Karlsruher Institut für Technologie

Algorithmen II
Übung am 21.01.2014

Approximation

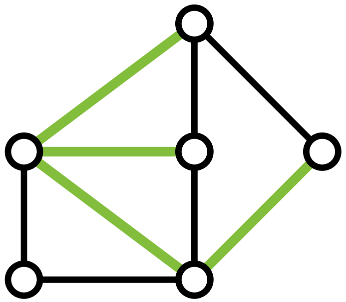
INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

Karlsruher Institut für Technologie

Minimale Schnittbasis

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

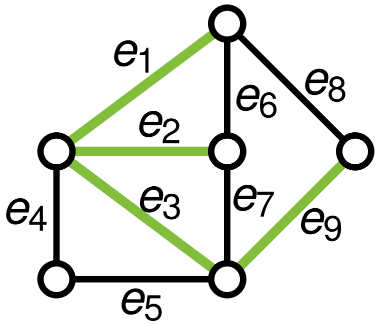
Der Schnittraum



Kantenmenge

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

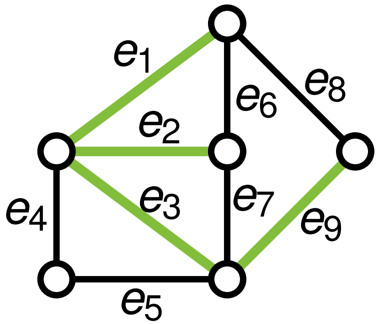
Der Schnittraum



Kantenmenge

in Vektordarstellung $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

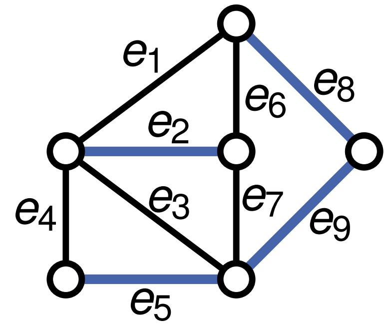
Der Schnittraum



Kantenmenge

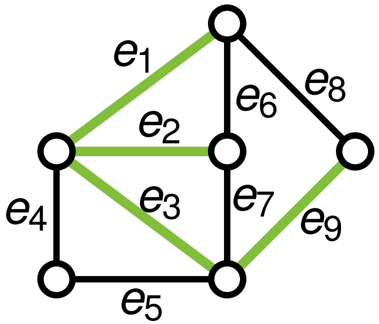
in Vektordarstellung

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$



symmetrische Differenz

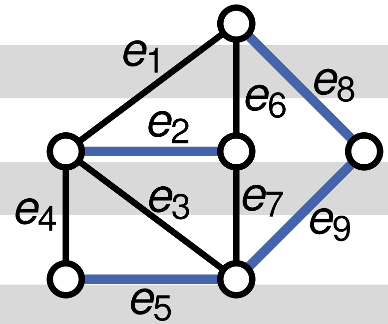
Der Schnittraum

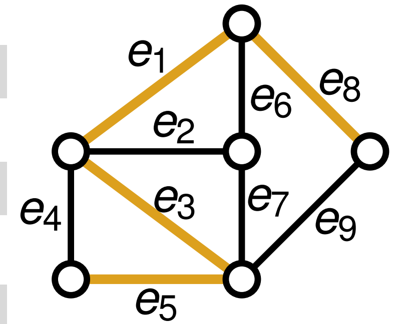


Kantenmenge

in Vektordarstellung

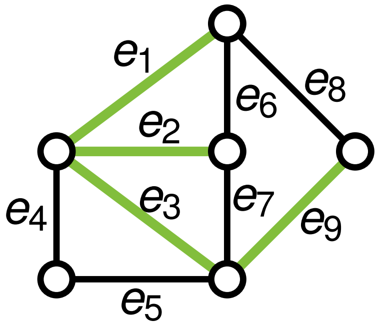
$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$





symmetrische Differenz

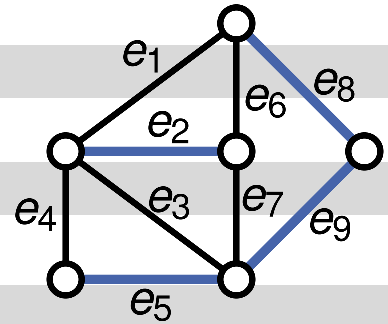
Der Schnittraum

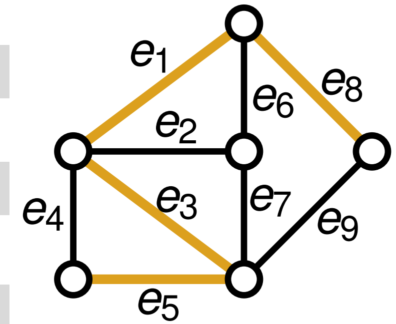


Kantenmenge

in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$

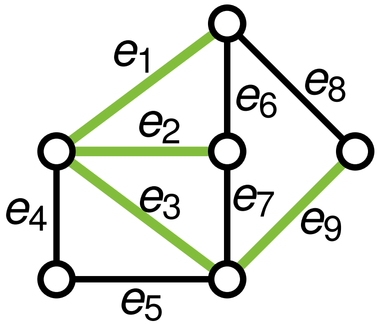




symmetrische Differenz

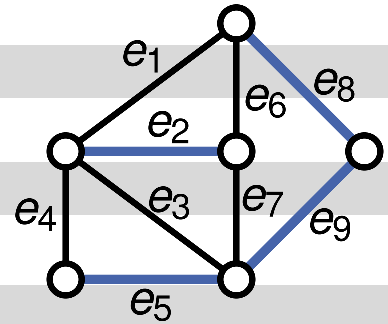
Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

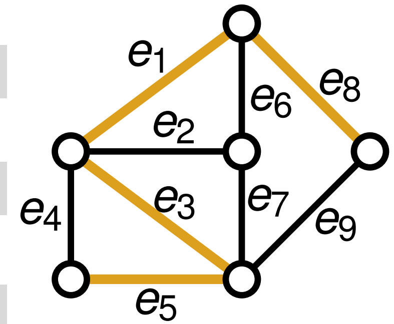
Der Schnittraum



Kantenmenge in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$



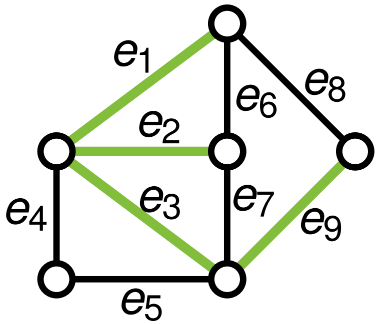


symmetrische Differenz

Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

- Aus der Vorlesung: Kreisraum
- heute: Schnittraum

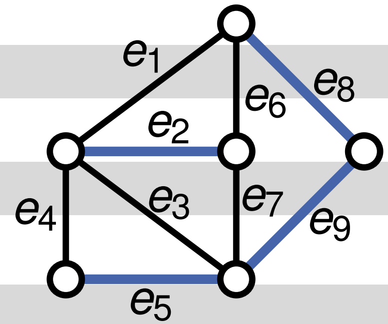
Der Schnittraum

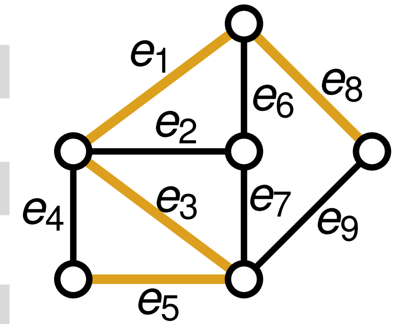


Kantenmenge

in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$

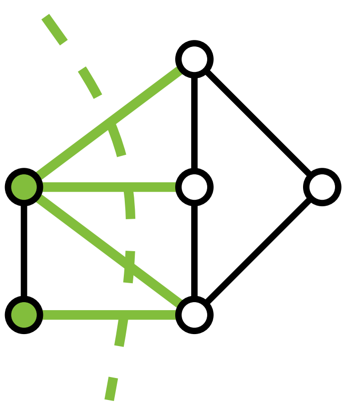




symmetrische Differenz

Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

- Aus der Vorlesung: Kreisraum
- heute: Schnittraum



Mögliche Darstellungen eines Schnitts:

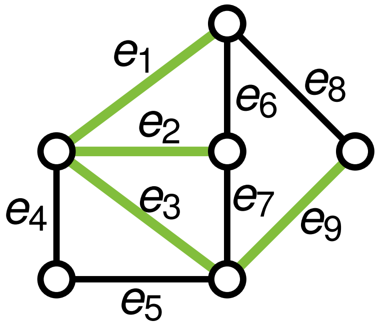
- Partition der Knoten in zwei Teilmengen S und $V \setminus S$.
- Menge der Kanten von S nach $V \setminus S$.

Achtung: Nicht jede Kantenmenge bildet Schnitt!

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Der Schnittraum

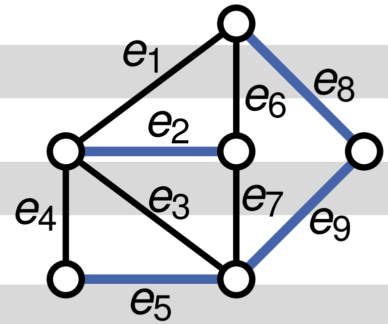
Karlsruher Institut für Technologie

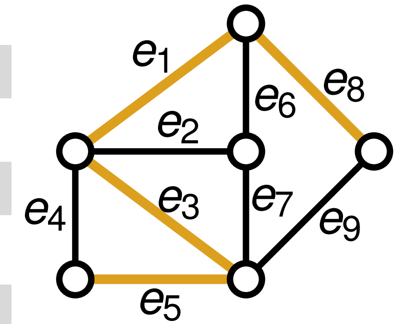


Kantenmenge

in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$

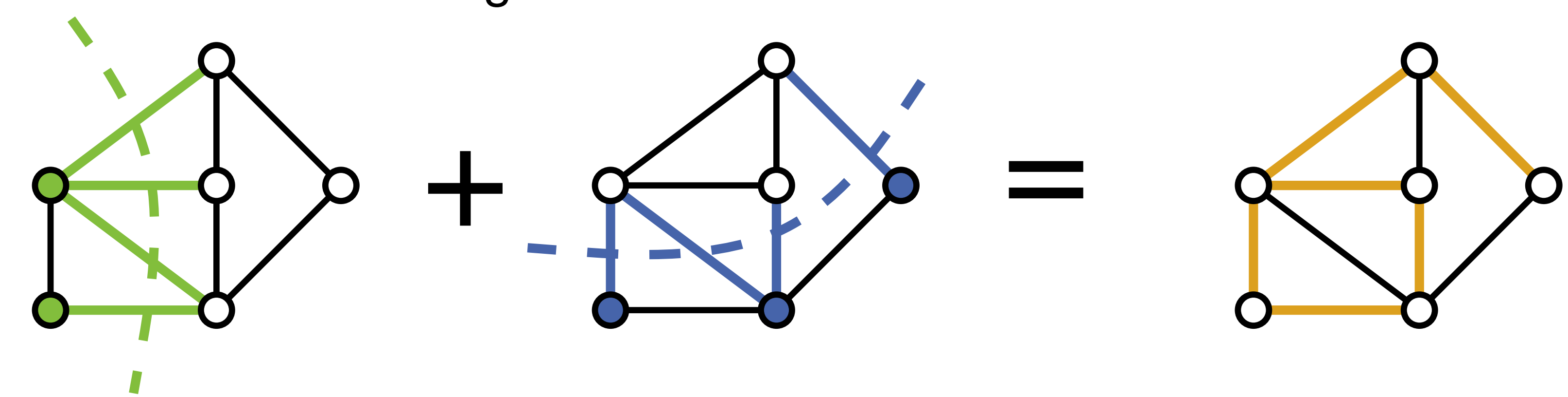




symmetrische Differenz

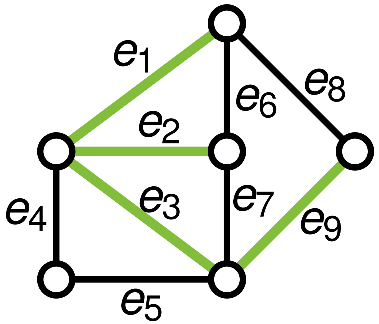
Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

- Aus der Vorlesung: Kreisraum
- heute: Schnittraum



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

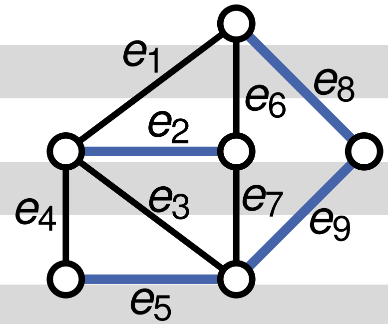
Der Schnittraum

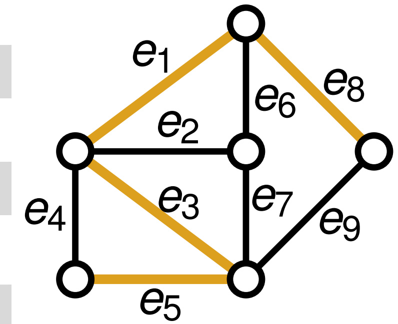


Kantenmenge

in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$

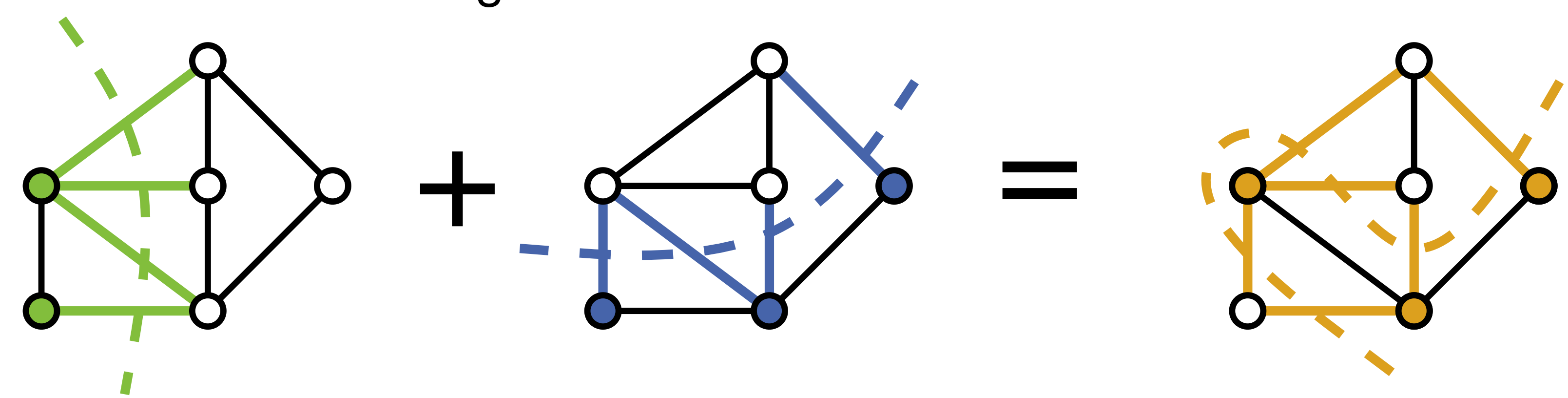




symmetrische Differenz

Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

- Aus der Vorlesung: Kreisraum
- heute: Schnittraum

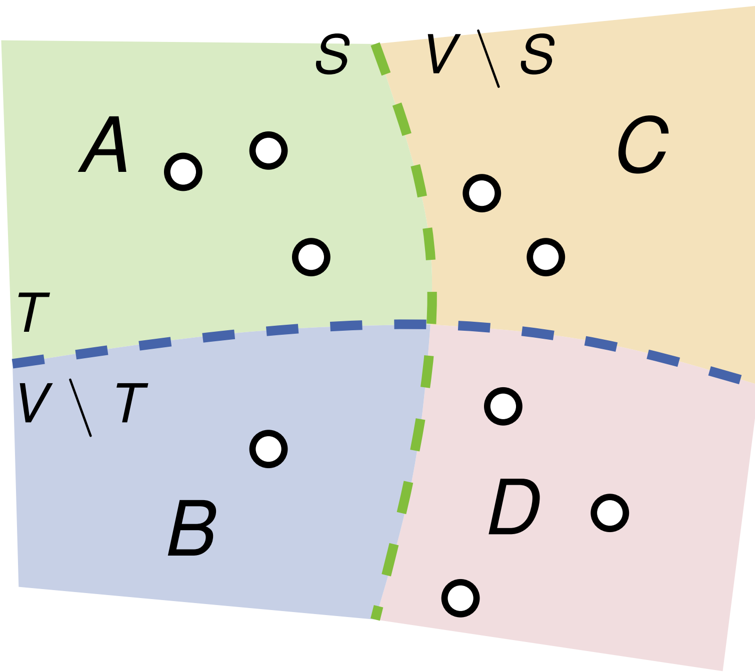


Problem 1 (a) Formulieren Sie die symmetrische Differenz in Partitionsdarstellung.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (a)

(a) Formulieren Sie die Partitionsdarstellung des Schnitts $s_3 = s_1 + s_2$ (mit $s_1 = (S, V \setminus S)$, $s_2 = (T, V \setminus T)$) in Abhängigkeit von S und T.

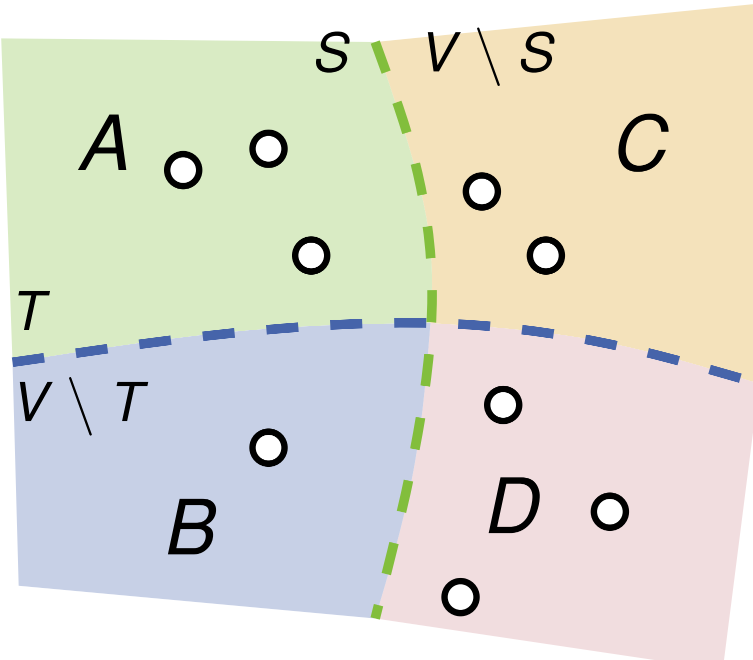


Interessante Knotenmengen:

- $A = S \cap T$
- $B = S \cap (V \setminus T)$
- $C = (V \setminus S) \cap T$
- $D = (V \setminus S) \cap (V \setminus T)$

Problem 1 (a)

(a) Formulieren Sie die Partitionsdarstellung des Schnitts $s_3 = s_1 + s_2$ (mit $s_1 = (S, V \setminus S)$, $s_2 = (T, V \setminus T)$) in Abhängigkeit von S und T.



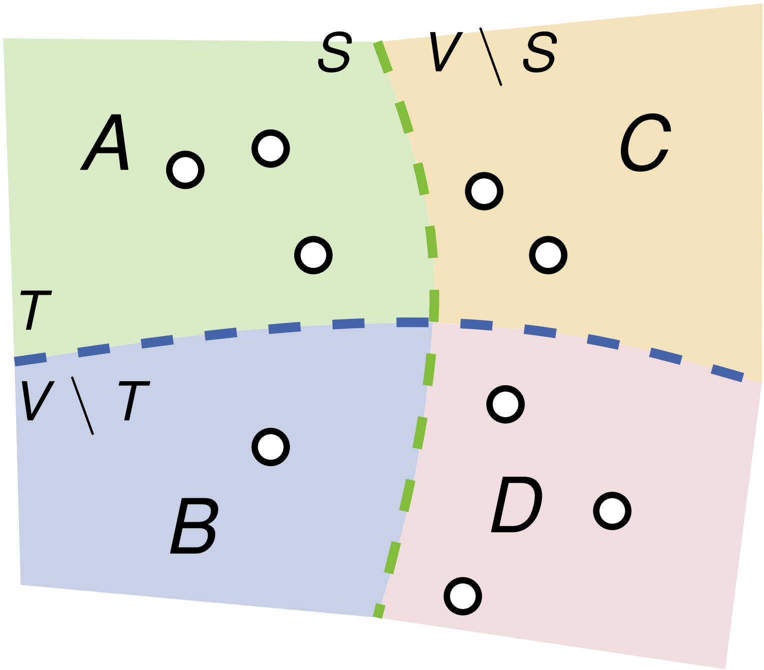
Interessante Knotenmengen:

- $A = S \cap T$
- $B = S \cap (V \setminus T)$
- $C = (V \setminus S) \cap T$
- $D = (V \setminus S) \cap (V \setminus T)$

Welche Kanten sind in s_3 enthalten?

Problem 1 (a)

(a) Formulieren Sie die Partitionsdarstellung des Schnitts $s_3 = s_1 + s_2$ (mit $s_1 = (S, V \setminus S)$, $s_2 = (T, V \setminus T)$) in Abhängigkeit von S und T.



Interessante Knotenmengen:

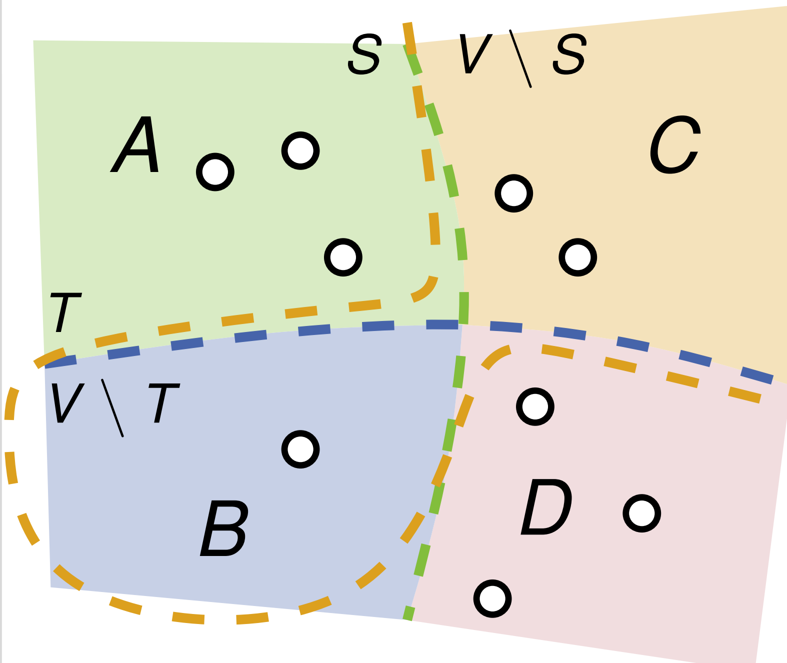
- $A = S \cap T$
- $B = S \cap (V \setminus T)$
- $C = (V \setminus S) \cap T$
- $D = (V \setminus S) \cap (V \setminus T)$

Welche Kanten sind in s_3 enthalten?

- Alle, die genau einen der beiden Schnitte kreuzen.

Problem 1 (a)

(a) Formulieren Sie die Partitionsdarstellung des Schnitts $s_3 = s_1 + s_2$ (mit $s_1 = (S, V \setminus S)$, $s_2 = (T, V \setminus T)$) in Abhängigkeit von S und T.



Interessante Knotenmengen:

- $A = S \cap T$
- $B = S \cap (V \setminus T)$
- $C = (V \setminus S) \cap T$
- $D = (V \setminus S) \cap (V \setminus T)$

Welche Kanten sind in s_3 enthalten?

- Alle, die genau einen der beiden Schnitte kreuzen.
- $\Rightarrow s_3 = (A \cup D, B \cup C)$

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \big(\{u\}, V \setminus \{u\}\big)$

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \big(\{u\}, V \setminus \{u\}\big)$

Behauptung:

Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \big(\{u\}, V \setminus \{u\}\big)$

Behauptung:

Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \left(\{u\}, V \setminus \{u\}\right)$

Behauptung:

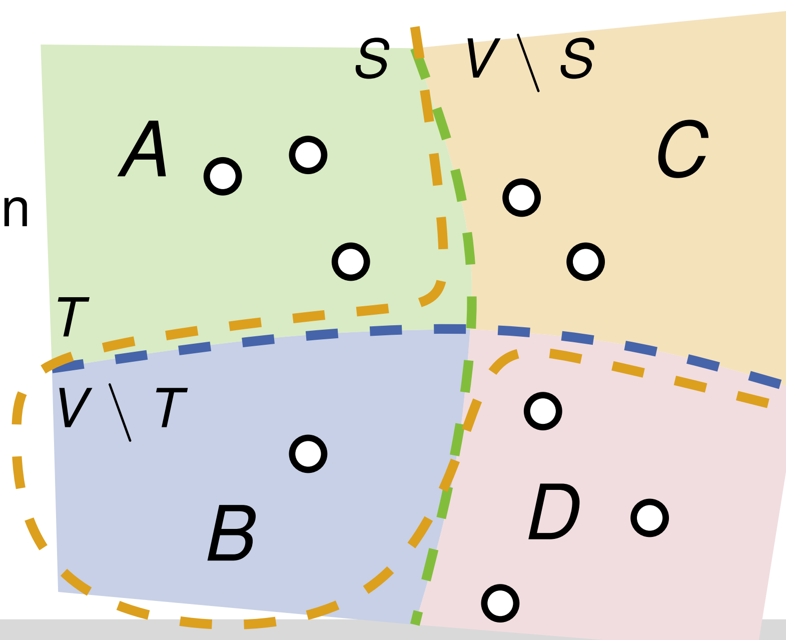
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \big(\{u\}, V \setminus \{u\}\big)$

Behauptung:

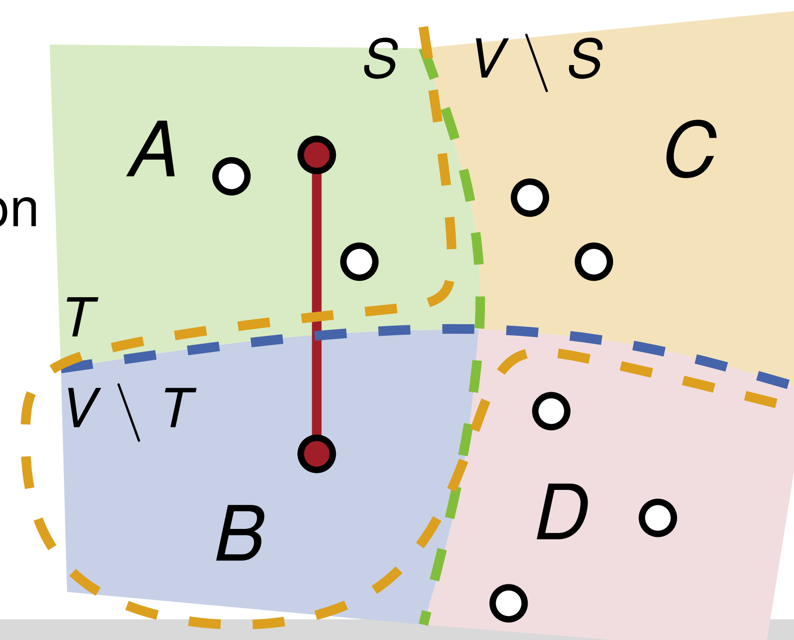
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \left(\{u\}, V \setminus \{u\}\right)$

Behauptung:

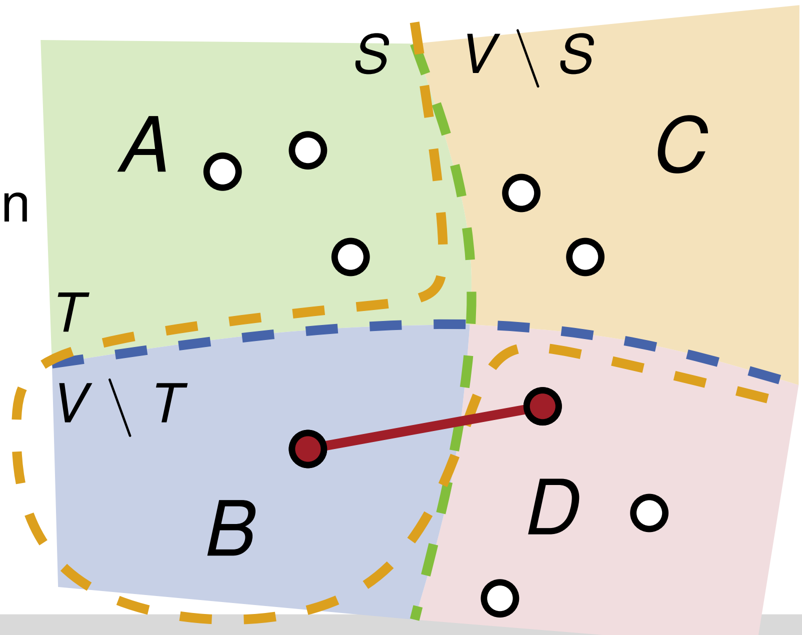
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \left(\{u\}, V \setminus \{u\}\right)$

Behauptung:

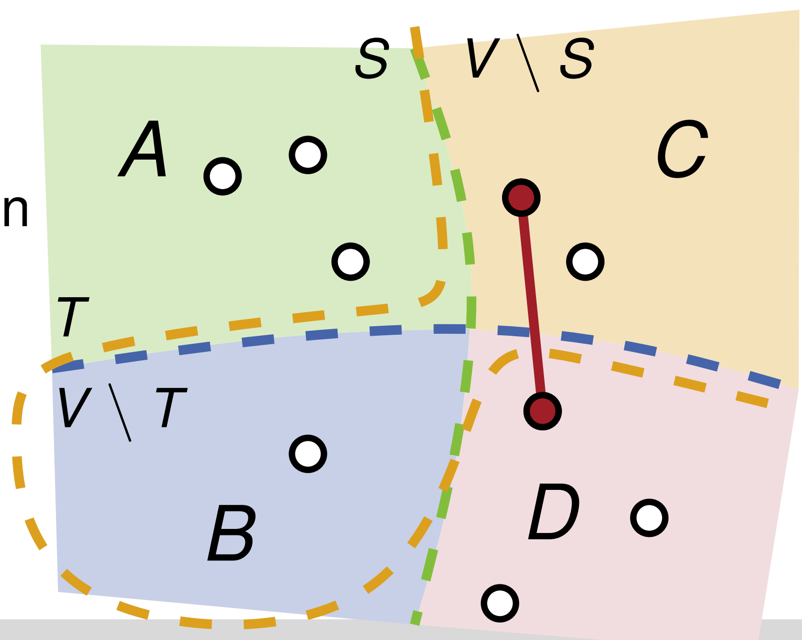
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \left(\{u\}, V \setminus \{u\}\right)$

Behauptung:

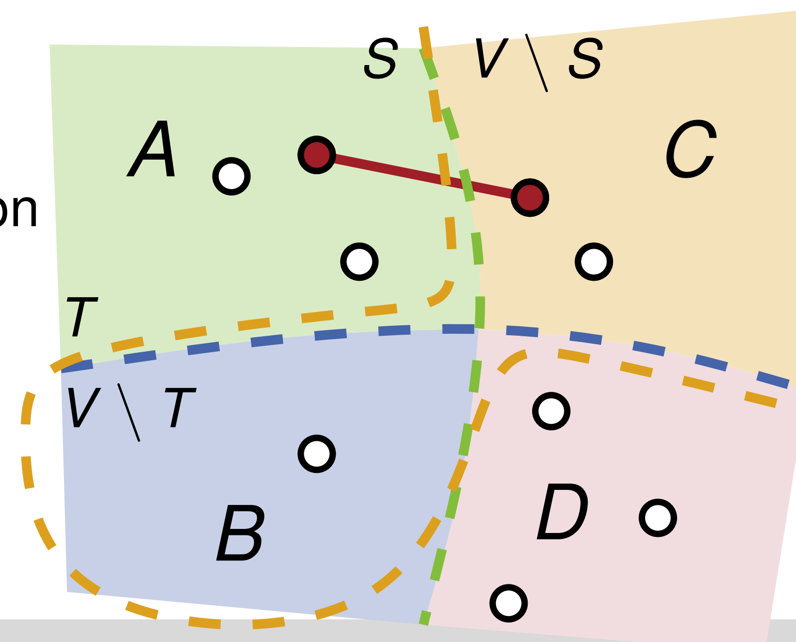
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Problem 1 (c)

Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen

Problem 1 (c)

Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen
- Gewicht der Basis B: $c(B) = \sum_{i=1}^{d} c(b_i)$

Problem 1 (c)

Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen
- Gewicht der Basis B: $c(B) = \sum_{i=1}^{d} c(b_i)$

Problem: MIN-SCHNITT-BASIS
Finde eine Basis des Schnittraums von G mit minimalem Gewicht.

