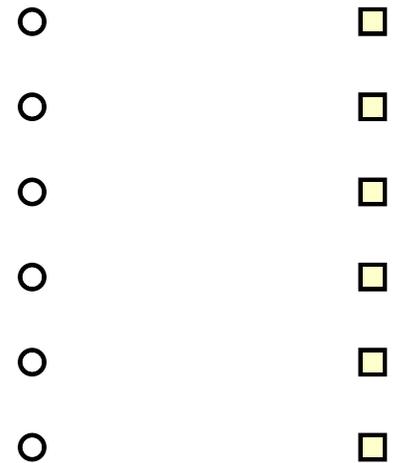
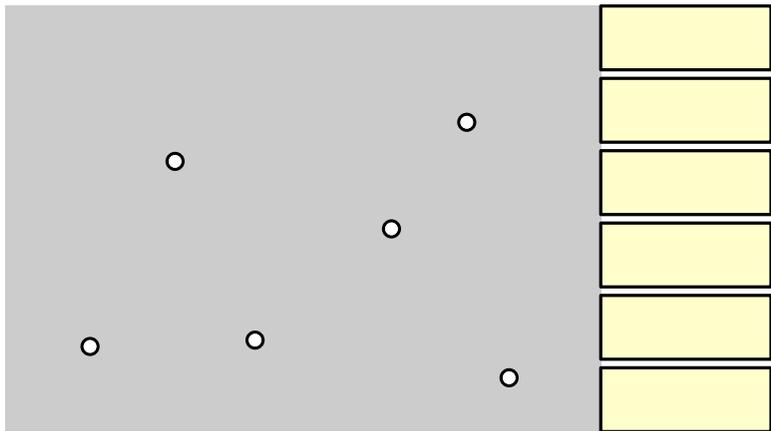
- Lemma.** Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.
- Lemma.** Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



Längenminimierung

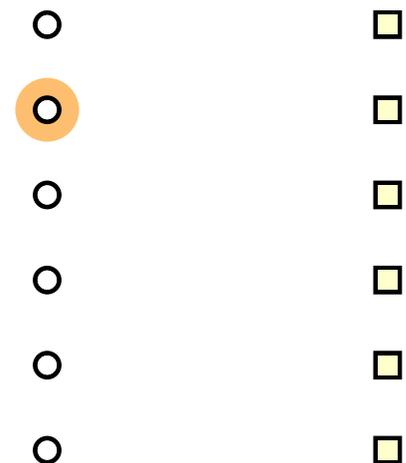
Lemma. Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.

Lemma. Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



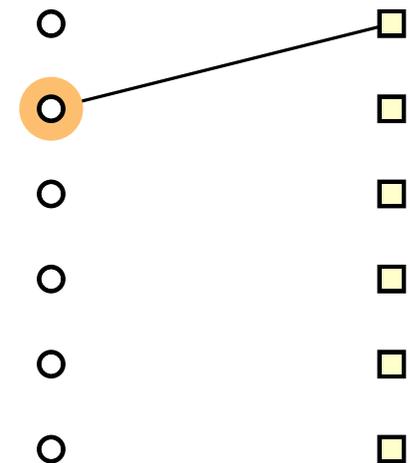
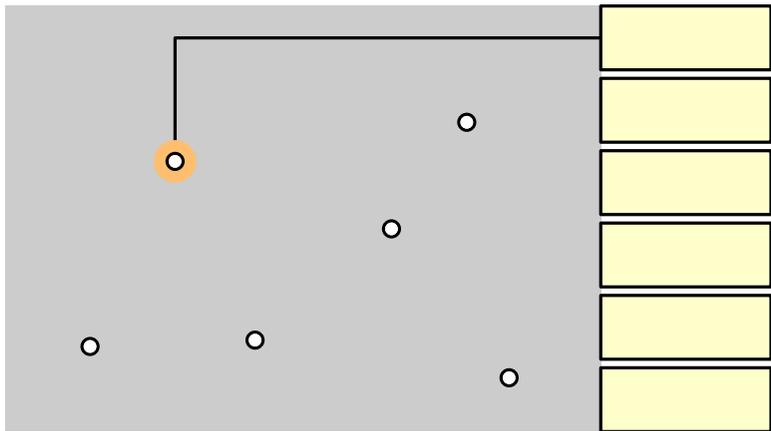
Längenminimierung

- Lemma.** Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.
- Lemma.** Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



Längenminimierung

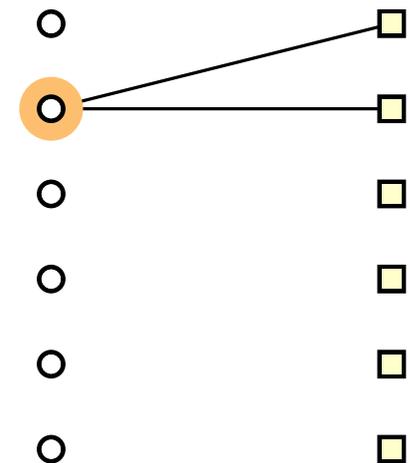
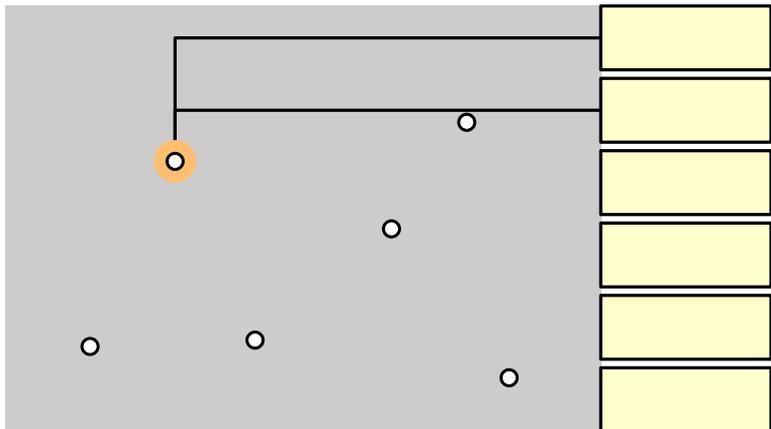
- Lemma.** Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.
- Lemma.** Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



Längenminimierung

Lemma. Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.

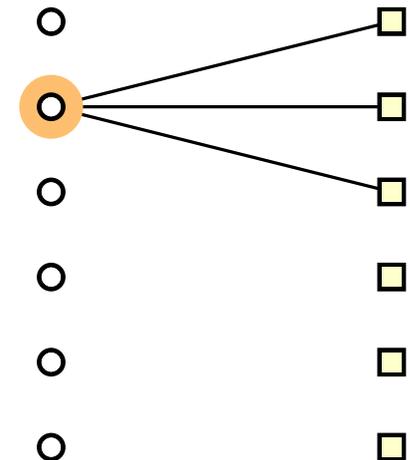
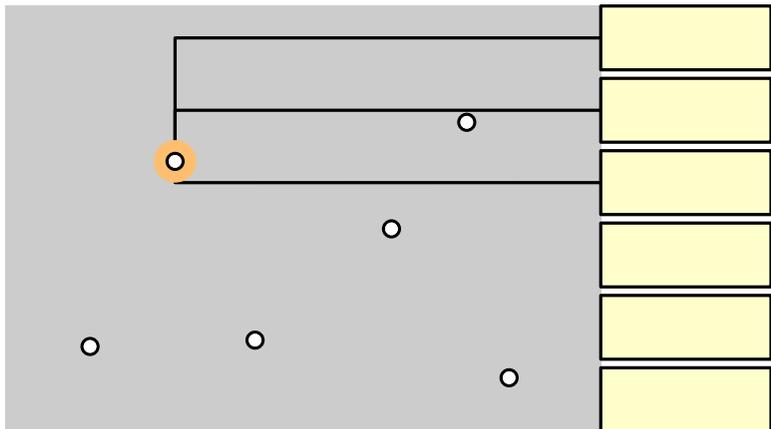
Lemma. Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



Längenminimierung

Lemma. Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.

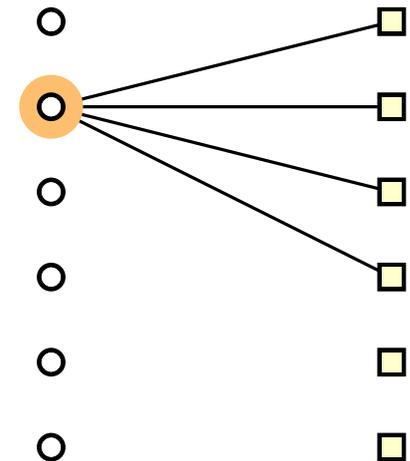
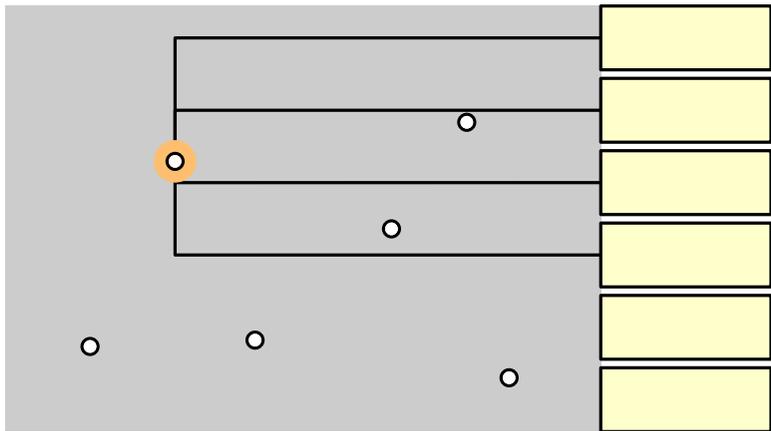
Lemma. Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



Längenminimierung

Lemma. Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.

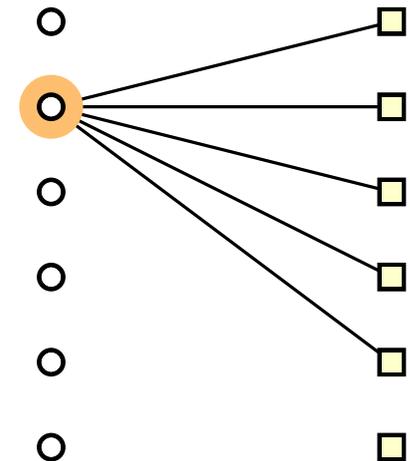
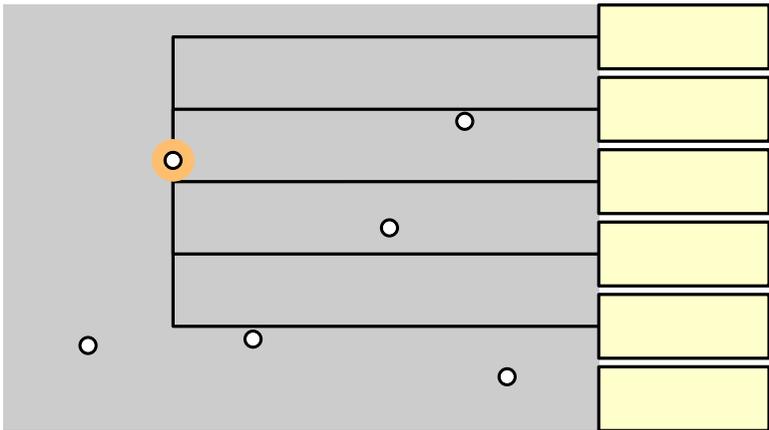
Lemma. Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



Längenminimierung

Lemma. Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.

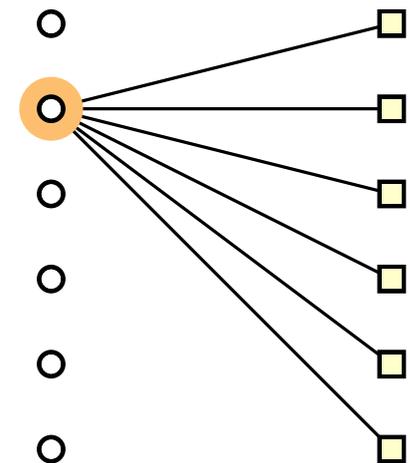
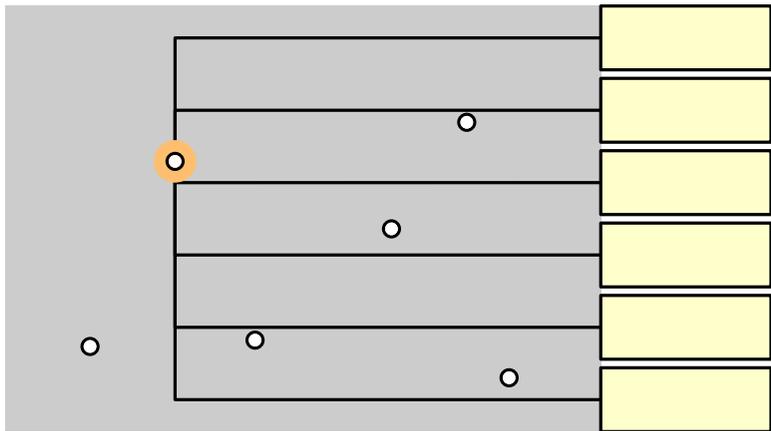
Lemma. Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



Längenminimierung

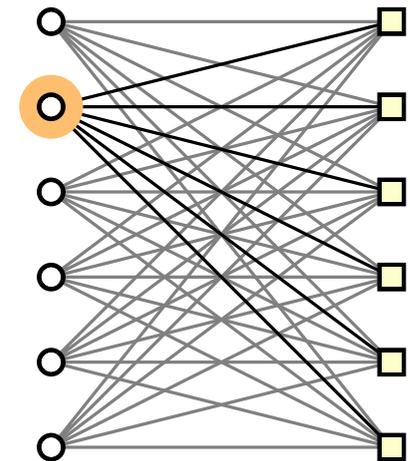
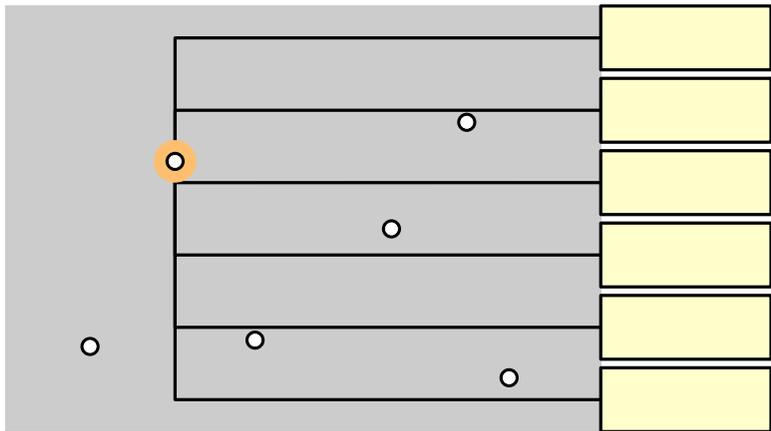
Lemma. Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.

Lemma. Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



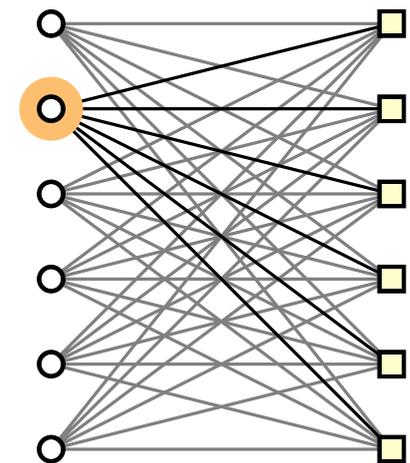
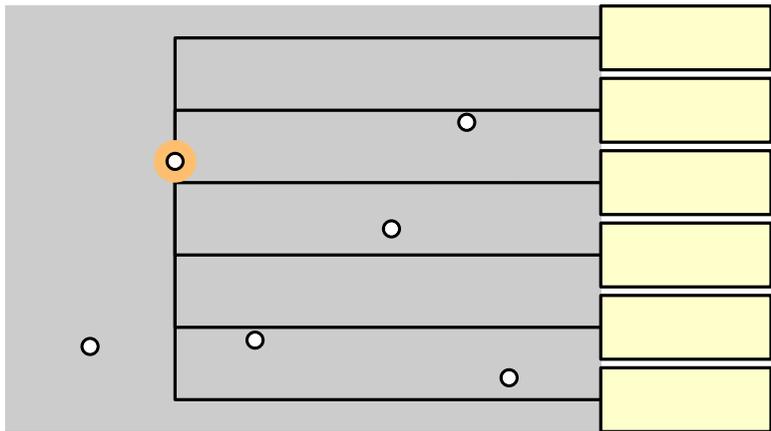
Längenminimierung

- Lemma.** Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.
- Lemma.** Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



Längenminimierung

- Lemma.** Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.
- Lemma.** Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



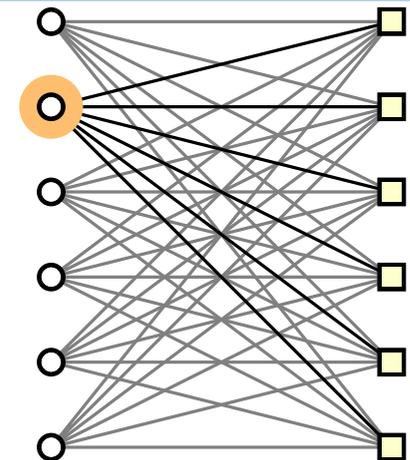
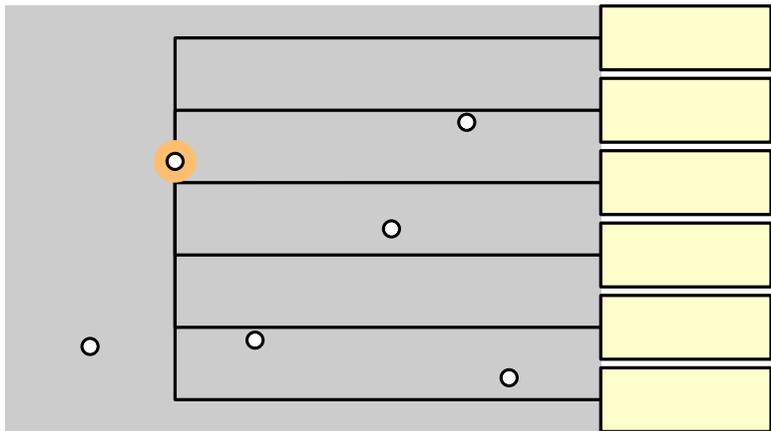
Minimum-weight bipartite matching!

Längenminimierung

Lemma. Für jede Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann eine kreuzungsfreie Beschriftung der gleichen Länge in $O(n^2)$ Zeit gefunden werden.

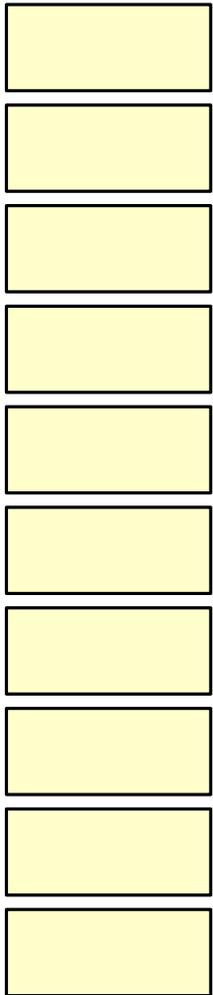
Lemma. Eine Beschriftung minimaler Länge mit Kreuzungen im 1-seitigen po-Leader-Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.

Satz. Eine kreuzungsfreie Beschriftung minimaler Länge im 1-seitigen po-Leader Model mit fixen Ports kann in $O(n^3)$ Zeit gefunden werden.



Minimum-weight bipartite matching!

Sweepline-Verfahren



○

○

○

○

○

○

○

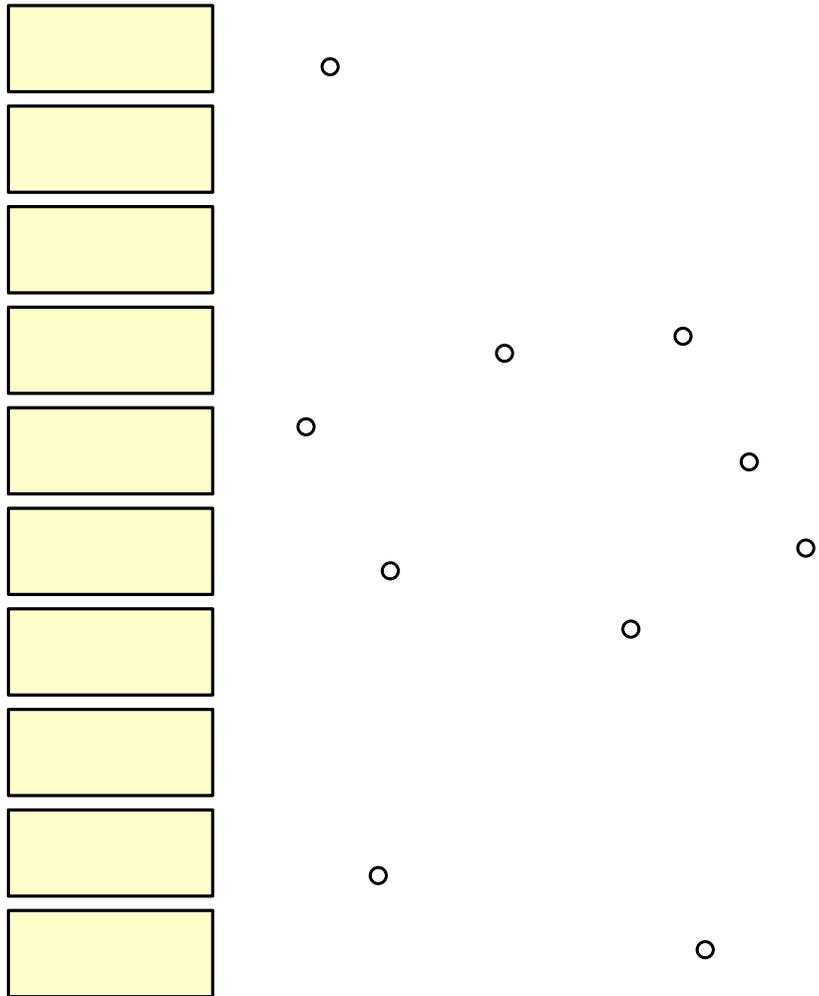
○

○

○

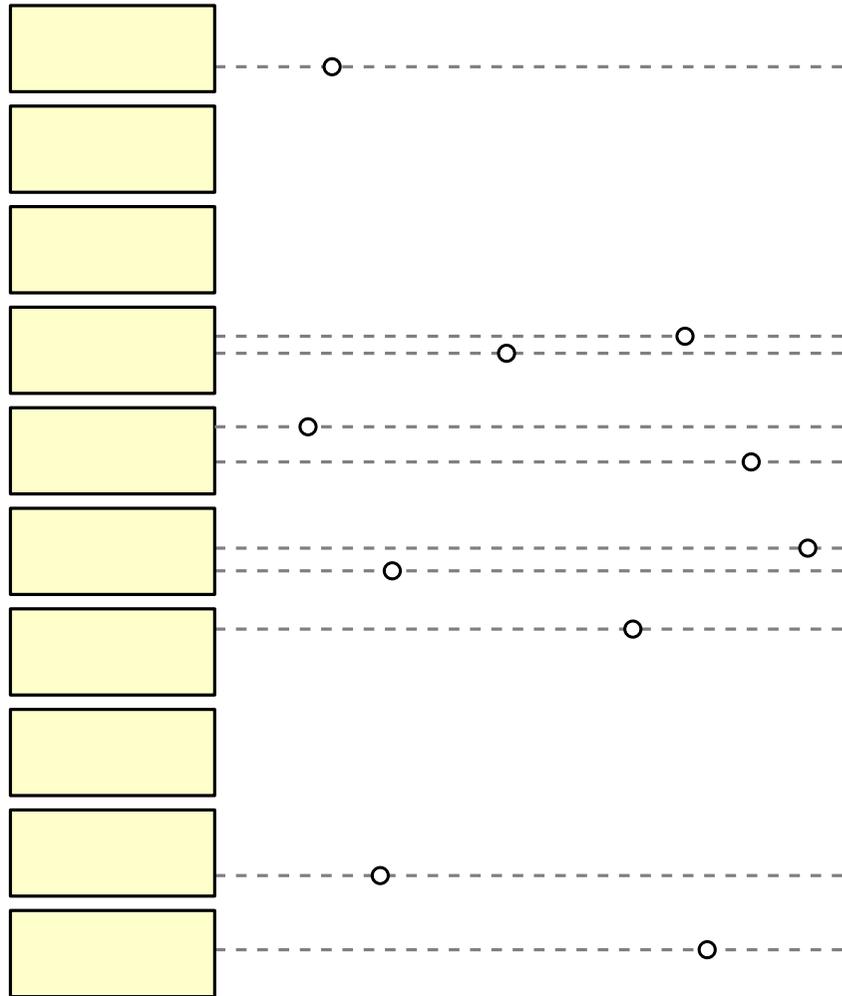
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte



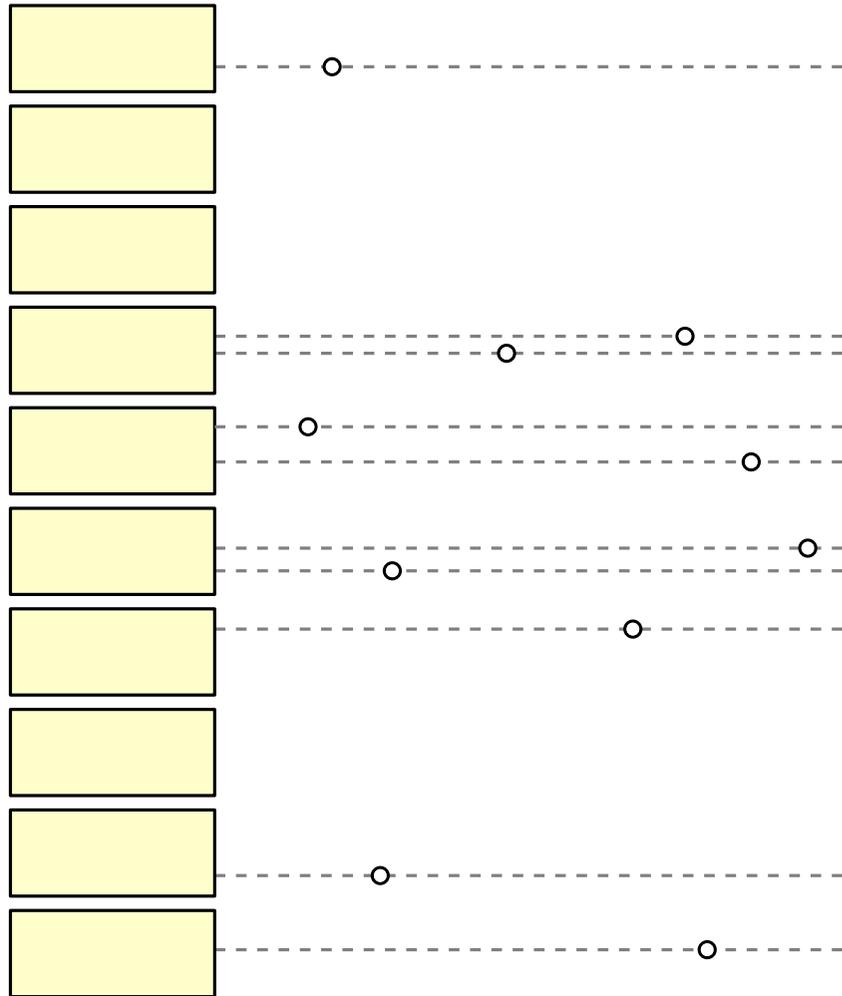
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte



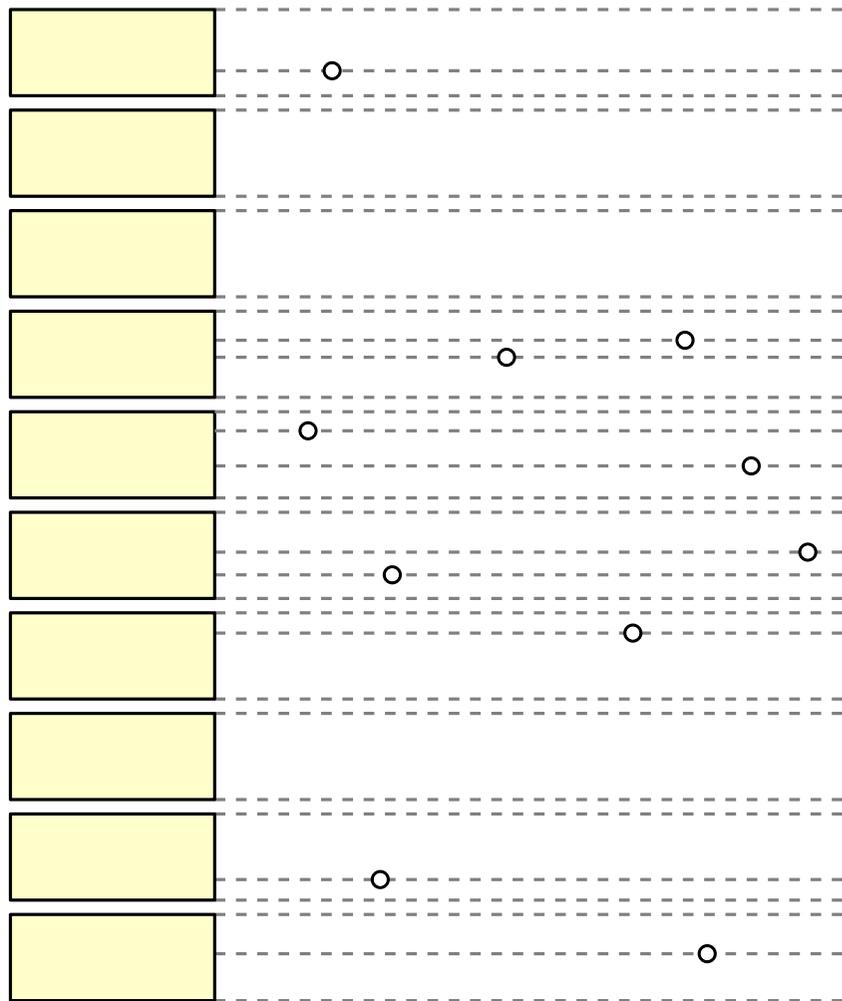
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.



Sweepline-Verfahren

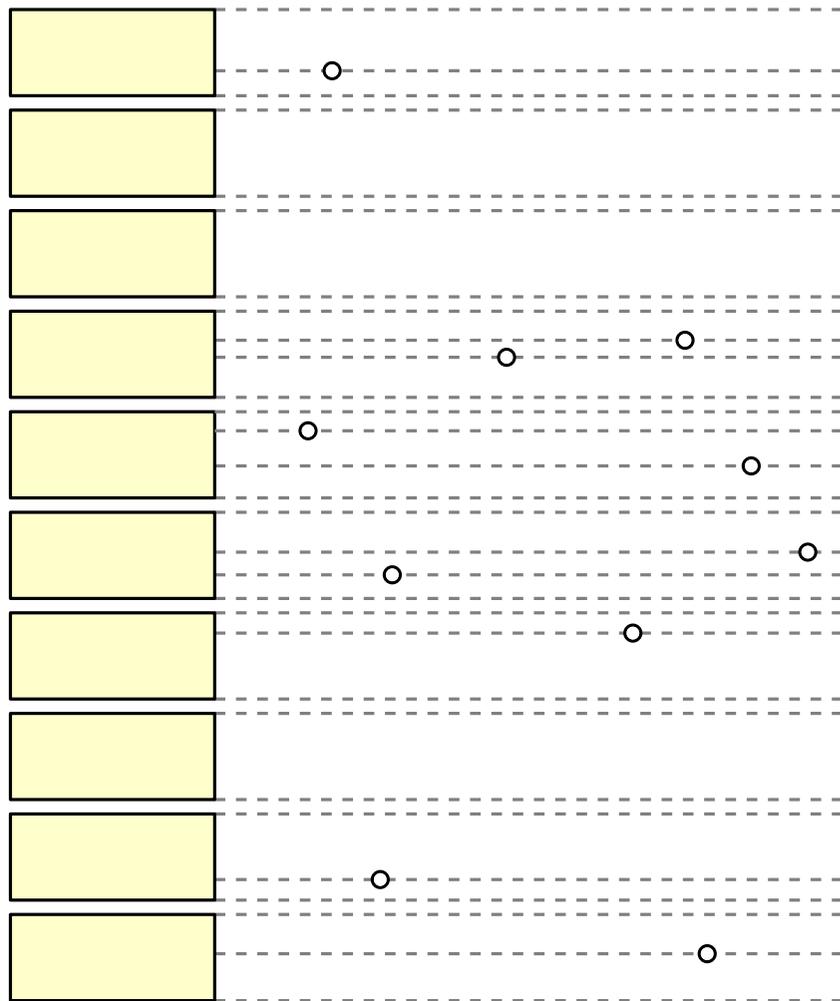
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.



Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

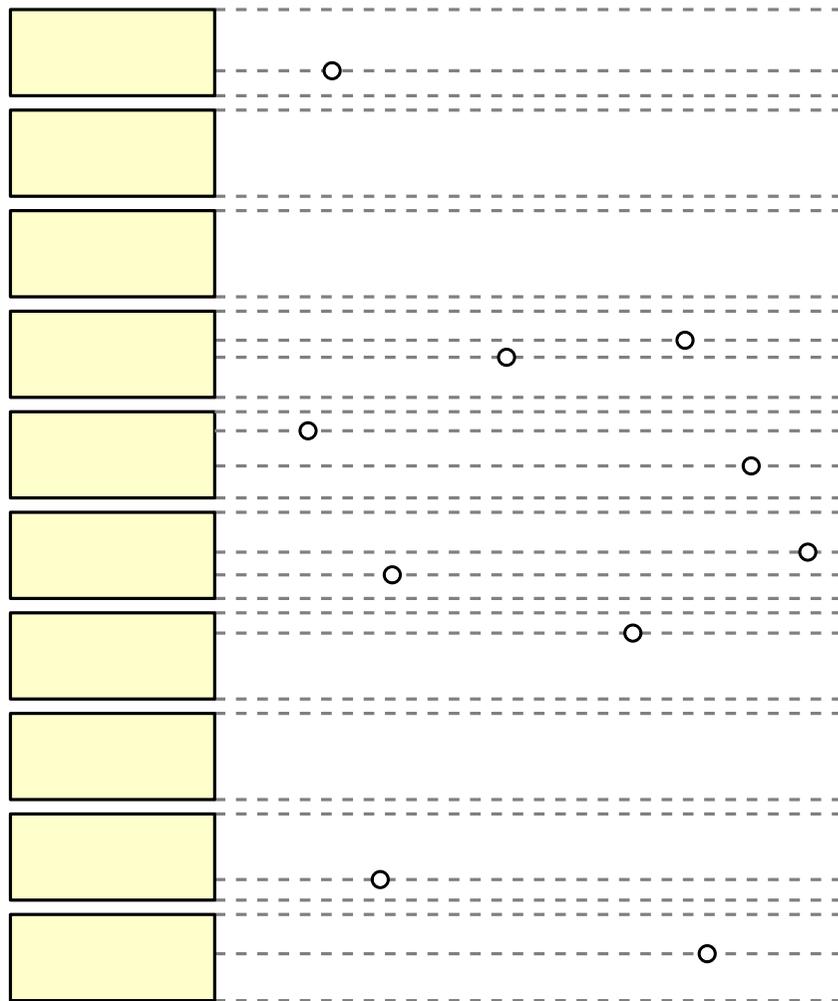


Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter.

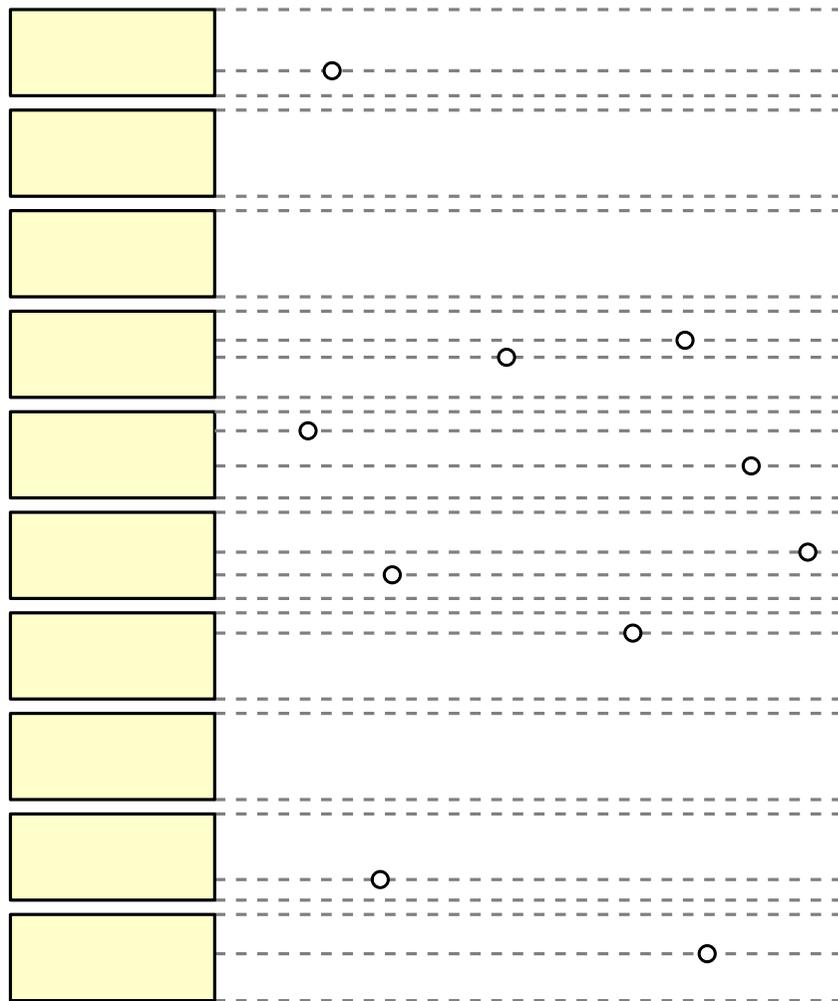


Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...



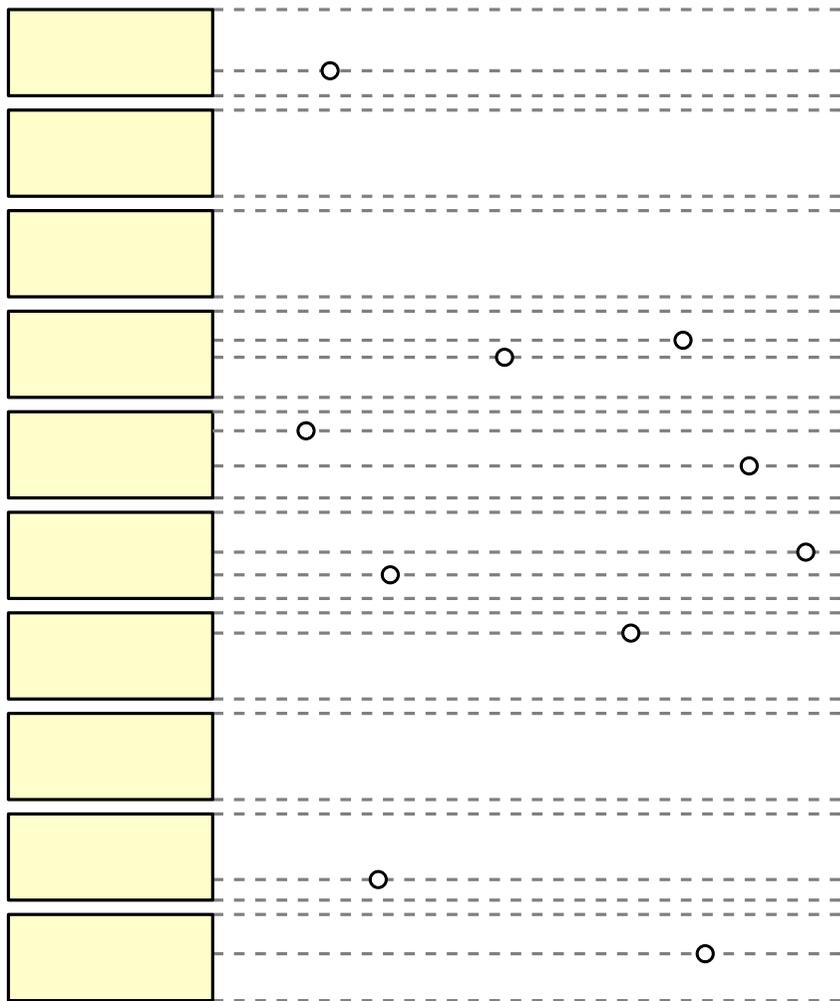
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.



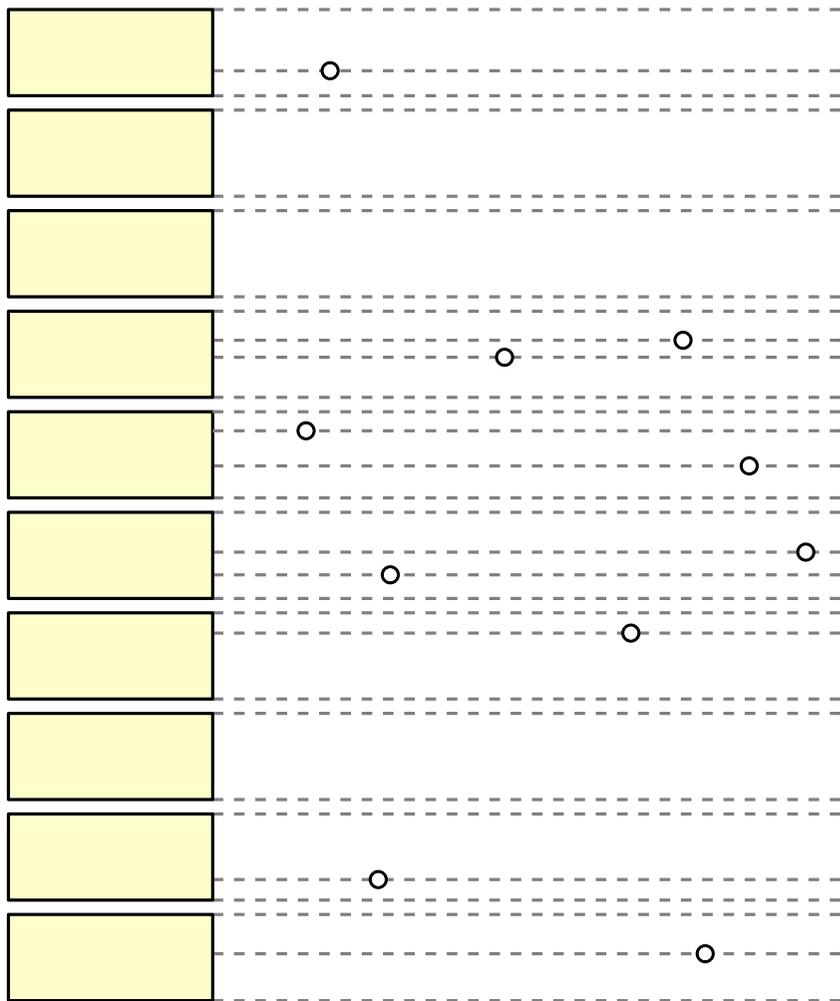
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.



