

# History-Independent Load Balancing

Michael A. Bender<sup>1</sup> Bill Kuszmaul<sup>2</sup> Elaine Shi<sup>2</sup> **Rose Silver<sup>2</sup>**

<sup>1</sup>Stony Brook University

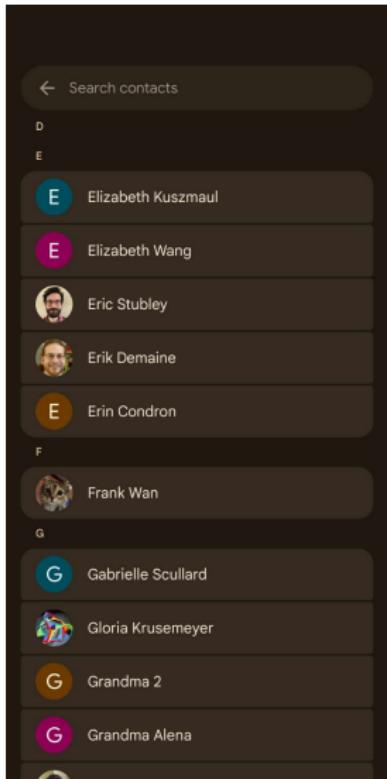
<sup>2</sup>Carnegie Mellon University

# HISTORY INDEPENDENT DATA STRUCTURES

**History Independence:** “If an adversary were to see the state of the data structure, they would learn only the current set of elements, and nothing else about the history of past operations.”

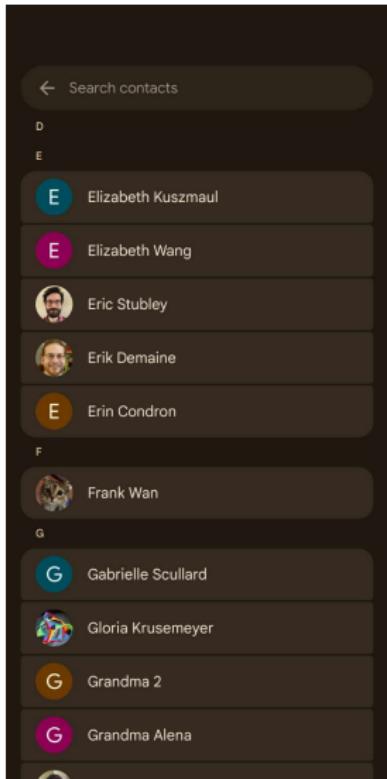
[Micciancio '97], [Naor, Teague '01]

# HISTORY VS CONTENT



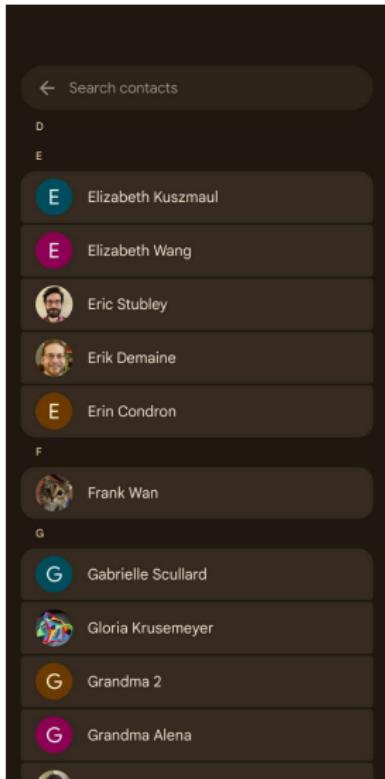
- ▶ If someone hacks my phone, they can learn my contacts list.

# HISTORY VS CONTENT



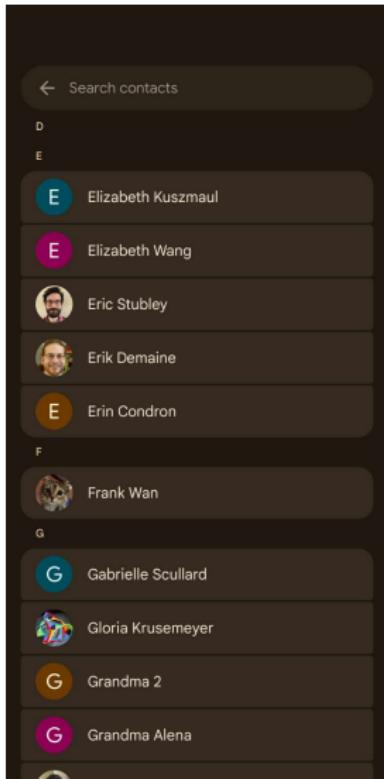
- ▶ If someone hacks my phone, they can learn my contacts list.
- ▶ But can they learn who my contacts were in the past?

# HISTORY VS CONTENT



- ▶ If someone hacks my phone, they can learn my contacts list.
- ▶ But can they learn who my contacts were in the past?
- ▶ What about the order in which contacts were added?

# HISTORY VS CONTENT



- ▶ If someone hacks my phone, they can learn my contacts list.
- ▶ But can they learn who my contacts were in the past?
- ▶ What about the order in which contacts were added?
- ▶ A history independent data structure protects this kind of information.

## HISTORY INDEPENDENT IS A SECURITY GUARANTEE

**History Independence:** “If an adversary were to see the state of the data structure, they would learn only the current set of elements, and nothing else about the history of past operations.”

[Micciancio '97], [Naor, Teague '01]

# HISTORY INDEPENDENT IS A SECURITY GUARANTEE

**History Independence:** “If an adversary were to see the state of the data structure, they would learn only the current set of elements, and nothing else about the history of past operations.”

[Micciancio '97], [Naor, Teague '01]

**Lost of successes:** Hash tables, trees, memory allocation, PMAs, graph algorithms, B-trees, cache-oblivious data structures...

[Micciancio '97], [Naor, Teague '01], [Buchbinder, Petrank '03], [Molnar, Kohno, Sastry, Wagner '06], [Blelloch, Golovin '07], [Moran, Naor, Segev '07] [Naor, Segev, Wieder '08], [Golovin '08 '09 '10], [Tzouramanis '12], [Bajaj, Sion '13] [Bajaj, Chakrabati, Sion '15], [Roche, Aviv, Choi '15], [Bender, Berry, Johnson, Kroeger, McCauley, Phillips, Simon, Singh, Zage '16], ...

# HISTORY INDEPENDENT IS A SECURITY GUARANTEE

**History Independence:** “If an adversary were to see the state of the data structure, they would learn only the current set of elements, and nothing else about the history of past operations.”

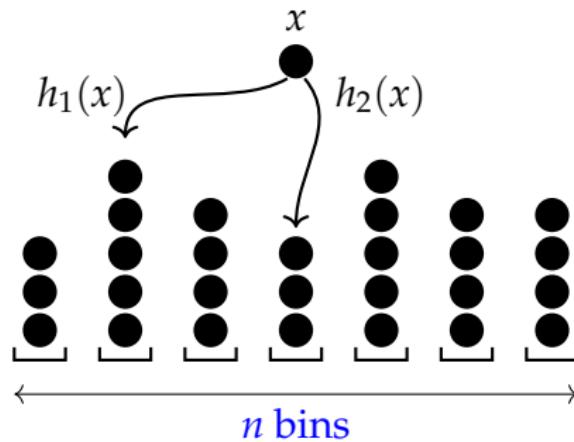
[Micciancio '97], [Naor, Teague '01]

**Lost of successes:** Hash tables, trees, memory allocation, PMAs, graph algorithms, B-trees, cache-oblivious data structures...

[Micciancio '97], [Naor, Teague '01], [Buchbinder, Petrank '03], [Molnar, Kohno, Sastry, Wagner '06], [Blelloch, Golovin '07], [Moran, Naor, Segev '07] [Naor, Segev, Wieder '08], [Golovin '08 '09 '10], [Tzouramanis '12], [Bajaj, Sion '13] [Bajaj, Chakrabati, Sion '15], [Roche, Aviv, Choi '15], [Bender, Berry, Johnson, Kroeger, McCauley, Phillips, Simon, Singh, Zage '16], ...

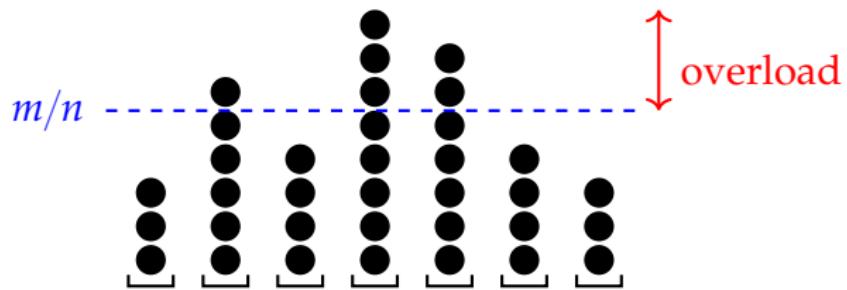
**But... some very basic questions also remain open.**

# TWO-CHOICE LOAD BALANCING



- ▶ Balls are **inserted/deleted**, with up to  $m$  present at a time.
- ▶ Each ball has two random bins where it can go.
- ▶ We must maintain a valid assignment of balls to bins.

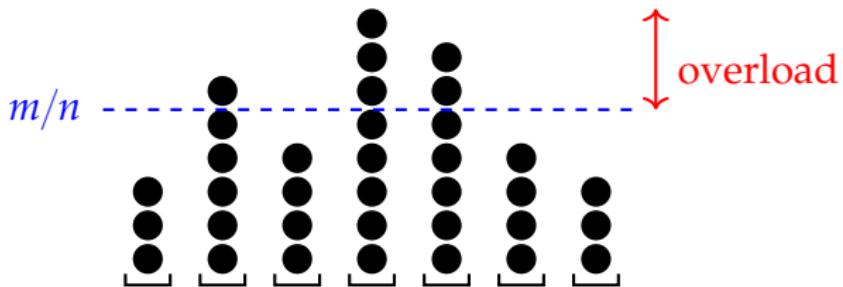
## TWO GOALS



### Minimize Overload:

The amount by which the fullest bin exceeds  $m/n$  is small.

## TWO GOALS



### Minimize Overload:

The amount by which the fullest bin exceeds  $m/n$  is small.

### Minimize Recourse:

On any given insertion/deletion, the number of balls moved around is small.

# THIS PAPER

**Question:** Does there exist a history-independent solution with small recourse and overload?

# THIS PAPER

**Question:** Does there exist a history-independent solution with small recourse and overload?

**Theorem:** There exists a history-independent solution with:

- ▶ Overload  $O(1)$ , with high probability.
- ▶ Expected recourse  $O(\log \log(m/n))$ .

# WHAT ABOUT NON-HISTORY-INDEPENDENT SOLUTIONS?

Lots of work on the insertion-only case.

[Azar, Broder, Karlin and Upfal '94] [Berenbrink, Czumaj, Steger, and Vöcking '00][Dietzfelbinger and Weidling '07]  
[Frieze and Petti '18] ...

# WHAT ABOUT NON-HISTORY-INDEPENDENT SOLUTIONS?

Lots of work on the insertion-only case.

[Azar, Broder, Karlin and Upfal '94] [Berenbrink, Czumaj, Steger, and Vöcking '00][Dietzfelbinger and Weidling '07]  
[Frieze and Petti '18] ...

But the fully dynamic case has remained largely open.