Problem 1 (c)

Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen
- Gewicht der Basis B: $c(B) = \sum_{i=1}^{d} c(b_i)$

Problem: MIN-SCHNITT-BASIS
Finde eine Basis des Schnittraums von G mit minimalem Gewicht.

APPROX-MIN-SCHNITT-BASIS(G)

Wähle einen Knoten $v \in V$
$B' \leftarrow \emptyset$
forall the $v' \in V \setminus \{v\}$ **do**
    $b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$
    $B' \leftarrow B' \cup \{b_{v'}\}$
Return B'

(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Problem 1 (c)

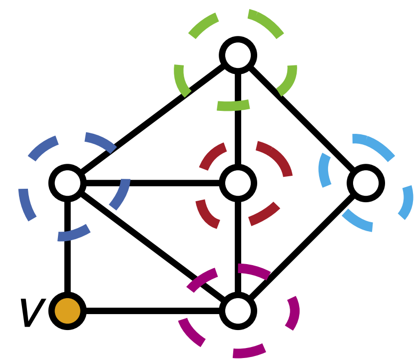
Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen
- Gewicht der Basis B: $c(B) = \sum_{i=1}^{d} c(b_i)$

Problem: MIN-SCHNITT-BASIS
Finde eine Basis des Schnittraums von G mit minimalem Gewicht.

APPROX-MIN-SCHNITT-BASIS(G)

Wähle einen Knoten $v \in V$
$B' \leftarrow \emptyset$
forall the $v' \in V \setminus \{v\}$ **do**
    $b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$
    $B' \leftarrow B' \cup \{b_{v'}\}$
Return B'



(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Problem 1 (c)

APPROX-MIN-SCHNITT-BASIS(G)

Wähle einen Knoten $v \in V$

$B' \leftarrow \emptyset$

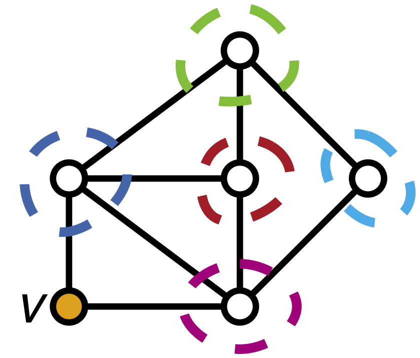
forall the $v' \in V \setminus \{v\}$ **do**

$b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$

$B' \leftarrow B' \cup \{b_{v'}\}$

Return B'

Wie groß ist $c(B')$**?**



(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Problem 1 (c)

$\textsc{Approx-Min-Schnitt-Basis}(G)$

Wähle einen Knoten $v \in V$

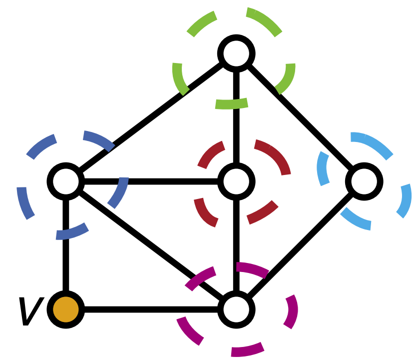
$B' \leftarrow \emptyset$

forall the $v' \in V \setminus \{v\}$ **do**

$b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$

$B' \leftarrow B' \cup \{b_{v'}\}$

Return B'



(c) Zeigen Sie: $\textsc{Approx-Min-Schnitt-Basis}$ ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Wie groß ist $c(B')$?

- Kanten inzident zu v sind in einem Schnitt enthalten.
- Alle anderen Kanten in zwei Schnitten.

$\Rightarrow c(B') = 2m - \deg(v) \leq 2m$ $\quad (m = |E|)$

Problem 1 (c)

APPROX-MIN-SCHNITT-BASIS(G)

Wähle einen Knoten $v \in V$

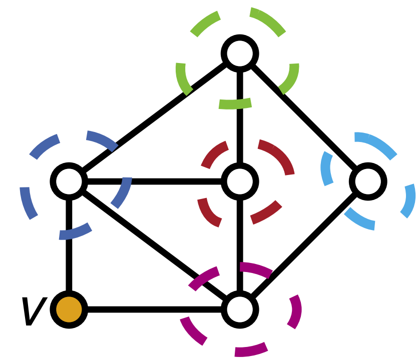
$B' \leftarrow \emptyset$

forall the $v' \in V \setminus \{v\}$ **do**

  $b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$

  $B' \leftarrow B' \cup \{b_{v'}\}$

Return B'



(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Wie groß ist $c(B')$?

- Kanten inzident zu v sind in einem Schnitt enthalten.
- Alle anderen Kanten in zwei Schnitten.

$$\Rightarrow c(B') = 2m - \deg(v) \leq 2m \qquad (m = |E|)$$

Wie groß ist $c(B)$ für eine minimale Basis B?

Problem 1 (c)

APPROX-MIN-SCHNITT-BASIS(G)

Wähle einen Knoten $v \in V$

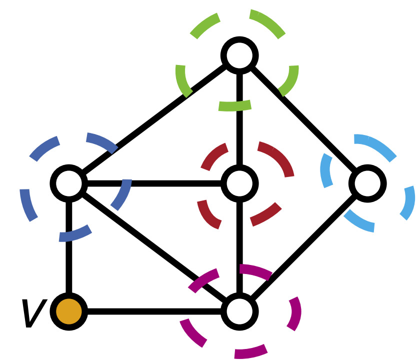
$B' \leftarrow \emptyset$

forall the $v' \in V \setminus \{v\}$ **do**

  $b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$

  $B' \leftarrow B' \cup \{b_{v'}\}$

Return B'



(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Wie groß ist $c(B')$?

- Kanten inzident zu v sind in einem Schnitt enthalten.
- Alle anderen Kanten in zwei Schnitten.

$\Rightarrow c(B') = 2m - \deg(v) \leq 2m$ $\quad (m = |E|)$

Wie groß ist $c(B)$ für eine minimale Basis B?

- Betrachte beliebige Kante $\{u, w\}$.
- Wegen (b) gilt: in B gibt es einen Schnitt, der u und w trennt.
- Jede Kante ist in mindestens einem Schnitt in B enthalten.

$\Rightarrow c(B) \geq m \Rightarrow$ APPROX-MIN-SCHNITT-BASIS hat Gütegarantie 2.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Organisatorisches

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Organisatorisches

Klausuren

Hauptklausur:

- Montag 24.02.2014 um 14:00 Uhr
- Bearbeitungszeit: 2 Stunden
- Die Anmeldung ist bis 17.02.2014 möglich.
  - Anmeldung erfolgt über das Studienportal.
  - Erasmusstudenten: Anmeldung im Studienbüro. Der (blaue?) Zettel muss bei uns abgegeben werden (ebenfalls bis 17.02.2014).
- Hörsaalbelegung wird auf der Homepage bekannt gegeben (natürlich erst nach dem Ende des Anmeldezeitraums).

Nachklausur:

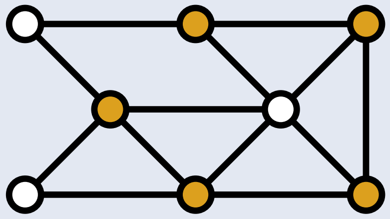
- Termin steht noch nicht fest (nach dem kommenden Sommersemester).
- Weitere Informationen gibt es zu gegebener Zeit auf der Homepage.

VERTEX COVER

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 2

Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)

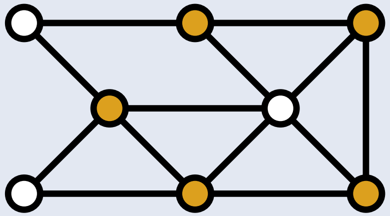


Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 2

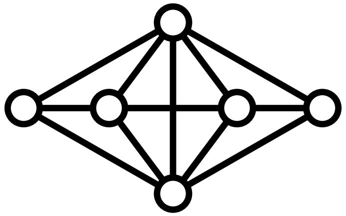
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

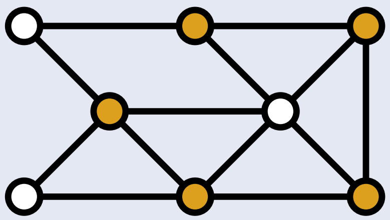
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

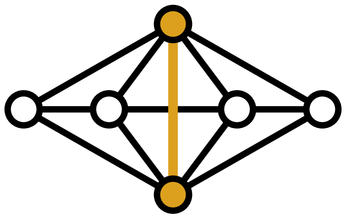
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$. (ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

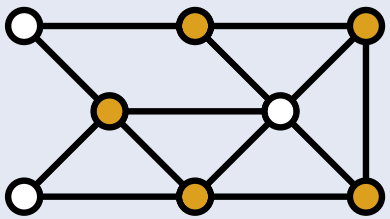
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

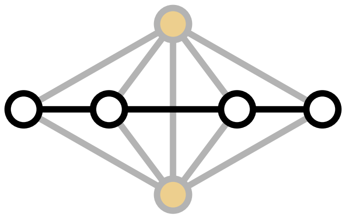
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$. (ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.

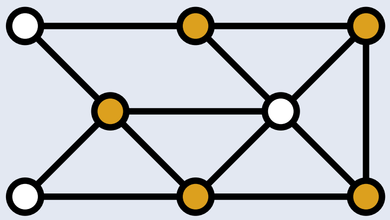


Problem 2

Problem: VERTEX COVER

Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.

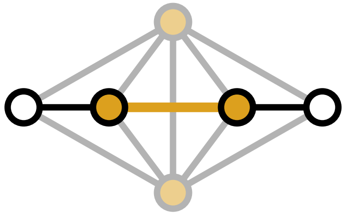
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

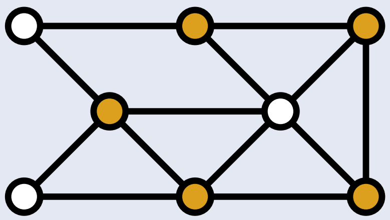
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

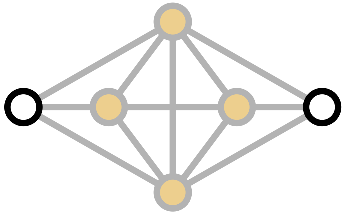
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

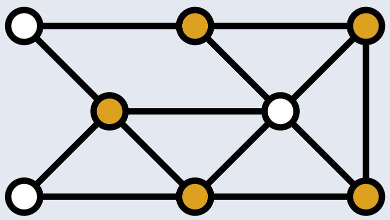
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

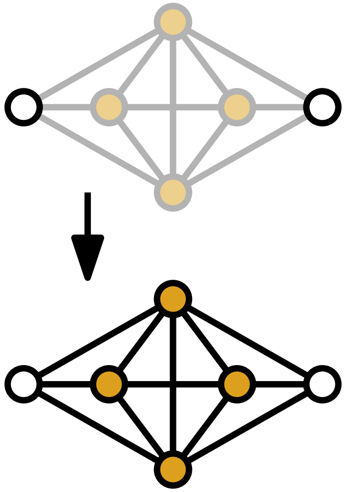
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

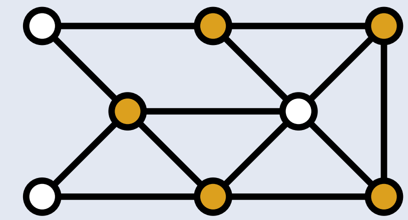
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

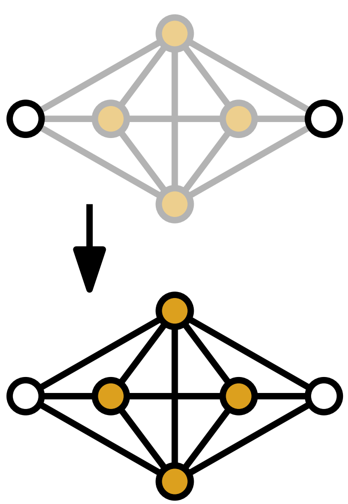
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

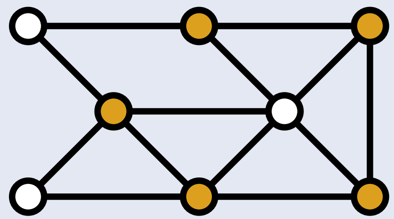
- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Approximationsgüte

Problem 2

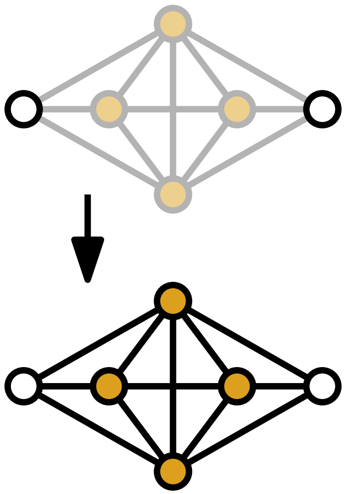
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$. (ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.

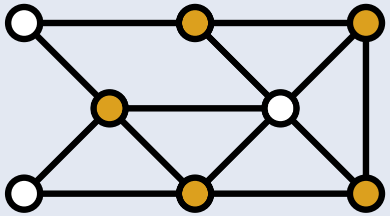


Approximationsgüte

- Seien $e_1, \ldots, e_k$ die vom Algorithmus ausgewählten Kanten.

Problem 2

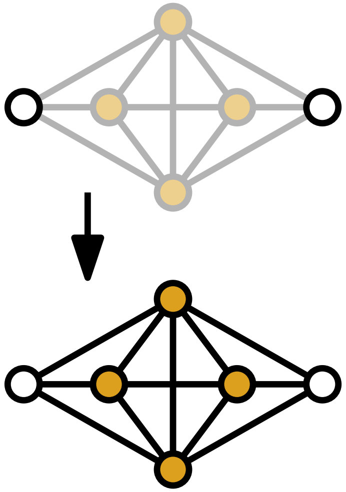
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



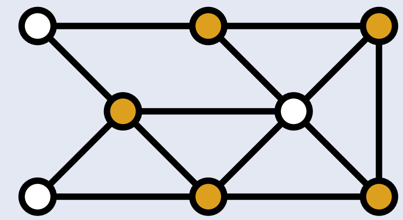
Approximationsgüte

- Seien $e_1, \ldots, e_k$ die vom Algorithmus ausgewählten Kanten.
- Der Algorithmus liefert ein Vertex Cover der Größe $2k$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 2

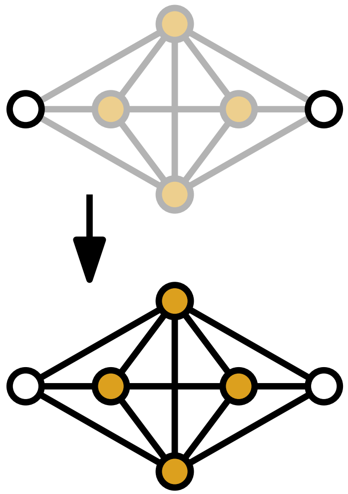
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$. (ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge *u* **und** *v* zum Vertex Cover hinzu und lösche sie aus dem Graph.



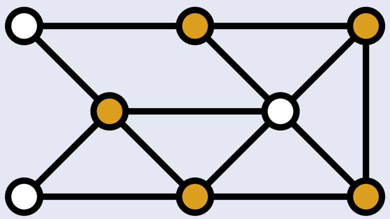
Approximationsgüte

- Seien $e_1, \dots, e_k$ die vom Algorithmus ausgewählten Kanten.
- Der Algorithmus liefert ein Vertex Cover der Größe $2k$.
- Jedes Vertex Cover enthält mindestens k Knoten, da es keinen Knoten gibt, der mehr als eine der Kanten $e_1, \dots, e_k$ abdeckt.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 2

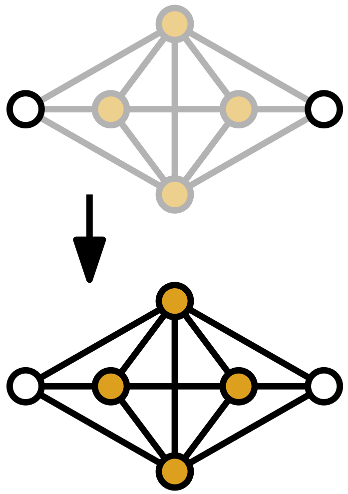
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Approximationsgüte

- Seien $e_1, \ldots, e_k$ die vom Algorithmus ausgewählten Kanten.
- Der Algorithmus liefert ein Vertex Cover der Größe $2k$.
- Jedes Vertex Cover enthält mindestens k Knoten, da es keinen Knoten gibt, der mehr als eine der Kanten $e_1, \ldots, e_k$ abdeckt.

$\Rightarrow$ Approximationsgüte 2

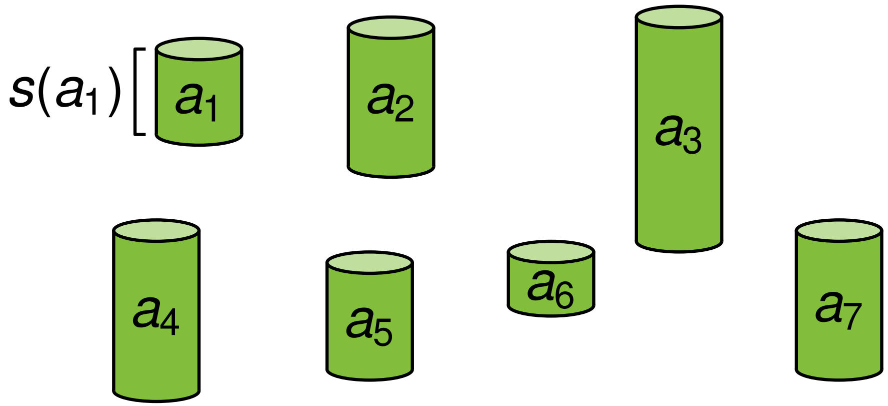
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

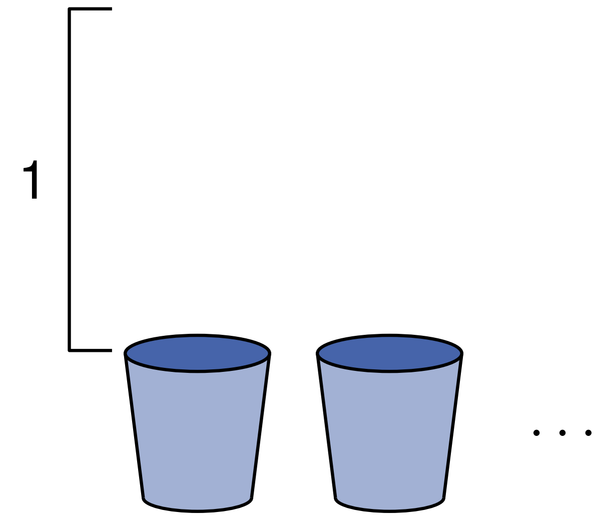
APAS für BIN PACKING

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Wiederholung: Bin Packing – Definition

endliche Menge $M = \{a_1, \ldots, a_n\}$
mit Gewichtsfunktion $s\colon M \longrightarrow (0, 1]$





Eimer (Bins) mit Fassungsvermögen 1

Schreibe für $s(a_i)$ auch kurz s_i.

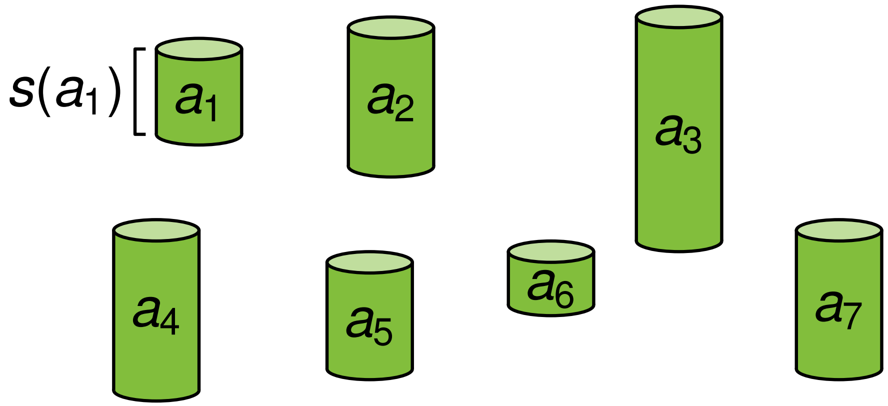
Problem: BIN PACKING
Weise die Elemente in M einer minimalen Anzahl an Bins $B_1, \ldots, B_m$ zu, sodass für jeden Bin B gilt:

$$\sum_{a_i \in B} s(a_i) \leq 1$$

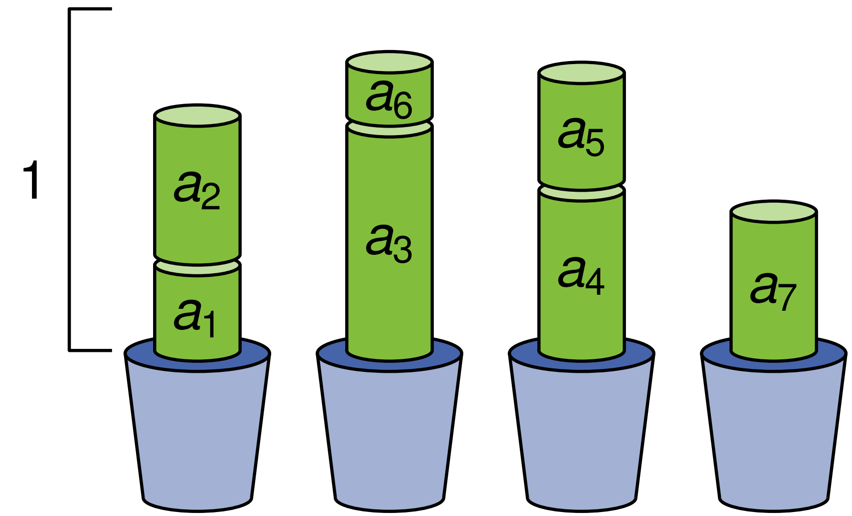
BIN PACKING ist $\mathcal{NP}$-schwer.

Wiederholung: Bin Packing – Definition

endliche Menge $M = \{a_1, \ldots, a_n\}$
mit Gewichtsfunktion $s\colon M \longrightarrow (0, 1]$



Schreibe für $s(a_i)$ auch kurz s_i.



Eimer (Bins) mit Fassungsvermögen 1

4 Bins

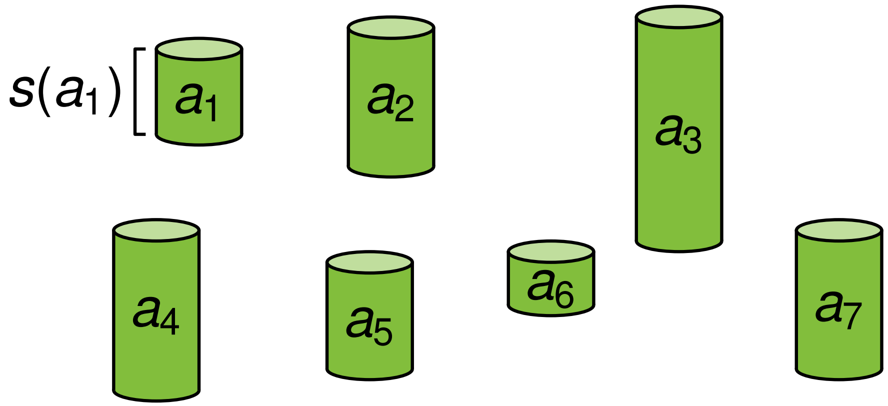
Problem: BIN PACKING
Weise die Elemente in M einer minimalen Anzahl an Bins $B_1, \ldots, B_m$ zu, sodass für jeden Bin B gilt:

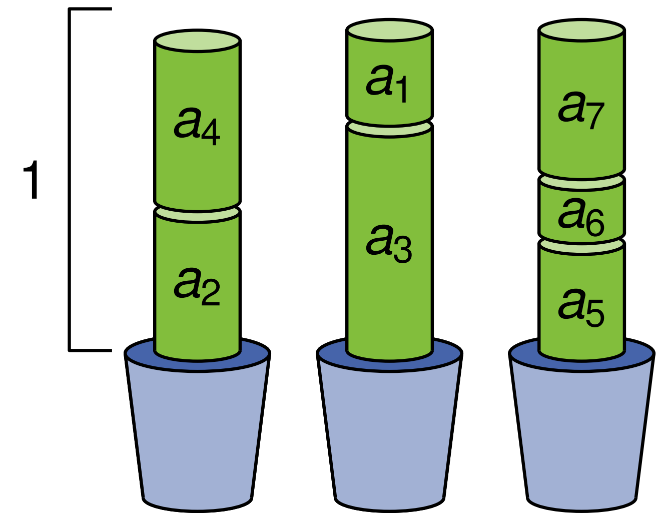
$$\sum_{a_i \in B} s(a_i) \leq 1$$

BIN PACKING ist $\mathcal{NP}$-schwer.

Wiederholung: Bin Packing – Definition

endliche Menge $M = \{a_1, \ldots, a_n\}$
mit Gewichtsfunktion $s\colon M \longrightarrow (0, 1]$





Schreibe für $s(a_i)$ auch kurz s_i.

Eimer (Bins) mit Fassungsvermögen 1

3 Bins

Problem: BIN PACKING
Weise die Elemente in M einer minimalen Anzahl an Bins $B_1, \ldots, B_m$ zu, sodass für jeden Bin B gilt:

$$\sum_{a_i \in B} s(a_i) \leq 1$$

BIN PACKING ist $\mathcal{NP}$-schwer.

Ist das Problem BIN PACKING überhaupt interessant?

Ist das Problem Bin Packing überhaupt interessant?

Mögliche Anwendungen:

- Wie viele CDs braucht man für die gesamte Musiksammlung?
- Wie viele Werbeblöcke (maximal 5 Minuten) braucht man, um alle Spots unterzubringen?
- Wie viele Regale braucht man für alle Bücher?
- Wie viele Bildschirme braucht man, um eine Menge von Fenstern gleichzeitig betrachten zu können?
- Wie oft muss man bei einem Umzug von der alten zur neuen Wohnung fahren?
- Wie viele Container braucht man für eine Warenlieferung?
- Wie viele Schiffe braucht man, um eine Menge Container zu transportieren?

Ist das Problem Bin Packing überhaupt interessant?

Mögliche Anwendungen:

- Wie viele CDs braucht man für die gesamte Musiksammlung?
- Wie viele Werbeblöcke (maximal 5 Minuten) braucht man, um alle Spots unterzubringen?
- Wie viele Regale braucht man für alle Bücher?

1-Dimensional

- Wie viele Bildschirme braucht man, um eine Menge von Fenstern gleichzeitig betrachten zu können?

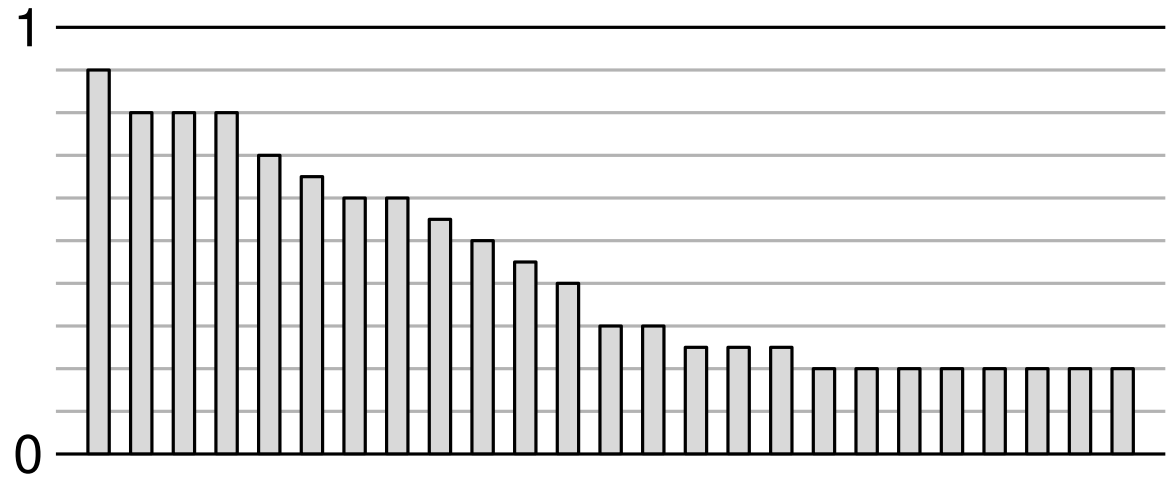
2-Dimensional

- Wie oft muss man bei einem Umzug von der alten zur neuen Wohnung fahren?
- Wie viele Container braucht man für eine Warenlieferung?
- Wie viele Schiffe braucht man, um eine Menge Container zu transportieren?

3-Dimensional

Problem 3

Führe das APAS für Bin Packing mit $\varepsilon = 0.6$ an einem Beispiel aus.



(a) Berechne die zu ε gehörenden Parameter δ, k und m. Betrachte anschließend die Instanz J bestehend aus allen Elementen mit Gewicht größer oder gleich δ.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

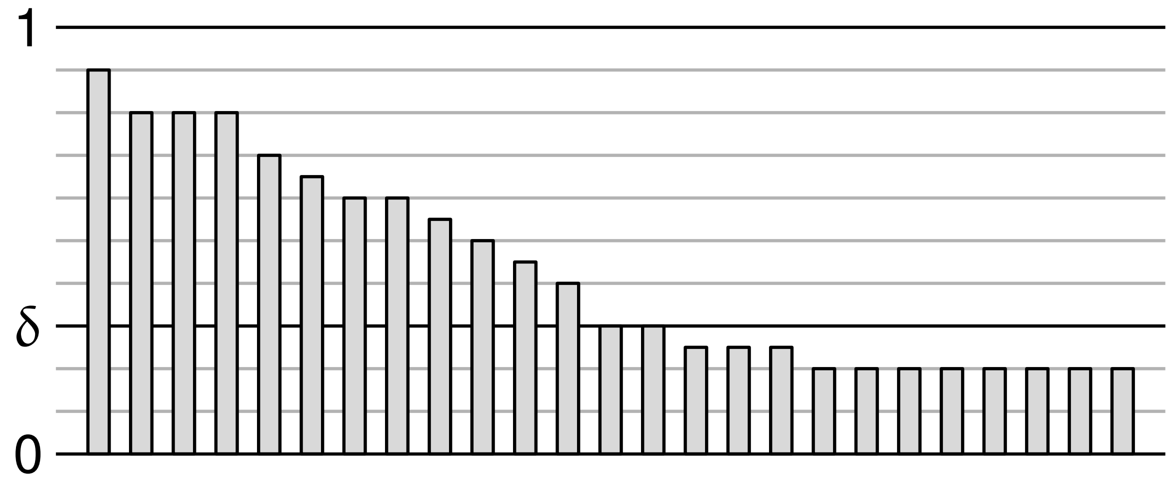
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$\delta = \frac{\varepsilon}{2} = 0.3$

$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil$

$m = \left\lfloor \frac{n'}{k} \right\rfloor$



(a) Berechne die zu ε gehörenden Parameter δ, k und m. Betrachte anschließend die Instanz J bestehend aus allen Elementen mit Gewicht größer oder gleich δ.

