Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

But also based on discussions with

Michael Albert and Jay Pantone

Sorting with restricted containers

Vince Vatter
University of Florida

Based on the paper

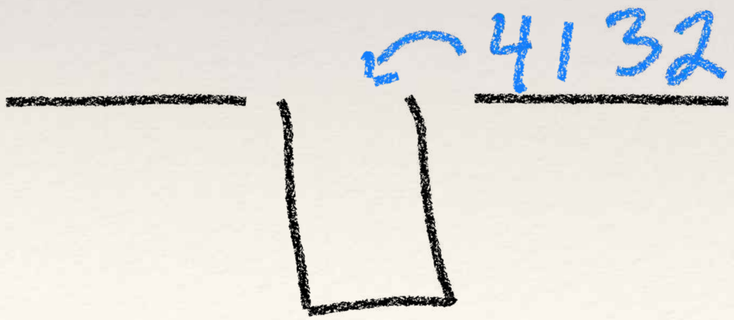
Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

But also based on discussions with

Michael Albert and Jay Pantone

Sorting with restricted containers

Vince Vatter
University of Florida



Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

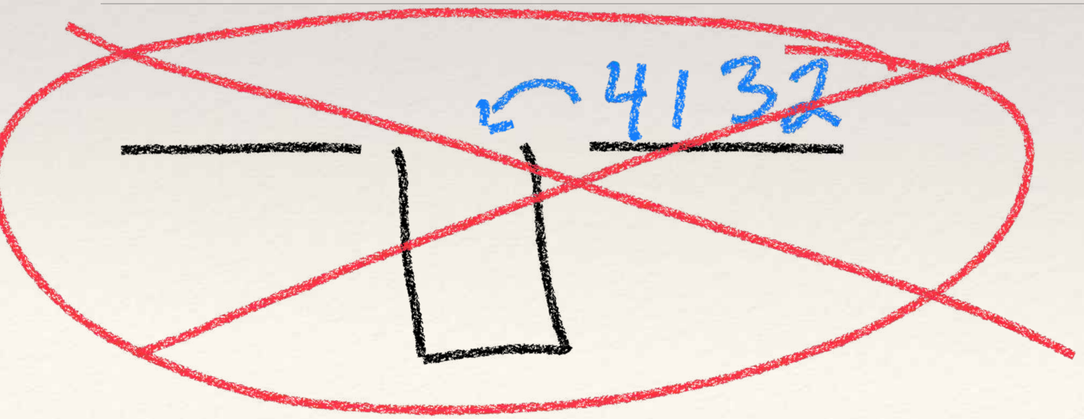
But also based on discussions with

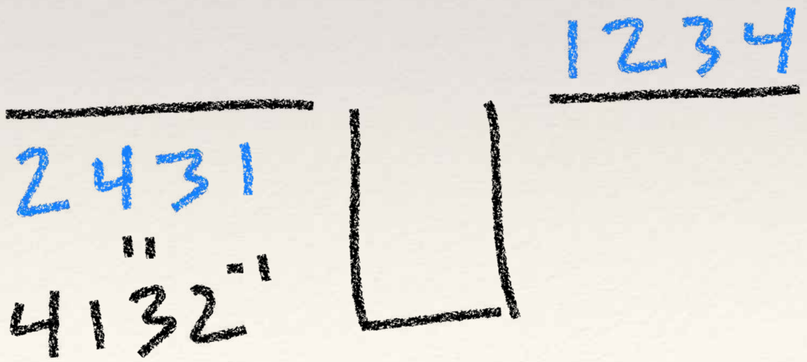
Michael Albert and Jay Pantone

Generating

~~Sorting~~ with restricted containers

Vince Vatter
University of Florida





Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

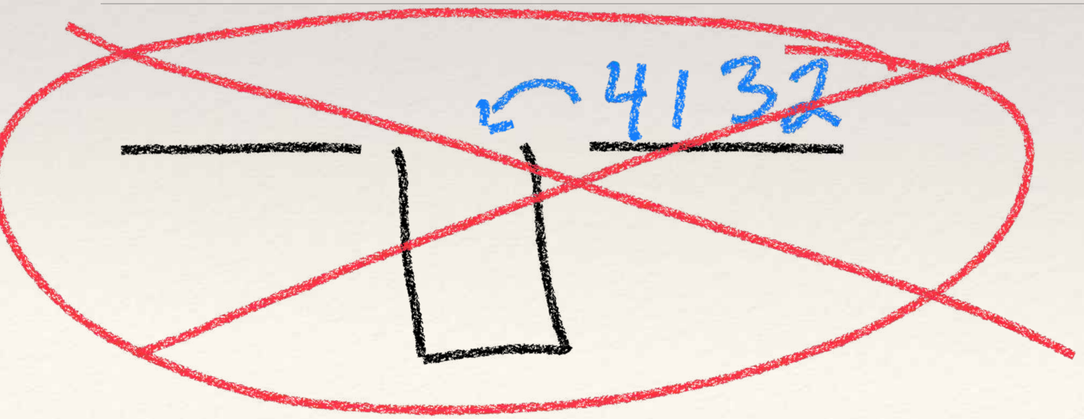
But also based on discussions with

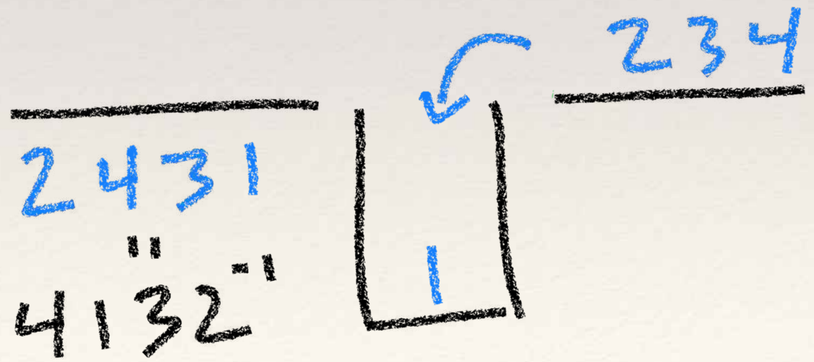
Michael Albert and Jay Pantone

Generating

~~Sorting~~ with restricted containers

Vince Vatter
University of Florida





Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

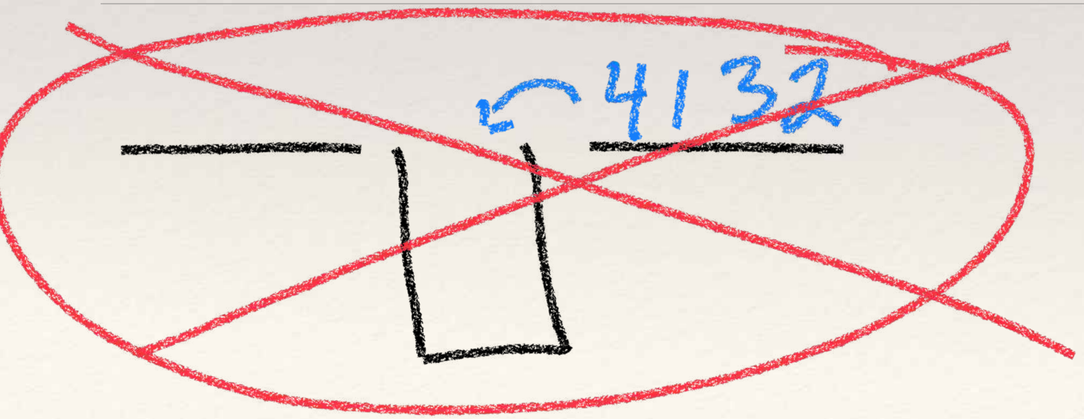
But also based on discussions with

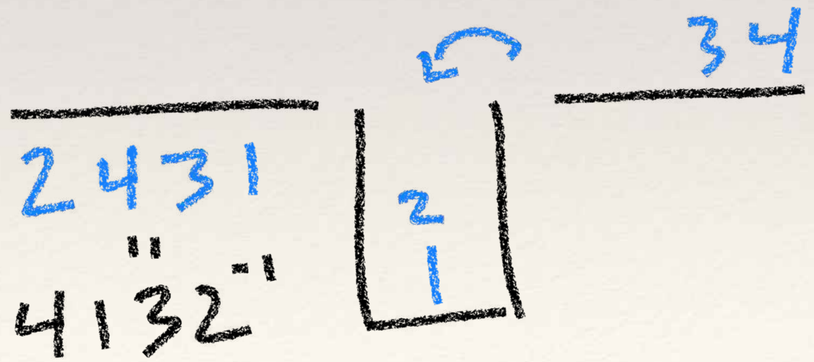
Michael Albert and Jay Pantone

Generating

~~Sorting~~ with restricted containers

Vince Vatter
University of Florida





Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

