



Stealing From the Dragon's Hoard

Online Unbounded Knapsack With Removal

Matthias Gehnen (RWTH Aachen University)

Moritz Stocker (ETH Zurich)

Online Problems

- Optimization problems

Online Problems

- Optimization problems
- Problem instance given as a sequence over time

Online Problems

- Optimization problems
- Problem instance given as a sequence over time
- Decisions on each prefix, usually irrevocable

Online Problems

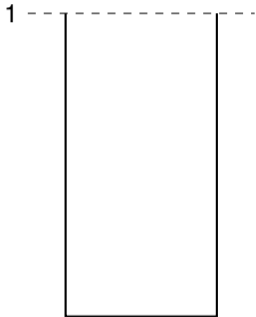
- Optimization problems
- Problem instance given as a sequence over time
- Decisions on each prefix, usually irrevocable
- Performance compared to algorithm with access to entire instance

Online Problems

- Optimization problems
- Problem instance given as a sequence over time
- Decisions on each prefix, usually irrevocable
- Performance compared to algorithm with access to entire instance
- Measured by *competitive ratio* $\frac{\text{gain}(\text{OPT})}{\text{gain}(\text{ALG})}$

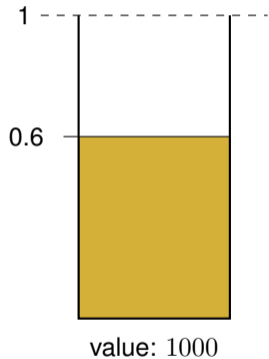
The Online Knapsack Problem

The Online Knapsack Problem

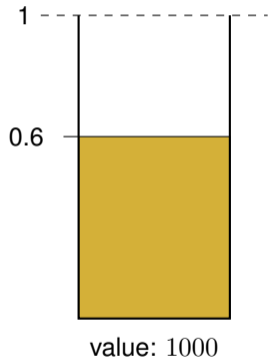


weight: 0.6
value: 1000

The Online Knapsack Problem

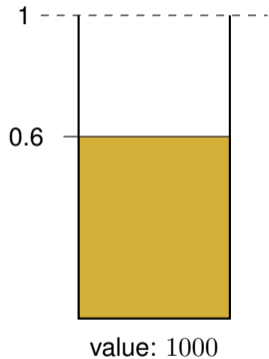


The Online Knapsack Problem



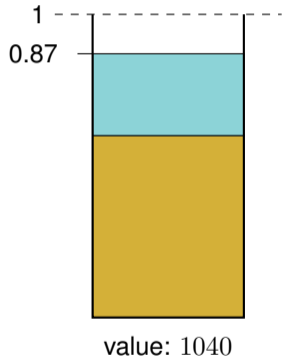
weight: 0.001
value: 0.001

The Online Knapsack Problem

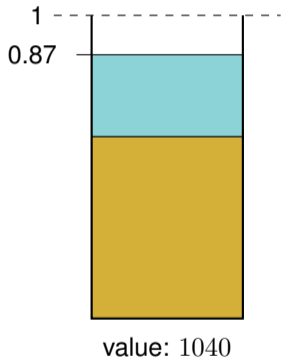


weight: 0.27
value: 40

The Online Knapsack Problem



The Online Knapsack Problem



weight: 1
value: 1

The Proportional Knapsack Problem

The Proportional Knapsack Problem

- Item restrictions: $\text{weight}=\text{value}$

The Proportional Knapsack Problem

- Item restrictions: weight=value
- Only peas

The Proportional Knapsack Problem

- Item restrictions: weight=value
- Only peas



The Proportional Knapsack Problem

- Item restrictions: weight=value
- Only peas



The Proportional Knapsack Problem

- Item restrictions: weight=value
- Only peas



The Proportional Knapsack Problem

- Item restrictions: weight=value
- Only peas



The Proportional Knapsack Problem

- Item restrictions: weight=value
- Only peas



No algorithm is competitive on all instances

The Proportional Knapsack Problem

- Item restrictions: weight=value
- Only peas



No (deterministic) algorithm is competitive on all instances

Power to the Algorithm: Online Knapsack with Removal

Power to the Algorithm: Online Knapsack with Removal

Algorithms may now remove items arbitrarily.

