



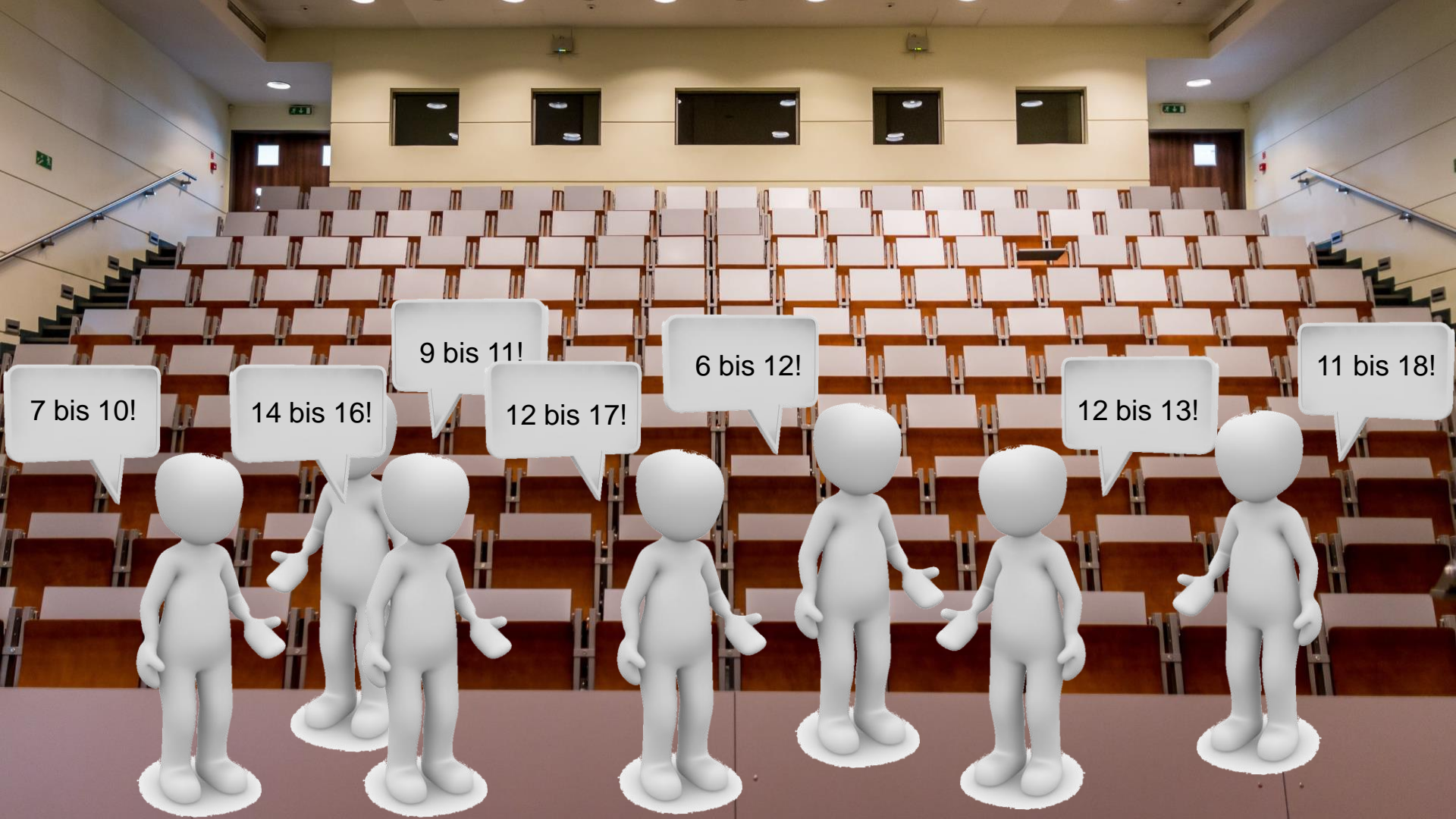
Technische
Universität
Braunschweig



Algorithmen und Datenstrukturen 2 – Übung #1

Matthias Konitzny

05.05.2021



7 bis 10!

14 bis 16!

9 bis 11!

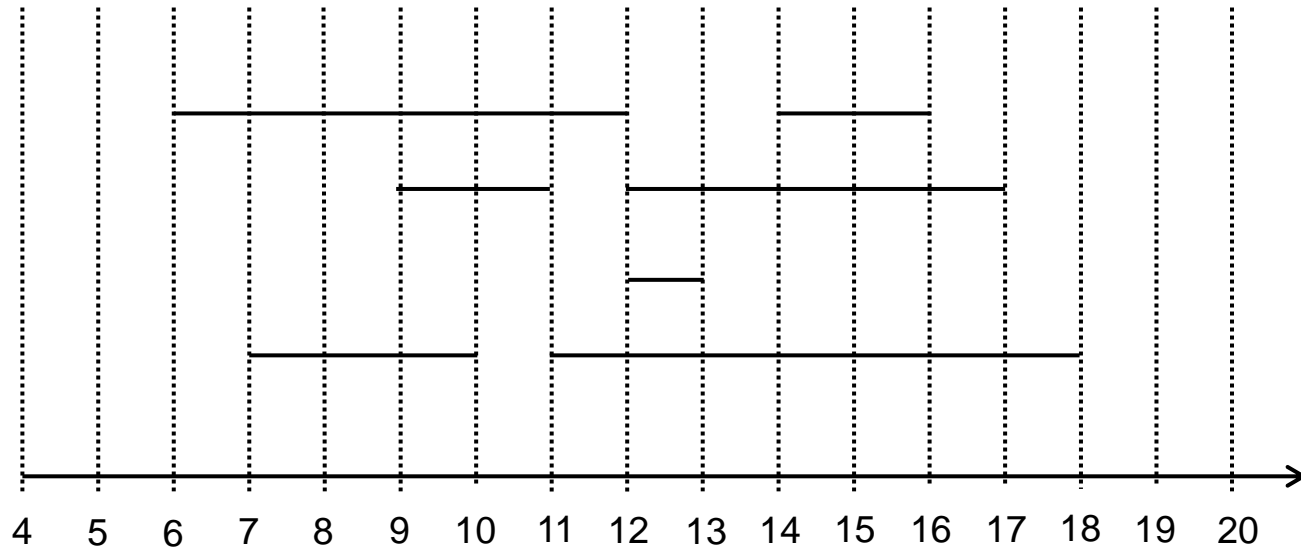
12 bis 17!

6 bis 12!

12 bis 13!

11 bis 18!

Hörsaal-Belegung



Hörsaal-Belegung – Das Problem

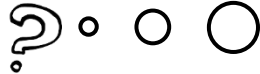
Gegeben: Menge von Intervallen $\mathcal{I} = \{I_1 = [s_1, e_1), \dots, I_n = [s_n, e_n)\}$

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.

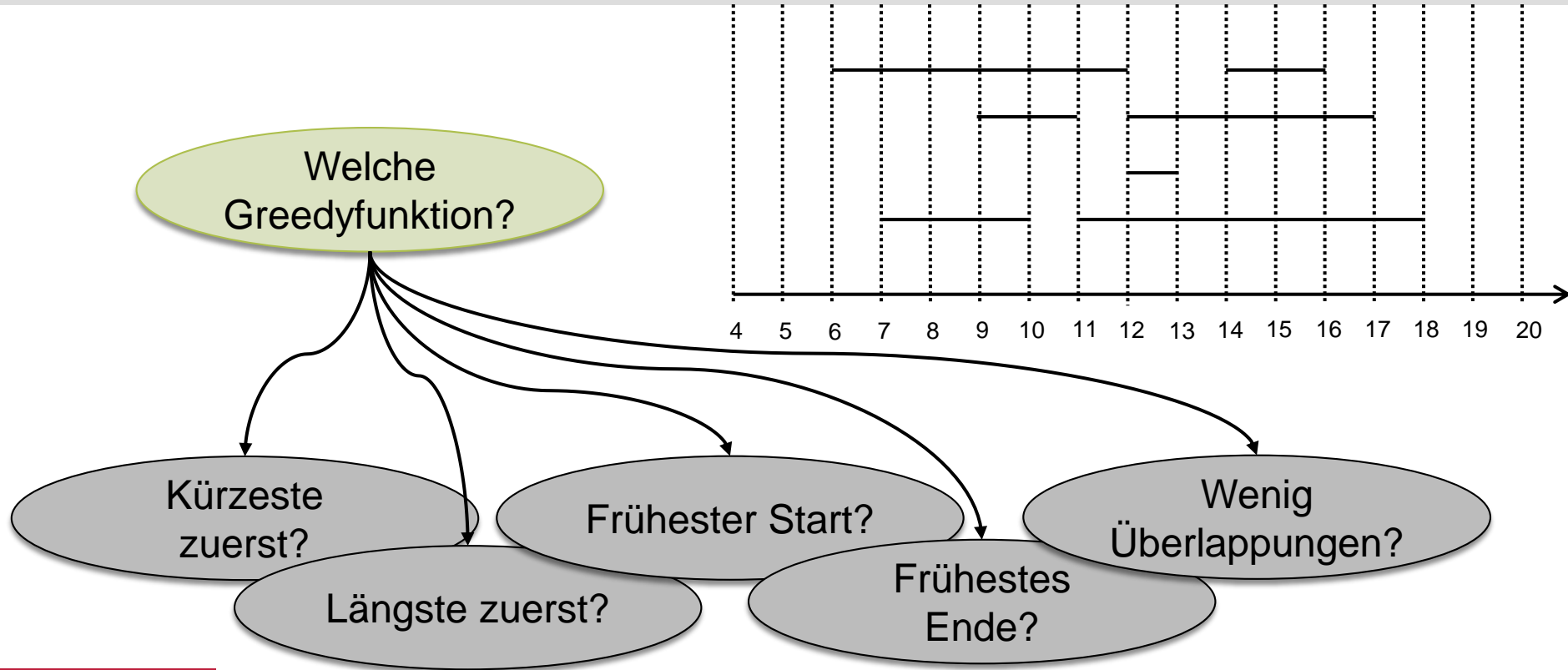
Die ausgewählten
Intervalle müssen
disjunkt sein.



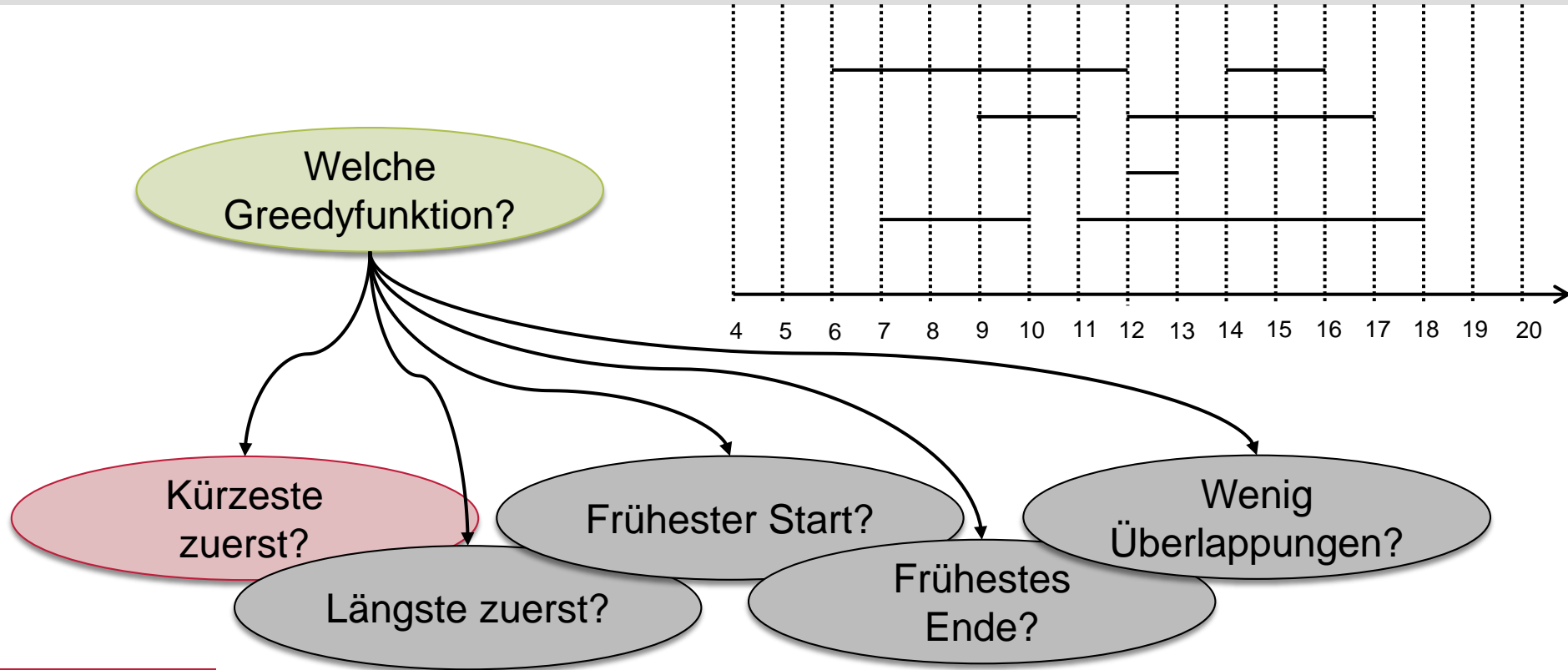
Wie löst
man das
Problem?

Greedy?!