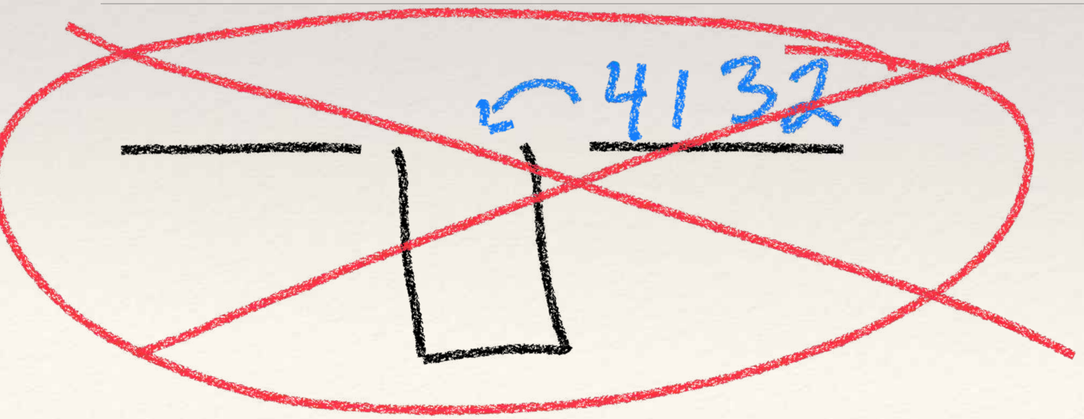
But also based on discussions with

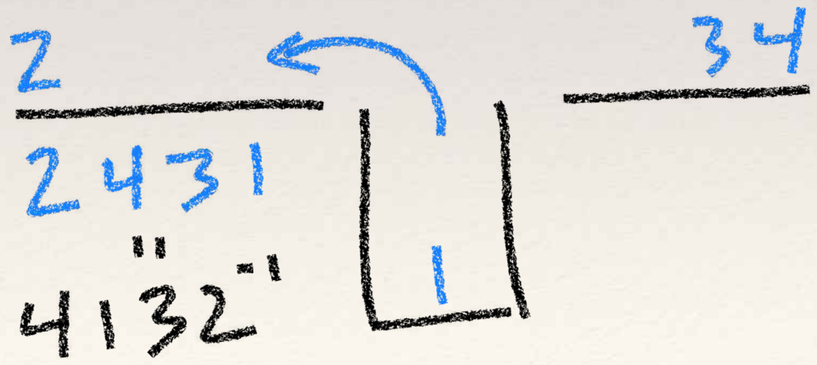
Michael Albert and Jay Pantone

Generating

~~Sorting~~ with restricted containers

Vince Vatter
University of Florida





Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

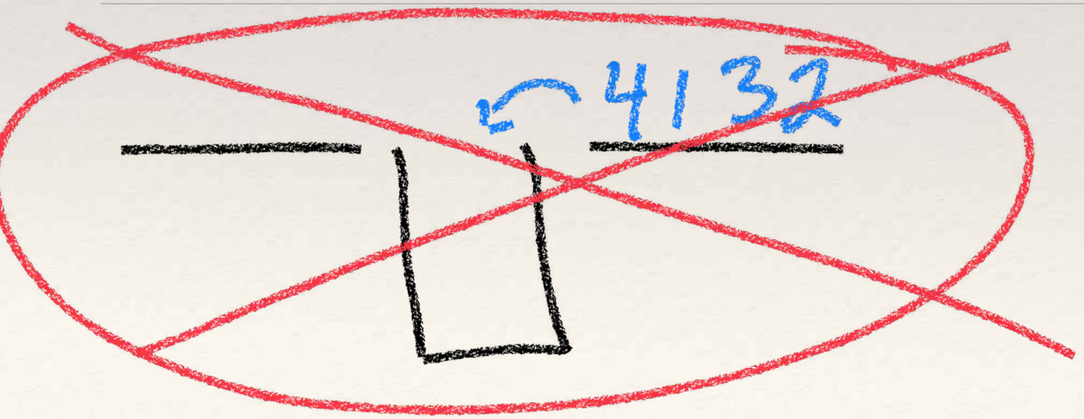
But also based on discussions with

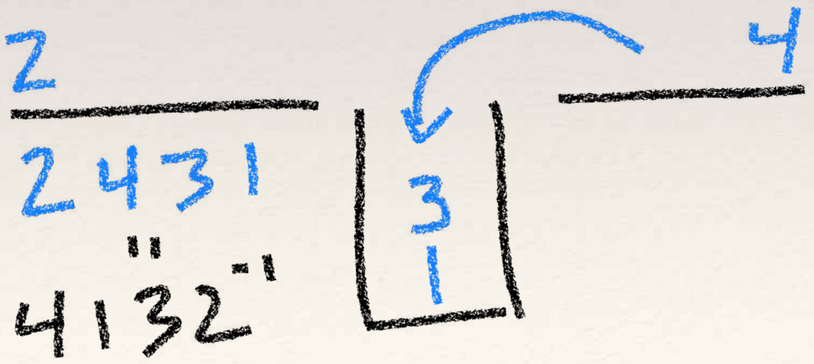
Michael Albert and Jay Pantone

Generating

Sorting with restricted containers

Vince Vatter
University of Florida





Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

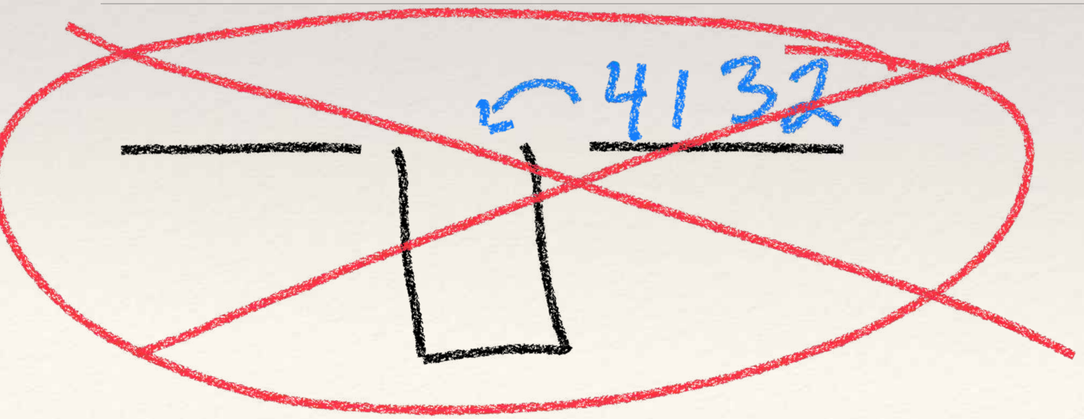
But also based on discussions with

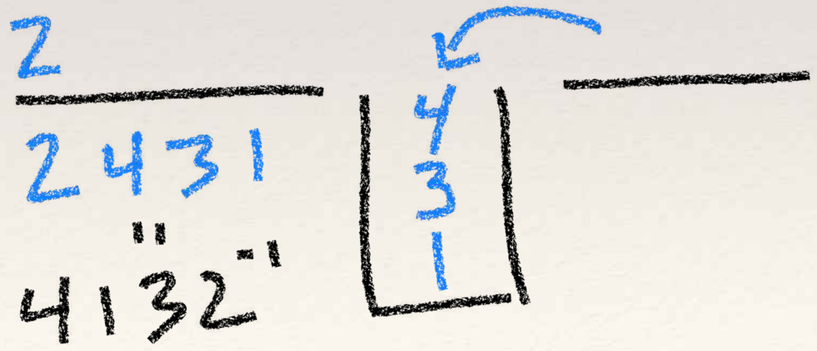
Michael Albert and Jay Pantone

Generating

~~Sorting~~ with restricted containers

Vince Vatter
University of Florida





Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

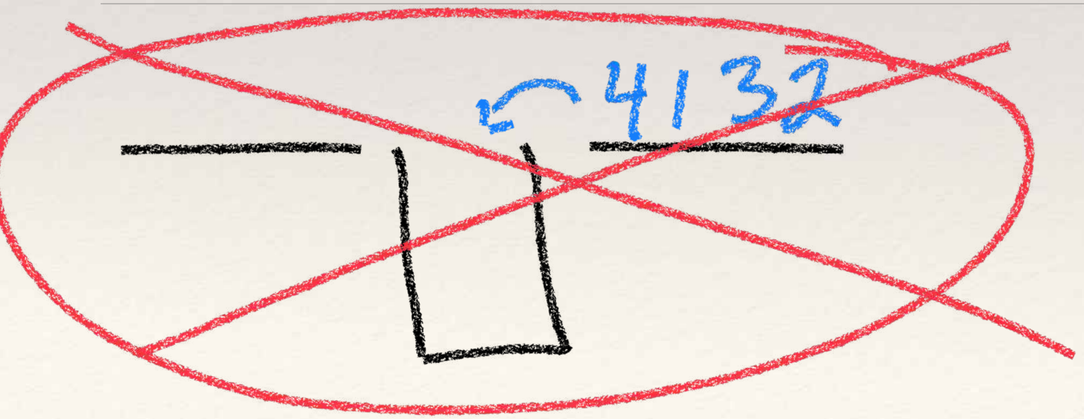
But also based on discussions with

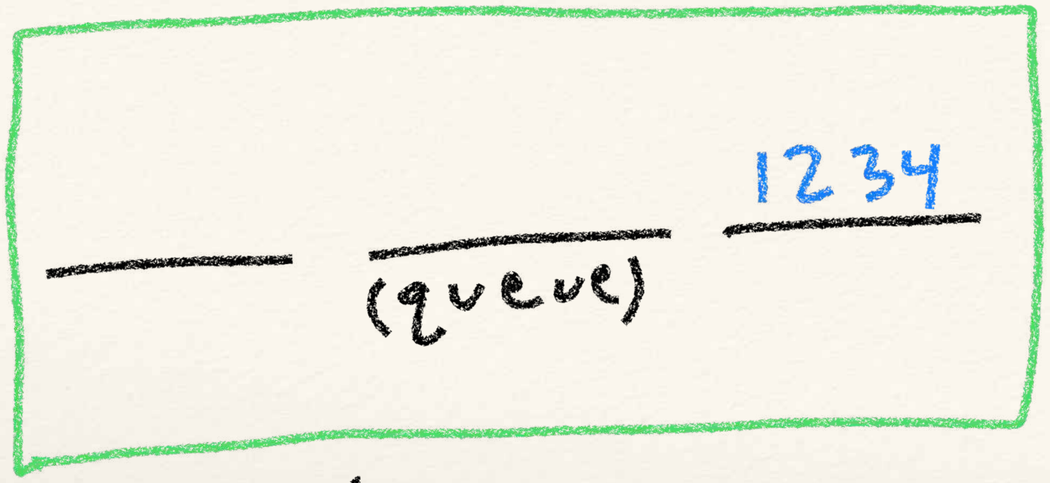
Michael Albert and Jay Pantone

Generating

Sorting with restricted containers

Vince Vatter
University of Florida





Generating

~~Sorting~~ with restricted containers

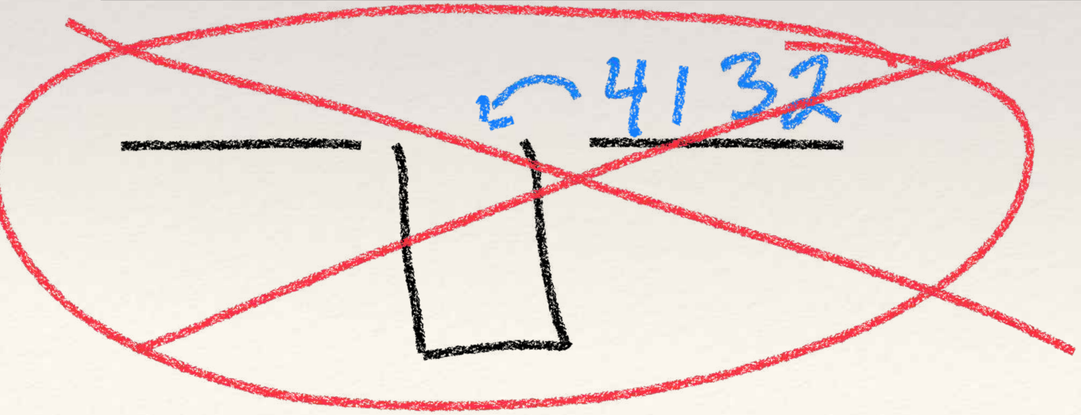
Vince Vatter
University of Florida

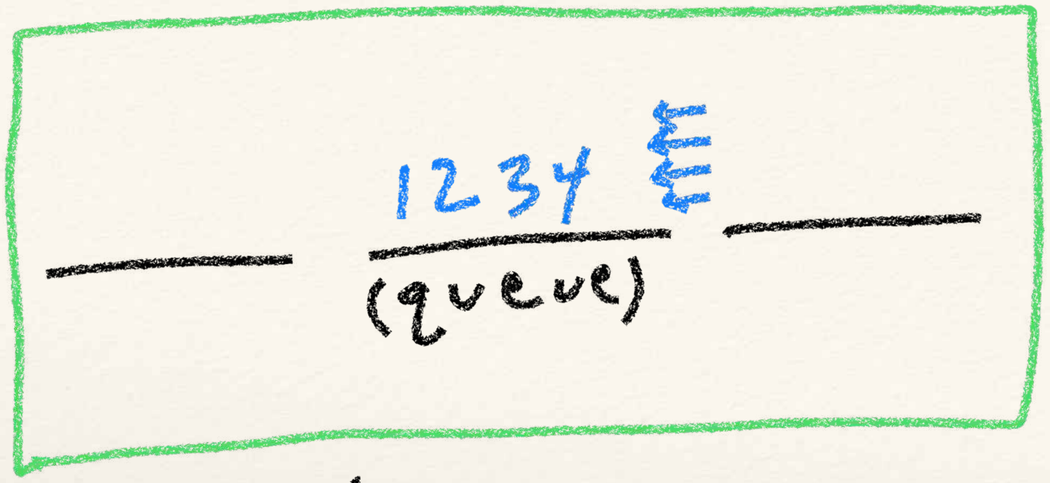
Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

