UNIVERSITÄT ZU LÜBECK

Fully Dynamic Bin Packing Revisited

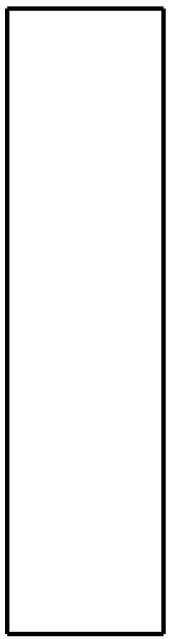
Sebastian Berndt [1] Klaus Jansen [2] Kim-Manuel Klein [2]

[1] Institute of Theoretical Computer Science, Universität zu Lübeck

[2] Department of Computer Science, Christian-Albrechts-University to Kiel

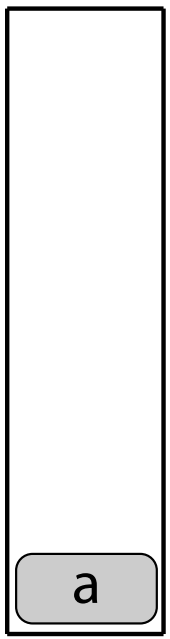
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2



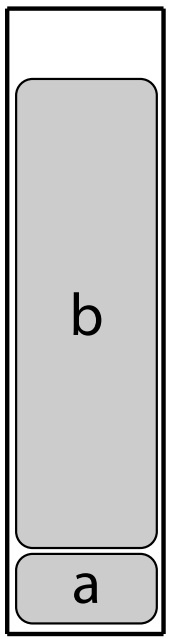
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7



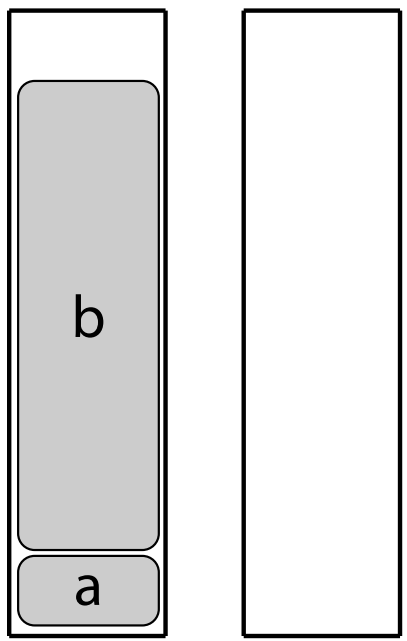
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4



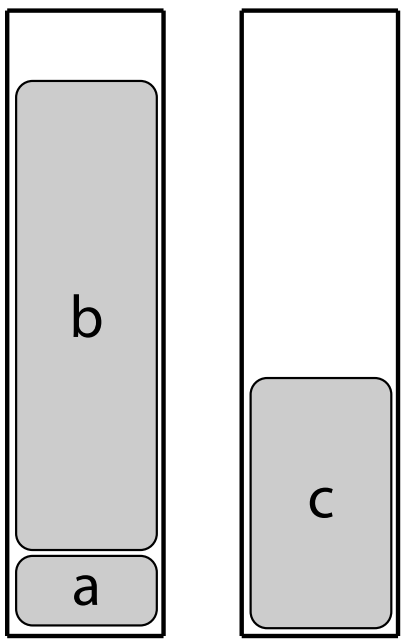
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4



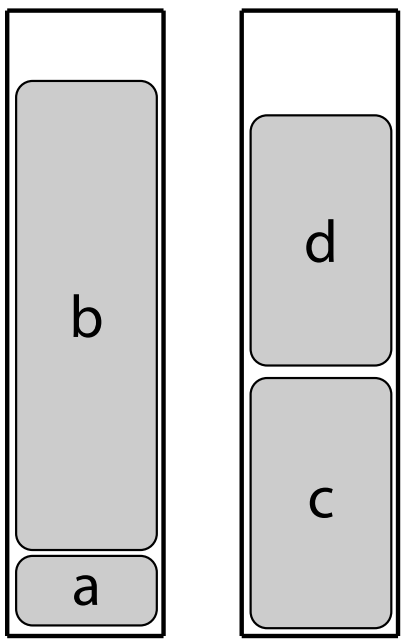
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4, **INSERT:** *d*/0.4



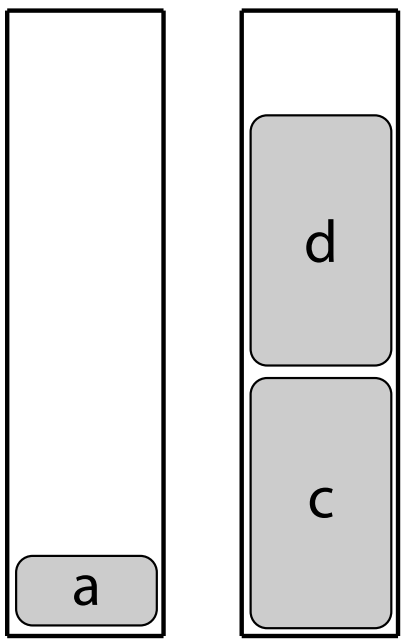
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4, **INSERT:** *d*/0.4, **REMOVE:** *b*/0.7



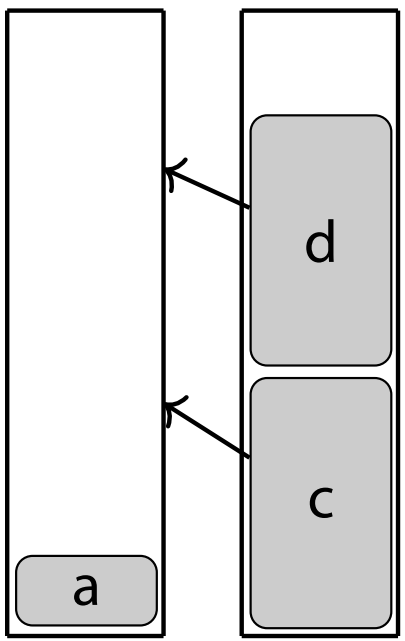
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4, **INSERT:** *d*/0.4, **REMOVE:** *b*/0.7



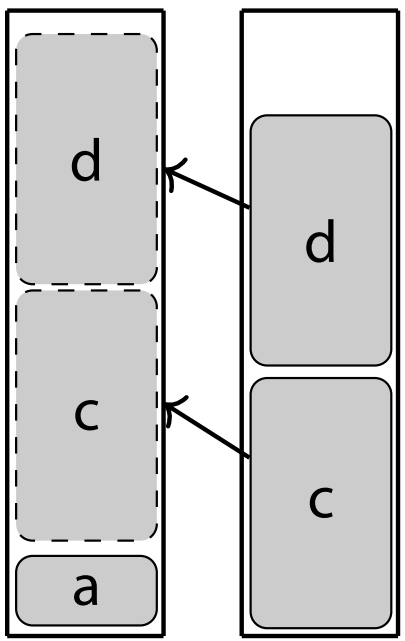
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4, **INSERT:** *d*/0.4, **REMOVE:** *b*/0.7



