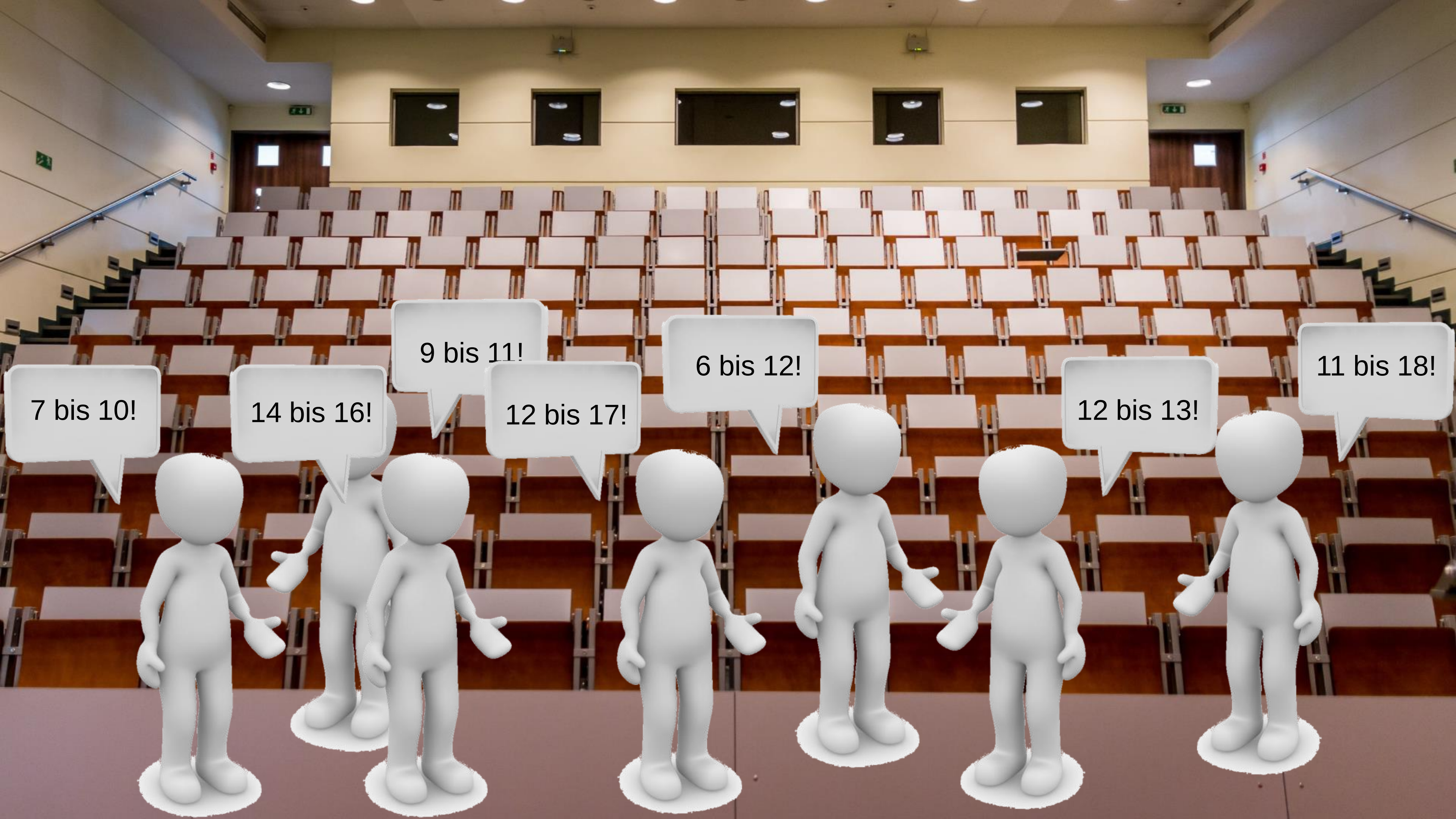
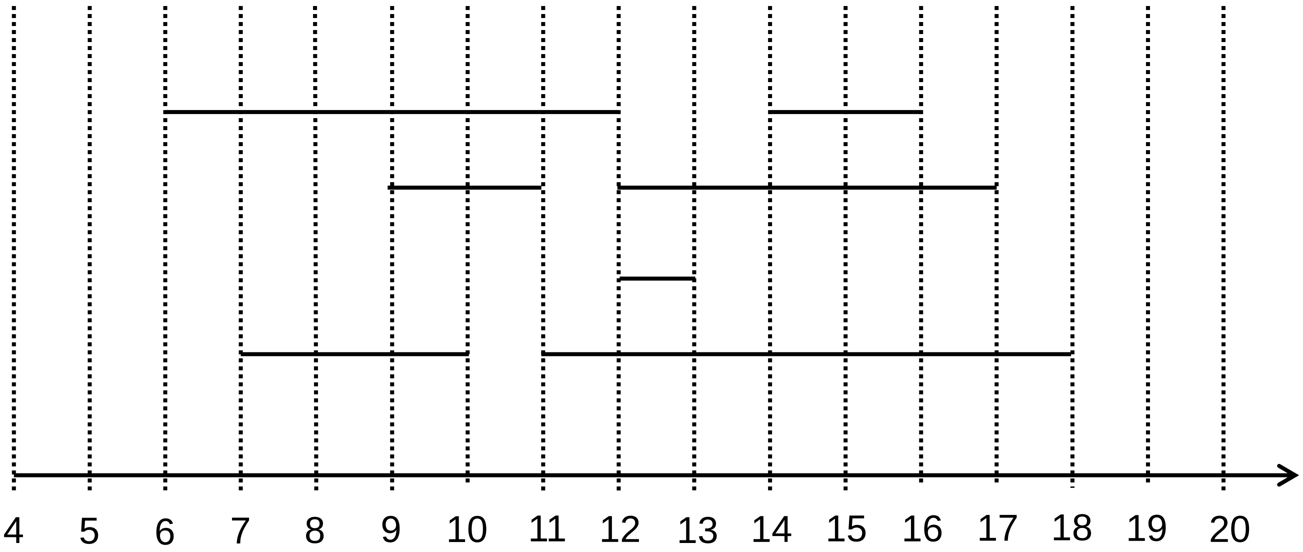
Technische Universität Braunschweig

Algorithmen und Datenstrukturen 2 – Übung #1

Arne Schmidt
06.05.2020


7 bis 10!
14 bis 16!
9 bis 11!
12 bis 17!
6 bis 12!
12 bis 13!
11 bis 18!

Hörsaal-Belegung



Hörsaal-Belegung – Das Problem

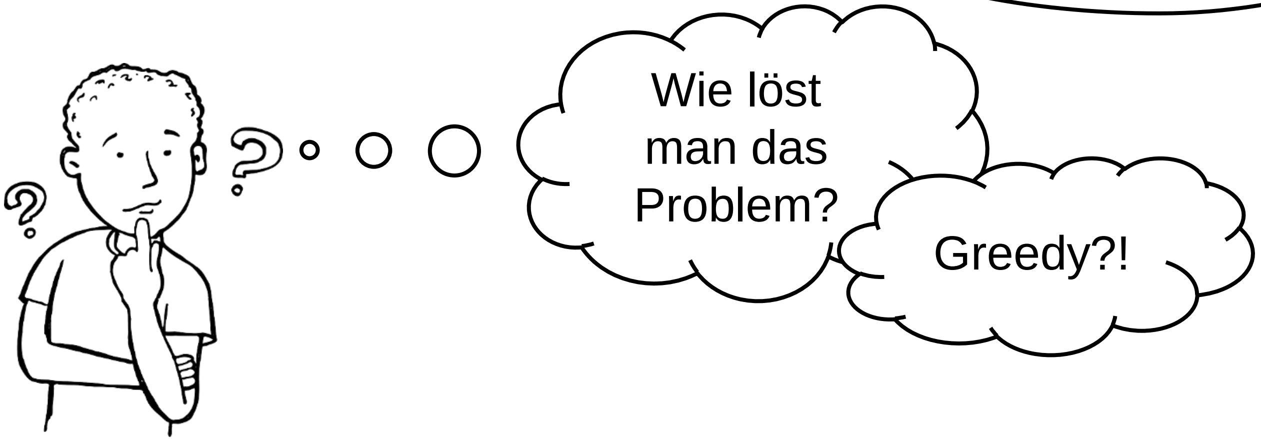
Gegeben: Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

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

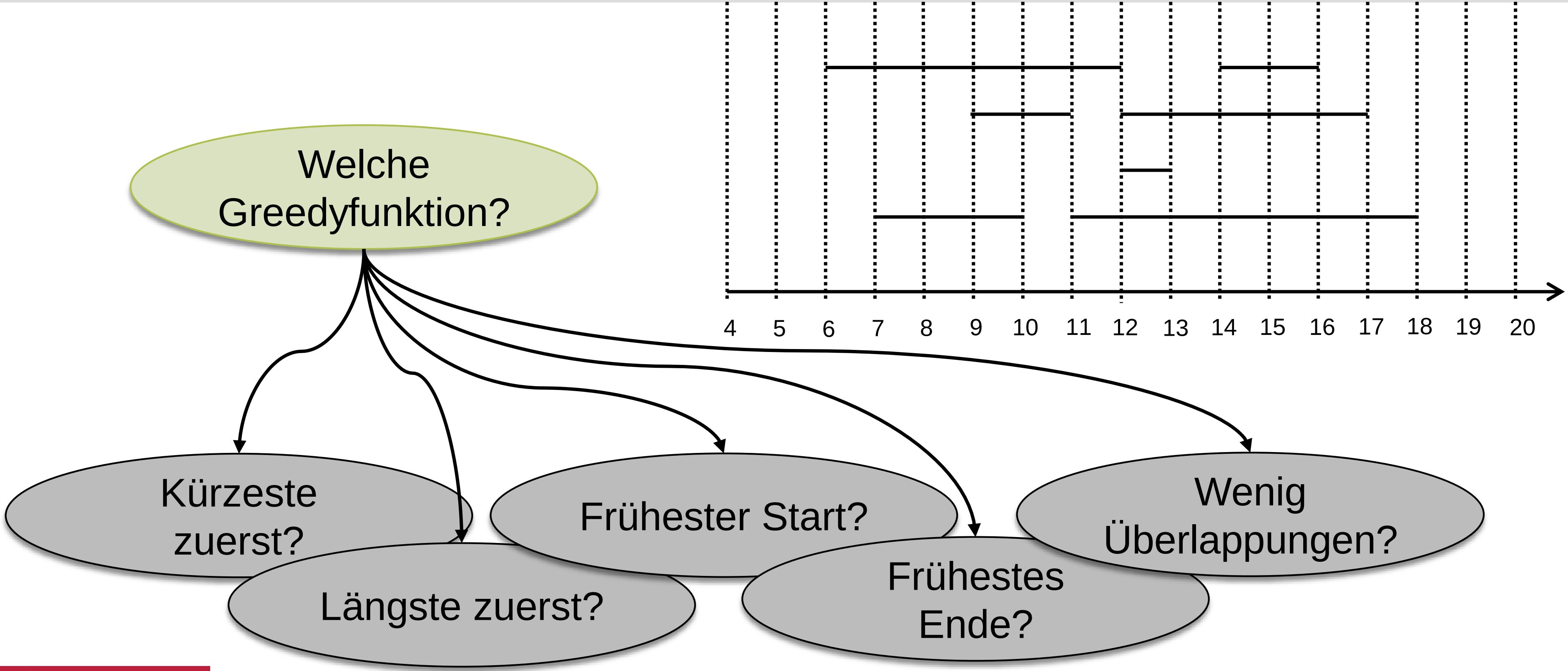
$$\forall I_i, I_j \in \mathcal{I}' : I_i \cap I_j = \emptyset$$

und $\mathcal{I}'$ ist größtmöglich.