But also based on discussions with

Michael Albert and Jay Pantone

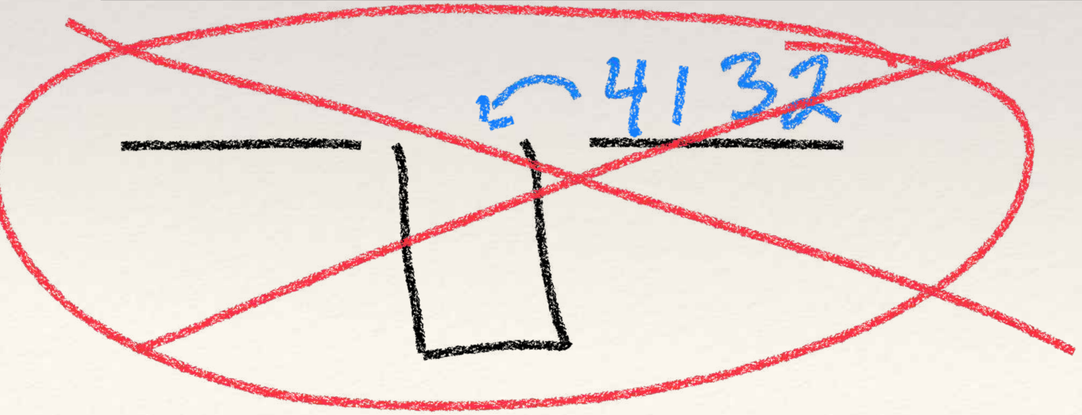




Generating

Sorting with restricted containers

Vince Vatter
University of Florida

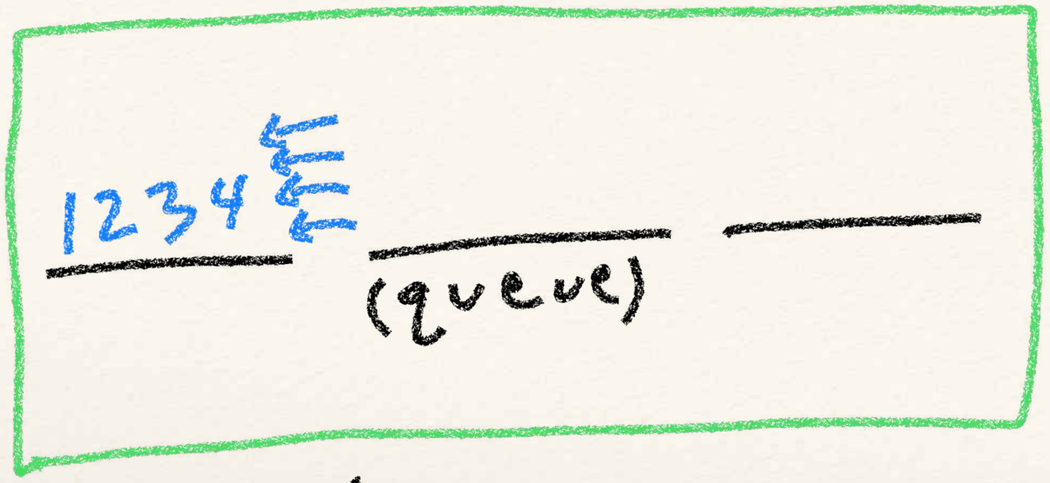


Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

But also based on discussions with

Michael Albert and Jay Pantone



Generating

~~Sorting~~ with restricted containers

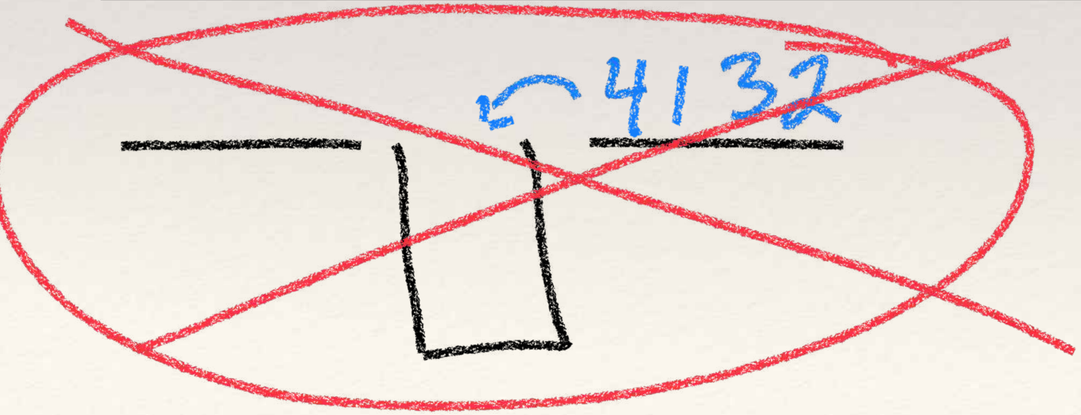
Vince Vatter
University of Florida

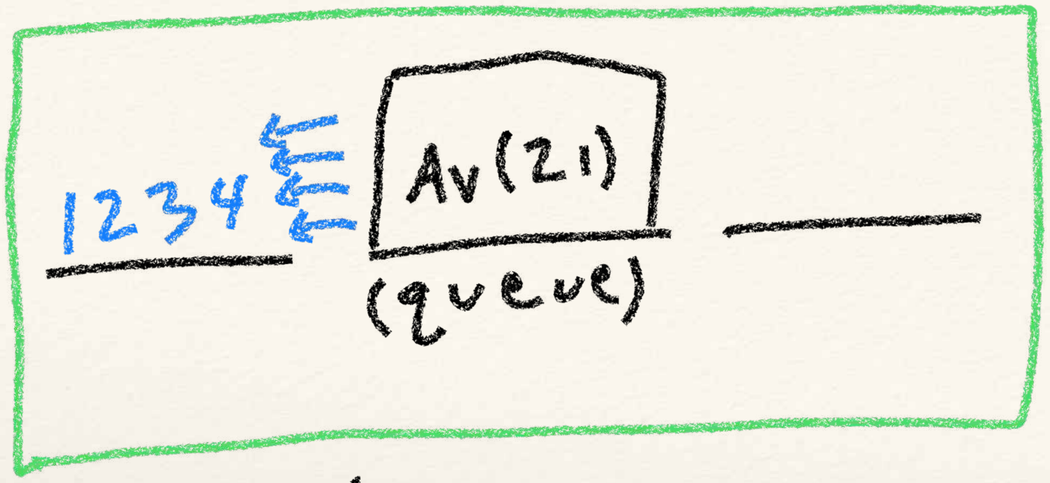
Based on the paper

Albert, Homberger, Pantone, Shar, and Vatter. “Generating permutations with restricted containers”, *J. Combin. Theory Ser. A* **157** (2018), 205–232.

But also based on discussions with

Michael Albert and Jay Pantone





Generating

Sorting with restricted containers

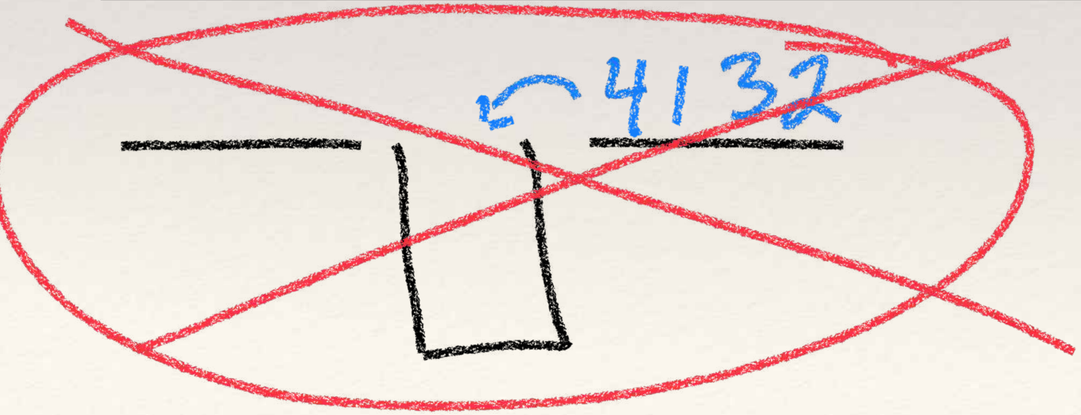
Vince Vatter
University of Florida

Based on the paper

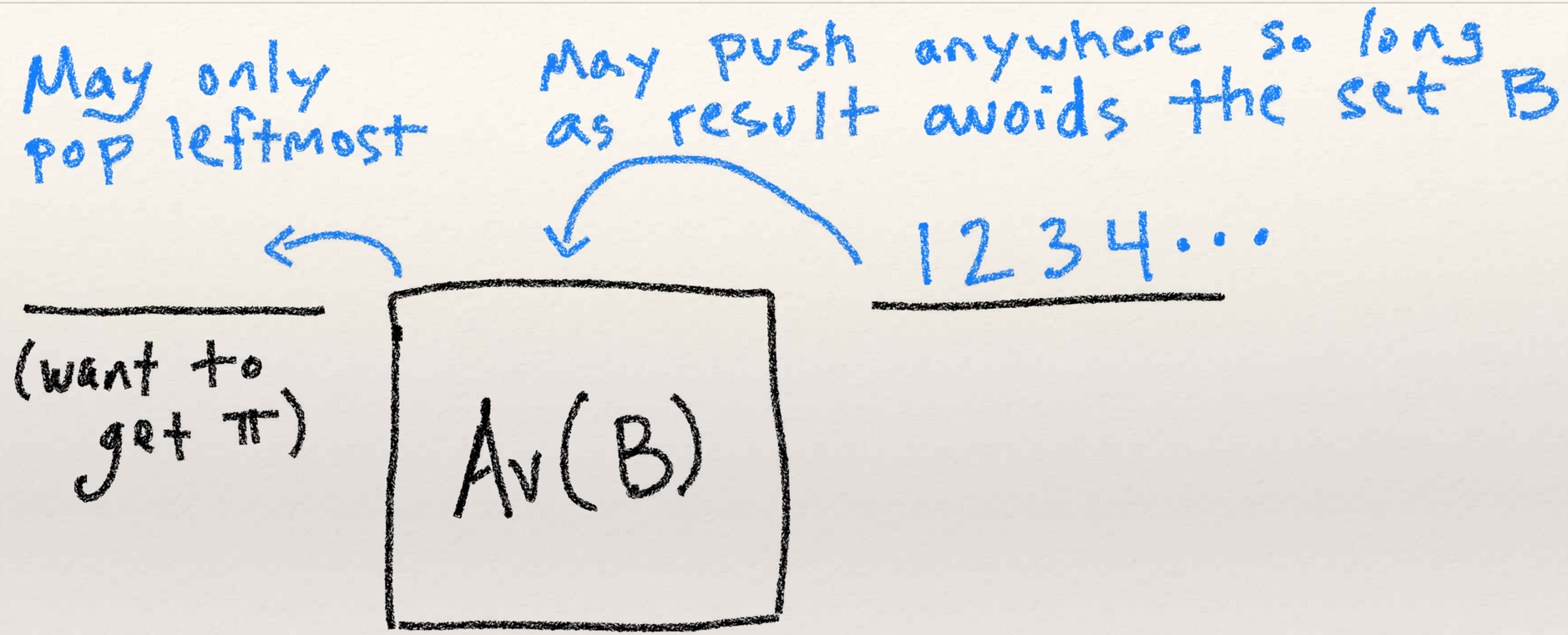
Albert, Homberger, Pantone, Shar, and Vatter. "Generating permutations with restricted containers", *J. Combin. Theory Ser. A* **157** (2018), 205–232.