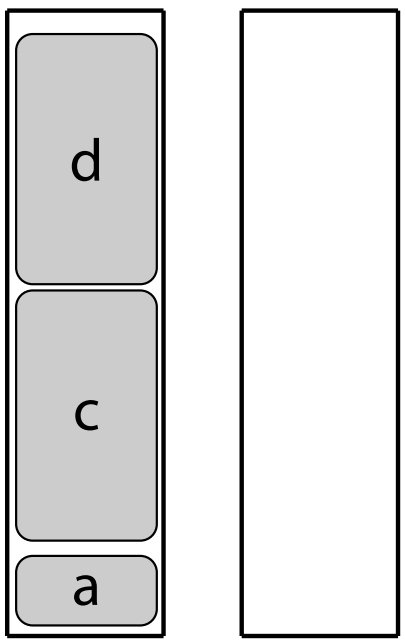
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4, **INSERT:** *d*/0.4, **REMOVE:** *b*/0.7



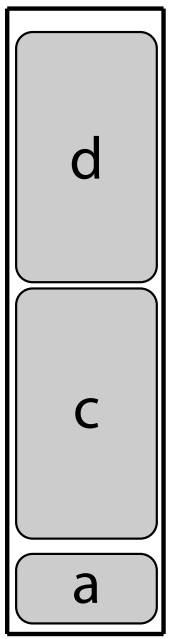
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4, **INSERT:** *d*/0.4, **REMOVE:** *b*/0.7



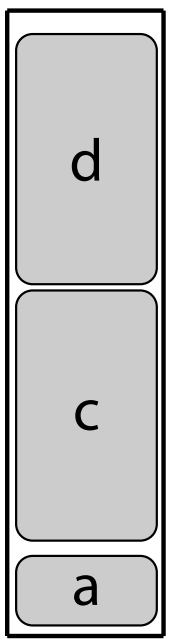
Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4, **INSERT:** *d*/0.4, **REMOVE:** *b*/0.7



Fully Dynamic Bin Packing = Online + Removal + Repacking

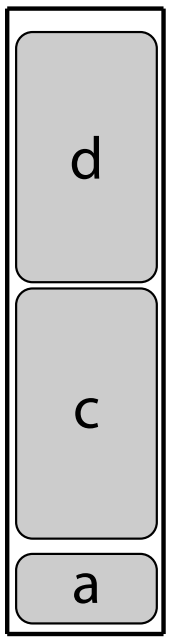
INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4, **INSERT:** *d*/0.4, **REMOVE:** *b*/0.7,



$$\textit{Migration Factor} = \frac{c + d}{b} = \frac{\text{SIZE(moved)}}{\text{SIZE(new/removed)}}$$

Fully Dynamic Bin Packing = Online + Removal + Repacking

INSERT: *a*/0.2, **INSERT:** *b*/0.7, **INSERT:** *c*/0.4, **INSERT:** *d*/0.4, **REMOVE:** *b*/0.7,



$$\textit{Migration Factor} = \frac{c + d}{b} = \frac{\text{SIZE(moved)}}{\text{SIZE(new/removed)}}$$

Shifting Moves = 2 = # moved

Lower Bound

MF of $\Omega(1/\varepsilon)$ is necessary for rate $1+\varepsilon$

$a = 1/2 - b/3, \quad b = 1/MF$

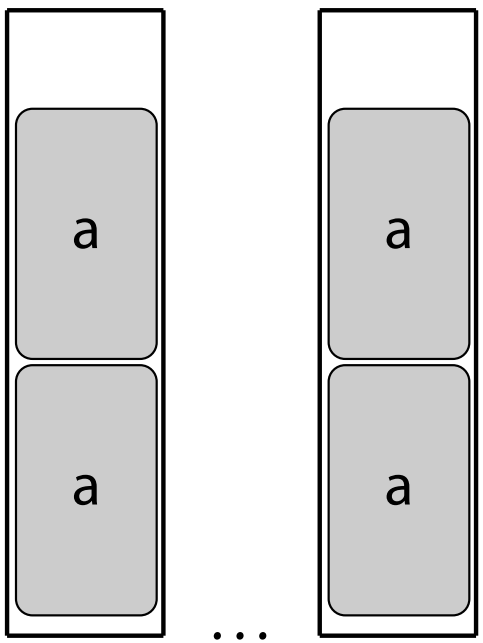
INSERT: a^+

Lower Bound

MF of $\Omega(1/\varepsilon)$ is necessary for rate $1+\varepsilon$

$a = 1/2 - b/3, \quad b = 1/MF$

INSERT: a^+



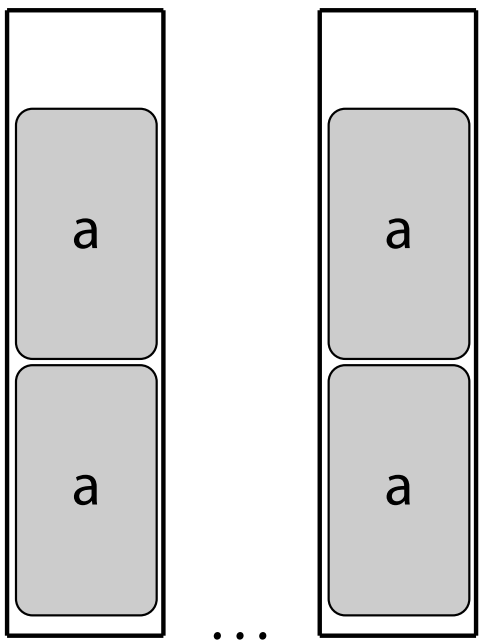
Lower Bound

MF of $\Omega(1/\varepsilon)$ is necessary for rate $1+\varepsilon$

$a = 1/2 - b/3, \quad b = 1/MF$

INSERT: a^+,

INSERT: b^+



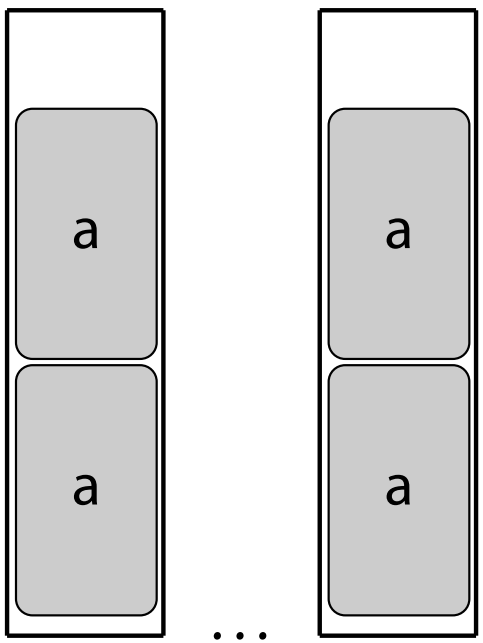
Lower Bound

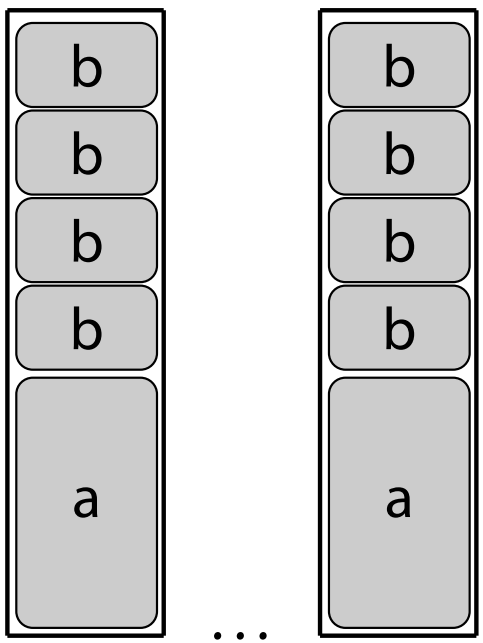
MF of $\Omega(1/\varepsilon)$ is necessary for rate $1+\varepsilon$