[Vöcking '99] [Dietzfelbinger and Weidling '07] [Bender, Conway, Farach-Colton, Kuszmaul, Tagliavini '21] [Bansal, Kuszmaul '22] ...

# WHAT ABOUT NON-HISTORY-INDEPENDENT SOLUTIONS?

Lots of work on the insertion-only case.

[Azar, Broder, Karlin and Upfal '94] [Berenbrink, Czumaj, Steger, and Vöcking '00][Dietzfelbinger and Weidling '07]  
[Frieze and Petti '18] ...

But the fully dynamic case has remained largely open.

[Vöcking '99] [Dietzfelbinger and Weidling '07] [Bender, Conway, Farach-Colton, Kuszmaul, Tagliavini '21] [Bansal, Kuszmaul '22] ...

**Open Question:**

Is there a **fully dynamic** solution with **recourse  $o(m/n)$**  and **overload  $O(1)$** ?

# WHAT ABOUT NON-HISTORY-INDEPENDENT SOLUTIONS?

Lots of work on the insertion-only case.

[Azar, Broder, Karlin and Upfal '94] [Berenbrink, Czumaj, Steger, and Vöcking '00][Dietzfelbinger and Weidling '07]  
[Frieze and Petti '18] ...

But the fully dynamic case has remained largely open.

[Vöcking '99] [Dietzfelbinger and Weidling '07] [Bender, Conway, Farach-Colton, Kuszmaul, Tagliavini '21] [Bansal, Kuszmaul '22] ...

## Open Question:

Is there a **fully dynamic** solution with **recourse  $o(m/n)$**  and **overload  $O(1)$** ?

## Answer:

Yes! We get **recourse  $O(\log \log(m/n))$**  and **overload  $O(1)$** !

# THIS PAPER

**Question:** Does there exist a history-independent solution with small recourse and overload?

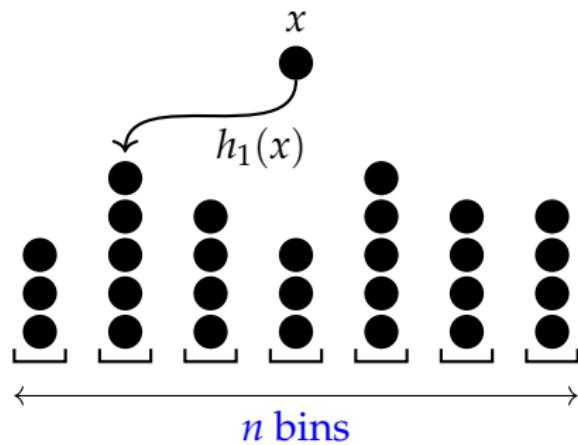
**Theorem:** There exists a history-independent solution with:

- ▶ Overload  $O(1)$ , with high probability.
- ▶ Expected recourse  $O(\log \log(m/n))$ .

**Rest of Talk:**  
Outlining a Solution with  
Overload  $O(\log \log n)$   
and Expected Recourse  $O(m/n)$ .

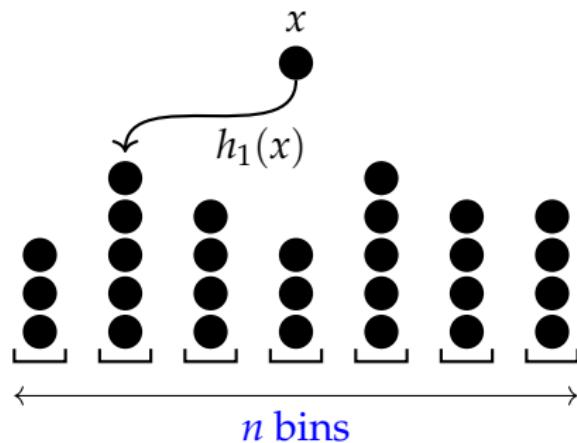
## WARMUP 1: THE SINGLE-CHOICE STRATEGY

To insert a ball  $x$ , just put it in bin  $h_1(x)$ :



## WARMUP 1: THE SINGLE-CHOICE STRATEGY

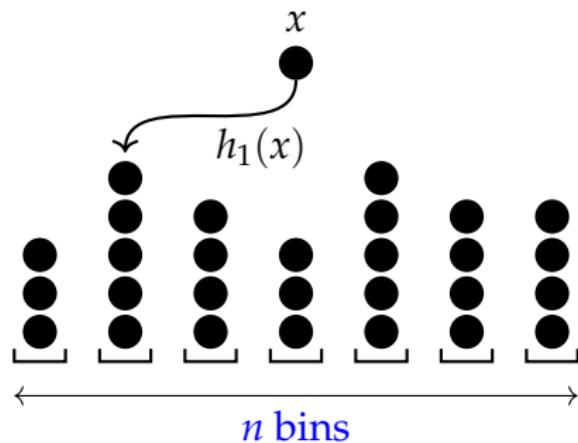
To insert a ball  $x$ , just put it in bin  $h_1(x)$ :



- ▶ This is history-independent ✓

## WARMUP 1: THE SINGLE-CHOICE STRATEGY

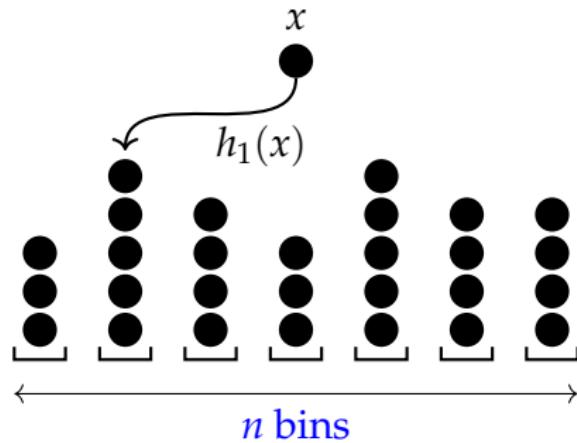
To insert a ball  $x$ , just put it in bin  $h_1(x)$ :



- ▶ This is history-independent ✓
- ▶ The recourse is 0 ✓

## WARMUP 1: THE SINGLE-CHOICE STRATEGY

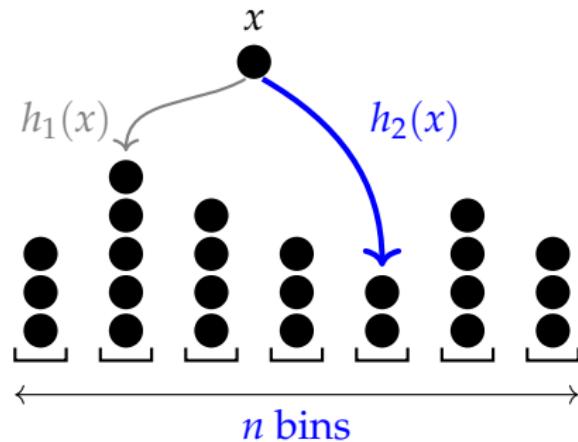
To insert a ball  $x$ , just put it in bin  $h_1(x)$ :



- ▶ This is history-independent ✓
- ▶ The recourse is 0 ✓
- ▶ But... the overload is huge, roughly  $\sqrt{m/n}$  ✗

## WARMUP 2: GREEDY INSERTIONS

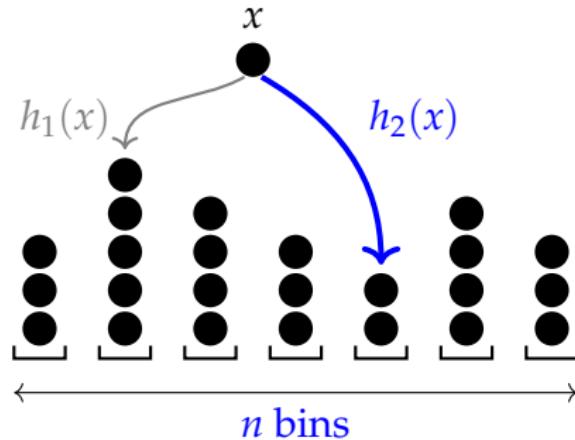
To insert a ball  $x$ , put it in the **emptier** of its choices:



- ▶ This is **not** history-independent  $\times$

## WARMUP 2: GREEDY INSERTIONS

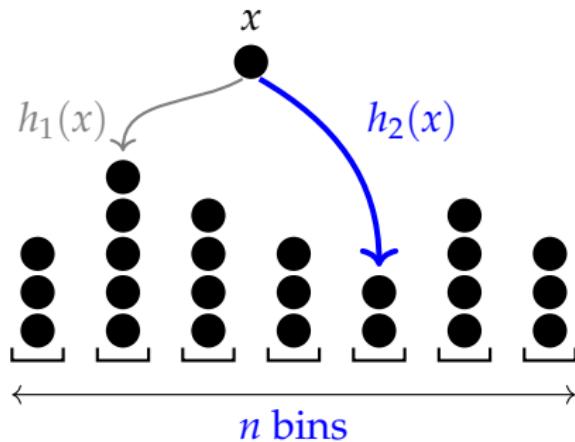
To insert a ball  $x$ , put it in the **emptier** of its choices:



- ▶ This is **not** history-independent ✗
- ▶ The recourse is 0 ✓

## WARMUP 2: GREEDY INSERTIONS

To insert a ball  $x$ , put it in the **emptier** of its choices:

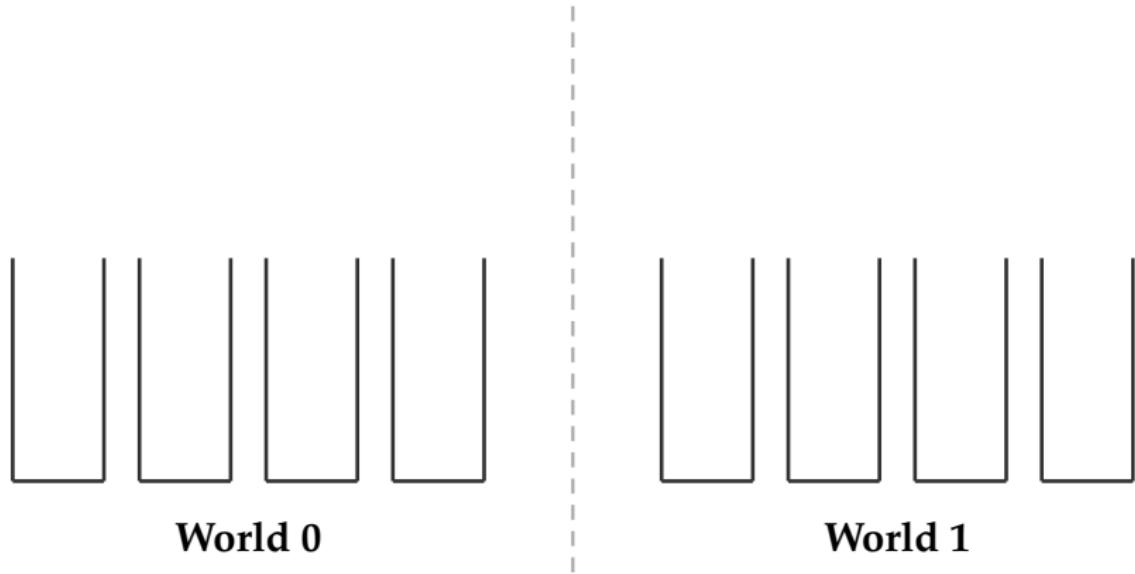


- ▶ This is **not** history-independent ✗
- ▶ The recourse is 0 ✓
- ▶ In the insertion-only case, the overload is  $O(\log \log n)$  ✓