Power to the Algorithm: Online Knapsack with Removal

Algorithms may now remove items arbitrarily.

Theorem (Kazuo Iwama and Shiro Taketomi, 2002)

PROPORTIONAL KNAPSACK WITH REMOVAL has competitiveness $\varphi \approx 1.618$.

Power to the Algorithm: Online Knapsack with Removal

Algorithms may now remove items arbitrarily.

Theorem (Kazuo Iwama and Shiro Taketomi, 2002)

PROPORTIONAL KNAPSACK WITH REMOVAL has competitiveness $\varphi \approx 1.618$.

Theorem (Kazuo Iwama and Guochuan Zhang, 2010)

GENERAL KNAPSACK WITH REMOVAL is not competitive.

The Industrial Setting: Online Unbounded Knapsack

The Industrial Setting: Online Unbounded Knapsack

- Restricted instances: each item must appear “in large quantities”

The Industrial Setting: Online Unbounded Knapsack

- Restricted instances: each item must appear “in large quantities”
- Equivalently: algorithms may pack any number of copies of each item

The Industrial Setting: Online Unbounded Knapsack

- Restricted instances: each item must appear “in large quantities”
- Equivalently: algorithms may pack any number of copies of each item

Theorem (Böckenhauer, Gehnen, Hromkovič, Klasing, Komm, Lotze, Mock, Rossmann, S., 2025)

PROPORTIONAL UNBOUNDED KNAPSACK has competitiveness 2.

The Industrial Setting: Online Unbounded Knapsack

- Restricted instances: each item must appear “in large quantities”
- Equivalently: algorithms may pack any number of copies of each item

Theorem (Böckenhauer, Gehnen, Hromkovič, Klasing, Komm, Lotze, Mock, Rossmann, S., 2025)

PROPORTIONAL UNBOUNDED KNAPSACK has competitiveness 2.

Theorem (Böckenhauer et al., 2025)

GENERAL UNBOUNDED KNAPSACK is not competitive.

The Dragon's Hoard: Unbounded Knapsack with Removal

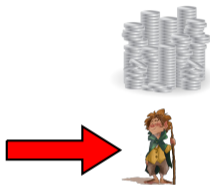
The Dragon's Hoard: Unbounded Knapsack with Removal



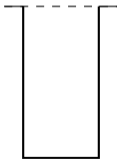
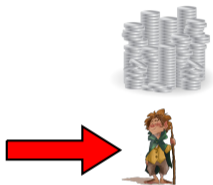
The Dragon's Hoard: Unbounded Knapsack with Removal



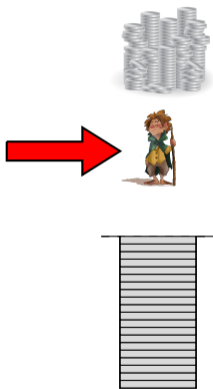
The Dragon's Hoard: Unbounded Knapsack with Removal



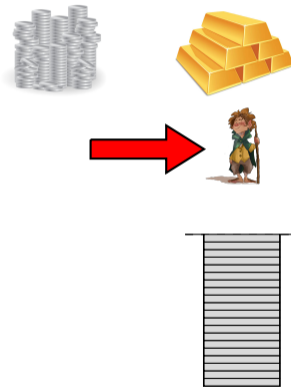
The Dragon's Hoard: Unbounded Knapsack with Removal



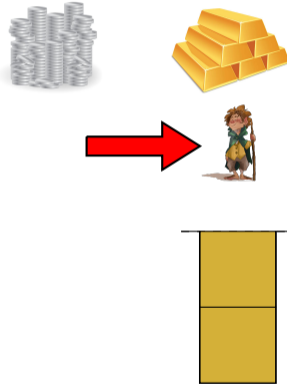
The Dragon's Hoard: Unbounded Knapsack with Removal



