

Algorithmen und Datenstrukturen 2 – Übung #1

Ramin Kosfeld und Chek-Manh Loi 17.04.2024

Kleine Übungen

Olone Obully

Mittwoch 15:00-16:30, SN 19.1 Erste Übung: 10.04.2024

Kleine Übungen

Die Anmeldung zu den kleinen Übungen ist abgelaufen. Falls ihr euch zu den kleinen Übungen angemeldet habt, aber *keine* Mail mit einer Übungsgruppe erhalten habt, schreibt bitte schnell eine Mail an Ramin. Dasselbe gilt auch, wenn sich eure Termine verändert haben oder ihr euch nachträglich zu den kleinen Übungen anmelden wollt.

Gruppe	Termin (siehe Semesterplan)	Raum	Tutor
01	Dienstag, 13:15 – 14:45	IZ 305	Tobias Wallner
02	Mittwoch, 16:45 – 18:15	IZ 305	Lisa Glowczewski
03	Donnerstag, 08:00 – 09:30	IZ 305	Tilo Hoitz
04	Donnerstag, 13:15 – 14:45	IZ 305	Kai Kobbe
05	Freitag, 11:30 – 13:00	IZ 305	Benjamin Hennies
06	Freitag: 13:15 – 14:45	IZ 305	Lisa Glowczewski

E-Mail-Adressen sind hier zu finden.

Klausur

Mittwoch 15:00-16:30 Uhr, SN 19 Erste Übung: 10.04.2024

Semesterplan (hier klicken)

Hier gibt es eine vollständige Übe über alle Termine der Veranstaltu

Mailingliste (hier klicken)

Bitte tragt euch hier ein! Hier wer spontane Änderungen etc. bekanntgegeben!

Informationen

Impressum

Datenschutz

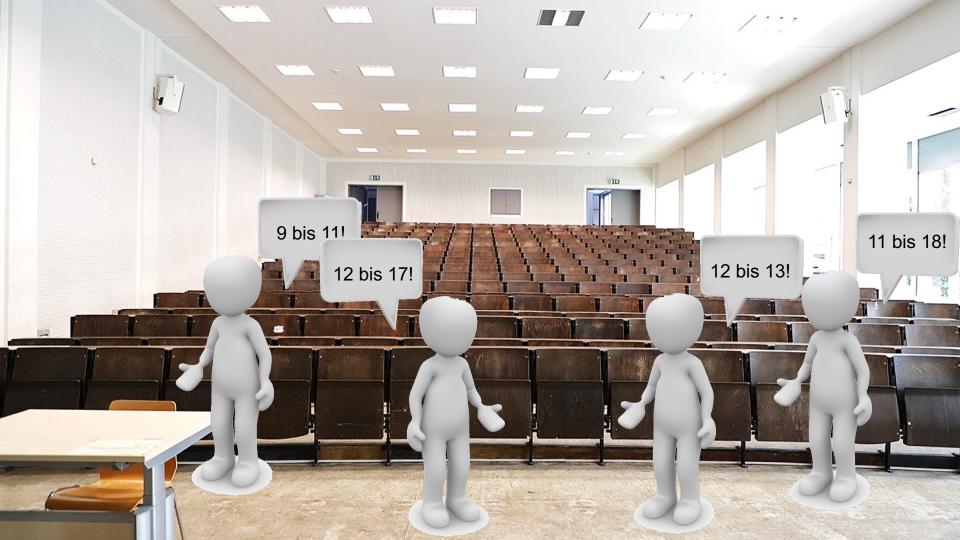


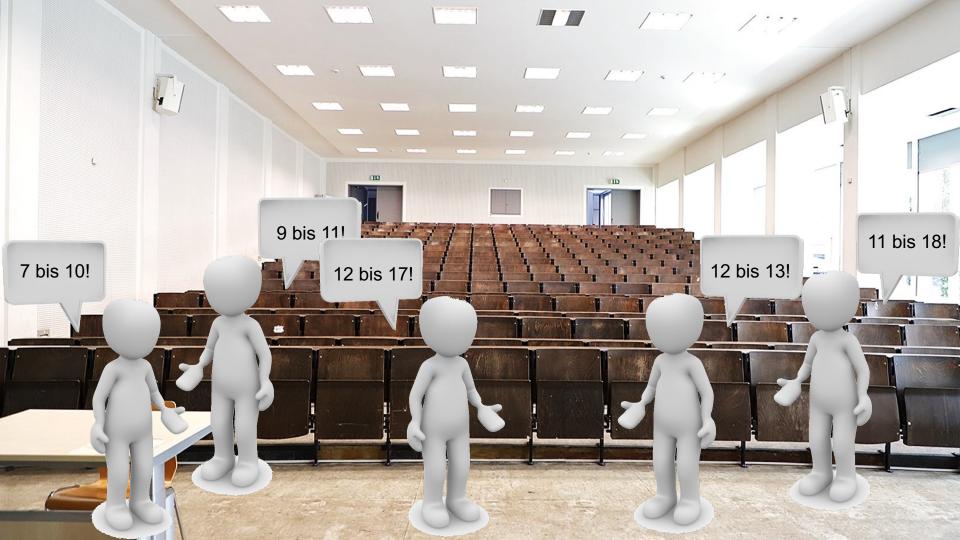


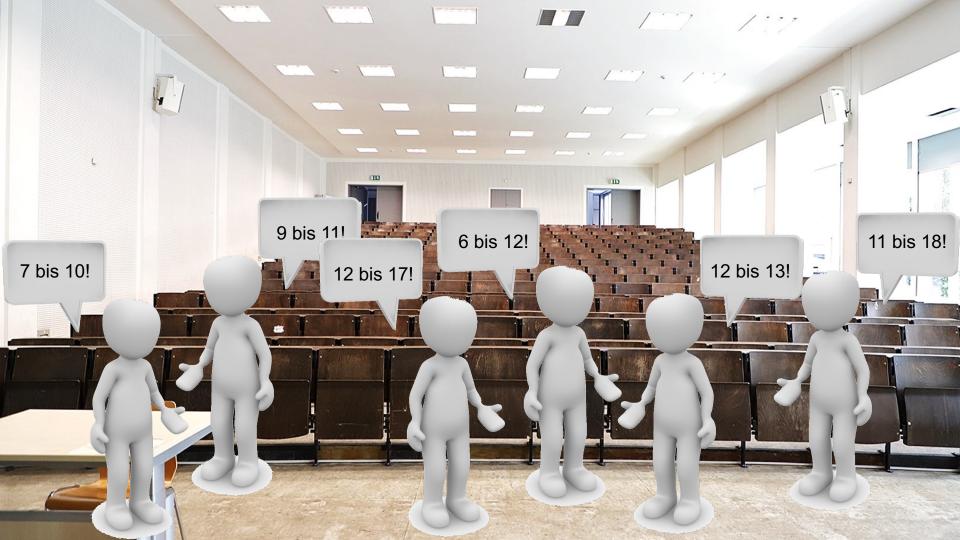


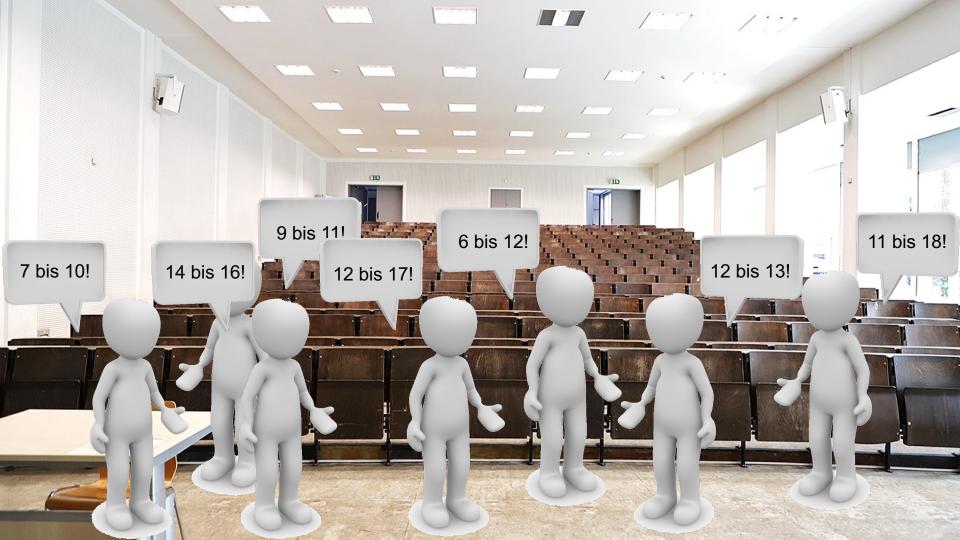




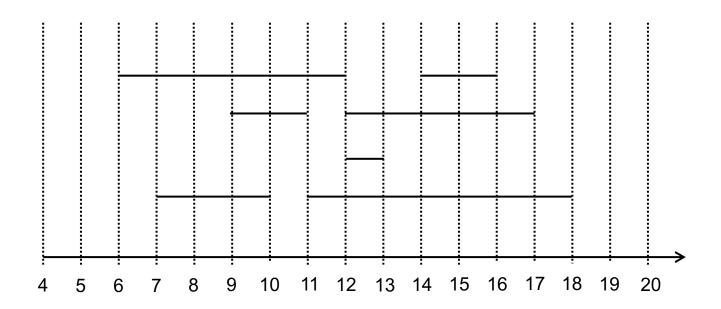






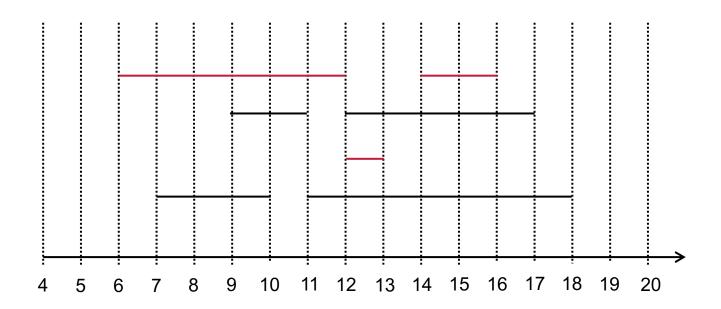


Hörsaal-Belegung





Hörsaal-Belegung







Gegeben



Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gesucht



Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gesucht

Teilmenge $\mathcal{I}' \subseteq \mathcal{I}$ mit den folgenden Eigenschaften

Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gesucht

Teilmenge $\mathcal{I}' \subseteq \mathcal{I}$ mit den folgenden Eigenschaften

1.
$$\forall I_i, I_j \in \mathcal{I}': I_i \cap I_j = \emptyset$$

Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gesucht

Teilmenge $\mathcal{I}' \subseteq \mathcal{I}$ mit den folgenden Eigenschaften

1.
$$\forall I_i, I_i \in \mathcal{I}': I_i \cap I_i = \emptyset$$

Die ausgewählten Intervalle müssen disjunkt sein.

Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gesucht

Teilmenge $\mathcal{I}' \subseteq \mathcal{I}$ mit den folgenden Eigenschaften

- 1. $\forall I_i, I_i \in \mathcal{I}': I_i \cap I_i = \emptyset$
- 2. \mathcal{I}' ist größtmöglich

Die ausgewählten Intervalle müssen disjunkt sein.

Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gesucht

Teilmenge $\mathcal{I}' \subseteq \mathcal{I}$ mit den folgenden Eigenschaften

- 1. $\forall I_i, I_i \in \mathcal{I}': I_i \cap I_i = \emptyset$
- 2. \mathcal{I}' ist größtmöglich

Die ausgewählten Intervalle müssen disjunkt sein.

Wie löst man das Problem?





Gegeben

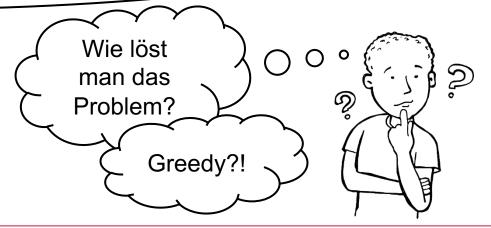
Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gesucht

Teilmenge $\mathcal{I}' \subseteq \mathcal{I}$ mit den folgenden Eigenschaften

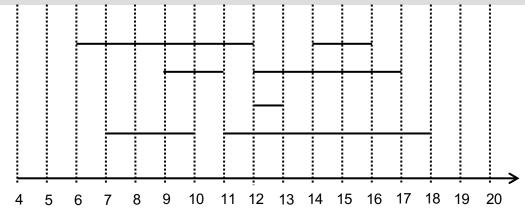
- 1. $\forall I_i, I_j \in \mathcal{I}': I_i \cap I_j = \emptyset$
- 2. \mathcal{I}' ist größtmöglich

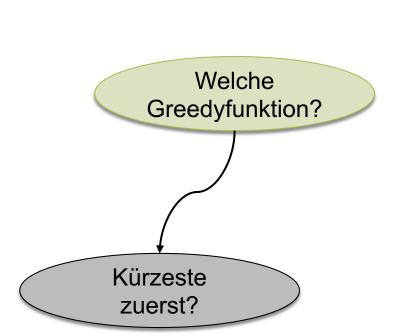
Die ausgewählten Intervalle müssen disjunkt sein.

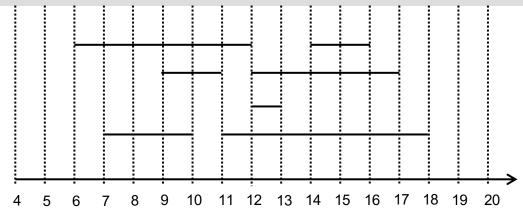


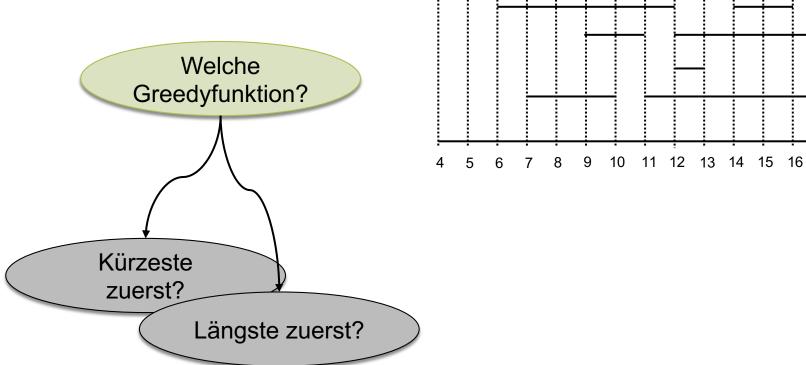


Welche Greedyfunktion?