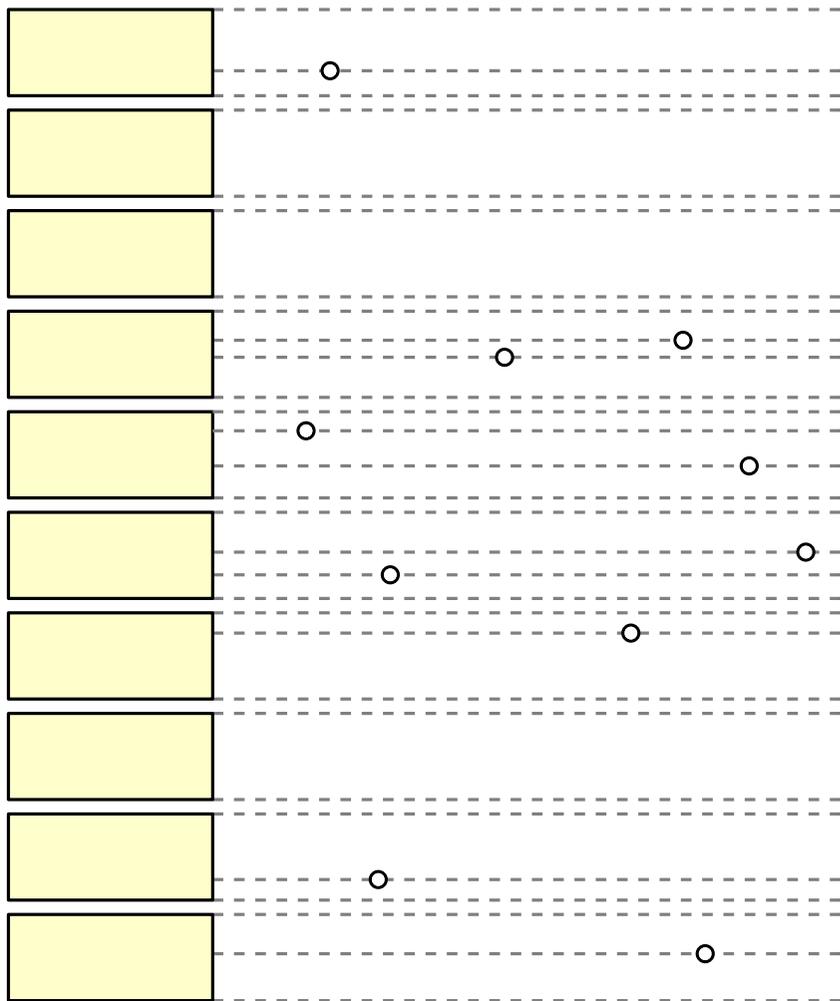
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



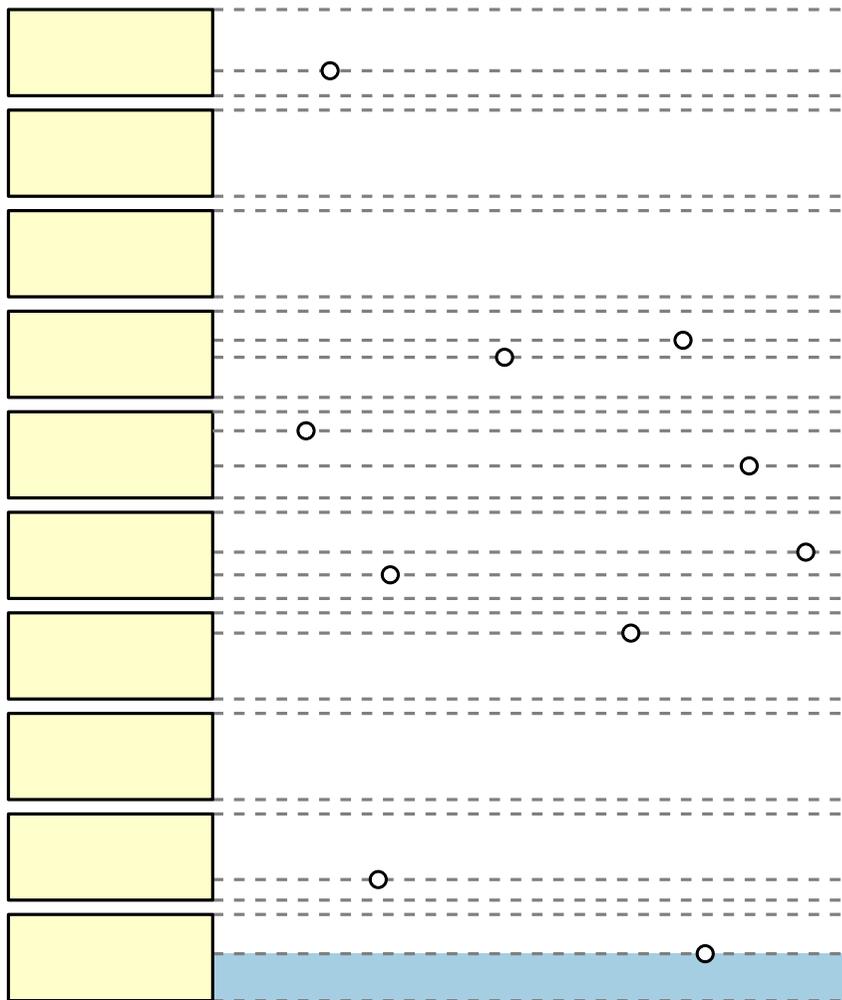
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



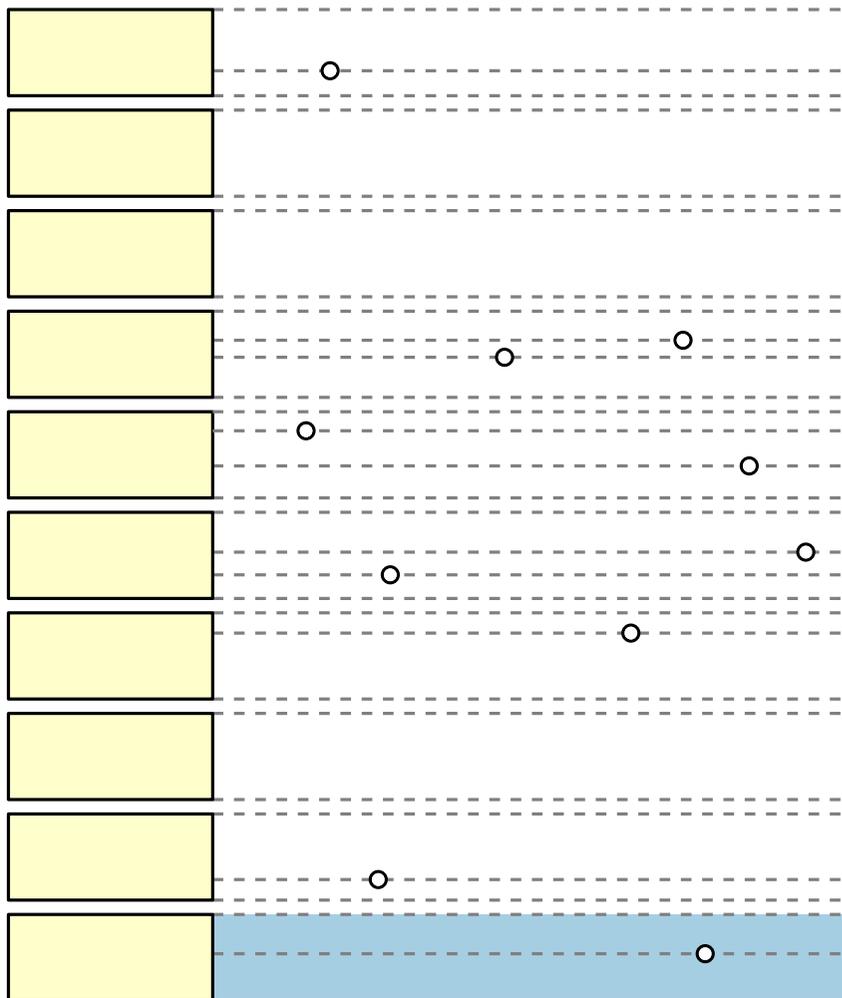
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



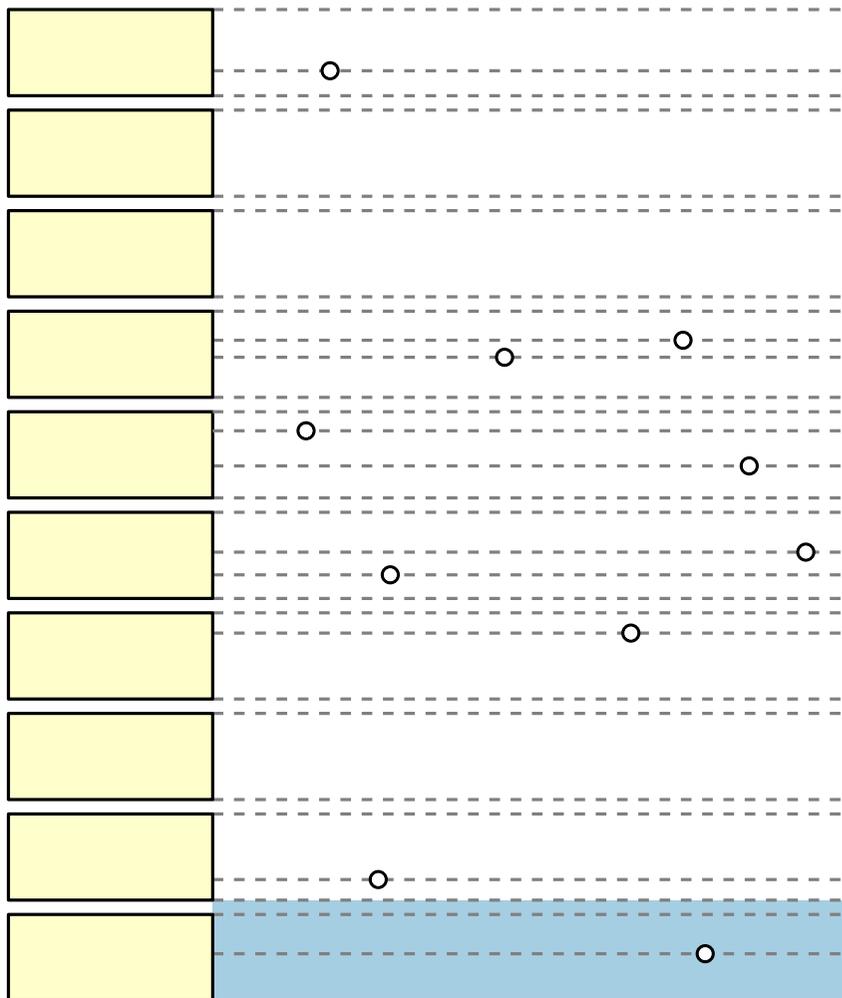
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



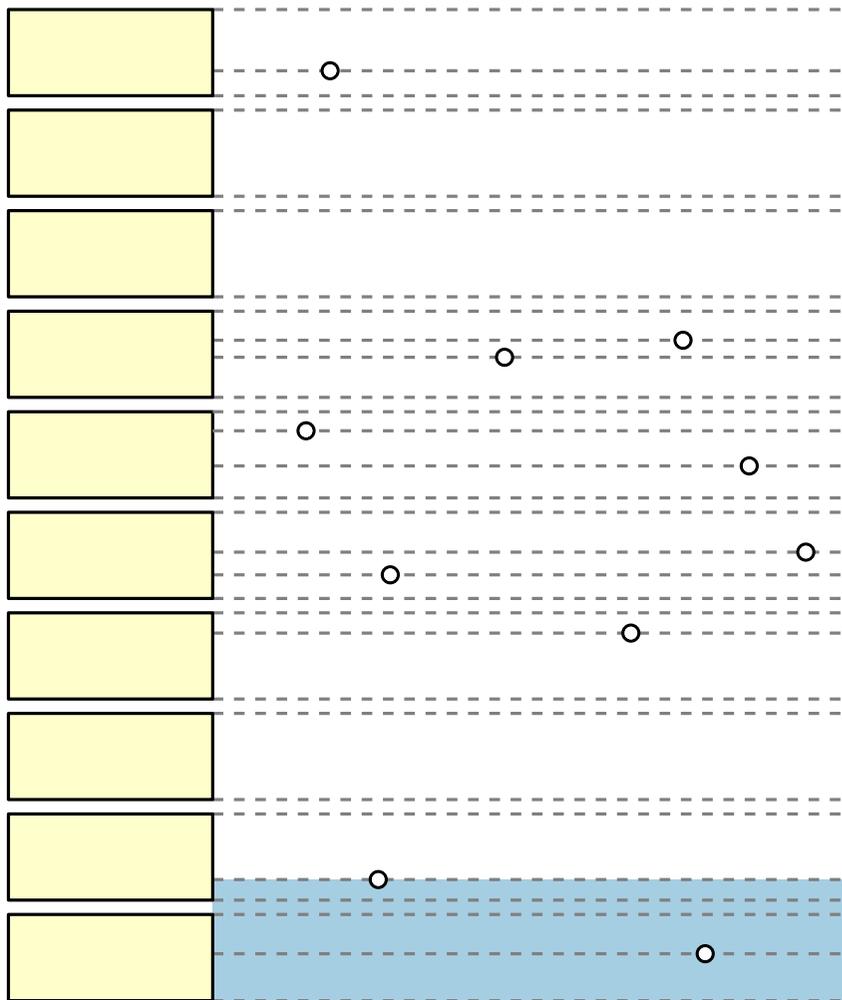
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



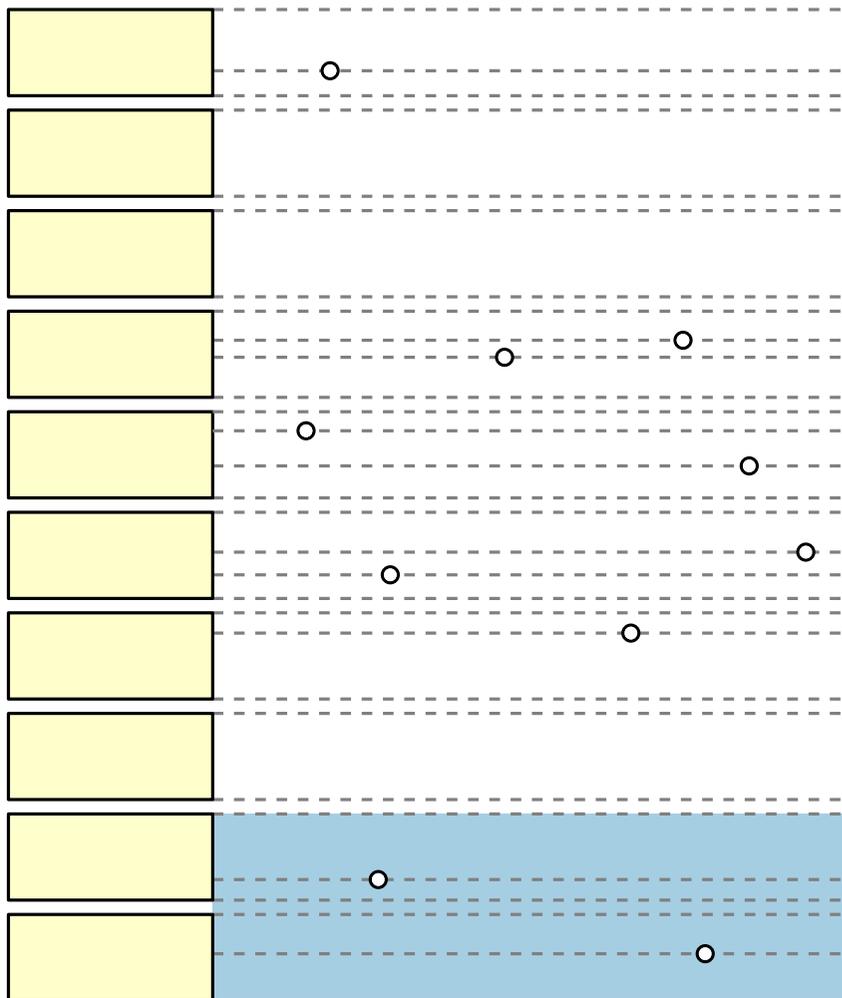
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



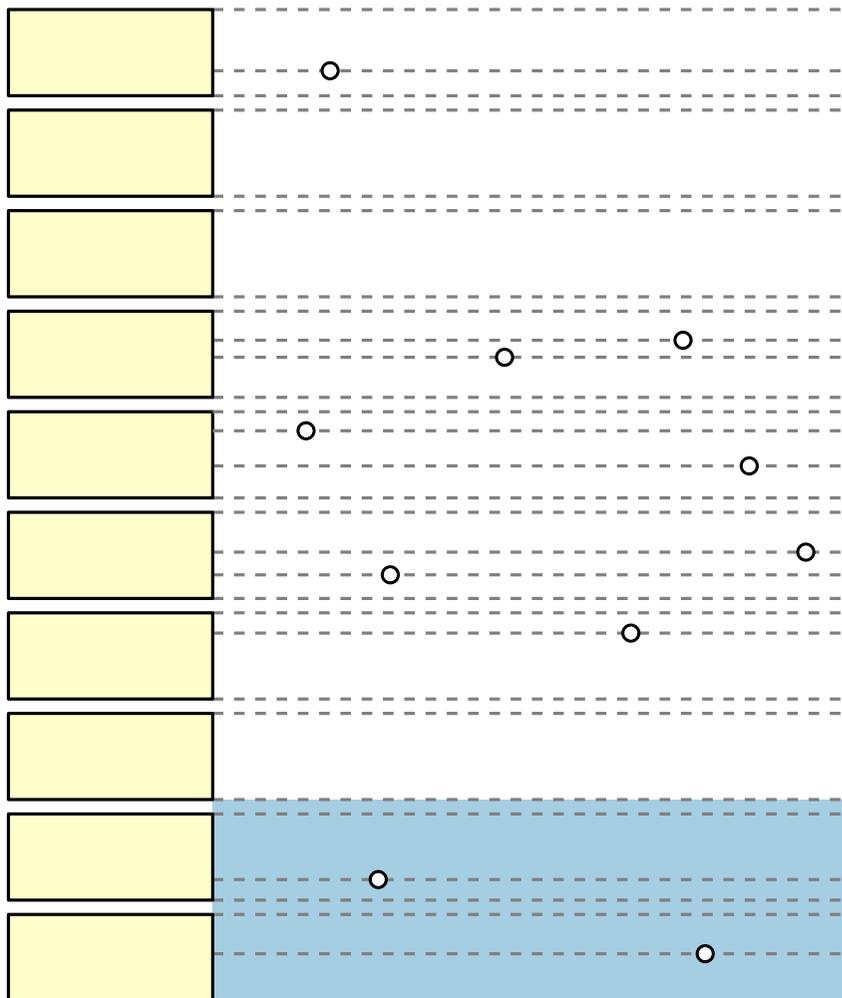
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



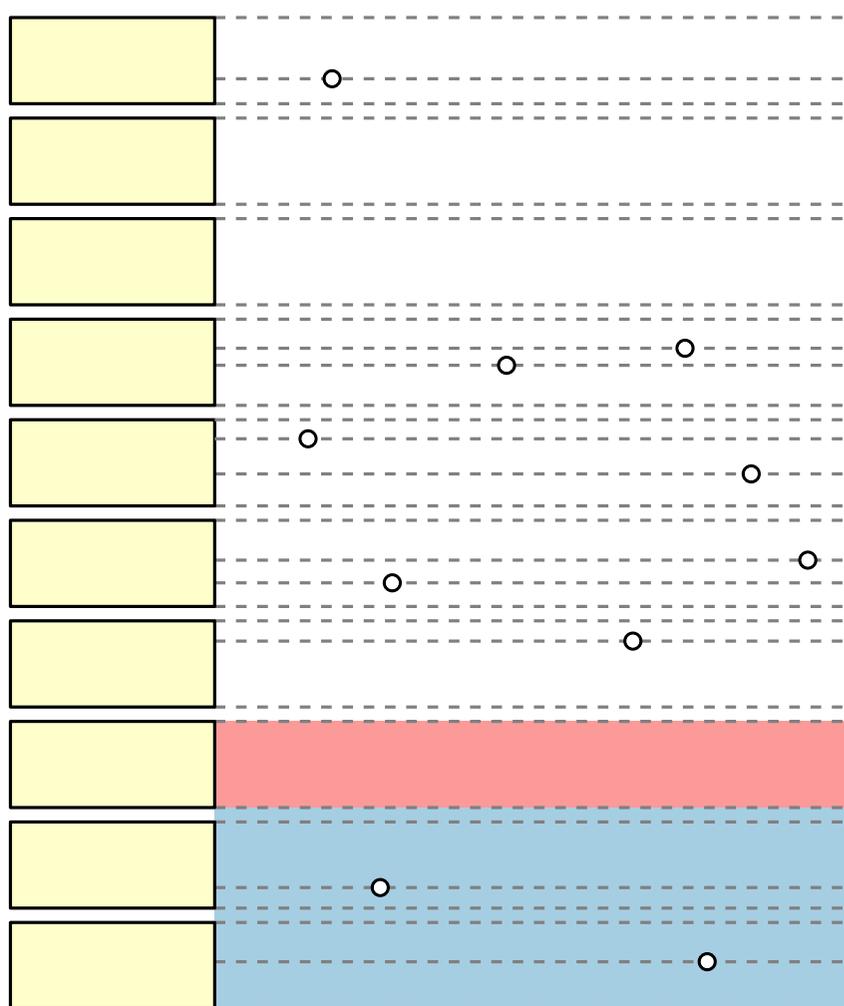
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



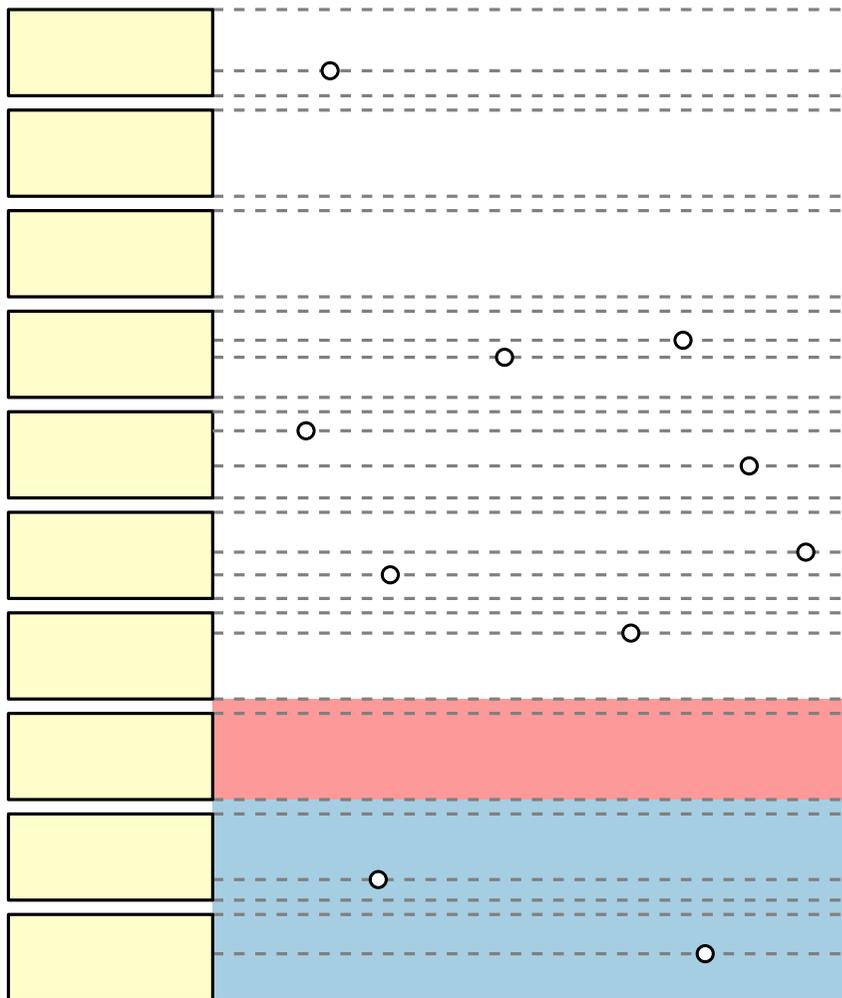
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



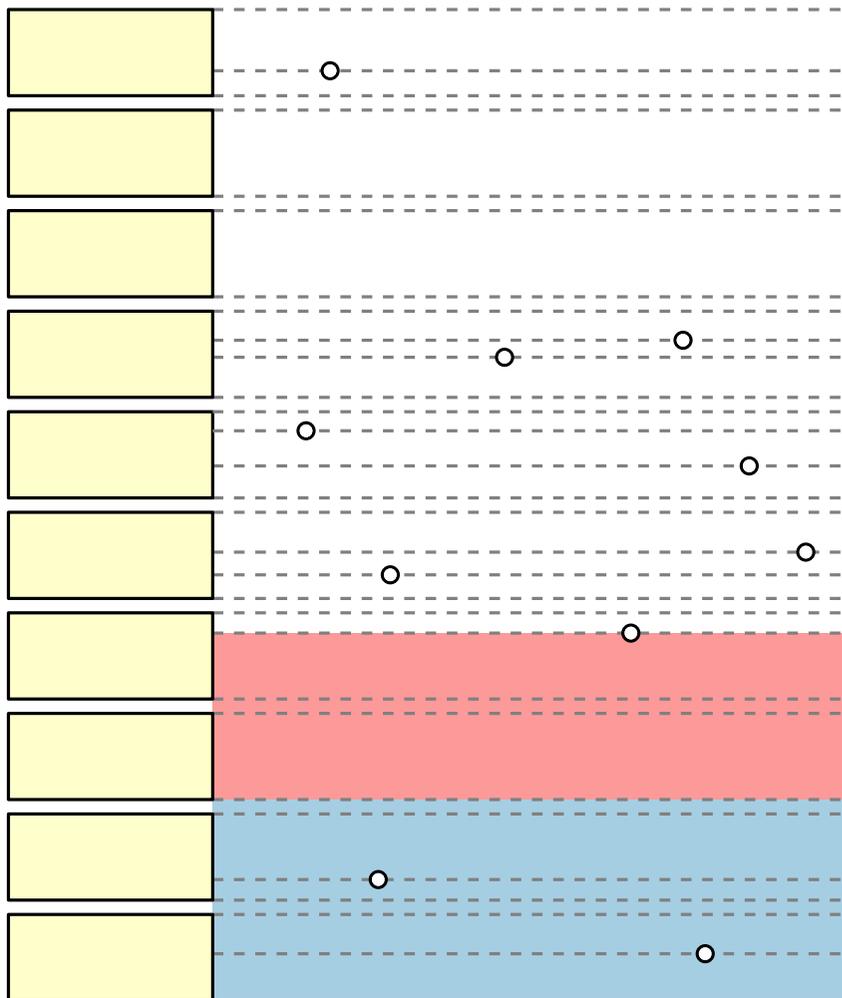
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



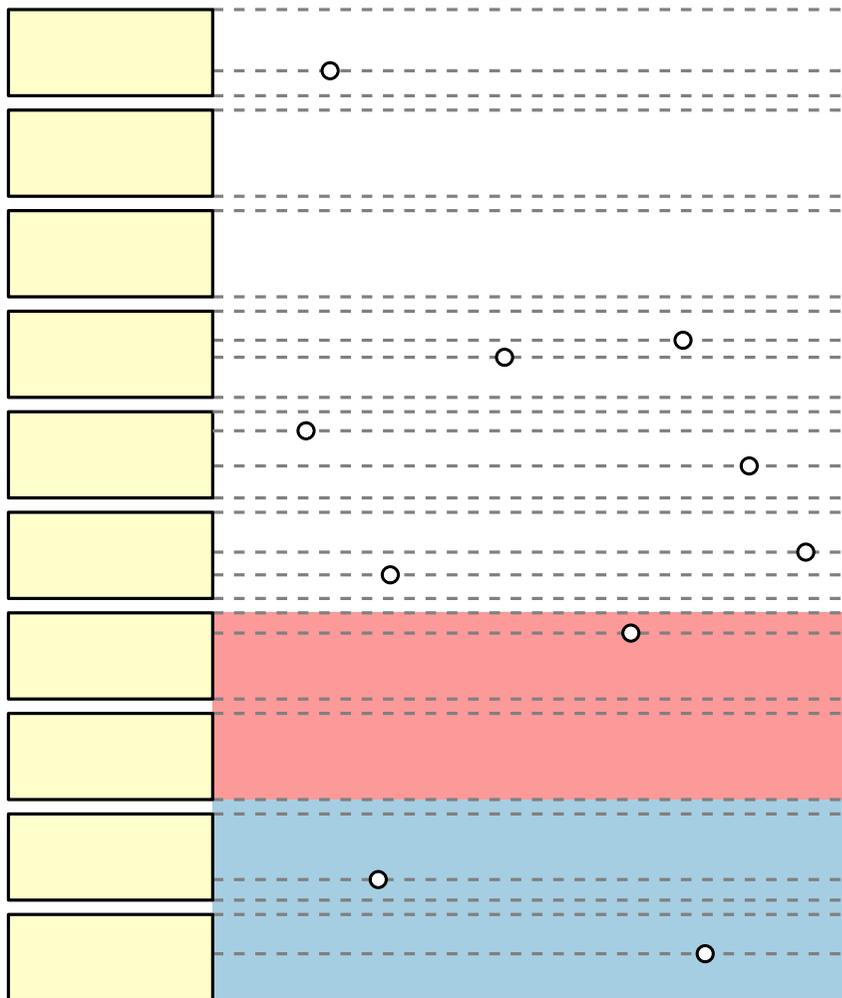
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



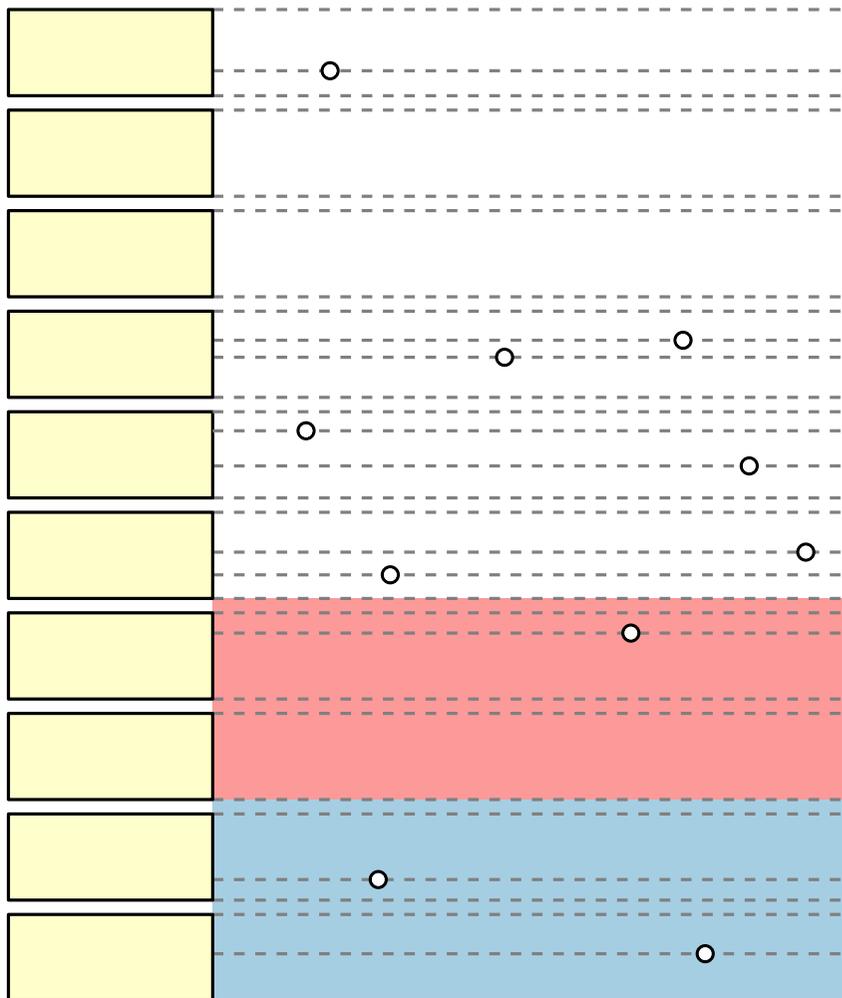
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



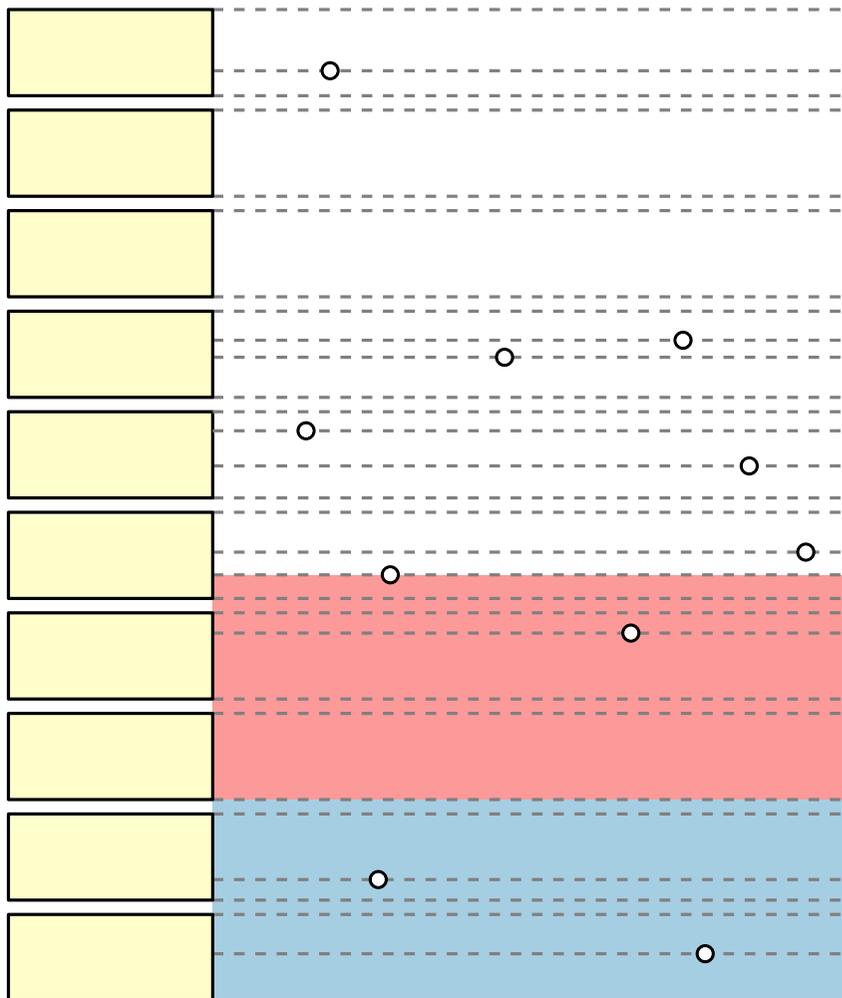
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



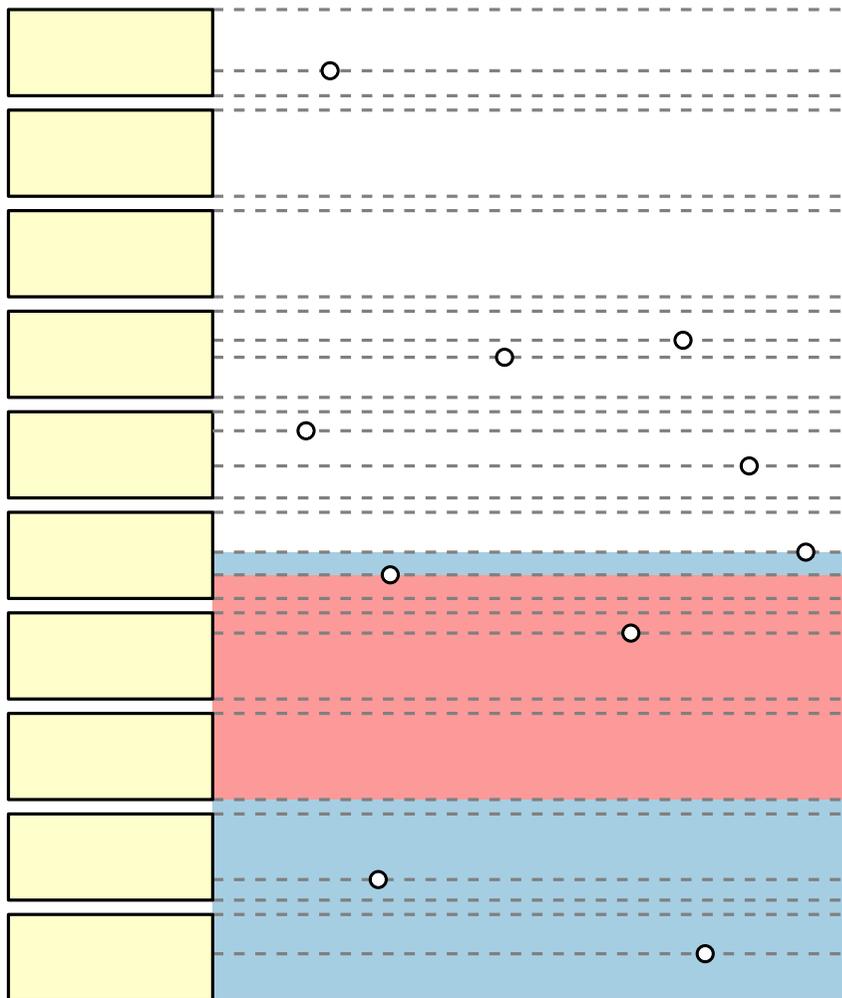
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



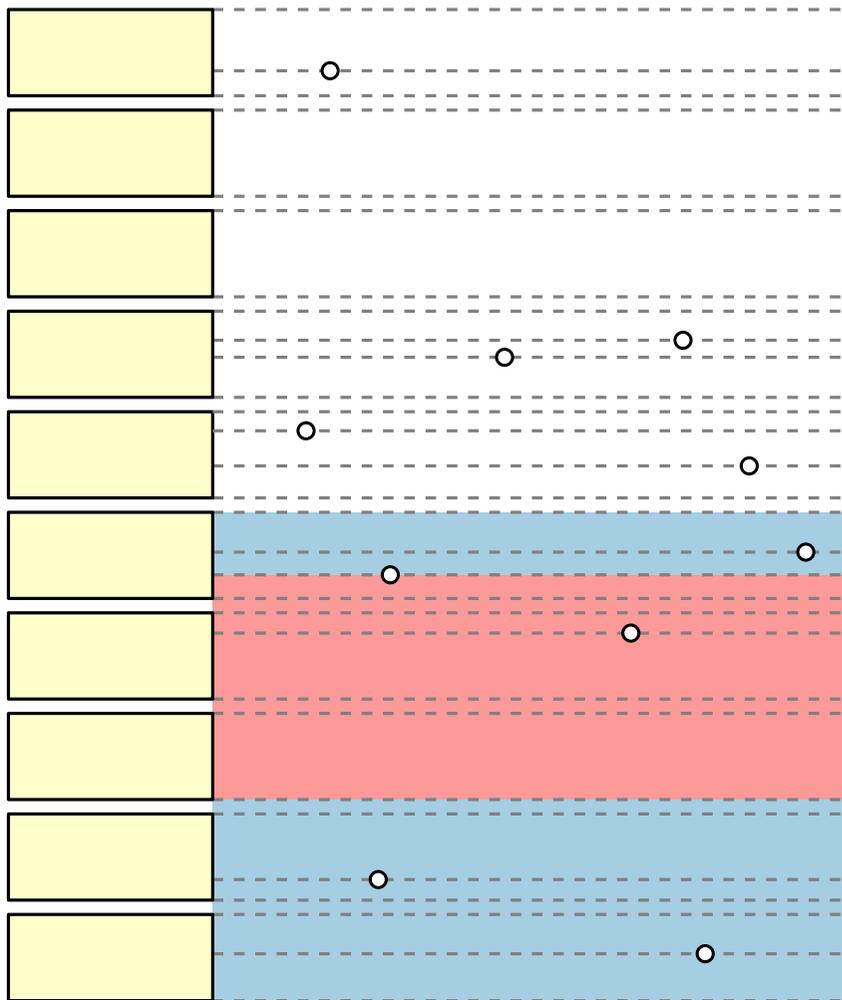
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



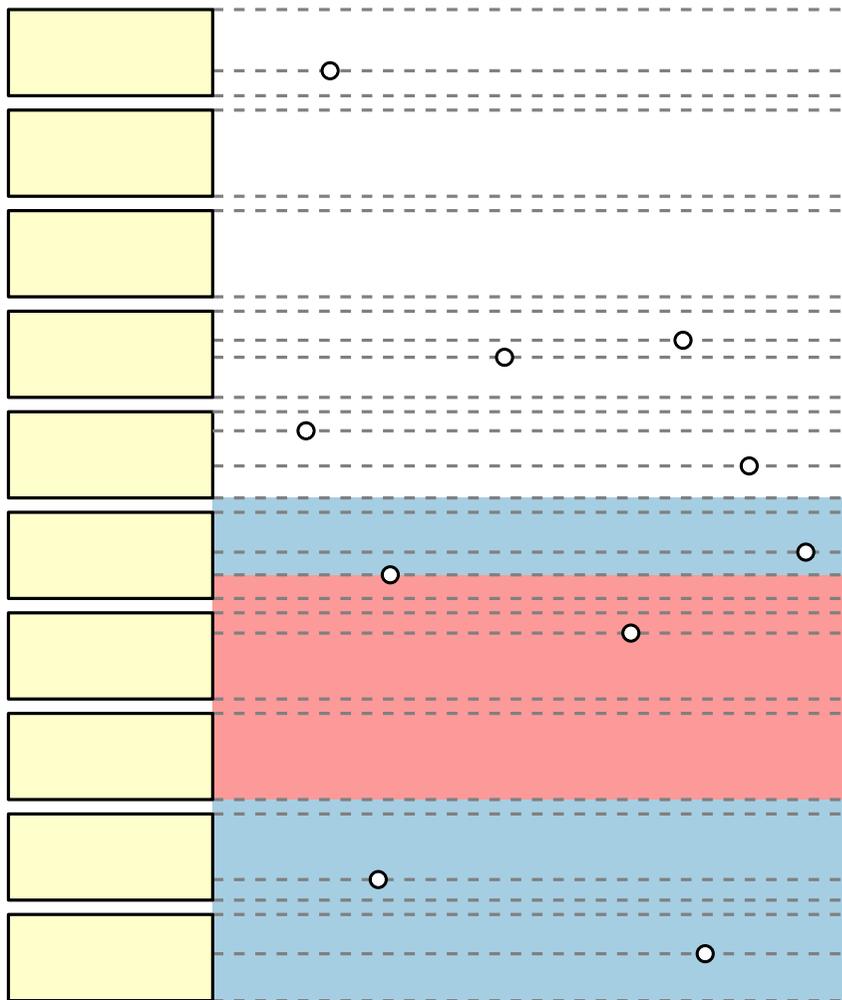
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