[Azar, Broder, Karlin and Upfal '94]

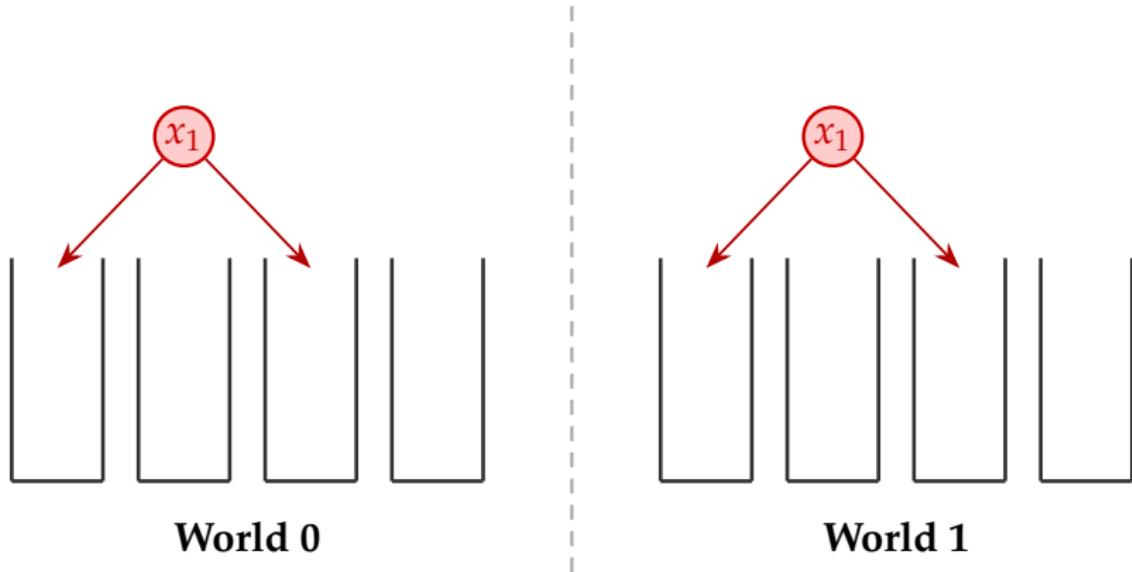
# TURNING GREEDY INTO A HISTORY-INDEPENDENT SOLUTION

## ANALYZING THE RE COURSE



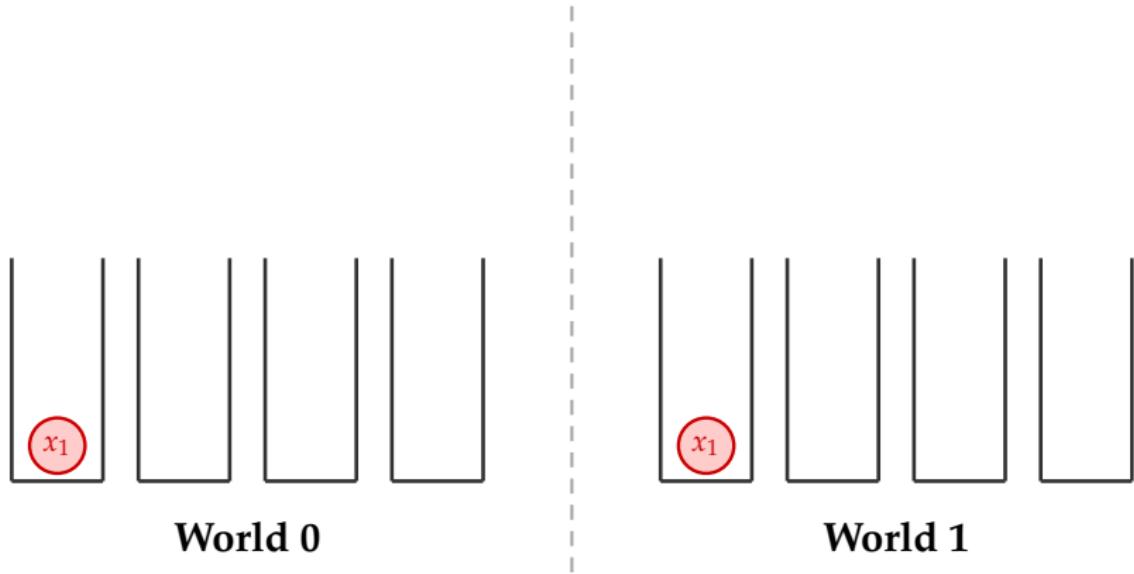
- ▶ Two identical worlds, same insertion sequence
- ▶ World 1 will later receive one extra ball  $x^*$

## ANALYZING THE RE COURSE



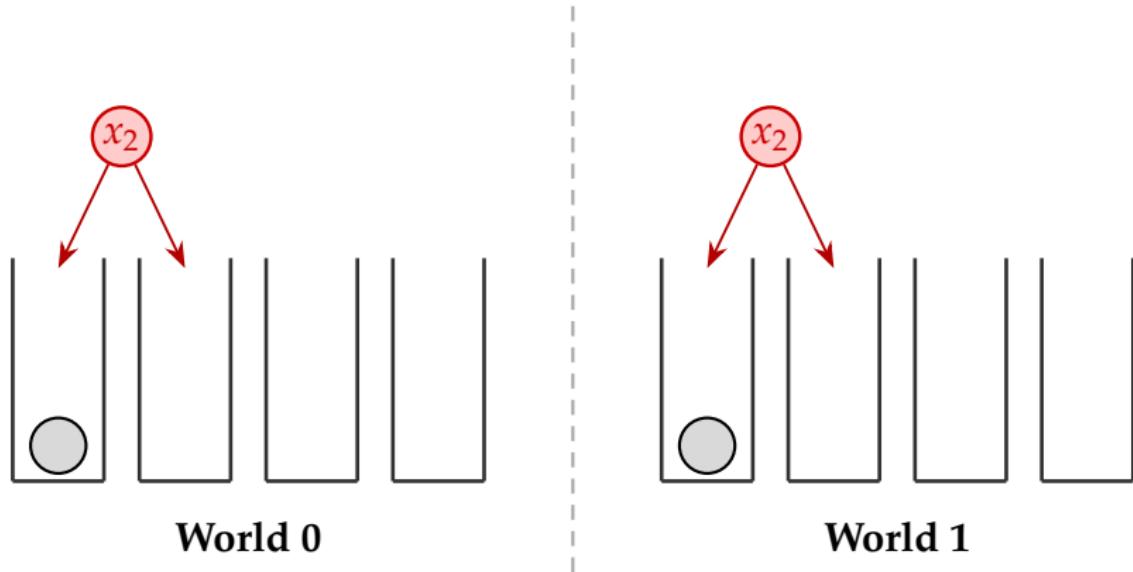
- ▶ Two identical worlds, same insertion sequence
- ▶ World 1 will later receive one extra ball  $x^*$

## ANALYZING THE RE COURSE



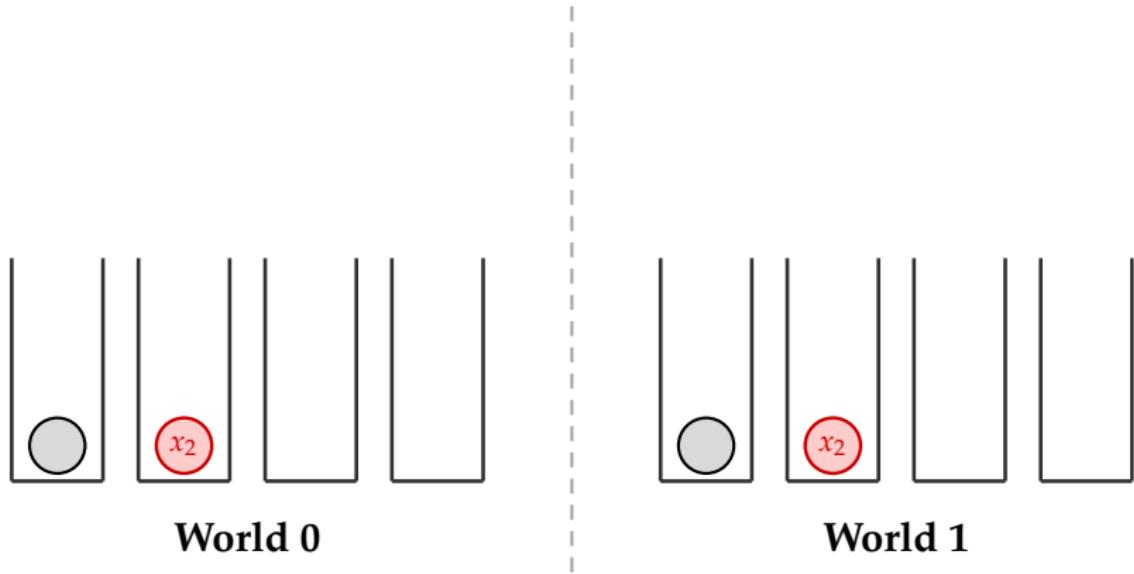
- ▶ Two identical worlds, same insertion sequence
- ▶ World 1 will later receive one extra ball  $x^*$

## ANALYZING THE RE COURSE



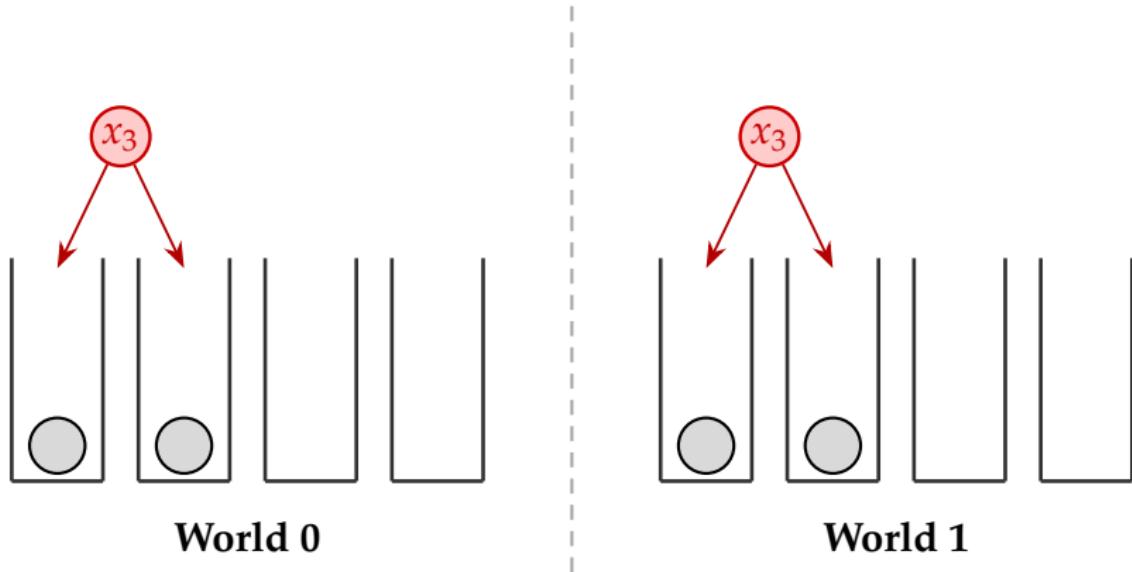
- ▶ Two identical worlds, same insertion sequence
- ▶ World 1 will later receive one extra ball  $x^*$