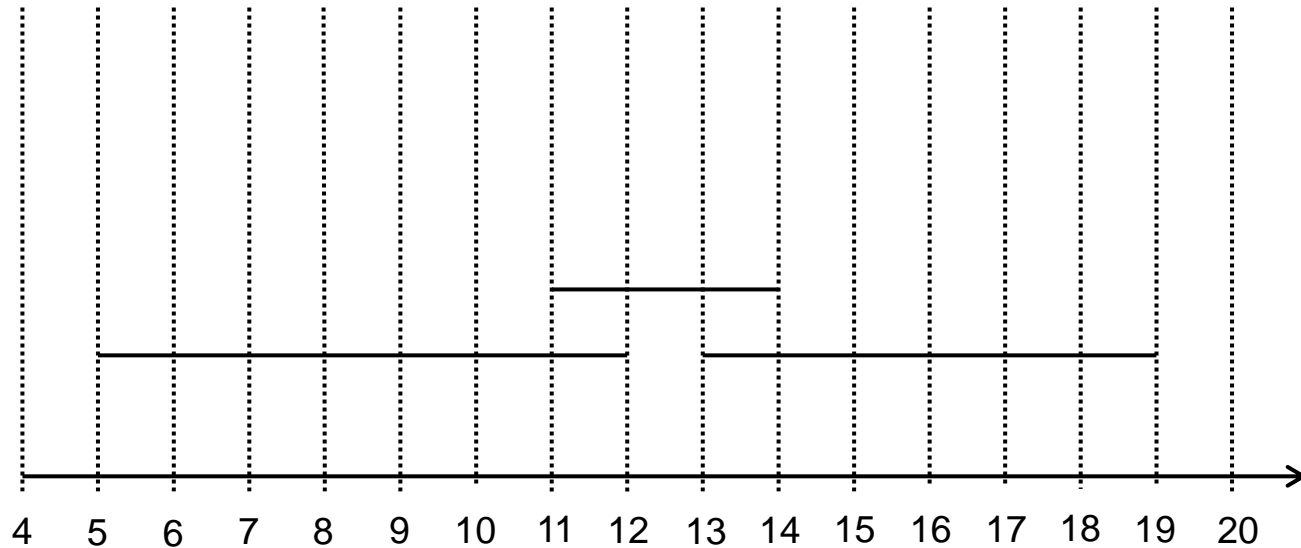
Hörsaal-Belegung – Strategien



Hörsaal-Belegung – Strategien



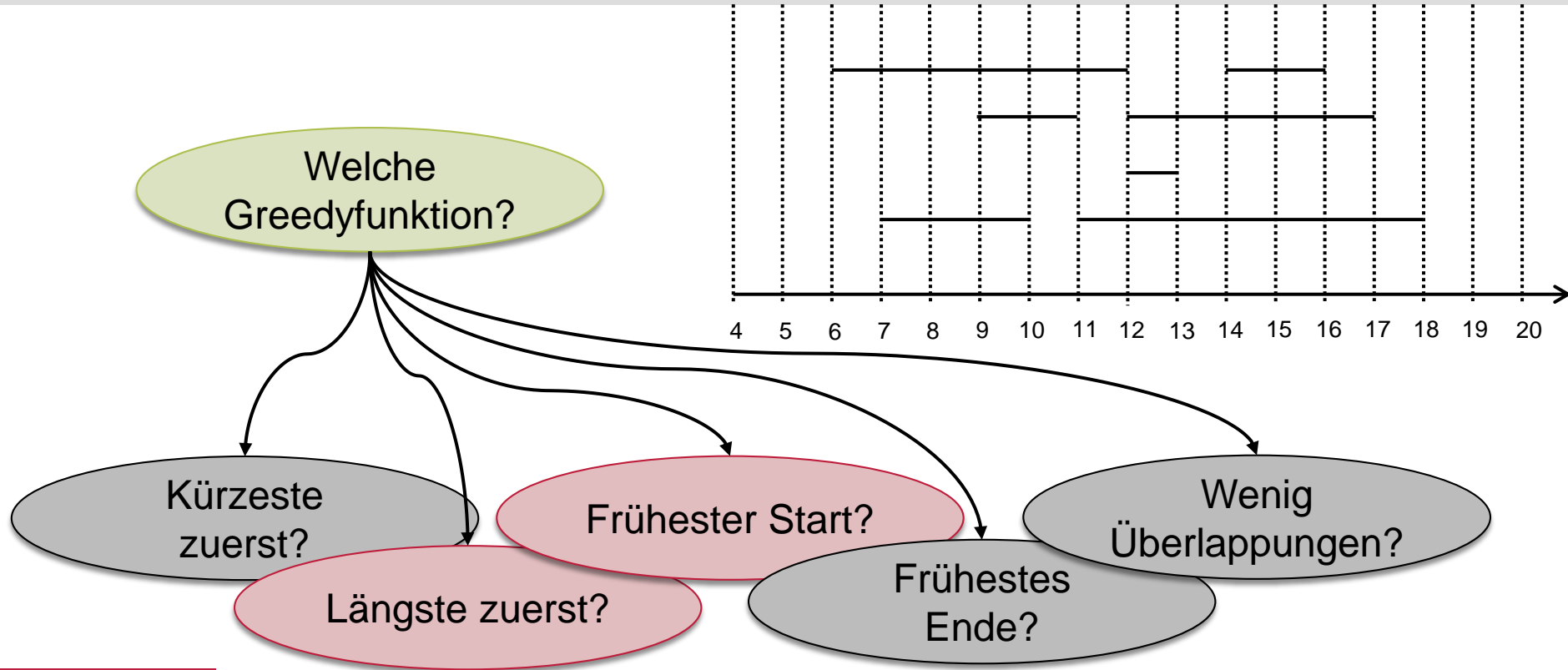
Hörsaal-Belegung – Kürzeste zuerst



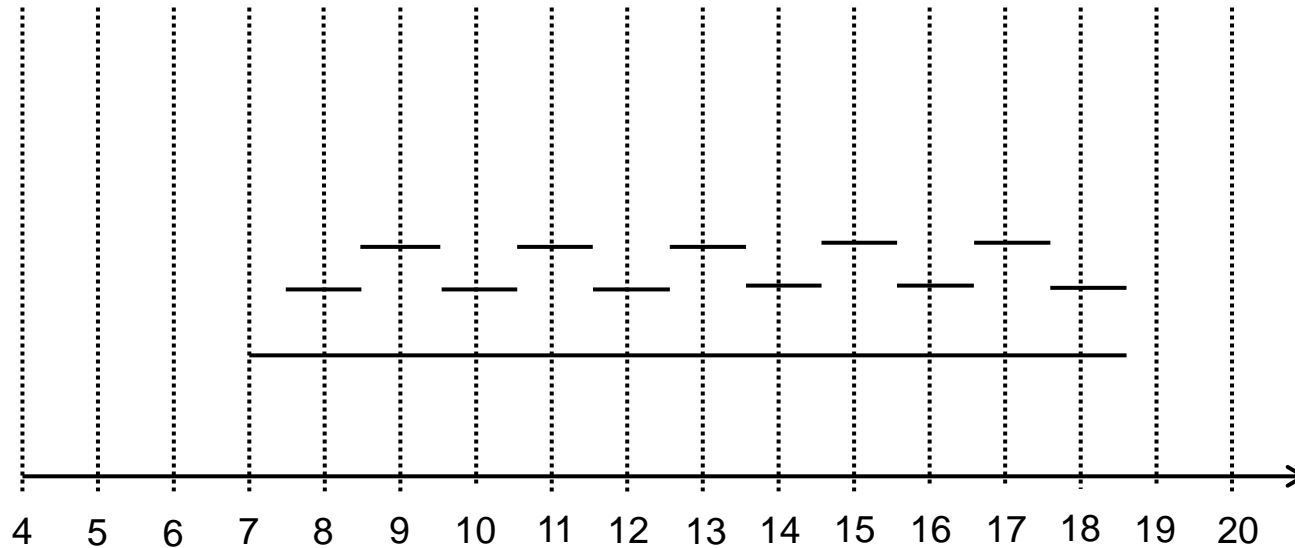
ALG: 1

OPT: 2

Hörsaal-Belegung – Strategien



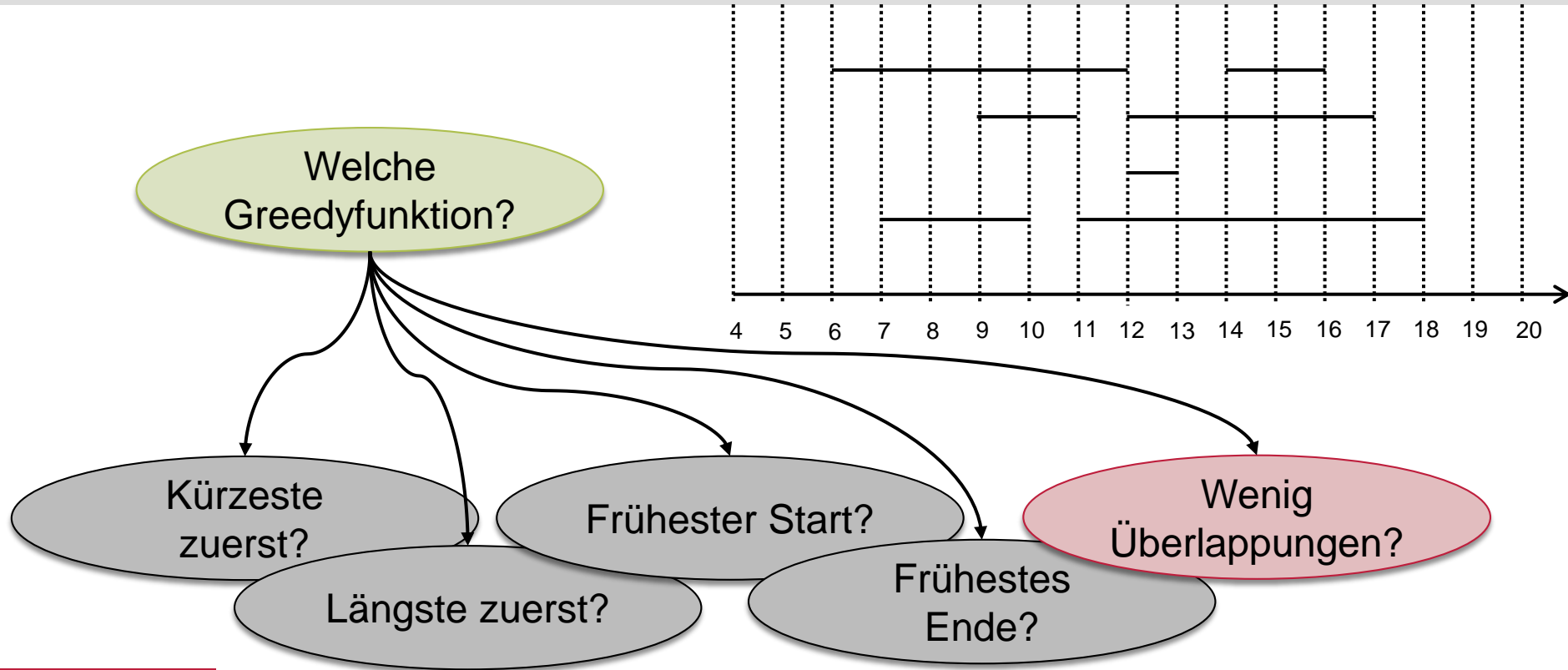
Hörsaal-Belegung – Frühester Start/Längstes Intervall