But also based on discussions with

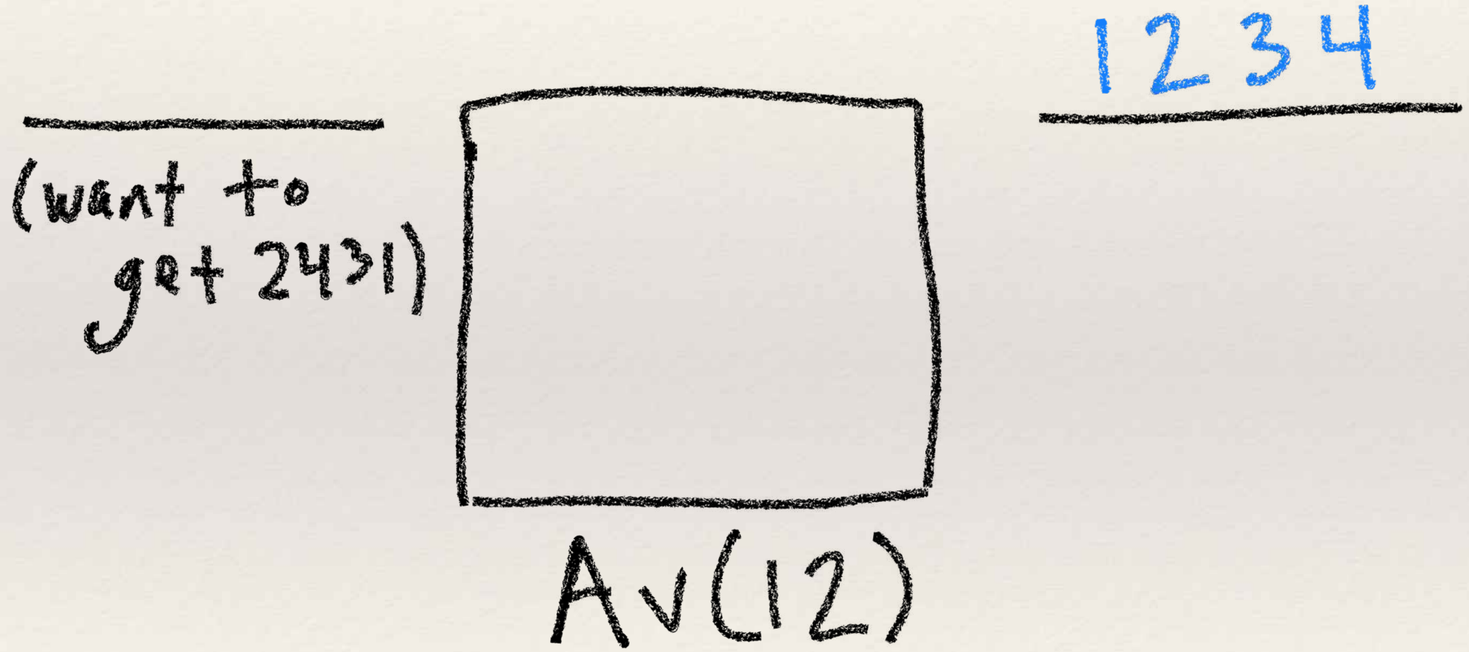
Michael Albert and Jay Pantone



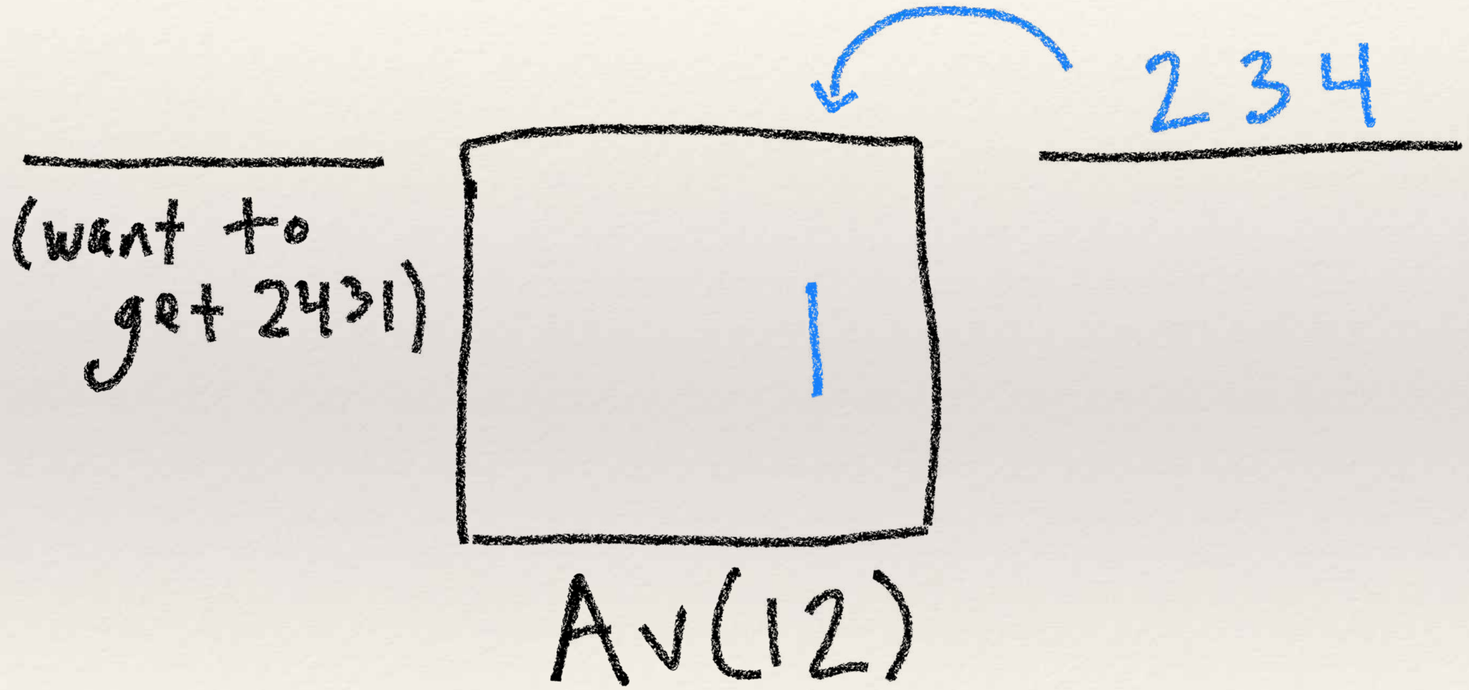
The (initial) restricted container machine



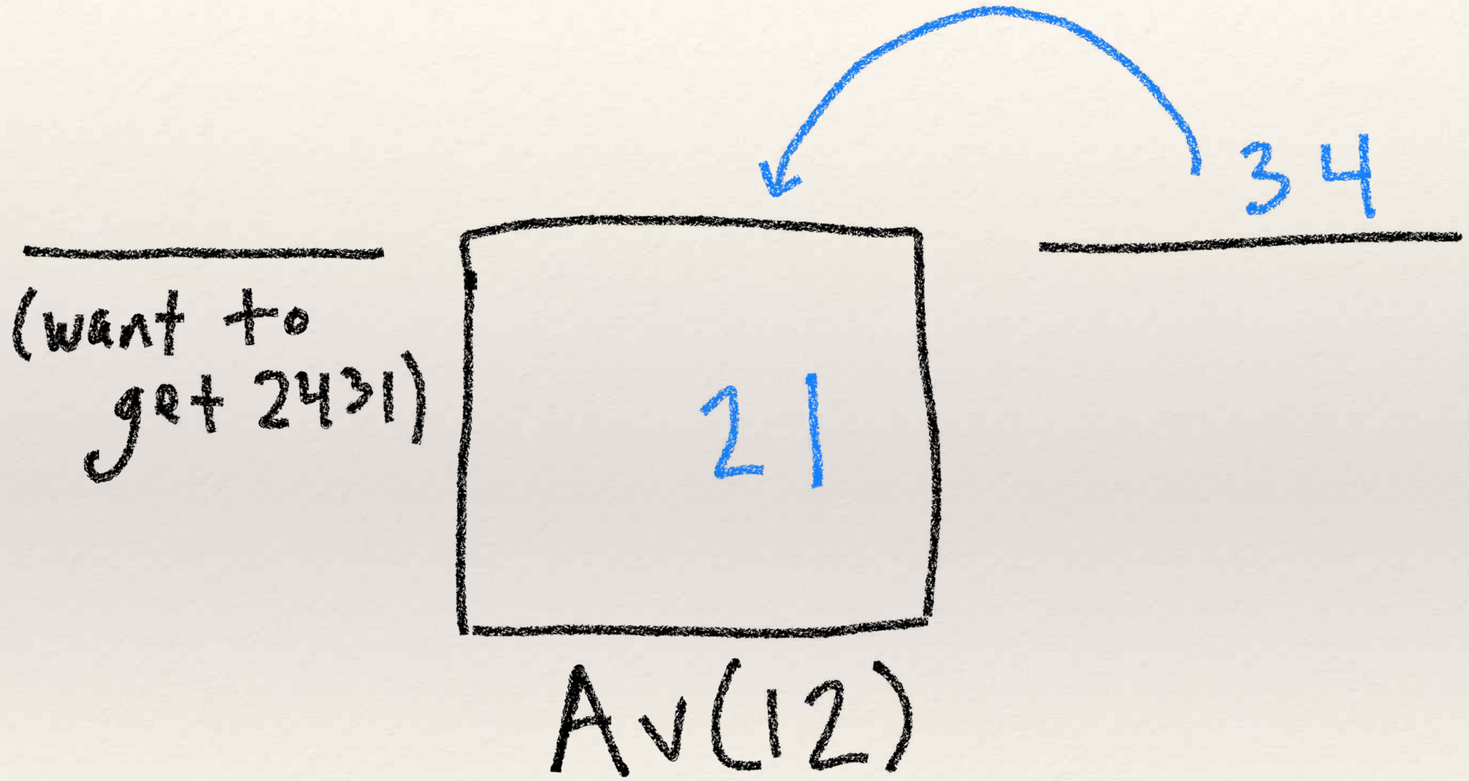
The (initial) restricted container machine



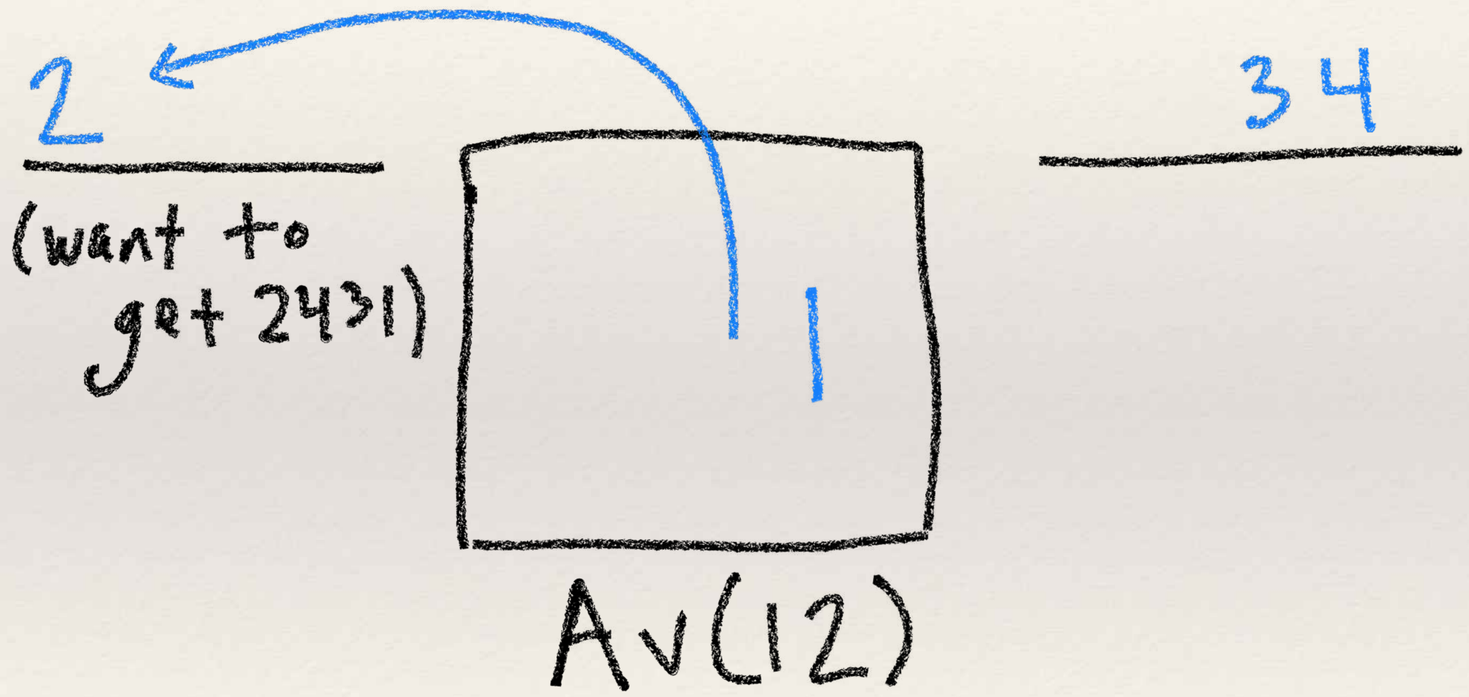
The (initial) restricted container machine



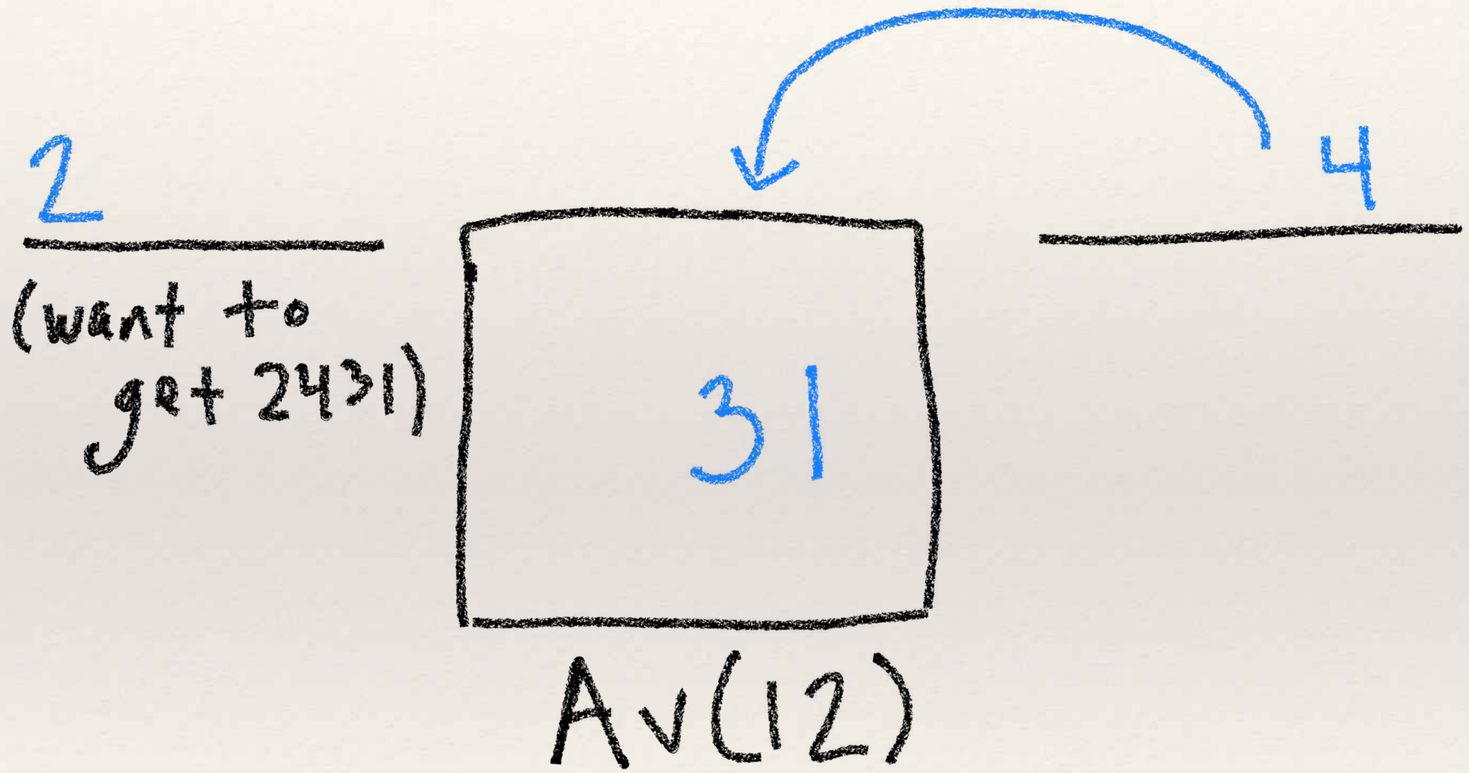
The (initial) restricted container machine