$a = 1/2 - b/3, \quad b = 1/MF$

INSERT: a^+,

INSERT: b^+





Fully Dynamic Bin Packing

Known Results

Rate	Shifting Moves	Migration Factor	Authors

Fully Dynamic Bin Packing

Known Results

Rate	Shifting Moves	Migration Factor	Authors
5/4	7 (amortized)	∞	Ivković, Lloyd (1998)

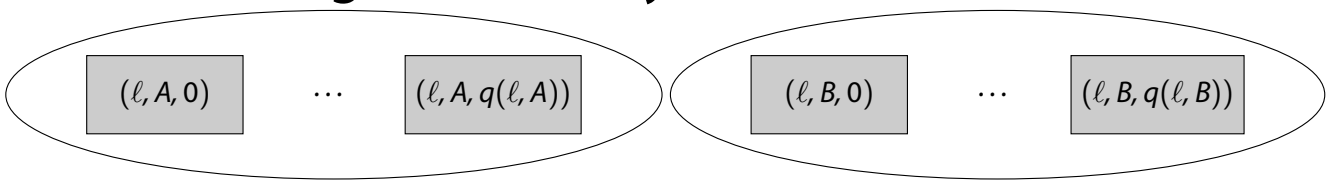
Fully Dynamic Bin Packing

Known Results

Rate	Shifting Moves	Migration Factor	Authors
5/4	7 (amortized)	∞	Ivković, Lloyd (1998)
$1+\varepsilon$	$\mathcal{O}(1/\varepsilon^4 \cdot \log(1/\varepsilon))$	$\mathcal{O}(1/\varepsilon^4 \cdot \log(1/\varepsilon))$	*B., Jansen, Klein (2015)*

Large Items

Round Items geometrically:

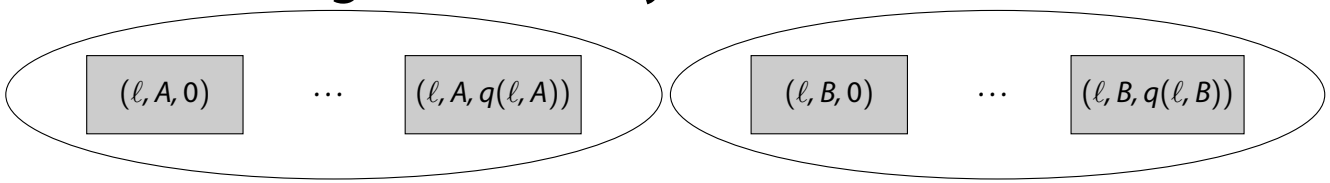


$s \in [2^{-(\ell+1)}, 2^{-\ell})$

- $|(\ell, A, \cdot)| = 2^{\ell} \cdot \text{SIZE}, \quad |(\ell, B, \cdot)| = 2^{\ell} \cdot (\text{SIZE} - 1)$

Large Items

Round Items geometrically:

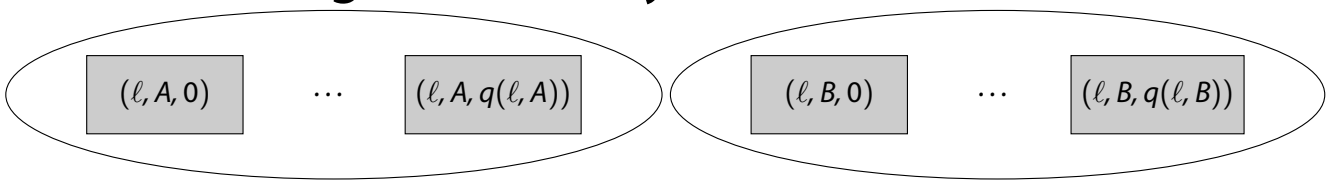


$s \in [2^{-(\ell+1)}, 2^{-\ell})$

- $|(\ell, A, \cdot)| = 2^{\ell} \cdot \text{SIZE}, \quad |(\ell, B, \cdot)| = 2^{\ell} \cdot (\text{SIZE} - 1)$
- Pack rounded items via LP

Large Items

Round Items geometrically:

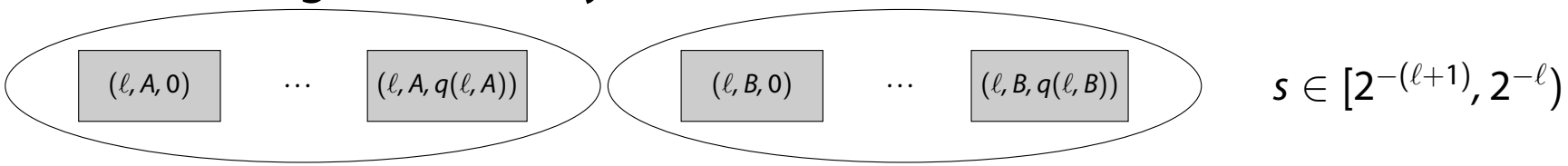


$s \in [2^{-(\ell+1)}, 2^{-\ell})$

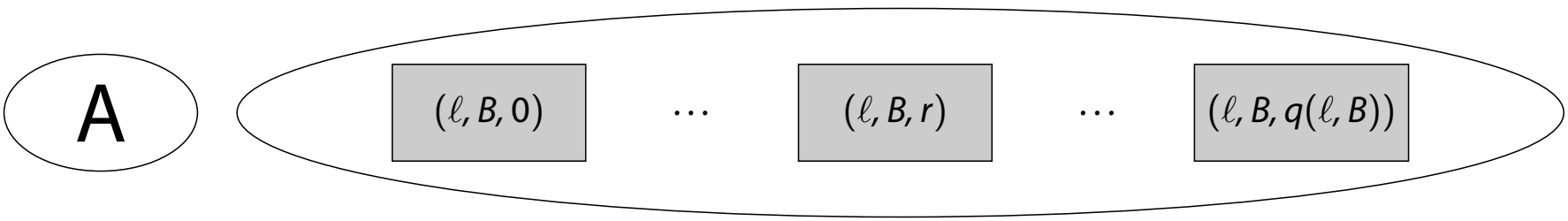
- $|(\ell, A, \cdot)| = 2^{\ell} \cdot \text{SIZE}, \quad |(\ell, B, \cdot)| = 2^{\ell} \cdot (\text{SIZE} - 1)$
- Pack rounded items via LP
- Value of SIZE may increase or decrease

Large Items

Round Items geometrically:

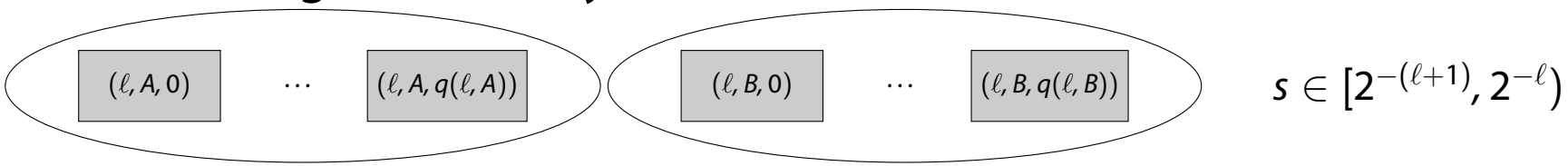


- $|(\ell, A, \cdot)| = 2^{\ell} \cdot \text{SIZE}, \quad |(\ell, B, \cdot)| = 2^{\ell} \cdot (\text{SIZE} - 1)$
- Pack rounded items via LP
- Value of SIZE may increase or decrease

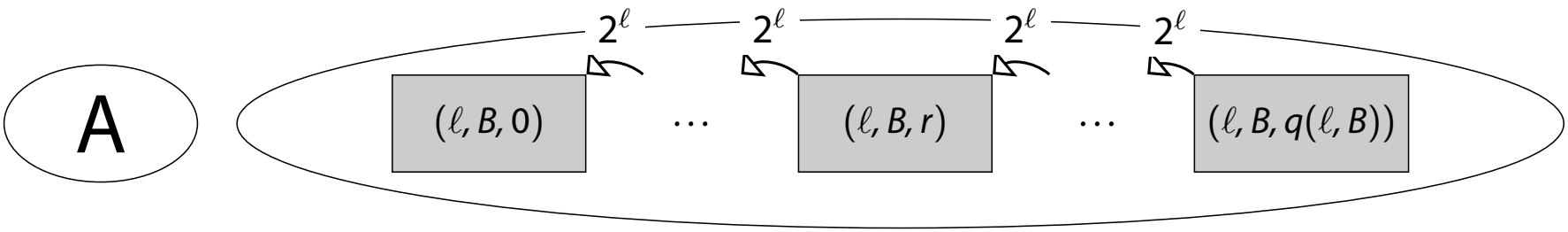


Large Items

Round Items geometrically:

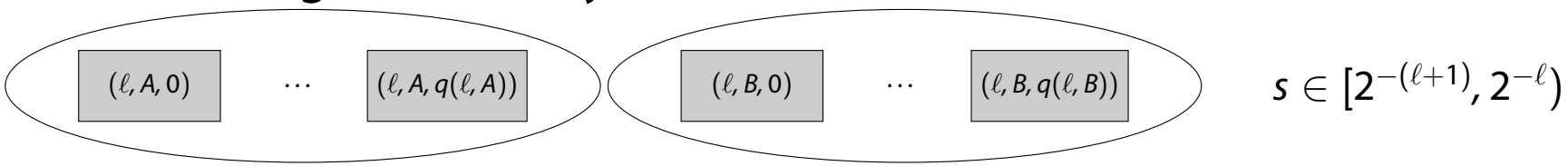


- $|(\ell, A, \cdot)| = 2^{\ell} \cdot \text{SIZE}, \quad |(\ell, B, \cdot)| = 2^{\ell} \cdot (\text{SIZE} - 1)$
- Pack rounded items via LP
- Value of SIZE may increase or decrease

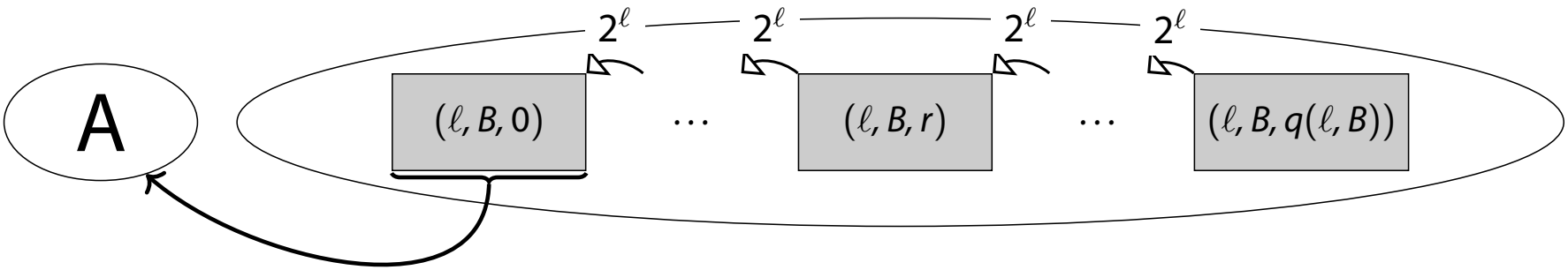


Large Items

Round Items geometrically:

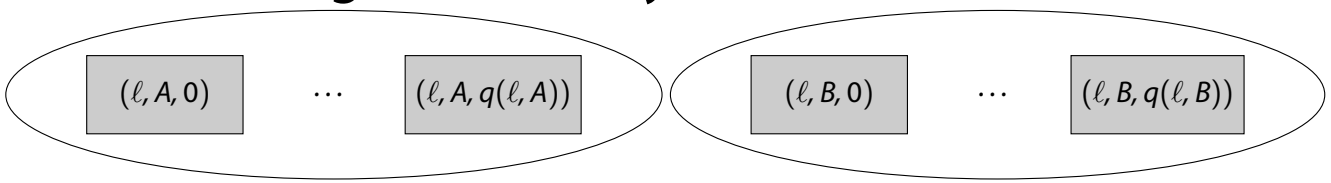


- $|(\ell, A, \cdot)| = 2^{\ell} \cdot \text{SIZE}, \quad |(\ell, B, \cdot)| = 2^{\ell} \cdot (\text{SIZE} - 1)$
- Pack rounded items via LP
- Value of SIZE may increase or decrease



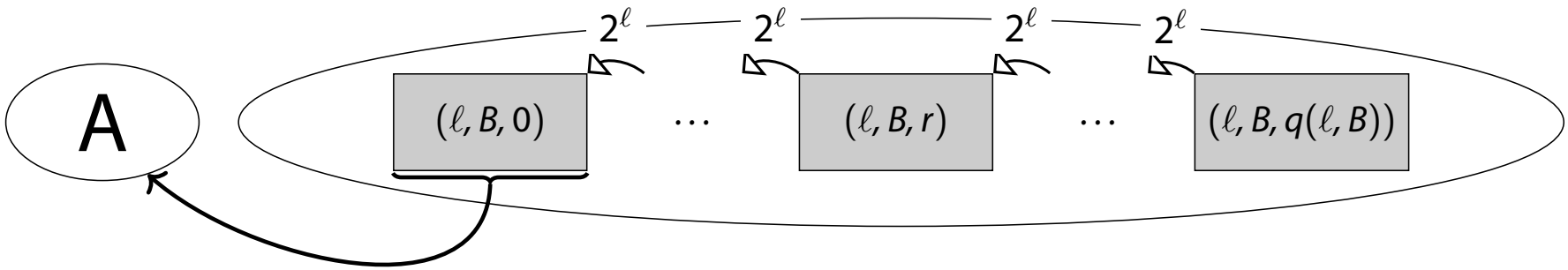
Large Items

Round Items geometrically:



$s \in [2^{-(\ell+1)}, 2^{-\ell})$

- $|(\ell, A, \cdot)| = 2^{\ell} \cdot \text{SIZE}, \quad |(\ell, B, \cdot)| = 2^{\ell} \cdot (\text{SIZE} - 1)$
- Pack rounded items via LP
- Value of SIZE may increase or decrease



Idea : Shifting Items to the left complies with the packing!

