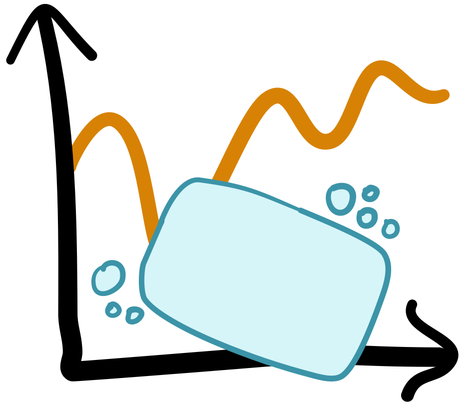
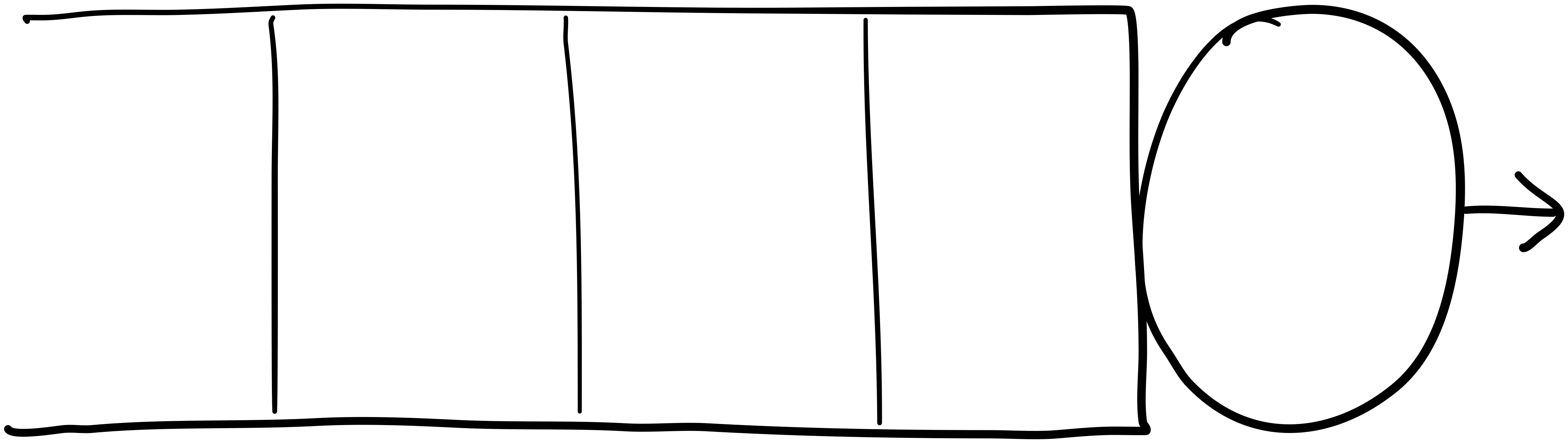
# *The Power of* **SOAP Scheduling**

Mor Harchol-Balter
Ziv Scully

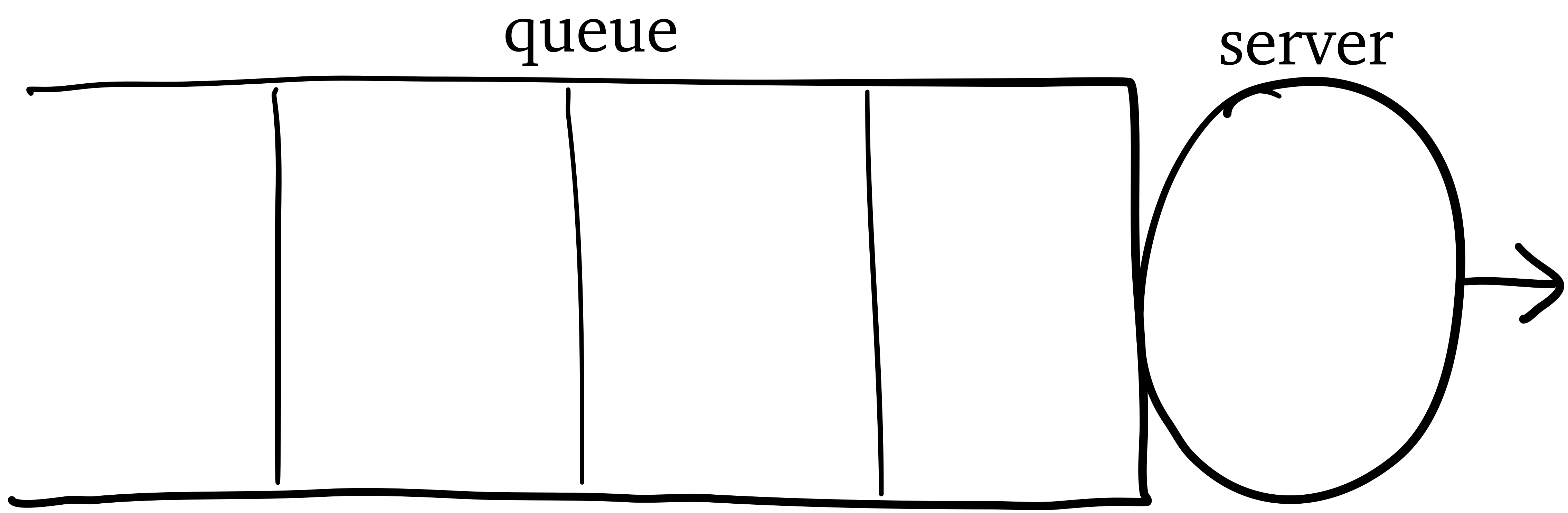
Carnegie Mellon University



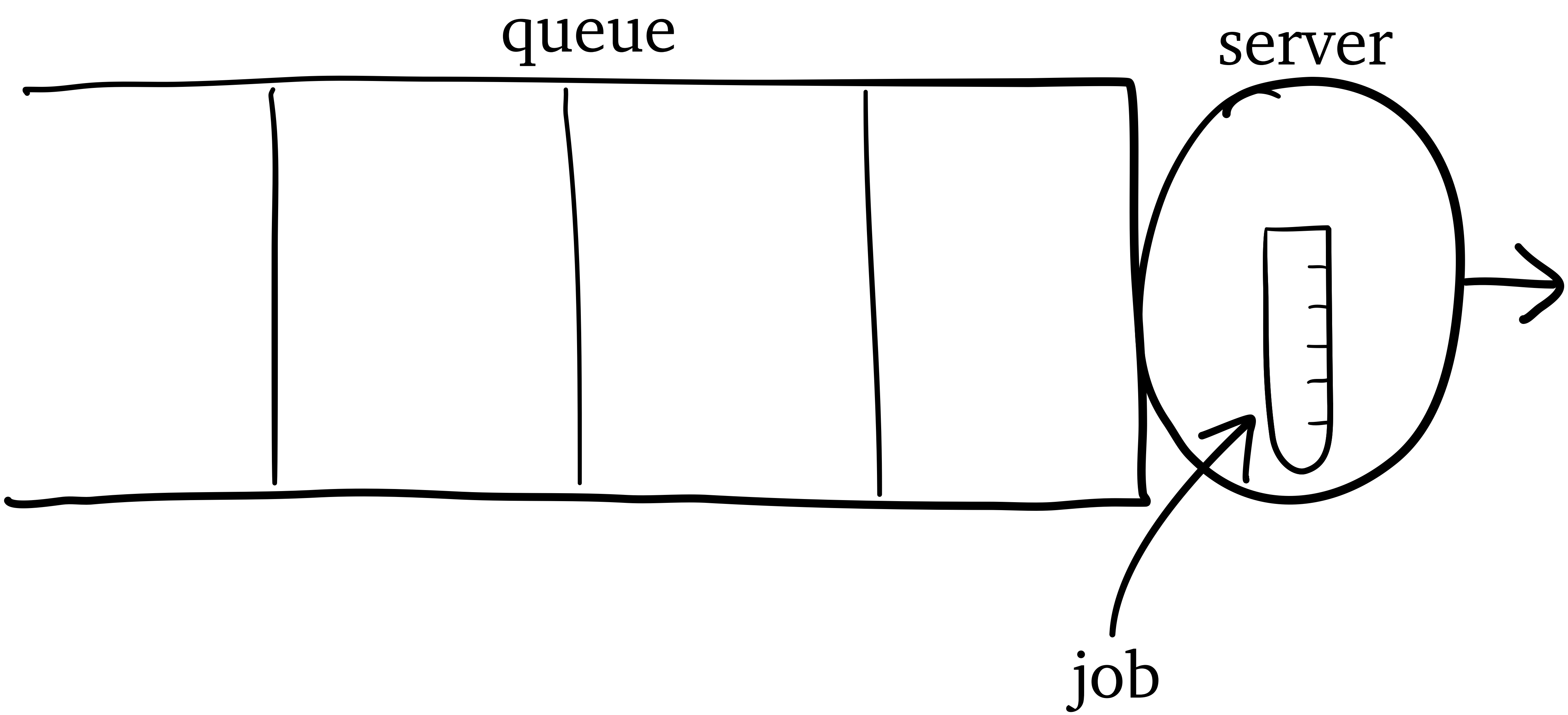
# M/G/1 Queue



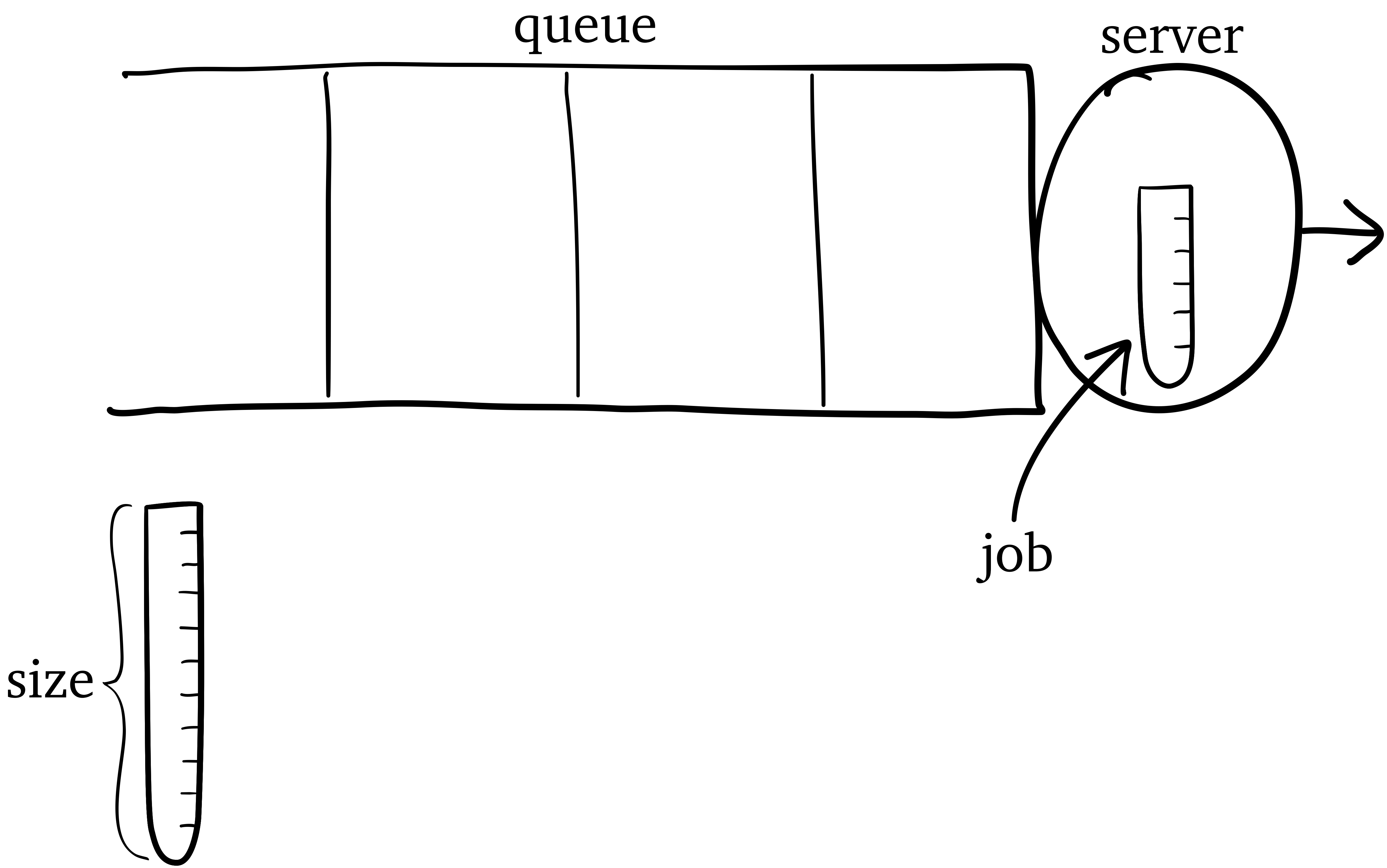
# M/G/1 Queue



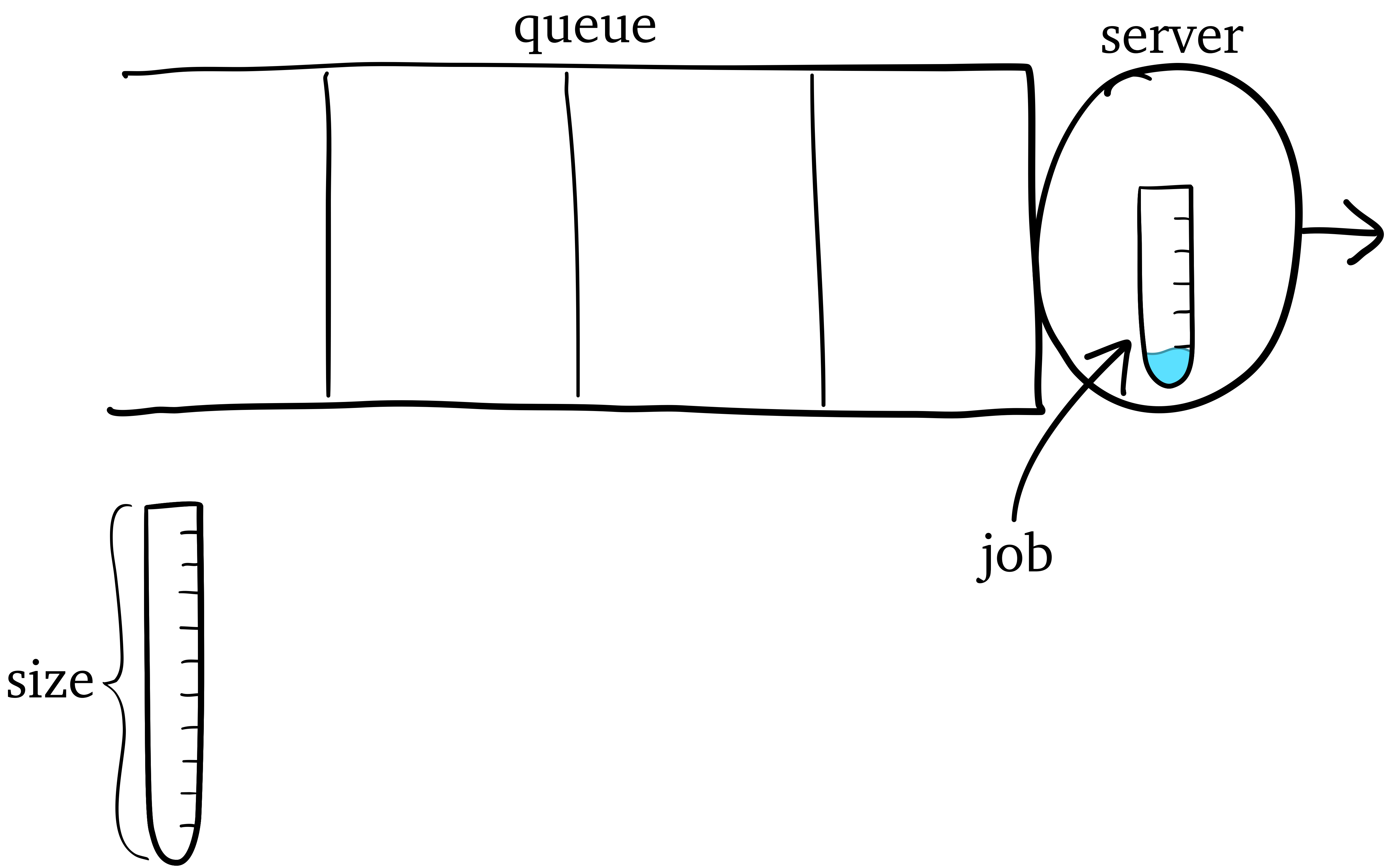
# M/G/1 Queue



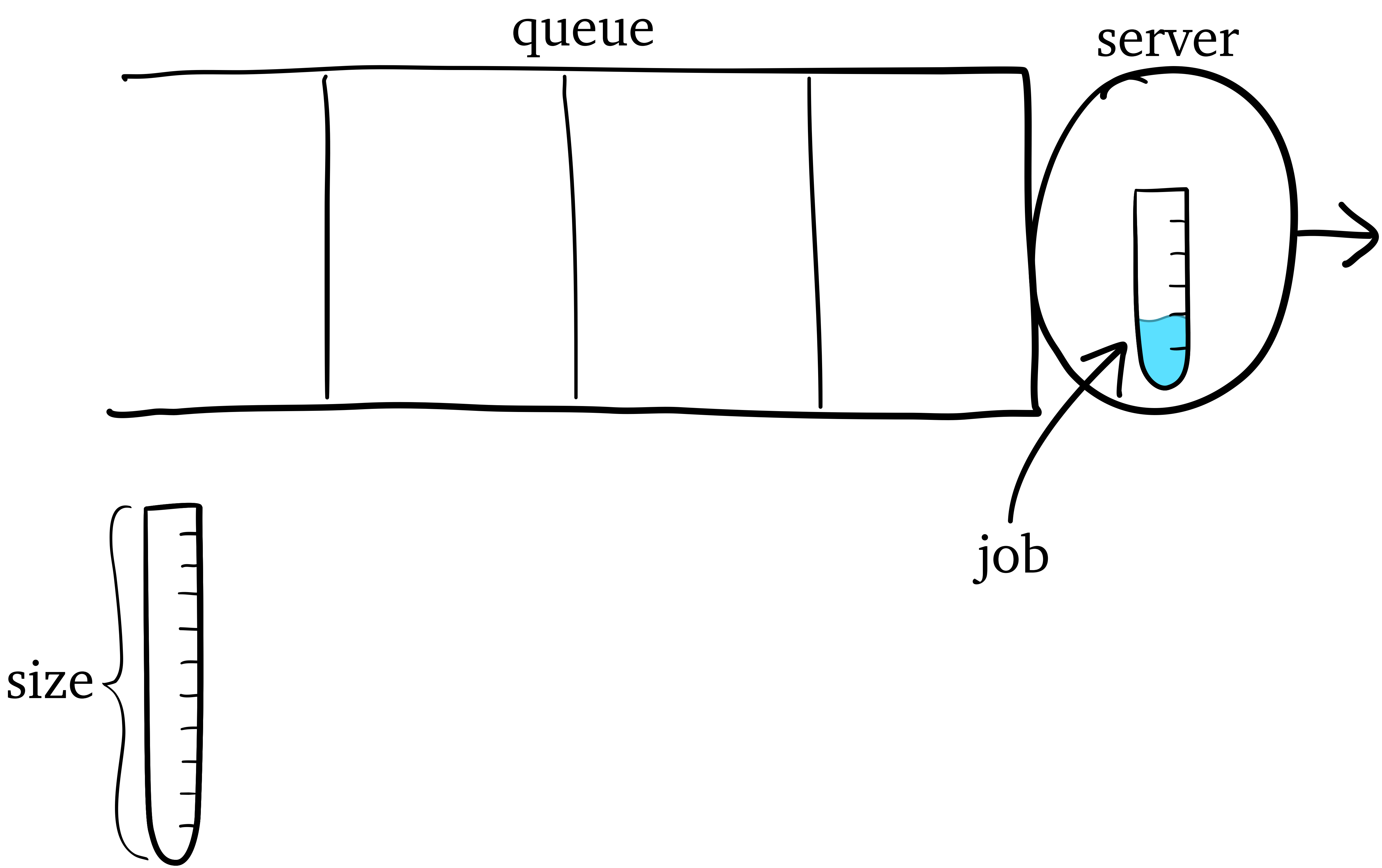
# M/G/1 Queue



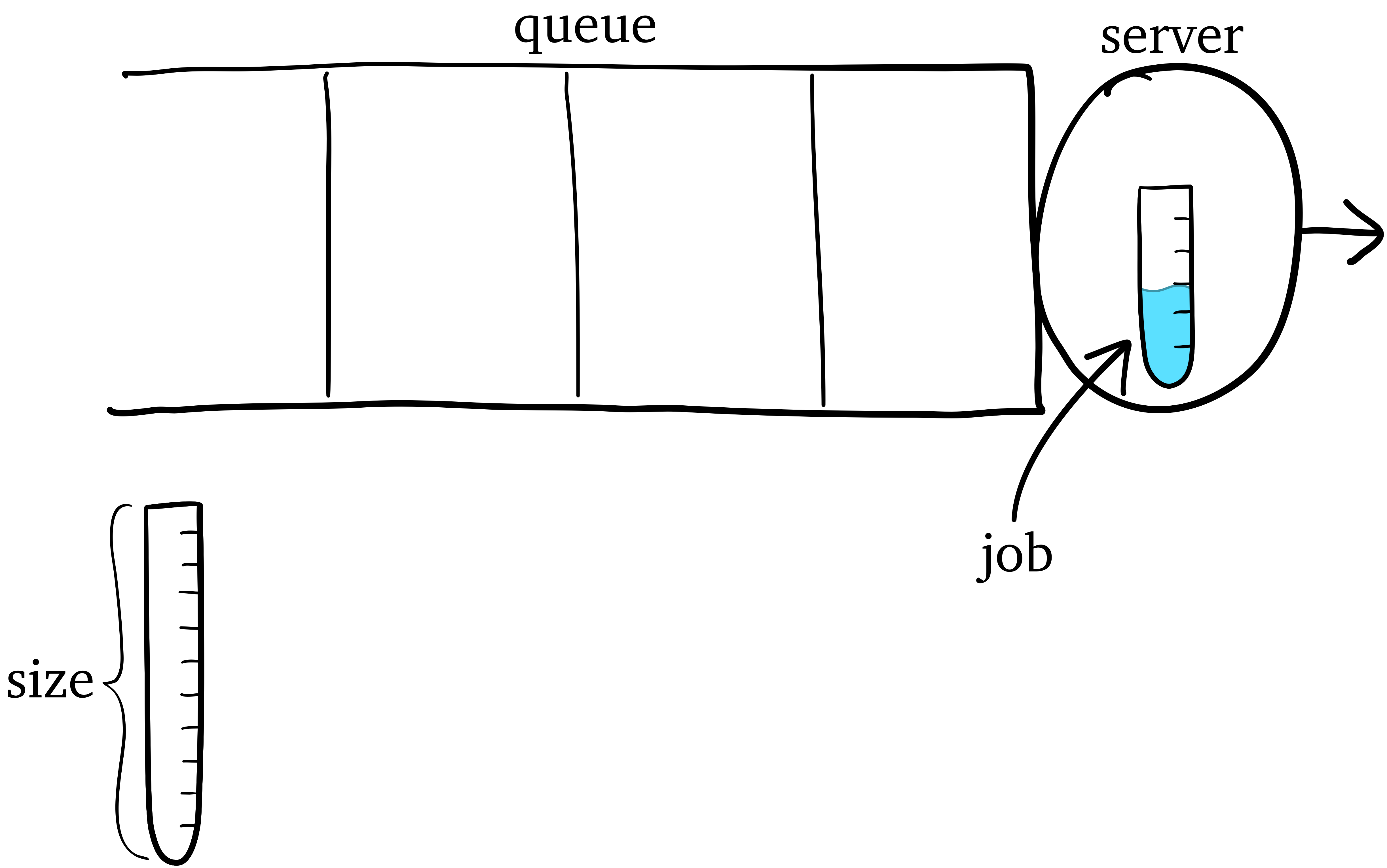
# M/G/1 Queue



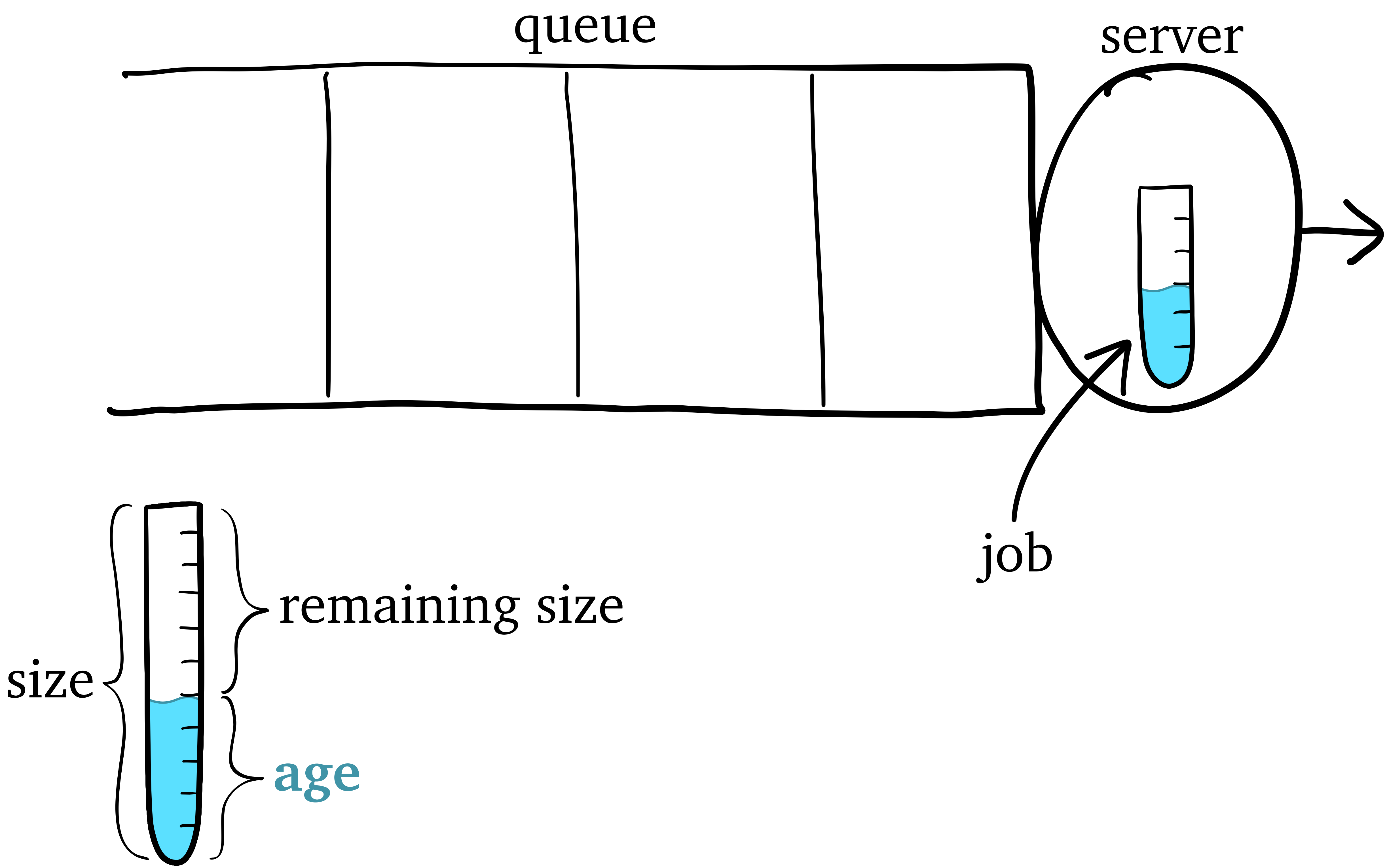
# M/G/1 Queue



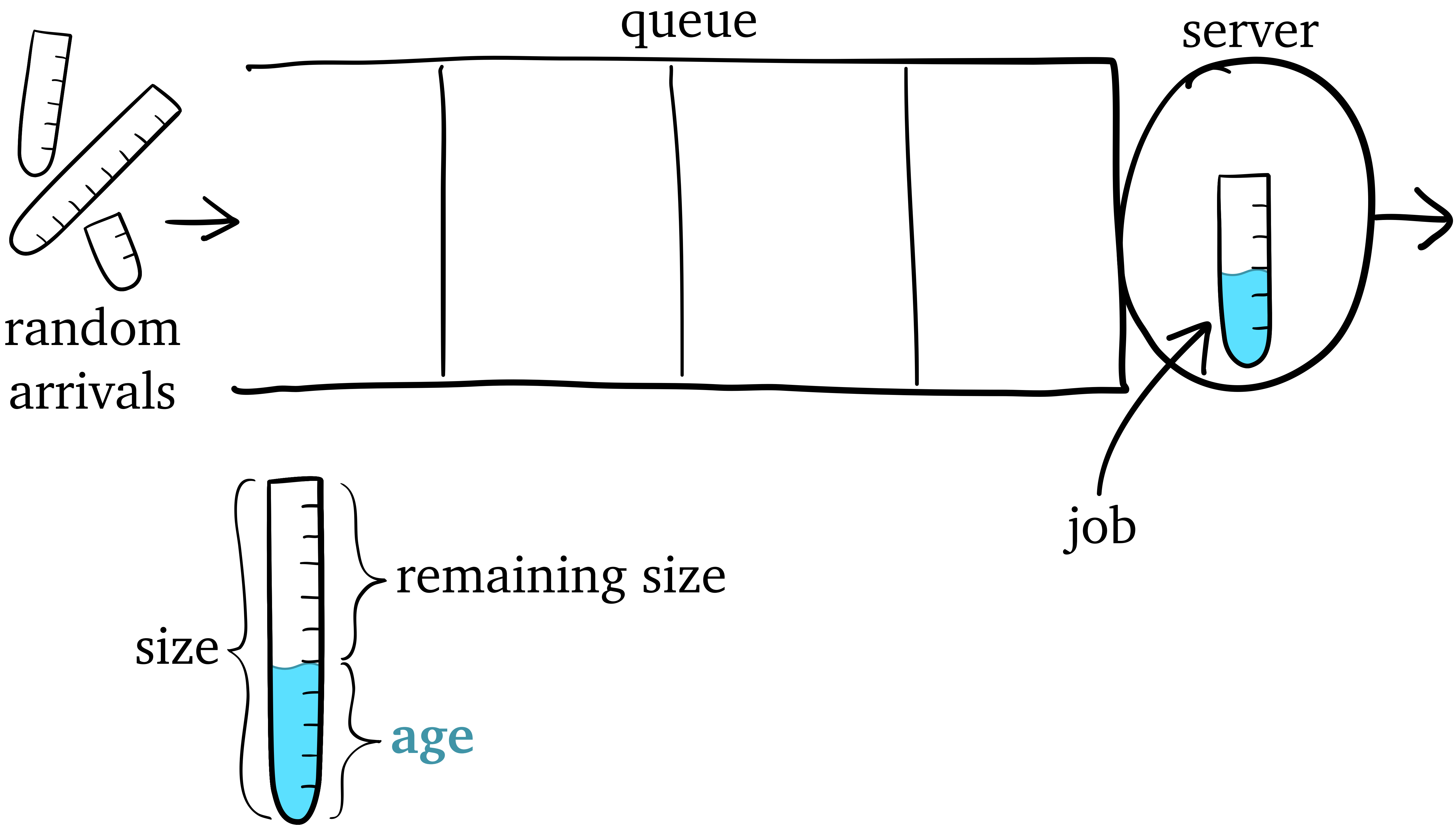
# M/G/1 Queue



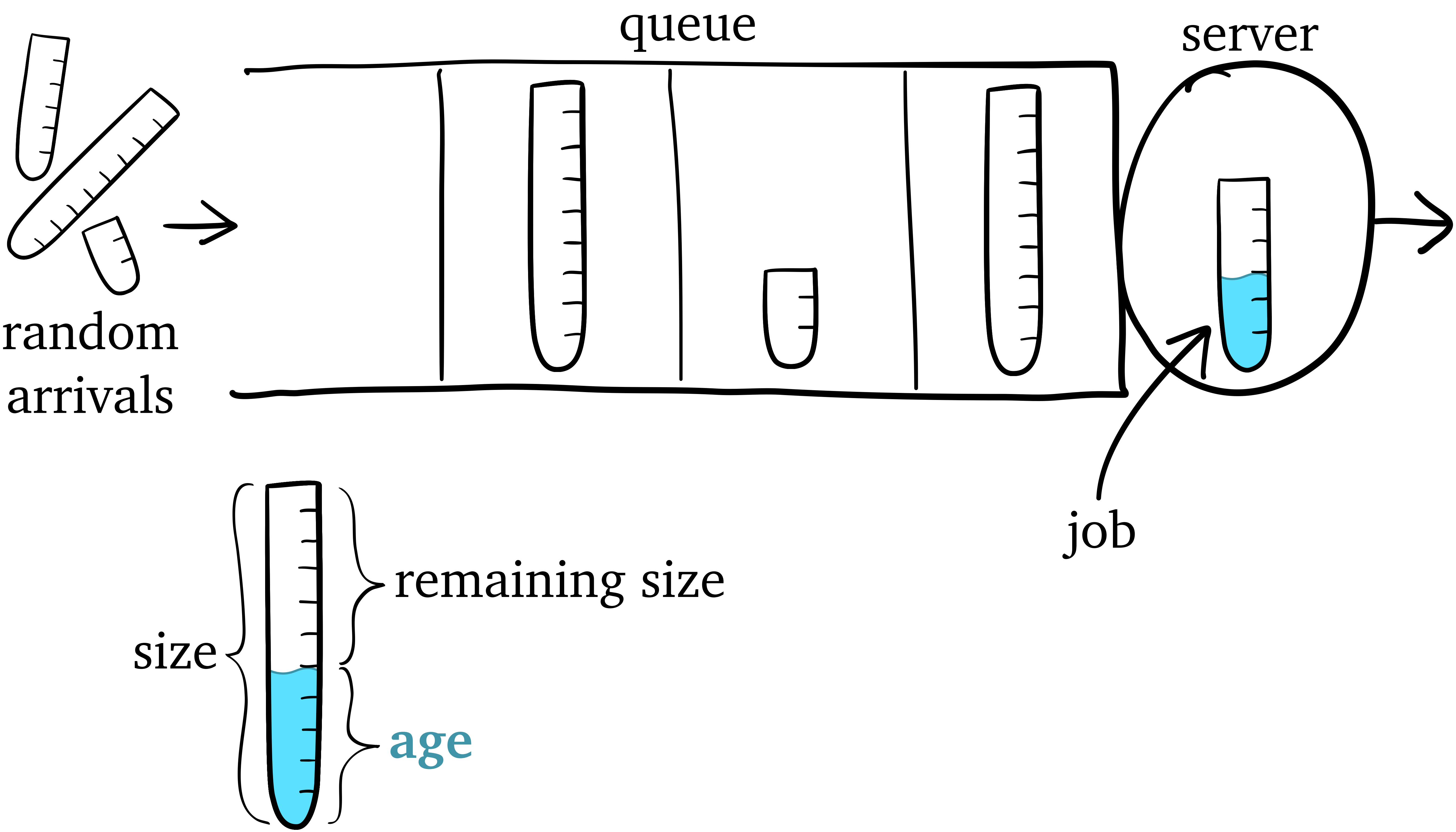
# M/G/1 Queue



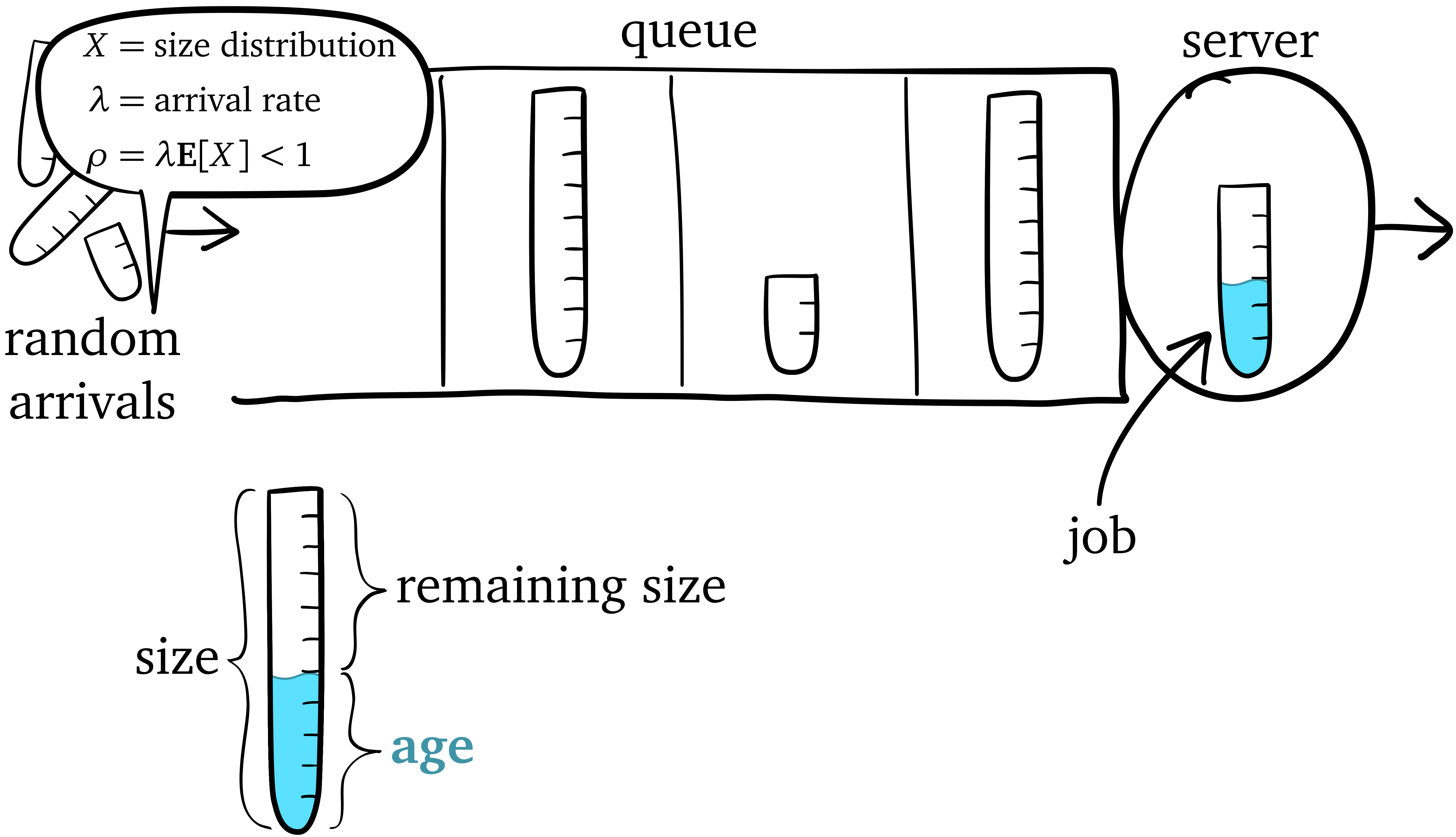
# M/G/1 Queue



# M/G/1 Queue

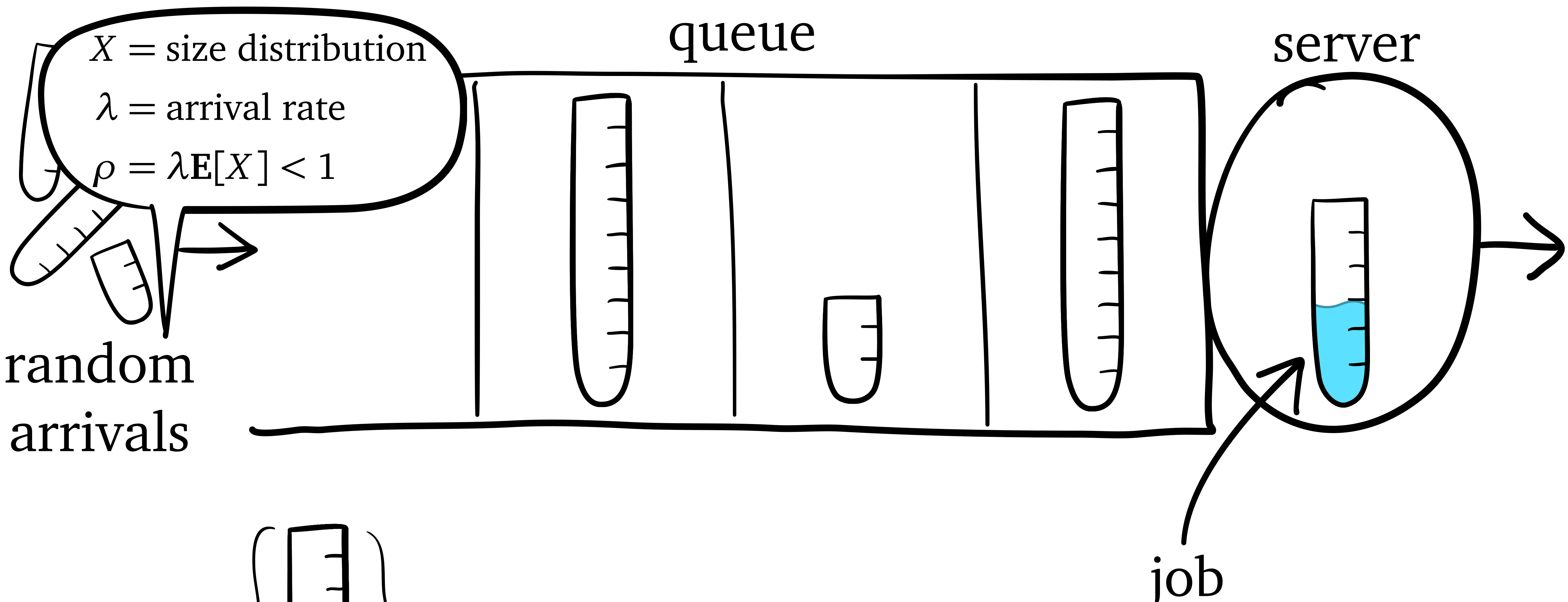


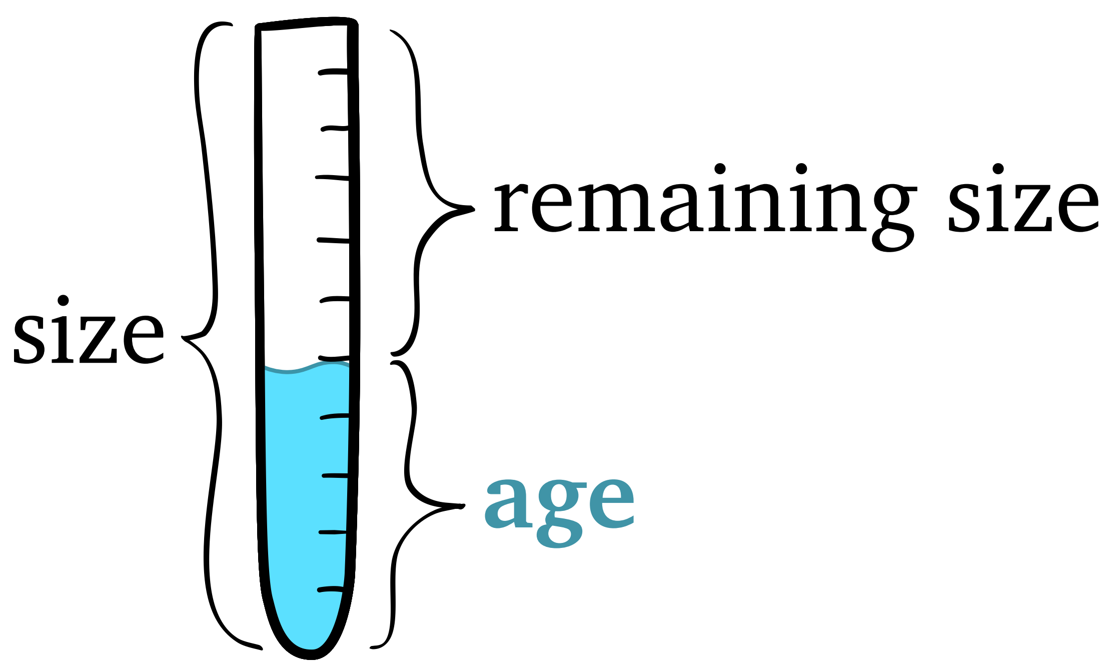
# M/G/1 Queue



$X =$ size distribution
$\lambda =$ arrival rate
$\rho = \lambda \mathbf{E}[X] < 1$
queue
server
random arrivals
job
size
remaining size
age