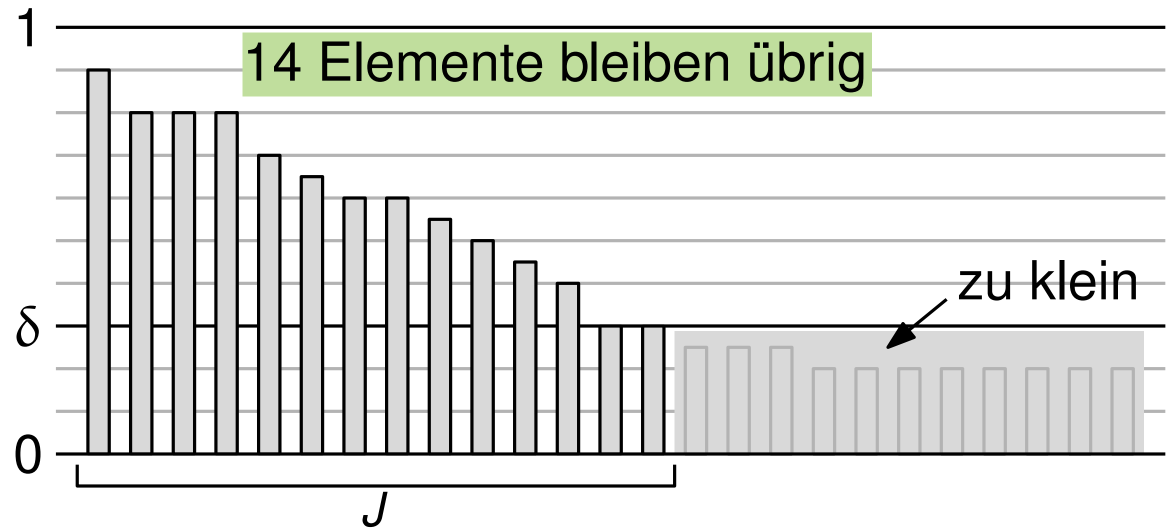
Problem 3

Führe das APAS für Bin Packing mit $\varepsilon = 0.6$ an einem Beispiel aus.

$\delta = \frac{\varepsilon}{2} = 0.3$

$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil$

$m = \left\lfloor \frac{n'}{k} \right\rfloor$



(a) Berechne die zu ε gehörenden Parameter δ, k und m. Betrachte anschließend die Instanz J bestehend aus allen Elementen mit Gewicht größer oder gleich δ.

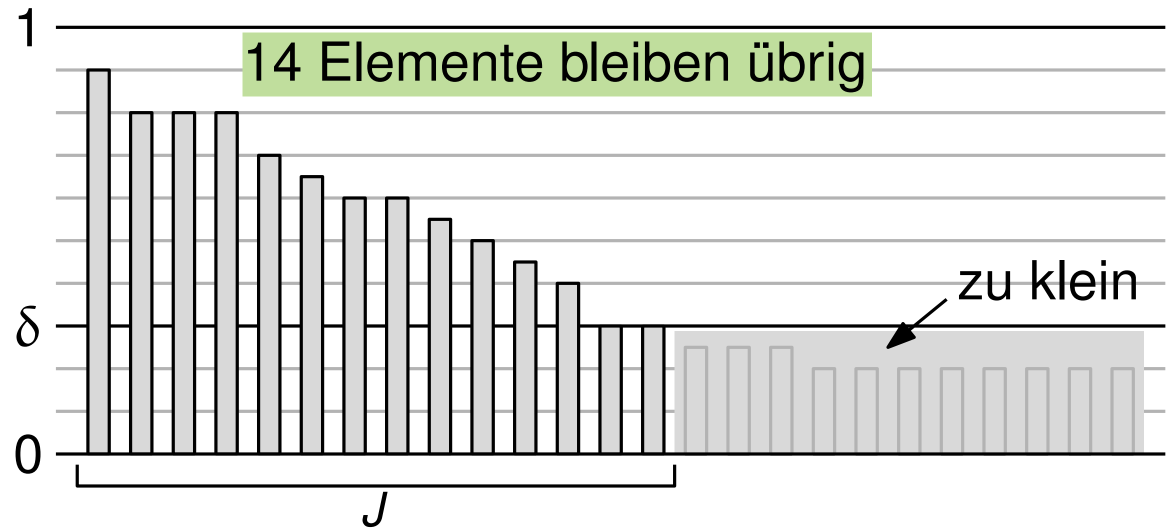
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(a) Berechne die zu ε gehörenden Parameter δ, k und m. Betrachte anschließend die Instanz J bestehend aus allen Elementen mit Gewicht größer oder gleich δ.

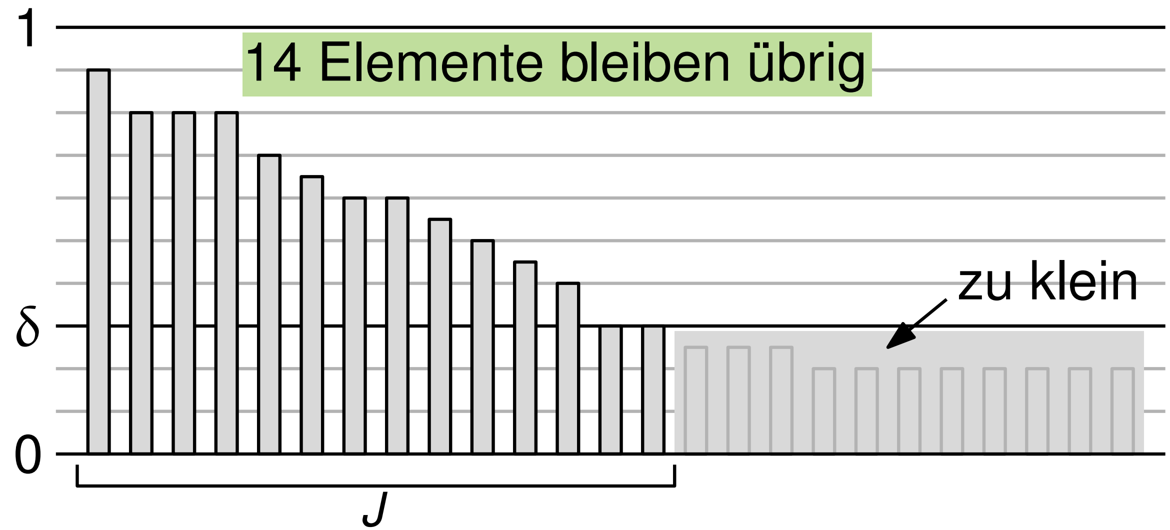
Problem 3

Führe das APAS für Bin Packing mit $\varepsilon = 0.6$ an einem Beispiel aus.

$\delta = \frac{\varepsilon}{2} = 0.3$

$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$

$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$



(b) Bestimme die $m + 1$ Gruppen $(H_1, \ldots, H_{m+1})$ und die dazugehörigen Instanzen J_{LO} und J_{HI}.

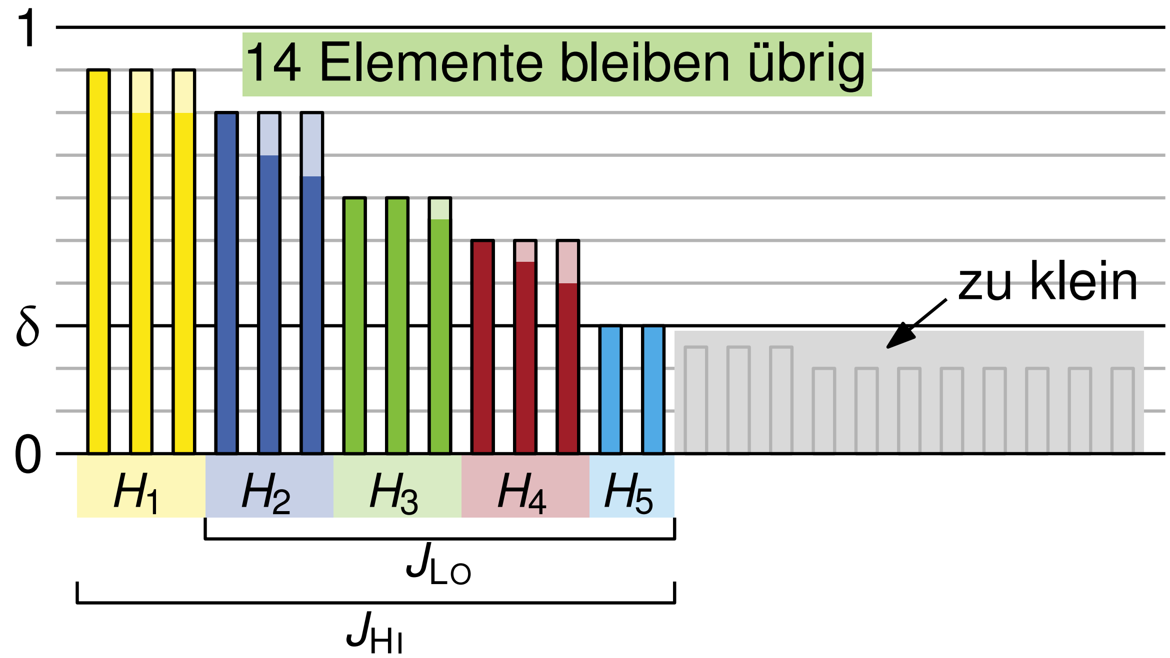
Problem 3

Führe das APAS für Bin Packing mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.18 \cdot \boxed{14} \rceil = 3$$

$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{14}}{3} \right\rfloor = 4$$



(b) Bestimme die $m+1$ Gruppen $(H_1, \ldots, H_{m+1})$ und die dazugehörigen Instanzen J_{LO} und J_{HI}.

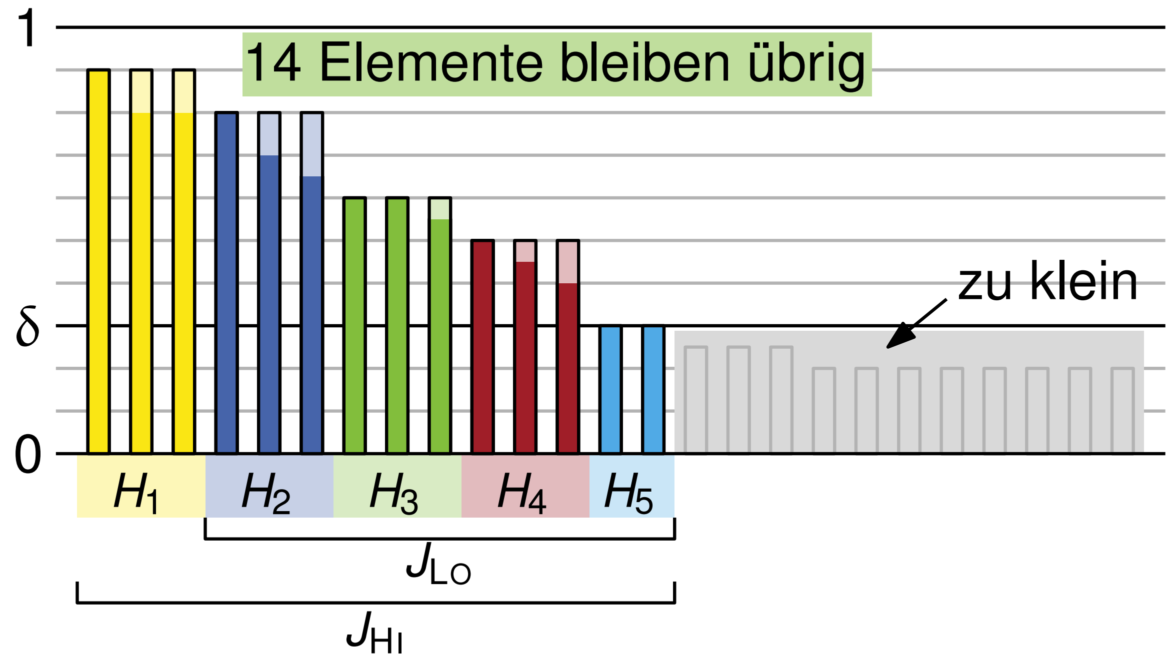
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(c) Lösen Sie die Instanz J_{LO} optimal (durch scharfes Hinsehen).

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

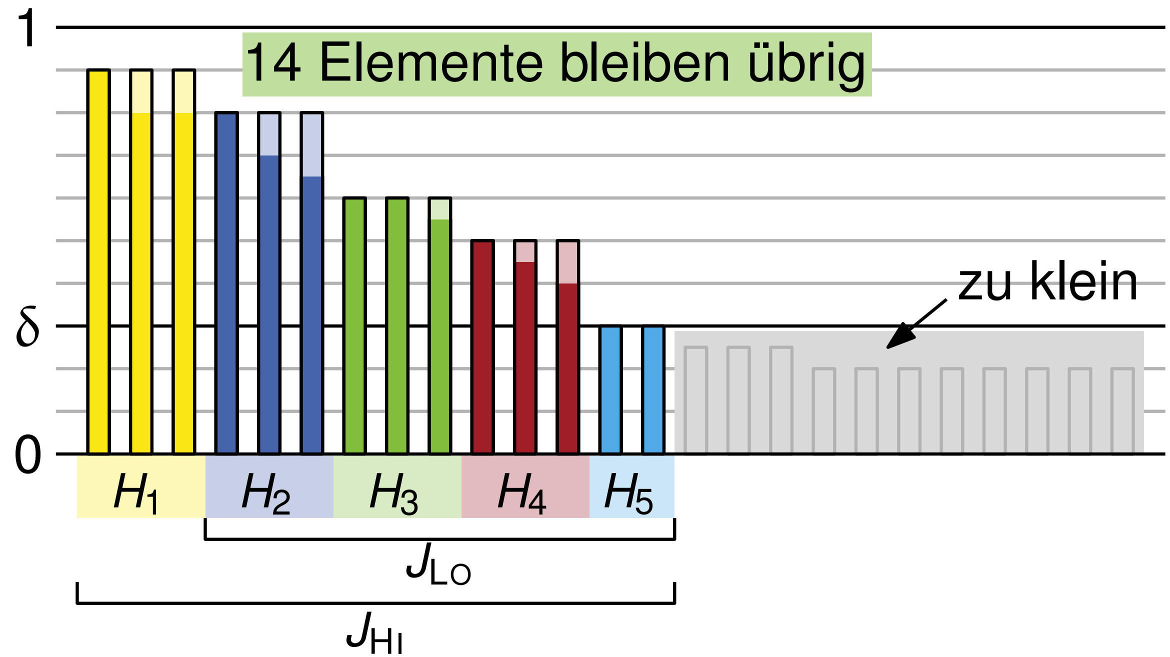
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

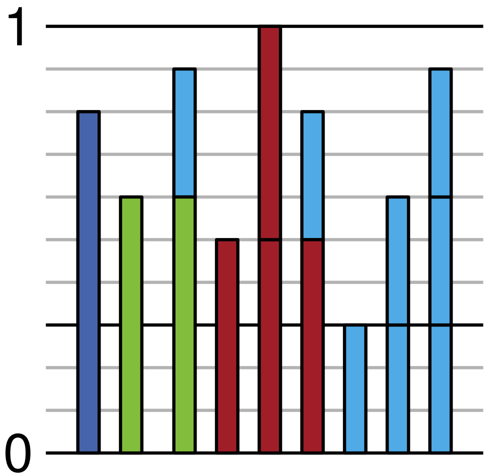
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(c) Lösen Sie die Instanz J_{LO} optimal (durch scharfes Hinsehen).

mögliche Bin-Typen



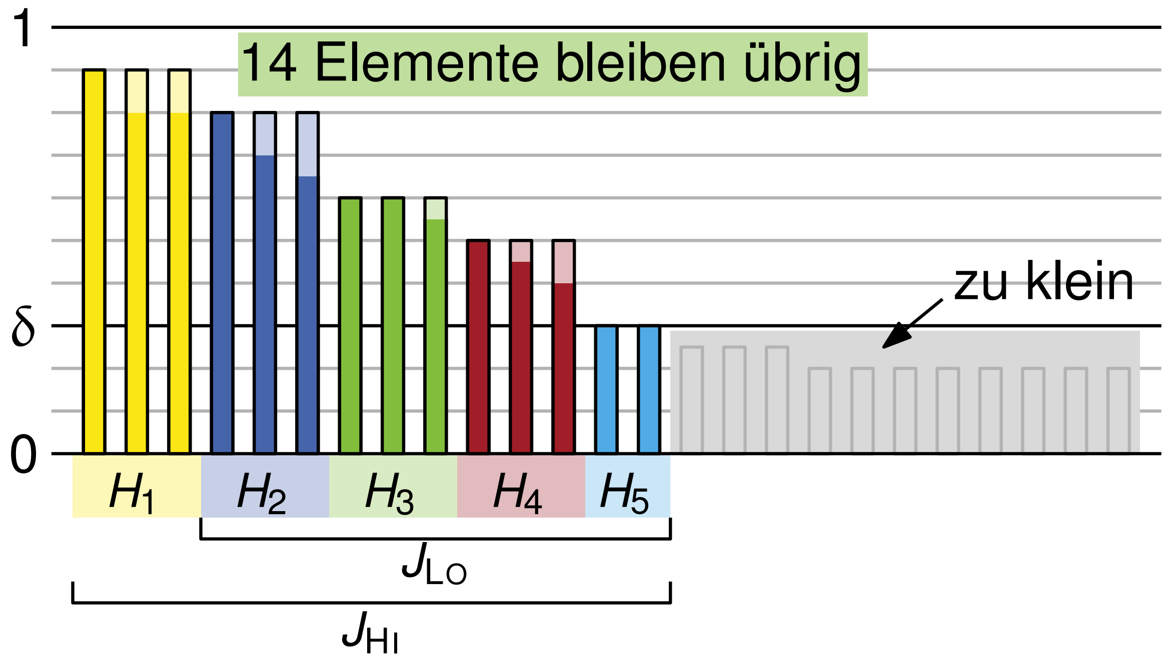
Problem 3

Führe das APAS für Bin Packing mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

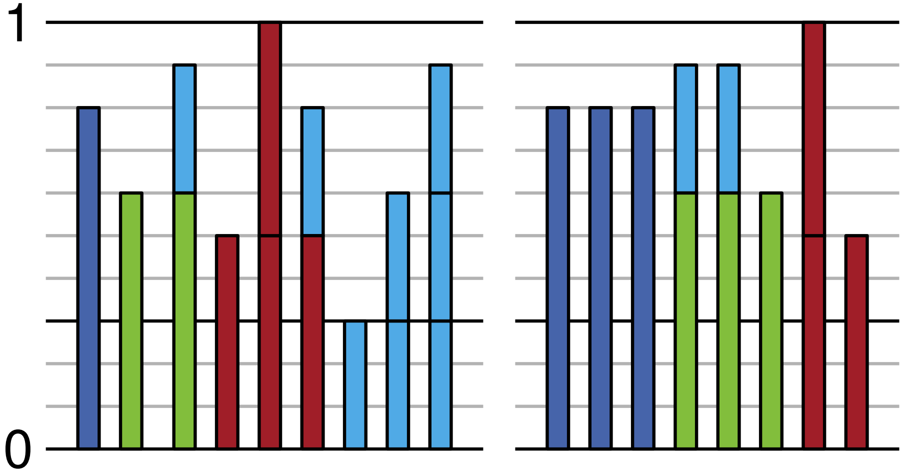
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(c) Lösen Sie die Instanz J_{LO} optimal (durch scharfes Hinsehen).

mögliche Bin-Typen **opt. Lsg für J_{LO}**



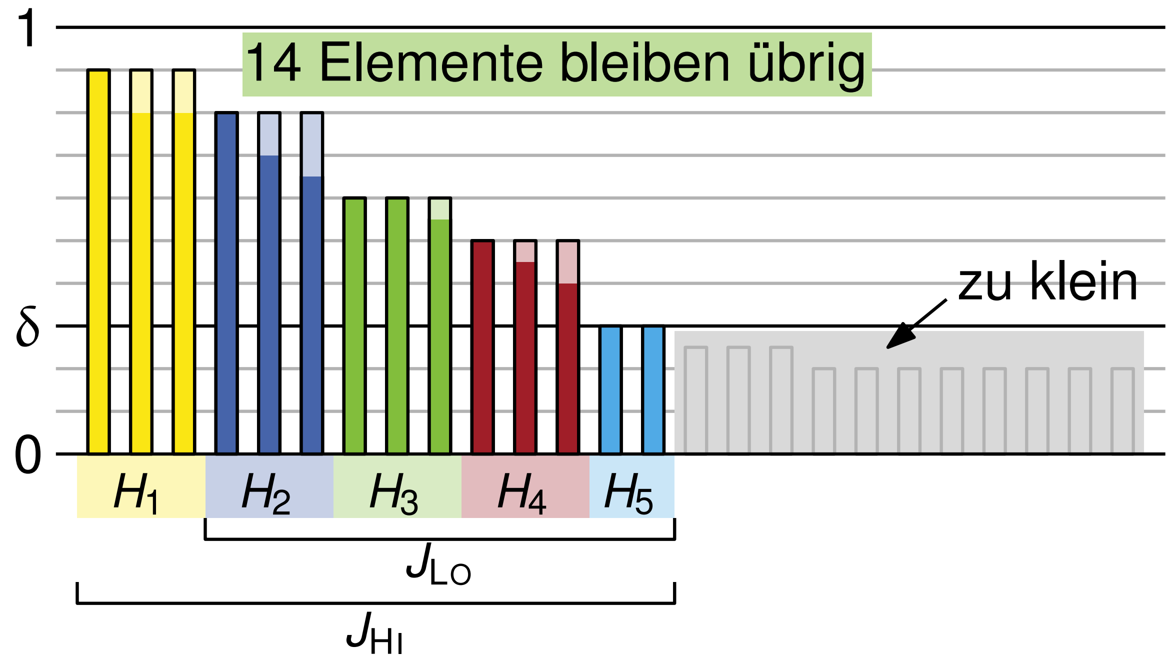
Problem 3

Führe das APAS für Bin Packing mit $\varepsilon = 0.6$ an einem Beispiel aus.

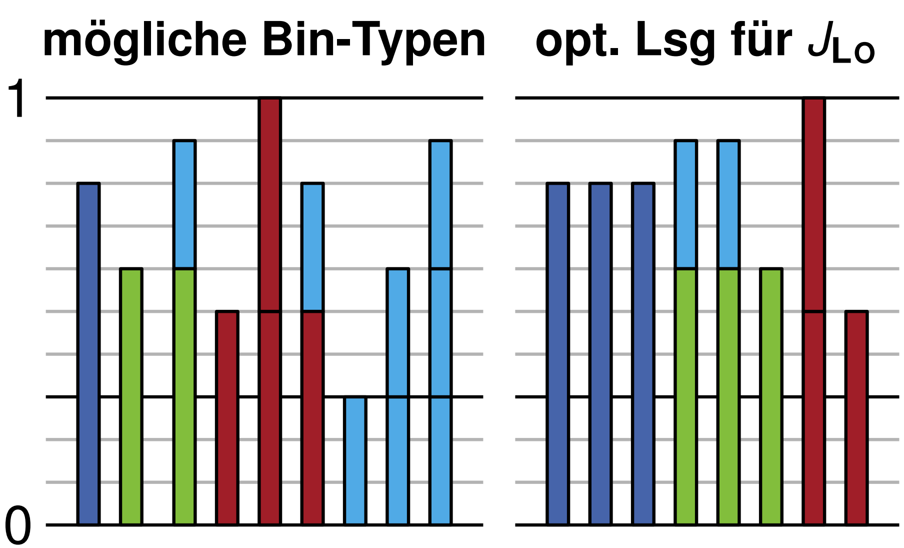
$\delta = \frac{\varepsilon}{2} = 0.3$

$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$

$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$



(d) Erweitere die optimale Lösung von J_{LO} zu einer Lösung von J_{HI} . . .



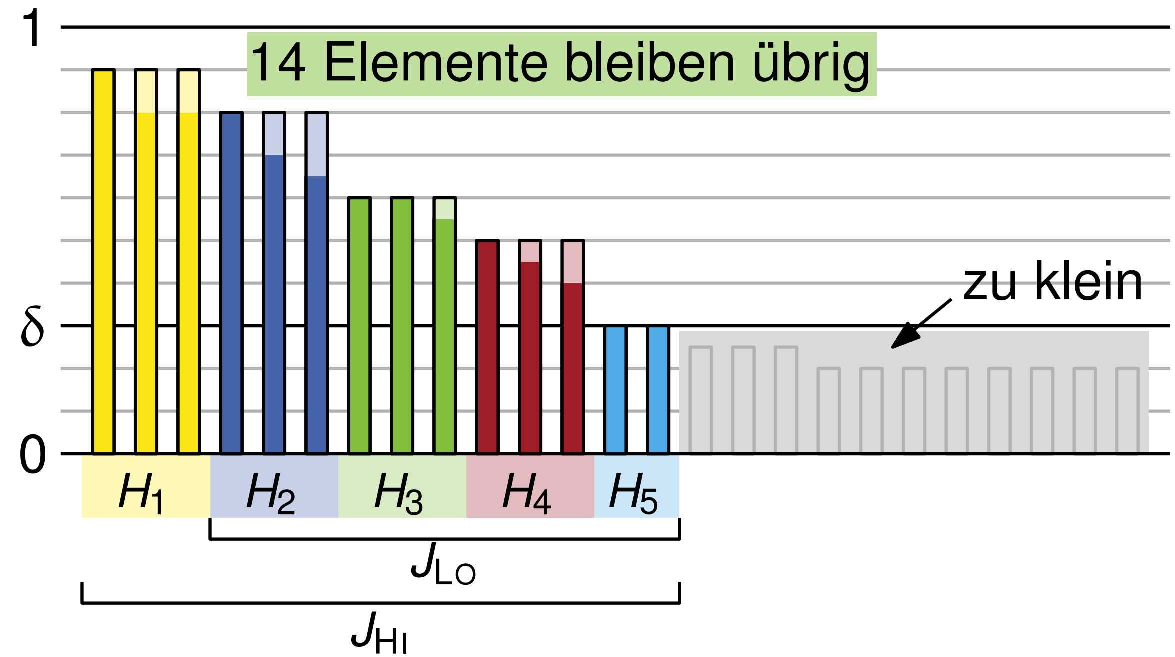
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

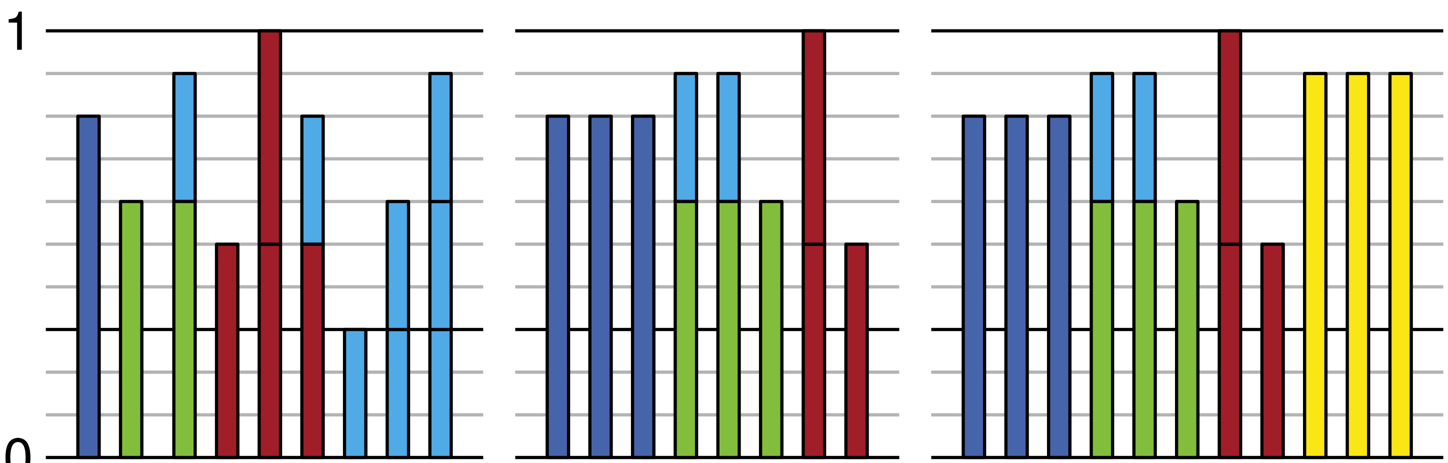
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.18 \cdot \boxed{14} \rceil = 3$$

$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{14}}{3} \right\rfloor = 4$$



(d) Erweitere die optimale Lösung von J_{LO} zu einer Lösung von J_{HI} . . .

mögliche Bin-Typen **opt. Lsg für J_{LO}** **Lösung für J_{HI}**



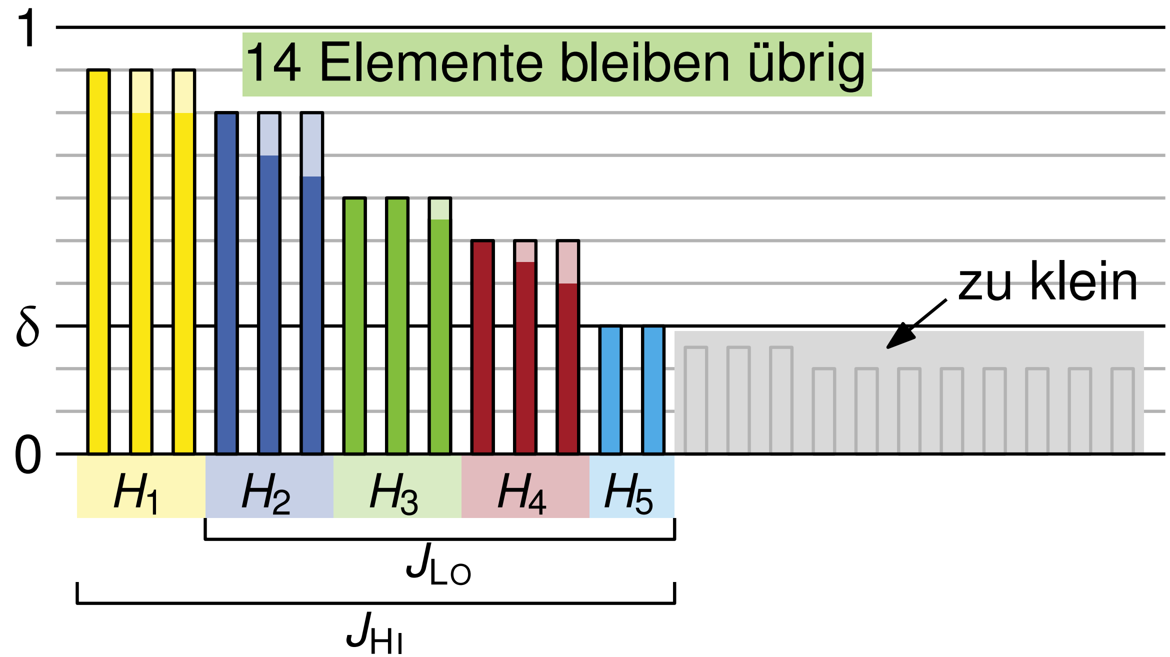
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

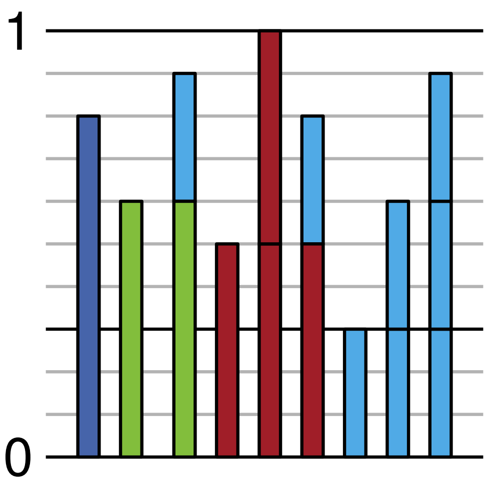
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.18 \cdot \boxed{14} \rceil = 3$$

$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{14}}{3} \right\rfloor = 4$$

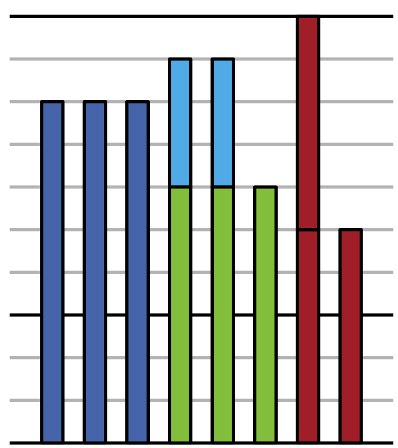


(d) ... und berechne daraus eine Lösung von J.