Small Items

Greedy fails: $\varepsilon \gg b \gg a$

Small Items

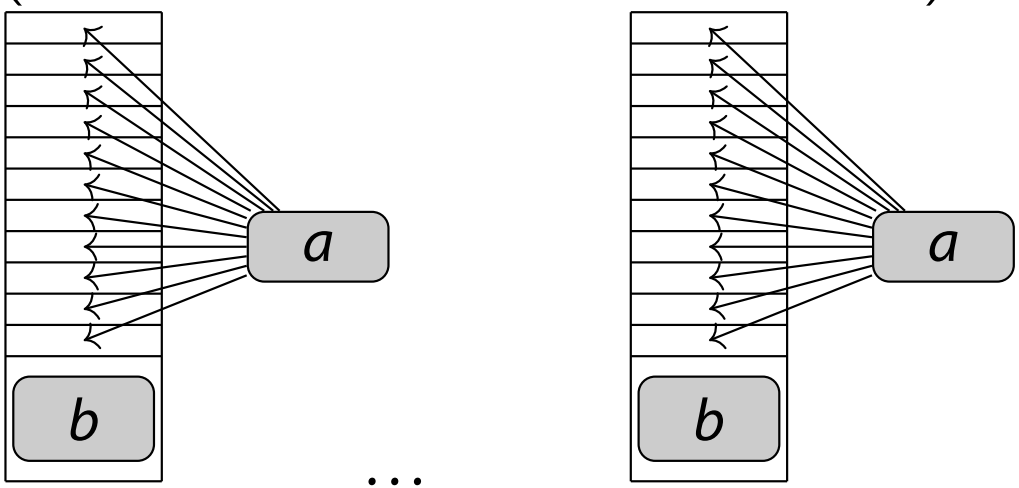
Greedy fails: $\varepsilon \gg b \gg a$

$$(\textbf{INSERT:}\, b, \textbf{INSERT:}\, a, \ldots, \textbf{INSERT:}\, a)^{+}$$

Small Items

Greedy fails: $\varepsilon \gg b \gg a$

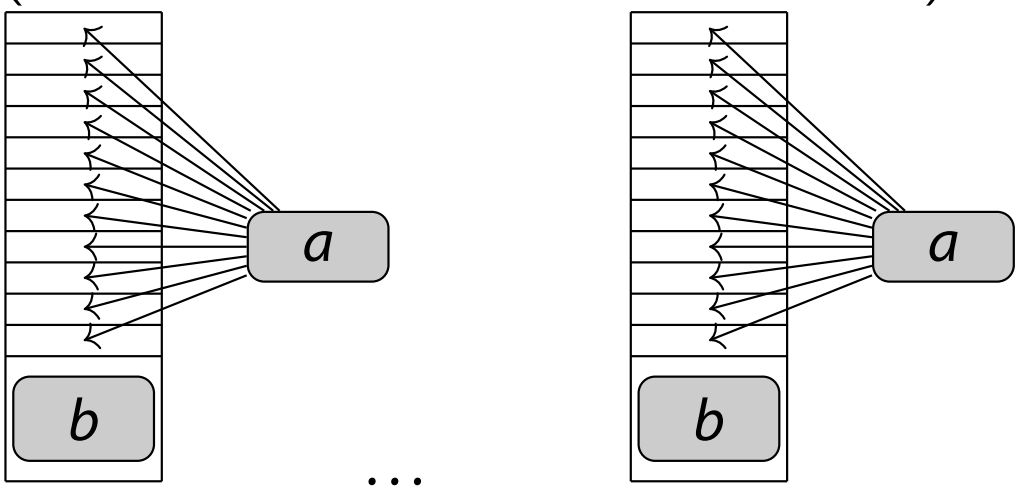
$(\textbf{INSERT:}\, b, \textbf{INSERT:}\, a, \ldots, \textbf{INSERT:}\, a)^{+}$



Small Items

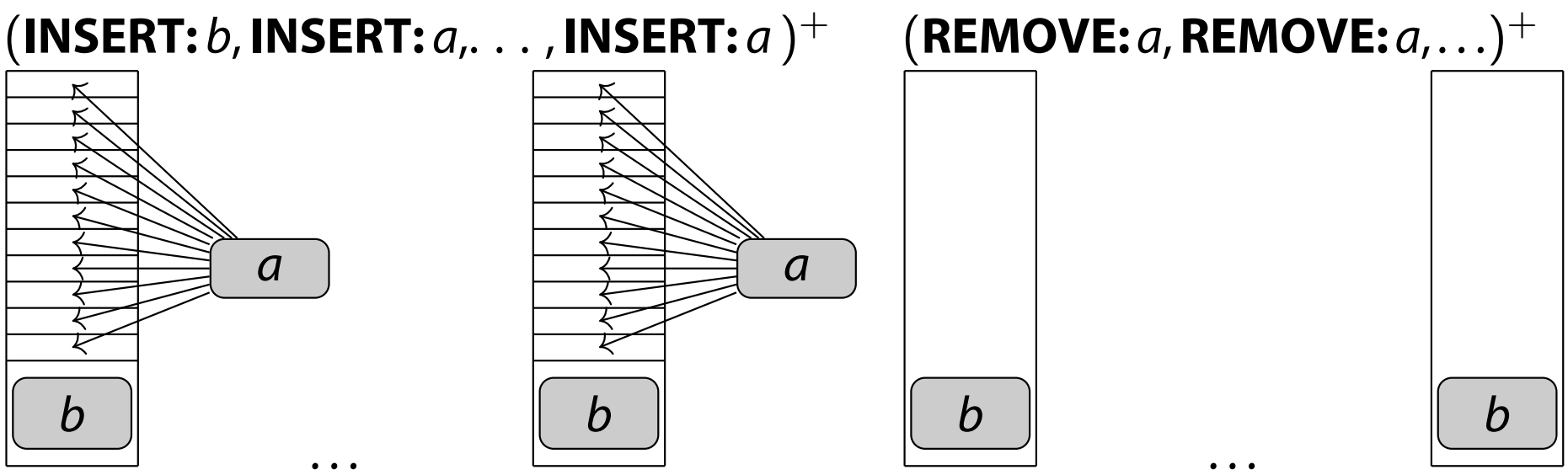
Greedy fails: $\varepsilon \gg b \gg a$

$(\textbf{INSERT:}\, b, \textbf{INSERT:}\, a, \ldots, \textbf{INSERT:}\, a)^+$ $(\textbf{REMOVE:}\, a, \textbf{REMOVE:}\, a, \ldots)^+$