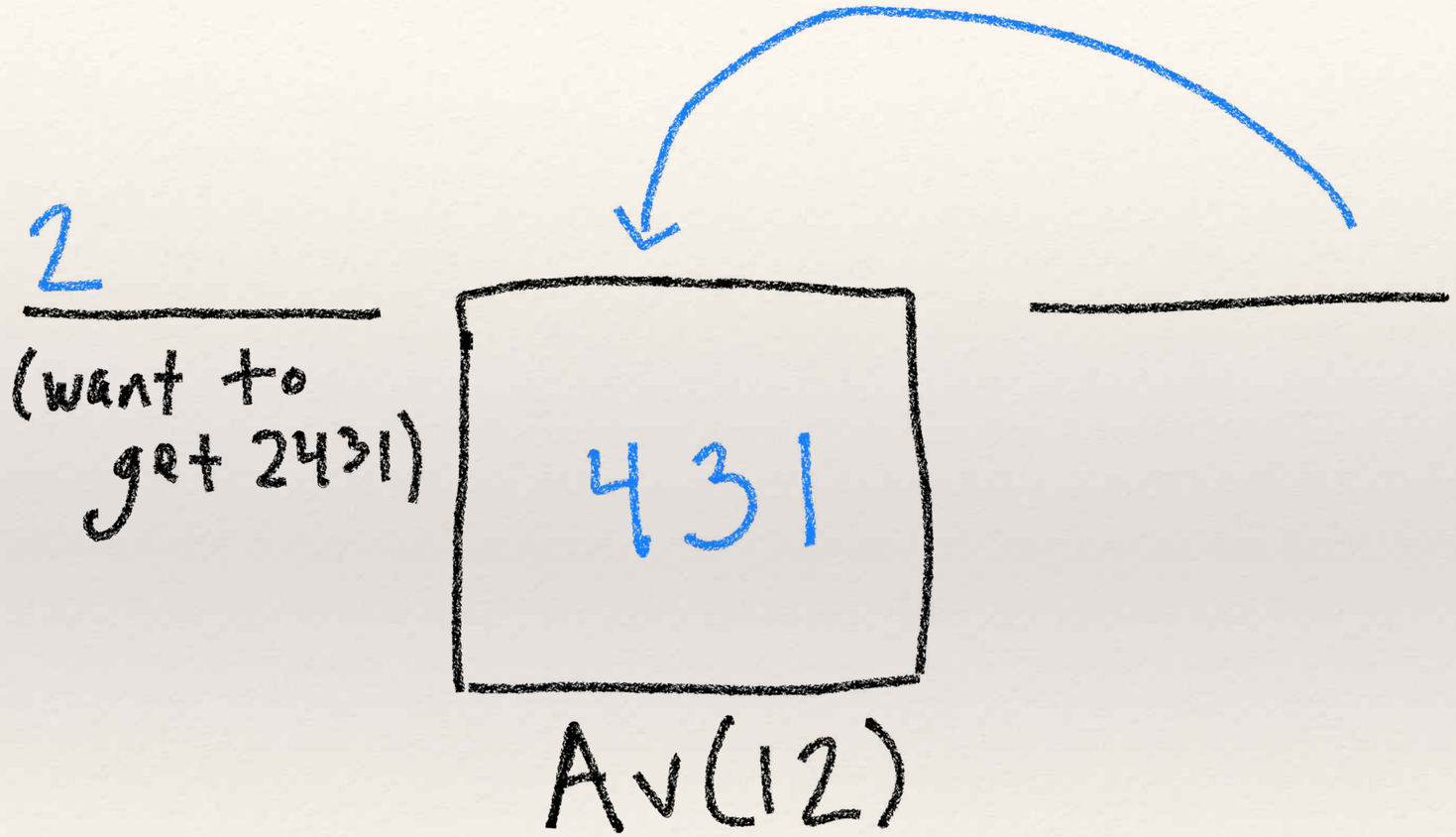
The (initial) restricted container machine



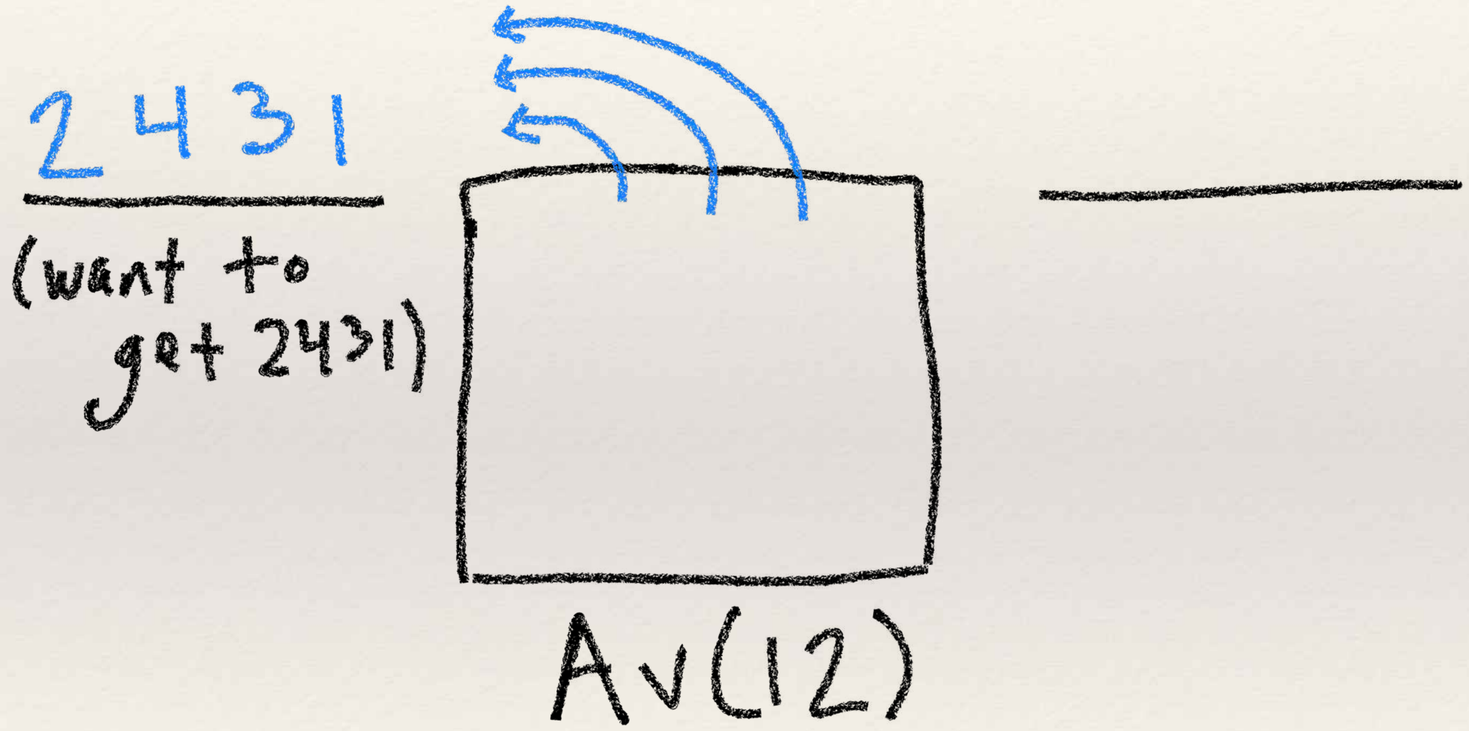
The (initial) restricted container machine



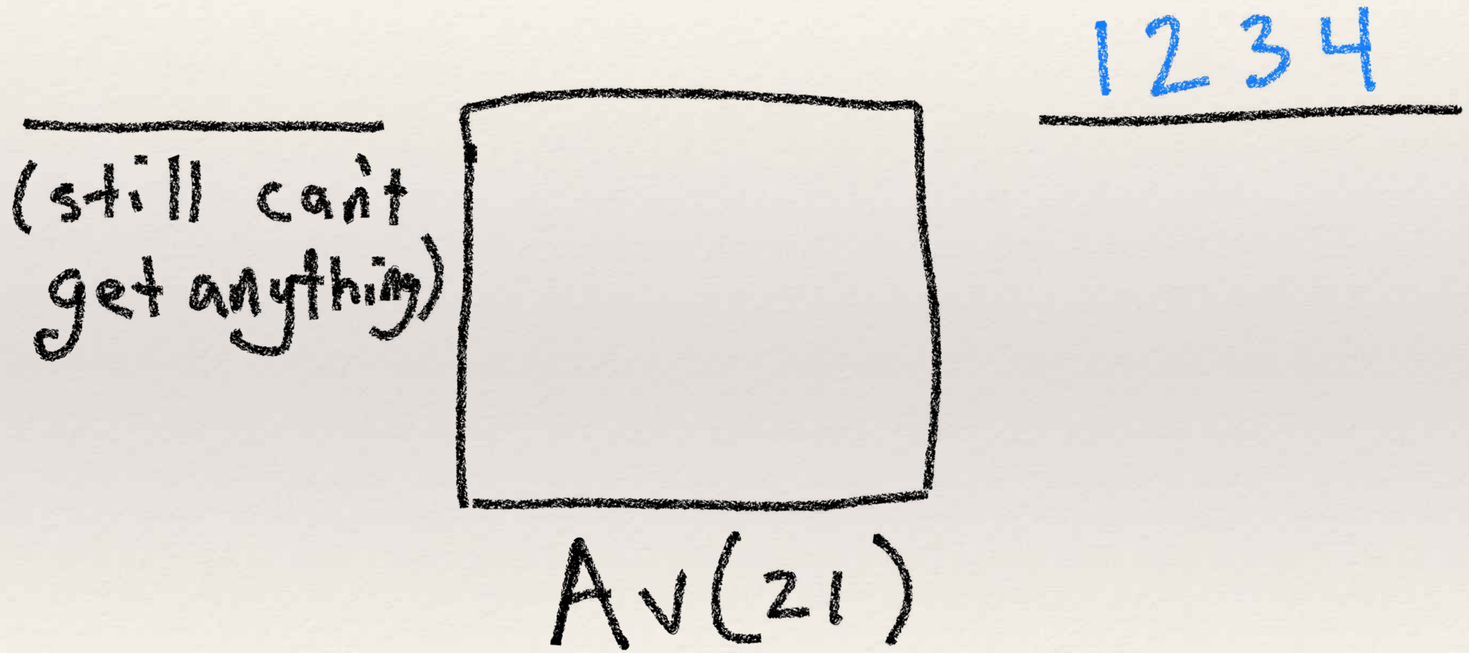
The (initial) restricted container machine



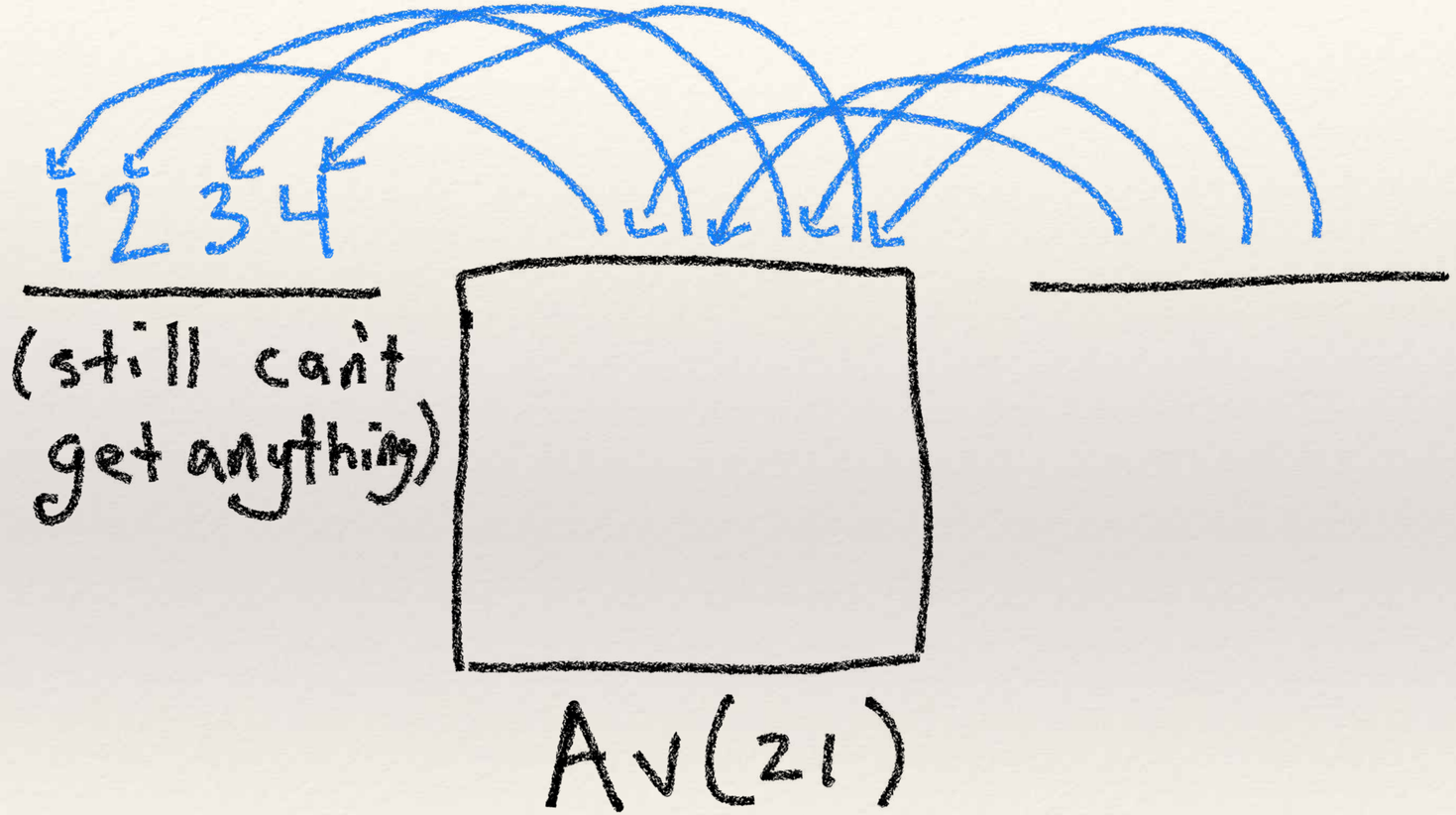
The (initial) restricted container machine