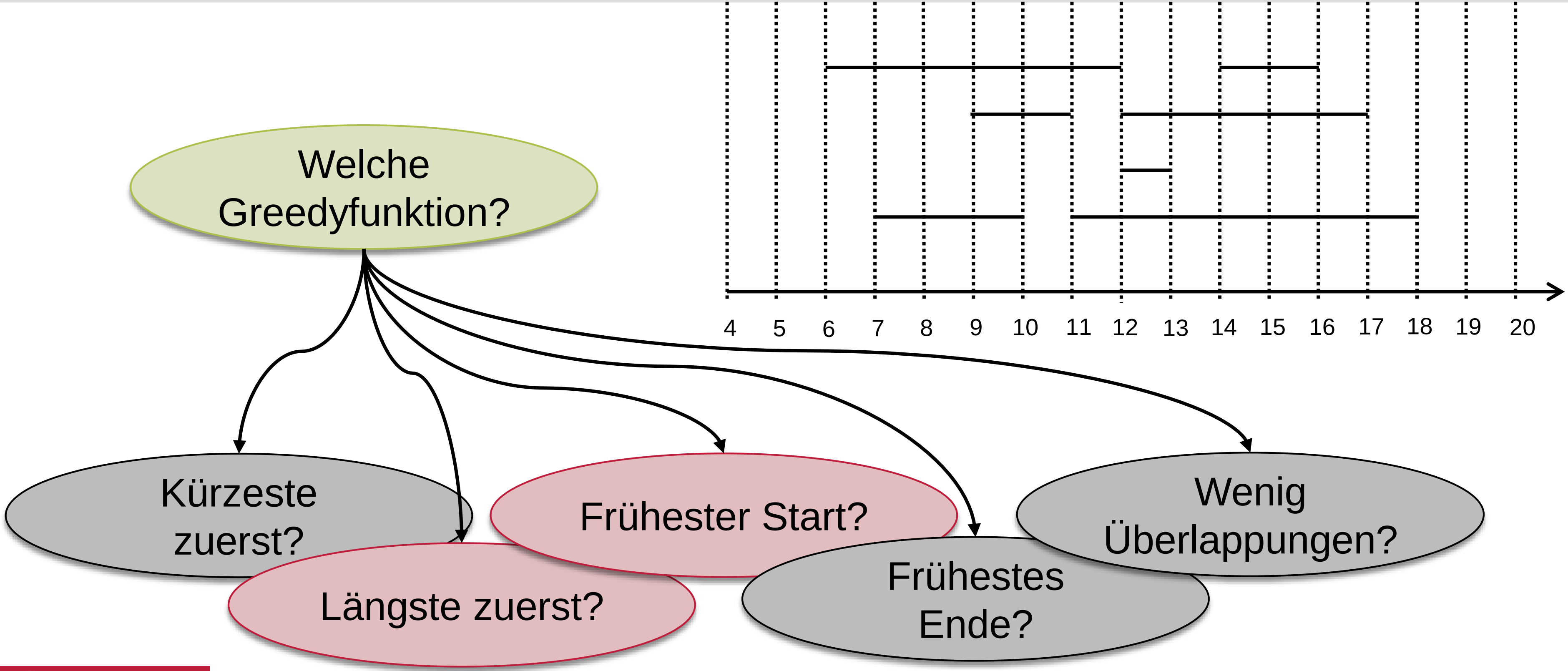
Die ausgewählten Intervalle müssen disjunkt sein.



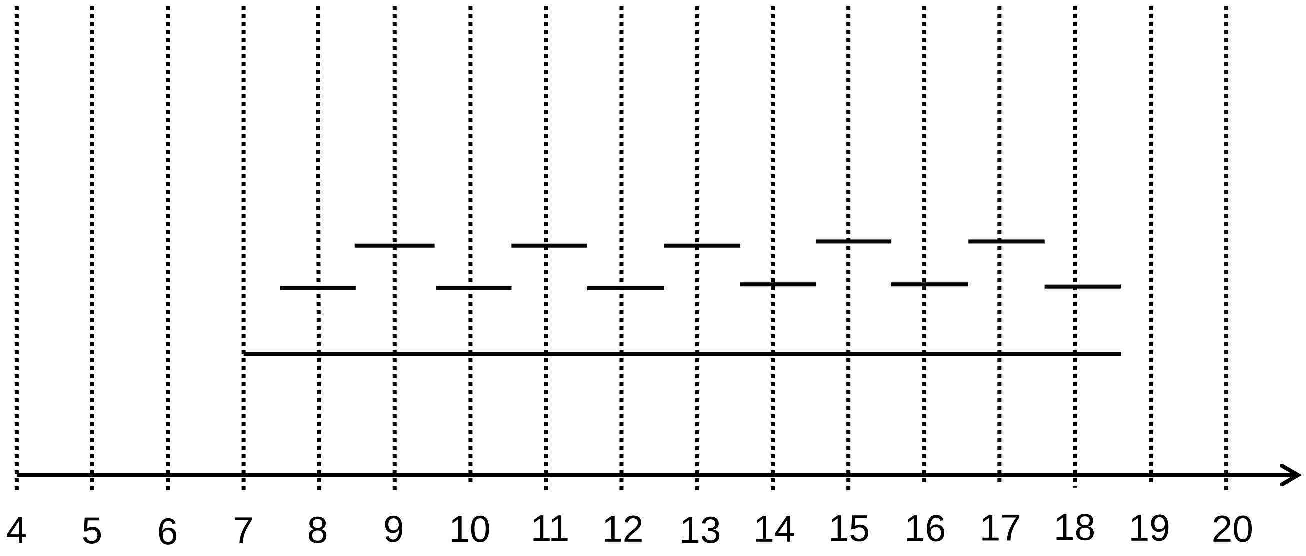
Hörsaal-Belegung – Strategien



Hörsaal-Belegung – Strategien



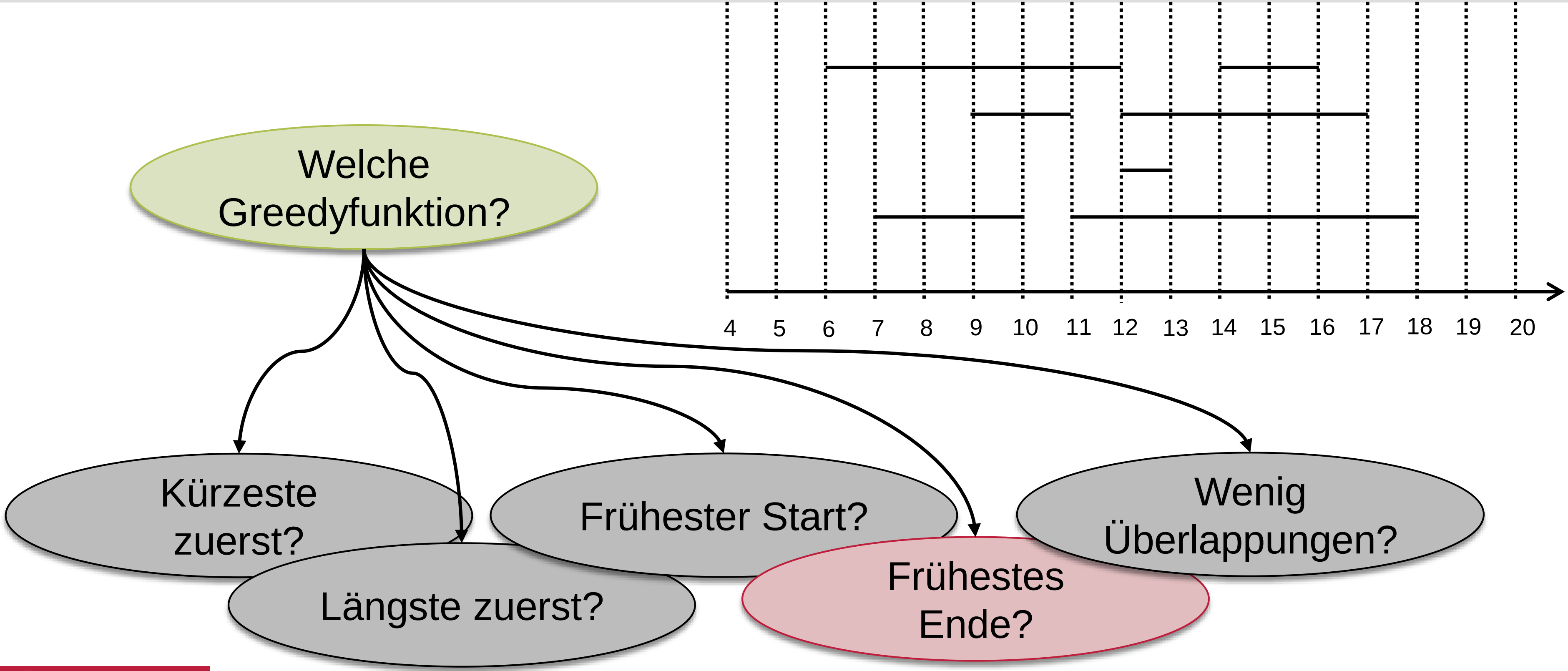
Hörsaal-Belegung – Frühester Start/Längstes Intervall



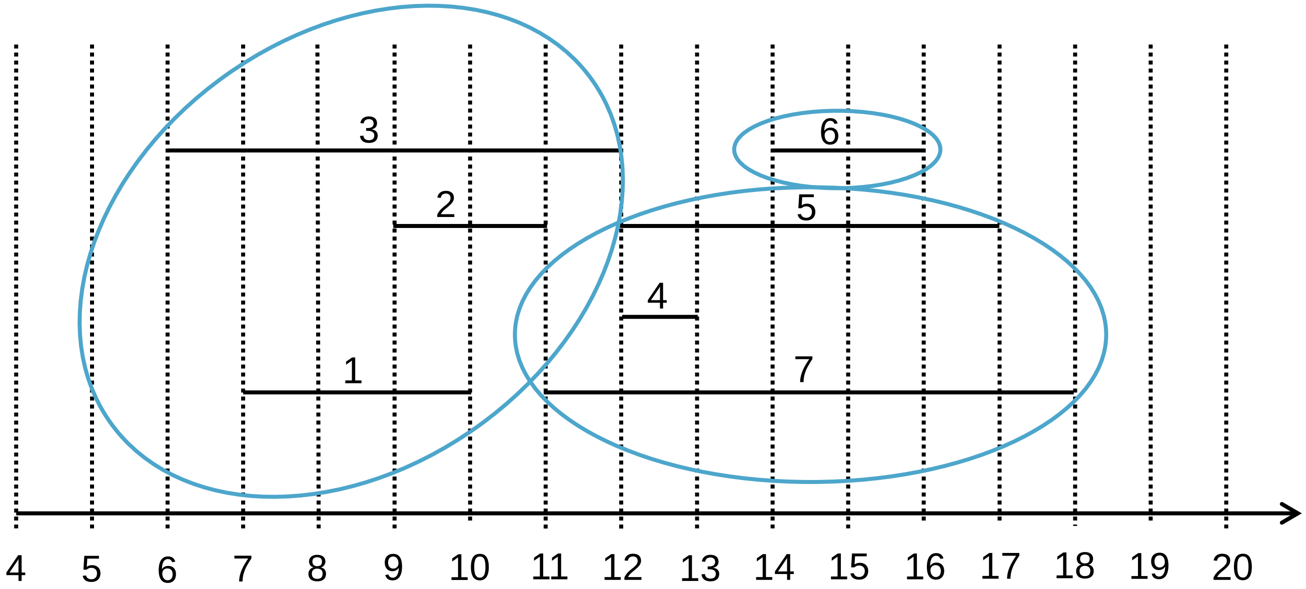
ALG: 1

OPT: 11

Hörsaal-Belegung – Strategien



Hörsaal-Belegung – Frühestes Ende



ALG: 3

OPT: 3?

Hörsaal-Belegung – Frühestes Ende

Algorithmus 1

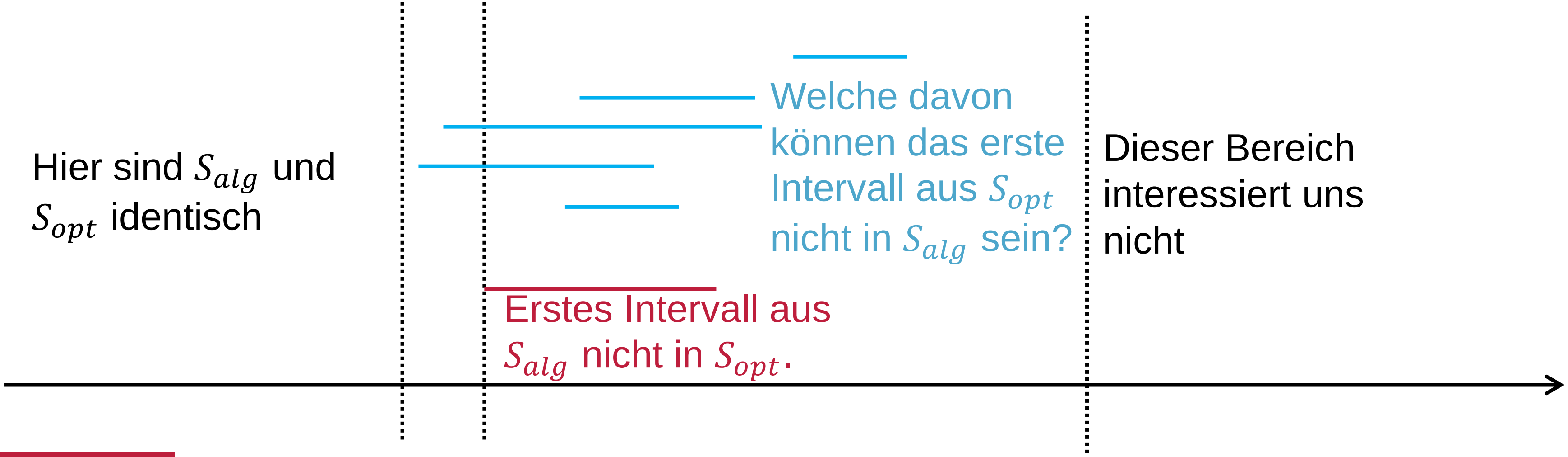
sortiere $\{1, \ldots, n\}$ nach e_i aufsteigend $\rightarrow$ Permutation $\pi(1), \ldots, \pi(n)$
$S := \{\pi(1)\}$
$e := e_{\pi(1)}$
for $k = 2$ **to** n **do**
    if $(s_{\pi(k)} \geq e)$ **then**
        $S := S \cup \{\pi(k)\}$
        $e := e_{\pi(k)}$
return S

Theorem:
Algorithmus 1 löst das Hörsaal-Problem optimal in Zeit $O(n \log n)$

Beweis

Annahme: Algorithmus 1 sei nicht optimal.