## ANALYZING THE RE COURSE



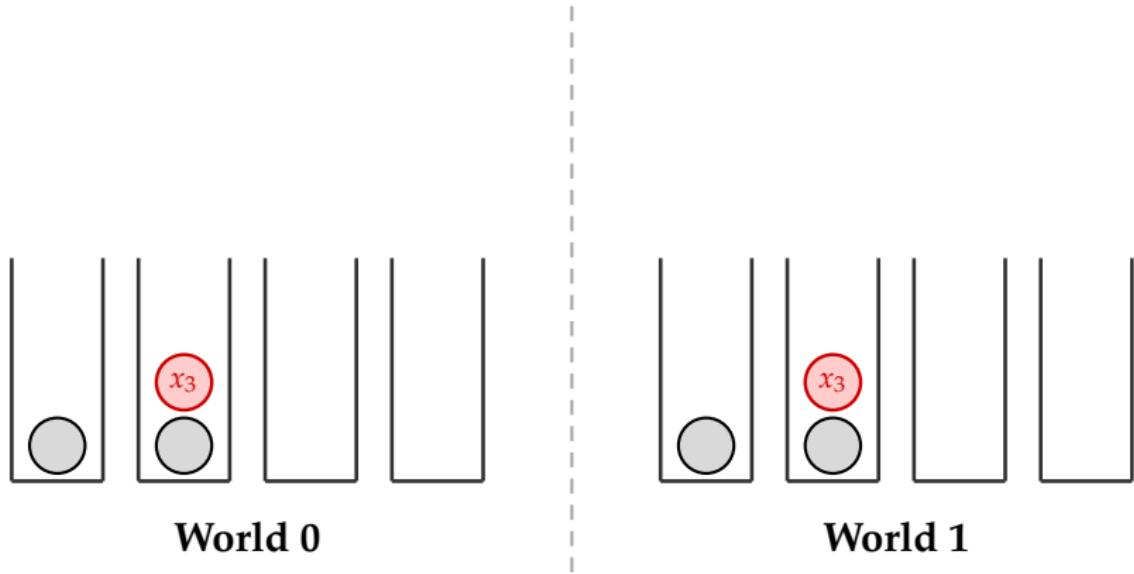
- ▶ Two identical worlds, same insertion sequence
- ▶ World 1 will later receive one extra ball  $x^*$

## ANALYZING THE RE COURSE



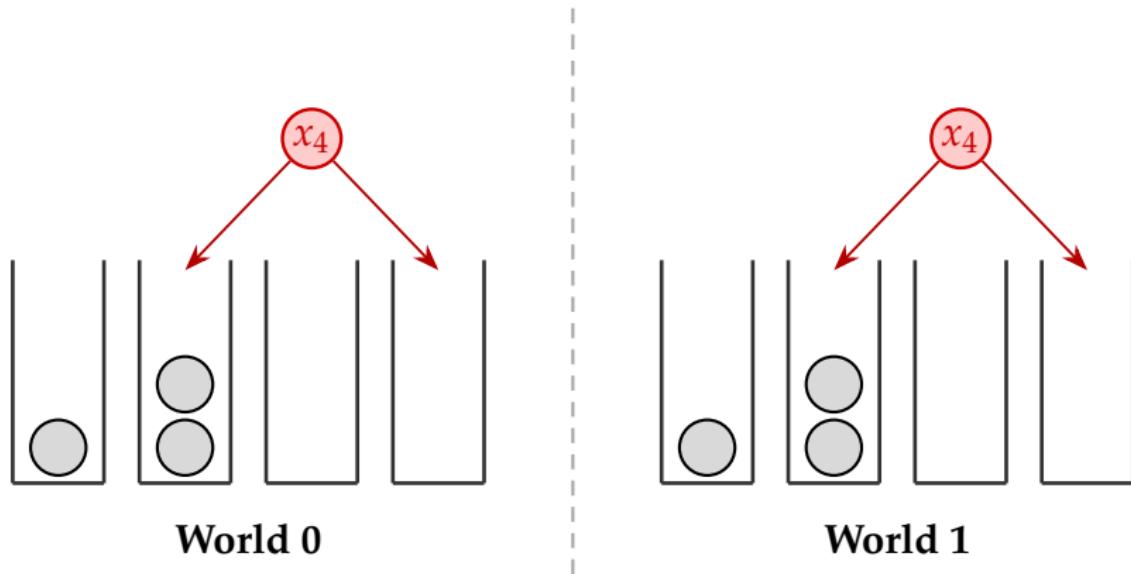
- ▶ Two identical worlds, same insertion sequence
- ▶ World 1 will later receive one extra ball  $x^*$

## ANALYZING THE RE COURSE



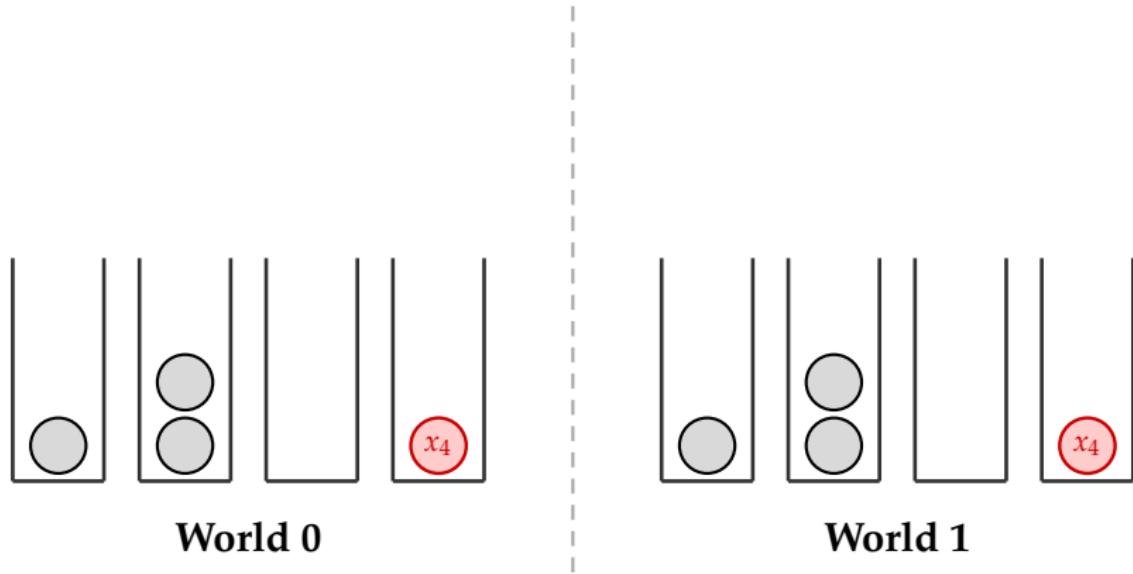
- ▶ Two identical worlds, same insertion sequence
- ▶ World 1 will later receive one extra ball  $x^*$

## ANALYZING THE RE COURSE



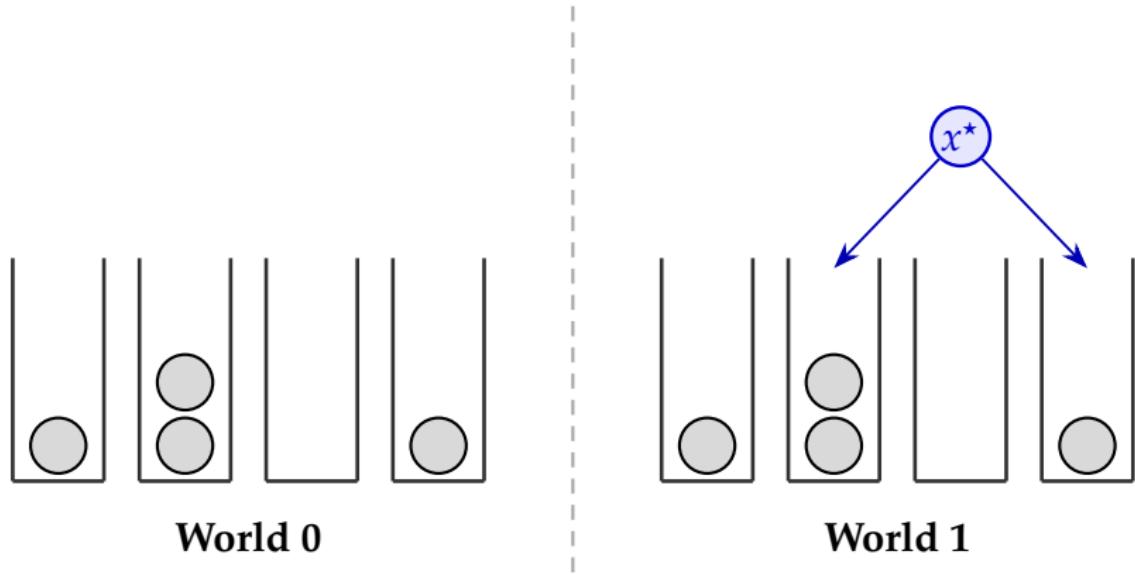
- ▶ Two identical worlds, same insertion sequence
- ▶ World 1 will later receive one extra ball  $x^*$

## ANALYZING THE RE COURSE



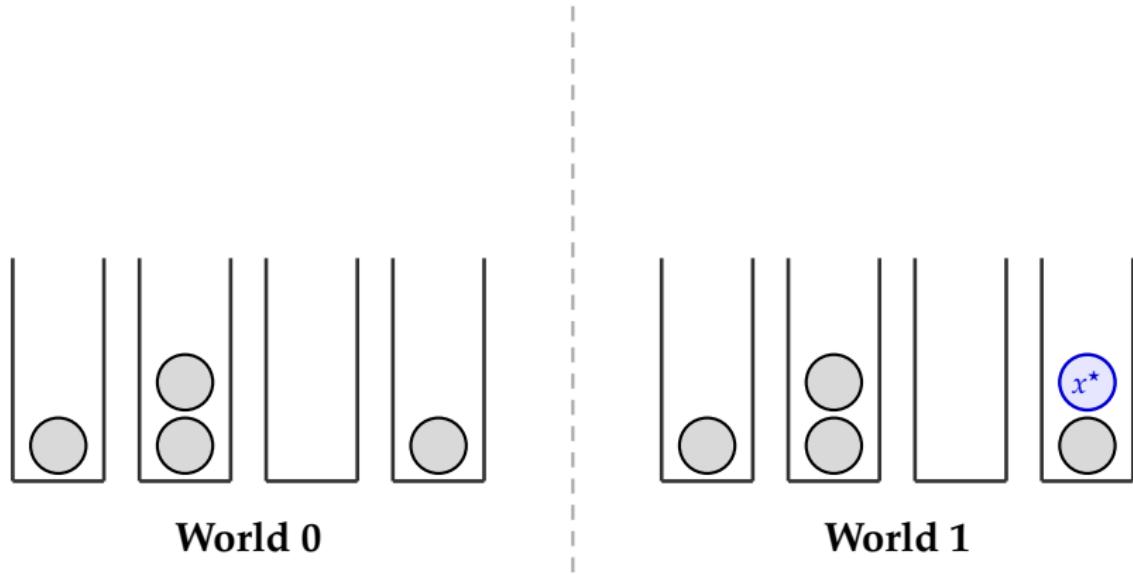
- ▶ Two identical worlds, same insertion sequence
- ▶ World 1 will later receive one extra ball  $x^*$