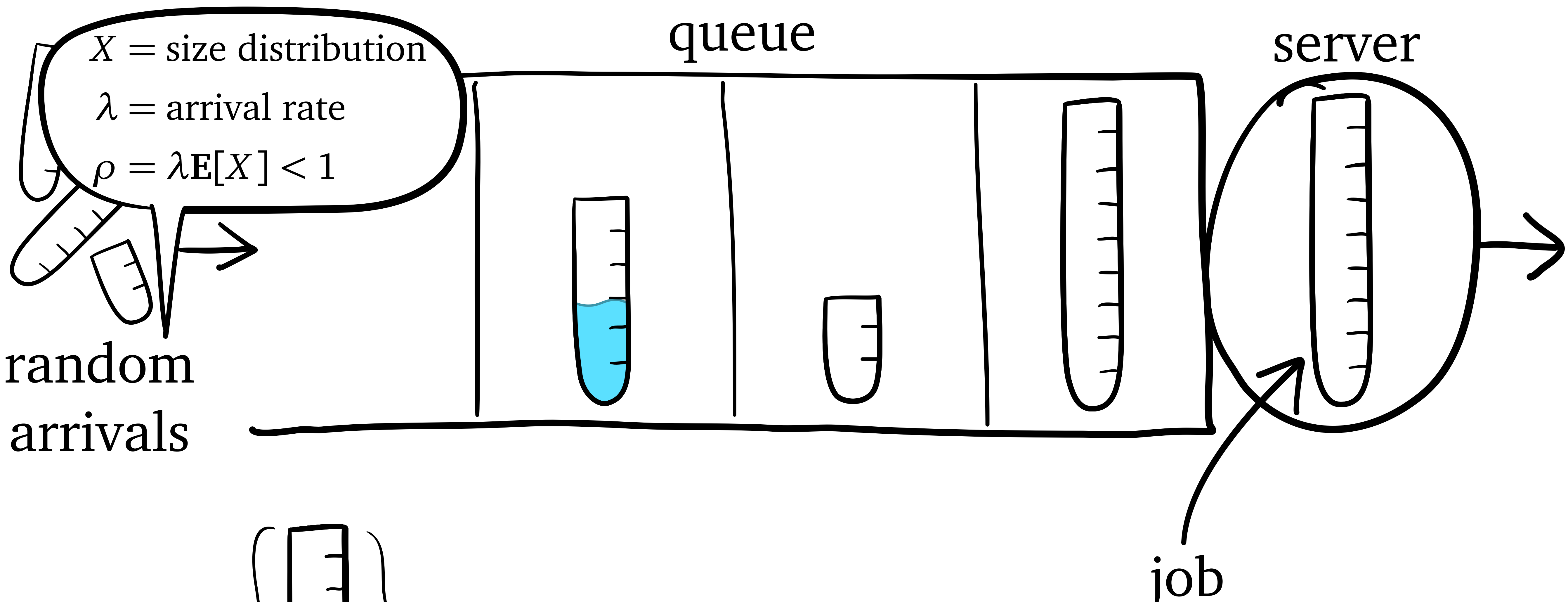
# M/G/1 Queue

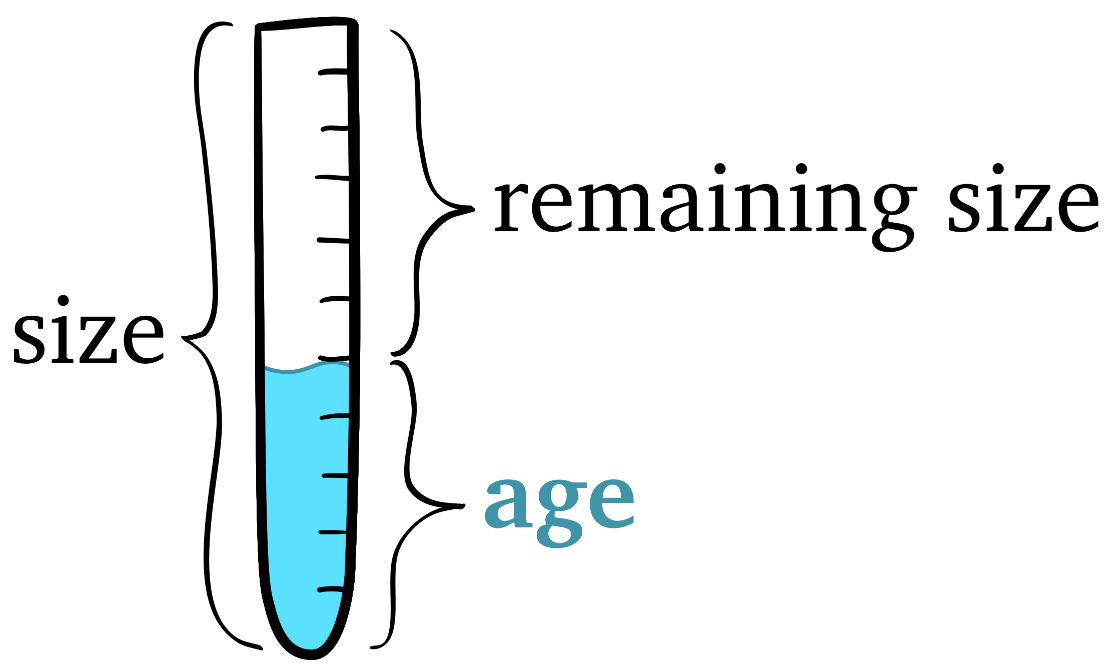



**Scheduling policy**:
picks which job to serve

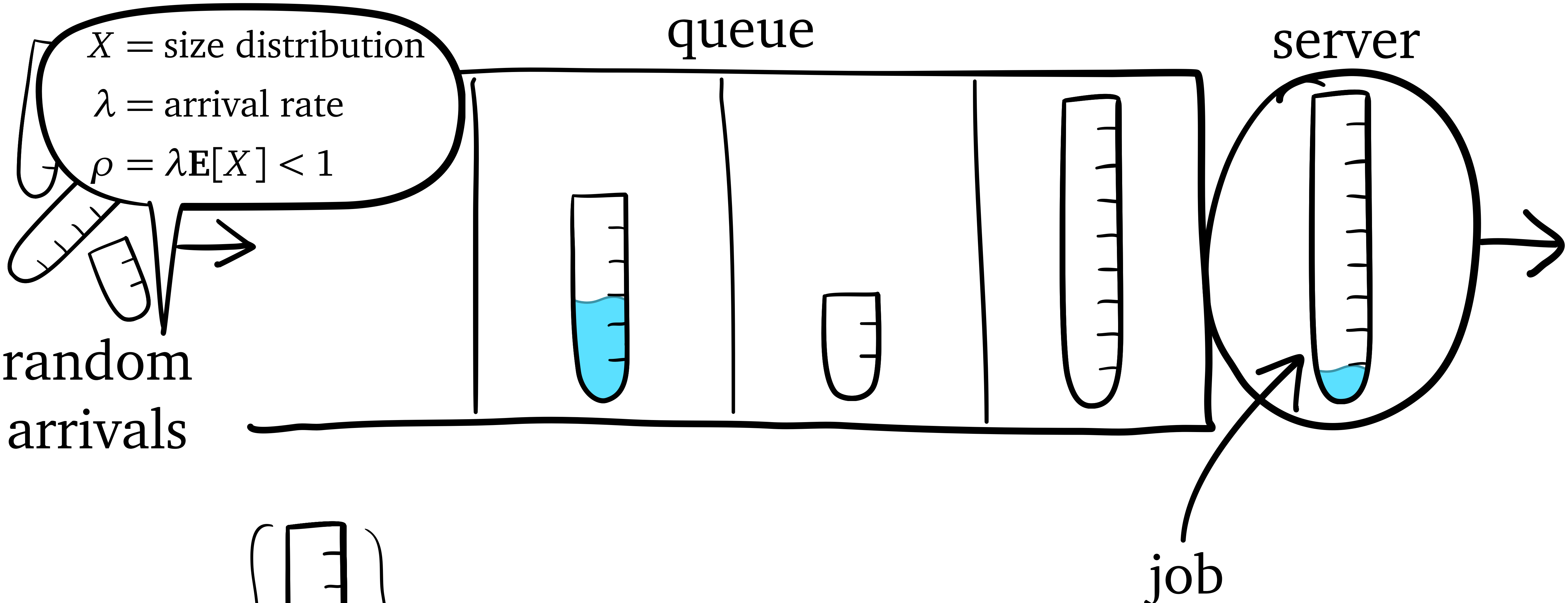
# M/G/1 Queue

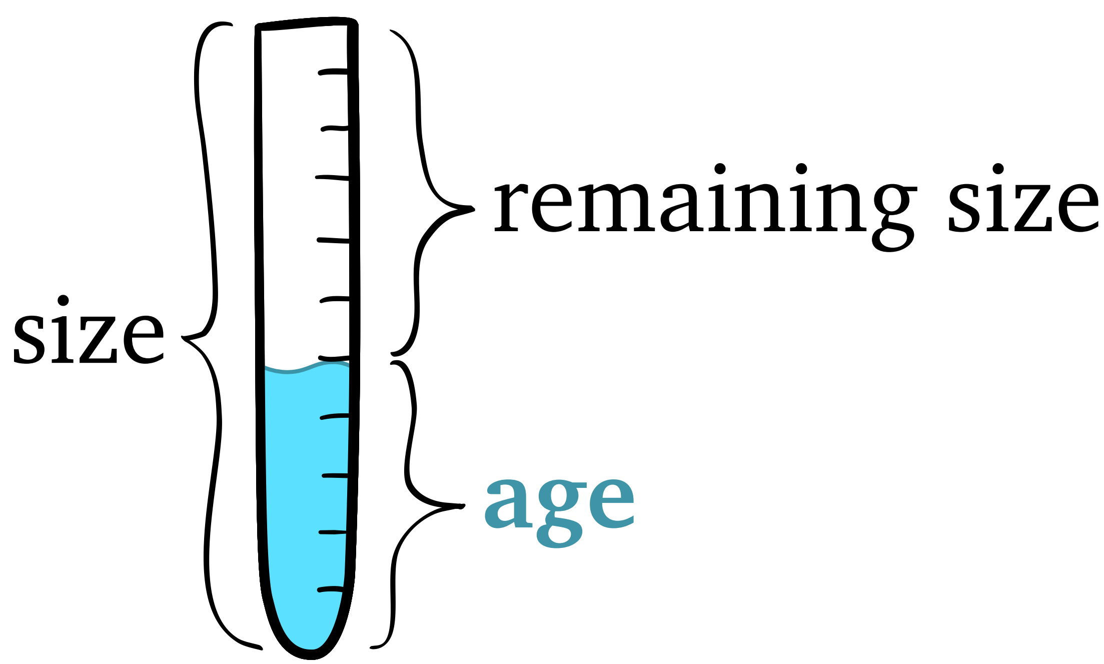



**Scheduling policy**:
picks which job to serve

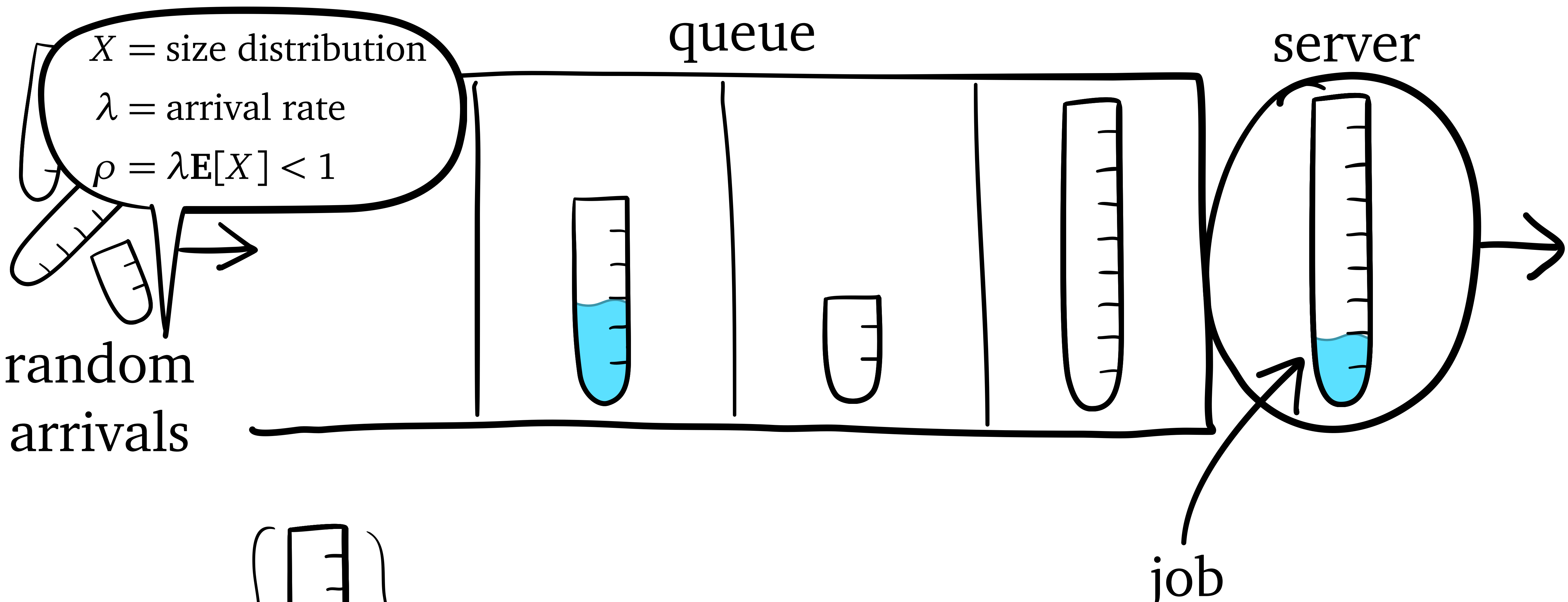
# M/G/1 Queue

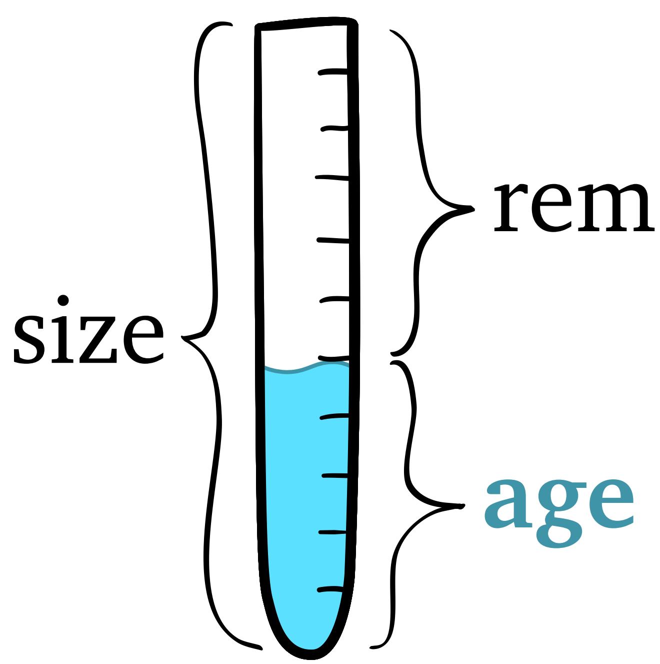



**Scheduling policy**:
picks which job to serve

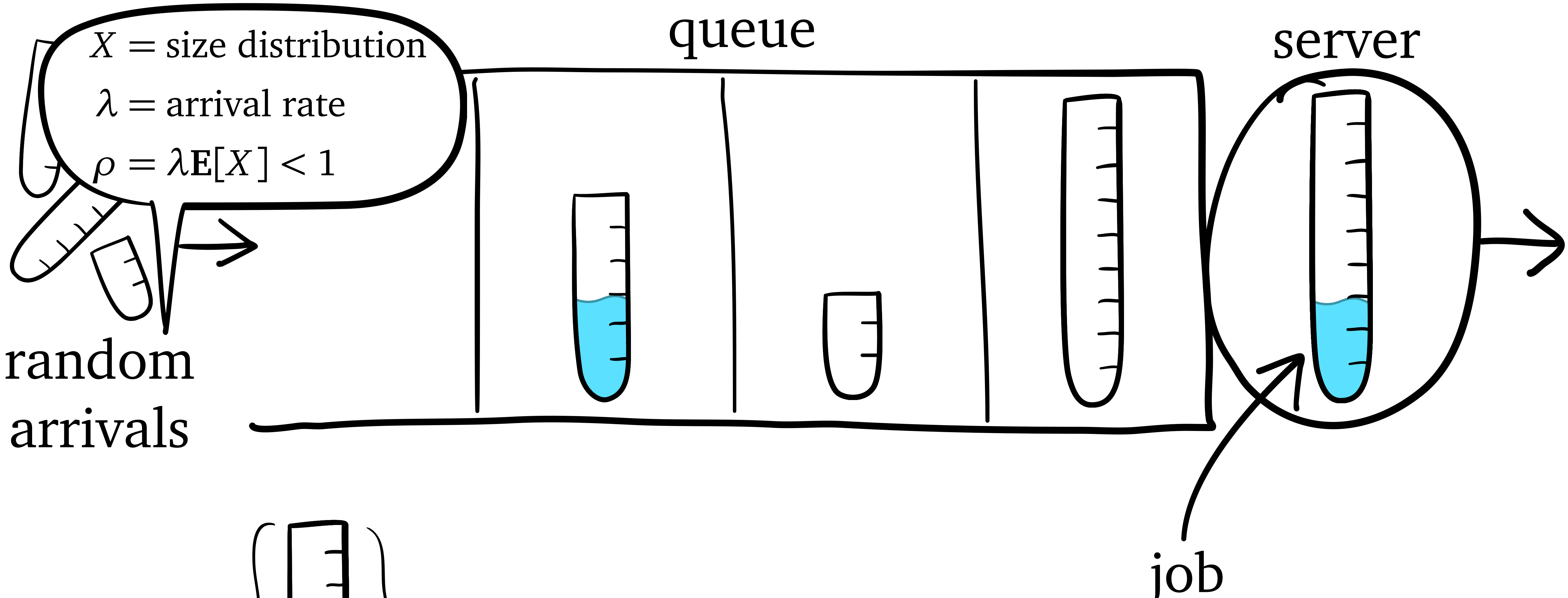
# M/G/1 Queue

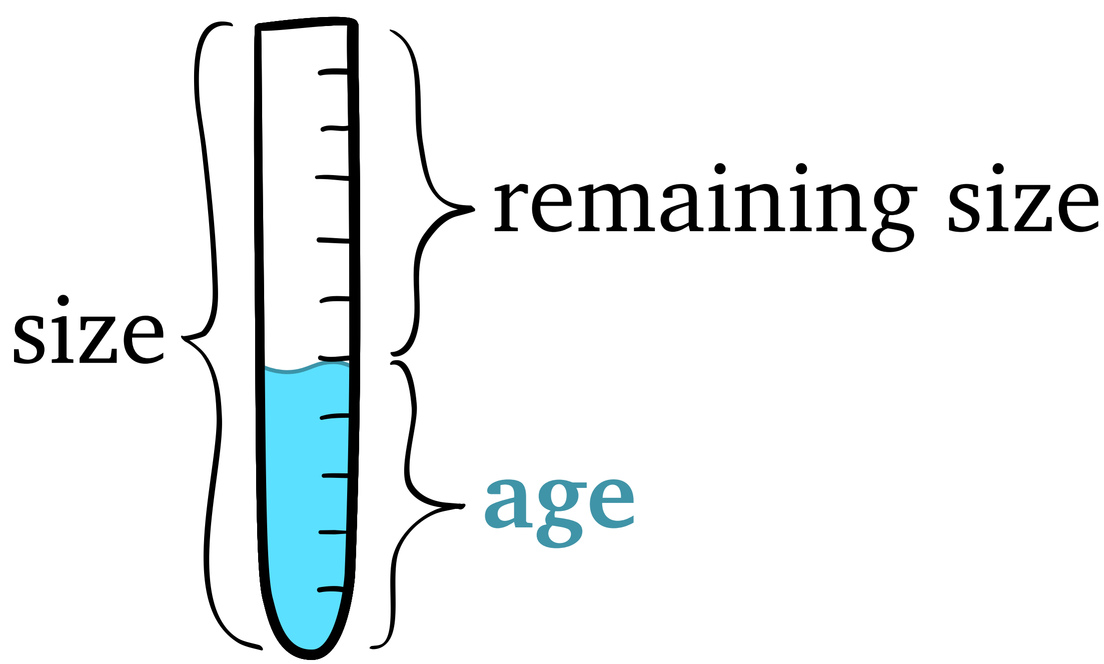



**Scheduling policy**:
picks which job to serve

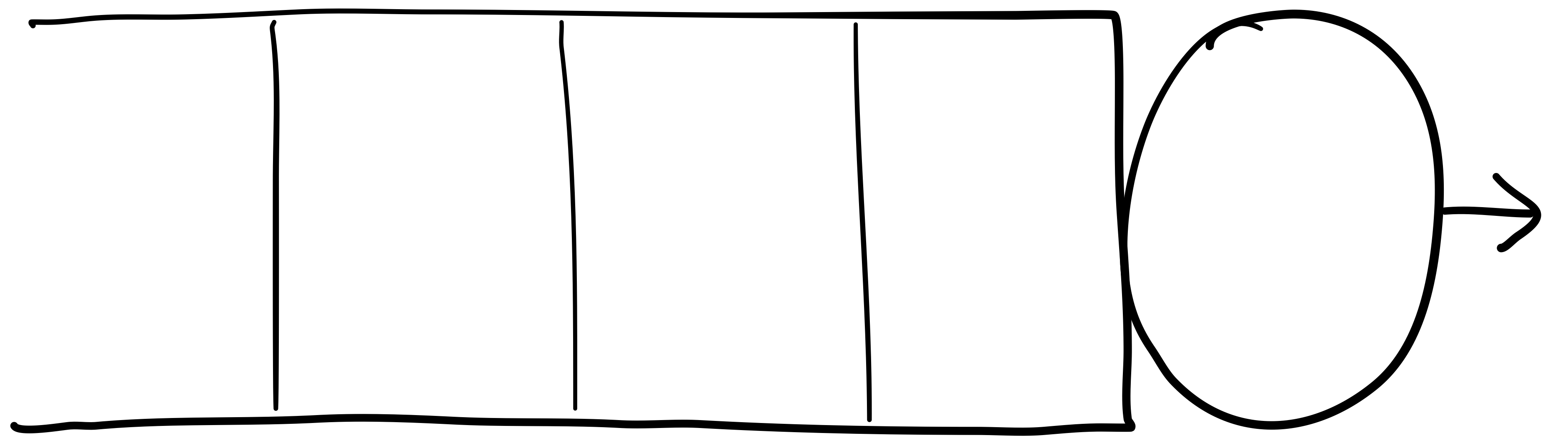
# M/G/1 Queue



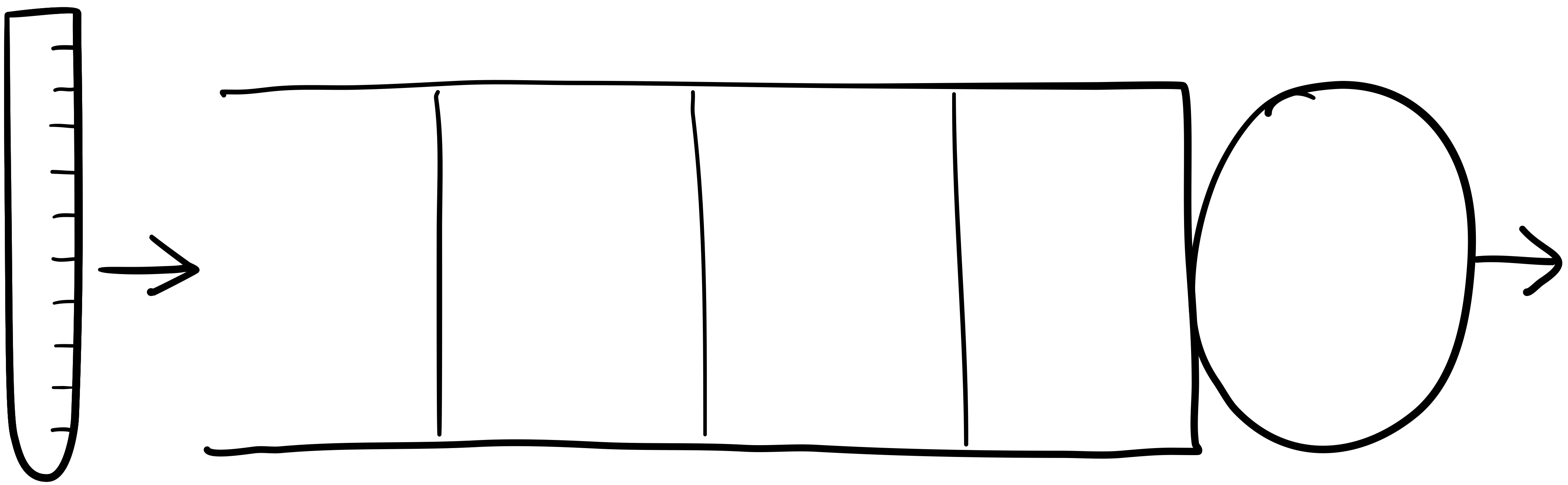




**Scheduling policy**:
picks which job to serve

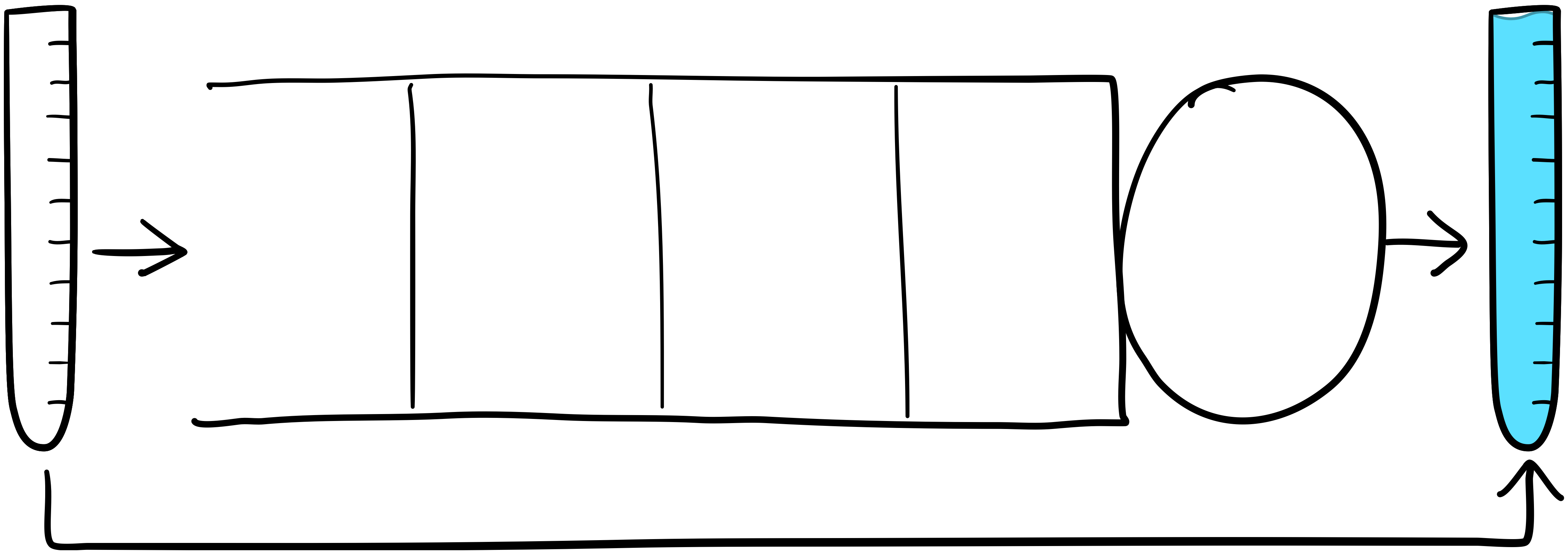
# Response Time

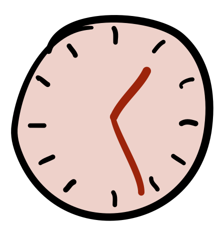


# Response Time

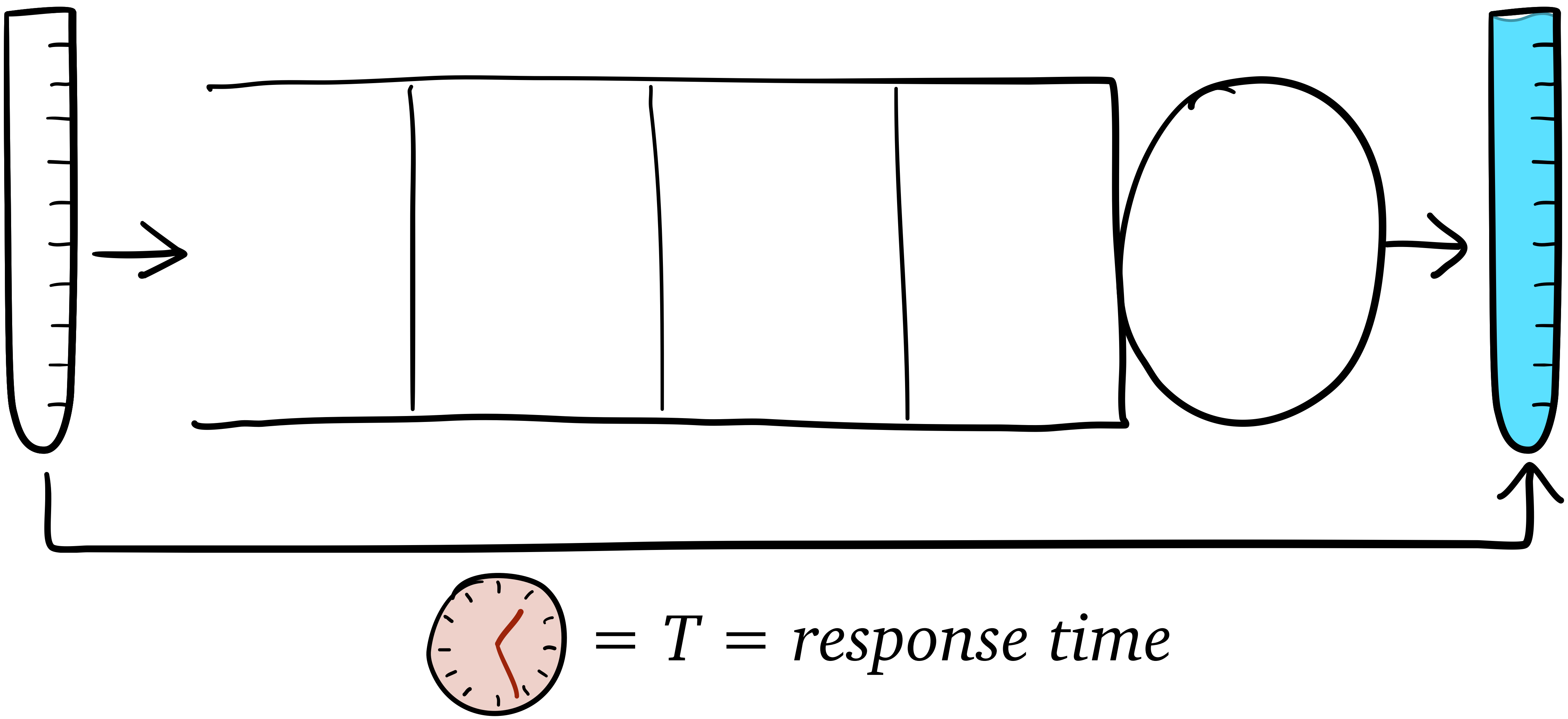


# Response Time



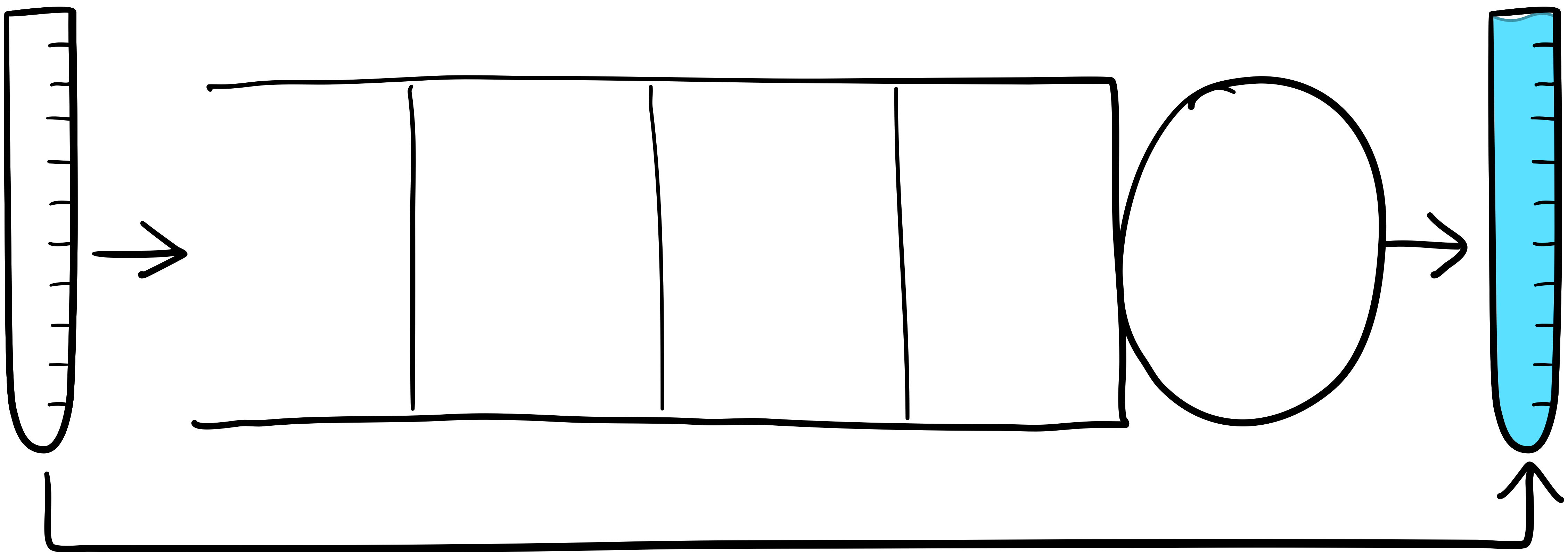
 $= T = \textit{response time}$

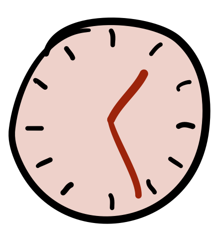
# Response Time




**Goal**: analyze *mean response time* $\mathbf{E}[T]$

# Response Time



 $= T =$ *response time*

**Goal**: analyze *mean response time* $\mathbf{E}[T]$

Depends on *scheduling policy*

# Impact of Scheduling

What scheduling policy minimizes $\mathbf{E}[T]$?

# Impact of Scheduling

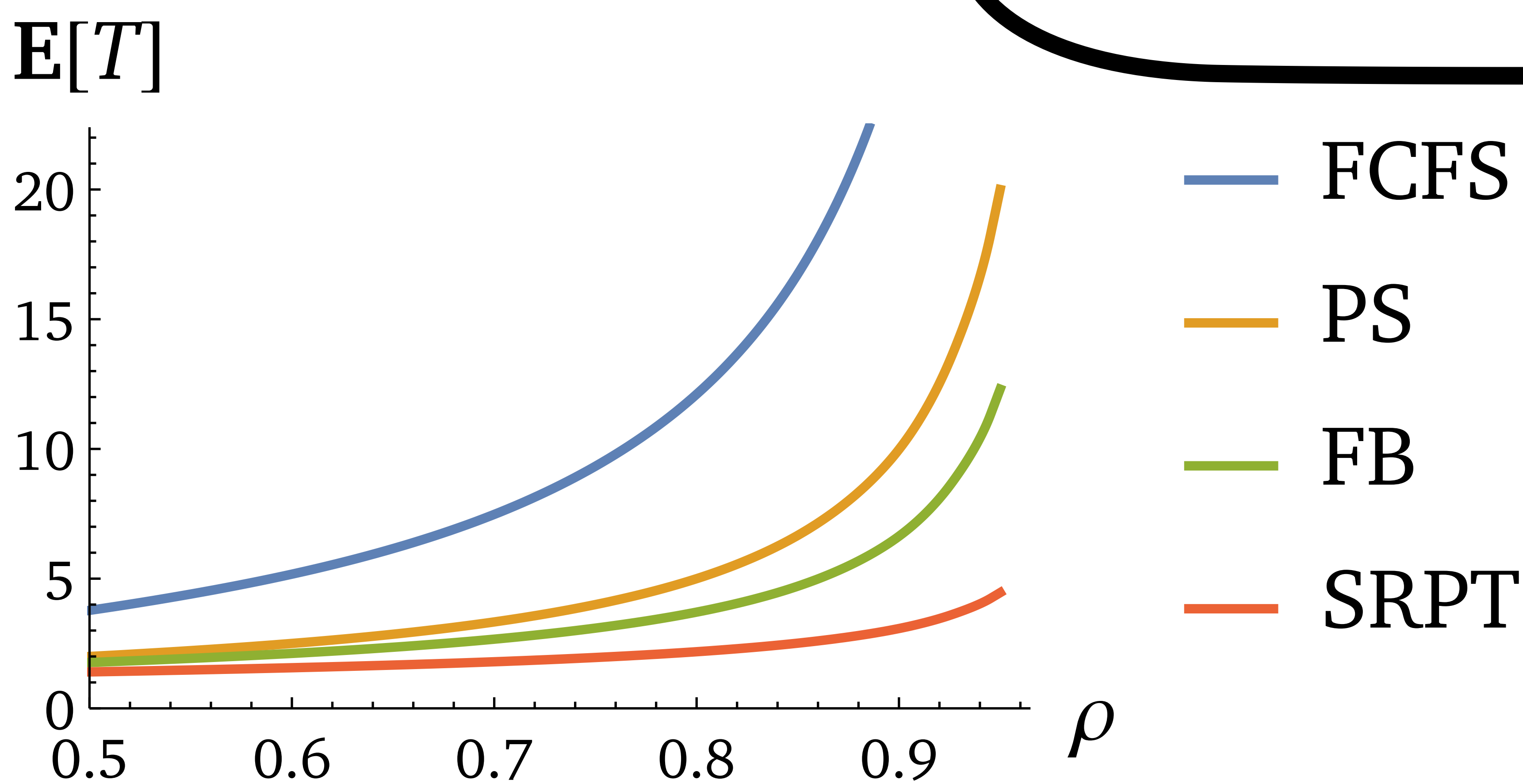
What scheduling policy minimizes $\mathbf{E}[T]$?

Shortest remaining processing time (SRPT)

# Impact of Scheduling

What scheduling policy minimizes $\mathbf{E}[T]$?

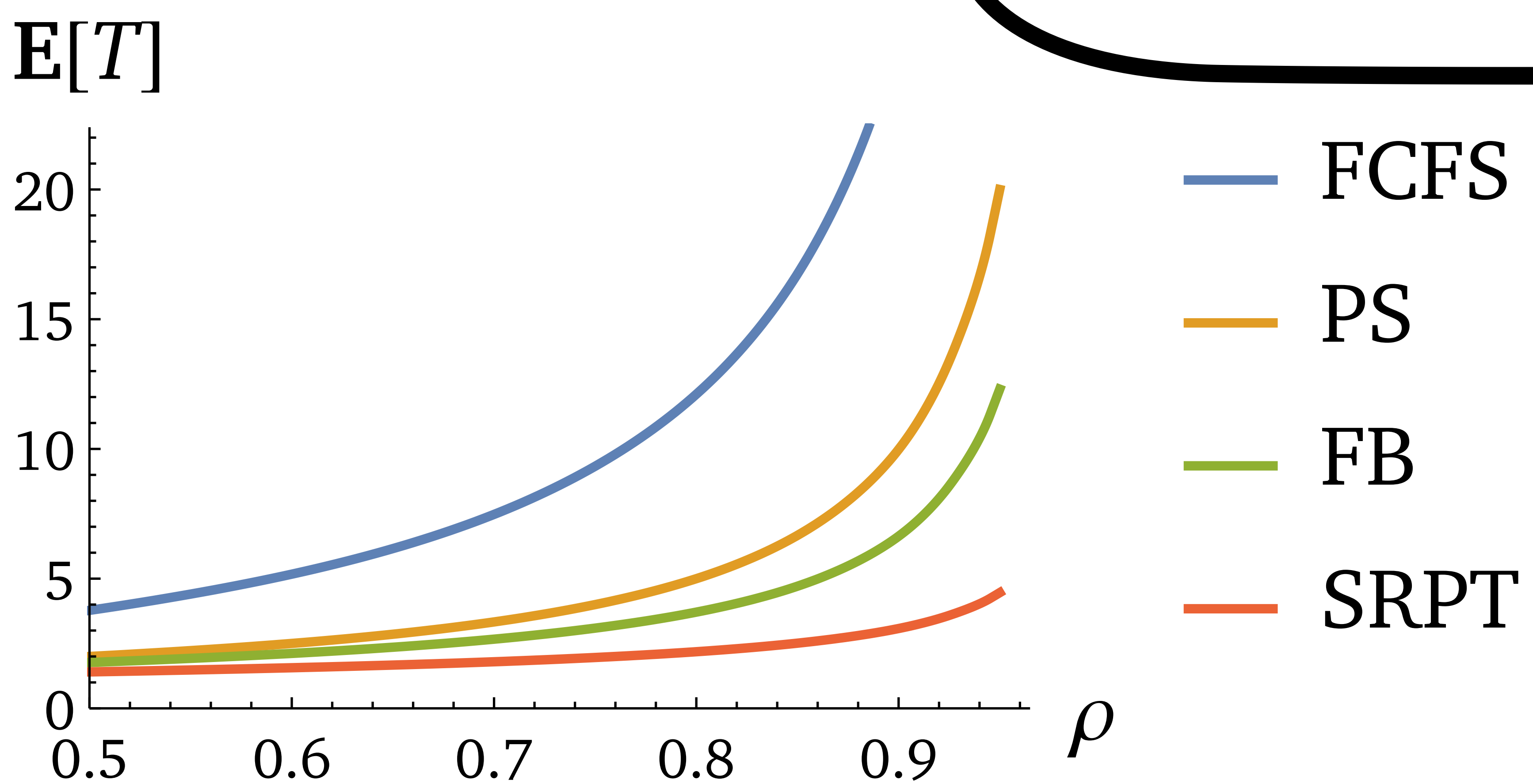
Shortest remaining processing time (SRPT)



# Impact of Scheduling

What scheduling policy minimizes $\mathbf{E}[T]$?

Shortest remaining processing time (SRPT)




… but nobody uses SRPT!

# Why Not SRPT?

# Why Not SRPT?

Unknown job sizes

# Why Not SRPT?

Unknown job sizes { FCFS (first come, first served)

# Why Not SRPT?

Unknown job sizes {
- FCFS (first come, first served)
- FB (foreground-background: least age)

# Why Not SRPT?

Unknown job sizes {
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least *expected* remaining size)

# Why Not SRPT?

Unknown job sizes {
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least *expected* remaining size)
- Gittins (optimal!)

# Why Not SRPT?

Unknown job sizes {
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least *expected* remaining size)
- Gittins (optimal!)

Hardware constraints

# Why Not SRPT?

Unknown job sizes:
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least *expected* remaining size)
- Gittins (optimal!)

Hardware constraints:
- "Discrete" SRPT
  (preempt only at checkpoints)

# Why Not SRPT?

Unknown job sizes:
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least *expected* remaining size)
- Gittins (optimal!)

Hardware constraints:
- "Discrete" SRPT (preempt only at checkpoints)
- "Bucketed" SRPT (limited number of priority levels)

# Why Not SRPT?

Unknown job sizes:
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least *expected* remaining size)
- Gittins (optimal!)

Hardware constraints:
- "Discrete" SRPT, FB, etc. (preempt only at checkpoints)
- "Bucketed" SRPT, FB, etc. (limited number of priority levels)

# Why Not SRPT?

Unknown job sizes:
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least *expected* remaining size)
- Gittins (optimal!)

Hardware constraints:
- "Discrete" SRPT, FB, etc. (preempt only at checkpoints)
- "Bucketed" SRPT, FB, etc. (limited number of priority levels)

Metric other than $\mathbf{E}[T]$

# Why Not SRPT?

Unknown job sizes
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least *expected* remaining size)
- Gittins (optimal!)

Hardware constraints
- "Discrete" SRPT, FB, etc. (preempt only at checkpoints)
- "Bucketed" SRPT, FB, etc. (limited number of priority levels)

Metric other than $\mathbf{E}[T]$
- Priority classes

# Why Not SRPT?

Unknown job sizes
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least *expected* remaining size)
- Gittins (optimal!)

Hardware constraints
- "Discrete" SRPT, FB, etc. (preempt only at checkpoints)
- "Bucketed" SRPT, FB, etc. (limited number of priority levels)

Metric other than $\mathbf{E}[T]$
- Priority classes
- RS (optimal for mean slowdown)

# Many Scheduling Policies

# Many Scheduling Policies

$\mathbf{E}[T]$ known

# Many Scheduling Policies

$\mathbf{E}[T]$ known

SRPT

# Many Scheduling Policies

$\mathbf{E}[T]$ known

SRPT

FCFS

# Many Scheduling Policies

$\mathbf{E}[T]$ known

SRPT

FCFS

FB

# Many Scheduling Policies

$\mathbf{E}[T]$ known

SRPT

FCFS

FB

Simple priority classes

# Many Scheduling Policies

$\mathbf{E}[T]$ known

$\mathbf{E}[T]$ unknown!

SRPT

FCFS

FB

Simple priority classes

# Many Scheduling Policies

**$\mathbf{E}[T]$ known**

SRPT

FCFS

FB

Simple priority classes

**$\mathbf{E}[T]$ unknown!**

SERPT

Gittins

Discrete SRPT

Discrete FB

Bucketed SRPT

Bucketed FB

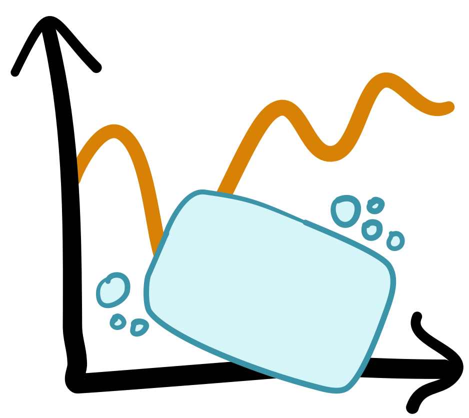
RS*

Complex priority classes

… and more!

# Many Scheduling Policies

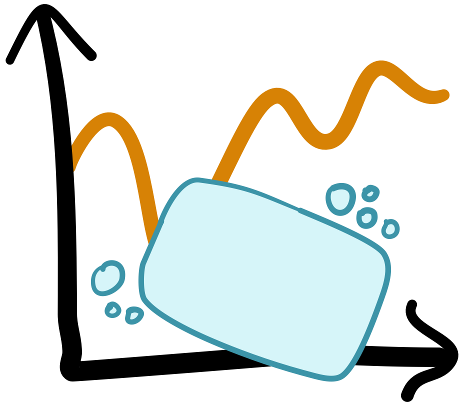
| **E**[$T$] known | **E**[$T$] unknown! |
| --- | --- |
| SRPT | SERPT |
| FCFS | Gittins |
| FB | Discrete SRPT |
| Simple priority classes | Discrete FB |
| | Bucketed SRPT |
| | Bucketed FB |
| | RS* |
| | Complex priority classes |
| | … and more! |



SOAP

# SOAP

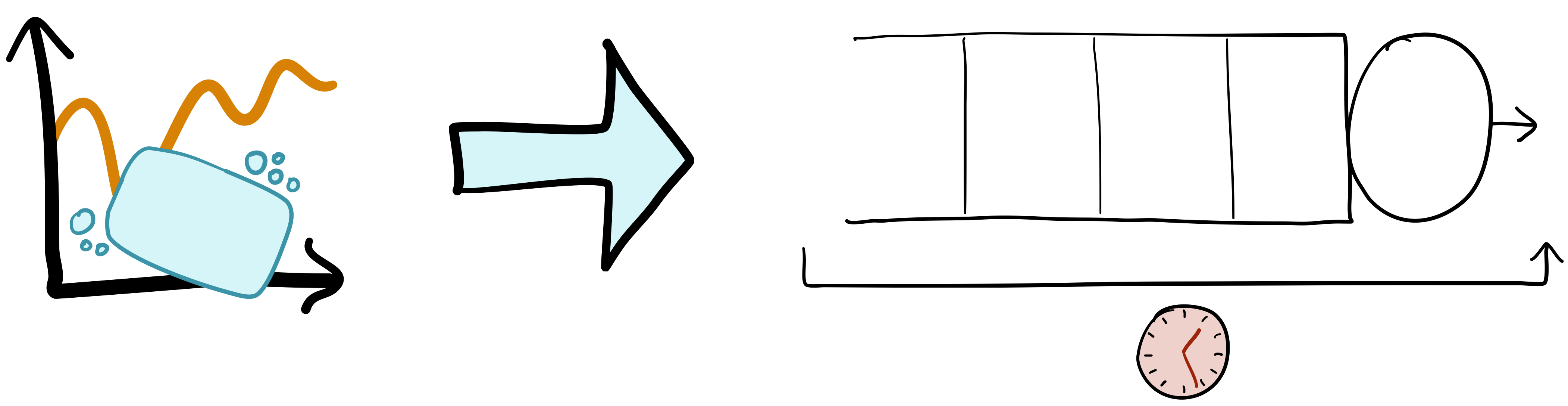
Broad *class* of scheduling policies…



# SOAP

Broad *class* of scheduling policies…

… with *universal* response time analysis

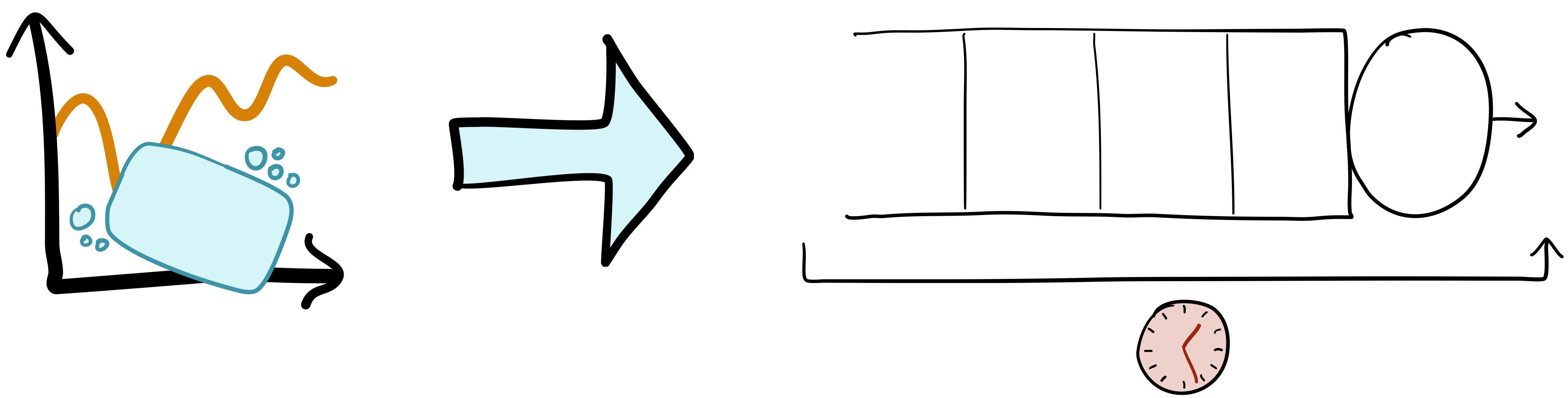


# SOAP

Schedule Ordered by Age-based Priority

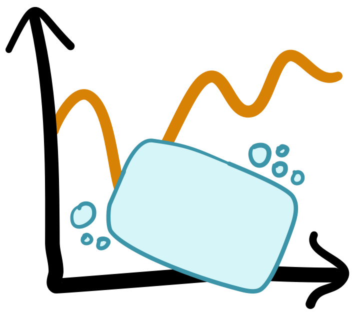
Broad *class* of scheduling policies…

… with *universal* response time analysis



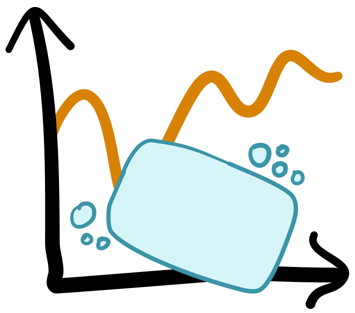
# Outline

# Outline

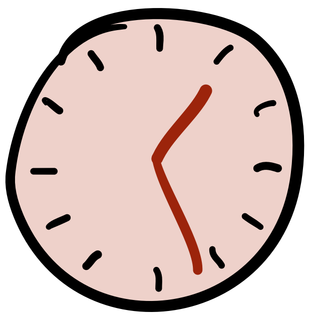


**Part 1**: *defining* **SOAP** policies

# Outline

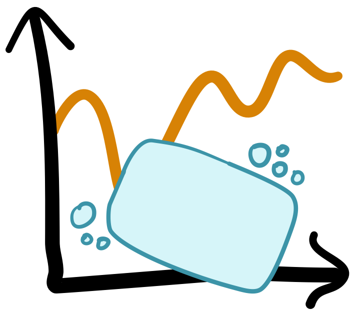


**Part 1**: *defining* **SOAP** policies

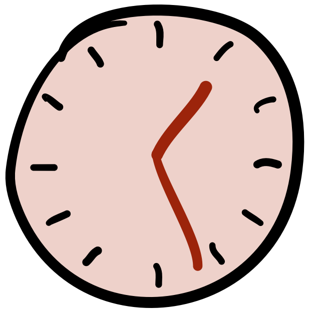


**Part 2**: *analyzing* **SOAP** policies

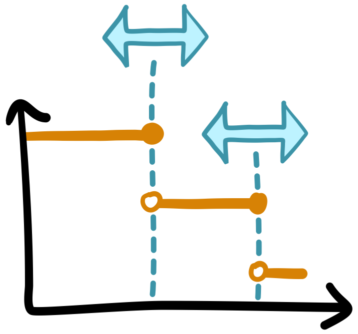
# Outline



**Part 1**: *defining* **SOAP** policies

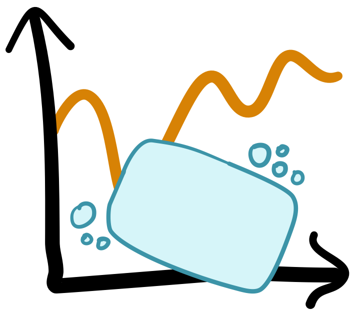


**Part 2**: *analyzing* **SOAP** policies

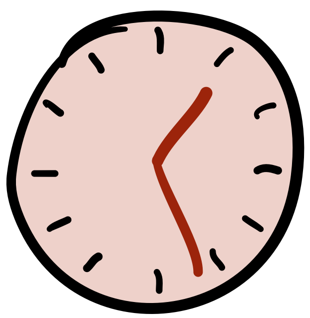


**Part 3**: *policy design* with **SOAP**

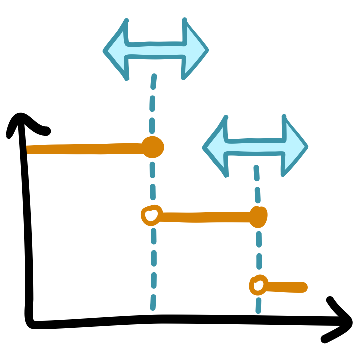
# Outline



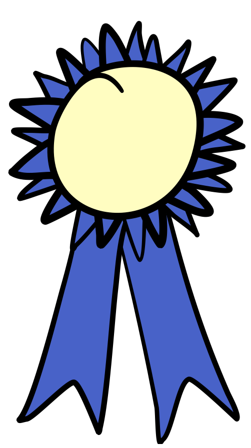
**Part 1**: *defining* **SOAP** policies



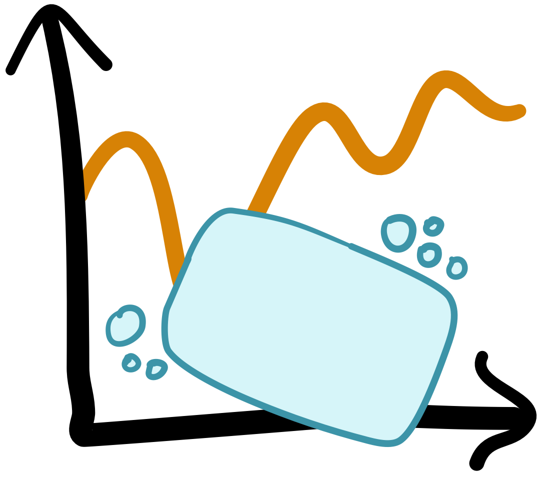
**Part 2**: *analyzing* **SOAP** policies



**Part 3**: *policy design* with **SOAP**



**Part 4**: *optimality proofs* with **SOAP**

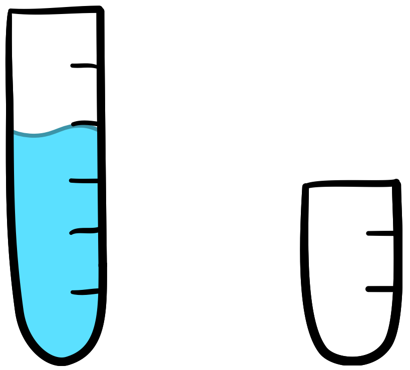


# Part 1:
*defining* **SOAP** policies

# Scheduling with **Ranks**

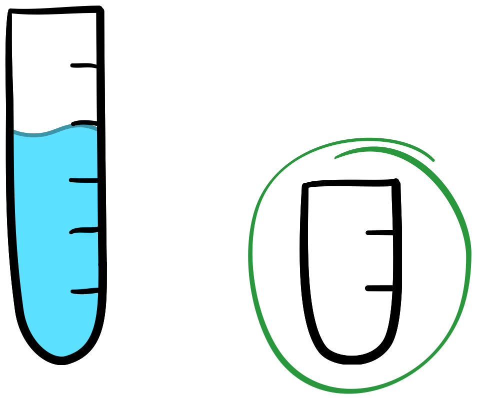
# Scheduling with **Ranks**

**FB**
serve by least age



# Scheduling with **Ranks**

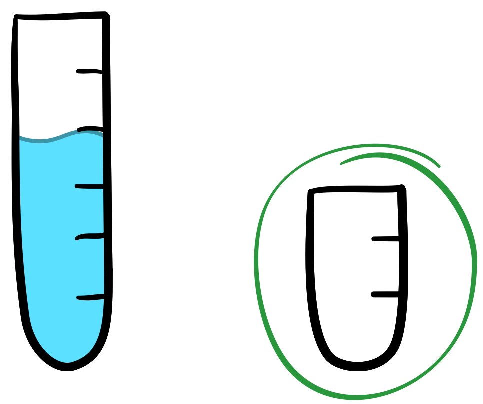
**FB**
serve by least age



# Scheduling with **Ranks**

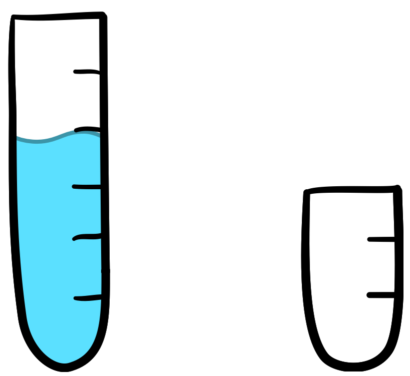
**FB**

serve by least age



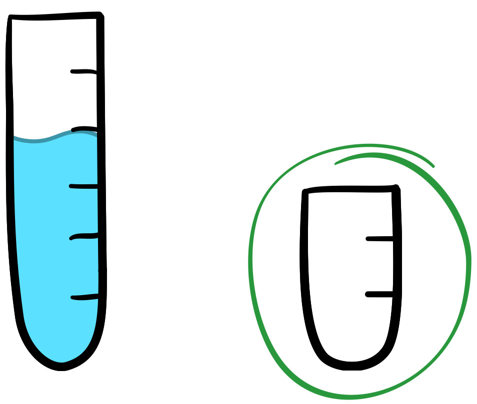
**SRPT**

serve by least remaining size

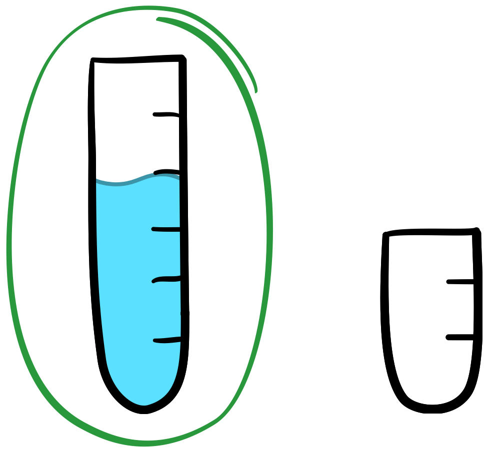


# Scheduling with Ranks

**FB**
serve by least age

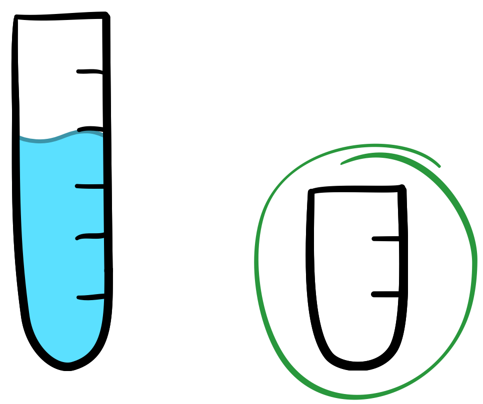


**SRPT**
serve by least remaining size

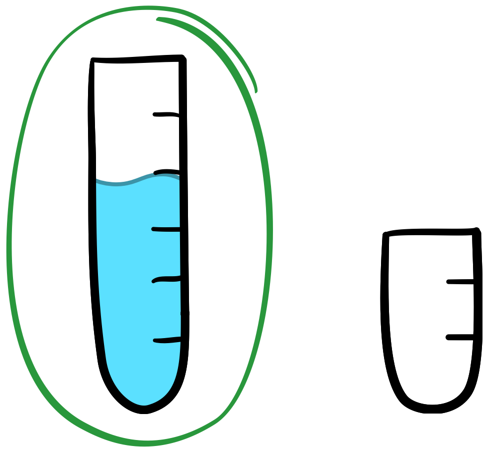


# Scheduling with Ranks

**FB**
serve by least age



**SRPT**
serve by least remaining size

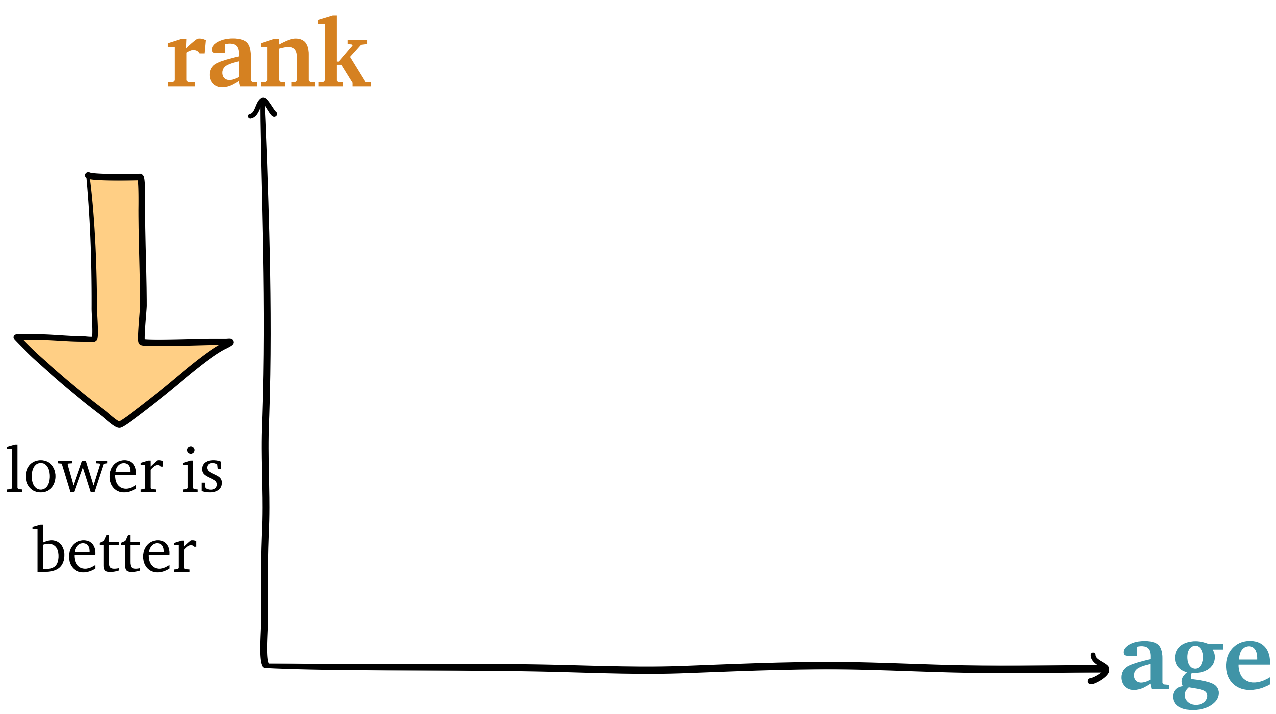


**Common theme**: a job's **rank** (priority) depends on its **age**

# Scheduling with **Ranks**

**FB**

serve by least age



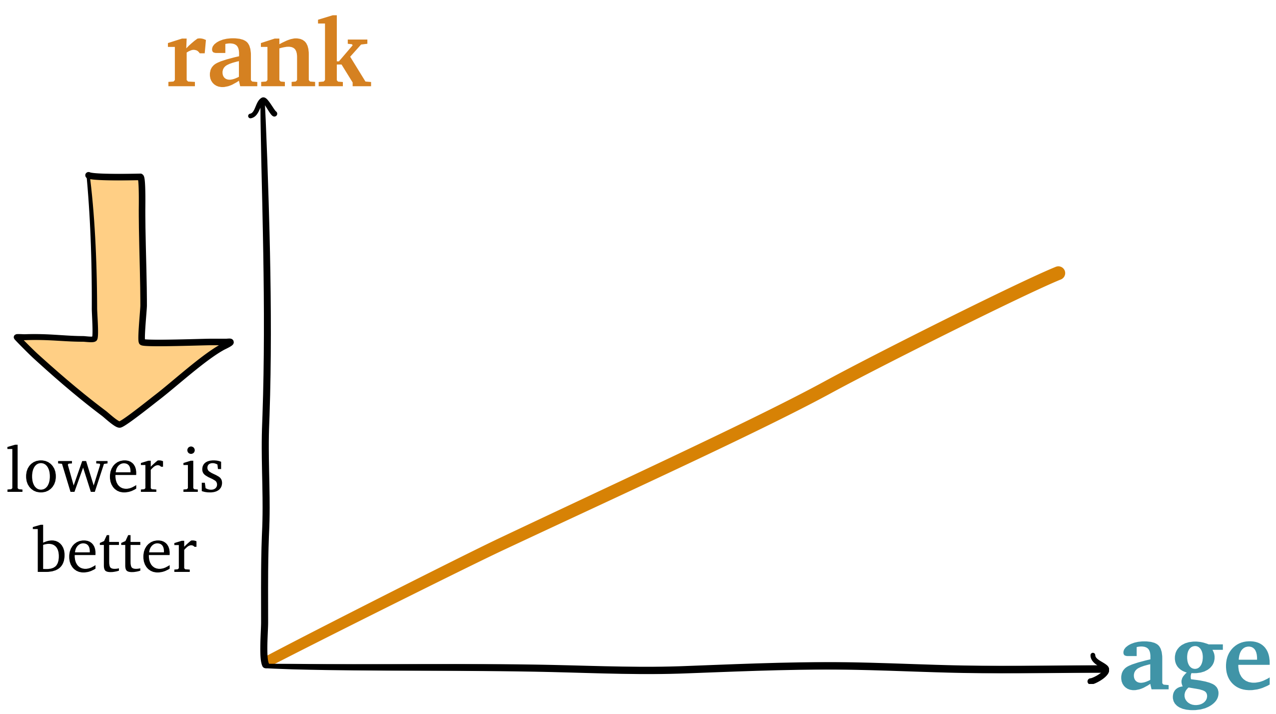

**SRPT**

serve by least remaining size


**Common theme**: a job's **rank** (priority) depends on its **age**

# Scheduling with Ranks

**FB**
serve by least age



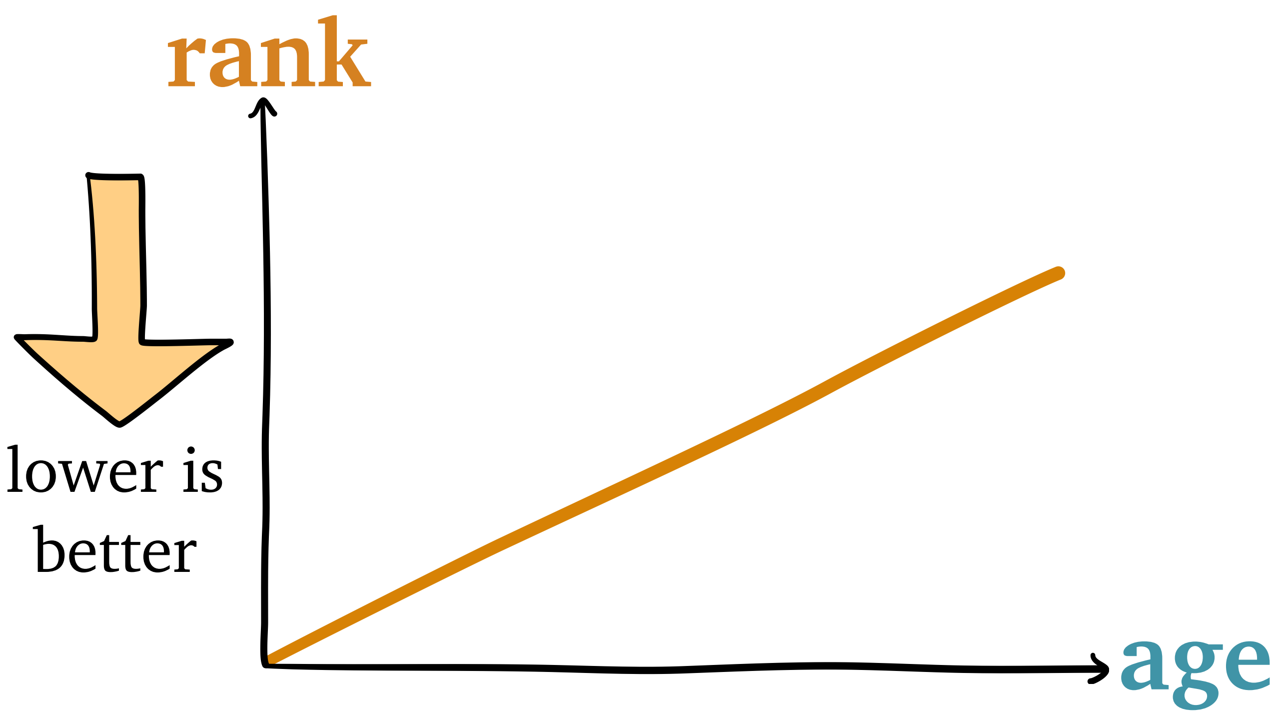

**SRPT**
serve by least remaining size


**Common theme**: a job's **rank** (priority) depends on its **age**

# Scheduling with **Ranks**

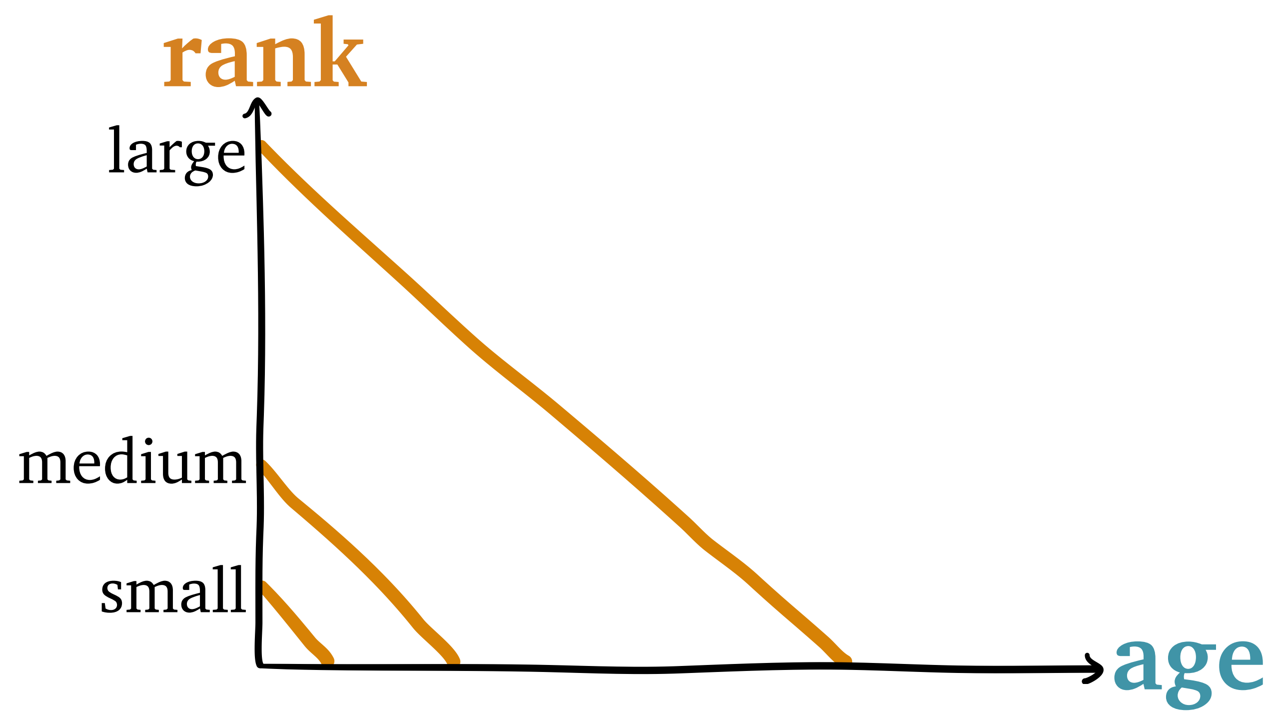
**FB**

serve by least age

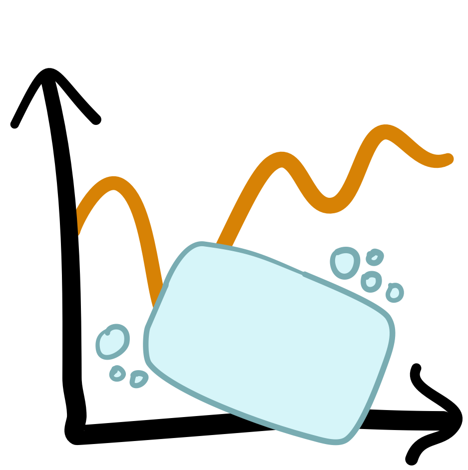



**SRPT**

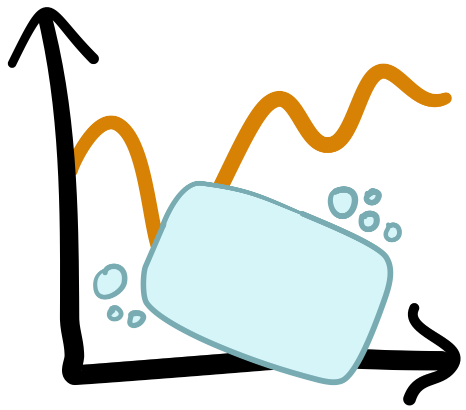
serve by least remaining size




**Common theme**: a job's **rank** (priority) depends on its **age**

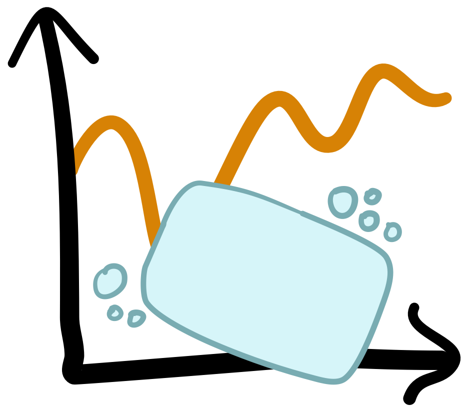


A ***SOAP*** *policy* is a ***rank*** *function* with one rule:



A ***SOAP*** *policy* is a ***rank*** *function* with one rule:

always serve the job of *minimum* ***rank***



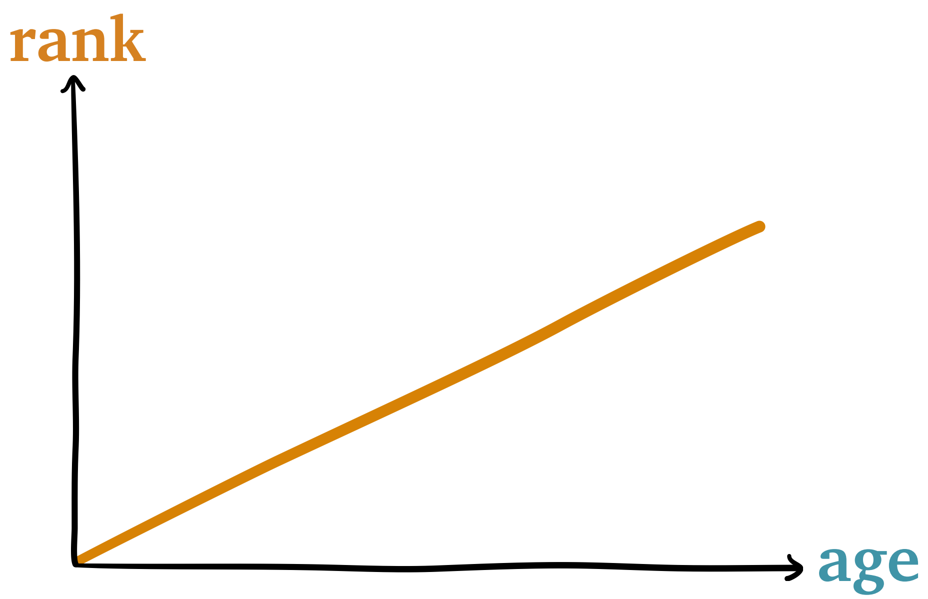
A ***SOAP*** *policy* is a ***rank*** *function* with one rule:

always serve the job of *minimum* ***rank***

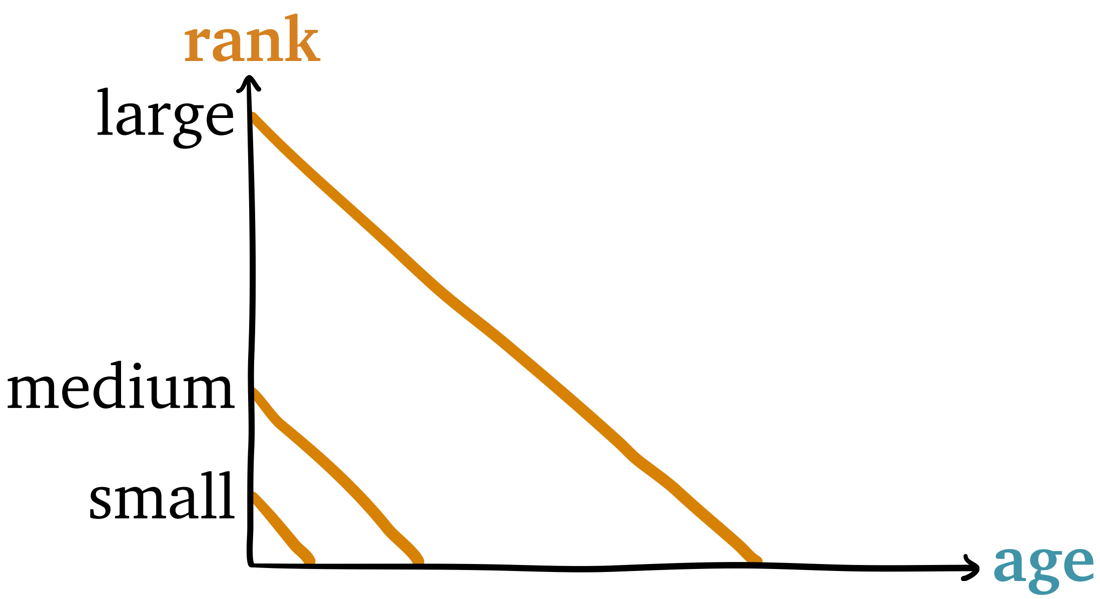
(break ties FCFS)