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



Beweis

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

Welche davon können das erste Intervall aus S_{opt} nicht in S_{alg} sein?

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

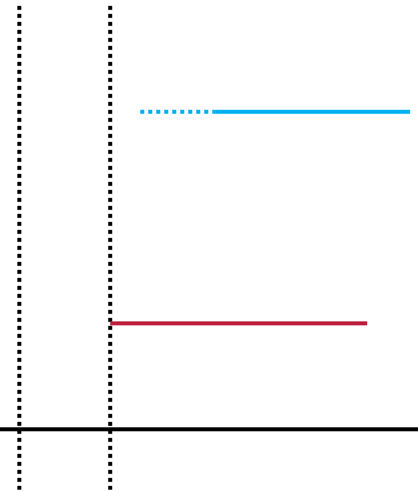
- Blau muss rot überlappen!
- Ende von blau kommt nach Ende von rot!

Beweis

Annahme: Algorithmus 1 sei nicht optimal.

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

Entferne blau aus S_{opt} und füge rot hinzu.



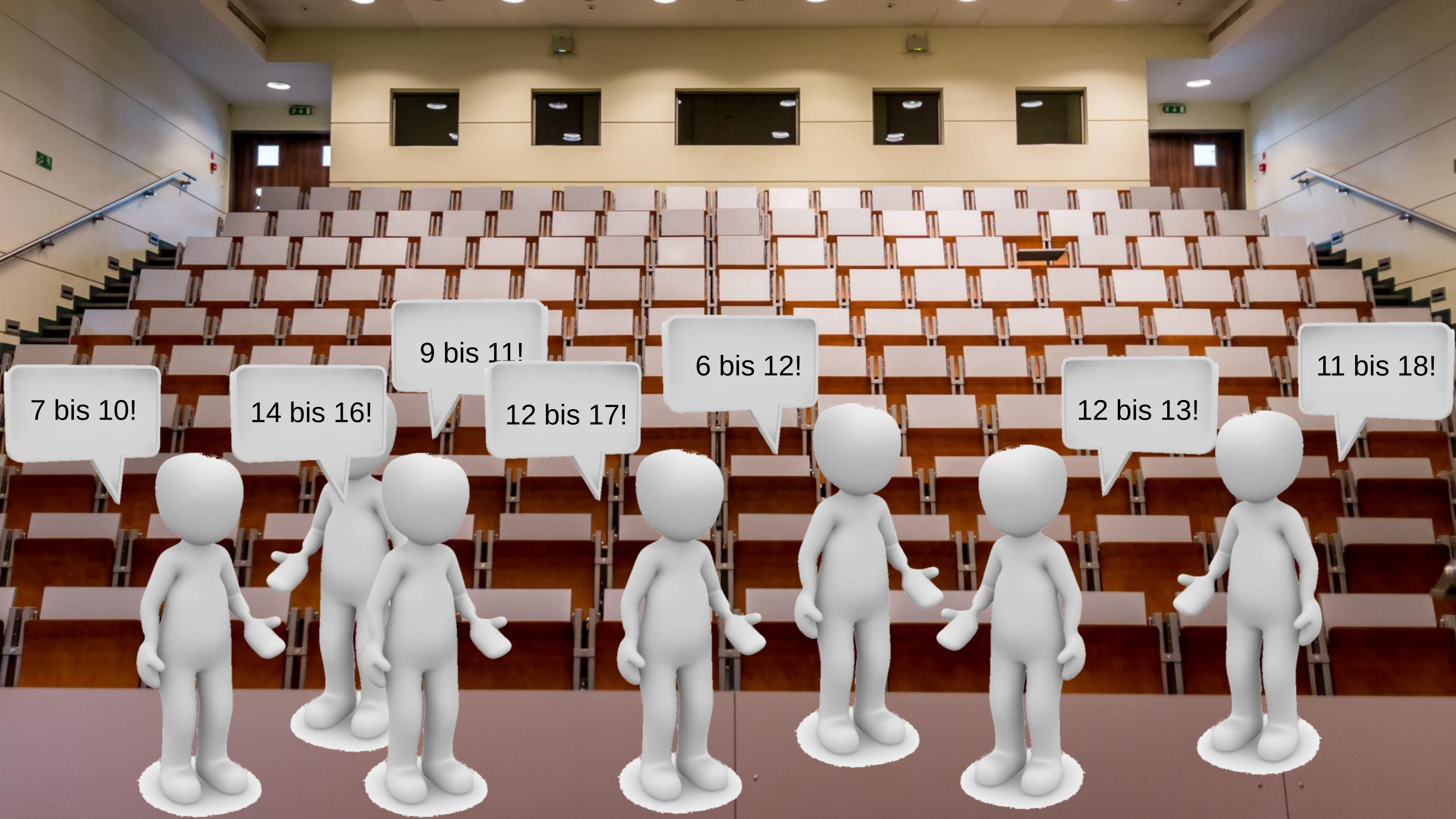
Hier sind S_{alg} und S_{opt} identisch

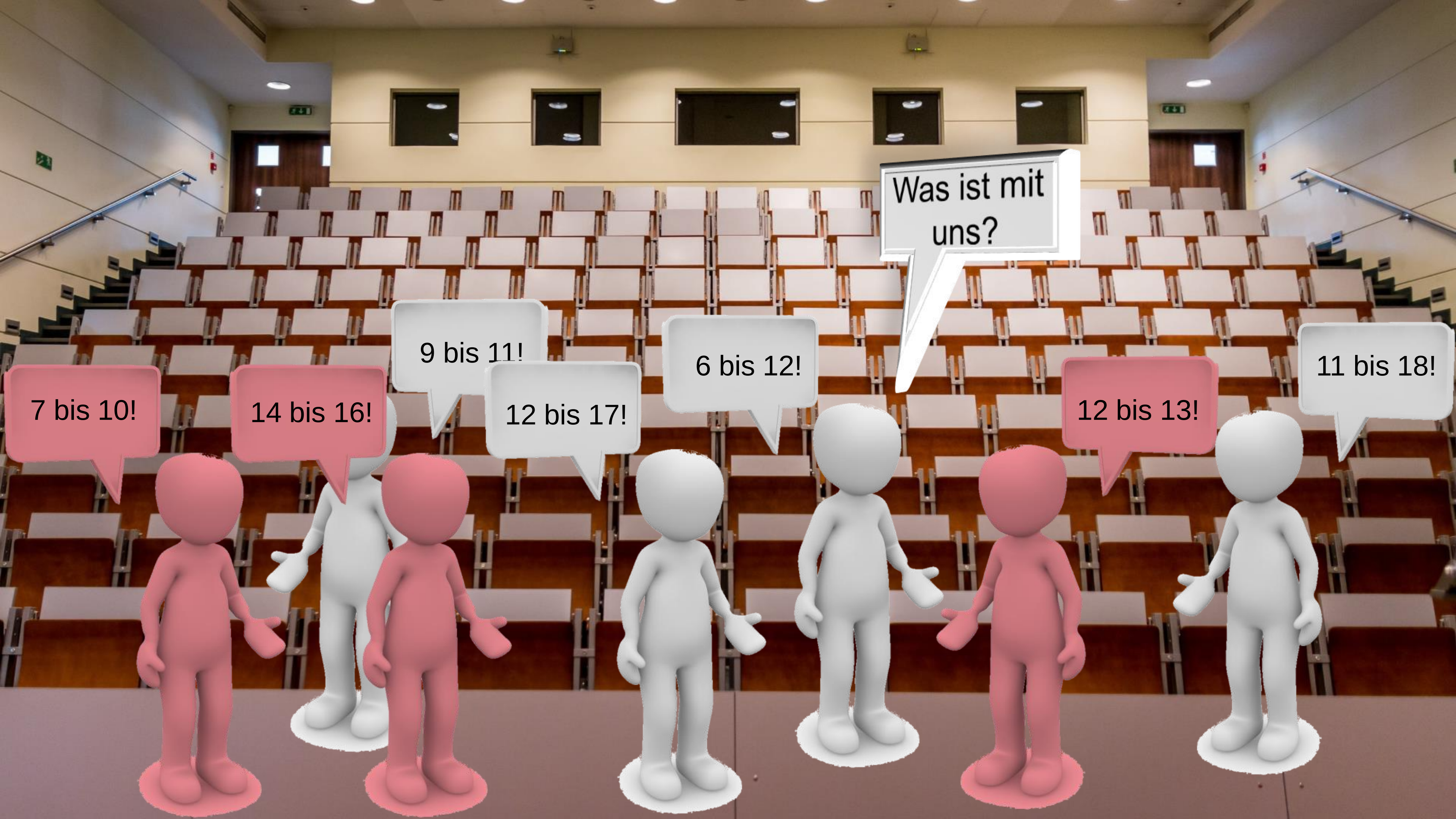
S_{opt} ist immer noch zulässig!

$\Rightarrow S_{alg} \subseteq S_{opt}$

Enthält S_{opt} nun noch mehr Intervalle, dann hätte Algorithmus 1 diese aufgenommen!

$\Rightarrow S_{alg} = S_{opt}$

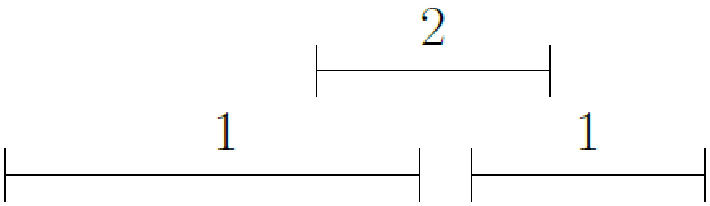

7 bis 10!
14 bis 16!
9 bis 11!
12 bis 17!
6 bis 12!
12 bis 13!
11 bis 18!

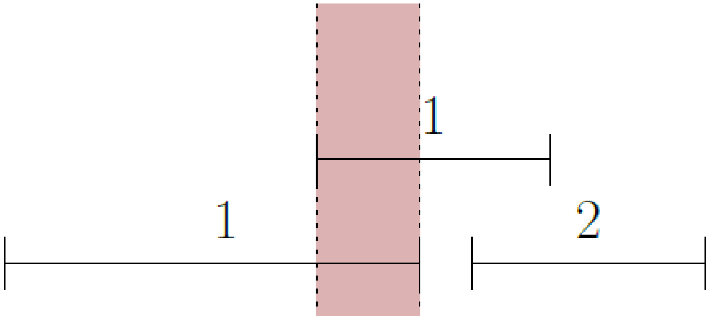

Was ist mit uns?
7 bis 10!
14 bis 16!
9 bis 11!
12 bis 17!
6 bis 12!
12 bis 13!
11 bis 18!