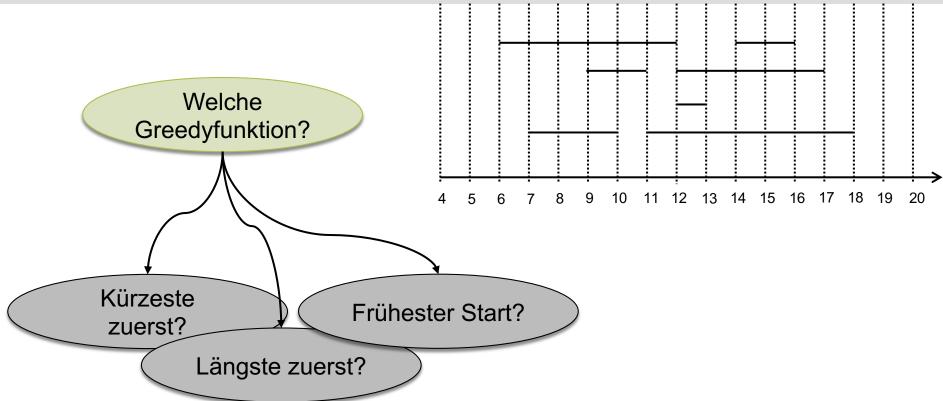




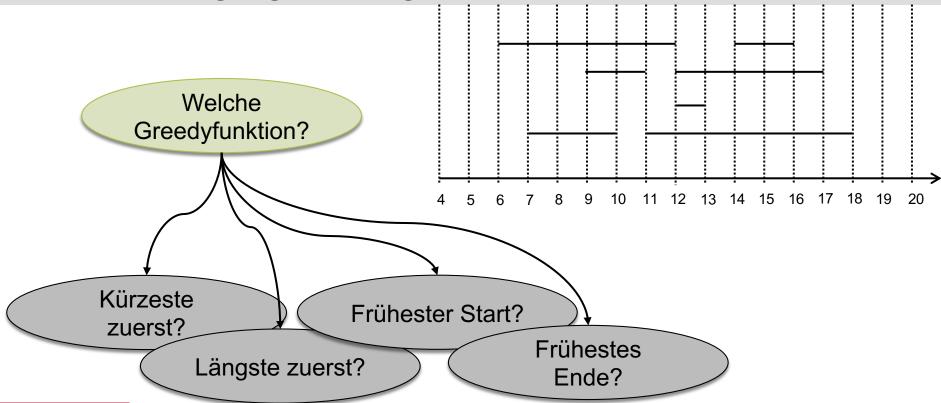


20

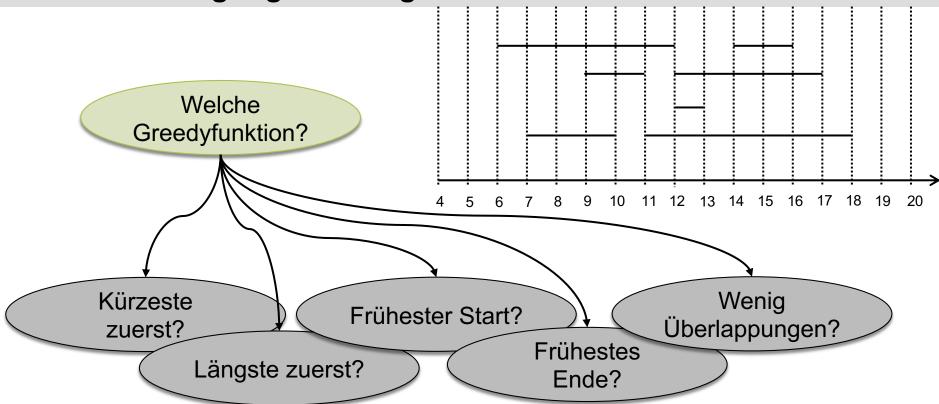




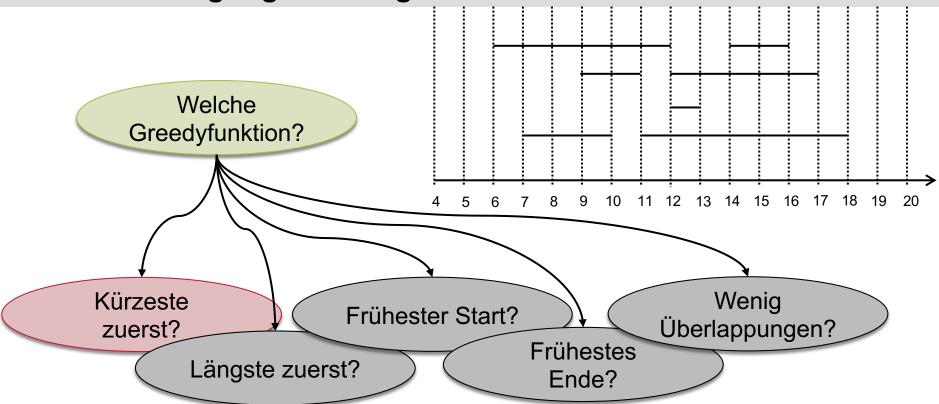






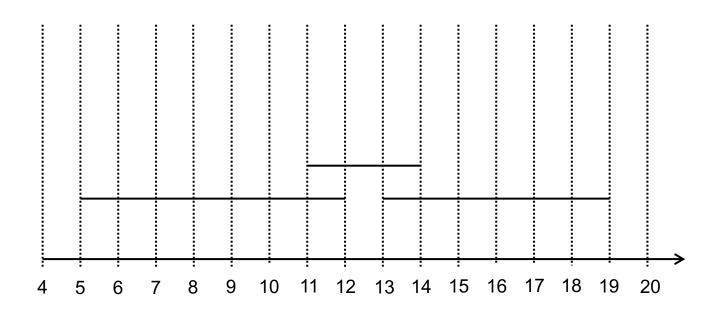






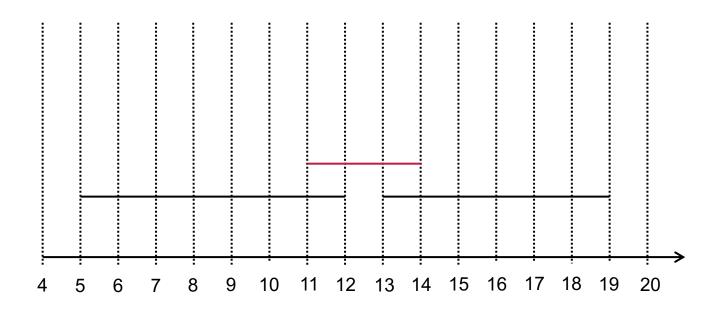


Hörsaal-Belegung – Kürzeste zuerst





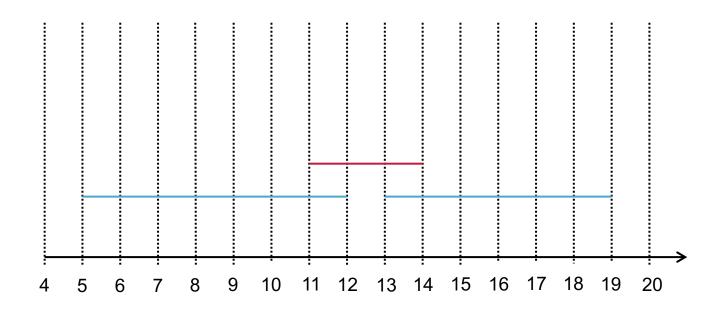
Hörsaal-Belegung – Kürzeste zuerst



ALG = 1

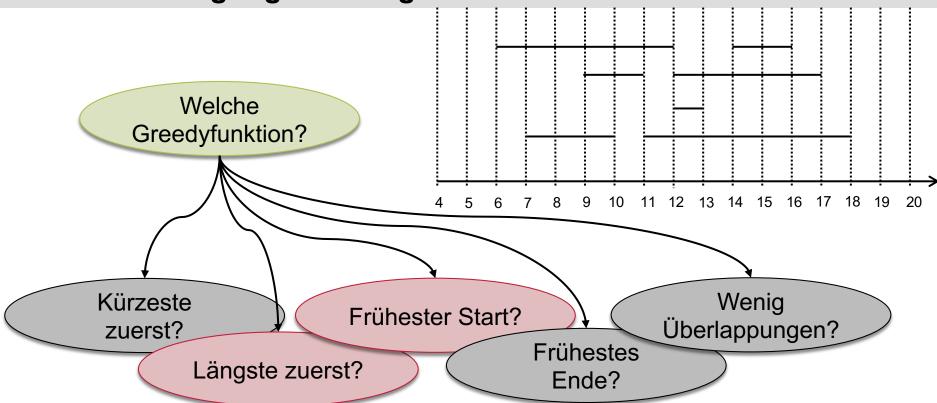


Hörsaal-Belegung – Kürzeste zuerst



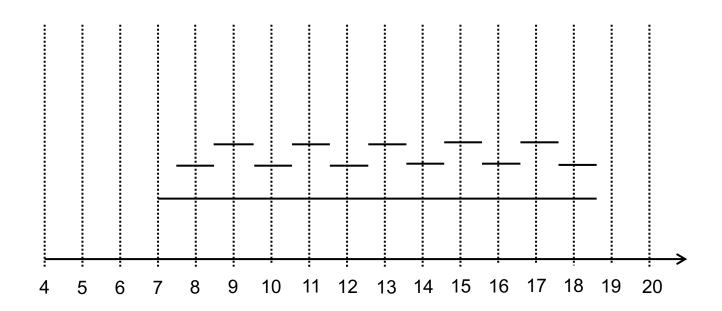


$$Opt = 2$$



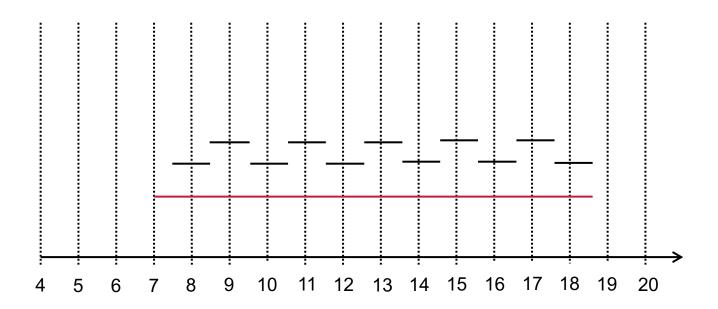


Hörsaal-Belegung – Frühester Start/Längstes Intervall





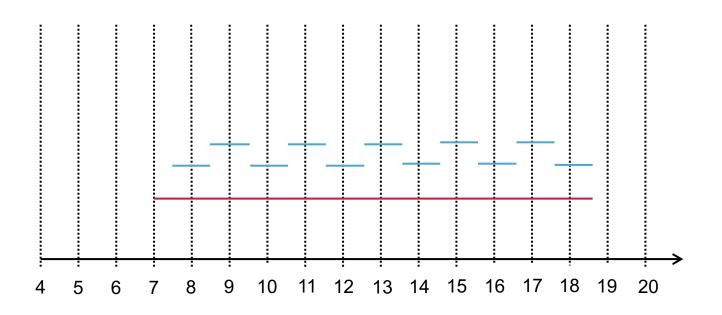
Hörsaal-Belegung – Frühester Start/Längstes Intervall



ALG = 1



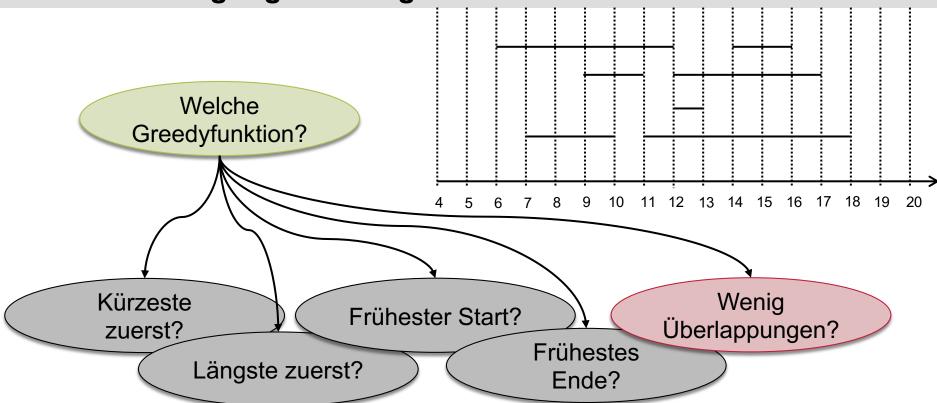
Hörsaal-Belegung – Frühester Start/Längstes Intervall