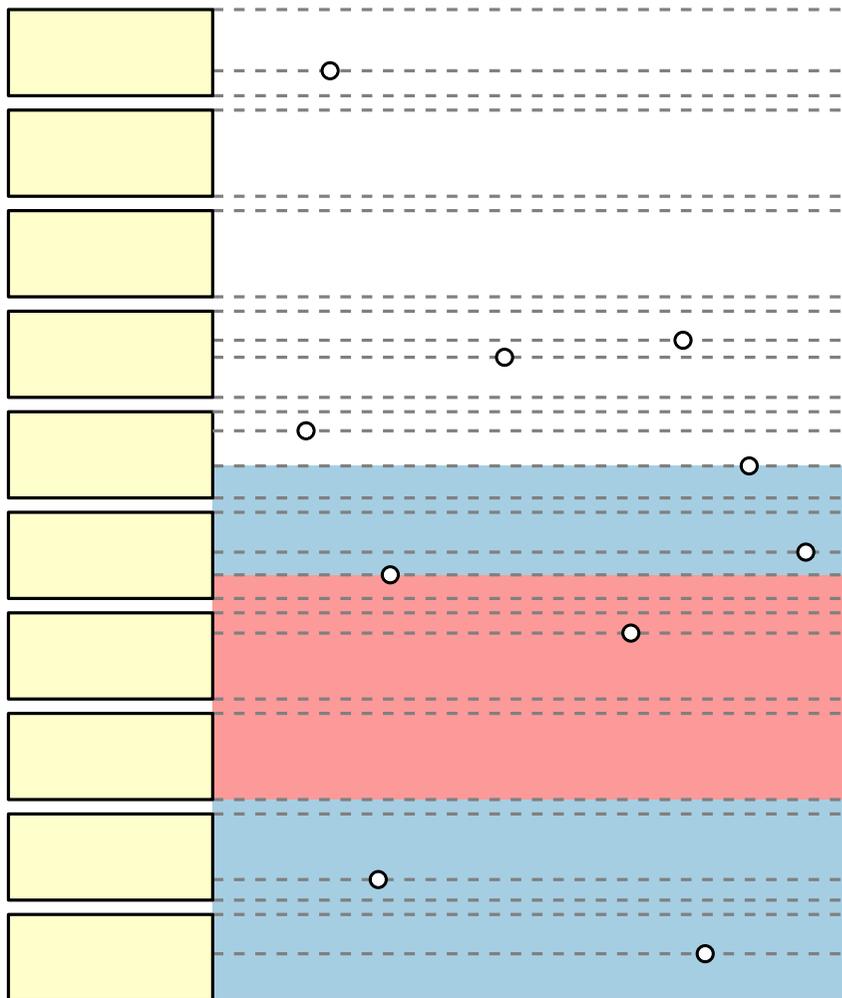
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



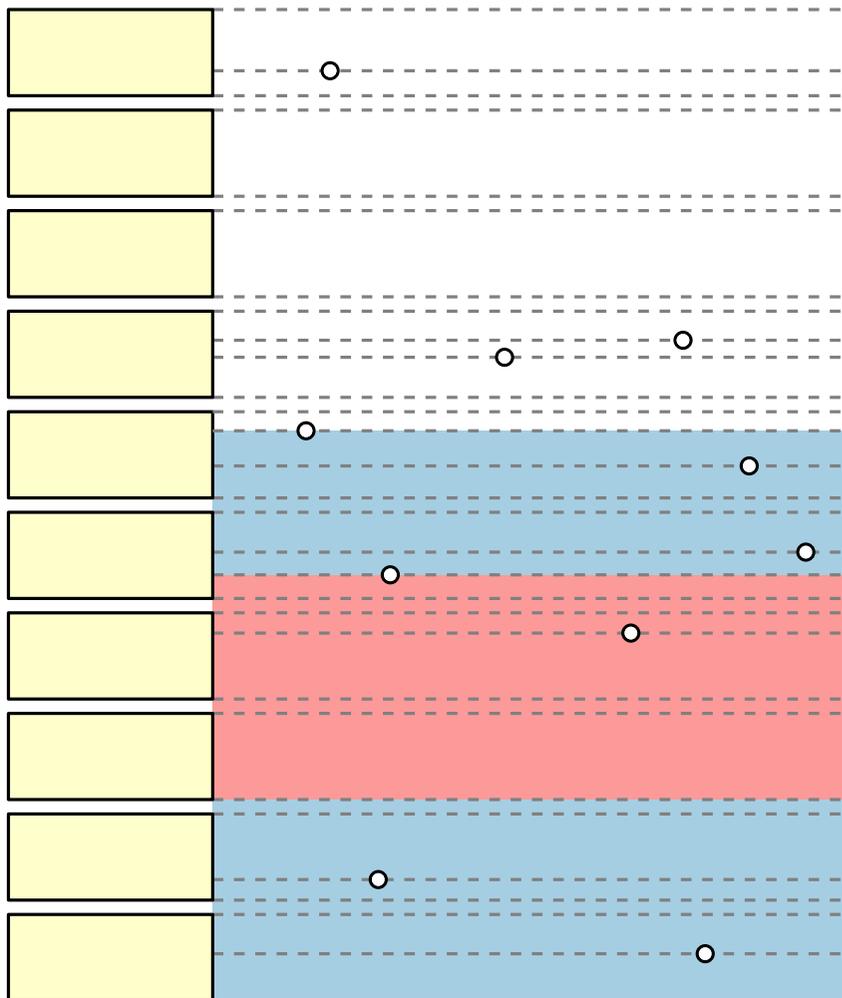
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



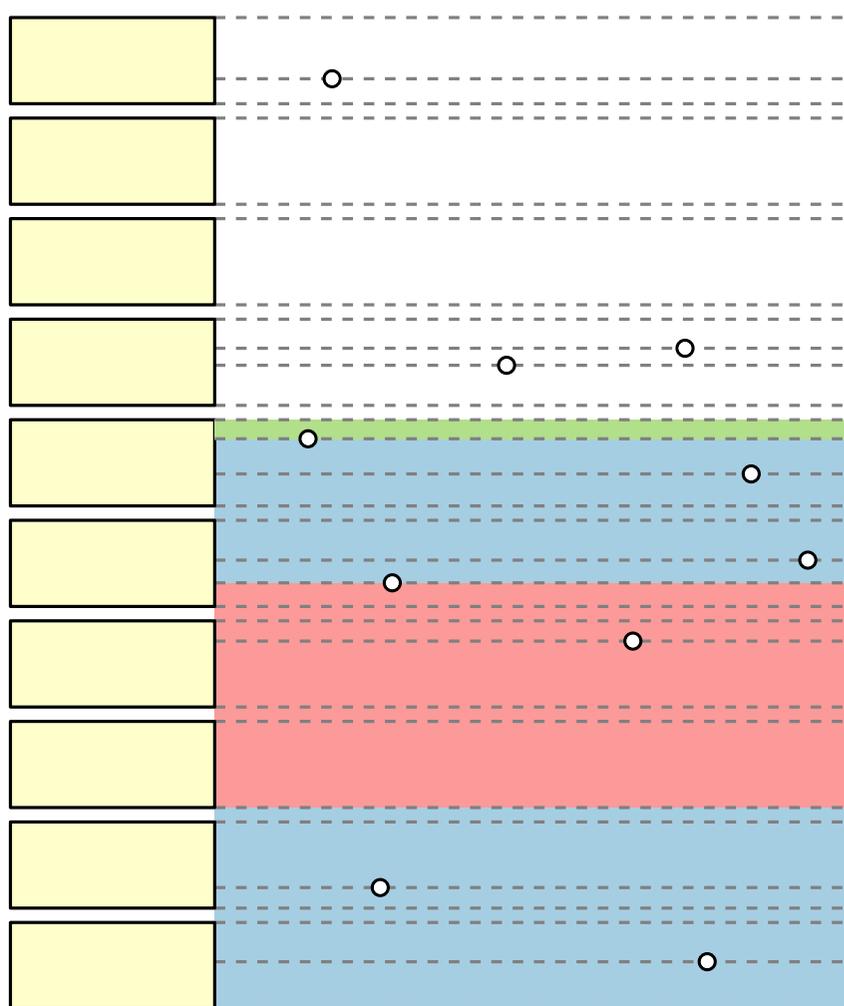
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



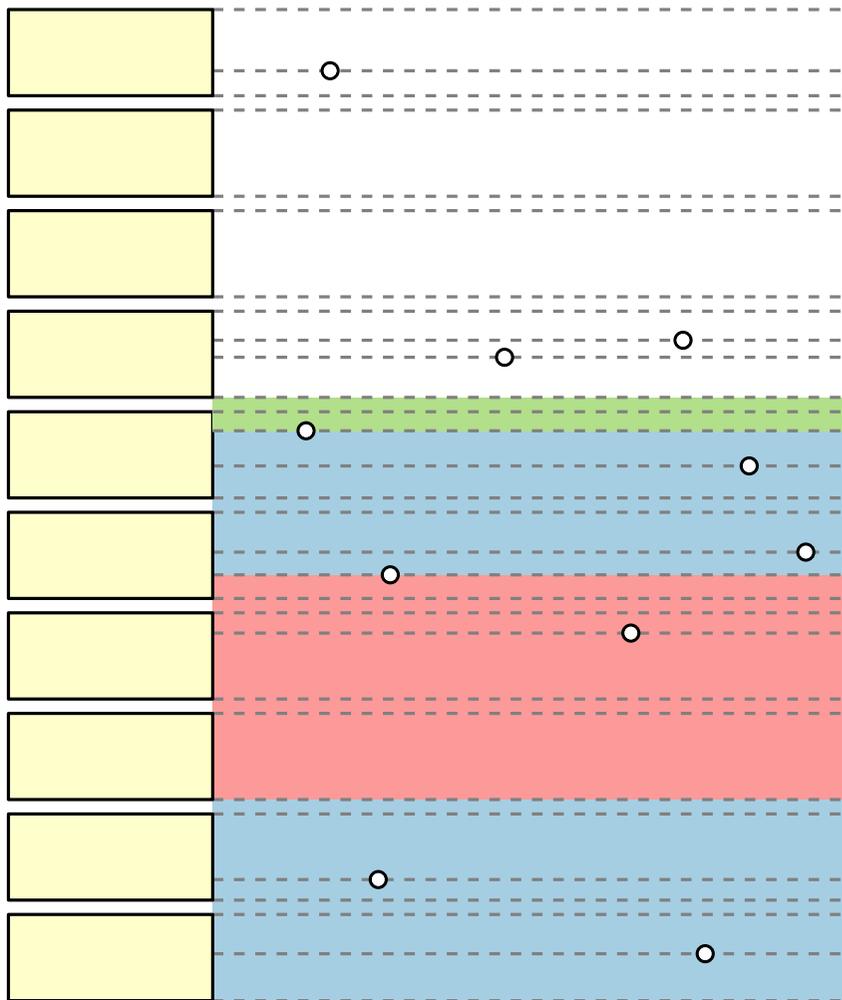
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



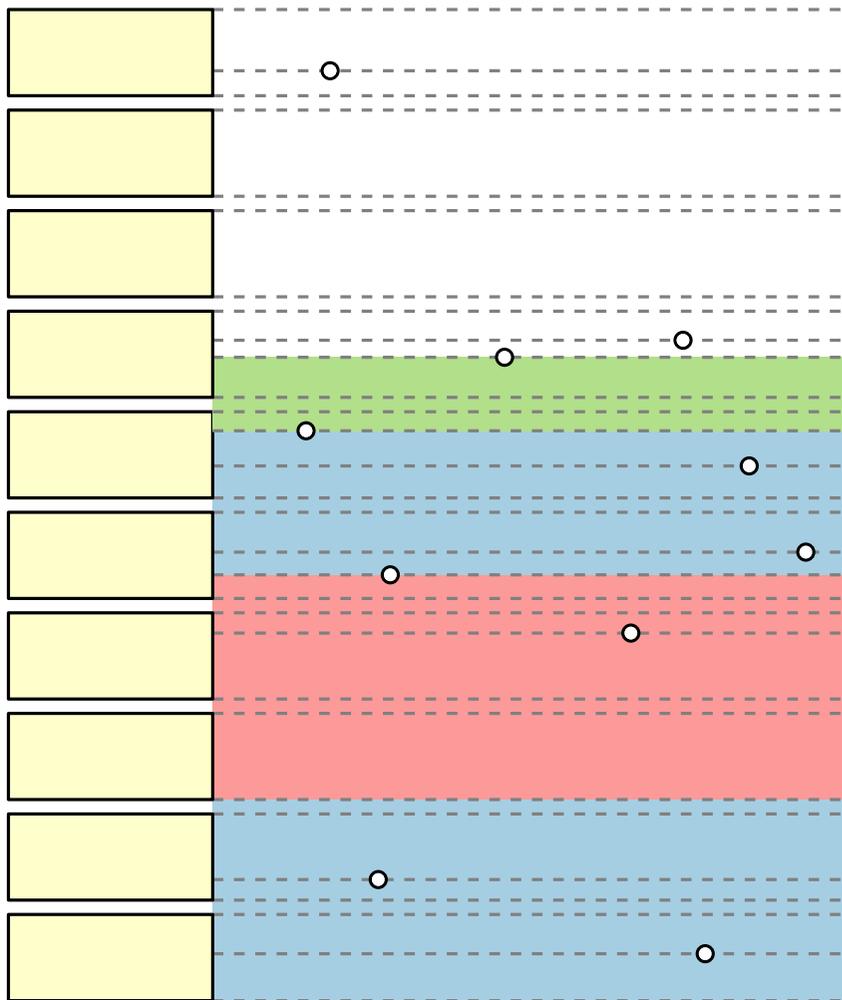
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



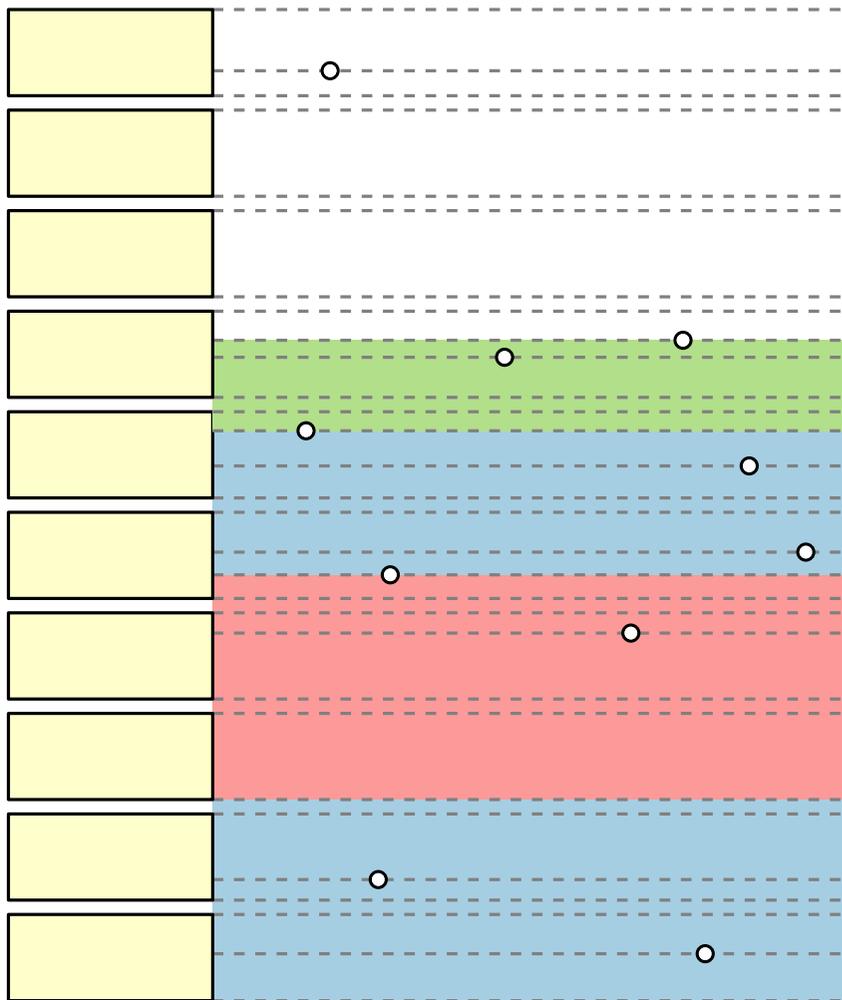
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



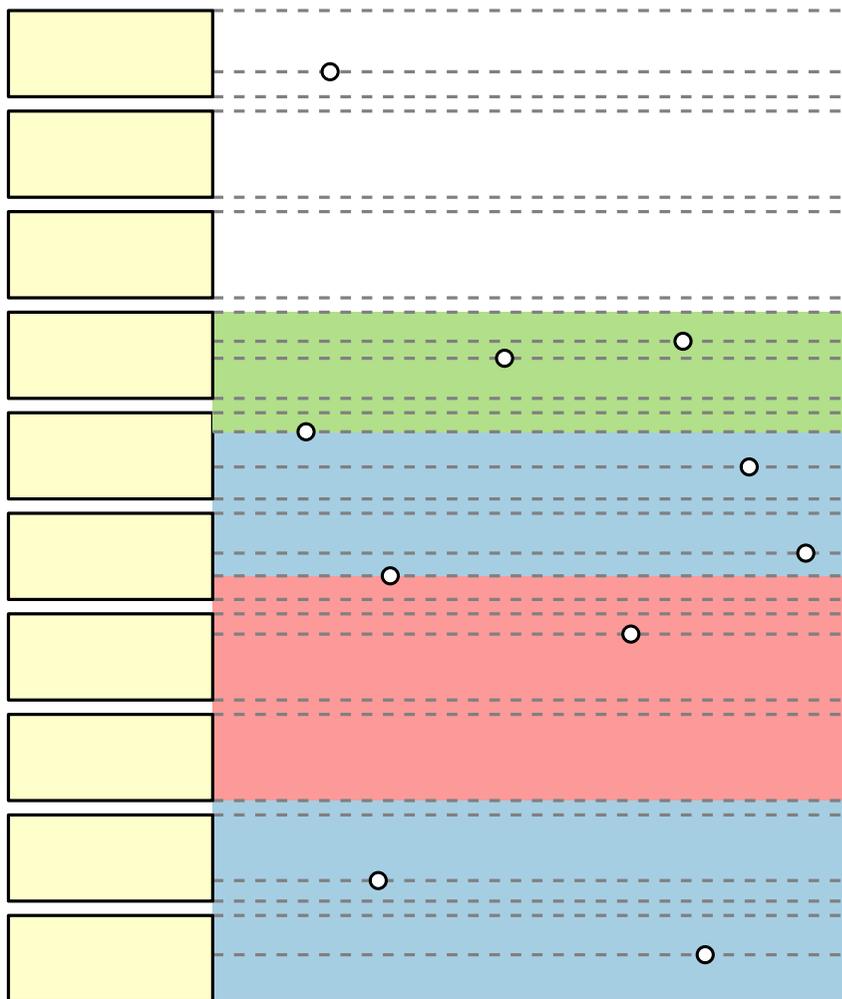
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



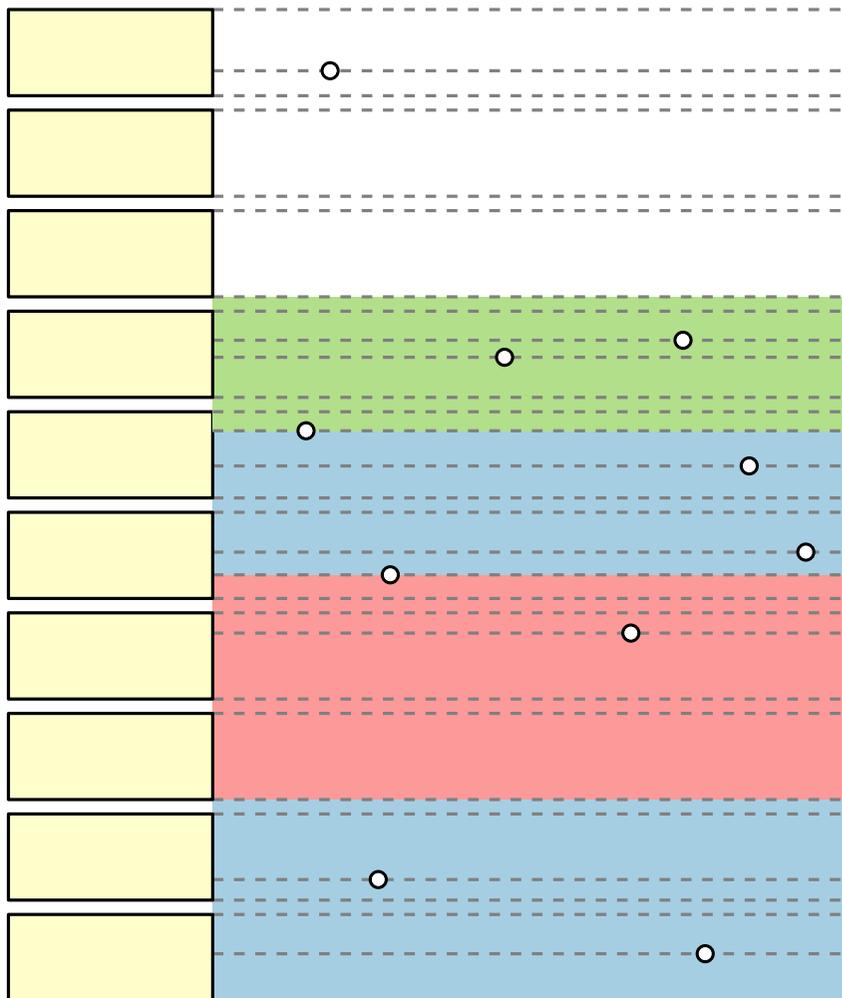
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



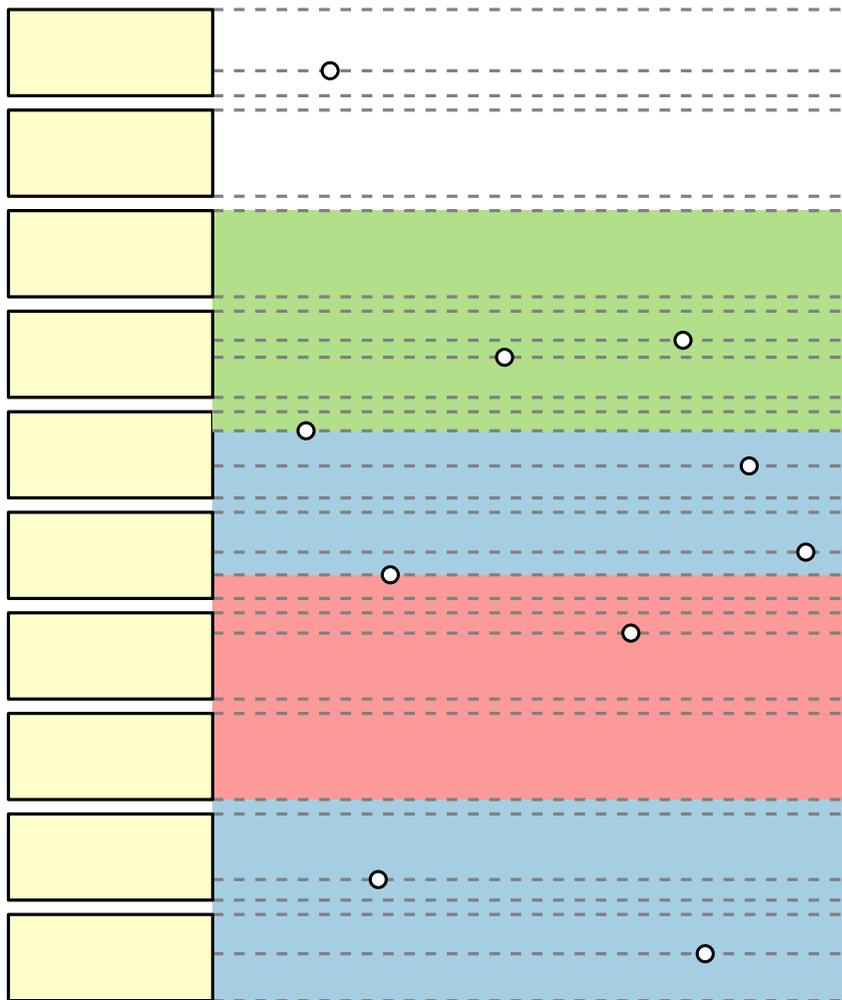
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



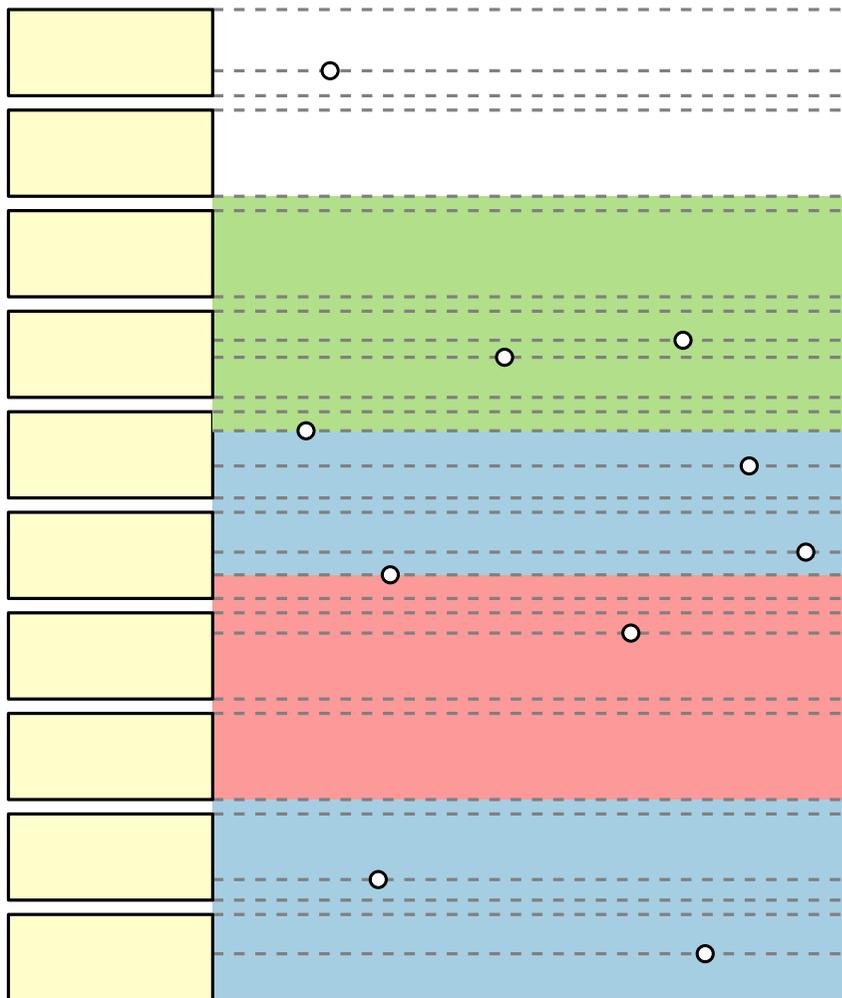
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



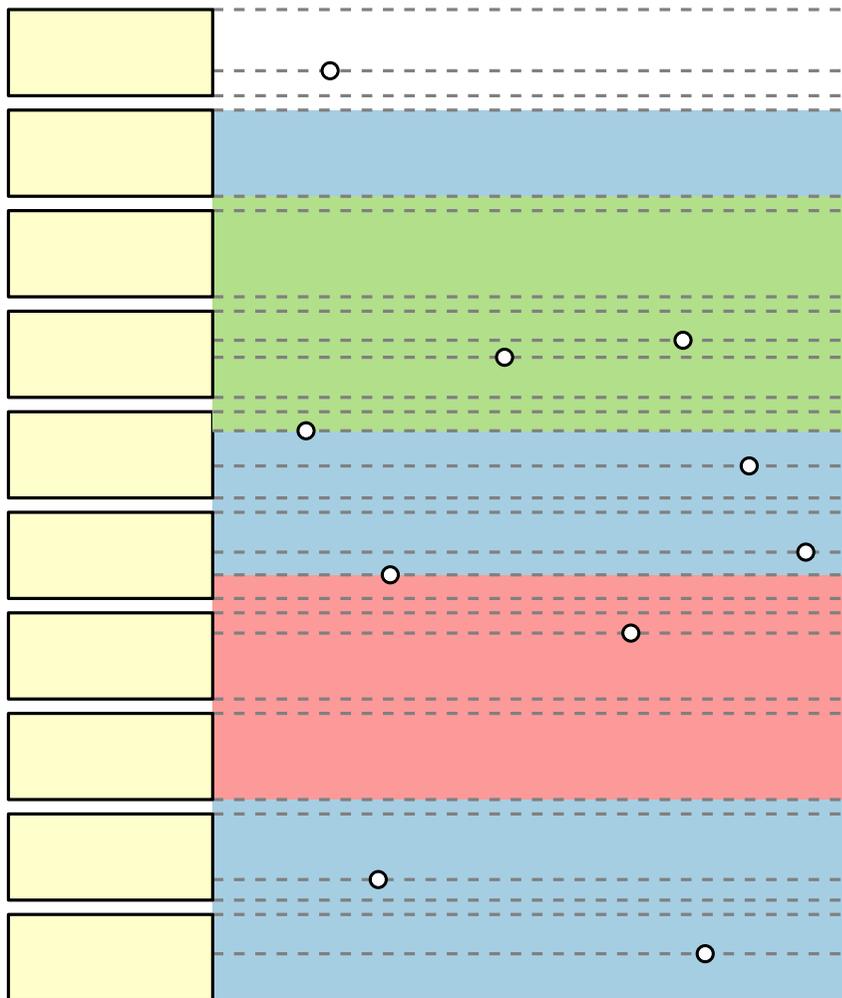
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



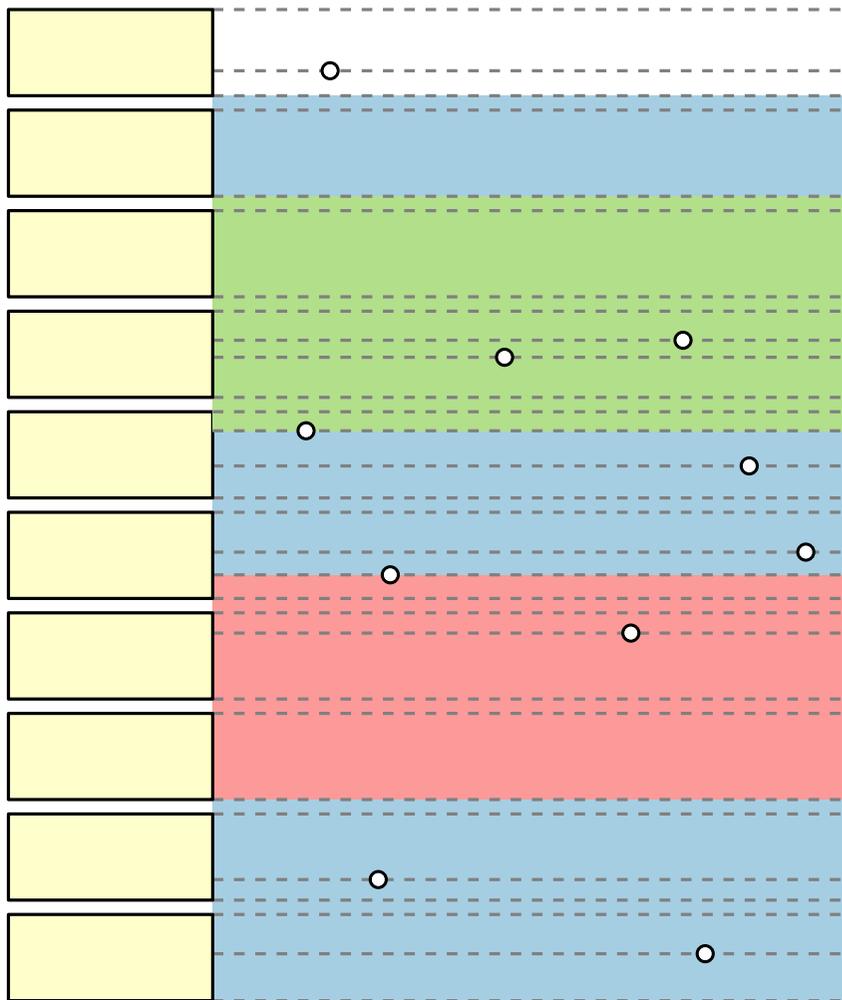
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



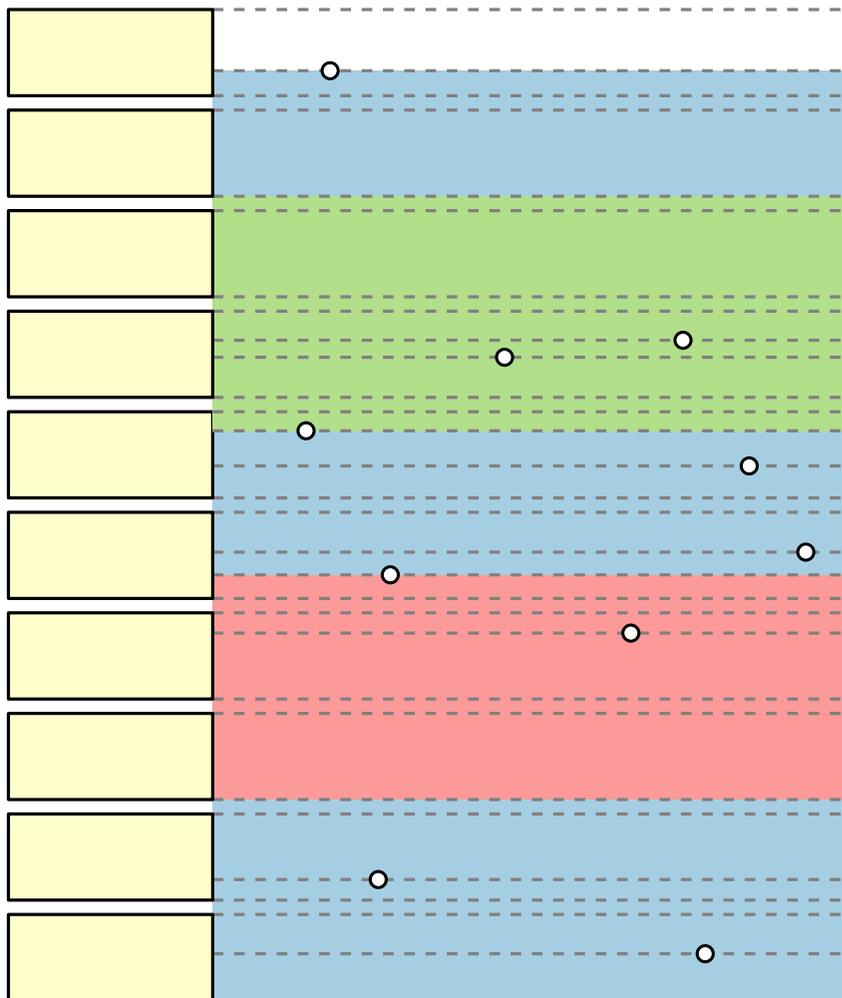
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



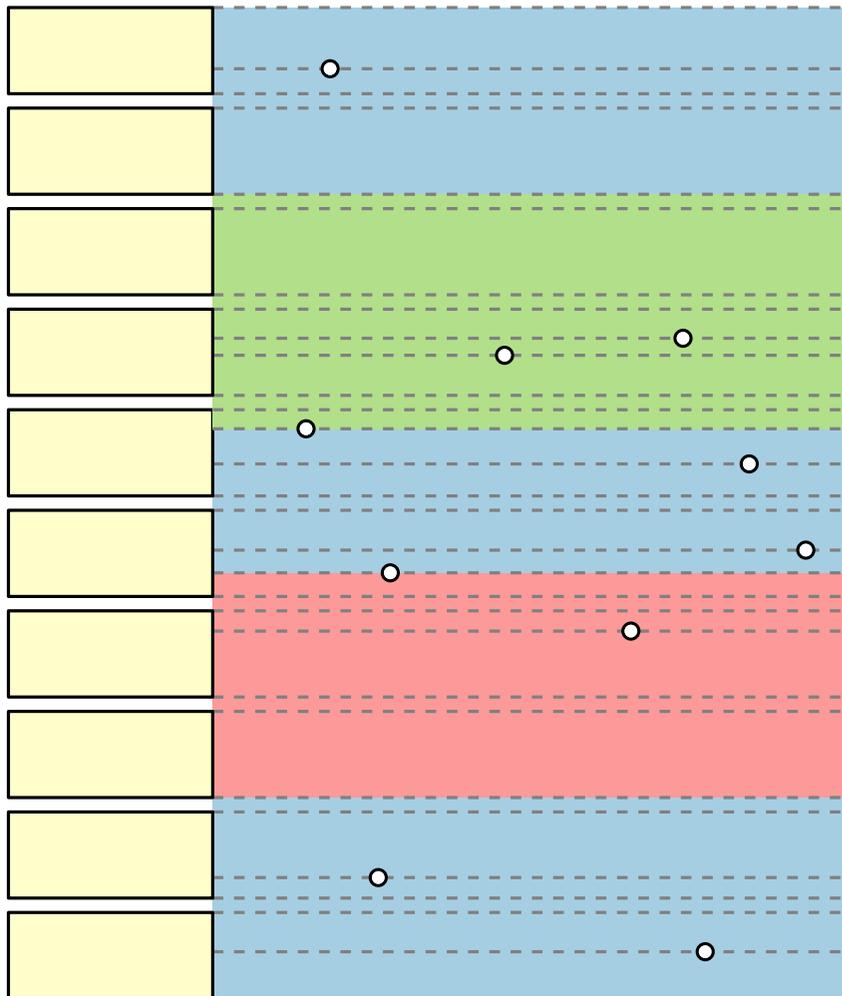
Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- neutral, falls gleiche Anzahl.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



Sweepline-Verfahren

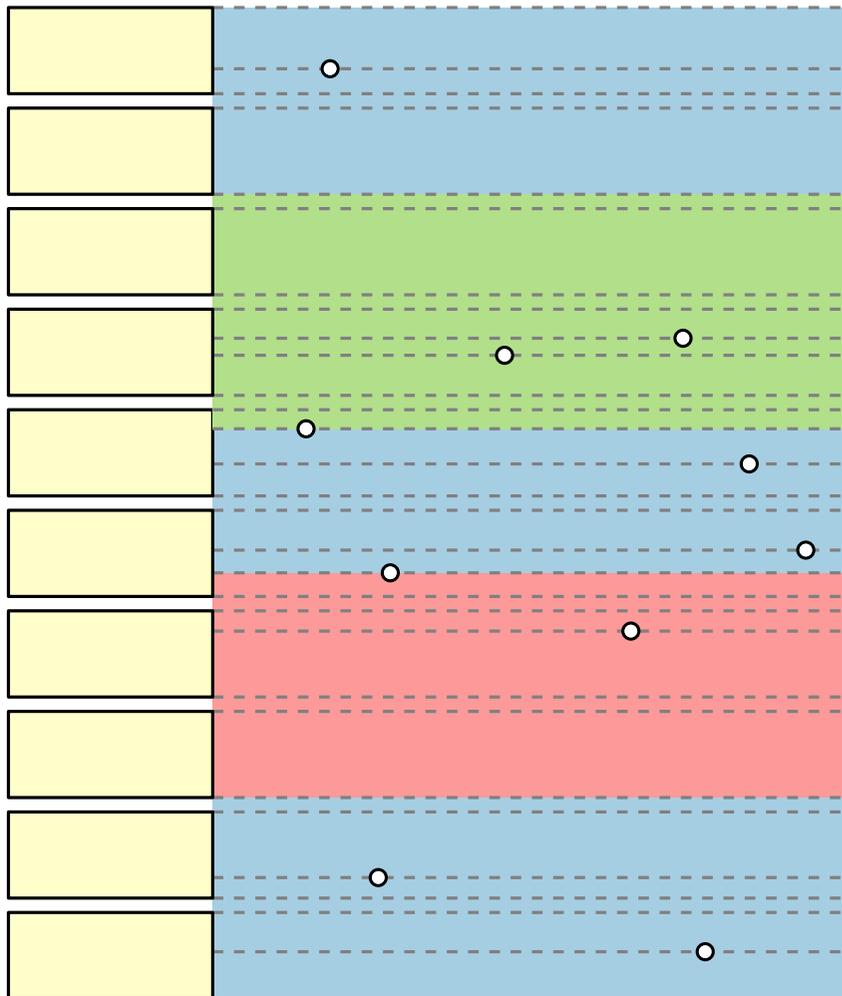
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

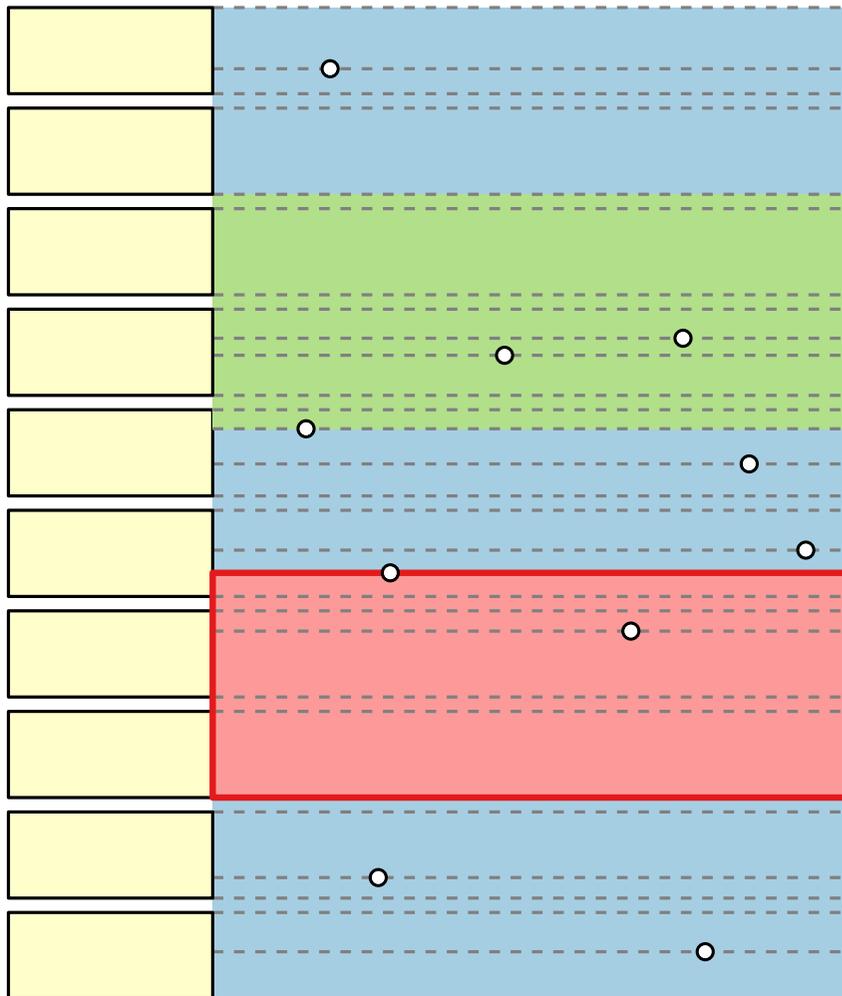
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