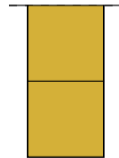
The Dragon's Hoard: Unbounded Knapsack with Removal



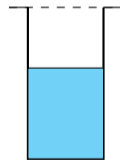
The Dragon's Hoard: Unbounded Knapsack with Removal



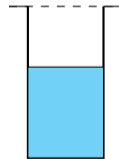
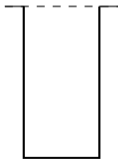
The Dragon's Hoard: Unbounded Knapsack with Removal



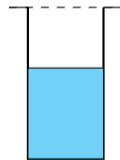
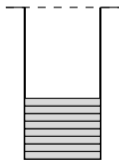
The Dragon's Hoard: Unbounded Knapsack with Removal



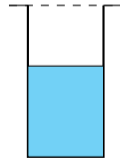
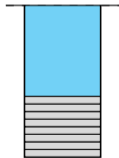
The Dragon's Hoard: Unbounded Knapsack with Removal



The Dragon's Hoard: Unbounded Knapsack with Removal



The Dragon's Hoard: Unbounded Knapsack with Removal



The Dragon's Hoard: Unbounded Knapsack with Removal

- The Dragon's Hoard is competitive for general items!

The Dragon's Hoard: Unbounded Knapsack with Removal

- The Dragon's Hoard is competitive for general items!
- Idea: Keep only “best” item

The Dragon's Hoard: Unbounded Knapsack with Removal

- The Dragon's Hoard is competitive for general items!
- Idea: Keep only “best” item
- Maximize value $\cdot \lfloor 1/\text{weight} \rfloor$

The Dragon's Hoard: Unbounded Knapsack with Removal

- The Dragon's Hoard is competitive for general items!
- Idea: Keep only “best” item
- Maximize value $\cdot \lfloor 1/\text{weight} \rfloor$

Theorem

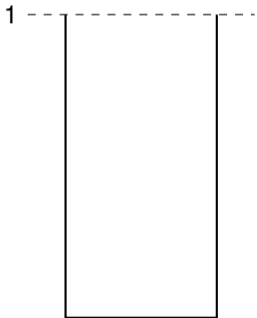
This algorithm has a competitiveness of exactly $S_\infty = 1.6910302\dots$

The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number?

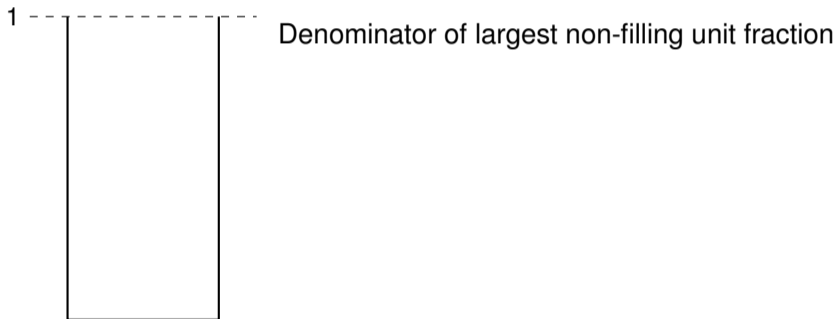
The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



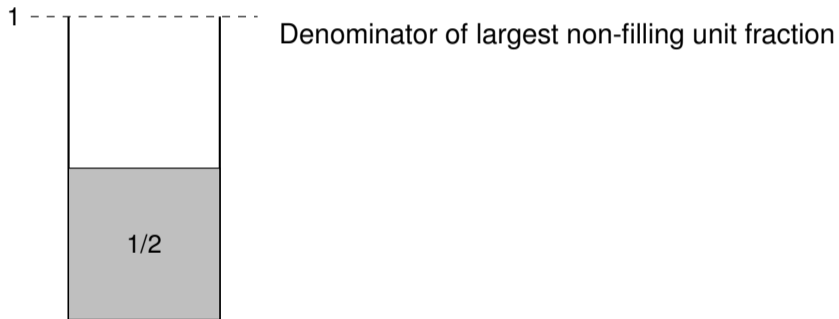
The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