The (initial) restricted container machine



The (initial) restricted container machine



Basis theorem (initial version)

Theorem. If β does not begin with its largest entry, then the $Av(\beta)$-machine generates $Av(1 \ominus \beta)$.

prepend a new maximum
ex: 12 → 312

Basis theorem (initial version)

Theorem. If β does not begin with its largest entry, then the $Av(\beta)$-machine generates $Av(1 \ominus \beta)$.

If β does begin with its largest entry, then the $Av(\beta)$ machine just generates $Av(\beta)$.

prepend a new maximum
ex: 12 → 312

Basis theorem (initial version)

Theorem. If β does **not** begin with its largest entry, then the $Av(\beta)$-machine generates $Av(1 \ominus \beta)$.

If β does begin with its largest entry, then the $Av(\beta)$ machine just generates $Av(\beta)$. ☹

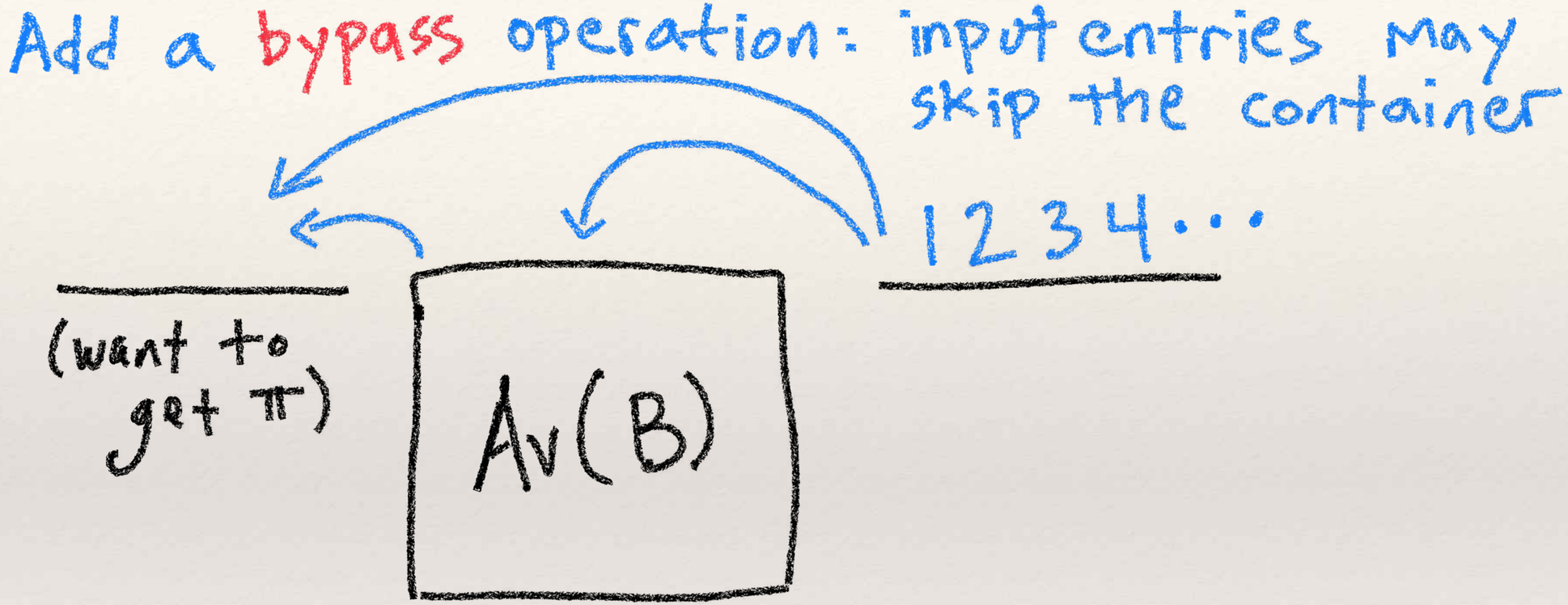
prepend a new maximum
ex: 12 → 312

Examples:

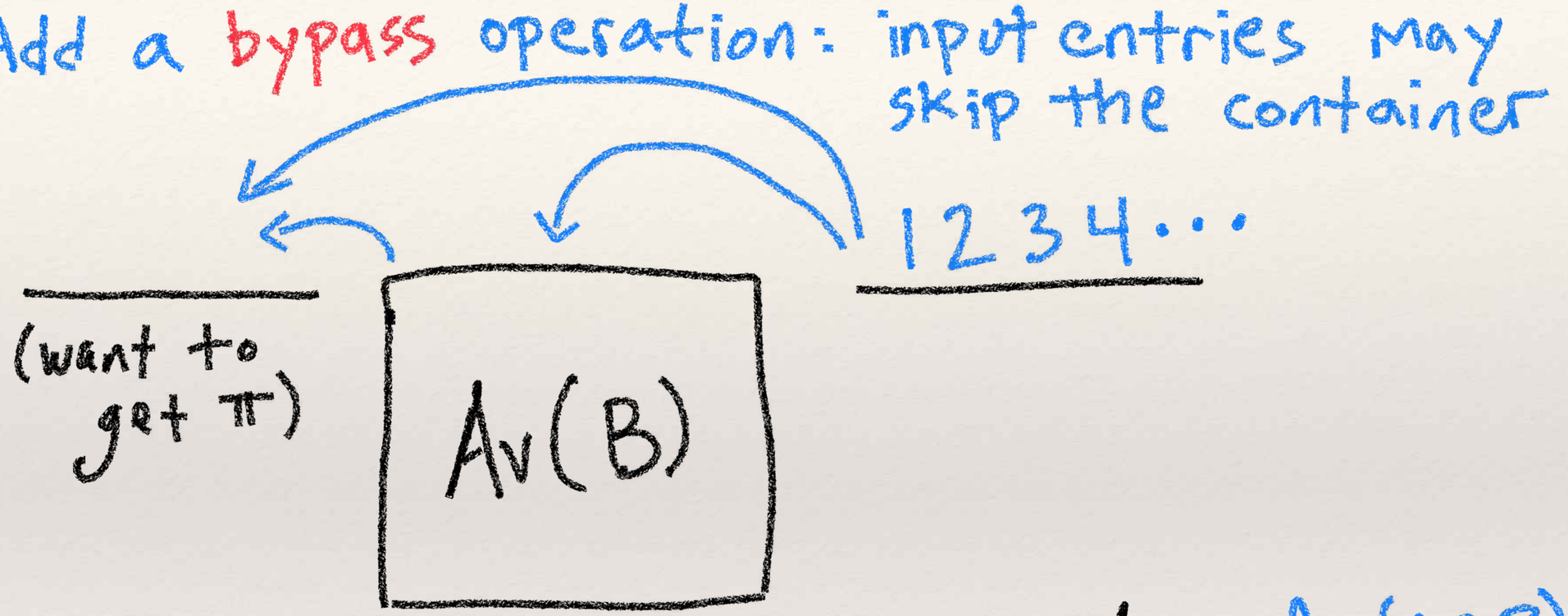
$Av(12)$-machine generates $Av(312)$ — stack

$Av(21)$-machine generates $Av(21)$ — queue

The restricted container machine



The restricted container machine



Theorem. The Av(B)-machine generates Av(1⊕B).

prepend new maximums to every member of B

The restricted container machine

Theorem. The $\mathrm{Av}(B)$-machine generates $\mathrm{Av}(1 \ominus B)$.

Idea. Look at left-to-right maxima of the permutation to generate.

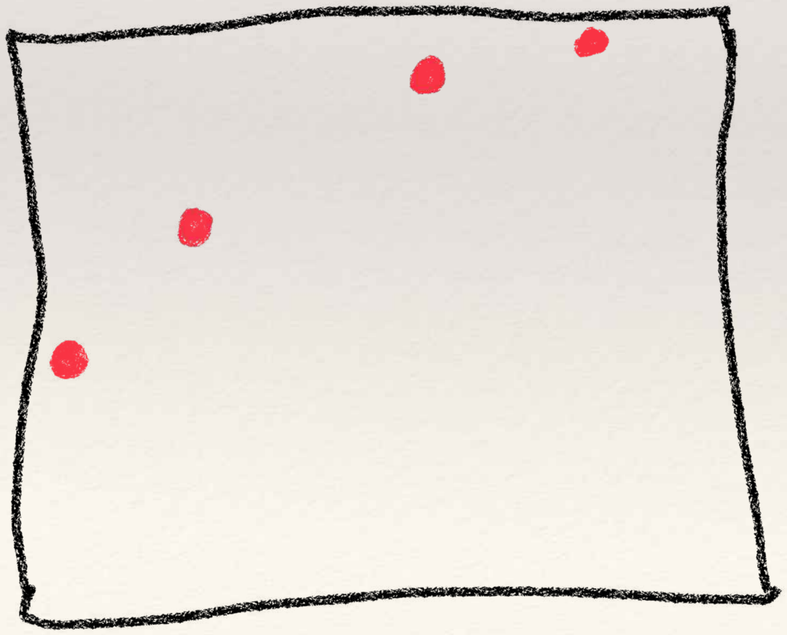
prepend new maximums to every member of B

The restricted container machine

Theorem. The Av(B)-machine generates Av(1⊖B).

Idea. Look at left-to-right maxima of the permutation to generate.

prepend new maximums to every member of B

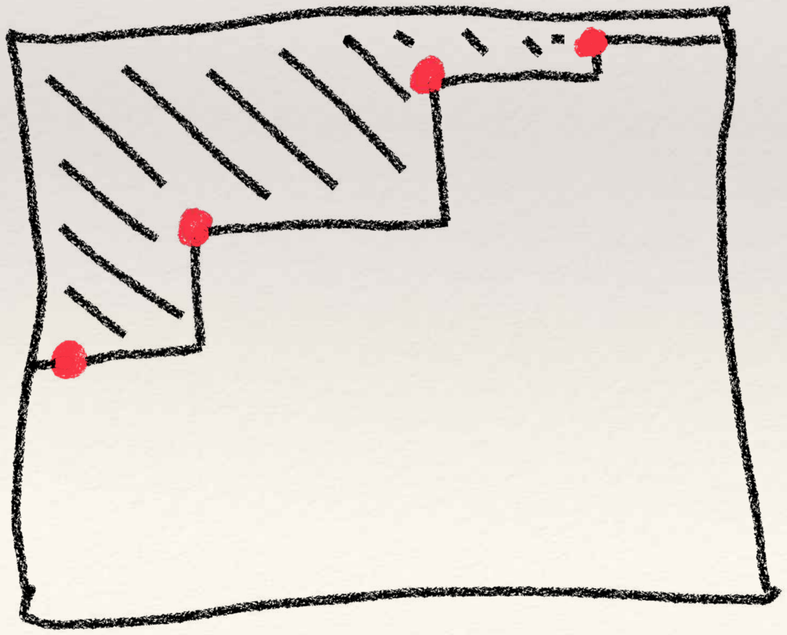


The restricted container machine

Theorem. The $\mathrm{Av}(B)$-machine generates $\mathrm{Av}(1 \ominus B)$.