# Classic **SOAP** Policies

**FB**

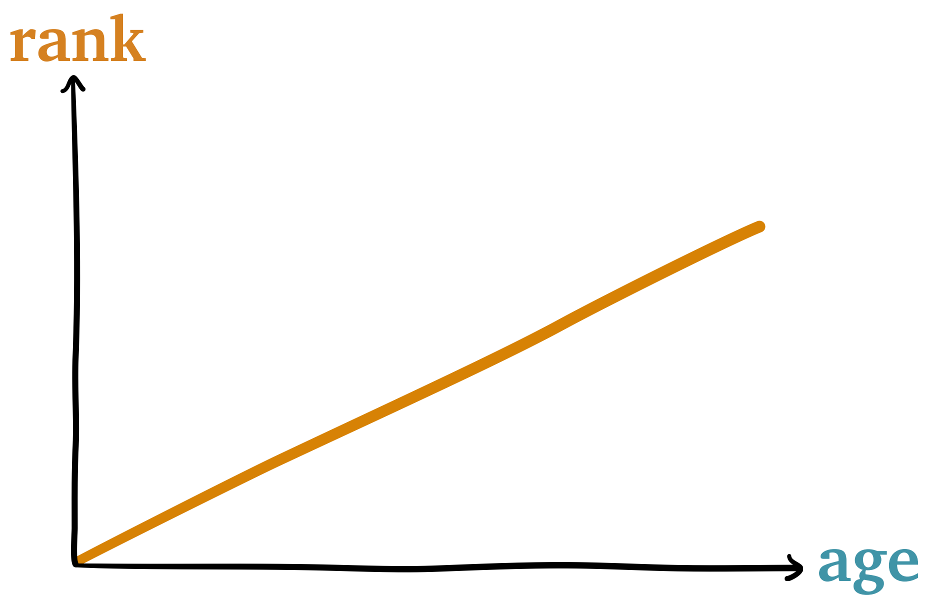



**SRPT**

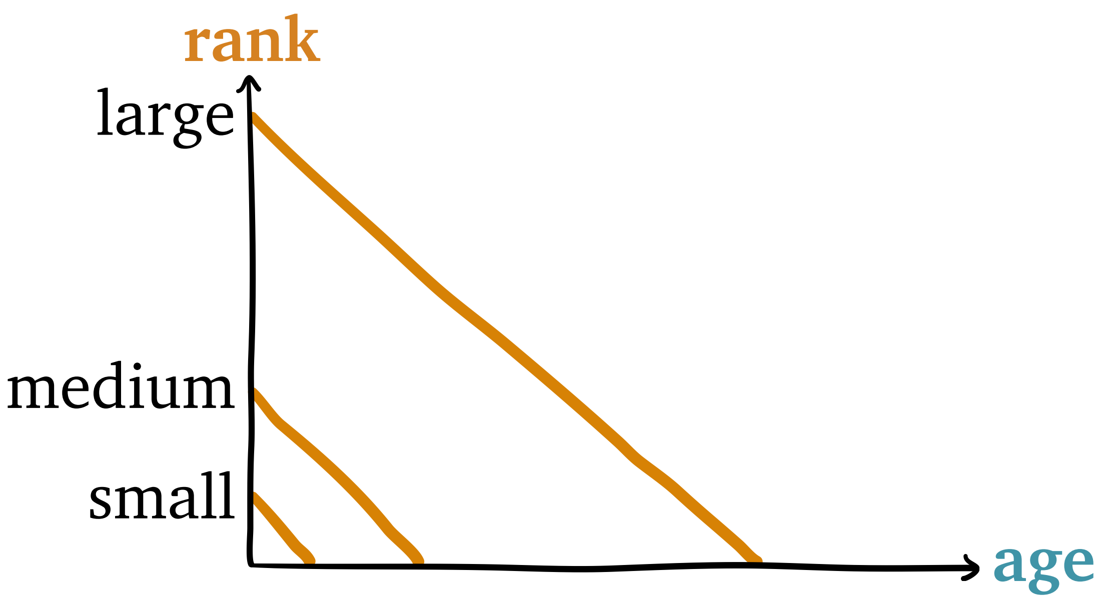


# Classic **SOAP** Policies

**FB**

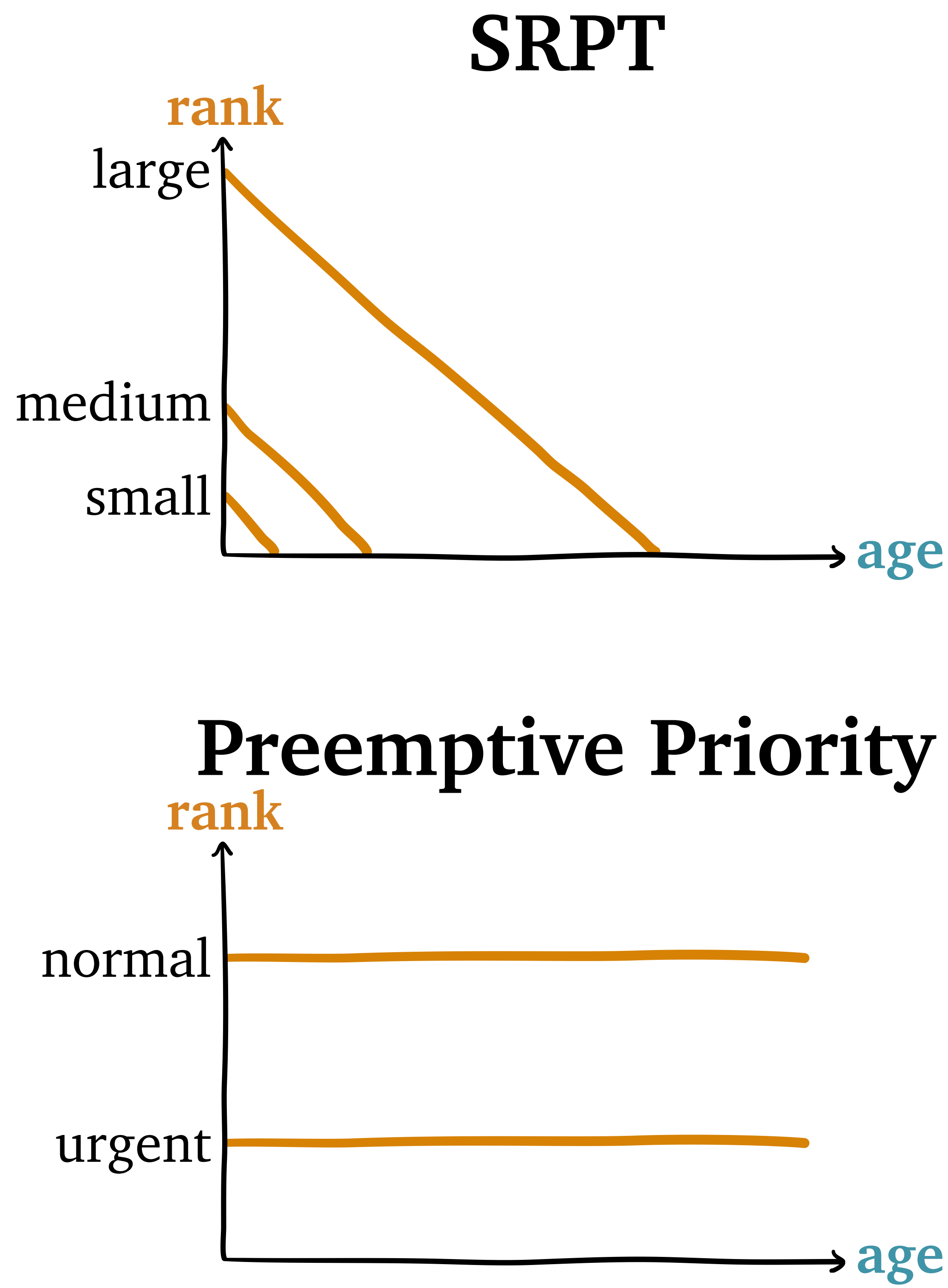



**SRPT**

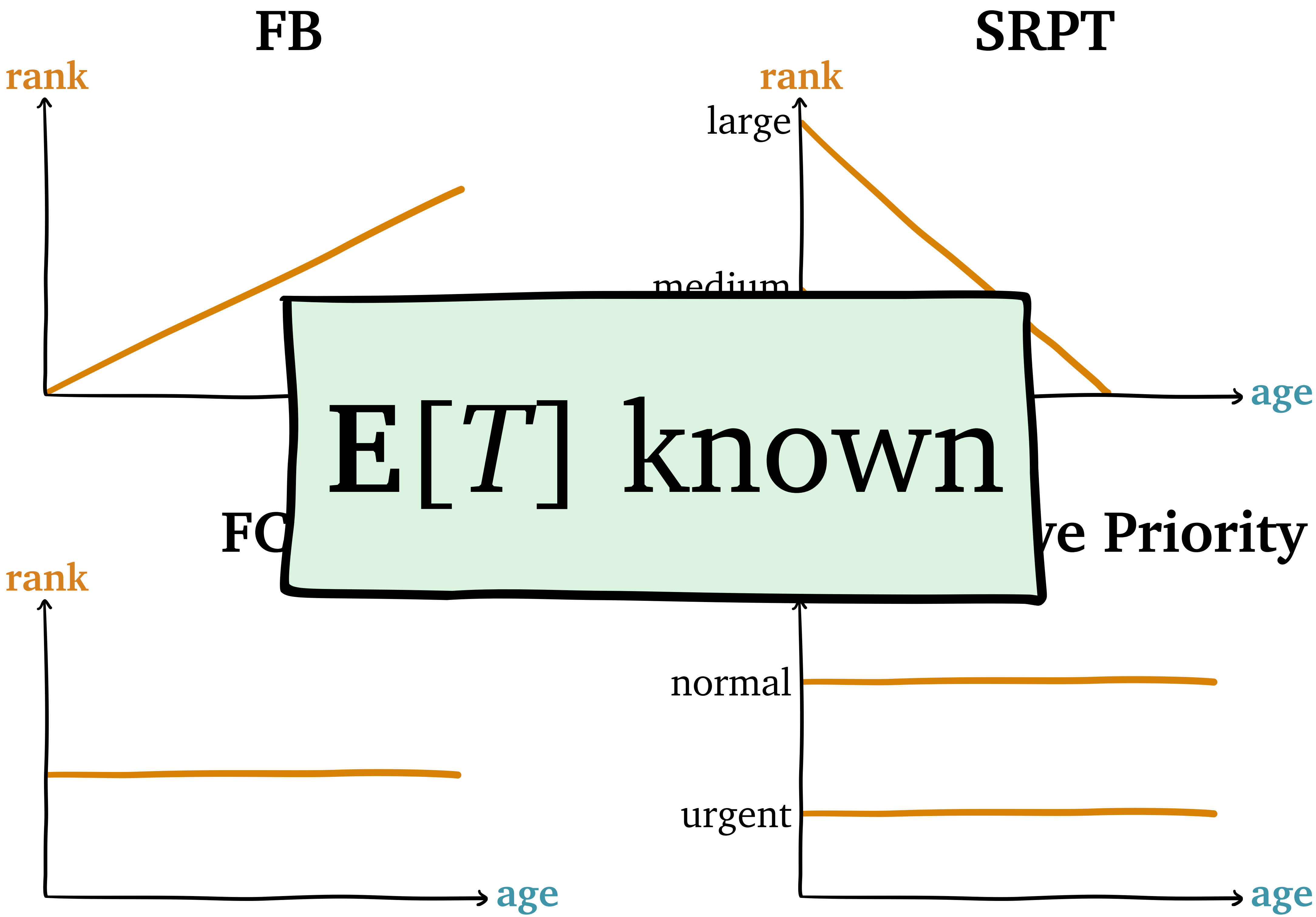



**FCFS**

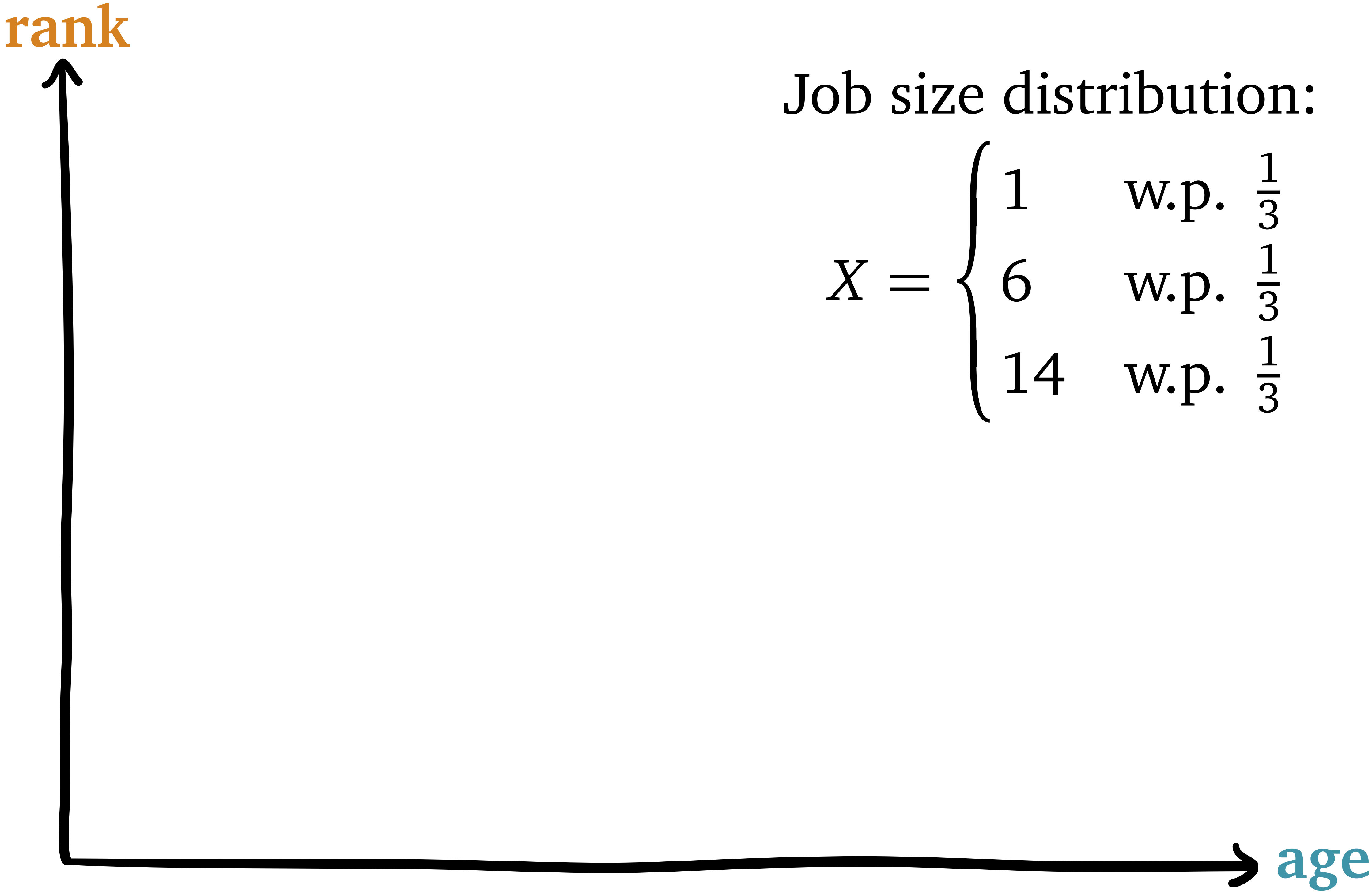
# Classic SOAP Policies



# Classic **SOAP** Policies



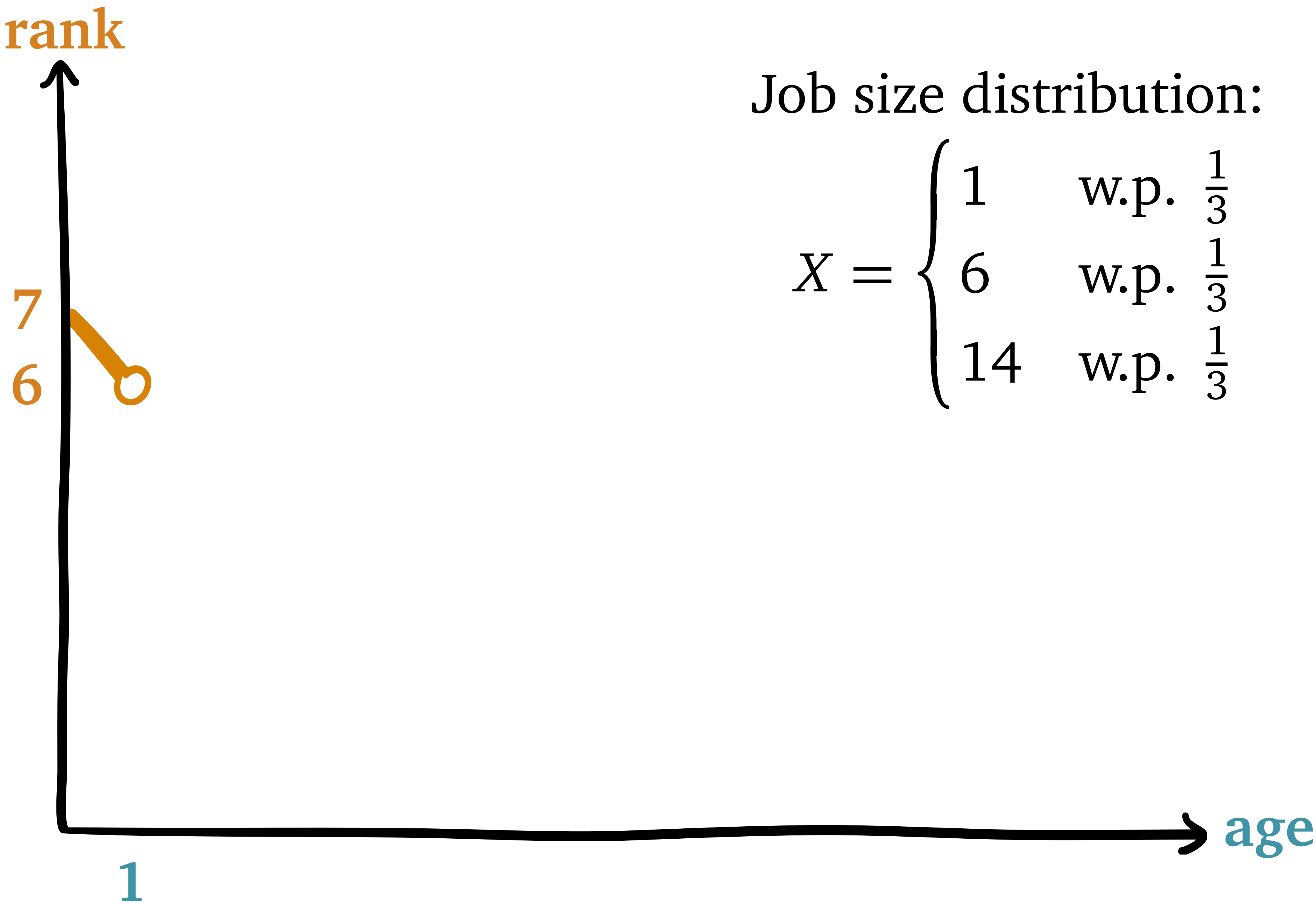
# SOAP Policy: SERPT




Job size distribution:

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

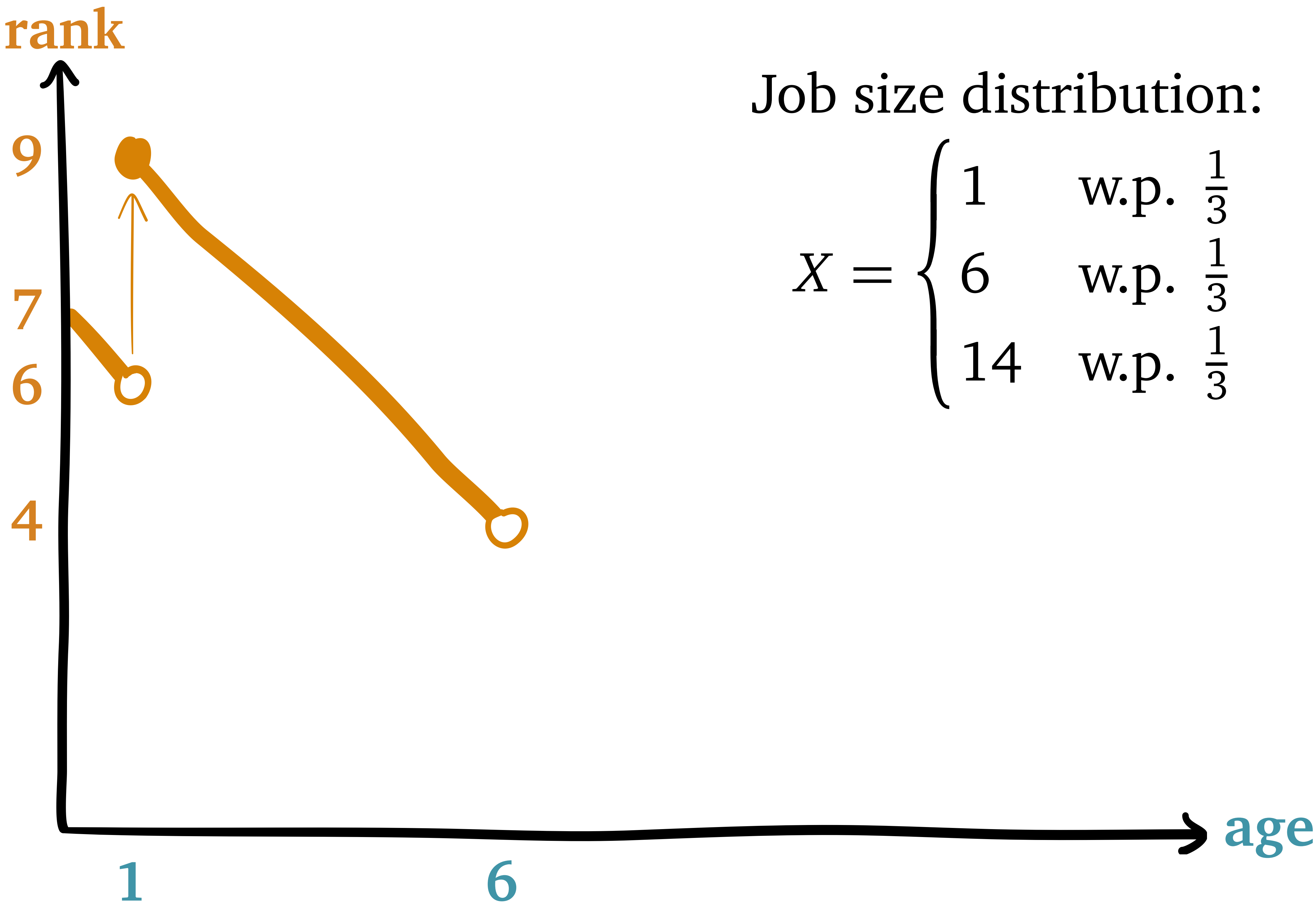
# SOAP Policy: SERPT




Job size distribution:

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

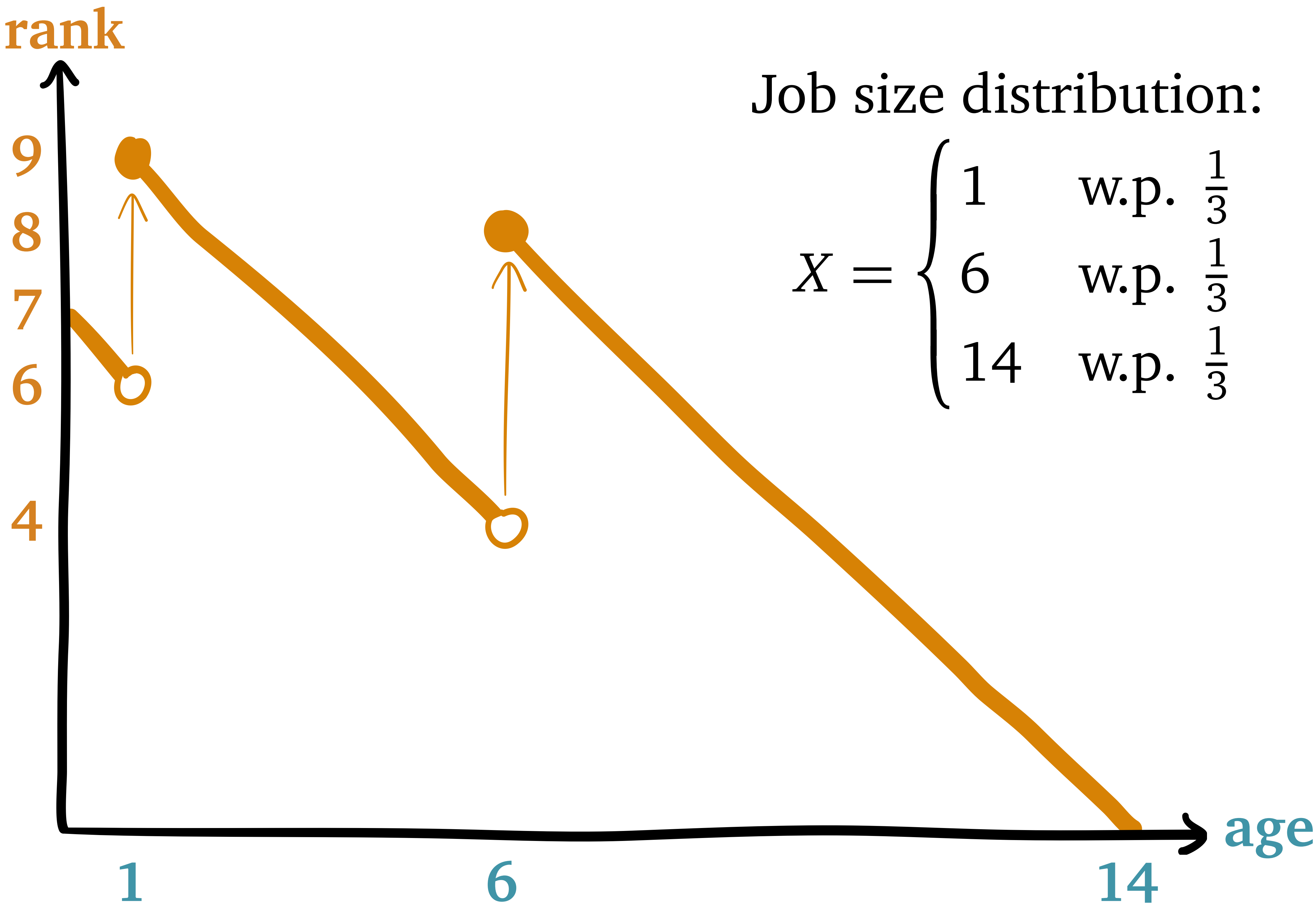
# SOAP Policy: SERPT




Job size distribution:

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

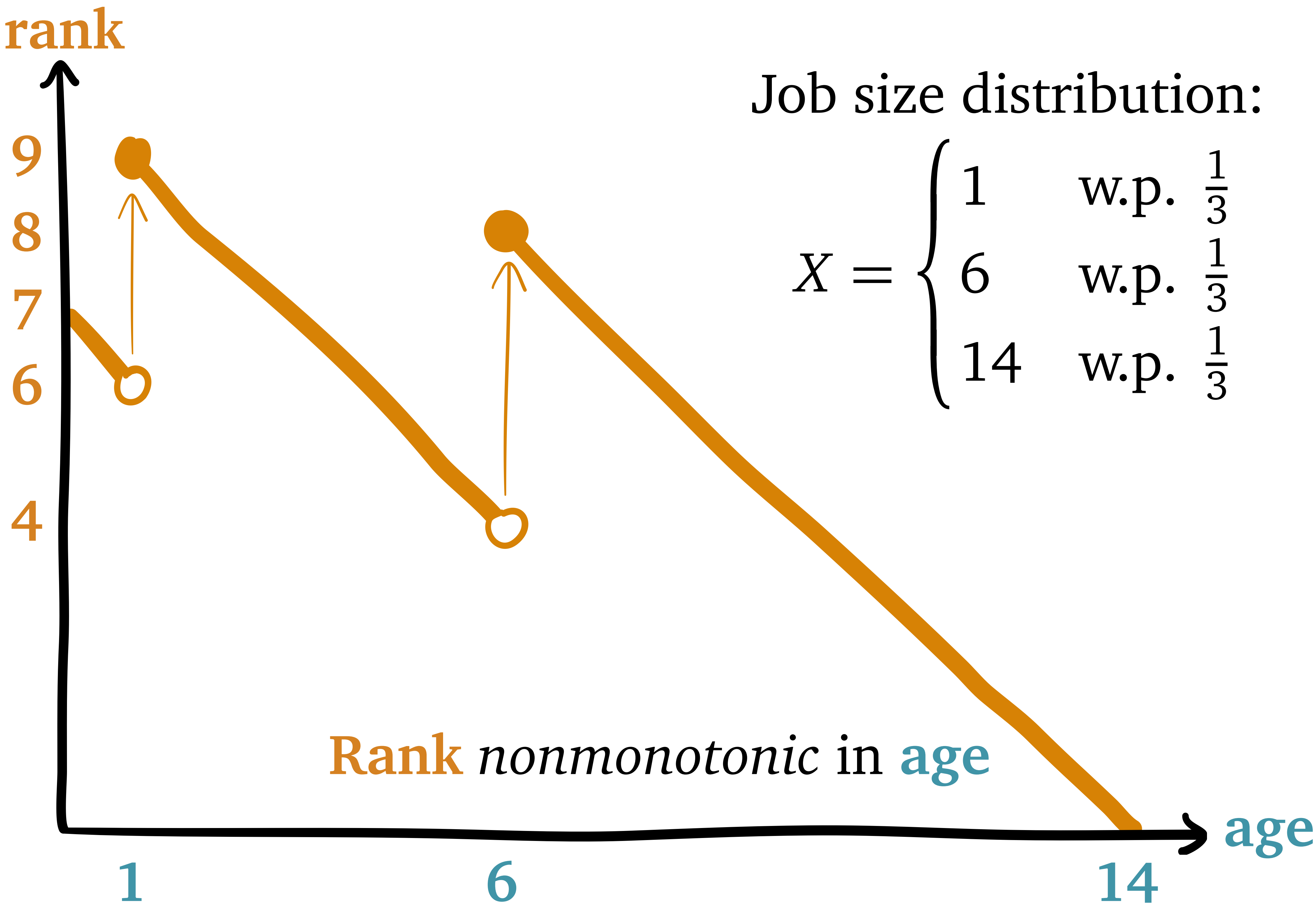
# SOAP Policy: SERPT




Job size distribution:

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# SOAP Policy: SERPT

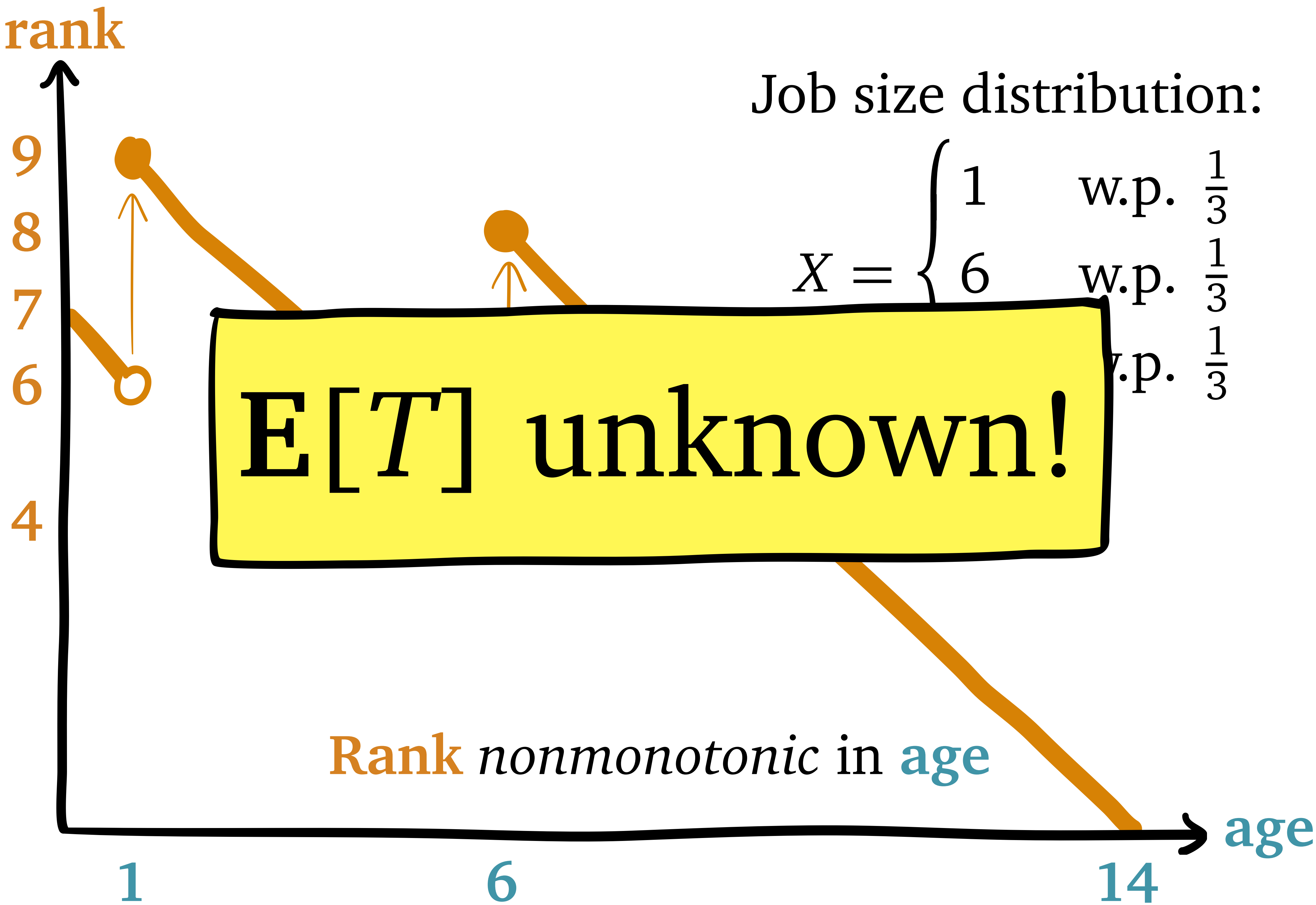



Job size distribution:

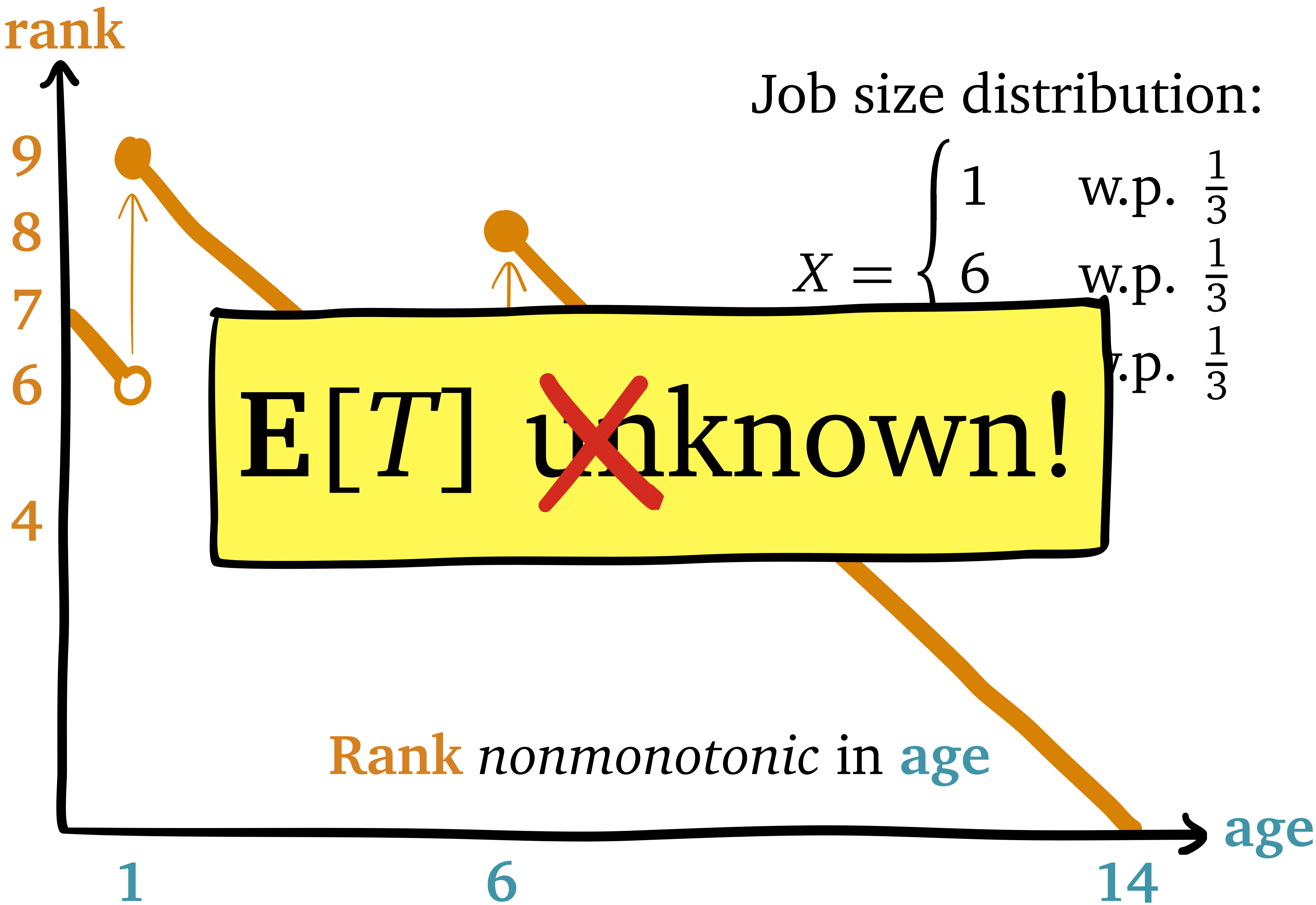
$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

**Rank** *nonmonotonic* in **age**

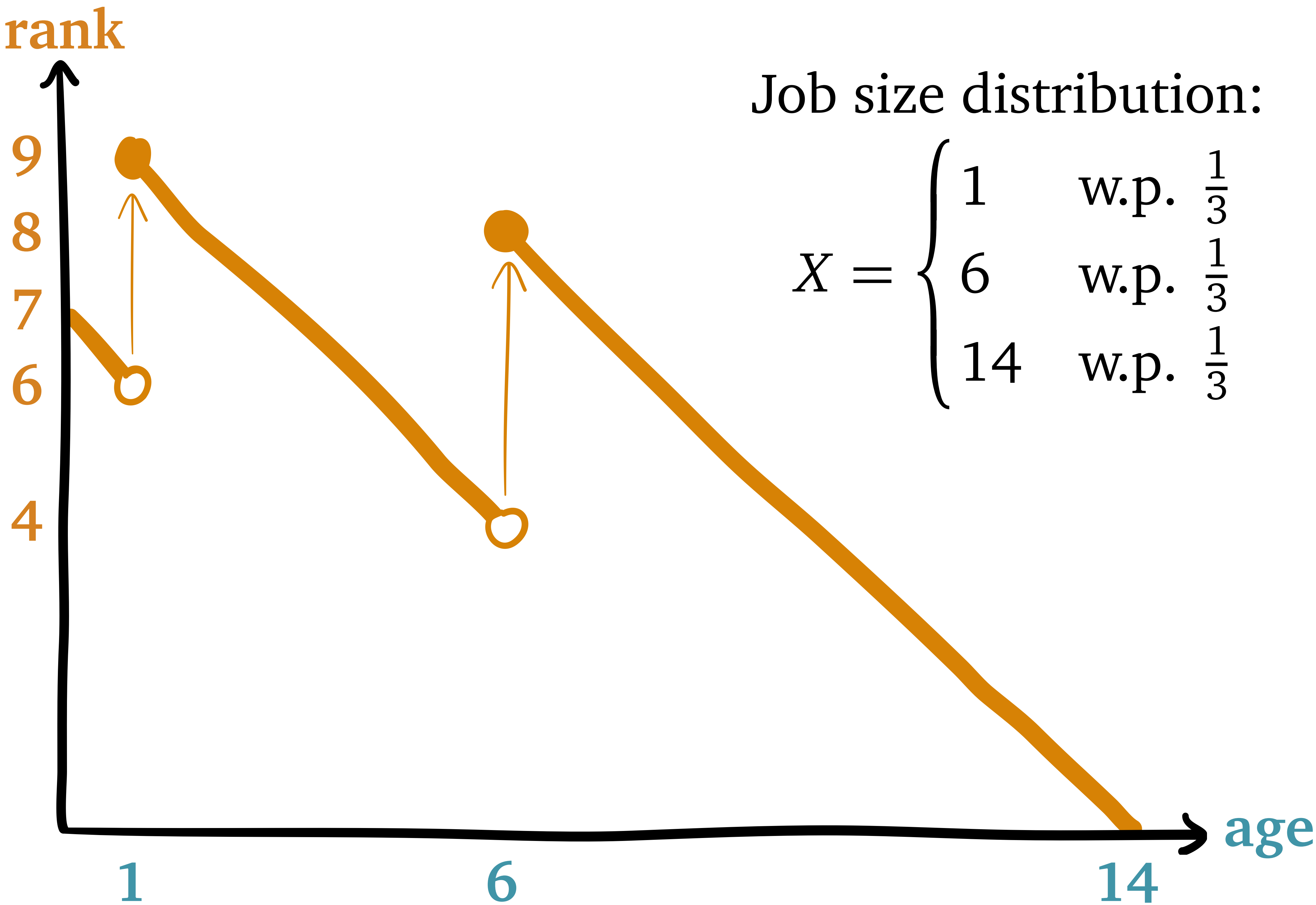
# SOAP Policy: SERPT



# SOAP Policy: SERPT



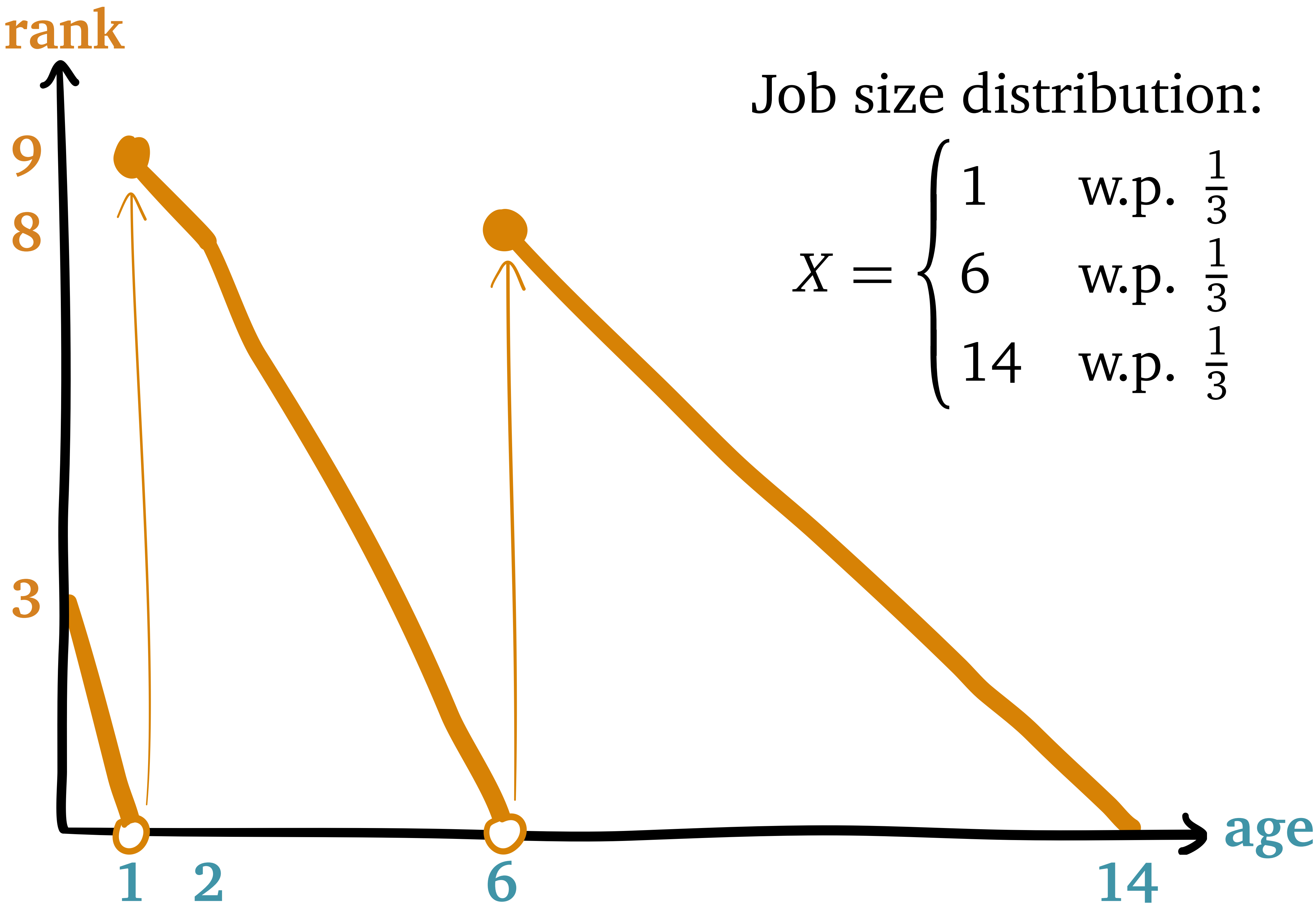
# SOAP Policy: SERPT




Job size distribution:

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

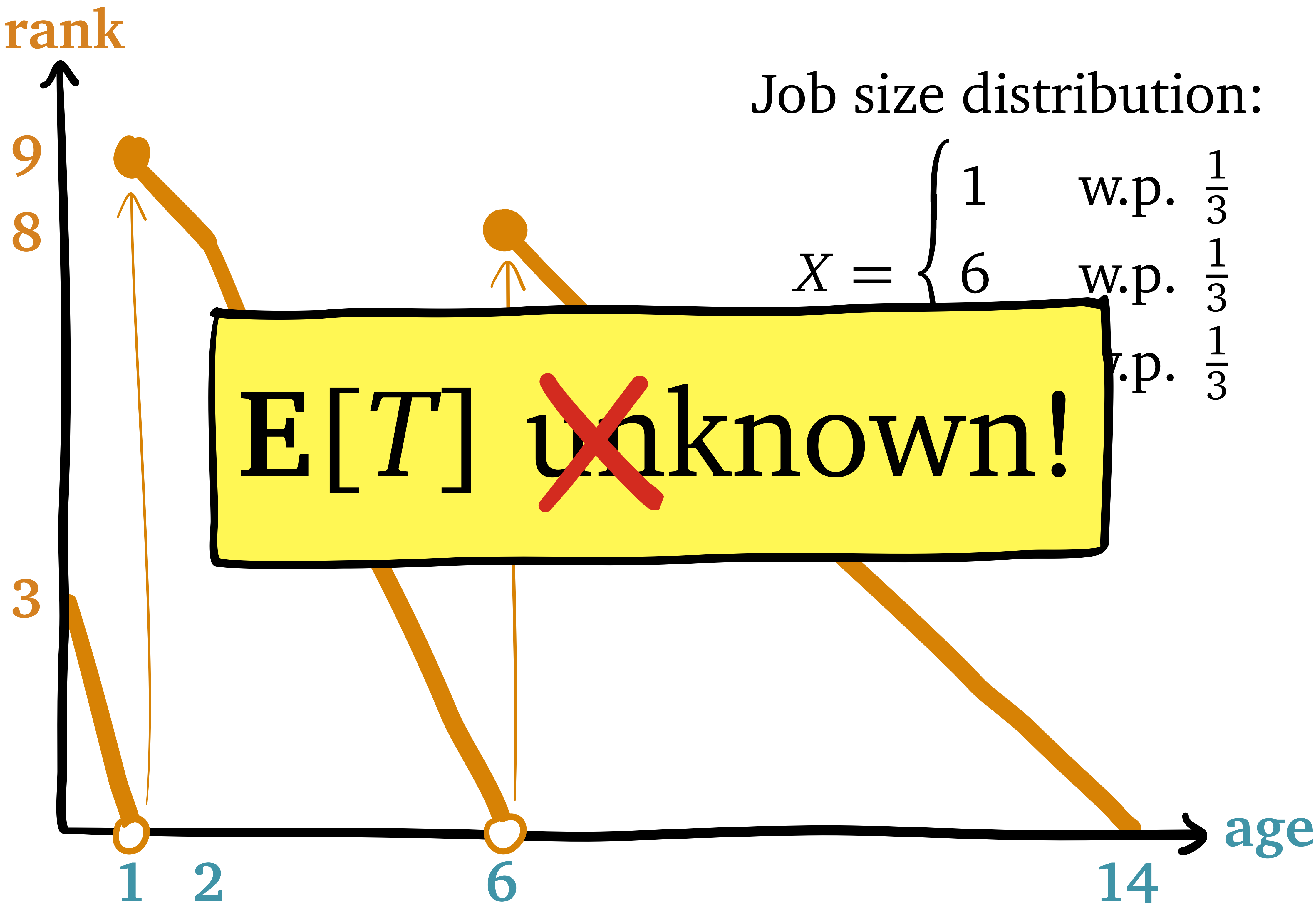
# SOAP Policy: Gittins



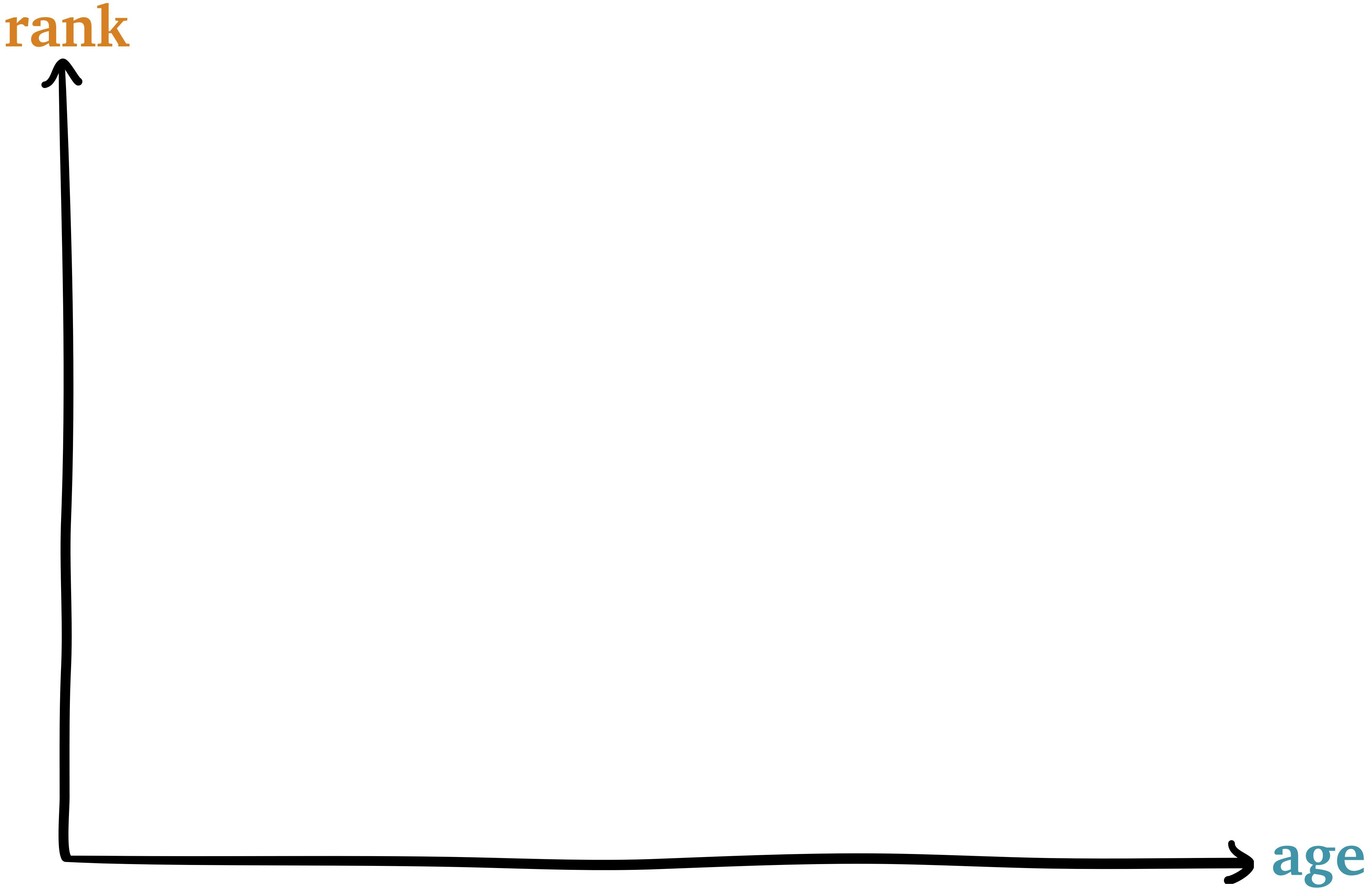

Job size distribution:

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# SOAP Policy: Gittins

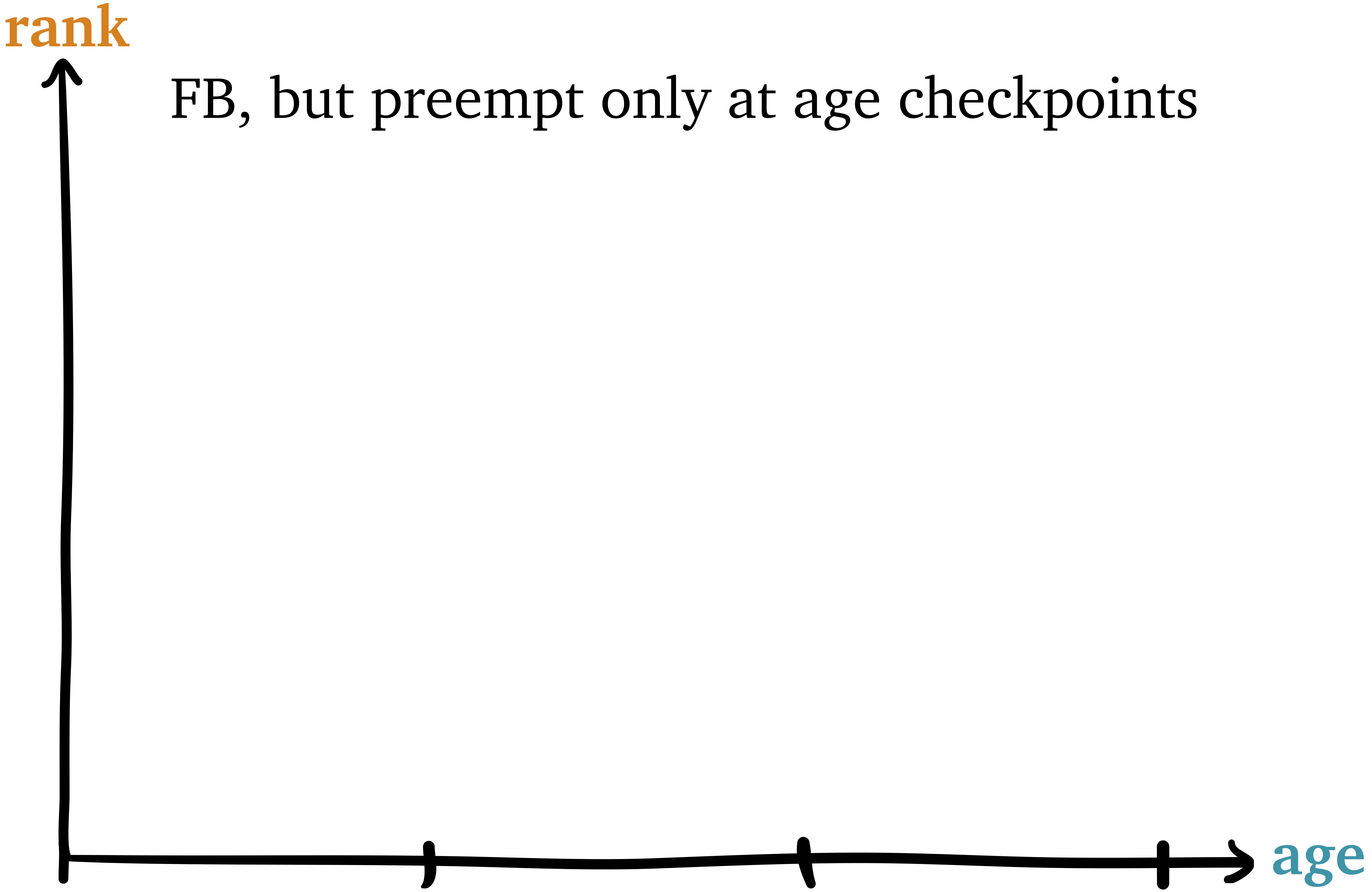


# **SOAP** Policy: Discrete FB

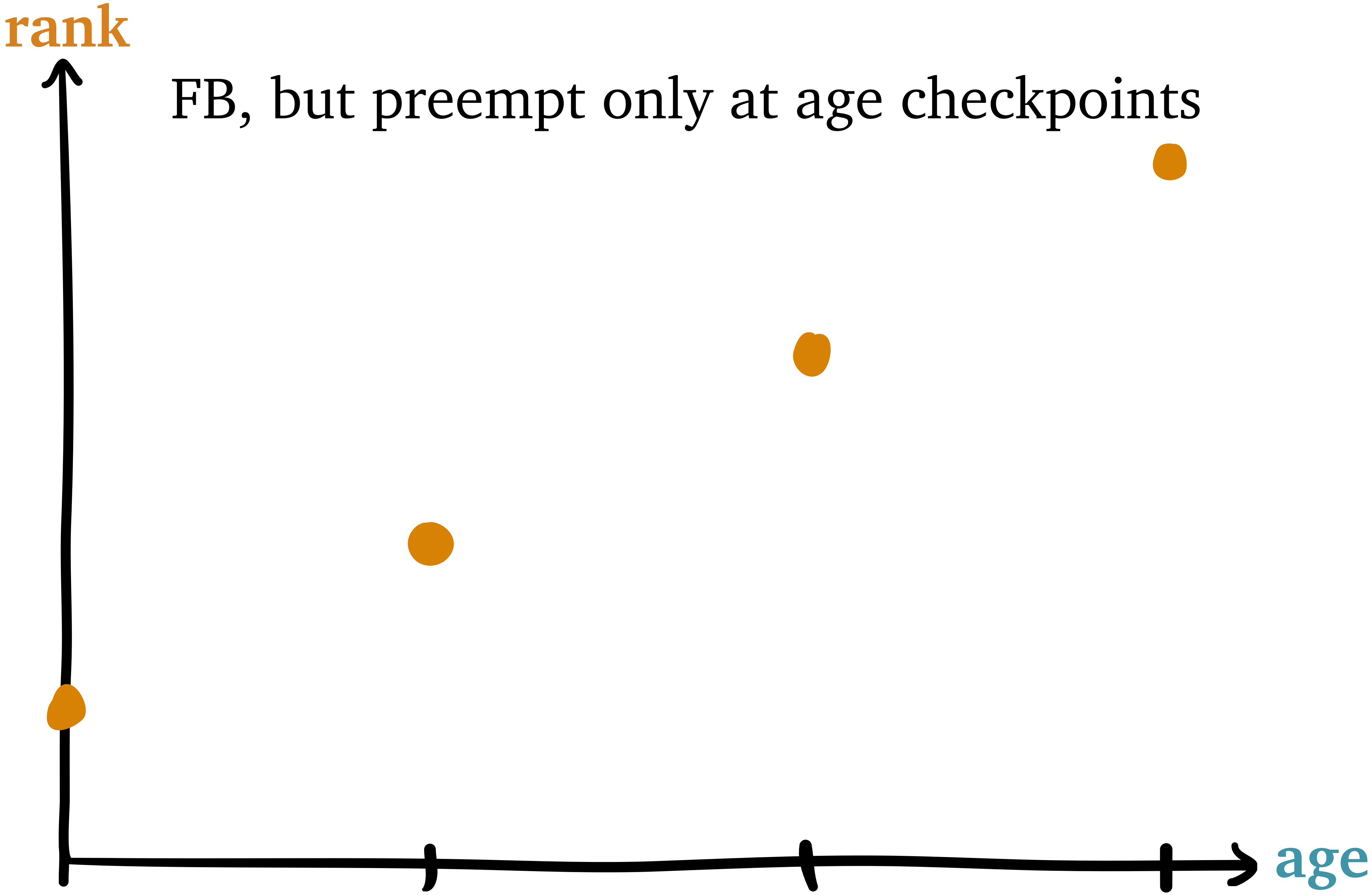


# SOAP Policy: Discrete FB

FB, but preempt only at age checkpoints

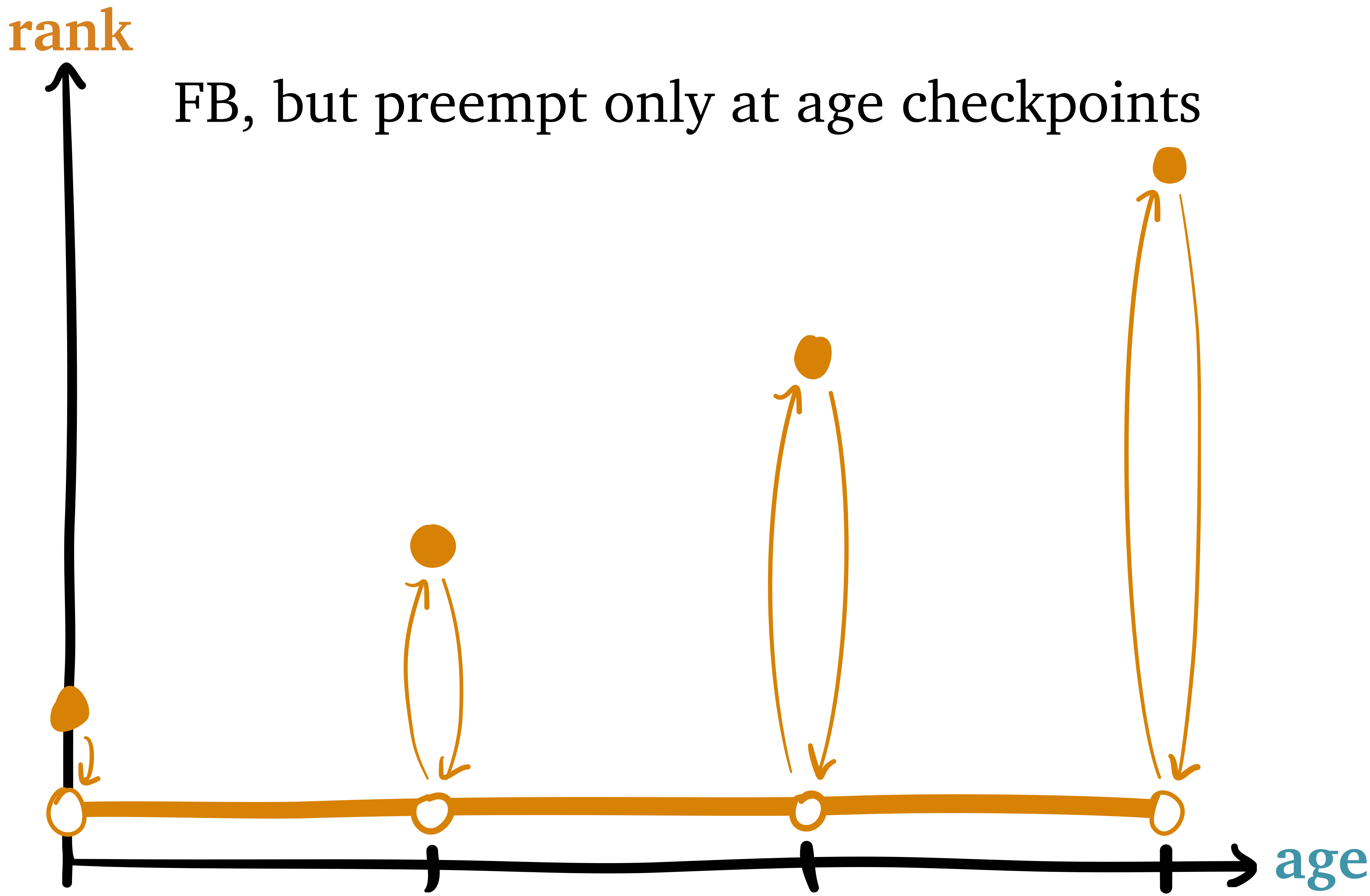


# SOAP Policy: Discrete FB

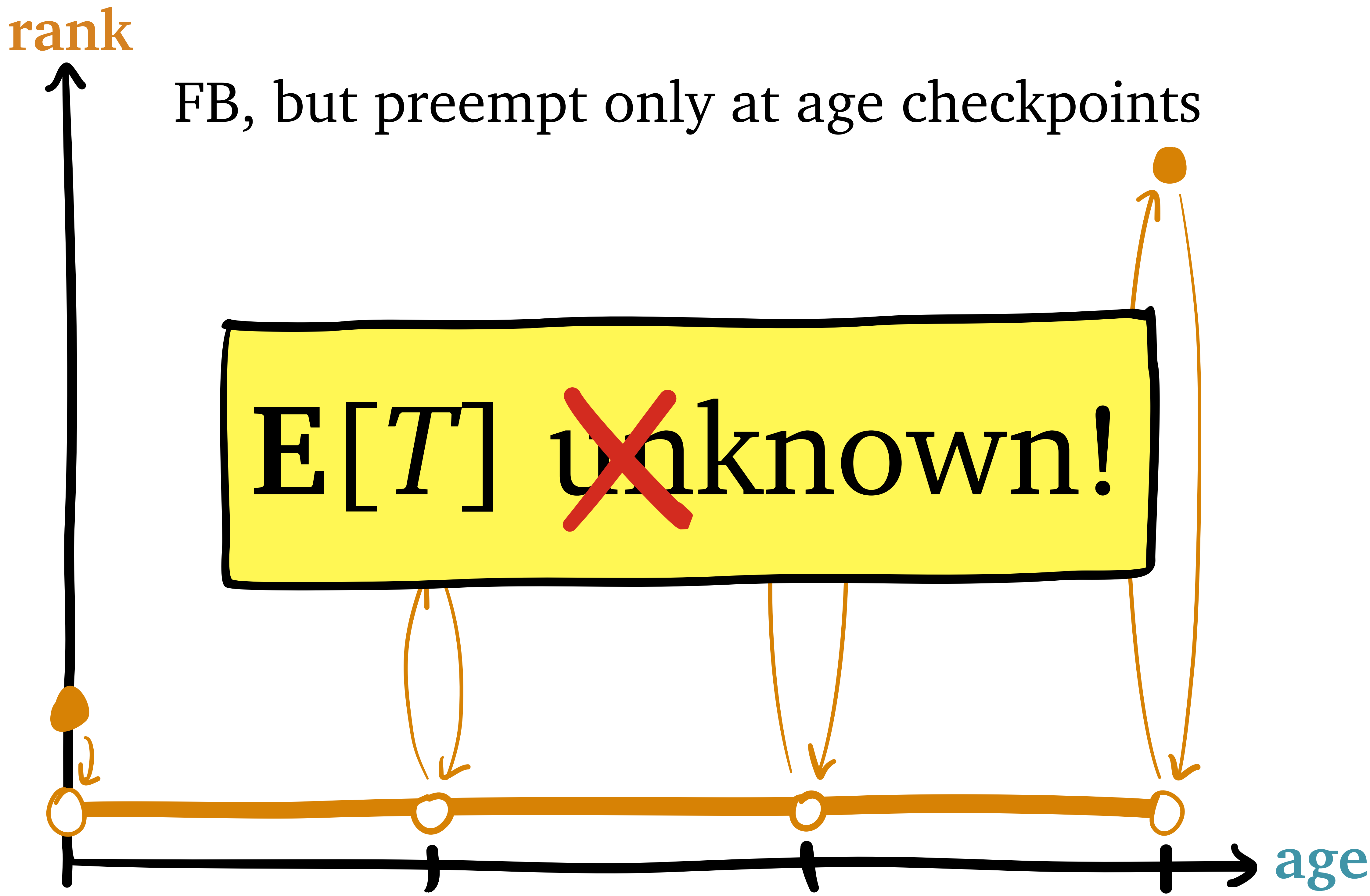


# SOAP Policy: Discrete FB

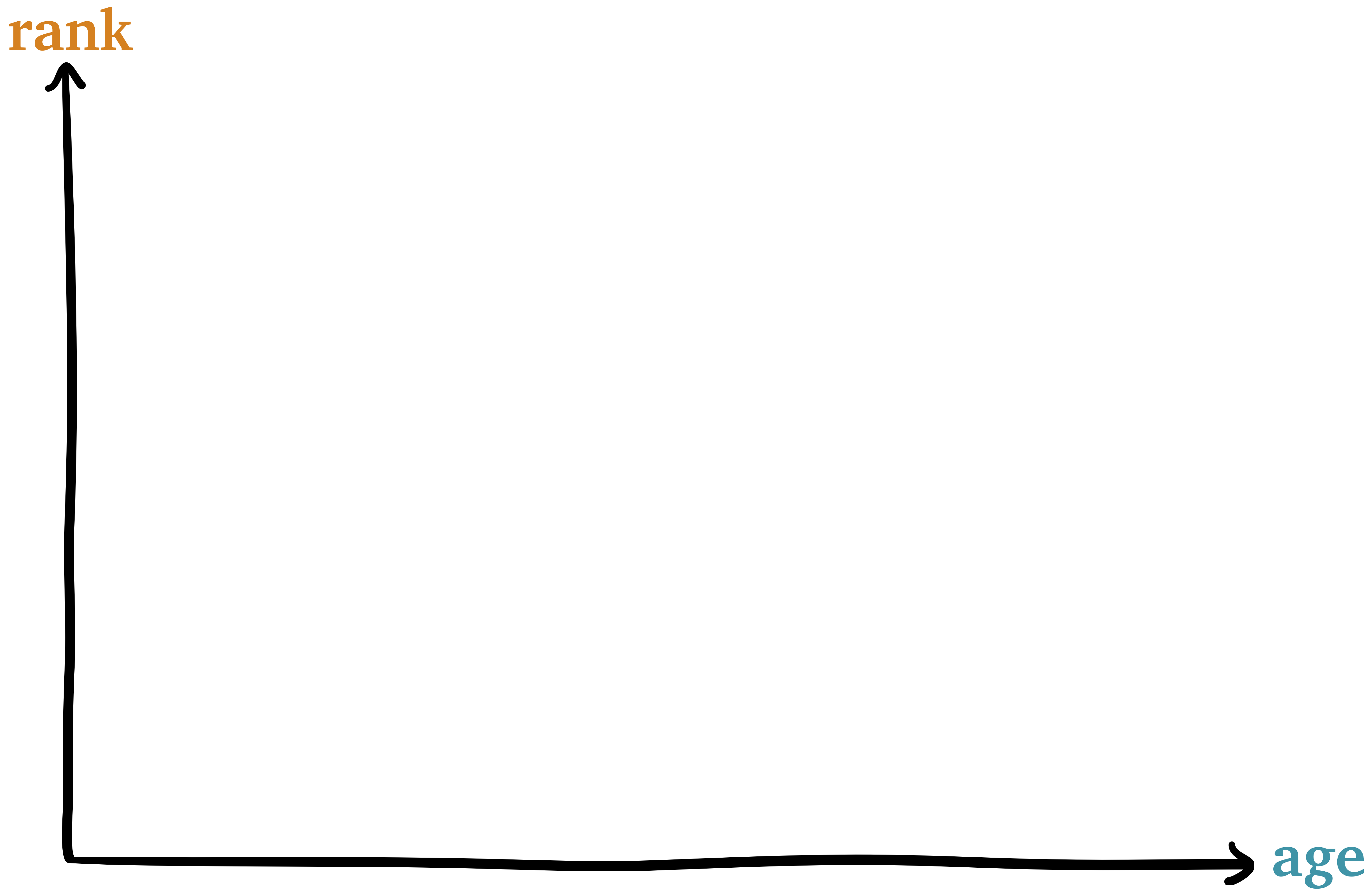
FB, but preempt only at age checkpoints



# SOAP Policy: Discrete FB

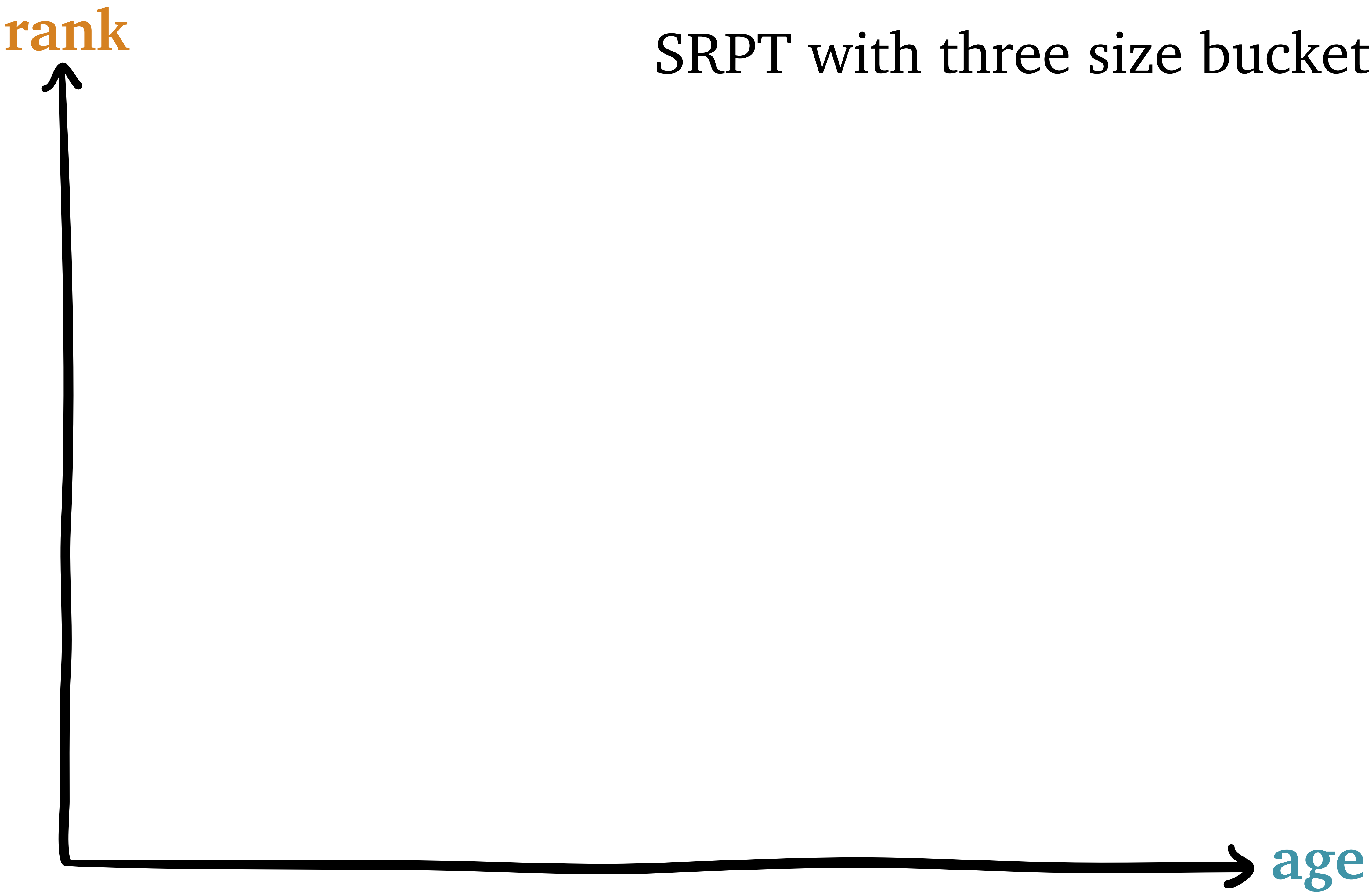


# SOAP Policy: Bucketed SRPT



# SOAP Policy: Bucketed SRPT

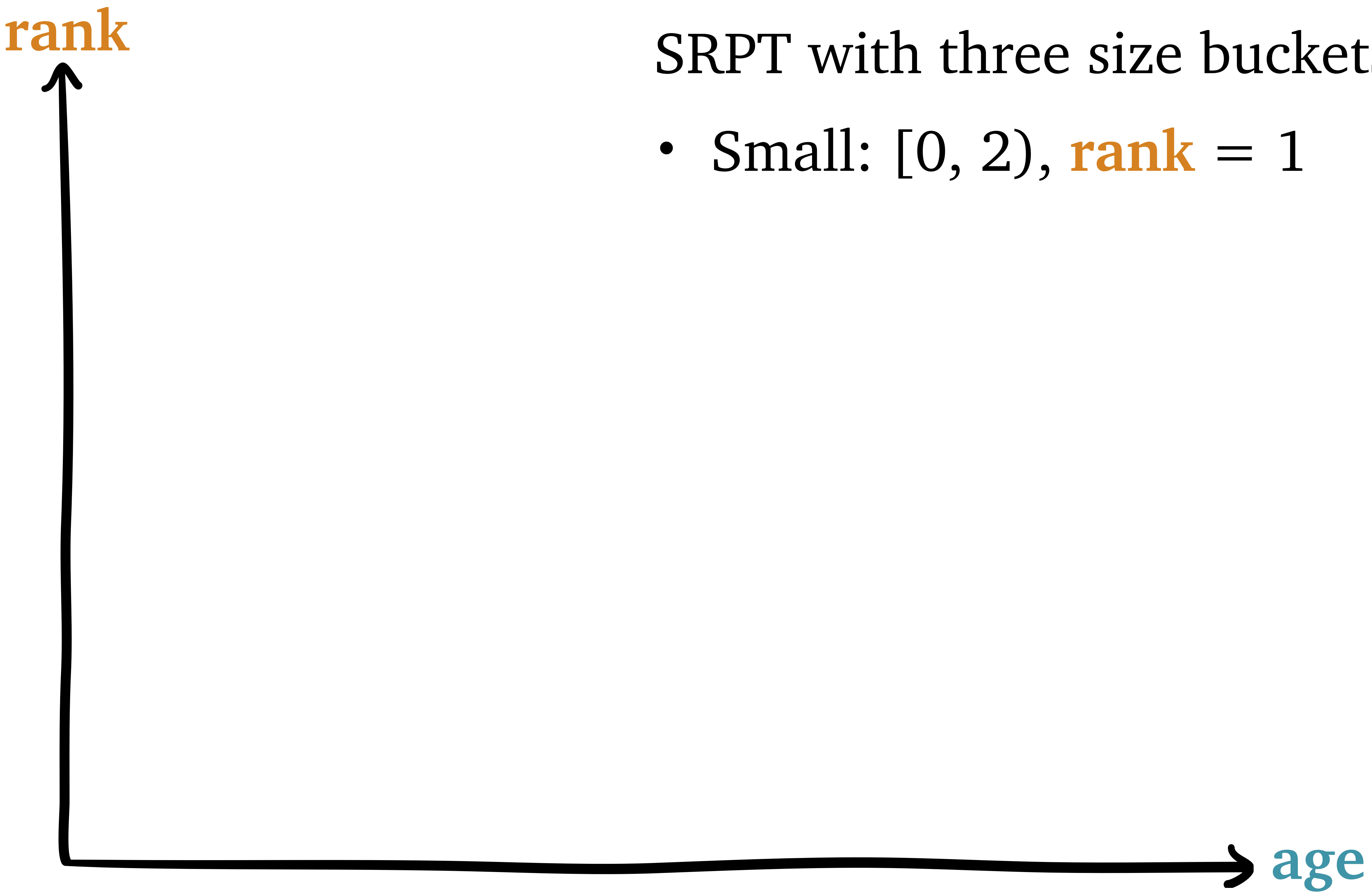
SRPT with three size buckets:



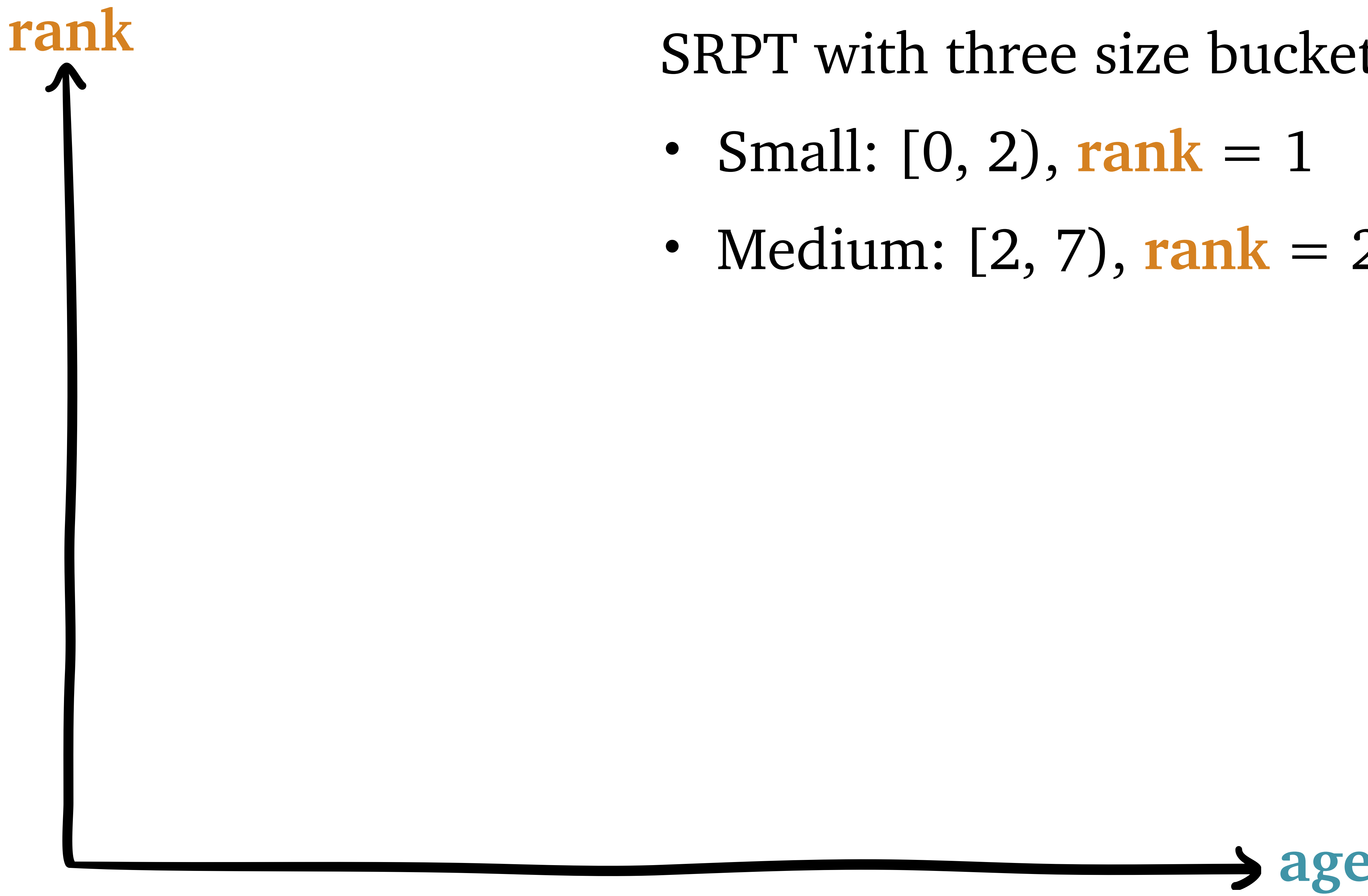
# SOAP Policy: Bucketed SRPT

SRPT with three size buckets:

- Small: $[0, 2)$, **rank** $= 1$



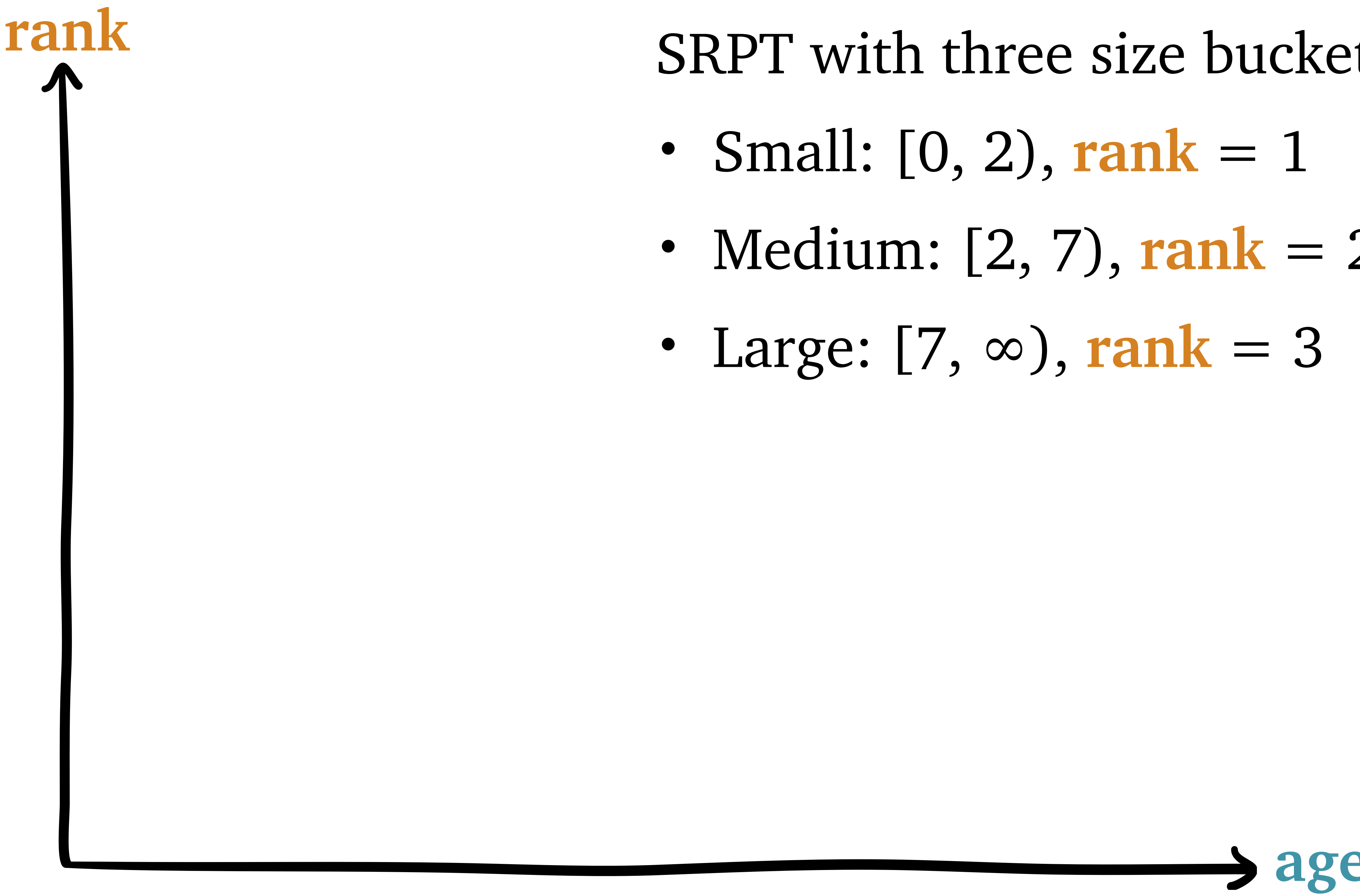
# SOAP Policy: Bucketed SRPT




SRPT with three size buckets:

- Small: $[0, 2)$, rank $= 1$
- Medium: $[2, 7)$, rank $= 2$

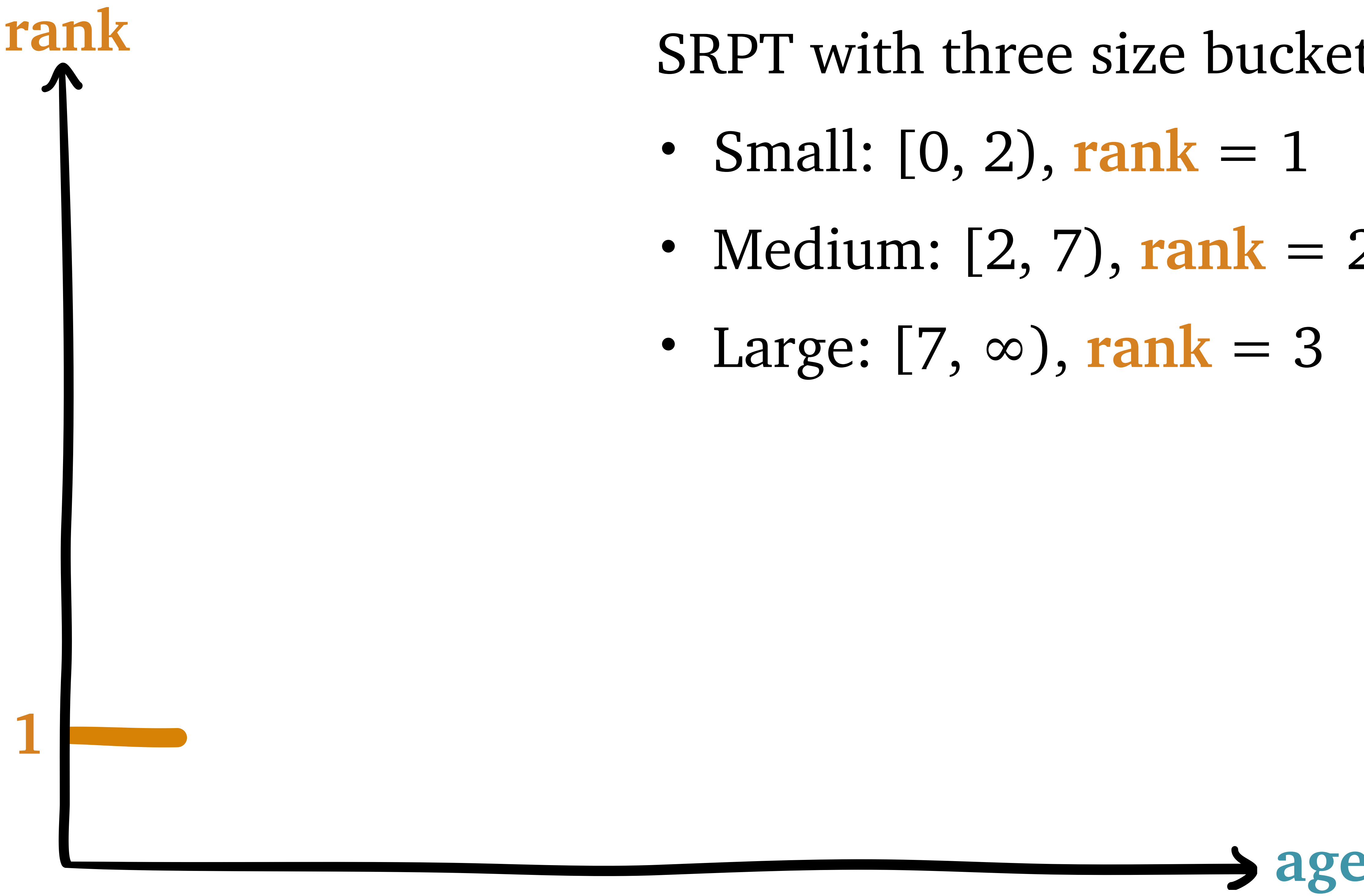
# SOAP Policy: Bucketed SRPT




SRPT with three size buckets:

- Small: $[0, 2)$, **rank** $= 1$
- Medium: $[2, 7)$, **rank** $= 2$
- Large: $[7, \infty)$, **rank** $= 3$

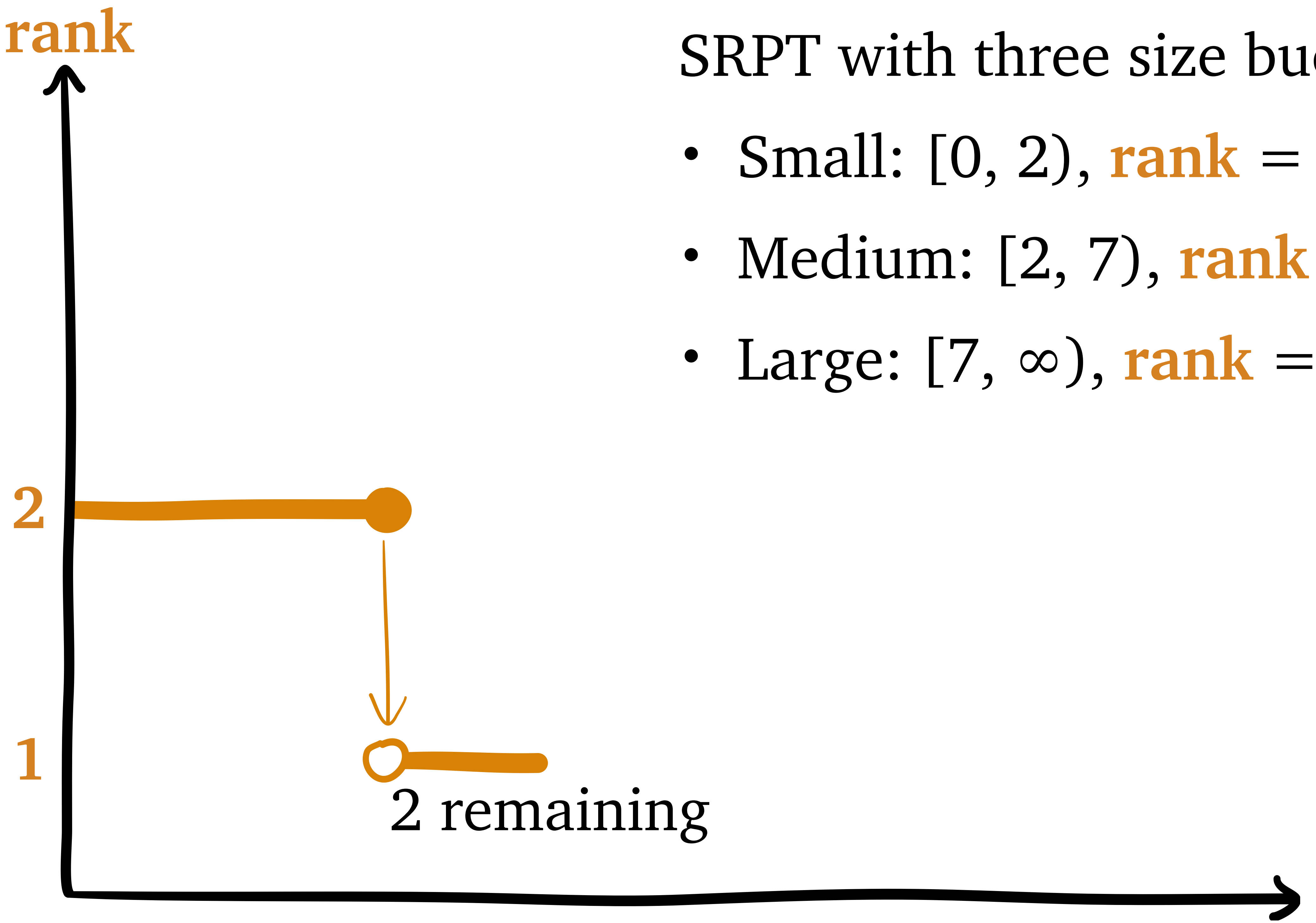
# SOAP Policy: Bucketed SRPT




SRPT with three size buckets:

- Small: $[0, 2)$, **rank** $= 1$
- Medium: $[2, 7)$, **rank** $= 2$
- Large: $[7, \infty)$, **rank** $= 3$

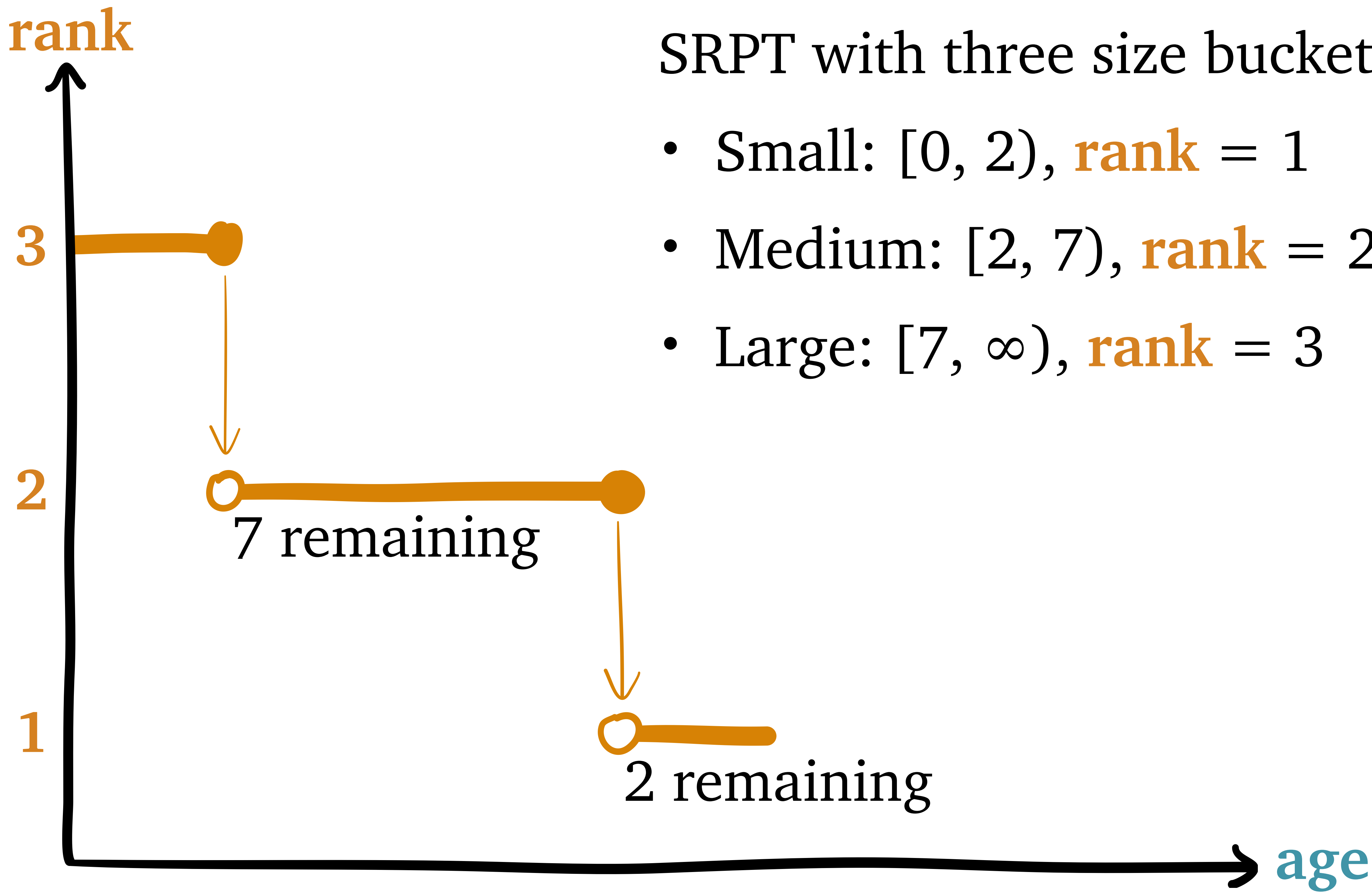
# SOAP Policy: Bucketed SRPT




SRPT with three size buckets:

- Small: [0, 2), rank = 1
- Medium: [2, 7), rank = 2
- Large: [7, ∞), rank = 3

# SOAP Policy: Bucketed SRPT



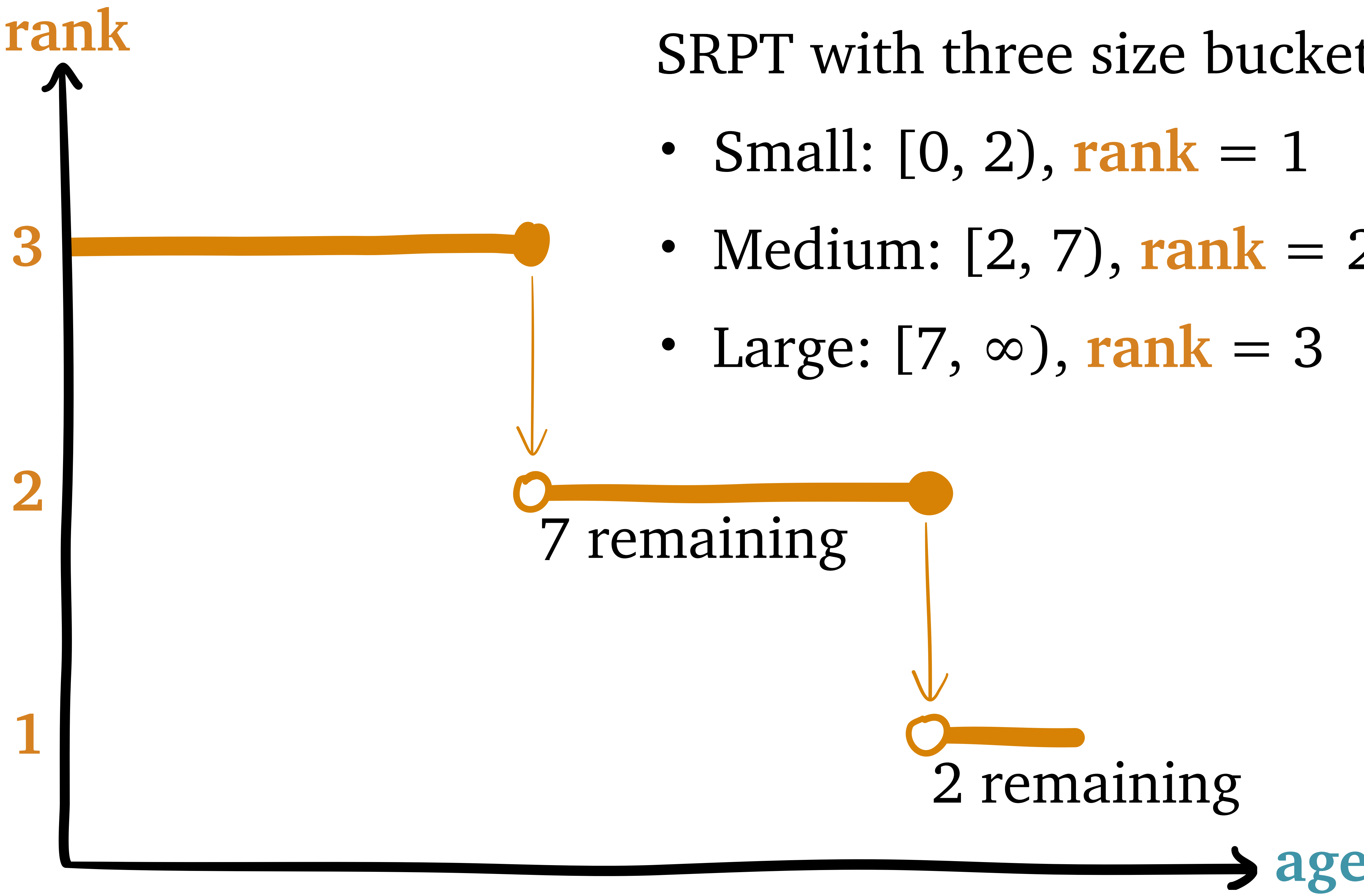

SRPT with three size buckets:

- Small: [0, 2), **rank** = 1
- Medium: [2, 7), **rank** = 2
- Large: [7, ∞), **rank** = 3

# SOAP Policy: Bucketed SRPT

SRPT with three size buckets:

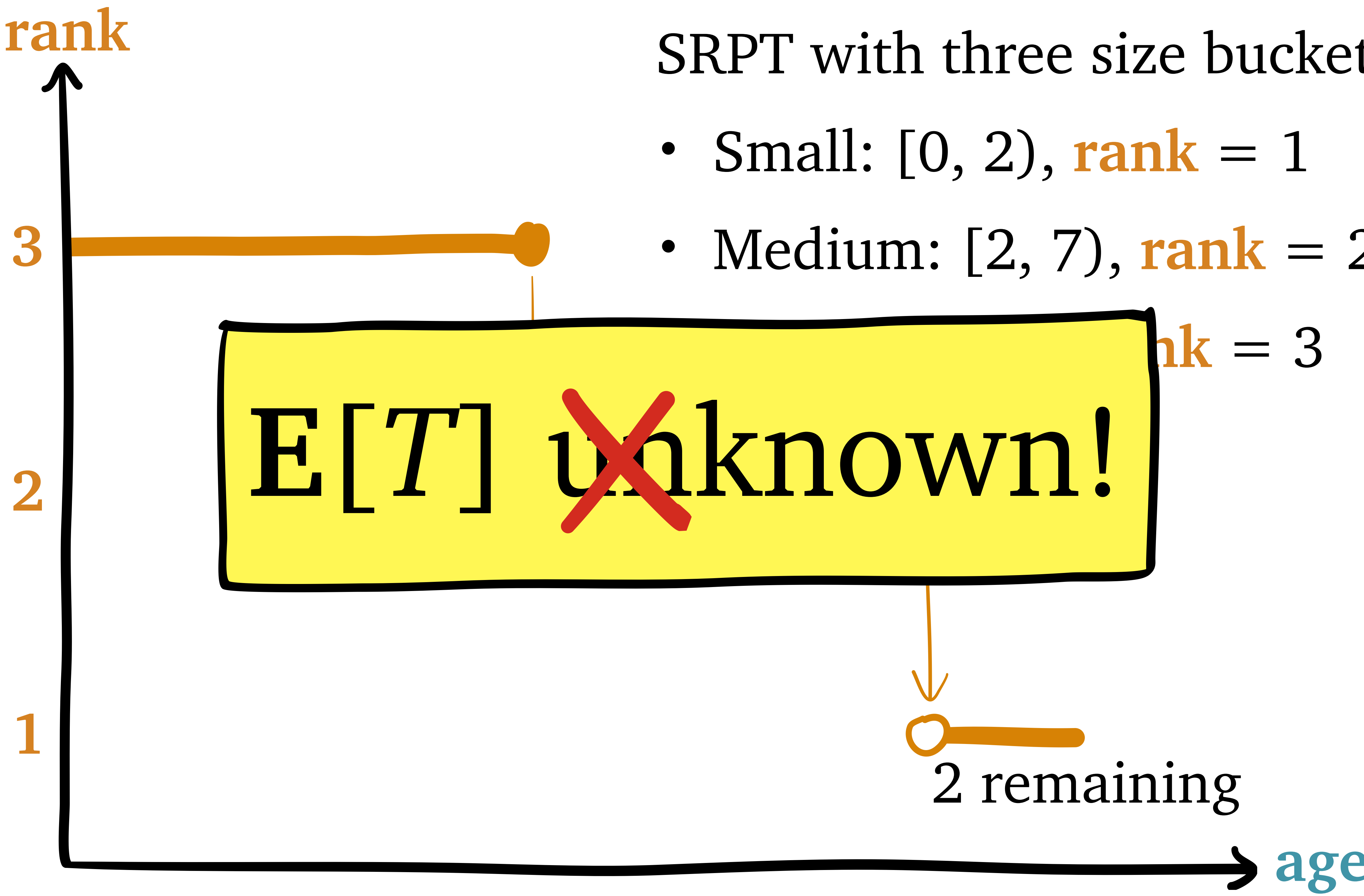
- Small: [0, 2), **rank** = 1
- Medium: [2, 7), **rank** = 2
- Large: [7, ∞), **rank** = 3



# **SOAP** Policy: Bucketed SRPT

SRPT with three size buckets:

- Small: [0, 2), **rank** = 1
- Medium: [2, 7), **rank** = 2
- …nk = 3

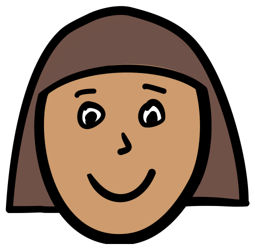


# SOAP Policy: Mixture

Two customer classes: *humans* and *robots*

# SOAP Policy: Mixture

Two customer classes: *humans* and *robots*

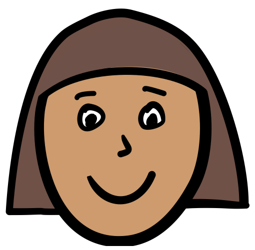


**Humans**

- unknown size
- nonpreemptible
- FCFS

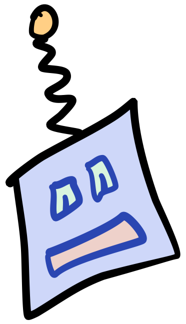
# SOAP Policy: Mixture

Two customer classes: *humans* and *robots*



**Humans**

- unknown size
- nonpreemptible
- FCFS

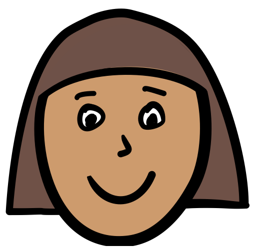


**Robots**

- known size
- preemptible
- SRPT

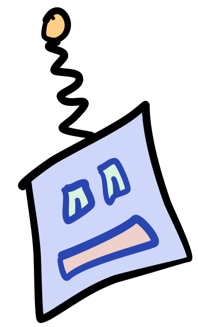
# SOAP Policy: Mixture

Two customer classes: *humans* and *robots*



**Humans**

- unknown size
- nonpreemptible
- FCFS
- priority over robots

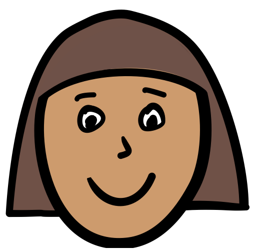


**Robots**

- known size
- preemptible
- SRPT

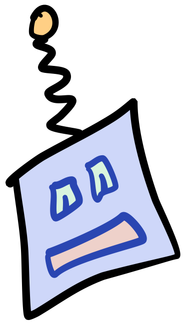
# **SOAP** Policy: Mixture

Two customer classes: *humans* and *robots*



**Humans**

- unknown size
- nonpreemptible
- FCFS
- priority over robots



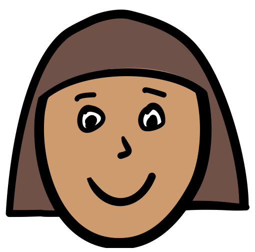
**Robots**

- known size
- preemptible
- SRPT

**Twist**: small robots outrank humans

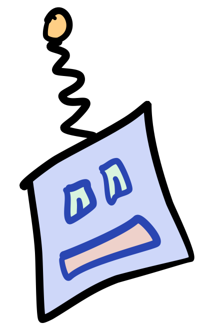
# SOAP Policy: Mixture

Two customer classes: *humans* and *robots*



**Humans**

- unknown size
- nonpreemptible
- FCFS
- priority over robots



**Robots**

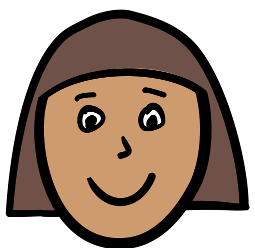
- known size
- preemptible
- SRPT

size $< x_{\mathrm{H}}$

**Twist:** small robots outrank humans

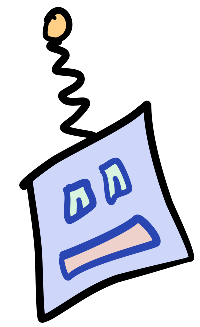
# SOAP Policy: Mixture

Two customer classes: *humans* and *robots*



**Humans**

- unknown size
- nonpreemptible
- FCFS
- priority over robots



**Robots**

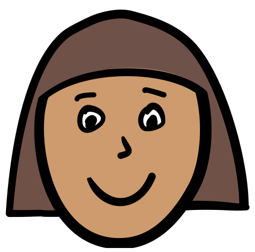
- known size
- preemptible
- SRPT

size $< x_{\mathrm{H}}$

**Twist:** small robots outrank humans

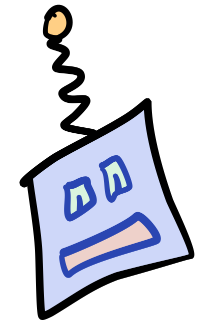
# SOAP Policy: Mixture

Two customer classes: *humans* and *robots*



**Humans**

- unknown size
-
-
-



**Robots**

- known size
- …mptible
- …T

**E[$T$] ~~unknown~~ known!**

size $< x_{\mathrm{H}}$

**Twist:** small robots outrank humans

# Full **SOAP** Definition

A ***SOAP*** *policy* is any policy expressible by a ***rank*** *function* of the form:

# Full **SOAP** Definition

A ***SOAP*** *policy* is any policy expressible by a ***rank*** *function* of the form:

$$\textbf{descriptor} \times \textbf{age} \rightarrow \textbf{rank}$$

# Full SOAP Definition

A ***SOAP*** *policy* is any policy expressible by a ***rank*** *function* of the form:

size, class, etc.

**descriptor** × **age** → **rank**

# Full SOAP Definition

A ***SOAP*** *policy* is any policy expressible by a ***rank*** *function* of the form:

size, class, etc.

$$\textbf{descriptor} \times \textbf{age} \rightarrow \textbf{rank}$$

**FB**

$$r_{\varnothing}(a) = a$$

# Full SOAP Definition

A ***SOAP*** *policy* is any policy expressible by a ***rank*** *function* of the form:

size, class, etc.

$$\textbf{descriptor} \times \textbf{age} \rightarrow \textbf{rank}$$

**FB**

$$r_{\varnothing}(a) = a$$

**SRPT**

$$r_x(a) = x - a$$

# Full SOAP Definition

A ***SOAP*** *policy* is any policy expressible by a ***rank*** *function* of the form:

size, class, etc.

$$\textbf{descriptor} \times \textbf{age} \rightarrow \textbf{rank}$$

**FB**

$r_{\varnothing}(a) = a$

**SRPT**

$r_x(a) = x - a$

**Descriptor** can be anything that:

- does not change while a job is in the system
- is i.i.d. for each job

# FAQ:
## What *isn't* a **SOAP** policy?

# FAQ:
## What *isn't* a **SOAP** policy?

- **Rank** changes when not in service

# FAQ:
## What *isn't* a **SOAP** policy?

- **Rank** changes when not in service
- **Rank** depends on system-wide state

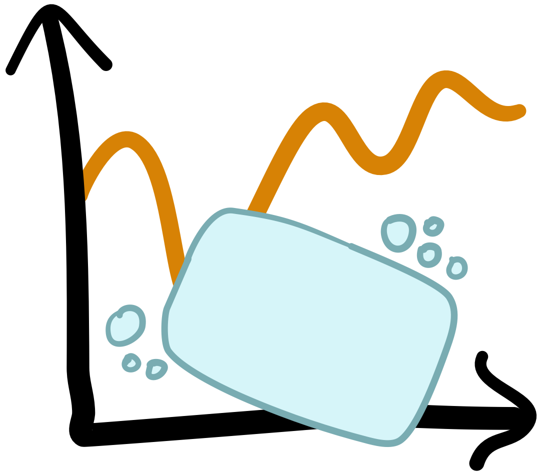
# FAQ:
## What *isn't* a **SOAP** policy?

- **Rank** changes when not in service
- **Rank** depends on system-wide state
- Non-FCFS tiebreaking

# **FAQ:**
## What *isn't* a **SOAP** policy?

- **Rank** changes when not in service
- **Rank** depends on system-wide state
- Non-FCFS tiebreaking

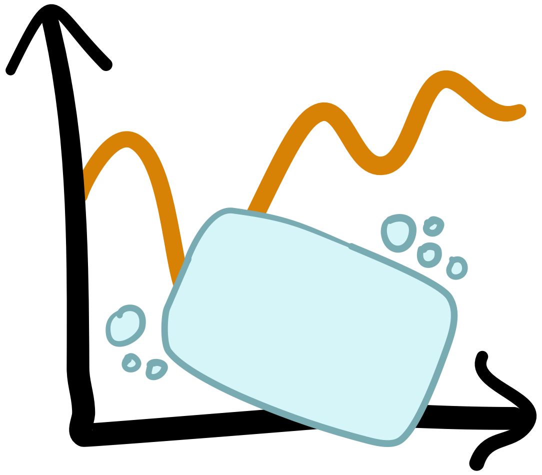
**Excludes**: EDF, accumulating priority, PS



# Part 1:

*defining* **SOAP** policies

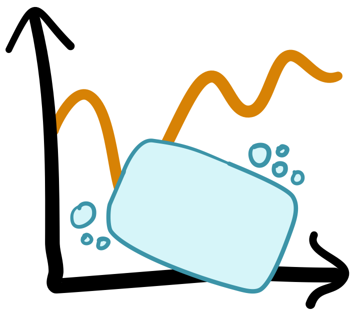
Practice!