Hörsaal-Belegung – Eine Variante

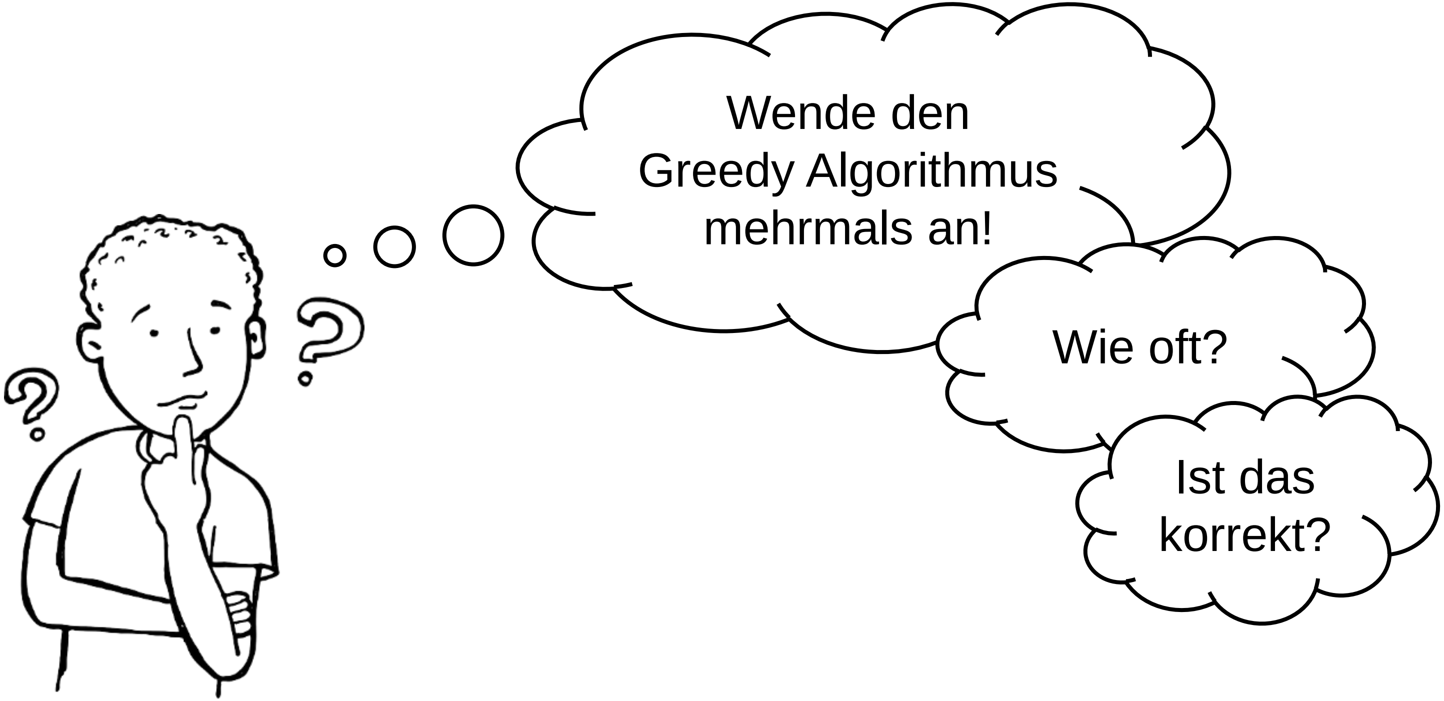
Gegeben: Intervalle $\mathcal{I} := \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