Idea. Look at left-to-right maxima of the permutation to generate.

prepend new maximums to every member of B

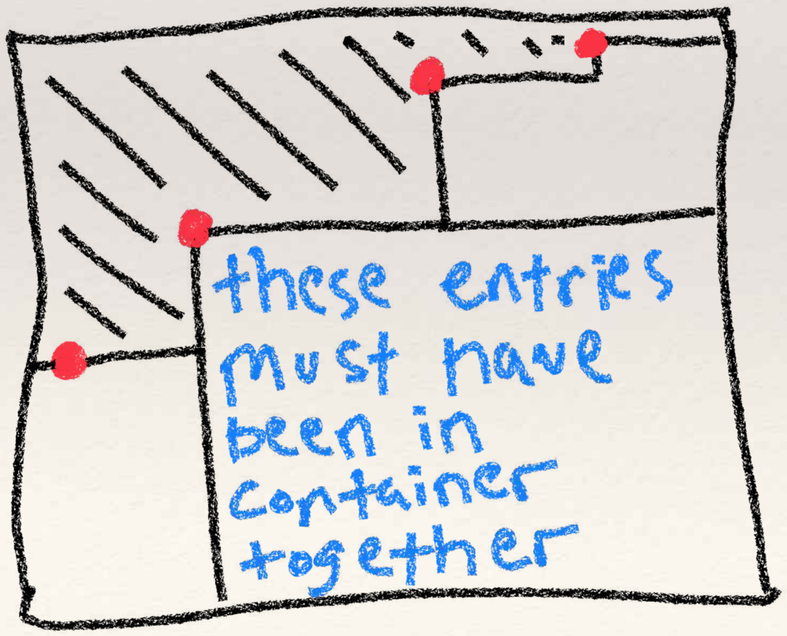


The restricted container machine

Theorem. The $\mathrm{Av}(B)$-machine generates $\mathrm{Av}(1 \ominus B)$.

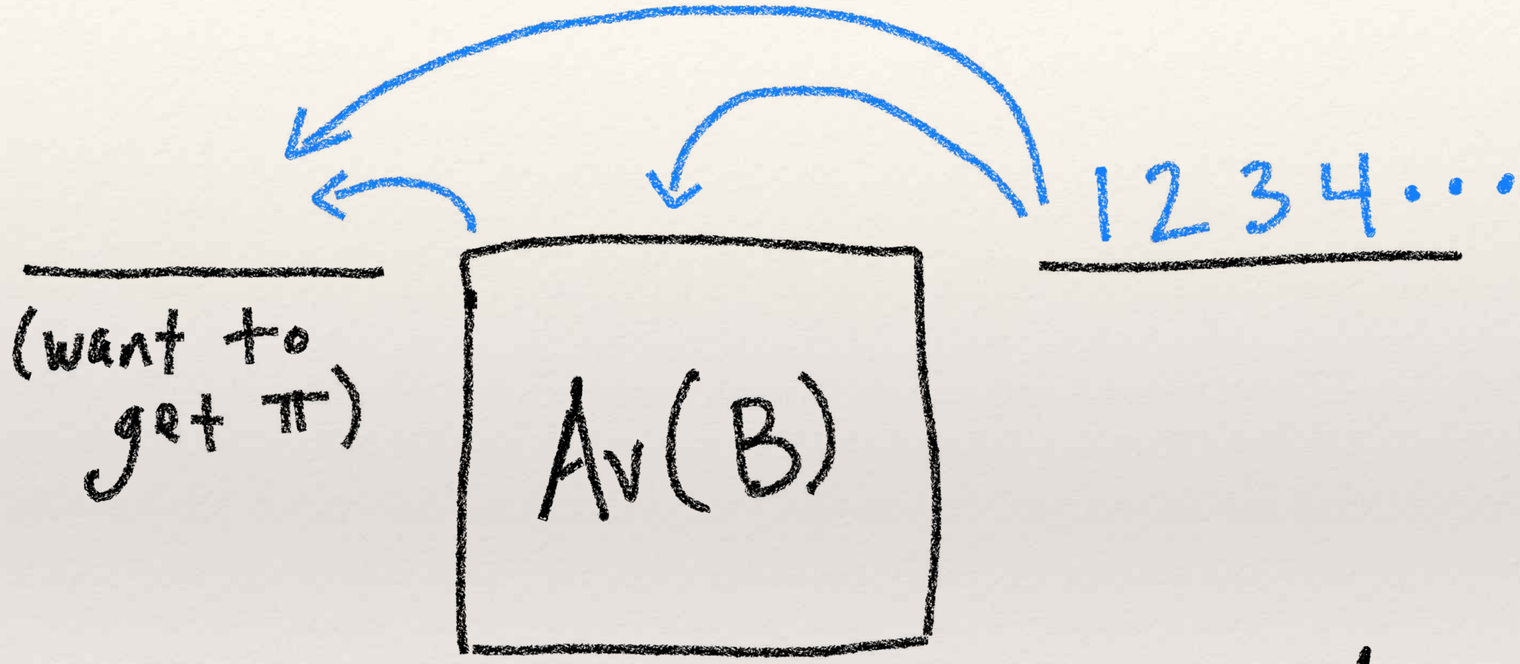
Idea. Look at left-to-right maxima of the permutation to generate.

prepend new maximums to every member of B



May assume all left-to-right maxima were output with bypasses.

The restricted container machine

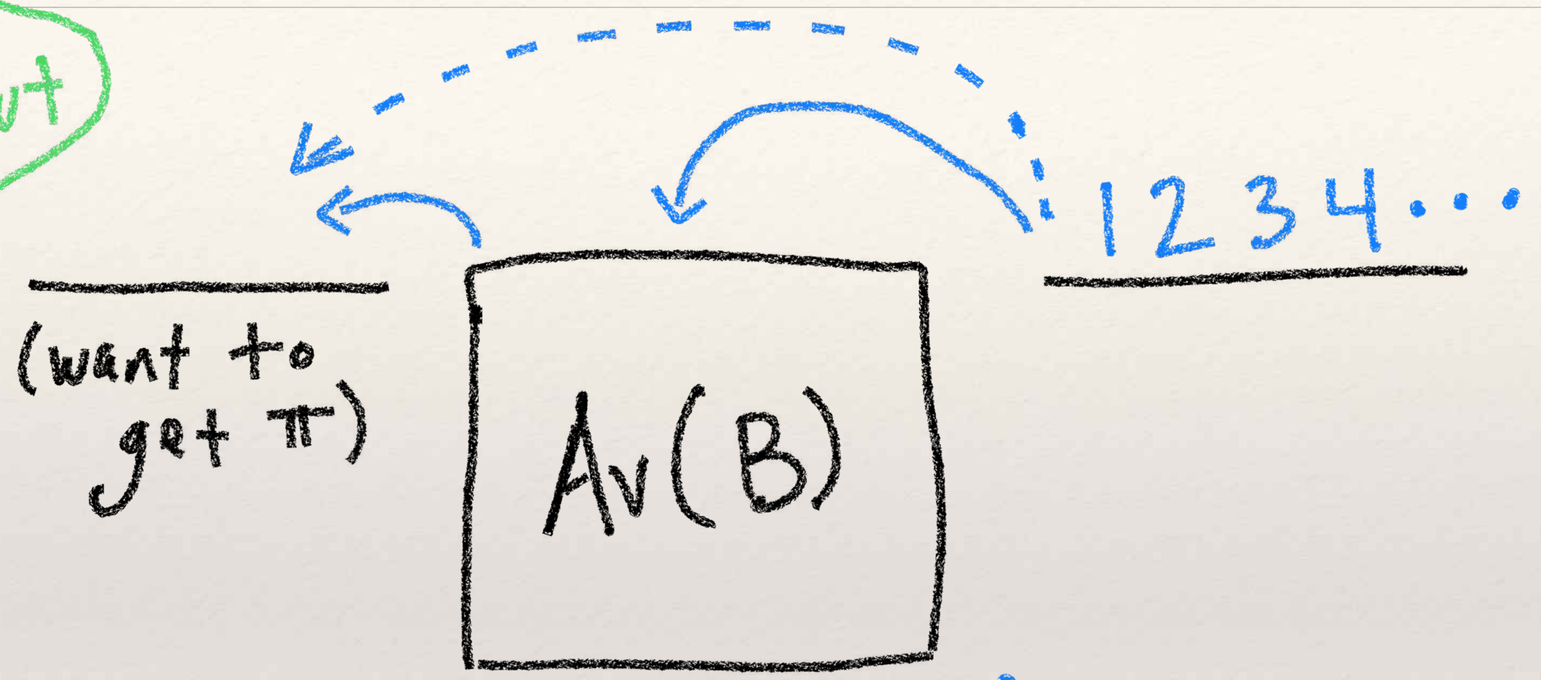


Theorem. The Av(B)-machine generates Av(1⊕B).

prepend new maximums to every member of B

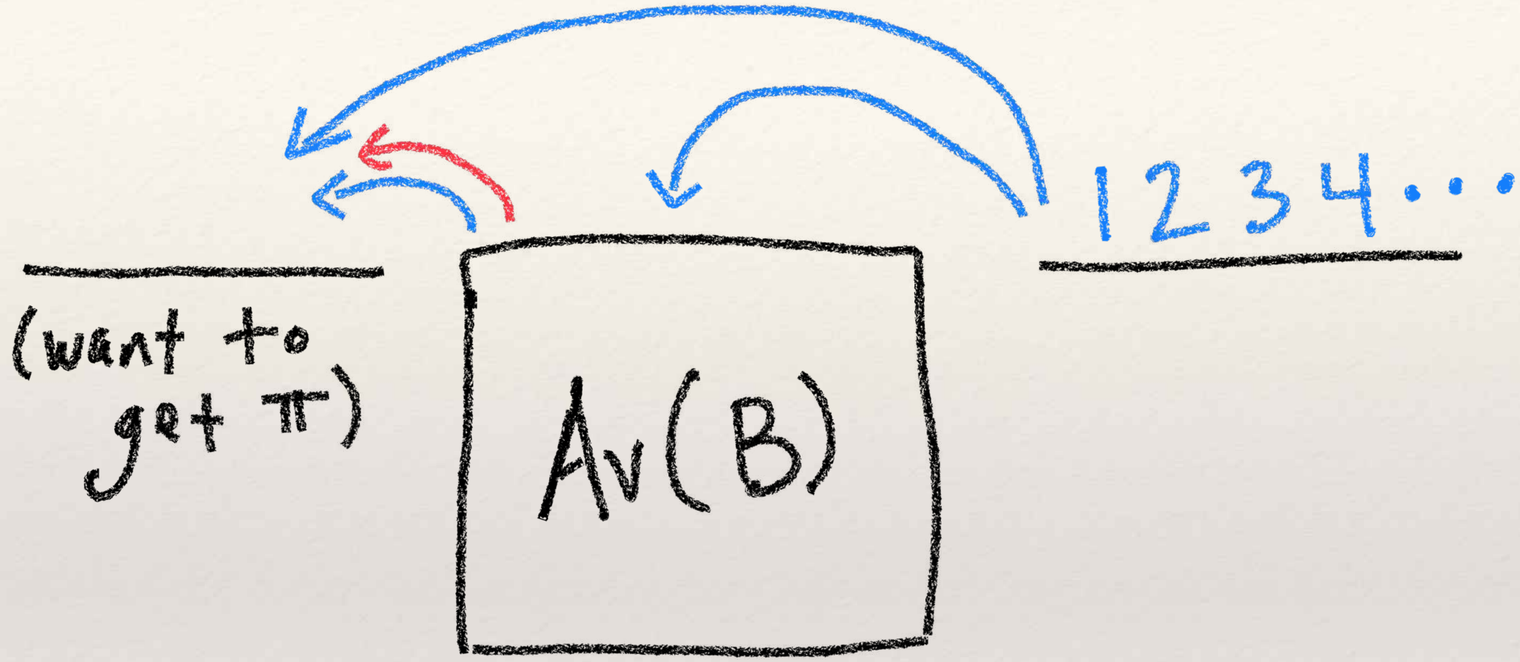
What about a "pop version" (like pop-stacks)

With/without bypasses give different answers!



