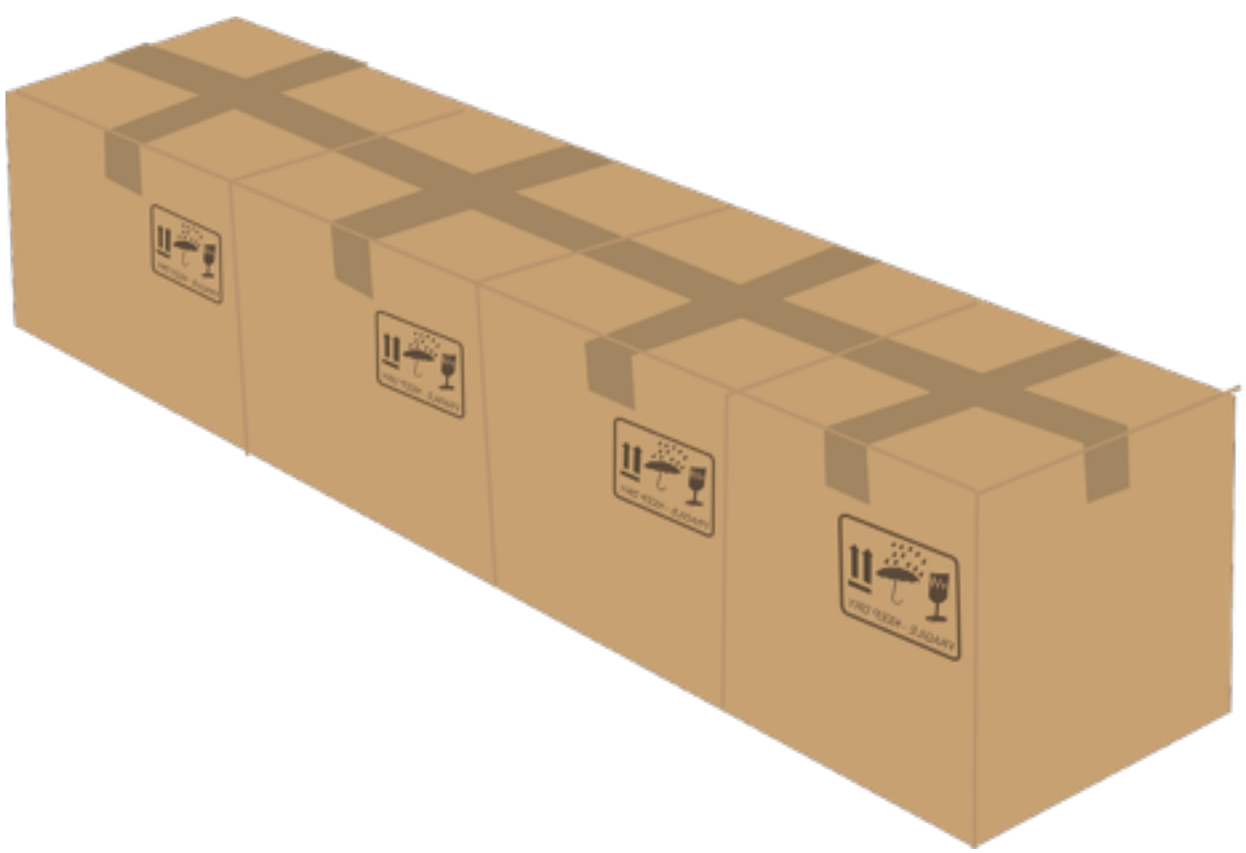
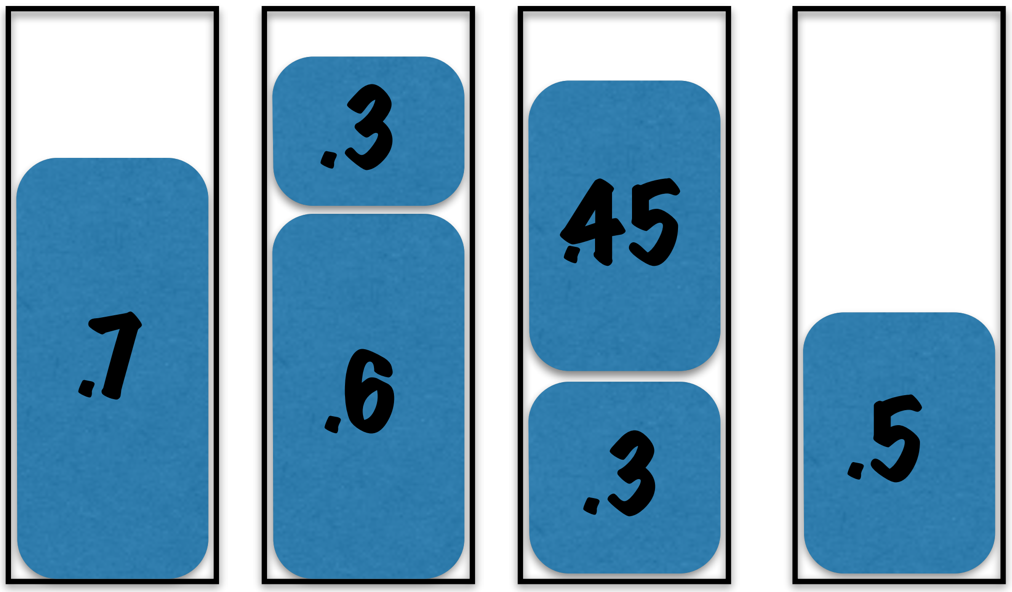
Bin packing, linear programming and rounding