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

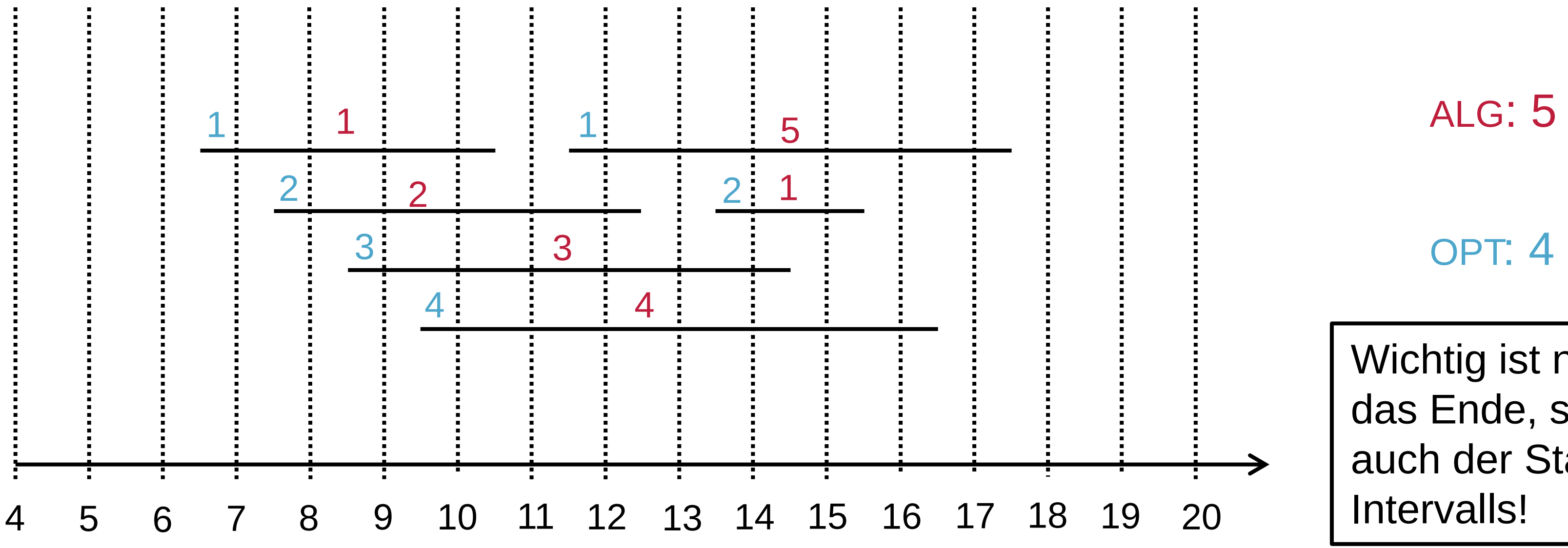




Hörsaal-Belegung – Eine Variante

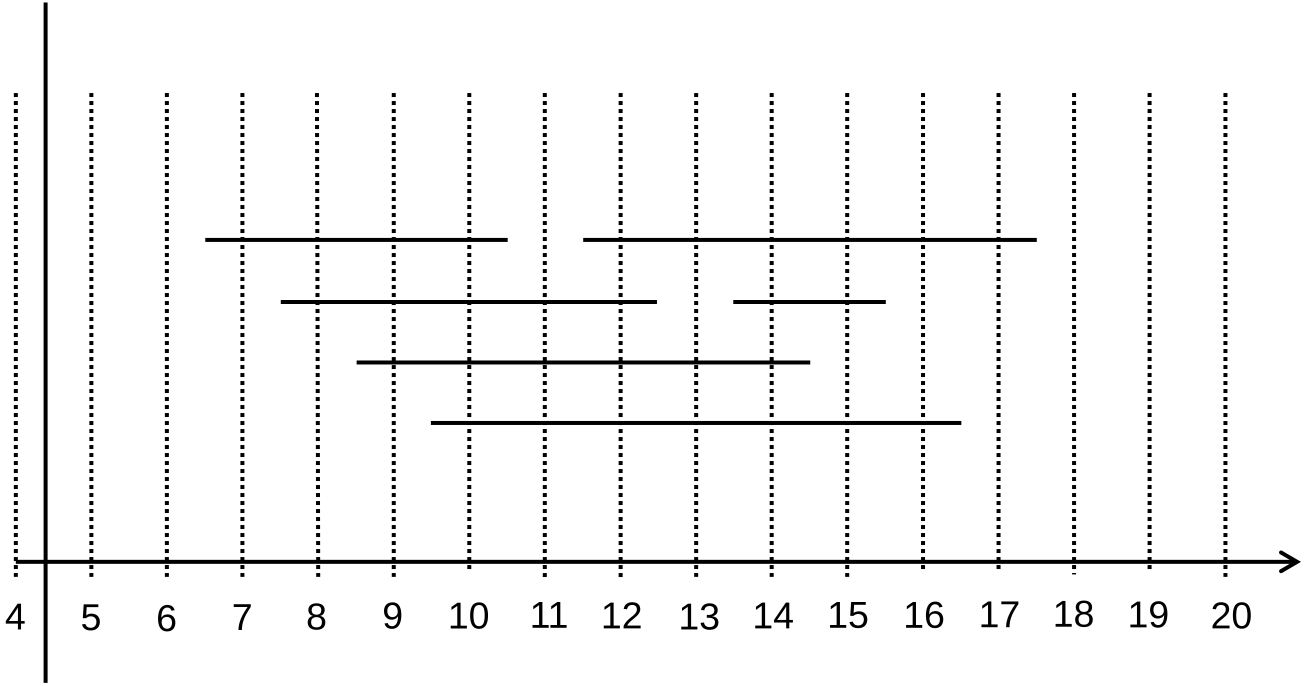


Hörsaal-Belegung – Eine Variante



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

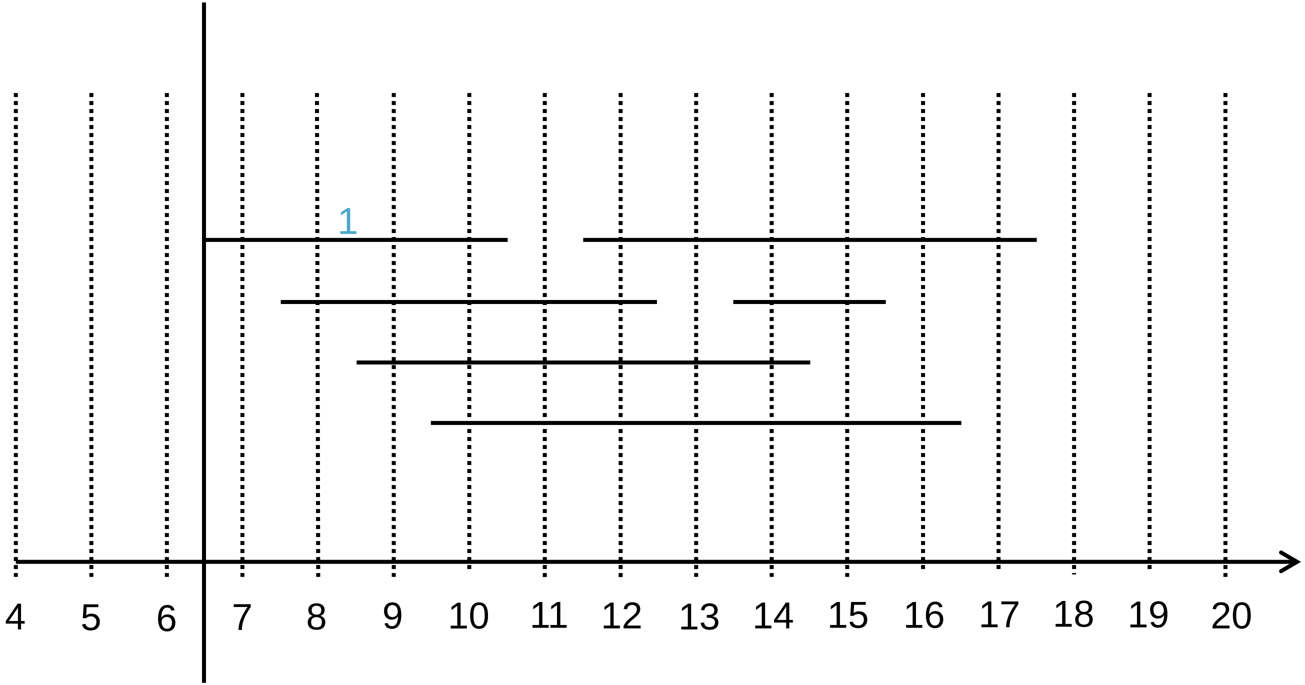
Hörsaal-Belegung – Eine Variante



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

Bewege eine Linie von links nach rechts

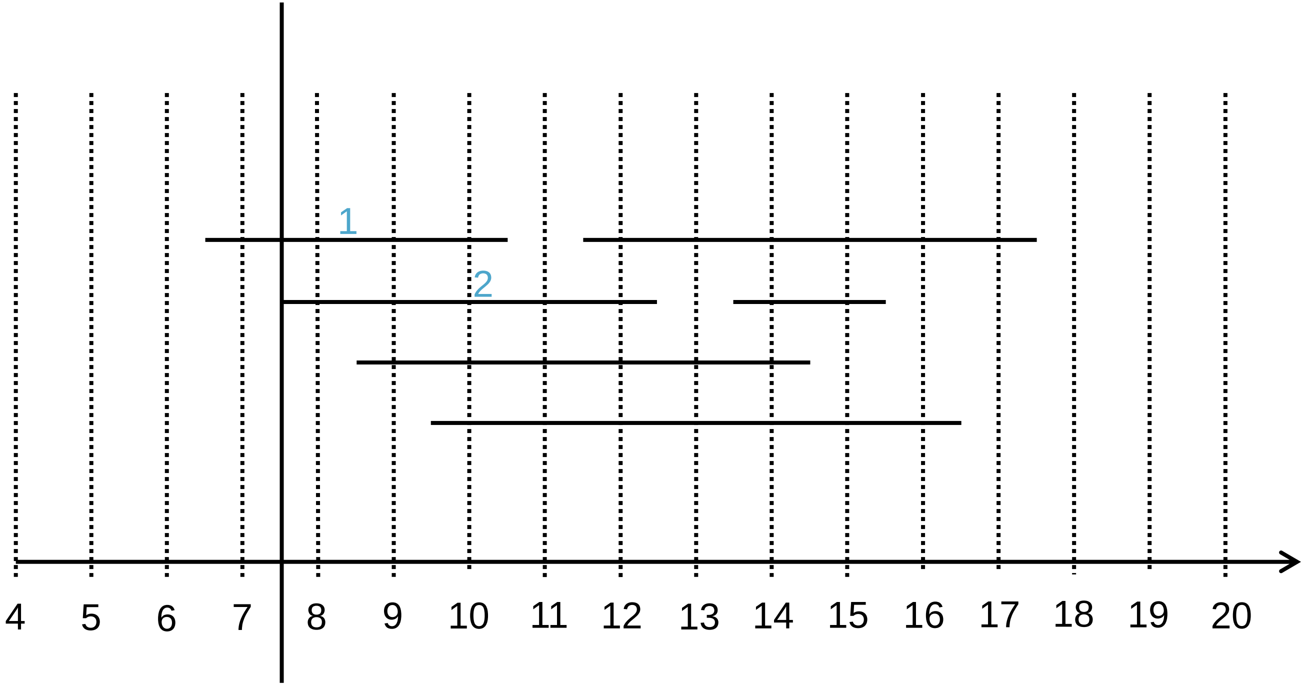
Hörsaal-Belegung – Eine Variante



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

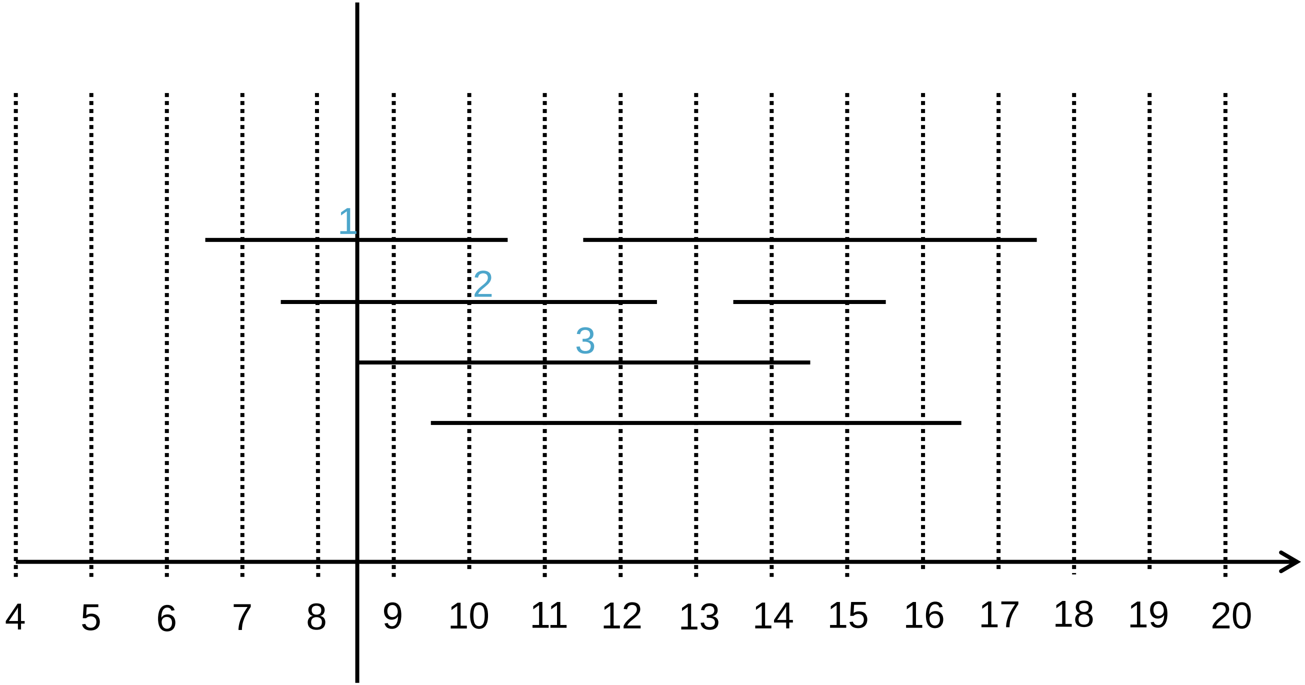
Hörsaal-Belegung – Eine Variante



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

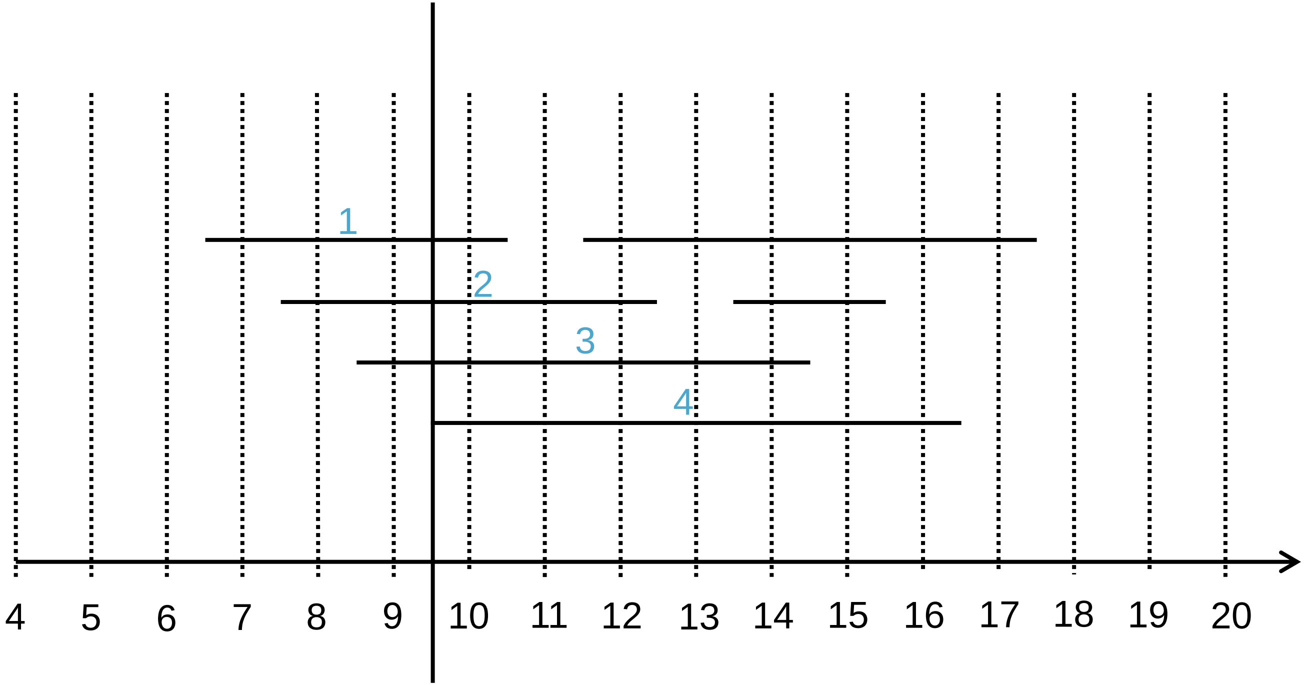
Hörsaal-Belegung – Eine Variante



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

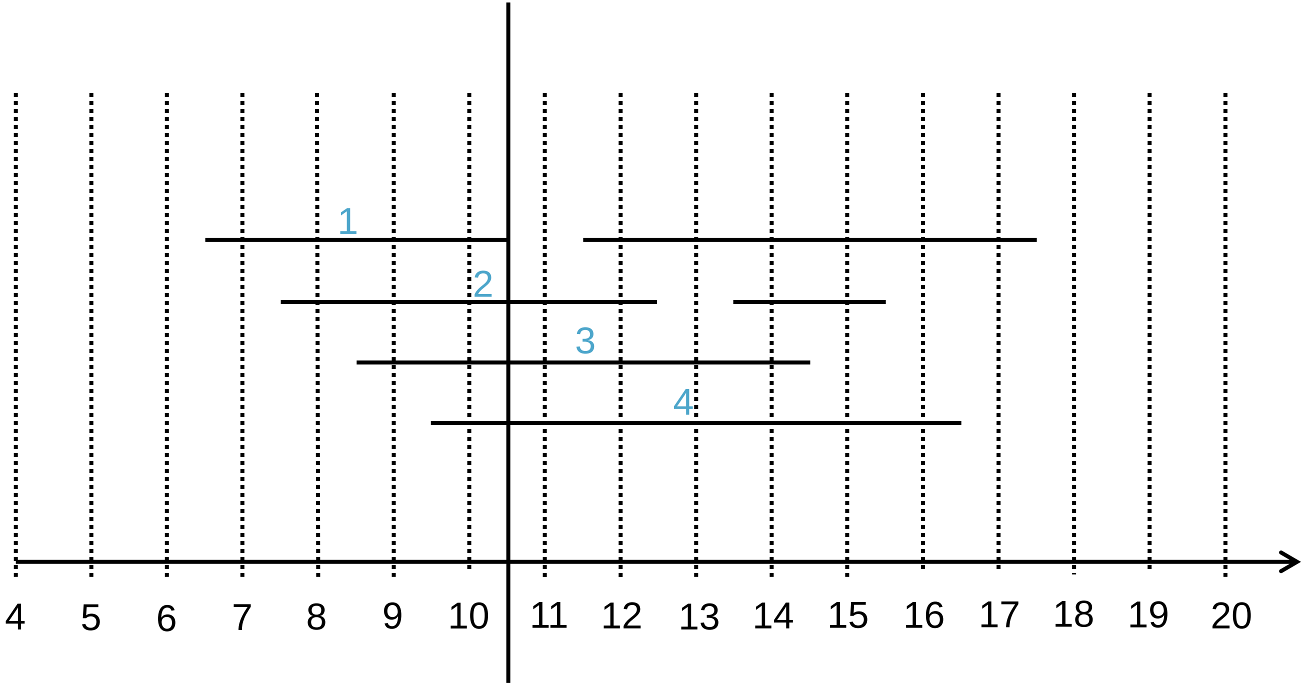
Hörsaal-Belegung – Eine Variante