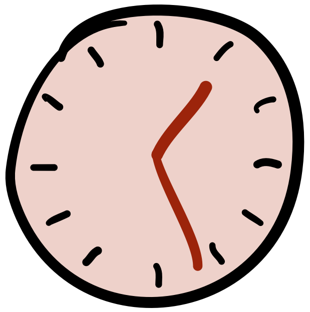
# Part 1:

*defining* **SOAP** policies

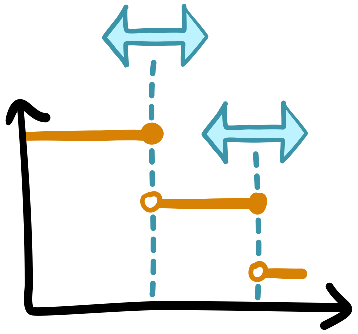
# Outline



**Part 1**: *defining* **SOAP** policies



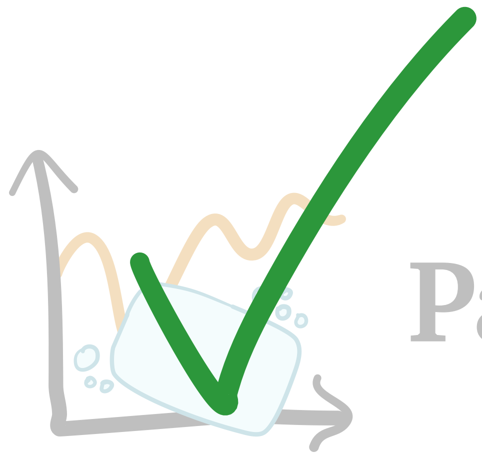
**Part 2**: *analyzing* **SOAP** policies



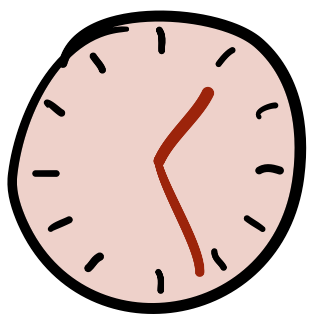
**Part 3**: *policy design* with **SOAP**

**Part 4**: *optimality proofs* with **SOAP**

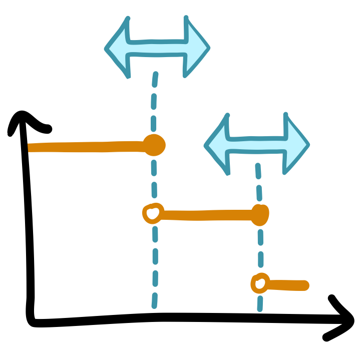
# Outline



Part 1: *defining* SOAP policies

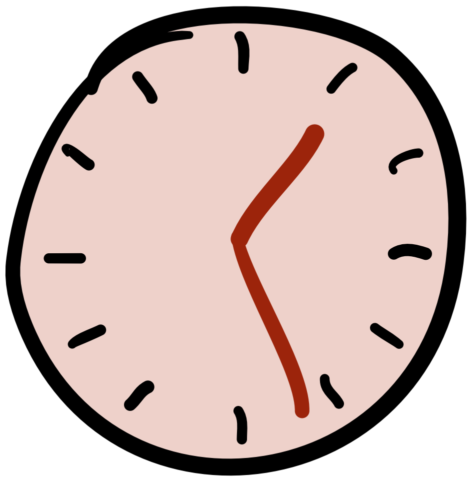


**Part 2**: *analyzing* **SOAP** policies



**Part 3**: *policy design* with **SOAP**

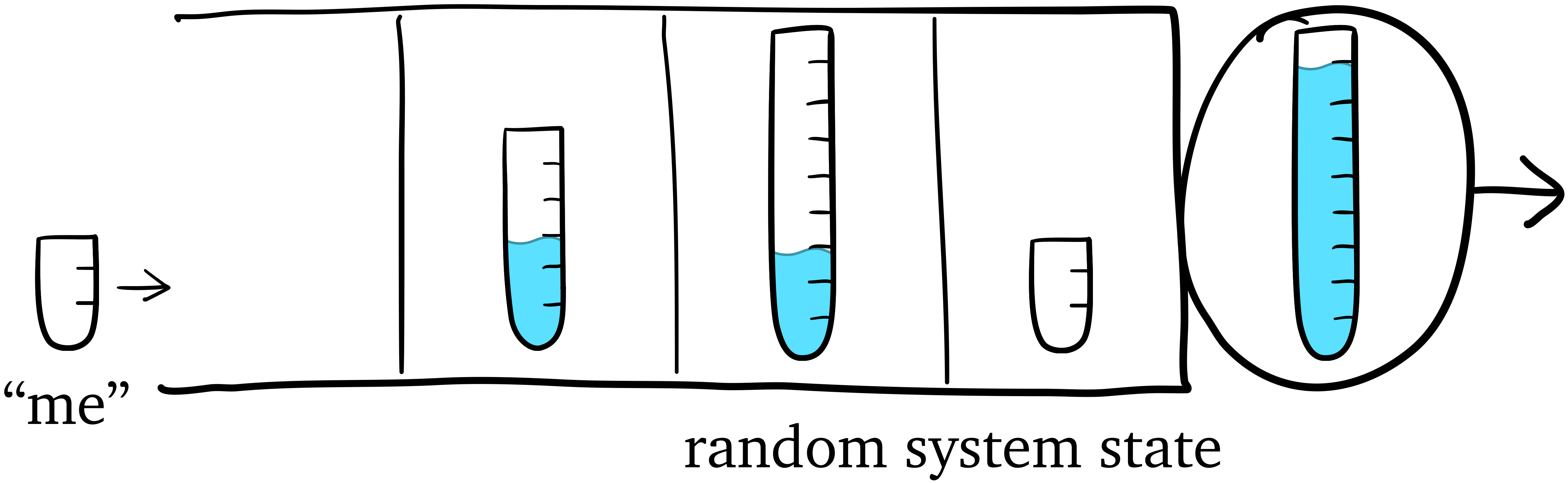
**Part 4**: *optimality proofs* with **SOAP**



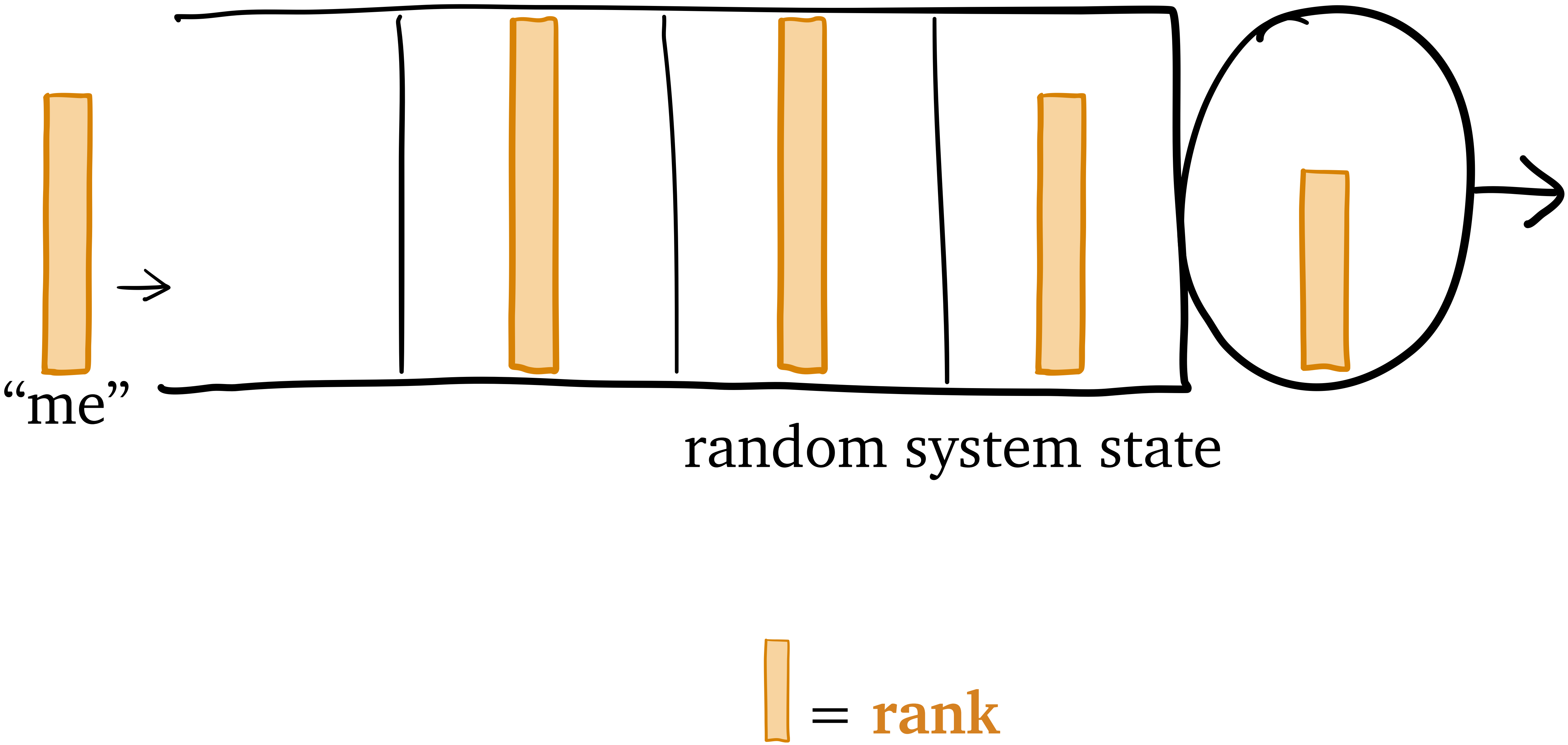
# Part 2:

*analyzing* **SOAP** policies

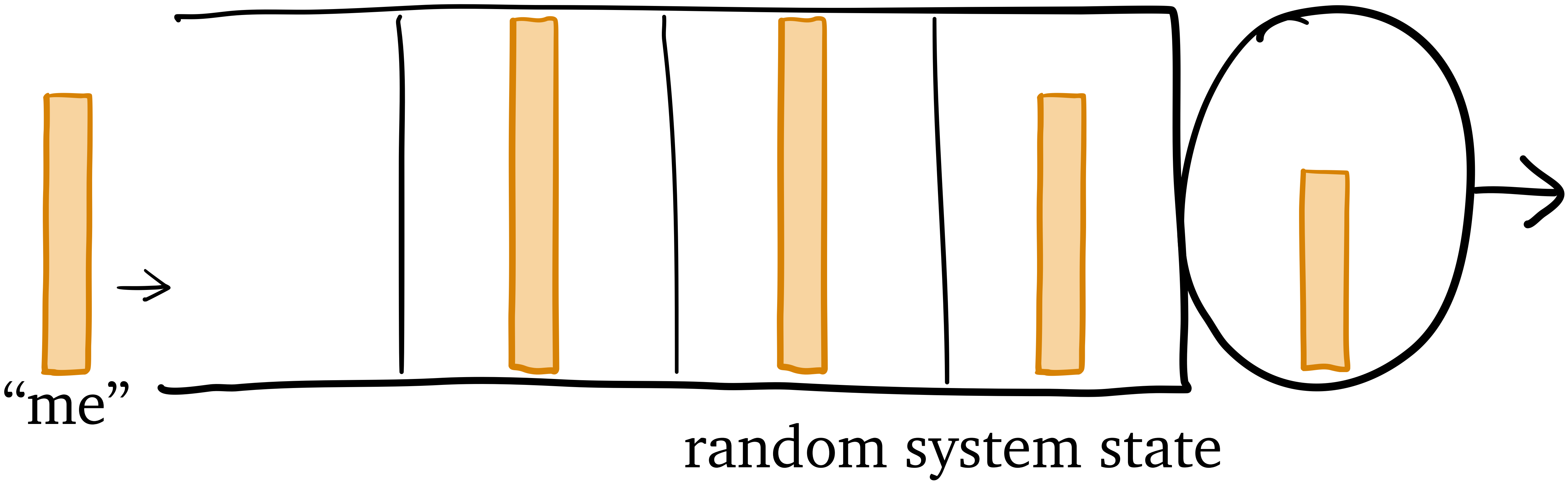
# Tagged Job Analysis

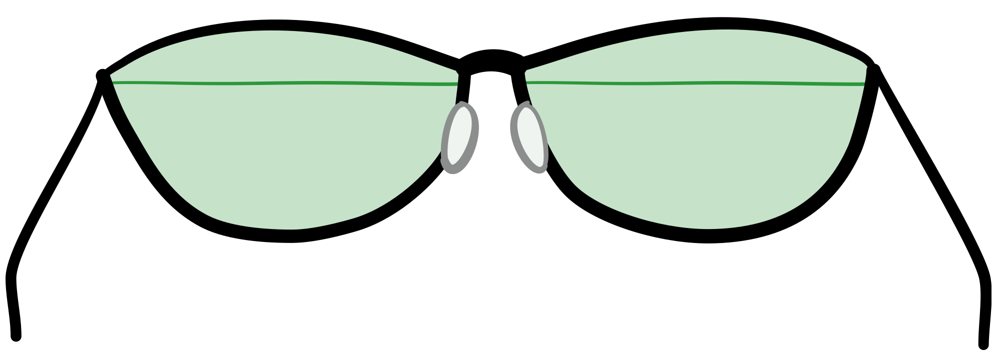


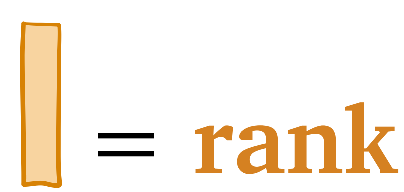
# Tagged Job Analysis



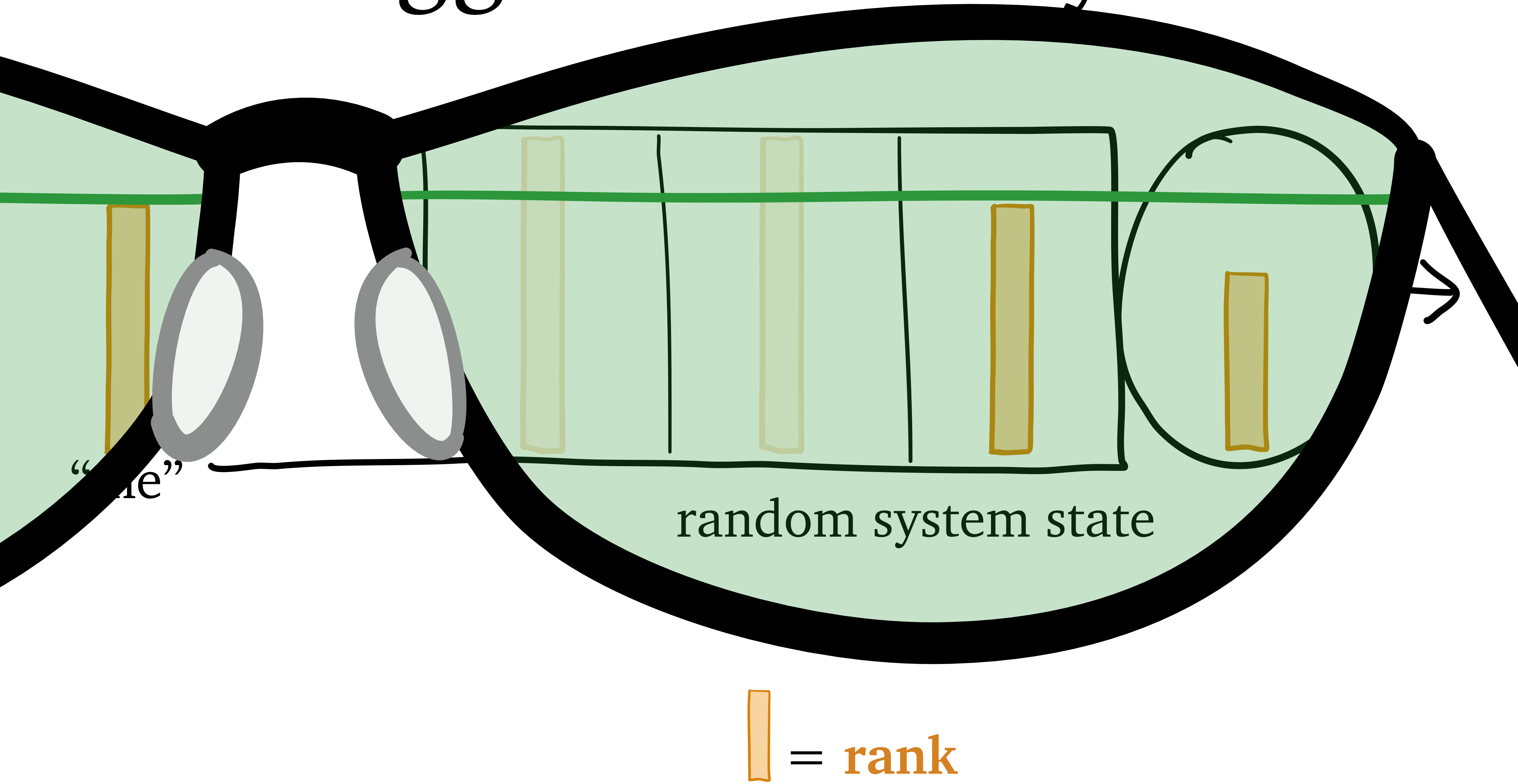
# Tagged Job Analysis



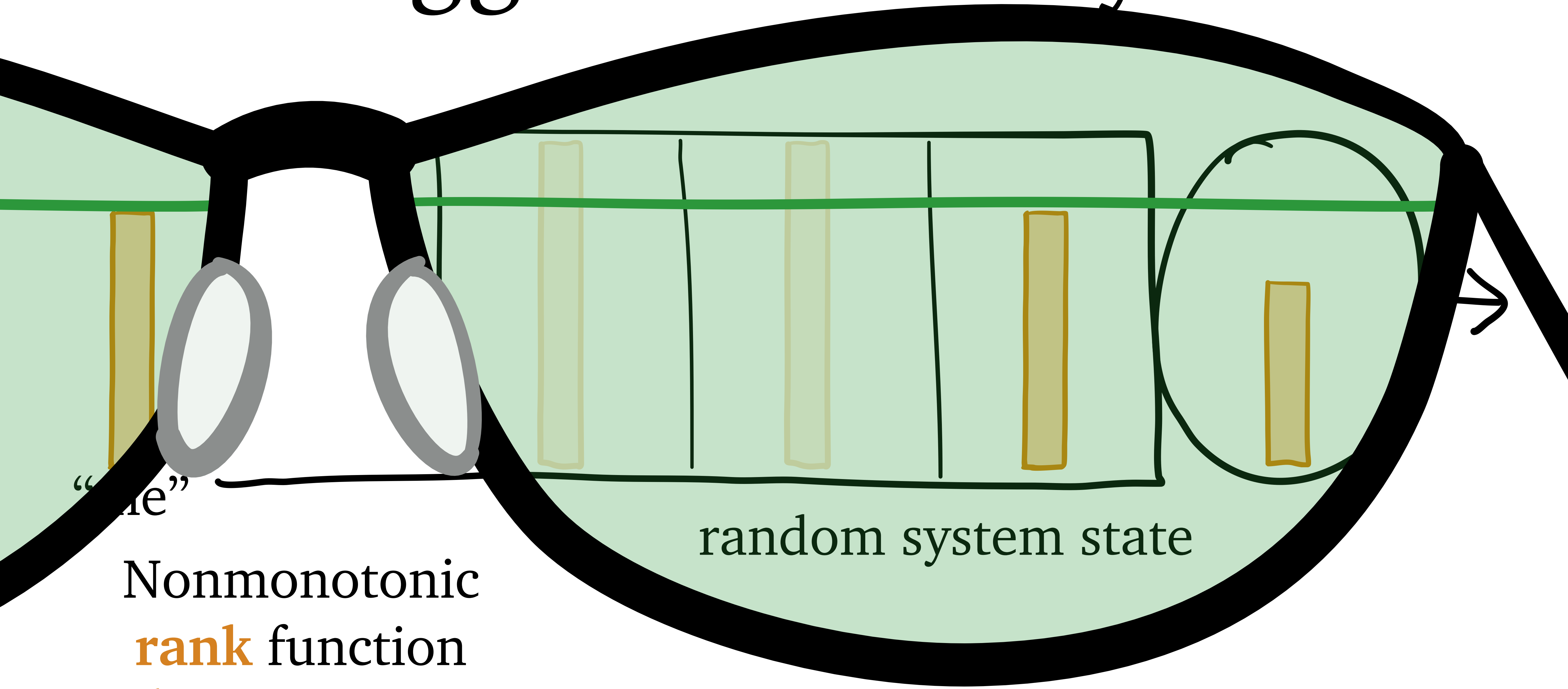


# Tagged Job Analysis

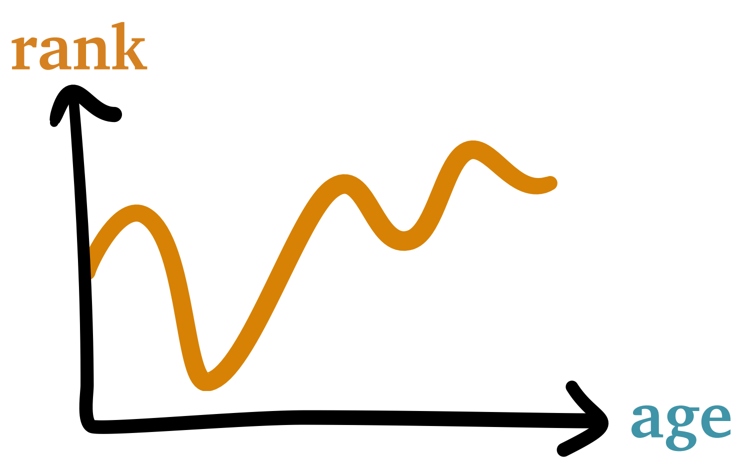



= **rank**

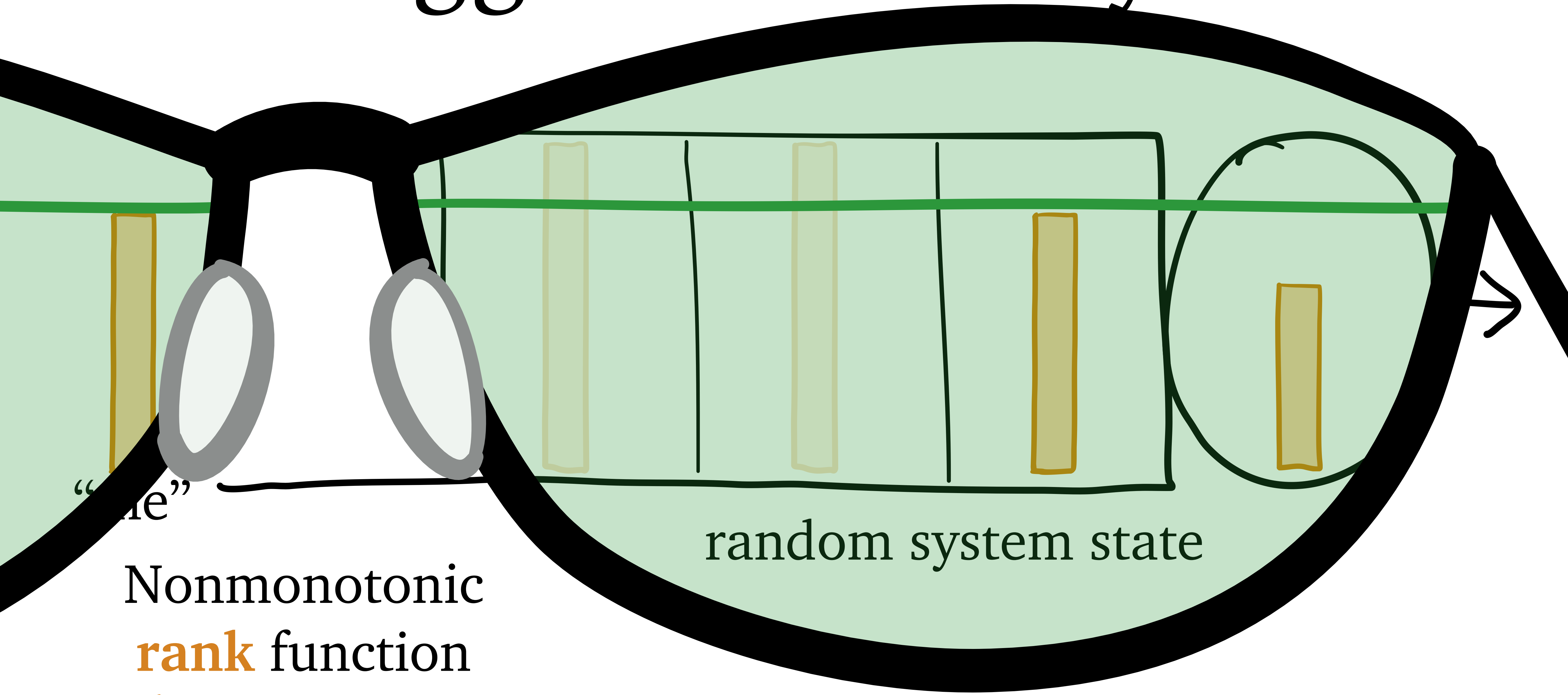
# Tagged Job Analysis



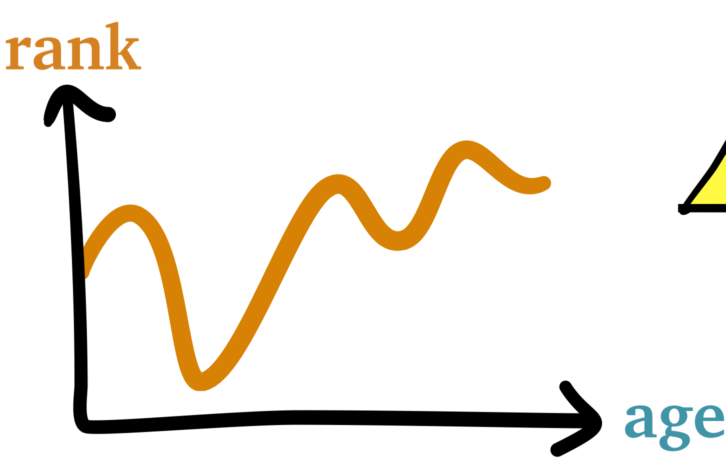

Nonmonotonic **rank** function



# Tagged Job Analysis

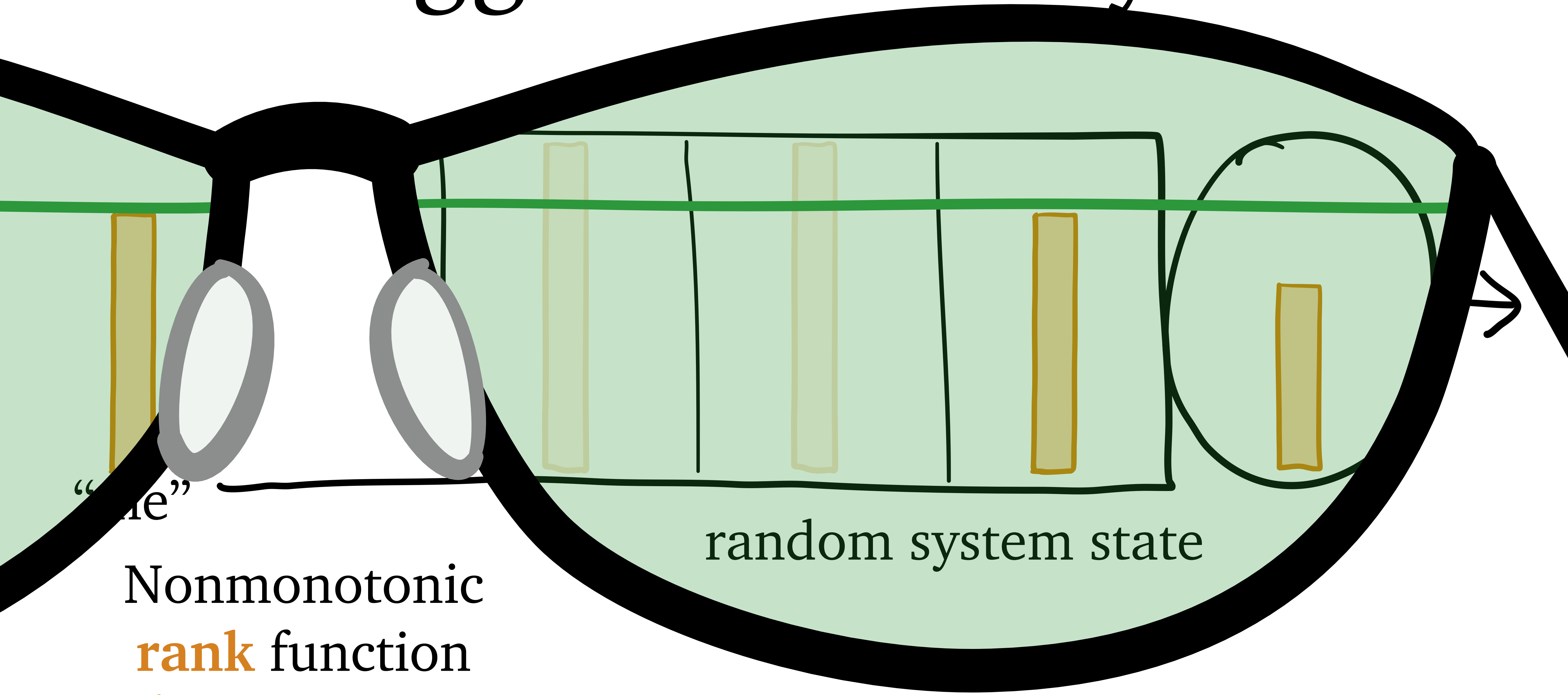



Nonmonotonic
**rank** function

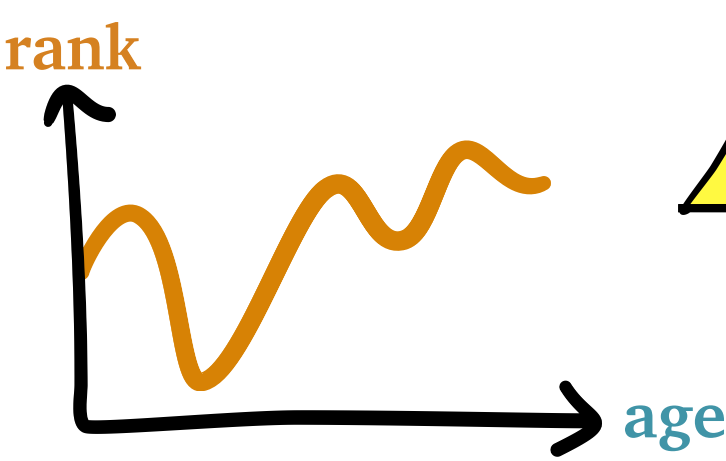



⚠ **Two obstacles:**

# Tagged Job Analysis



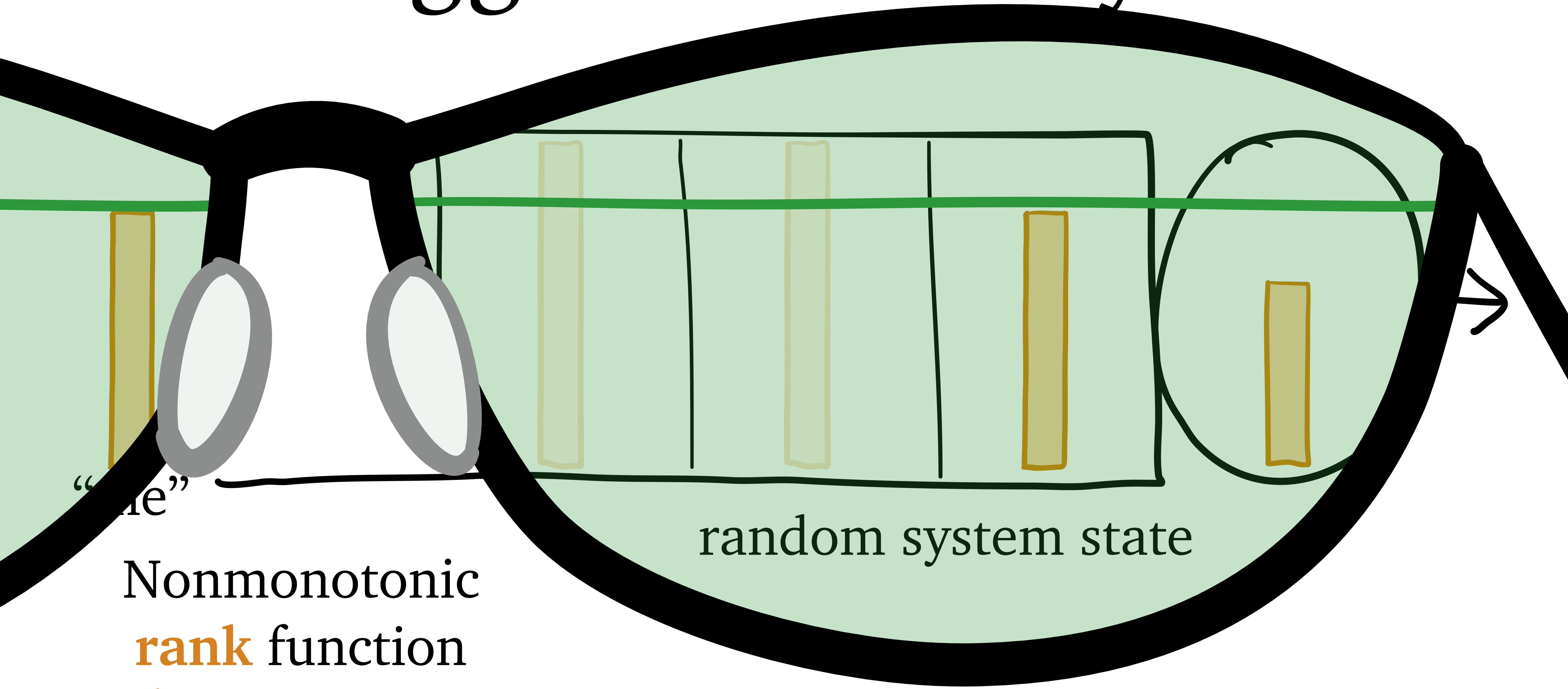

Nonmonotonic
**rank** function



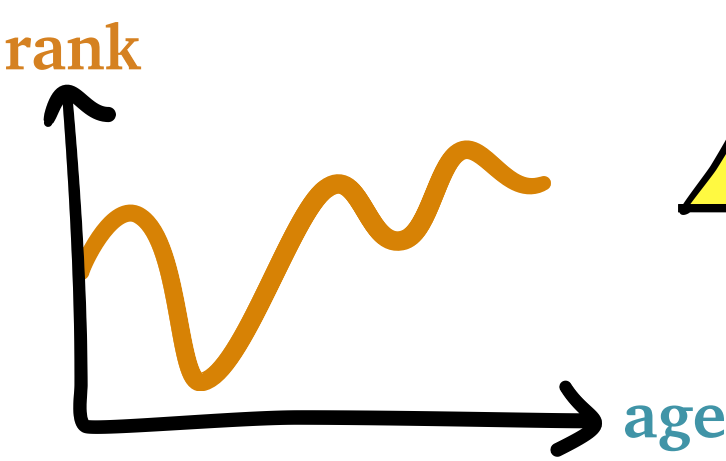

**Two obstacles:**

- *My rank* goes up and down

# Tagged Job Analysis



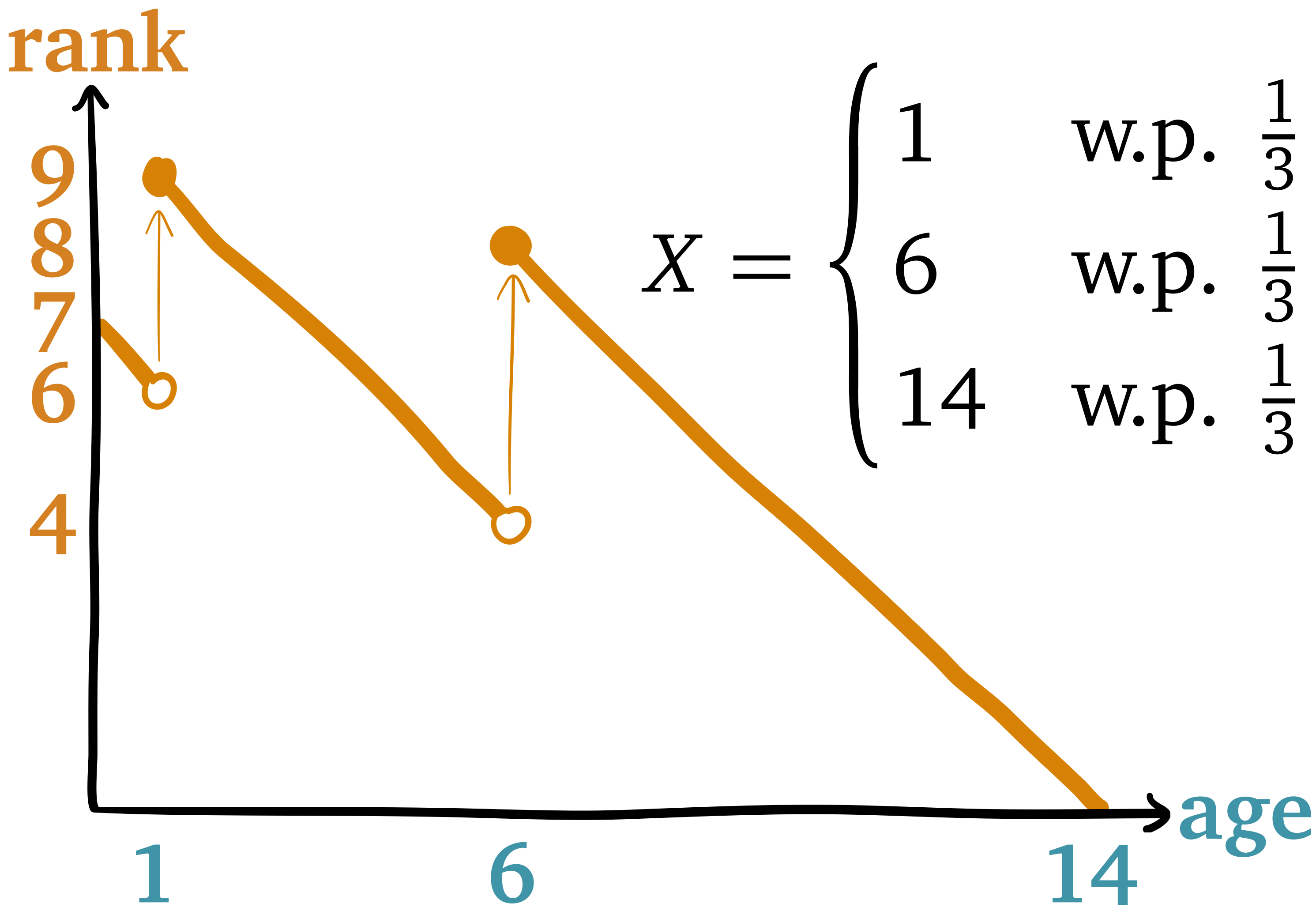

Nonmonotonic **rank** function




**Two obstacles:**

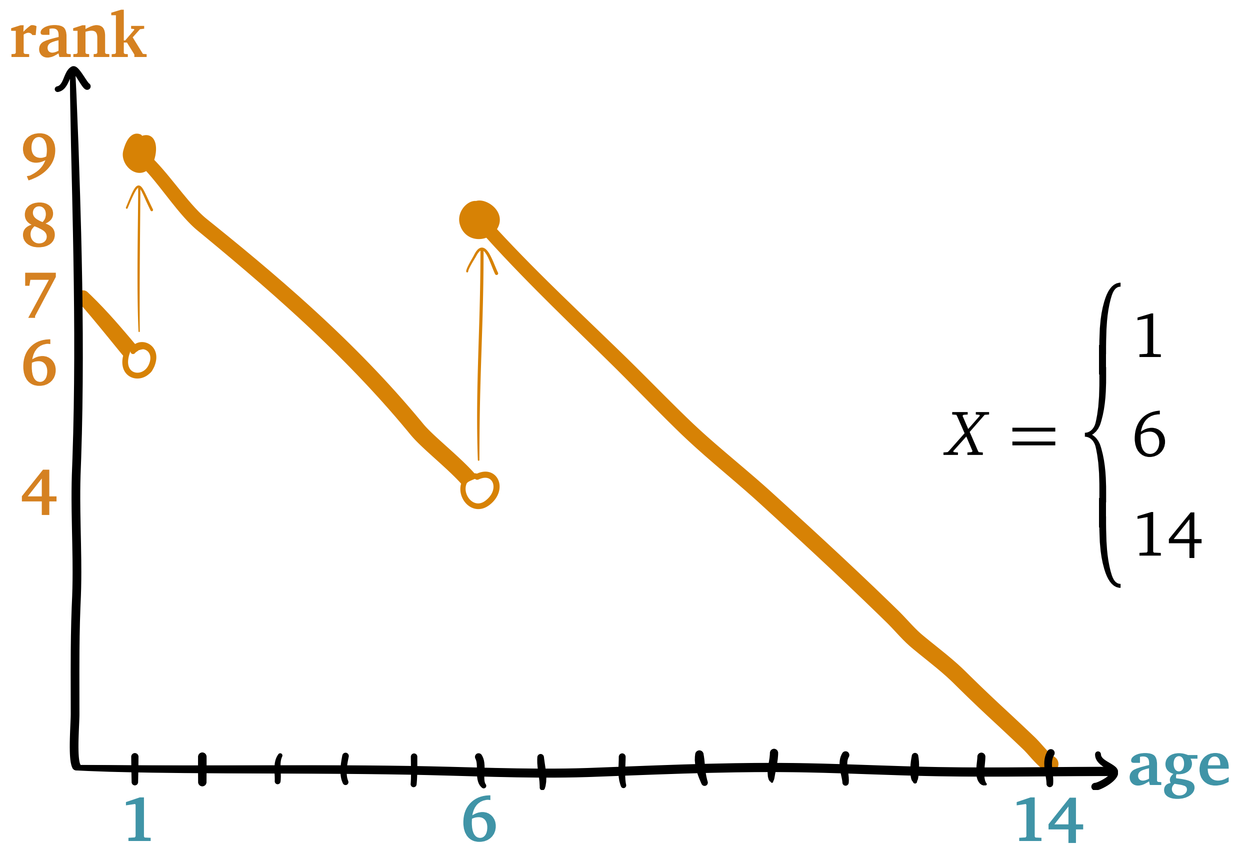
- *My rank* goes up and down
- *Others' ranks* go up and down too

# Running example:
# **SERPT**



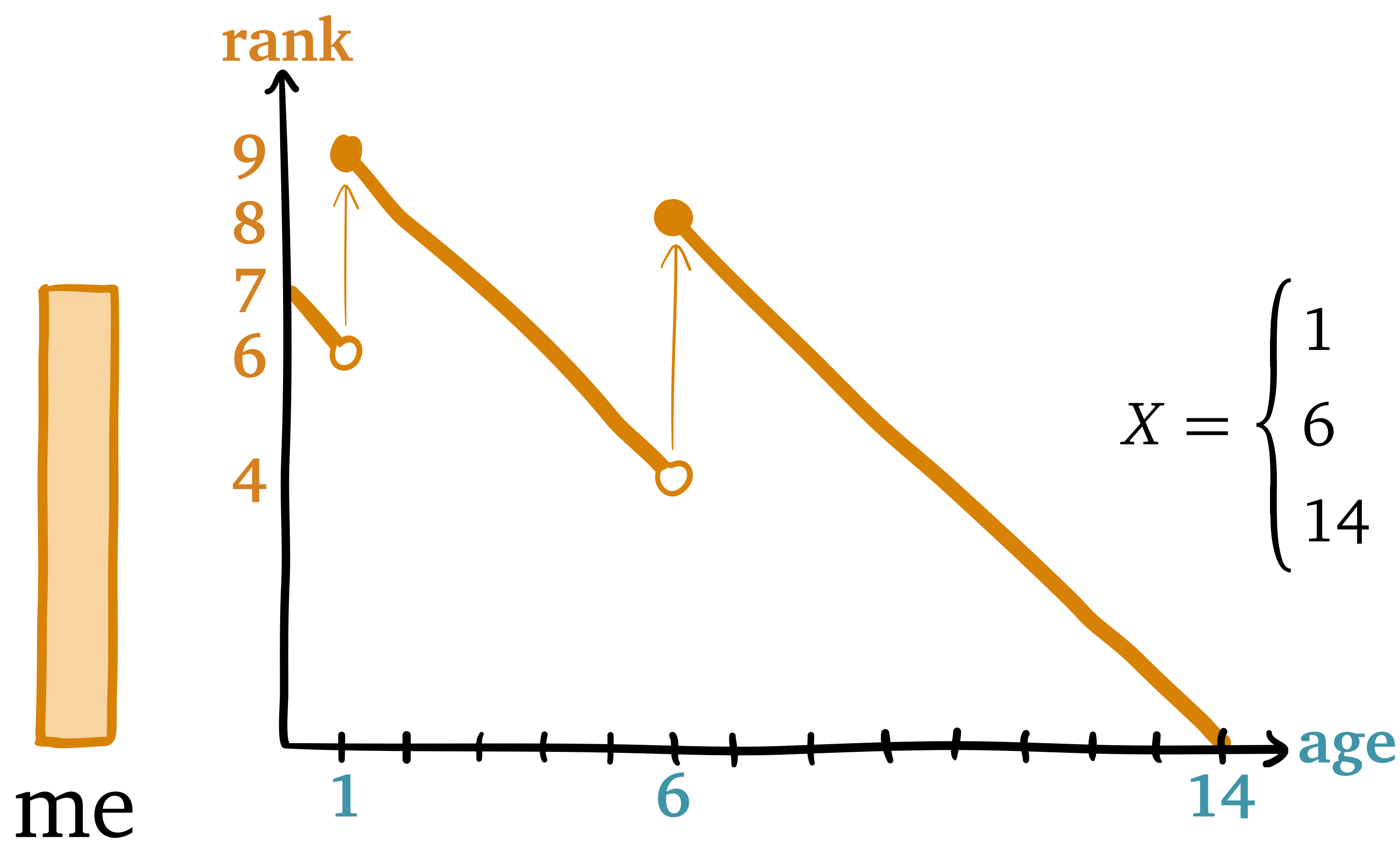

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Warmup: Empty System



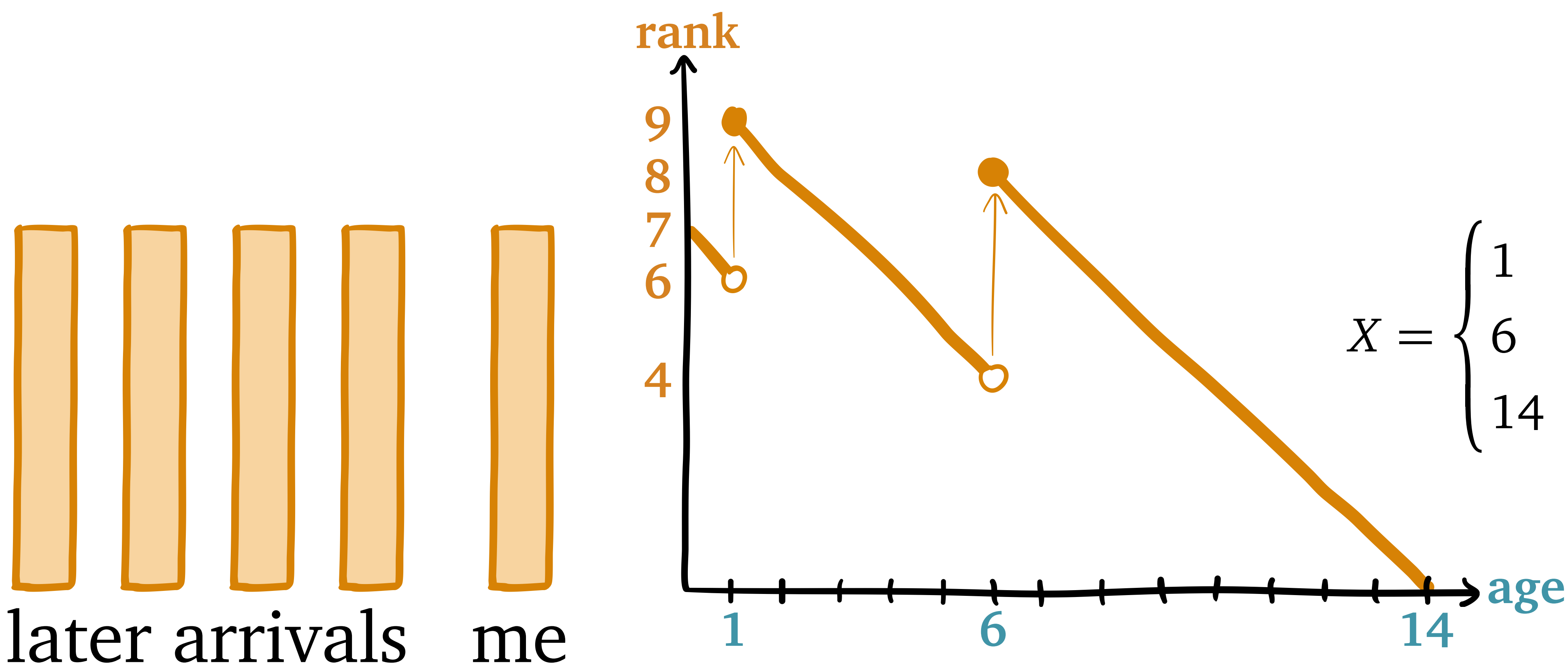

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Warmup: Empty System



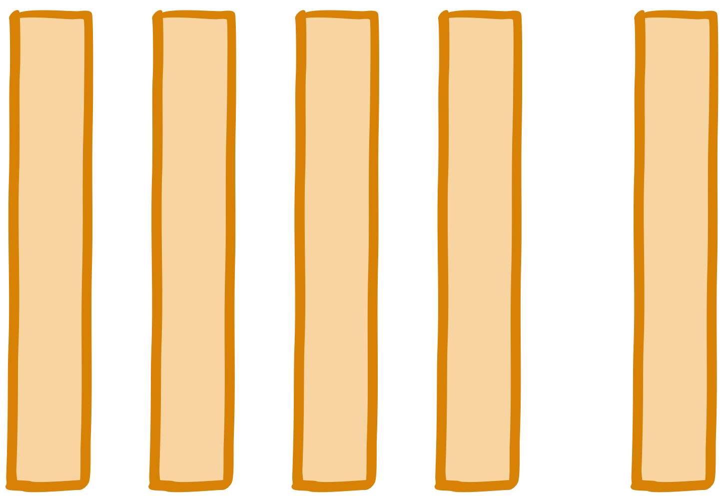

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Warmup: Empty System

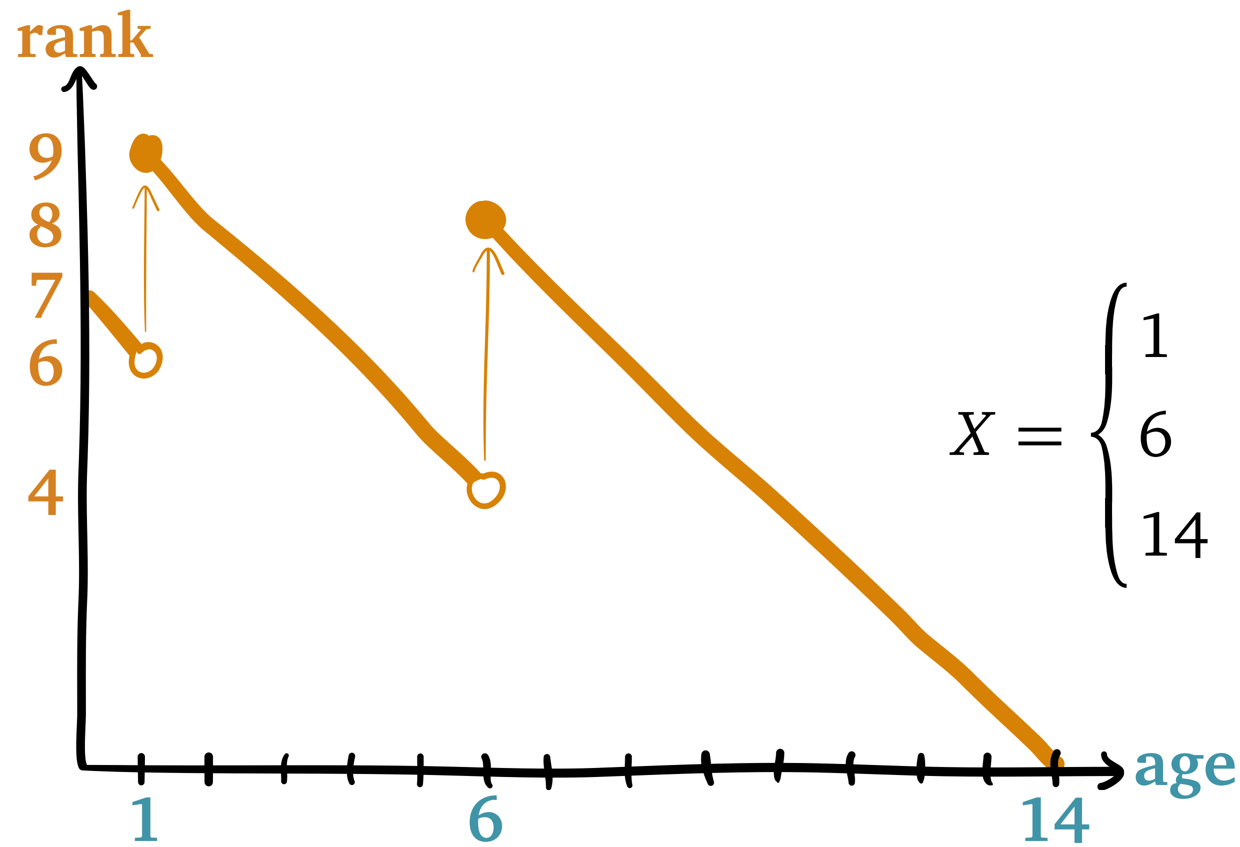



$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Warmup: Empty System



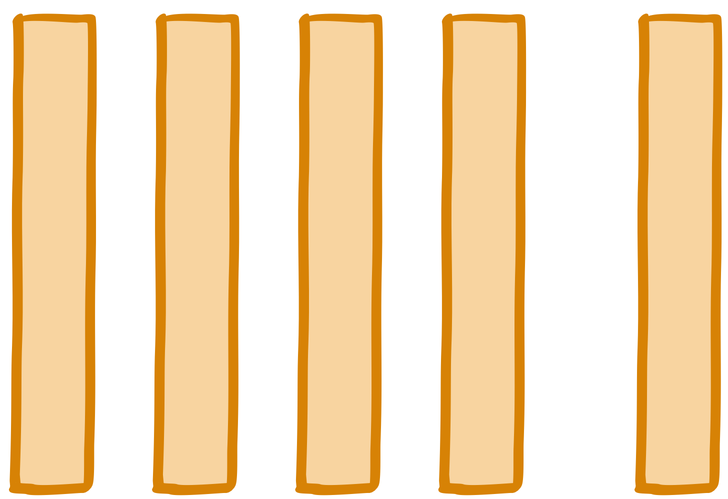
later arrivals me



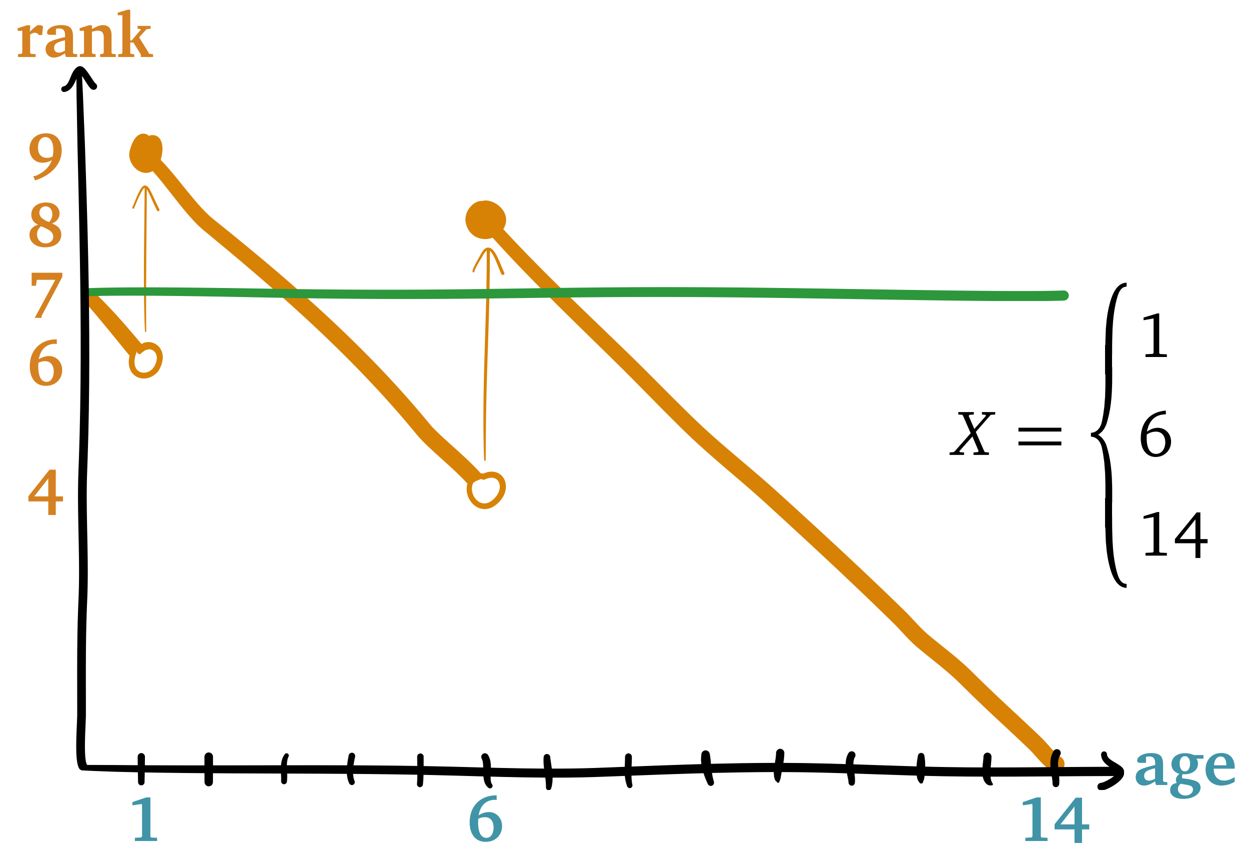

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | | |
| 6 | | |
| 14 | | |

# Warmup: Empty System



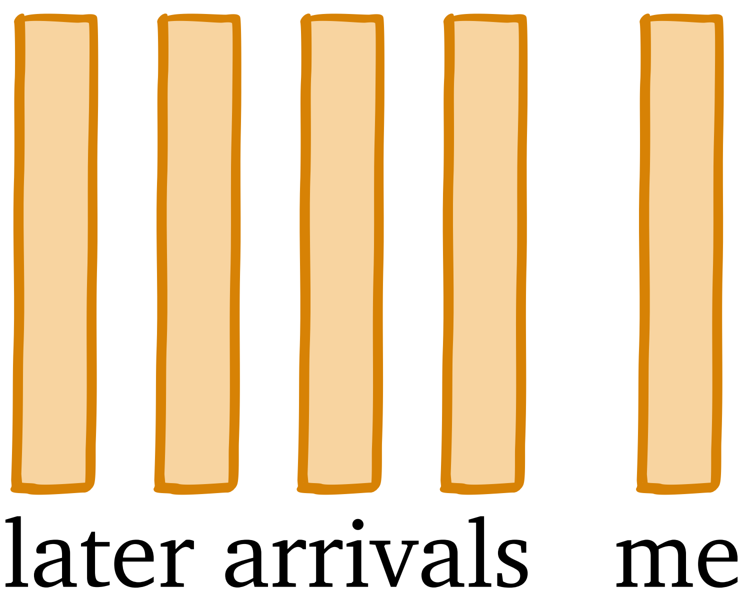
later arrivals me

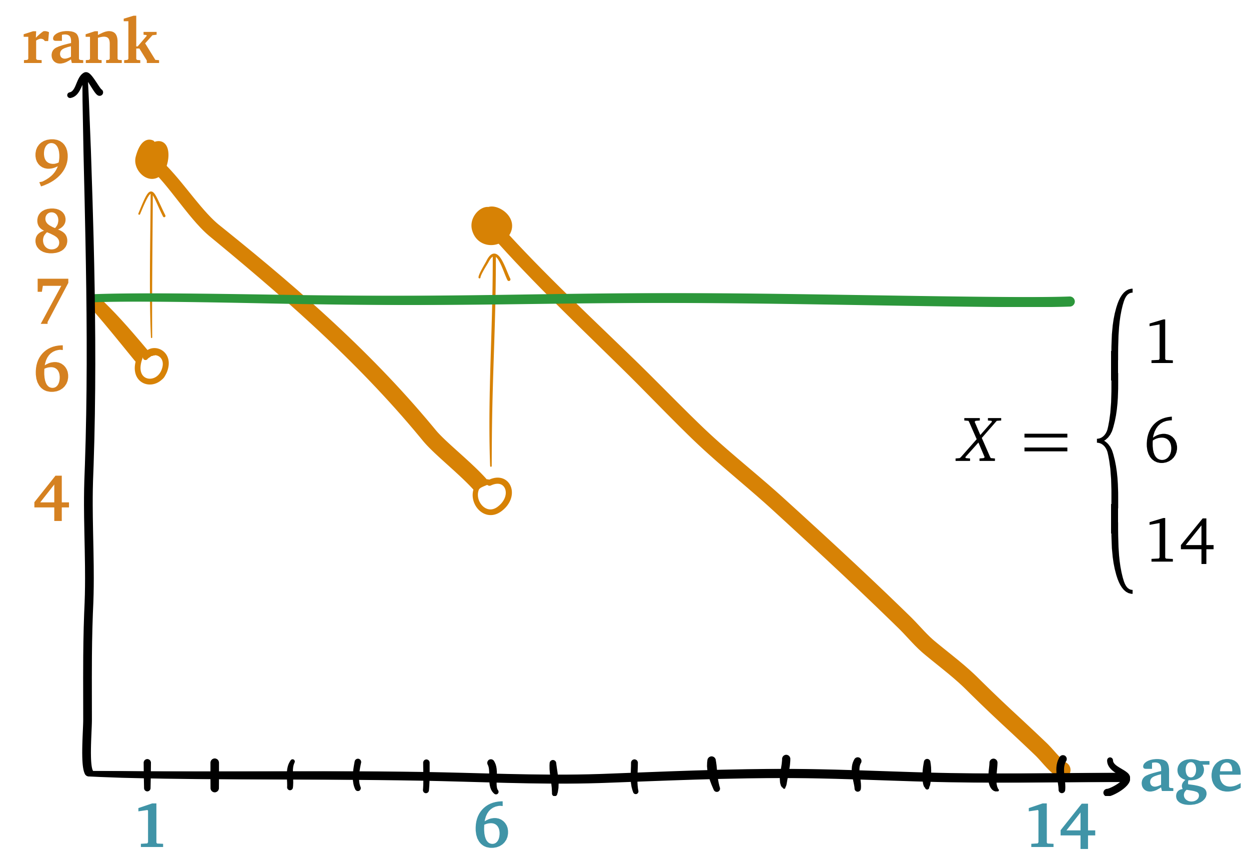



$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | | |
| 6 | | |
| 14 | | |

# Warmup: Empty System

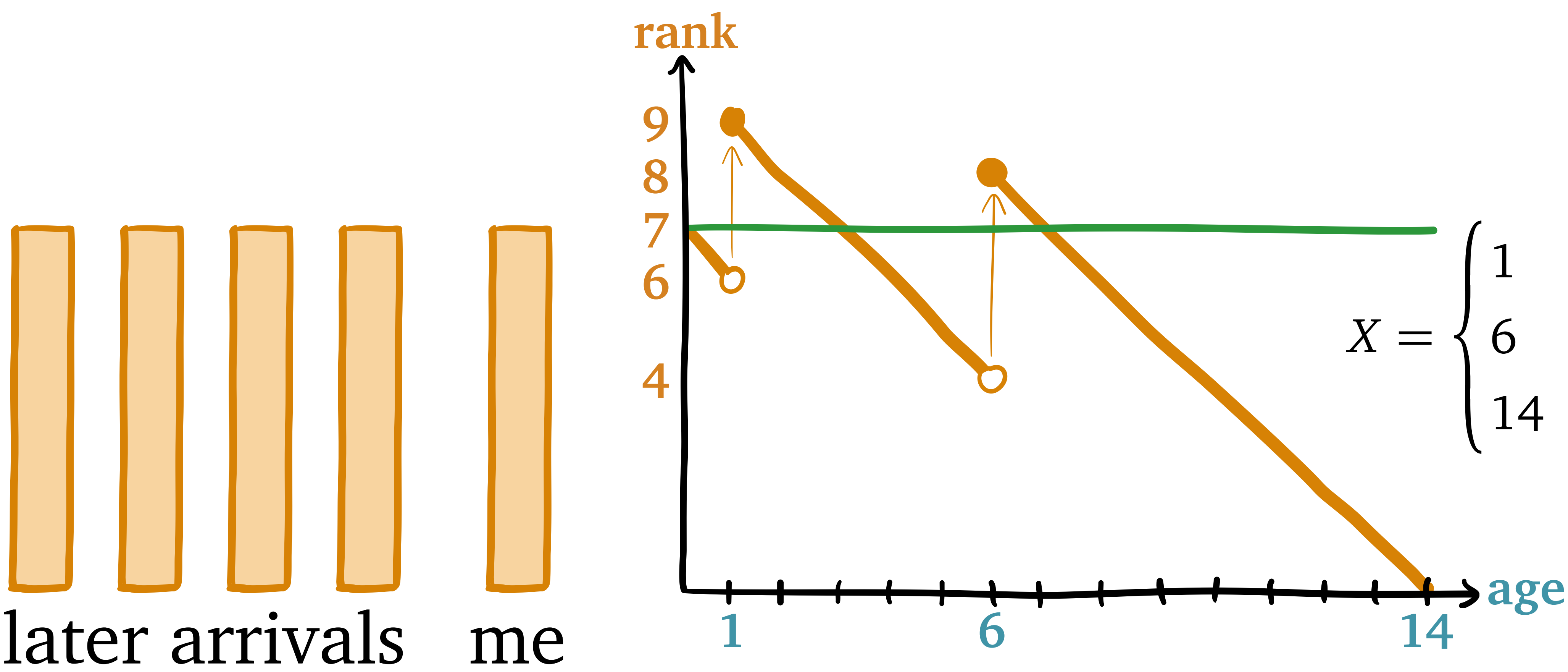



$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | |
| 6 | | |
| 14 | | |

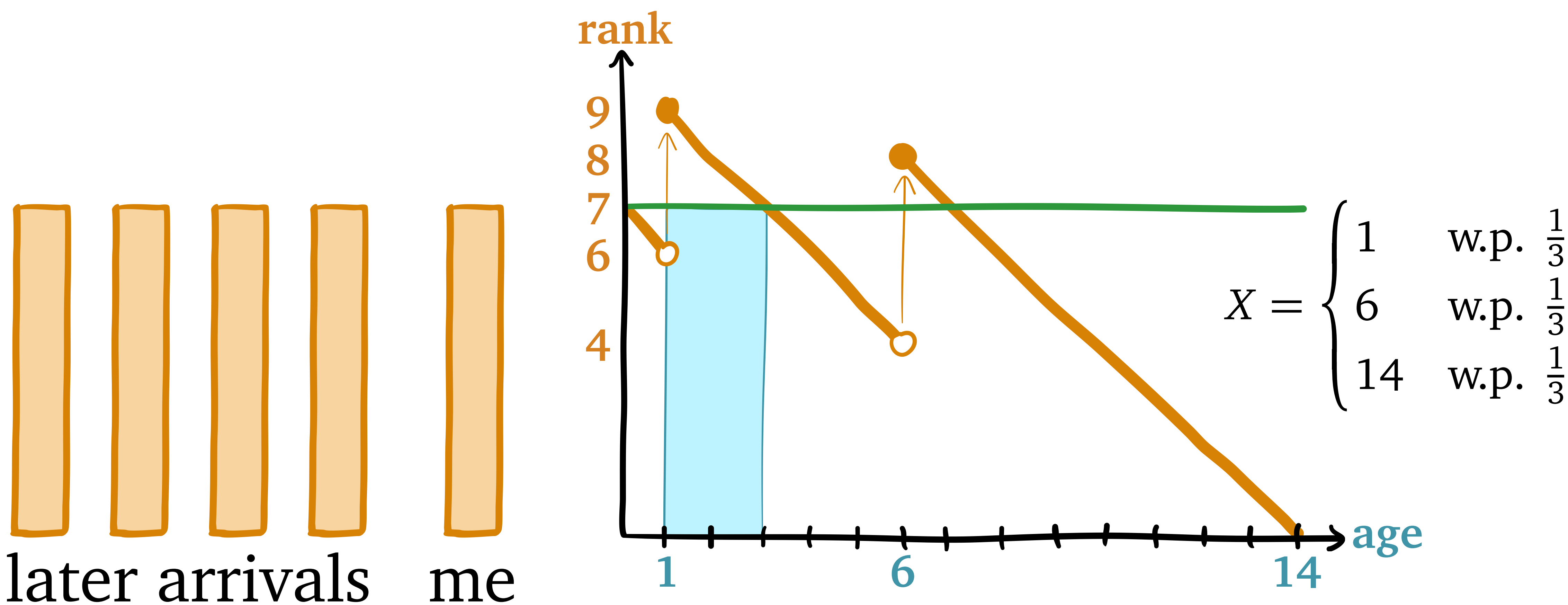
# Warmup: Empty System




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

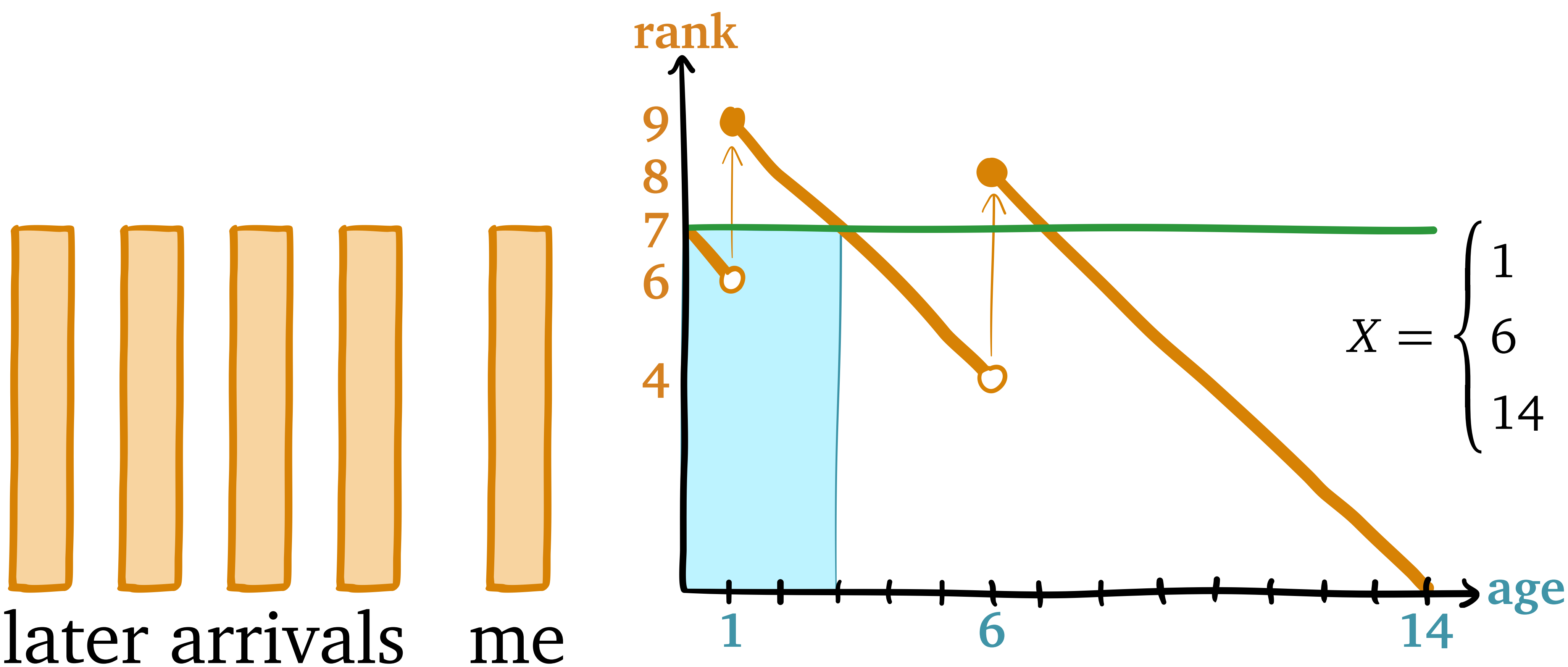
| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | n/a |
| 6 | | |
| 14 | | |