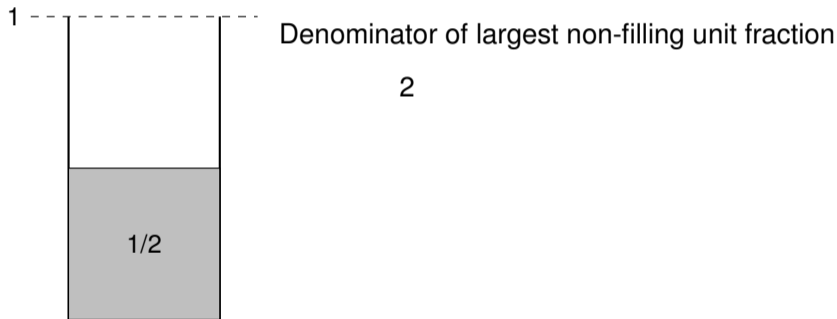
The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



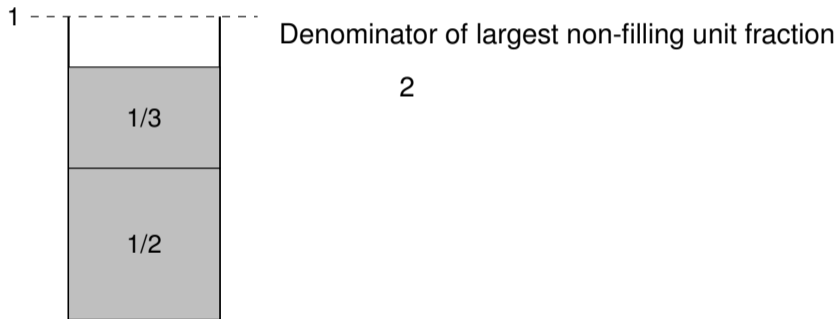
The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



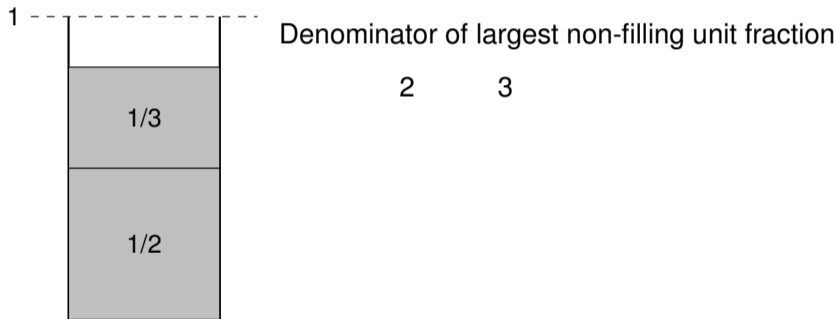
The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



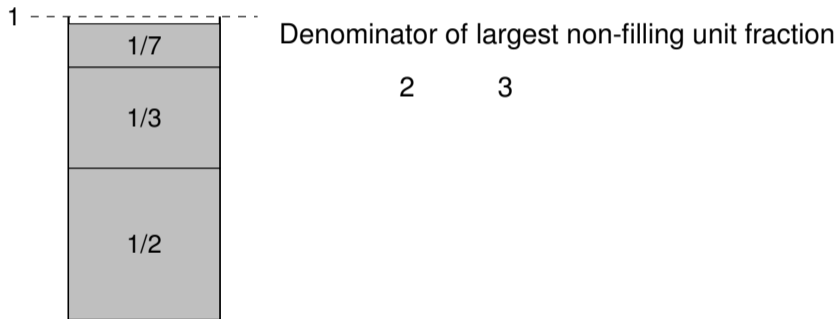
The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