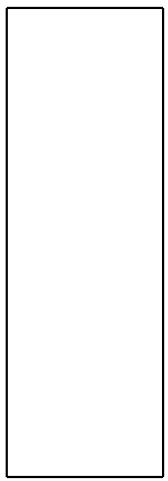
Small Items

Greedy fails: $\varepsilon \gg b \gg a$



Small Items

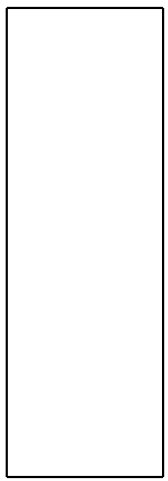
Idea: "Sort" small items from left to right



Small Items

Idea: "Sort" small items from left to right

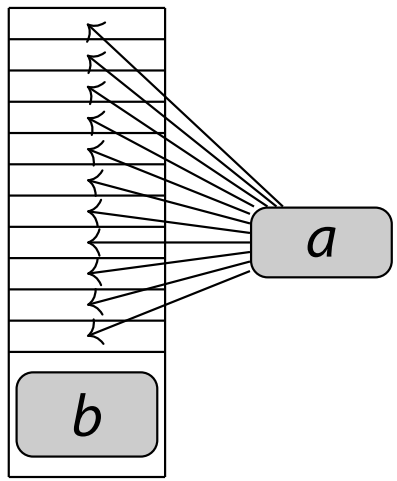
INSERT: *b*, **INSERT:** *a*,. . . , **INSERT:** *a*,



Small Items

Idea: "Sort" small items from left to right

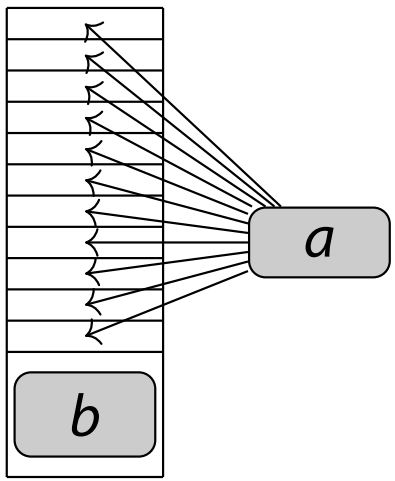
INSERT: *b*, **INSERT:** *a*,. . . , **INSERT:** *a*,



Small Items

Idea: "Sort" small items from left to right

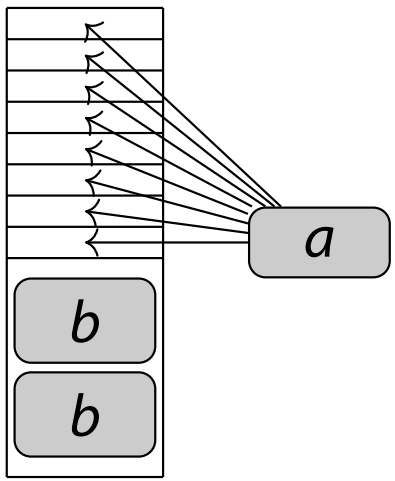
INSERT: *b*, **INSERT:** *a*,. . . , **INSERT:** *a*, **INSERT:** *b*



Small Items

Idea: "Sort" small items from left to right

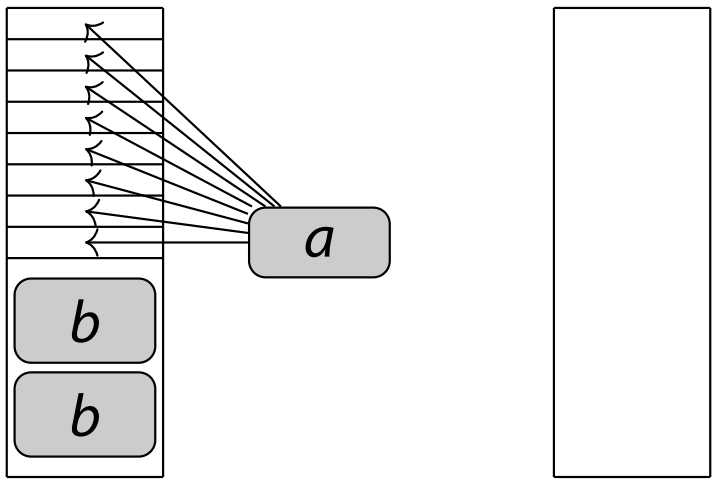
INSERT: *b*, **INSERT:** *a*,. . . , **INSERT:** *a*, **INSERT:** *b*



Small Items

Idea: "Sort" small items from left to right

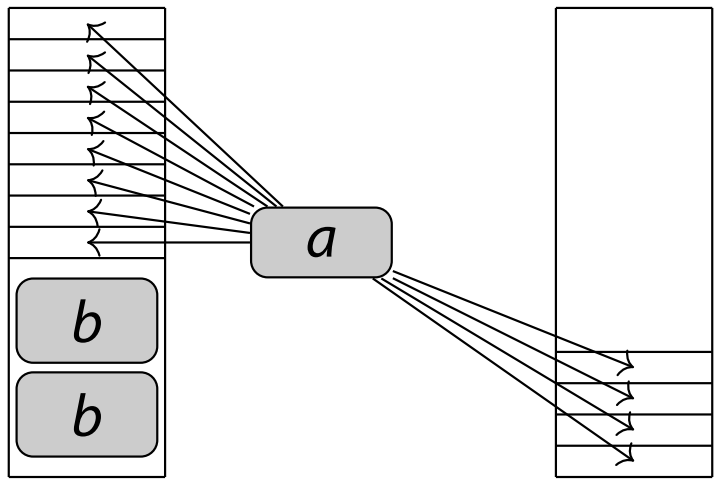
INSERT: *b*, **INSERT:** *a*,. . . , **INSERT:** *a*, **INSERT:** *b*



Small Items

Idea: "Sort" small items from left to right

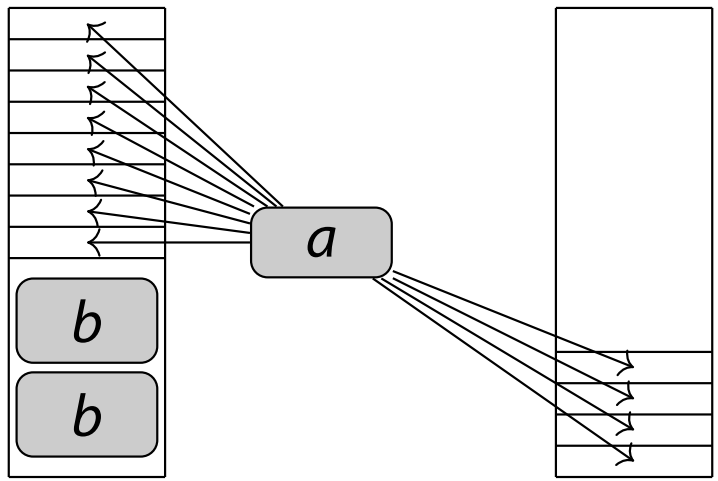
INSERT: *b*, **INSERT:** *a*,. . . , **INSERT:** *a*, **INSERT:** *b*