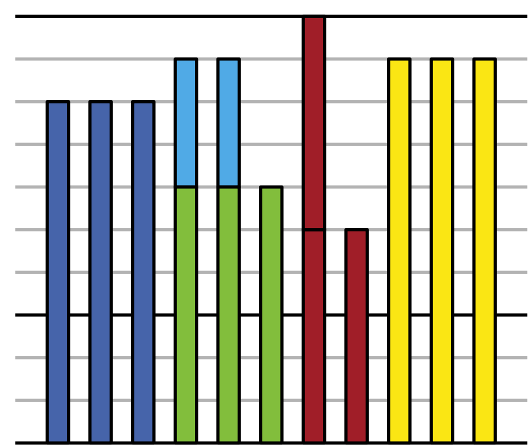
mögliche Bin-Typen



opt. Lsg für J_{LO}



Lösung für J_{HI}



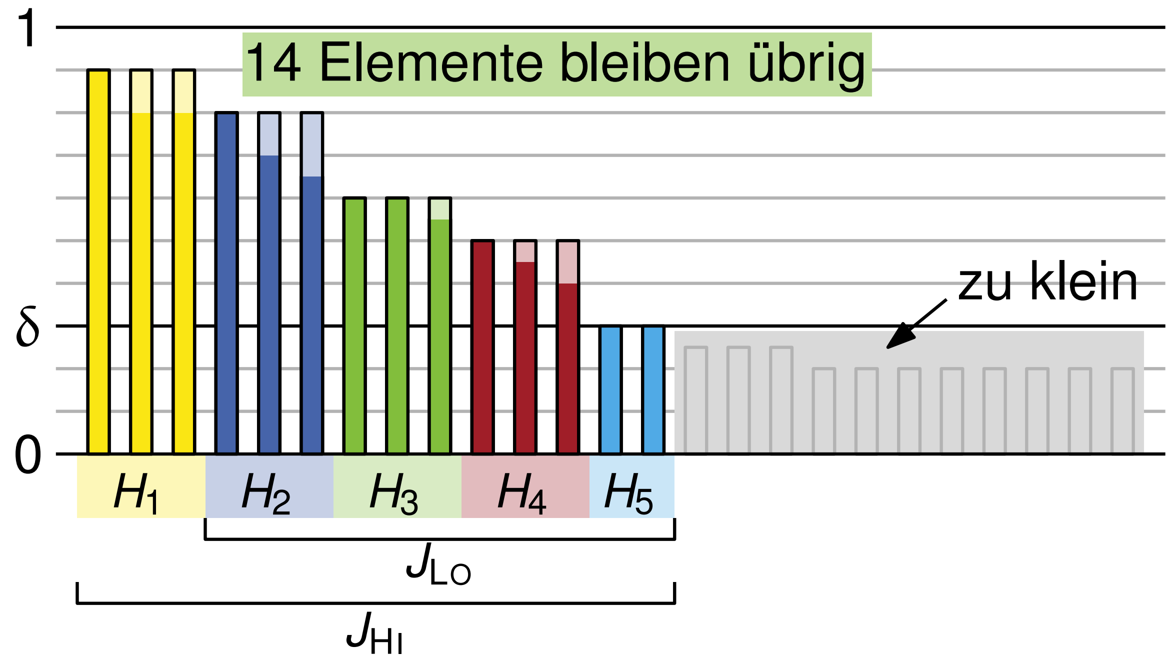
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

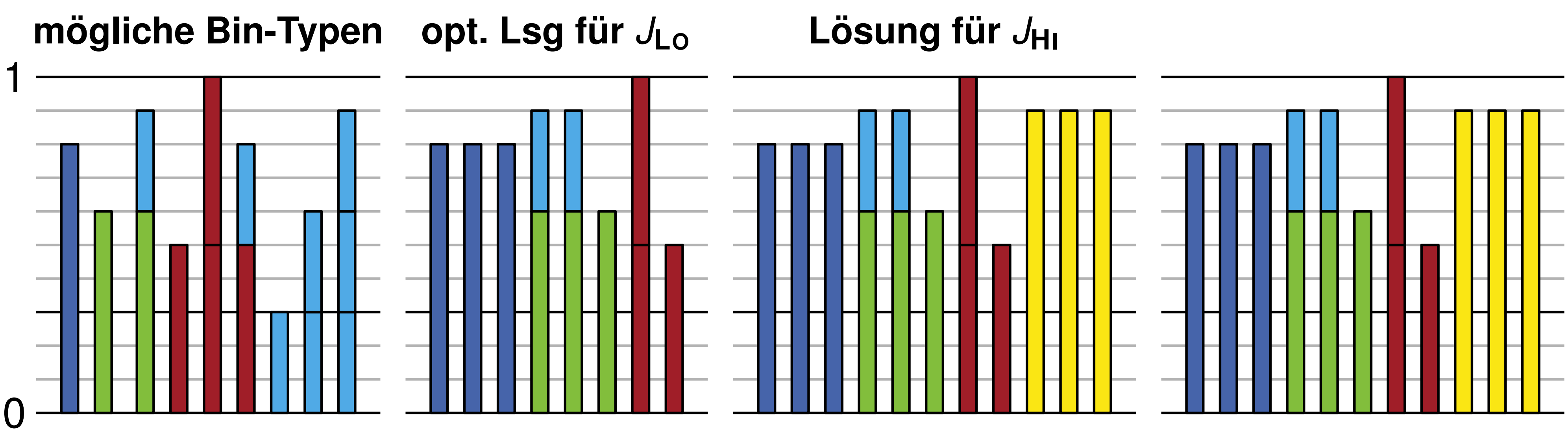
$\delta = \frac{\varepsilon}{2} = 0.3$

$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$

$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$



(d) ...und berechne daraus eine Lösung von J.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

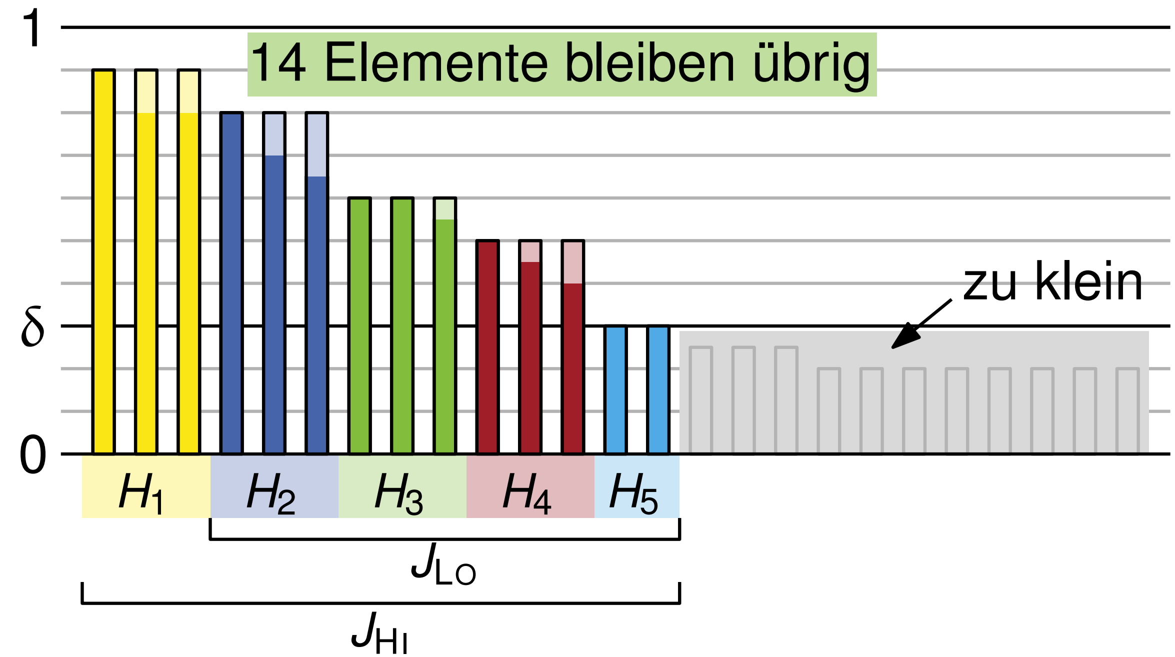
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

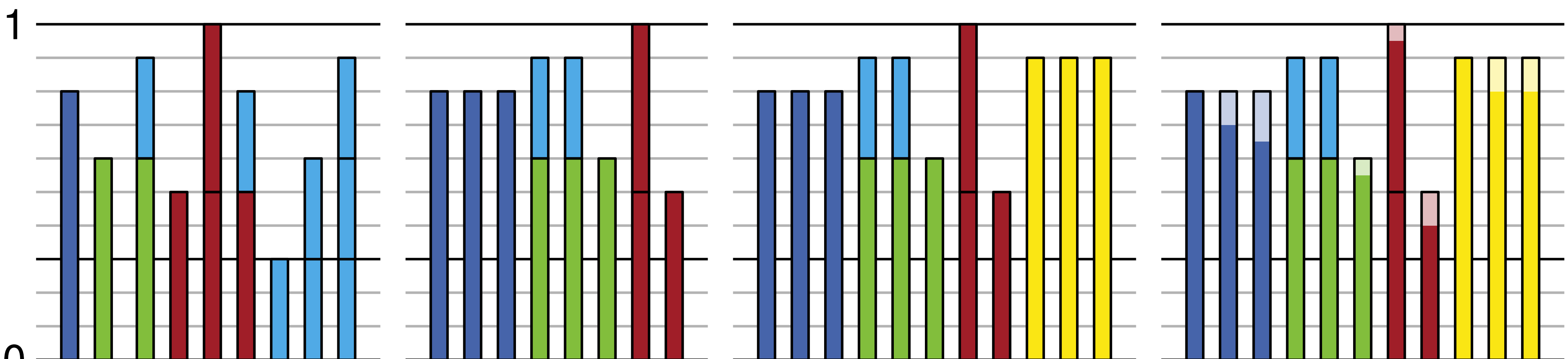
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.18 \cdot \boxed{14} \rceil = 3$$

$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{14}}{3} \right\rfloor = 4$$



(d) . . . und berechne daraus eine Lösung von J.

mögliche Bin-Typen **opt. Lsg für J_{LO}** **Lösung für J_{HI}**



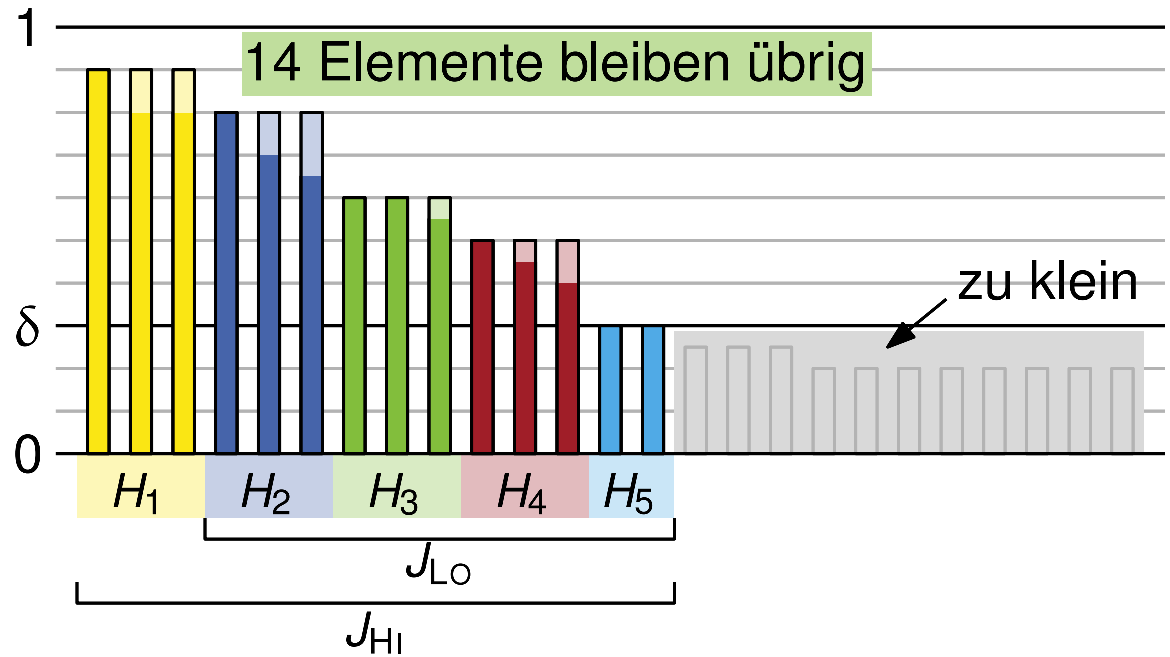
Problem 3

Führe das APAS für Bin Packing mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

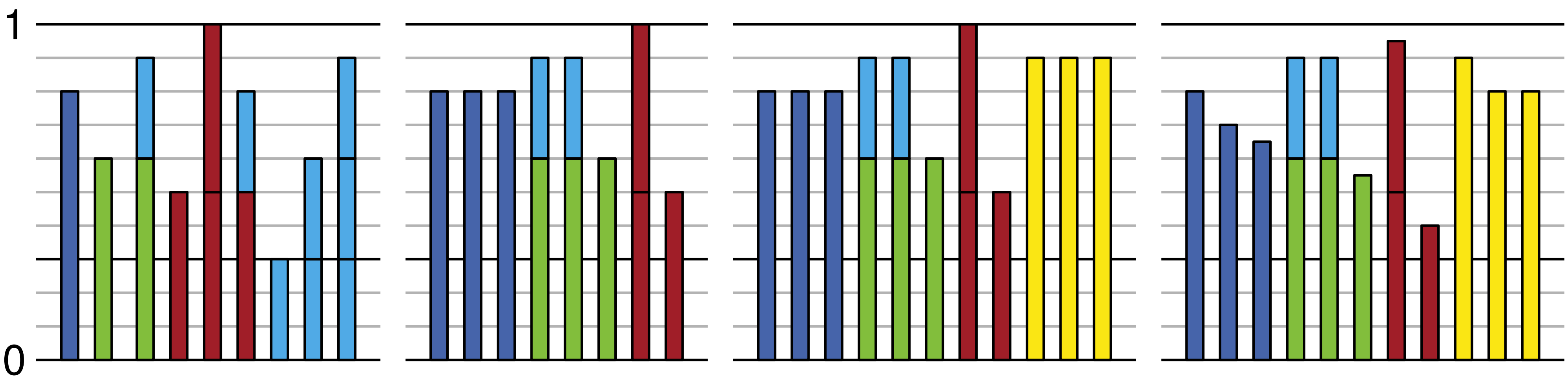
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.18 \cdot \boxed{14} \rceil = 3$$

$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{14}}{3} \right\rfloor = 4$$



(d) . . . und berechne daraus eine Lösung von J.

mögliche Bin-Typen **opt. Lsg für J_{LO}** **Lösung für J_{HI}** **Lösung für J**



Problem 3

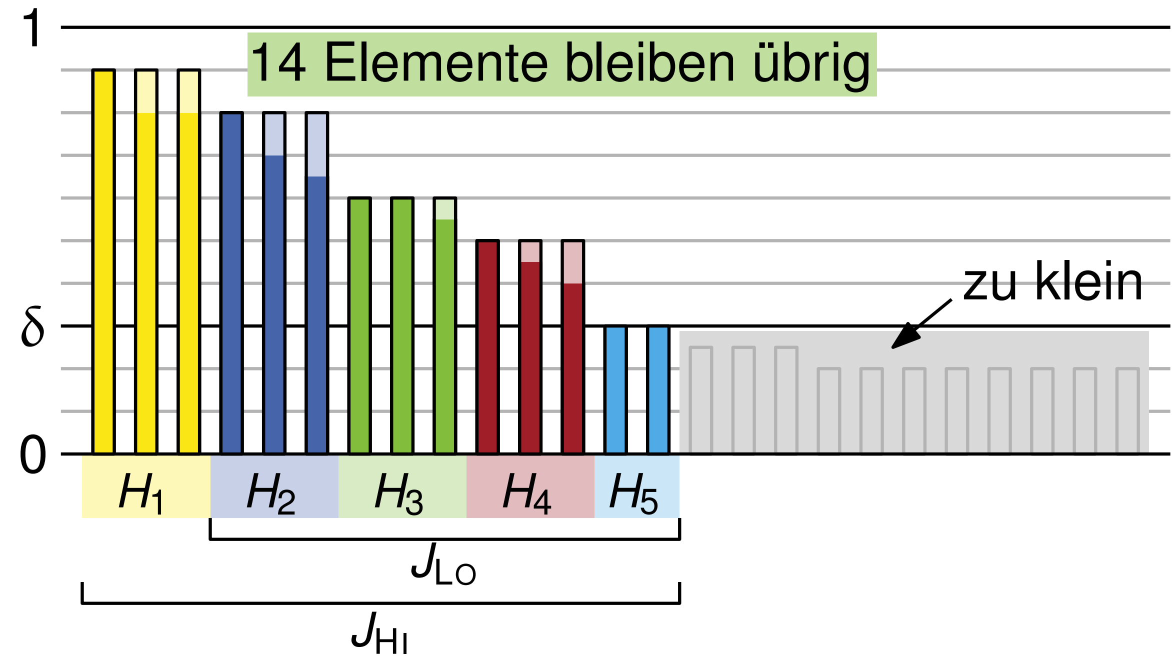
Karlsruher Institut für Technologie

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

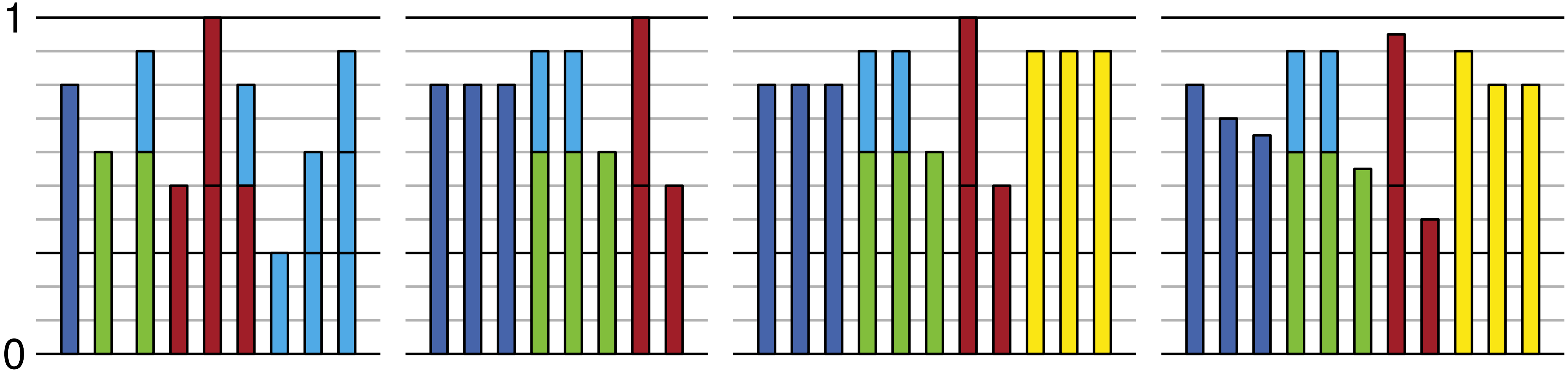
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.18 \cdot \boxed{14} \rceil = 3$$

$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{14}}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.

mögliche Bin-Typen | **opt. Lsg für J_{LO}** | **Lösung für J_{HI}** | **Lösung für J**



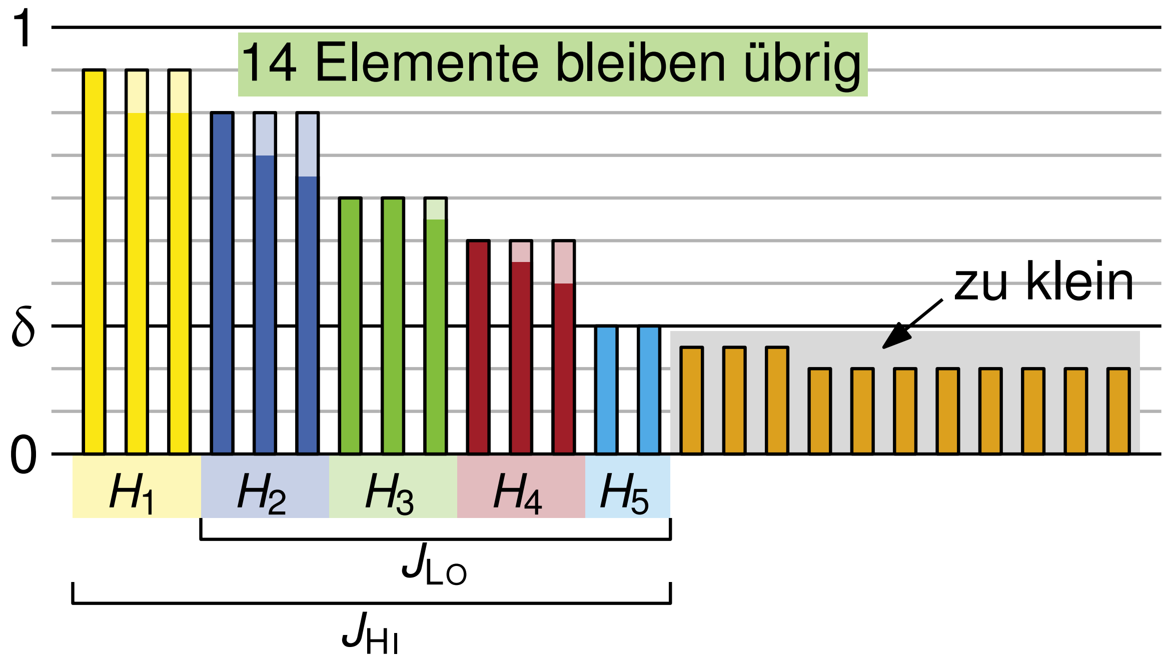
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

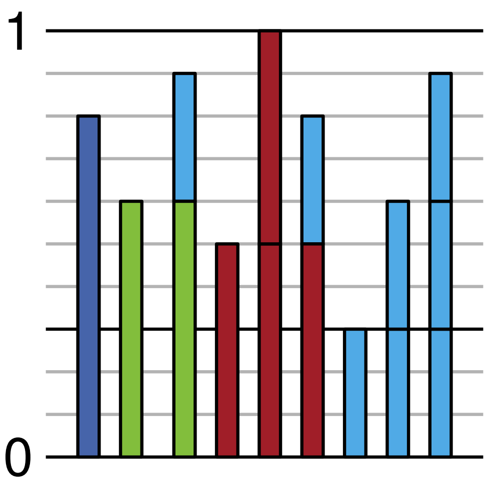
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$

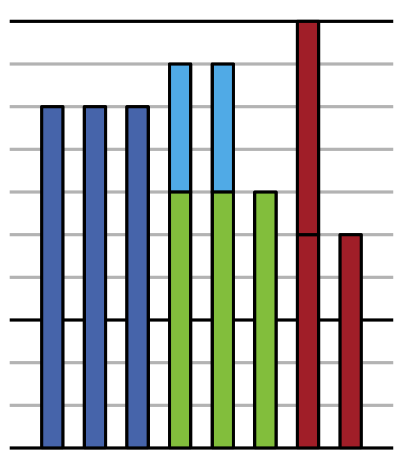


(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.

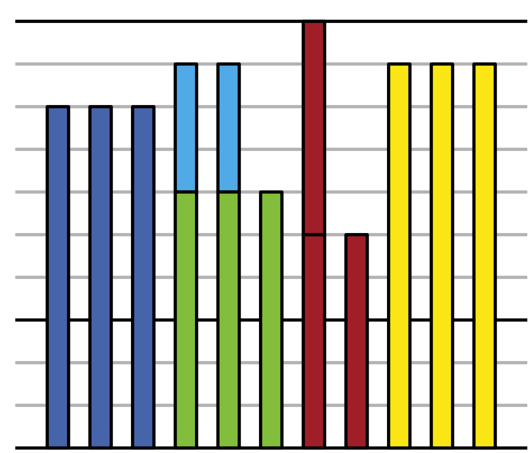
mögliche Bin-Typen



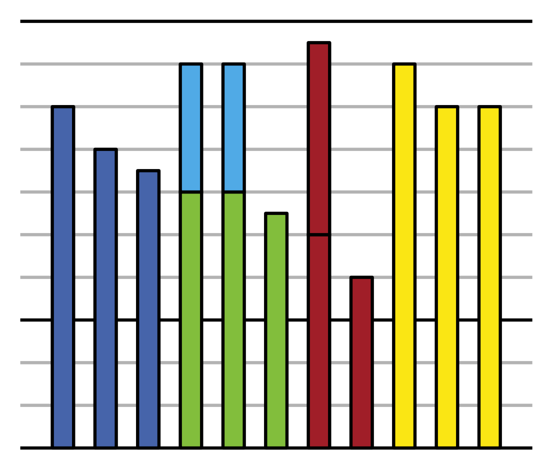
opt. Lsg für J_{LO}



Lösung für J_{HI}



Lösung für J



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

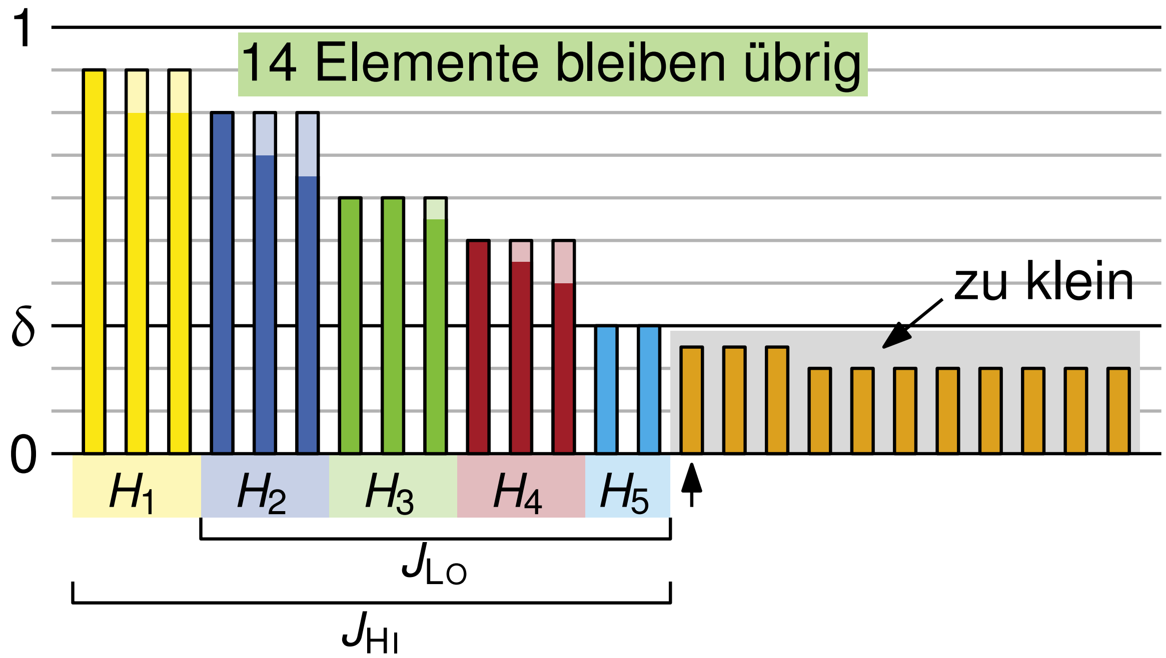
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

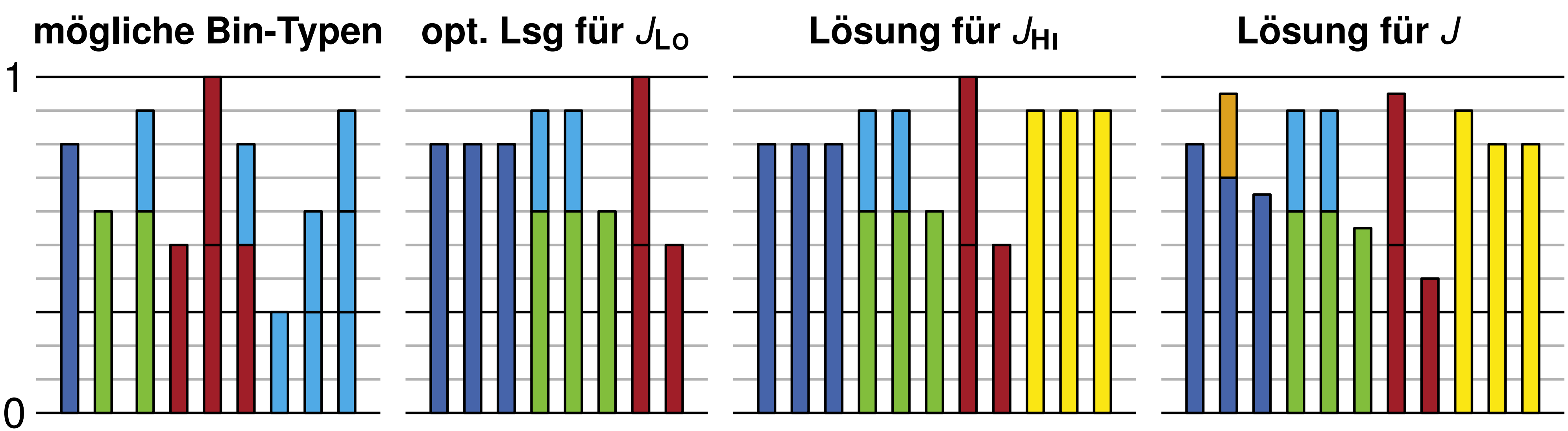
$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.