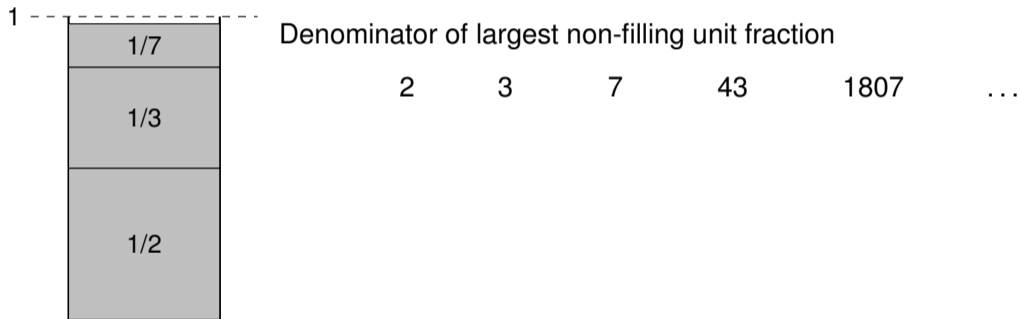
The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



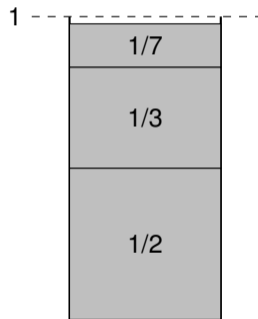
The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



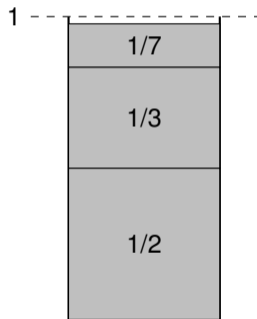
Denominator of largest non-filling unit fraction

2 3 7 43 1807 ...

$$s_{n+1} = 1 + \prod_{i=1}^n s_i$$

The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



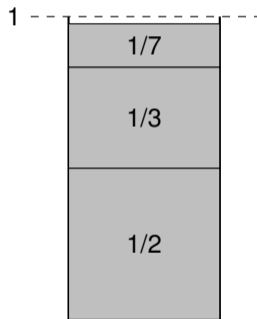
Denominator of largest non-filling unit fraction

$$S_{\infty} = \quad 2 \quad 3 \quad 7 \quad 43 \quad 1807 \quad \dots$$

$$s_{n+1} = 1 + \prod_{i=1}^n s_i$$

The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



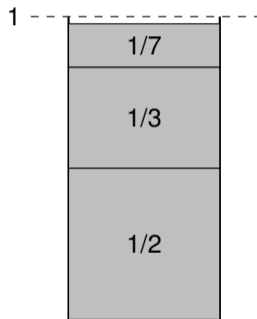
Denominator of largest non-filling unit fraction

$$S_{\infty} = 1 + \frac{2}{1} + \frac{3}{1/2} + \frac{7}{1/6} + \frac{43}{1/42} + \frac{1807}{1/1806} + \dots$$

$$s_{n+1} = 1 + \prod_{i=1}^n s_i$$

The Dragon's Hoard: Unbounded Knapsack with Removal

Why this number? Introducing the Sylvester numbers...