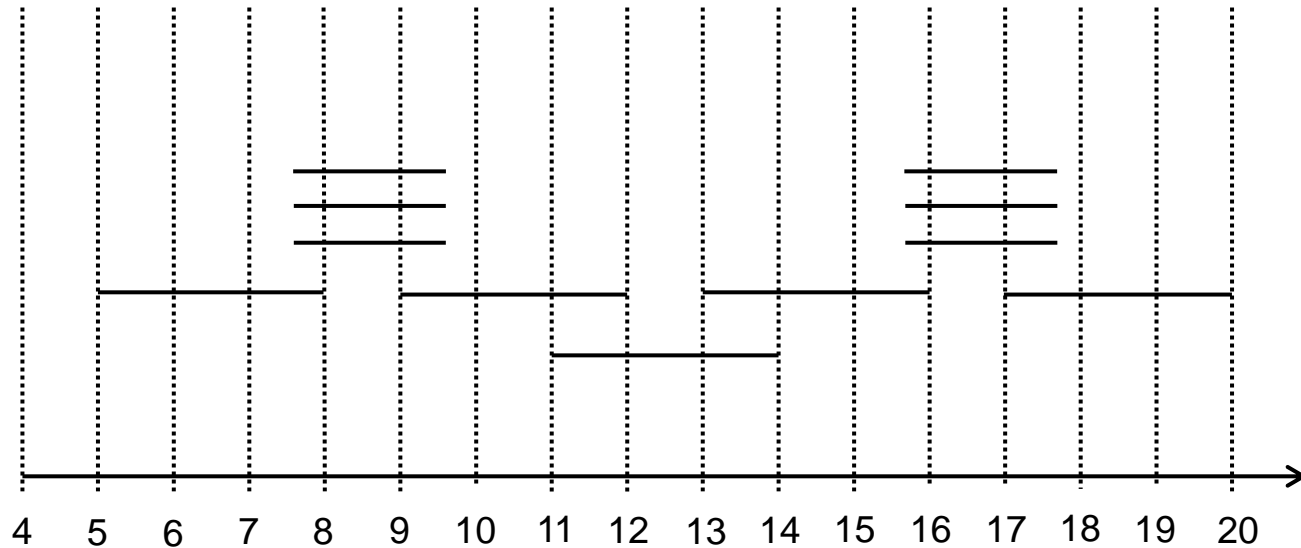
ALG: 1

OPT: 11

Hörsaal-Belegung – Strategien

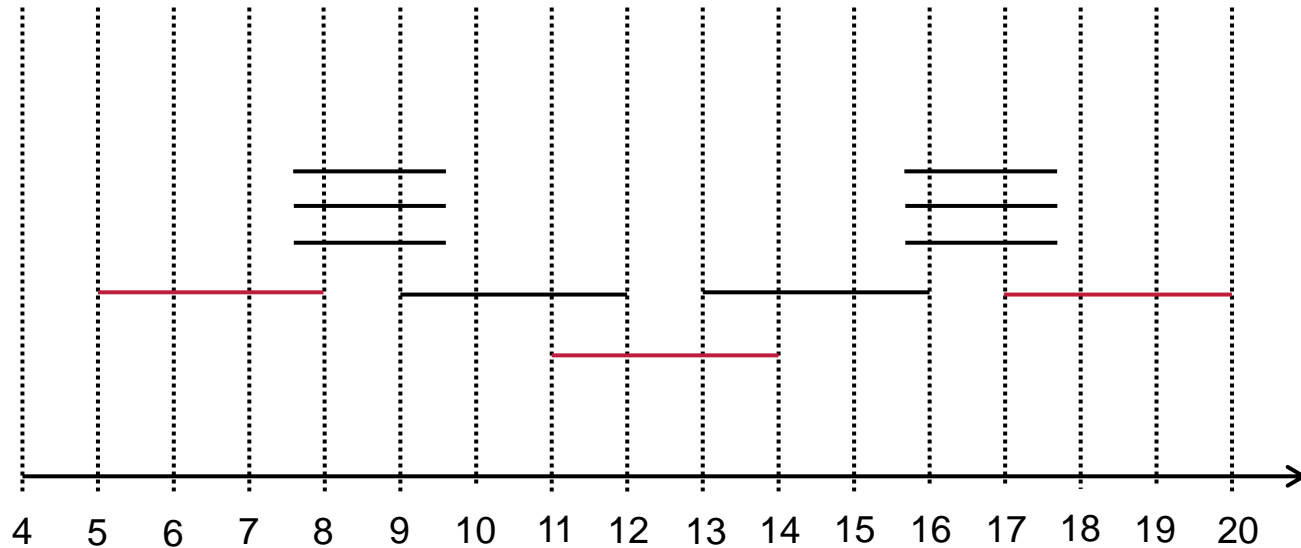


Hörsaal-Belegung – Wenig Überlappungen



ALG: 3

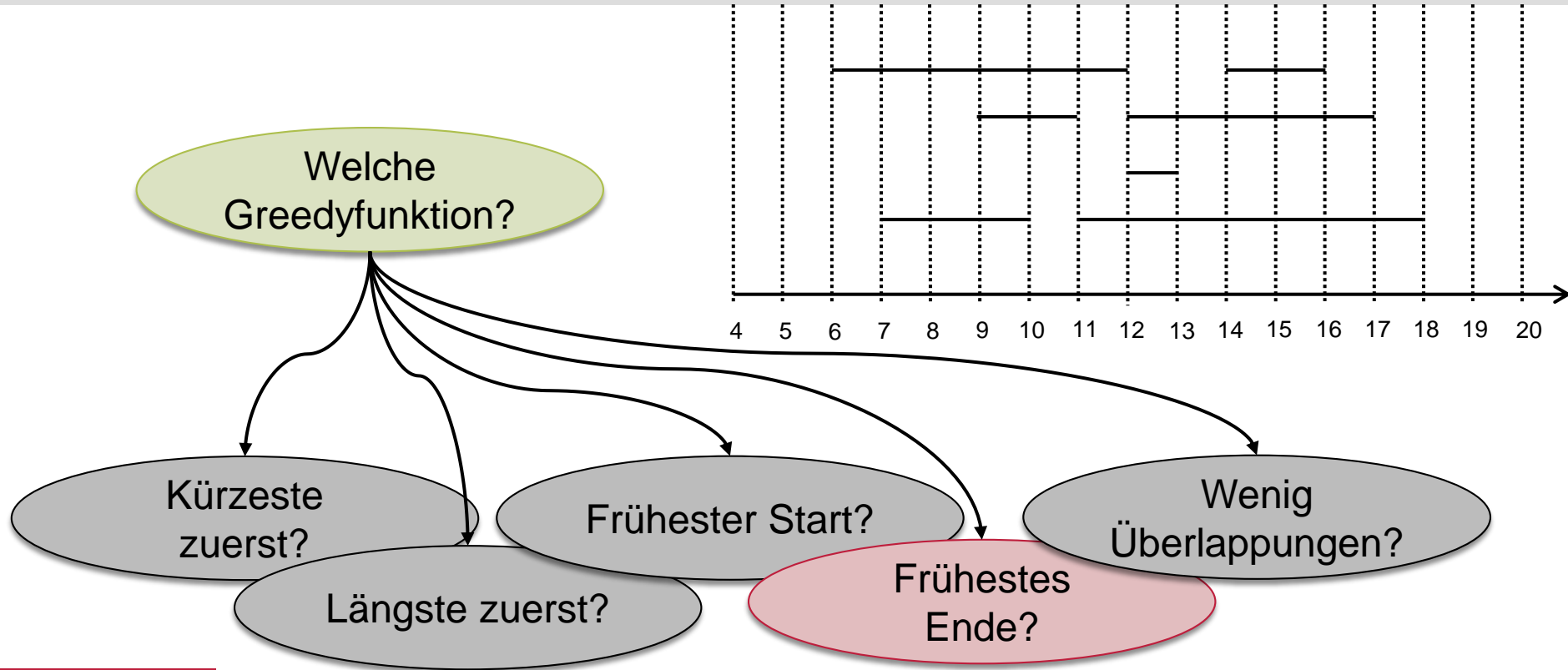
Hörsaal-Belegung – Wenig Überlappungen



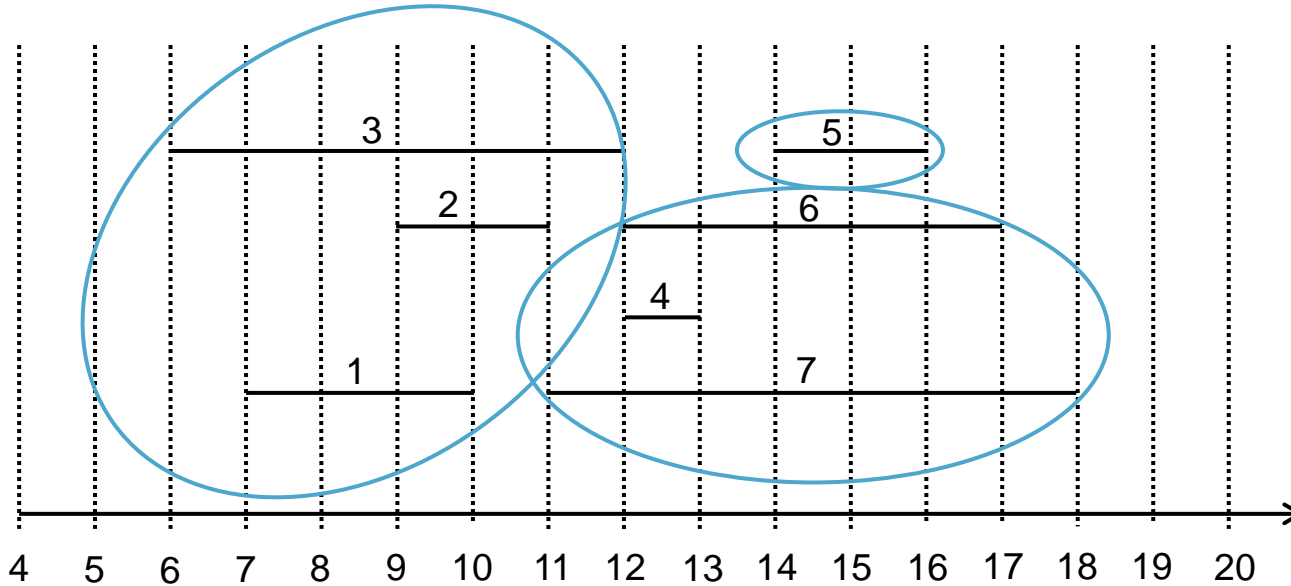
ALG: 3

OPT: 4

Hörsaal-Belegung – Strategien



Hörsaal-Belegung – Frühestes Ende



ALG: 3

OPT: 3?

Hörsaal-Belegung – Frühestes Ende

Algorithmus 1

```
1: sortiere  $\{1, \dots, n\}$  nach  $e_i$  aufsteigend  $\rightarrow$  Permutation  $\pi(1), \dots, \pi(n)$ 
2:  $S := \{\pi(1)\}$ 
3:  $e := e_{\pi(1)}$ 
4: for  $k = 2$  to  $n$  do
5:   if  $(s_{\pi(k)} \geq e)$  then
6:      $S := S \cup \{\pi(k)\}$ 
7:      $e := e_{\pi(k)}$ 
8: return  $S$ 
```

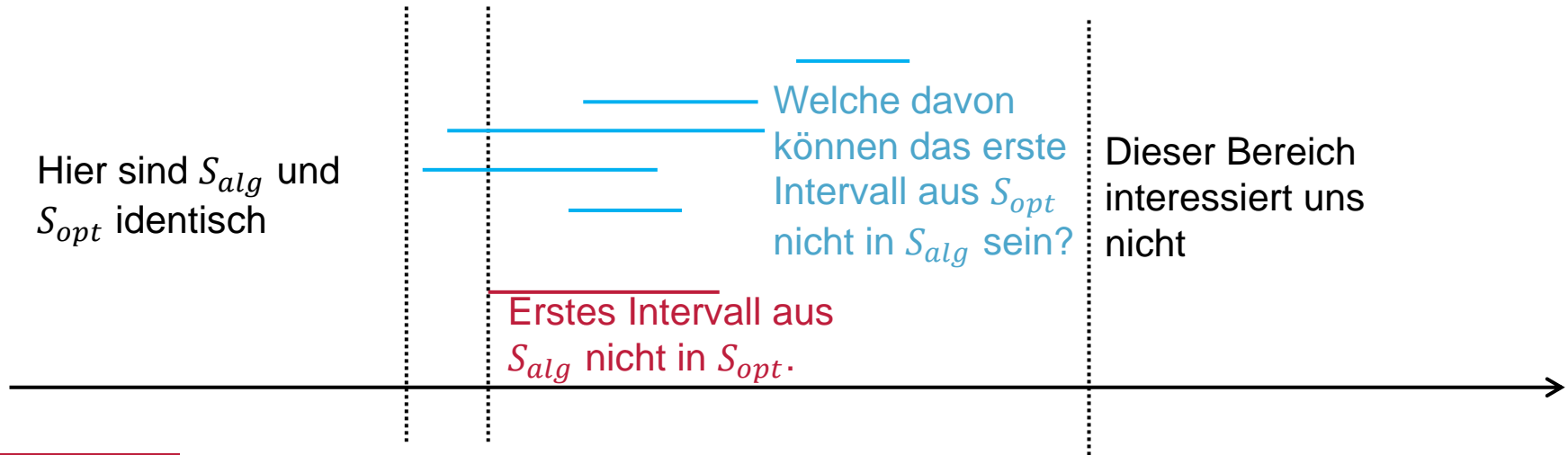
Theorem:

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.

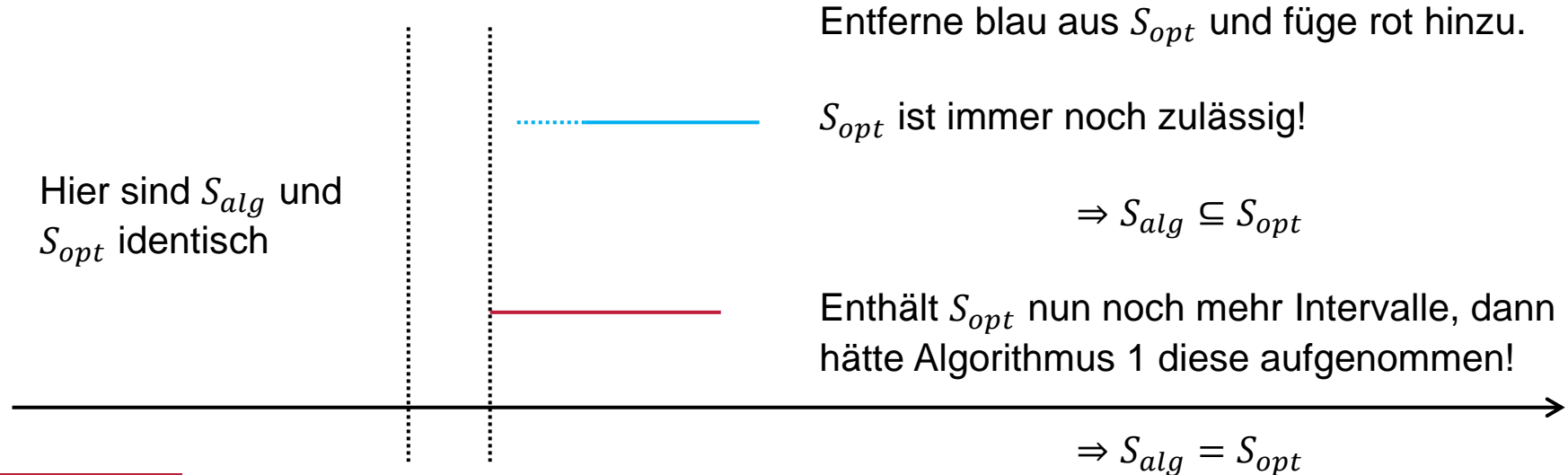
Sei S_{alg} die Lösung von Algorithmus 1 und S_{opt} eine optimale Lösung.