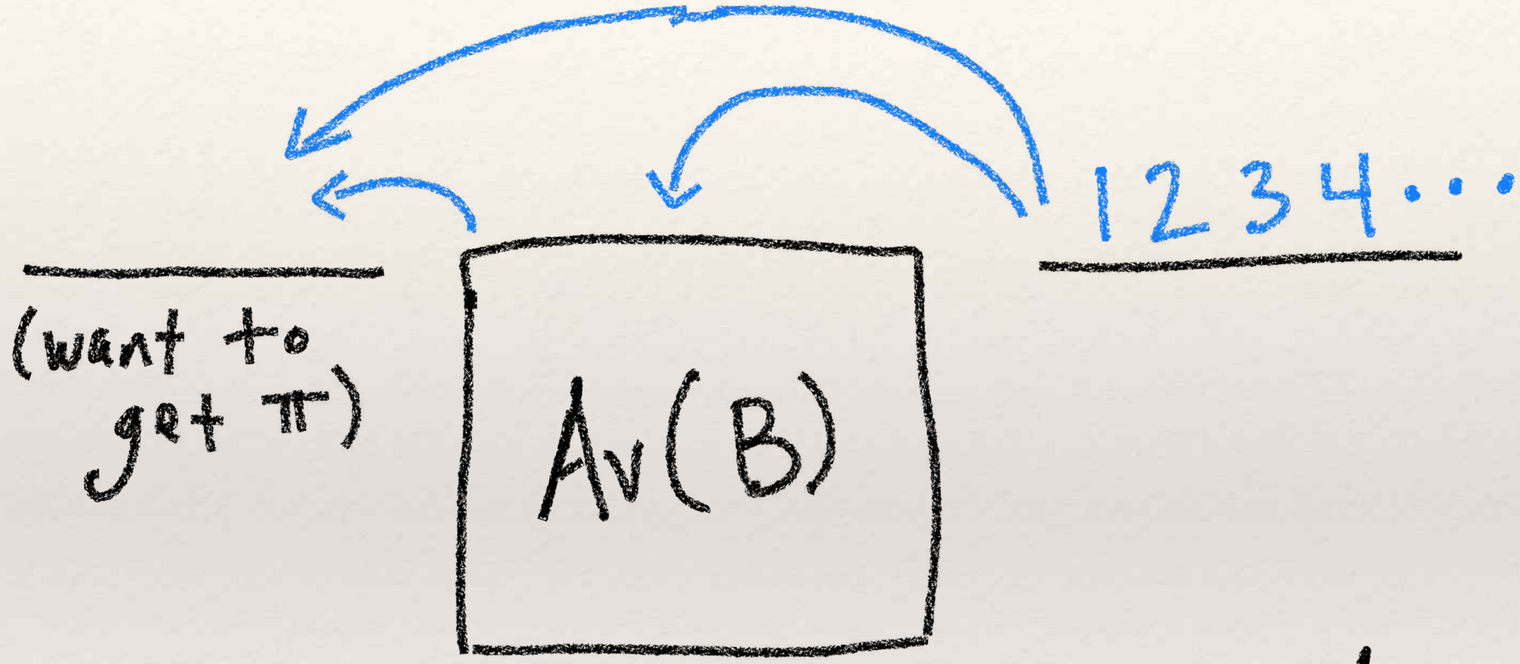
In a pop-Av(B)-machine, if you pop one entry from the container, you must empty it.

What about allowing more pops?



Could allow pops from first or second position.

The restricted container machine



Theorem. The Av(B)-machine generates Av(1⊕B).

prepend new maximums to every member of B

The restricted container machine

Theorem. The $\mathrm{Av}(B)$-machine generates $\mathrm{Av}(1 \oplus B)$.

Easy to see if a class can be generated!

The restricted container machine

Theorem. The $\mathrm{Av}(B)$-machine generates $\mathrm{Av}(1 \oplus B)$.

Easy to see if a class can be generated!

Bóna (PP 2007): Is there a machine that sorts $\mathrm{Av}(1342)$?

The restricted container machine

Theorem. The Av(B)-machine generates Av(1⊕B).

Easy to see if a class can be generated!

Bóna (PP 2007): Is there a machine that sorts Av(1342)?

Answer: Av(1342) ~ Av(4213), and Av(4213) is generated by Av(213)-machine.

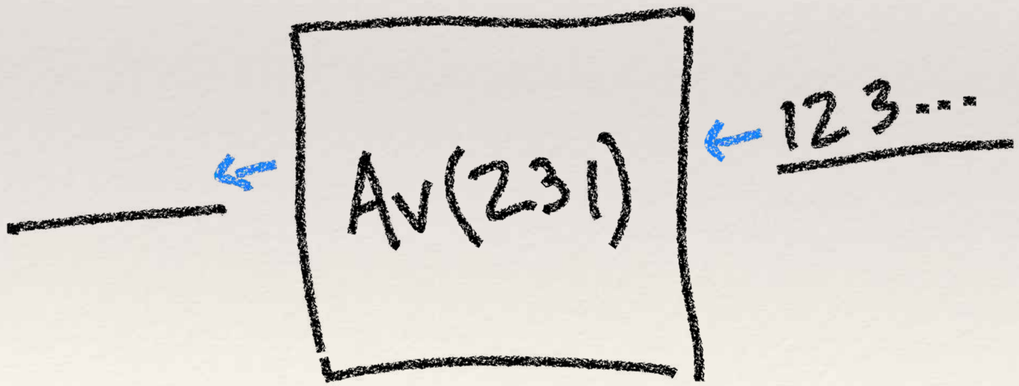
Question: Does this explain bijection to two increasing stacks in series? (Atkinson, Murphy, Ruškuc 2002)

The restricted container machine

Theorem. The Av(B)-machine generates Av(1⊕B).

Easy to see if a class can be generated!

What about Av(4231)?



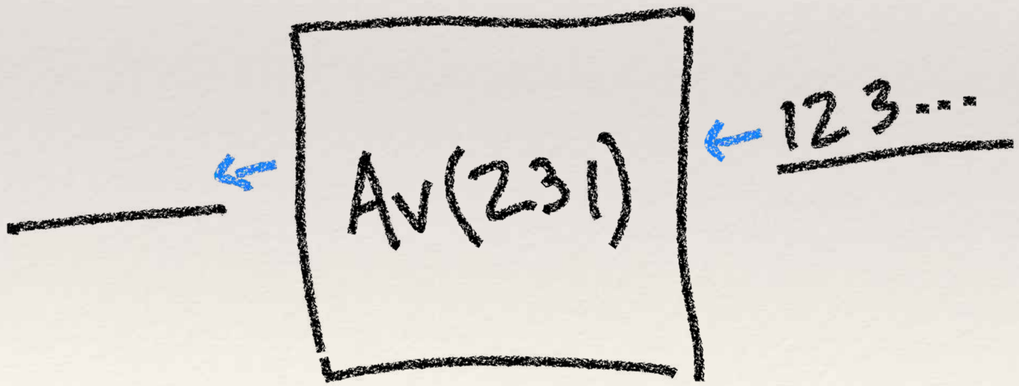
generates Av(4231)

The restricted container machine

Theorem. The Av(B)-machine generates Av(1⊕B).

Easy to see if a class can be generated!

What about Av(4231)?



generates Av(4231)

To compute $|Av_n(4231)|$ this way takes $\approx C_n \approx 4^n$ space and time.

Conway, Guttmann, Zinn-Justin (2018) do this in about 2^n space & time.

There must be a translation of their tricks.

The restricted container machine

Theorem. The Av(B)-machine generates Av(1⊕B).

Easy to see if a class can be generated!

If the container is simple enough...

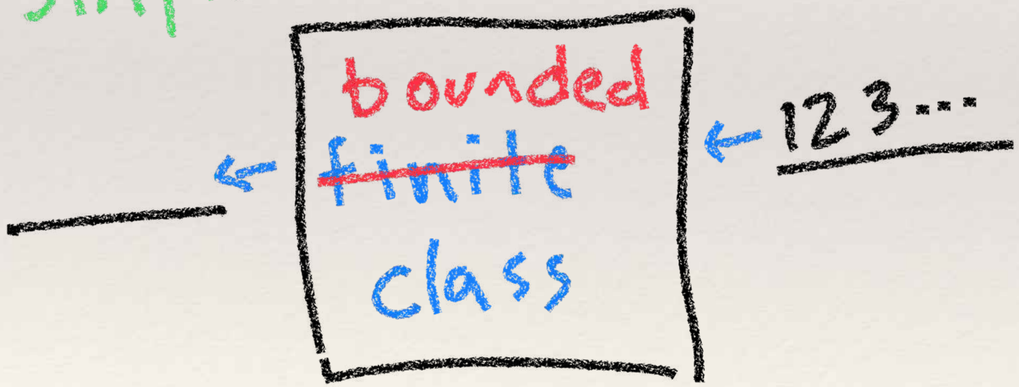
⇛ **Theorem.** generated class has rational generating function

The restricted container machine

Theorem. The Av(B)-machine generates Av(1⊕B).

Easy to see if a class can be generated!

If the container is simple enough...



(bounded: enumeration bounded by constant)

⇒

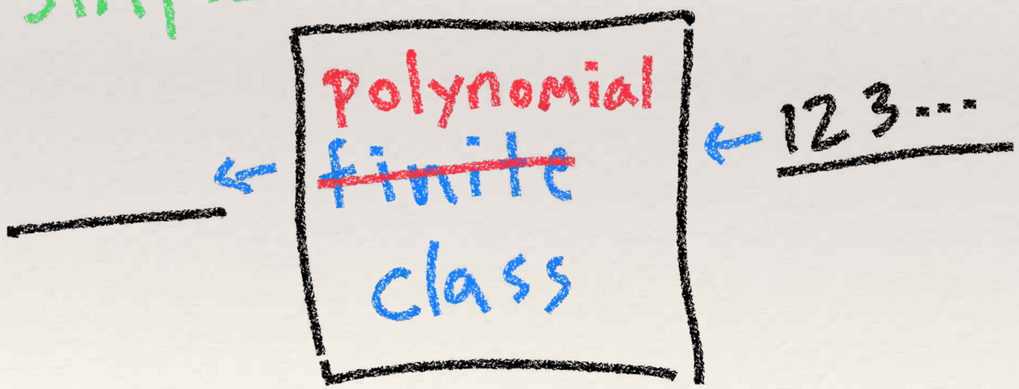
Theorem. generated class has ~~rational~~ algebraic generating function

The restricted container machine

Theorem. The Av(B)-machine generates Av(10B).

Easy to see if a class can be generated!

If the container is simple enough...



(polynomial: enumeration bounded by polynomial)

⇒

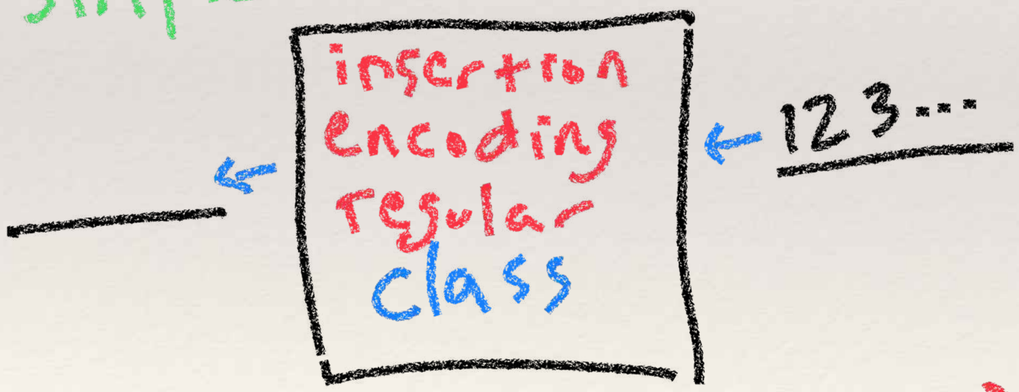
Theorem. ~~generated~~ class has ~~rational~~ ~~generating~~ ~~function~~ poly time enumeration

The restricted container machine

Theorem. The $\mathrm{Av}(B)$-machine generates $\mathrm{Av}(1 \oplus B)$.

Easy to see if a class can be generated!

If the container is simple enough...



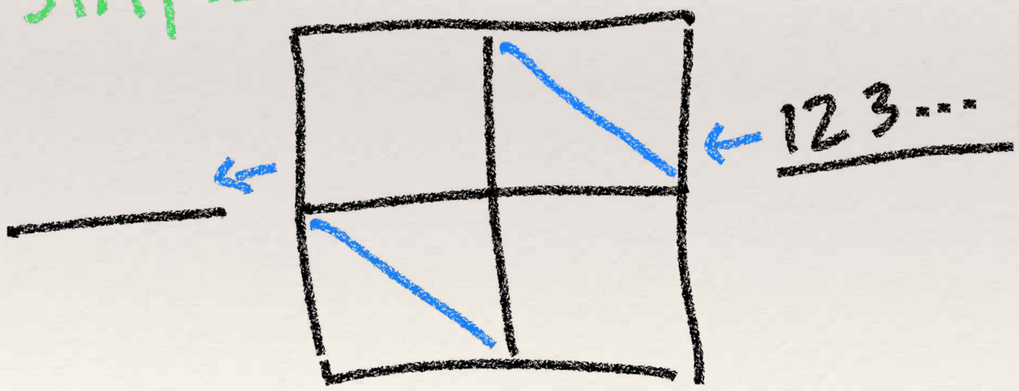
(or anything else nice)

The restricted container machine

Theorem. The Av(B)-machine generates Av(1⊕B).

Easy to see if a class can be generated!

If the container is simple enough...



Av(123, 231, 312)

⇒

5,000 terms for Av(4123, 4231, 4312)

these fit no algebraic differential equation