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

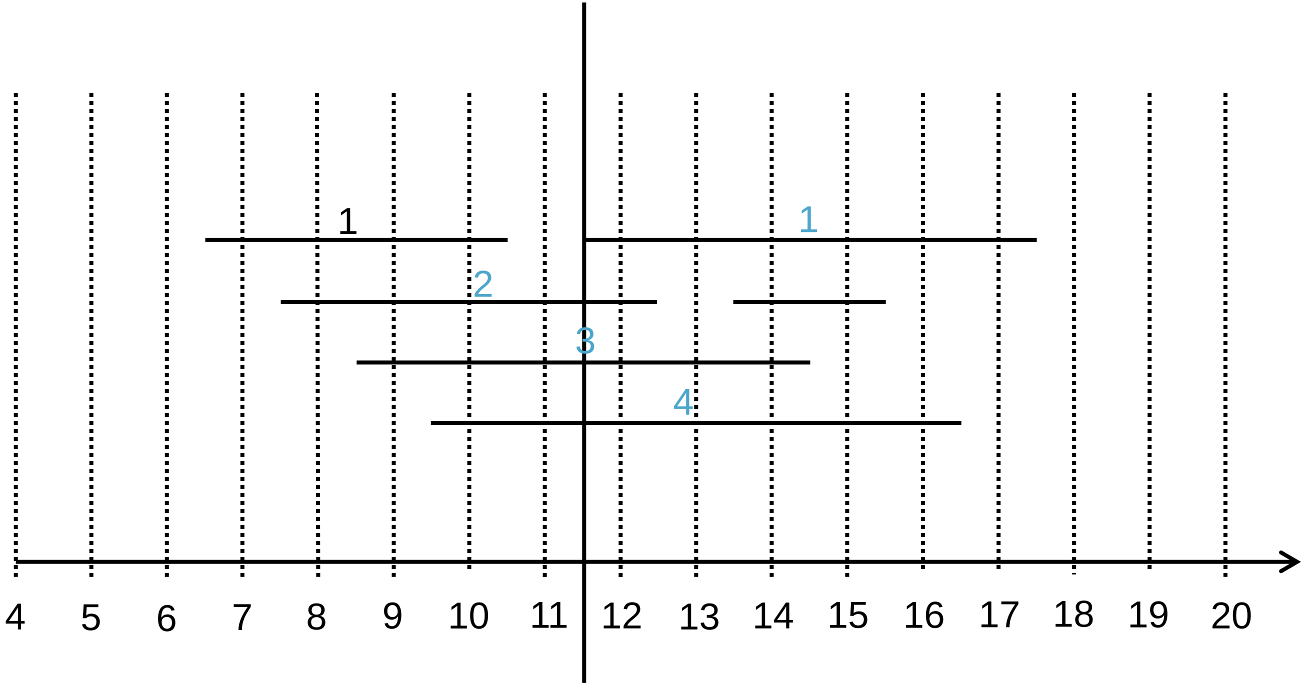
Hörsaal-Belegung – Eine Variante



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

Hörsaal-Belegung – Eine Variante

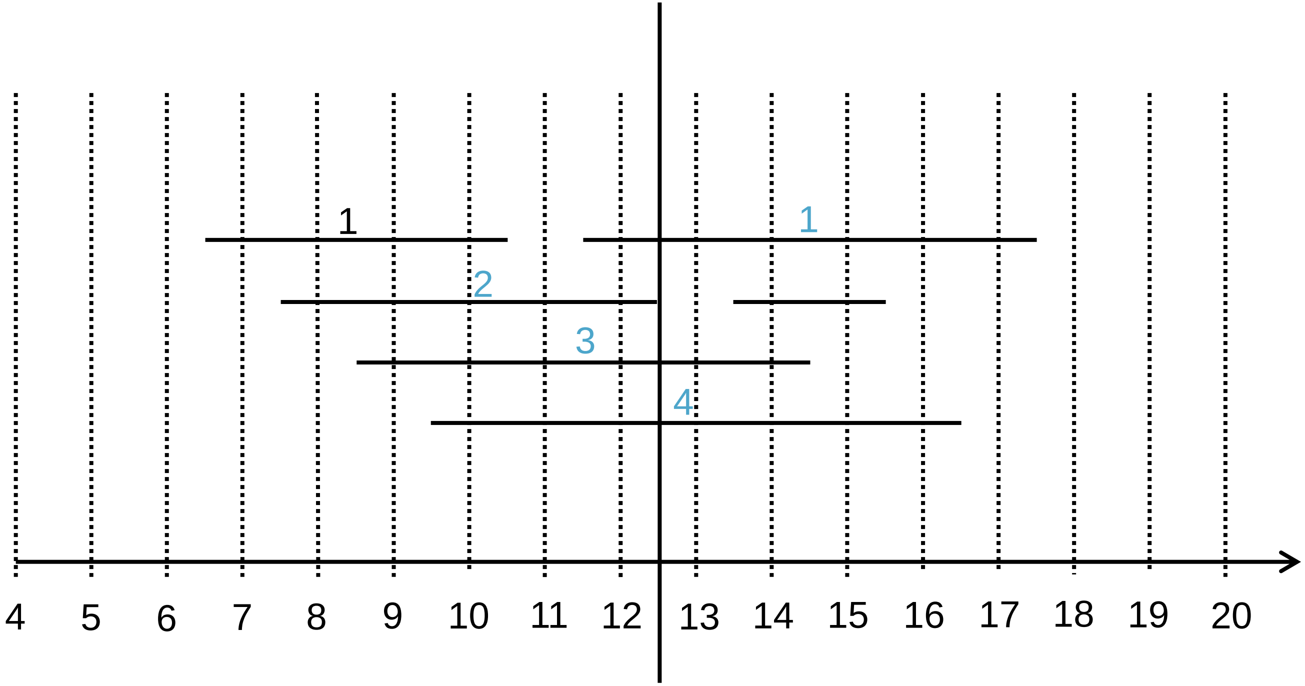


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

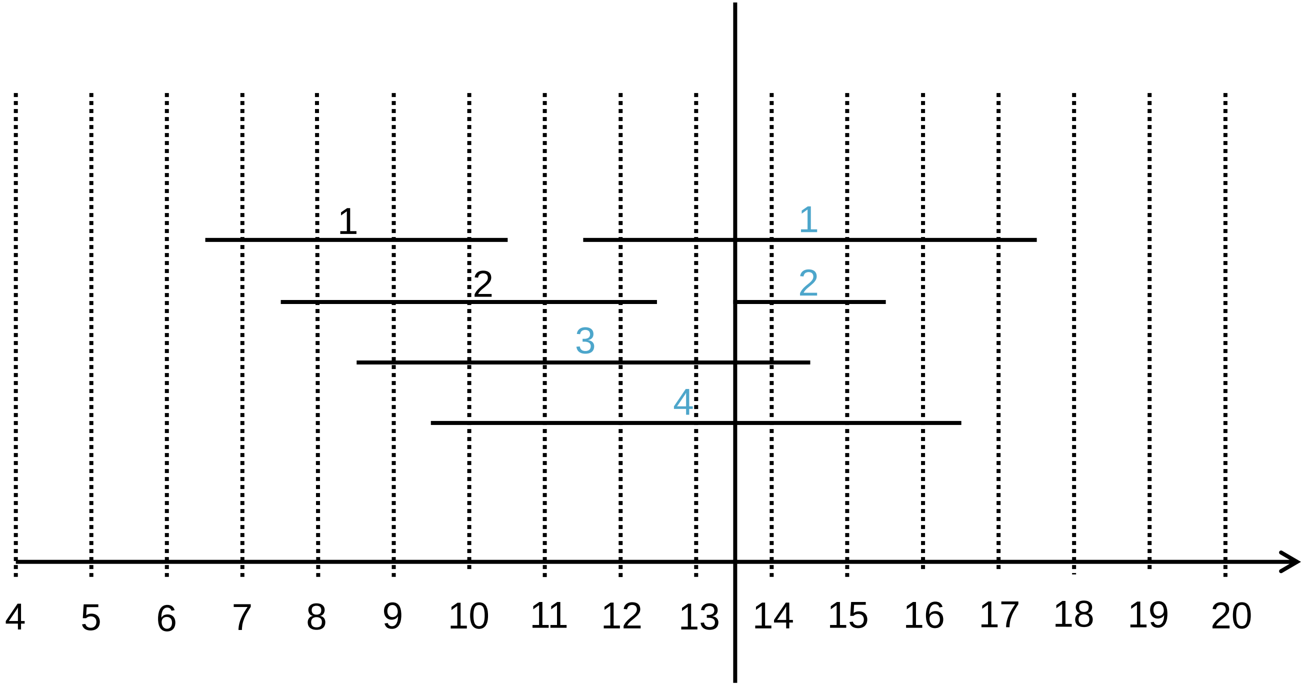
Hörsaal-Belegung – Eine Variante



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

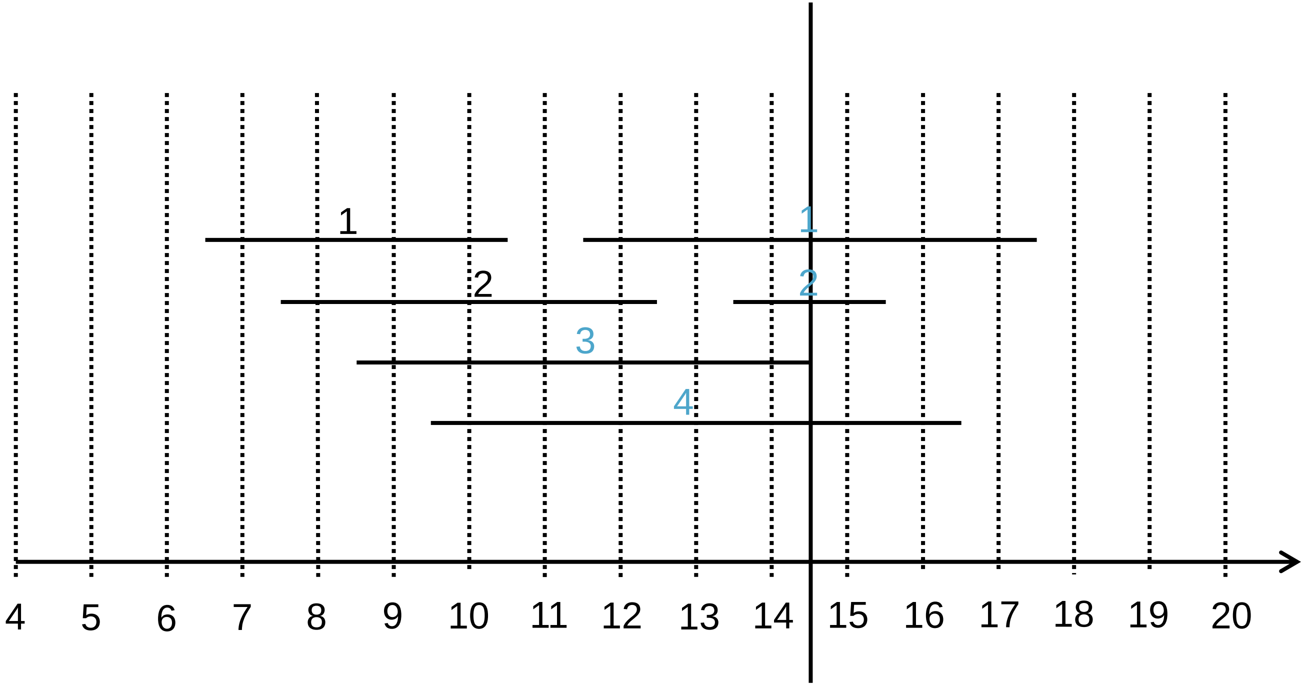
Hörsaal-Belegung – Eine Variante



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

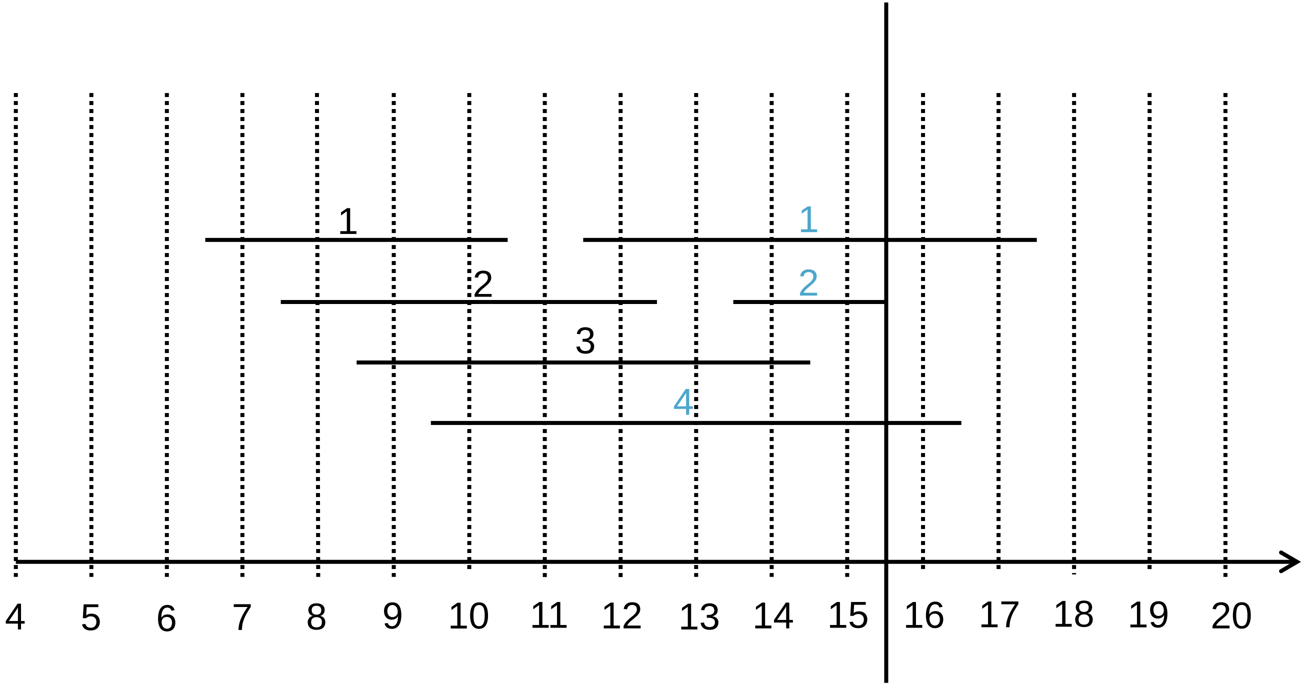
Hörsaal-Belegung – Eine Variante



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

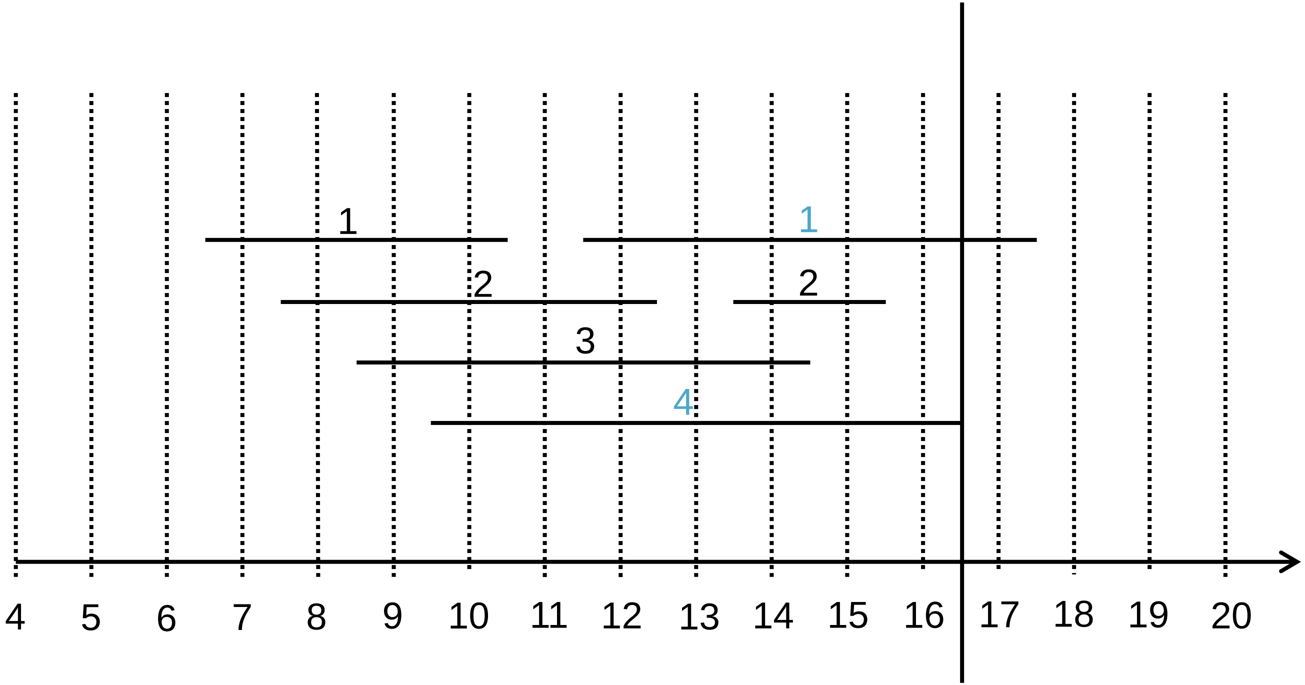
Hörsaal-Belegung – Eine Variante



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

Hörsaal-Belegung – Eine Variante

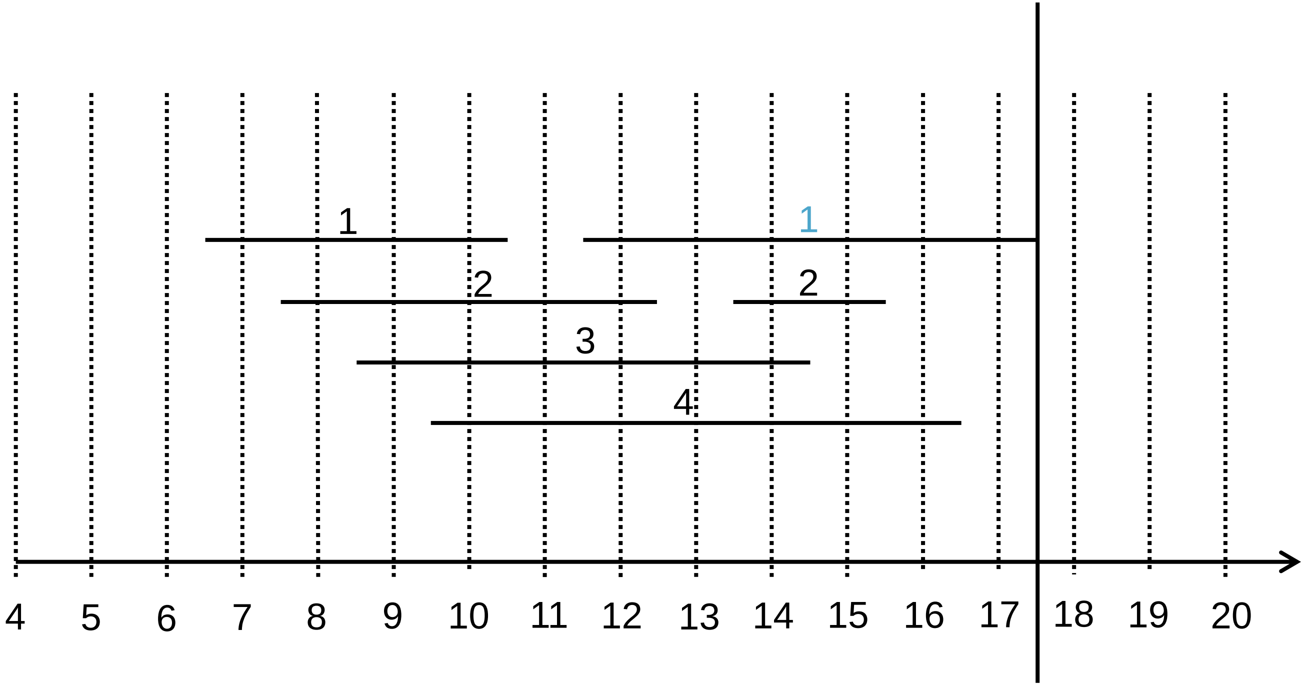