Small Items

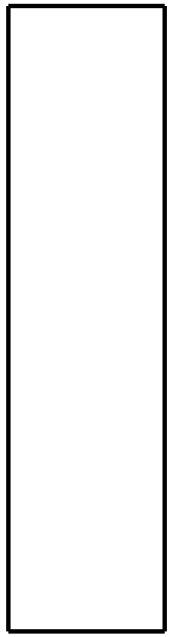
Idea: "Sort" small items from left to right

INSERT: *b*, **INSERT:** *a*,. . . , **INSERT:** *a*, **INSERT:** *b*

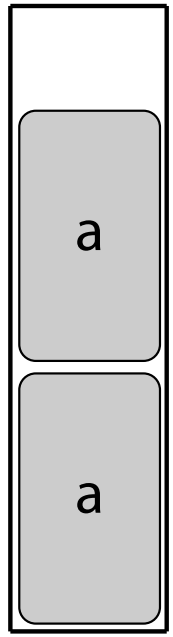


Stop at every $1/\varepsilon$-th bin (buffer bin) to bound MF

Everything

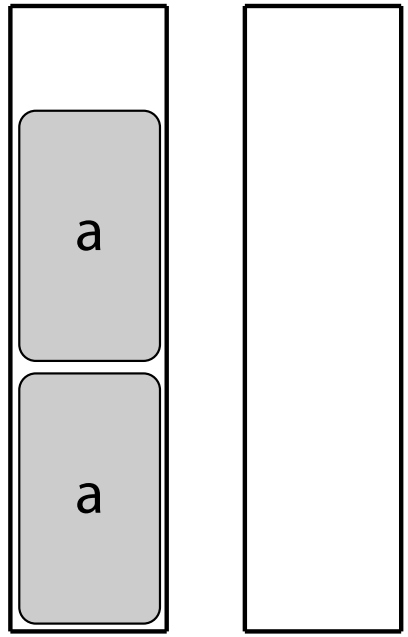


Everything



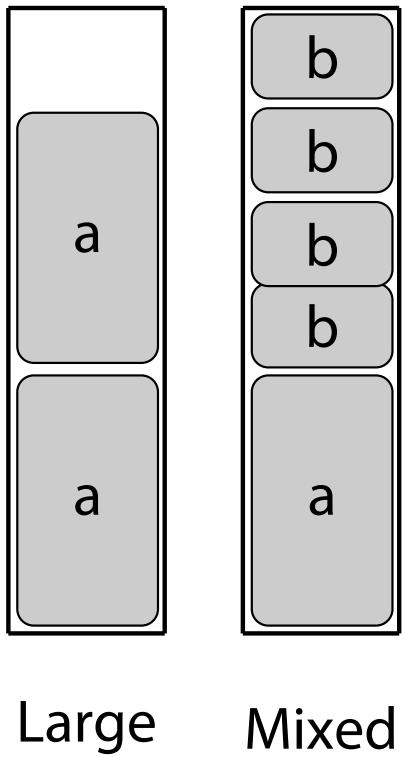
Large

Everything

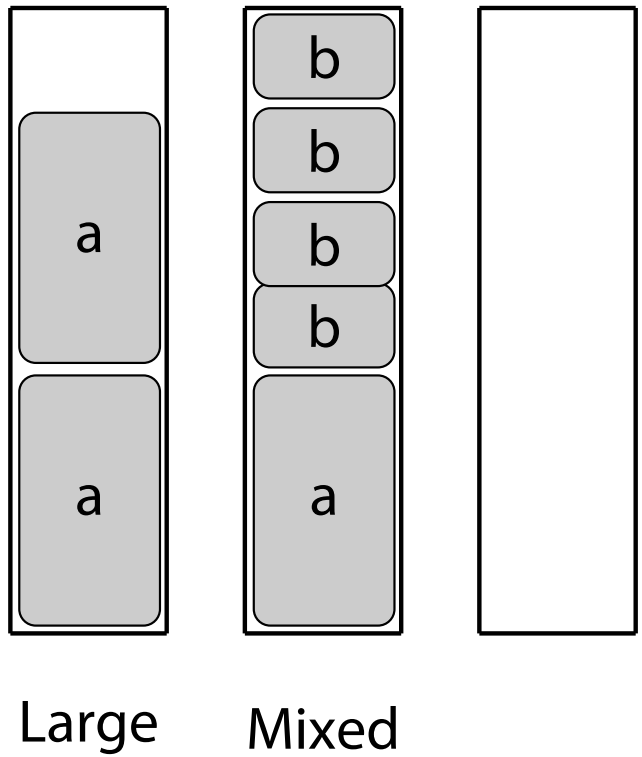


Large

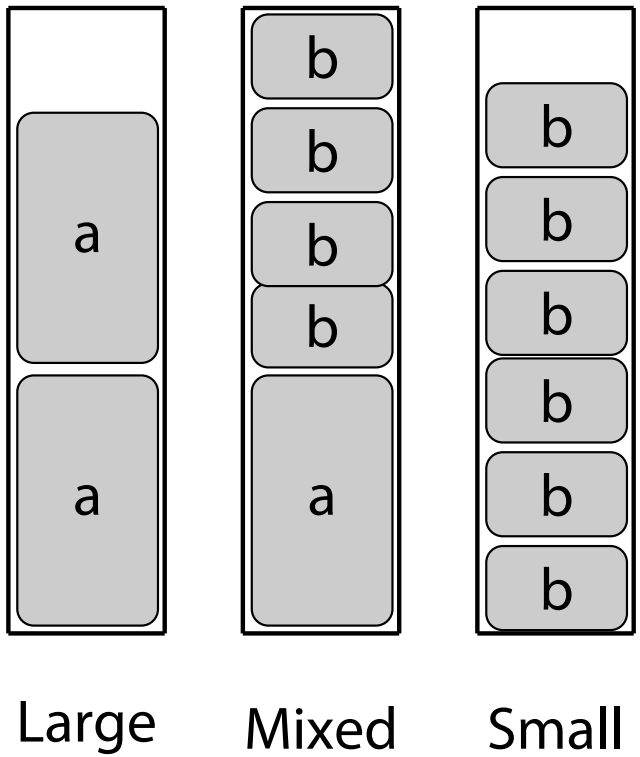
Everything



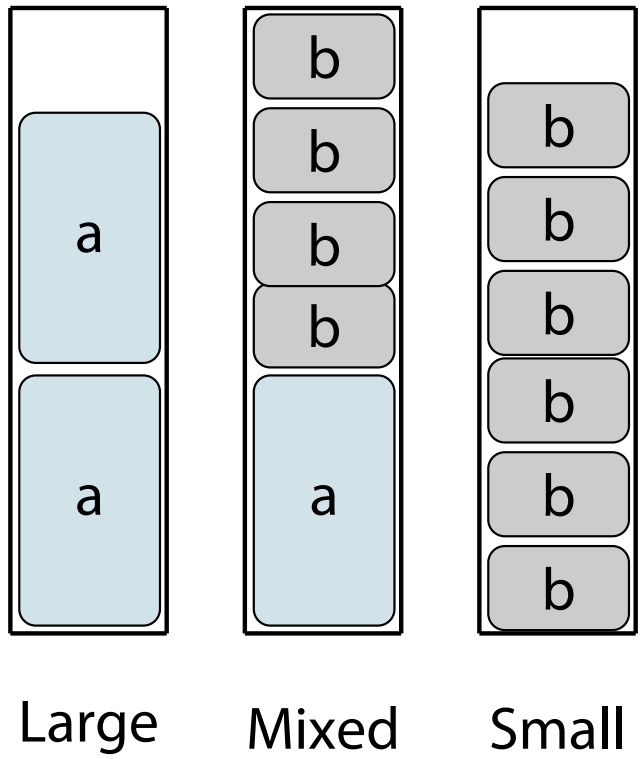
Everything



Everything

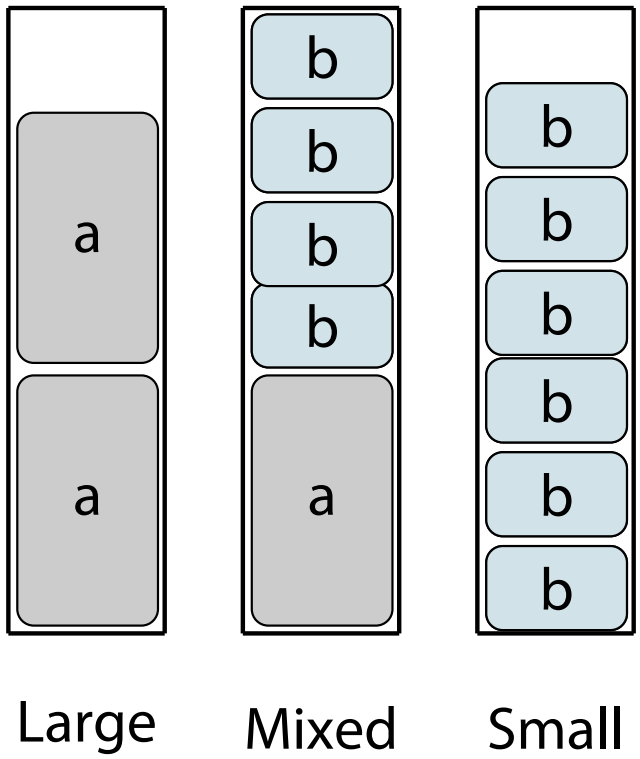


Everything



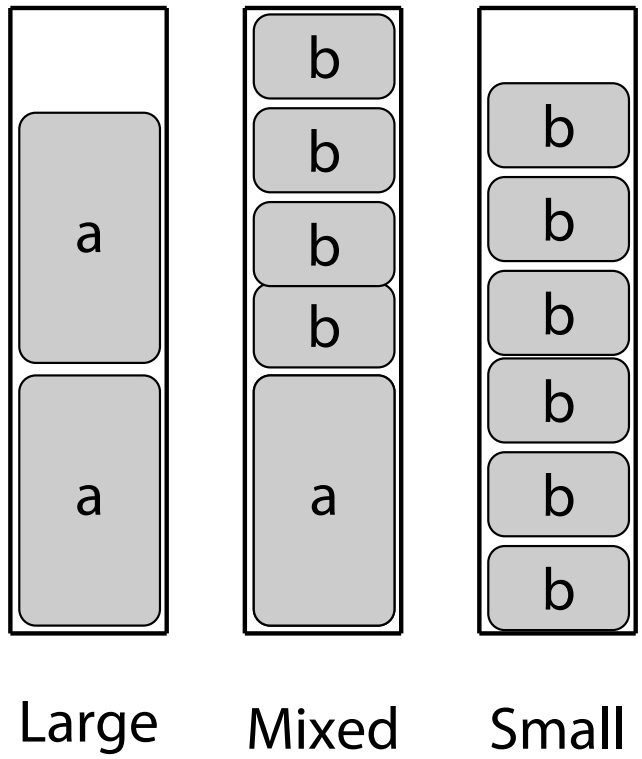
- *Pack via LP*

Everything



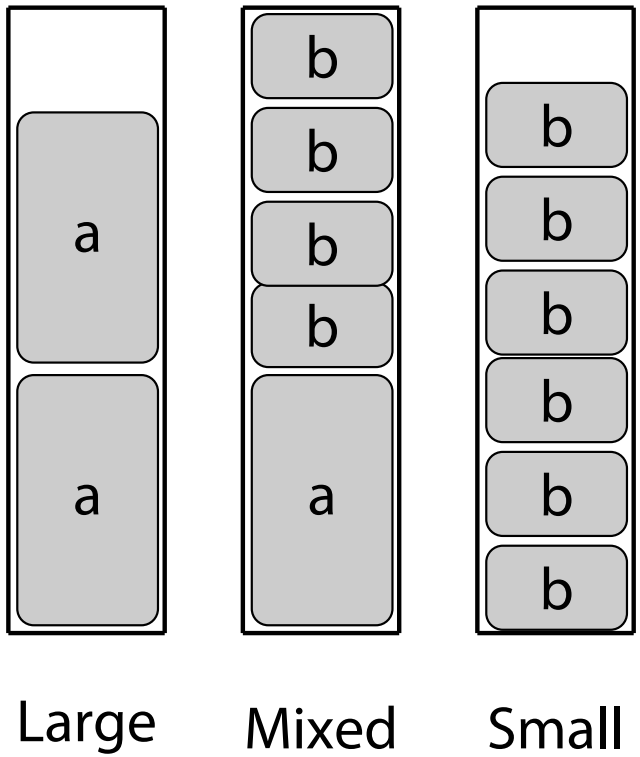
- Pack via LP
- *Pack via "Sorting"*

Everything



- Pack via LP
- Pack via "Sorting"
- *Small bins $\Rightarrow$ little free space in other bins*

Everything



- Pack via LP
- Pack via "Sorting"
- Small bins $\Rightarrow$ little free space in other bins