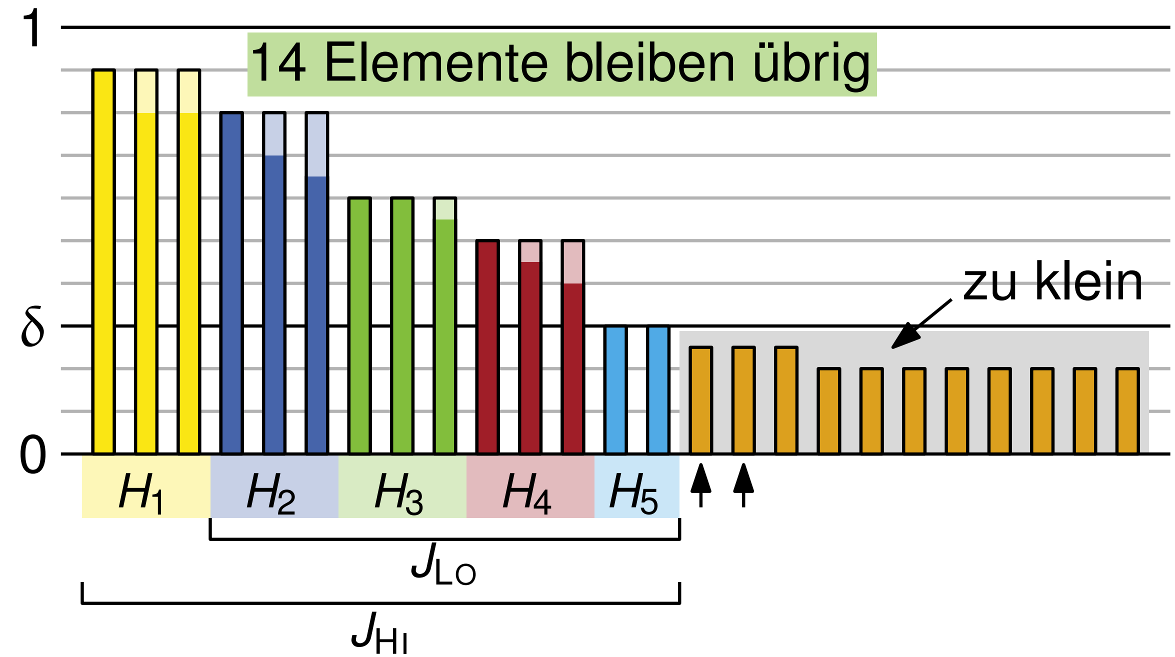
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

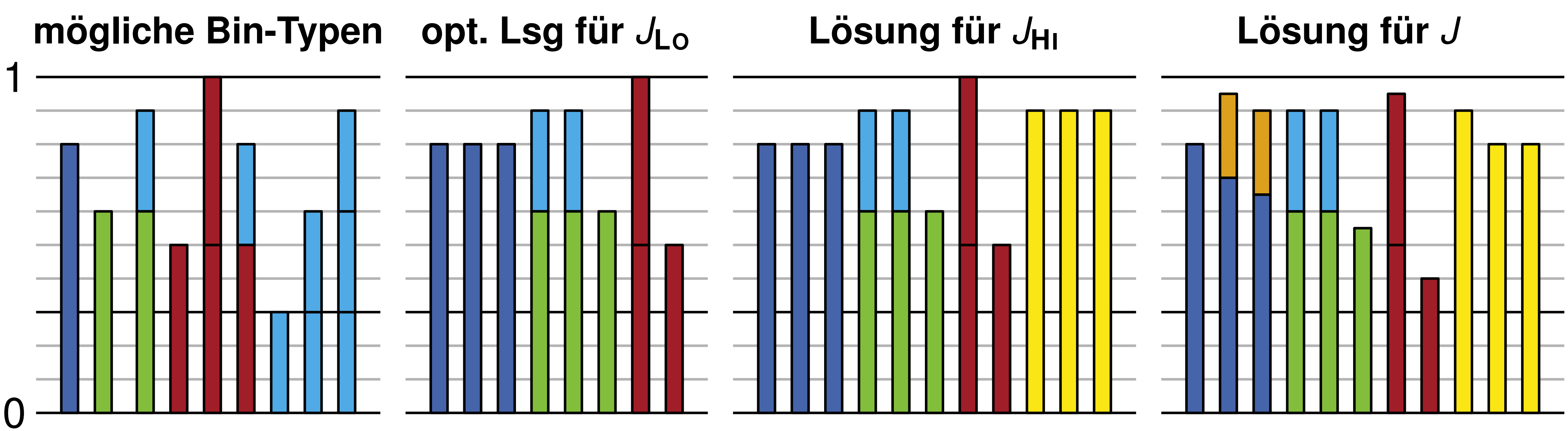
$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.



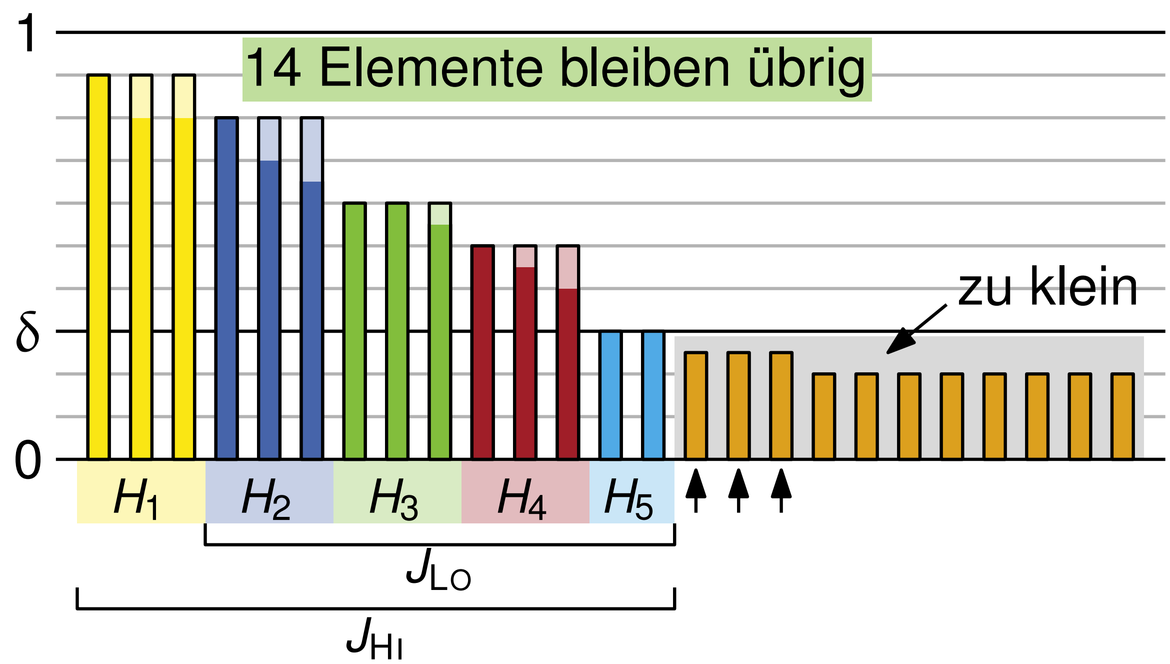
Problem 3

Führe das APAS für Bin Packing mit $\varepsilon = 0.6$ an einem Beispiel aus.

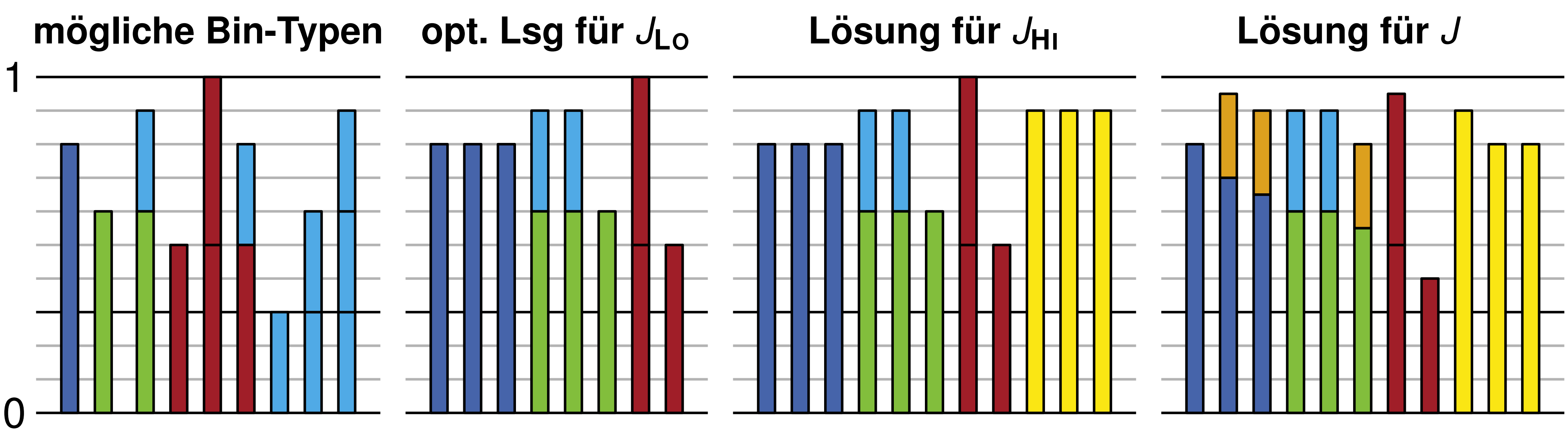
$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels First Fit zu einer Lösung von I.



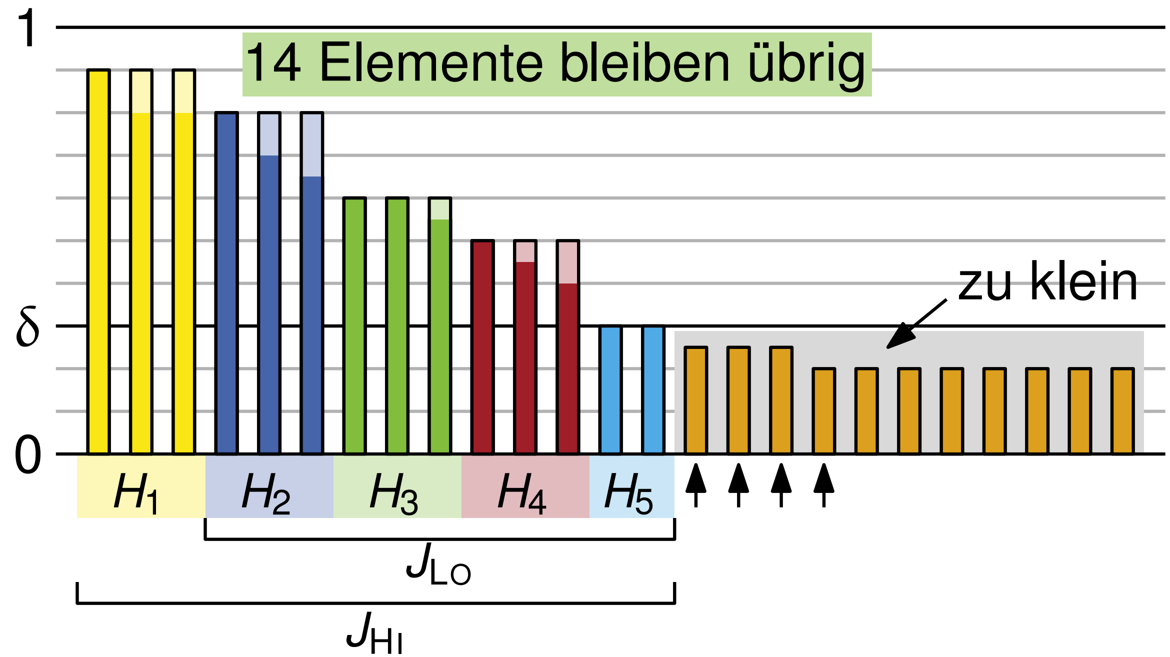
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

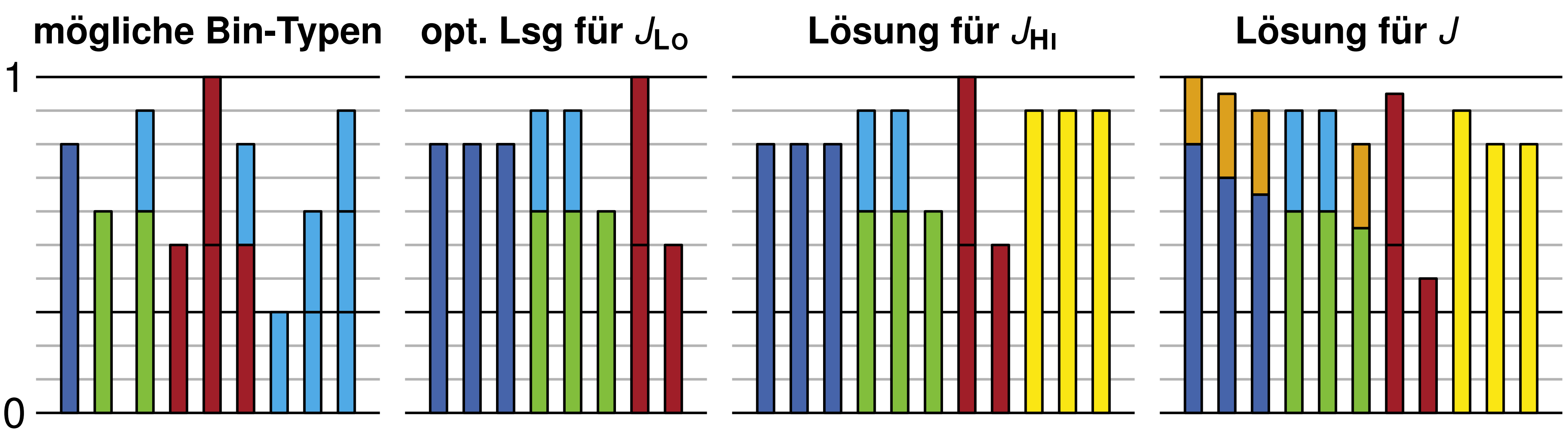
$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

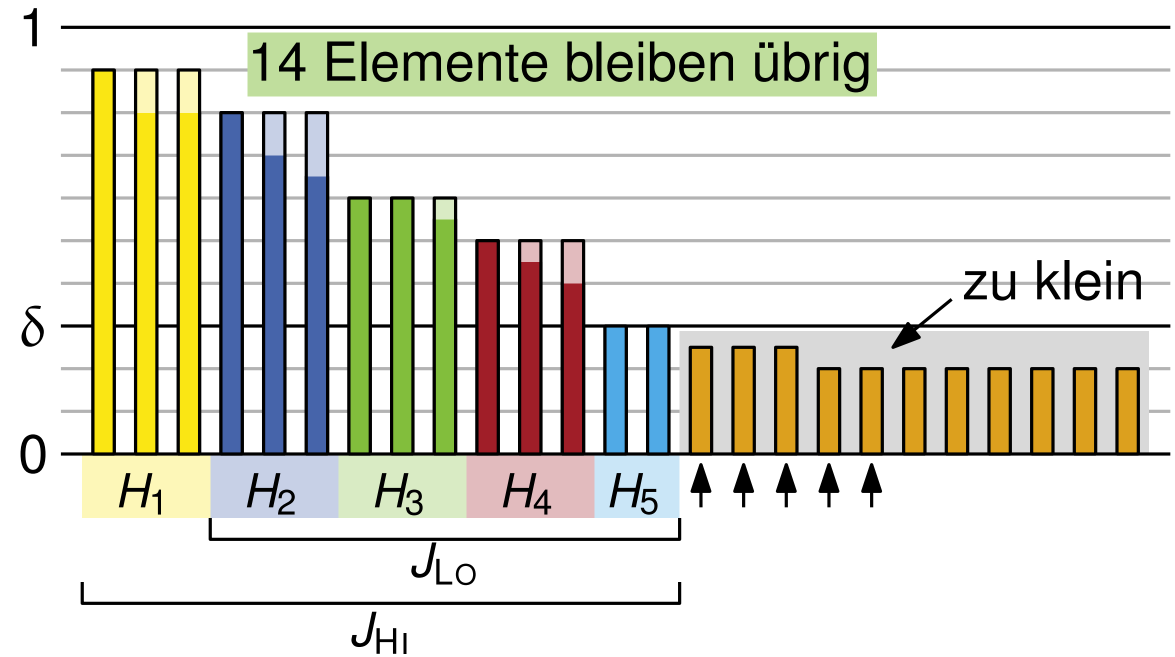
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

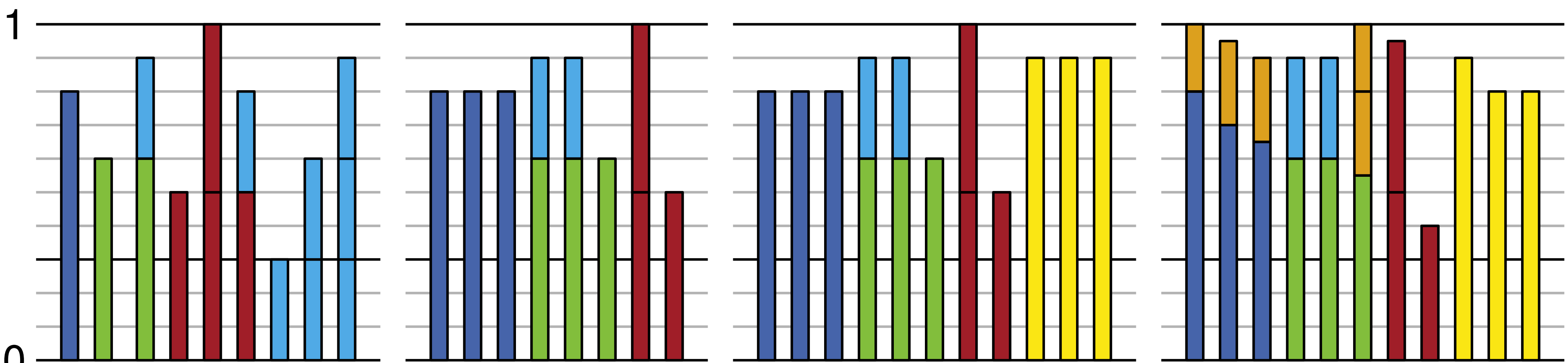
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.

mögliche Bin-Typen | **opt. Lsg für J_{Lo}** | **Lösung für J_{Hi}** | **Lösung für J**

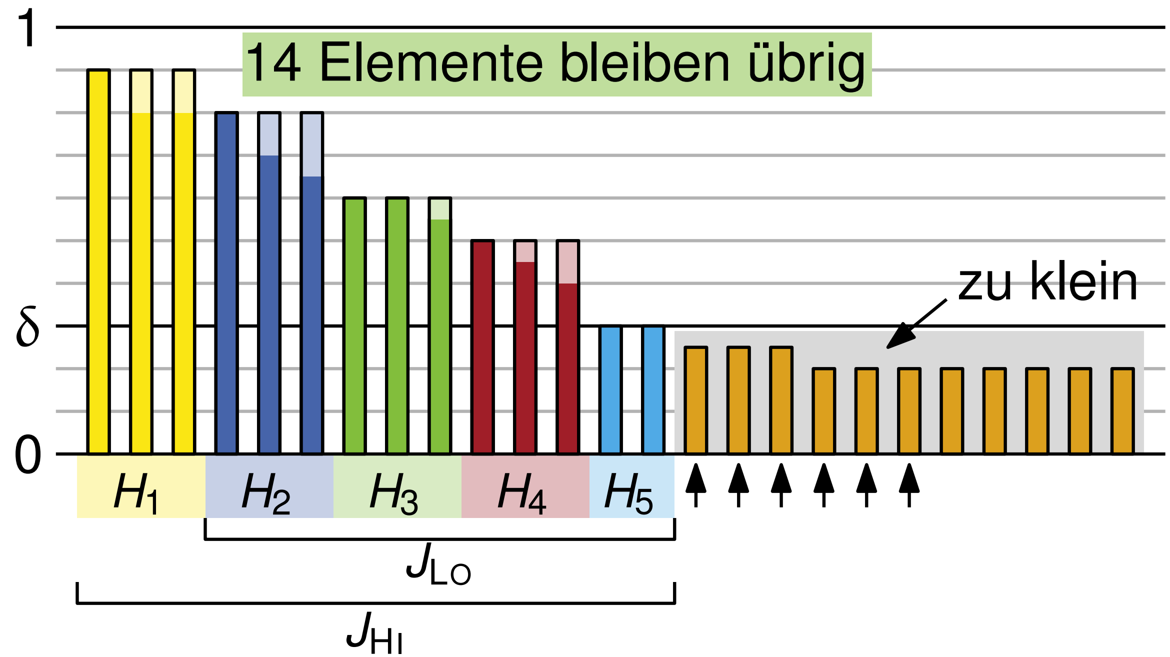


Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

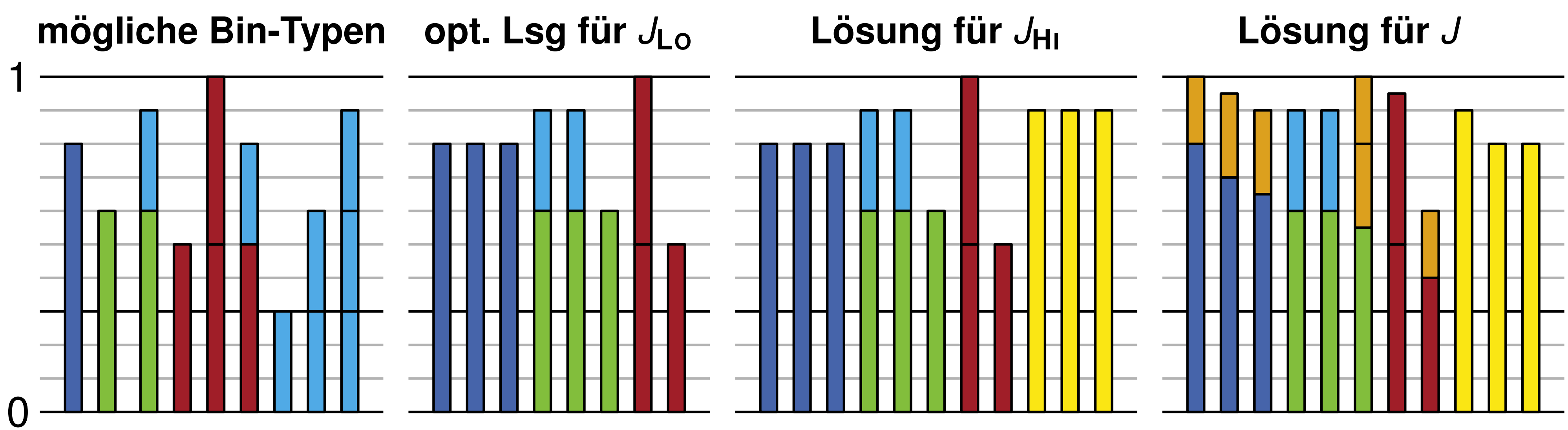
$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.



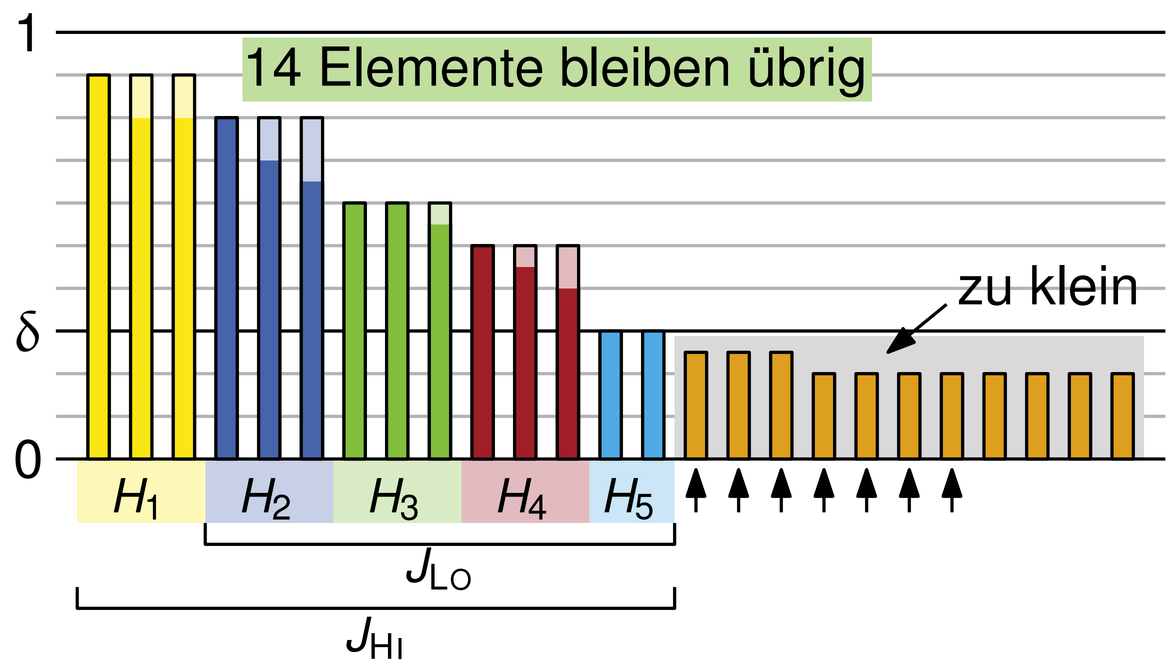
Problem 3

Führe das APAS für Bin Packing mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

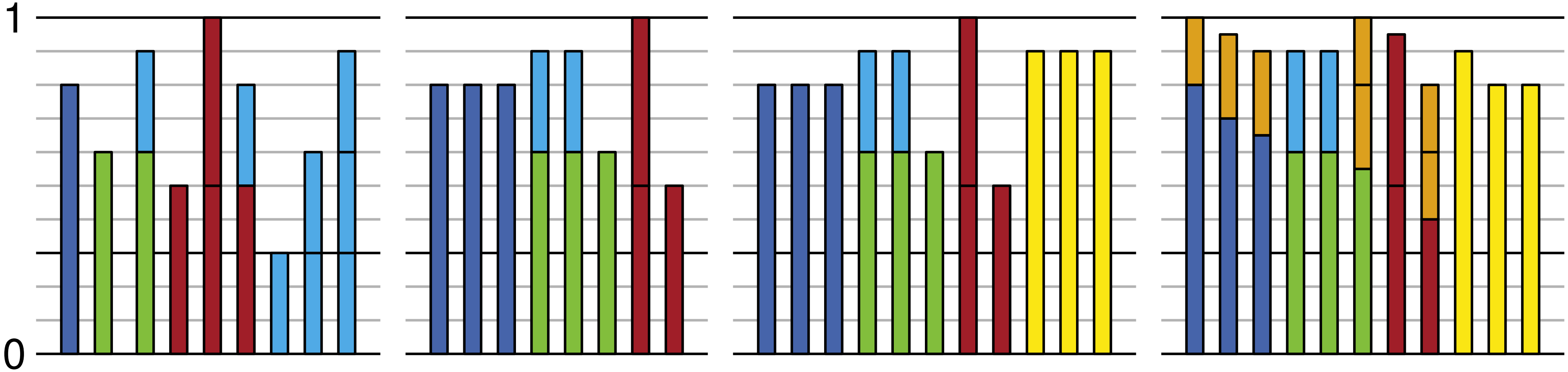
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels First Fit zu einer Lösung von I.

mögliche Bin-Typen **opt. Lsg für J_{LO}** **Lösung für J_{HI}** **Lösung für J**



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

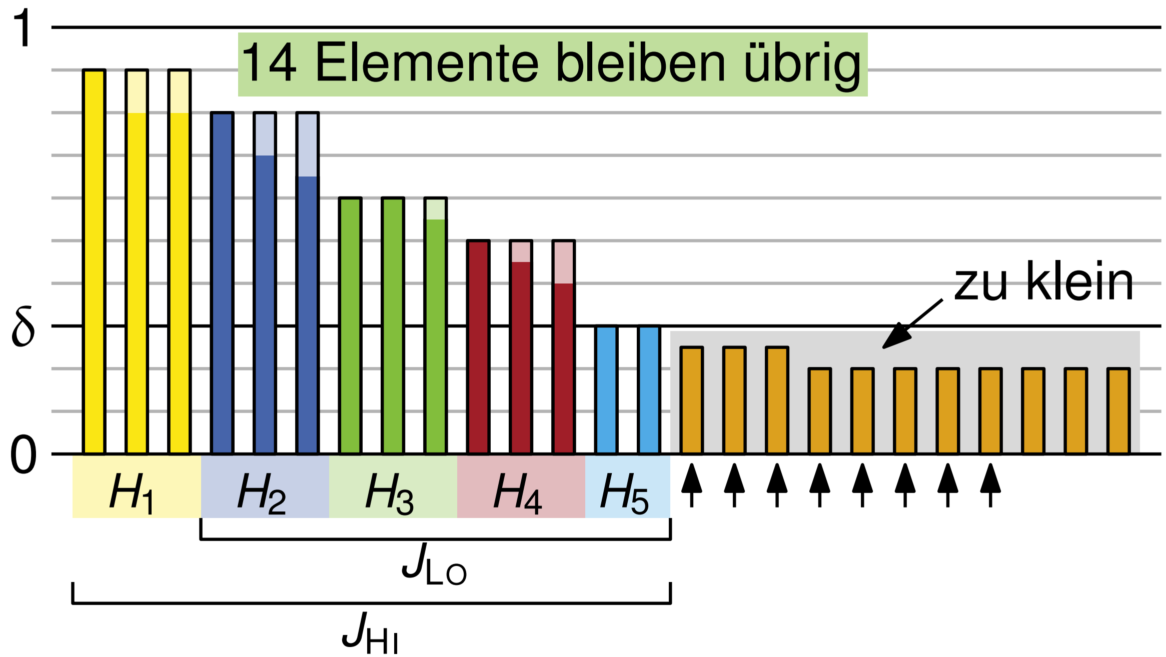
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

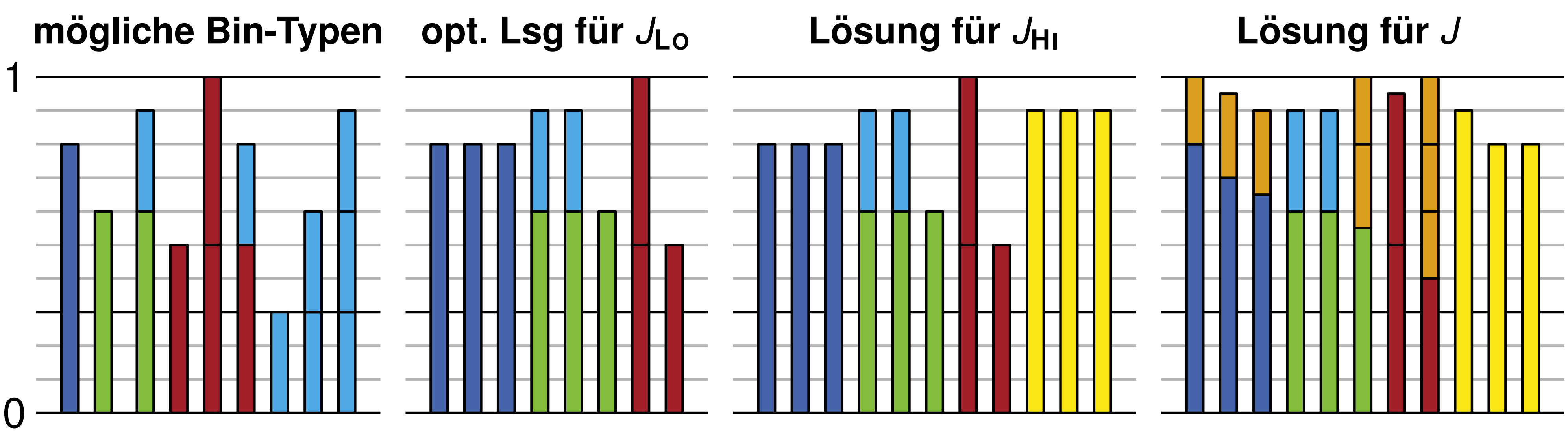
$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