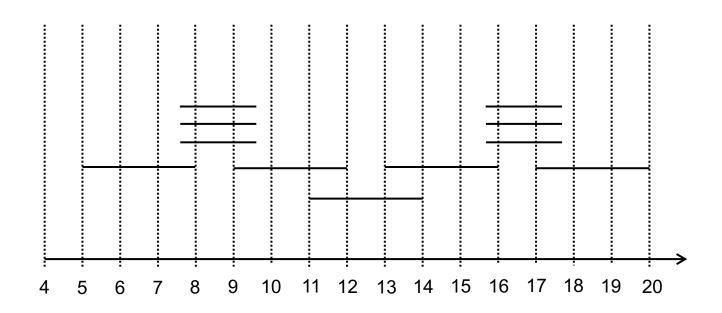
ALG = 1

Opt = 11

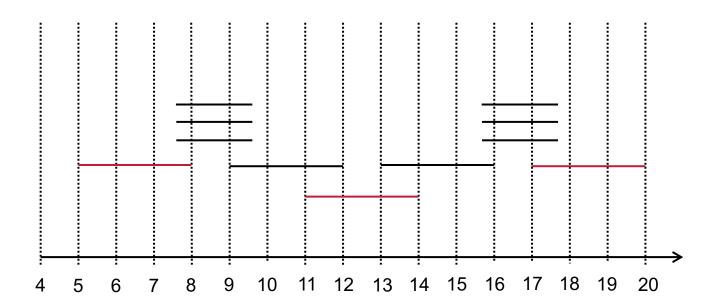
Hörsaal-Belegung – Strategien





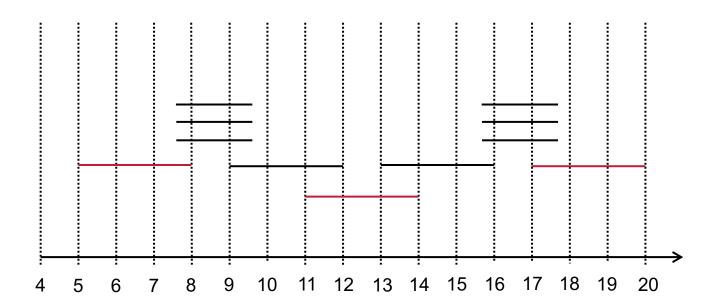






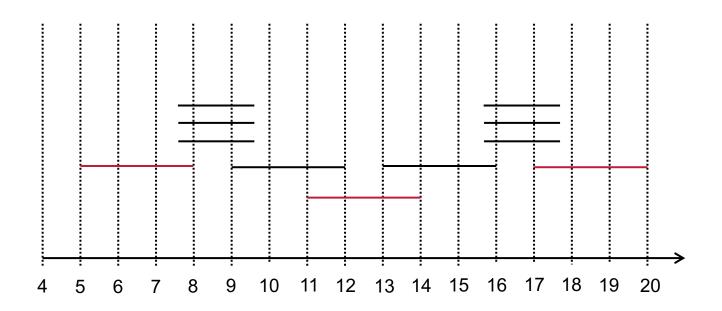
ALG = 3





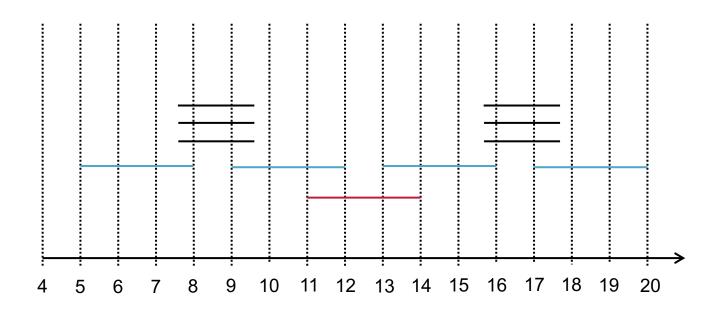
ALG = 3





ALG = 3

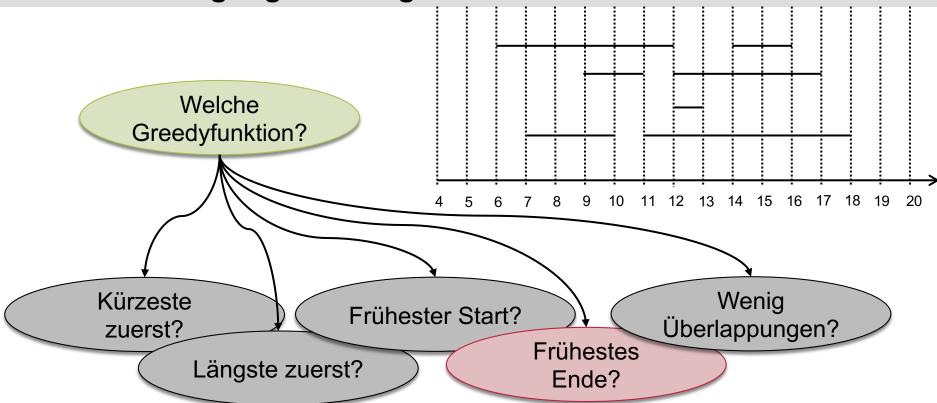
Opt = 4



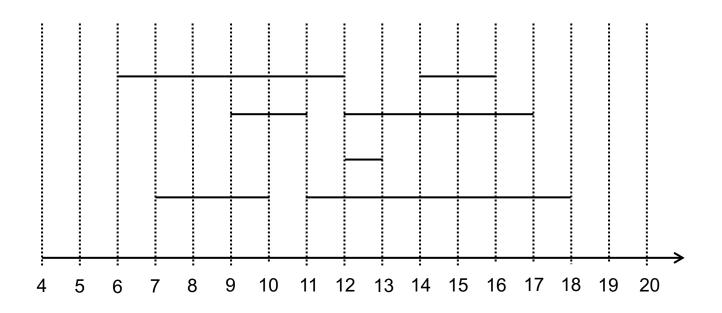
ALG = 3

Opt = 4

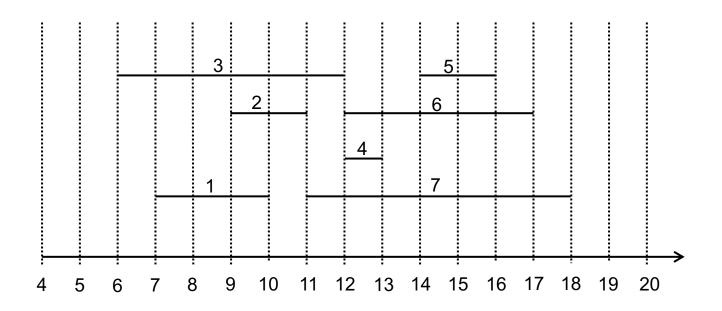
Hörsaal-Belegung – Strategien



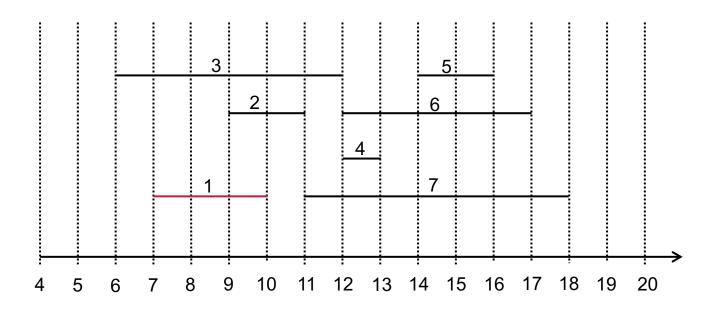




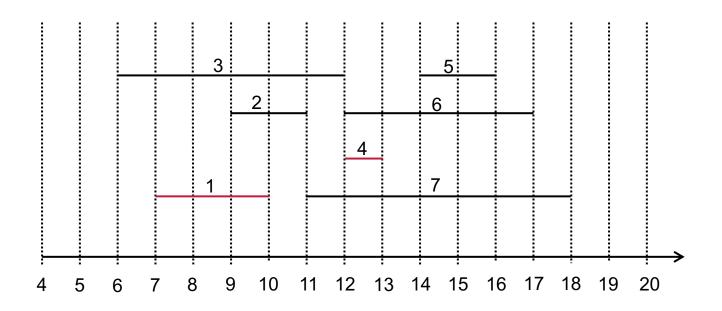




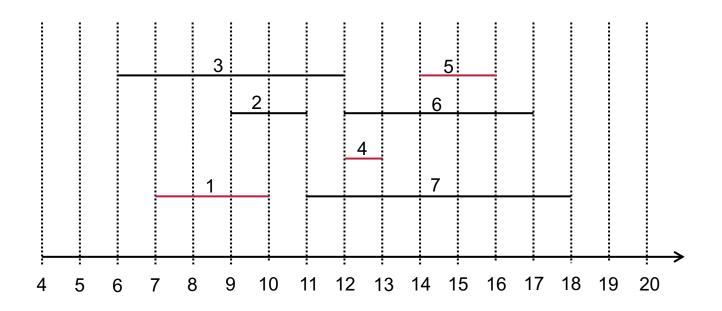






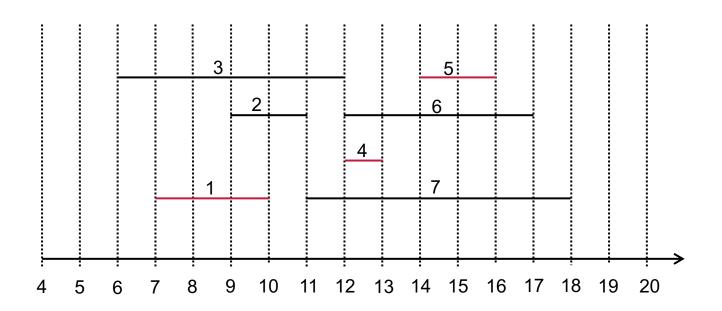






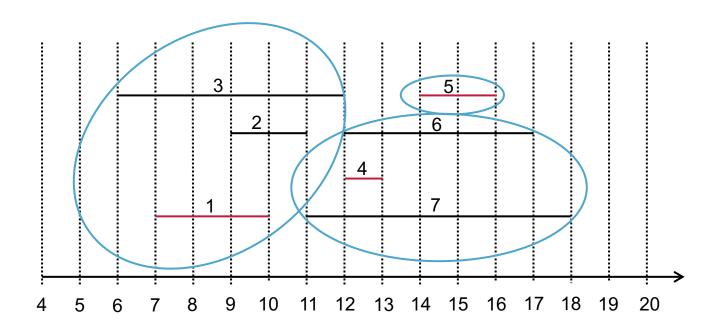
ALG = 3





ALG = 3

Opt = 3?



ALG = 3

OPT = 3?



Greedy-Algorithmus: Frühestes Ende

1. Sortiere $\{1, ..., n\}$ nach e_i aufsteigend; erhalte Permutation $\pi(1), ..., \pi(n)$

- 1. Sortiere $\{1, ..., n\}$ nach e_i aufsteigend; erhalte Permutation $\pi(1), ..., \pi(n)$
- 2. $S := {\pi(1)}$

- 1. Sortiere $\{1, ..., n\}$ nach e_i aufsteigend; erhalte Permutation $\pi(1), ..., \pi(n)$
- 2. $S := {\pi(1)}$
- 3. $e \coloneqq e_{\pi(1)}$

- 1. Sortiere $\{1, ..., n\}$ nach e_i aufsteigend; erhalte Permutation $\pi(1), ..., \pi(n)$
- 2. $S := {\pi(1)}$
- 3. $e \coloneqq e_{\pi(1)}$
- 4. for k = 2 to n do

- 1. Sortiere $\{1, ..., n\}$ nach e_i aufsteigend; erhalte Permutation $\pi(1), ..., \pi(n)$
- 2. $S := {\pi(1)}$
- 3. $e \coloneqq e_{\pi(1)}$
- 4. for k = 2 to n do
- 5. if $(s_{\pi(k)} \ge e)$ then

- 1. Sortiere $\{1, ..., n\}$ nach e_i aufsteigend; erhalte Permutation $\pi(1), ..., \pi(n)$
- 2. $S := {\pi(1)}$
- 3. $e \coloneqq e_{\pi(1)}$
- 4. for k = 2 to n do
- 5. if $(s_{\pi(k)} \ge e)$ then
- 6. $S := S \cup \{\pi(k)\}$

- 1. Sortiere $\{1, ..., n\}$ nach e_i aufsteigend; erhalte Permutation $\pi(1), ..., \pi(n)$
- 2. $S := {\pi(1)}$
- 3. $e \coloneqq e_{\pi(1)}$
- 4. for k = 2 to n do
- 5. if $(s_{\pi(k)} \ge e)$ then
- 6. $S \coloneqq S \cup \{\pi(k)\}$
- 7. $e \coloneqq e_{\pi(k)}$

```
1. Sortiere \{1, \ldots, n\} nach e_i aufsteigend; erhalte Permutation \pi(1), \ldots, \pi(n)

2. S \coloneqq \{\pi(1)\}

3. e \coloneqq e_{\pi(1)}

4. for k = 2 to n do

5. if (s_{\pi(k)} \ge e) then

6. S \coloneqq S \cup \{\pi(k)\}

7. e \coloneqq e_{\pi(k)}

8. return S
```

Greedy-Algorithmus: Frühestes Ende

- 1. Sortiere $\{1, ..., n\}$ nach e_i aufsteigend; erhalte Permutation $\pi(1), ..., \pi(n)$
- 2. $S := {\pi(1)}$
- 3. $e \coloneqq e_{\pi(1)}$
- 4. for k=2 to n do
- 5. if $(s_{\pi(k)} \ge e)$ then
- 6. $S \coloneqq S \cup \{\pi(k)\}$
- 7. $e \coloneqq e_{\pi(k)}$
- 8. **return** *S*

Theorem

Der Algorithmus löst das Hörsaal-Problem optimal in Zeit $O(n \log n)$

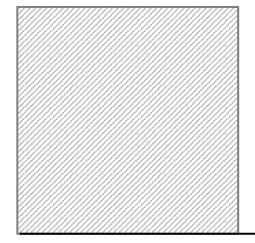


Annahme: Algorithmus 1 sei nicht optimal.

Annahme: Algorithmus 1 sei nicht optimal.



Annahme: Algorithmus 1 sei nicht optimal.



Annahme: Algorithmus 1 sei nicht optimal.

Sei S_{alg} die Lösung von Algorithmus 1 und S_{opt} eine optimale Lösung.

Hier sind S_{alg} und S_{opt} identisch



Annahme: Algorithmus 1 sei nicht optimal.