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

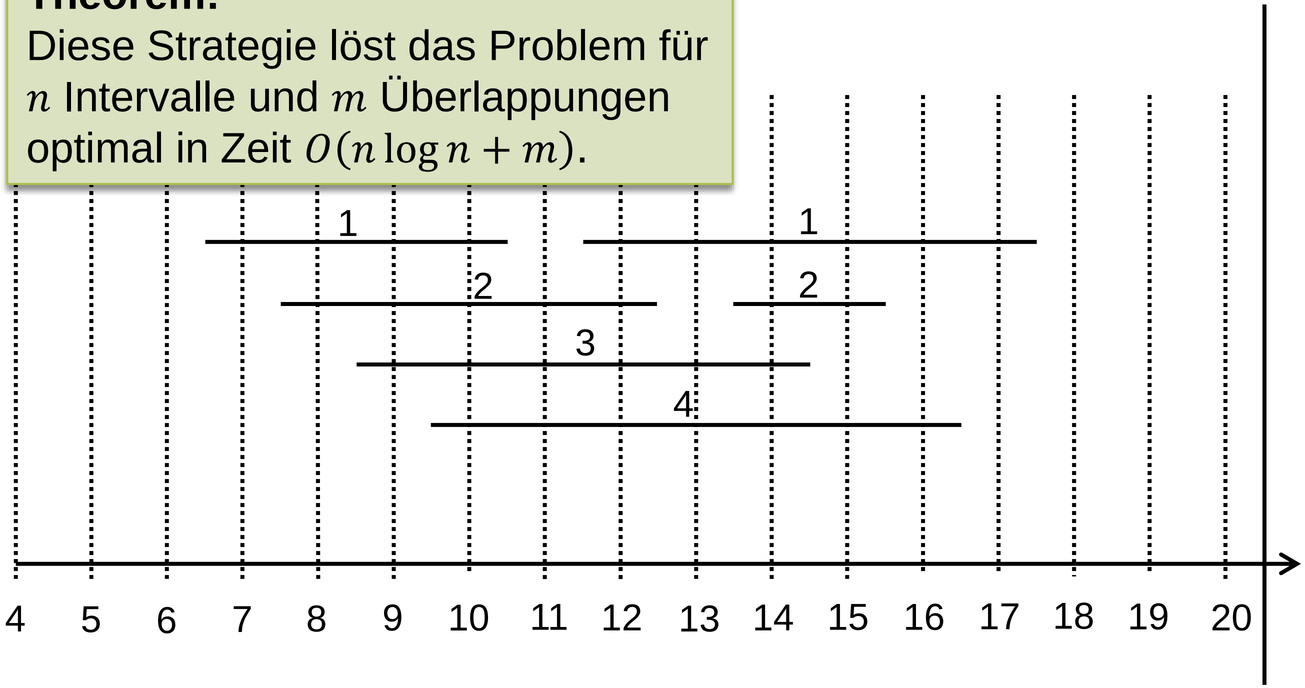


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

Hörsaal-Belegung – Eine Variante

Theorem:
Diese Strategie löst das Problem für n Intervalle und m Überlappungen optimal in Zeit $O(n \log n + m)$.



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

Bei Erreichen eines
- Startpunktes:
  Weise kleinste Raumnummer zu
- Endpunktes:
  Gib Raum frei

Beweis

Wie viele Räume braucht man mindestens?

Maximale Anzahl an Intervallen, die sich gleichzeitig überlappen.

Sei

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

Beobachtung: $opt \geq \chi$

Wir zeigen: $alg \leq \chi$

Beweis

Beweis per Induktion!

Für das erste Intervall wird ein Raum benötigt, und $\chi \geq 1$. Das passt!

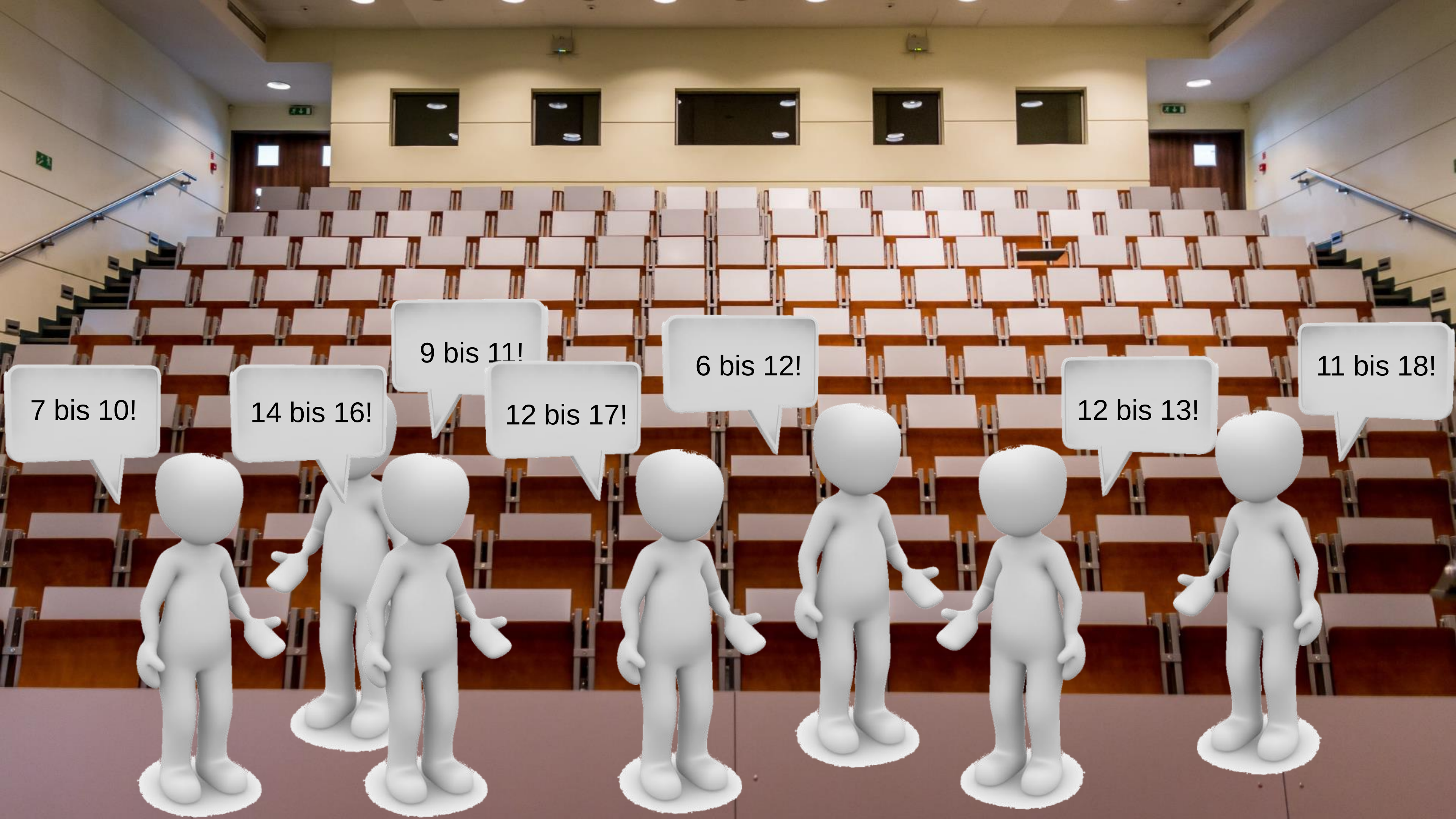
Annahme: $alg \leq \chi$ für die ersten i Intervalle.