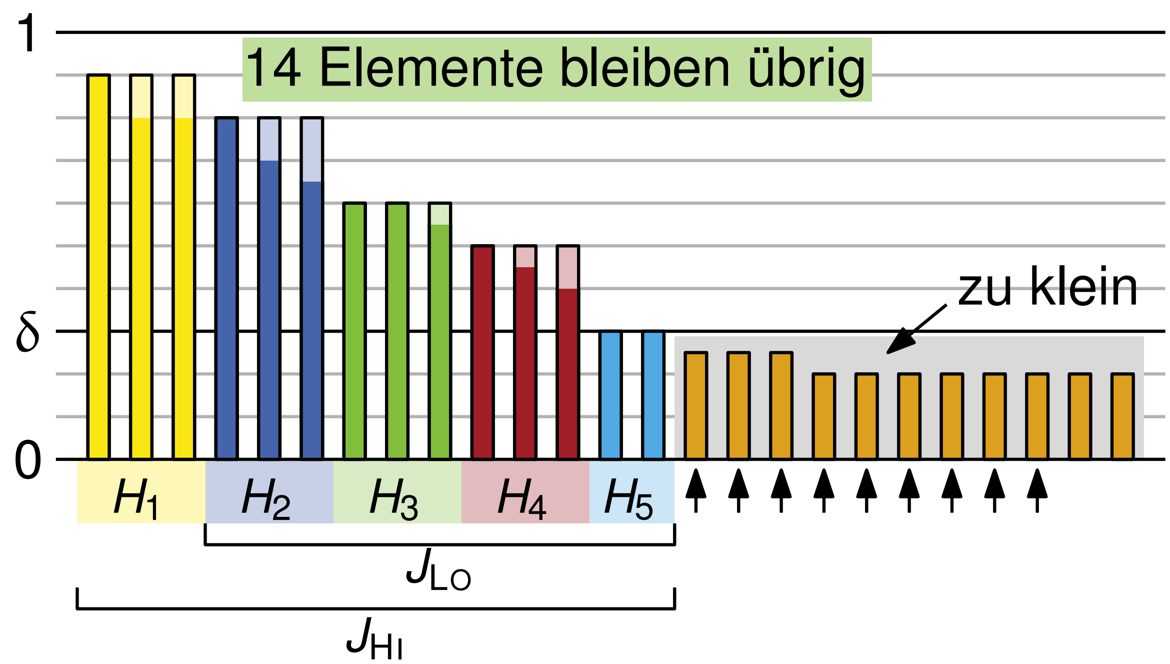
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

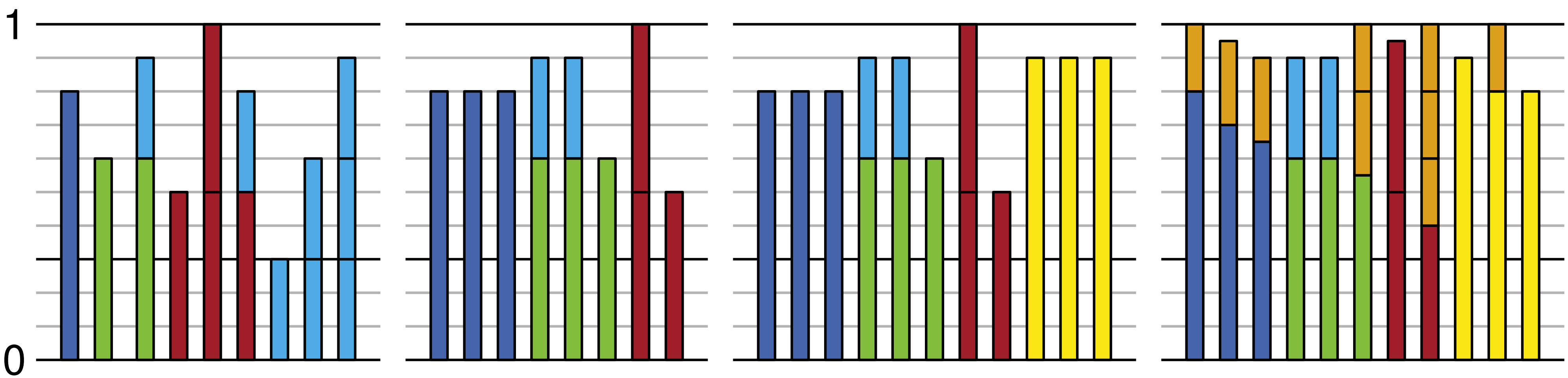
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.

mögliche Bin-Typen **opt. Lsg für J_{LO}** **Lösung für J_{HI}** **Lösung für J**



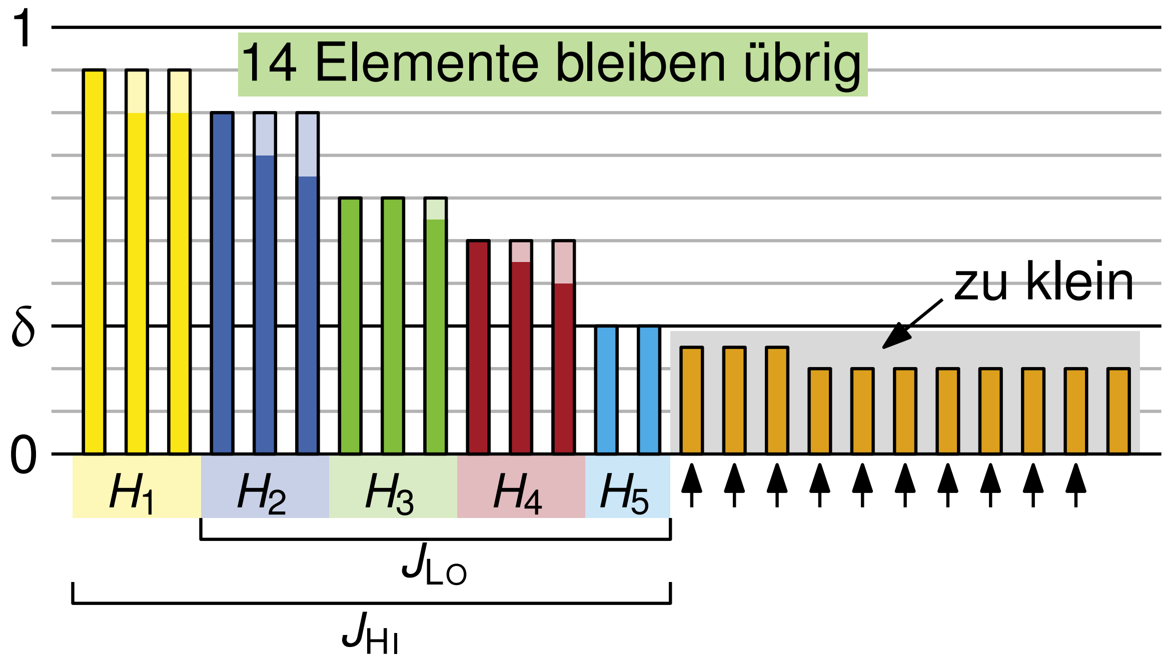
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

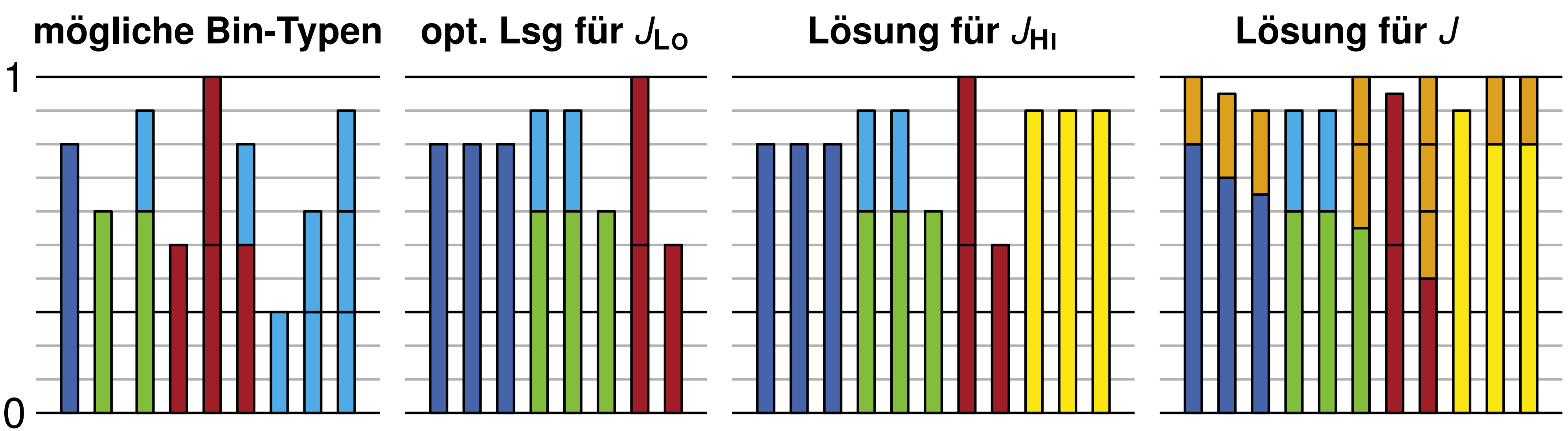
$$\delta = \frac{\varepsilon}{2} = 0.3$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

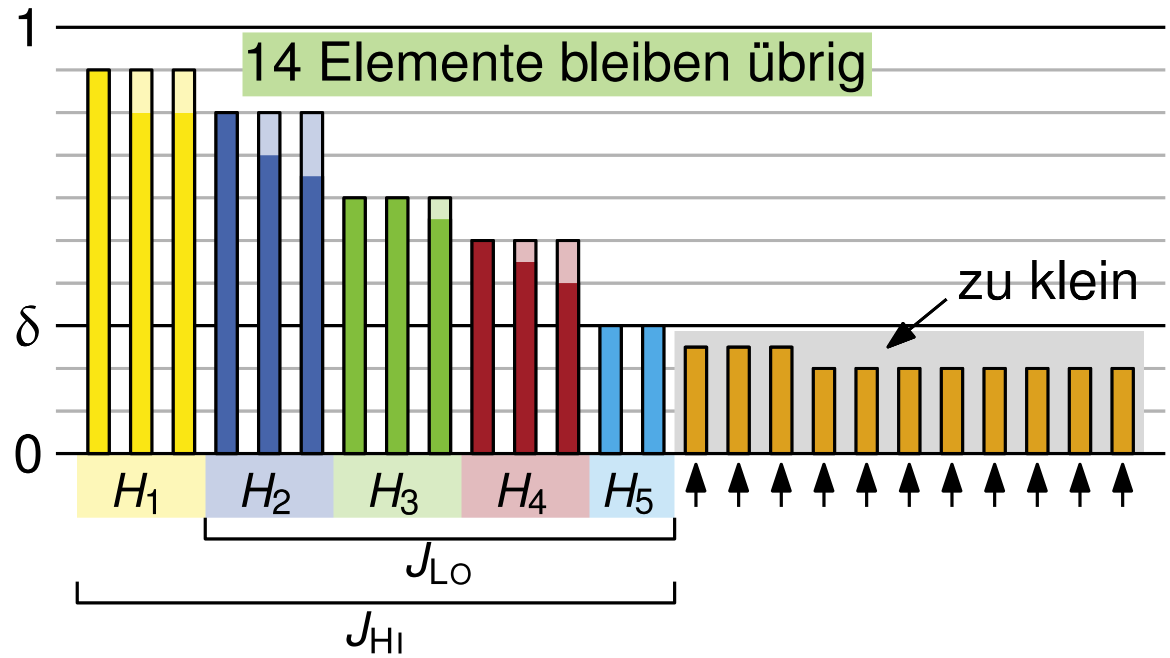
Problem 3

Führe das APAS für BIN PACKING mit $\varepsilon = 0.6$ an einem Beispiel aus.

$$\delta = \frac{\varepsilon}{2} = 0.3$$

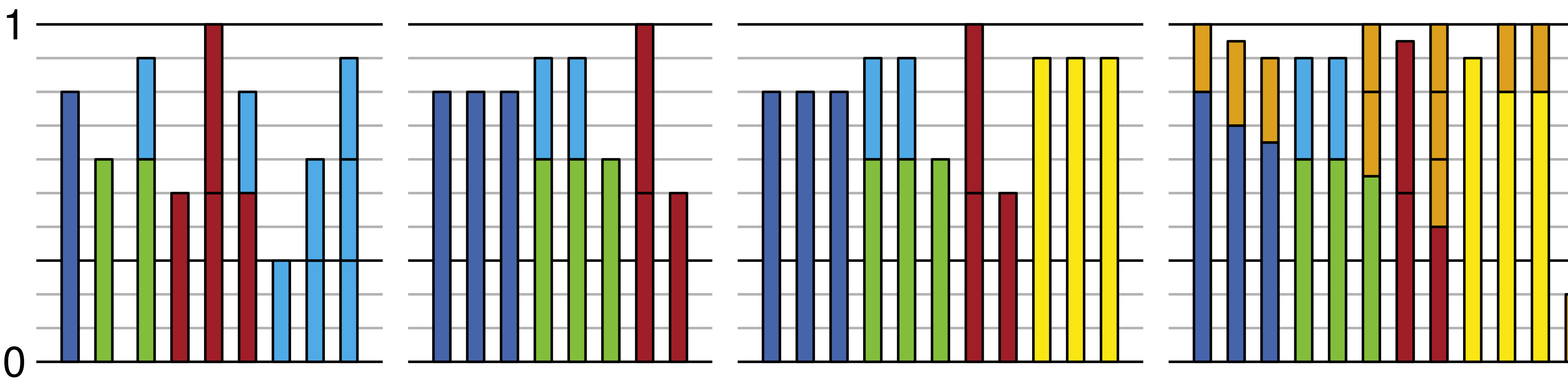
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.18 \cdot 14 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{14}{3} \right\rfloor = 4$$



(e) Erweitere die Lösung von J mittels FIRST FIT zu einer Lösung von I.

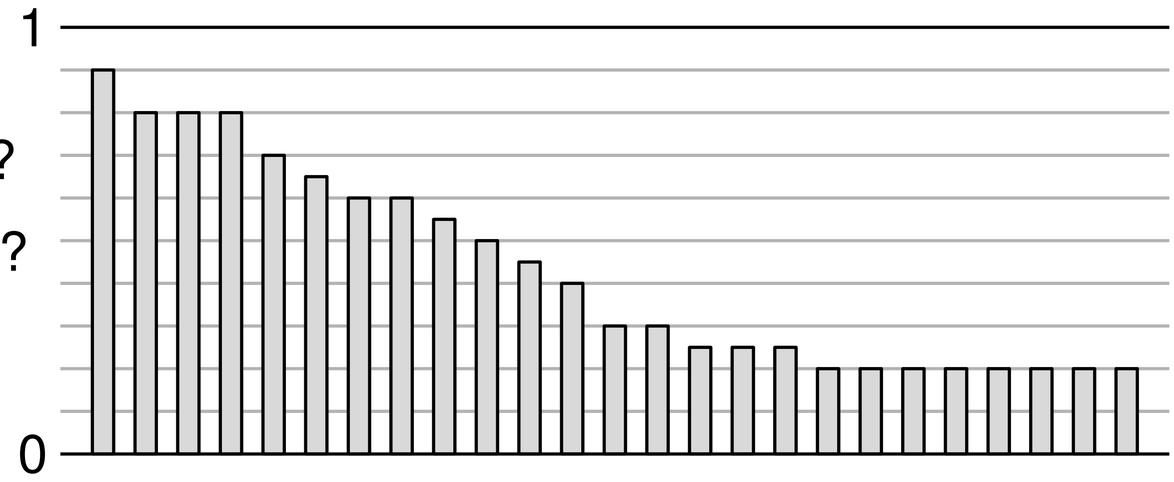
mögliche Bin-Typen **opt. Lsg für J_{LO}** **Lösung für J_{HI}** **Lösung für ~~J~~ I'**



Problem 3

Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

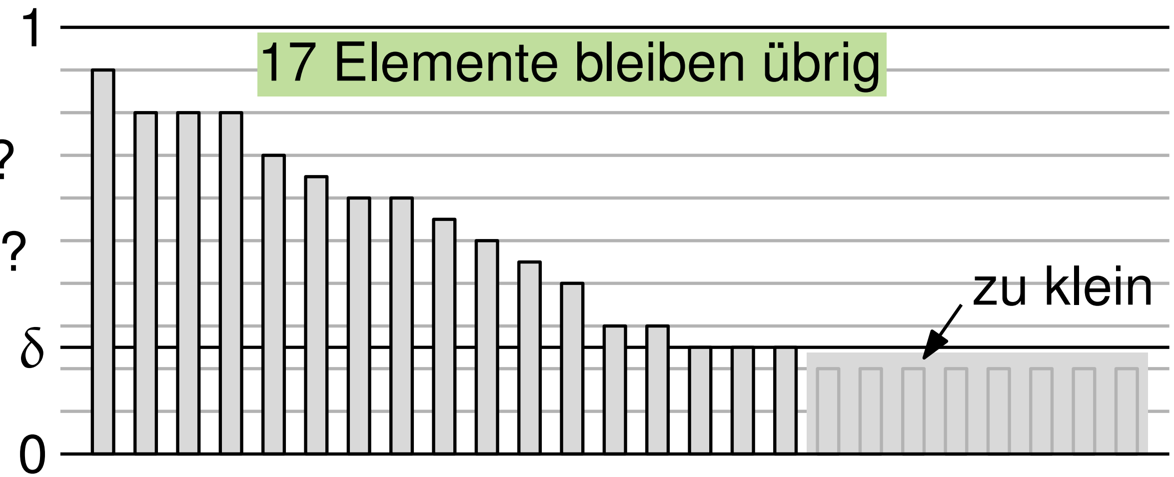


Problem 3

Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

$\delta = \frac{\varepsilon}{2} = 0.25$



Problem 3

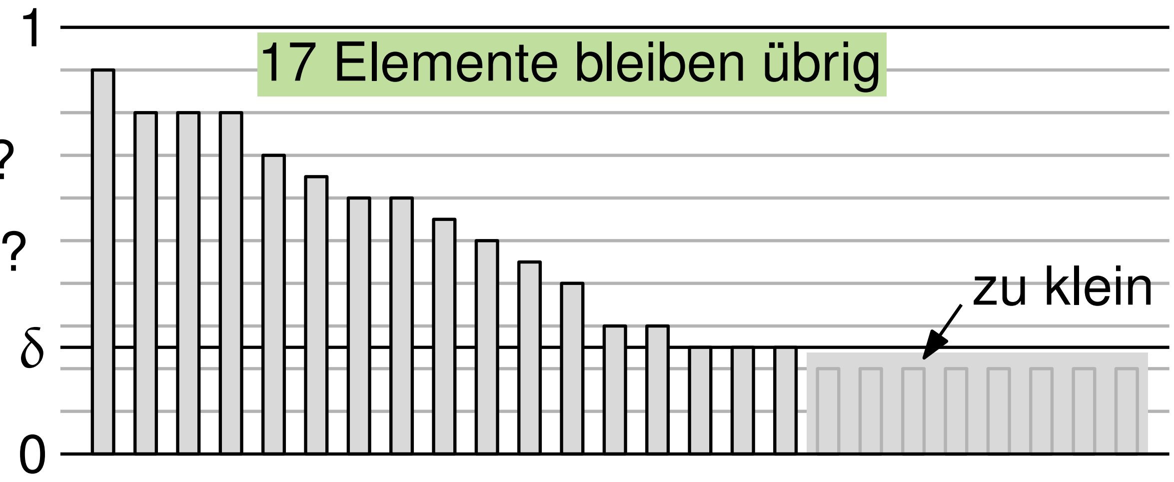
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$$



Problem 3

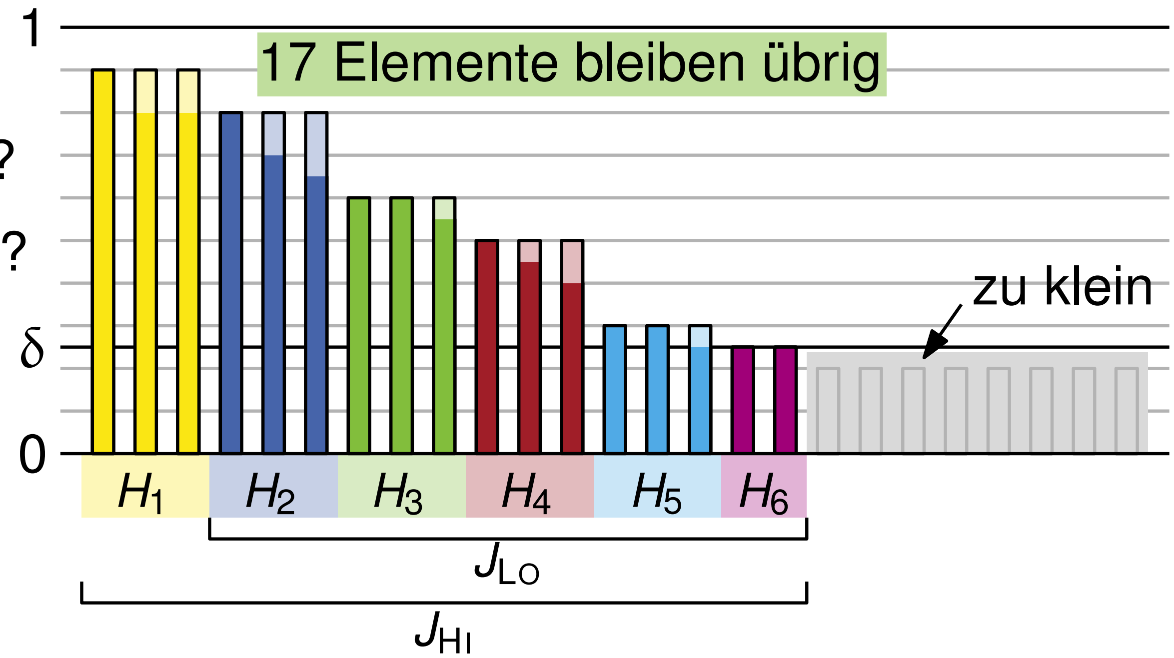
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 3

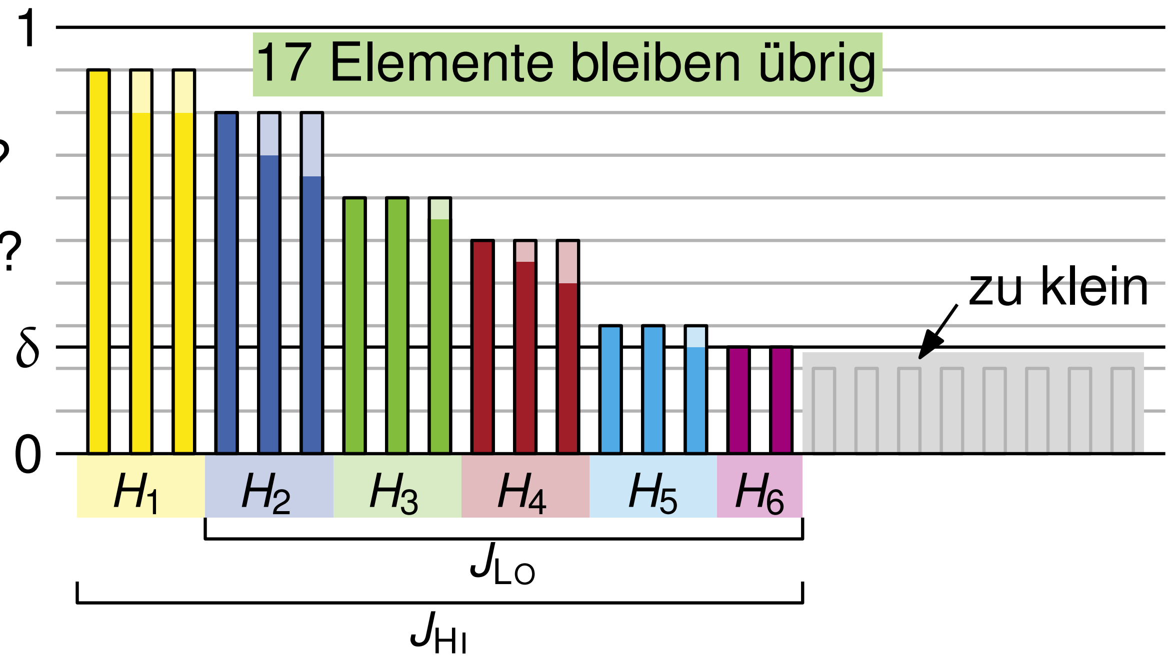
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

$$\delta = \frac{\varepsilon}{2} = 0.25$$

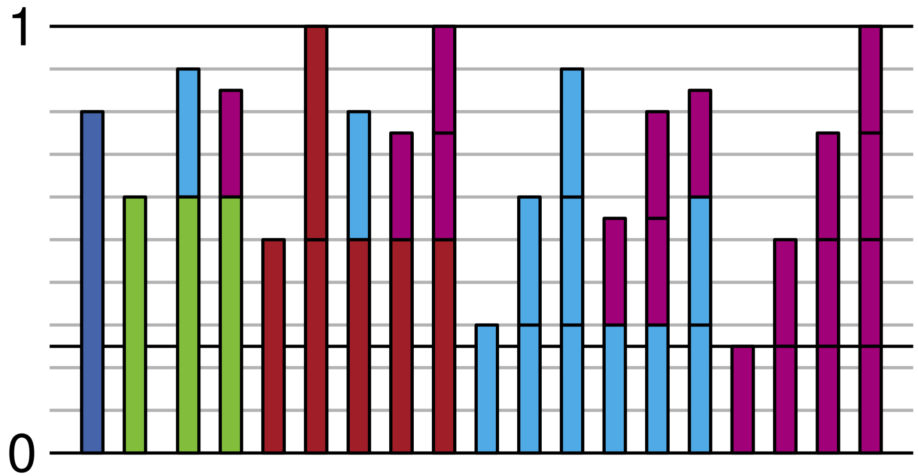
$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$$



Warum gibt es so viel mehr Bin-Typen?

mögliche Bin-Typen



Problem 3

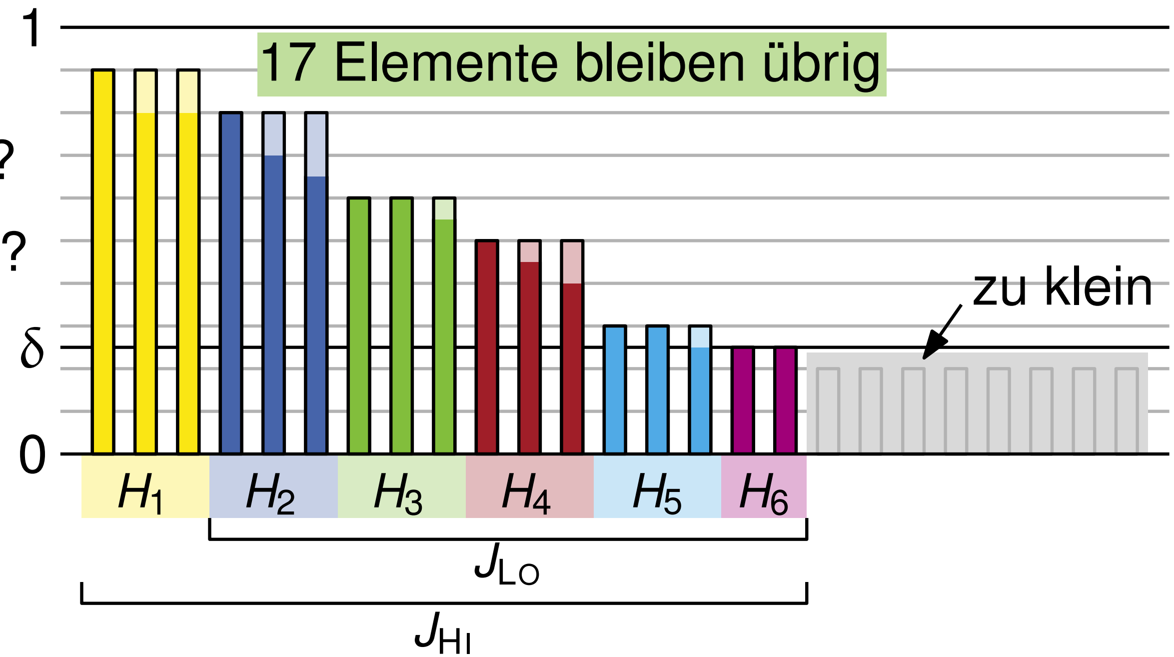
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

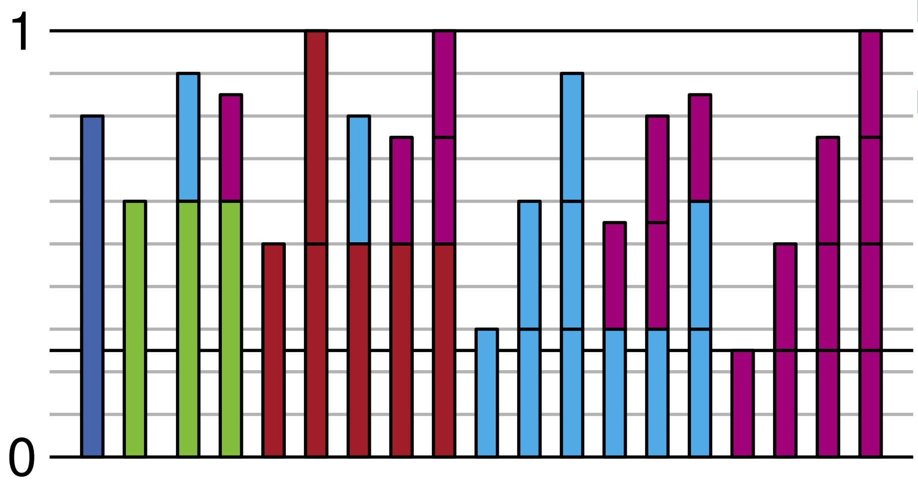
$\delta = \frac{\varepsilon}{2} = 0.25$

$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$

$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$



mögliche Bin-Typen



Warum gibt es so viel mehr Bin-Typen?