- *Relate nearly full / nearly empty bins via potential function*

Lookout

- *Improve lower bound on MF from $\Omega(1/\varepsilon)$ (**REMOVE**, $\omega(1)$ sizes)*

Lookout

- Improve lower bound on MF from $\Omega(1/\varepsilon)$ (**REMOVE**, $\omega(1)$ sizes)
- *Improve running time from $\mathcal{O}(1/\varepsilon^6 \cdot \log^4(1/\varepsilon) \cdot n)$ (Plotkin, Shachnai and Yehezkely, Jansen and Kraft)*

Lookout

- Improve lower bound on MF from $\Omega(1/\varepsilon)$ (**REMOVE**, $\omega(1)$ sizes)
- Improve running time from $\mathcal{O}(1/\varepsilon^6 \cdot \log^4(1/\varepsilon) \cdot n)$ (Plotkin, Shachnai and Yehezkely, Jansen and Kraft)
- *Simplify handling of small items*

Lookout

- Improve lower bound on MF from $\Omega(1/\varepsilon)$ (**REMOVE**, $\omega(1)$ sizes)
- Improve running time from $\mathcal{O}(1/\varepsilon^6 \cdot \log^4(1/\varepsilon) \cdot n)$ (Plotkin, Shachnai and Yehezkely, Jansen and Kraft)
- Simplify handling of small items
- *Adapt techniques to other problems*