# Warmup: Empty System




| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | n/a |
| 6 | | |
| 14 | | |

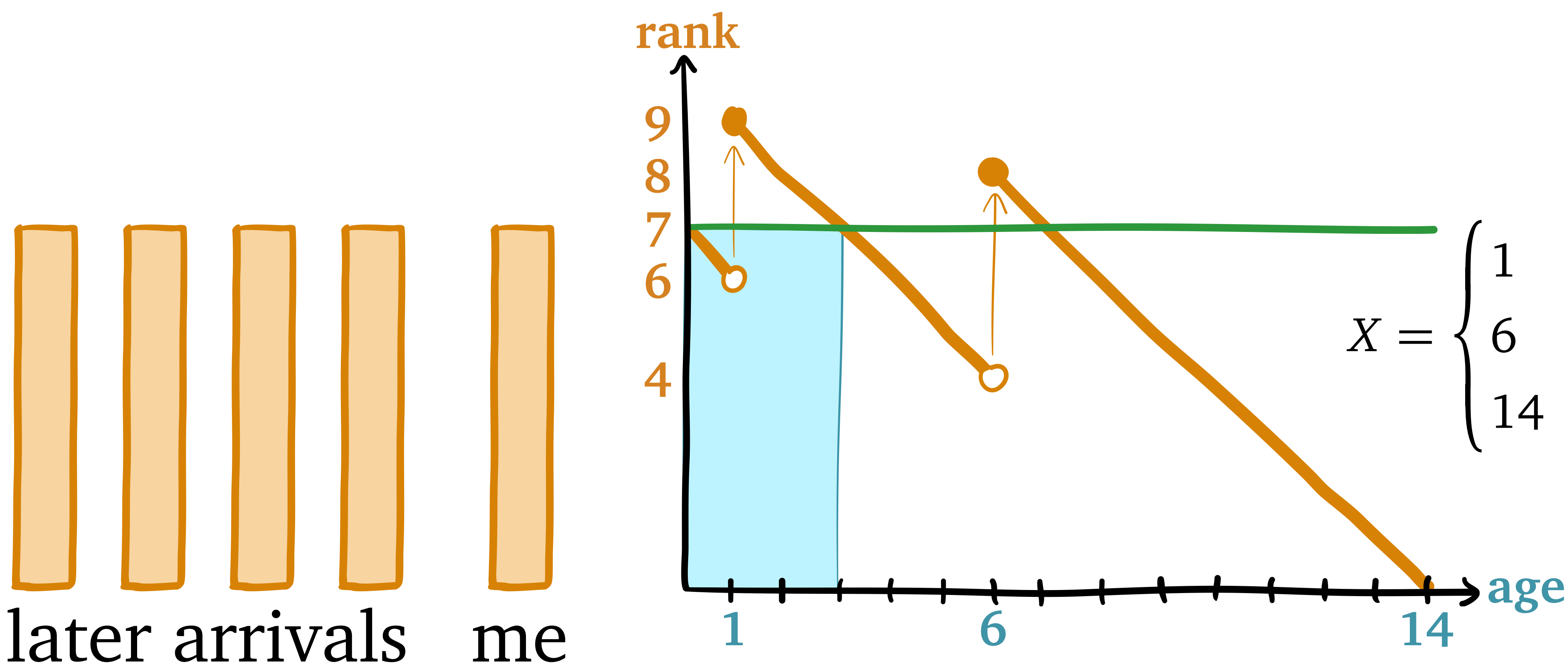
# Warmup: Empty System




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | n/a |
| 6 | | |
| 14 | | |

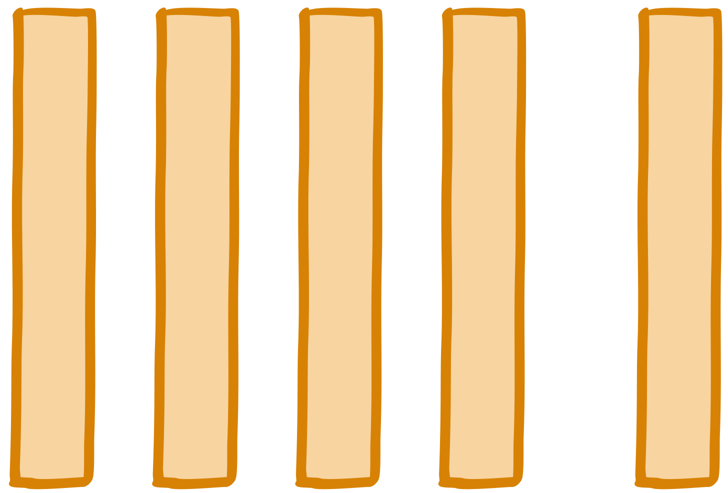
# Warmup: Empty System

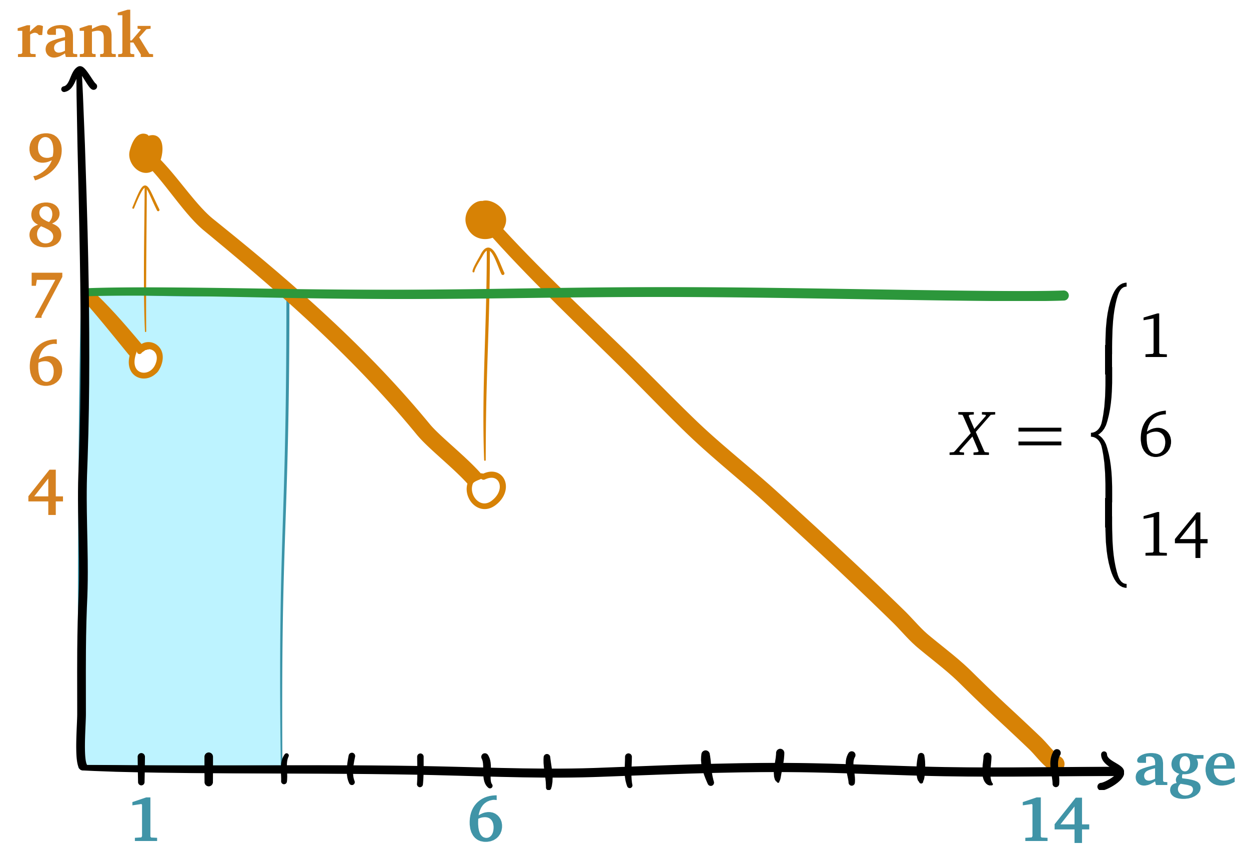



$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | n/a |
| 6 | when $0 \le$ my age $< 3$ | |
| 14 | | |

# Warmup: Empty System

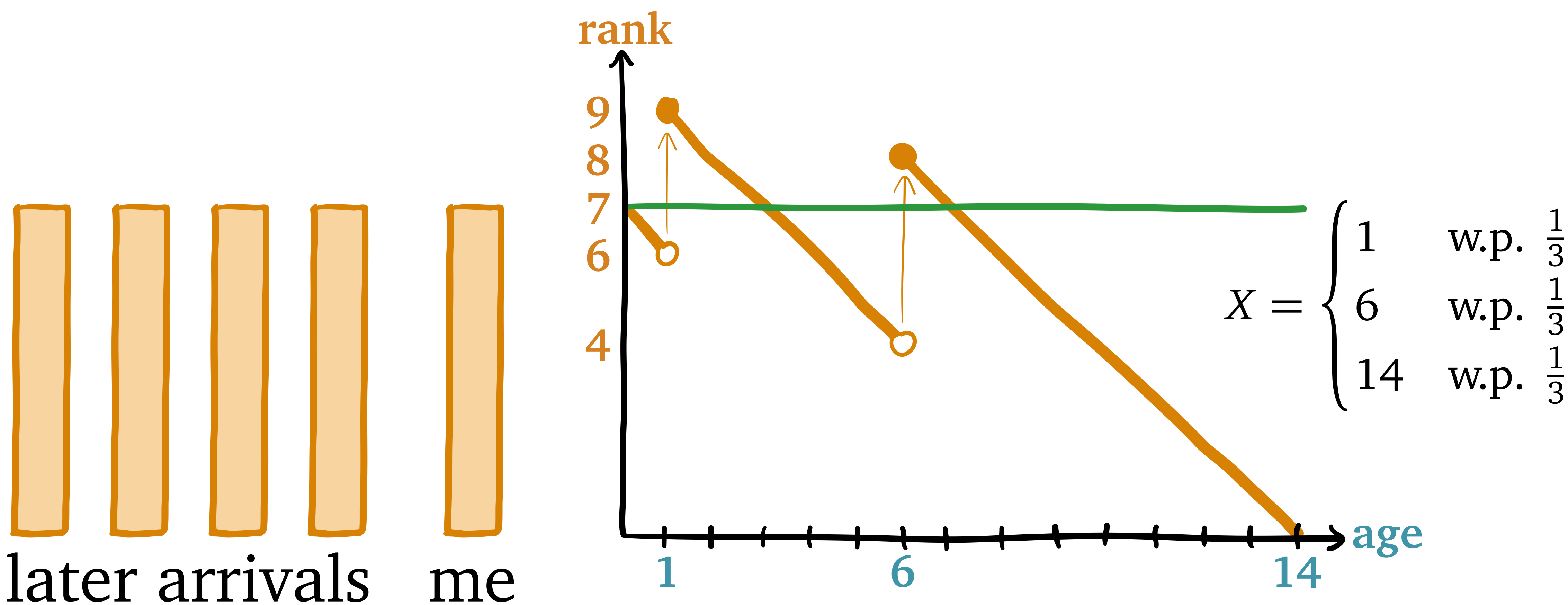



$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

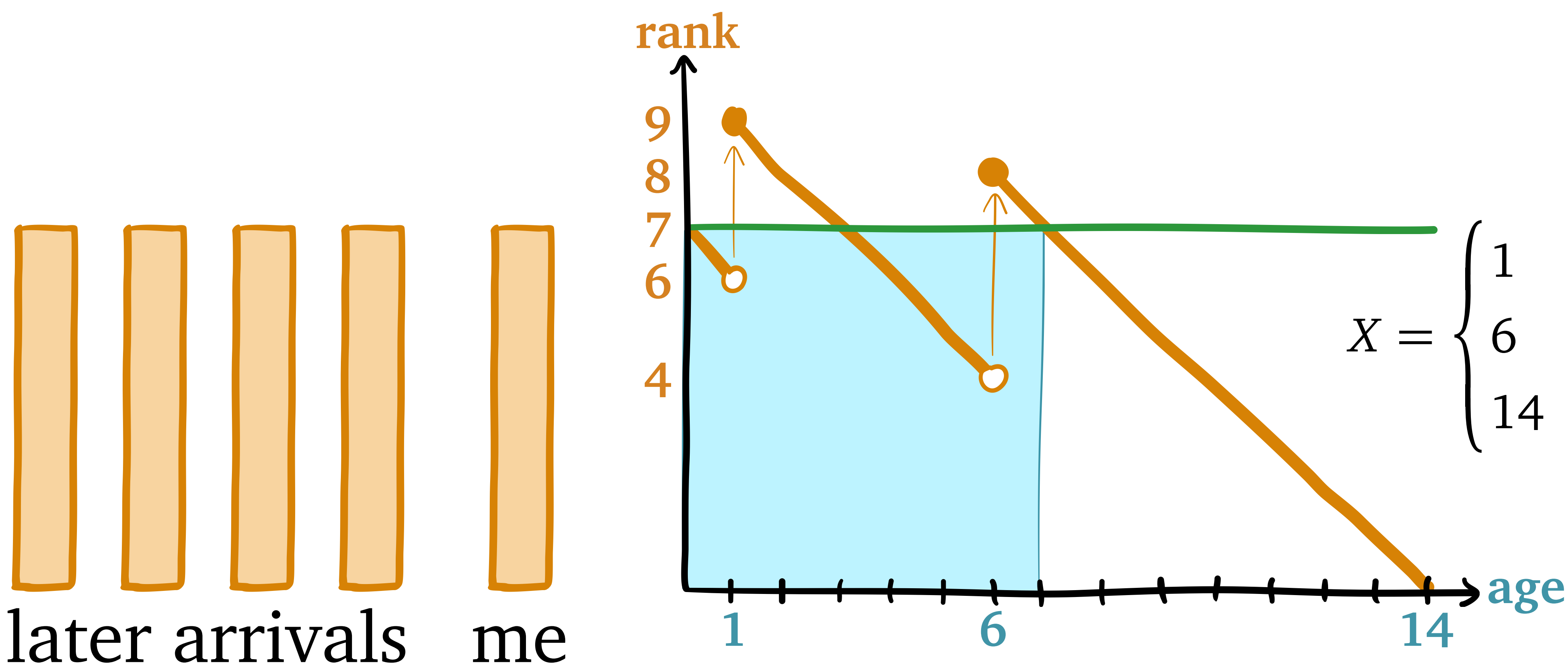
| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | n/a |
| 6 | when $0 \leq$ my age $< 3$ | 1 |
| 14 | | |

# Warmup: Empty System




| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | n/a |
| 6 | when **0** ≤ my age < **3** | 1 |
| 14 | | |

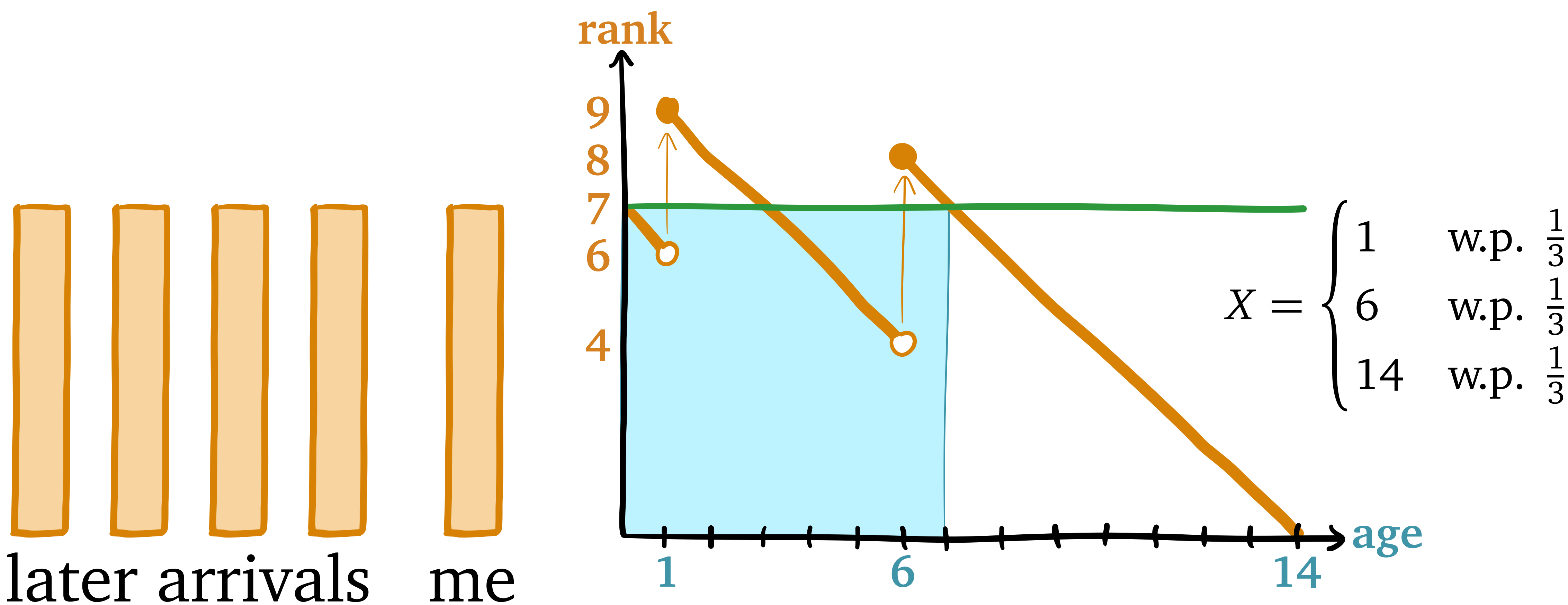
# Warmup: Empty System




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

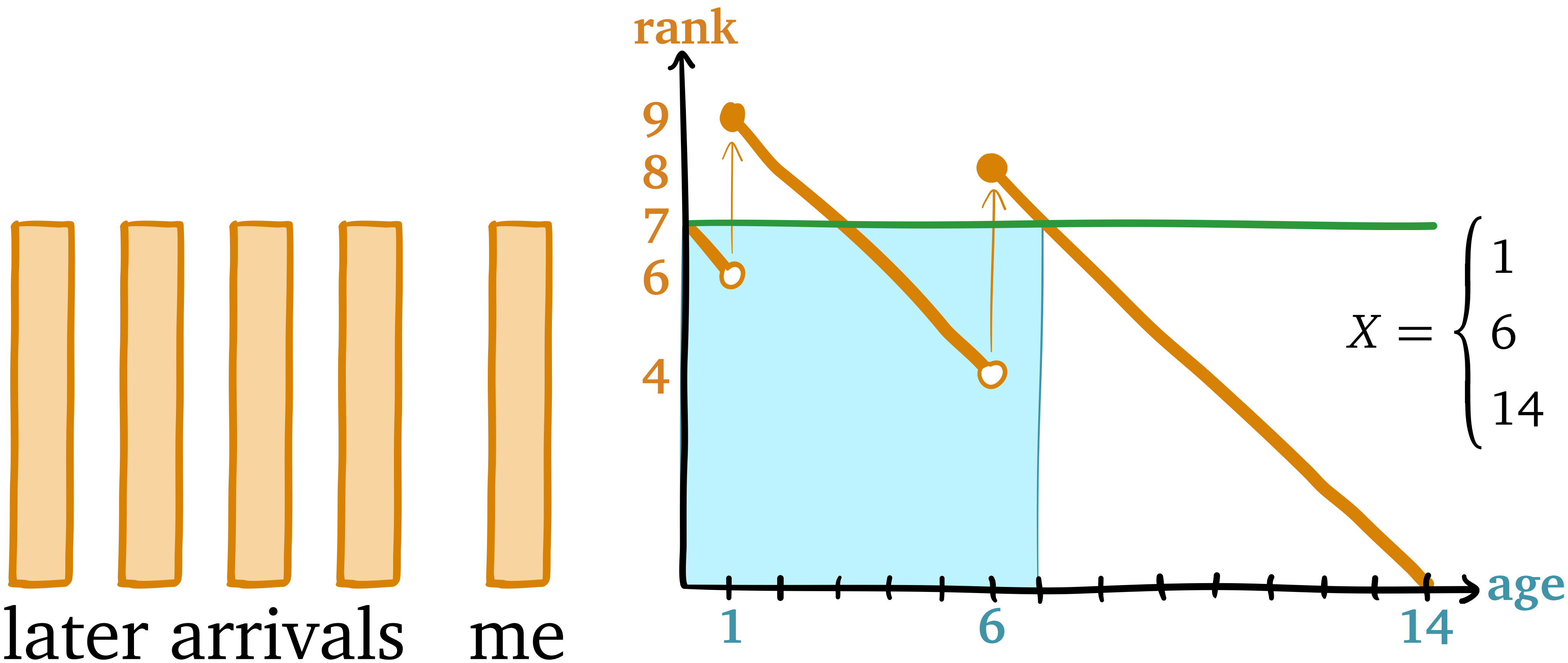
| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | n/a |
| 6 | when $0 \leq$ my age $< 3$ | 1 |
| 14 | | |

# Warmup: Empty System




| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | n/a |
| 6 | when $0 \le$ my age $< 3$ | 1 |
| 14 | when $0 \le$ my age $< 7$ | |

# Warmup: Empty System




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

| My size | Which arrivals delay me? | By how much? |
|---|---|---|
| 1 | none | n/a |
| 6 | when $0 \le$ my age $< 3$ | 1 |
| 14 | when $0 \le$ my age $< 7$ | 1 |

# **SOAP** Insight #1:
# **Pessimism Principle**

Replace my **rank** with my **worst** future rank

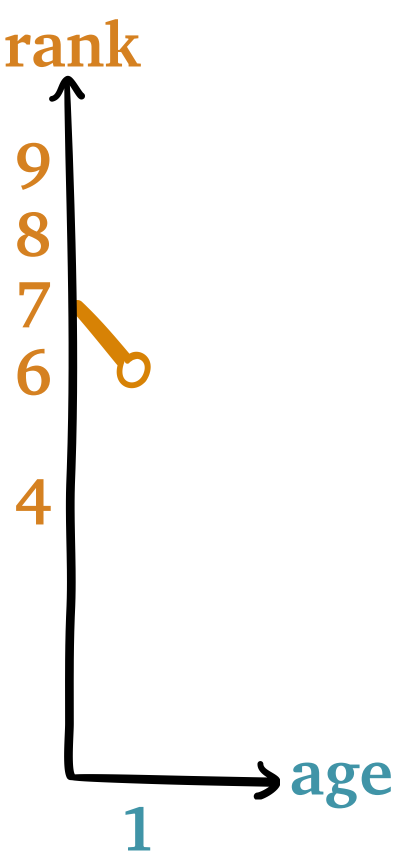
# Pessimism Principle

Replace my **rank** with my **worst** future rank

# Pessimism Principle

Replace my **rank** with my **worst** future rank

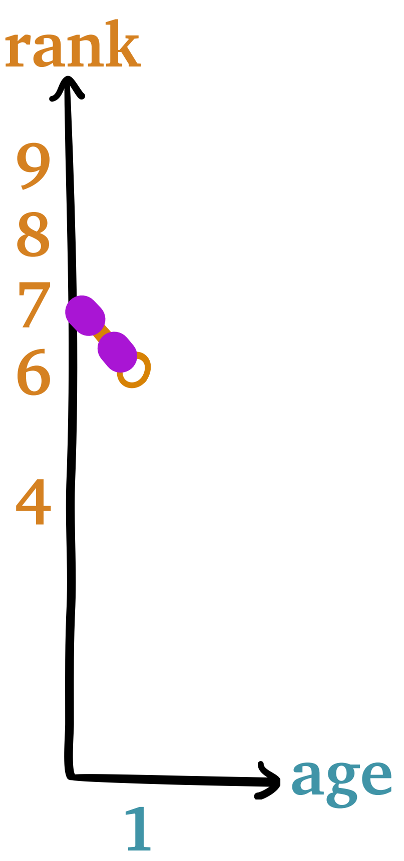
my size = 1



# Pessimism Principle

Replace my **rank** with my **worst** future rank

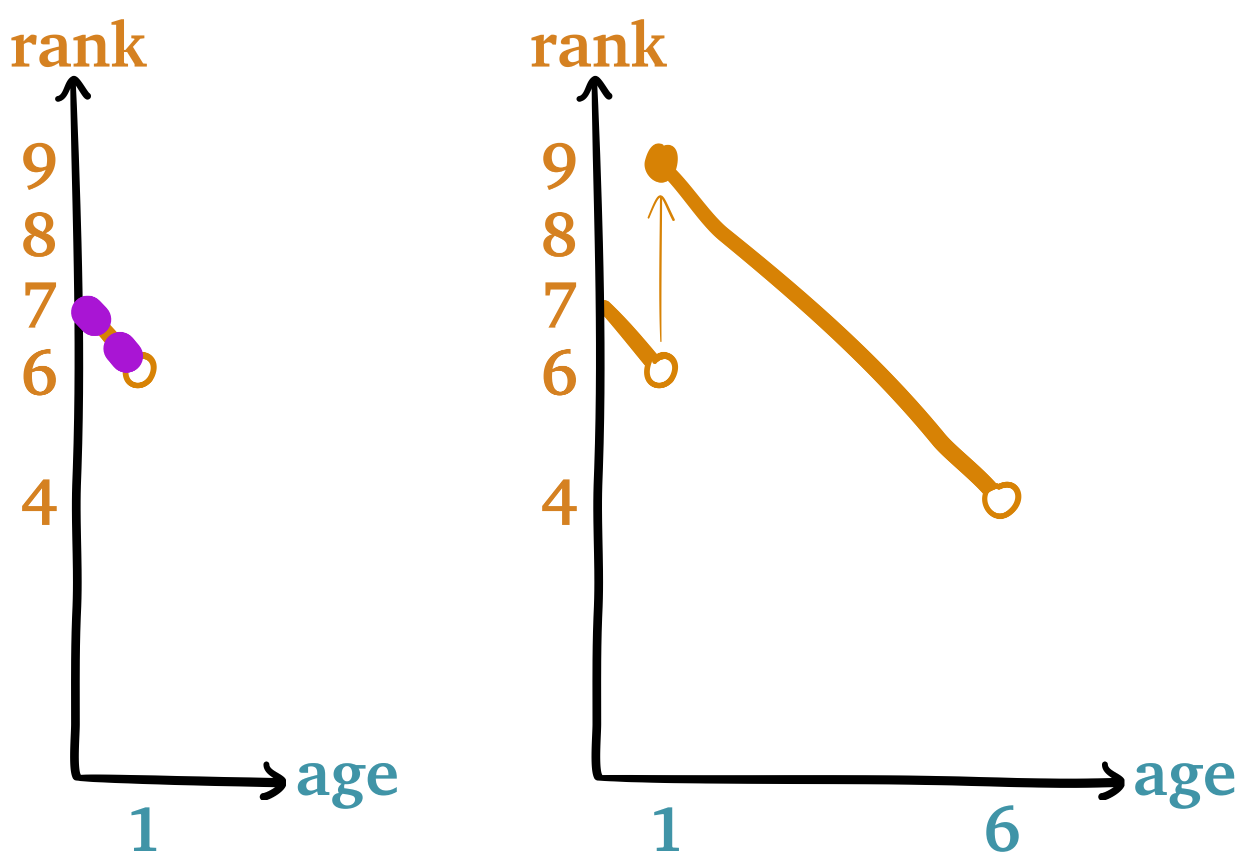
my size = 1



# Pessimism Principle

Replace my **rank** with my **worst** future rank

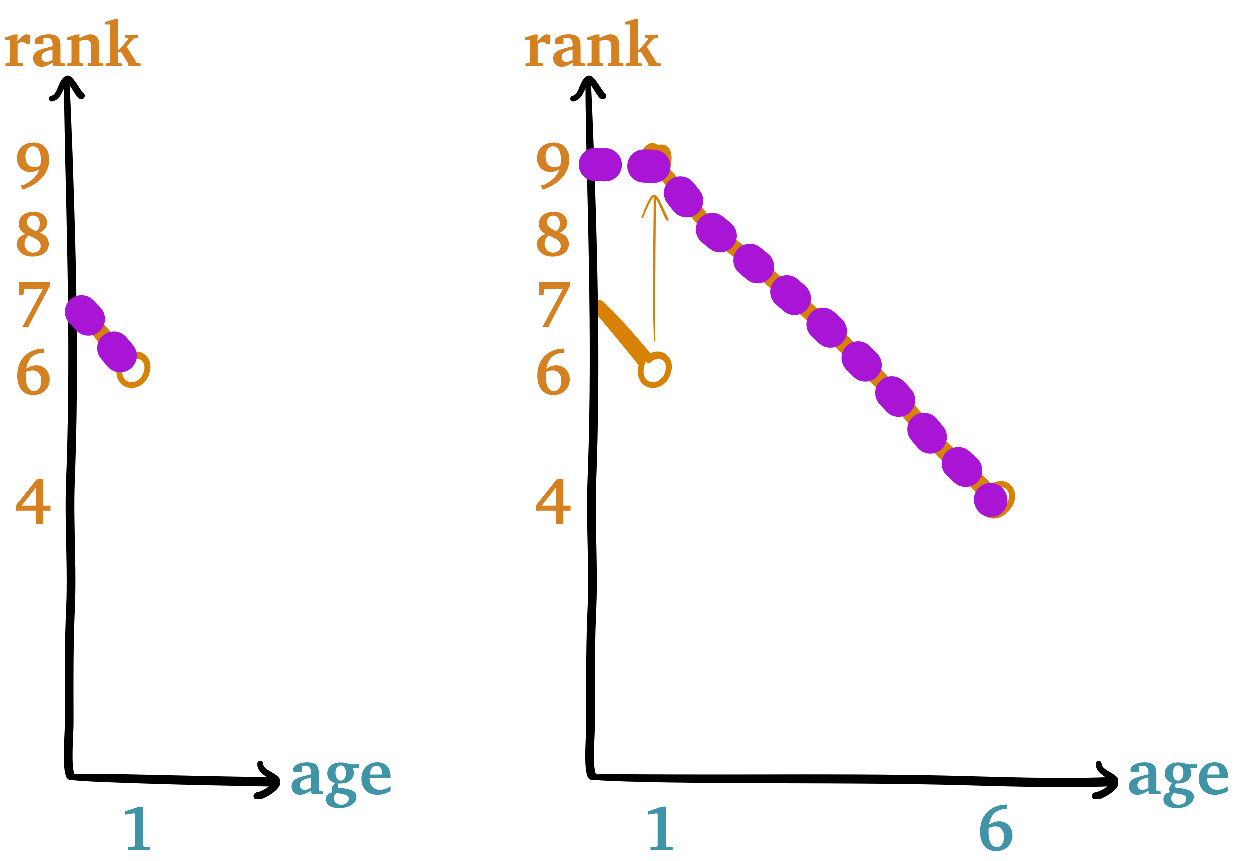
my size = 1 my size = 6



# Pessimism Principle

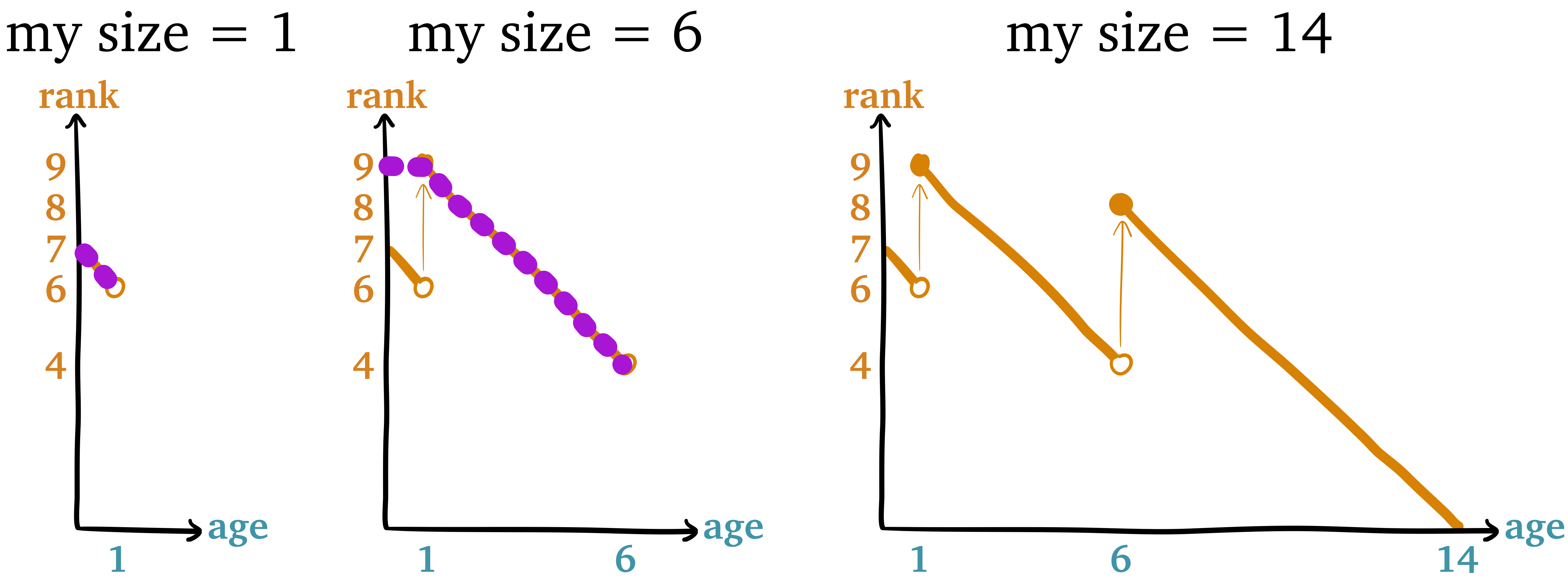
Replace my **rank** with my **worst** future rank

my size = 1 my size = 6



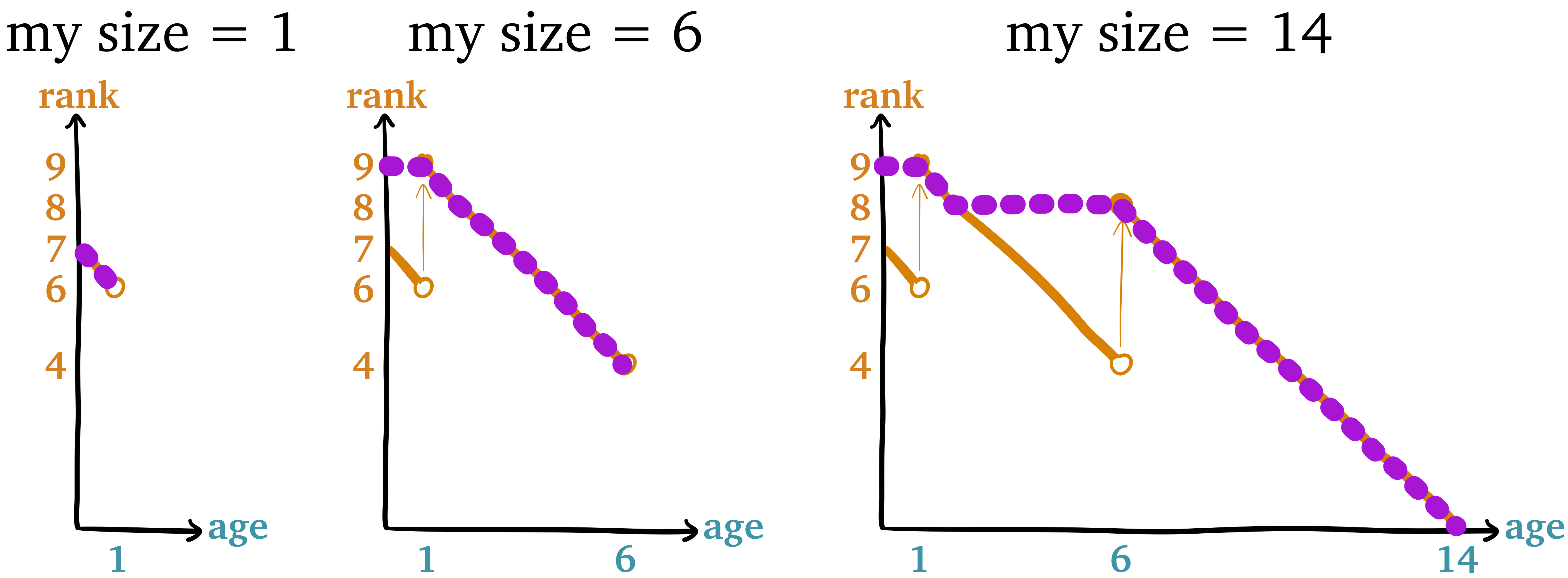
# Pessimism Principle

Replace my **rank** with my **worst** future rank



# Pessimism Principle

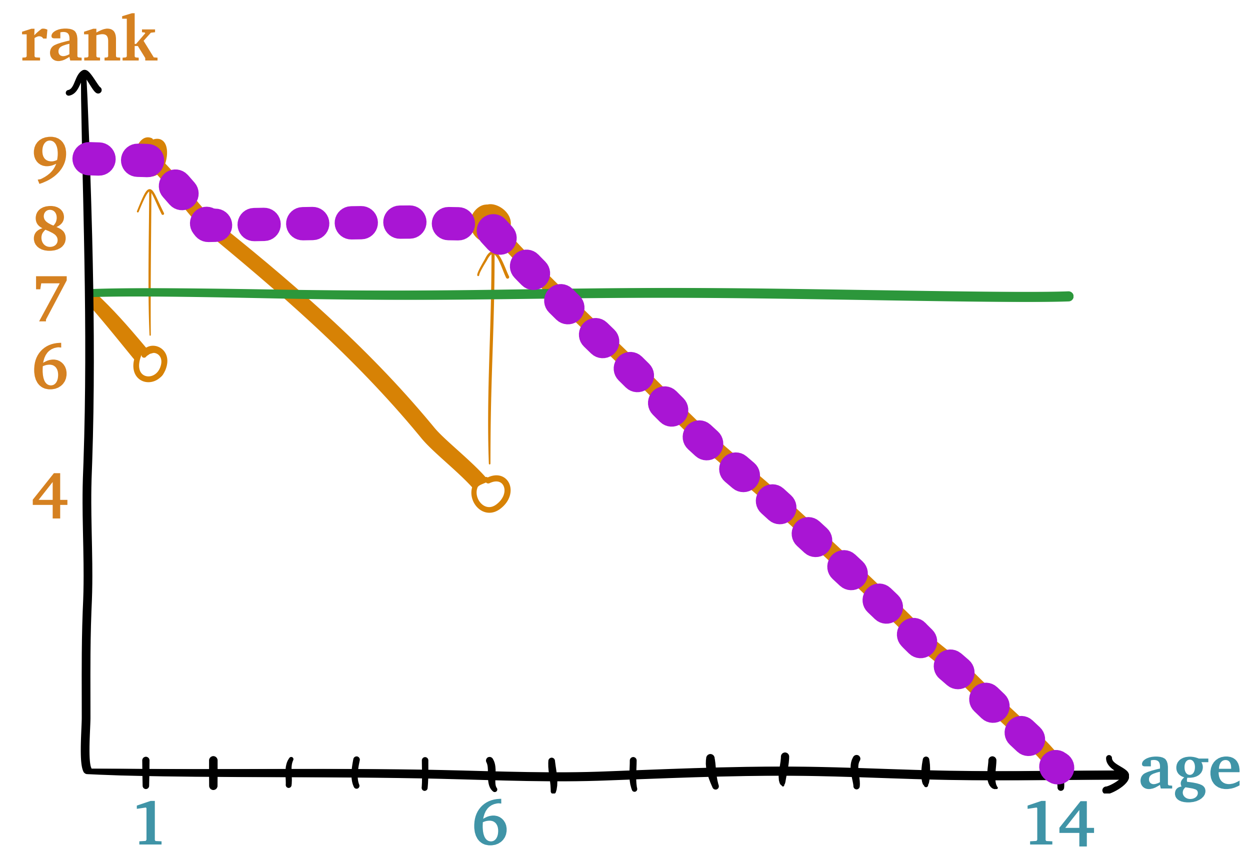
Replace my **rank** with my **worst** future rank



# Pessimism Principle

Replace my **rank** with my **worst** future rank

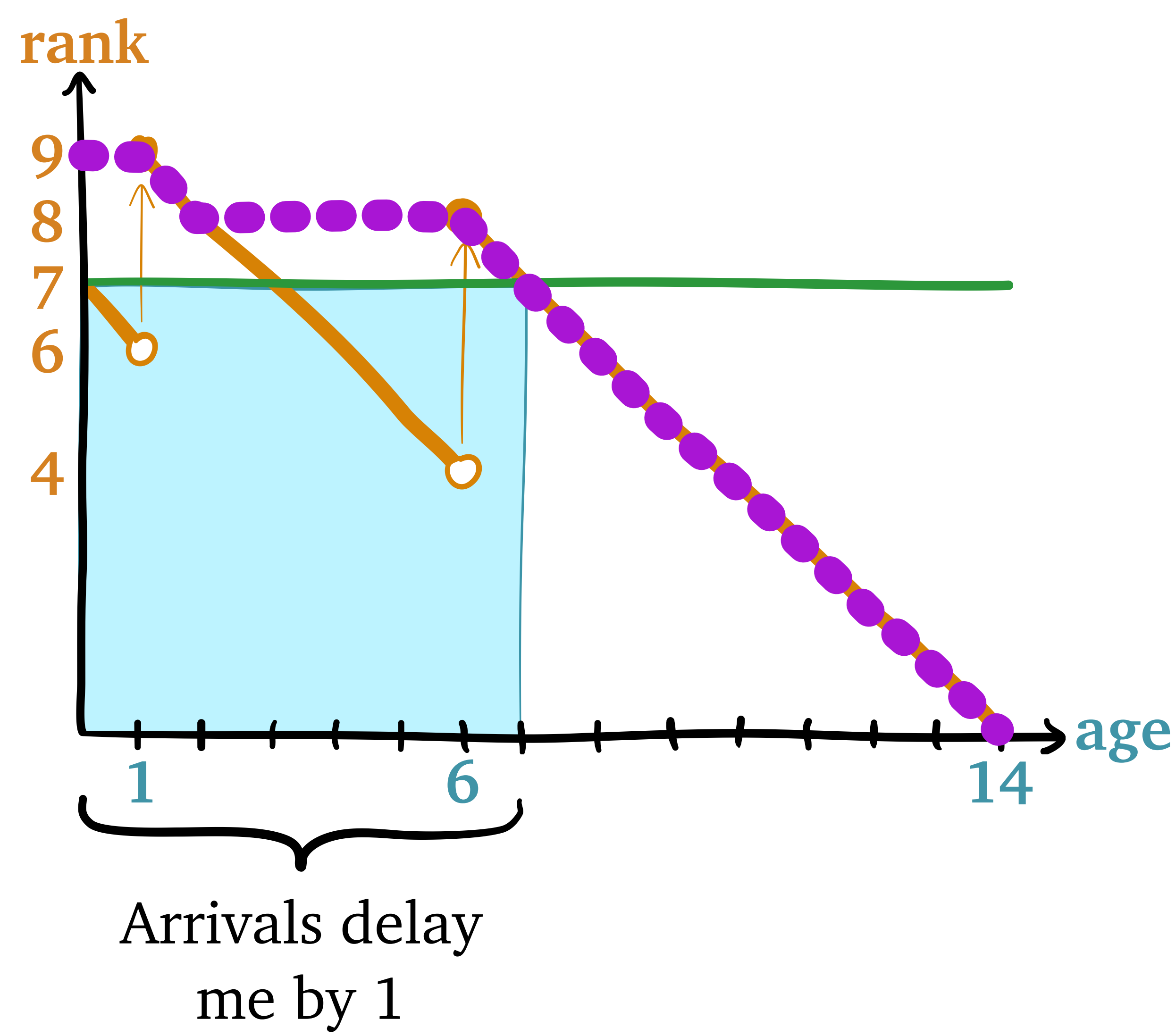
my size = 14



# Pessimism Principle

Replace my **rank** with my **worst** future rank

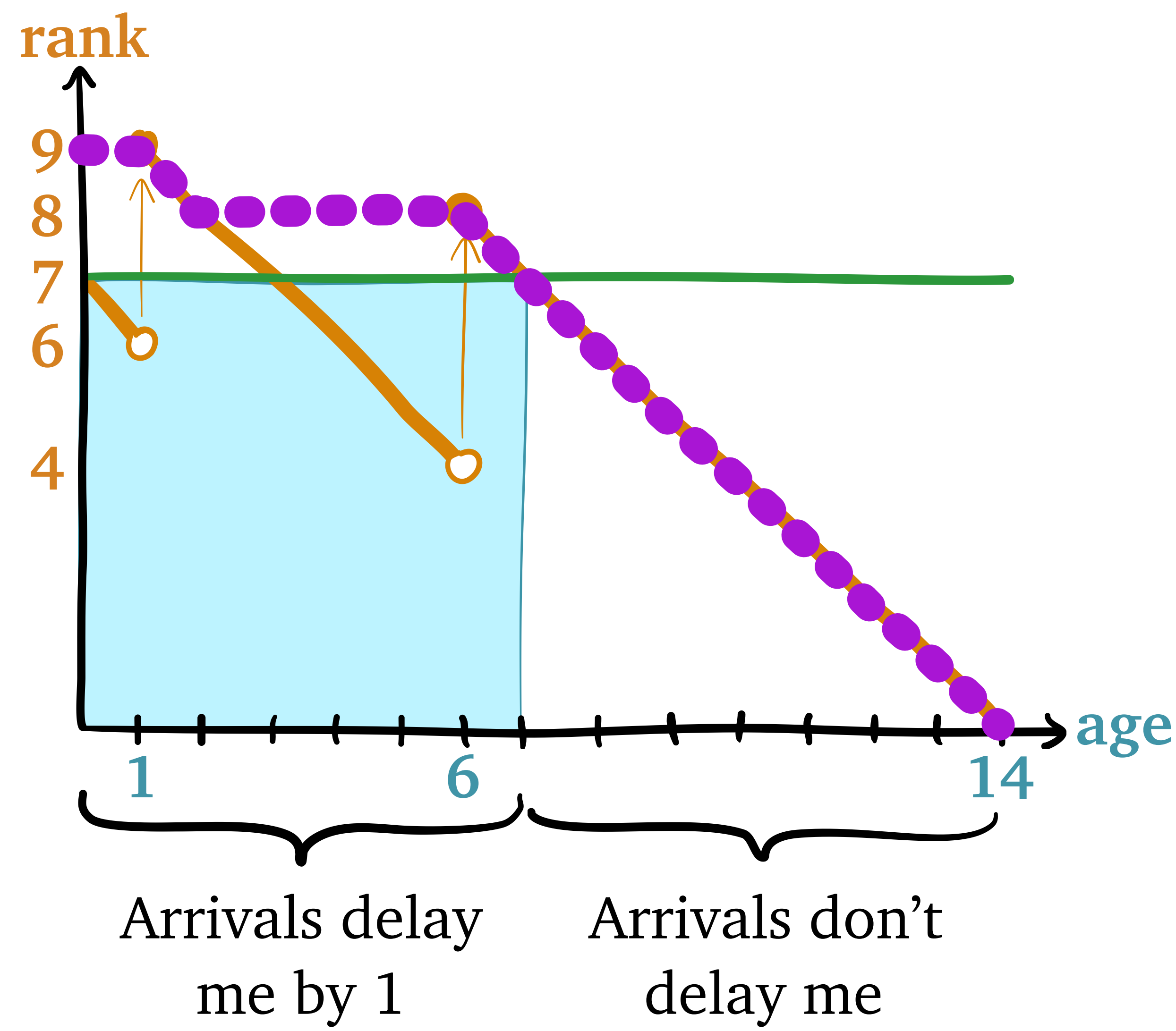
my size = 14



# Pessimism Principle

Replace my **rank** with my **worst** future rank

my size = 14

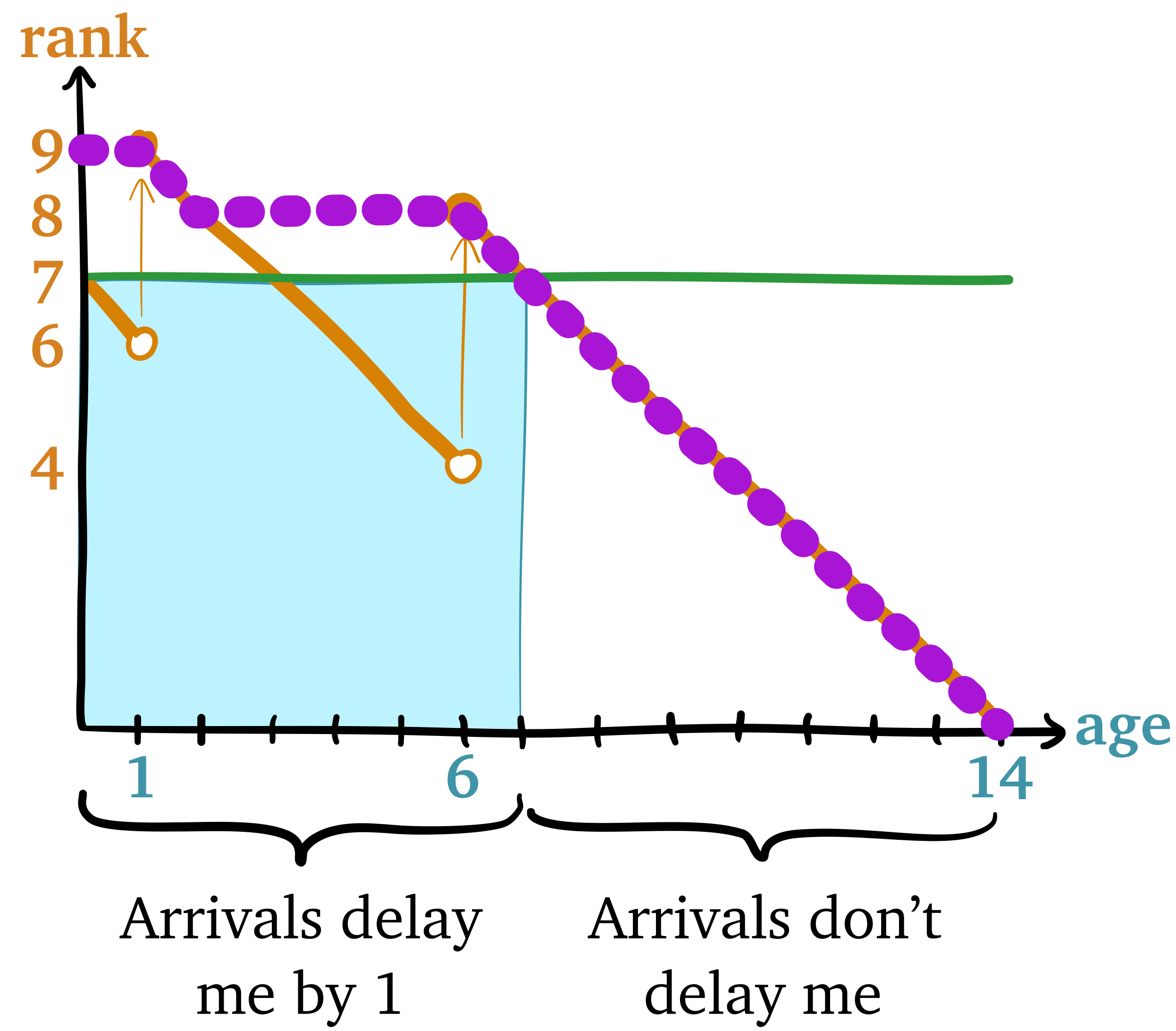


# Pessimism Principle

Replace my **rank** with my **worst** future rank

$$\rho_{\text{new}}(a) = \begin{cases} \lambda \cdot 1 & 0 \le a < 7 \\ \lambda \cdot 0 & 7 \le a < 14 \end{cases}$$

my size = 14



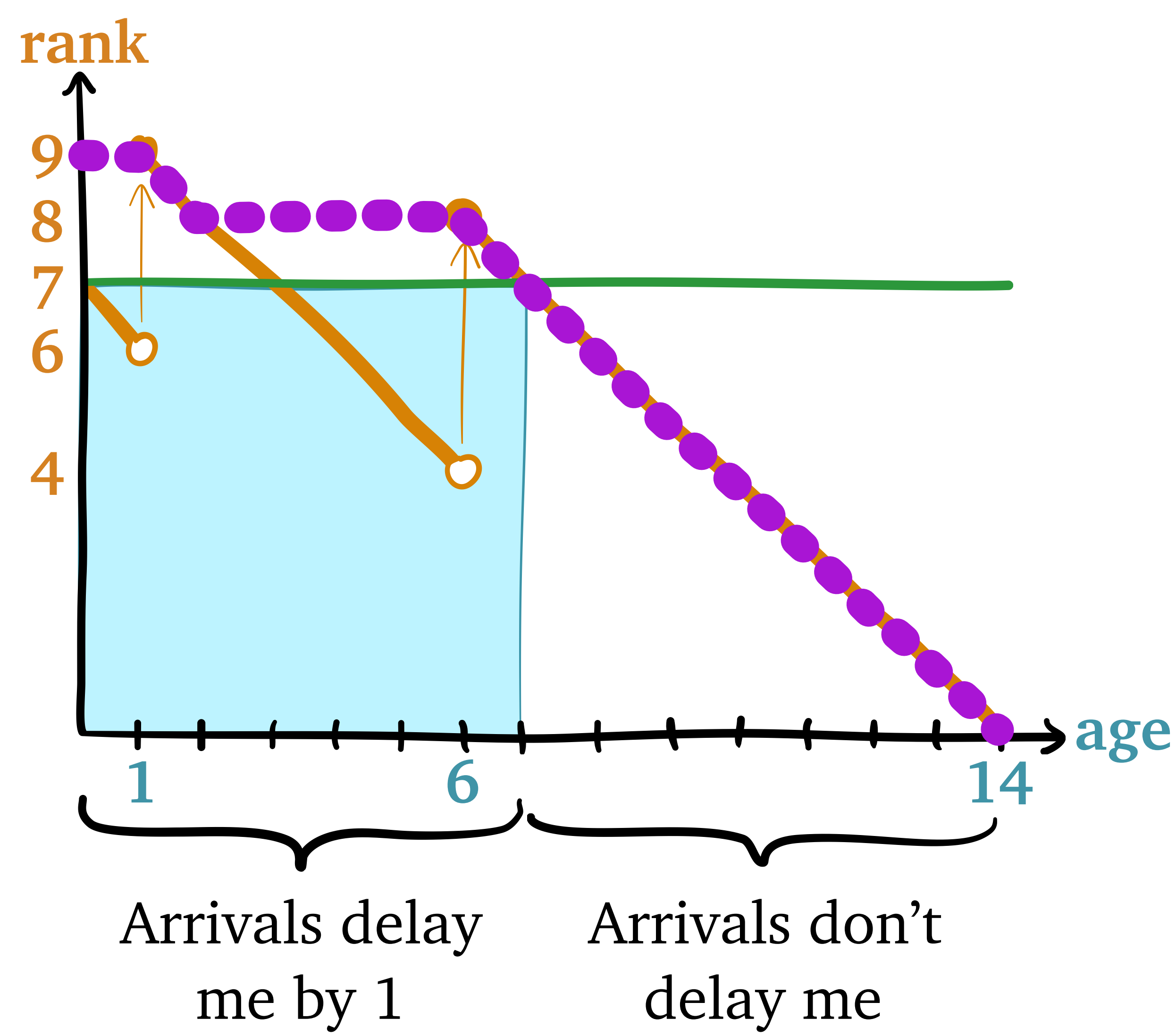
# Pessimism Principle

Replace my **rank** with my **worst** future rank

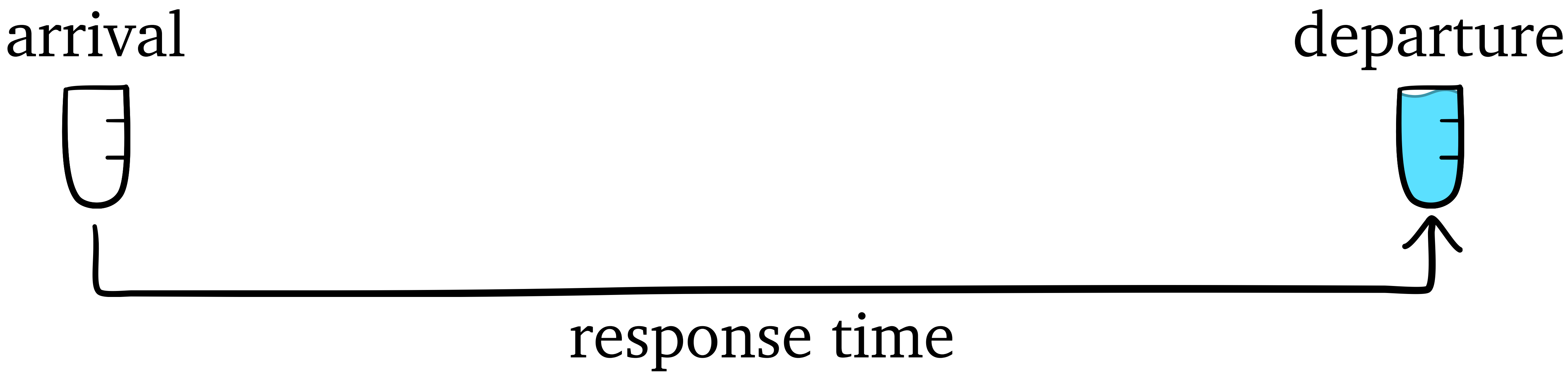
$$\rho_{\text{new}}(a) = \begin{cases} \lambda \cdot 1 & 0 \le a < 7 \\ \lambda \cdot 0 & 7 \le a < 14 \end{cases}$$

$$\mathbf{E}[T_{14} \mid \text{empty}] = \int_0^{14} \frac{\mathrm{d}a}{1 - \rho_{\text{new}}(a)}$$

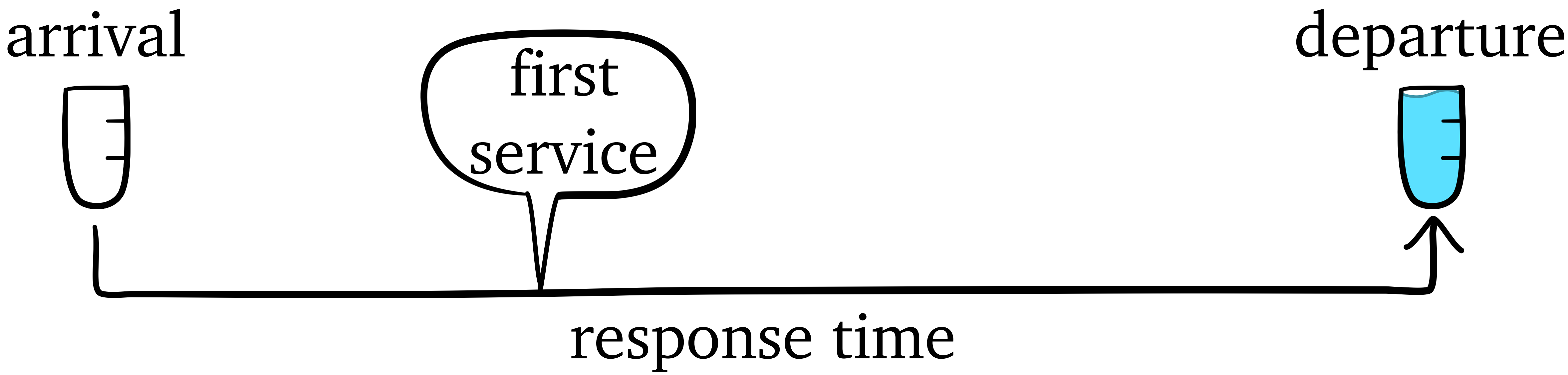
my size = 14



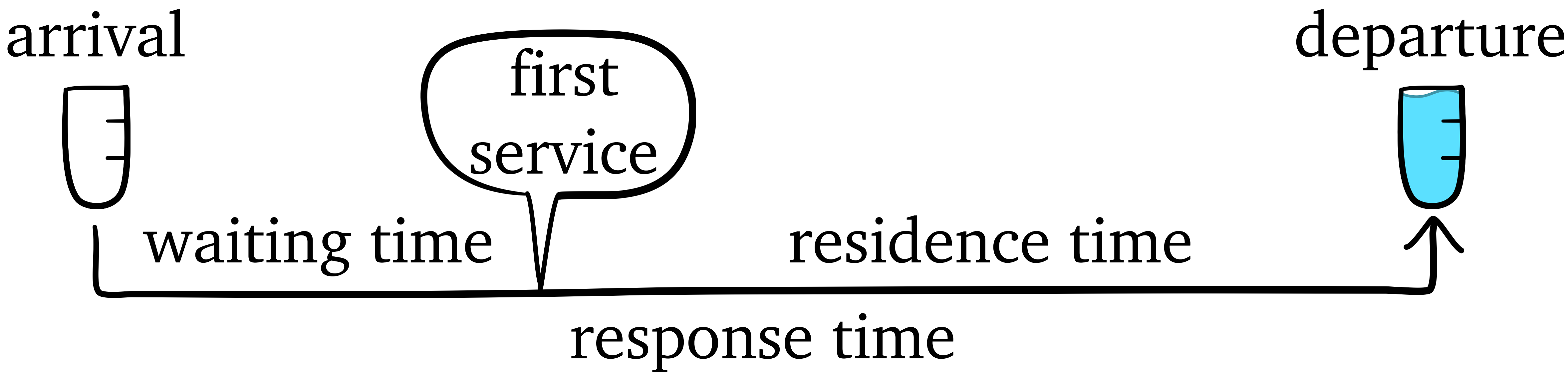
# Response Time Analysis



# Response Time Analysis



# Response Time Analysis



# Residence Time

# Residence Time

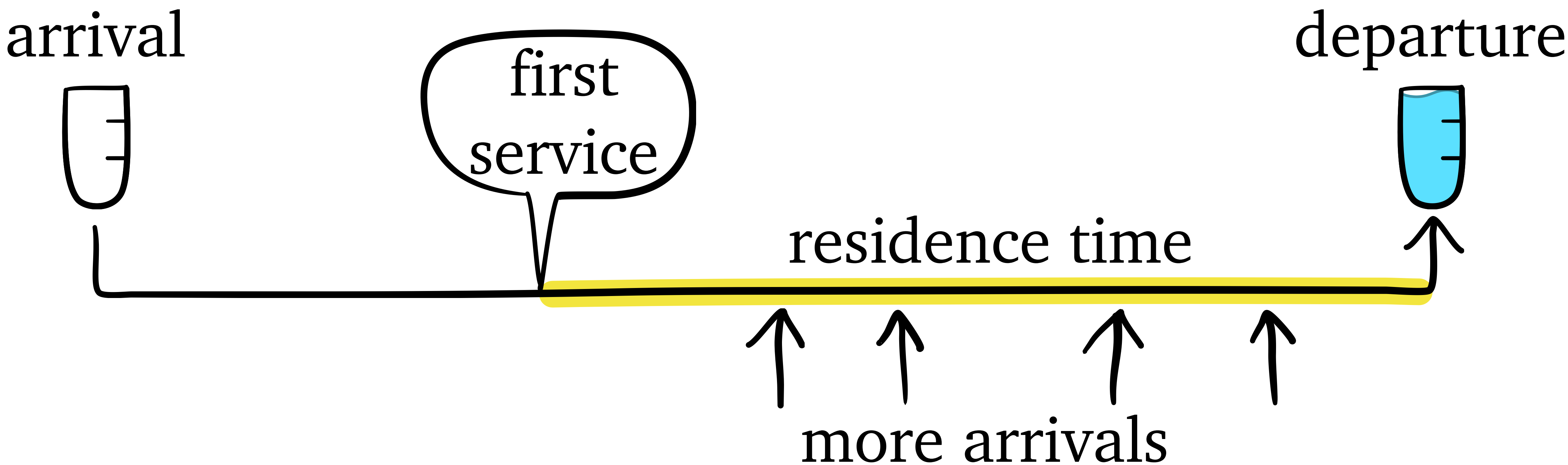

**Question**: is residence time…

# Residence Time


**Question**: is residence time…

- my size?