## ANALYZING THE RE COURSE



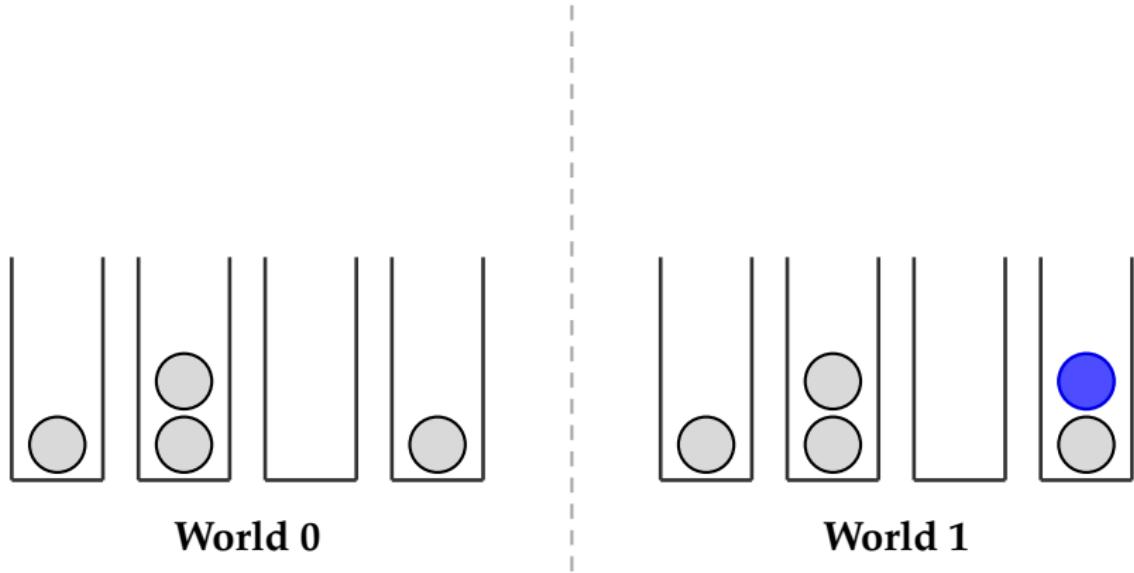
- $x^*$  arrives only in World 1

## ANALYZING THE RE COURSE



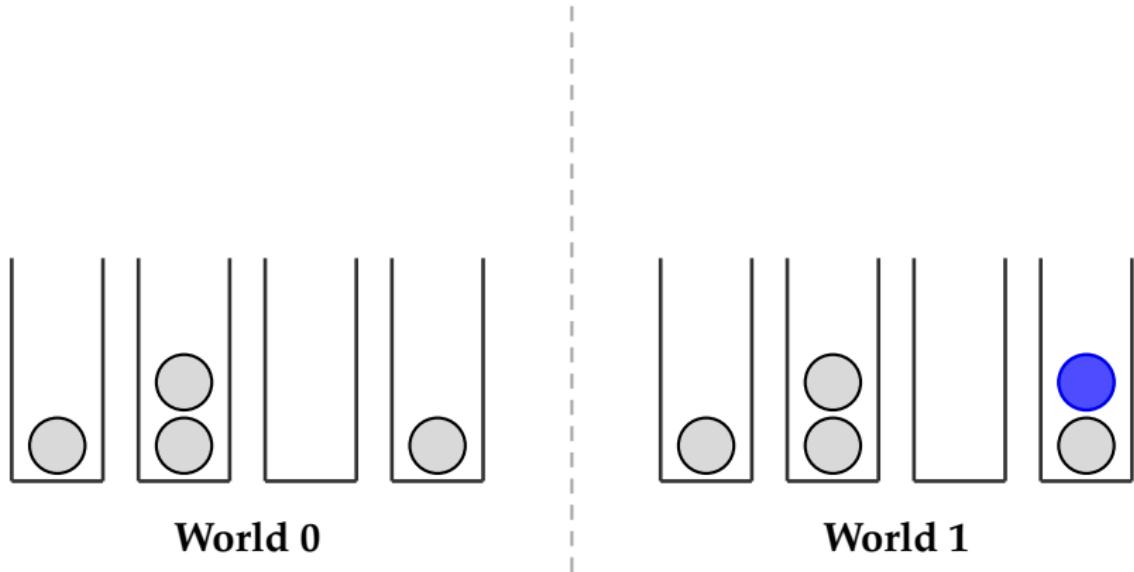
- $x^*$  arrives only in World 1

## ANALYZING THE RE COURSE



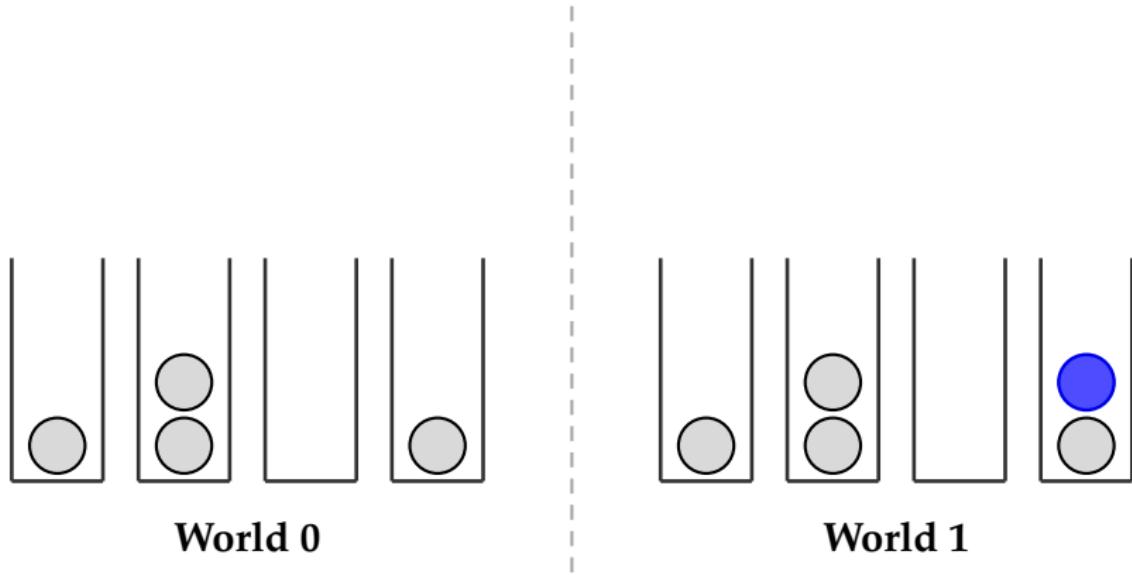
**Question:** How do subsequent insertions differ between the two worlds?

## ANALYZING THE RE COURSE



Future insertions will experience either:

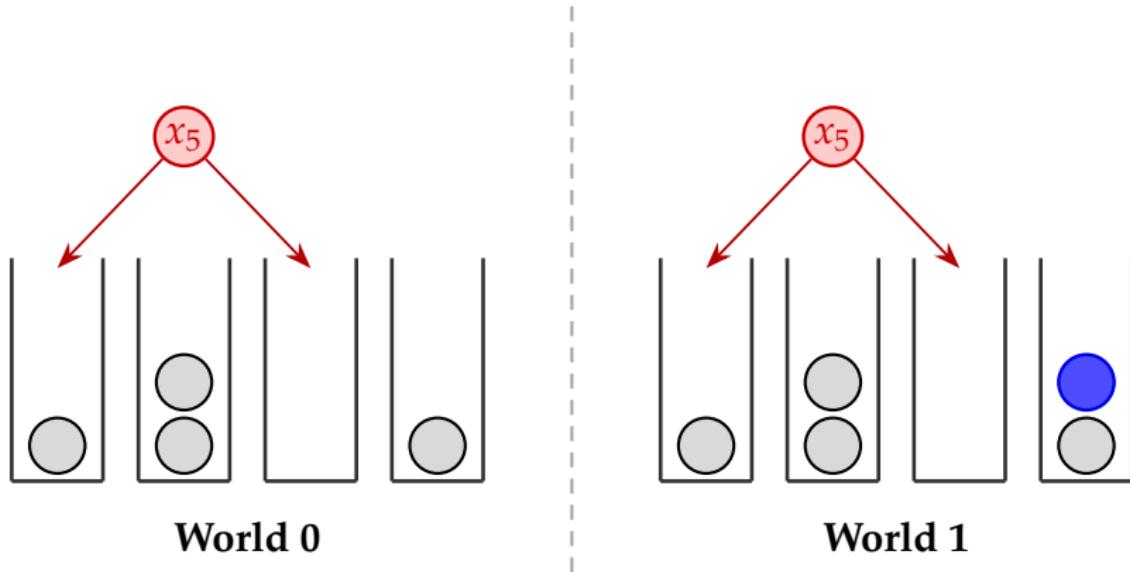
## ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse

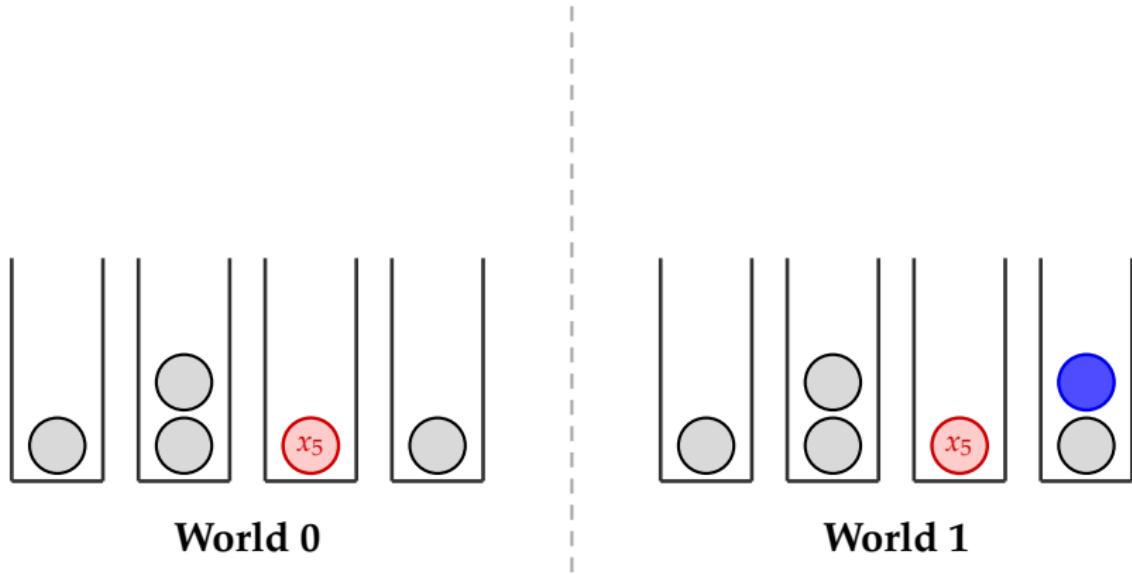
## ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse