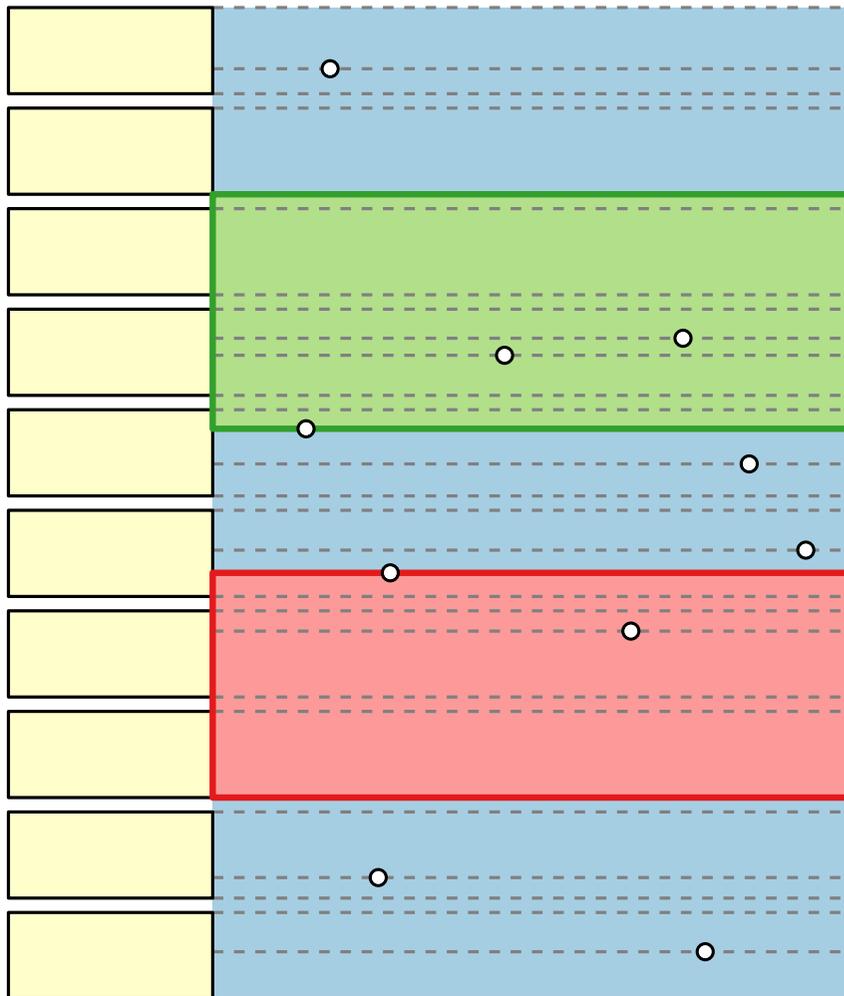
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

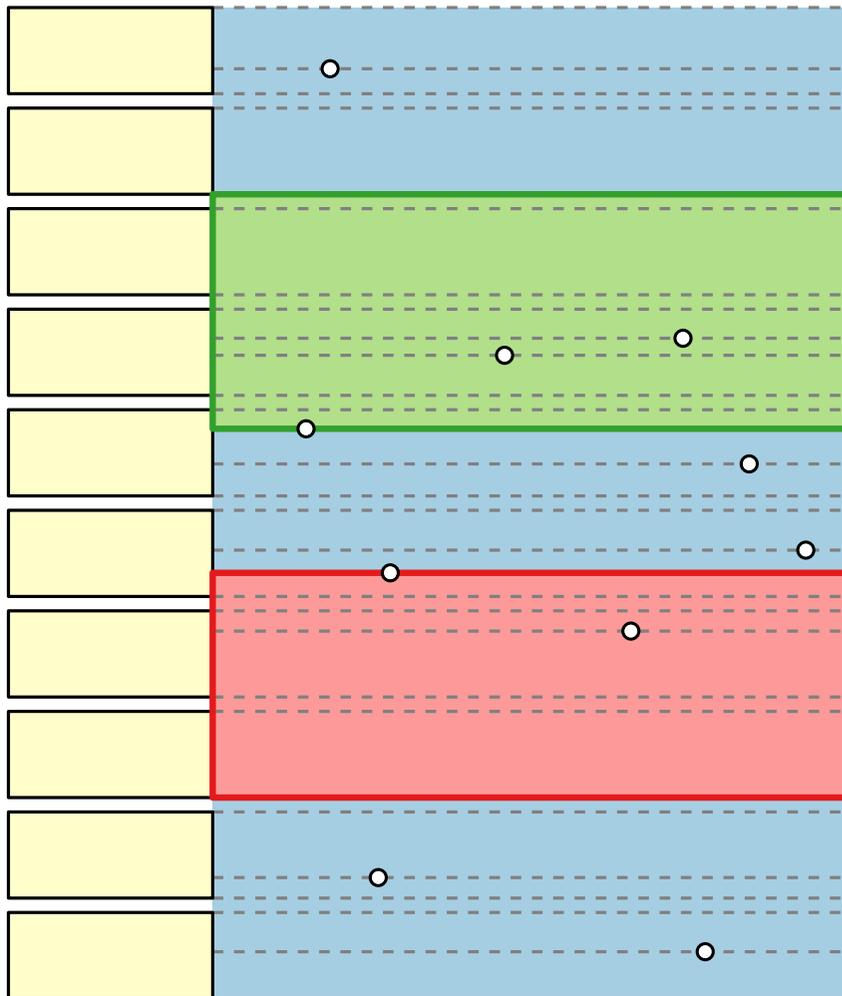
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

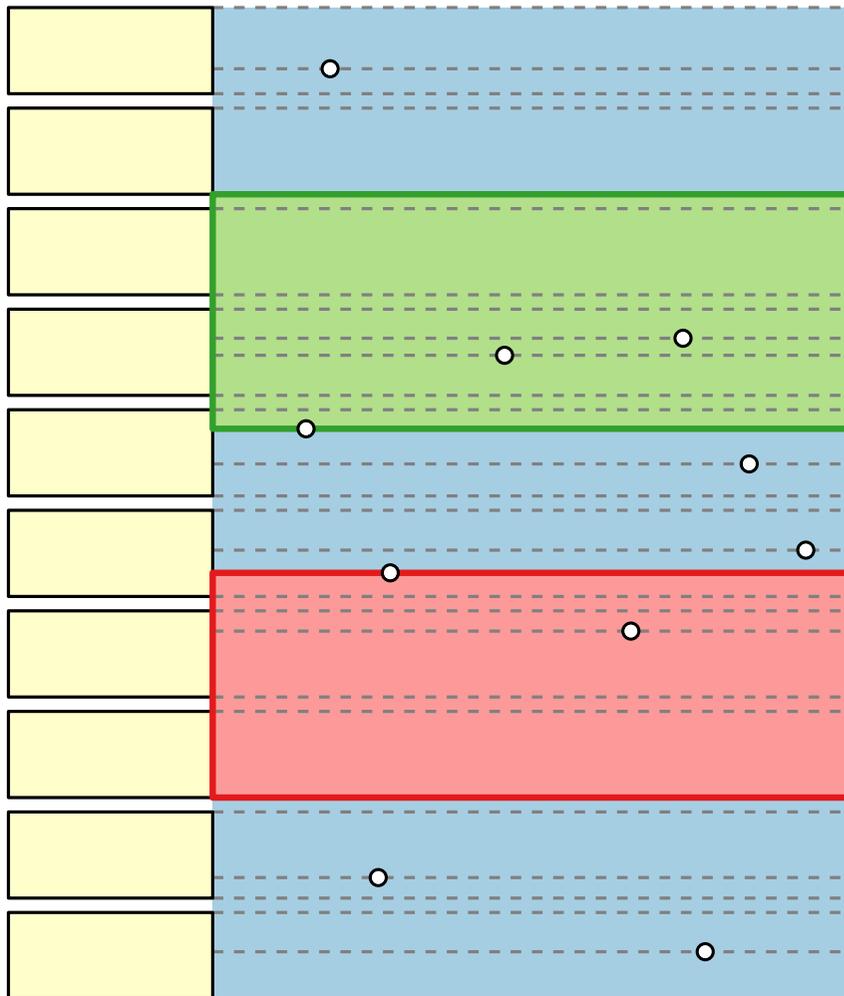
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

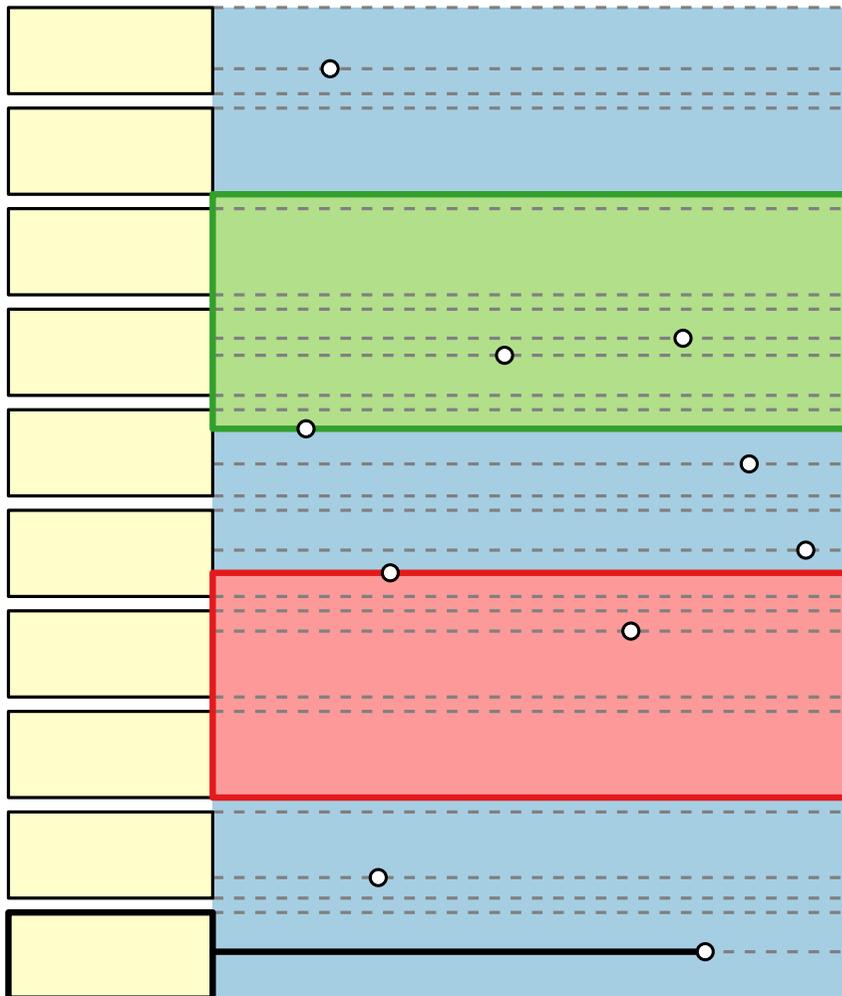
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

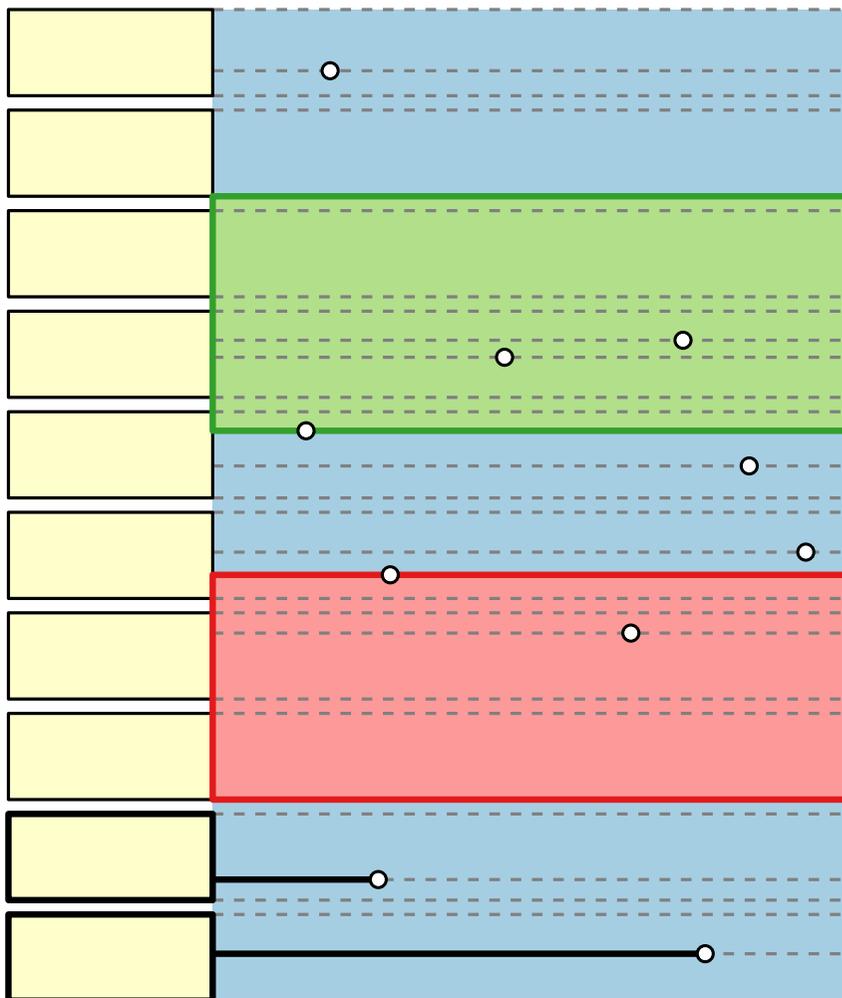
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- neutral, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- aufsteigend, falls mehr Punkte.
- absteigend, falls mehr Beschriftungen.



Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

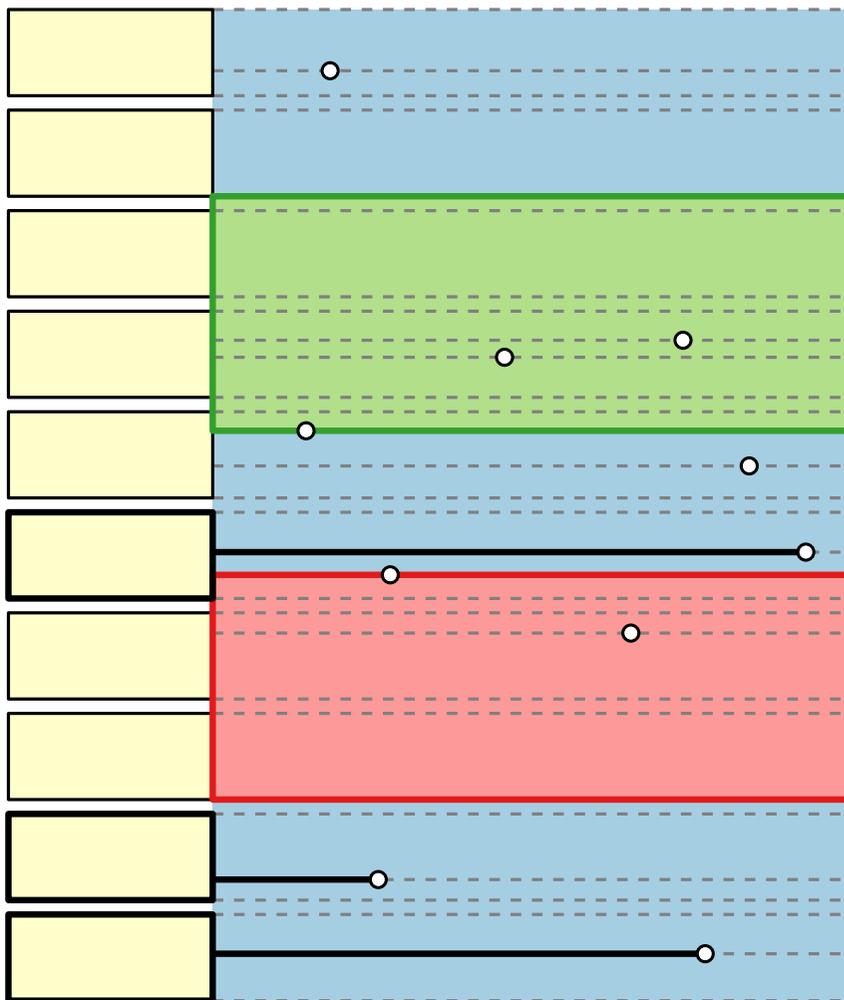
1. Unterteile Instanz in einfache Teilinstanzen.

2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.



Sweepline-Verfahren

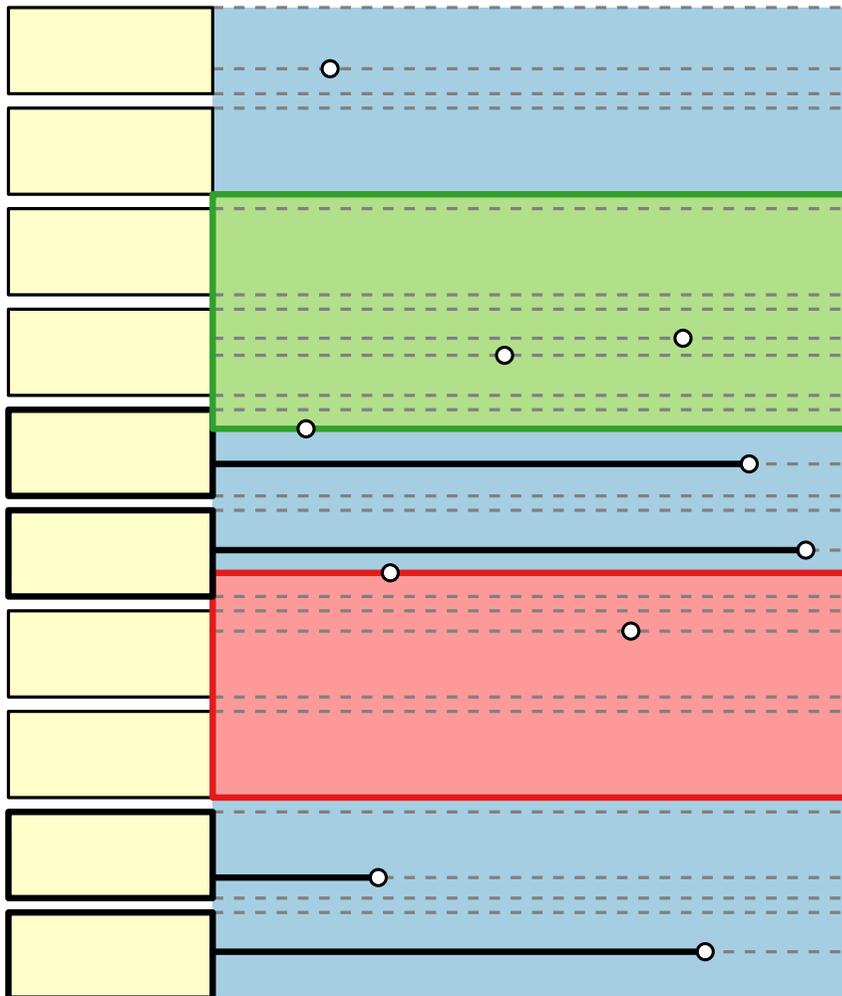
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

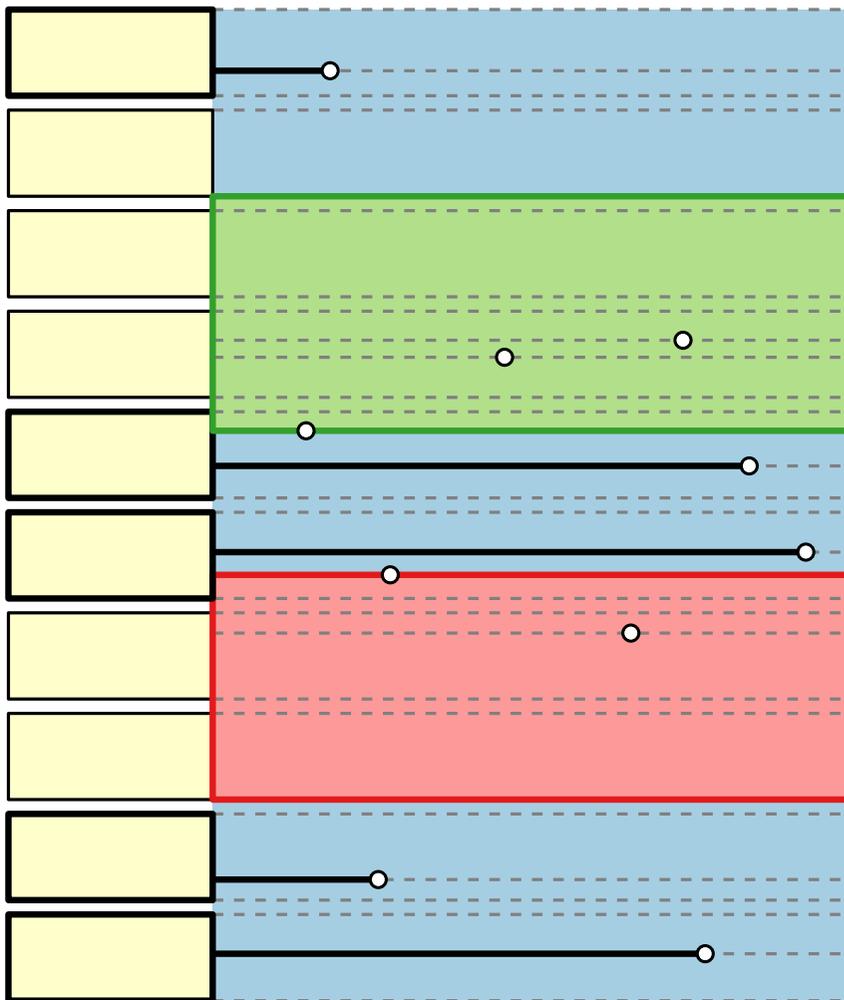
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

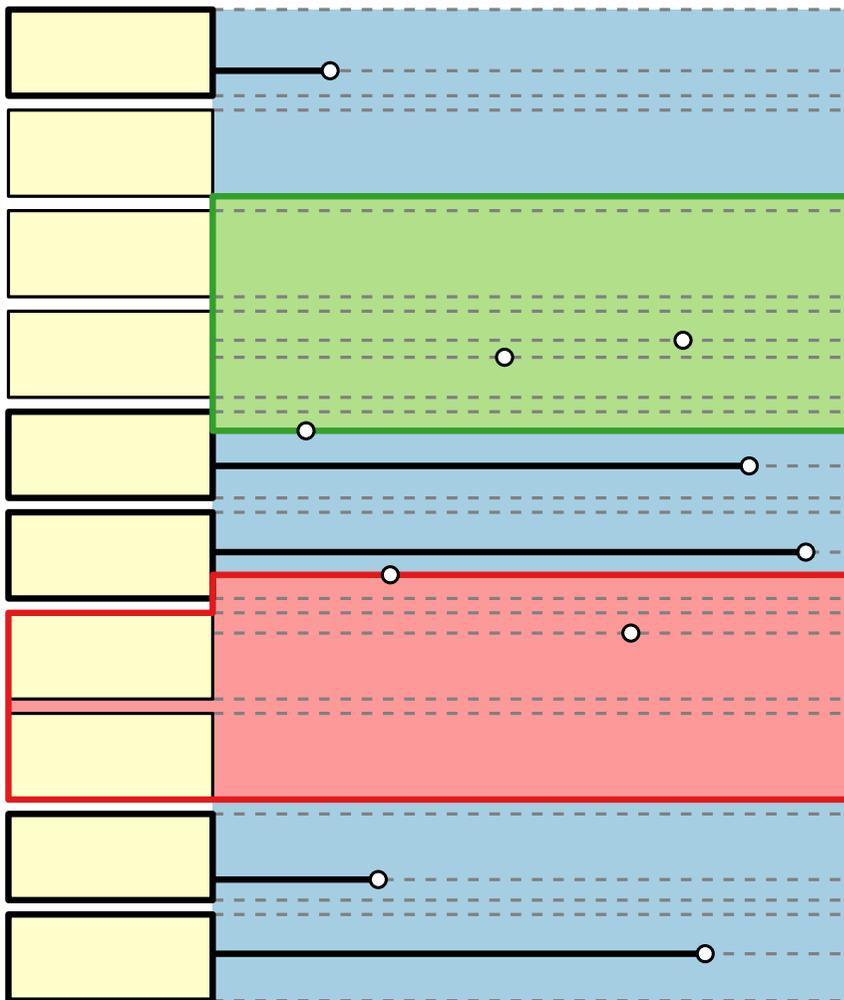
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

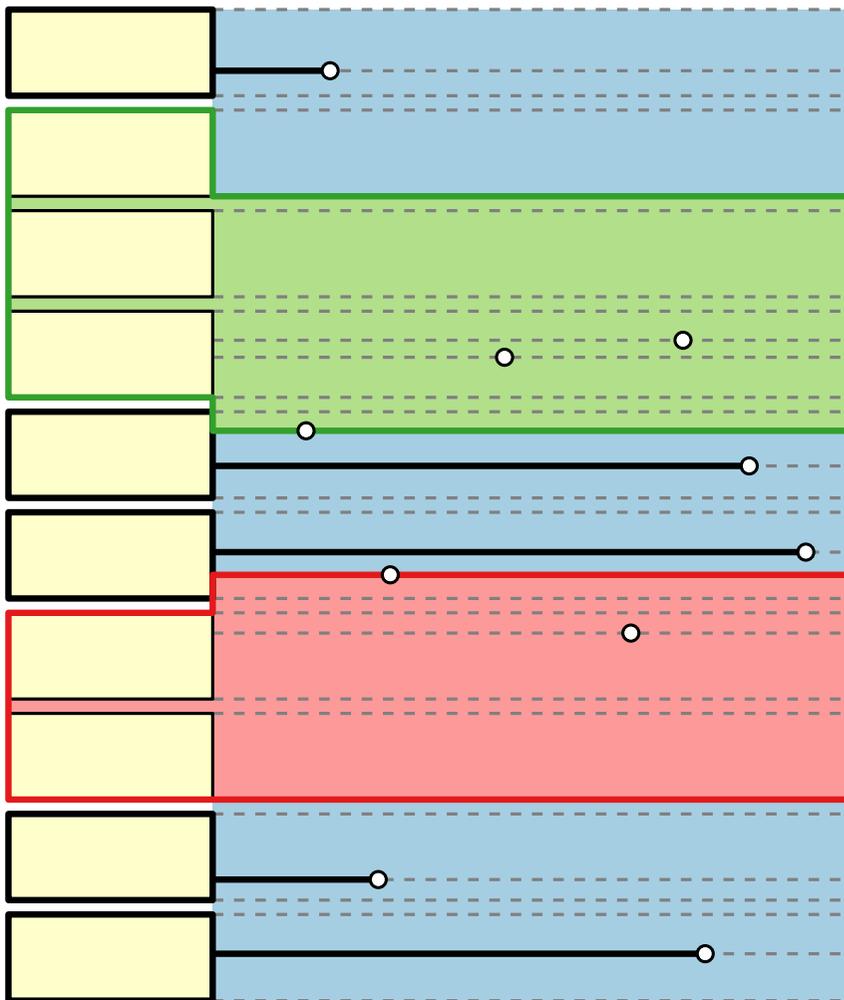
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
- **absteigend**, falls mehr Beschriftungen.

ohne Rand



Sweepline-Verfahren

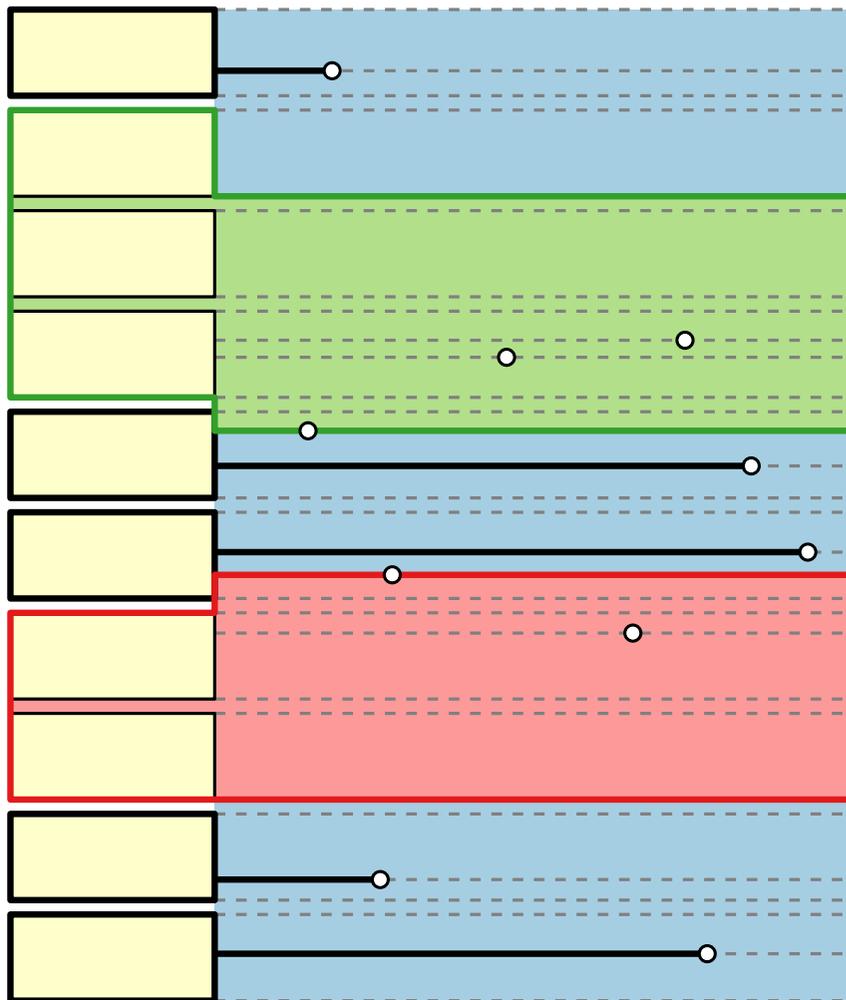
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline:
- **absteigend**, falls mehr Beschriftungen.
Sweepline:



Sweepline-Verfahren

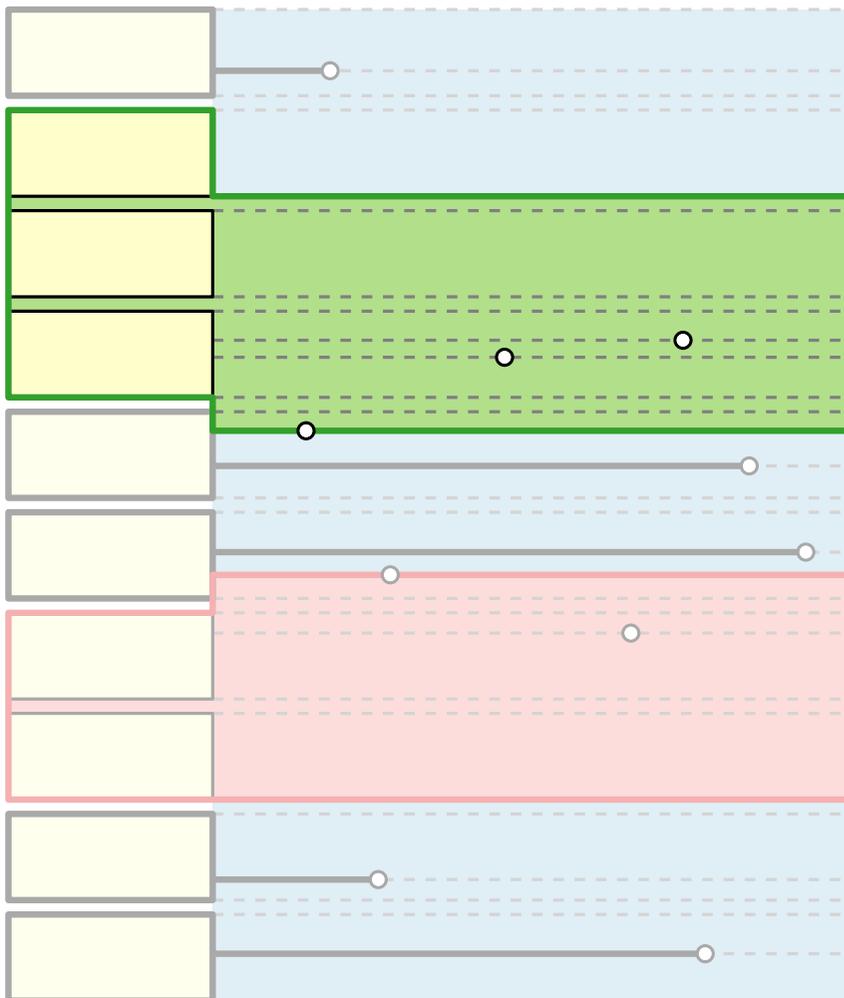
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline:
- **absteigend**, falls mehr Beschriftungen.
Sweepline:



Sweepline-Verfahren

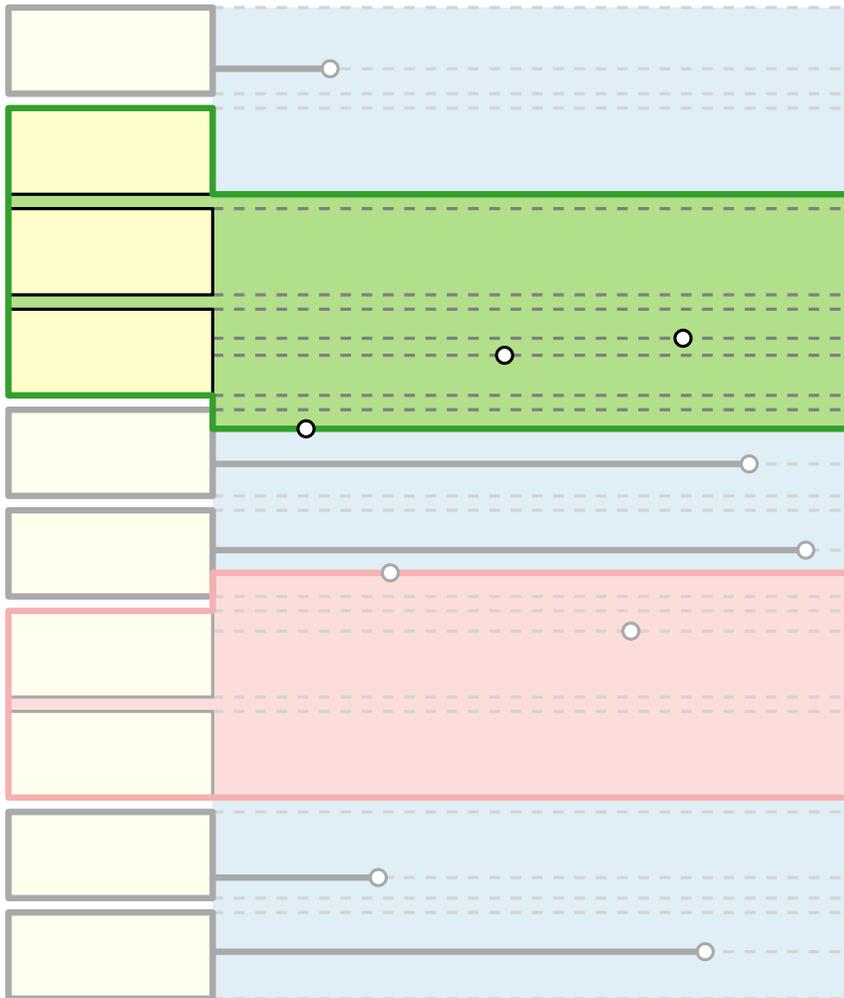
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*

ohne Rand



Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

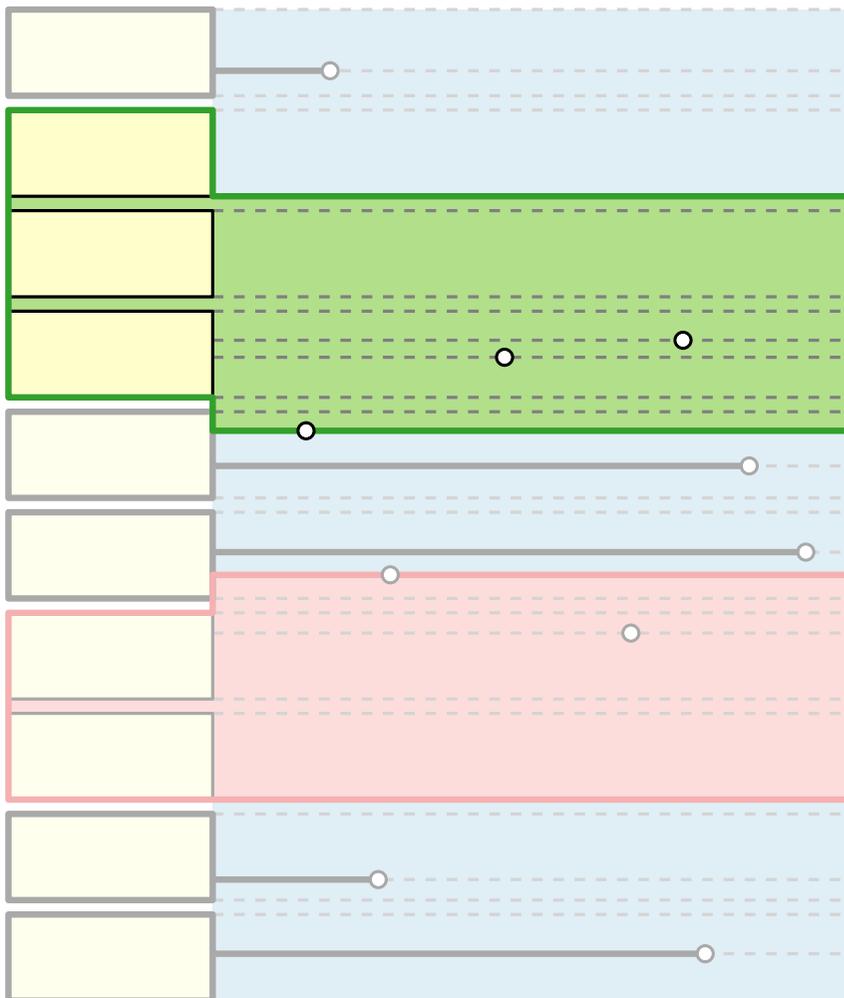
1. Unterteile Instanz in einfache Teilinstanzen.

2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linkensten Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linkensten Punkt mit oberster Beschriftung.



Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

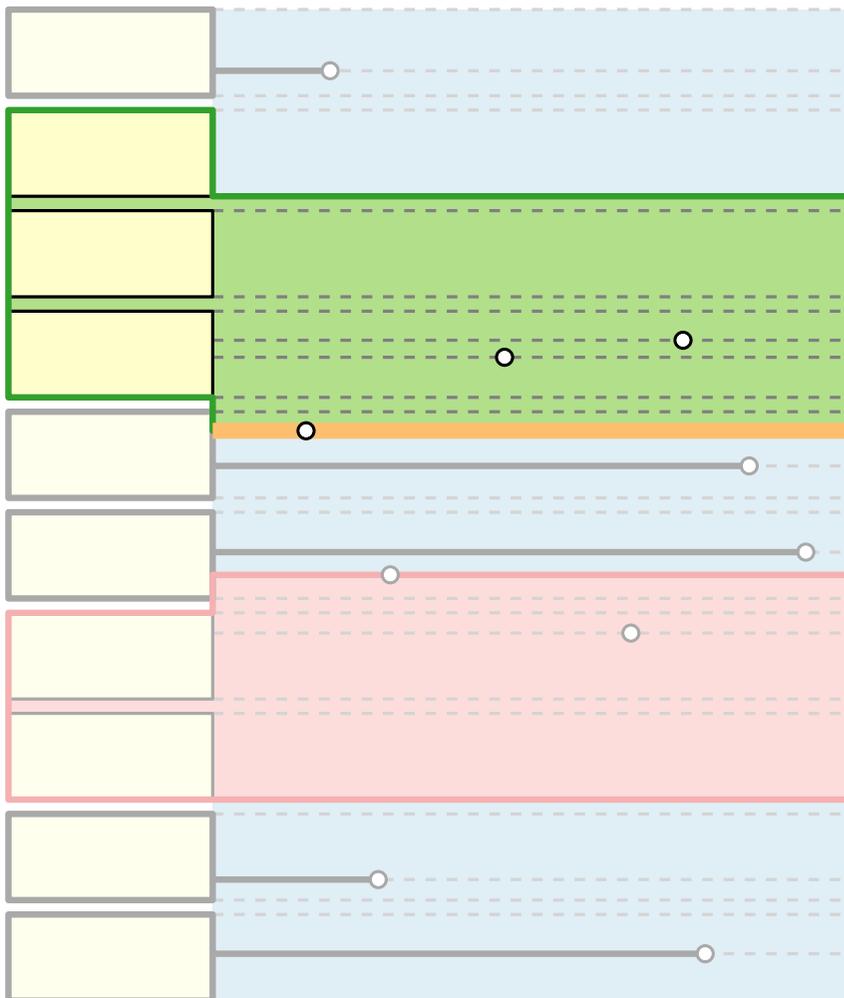
1. Unterteile Instanz in einfache Teilinstanzen.

2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linkensten Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linkensten Punkt mit oberster Beschriftung.



Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

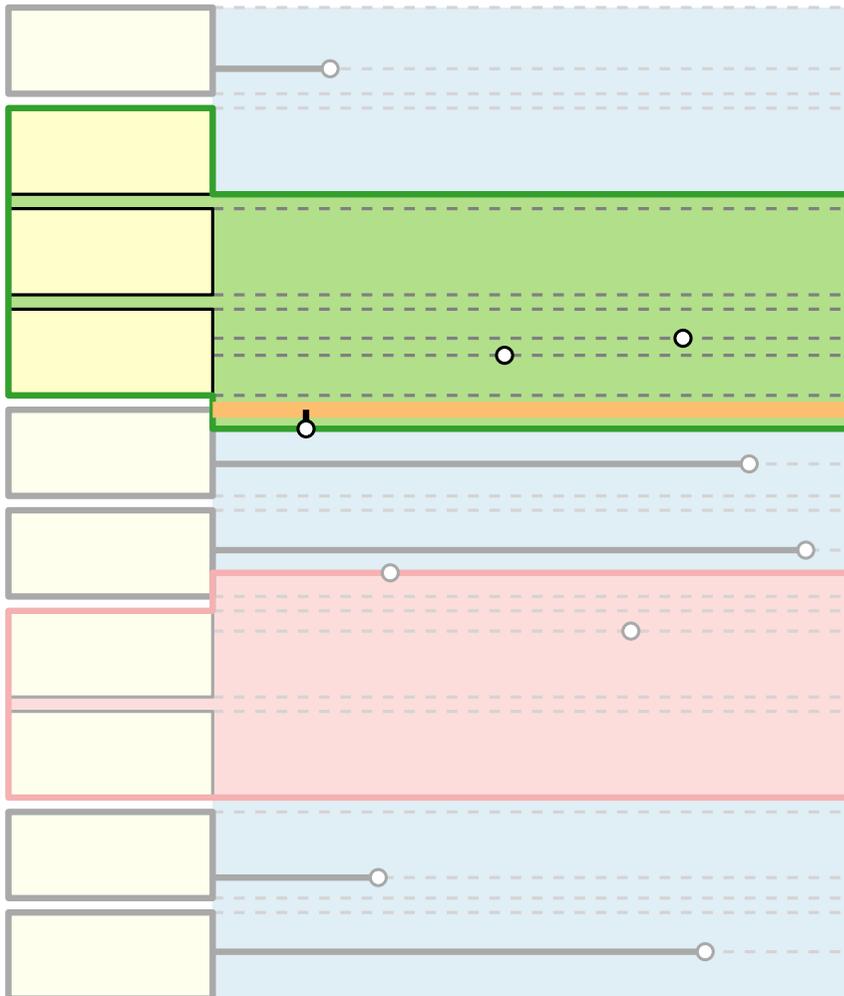
1. Unterteile Instanz in einfache Teilinstanzen.

2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linkensten Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linkensten Punkt mit oberster Beschriftung.



Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

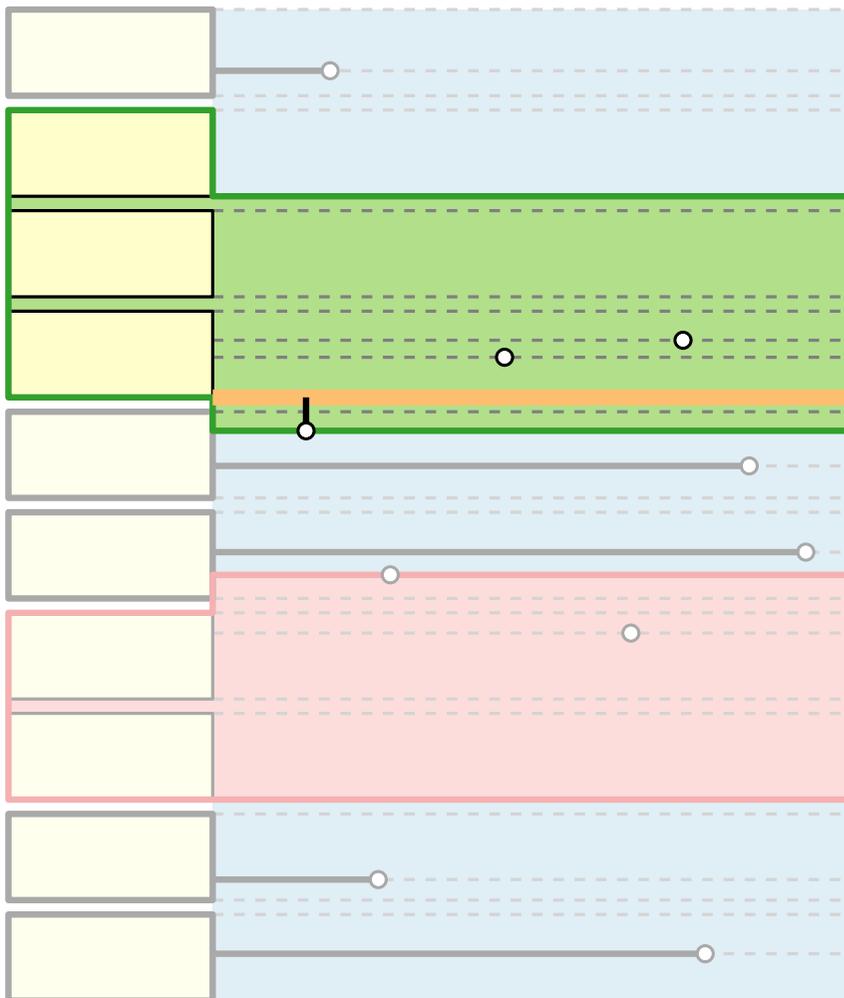
1. Unterteile Instanz in einfache Teilinstanzen.

2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linken Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linken Punkt mit oberster Beschriftung.



Sweepline-Verfahren

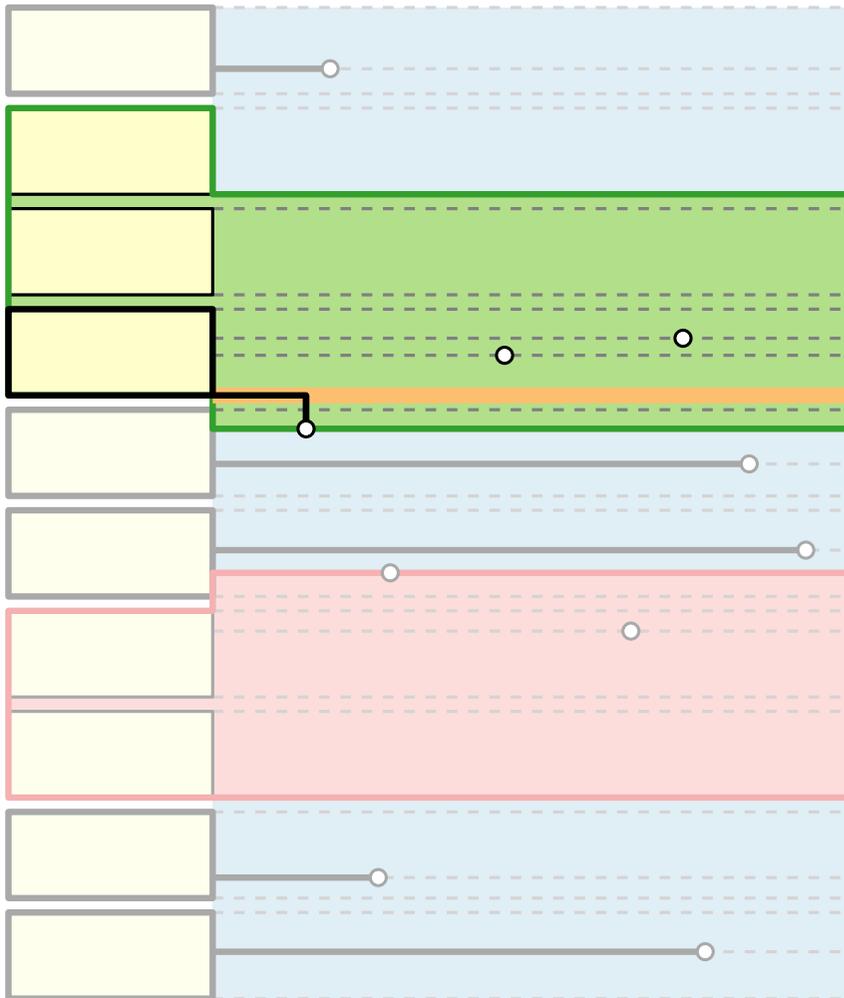
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linkensten Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linkensten Punkt mit oberster Beschriftung.



Sweepline-Verfahren

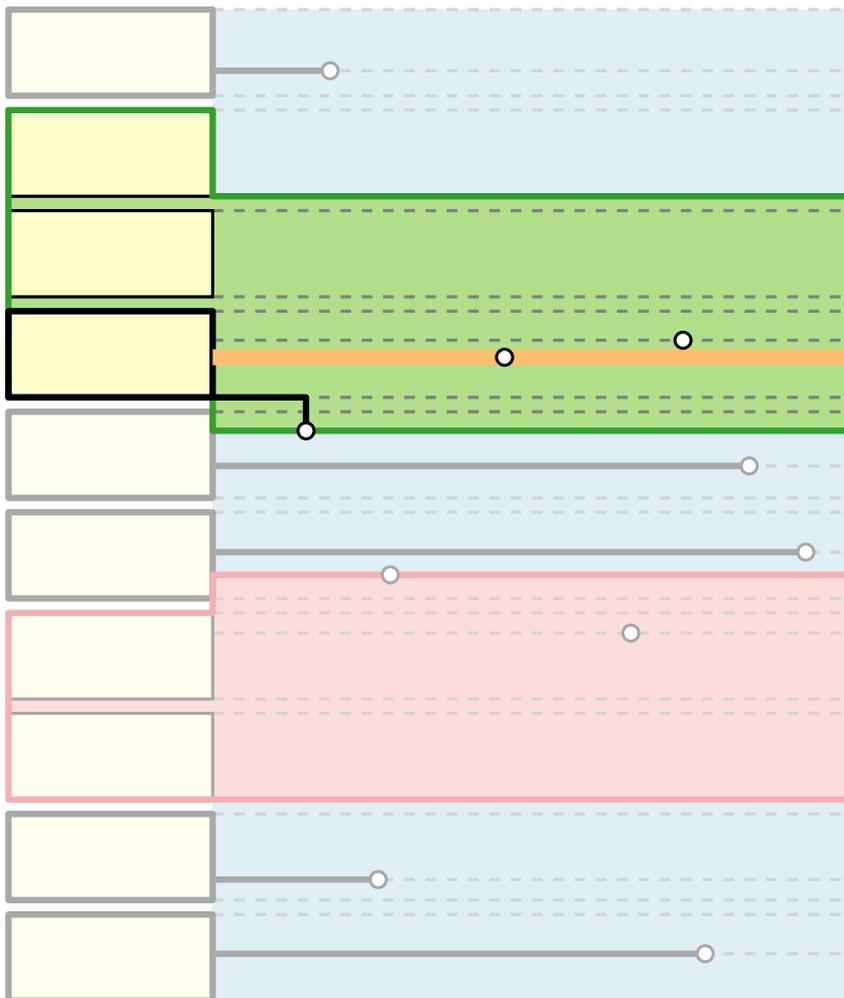
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linkensten Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linkensten Punkt mit oberster Beschriftung.



Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

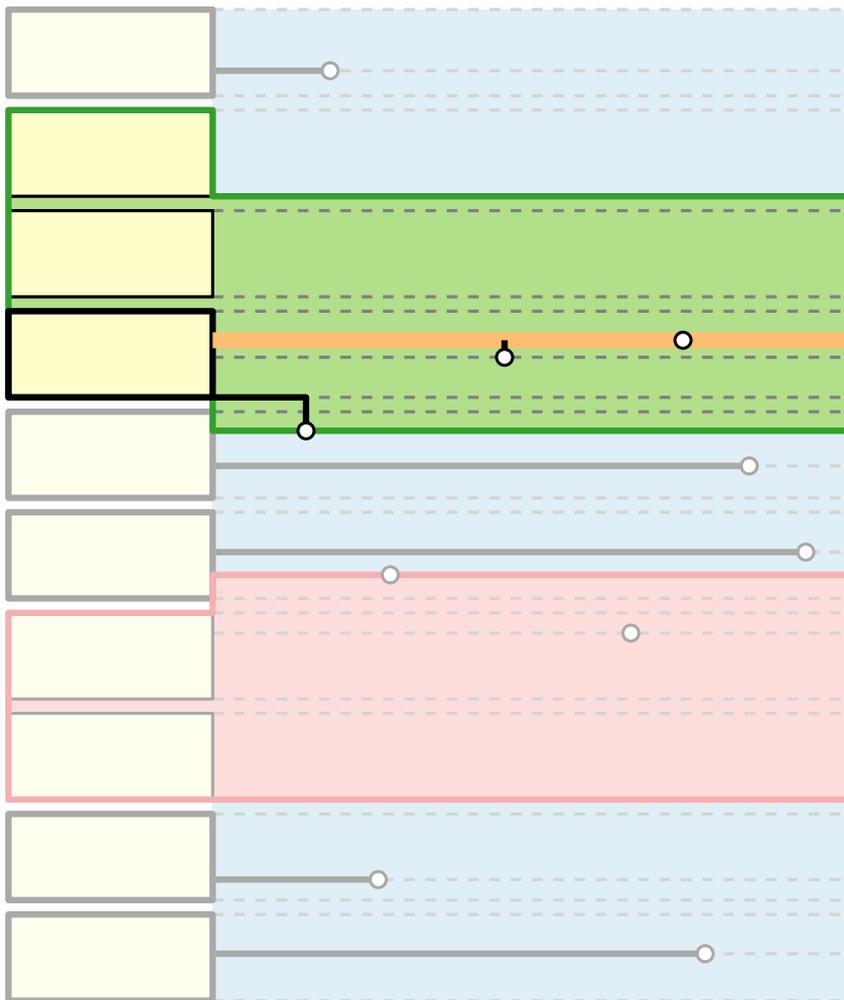
1. Unterteile Instanz in einfache Teilinstanzen.

2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linkensten Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linkensten Punkt mit oberster Beschriftung.



Sweepline-Verfahren

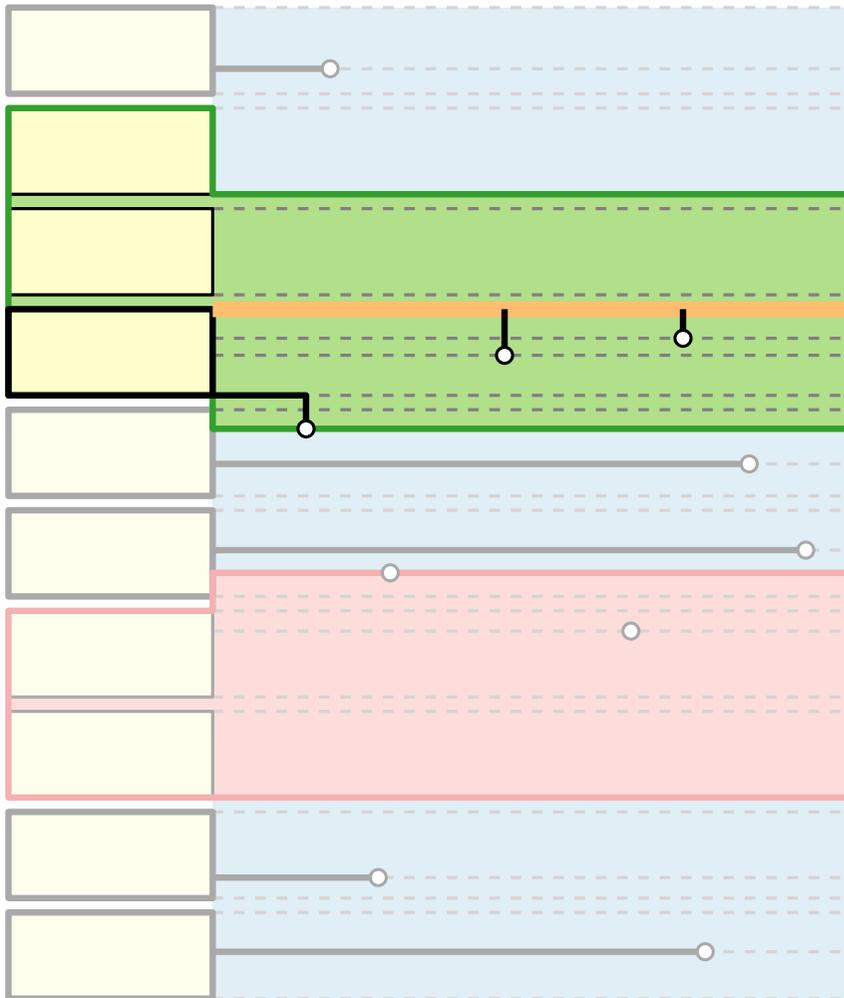
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linkensten Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linkensten Punkt mit oberster Beschriftung.



Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

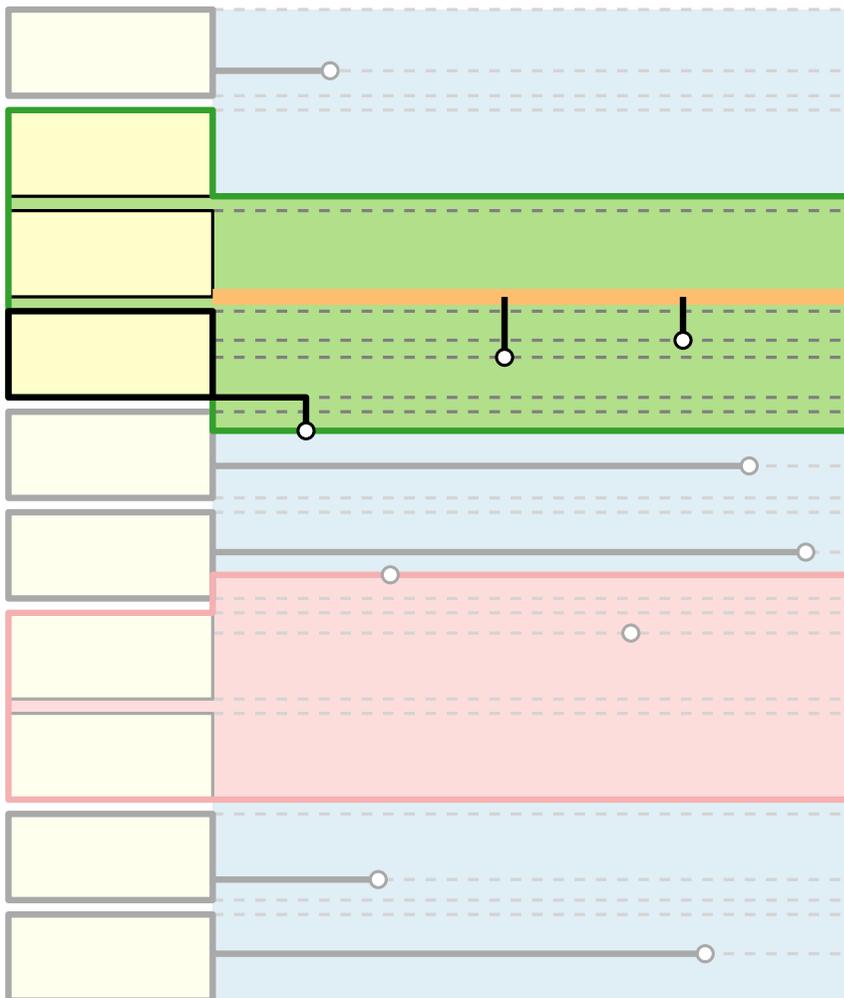
1. Unterteile Instanz in einfache Teilinstanzen.

2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linkensten Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linkensten Punkt mit oberster Beschriftung.



Sweepline-Verfahren

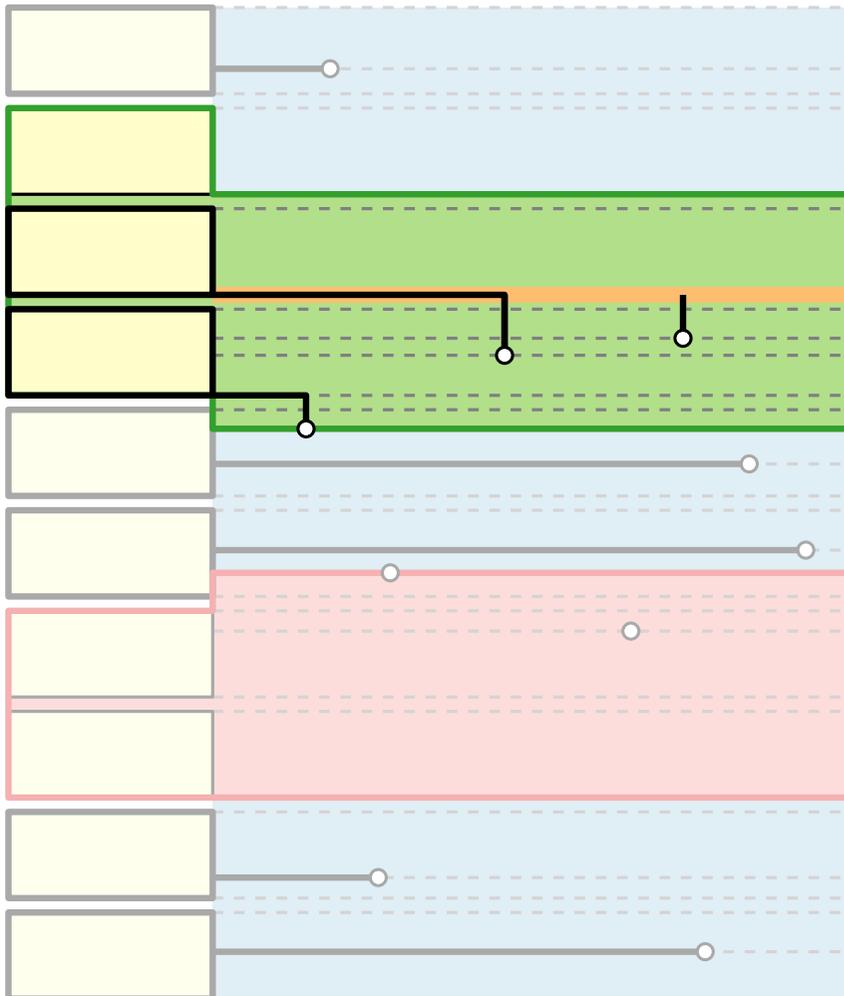
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linkensten Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linkensten Punkt mit oberster Beschriftung.



Sweepline-Verfahren

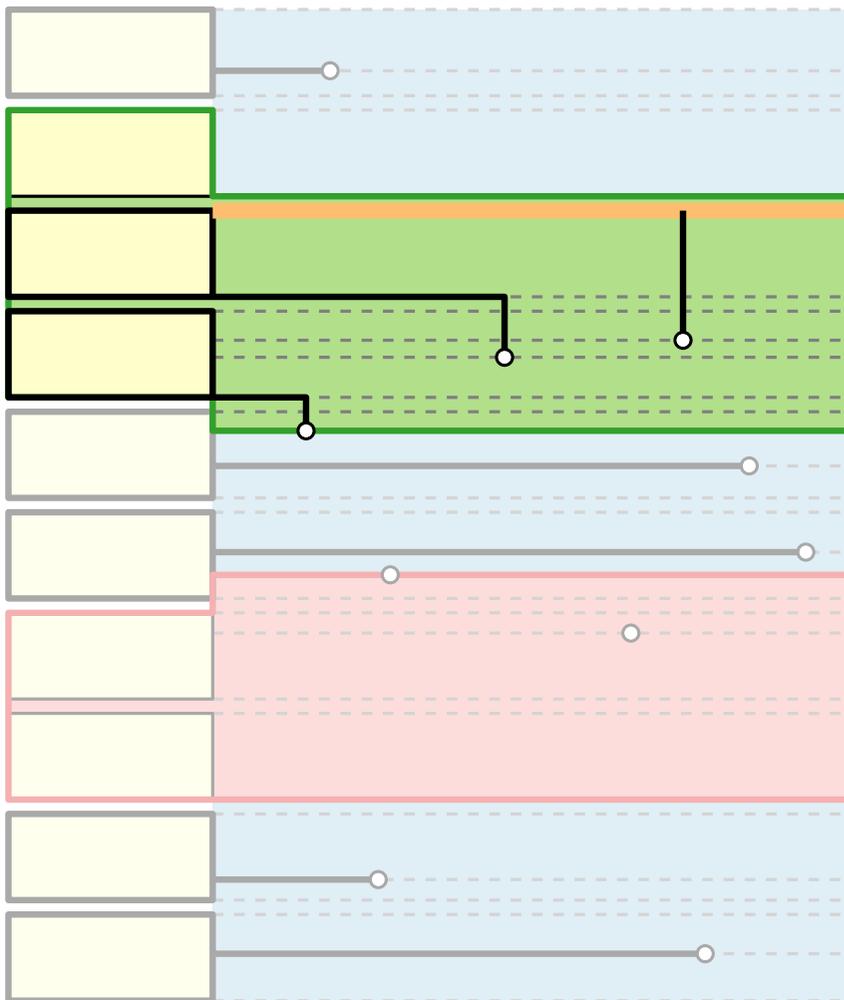
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linken Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linken Punkt mit oberster Beschriftung.



Sweepline-Verfahren

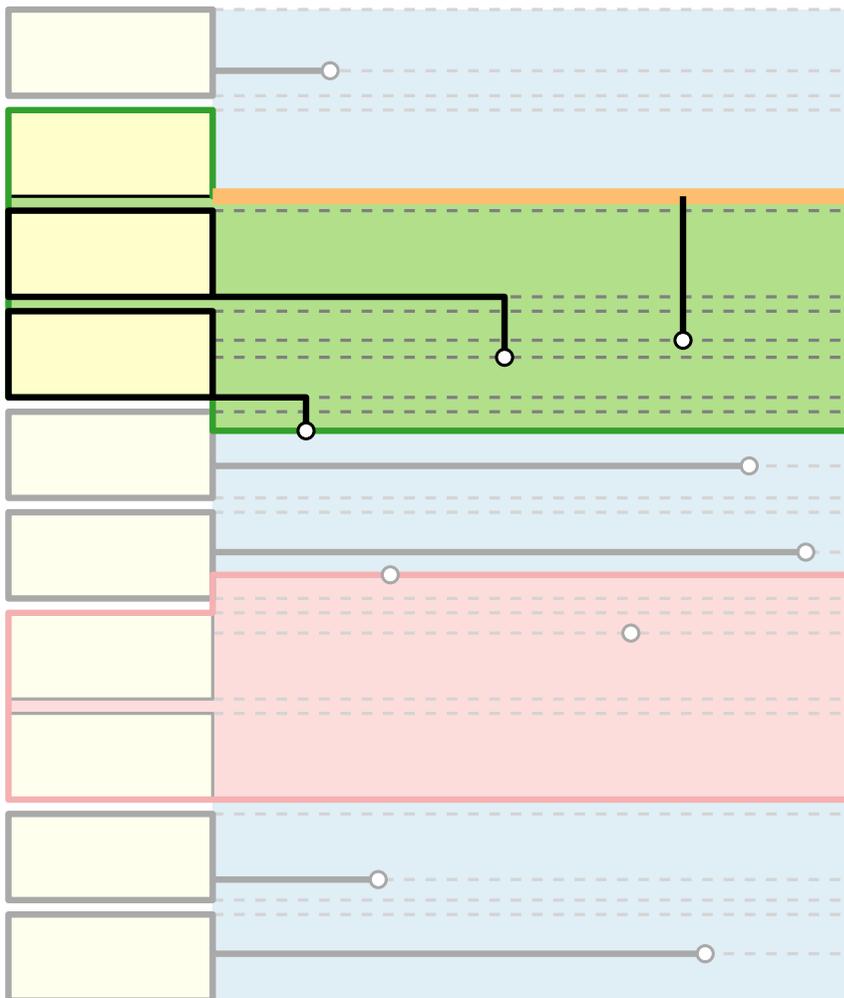
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linken Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linken Punkt mit oberster Beschriftung.



Sweepline-Verfahren

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

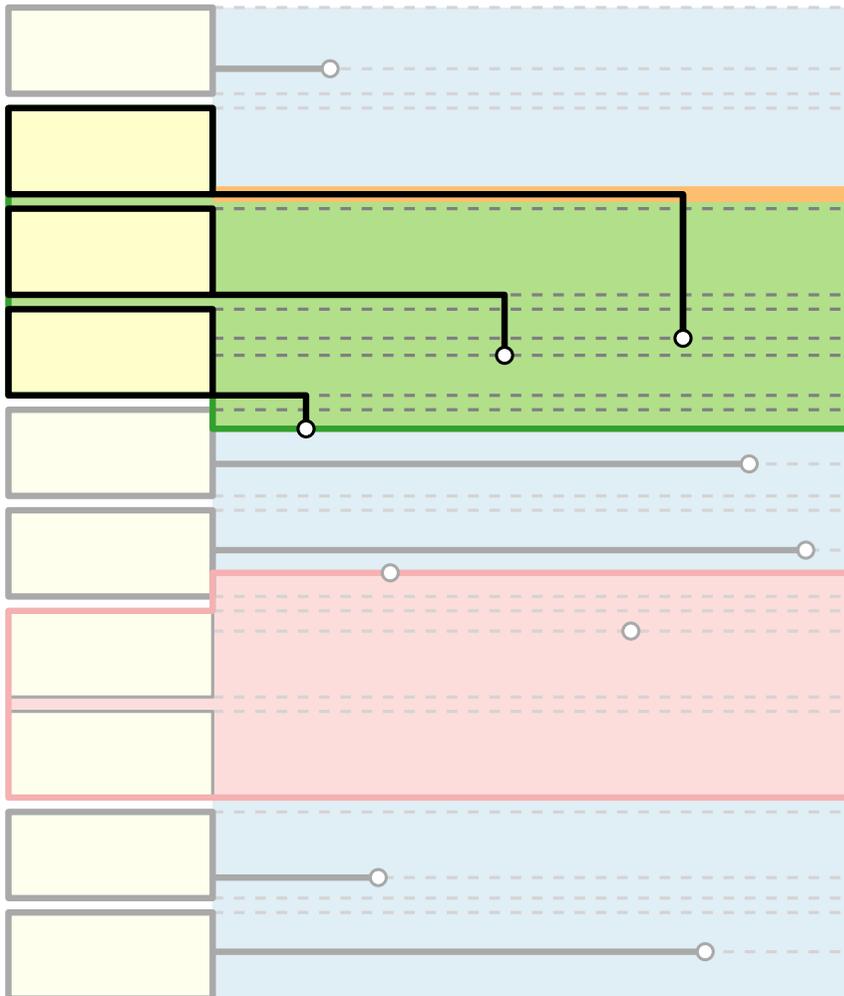
1. Unterteile Instanz in einfache Teilinstanzen.

2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linken Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linken Punkt mit oberster Beschriftung.



Sweepline-Verfahren

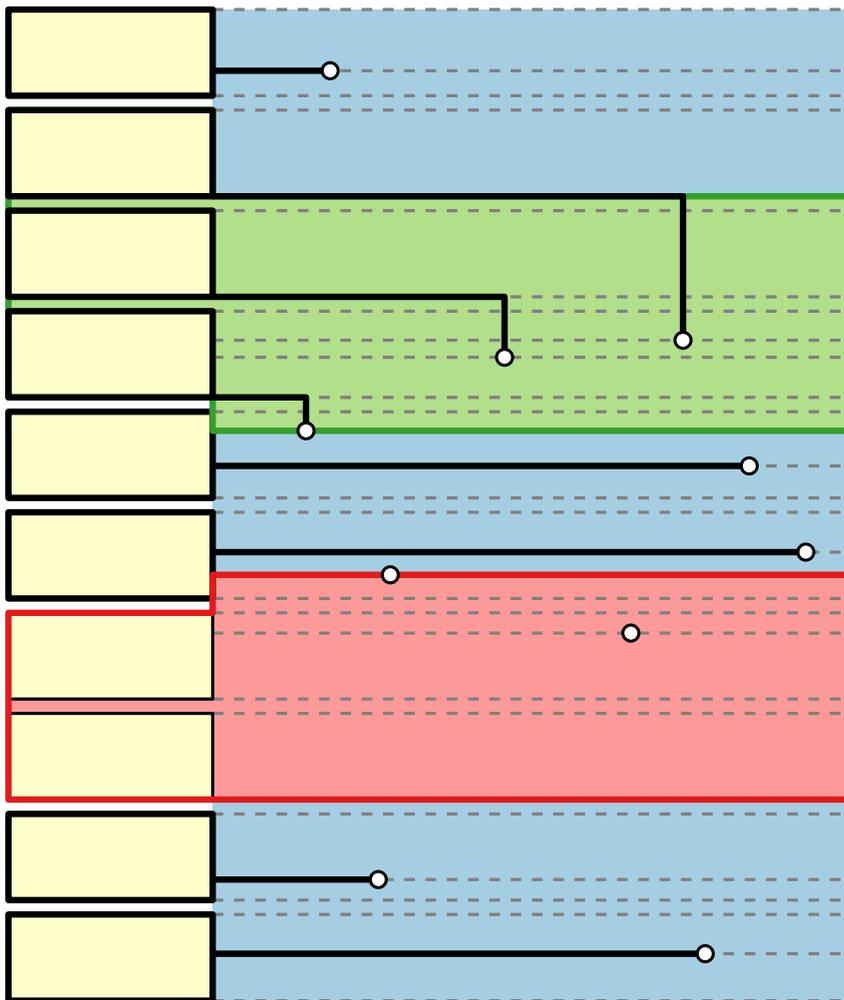
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linken Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linken Punkt mit oberster Beschriftung.



Sweepline-Verfahren

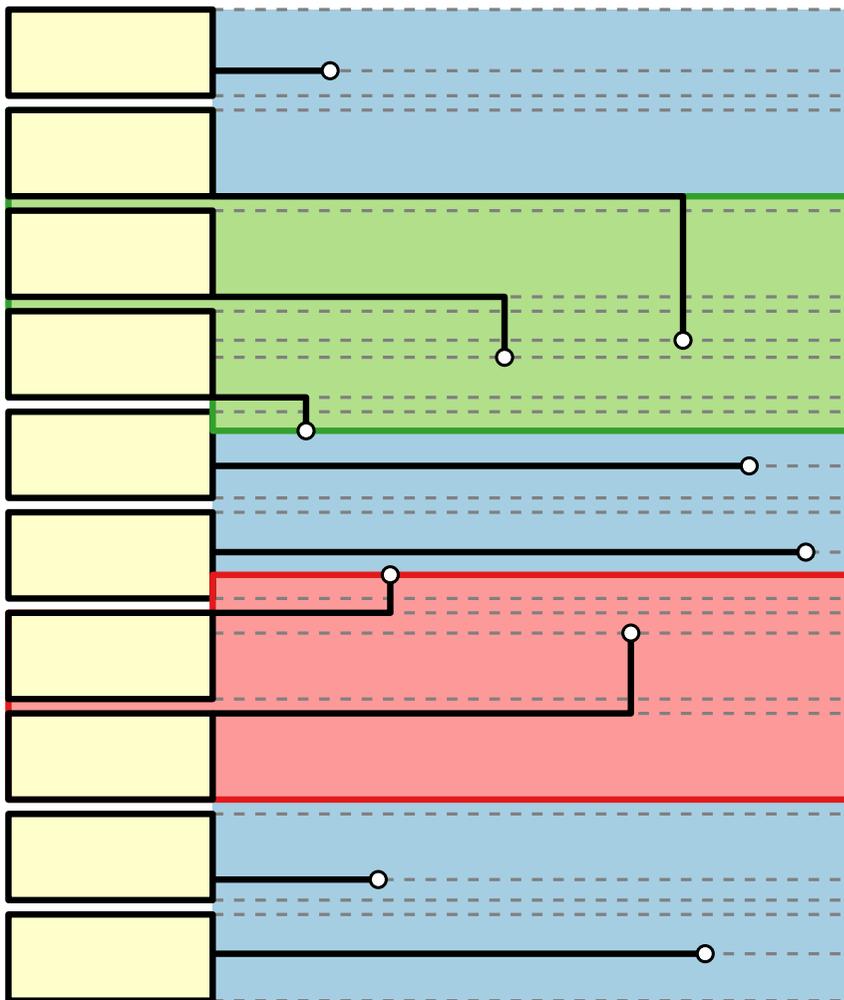
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in einfache Teilinstanzen.
2. Löse Teilinstanzen.

Für jeden Streifen, zähle Anzahl Punkte und Beschriftungen im Streifen oder darunter. Ein Streifen ist...

ohne Rand

- **neutral**, falls gleiche Anzahl.
Verbinde Punkte horizontal.
- **aufsteigend**, falls mehr Punkte.
Sweepline: *von unten nach oben*
Verbinde linken Punkt mit unterster Beschriftung.
- **absteigend**, falls mehr Beschriftungen.
Sweepline: *von oben nach unten*
Verbinde linken Punkt mit oberster Beschriftung.



Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Sweep-line-Verfahren

- Lemma.** Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.
- Beweis.** Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

Sweep-line-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

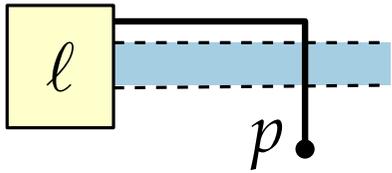
1. Fall. ℓ umschließt σ

Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ

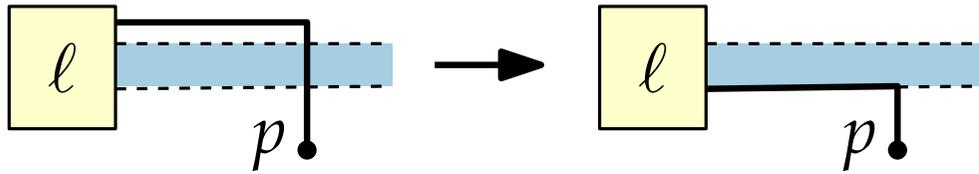


Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ

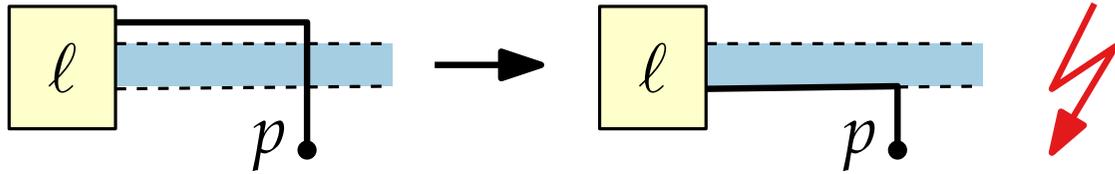


Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ

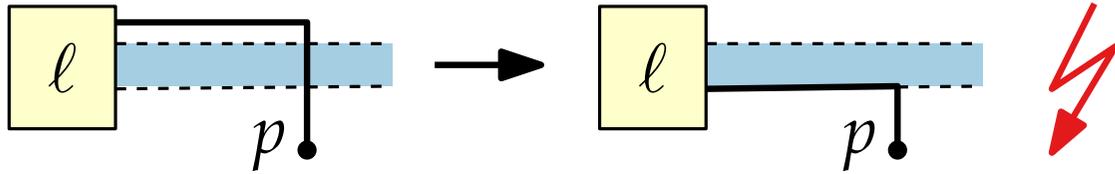


Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ



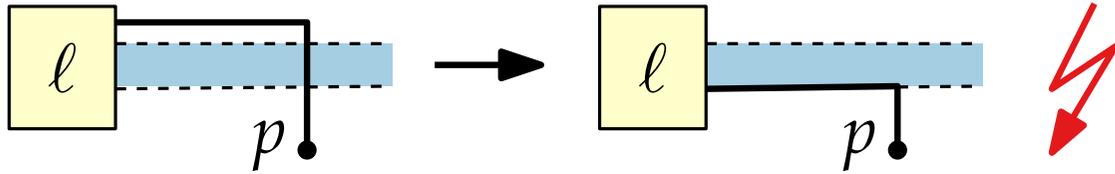
2. Fall. p oberhalb, ℓ unterhalb von σ .

Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ



2. Fall. p oberhalb, ℓ unterhalb von σ .

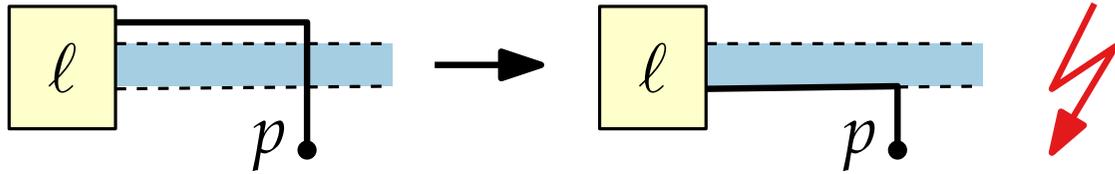
3. Fall. p unterhalb, ℓ oberhalb von σ .

Sweepline-Verfahren

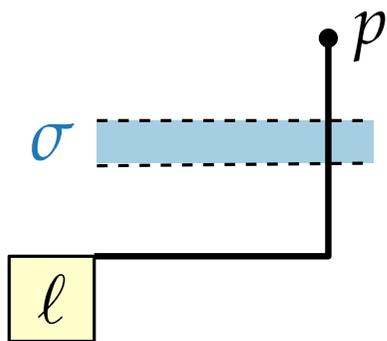
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ



2. Fall. p oberhalb, ℓ unterhalb von σ .



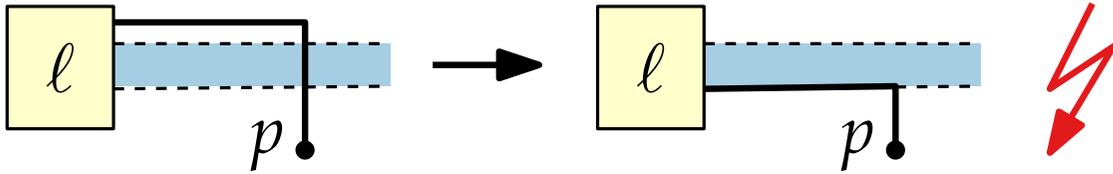
3. Fall. p unterhalb, ℓ oberhalb von σ .

Sweepline-Verfahren

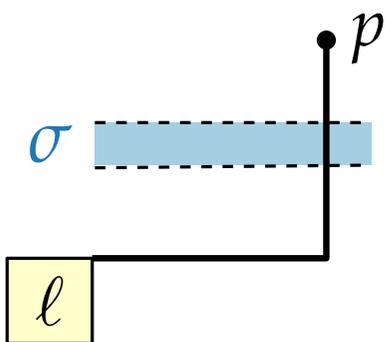
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ



2. Fall. p oberhalb, ℓ unterhalb von σ .



Punkte = # Beschriftungen bis σ

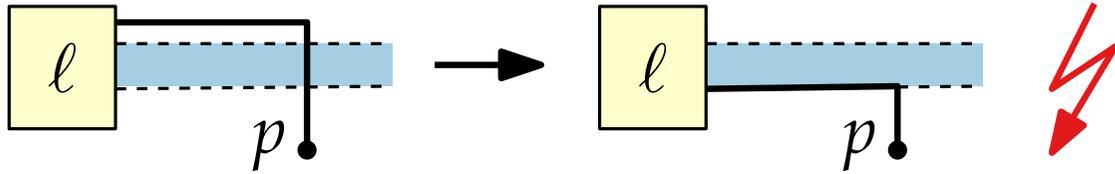
3. Fall. p unterhalb, ℓ oberhalb von σ .