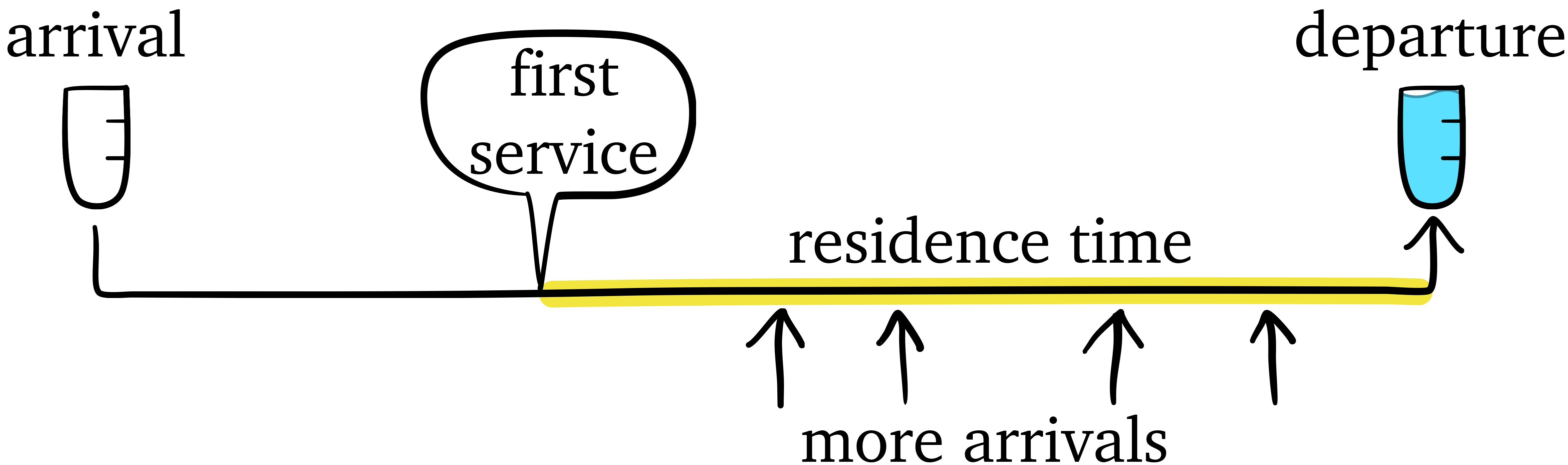
# Residence Time




**Question**: is residence time…

- my size?

# Residence Time




**Question**: is residence time…

- my size? ✗

# Residence Time


**Question**: is residence time…

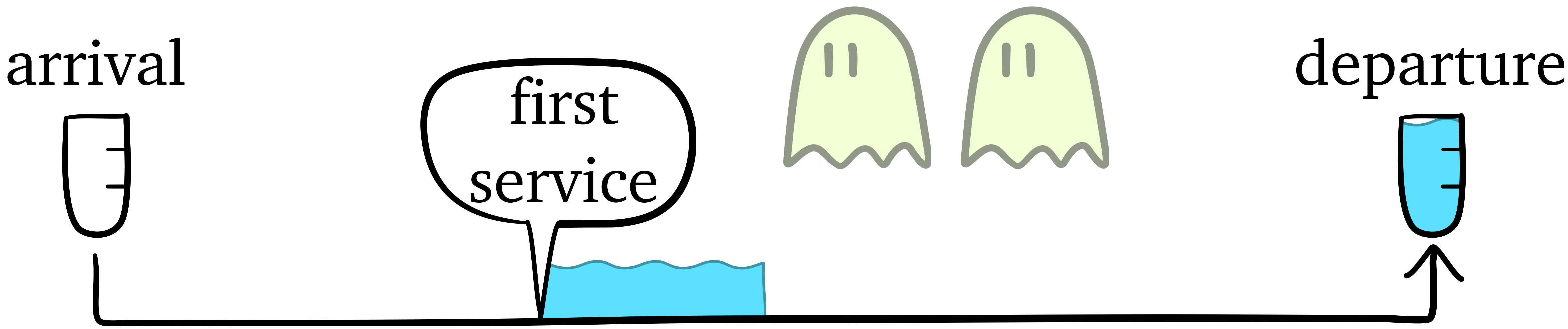
- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$?

# Residence Time


**Question**: is residence time…

- my size? ❌
- $\mathbf{E}[T \mid \text{empty}]$?

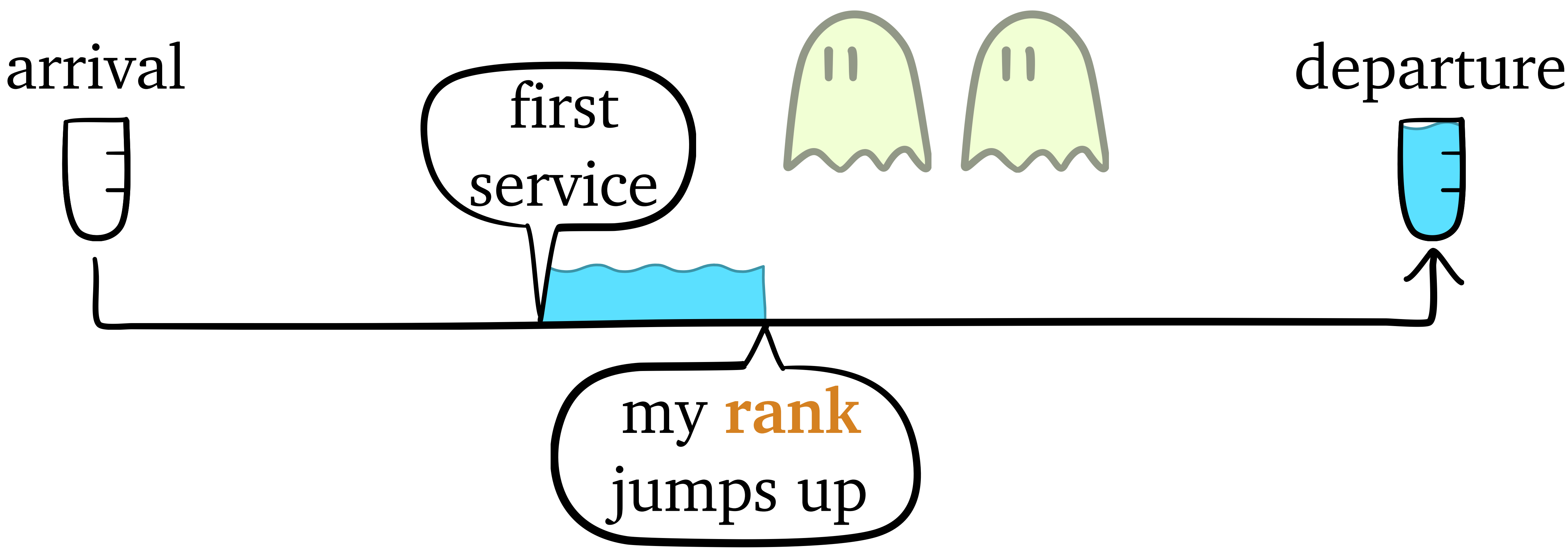
# Residence Time




**Question**: is residence time…

- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$?

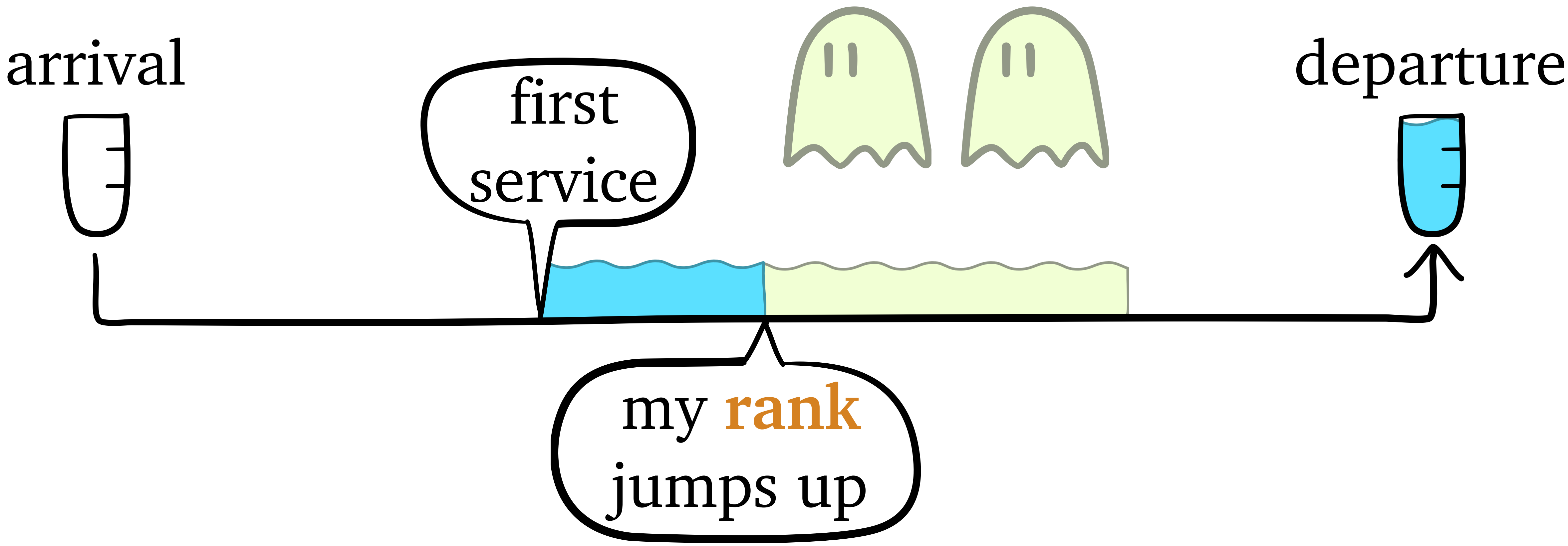
# Residence Time




**Question**: is residence time…

- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$?

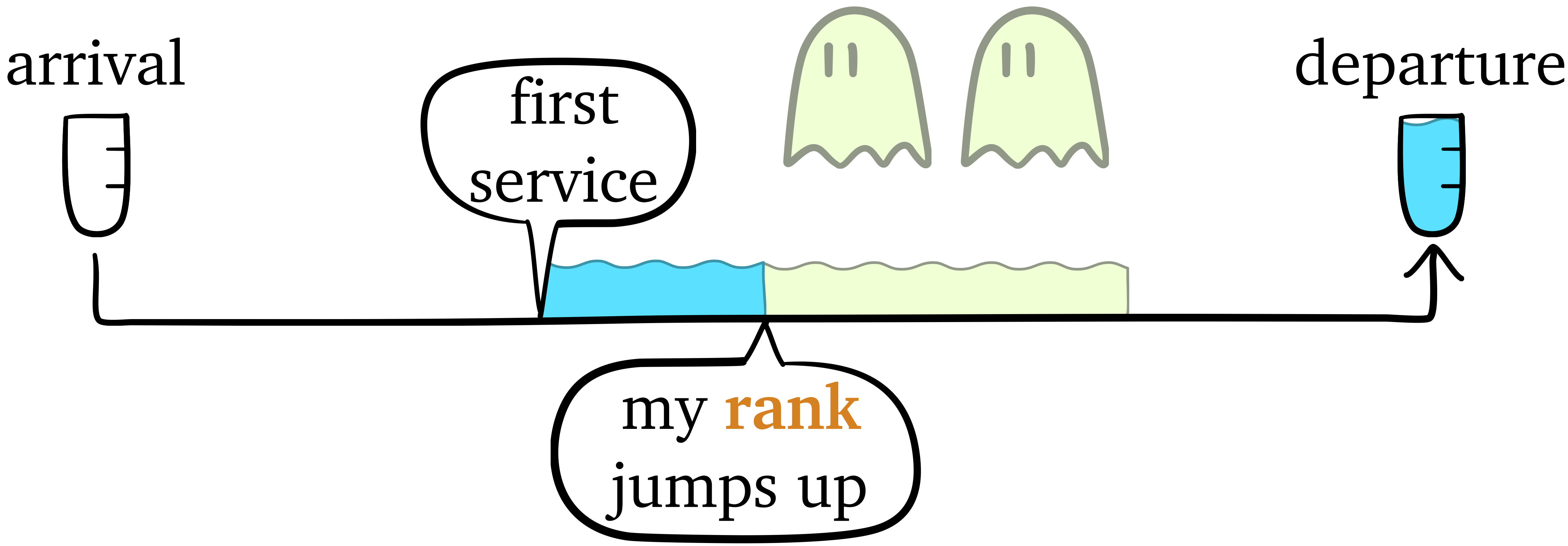
# Residence Time




**Question**: is residence time…

- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$?

# Residence Time

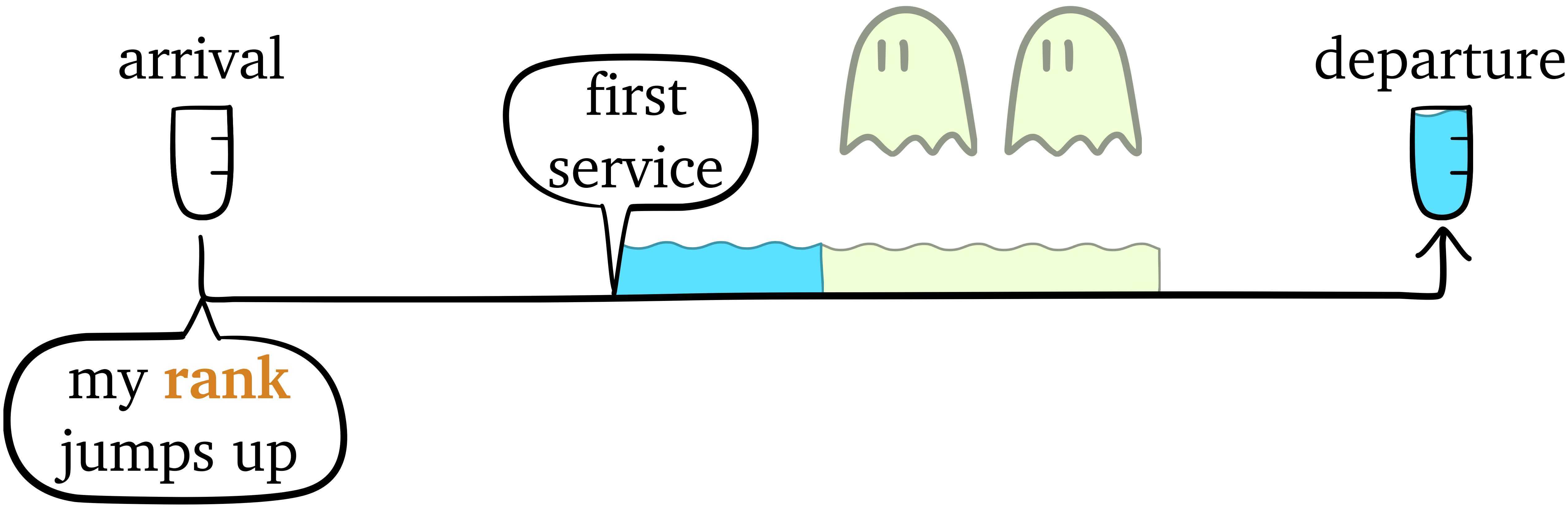



**Question**: is residence time…

- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$?

**Pessimism Principle**:
replace my **rank** with my **worst** future rank

# Residence Time

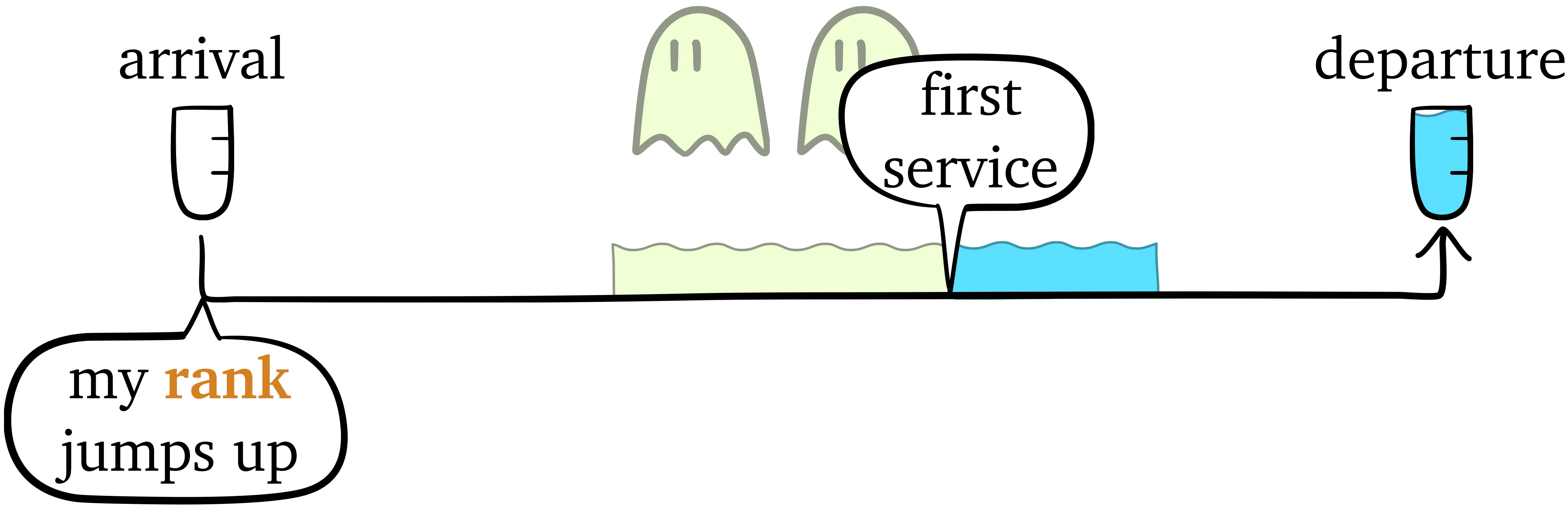



**Question**: is residence time…

- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$?

**Pessimism Principle**:
replace my **rank** with my **worst** future rank

# Residence Time

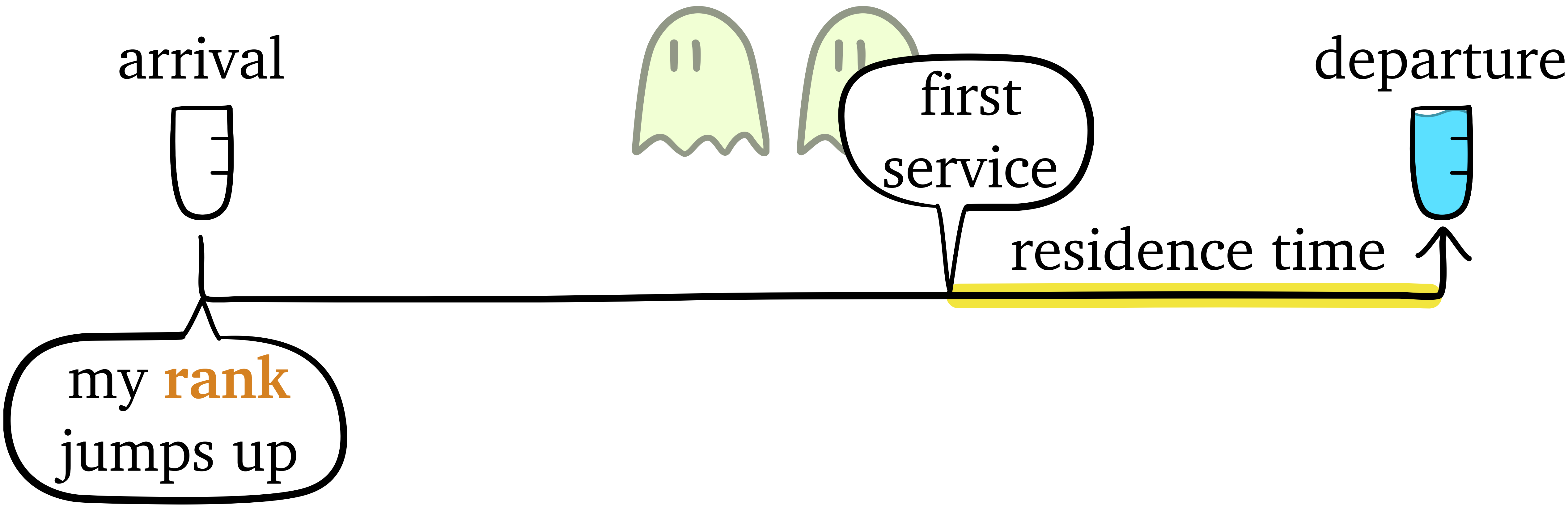



**Question**: is residence time…

- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$?

**Pessimism Principle**: replace my **rank** with my **worst** future rank

# Residence Time

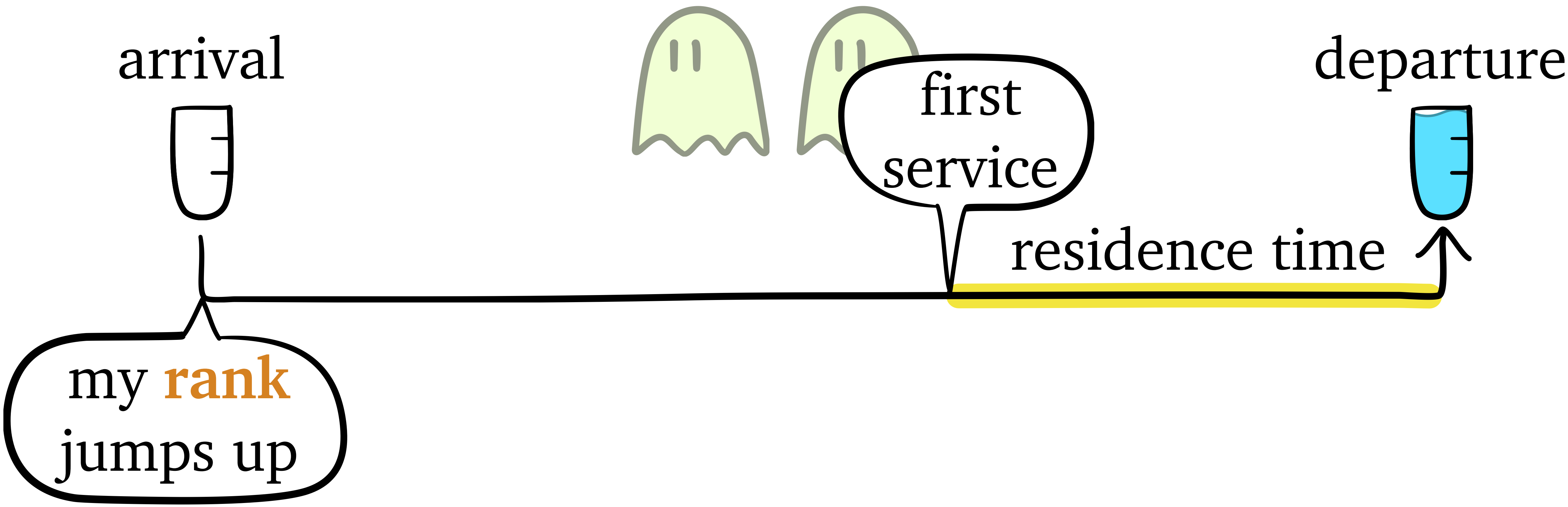



**Question**: is residence time…

- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$?

**Pessimism Principle**:
replace my **rank** with my **worst** future rank

# Residence Time

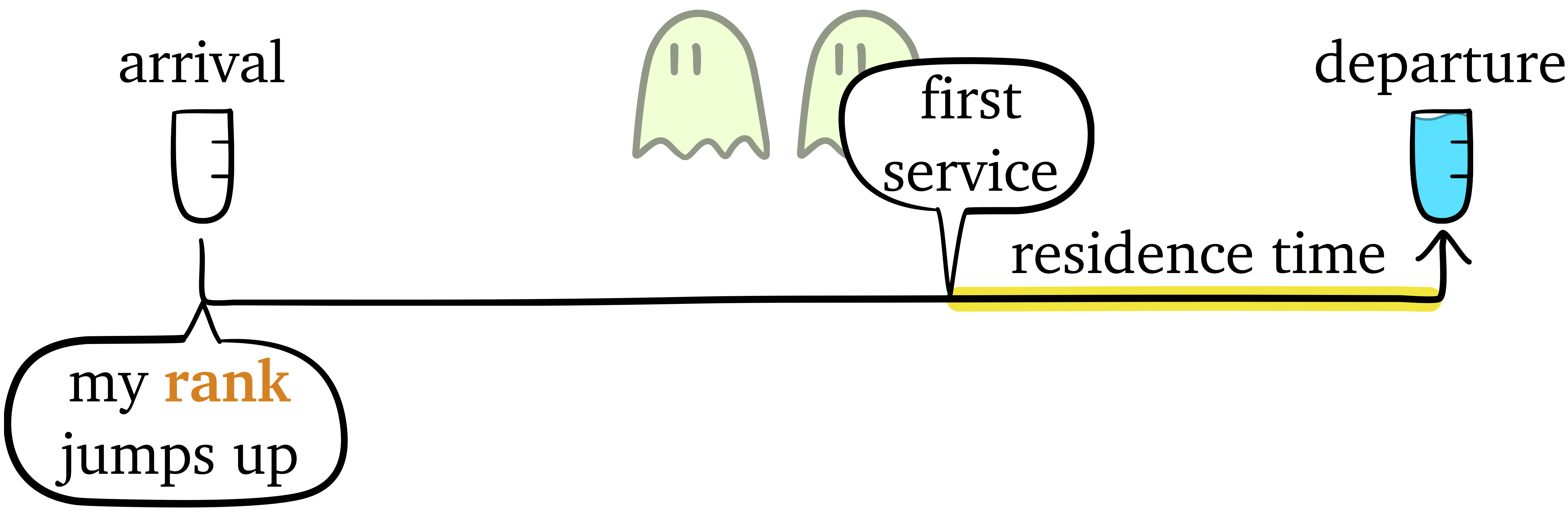



**Question**: is residence time…

- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$? ✓

**Pessimism Principle**: replace my **rank** with my **worst** future rank

# Residence Time



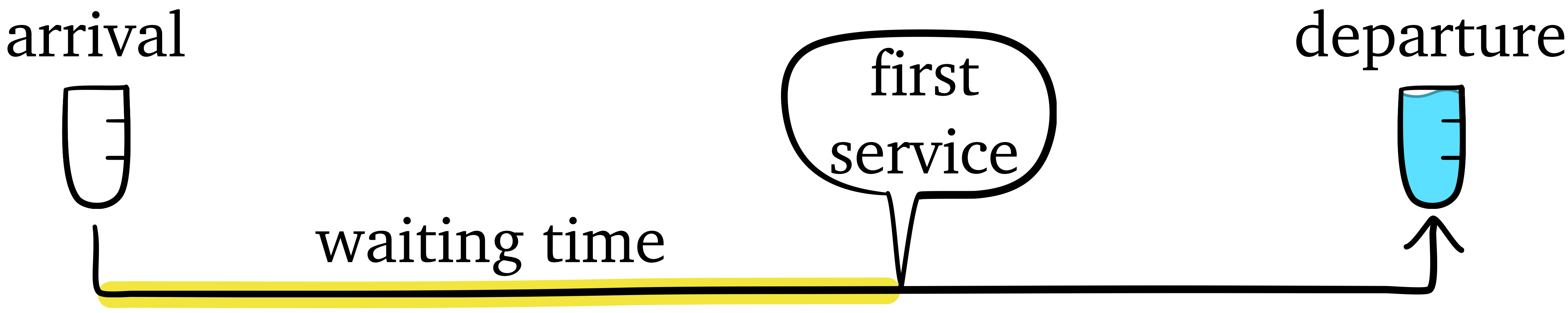

**Question**: is residence time…

- my size? ✗
- $\mathbf{E}[T \mid \text{empty}]$? ✓

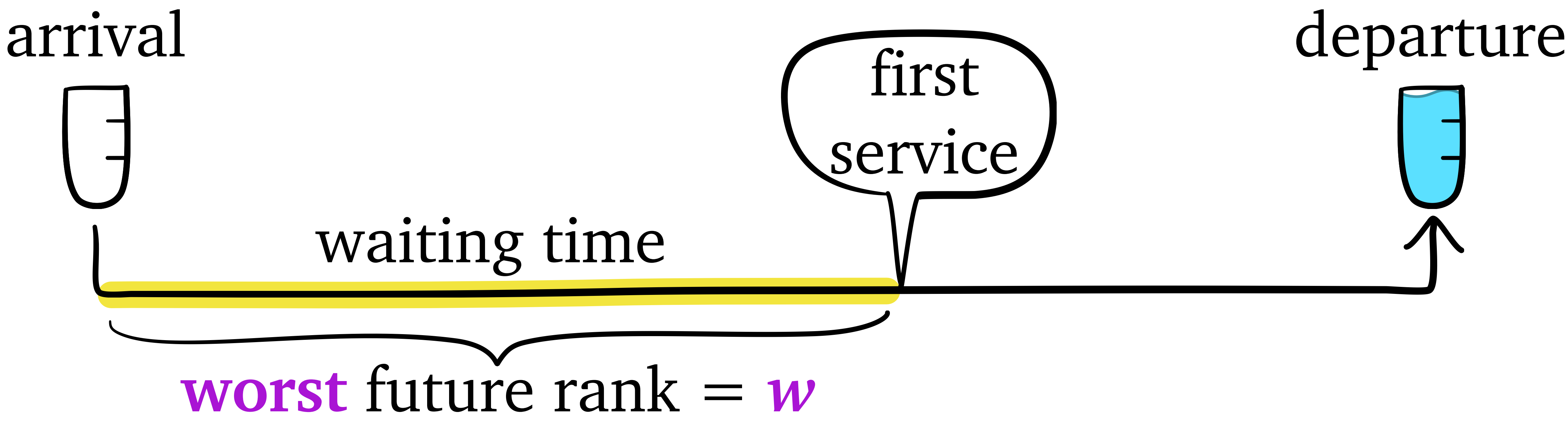
**Pessimism Principle**: replace my **rank** with my **worst** future rank

$$\text{e.g.}\quad \mathbf{E}[R_{14}] = \mathbf{E}[T_{14} \mid \text{empty}] = \int_0^{14} \frac{\mathrm{d}a}{1-\rho_{\text{new}}(a)}$$

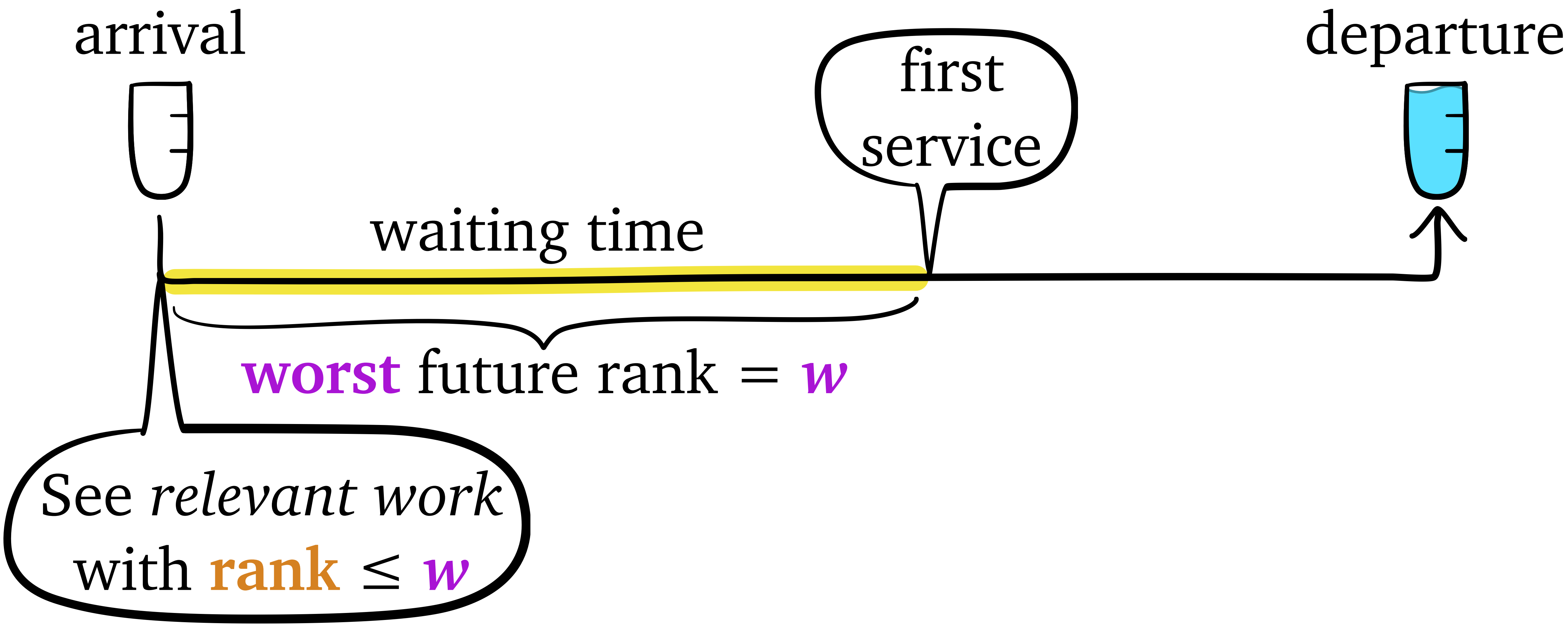
# Waiting Time



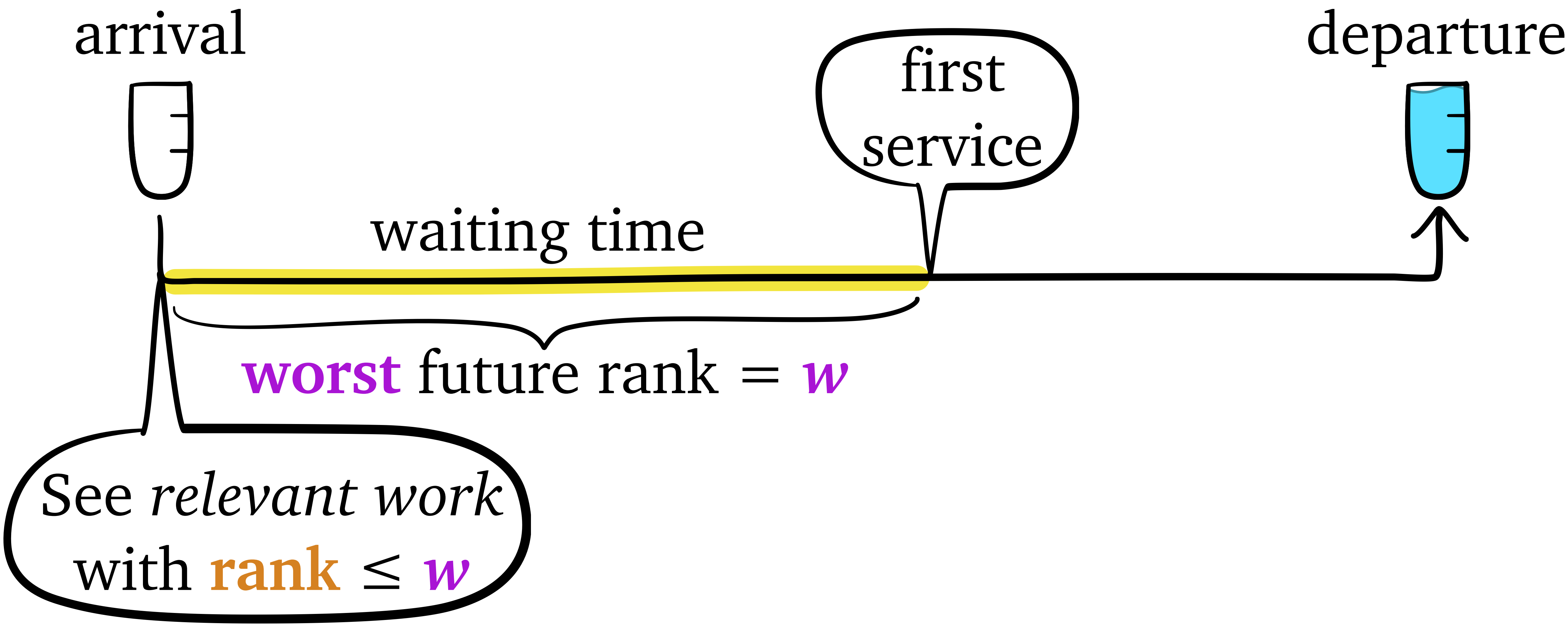
# Waiting Time



# Waiting Time

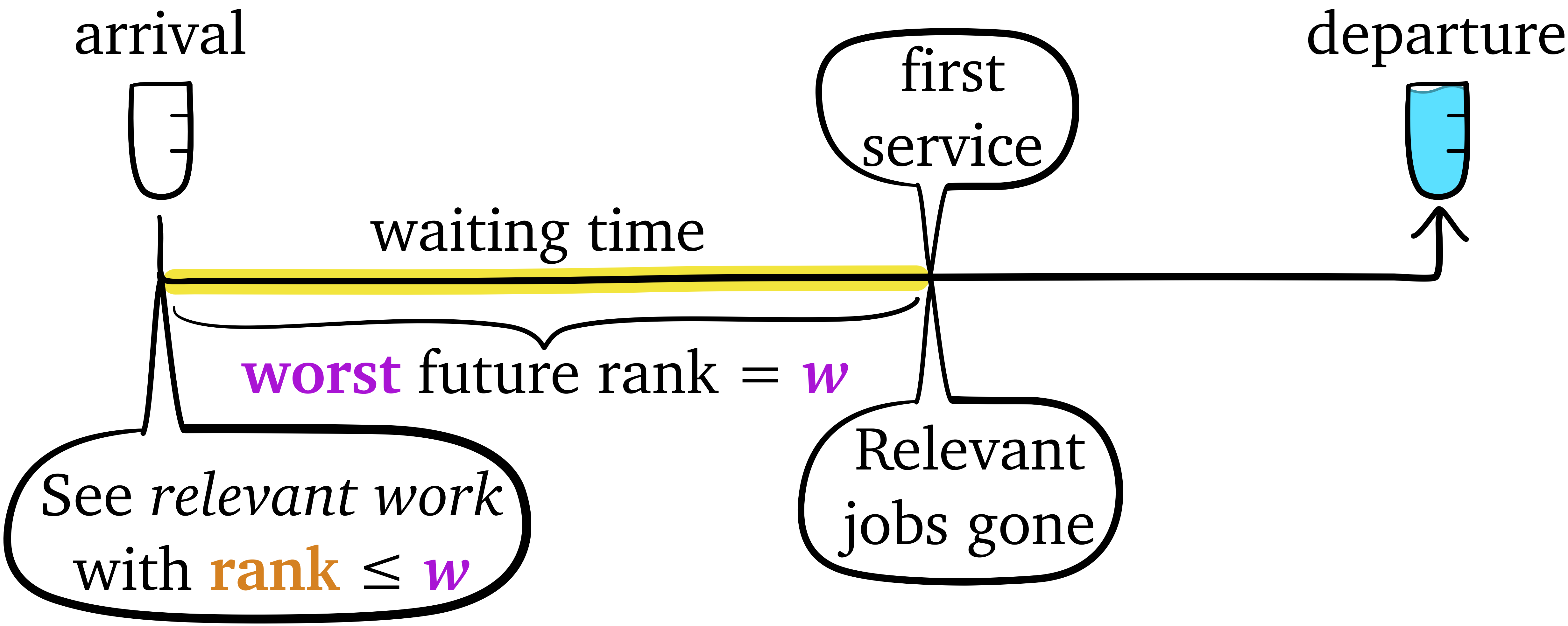


# Waiting Time



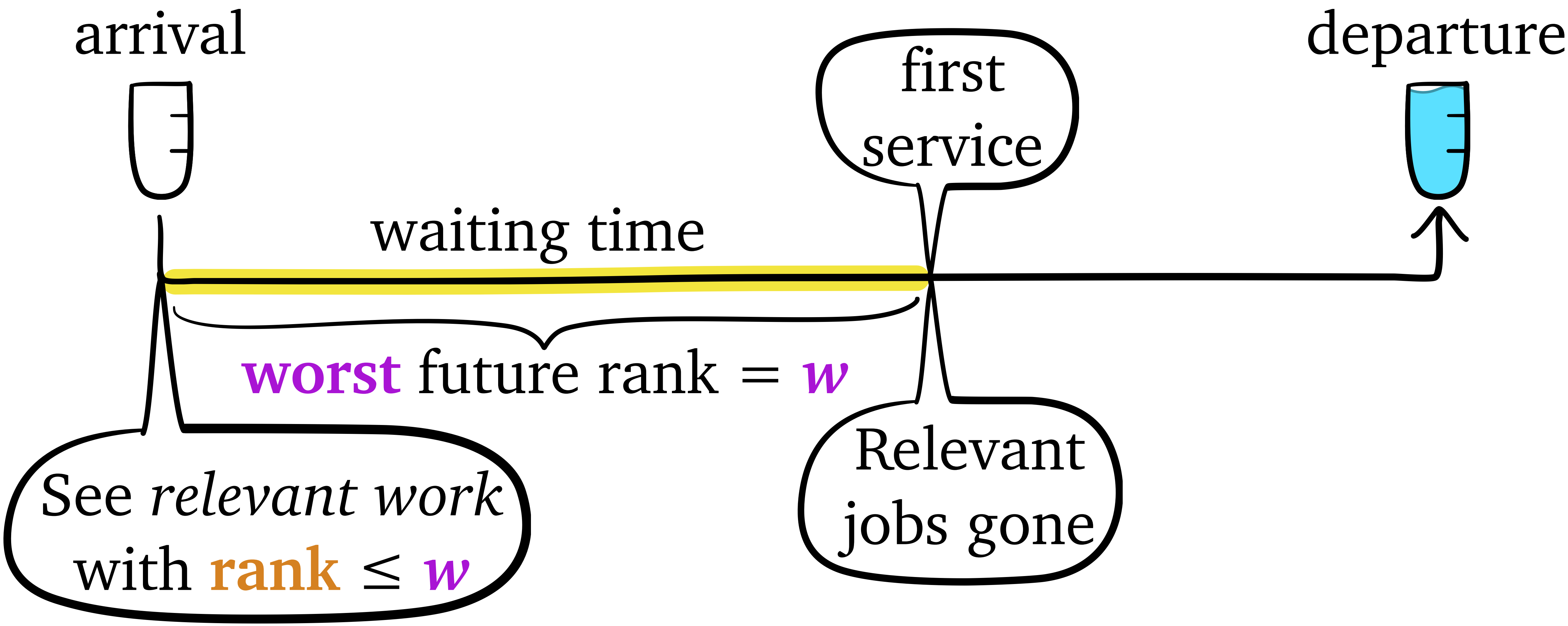

$U[w]$ = *relevant work*

# Waiting Time




$U[w]$ = *relevant work*

# Waiting Time

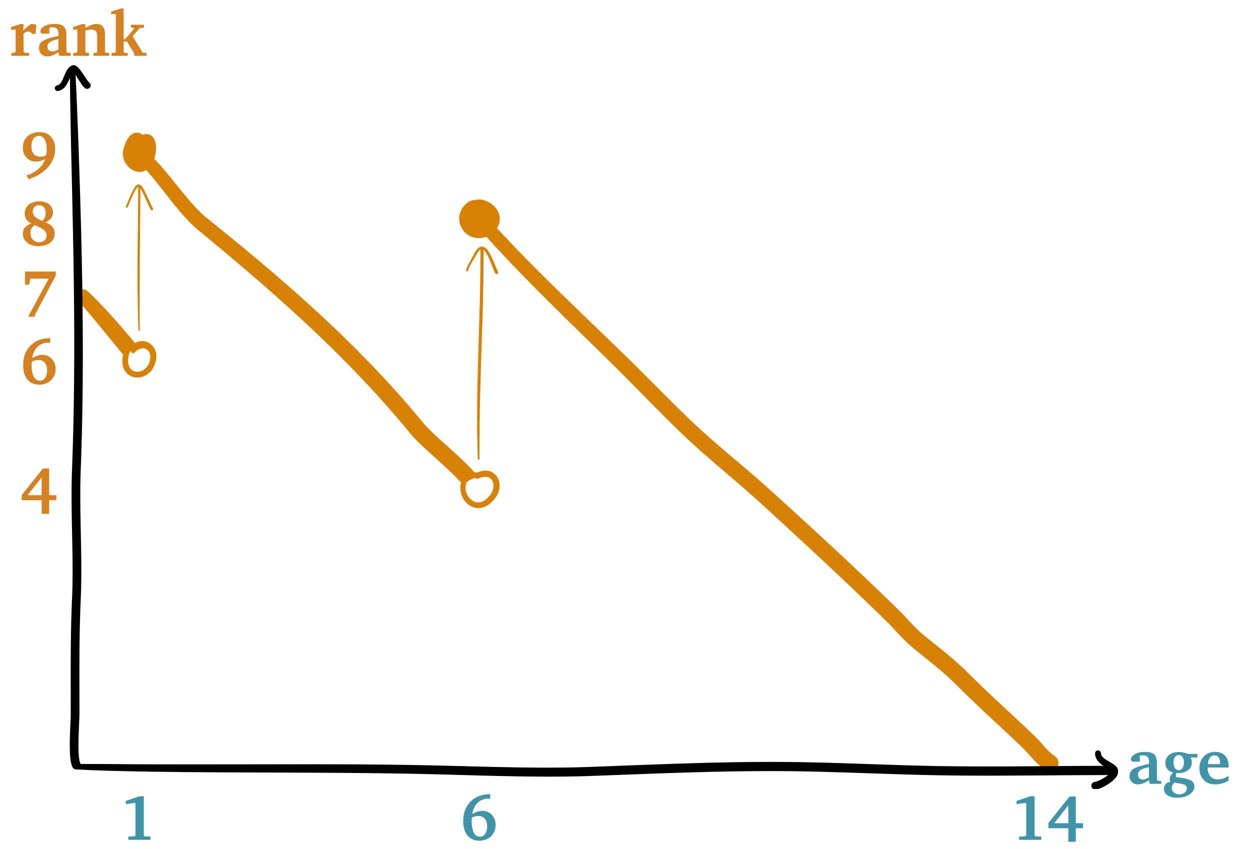



$U[w]$ = *relevant work*

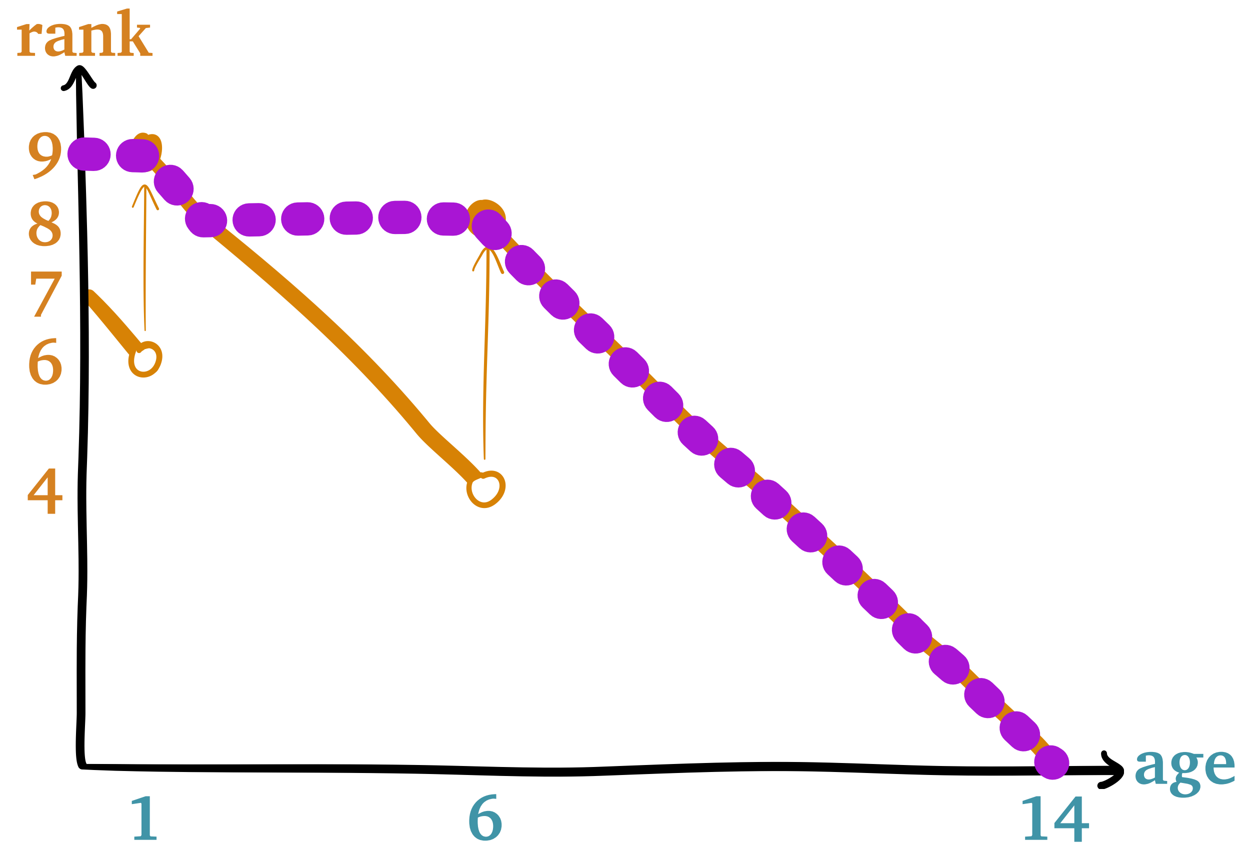
Waiting time is *busy period* started by $U[w]$

# Response Time: Size 14

# Response Time: Size 14

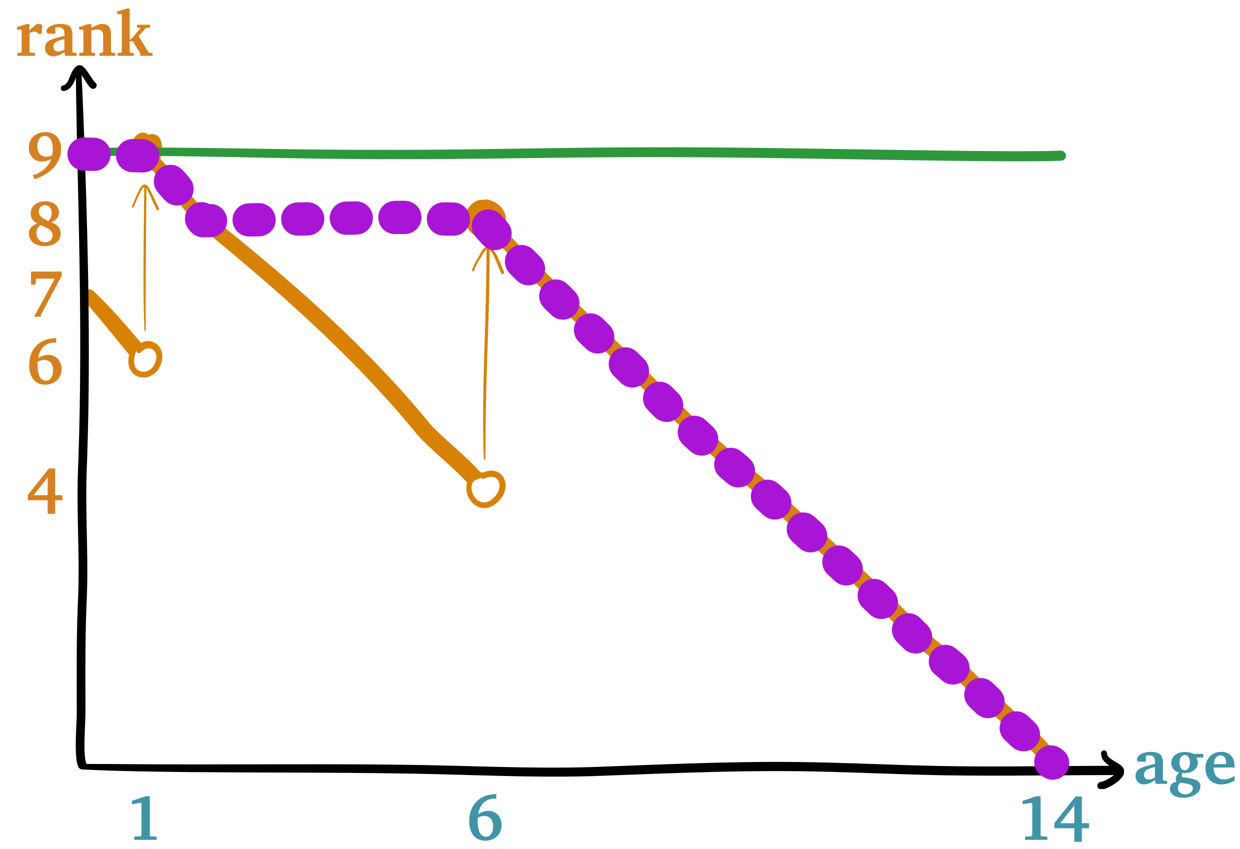


# Response Time: Size 14



# Response Time: Size 14

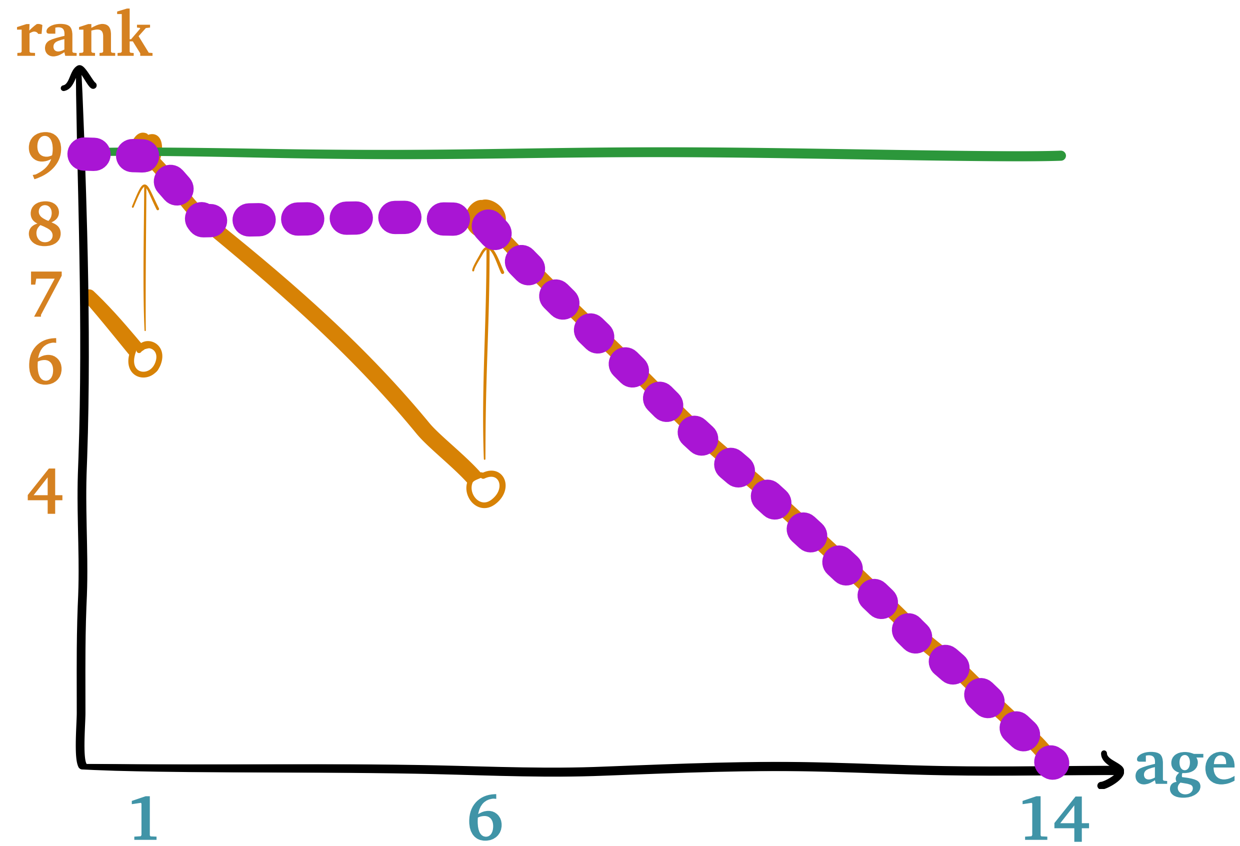
**Relevant work** ($w$ = 9):



# Response Time: Size 14

**Relevant work** ($w = 9$):

$$\mathbf{E}[U[9]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X^2]}{1-\rho}$$



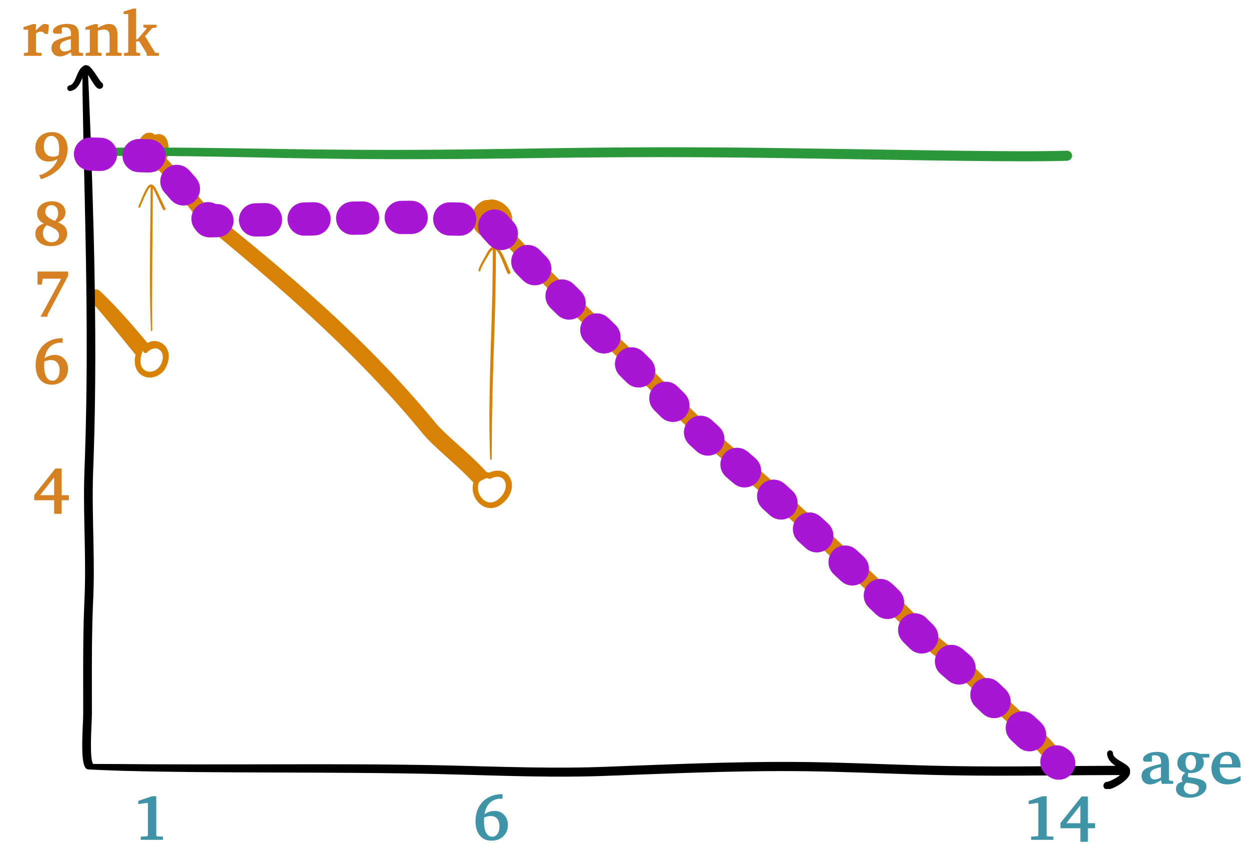
# Response Time: Size 14

**Relevant work** ($w = 9$):

$$\mathbf{E}[U[9]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X^2]}{1-\rho}$$