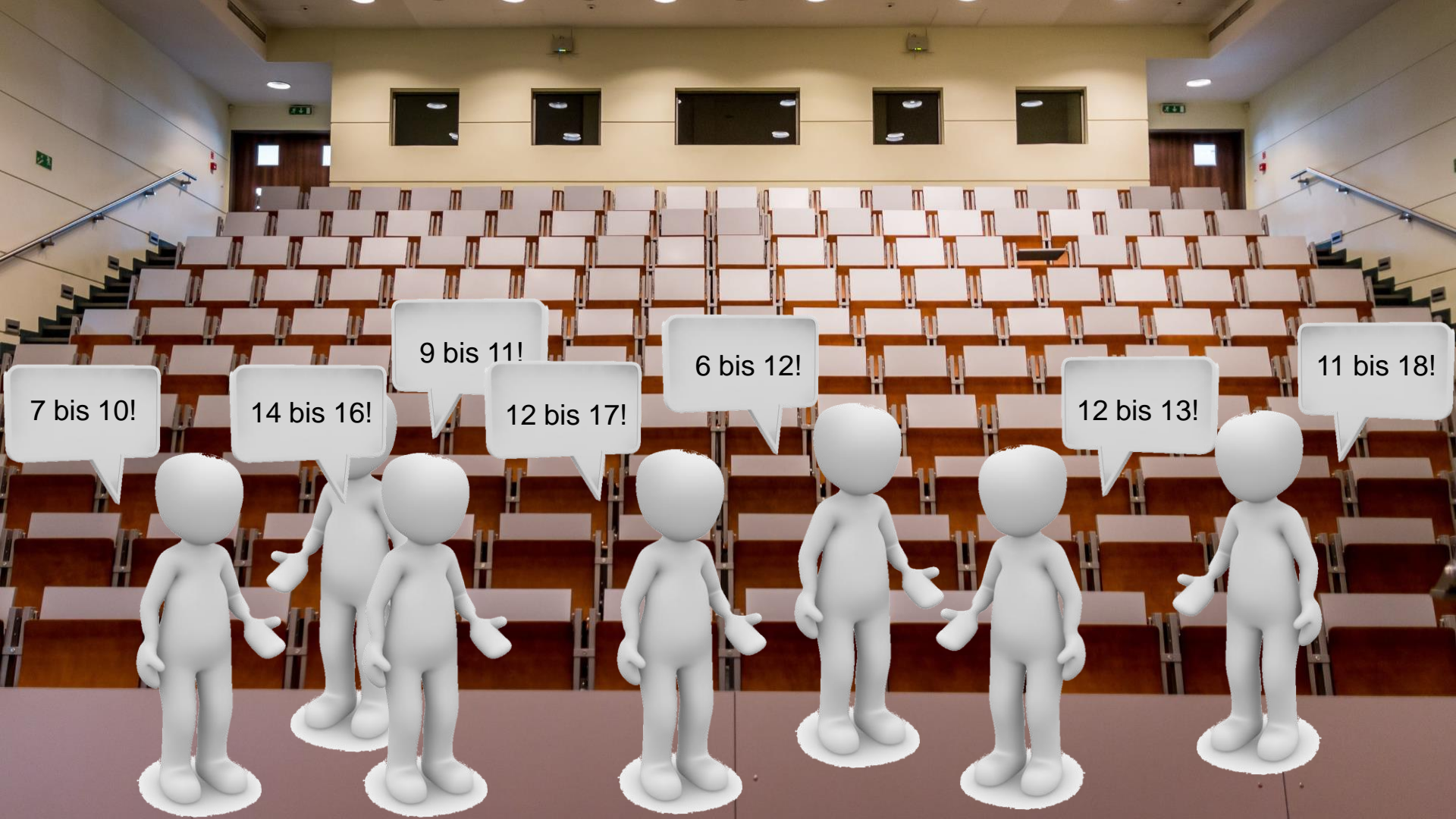


Beweis

Annahme: Algorithmus 1 sei nicht optimal.

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





7 bis 10!

14 bis 16!

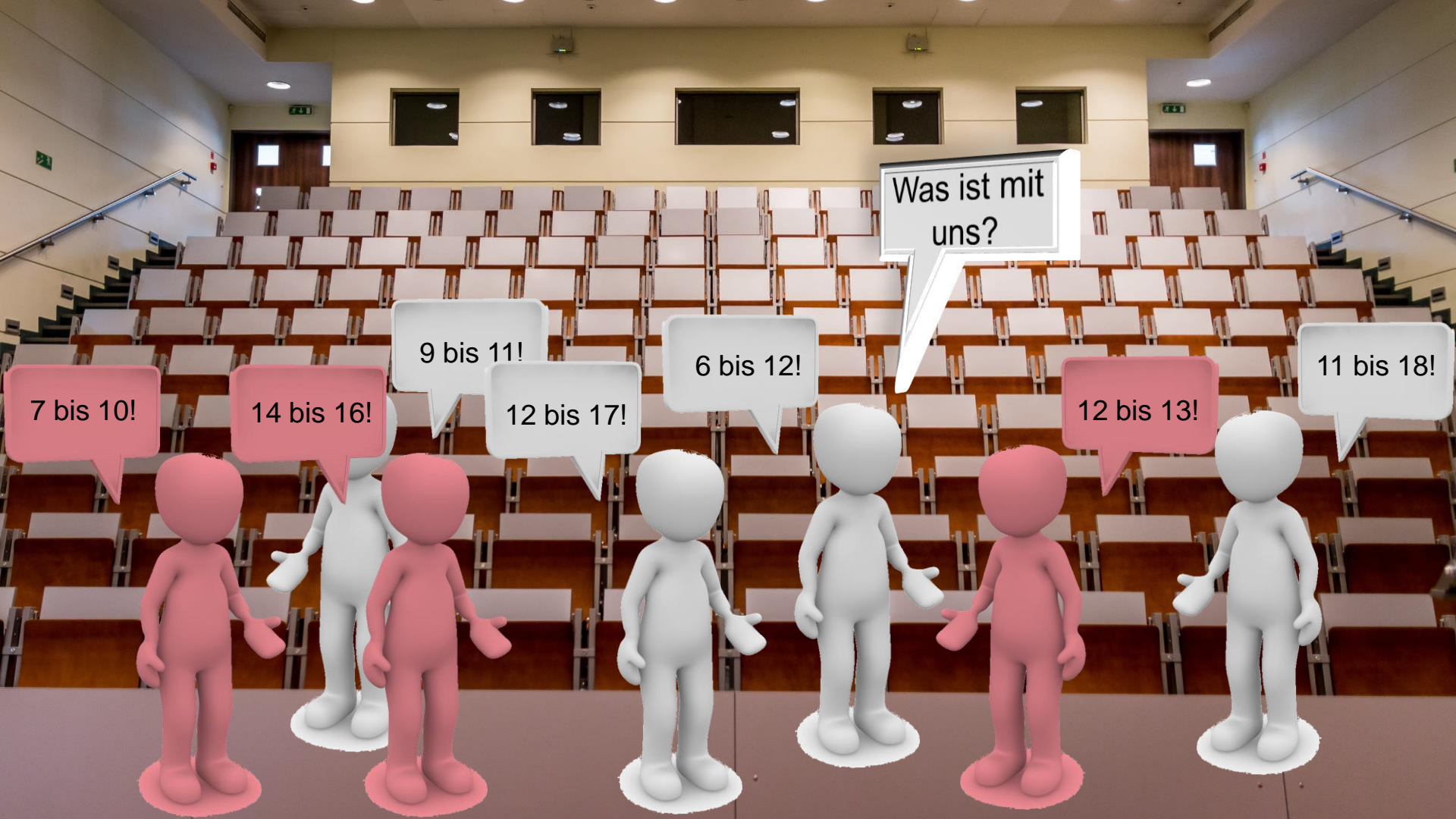
9 bis 11!

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!

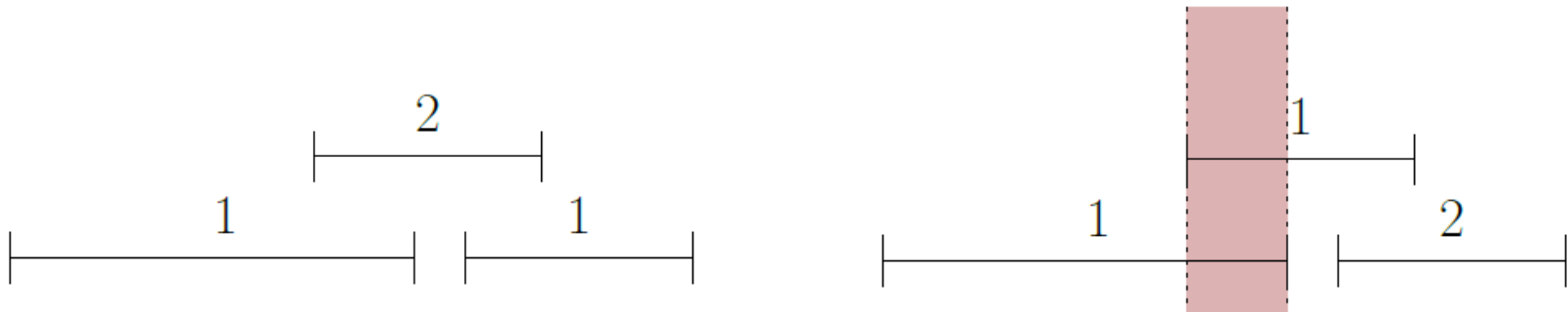
11 bis 18!

Was ist mit uns?

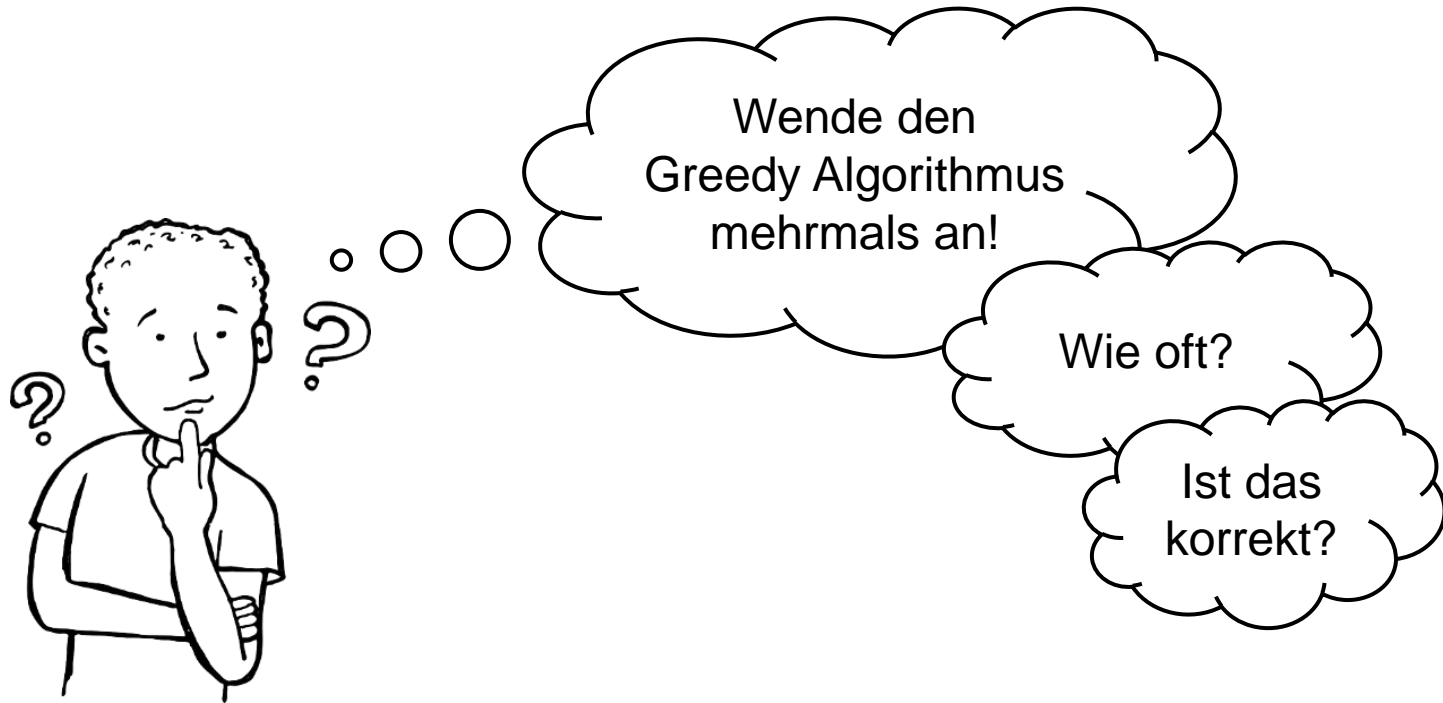
Hörsaal-Belegung – Eine Variante

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

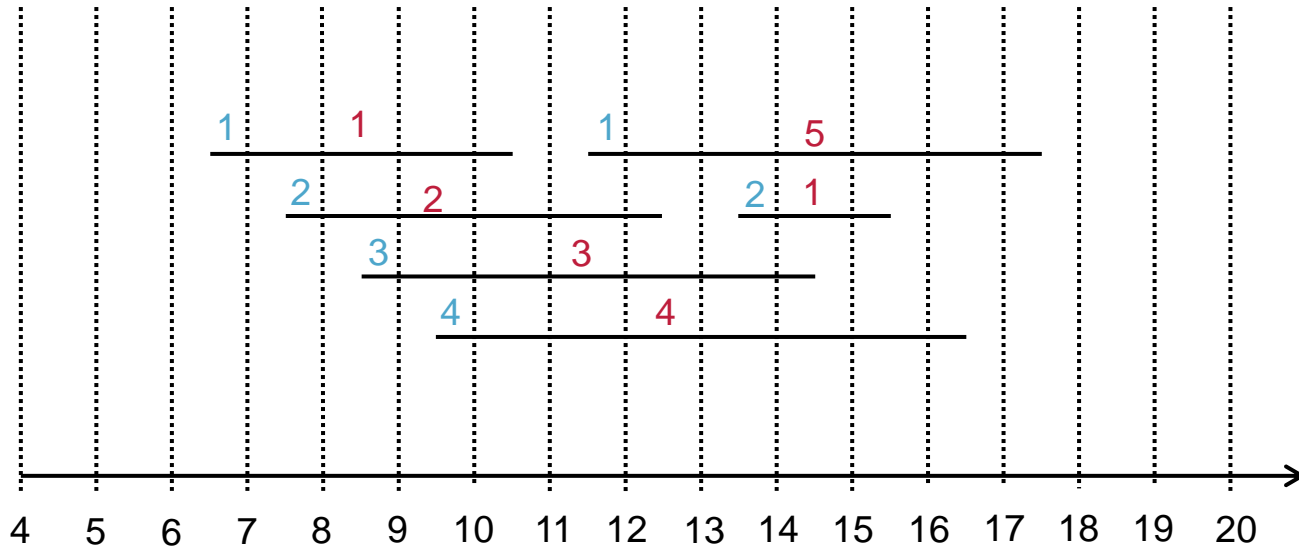
Gesucht: Die kleinste Zahl $k \in \mathbb{N}$, sodass folgendes gilt. Es gibt eine Funktion $f : \mathcal{I} \rightarrow \{1, \dots, k\}$, sodass für je zwei Intervalle I und I' gilt $I \cap I' \neq \emptyset \Rightarrow f(I) \neq f(I')$