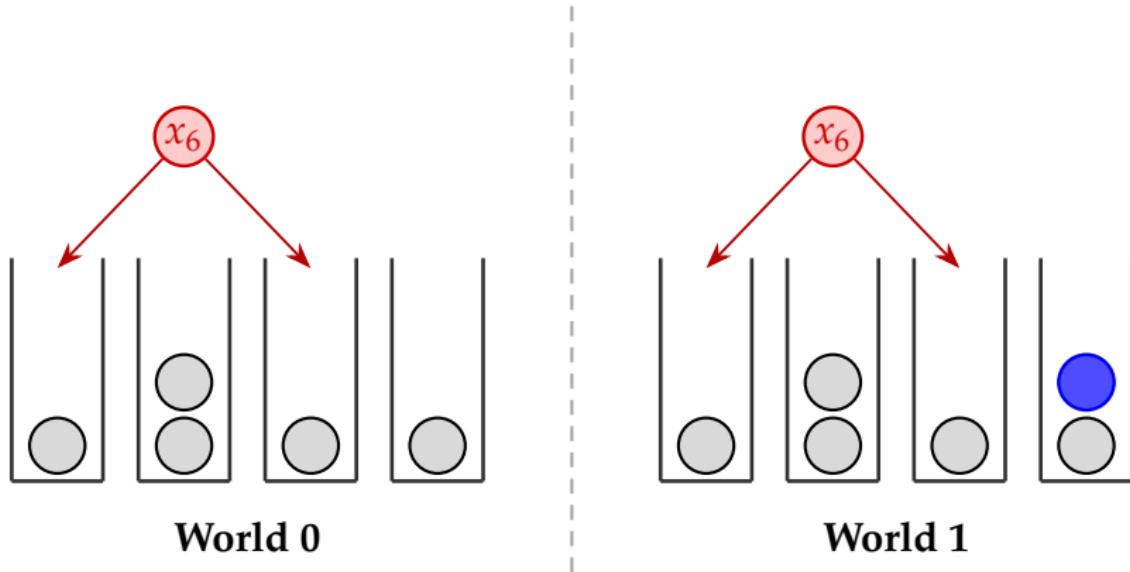
# ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse

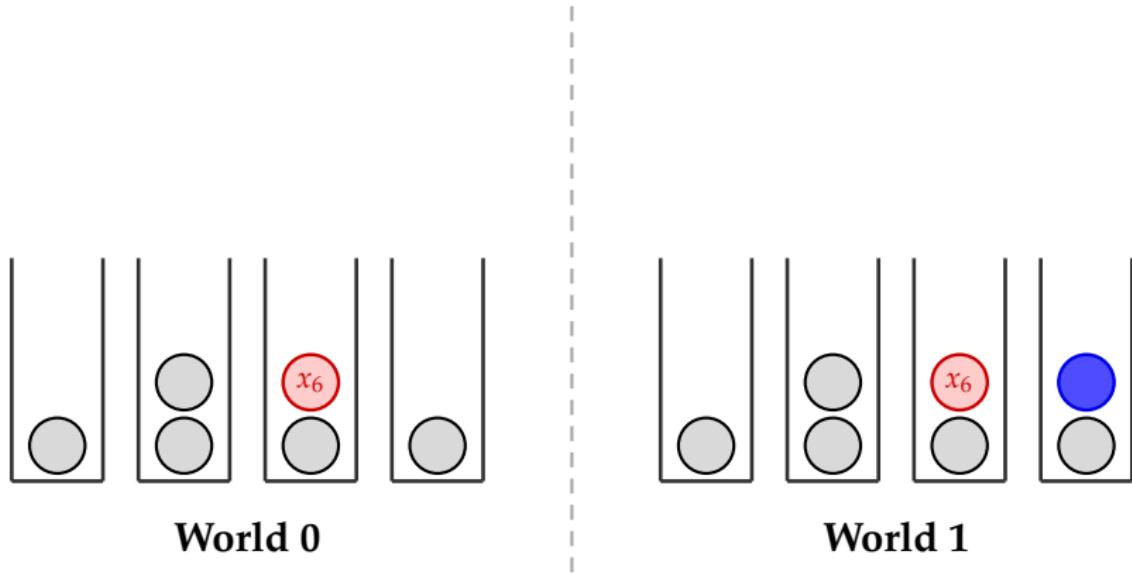
## ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse

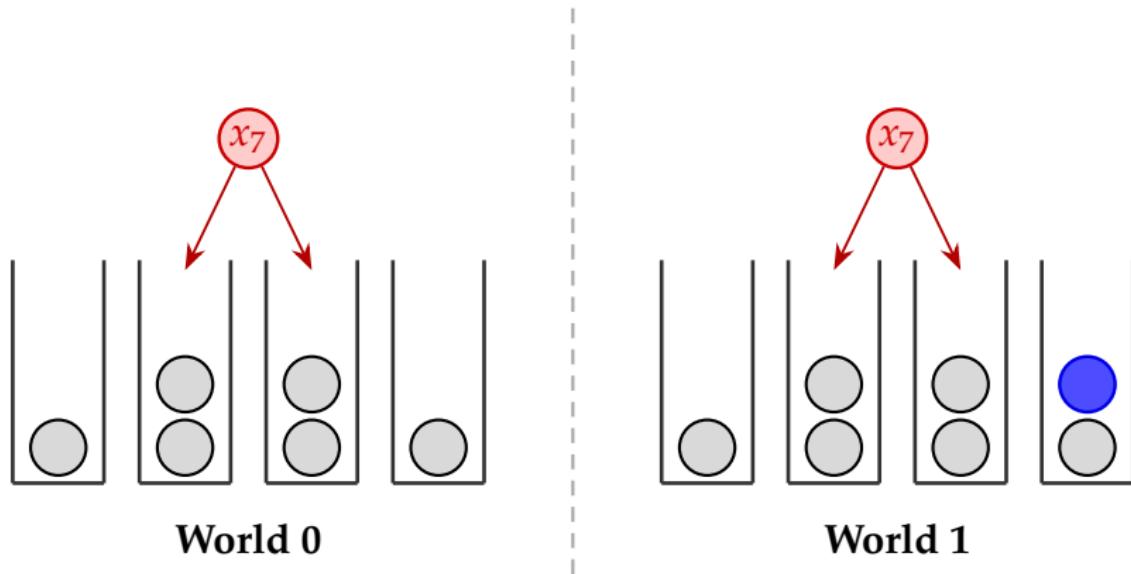
# ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse

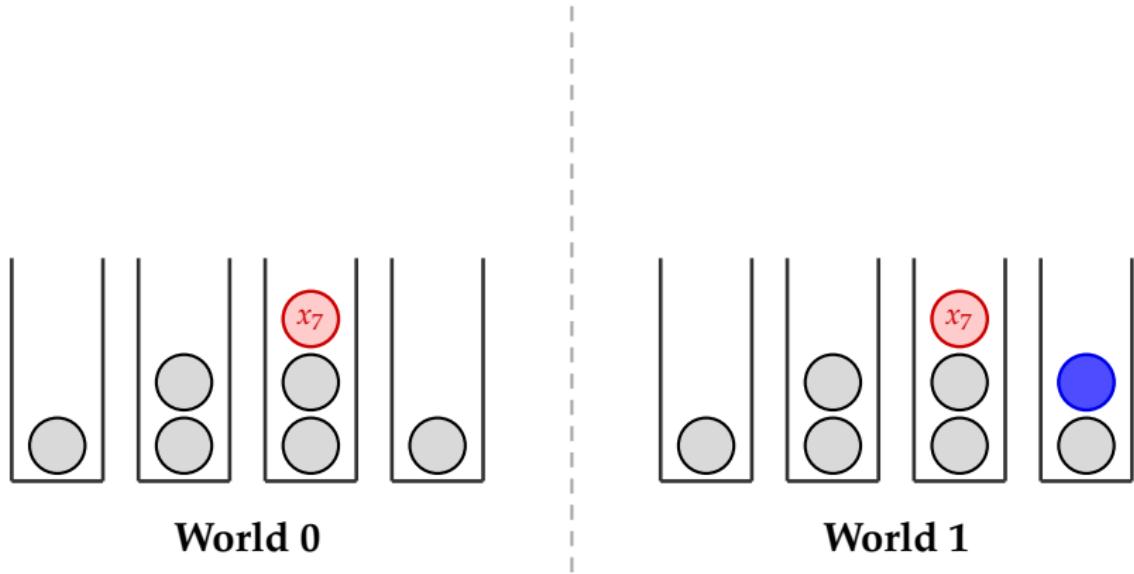
## ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse

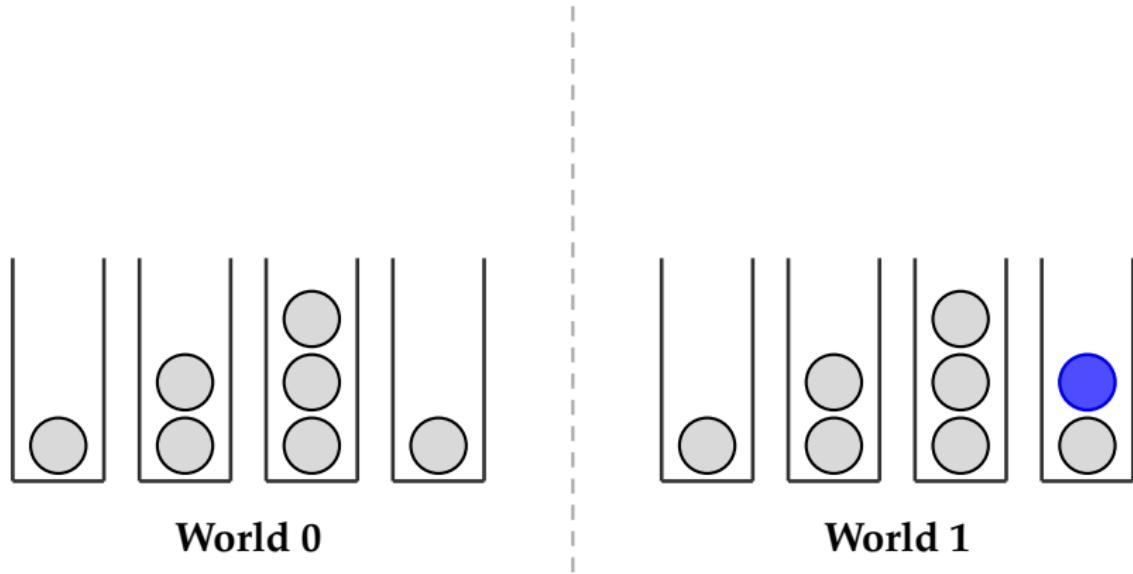
# ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse

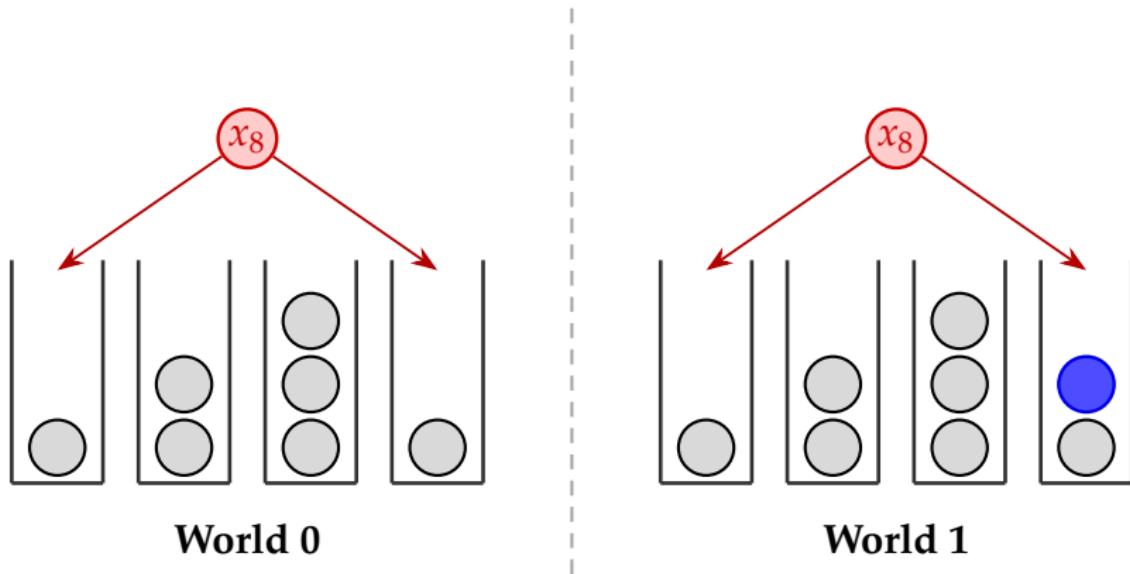
# ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse
2. Recourse