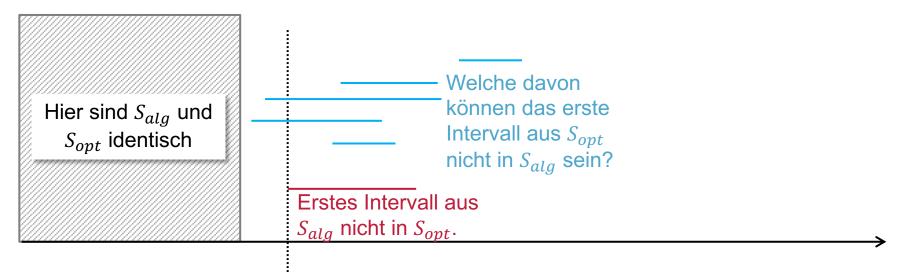
Sei S_{alg} die Lösung von Algorithmus 1 und S_{opt} eine optimale Lösung.

Hier sind S_{alg} und S_{opt} identisch

Erstes Intervall aus S_{alg} nicht in S_{opt} .

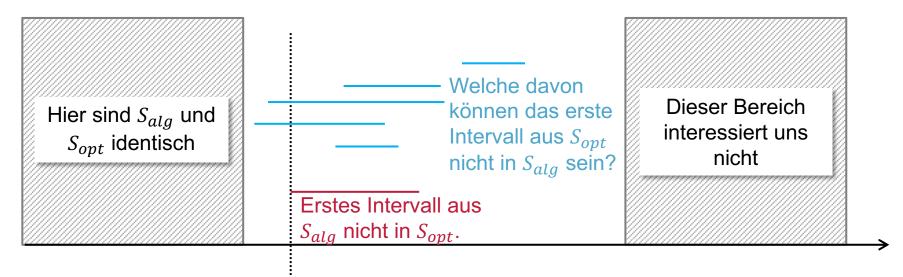


Annahme: Algorithmus 1 sei nicht optimal.



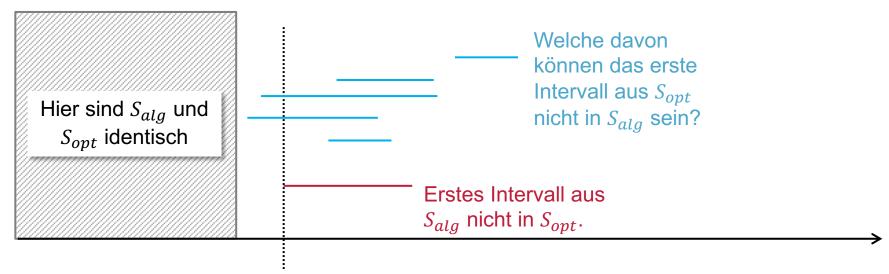


Annahme: Algorithmus 1 sei nicht optimal.



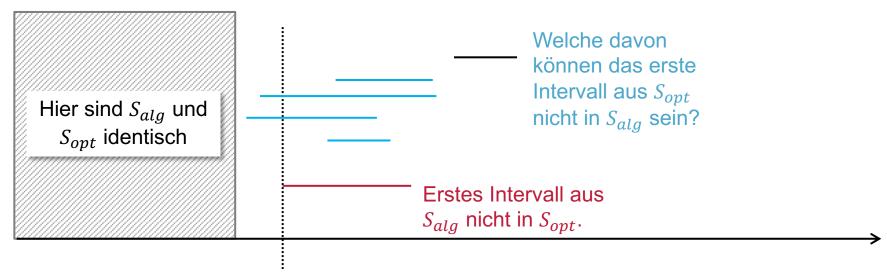


Annahme: Algorithmus 1 sei nicht optimal.



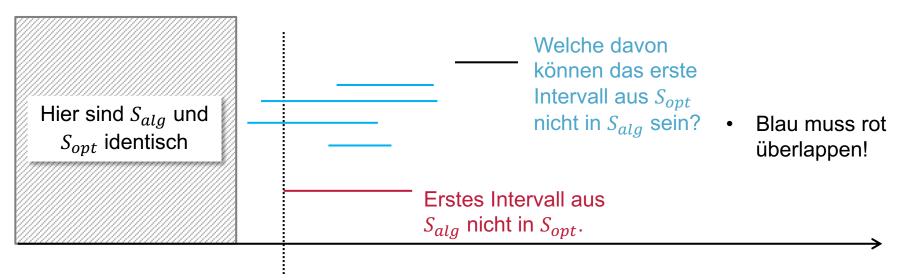


Annahme: Algorithmus 1 sei nicht optimal.





Annahme: Algorithmus 1 sei nicht optimal.



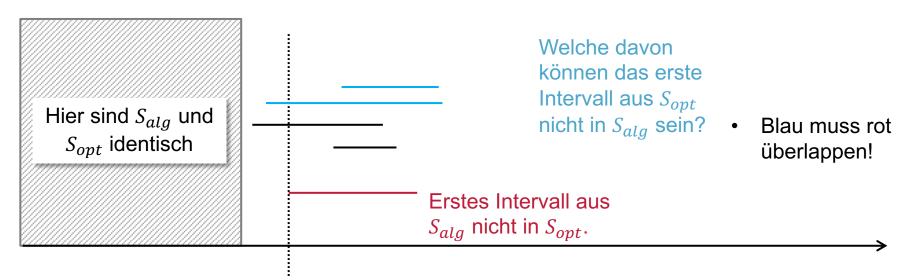


Annahme: Algorithmus 1 sei nicht optimal.





Annahme: Algorithmus 1 sei nicht optimal.





Annahme: Algorithmus 1 sei nicht optimal.



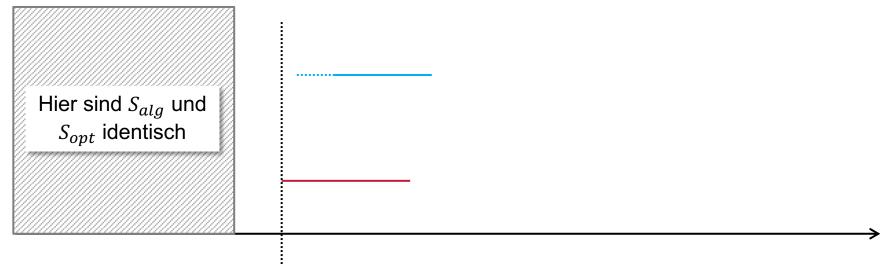


Annahme: Algorithmus 1 sei nicht optimal.



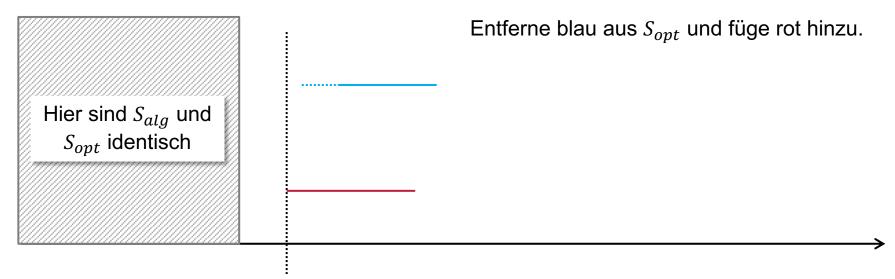


Annahme: Algorithmus 1 sei nicht optimal.



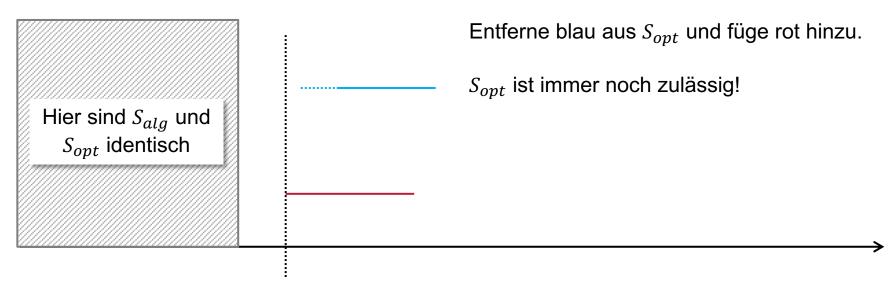


Annahme: Algorithmus 1 sei nicht optimal.



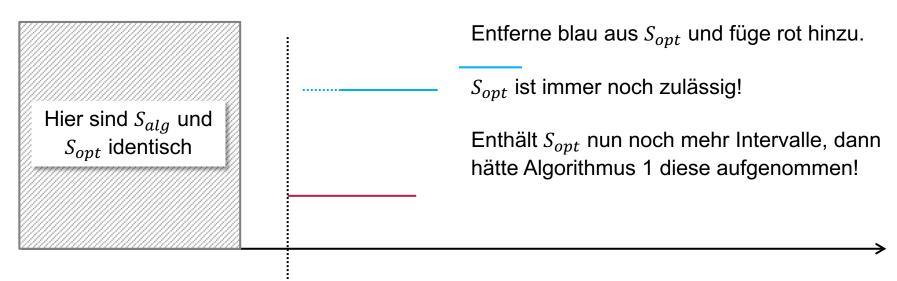


Annahme: Algorithmus 1 sei nicht optimal.



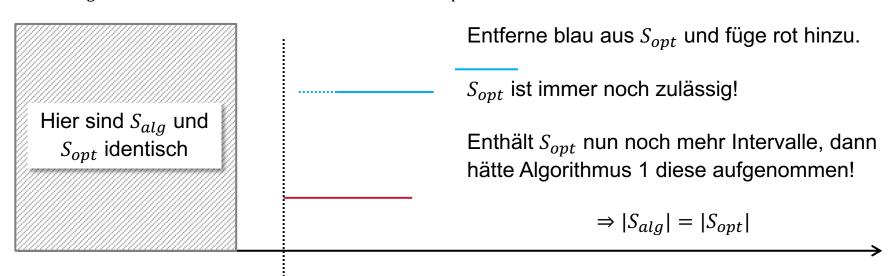


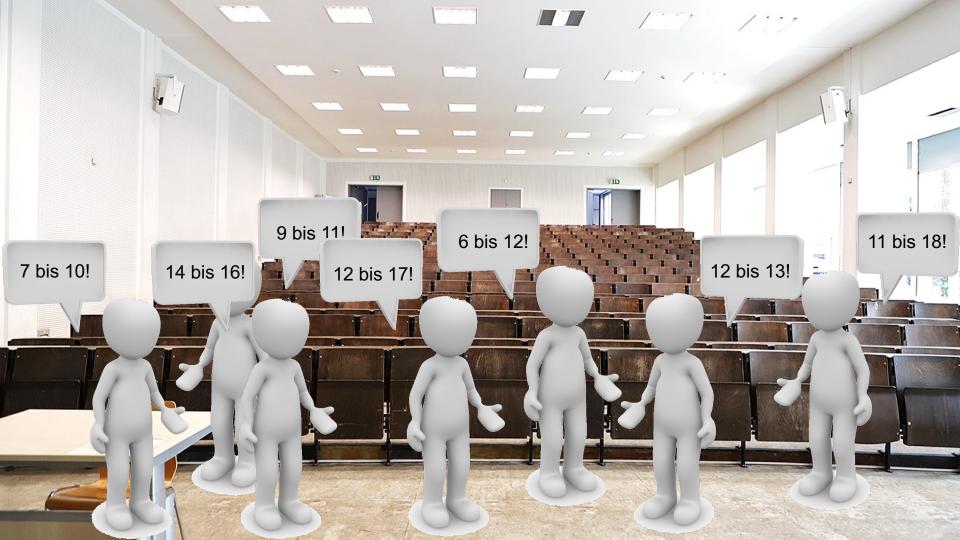
Annahme: Algorithmus 1 sei nicht optimal.





Annahme: Algorithmus 1 sei nicht optimal.

















Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gegeben

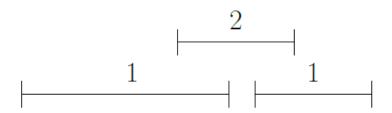
Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), ..., I_n = [s_n, e_n)\}$

Gesucht

Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), ..., I_n = [s_n, e_n)\}$

Gesucht



Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Gesucht

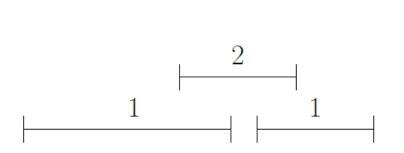


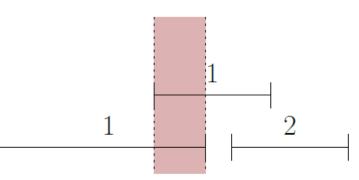
Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

Können jedem Intervall (Veranstaltung) k verschiedene Werte (Räume) zuordnen mit f

Gesucht





Gegeben

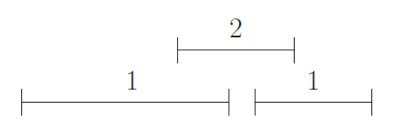
Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

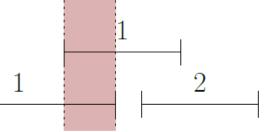
Können jedem Intervall (Veranstaltung) k verschiedene Werte (Räume) zuordnen mit f

Gesucht

Die kleinste Zahl $k \in \mathbb{N}$, sodass eine Funktion $f: I \to \{1, ..., k\}$ existiert und für je zwei Intervalle I und I' gilt $I \cap I' \neq \emptyset \Rightarrow f(I) \neq f(I')$

Wenn zwei Intervalle überlappen, haben sie verschiedenen Wert in *f* (anderen Raum)





Gegeben

Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

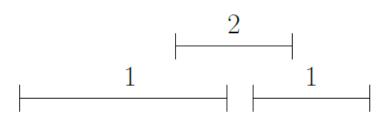
Können jedem Intervall (Veranstaltung) k verschiedene Werte (Räume) zuordnen mit f

Gesucht

Die kleinste Zahl $k \in \mathbb{N}$, sodass eine Funktion $f: I \to \{1, ..., k\}$ existiert und für je zwei Intervalle I und I' gilt $I \cap I' \neq \emptyset \Rightarrow f(I) \neq f(I')$

Minimiere k!