Hörsaal-Belegung – Eine Variante



Hörsaal-Belegung – Eine Variante

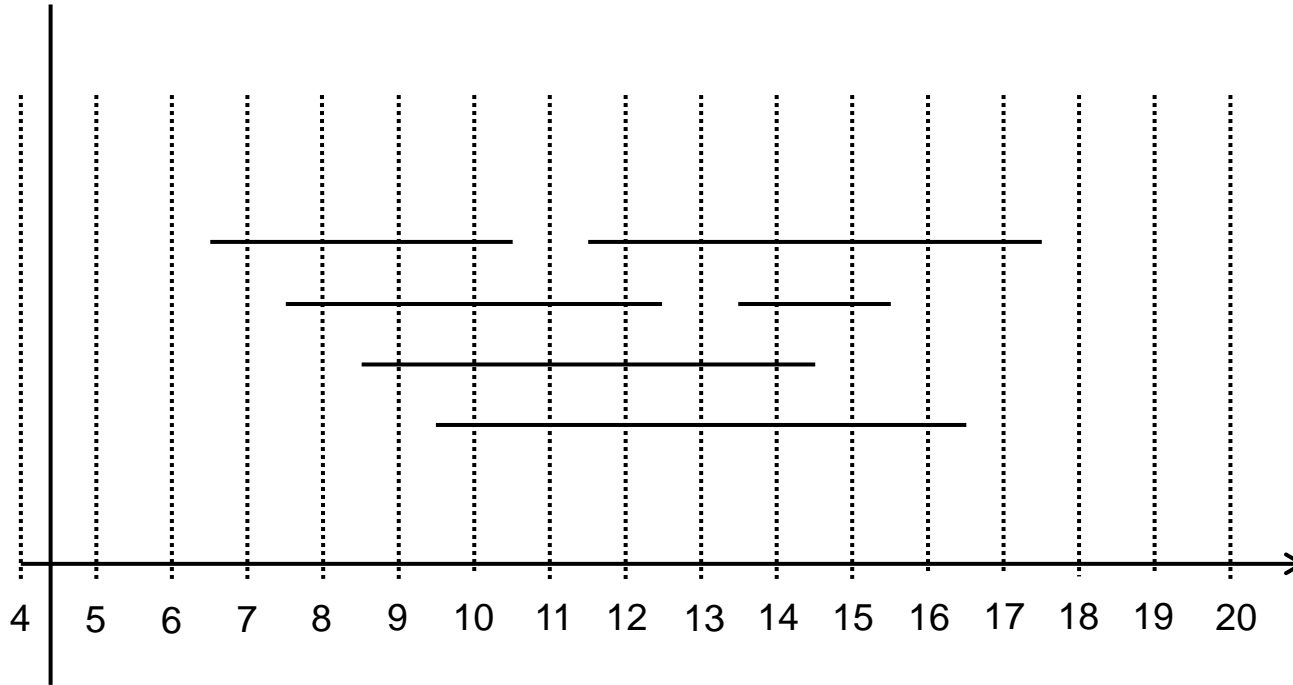


ALG: 5

OPT: 4

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

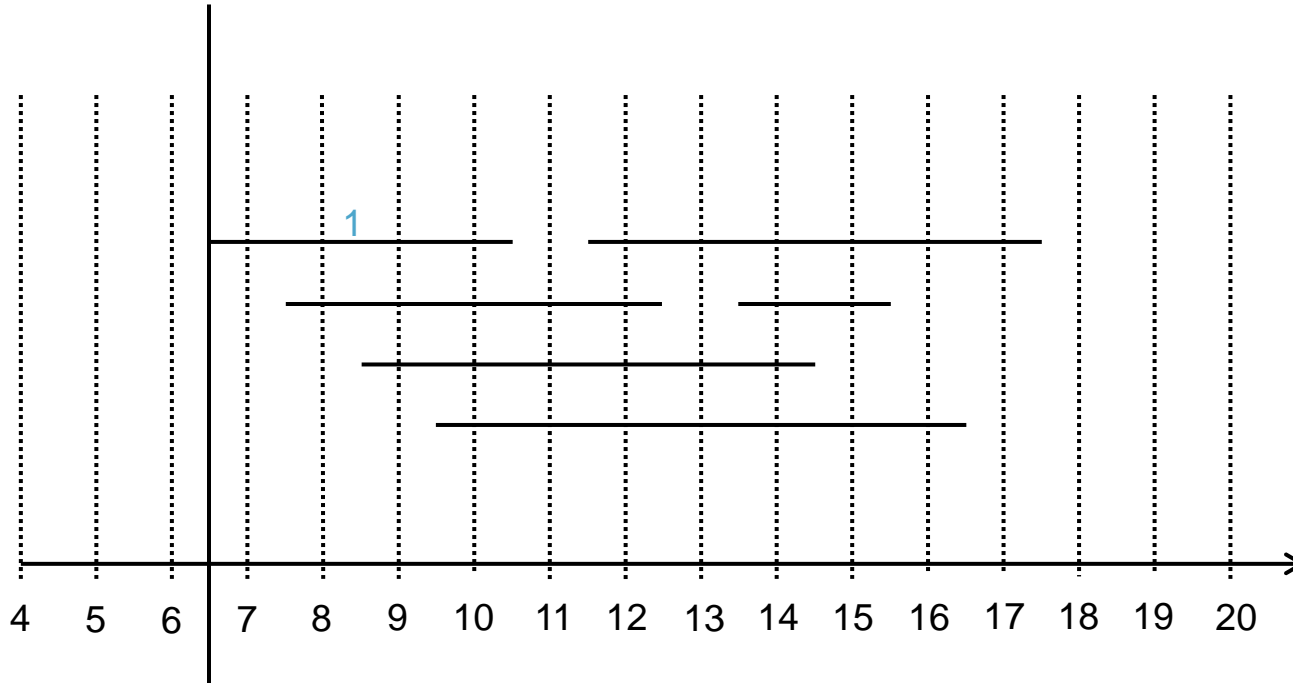
Hörsaal-Belegung – Eine Variante



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

Bewege eine Linie von links nach rechts

Hörsaal-Belegung – Eine Variante

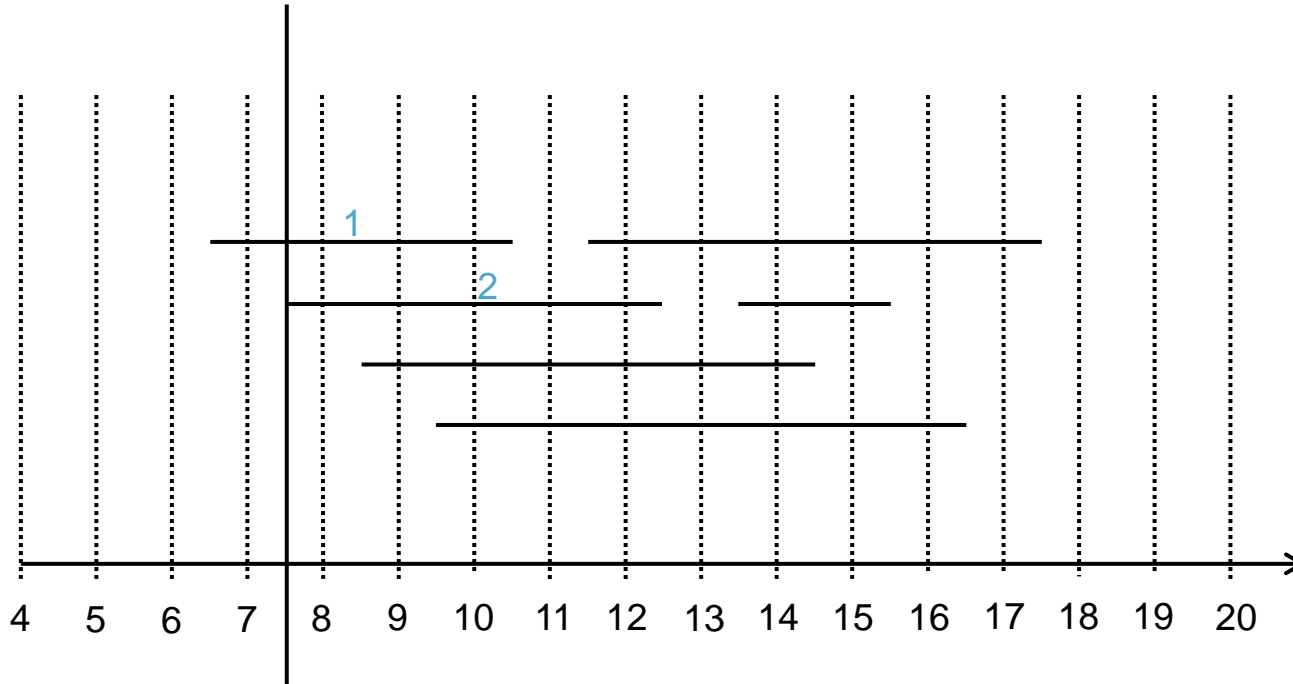


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

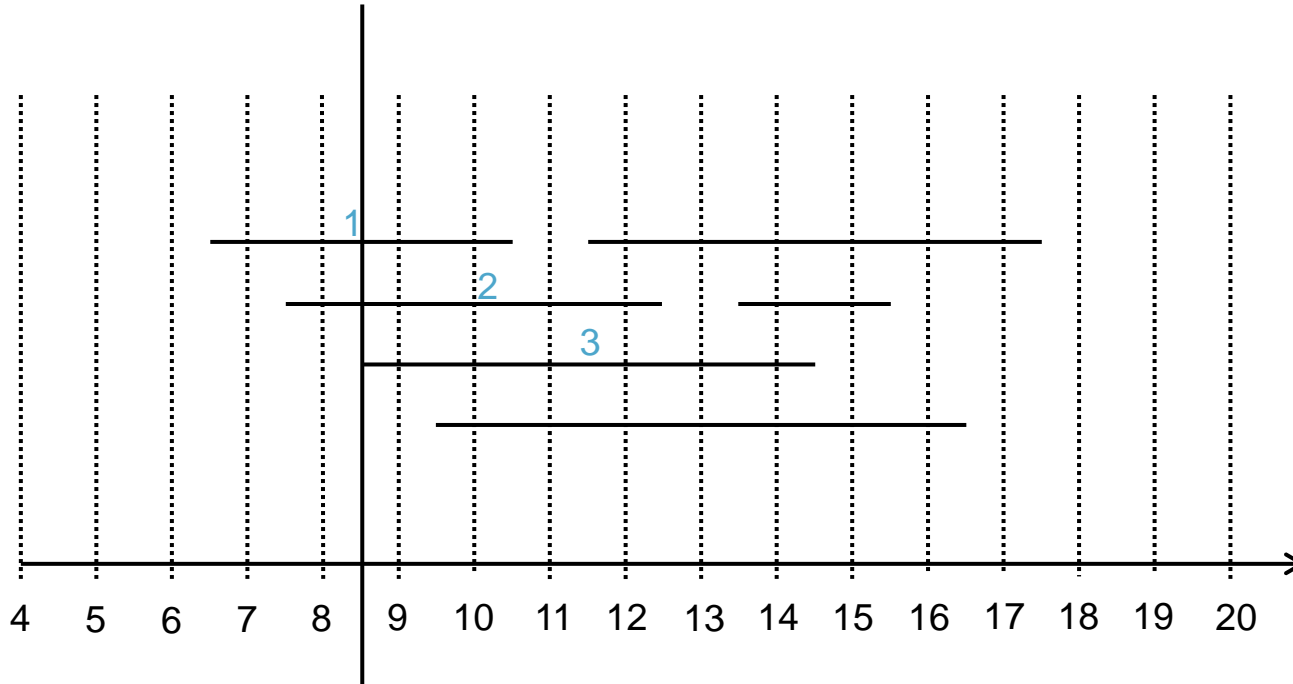


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

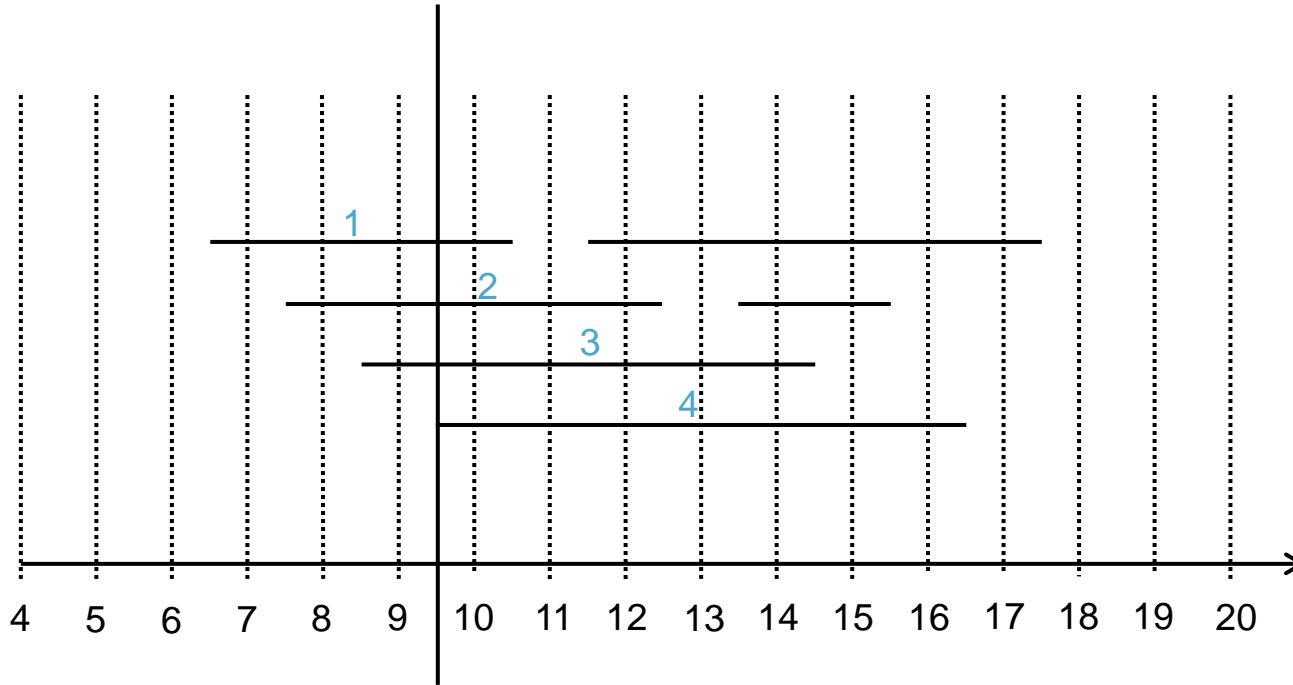