Meta-tool: special cases

Large items

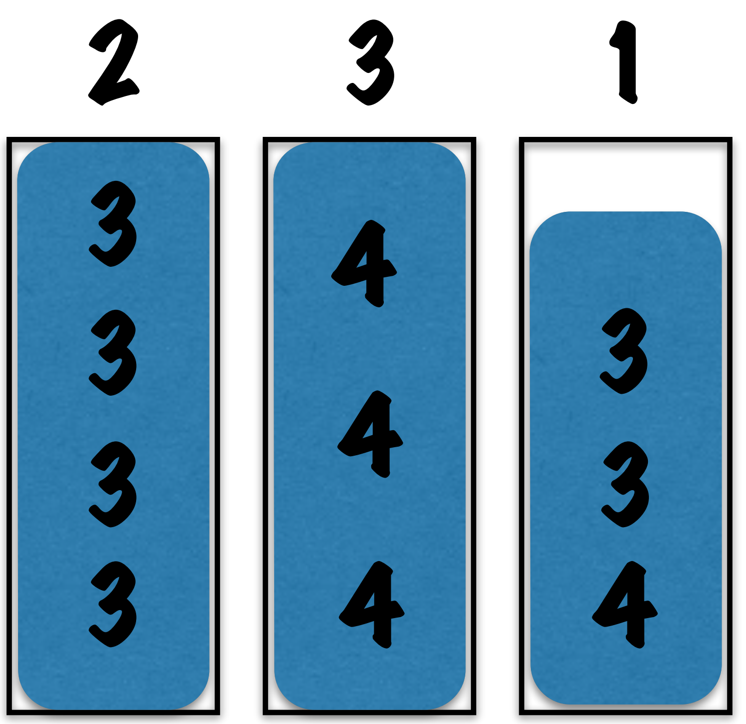
Special special case

Large items,
few distinct sizes

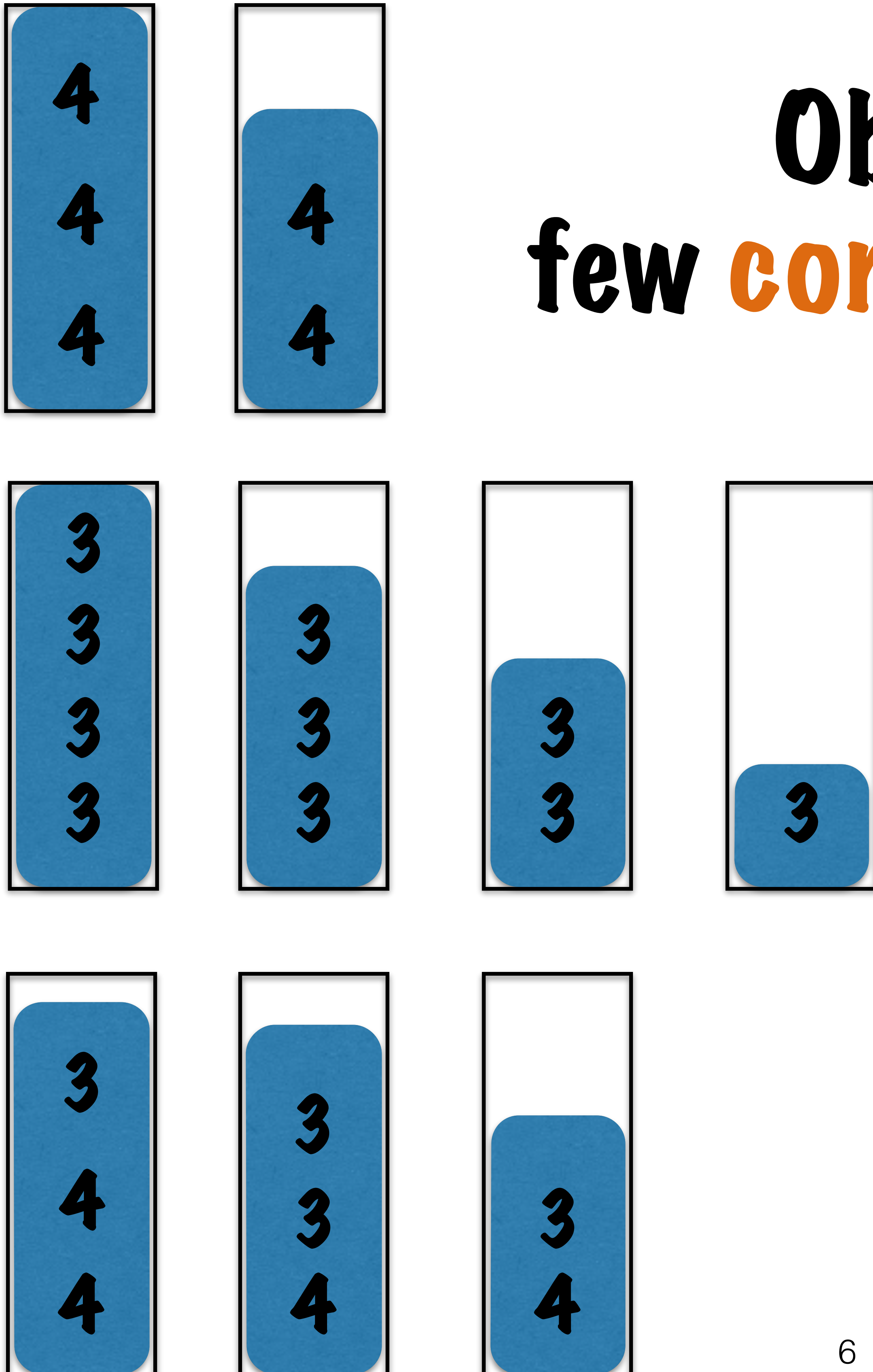
Example

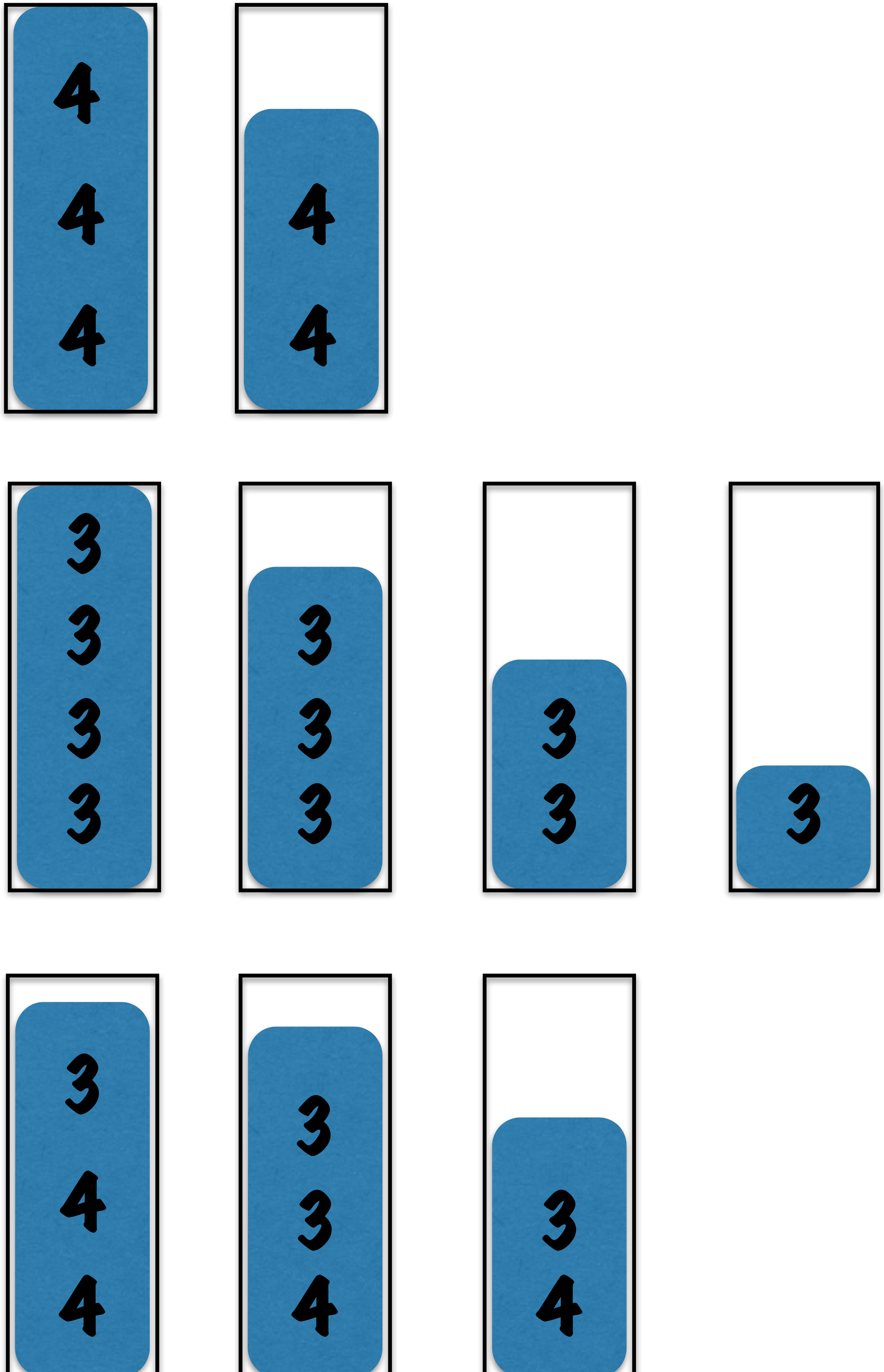
Bin capacity 12
sizes: { 3, 4 }
10 items of size 3,
10 items of size 4.

Bin capacity 12
sizes: { 3, 4 }
10 items of size 3,
10 items of size 4.



Observe:
few configurations





Large items, few distinct sizes C={configurations}

In configuration c size s occurs $a_{s,c}$ times

Integer program

Input: S={size}

number of items of size s: n_s

Output: C={configurations}

number of bins in configuration c: x_c

Constraints: $\sum_c a_{s,c} x_c \geq n_s$

Number of bins: $\sum_c x_c$

integer

If size > capacity/10 then:
< 10 items per configuration

If < 10 sizes then:
< 10^10 configurations

Solve LP relaxation
10 constraints, 10^10 variables
Round up to nearest integer

#bins < OPT + 10^10

(Exhaustive search also ok)

For every $(\mathbf{x_c})_{\mathbf{c} \in \mathcal{C}} \in \{0, 1, \cdots, \mathbf{n}\}^{|\mathcal{C}|}$

Check whether, for every size s, enough slots for items of size s

Output solution with min #bins

Runtime if size>capacity/10:

$$|\mathbf{S}| \times \mathbf{n}^{|\mathbf{S}|^{10}}$$

Bin packing, linear programming and rounding