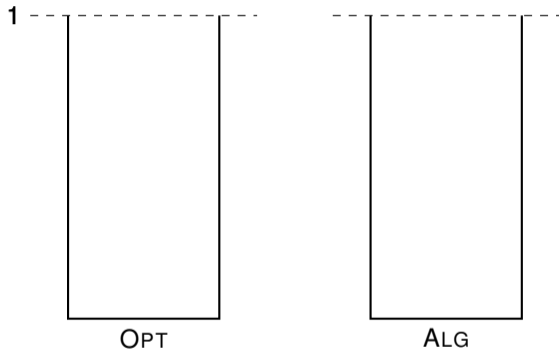
Denominator of largest non-filling unit fraction

$$S_{\infty} = 1 + \frac{2}{1} + \frac{3}{1/2} + \frac{7}{1/6} + \frac{43}{1/42} + \frac{1807}{1/1806} + \dots$$

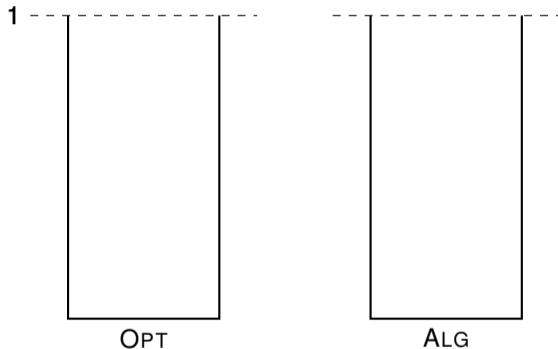
$$s_{n+1} = 1 + \prod_{i=1}^n s_i \quad S_{\infty} = \sum_{n=1}^{\infty} \frac{1}{s_n - 1}$$