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

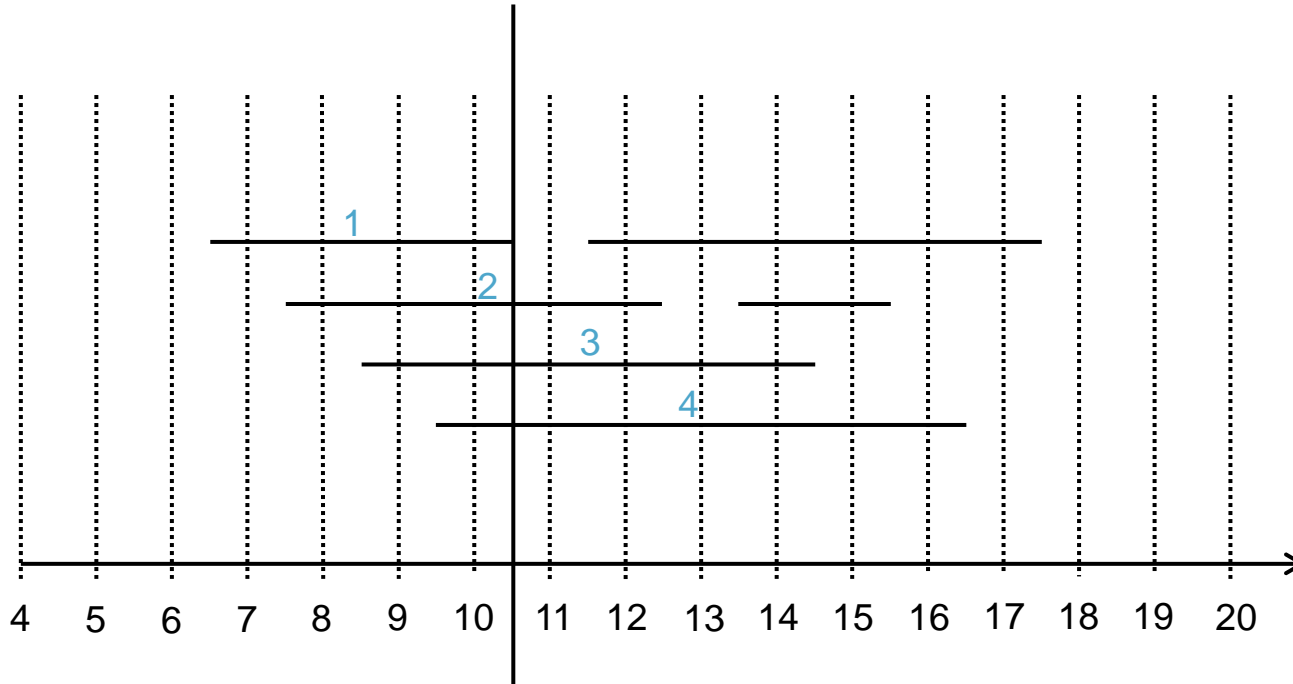


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

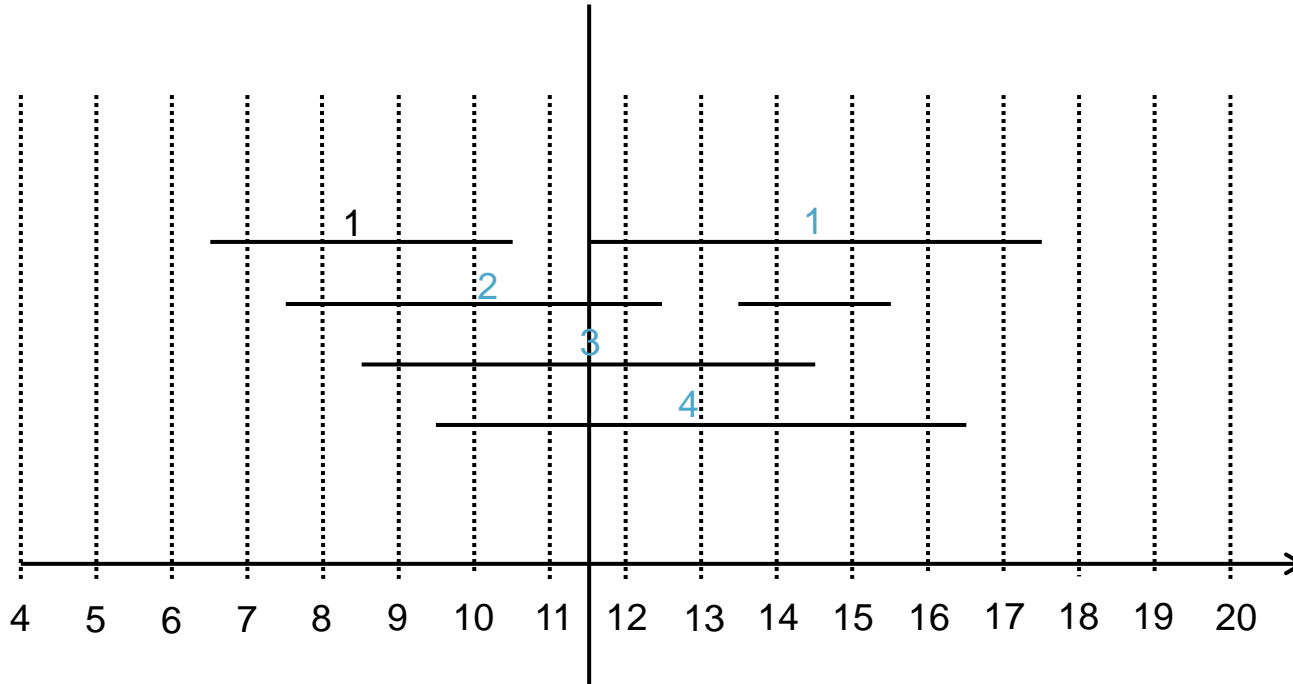


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

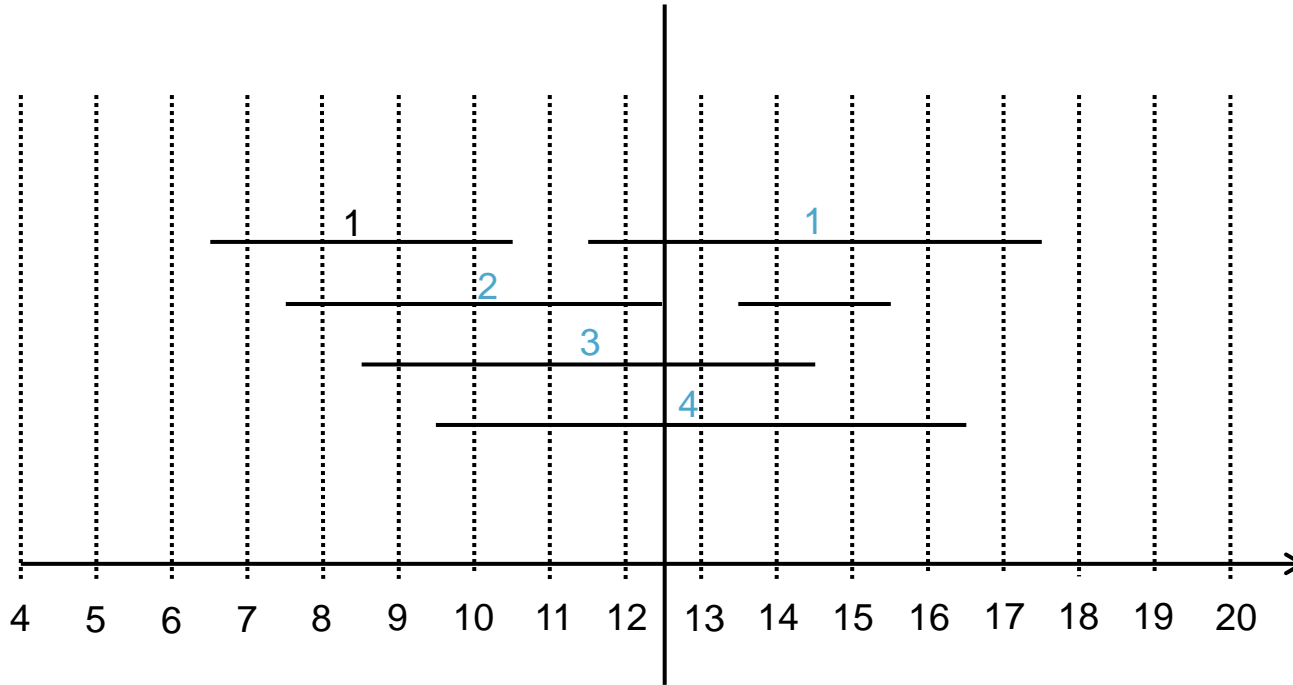


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

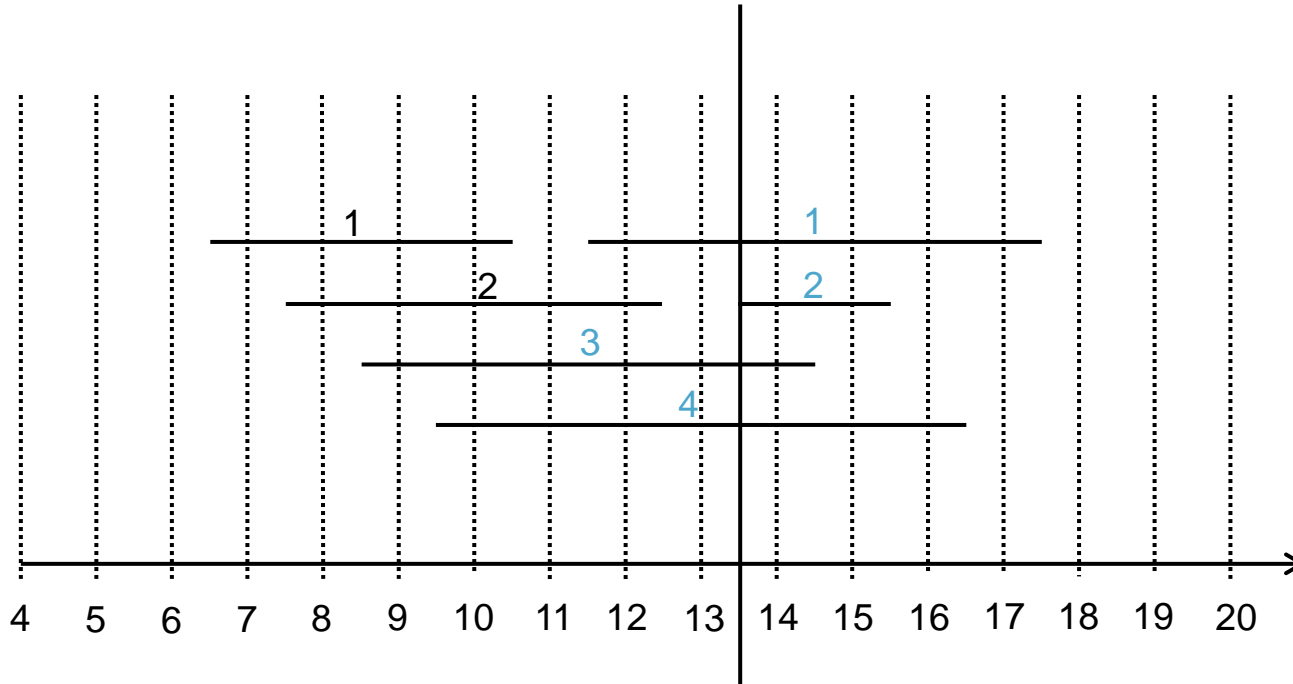


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

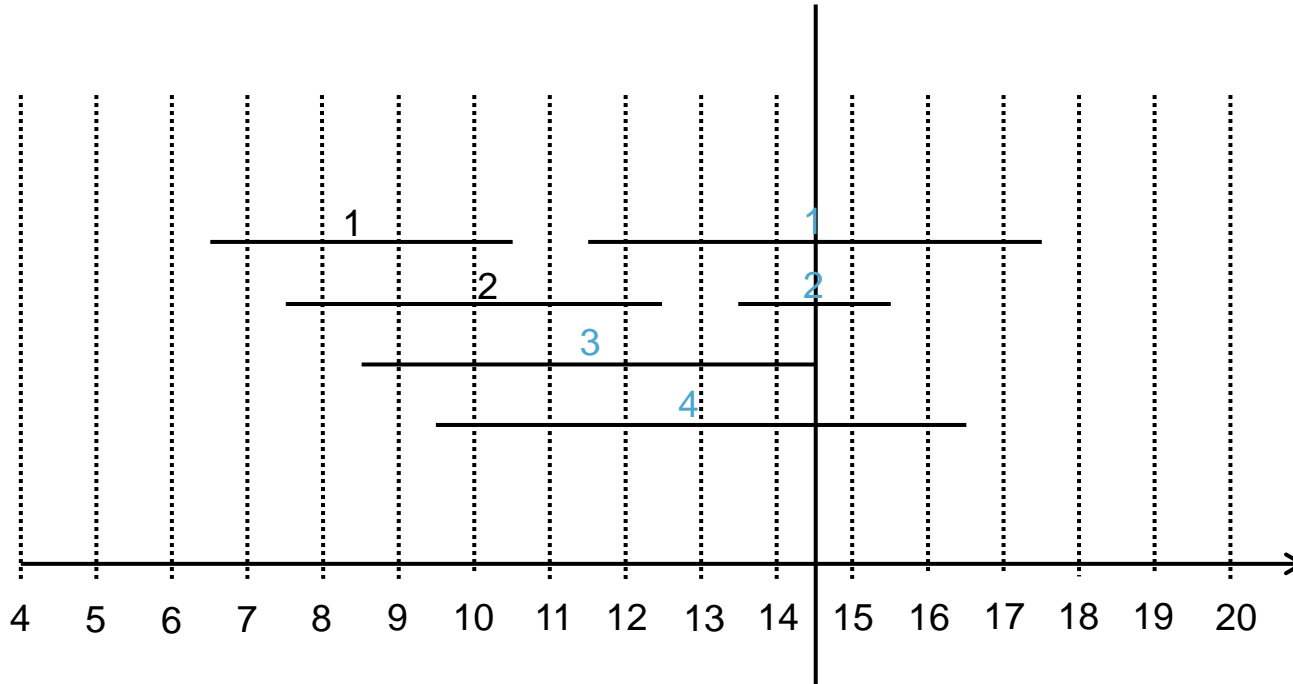


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

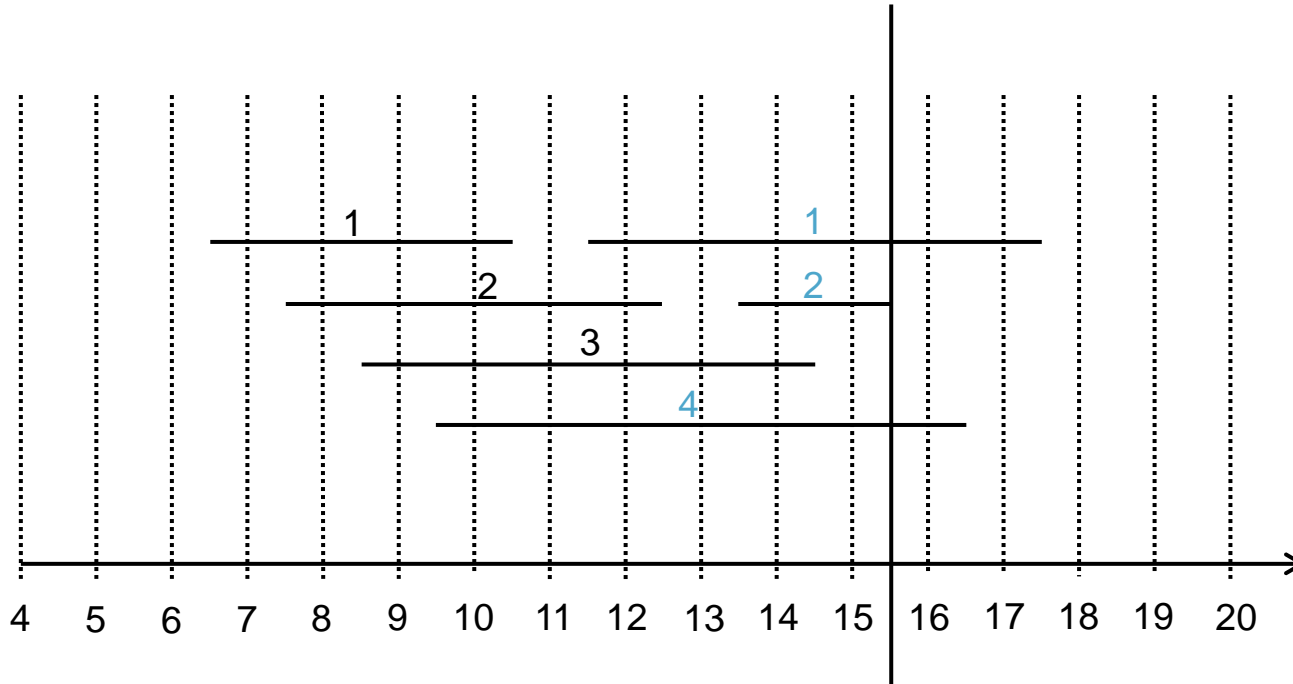


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