Sweepline-Verfahren

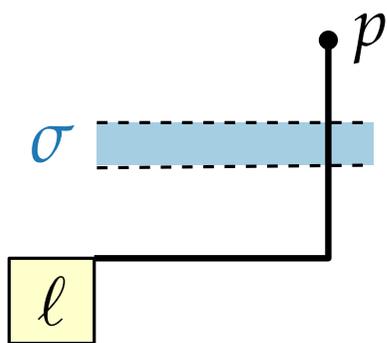
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ



2. Fall. p oberhalb, ℓ unterhalb von σ .



Punkte = # Beschriftungen bis σ
 $\Rightarrow \exists q$ in oder unter σ
mit Beschriftung über σ

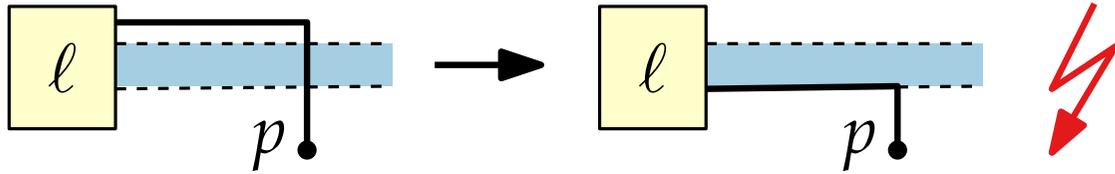
3. Fall. p unterhalb, ℓ oberhalb von σ .

Sweepline-Verfahren

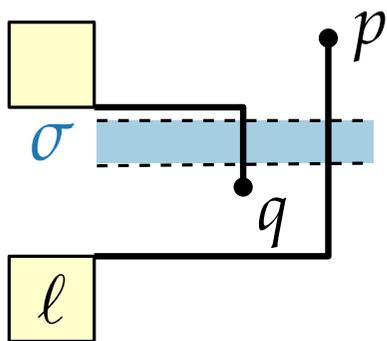
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ



2. Fall. p oberhalb, ℓ unterhalb von σ .



Punkte = # Beschriftungen bis σ
 $\Rightarrow \exists q$ in oder unter σ
mit Beschriftung über σ

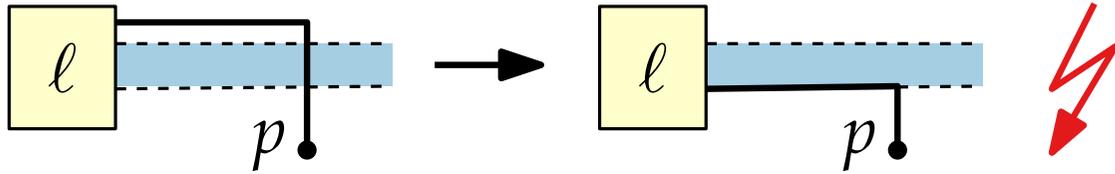
3. Fall. p unterhalb, ℓ oberhalb von σ .

Sweepline-Verfahren

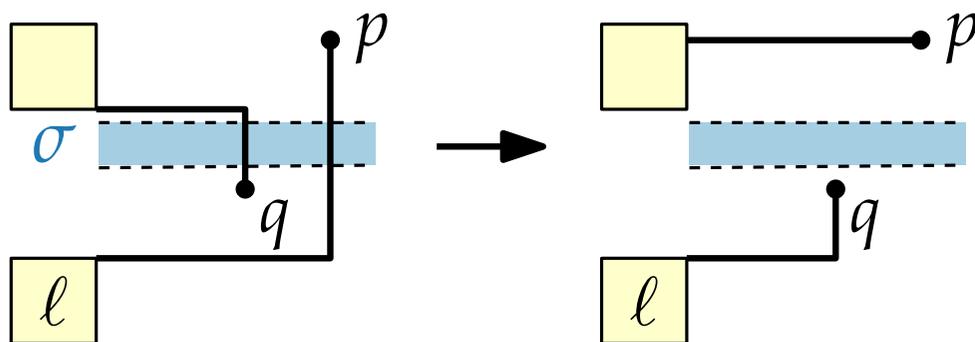
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ



2. Fall. p oberhalb, ℓ unterhalb von σ .



Punkte = # Beschriftungen bis σ
 $\Rightarrow \exists q$ in oder unter σ
 mit Beschriftung über σ

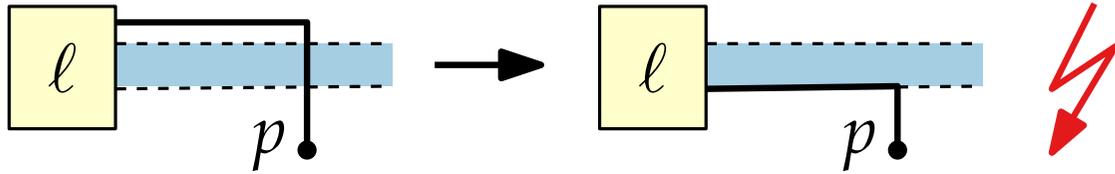
3. Fall. p unterhalb, ℓ oberhalb von σ .

Sweepline-Verfahren

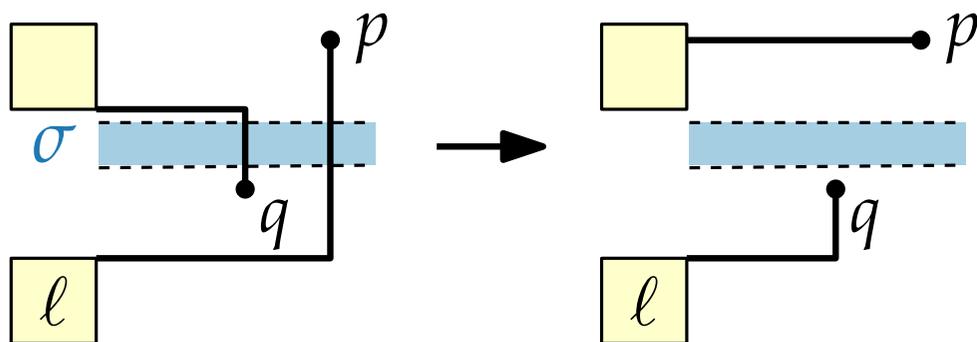
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Beweis. Angenommen Leader (p, ℓ) kreuzt neutralen Streifen σ .

1. Fall. ℓ umschließt σ



2. Fall. p oberhalb, ℓ unterhalb von σ .



Punkte = # Beschriftungen bis σ
 $\Rightarrow \exists q$ in oder unter σ
mit Beschriftung über σ

3. Fall. p unterhalb, ℓ oberhalb von σ .

Sweepline-Verfahren

- Lemma.** Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.
- Lemma.** Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

Sweep-line-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

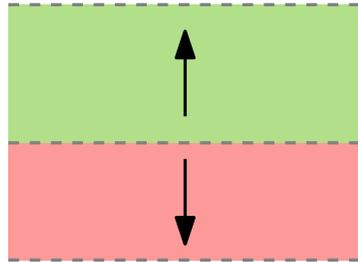
Beweis.

Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

Beweis.



Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

Beweis.

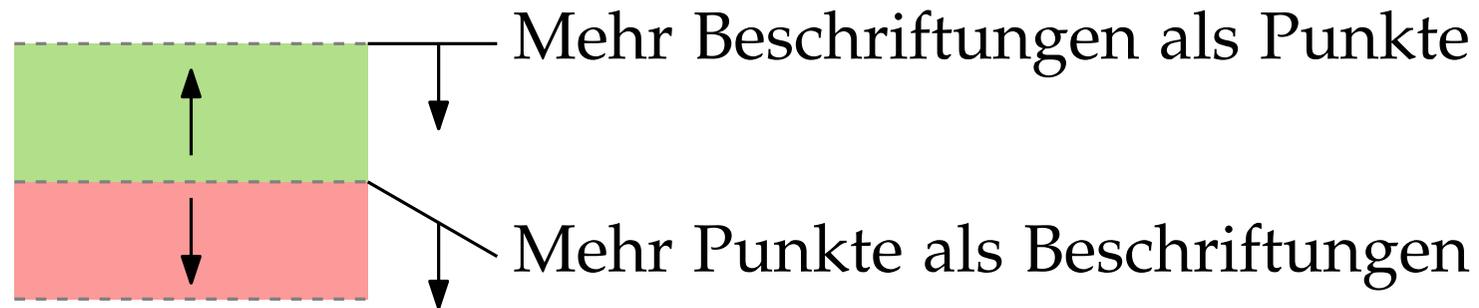


Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

Beweis.

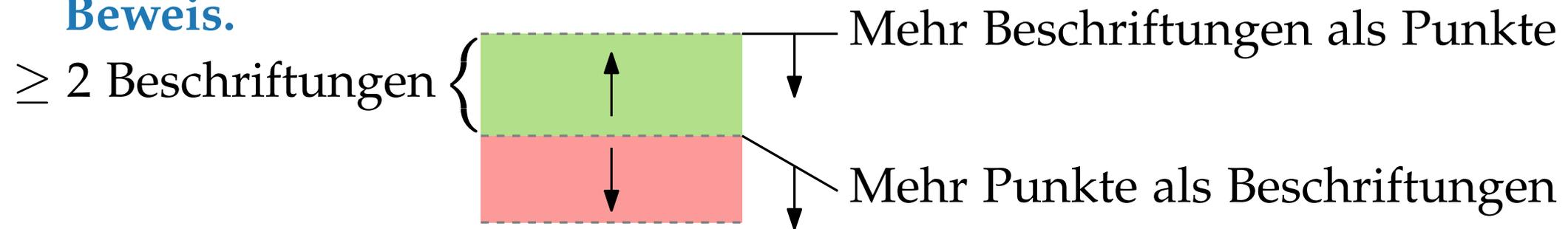


Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

Beweis.

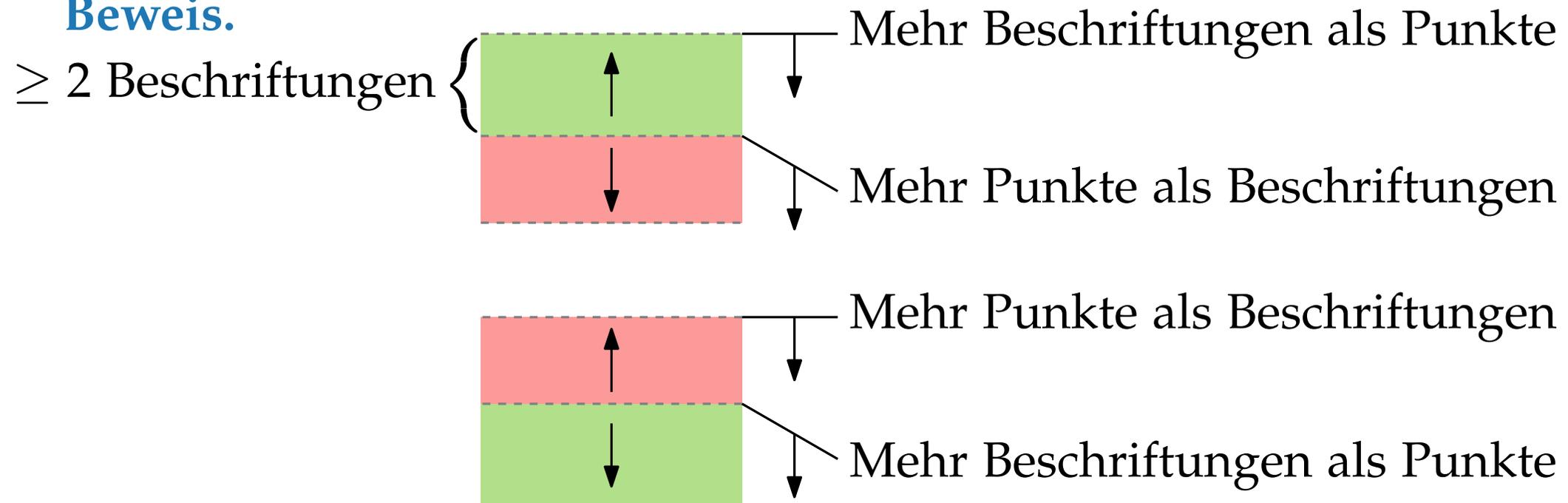


Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

Beweis.

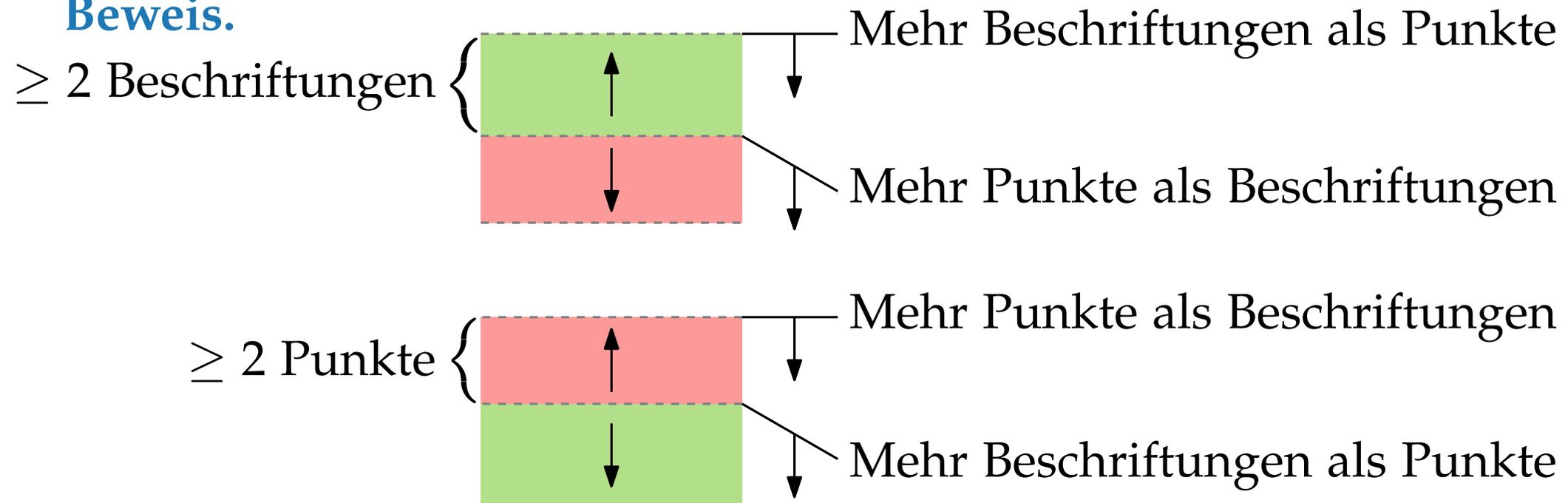


Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

Beweis.



Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Sweepline-Verfahren

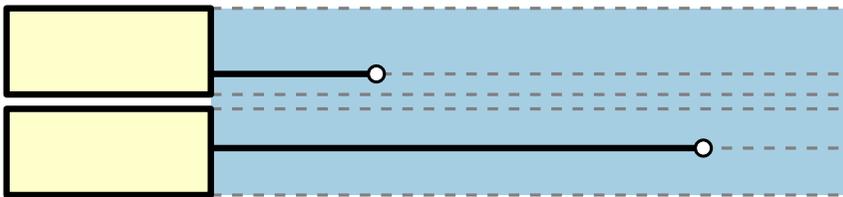
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Beweis.



Sweepline-Verfahren

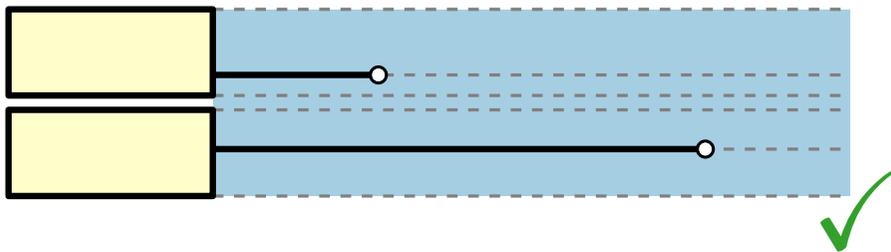
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Beweis.



Sweepline-Verfahren

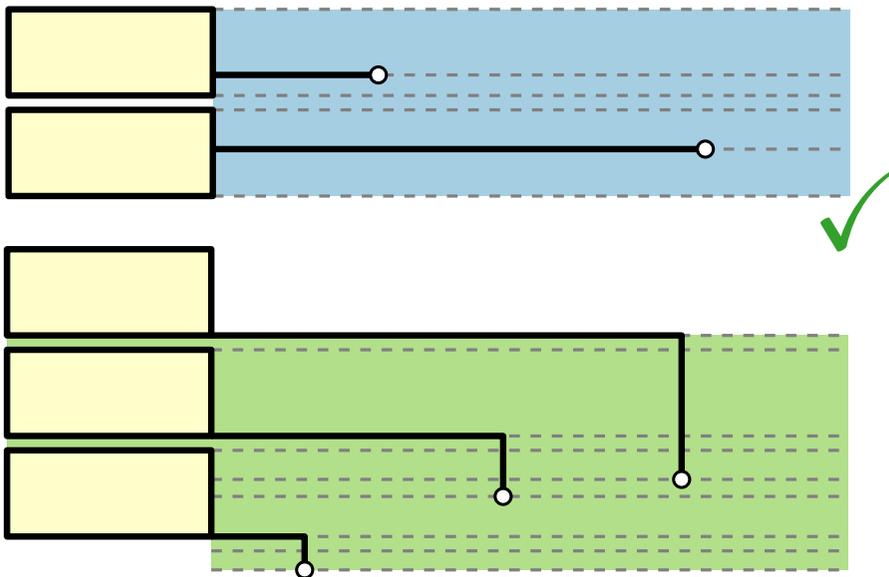
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Beweis.



Sweepline-Verfahren

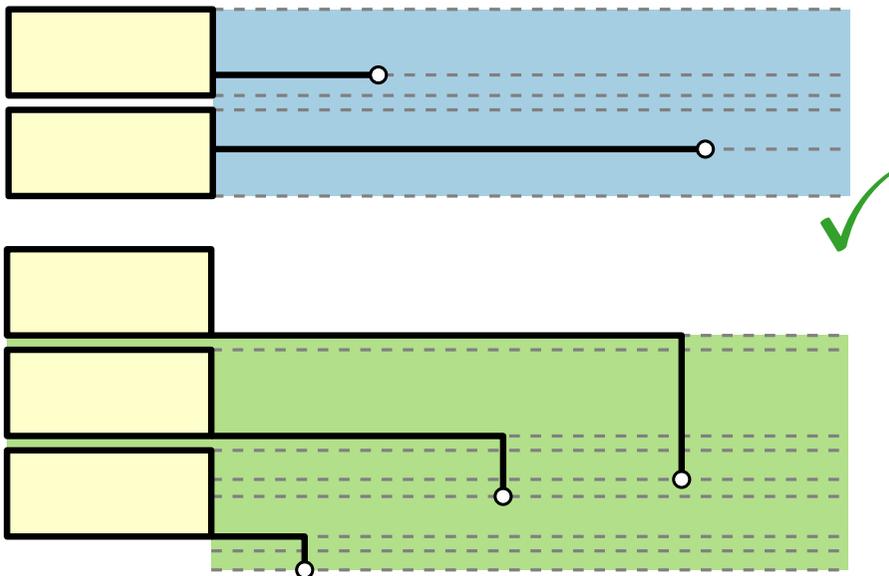
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Beweis.



Kreuzungsfrei:

Sweepline-Verfahren

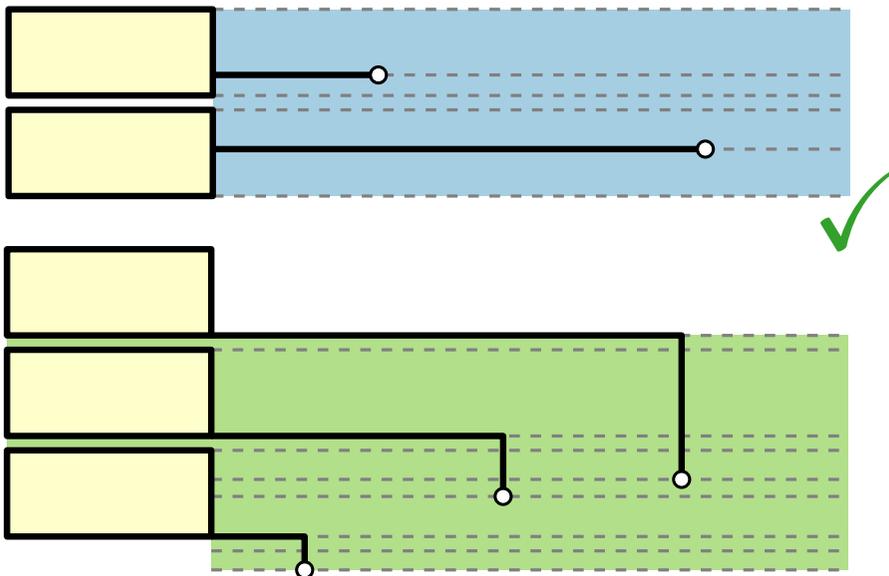
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Beweis.



Kreuzungsfrei:

Horizontale Segmente: Nur auf Sweepline, „linkester“ Punkt

Sweepline-Verfahren

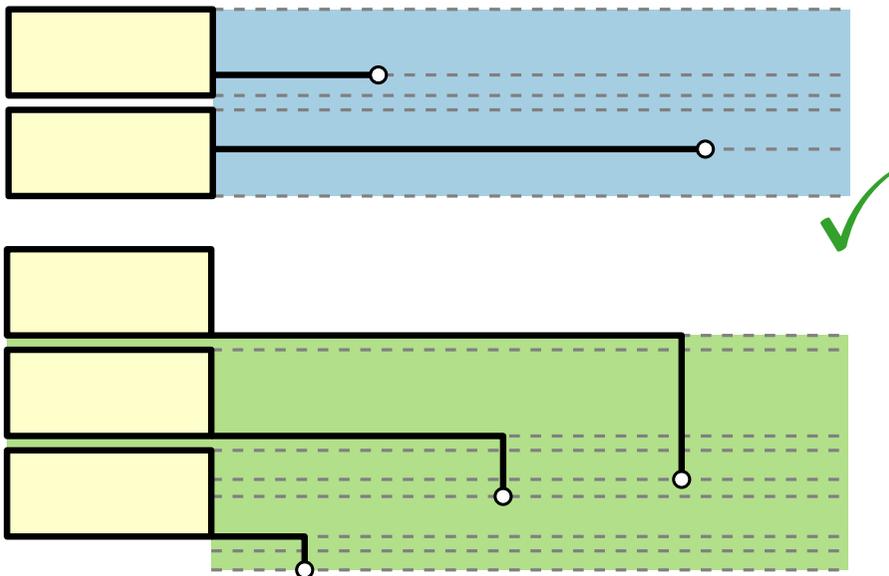
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Beweis.



Kreuzungsfrei: ✓

Horizontale Segmente: Nur auf Sweepline, „linkester“ Punkt

Sweepline-Verfahren

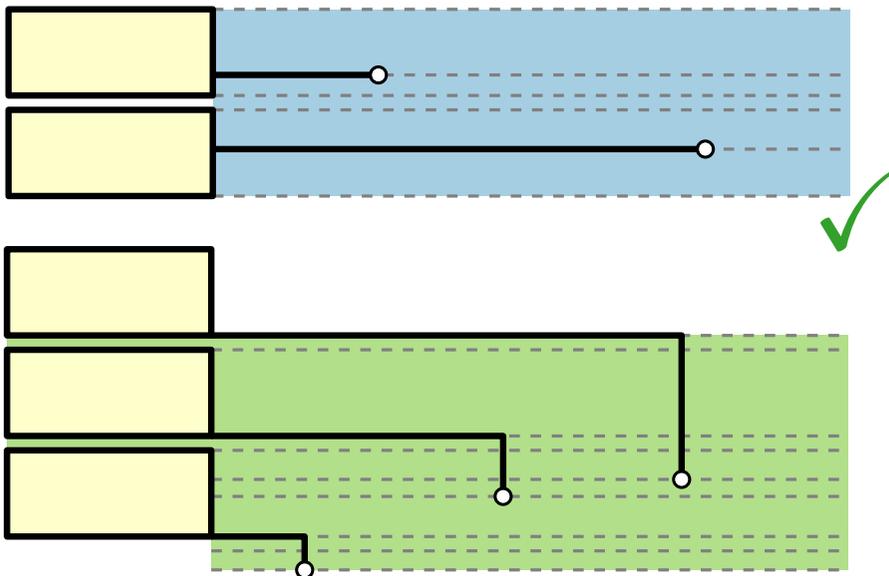
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Beweis.



Kreuzungsfrei: ✓

Horizontale Segmente: Nur auf Sweepline, „linkester“ Punkt

Minimale Länge:

Sweepline-Verfahren

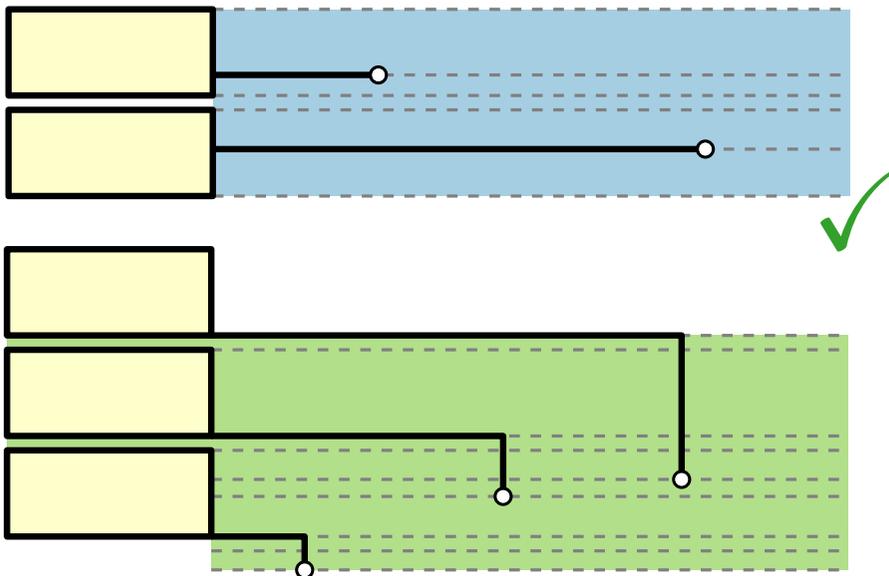
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Beweis.



Kreuzungsfrei: ✓

Horizontale Segmente: Nur auf Sweepline, „linkester“ Punkt

Minimale Länge:

Alle aufsteigenden Lösungen haben gleiche Länge!

Sweepline-Verfahren

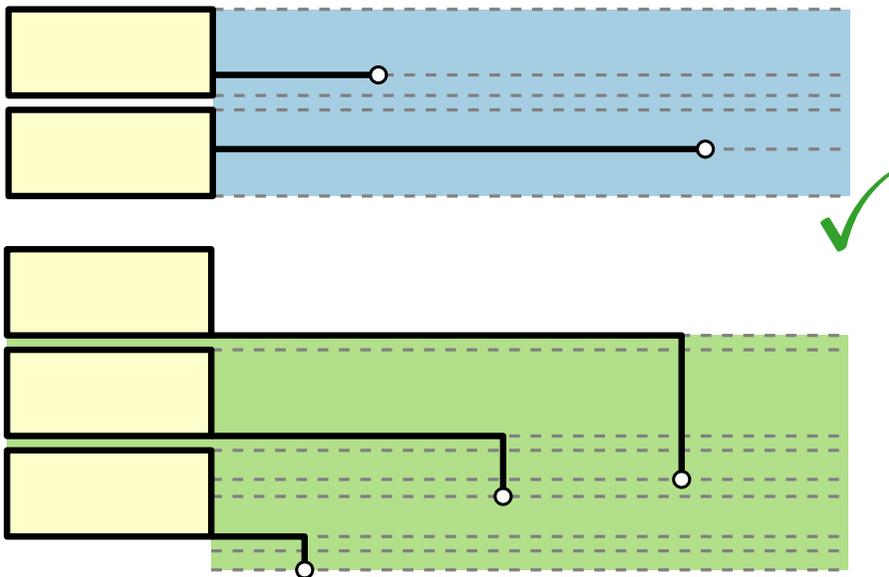
Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Beweis.



Kreuzungsfrei: ✓

Horizontale Segmente: Nur auf Sweepline, „linkester“ Punkt

Minimale Länge: ✓

Alle aufsteigenden Lösungen haben gleiche Länge!

Sweepline-Verfahren

Lemma. Für jede kreuzungsfreie Zuordnung minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt.

Lemma. Zwischen absteigendem Streifen und aufsteigendem Streifen befindet sich immer ein neutraler Streifen.

⇒ Jede optimale Lösung respektiert die Aufteilung in Teilinstanzen.

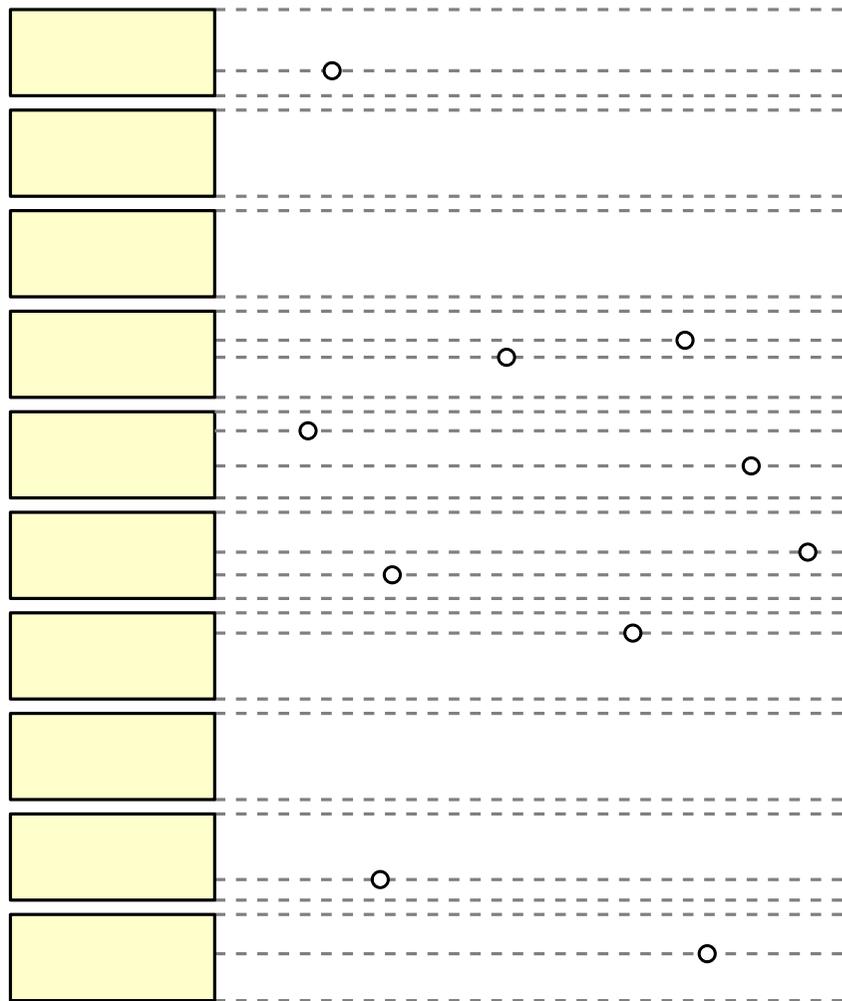
Lemma. Für jede Teilinstanz berechnet der Algorithmus eine kreuzungsfreie Lösung minimaler Länge.

Satz. Eine kreuzungsfreie Beschriftung minimaler Länge im 1-seitigen po-Leader-Model mit freien Ports kann in $O(n \log n)$ Zeit gefunden werden.

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.

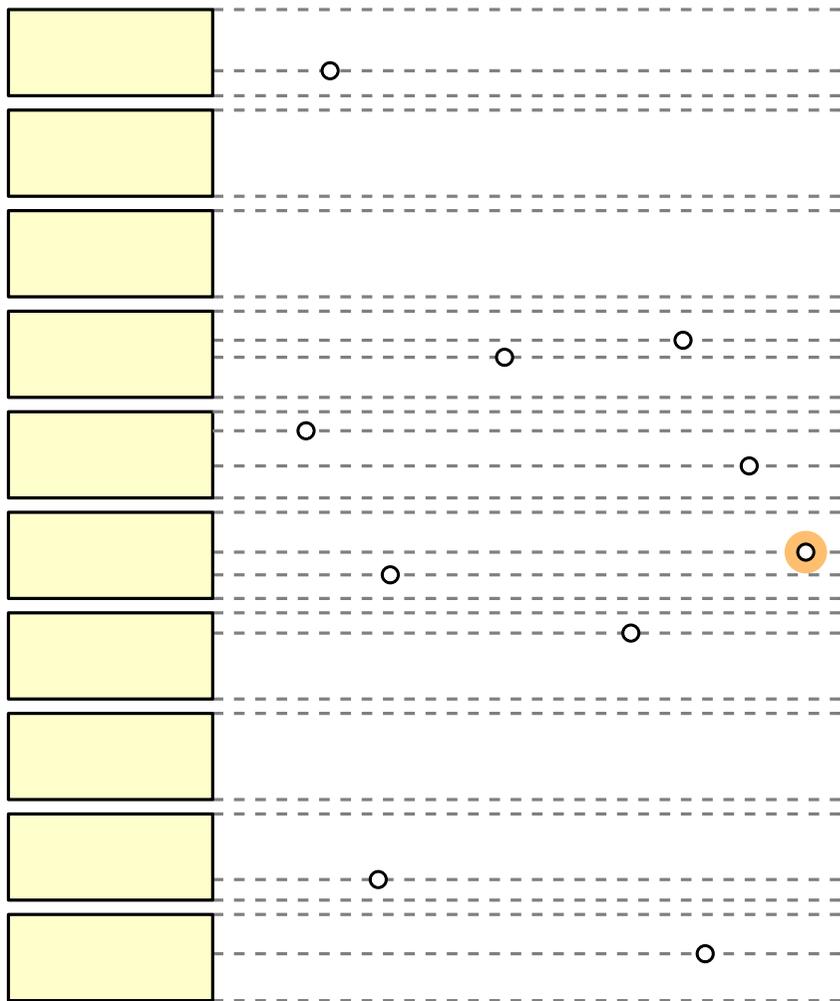


Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.

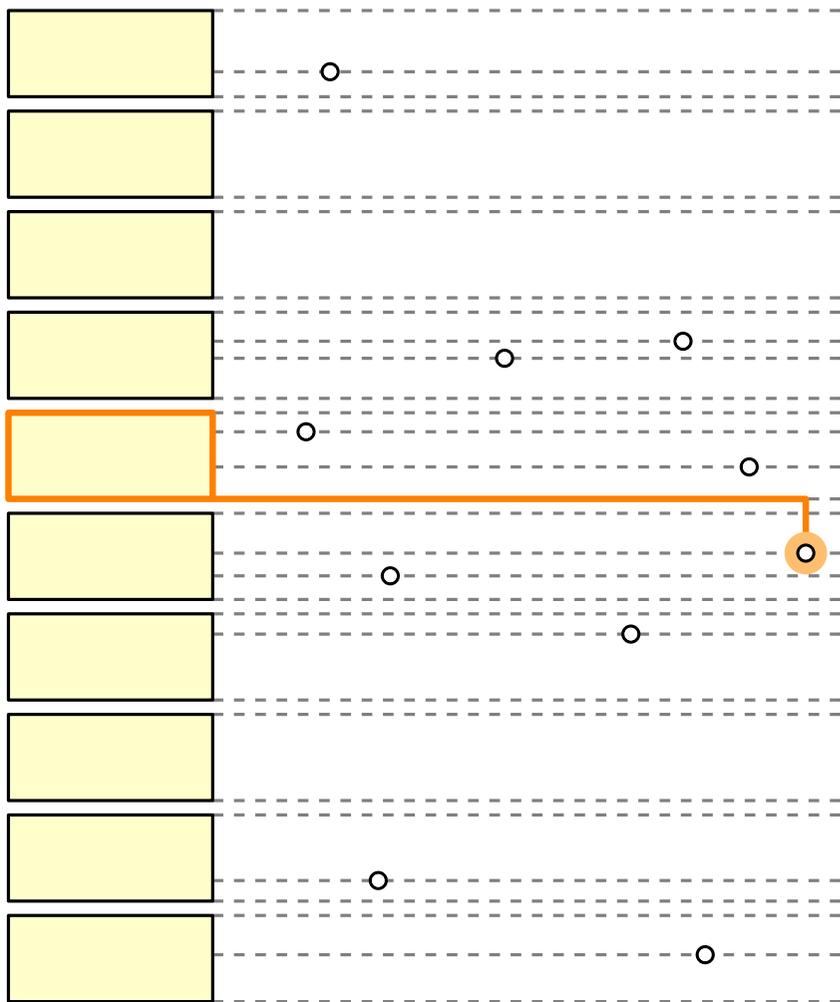
Betrachte rechten Punkt.



Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



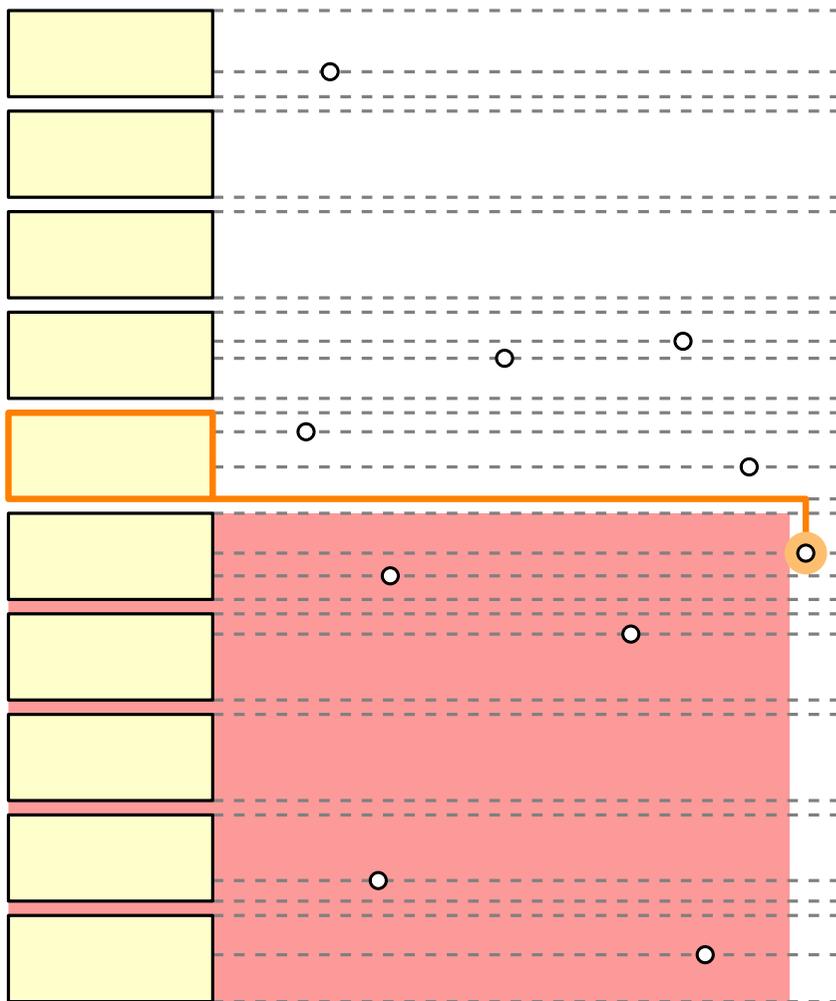
Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



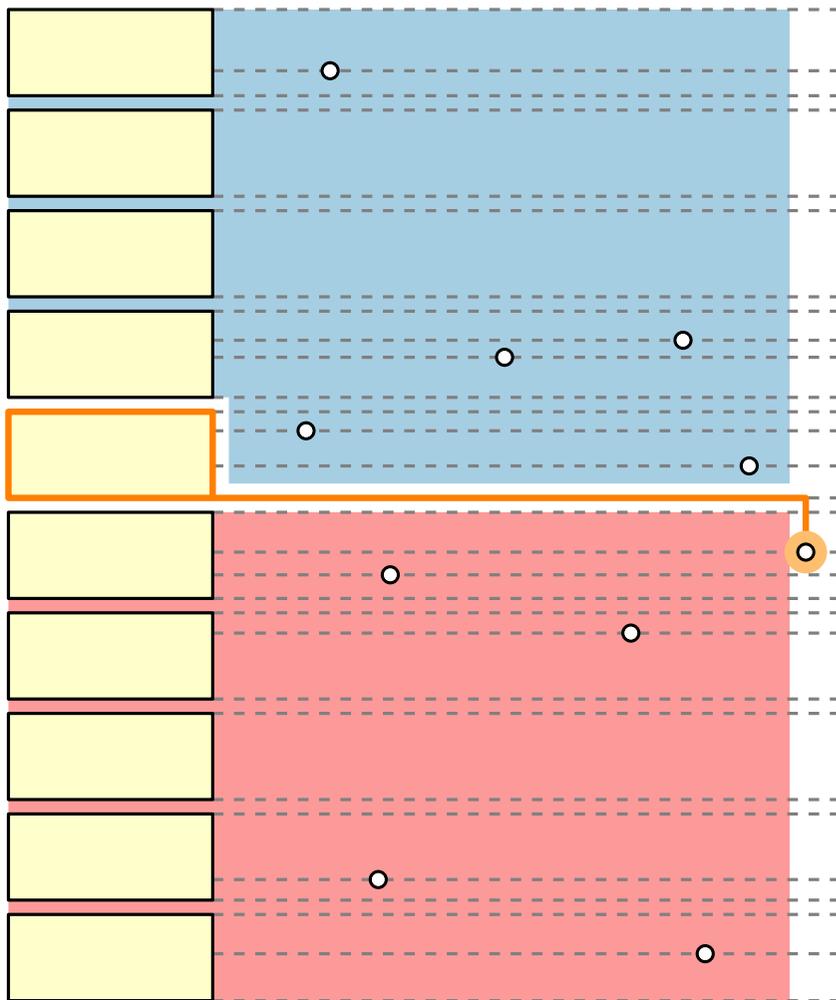
Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



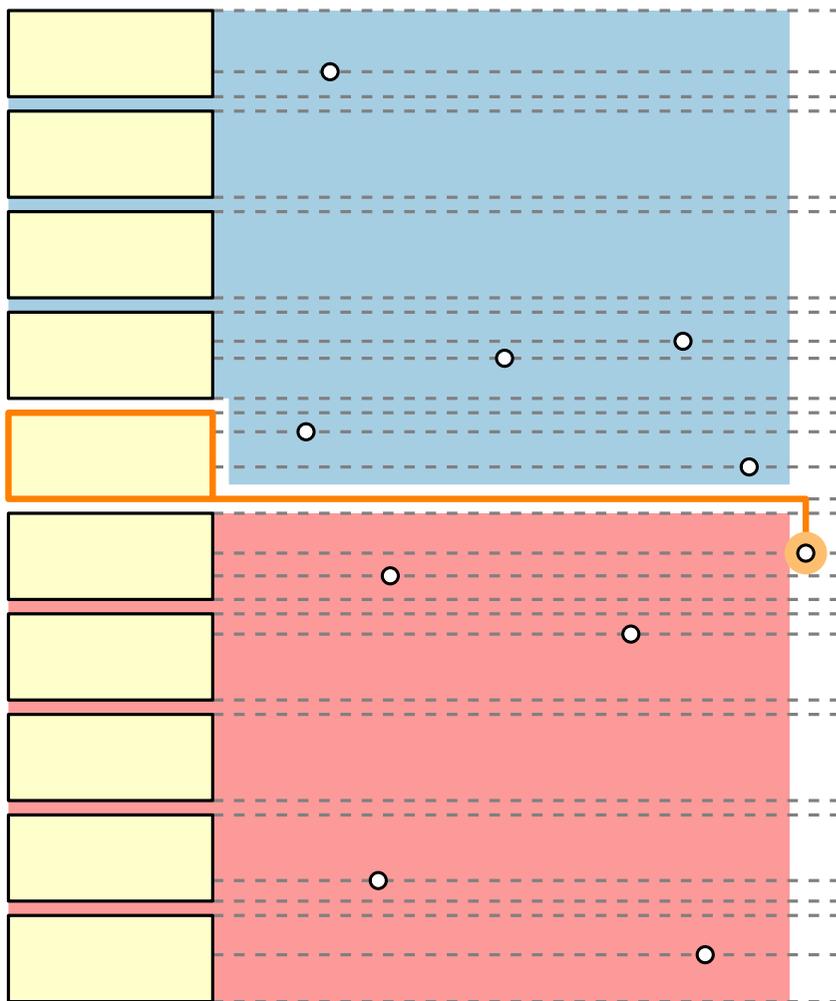
Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

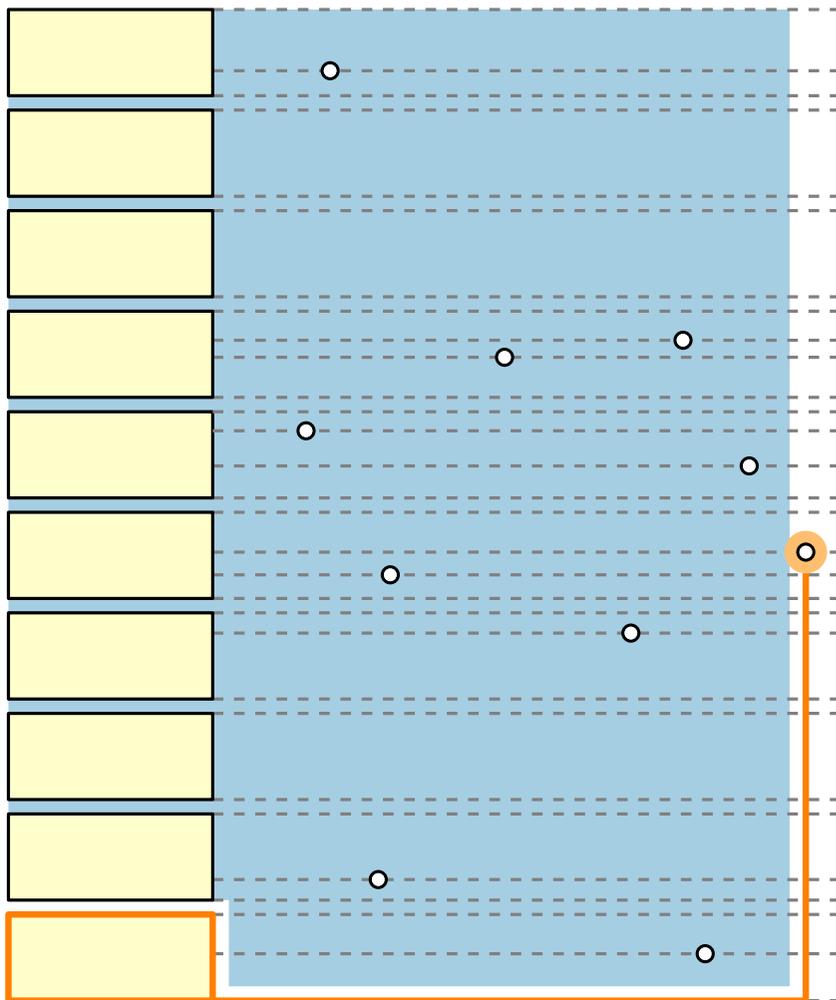
Teilinstanzen müssen *balanciert* sein

(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

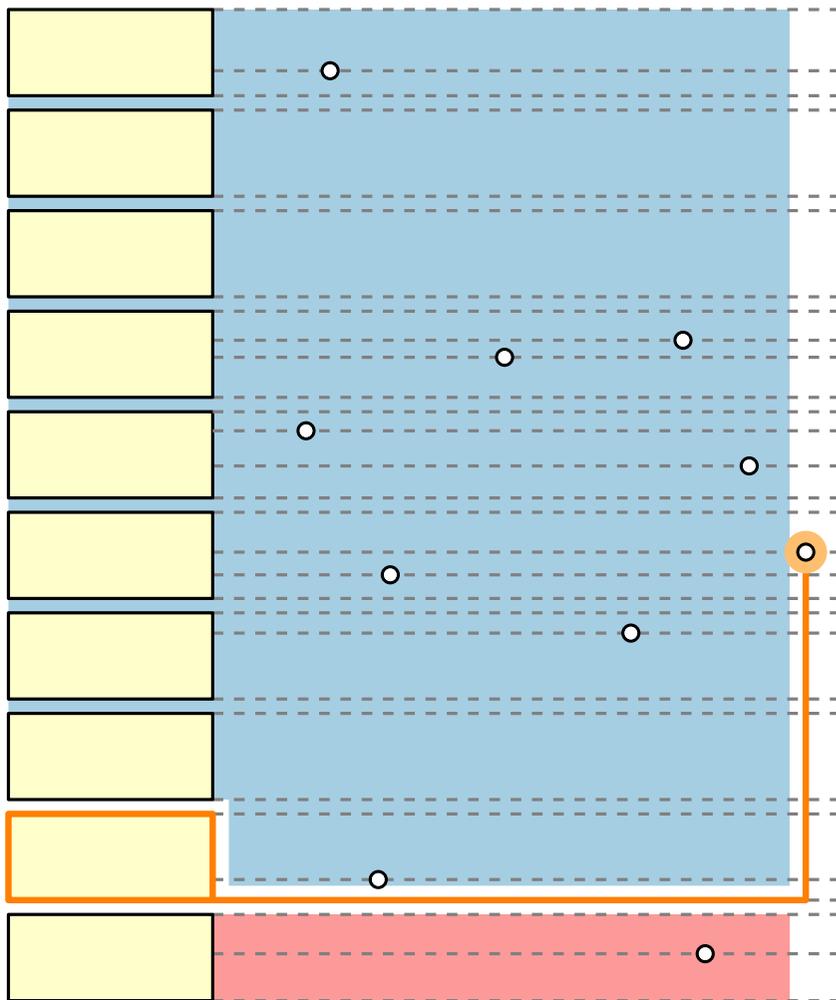
Teilinstanzen müssen *balanciert* sein

(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

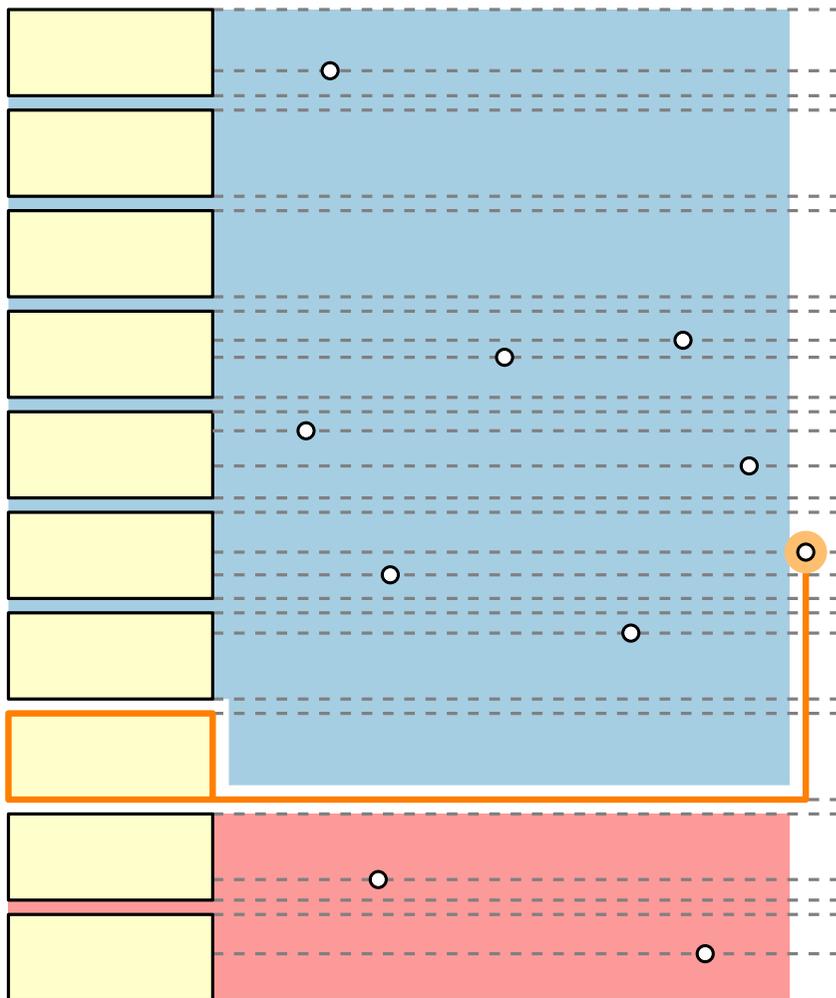
Teilinstanzen müssen *balanciert* sein

(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

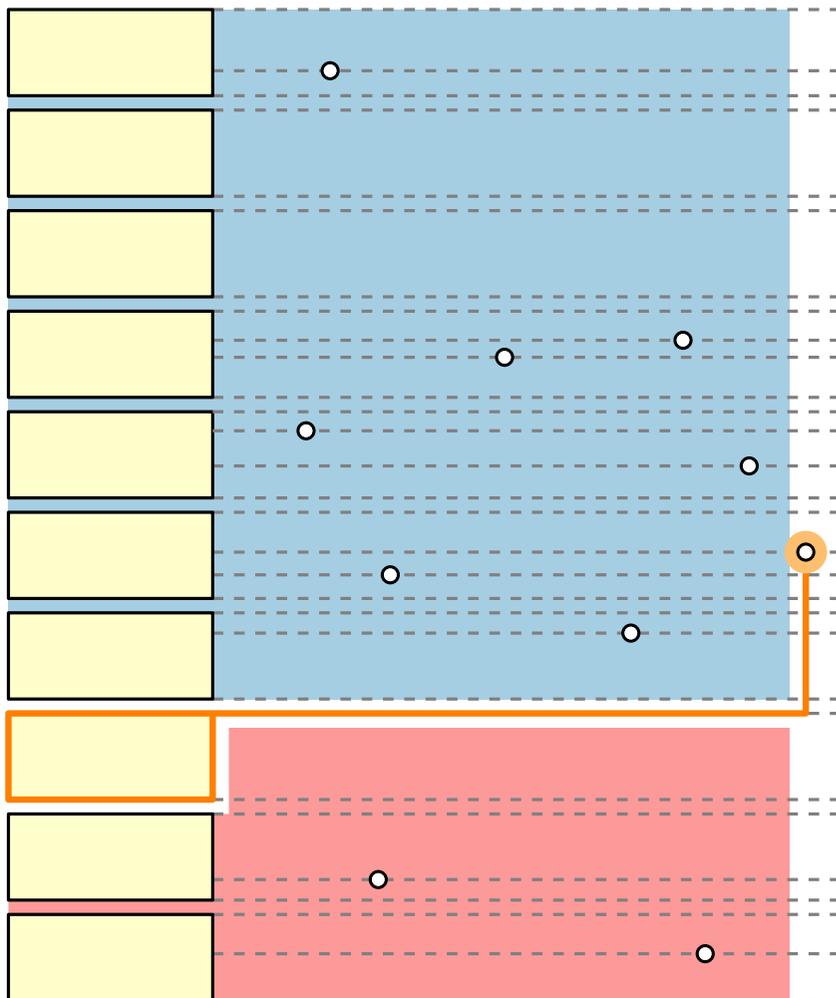
Teilinstanzen müssen *balanciert* sein

(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

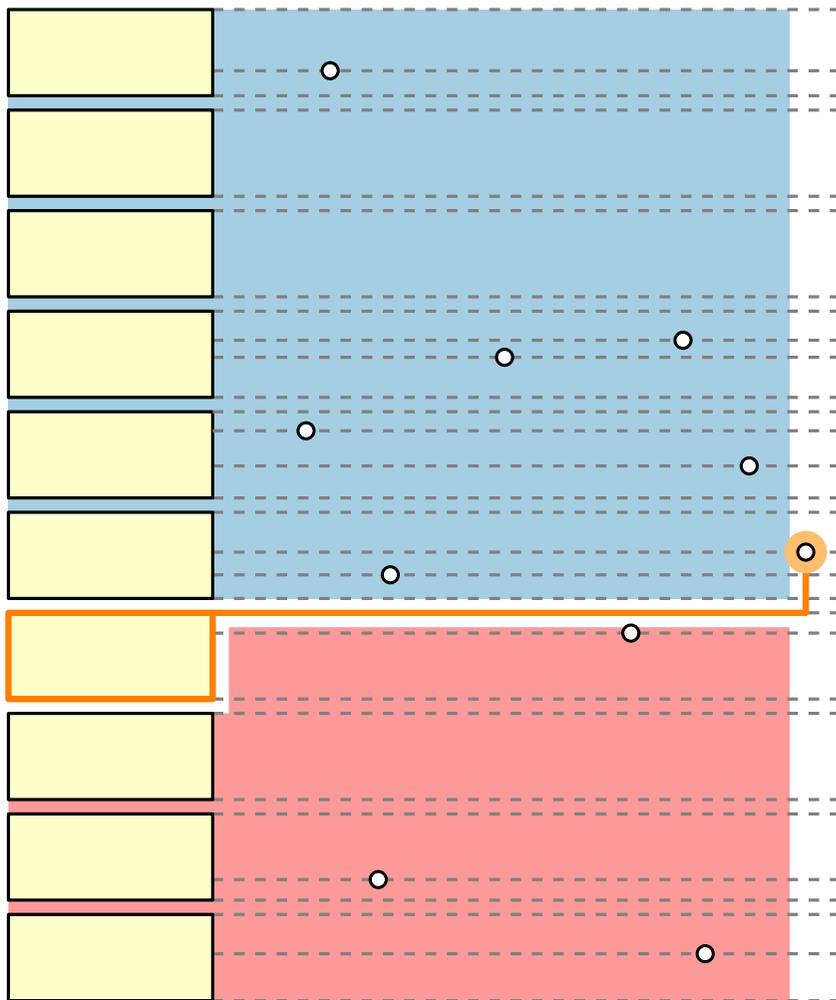
Teilinstanzen müssen *balanciert* sein

(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

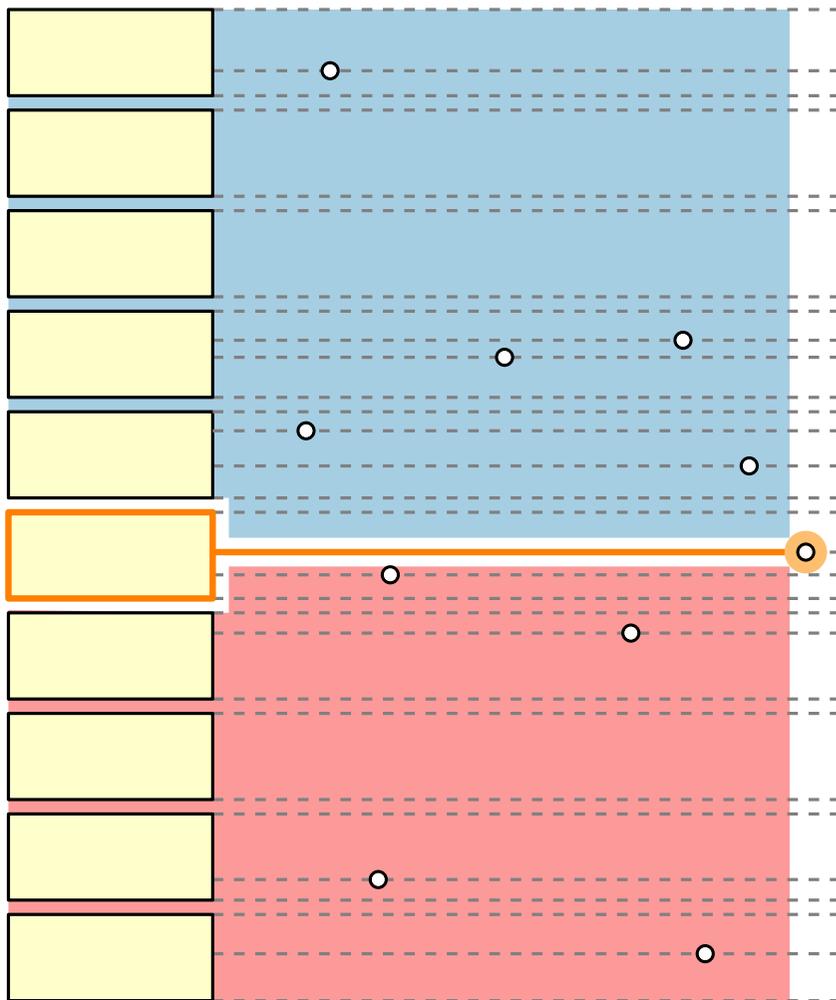
Teilinstanzen müssen *balanciert* sein

(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

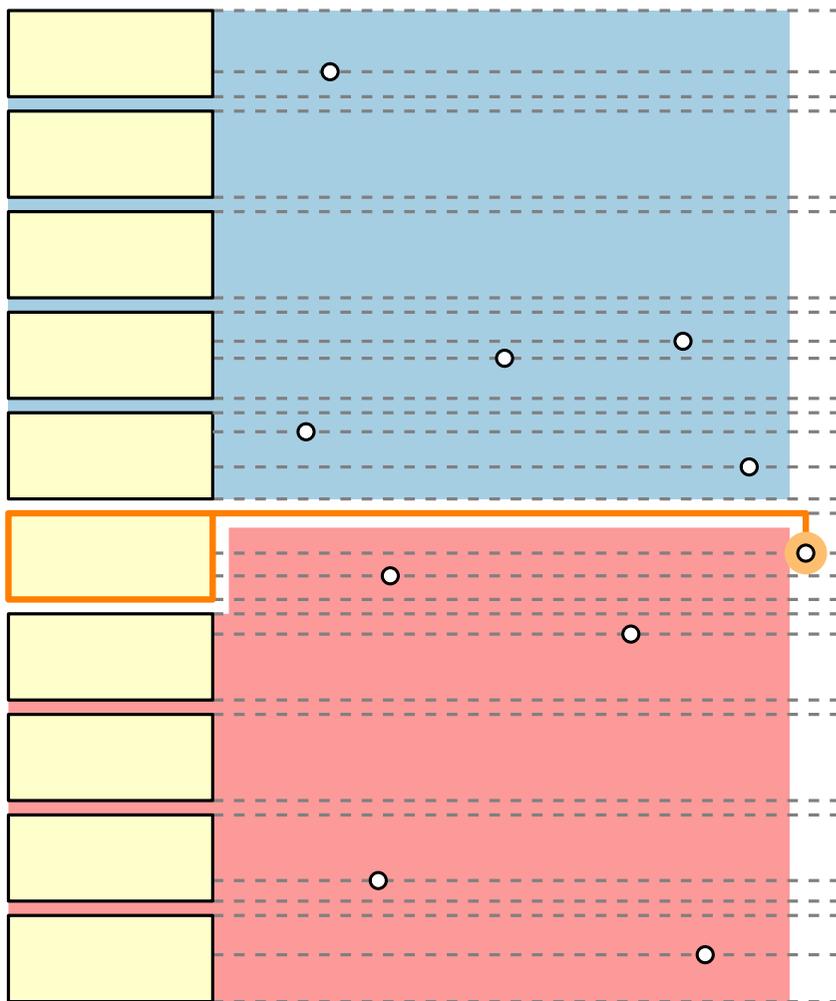
Teilinstanzen müssen *balanciert* sein

(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

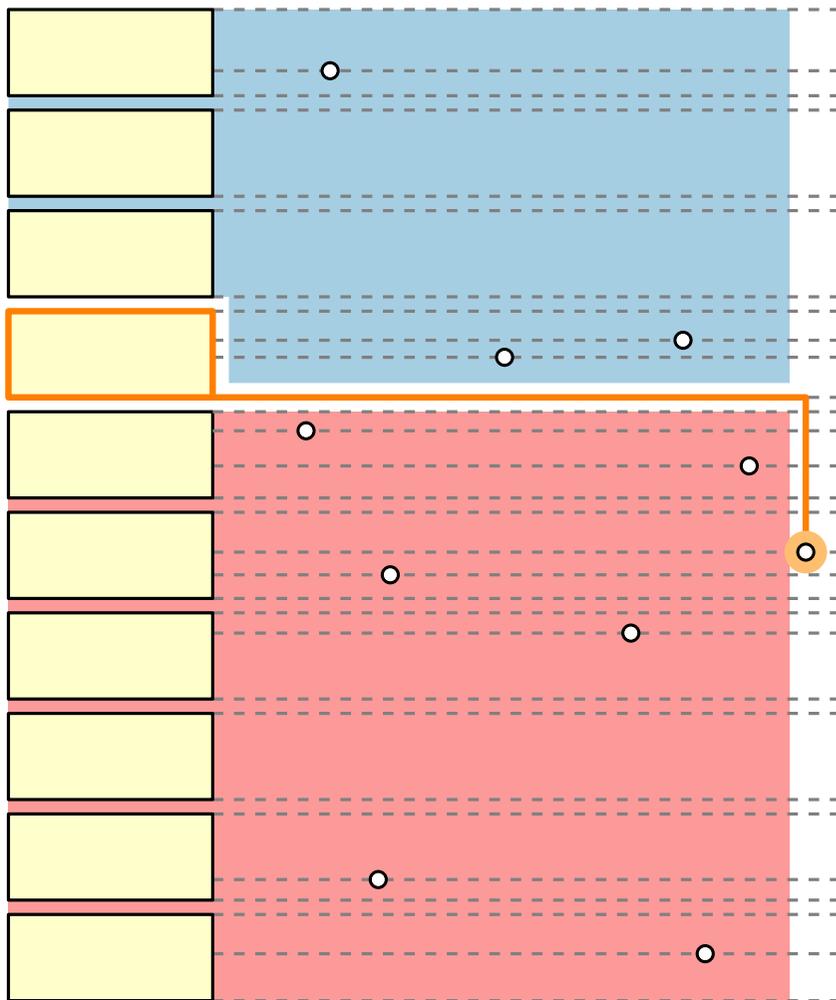
Teilinstanzen müssen *balanciert* sein

(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

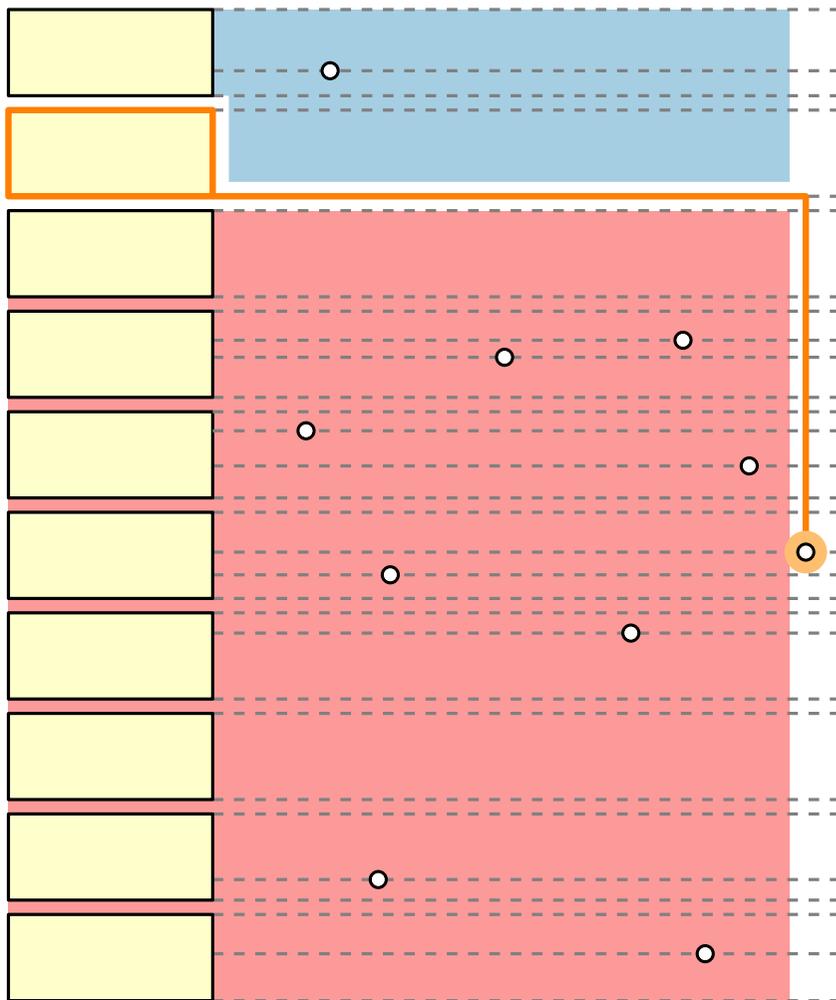
Teilinstanzen müssen *balanciert* sein

(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

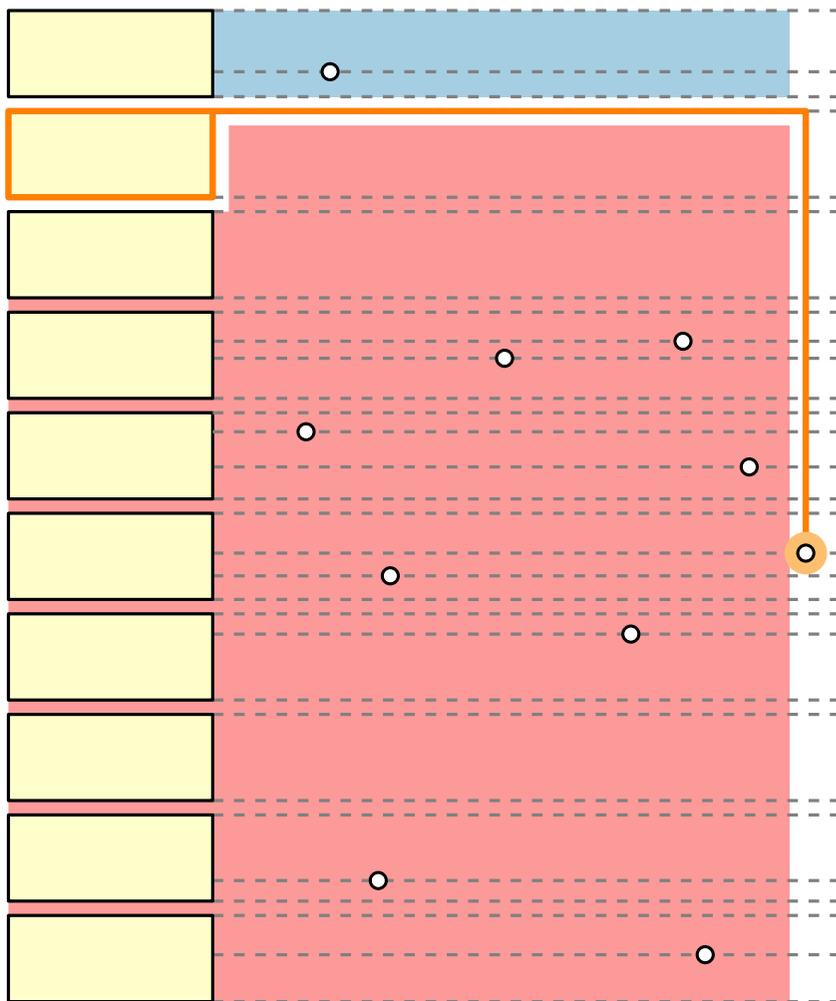
Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

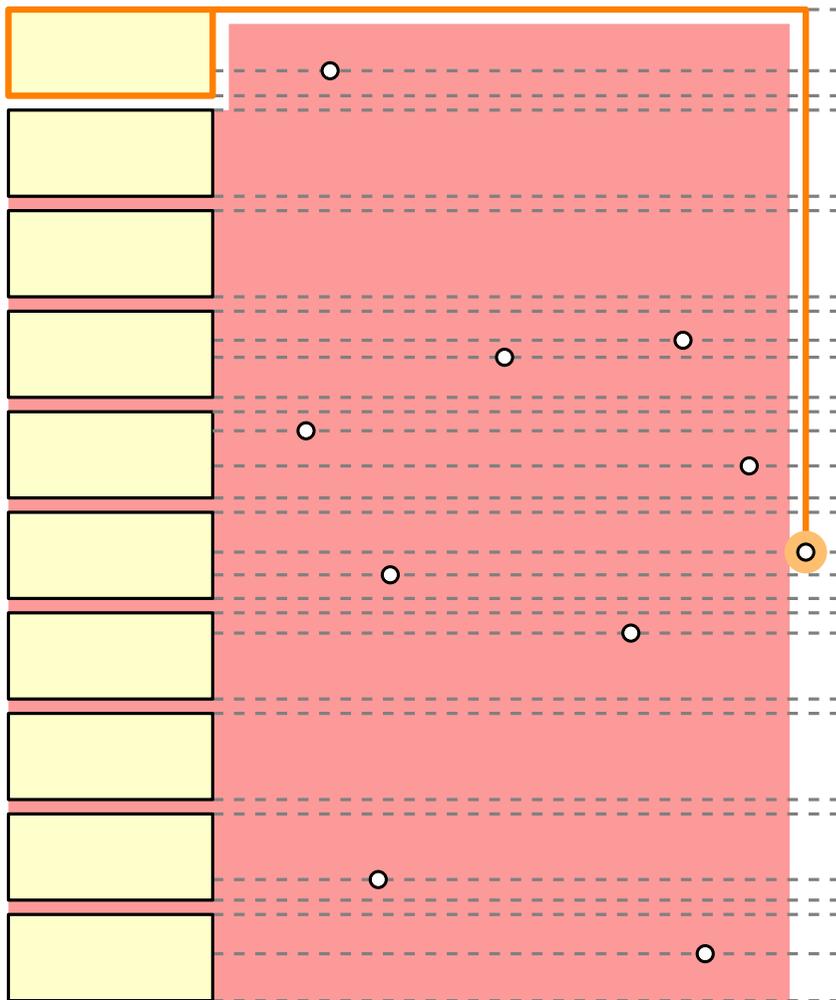
Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Allgemeines Qualitätsmaß

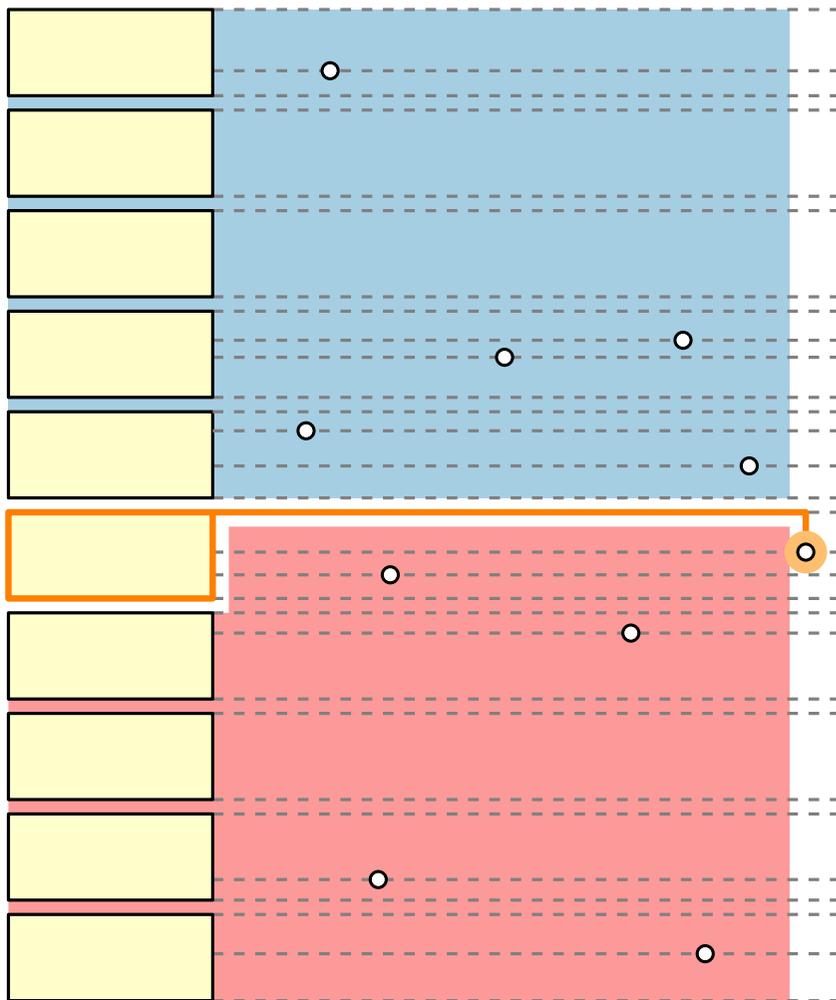
Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm

Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

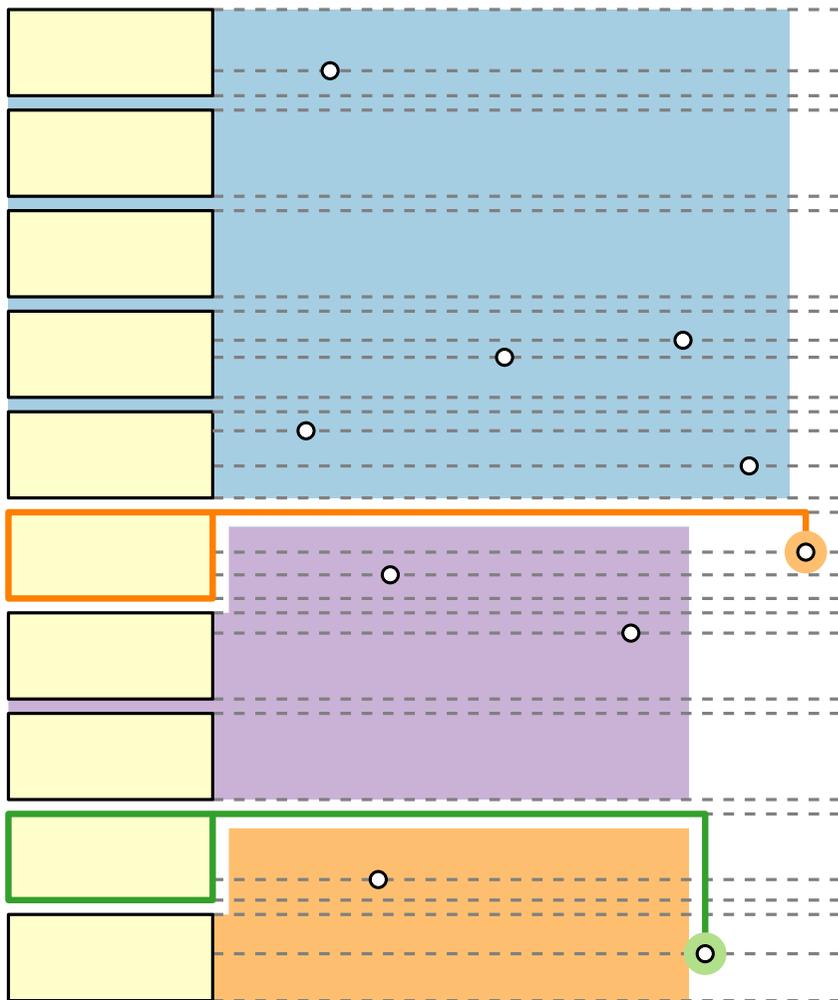


Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen. } Dynamisches Programm
2. Löse Teilinstanzen rekursiv.

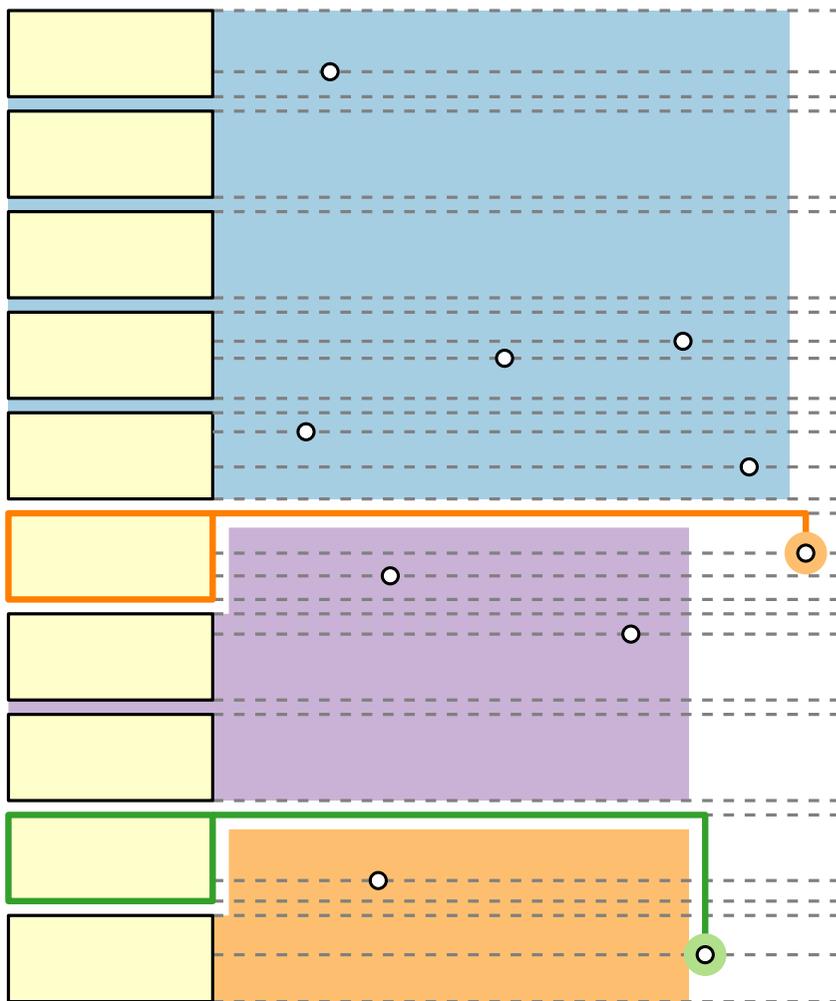
Betrachte rechtesten Punkt.
Wähle Leader aus einem Streifenrand.
Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)



Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen. } Dynamisches Programm
2. Löse Teilinstanzen rekursiv.