Focussing on a Single Item

Focussing on a Single Item

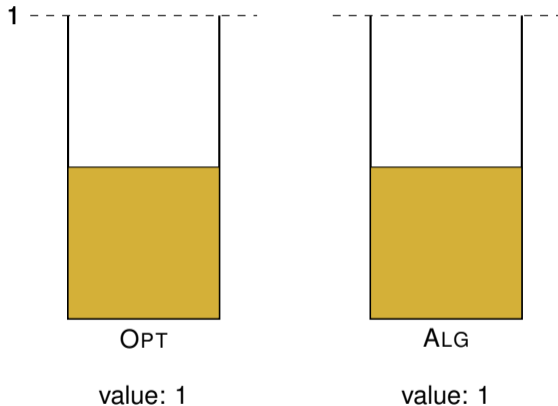


Focussing on a Single Item



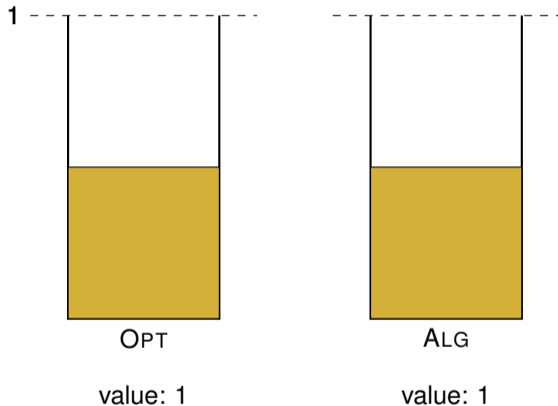
weight: $1/2 + \bullet$
value: 1

Focussing on a Single Item



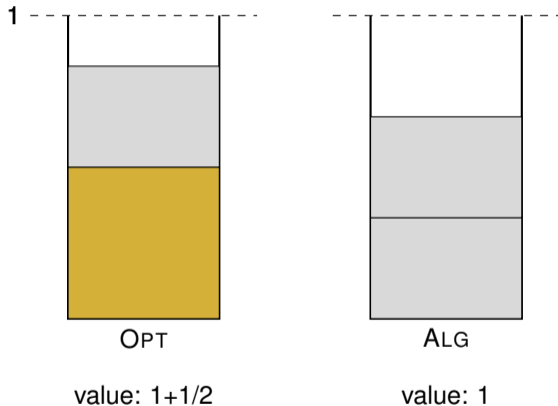
weight: $1/2 + \bullet$
value: 1

Focussing on a Single Item



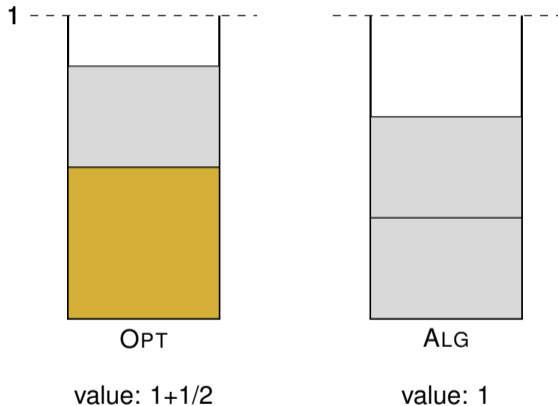
weight: $1/3 + \bullet$
value: $1/2$

Focussing on a Single Item



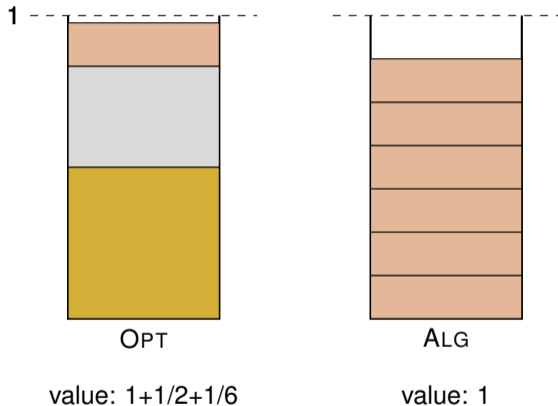
weight: $1/3 + \bullet$
value: $1/2$

Focussing on a Single Item



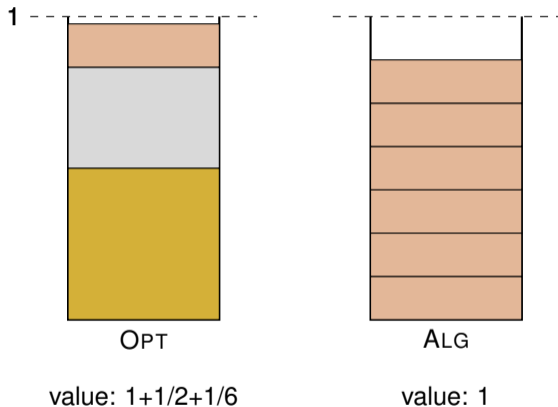
weight: $1/7 + \bullet$
value: $1/6$


Focussing on a Single Item



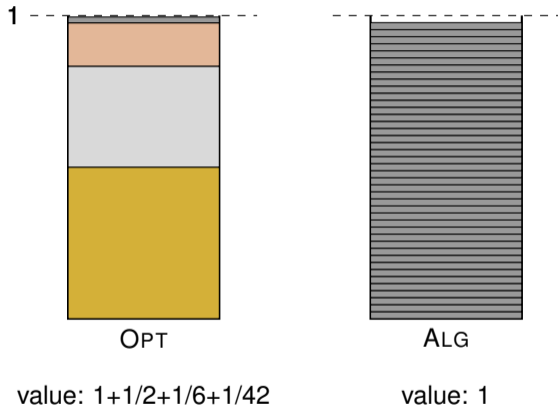
weight: $1/7 + \bullet$
value: $1/6$


Focussing on a Single Item



 weight: $1/43 +$ 
value: $1/42$

Focussing on a Single Item



 weight: $1/43 +$ 
value: $1/42$

Comparing Competitive Ratios

Results for Deterministic Algorithms

Proportional Variant

General Variant

Classical Knapsack

∞

∞

KP with Removal

φ

∞

Unbounded KP

2

∞

Comparing Competitive Ratios

Results for Deterministic Algorithms

Proportional Variant

General Variant

Classical Knapsack

∞

∞

KP with Removal

φ

∞

Unbounded KP

2

∞

Dragon's Hoard

Comparing Competitive Ratios

Results for Deterministic Algorithms

	Proportional Variant	General Variant	
		lower bound	upper bound
Classical Knapsack	∞	∞	
KP with Removal	φ	∞	
Unbounded KP	2	∞	
Dragon's Hoard			1.6911