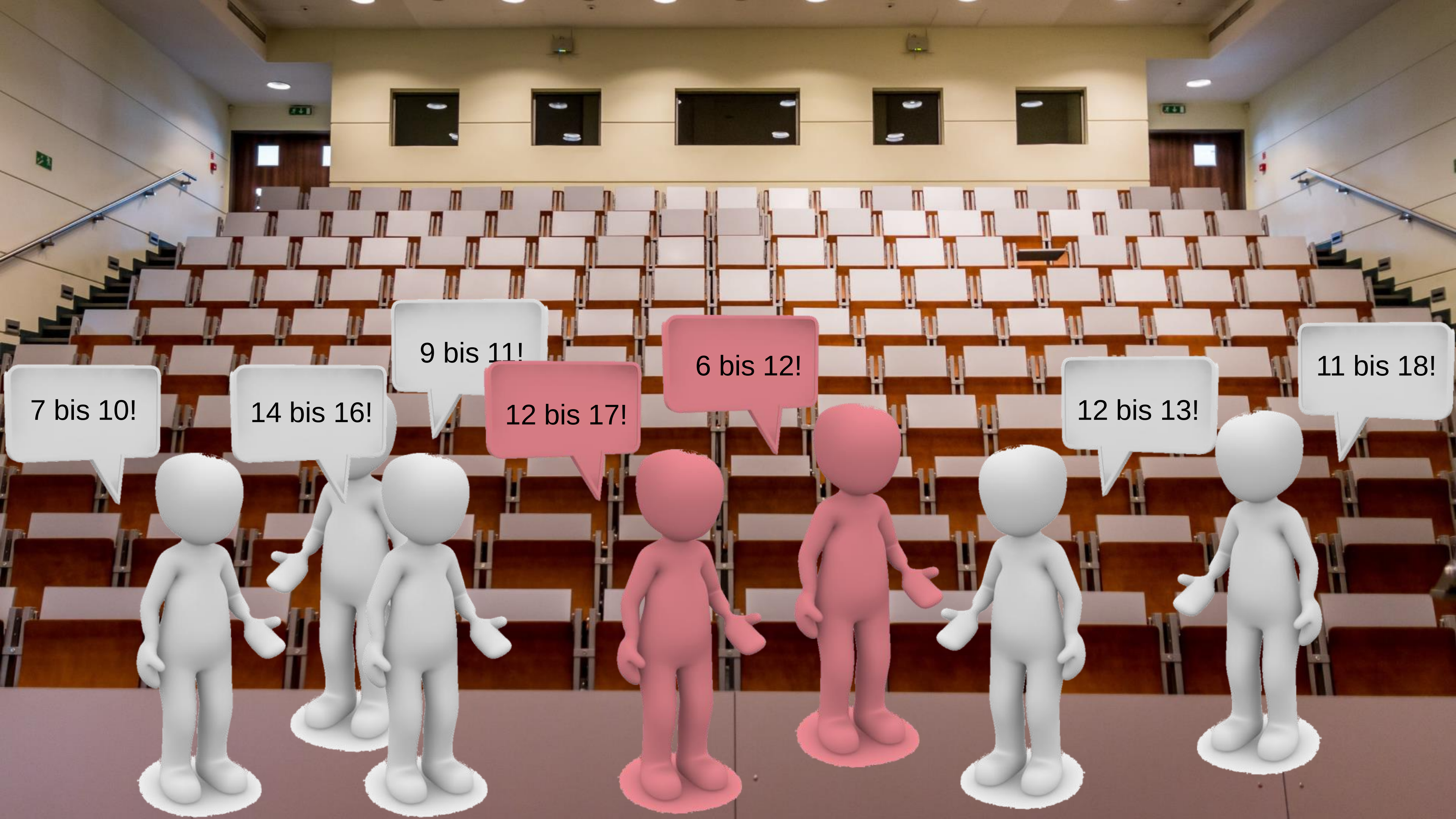
Betrachte das $i + 1$-ste Intervall I_{i+1}, welches den Raum k bekommt.

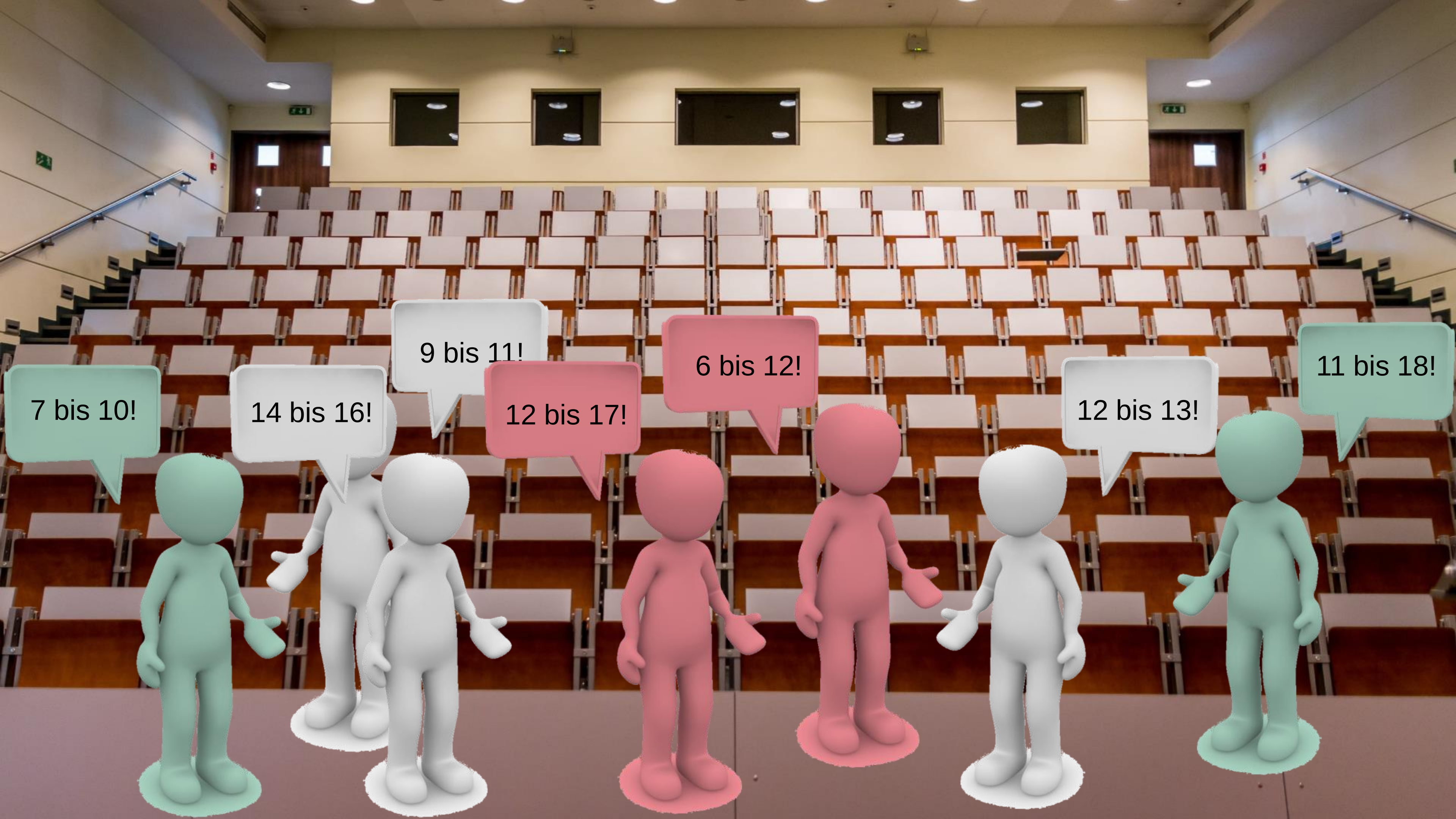
Fall 1: k wurde schon mal vergeben. Nach Annahme gilt weiterhin $alg \leq \chi$

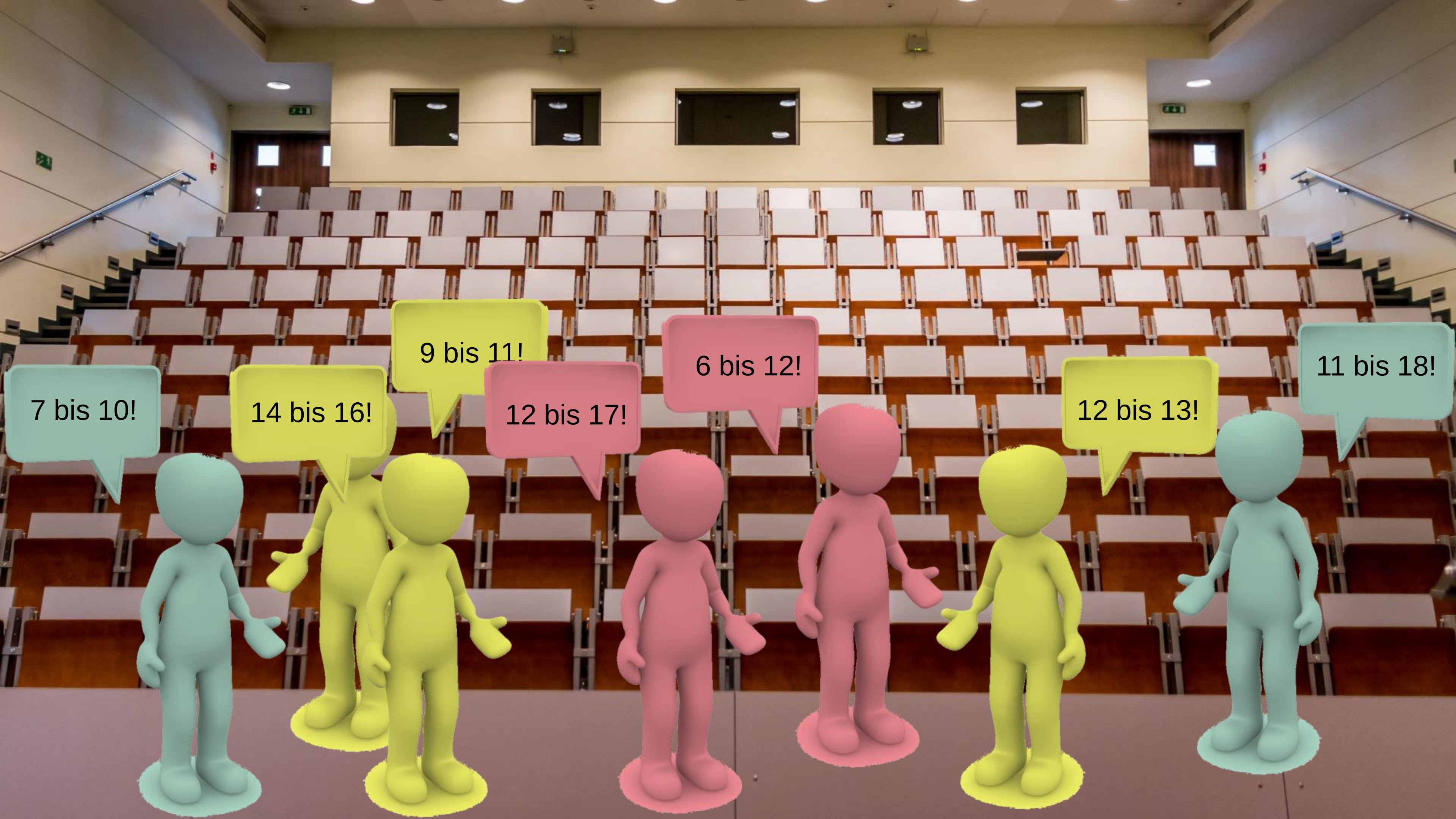
Fall 2: k wurde noch nicht vergeben.
$\Rightarrow$ Zum Zeitpunkt s_{i+1} überlappen sich k Intervalle.
$\Rightarrow \chi \geq k$

In beiden Fällen haben wir auch für das $i + 1$-ste Intervall $\leq \chi$ Räume benutzt.


7 bis 10!
14 bis 16!
9 bis 11!
12 bis 17!
6 bis 12!
12 bis 13!
11 bis 18!


7 bis 10!
14 bis 16!
9 bis 11!
12 bis 17!
6 bis 12!
12 bis 13!
11 bis 18!


7 bis 10!
14 bis 16!
9 bis 11!
12 bis 17!
6 bis 12!
12 bis 13!
11 bis 18!


7 bis 10!
14 bis 16!
9 bis 11!
12 bis 17!
6 bis 12!
12 bis 13!
11 bis 18!