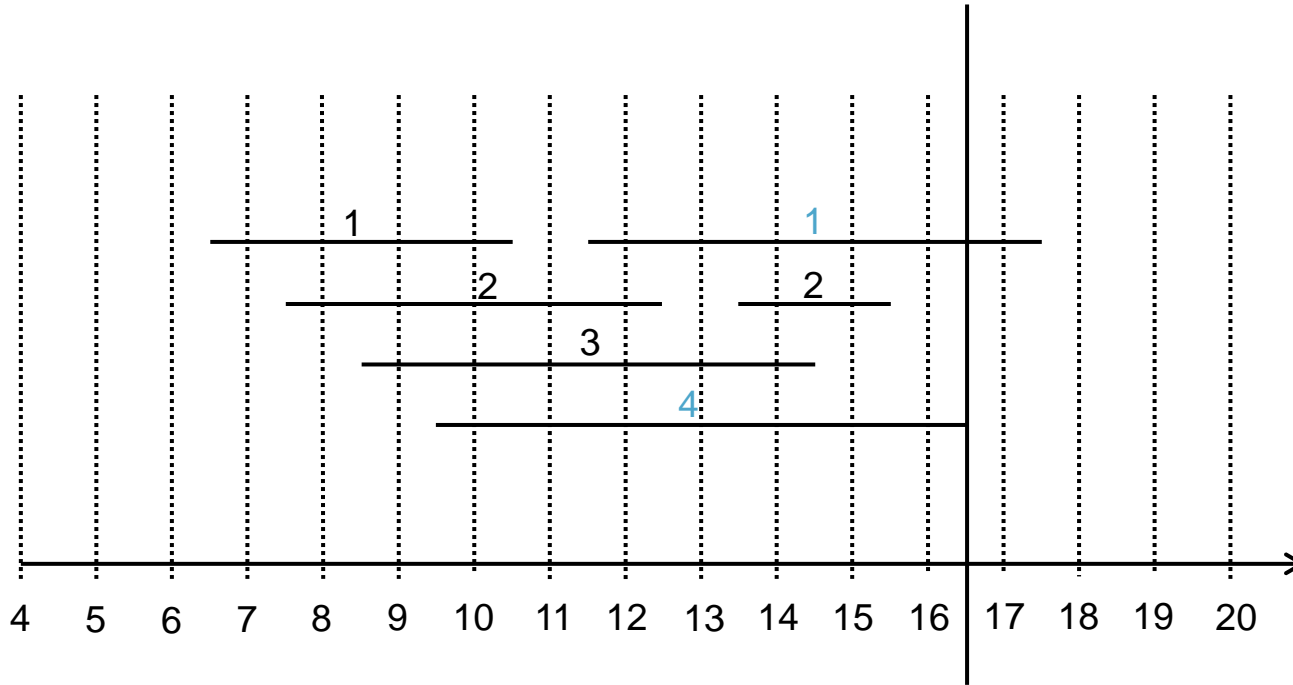


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

Bei Erreichen eines

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

Hörsaal-Belegung – Eine Variante

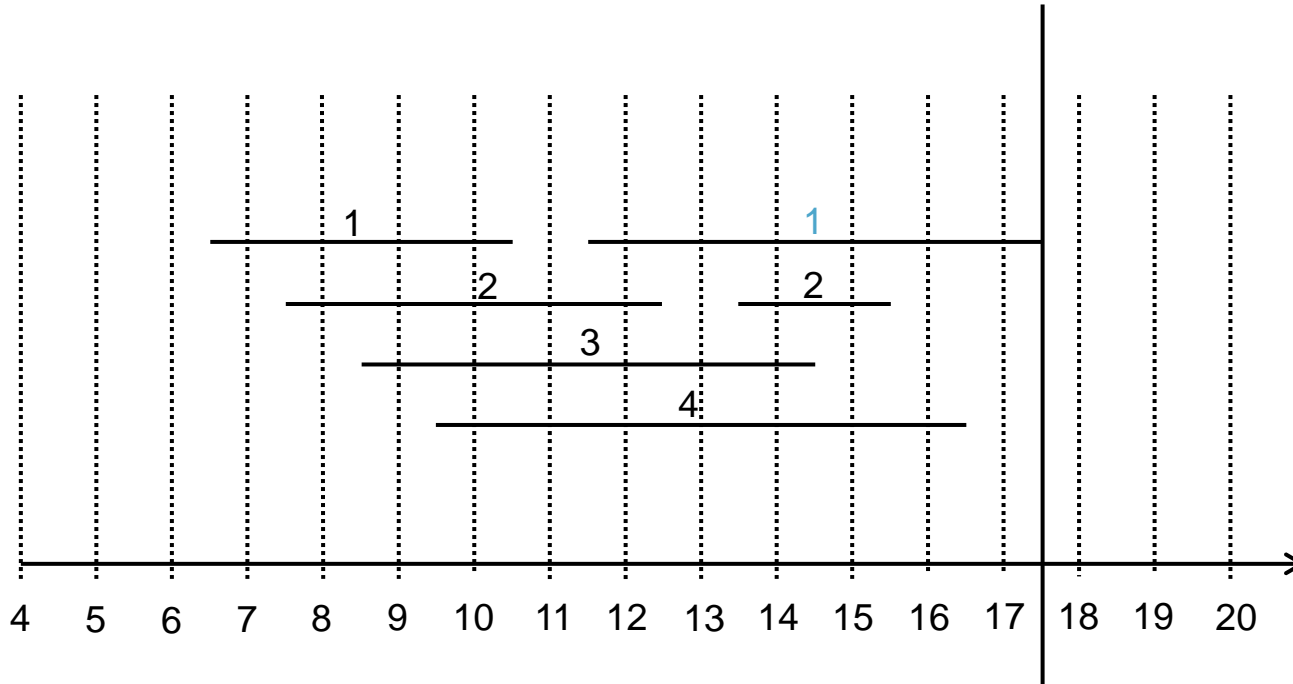


Wichtig ist nicht nur das Ende, sondern auch der Start eines Intervalls!

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!

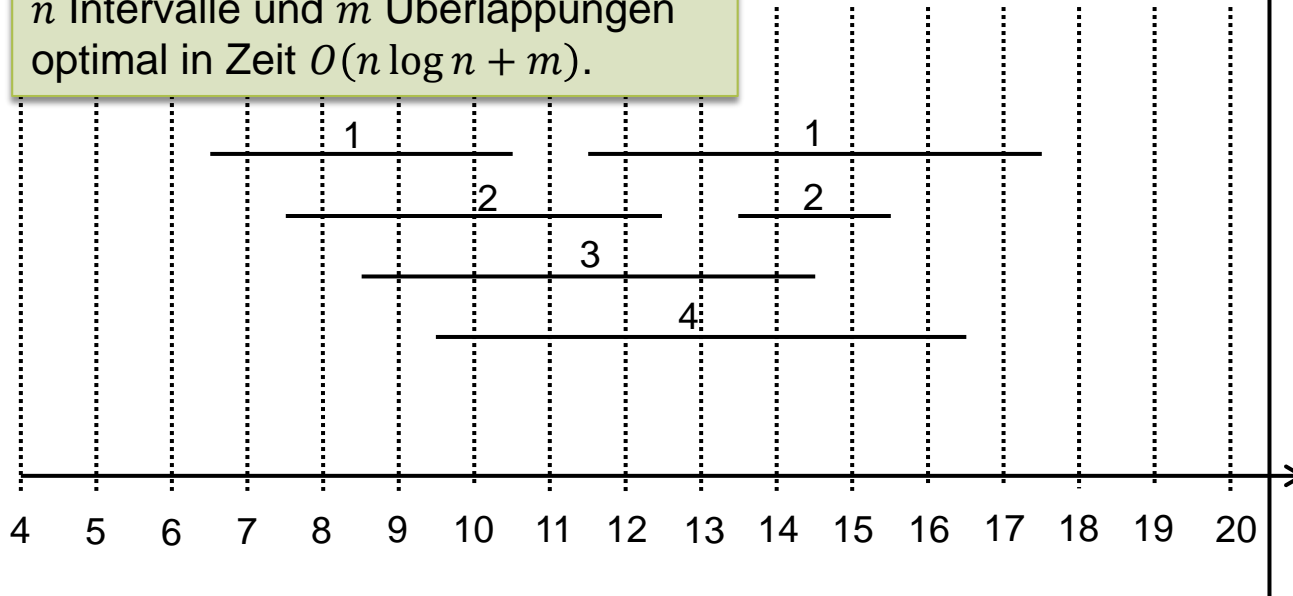
Bei Erreichen eines

- 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

- $\chi := \max_i (|\{I \in \mathcal{I} : i \in I\}|)$,
- 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.