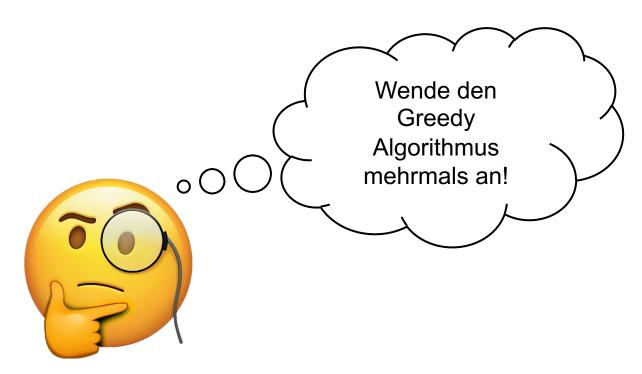
Wenn zwei Intervalle überlappen, haben sie verschiedenen Wert in *f* (anderen Raum)



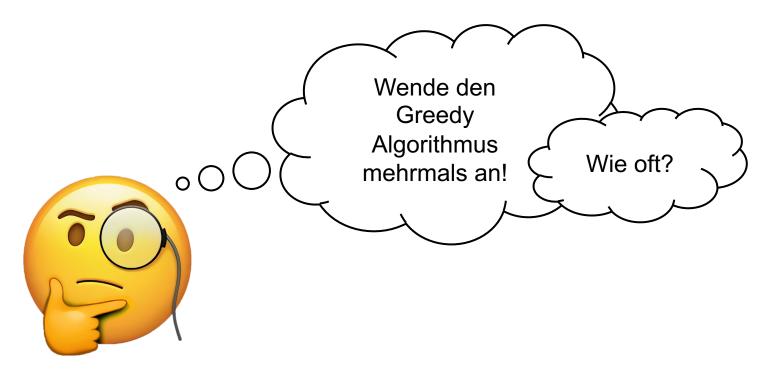




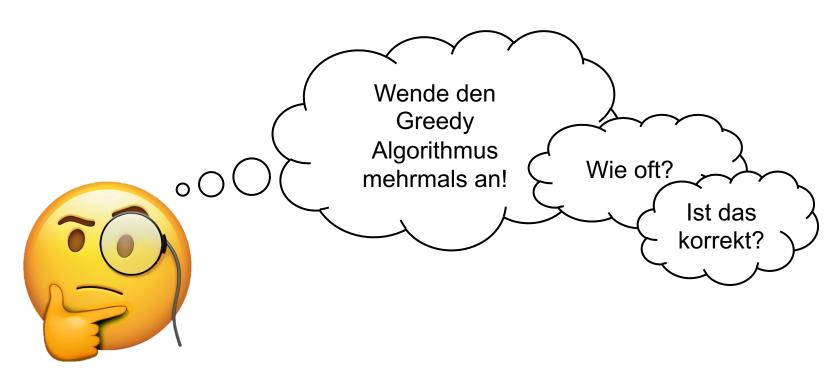




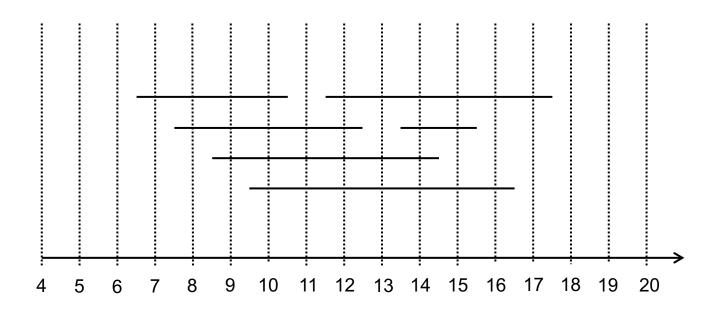




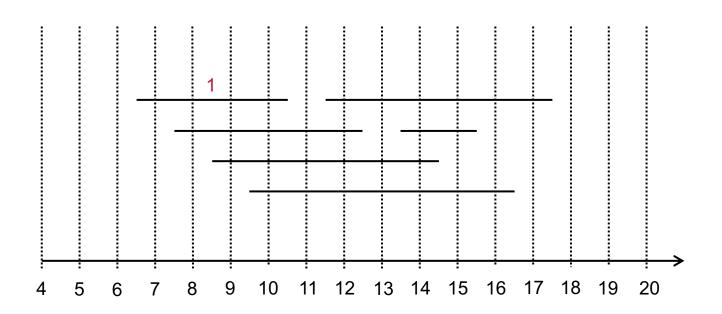




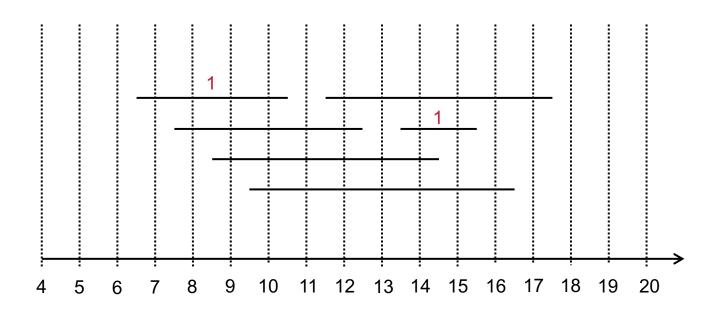




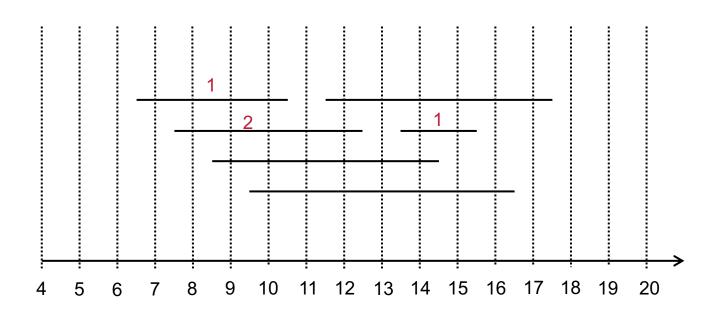




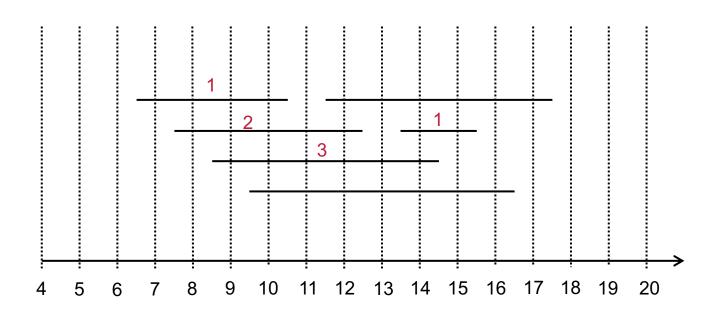




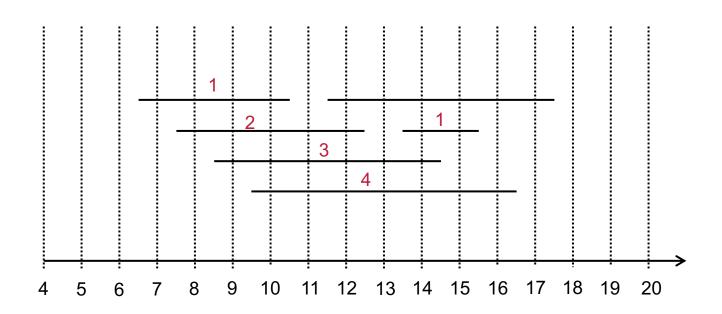




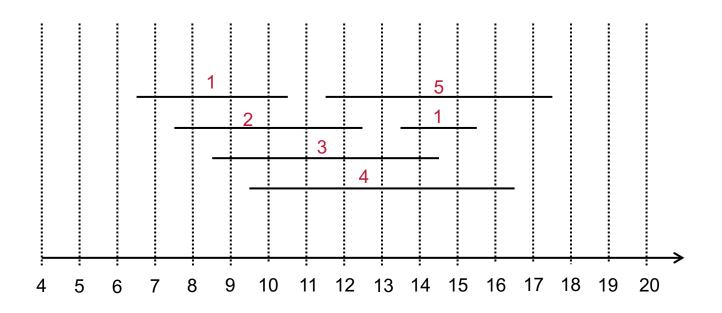




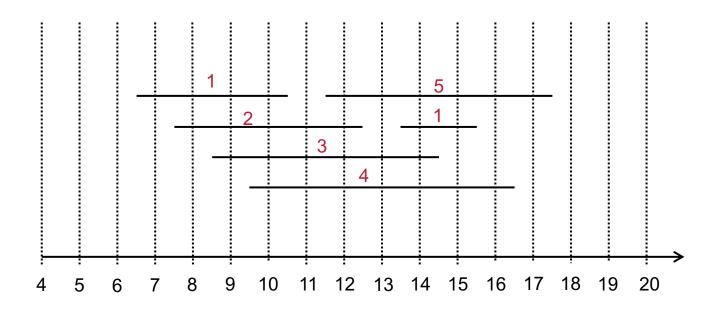






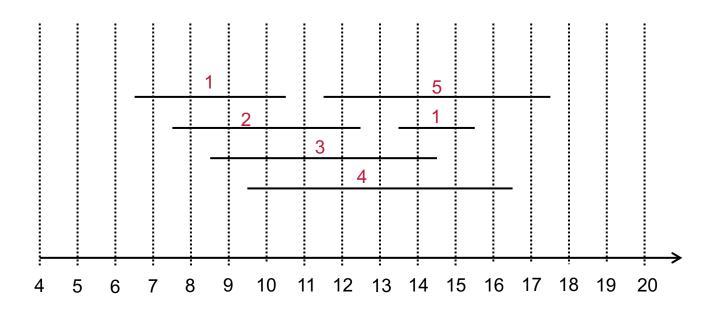






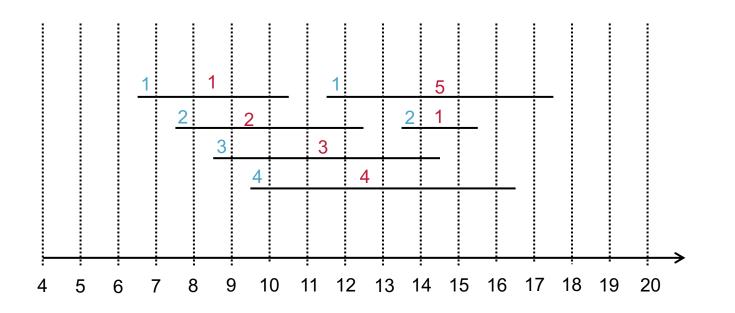
ALG = 5





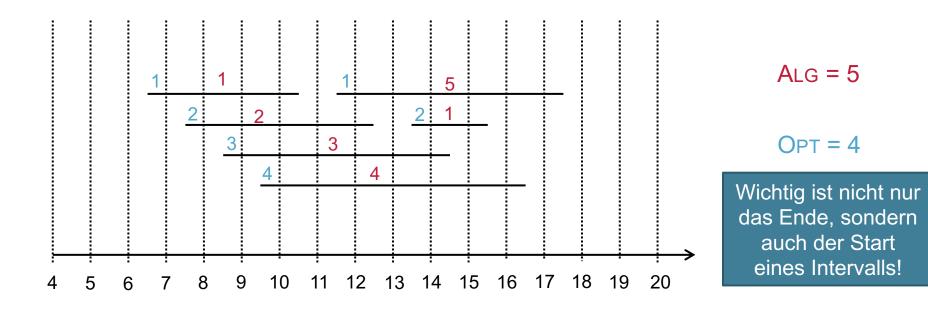


$$OPT = 4$$

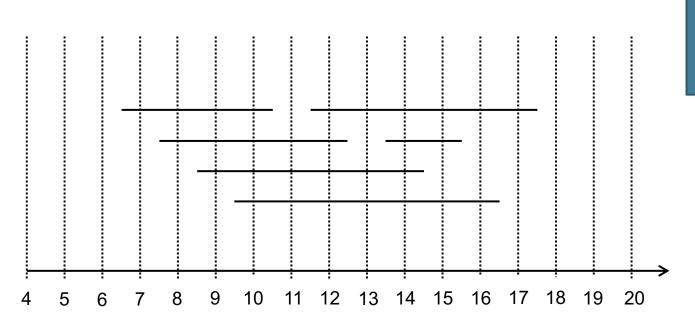


ALG = 5

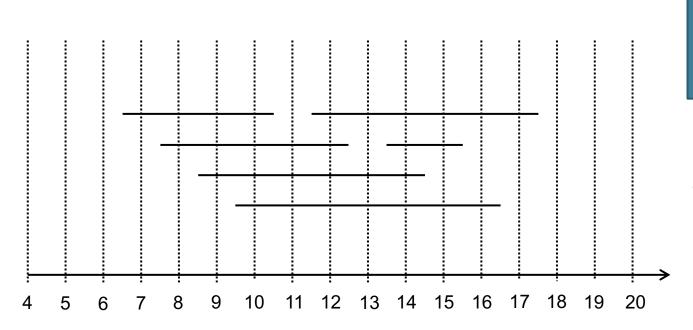
OPT = 4







Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

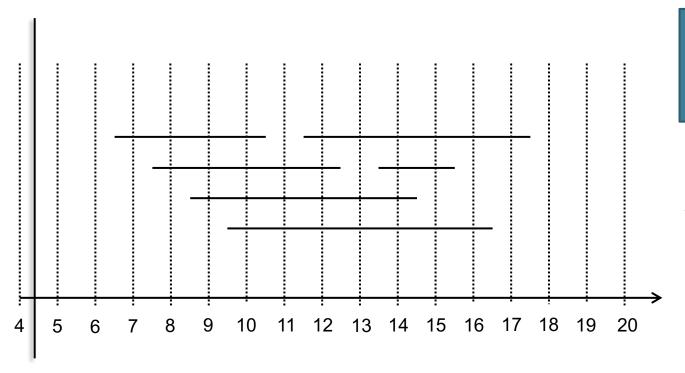


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Idee:

Bewege eine Linie von links nach rechts



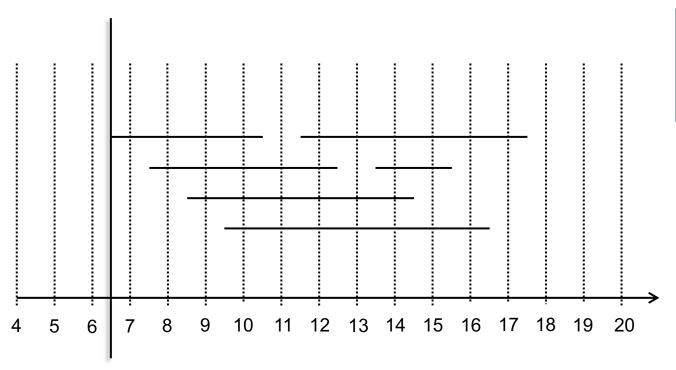


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

ldee:

Bewege eine Linie von links nach rechts

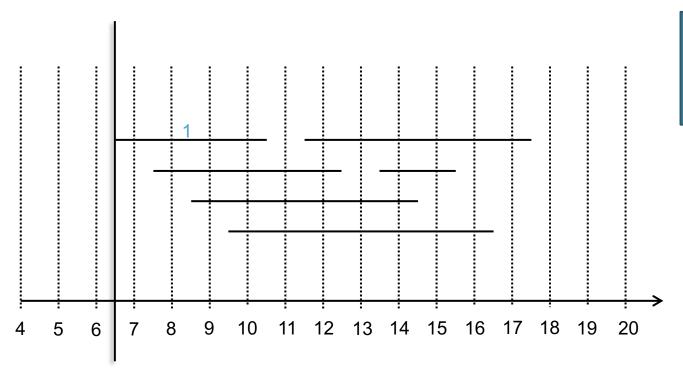




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

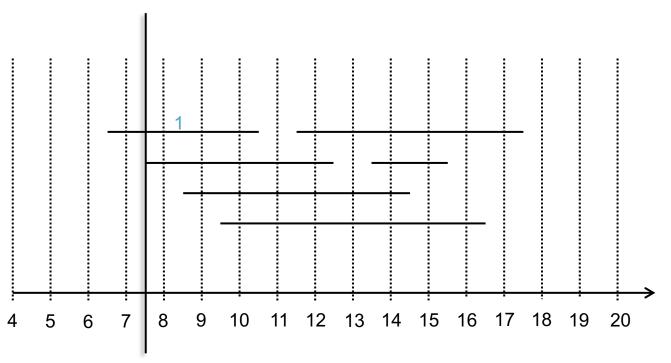




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei



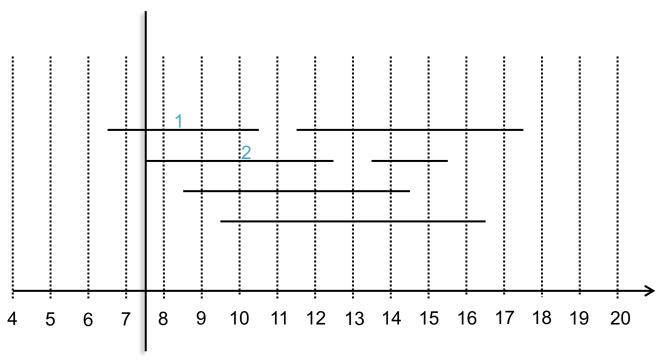


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:

 Gib Raum frei

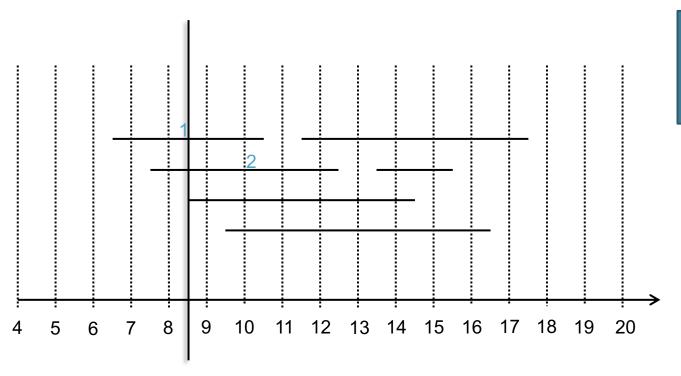




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei



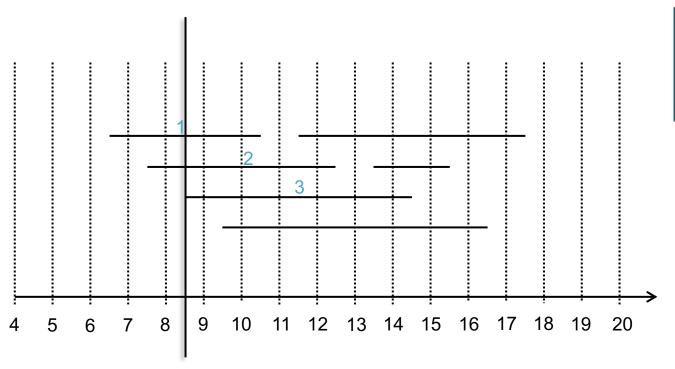


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:

 Gib Raum frei

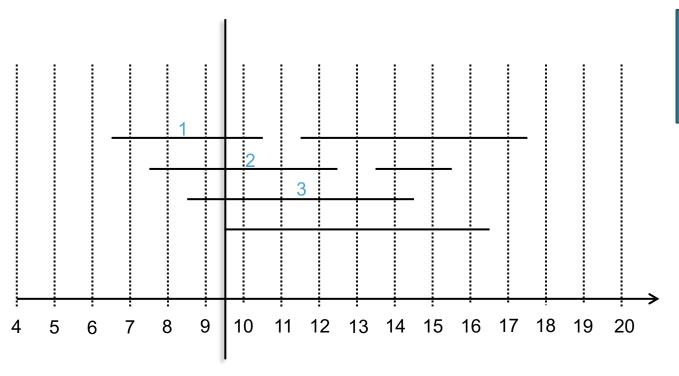




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

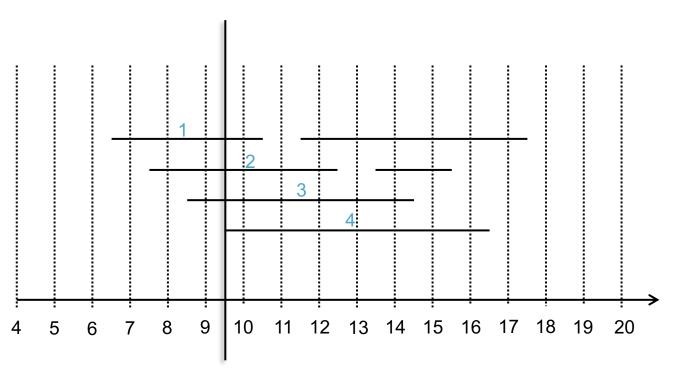




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

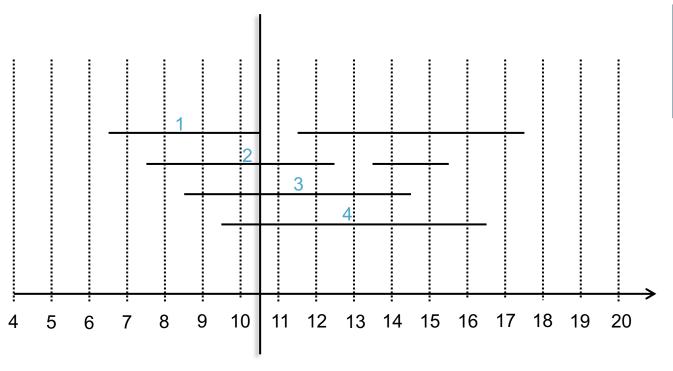




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

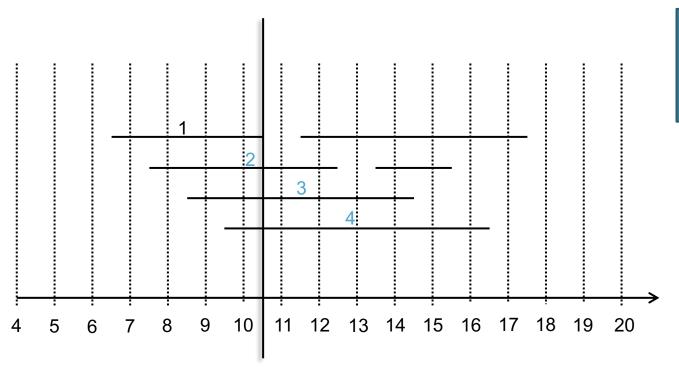




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei



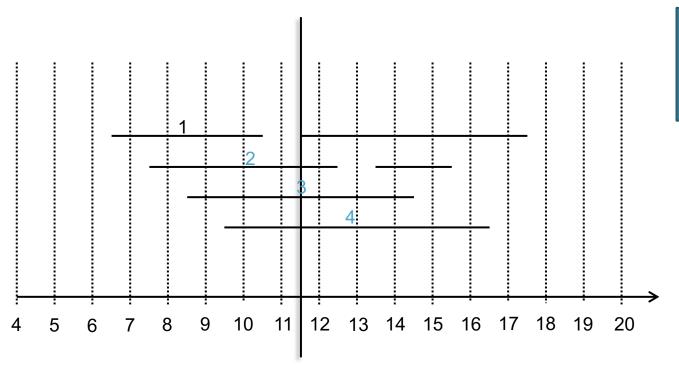


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:

 Gib Raum frei

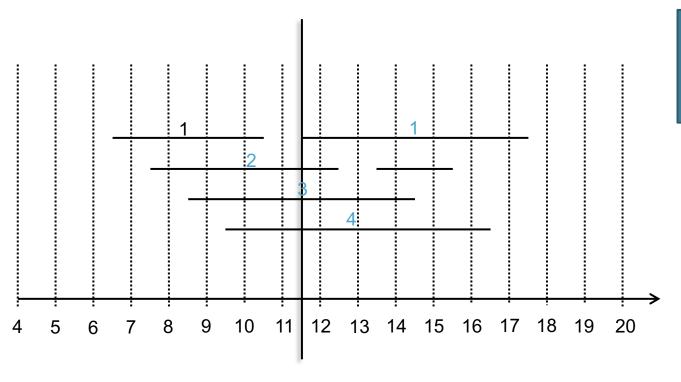




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

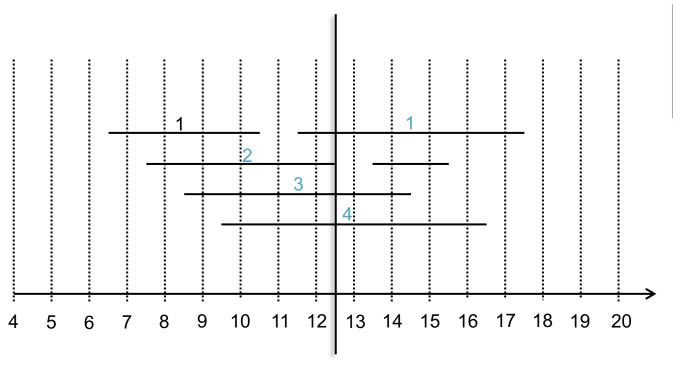




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

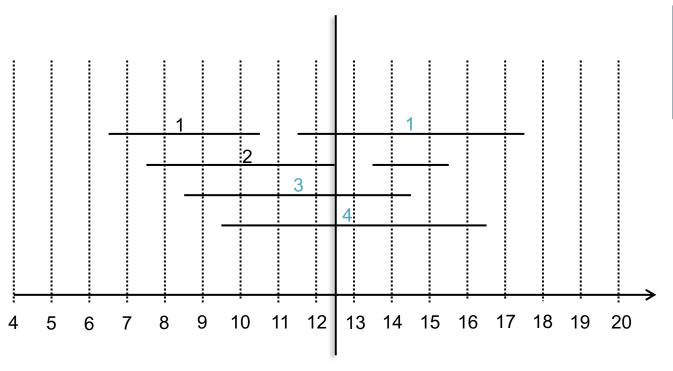
- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei





Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

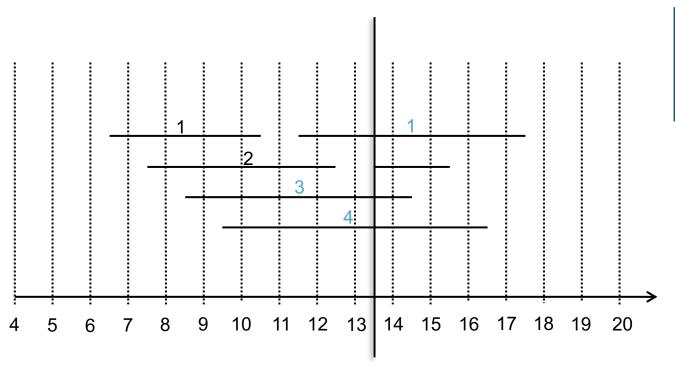
- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei



Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

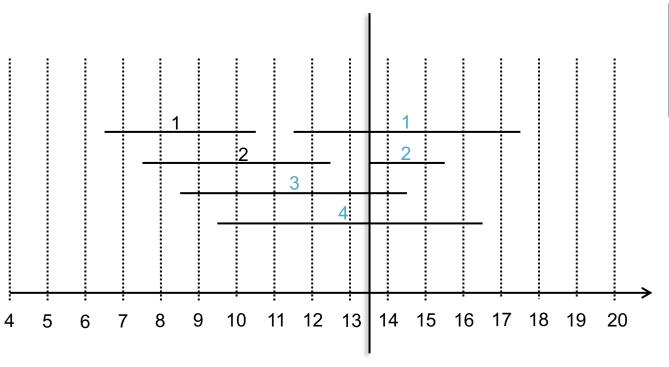




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

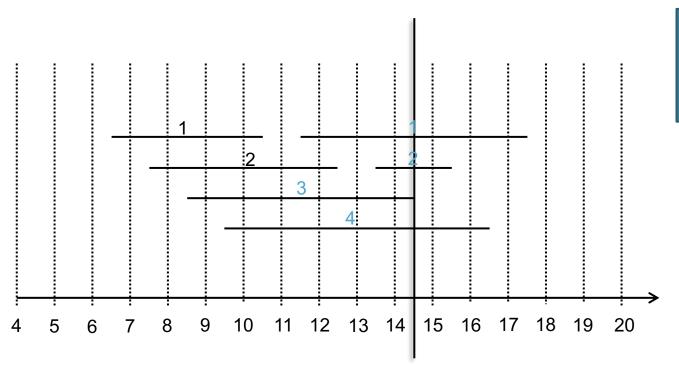




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

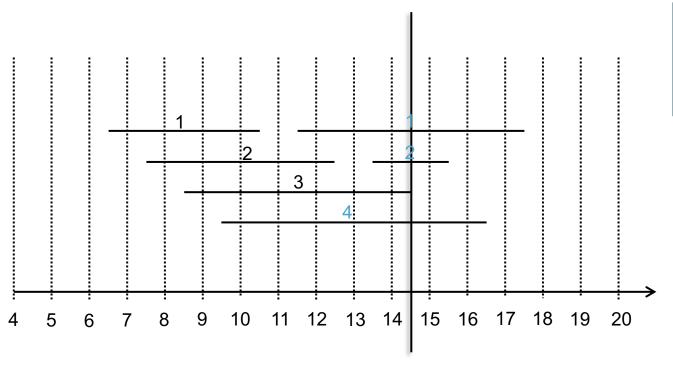




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei



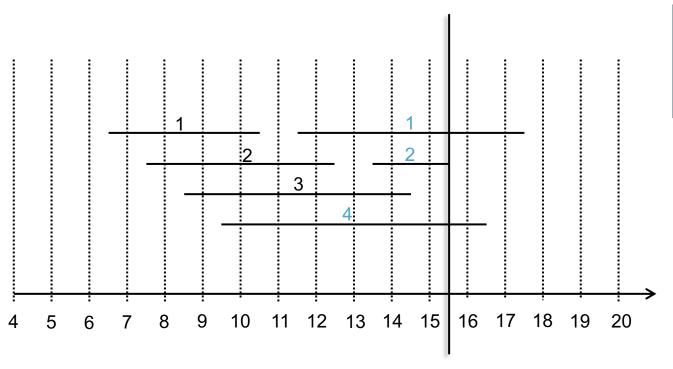


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:

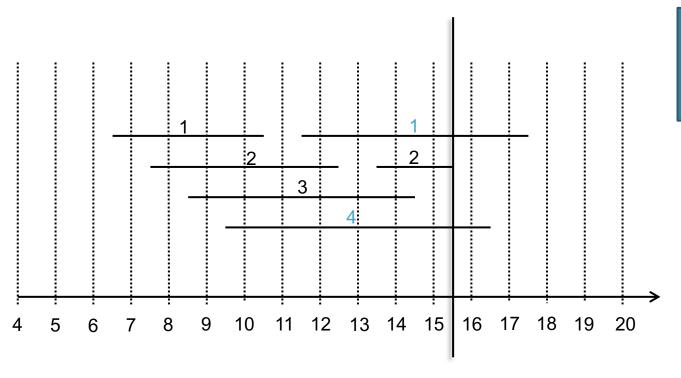
 Gib Raum frei





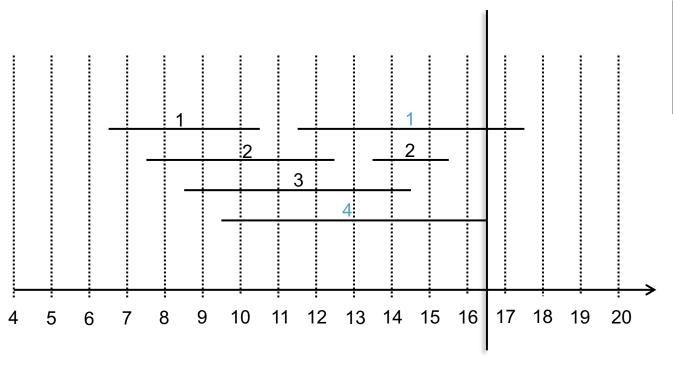
Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei



Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

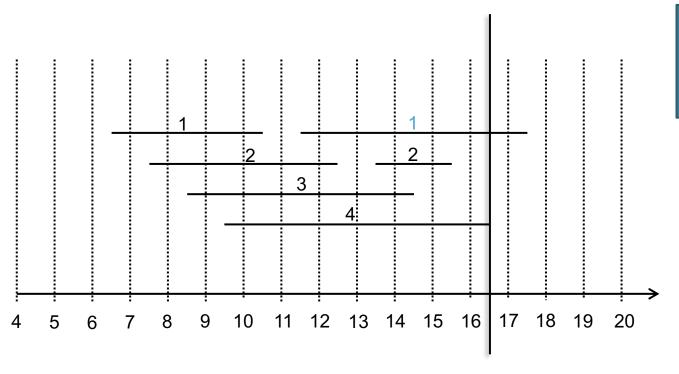
- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei



Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

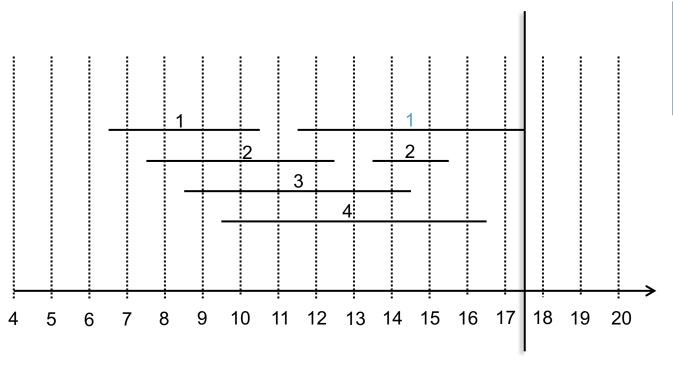
- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei





Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

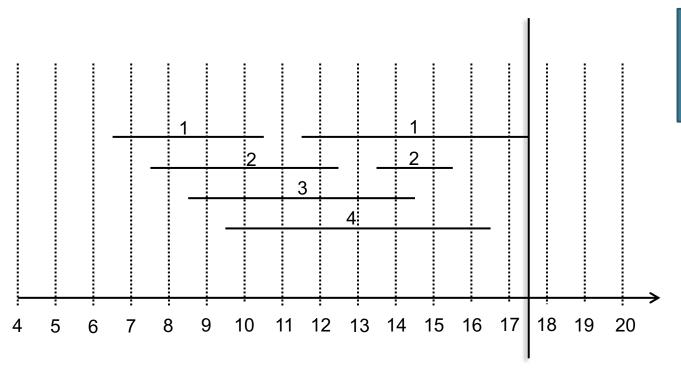
- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei



Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

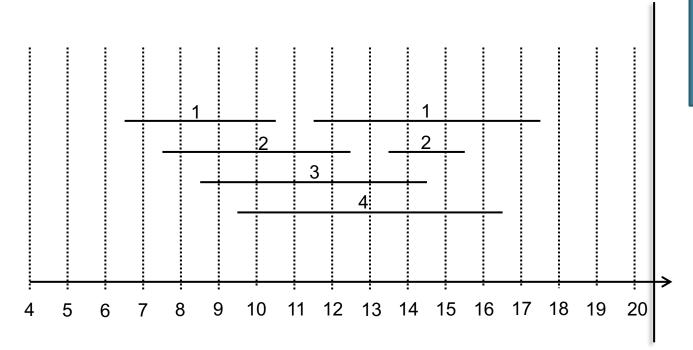




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

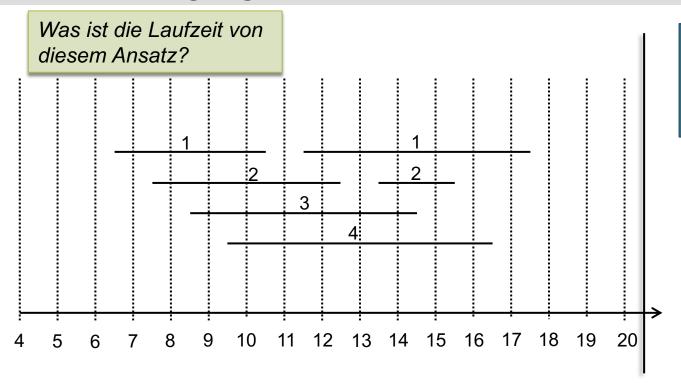




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

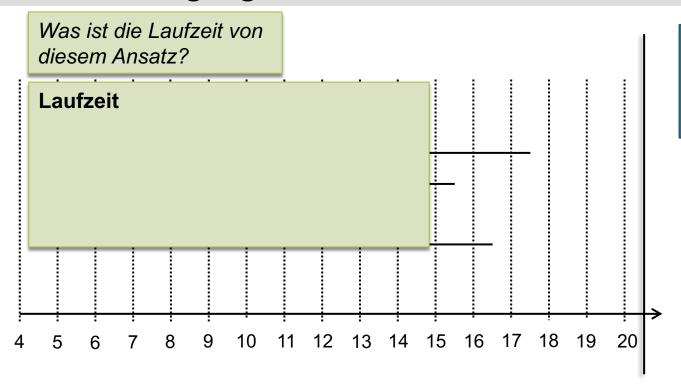




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei

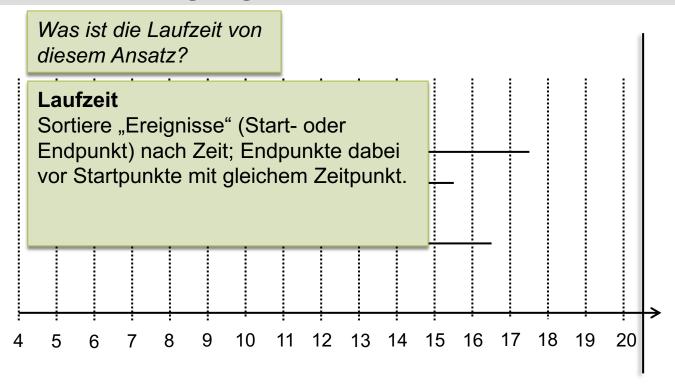




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei



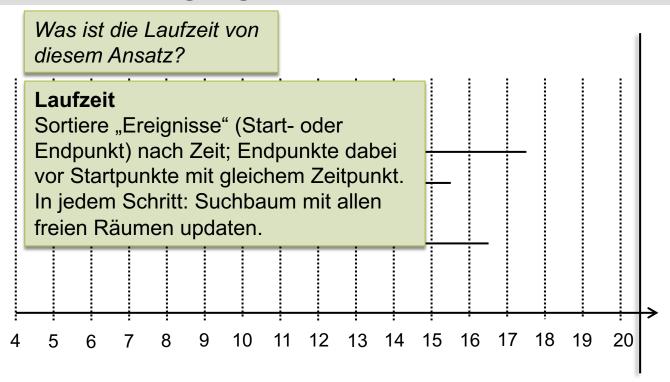


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:

 Gib Raum frei

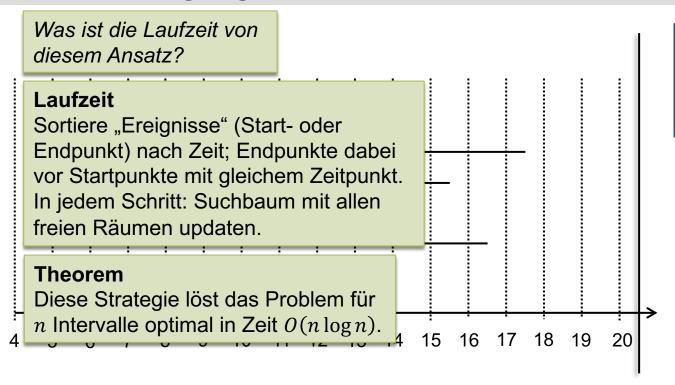




Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:
 Gib Raum frei





Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

- Startpunktes:
 Weise kleinste
 Raumnummer zu
- Endpunktes:

 Gib Raum frei



Beweis

Wie viele Räume braucht man mindestens?



Beweis

Wie viele Räume braucht man mindestens?

Sei



Wie viele Räume braucht man mindestens?

Sei

• $\chi := \max_{i} (|\{I \in \mathcal{I} : i \in I\}|),$

Wie viele Räume braucht man mindestens?

Sei

• $\chi := \max_{i} (|\{I \in \mathcal{I} : i \in I\}|), \leftarrow$

Wie viele Räume braucht man mindestens?

Sei

- $\chi := \max_{i} (|\{I \in \mathcal{I} : i \in I\}|), \longleftarrow$
- opt der Wert einer optimalen Lösung,

Wie viele Räume braucht man mindestens?

Sei

- $\chi := \max_{i} (|\{I \in \mathcal{I} : i \in I\}|), \leftarrow$
- opt der Wert einer optimalen Lösung,
- alg der Wert der Lösung unserer Strategie

Wie viele Räume braucht man mindestens?

Sei

- $\chi := \max_{i} (|\{I \in \mathcal{I} : i \in I\}|), \leftarrow$
- opt der Wert einer optimalen Lösung,
- alg der Wert der Lösung unserer Strategie

Beobachtung: $opt \ge \chi$



Wie viele Räume braucht man mindestens?

Sei

- $\chi := \max_{i} (|\{I \in \mathcal{I} : i \in I\}|), \leftarrow$
- opt der Wert einer optimalen Lösung,
- alg der Wert der Lösung unserer Strategie

Beobachtung: $opt \ge \chi$

Wir zeigen: $alg \le \chi$



Wie viele Räume braucht man mindestens?

Sei

- $\chi := \max_{i} (|\{I \in \mathcal{I} : i \in I\}|), \leftarrow$
- opt der Wert einer optimalen Lösung,
- alg der Wert der Lösung unserer Strategie

Maximale Anzahl an Intervallen, die sich gleichzeitig überlappen.

Beobachtung: $opt \ge \chi$

$$\Rightarrow alg = \mathcal{X}$$

Wir zeigen: $alg \le \chi$





Angenommen, wir brauchen k Räume, d.h. alg = k.

Angenommen, wir brauchen k Räume, d.h. alg = k.

Betrachte den ersten Zeitpunkt, zu dem wir Raum k benutzen.