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

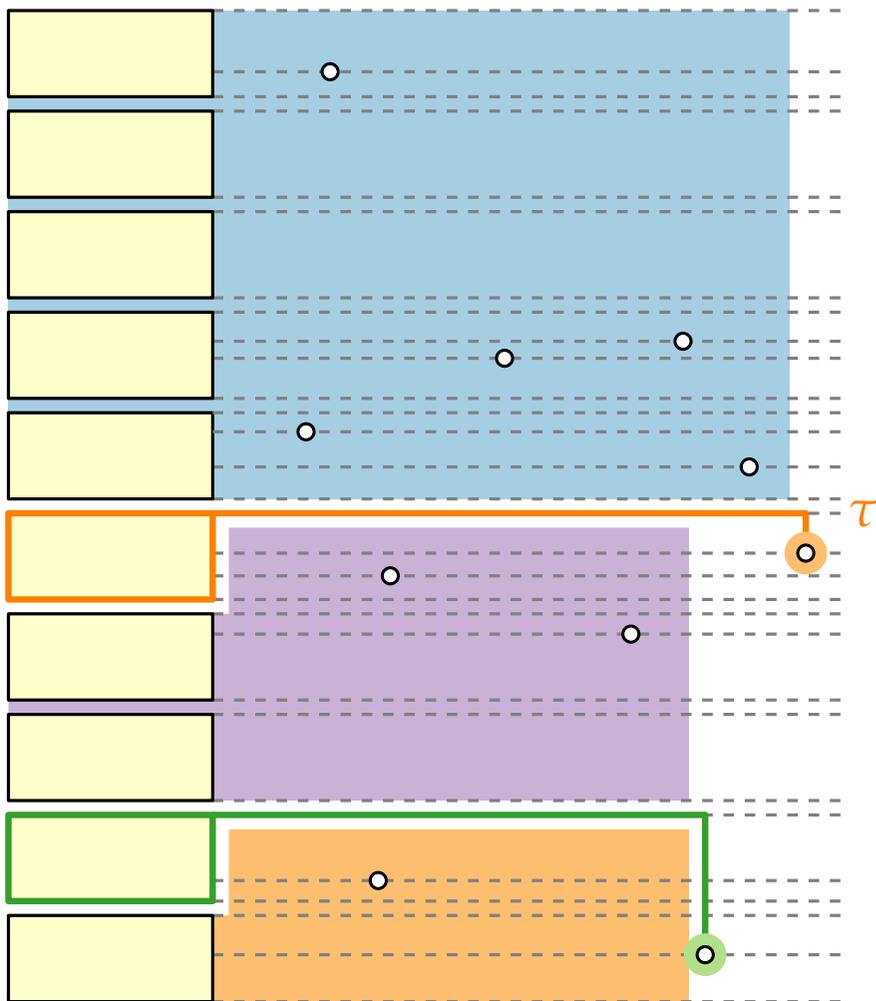
Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

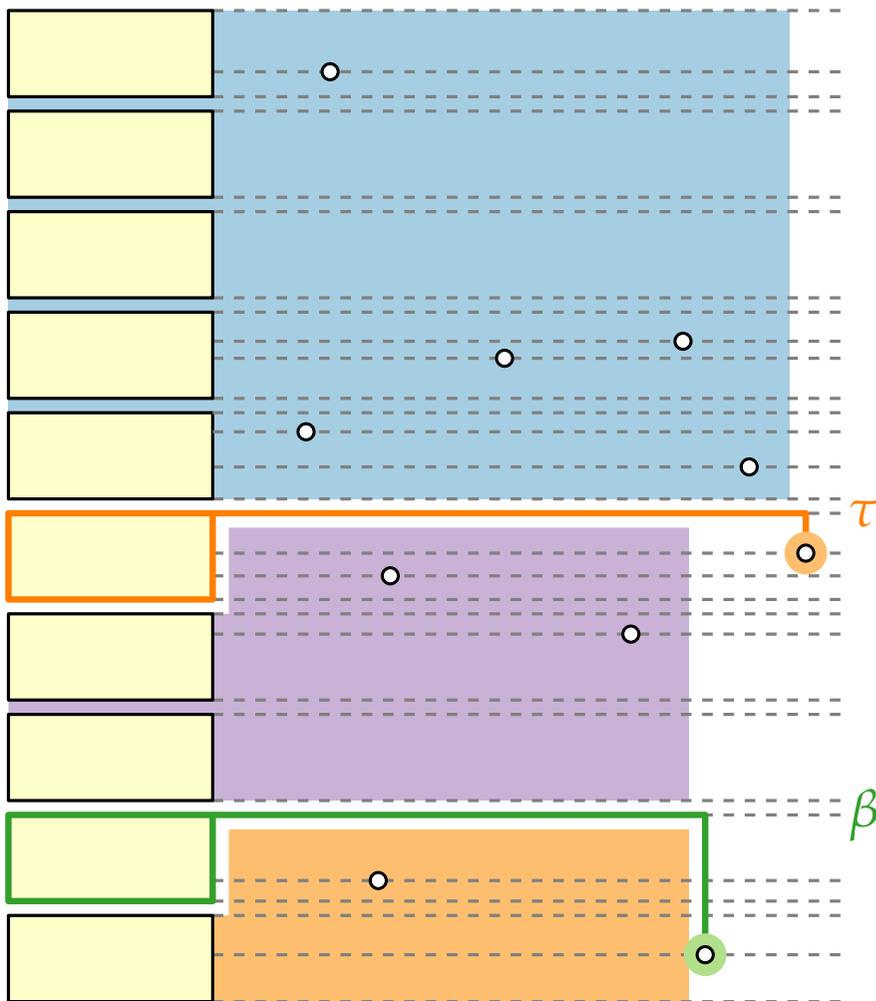
Teilinstanz definiert durch:

τ : oberer Rand

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.
Wähle Leader aus einem Streifenrand.
Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

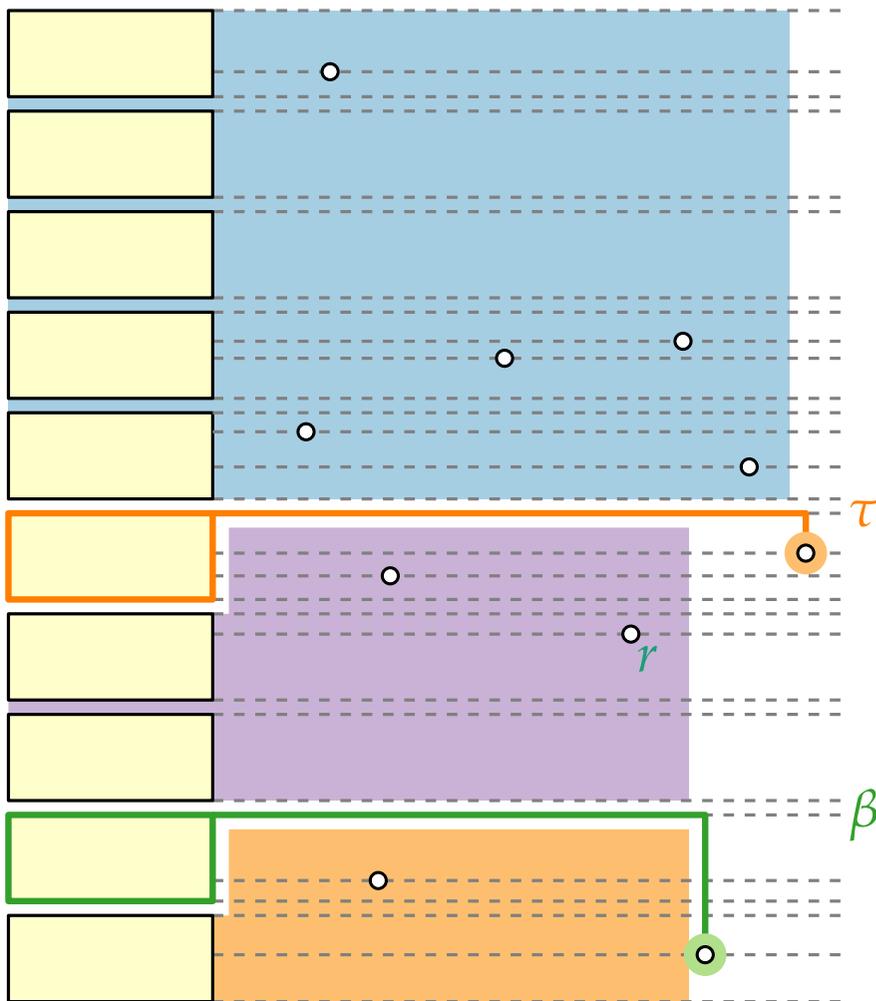
τ : oberer Rand

β : unterer Rand

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

τ : oberer Rand

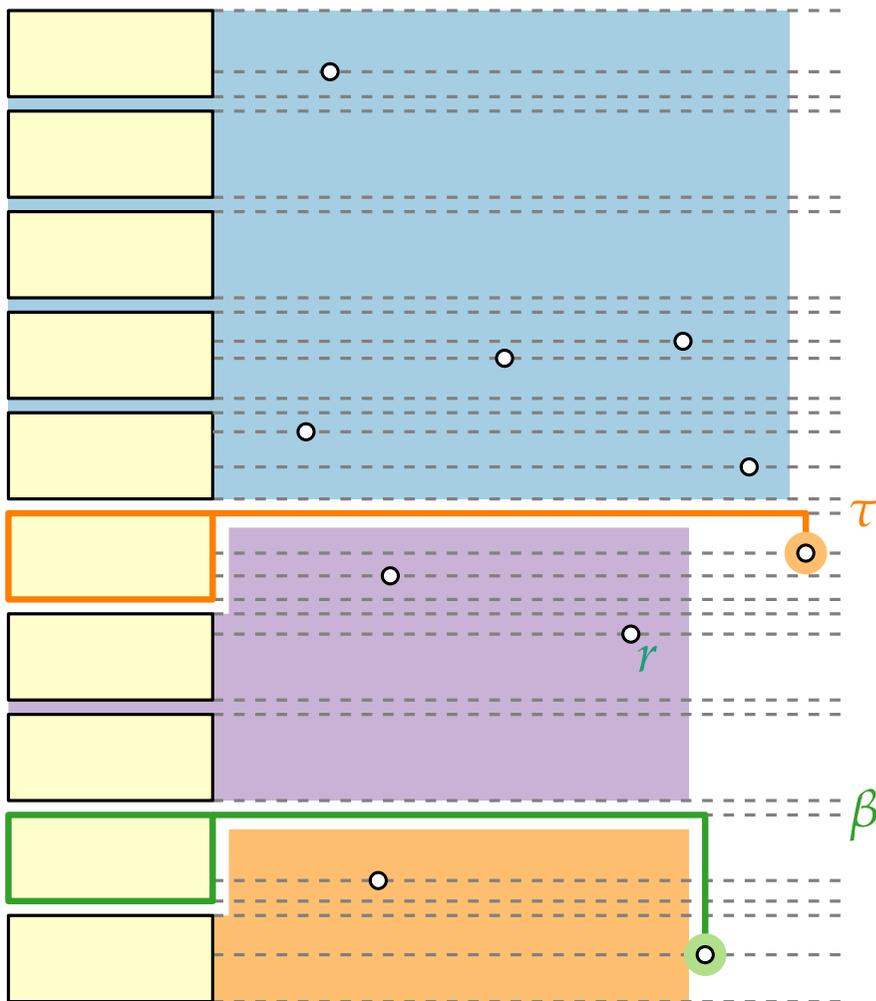
β : unterer Rand

r : rechtester Punkt

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

τ : oberer Rand

β : unterer Rand

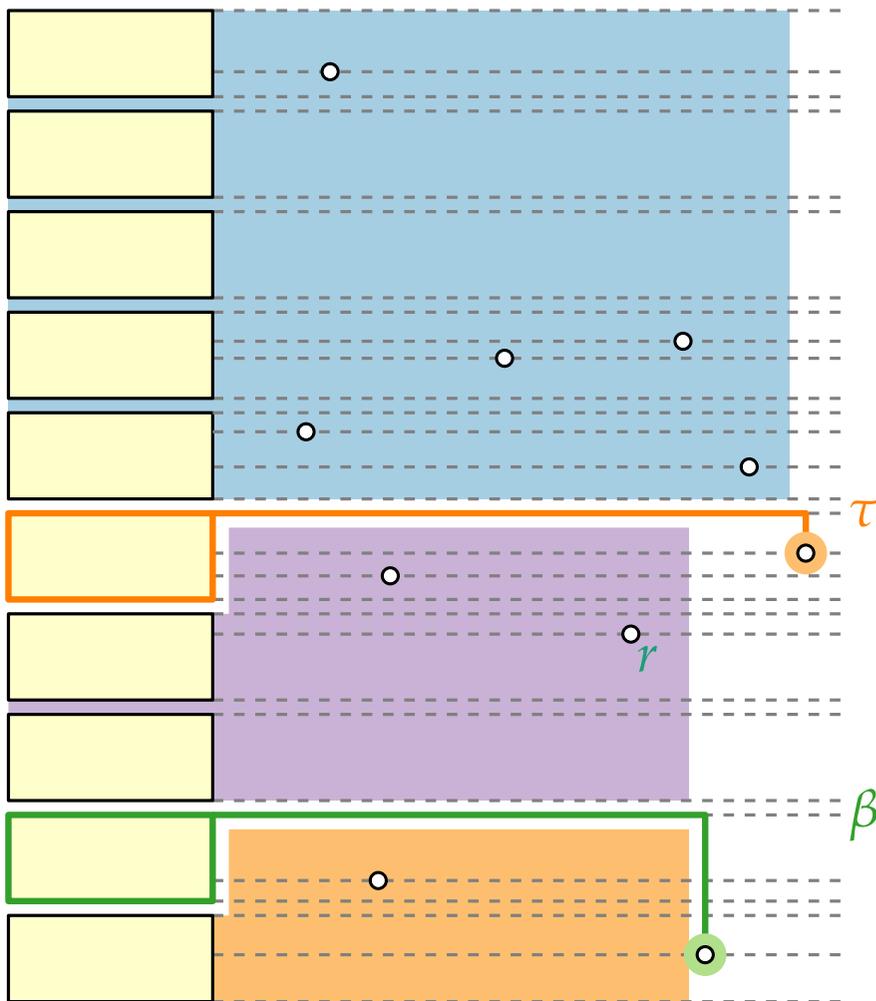
~~r : rechtester Punkt~~

$r(\beta, \tau)$ eindeutig!

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

τ : oberer Rand

β : unterer Rand

~~r : rechtester Punkt~~

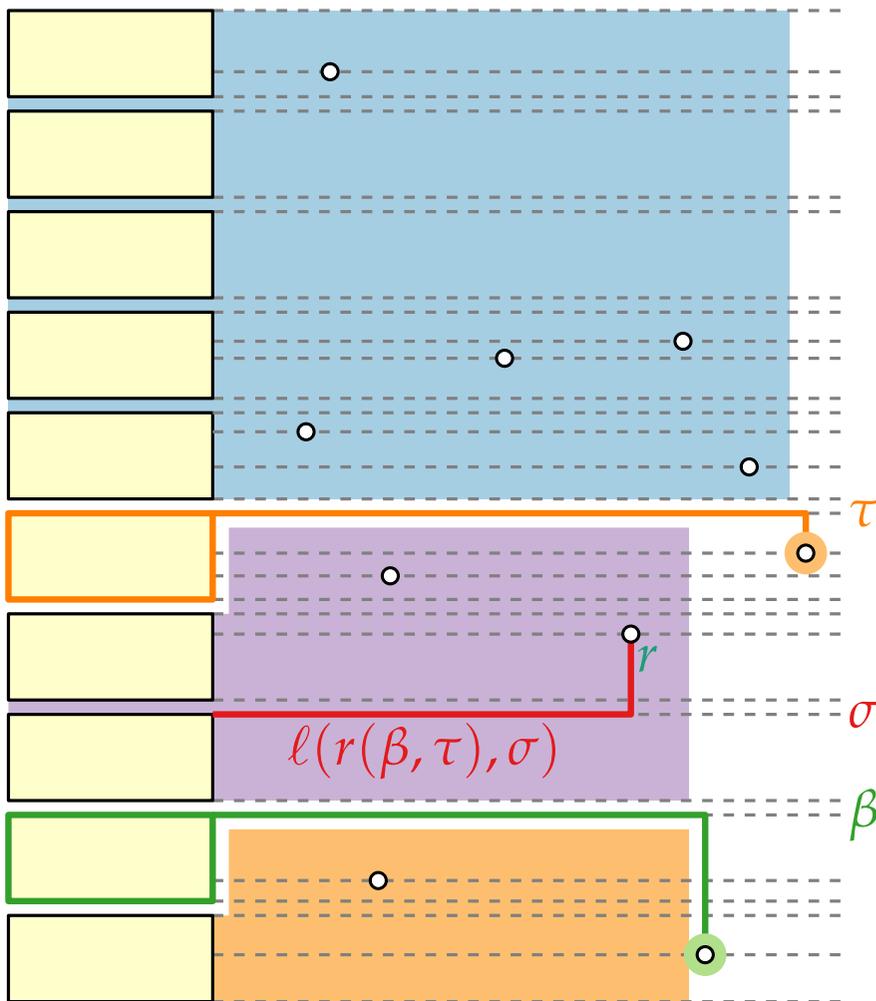
$r(\beta, \tau)$ eindeutig!

$$\beta T[\beta, \tau] =$$

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

τ : oberer Rand

β : unterer Rand

~~r : rechtester Punkt~~

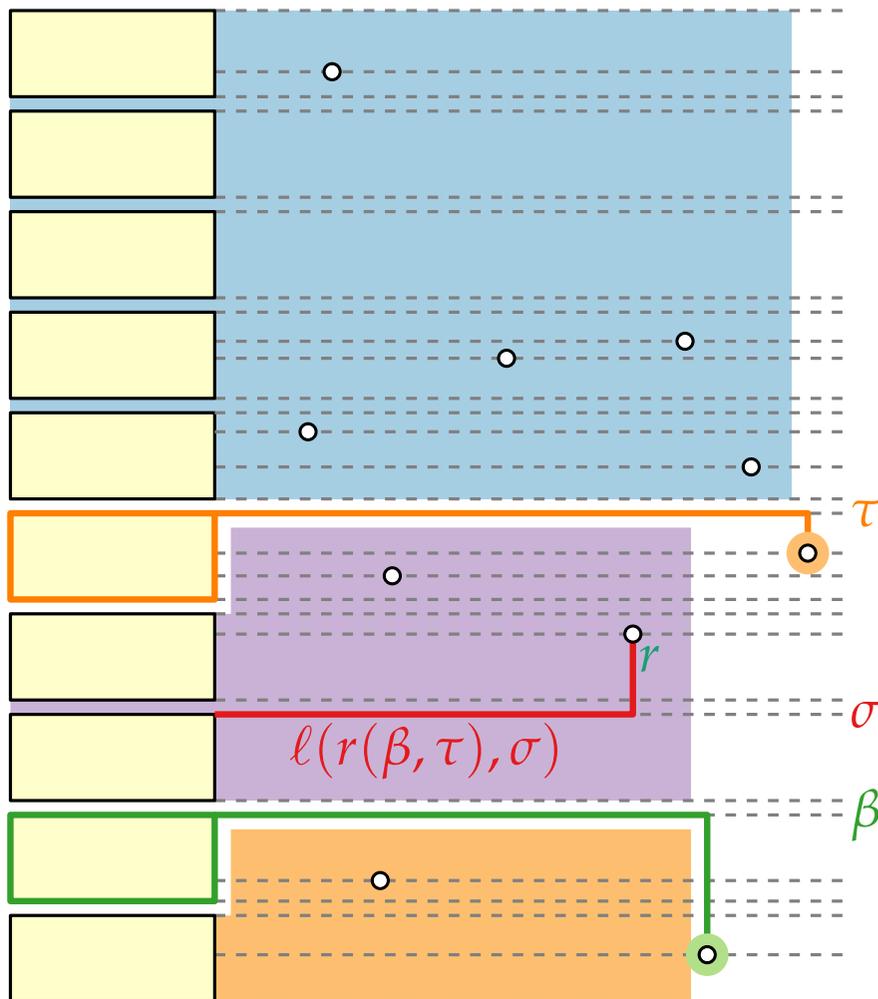
$r(\beta, \tau)$ eindeutig!

$$\beta T[\beta, \tau] =$$

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

τ : oberer Rand

β : unterer Rand

~~r : rechtester Punkt~~

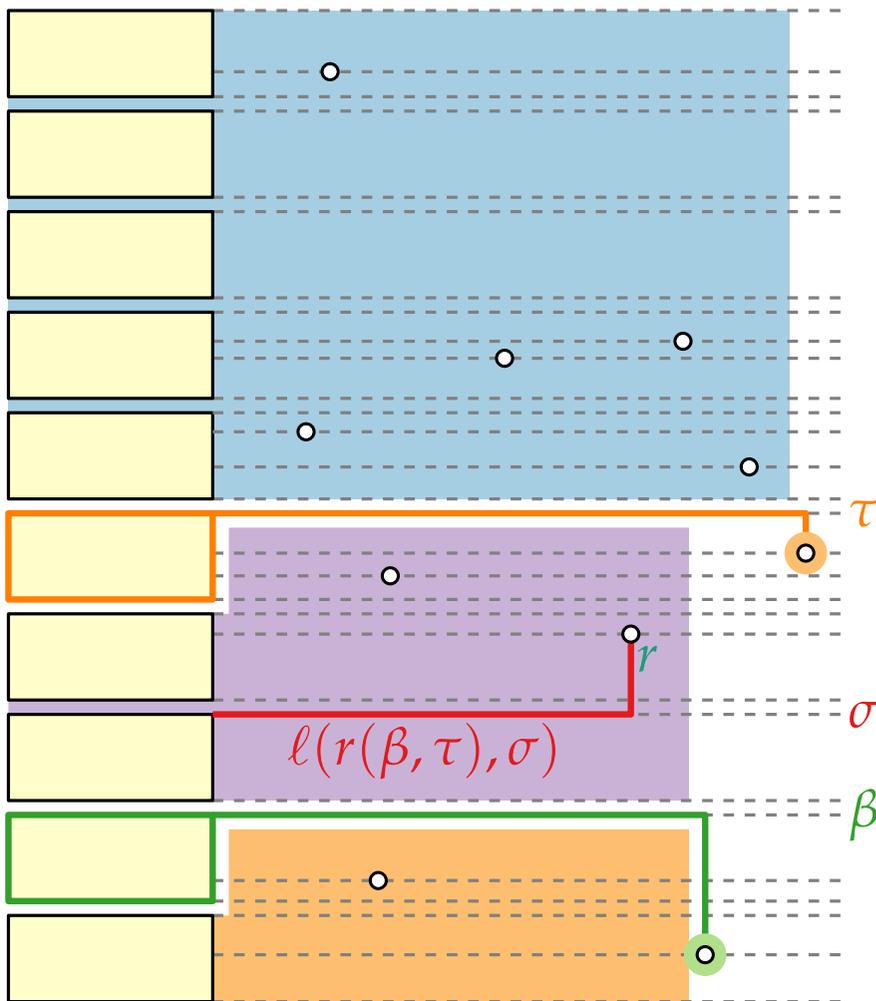
$r(\beta, \tau)$ eindeutig!

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau} l(r(\beta, \tau), \sigma) \text{ balanciert } \{$$

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

τ : oberer Rand

β : unterer Rand

~~r : rechtester Punkt~~

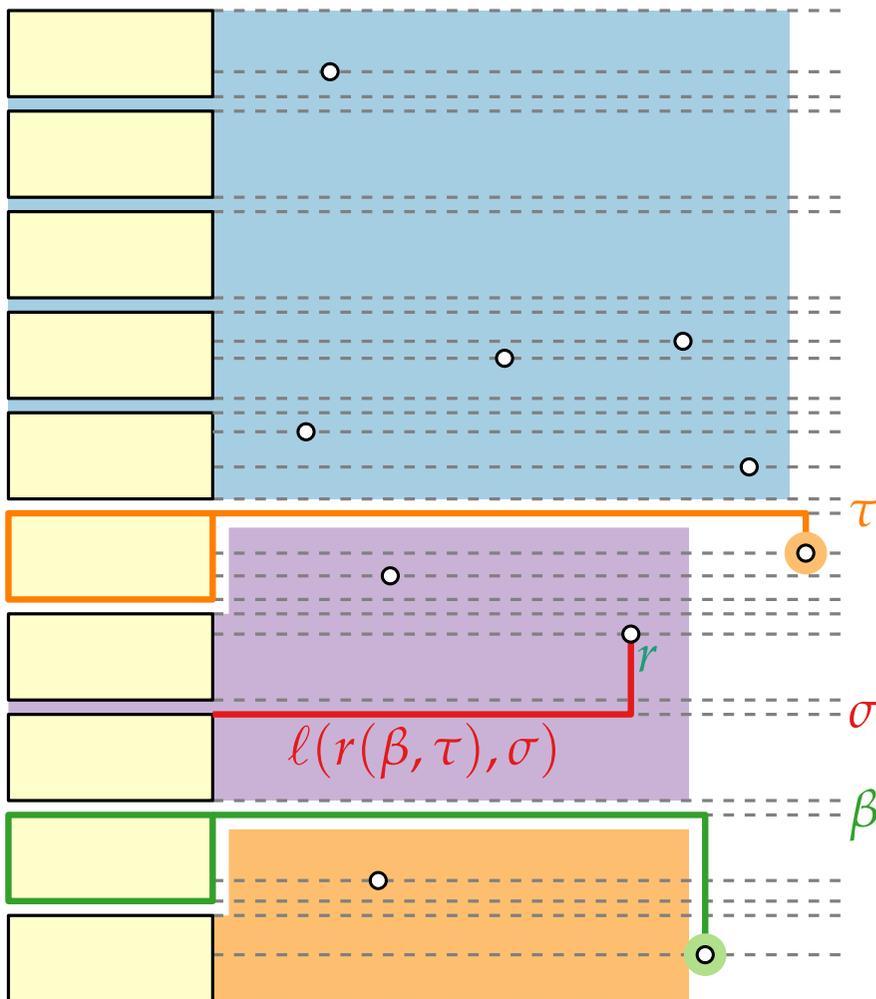
$r(\beta, \tau)$ eindeutig!

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau} l(r(\beta, \tau), \sigma) \text{ balanciert } \{ T[\beta, \sigma] + \dots \}$$

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.
Wähle Leader aus einem Streifenrand.
Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

τ : oberer Rand

β : unterer Rand

~~r : rechtester Punkt~~

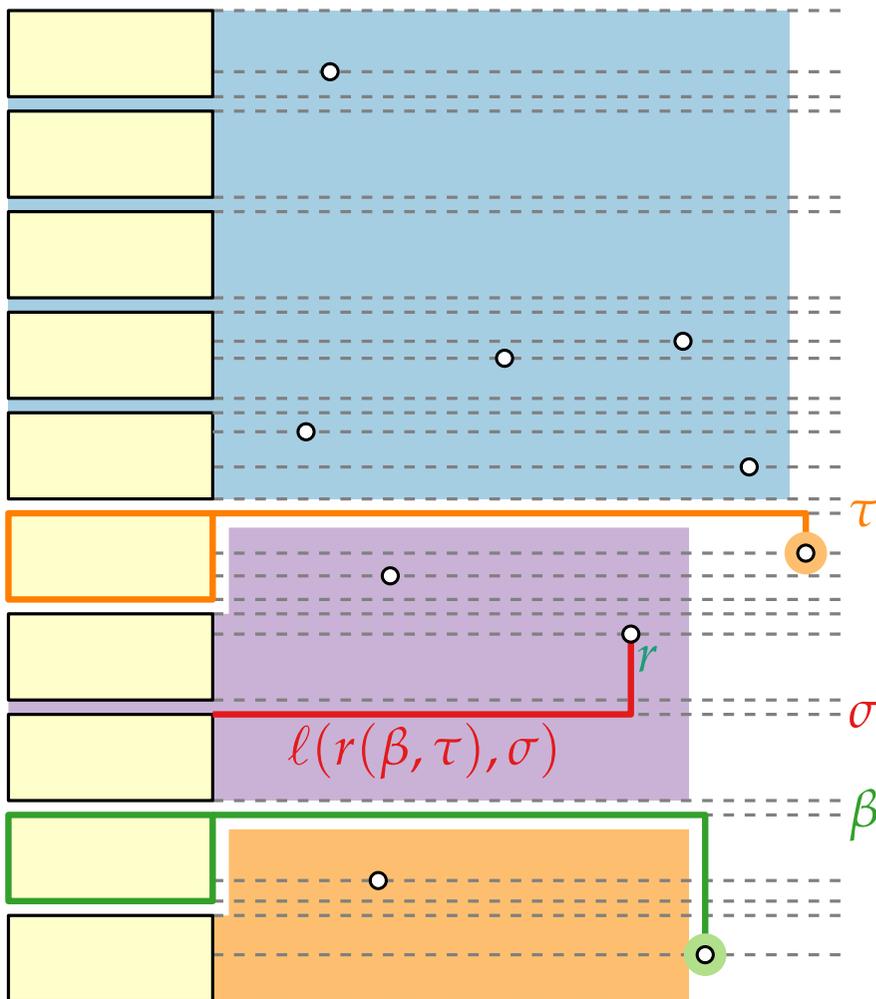
$r(\beta, \tau)$ eindeutig!

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau} \{ \ell(r(\beta, \tau), \sigma) \text{ balanciert } T[\beta, \sigma] + T[\sigma, \tau] + \dots \}$$

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.

Wähle Leader aus einem Streifenrand.

Teilinstanzen müssen *balanciert* sein
(#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

τ : oberer Rand

β : unterer Rand

~~r : rechtester Punkt~~

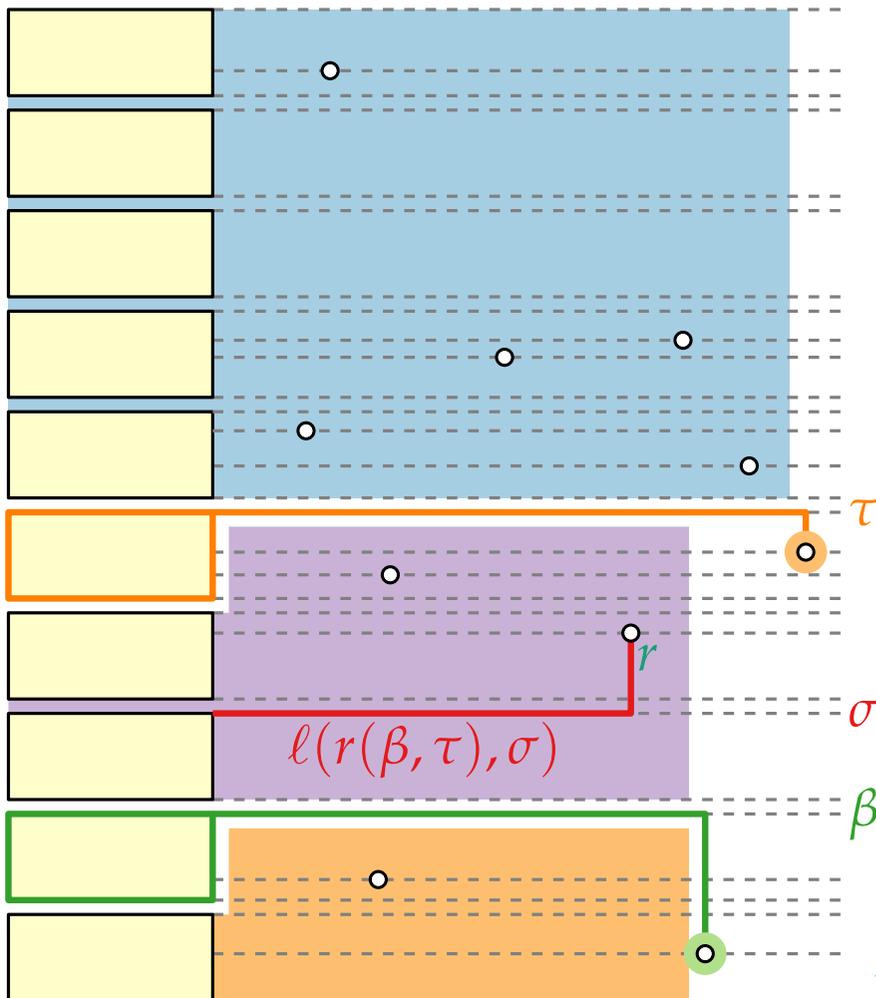
$r(\beta, \tau)$ eindeutig!

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau} \{ \text{qual}(\ell(r(\beta, \tau), \sigma)) \text{ balanciert } T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Allgemeines Qualitätsmaß

Definiere $O(n)$ Streifen durch Punkte und Beschriftungen.

1. Unterteile Instanz in Teilinstanzen.
 2. Löse Teilinstanzen rekursiv.
- } Dynamisches Programm



Betrachte rechtesten Punkt.
 Wähle Leader aus einem Streifenrand.
 Teilinstanzen müssen *balanciert* sein
 (#Punkte = #Beschriftungen)

Teilinstanz definiert durch:

τ : oberer Rand

β : unterer Rand

~~r : rechtester Punkt~~

$r(\beta, \tau)$ eindeutig!

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau} \{ \text{balanciert } l(r(\beta, \tau), \sigma) + T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(l(r(\beta, \tau), \sigma)) \}$$

Allgem. Qualitätsmaß:

$\text{qual}(l(r(\beta, \tau), \sigma))$

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Laufzeit:

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Laufzeit:

$O(n)$ Streifen, 2 Dimensionen \Rightarrow DP-Tabelle hat Größe $O(n^2)$.

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Laufzeit:

$O(n)$ Streifen, 2 Dimensionen \Rightarrow DP-Tabelle hat Größe $O(n^2)$.

Berechne $\text{qual}(\ell(r(\beta, \tau), \sigma))$ vor $\Rightarrow O(n^3 \cdot f(n))$ Zeit.

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Laufzeit:

$O(n)$ Streifen, 2 Dimensionen \Rightarrow DP-Tabelle hat Größe $O(n^2)$.

Berechne $\text{qual}(\ell(r(\beta, \tau), \sigma))$ vor $\Rightarrow O(n^3 \cdot f(n))$ Zeit.

Sortiere Punkte nach x - und y -Koordinate $\Rightarrow O(n \log n)$ Zeit.

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Laufzeit:

$O(n)$ Streifen, 2 Dimensionen \Rightarrow DP-Tabelle hat Größe $O(n^2)$.

Berechne $\text{qual}(\ell(r(\beta, \tau), \sigma))$ vor $\Rightarrow O(n^3 \cdot f(n))$ Zeit.

Sortiere Punkte nach x - und y -Koordinate $\Rightarrow O(n \log n)$ Zeit.

Berechne einen Eintrag $T[\beta, \tau]$ der DP-Tabelle:

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Laufzeit:

$O(n)$ Streifen, 2 Dimensionen \Rightarrow DP-Tabelle hat Größe $O(n^2)$.

Berechne $\text{qual}(\ell(r(\beta, \tau), \sigma))$ vor $\Rightarrow O(n^3 \cdot f(n))$ Zeit.

Sortiere Punkte nach x - und y -Koordinate $\Rightarrow O(n \log n)$ Zeit.

Berechne einen Eintrag $T[\beta, \tau]$ der DP-Tabelle:

Finde $r(\beta, \tau)$ in $O(n)$ Zeit.

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Laufzeit:

$O(n)$ Streifen, 2 Dimensionen \Rightarrow DP-Tabelle hat Größe $O(n^2)$.

Berechne $\text{qual}(\ell(r(\beta, \tau), \sigma))$ vor $\Rightarrow O(n^3 \cdot f(n))$ Zeit.

Sortiere Punkte nach x - und y -Koordinate $\Rightarrow O(n \log n)$ Zeit.

Berechne einen Eintrag $T[\beta, \tau]$ der DP-Tabelle:

Finde $r(\beta, \tau)$ in $O(n)$ Zeit.

Alle σ können durch Sortierung in $O(n)$ Zeit gefunden werden.

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Laufzeit:

$O(n)$ Streifen, 2 Dimensionen \Rightarrow DP-Tabelle hat Größe $O(n^2)$.

Berechne $\text{qual}(\ell(r(\beta, \tau), \sigma))$ vor $\Rightarrow O(n^3 \cdot f(n))$ Zeit.

Sortiere Punkte nach x - und y -Koordinate $\Rightarrow O(n \log n)$ Zeit.

Berechne einen Eintrag $T[\beta, \tau]$ der DP-Tabelle:

Finde $r(\beta, \tau)$ in $O(n)$ Zeit.

Alle σ können durch Sortierung in $O(n)$ Zeit gefunden werden. $\Rightarrow O(n)$ Zeit

Allgemeines Qualitätsmaß

$$T[\beta, \tau] = \min_{\beta < \sigma < \tau, \text{ balanciert}} \{ T[\beta, \sigma] + T[\sigma, \tau] + \text{qual}(\ell(r(\beta, \tau), \sigma)) \}$$

Annahme: Funktion $\text{qual}(\ell(r(\beta, \tau), \sigma))$ berechenbar in $f(n)$ Zeit.

Laufzeit:

$O(n)$ Streifen, 2 Dimensionen \Rightarrow DP-Tabelle hat Größe $O(n^2)$.

Berechne $\text{qual}(\ell(r(\beta, \tau), \sigma))$ vor $\Rightarrow O(n^3 \cdot f(n))$ Zeit.

Sortiere Punkte nach x - und y -Koordinate $\Rightarrow O(n \log n)$ Zeit.

Berechne einen Eintrag $T[\beta, \tau]$ der DP-Tabelle:

Finde $r(\beta, \tau)$ in $O(n)$ Zeit.

Alle σ können durch Sortierung in $O(n)$ Zeit gefunden werden. $\Rightarrow O(n)$ Zeit

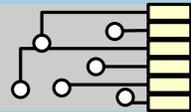
Satz. Eine kreuzungsfreie Beschriftung im 1-seitigen po-Leader Model mit freien Ports, die eine beliebiges Qualitätsmaß für Leader minimiert, kann in $O(n^3 \cdot f(n))$ Zeit gefunden werden.

Ergebnisse

Leader	Seiten	Ports	Zielfkt.	Laufzeit

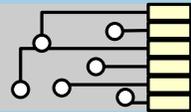
Ergebnisse

[1] Benkert et al. '09

Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]

Ergebnisse

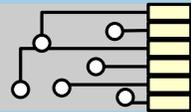
[1] Benkert et al. '09

Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

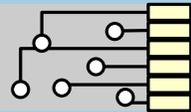
[1] Benkert et al. '09

Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
do 	1	frei	Länge	$O(n^2)$ [1]

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

[1] Benkert et al. '09

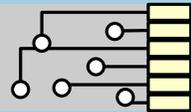
Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
do 	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

[1] Benkert et al. '09

[2] Bekos et al. '07

Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
do 	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]

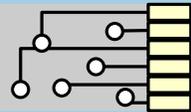
Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

[1] Benkert et al. '09

[2] Bekos et al. '07

[3] Kindermann et al. '16

Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
do 	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

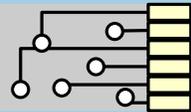
Ergebnisse

[1] Benkert et al. '09

[2] Bekos et al. '07

[3] Kindermann et al. '16

[4] Bose et al. '18

Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
do 	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

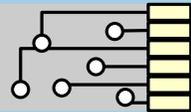
Ergebnisse

[1] Benkert et al. '09

[2] Bekos et al. '07

[3] Kindermann et al. '16

[4] Bose et al. '18

Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
	3	fix	–	$O(n^4)$ [3]
do 	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

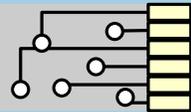
Ergebnisse

[1] Benkert et al. '09

[2] Bekos et al. '07

[3] Kindermann et al. '16

[4] Bose et al. '18

Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
	3	fix	–	$O(n^4)$ [3]
	3, 4	fix	Länge	$O(n^6)$ [4]
do 	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

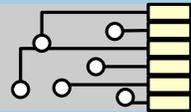
Ergebnisse

[1] Benkert et al. '09

[2] Bekos et al. '07

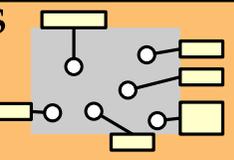
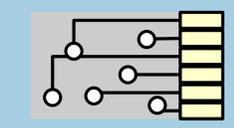
[3] Kindermann et al. '16

[4] Bose et al. '18

Leader	Seiten	Ports	Zielfkt.	Laufzeit
po 	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
	3	fix	–	$O(n^4)$ [3]
	3, 4	fix	Länge	$O(n^6)$ [4]
	1, 2, 3, 4	beides	Knicke	$O(n^6)$ [4]
do 	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

Leader	Seiten	Ports	Zielfkt.	Laufzeit
 s	1, 2, 3, 4	fix	Länge	$O(n^{2+\varepsilon})$ [2]
 po	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
	3	fix	–	$O(n^4)$ [3]
	3, 4	fix	Länge	$O(n^6)$ [4]
 do	1, 2, 3, 4	beides	Knicke	$O(n^6)$ [4]
	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]

[1] Benkert et al. '09

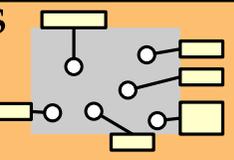
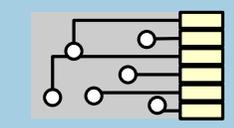
[2] Bekos et al. '07

[3] Kindermann et al. '16

[4] Bose et al. '18

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

Leader	Seiten	Ports	Zielfkt.	Laufzeit
	1, 2, 3, 4	fix	Länge	$O(n^{2+\varepsilon})$ [2]
	1, 2, 3, 4	frei	Länge	$O(n^3)$ [2]
	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
	3	fix	–	$O(n^4)$ [3]
	3, 4	fix	Länge	$O(n^6)$ [4]
1, 2, 3, 4	beides	Knicke	$O(n^6)$ [4]	
	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]

[1] Benkert et al. '09

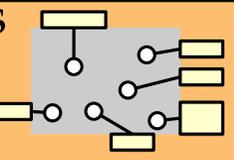
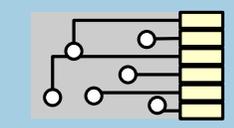
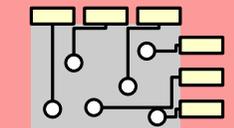
[2] Bekos et al. '07

[3] Kindermann et al. '16

[4] Bose et al. '18

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

Leader	Seiten	Ports	Zielfkt.	Laufzeit
 s	1, 2, 3, 4	fix	Länge	$O(n^{2+\varepsilon})$ [2]
	1, 2, 3, 4	frei	Länge	$O(n^3)$ [2]
 po	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
	3	fix	–	$O(n^4)$ [3]
	3, 4	fix	Länge	$O(n^6)$ [4]
 do	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]
 opo	1	fix	Länge	$O(n \log n)$ [2]

[1] Benkert et al. '09

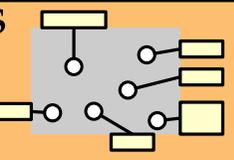
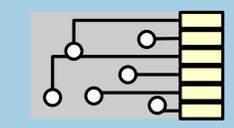
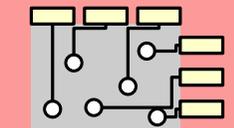
[2] Bekos et al. '07

[3] Kindermann et al. '16

[4] Bose et al. '18

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

Leader	Seiten	Ports	Zielfkt.	Laufzeit
	1, 2, 3, 4	fix	Länge	$O(n^{2+\varepsilon})$ [2]
	1, 2, 3, 4	frei	Länge	$O(n^3)$ [2]
	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
	3	fix	–	$O(n^4)$ [3]
	3, 4	fix	Länge	$O(n^6)$ [4]
	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]
	1	fix	Länge	$O(n \log n)$ [2]
	2	beides	Länge	$O(n^2)$ [2]

[1] Benkert et al. '09

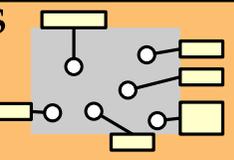
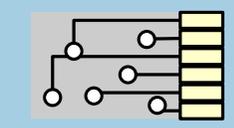
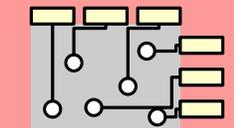
[2] Bekos et al. '07

[3] Kindermann et al. '16

[4] Bose et al. '18

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

Leader	Seiten	Ports	Zielfkt.	Laufzeit
	1, 2, 3, 4	fix	Länge	$O(n^{2+\varepsilon})$ [2]
	1, 2, 3, 4	frei	Länge	$O(n^3)$ [2]
	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
	3	fix	–	$O(n^4)$ [3]
	3, 4	fix	Länge	$O(n^6)$ [4]
	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]
	1	fix	Länge	$O(n \log n)$ [2]
	2	beides	Länge	$O(n^2)$ [2]
	4	fix	Länge	$O(n^2 \log^3 n)$ [2]

[1] Benkert et al. '09

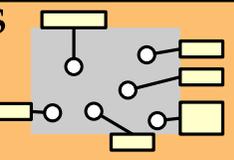
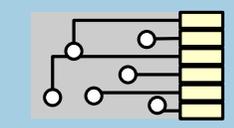
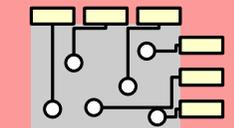
[2] Bekos et al. '07

[3] Kindermann et al. '16

[4] Bose et al. '18

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$

Ergebnisse

Leader	Seiten	Ports	Zielfkt.	Laufzeit
	1, 2, 3, 4	fix	Länge	$O(n^{2+\varepsilon})$ [2]
	1, 2, 3, 4	frei	Länge	$O(n^3)$ [2]
	1	frei	Länge	$O(n \log n)$ [1]
	1	frei	beliebig	$O(n^3)$ [1]
	2 (geg.)	frei	Länge	$O(n^2)$ [2]
	2 (adj.)	fix	–	$O(n^2)$ [3]
	2 (adj.)	fix	Länge	$O(n^3 \log n)$ [4]
	3	fix	–	$O(n^4)$ [3]
	3, 4	fix	Länge	$O(n^6)$ [4]
	1	frei	Länge	$O(n^2)$ [1]
	1	frei	beliebig	$O(n^5)$ [1]
	1	fix	Länge	$O(n \log n)$ [2]
	2	beides	Länge	$O(n^2)$ [2]
	4	fix	Länge	$O(n^2 \log^3 n)$ [2]
	1, 2, 3, 4	beides	Länge	$O(n^{12})$ [4]

[1] Benkert et al. '09

[2] Bekos et al. '07

[3] Kindermann et al. '16

[4] Bose et al. '18

Für beliebige Zielfunktion:
Annahme $f(n) \in O(1)$