Comparing Competitive Ratios

Results for Deterministic Algorithms

	Proportional Variant	General Variant	
		lower bound	upper bound
Classical Knapsack	∞		∞
KP with Removal	φ		∞
Unbounded KP	2		∞
Dragon's Hoard		1.5877	1.6911

Comparing Competitive Ratios

Results for Deterministic Algorithms

	Proportional Variant	General Variant	
		lower bound	upper bound
Classical Knapsack	∞		∞
KP with Removal	φ		∞
Unbounded KP	2		∞
Dragon's Hoard	1.5	1.5877	1.6911

An Annoying Coincidence: The Bin-Packing Problem

An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online

An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item

An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)

An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



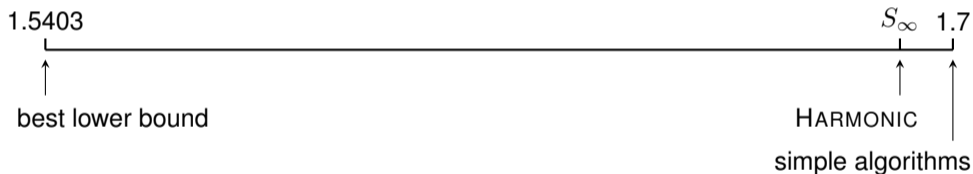
An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



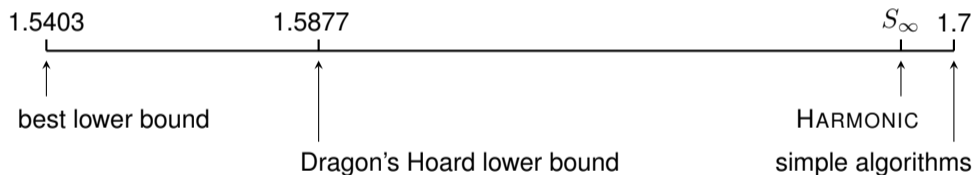
An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



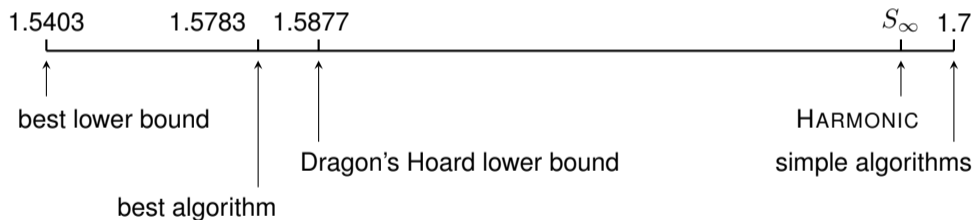
An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



SUPER-HARMONIC framework (SH)

An Annoying Coincidence: The Bin-Packing Problem

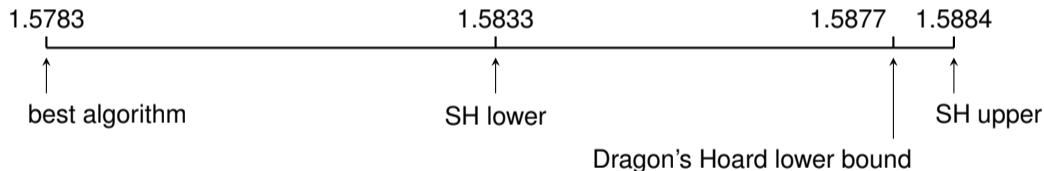
- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



SUPER-HARMONIC framework (SH)

An Annoying Coincidence: The Bin-Packing Problem

- Items with varying weight arrive online
- Pack *every* item
- Minimize number of “bins” used (unit capacity)



SUPER-HARMONIC framework (SH)

Thank You