Angenommen, wir brauchen k Räume, d.h. alg = k.

Betrachte den ersten Zeitpunkt, zu dem wir Raum k benutzen.

Das passiert nur beim Startpunkt eines neuen Intervalls *I*.

Angenommen, wir brauchen k Räume, d.h. alg = k.

Betrachte den ersten Zeitpunkt, zu dem wir Raum k benutzen.

Das passiert nur beim Startpunkt eines neuen Intervalls *I*.

Was muss noch gelten?



Angenommen, wir brauchen k Räume, d.h. alg = k.

Betrachte den ersten Zeitpunkt, zu dem wir Raum k benutzen.

Das passiert nur beim Startpunkt eines neuen Intervalls I.

Was muss noch gelten?

• Alle anderen Räume 1, ..., k-1 sind in Gebrauch!



Angenommen, wir brauchen k Räume, d.h. alg = k.

Betrachte den ersten Zeitpunkt, zu dem wir Raum k benutzen.

Das passiert nur beim Startpunkt eines neuen Intervalls I.

Was muss noch gelten?

- Alle anderen Räume 1, ..., k-1 sind in Gebrauch!
- D.h. I überlappt k-1 andere Intervalle an seinem Startpunkt.



Angenommen, wir brauchen k Räume, d.h. alg = k.

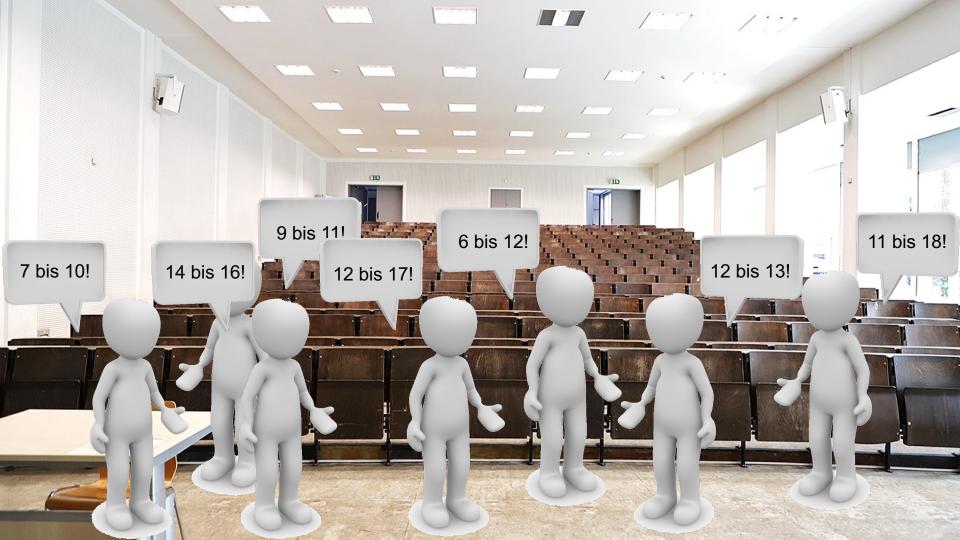
Betrachte den ersten Zeitpunkt, zu dem wir Raum k benutzen.

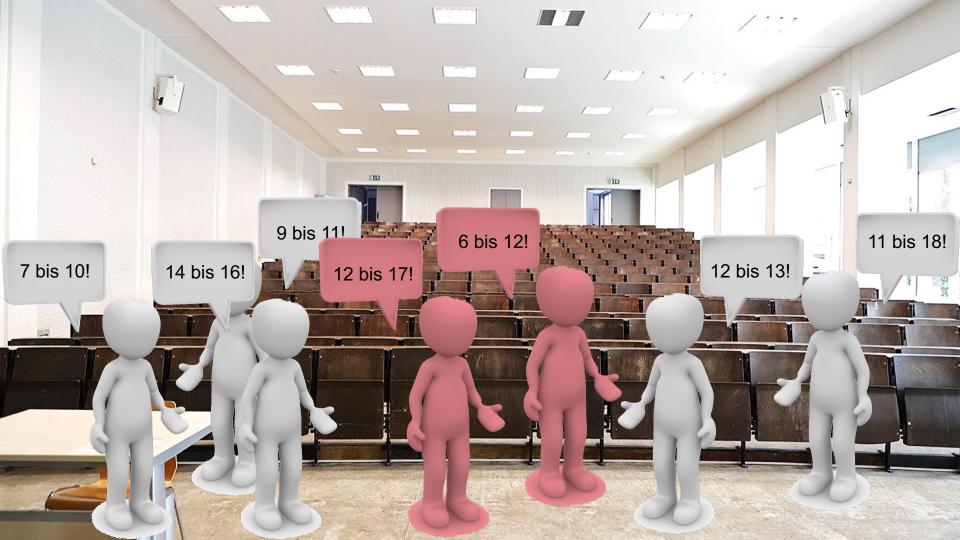
Das passiert nur beim Startpunkt eines neuen Intervalls I.

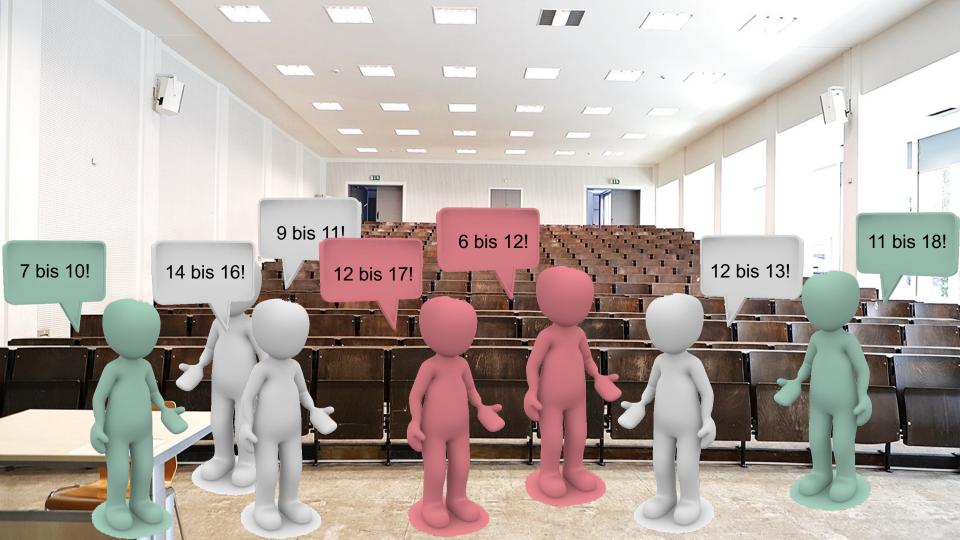
Was muss noch gelten?

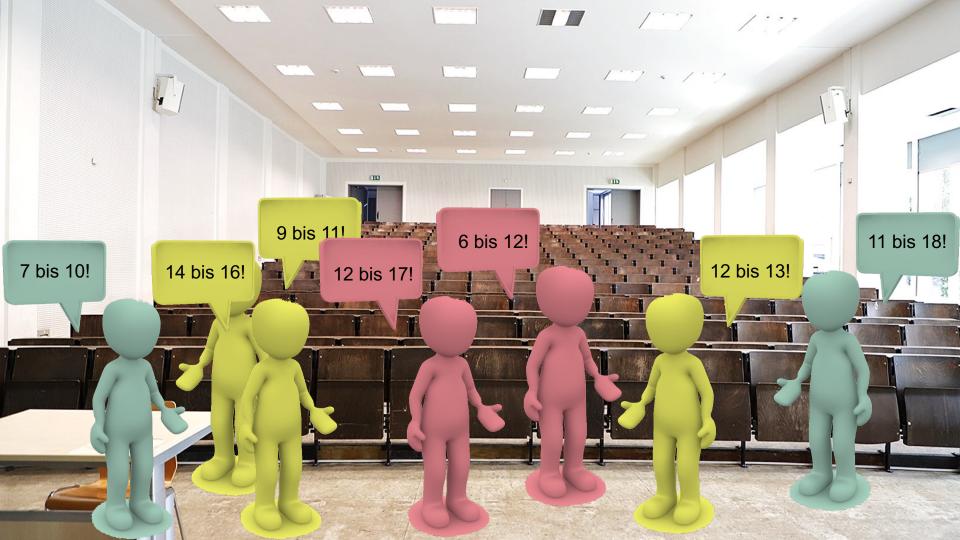
- Alle anderen Räume 1, ..., k-1 sind in Gebrauch!
- D.h. I überlappt k-1 andere Intervalle an seinem Startpunkt.
- Endpunkte vor Startpunkten zum gleichen Zeitpunkt bearbeitet: echte Überlappung!













Hier vorgestellte Probleme sind Spezialfälle von berühmten Graphproblemen:

Independent Set (unabhängige Menge bzw. stabile Menge)



- Independent Set (unabhängige Menge bzw. stabile Menge)
 - Gegeben: ungerichteter Graph G = (V, E)



- Independent Set (unabhängige Menge bzw. stabile Menge)
 - Gegeben: ungerichteter Graph G = (V, E)
 - **Gesucht:** möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S : vw \notin E$ (keine Kanten in S)

- Independent Set (unabhängige Menge bzw. stabile Menge)
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht:** möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S : vw \notin E$ (keine Kanten in S)
- Graphfärbung



- Independent Set (unabhängige Menge bzw. stabile Menge)
 - Gegeben: ungerichteter Graph G = (V, E)
 - **Gesucht:** möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S$: $vw \notin E$ (keine Kanten in S)
- Graphfärbung
 - **Gegeben:** ungerichteter Graph G = (V, E)



- Independent Set (unabhängige Menge bzw. stabile Menge)
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht:** möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S : vw \notin E$ (keine Kanten in S)
- Graphfärbung
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht**: Färbung $c: V \to \{1, ..., k\}$ mit $c(v) = c(w) \Rightarrow vw \notin E$

- Independent Set (unabhängige Menge bzw. stabile Menge)
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht**: möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S : vw \notin E$ (keine Kanten in S)
- Graphfärbung
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht:** Färbung $c: V \to \{1, ..., k\}$ mit $c(v) = c(w) \Rightarrow vw \notin E$
 - D.h. keine einfarbigen Kanten



- Independent Set (unabhängige Menge bzw. stabile Menge)
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht**: möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S : vw \notin E$ (keine Kanten in S)
- Graphfärbung
 - Gegeben: ungerichteter Graph G = (V, E)
 - **Gesucht:** Färbung $c: V \to \{1, ..., k\}$ mit $c(v) = c(w) \Rightarrow vw \notin E$
 - D.h. keine einfarbigen Kanten
 - Möglichst wenige Farben (möglichst kleines k)



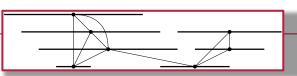
- Independent Set (unabhängige Menge bzw. stabile Menge)
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht:** möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S$: $vw \notin E$ (keine Kanten in S)
- Graphfärbung
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht:** Färbung $c: V \to \{1, ..., k\}$ mit $c(v) = c(w) \Rightarrow vw \notin E$
 - D.h. keine einfarbigen Kanten
 - Möglichst wenige Farben (möglichst kleines k)
 - Äquivalent: Möglichst wenige Independent Sets S_i , $1 \le i \le k$ mit $\bigcup_{i=1}^k S_i = V$.

- Independent Set (unabhängige Menge bzw. stabile Menge)
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht**: möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S : vw \notin E$ (keine Kanten in S)
- Graphfärbung
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht:** Färbung $c: V \to \{1, ..., k\}$ mit $c(v) = c(w) \Rightarrow vw \notin E$
 - D.h. keine einfarbigen Kanten
 - Möglichst wenige Farben (möglichst kleines k)
 - Äquivalent: Möglichst wenige Independent Sets S_i , $1 \le i \le k$ mit $\bigcup_{i=1}^k S_i = V$.
- Hier betrachtet auf so genannten Intervallgraphen



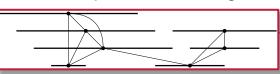
- Independent Set (unabhängige Menge bzw. stabile Menge)
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht:** möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S$: $vw \notin E$ (keine Kanten in S)
- Graphfärbung
 - Gegeben: ungerichteter Graph G = (V, E)
 - **Gesucht:** Färbung $c: V \to \{1, ..., k\}$ mit $c(v) = c(w) \Rightarrow vw \notin E$
 - D.h. keine einfarbigen Kanten
 - Möglichst wenige Farben (möglichst kleines k)
 - Äquivalent: Möglichst wenige Independent Sets S_i , $1 \le i \le k$ mit $\bigcup_{i=1}^k S_i = V$.
- Hier betrachtet auf so genannten Intervallgraphen





- Independent Set (unabhängige Menge bzw. stabile Menge)
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht:** möglichst großes $S \subseteq V$ mit $\forall \{v, w\} \subseteq S : vw \notin E$ (keine Kanten in S)
- Graphfärbung
 - **Gegeben:** ungerichteter Graph G = (V, E)
 - **Gesucht**: Färbung $c: V \to \{1, ..., k\}$ mit $c(v) = c(w) \Rightarrow vw \notin E$
 - D.h. keine einfarbigen Kanten
 - Möglichst wenige Farben (möglichst kleines k)
 - Äquivalent: Möglichst wenige Independent Sets S_i , $1 \le i \le k$ mit $\bigcup_{i=1}^k S_i = V$.
- Hier betrachtet auf so genannten Intervallgraphen
- Auf allgemeinen Graphen: Wohl kein effizienter Algorithmus für optimale Lösungen...



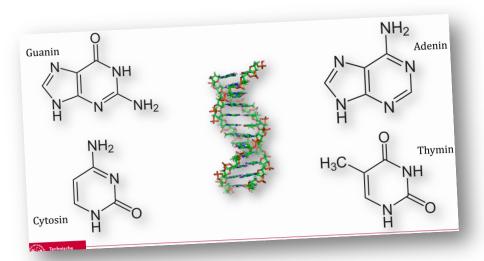




Dynamic Programming!



Dynamic Programming!



Dynamic Programming!