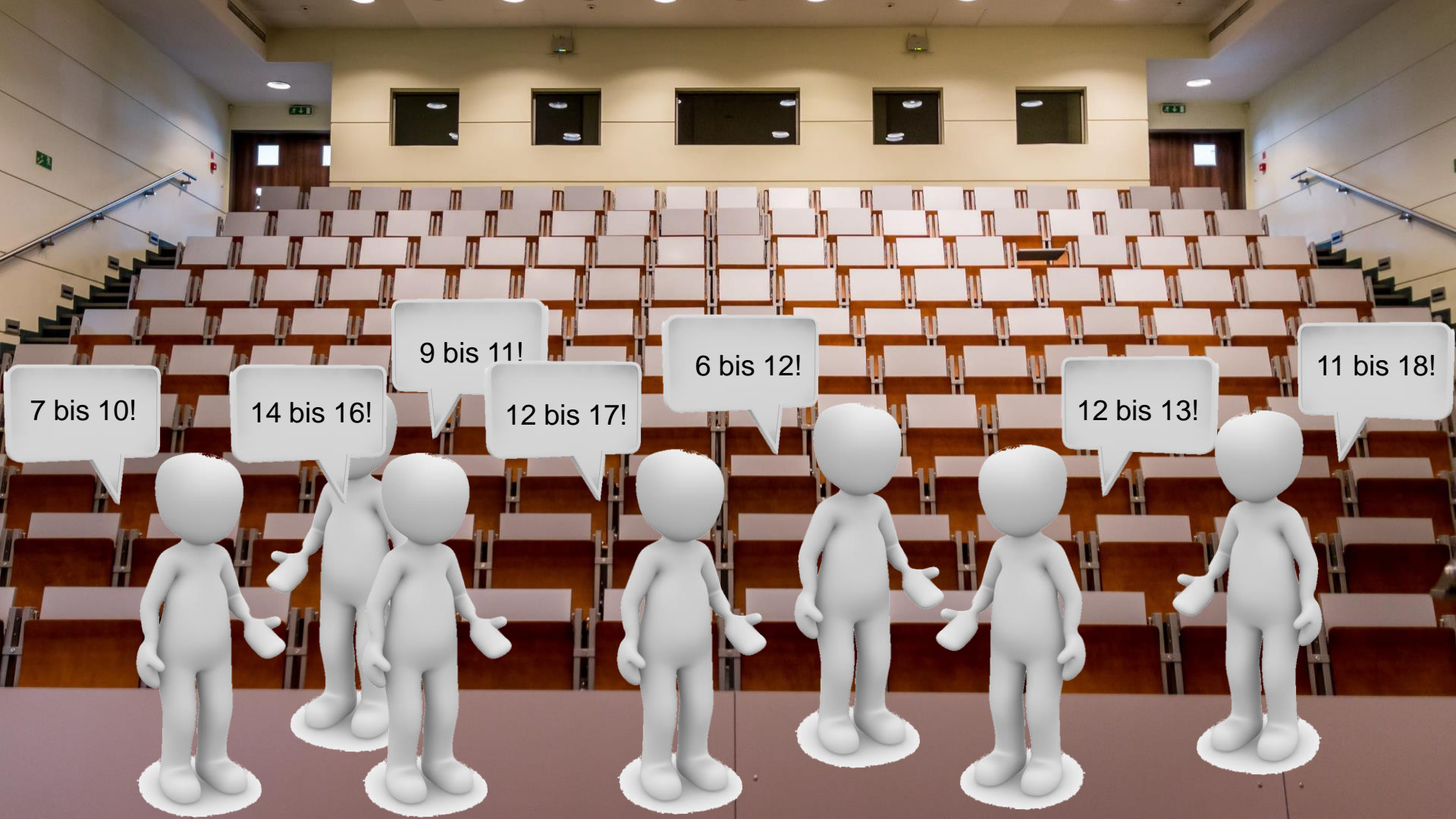
Fall 1: k wurde schon mal vergeben. Nach Annahme gilt weiterhin $alg \leq \chi$

Fall 2: k wurde noch nicht vergeben.

⇒ Zum Zeitpunkt s_{i+1} überlappen sich k Intervalle.

⇒ $\chi \geq k$

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



7 bis 10!

14 bis 16!

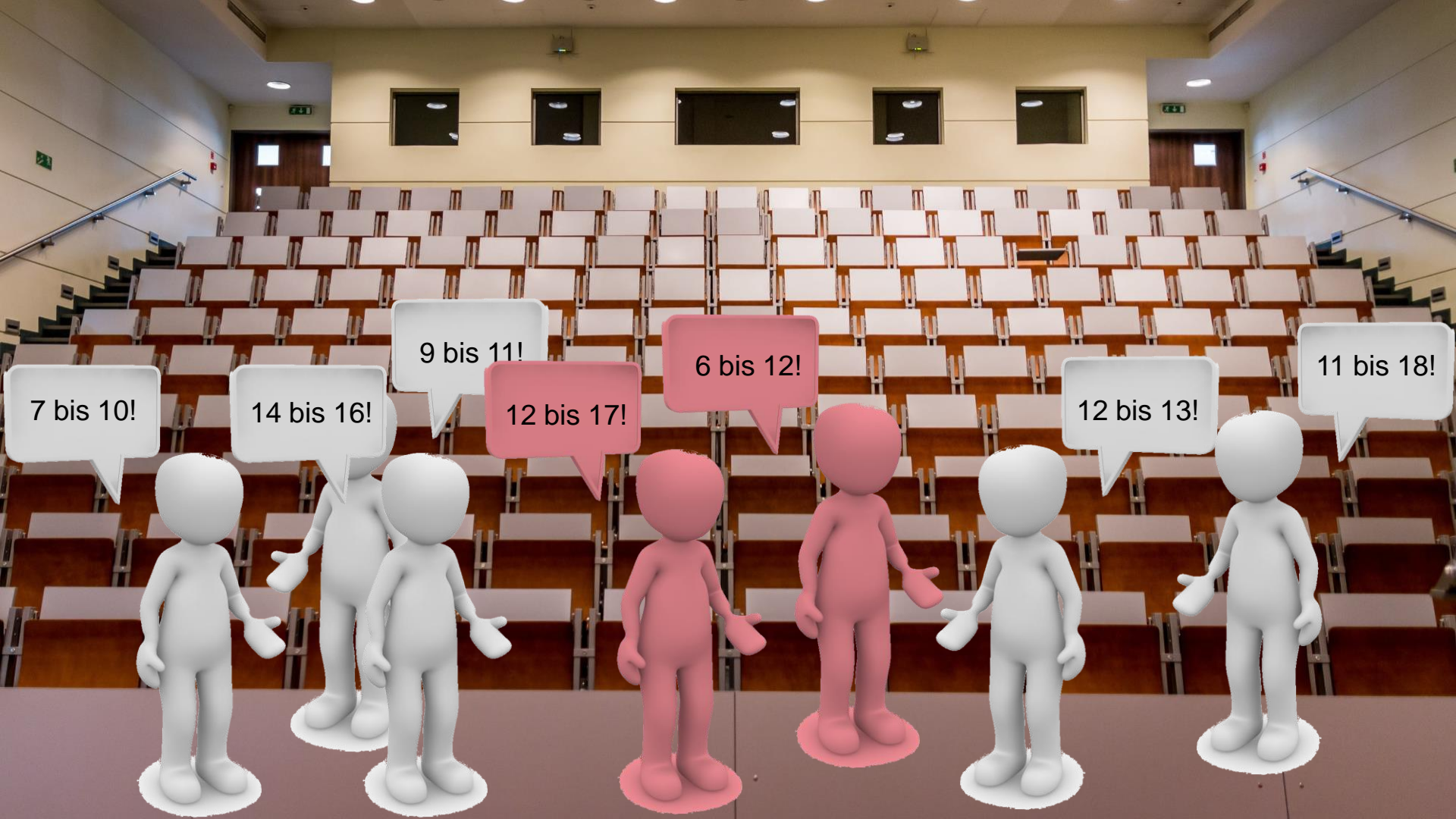
9 bis 11!

12 bis 17!

6 bis 12!

12 bis 13!

11 bis 18!



7 bis 10!

14 bis 16!

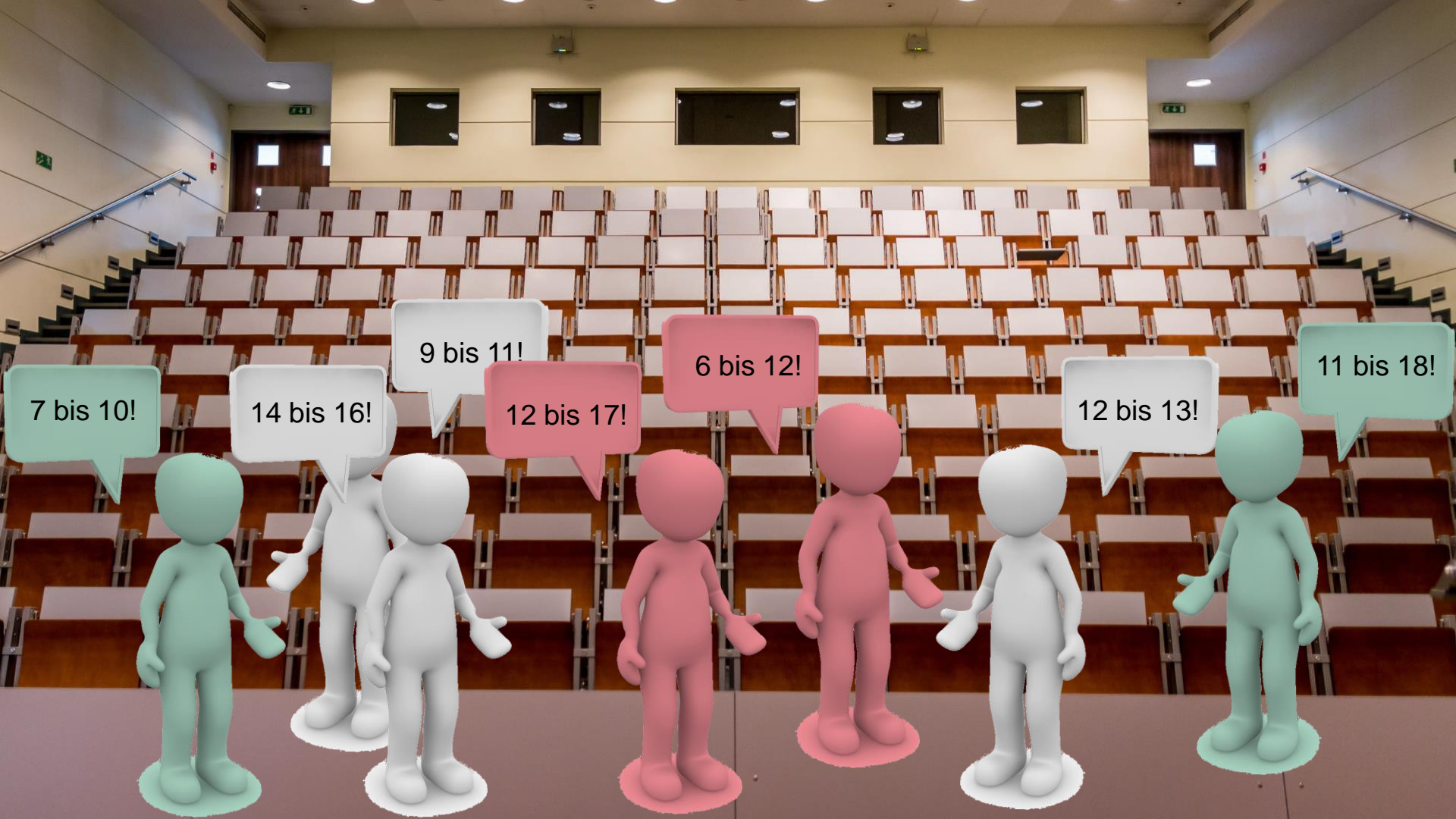
9 bis 11!

12 bis 17!

6 bis 12!

12 bis 13!

11 bis 18!



7 bis 10!

14 bis 16!

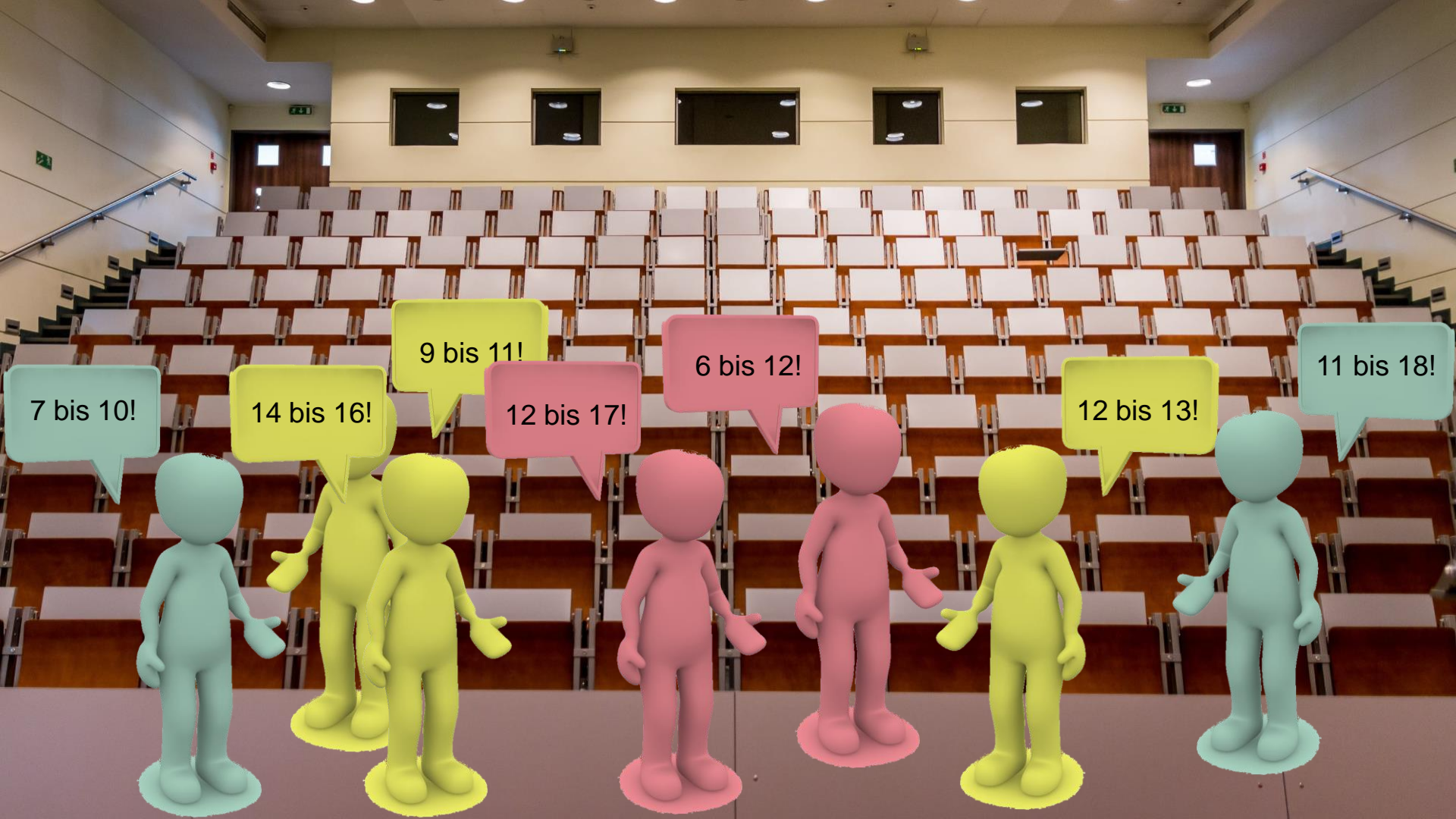
9 bis 11!

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!