- Es gibt mehr verschiedene Elementgrößen.
- Es gibt kleinere Elementgrößen.
  (kleine Elemente passen „überall noch irgendwie dazu“ $\rightarrow$ mehr Kombinationen)

Problem 3

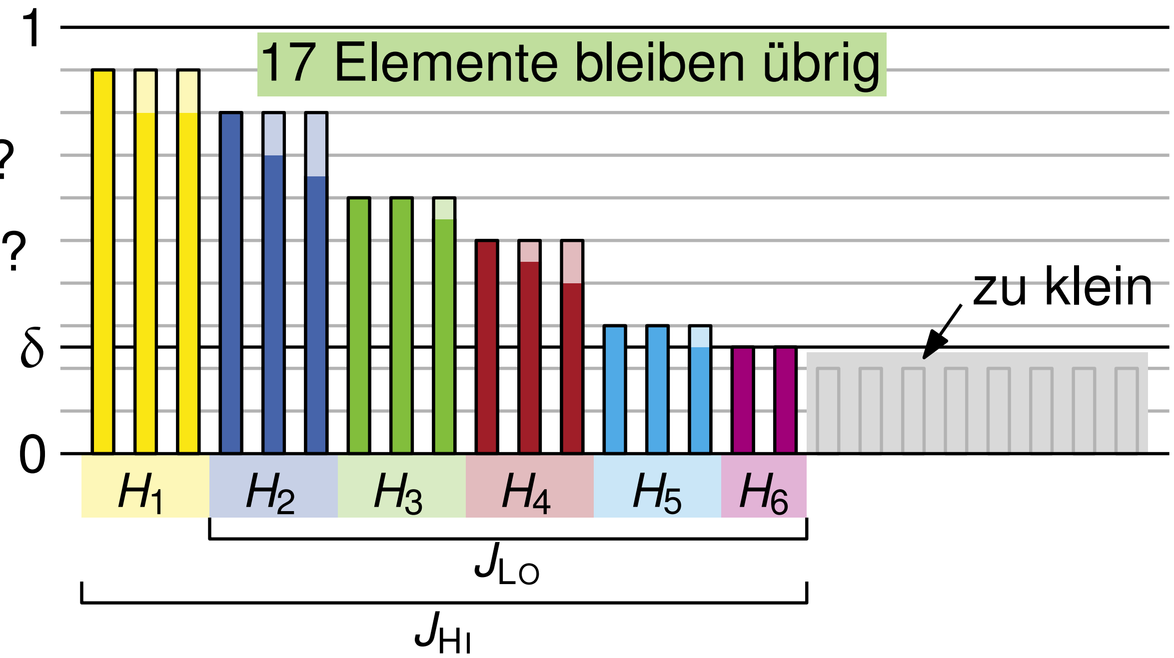
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

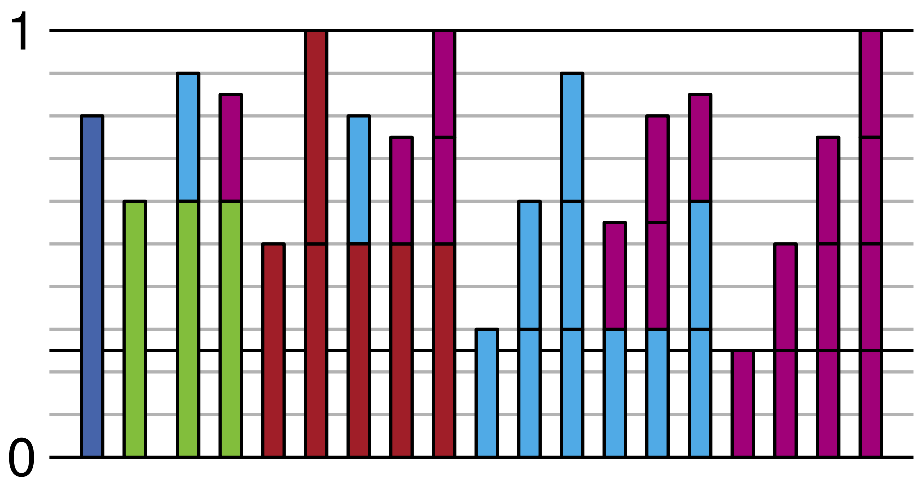
$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$$

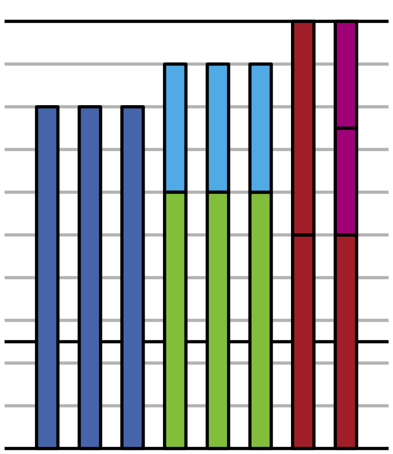
$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$$



mögliche Bin-Typen



opt. Lsg für J_{Lo}



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 3

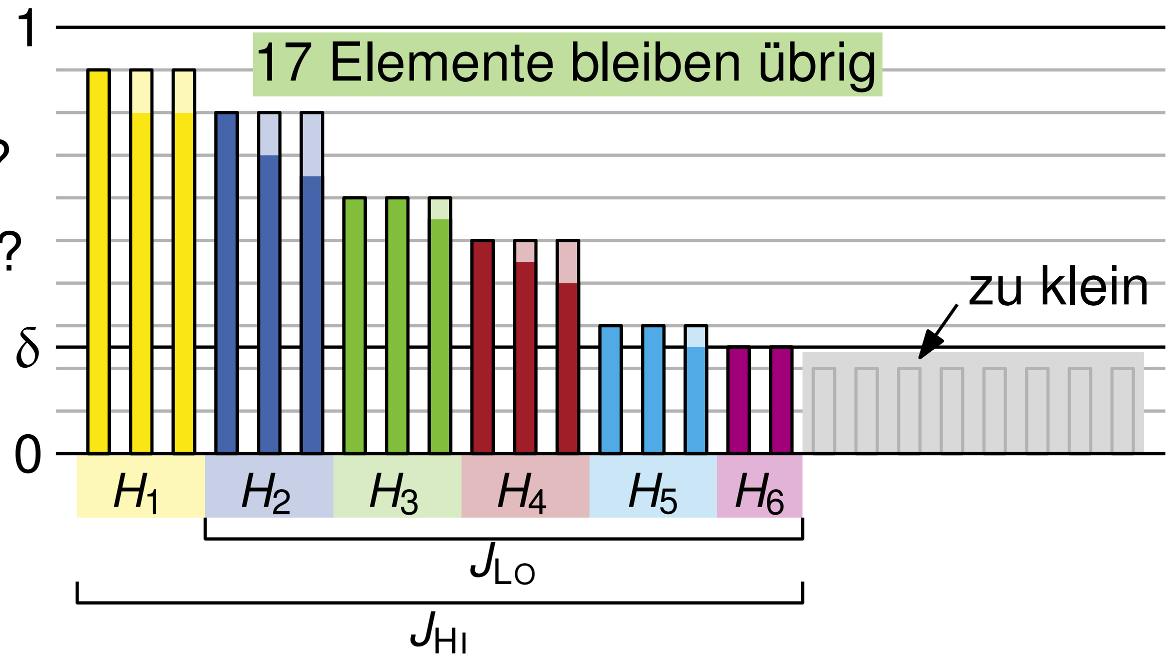
Wähle nun $\varepsilon = 0.5$.

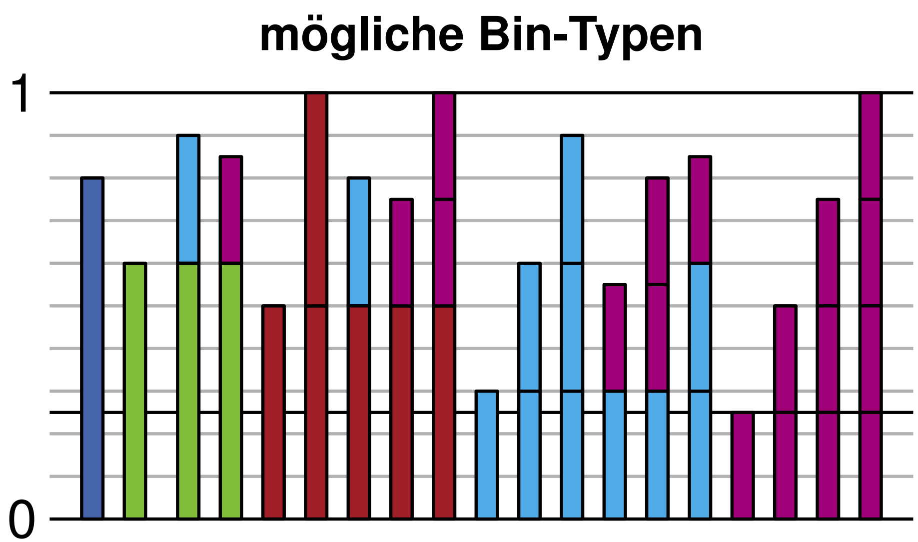
- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

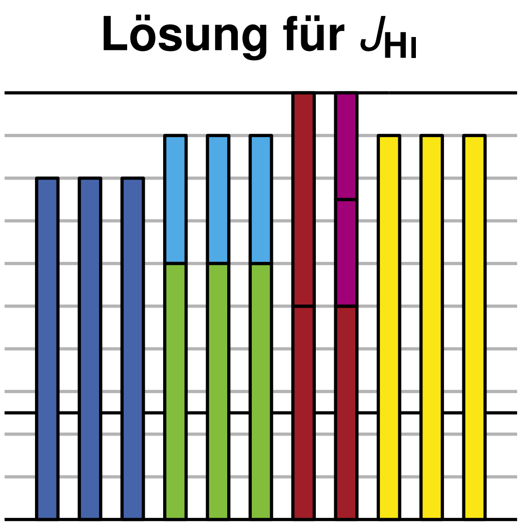
$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.125 \cdot \boxed{17} \rceil = 3$$

$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{17}}{3} \right\rfloor = 5$$







Problem 3

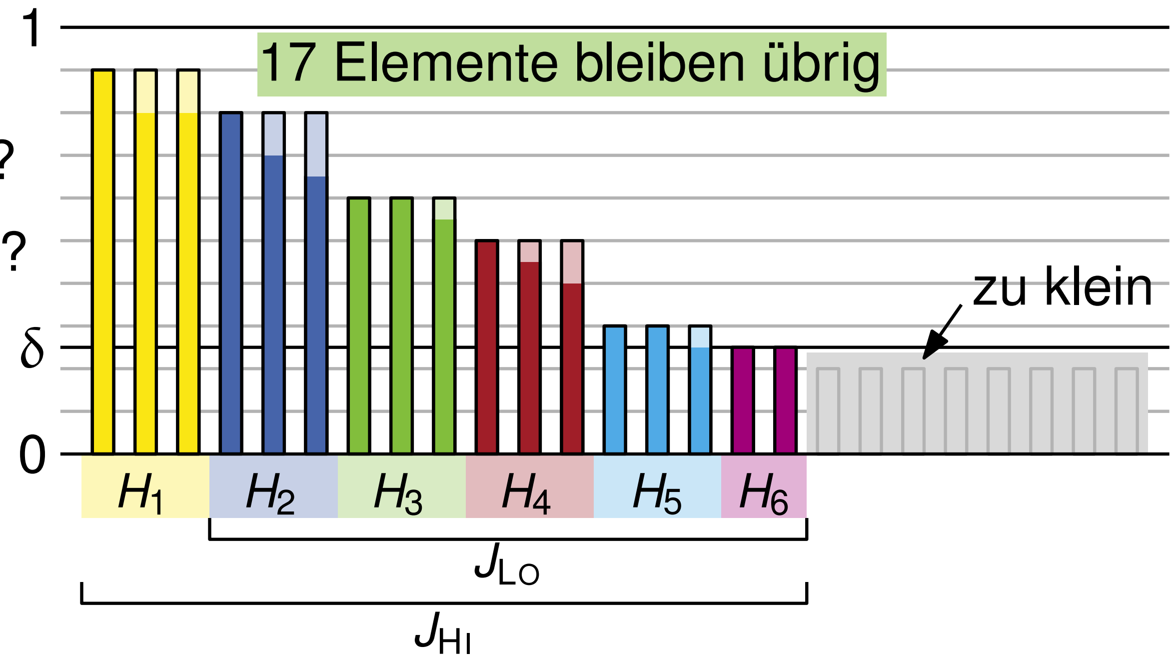
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

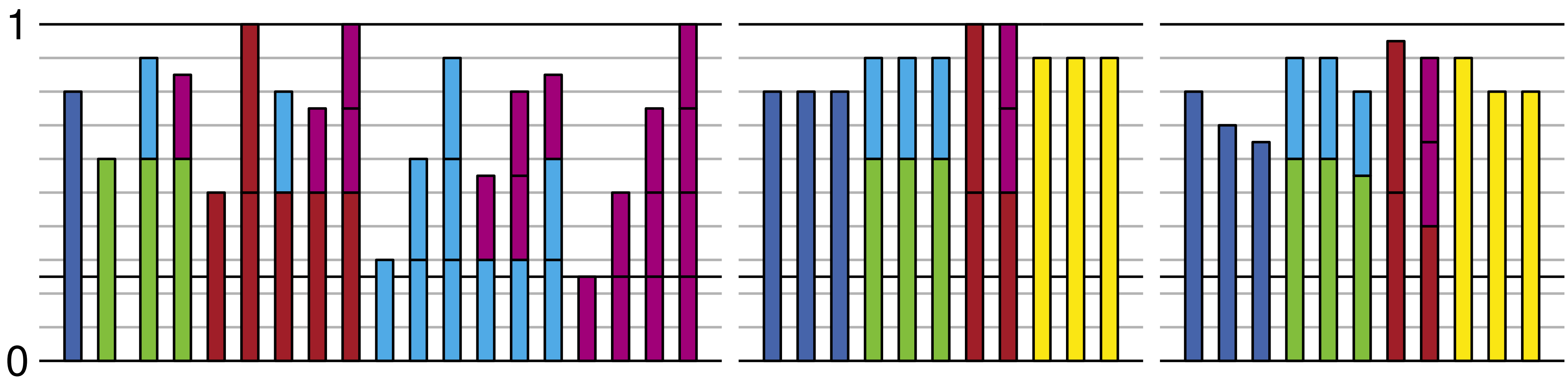
$\delta = \frac{\varepsilon}{2} = 0.25$

$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$

$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$



mögliche Bin-Typen **Lösung für J_{HI}** **Lösung für J**



Problem 3

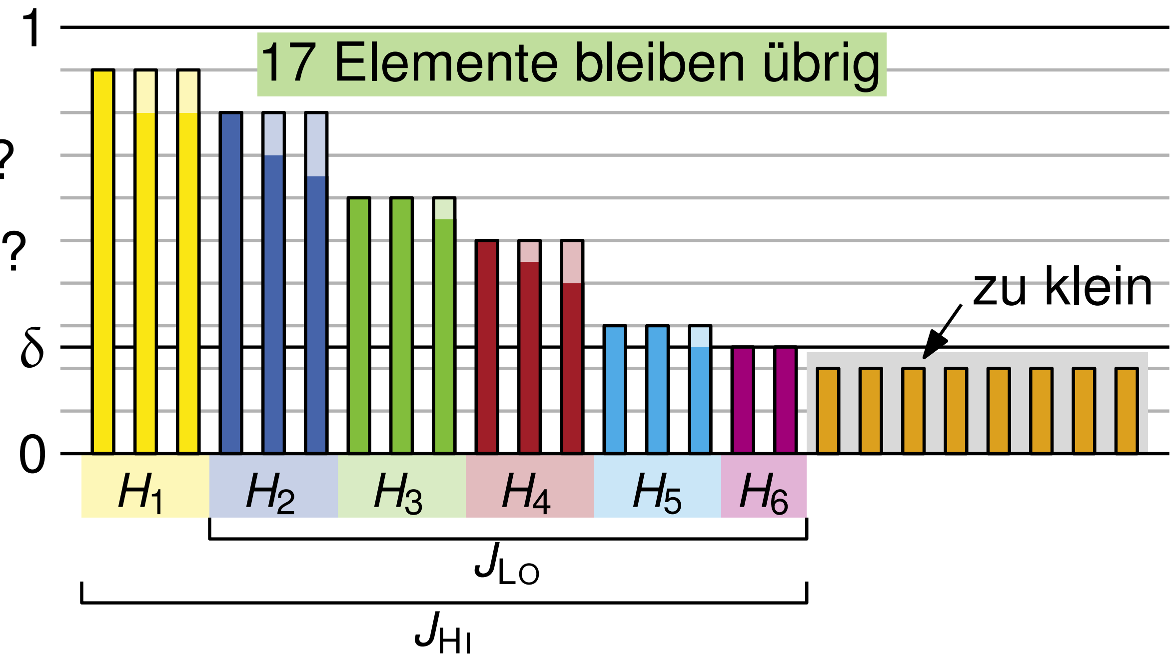
Wähle nun $\varepsilon = 0.5$.

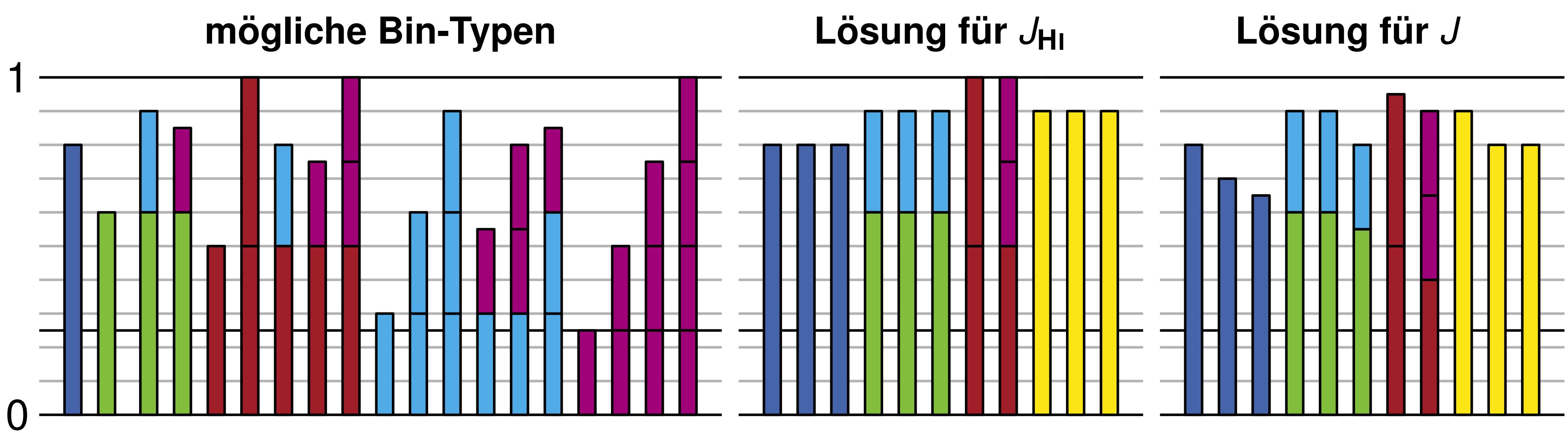
- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$$





Problem 3

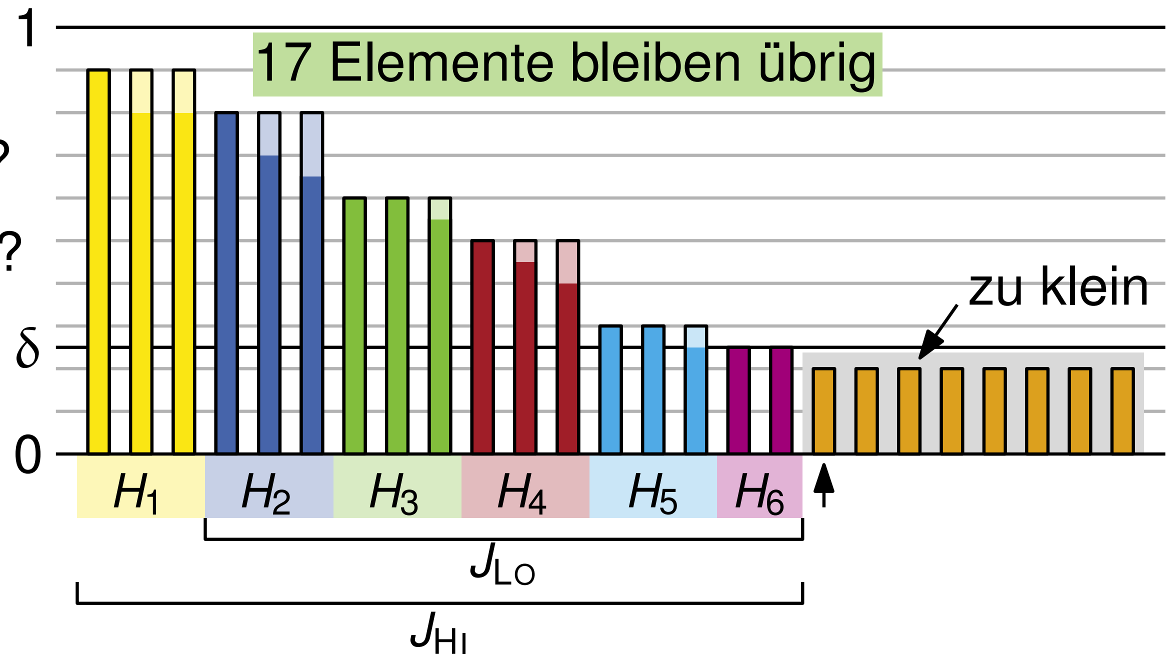
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

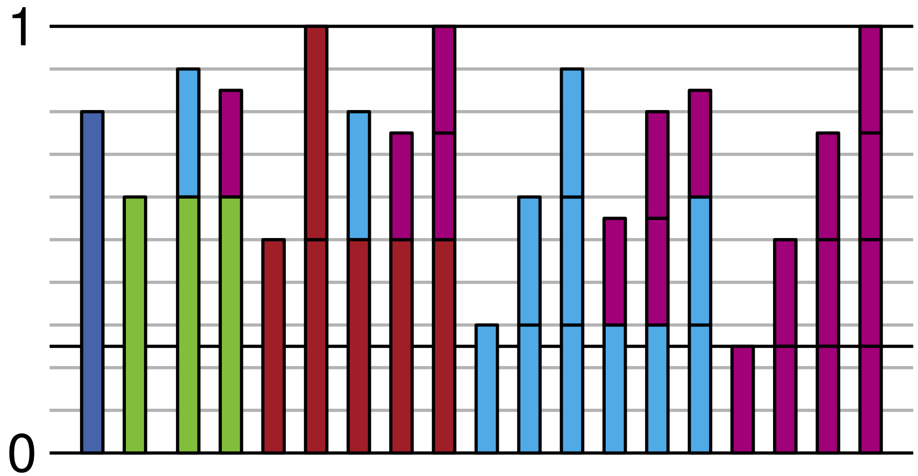
$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$$

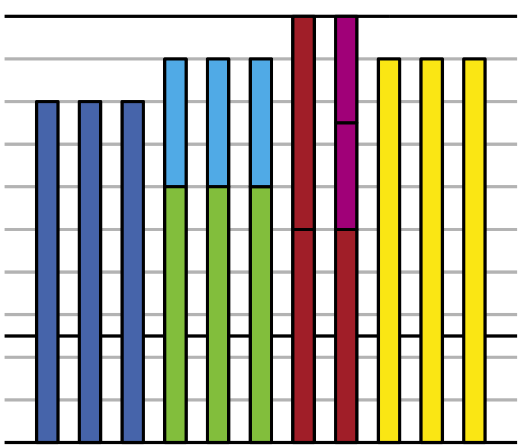
$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$$



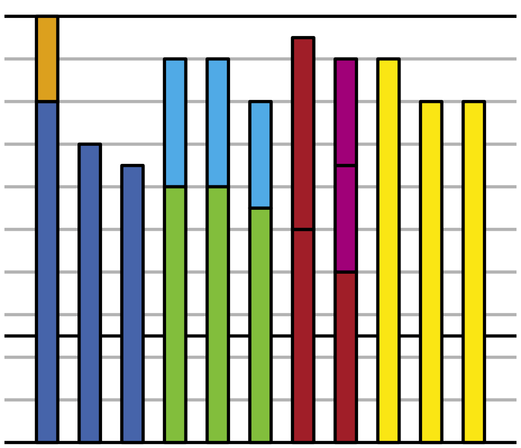
mögliche Bin-Typen



Lösung für J_{HI}



Lösung für J



Problem 3

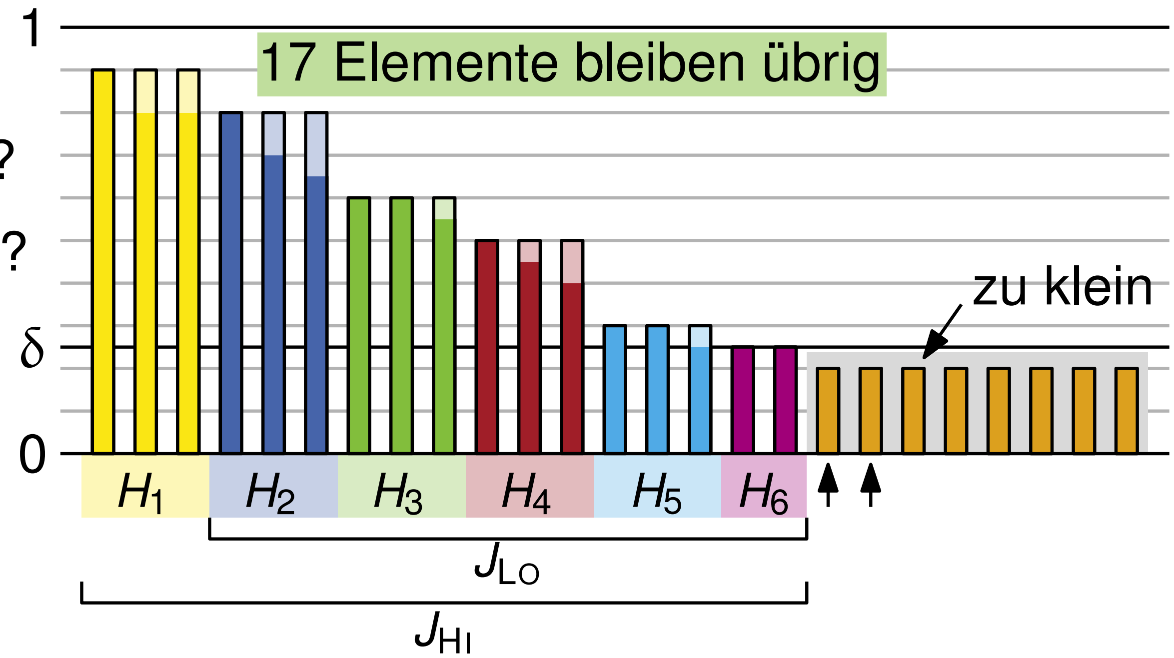
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

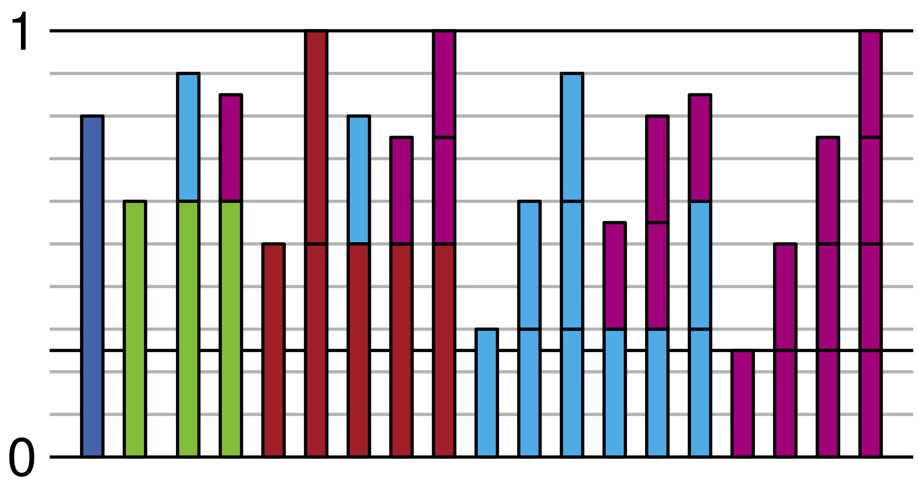
$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.125 \cdot \boxed{17} \rceil = 3$$

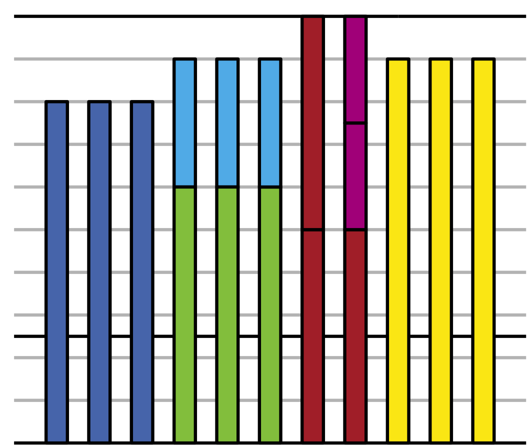
$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{17}}{3} \right\rfloor = 5$$



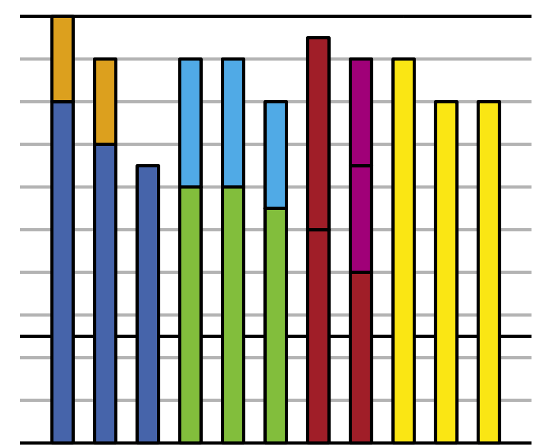
mögliche Bin-Typen



Lösung für J_{HI}



Lösung für J



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 3

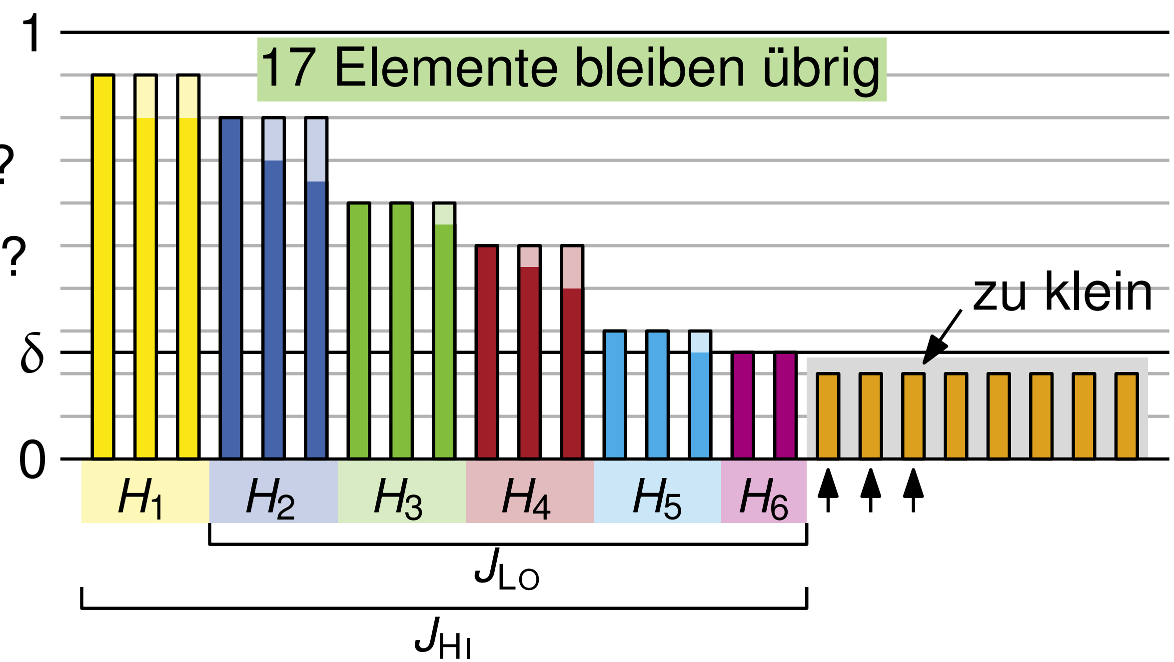
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

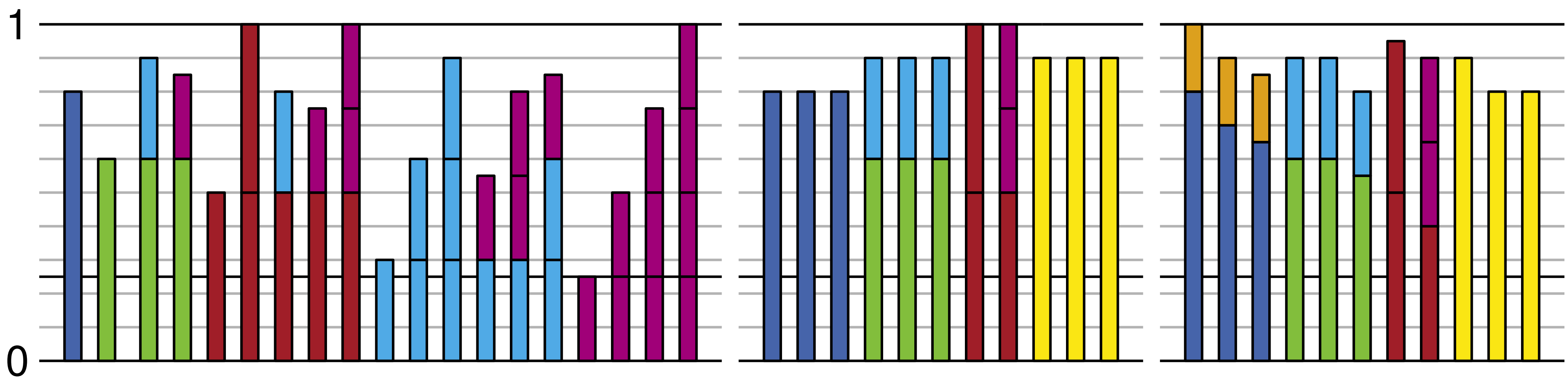
$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$$

$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$$



mögliche Bin-Typen **Lösung für J_{HI}** **Lösung für J**



Problem 3

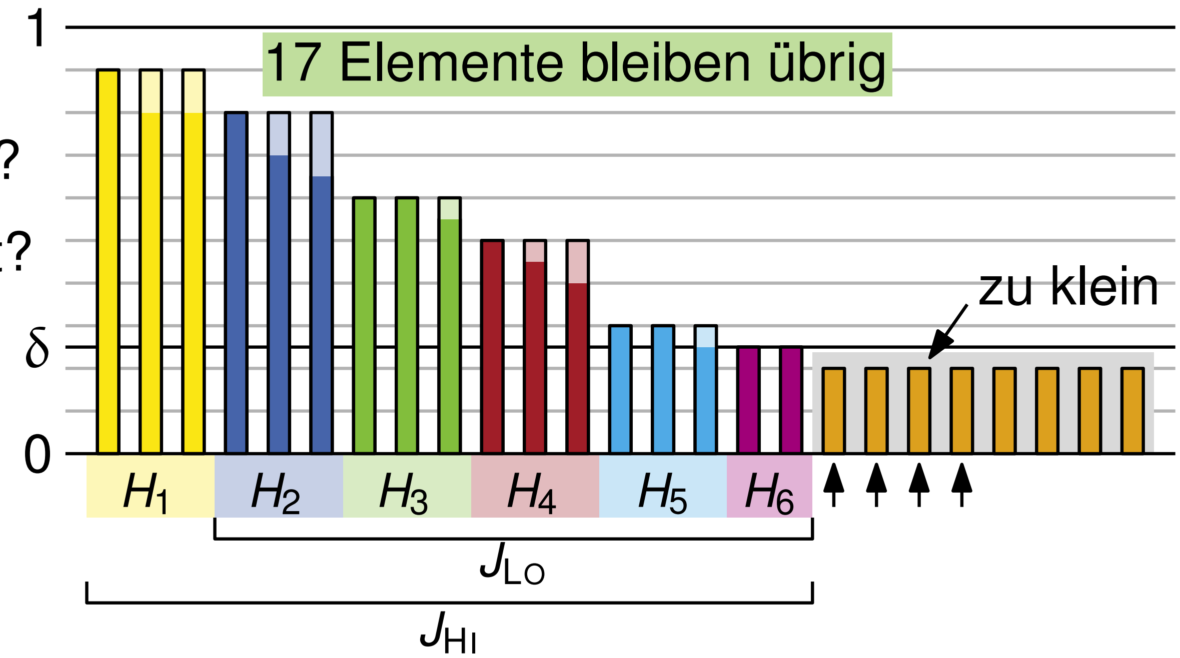
Wähle nun $\varepsilon = 0.5$.