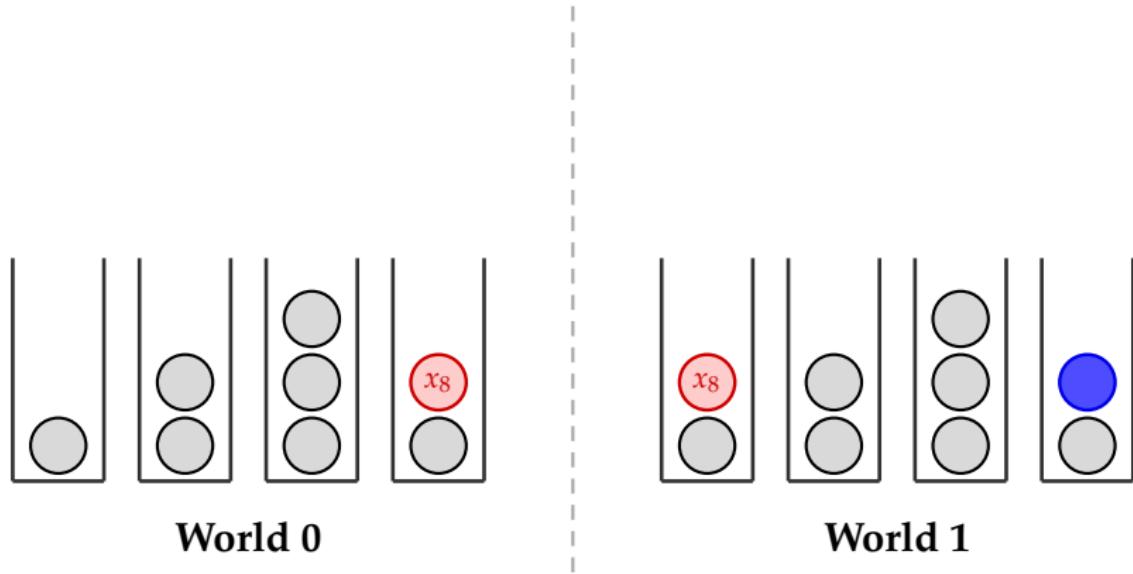
## ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse
2. Recourse

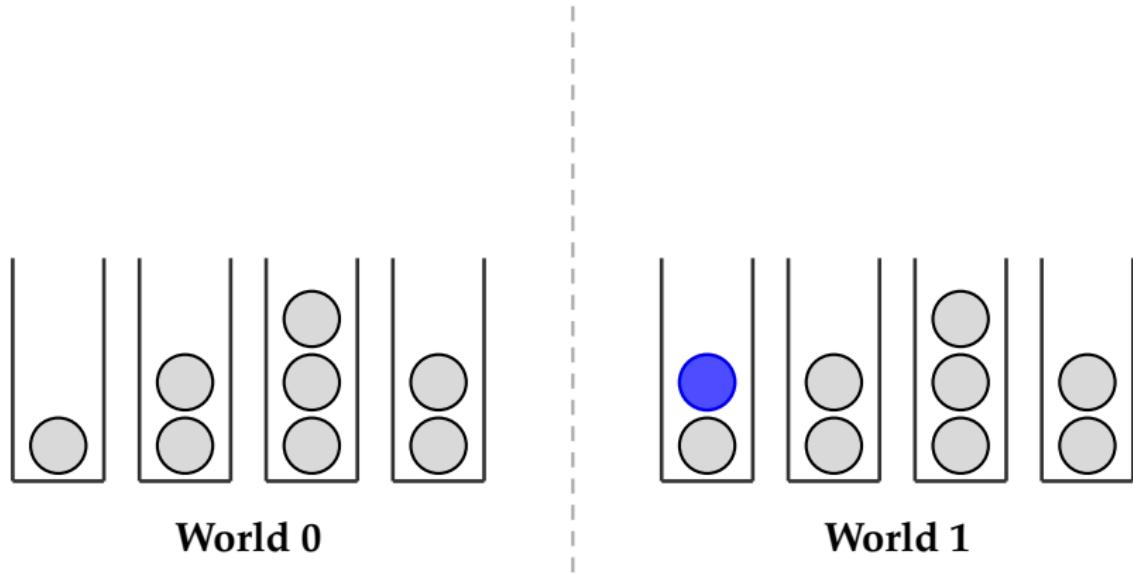
# ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse
2. Recourse

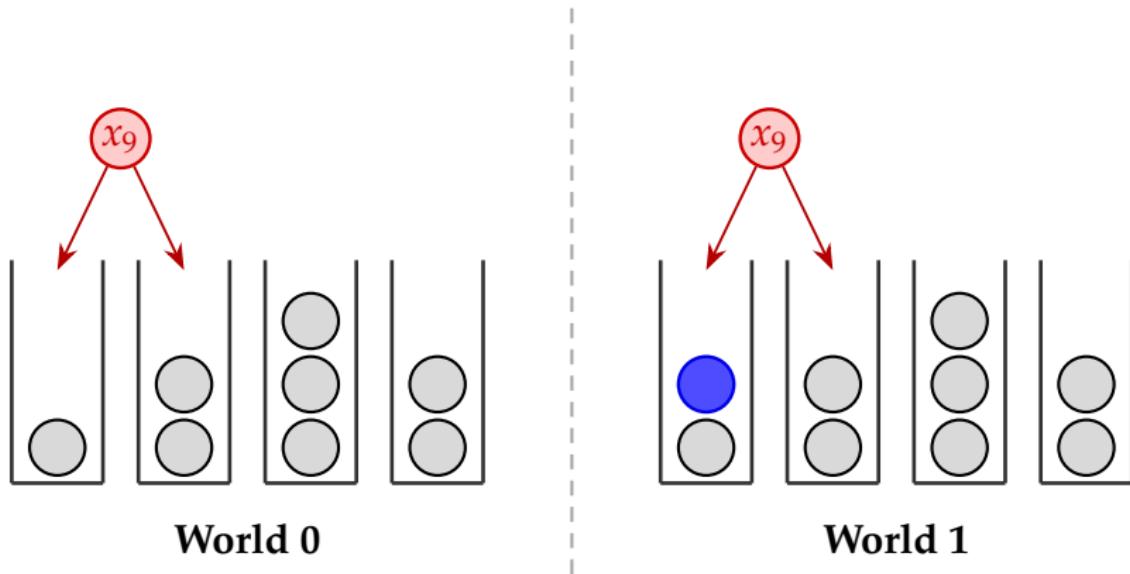
# ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse
2. Recourse

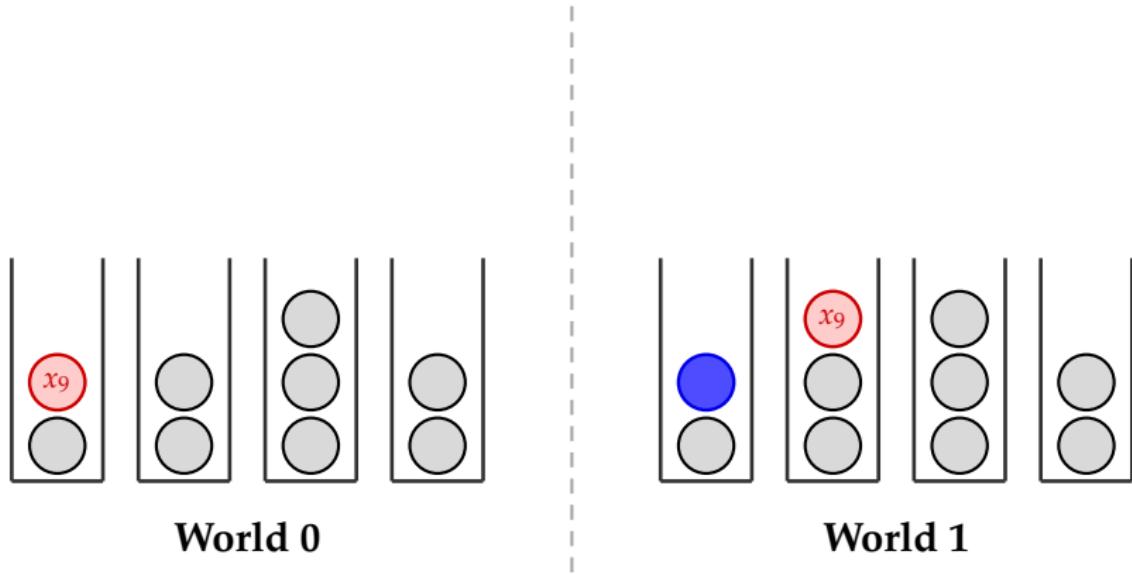
## ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse
2. Recourse

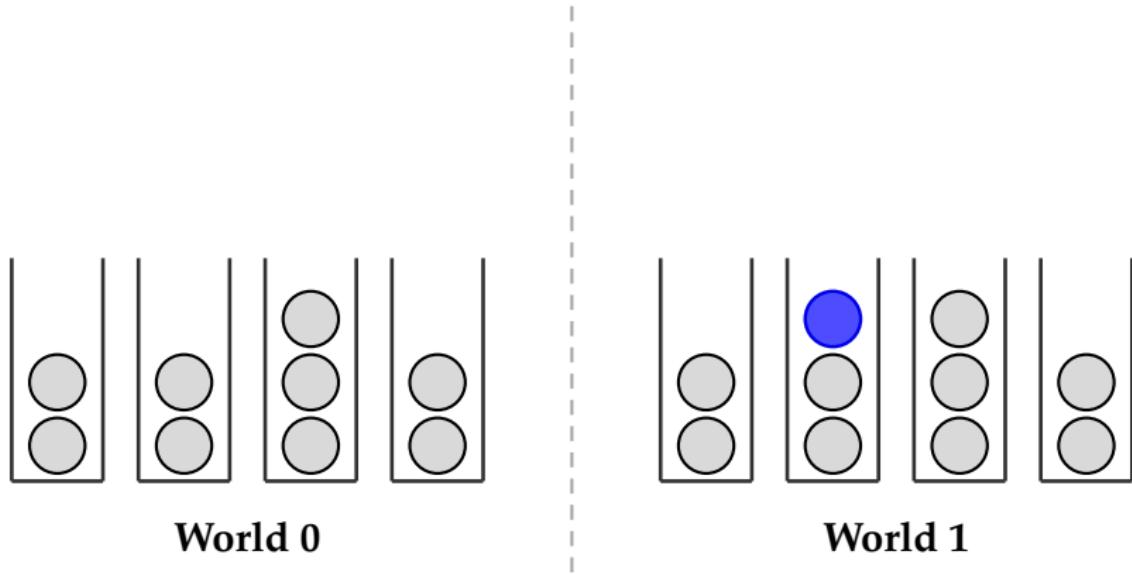
# ANALYZING THE RE COURSE



Future insertions will experience either:

1. No recourse
2. Recourse

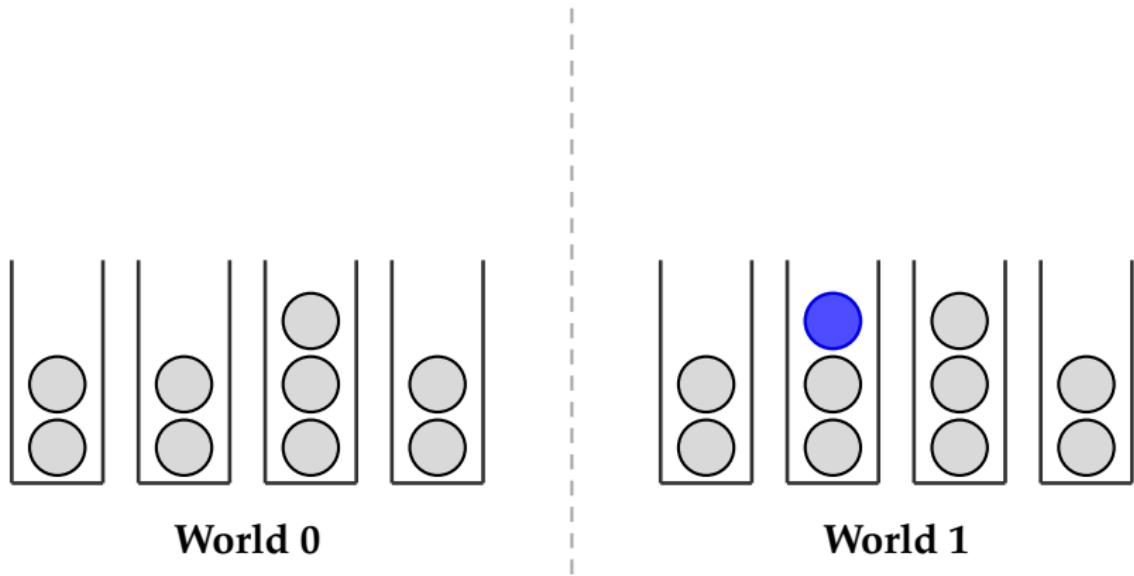
## ANALYZING THE RE COURSE



Future insertions will experience either:

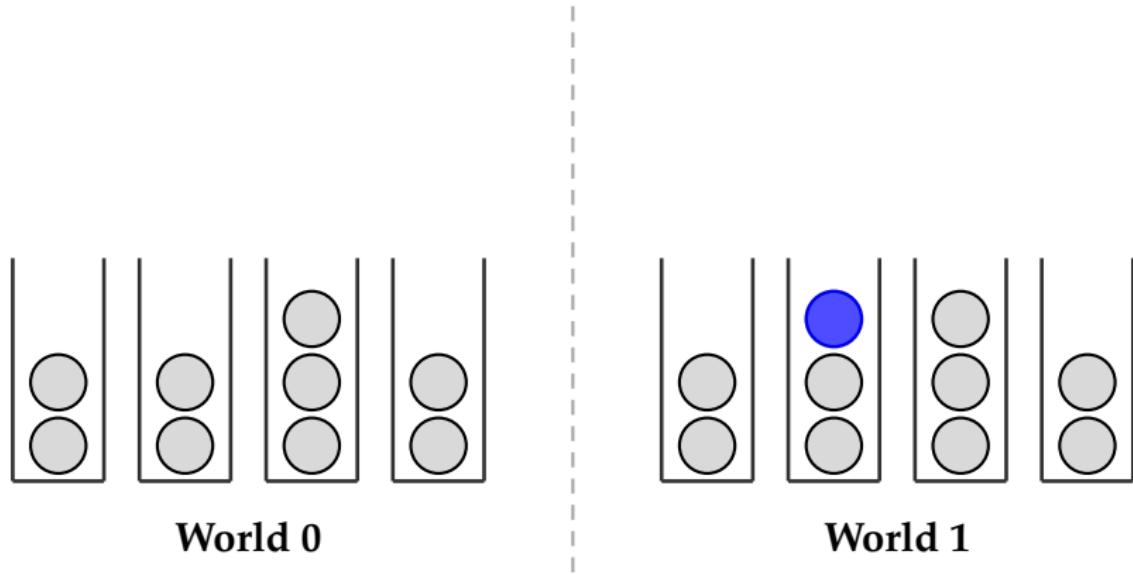
1. No recourse
2. Recourse

# ANALYZING THE RE COURSE



Two key observations:

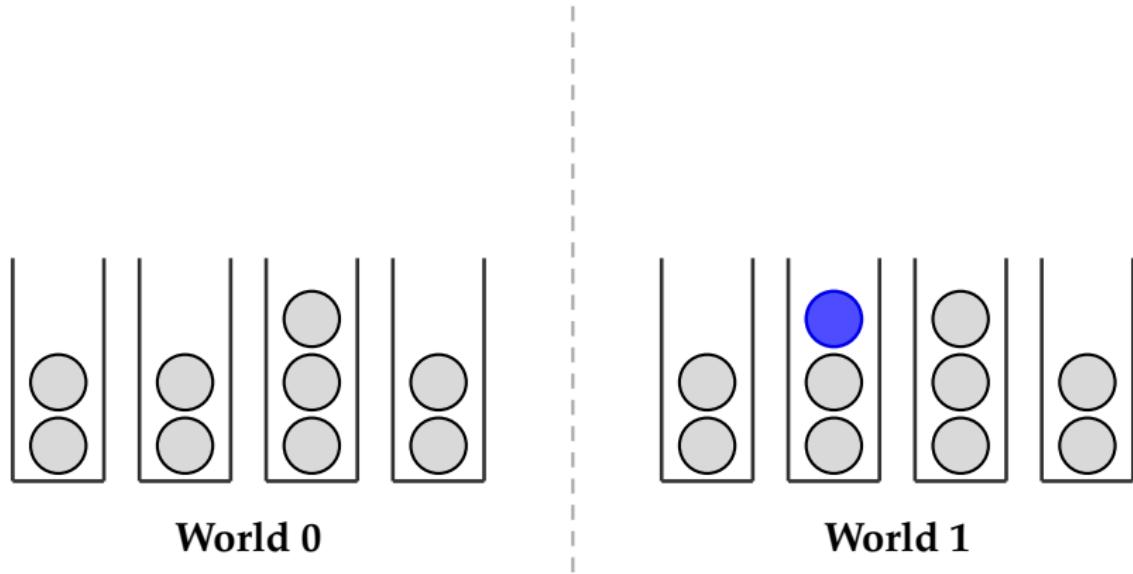
## ANALYZING THE RE COURSE



Two key observations:

1. There's always one special bin with an extra ball

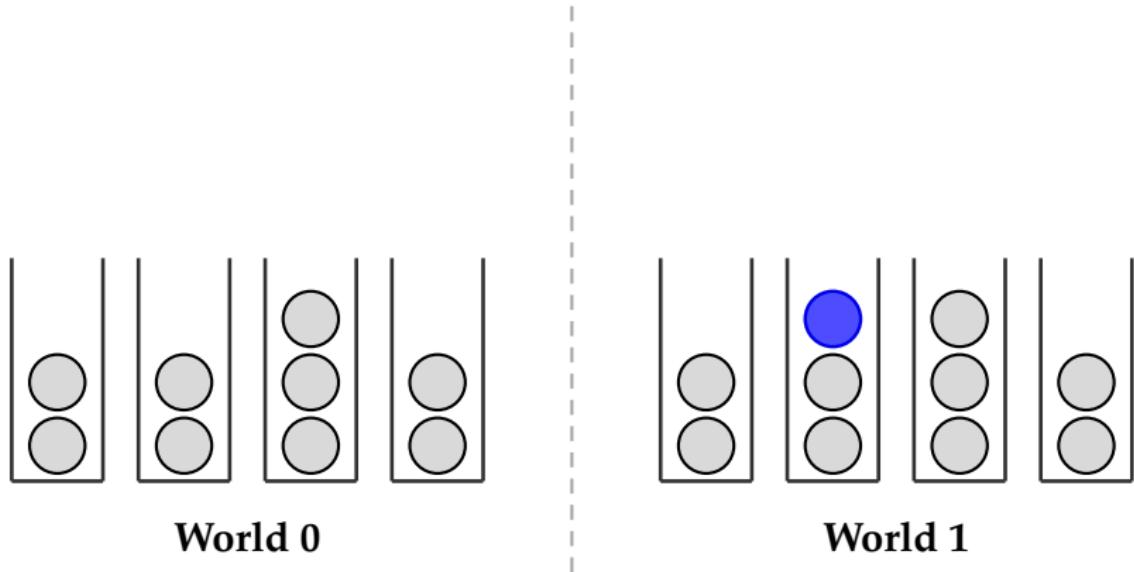
# ANALYZING THE RE COURSE



Two key observations:

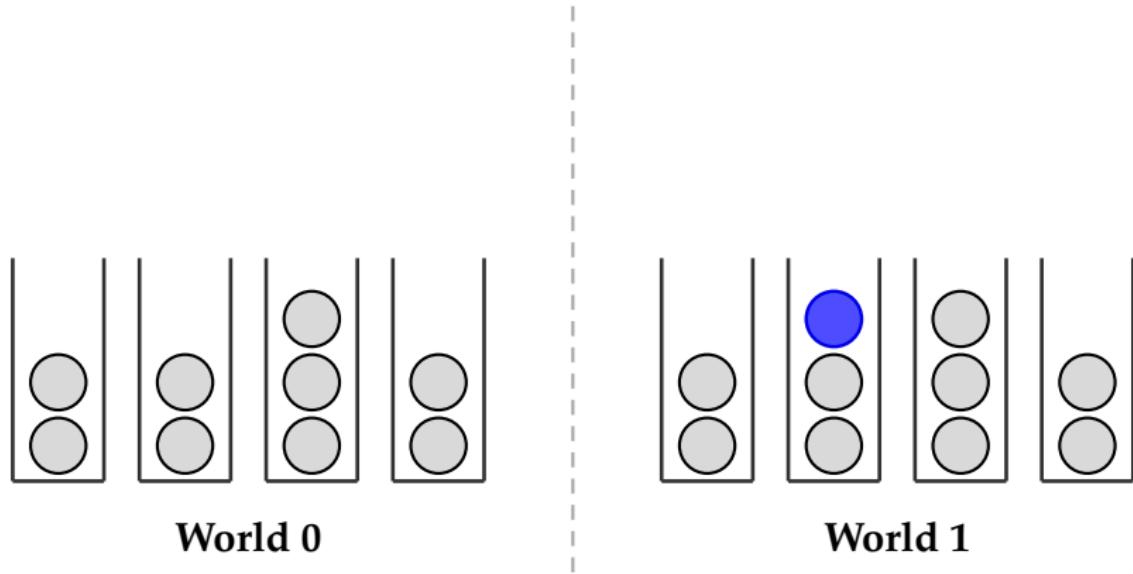
1. There's always one special bin with an extra ball
2. Recourse requires the special bin as one of the two choices

# ANALYZING THE RECOURSE



$$\Pr[\text{ball } x_i \text{ incurs recourse}] = O(1/n)$$

## ANALYZING THE RECOURSE



$$\Pr[\text{ball } x_i \text{ incurs recourse}] = O(1/n)$$

$$\implies \mathbb{E}[\text{total recourse}] = \sum_i \Pr[\text{ball } x_i \text{ incurs recourse}] = O(m/n)$$