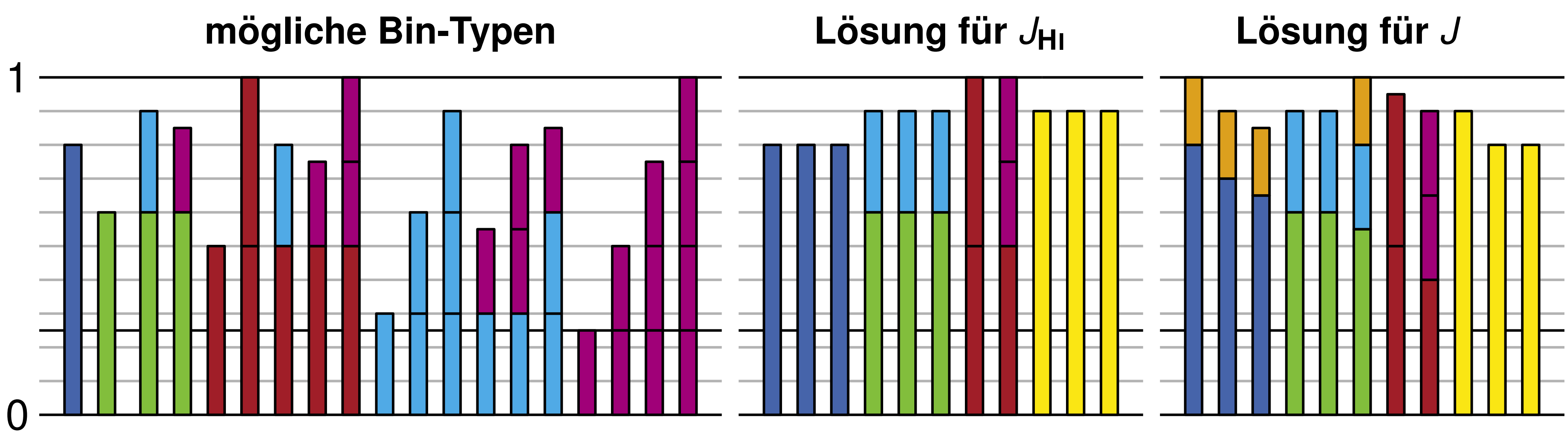
- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

$\delta = \frac{\varepsilon}{2} = 0.25$

$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$

$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$





Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 3

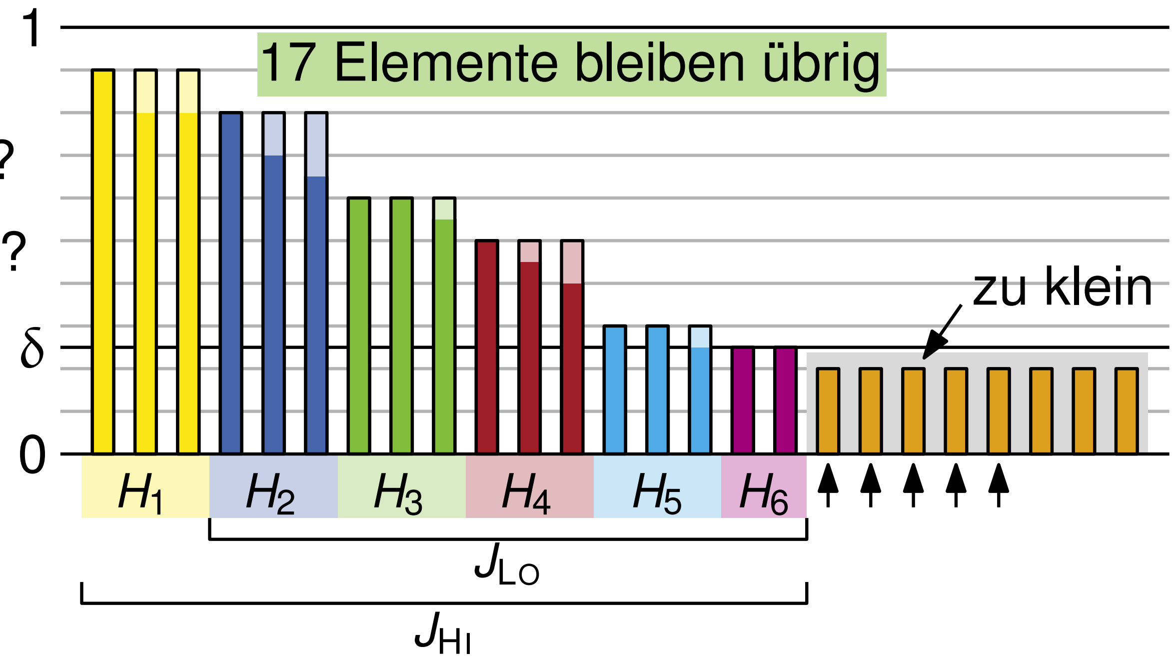
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

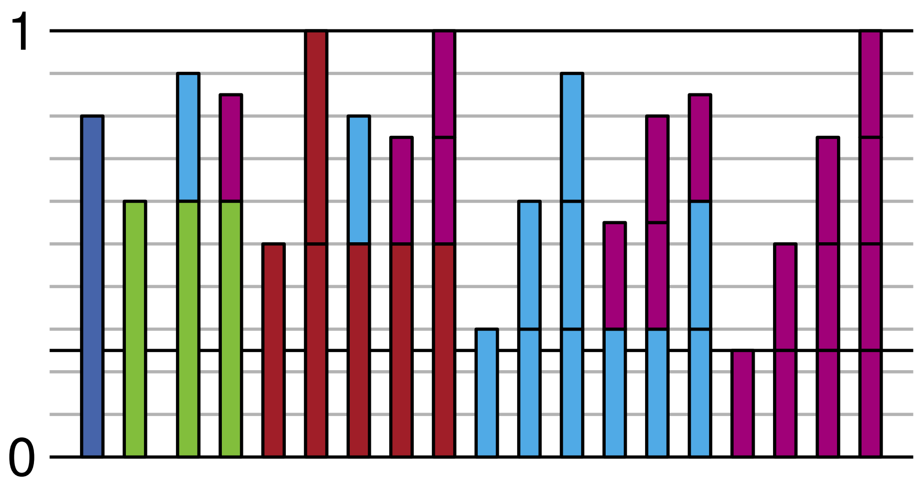
$\delta = \frac{\varepsilon}{2} = 0.25$

$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$

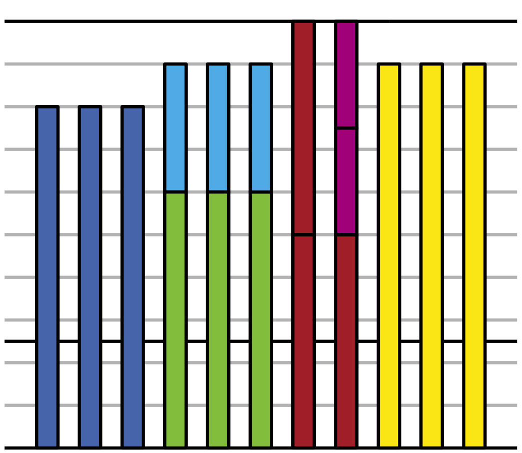
$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$



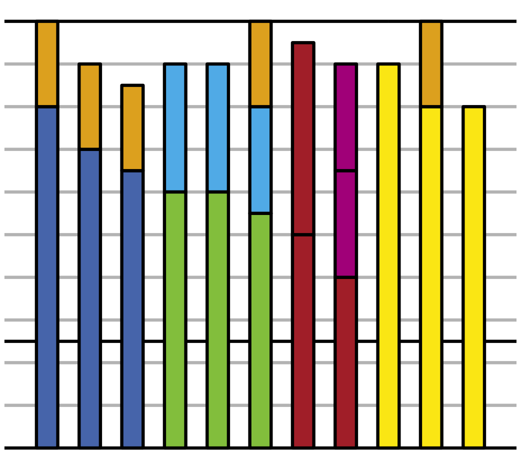
mögliche Bin-Typen



Lösung für J_{HI}



Lösung für J



Problem 3

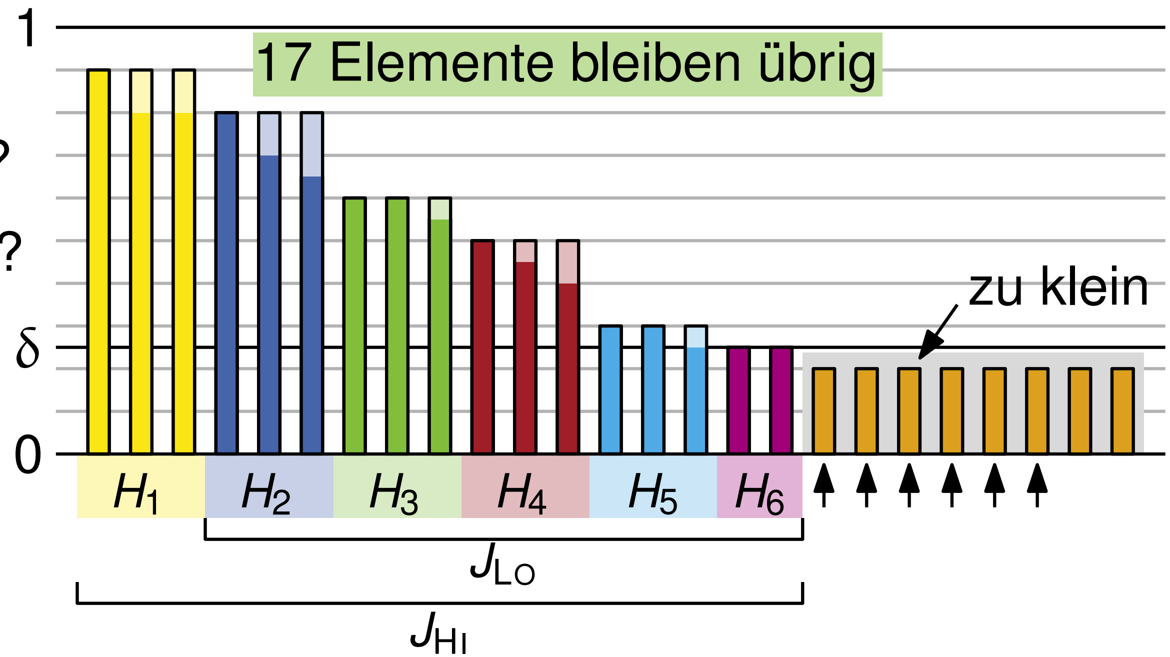
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

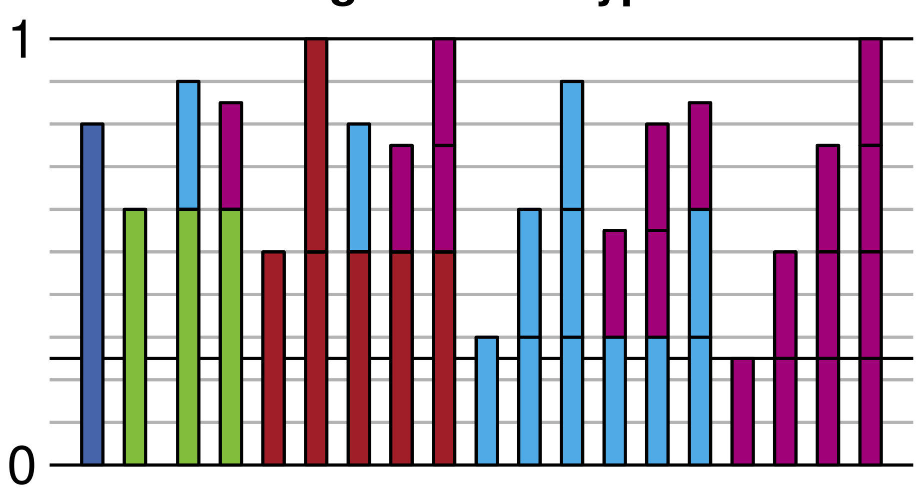
$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot n' \right\rceil = \lceil 0.125 \cdot 17 \rceil = 3$$

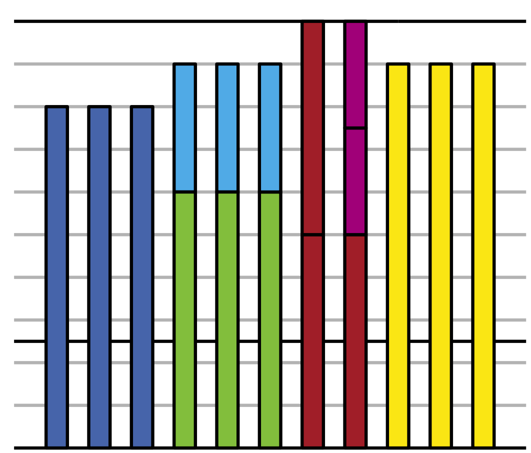
$$m = \left\lfloor \frac{n'}{k} \right\rfloor = \left\lfloor \frac{17}{3} \right\rfloor = 5$$



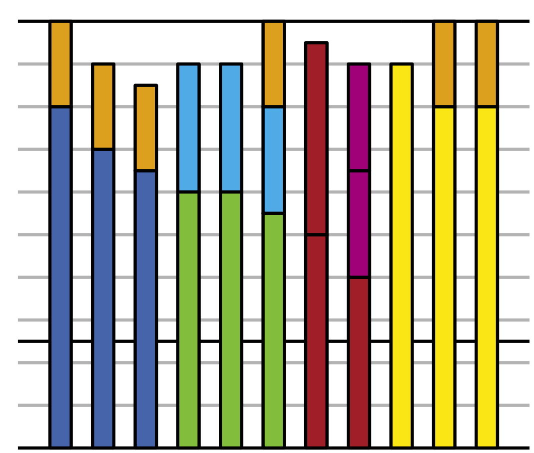
mögliche Bin-Typen



Lösung für J_{HI}



Lösung für J



Problem 3

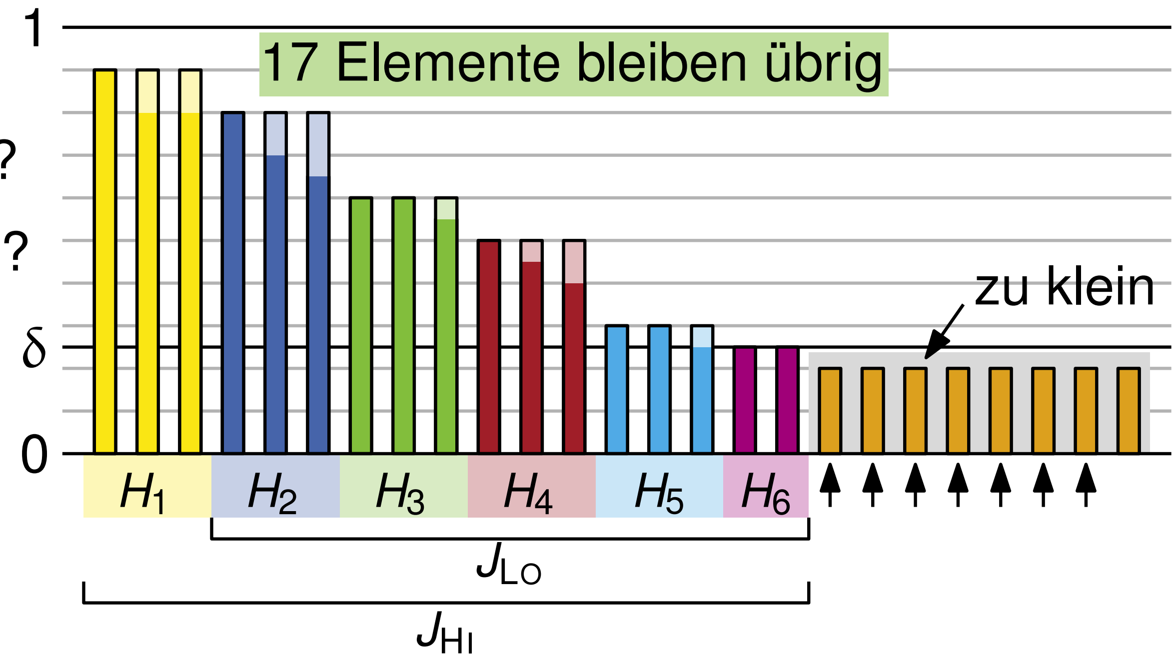
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

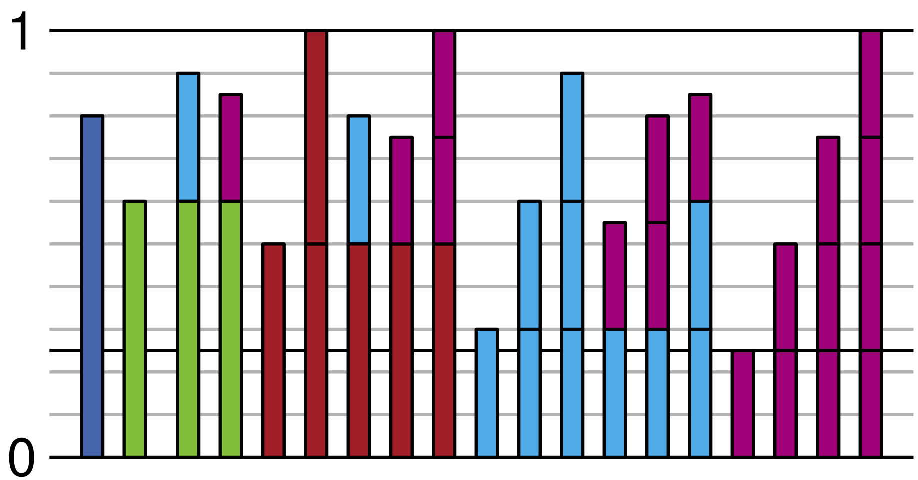
$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.125 \cdot \boxed{17} \rceil = 3$$

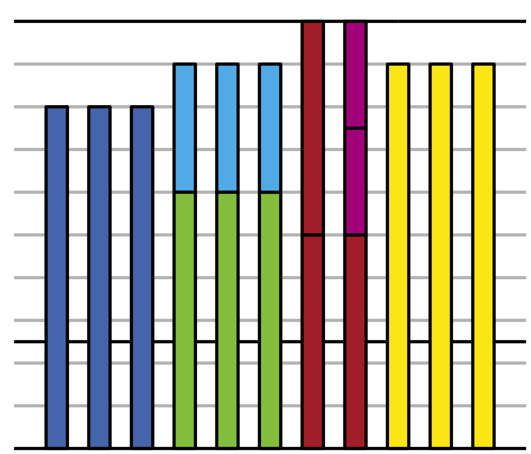
$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{17}}{3} \right\rfloor = 5$$



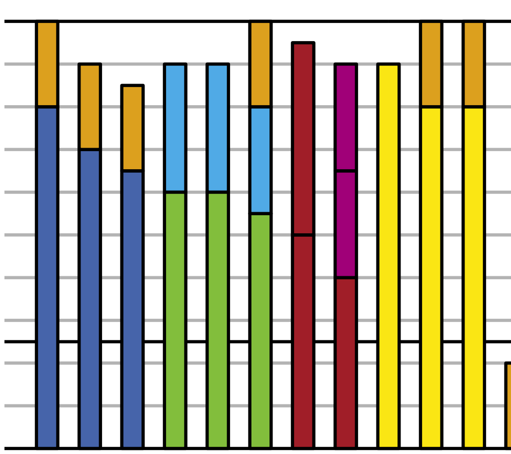
mögliche Bin-Typen



Lösung für J_{HI}



Lösung für J



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 3

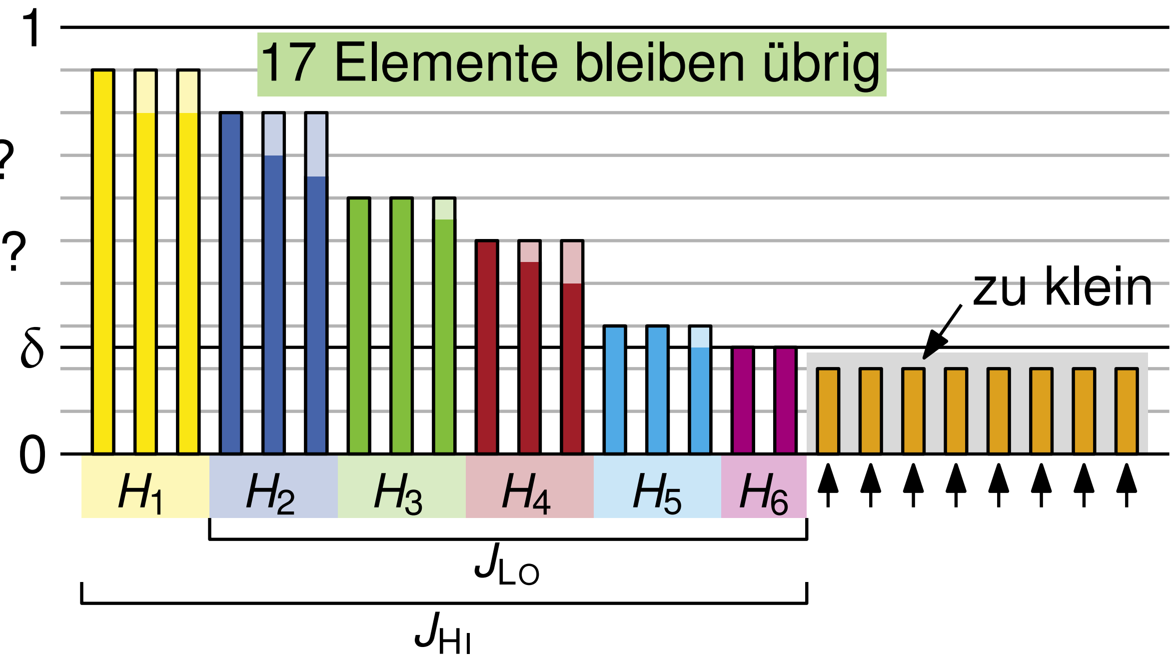
Wähle nun $\varepsilon = 0.5$.

- Warum ändert sich die Laufzeit?
- Was ändert sich an der Qualität?

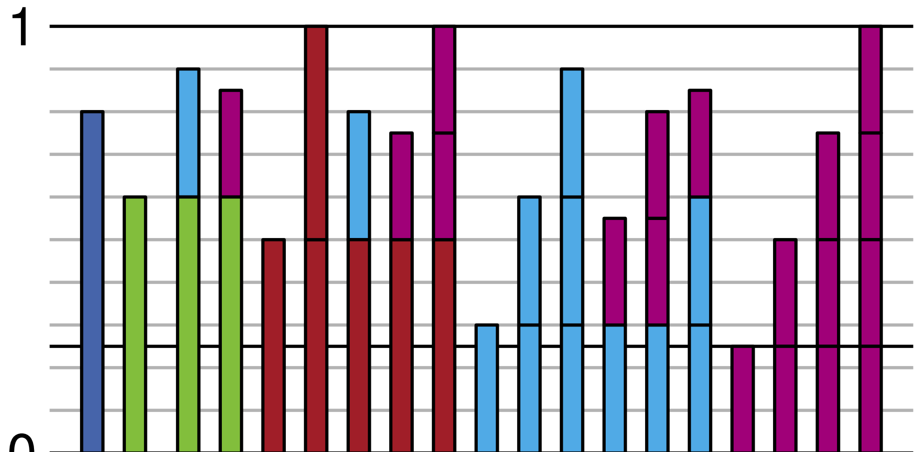
$$\delta = \frac{\varepsilon}{2} = 0.25$$

$$k = \left\lceil \frac{\varepsilon^2}{2} \cdot \boxed{n'} \right\rceil = \lceil 0.125 \cdot \boxed{17} \rceil = 3$$

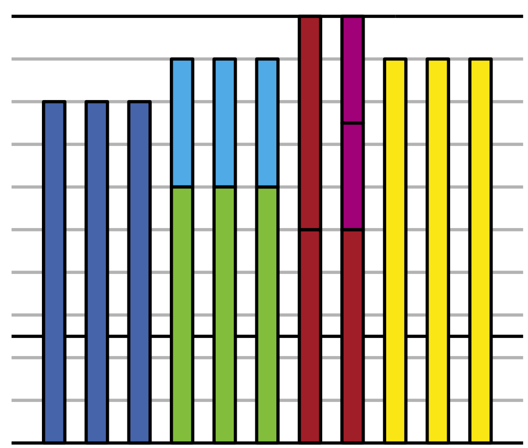
$$m = \left\lfloor \frac{\boxed{n'}}{k} \right\rfloor = \left\lfloor \frac{\boxed{17}}{3} \right\rfloor = 5$$



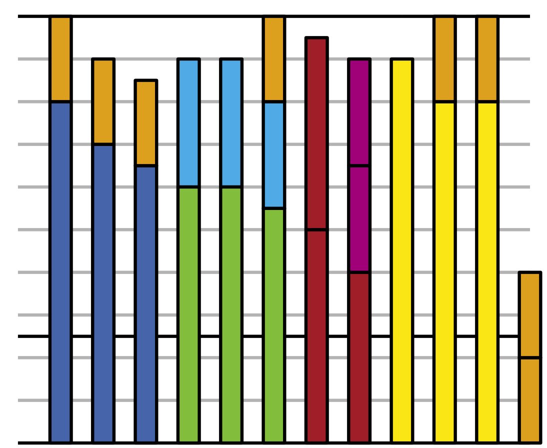
mögliche Bin-Typen



Lösung für J_{HI}



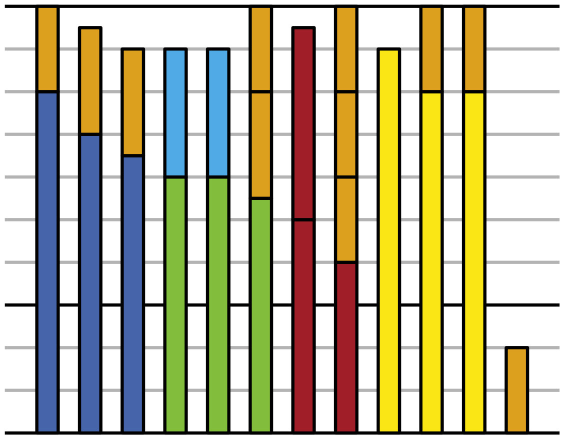
Lösung für $\cancel{J}'$



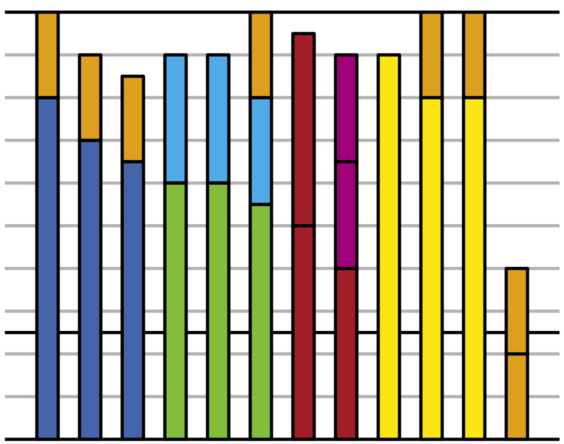
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 3

Lösung für ε = 0.6



Lösung für ε = 0.5

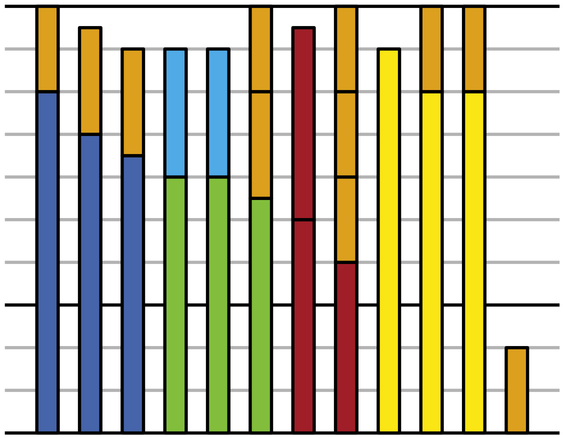


Bonusfrage: Geht es besser?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

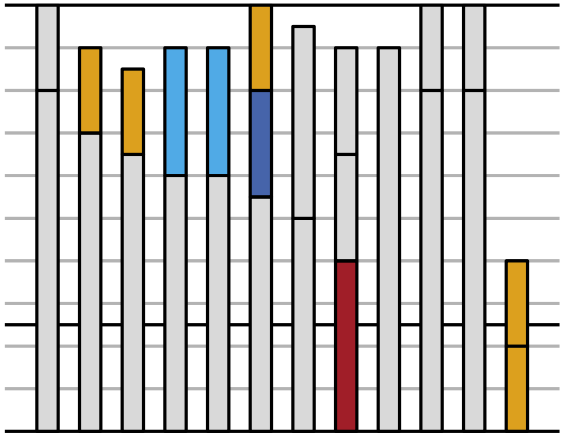
Problem 3

Lösung für $\varepsilon = 0.6$



Bonusfrage: Geht es besser?

Lösung für $\varepsilon = 0.5$



optimale Lösung