**Waiting time:**

$$\mathbf{E}[Q_{14}] = \frac{\mathbf{E}[U[9]]}{1-\rho_{\text{new}}(0)}$$



# Response Time: Size 14

**Relevant work** ($w = 9$):

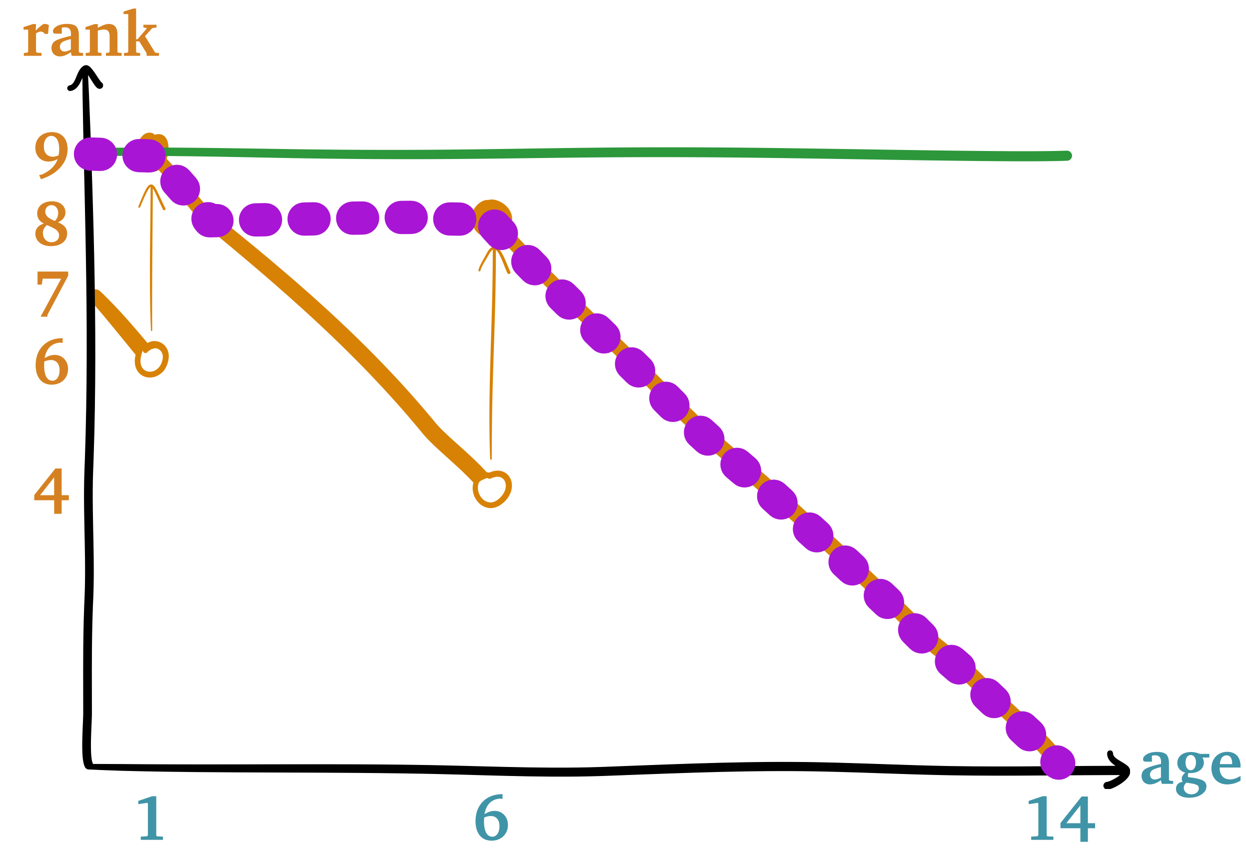
$$\mathbf{E}[U[9]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X^2]}{1-\rho}$$

**Waiting time:**

$$\mathbf{E}[Q_{14}] = \frac{\mathbf{E}[U[9]]}{1-\rho_{\text{new}}(0)}$$

**Residence time:**

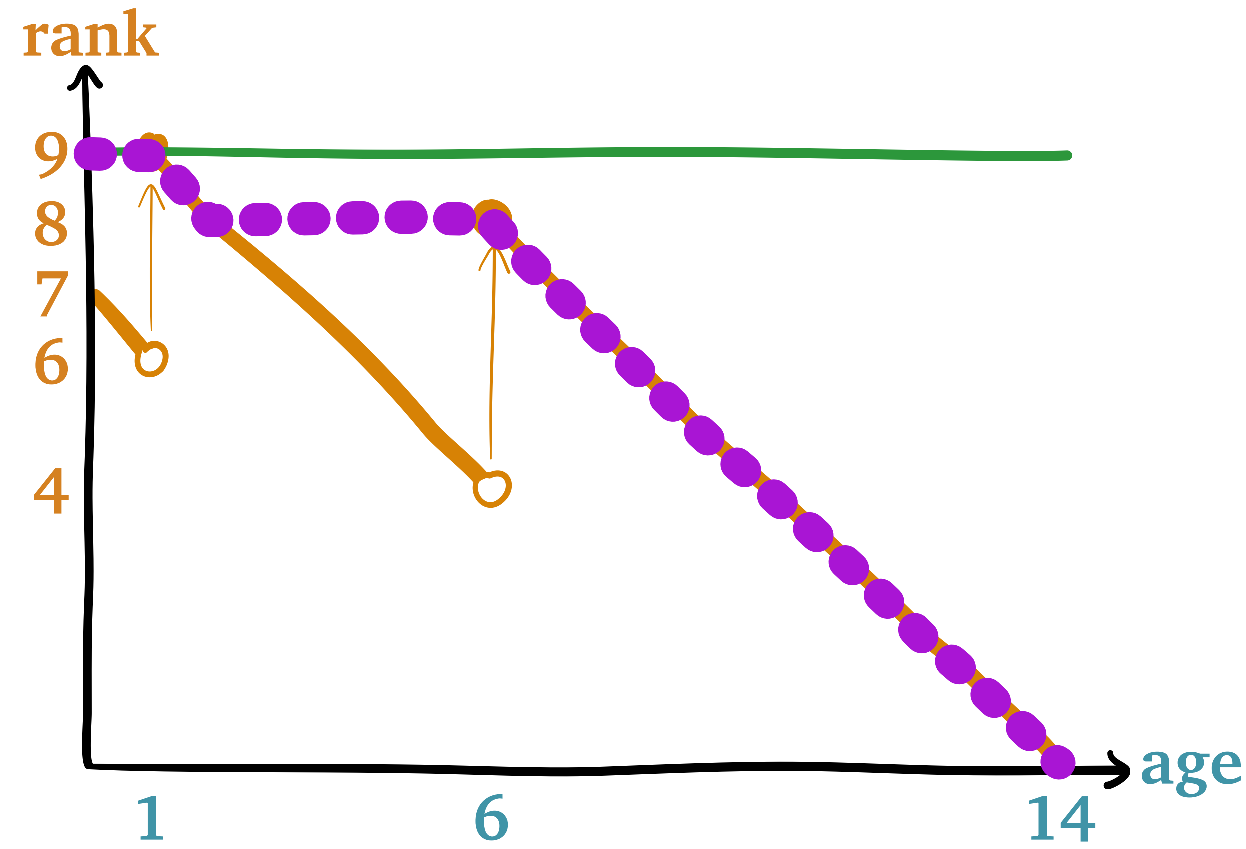
$$\mathbf{E}[R_{14}] = \int_0^{14} \frac{\mathrm{d}a}{1-\rho_{\text{new}}(a)}$$



# Response Time: Size 14

**Relevant work** ($w = 9$):

$$\mathbf{E}[U[9]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X^2]}{1-\rho}$$




**Waiting time:**

$$\mathbf{E}[Q_{14}] = \frac{\mathbf{E}[U[9]]}{1-\rho_{\text{new}}(0)}$$

**Residence time:**

$$\mathbf{E}[R_{14}] = \int_0^{14} \frac{\mathrm{d}a}{1-\rho_{\text{new}}(a)}$$

**Response time:**

$$\mathbf{E}[T_{14}] = \mathbf{E}[Q_{14}] + \mathbf{E}[R_{14}]$$

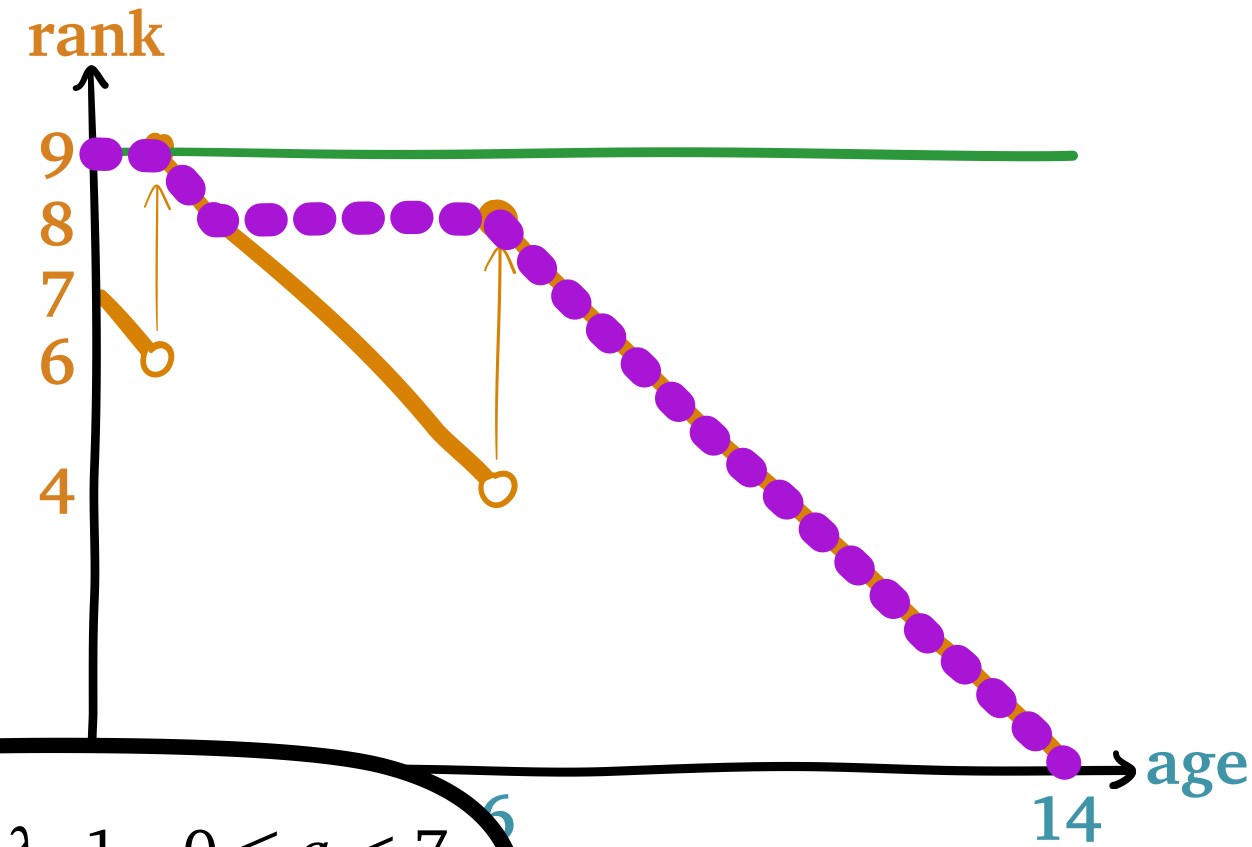
# Response Time: Size 14

**Relevant work** ($w = 9$):

$$\mathbf{E}[U[9]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X^2]}{1-\rho}$$

**Waiting time:**

$$\mathbf{E}[Q_{14}] = \frac{\mathbf{E}[U[9]]}{1-\rho_{\text{new}}(0}$$




$$\rho_{\text{new}}(a) = \begin{cases} \lambda \cdot 1 & 0 \le a < 7 \\ \lambda \cdot 0 & 7 \le a < 14 \end{cases}$$

**Residence time:**

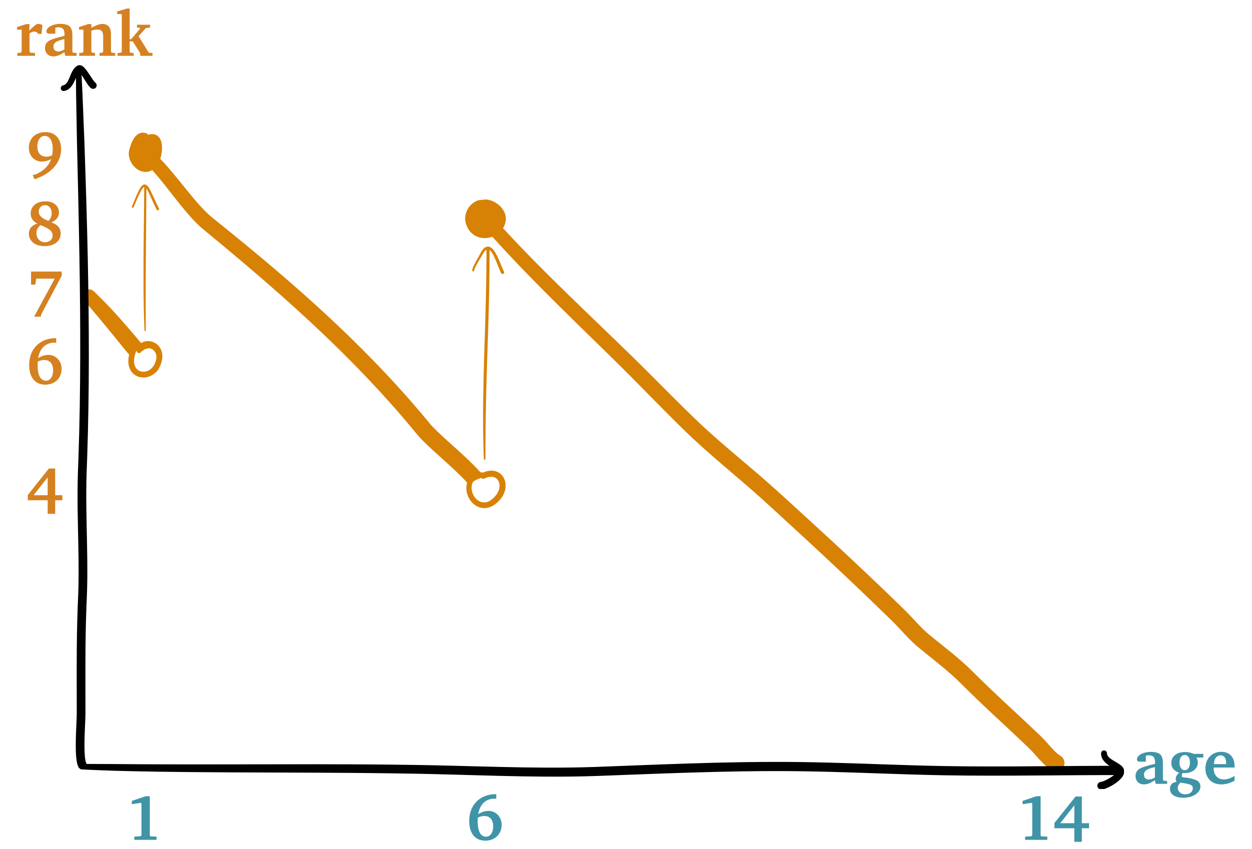
$$\mathbf{E}[R_{14}] = \int_0^{14} \frac{\mathrm{d}a}{1-\rho_{\text{new}}(a)}$$

**time:**

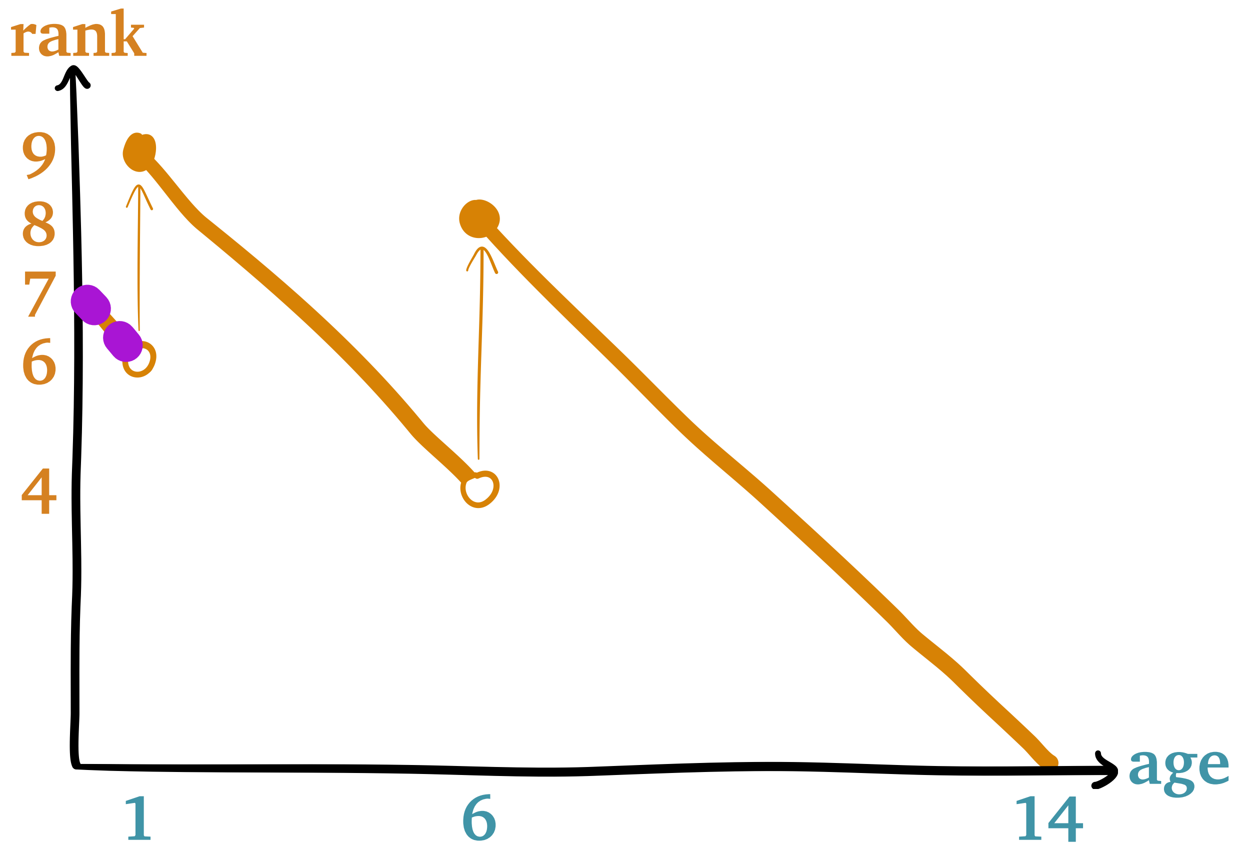
$$\mathbf{E}[T_{14}] = \mathbf{E}[Q_{14}] + \mathbf{E}[R_{14}]$$

# Response Time: Size 1

# Response Time: Size 1

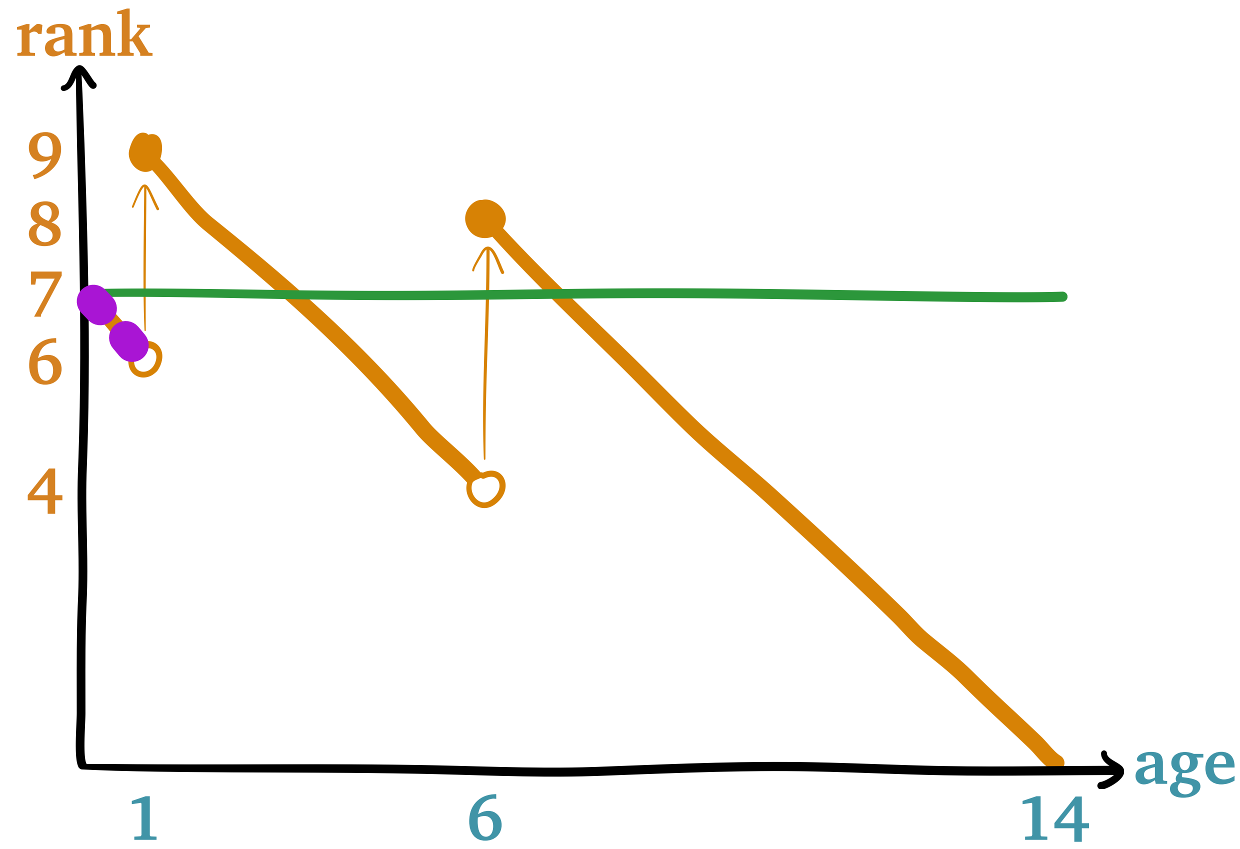


# Response Time: Size 1



# Response Time: Size 1

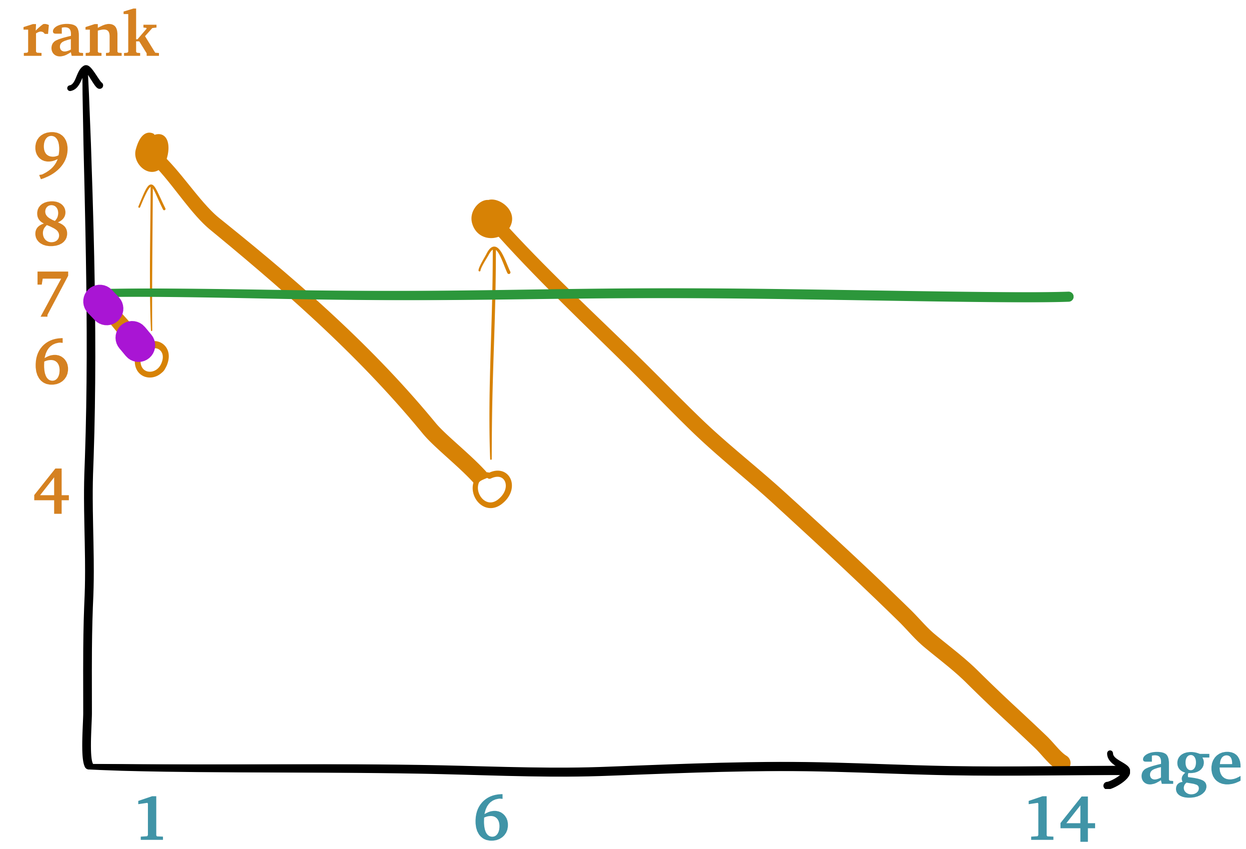
**Relevant work** ($w = 7$):



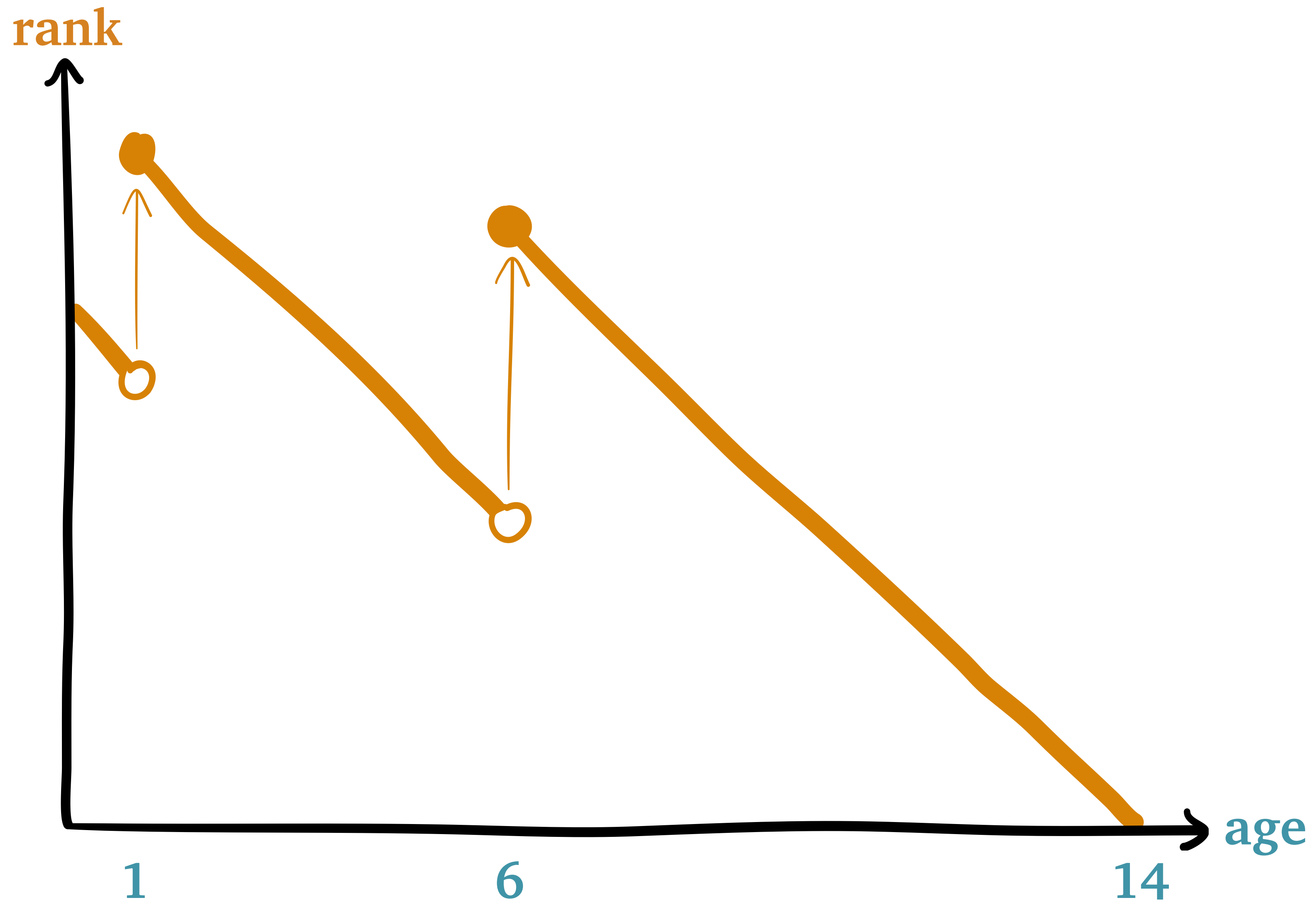
# Response Time: Size 1

**Relevant work** ($w = 7$):

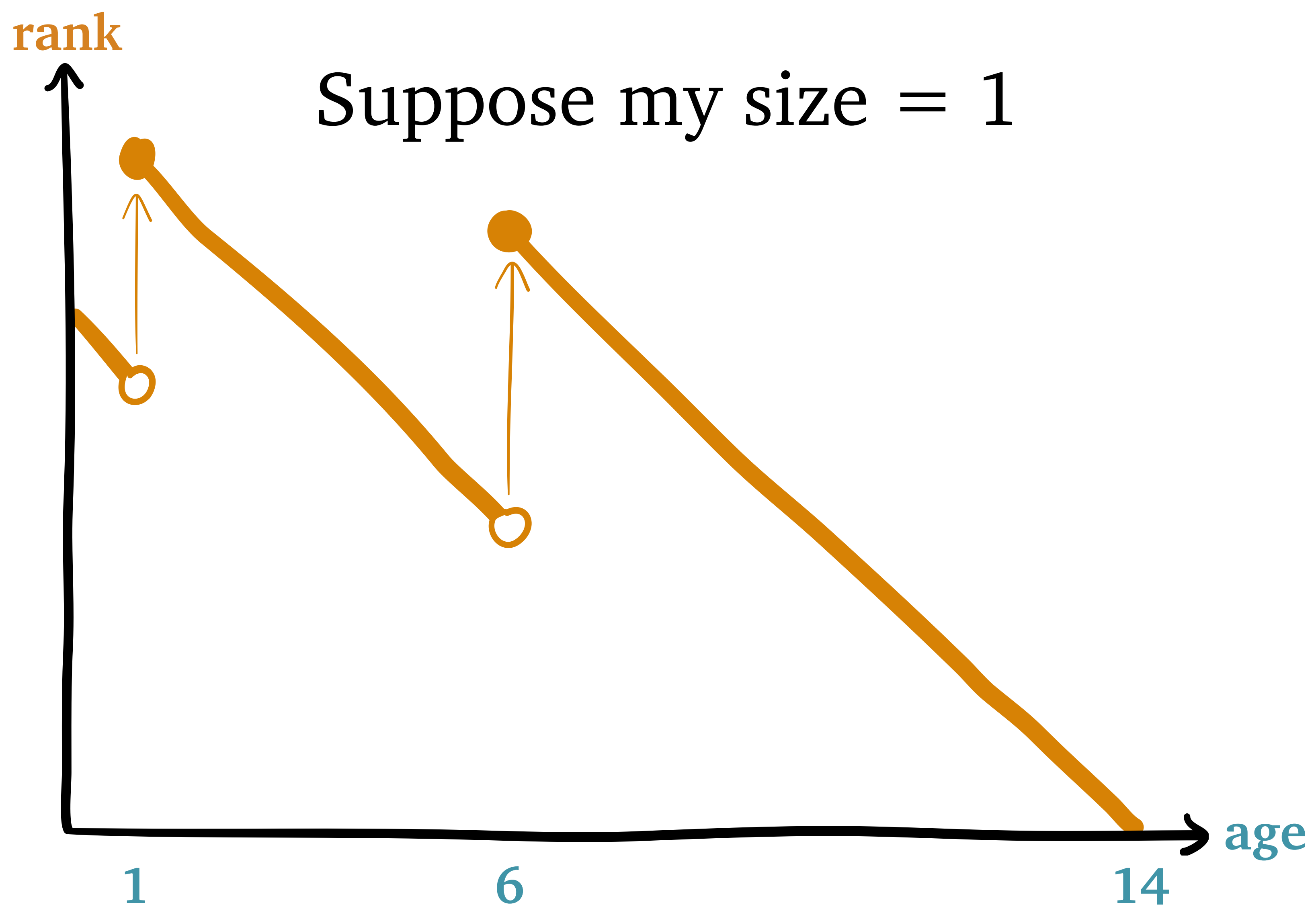
$\mathbf{E}[U[7]] = ???$



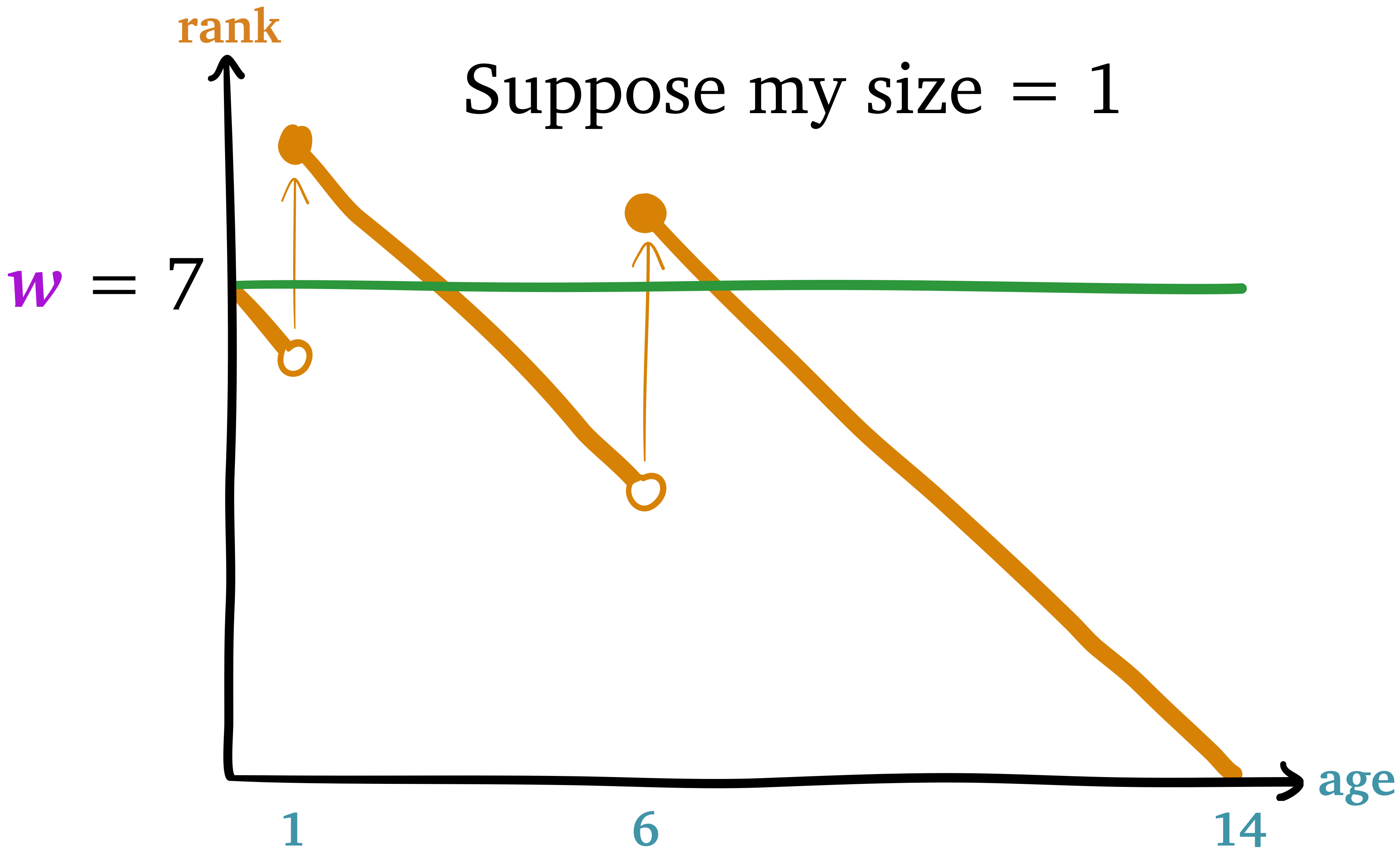
# Relevant Work



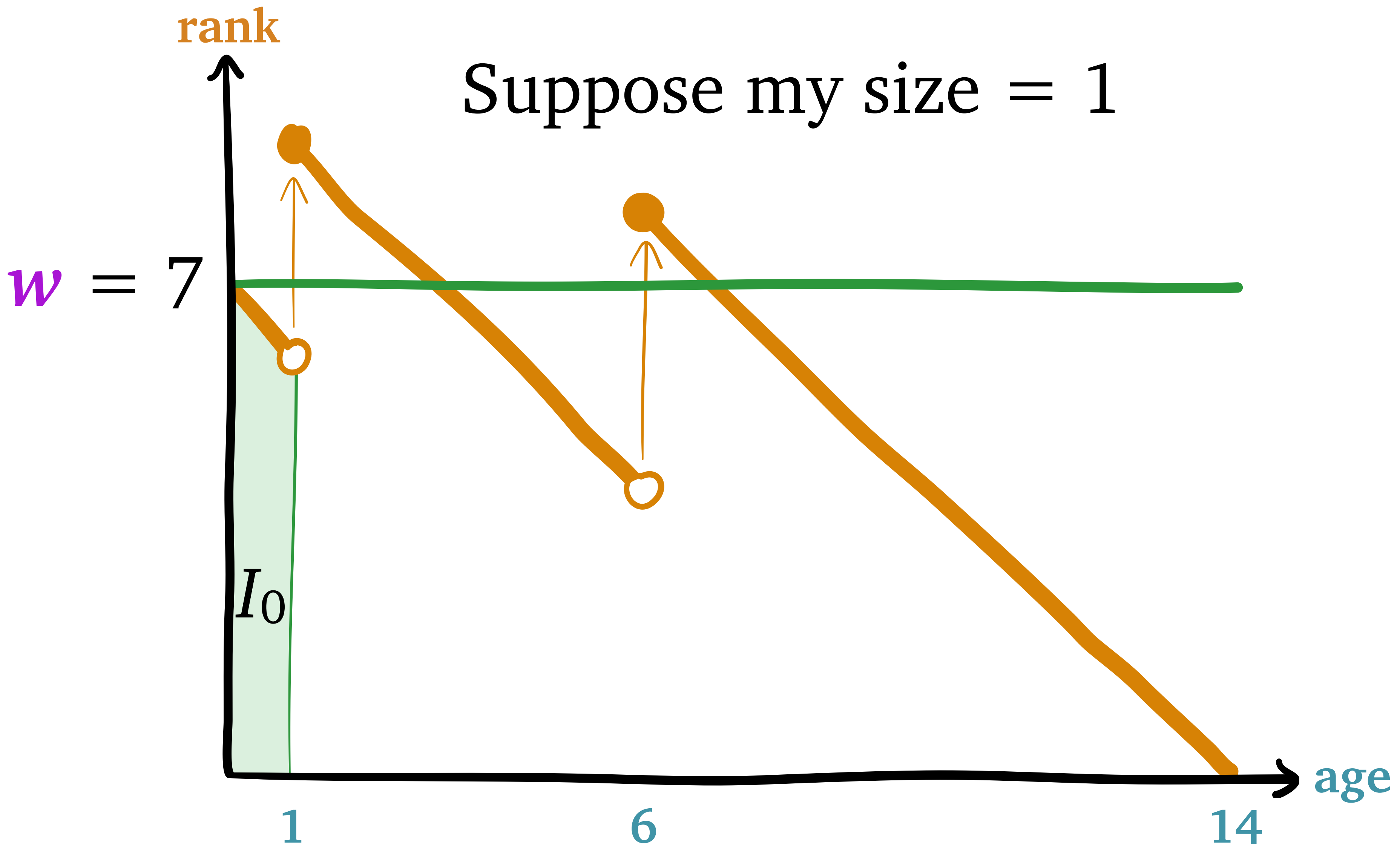
# Relevant Work



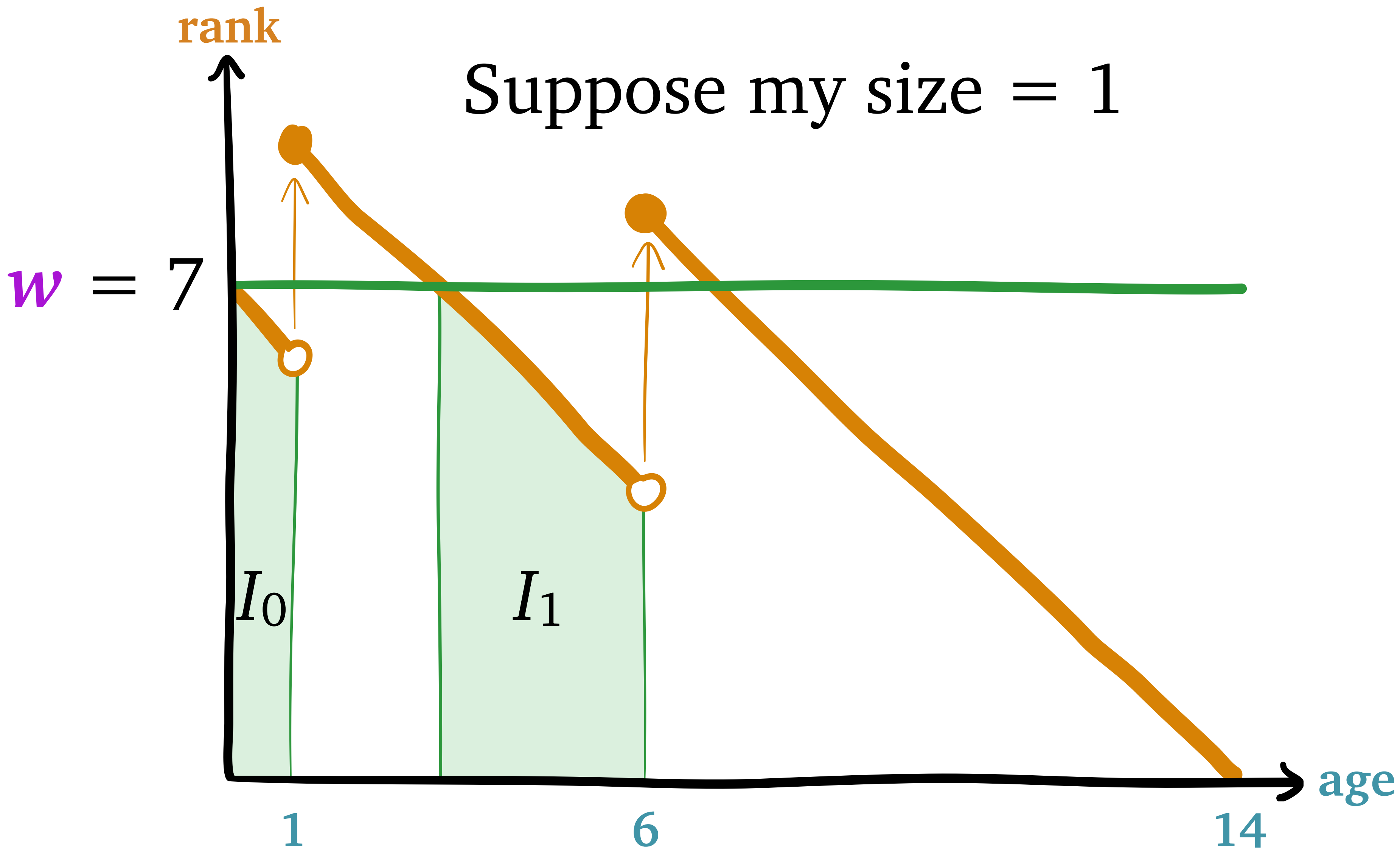
# Relevant Work



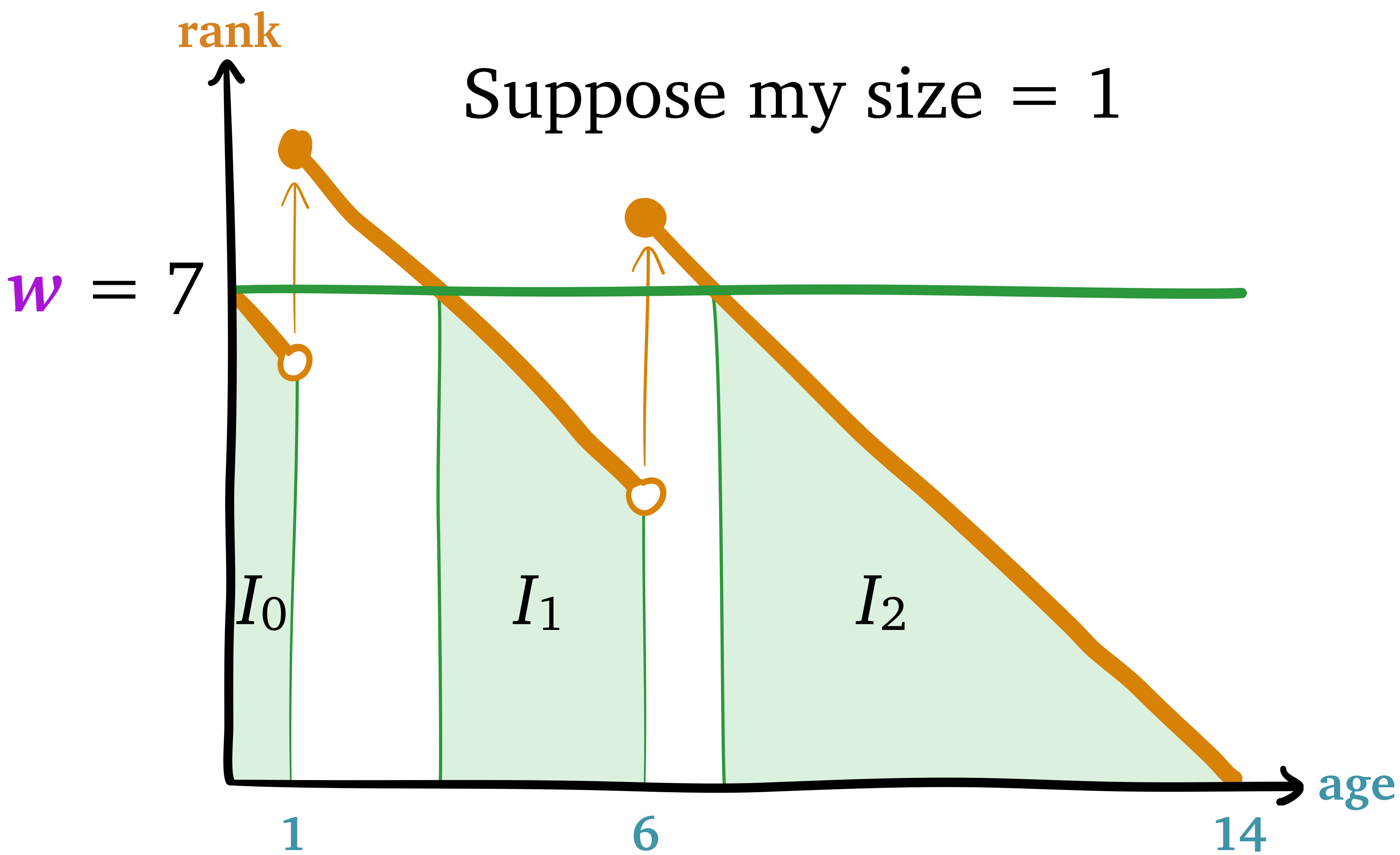
# Relevant Work



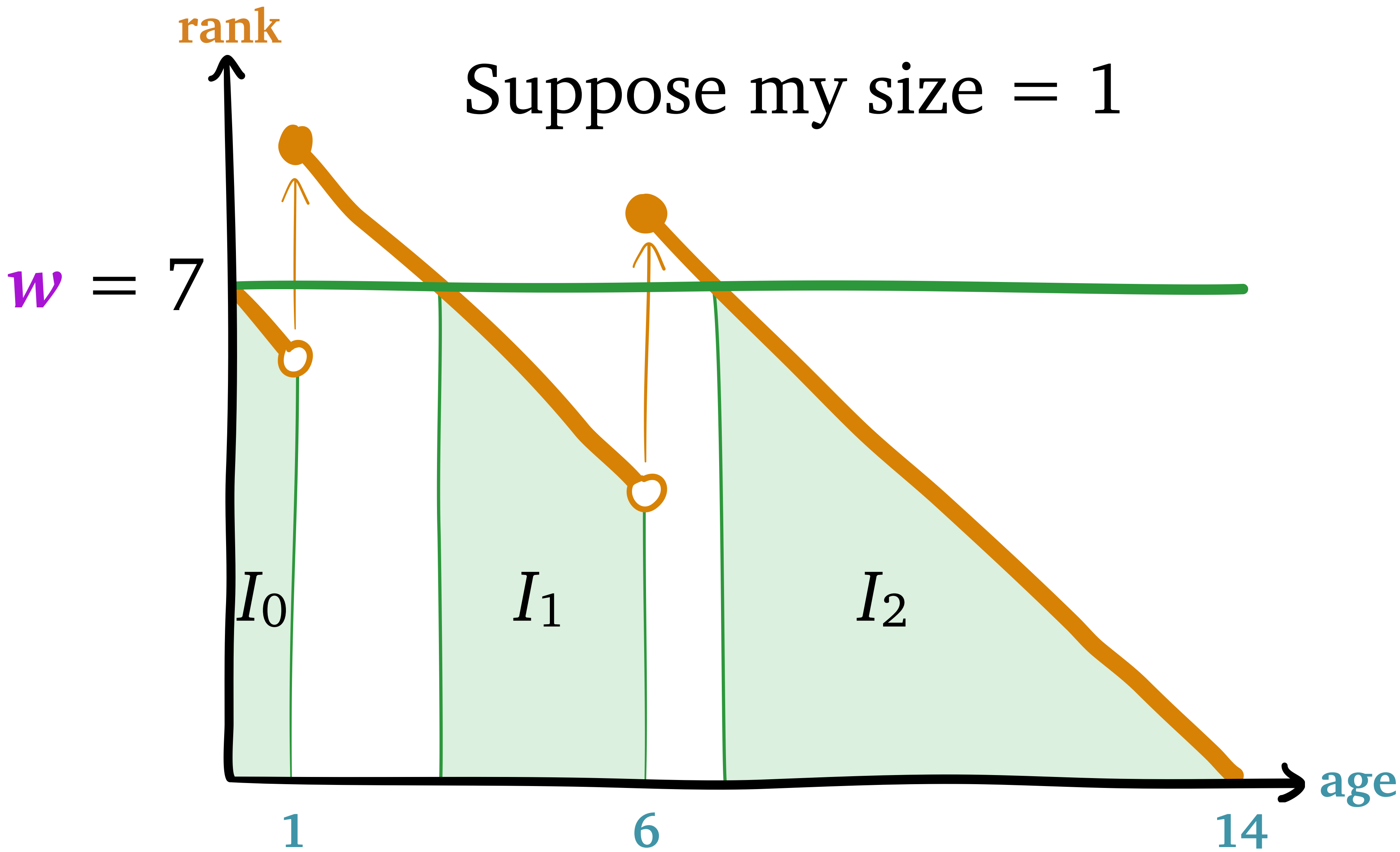
# Relevant Work



# Relevant Work

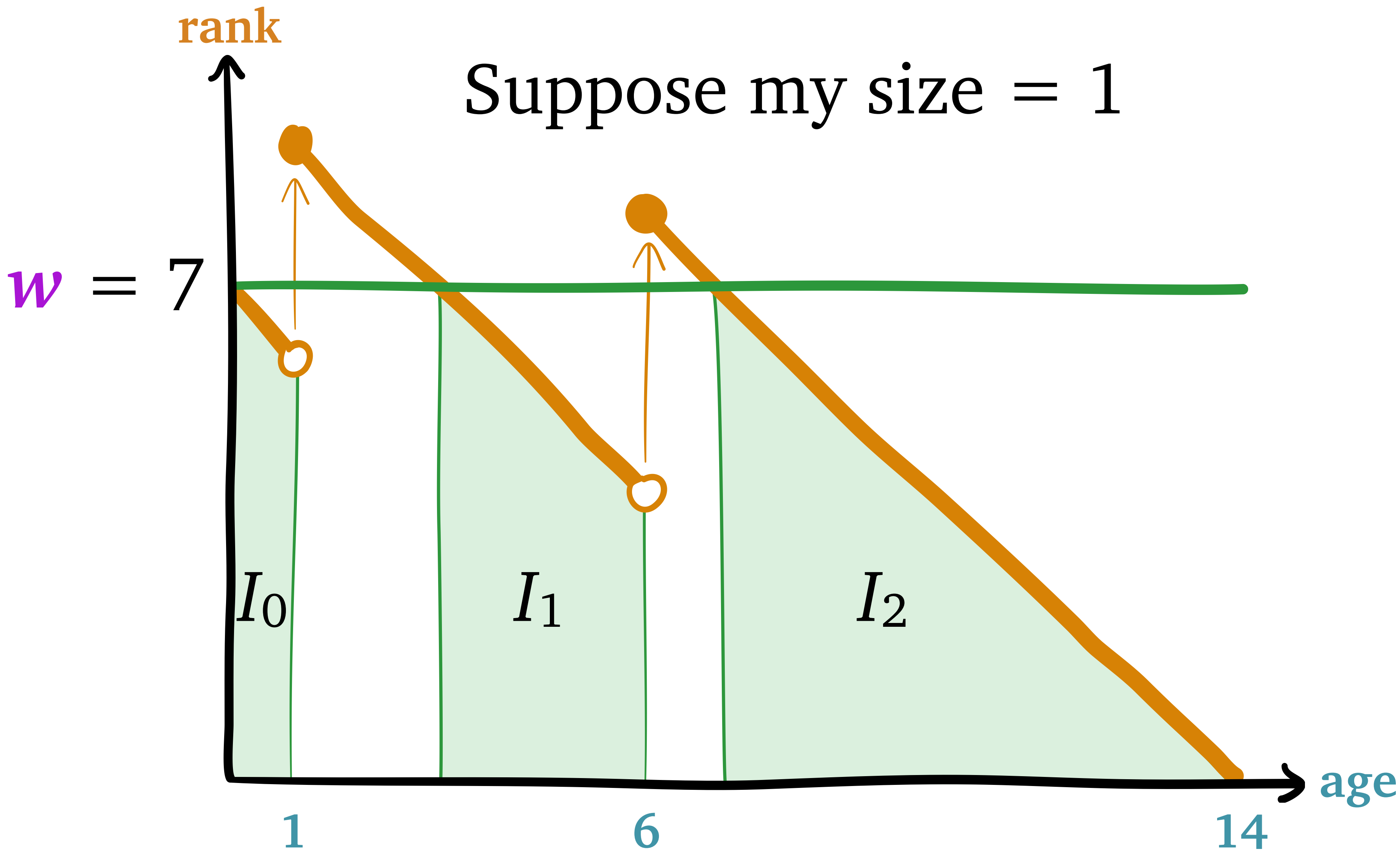


# Relevant Work




Two causes of relevant work:

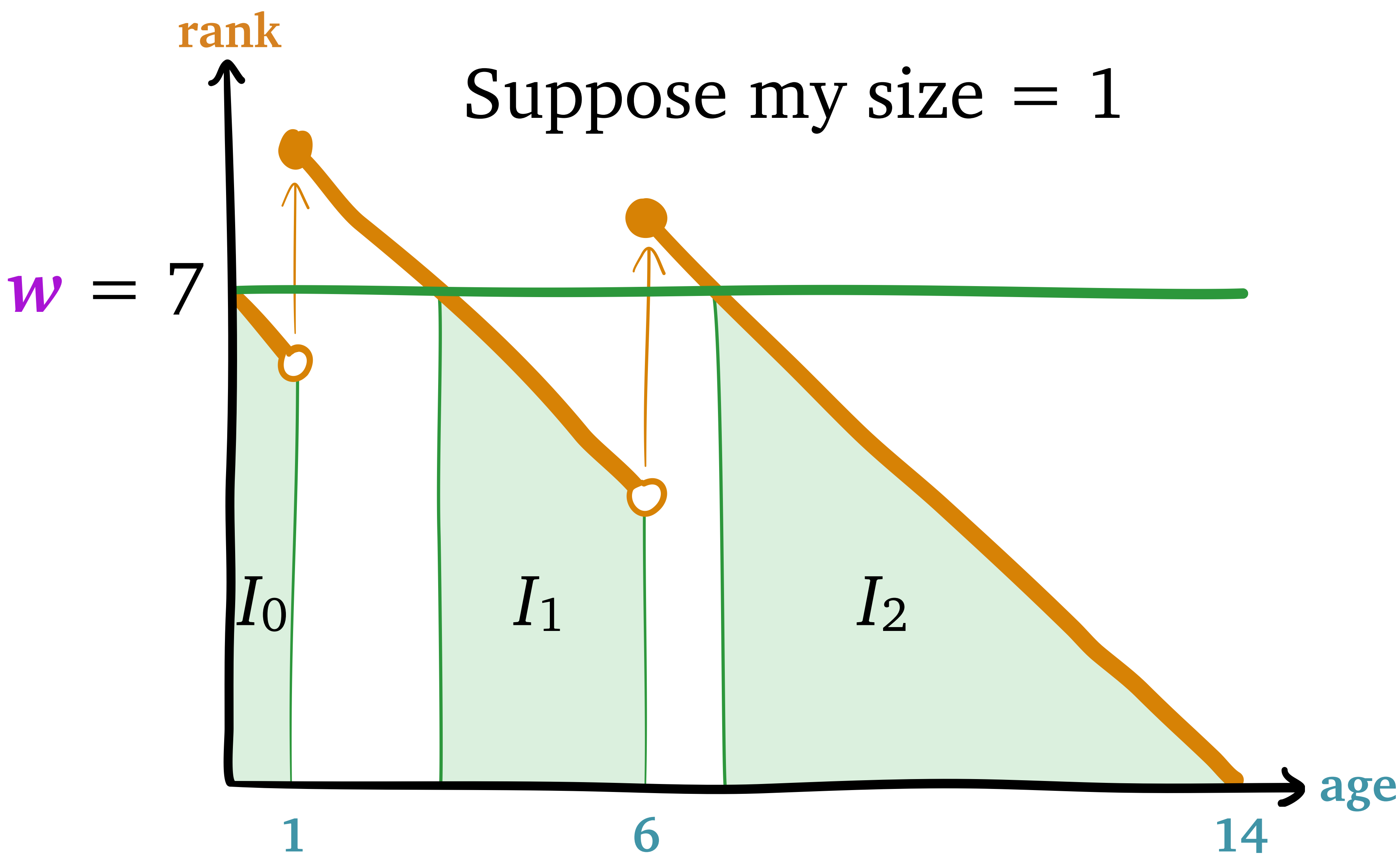
# Relevant Work




Two causes of relevant work:

- $I_0$: arrivals

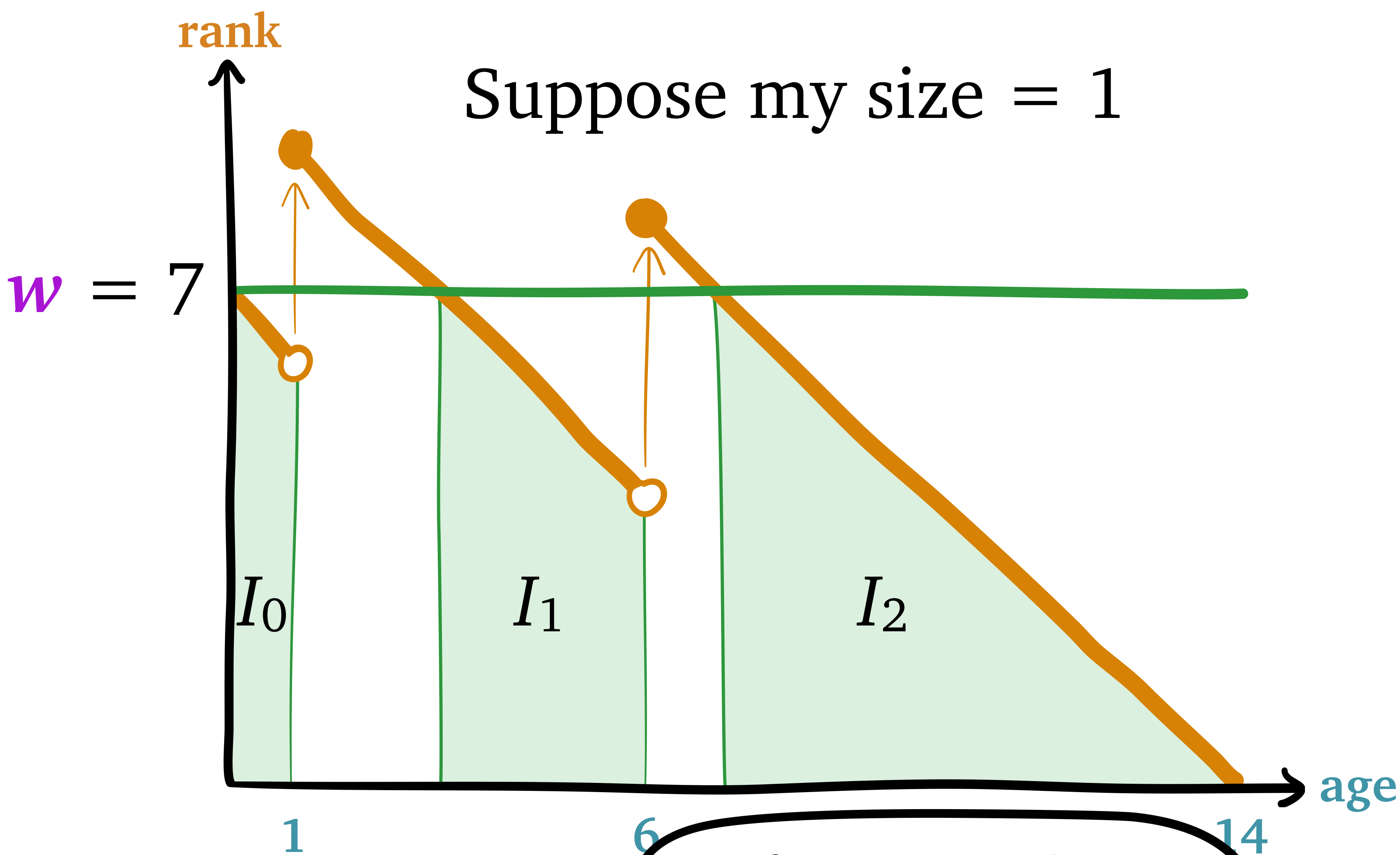
# Relevant Work




Two causes of relevant work:

- $I_0$: arrivals
- $I_1$, $I_2$: **recyclings**

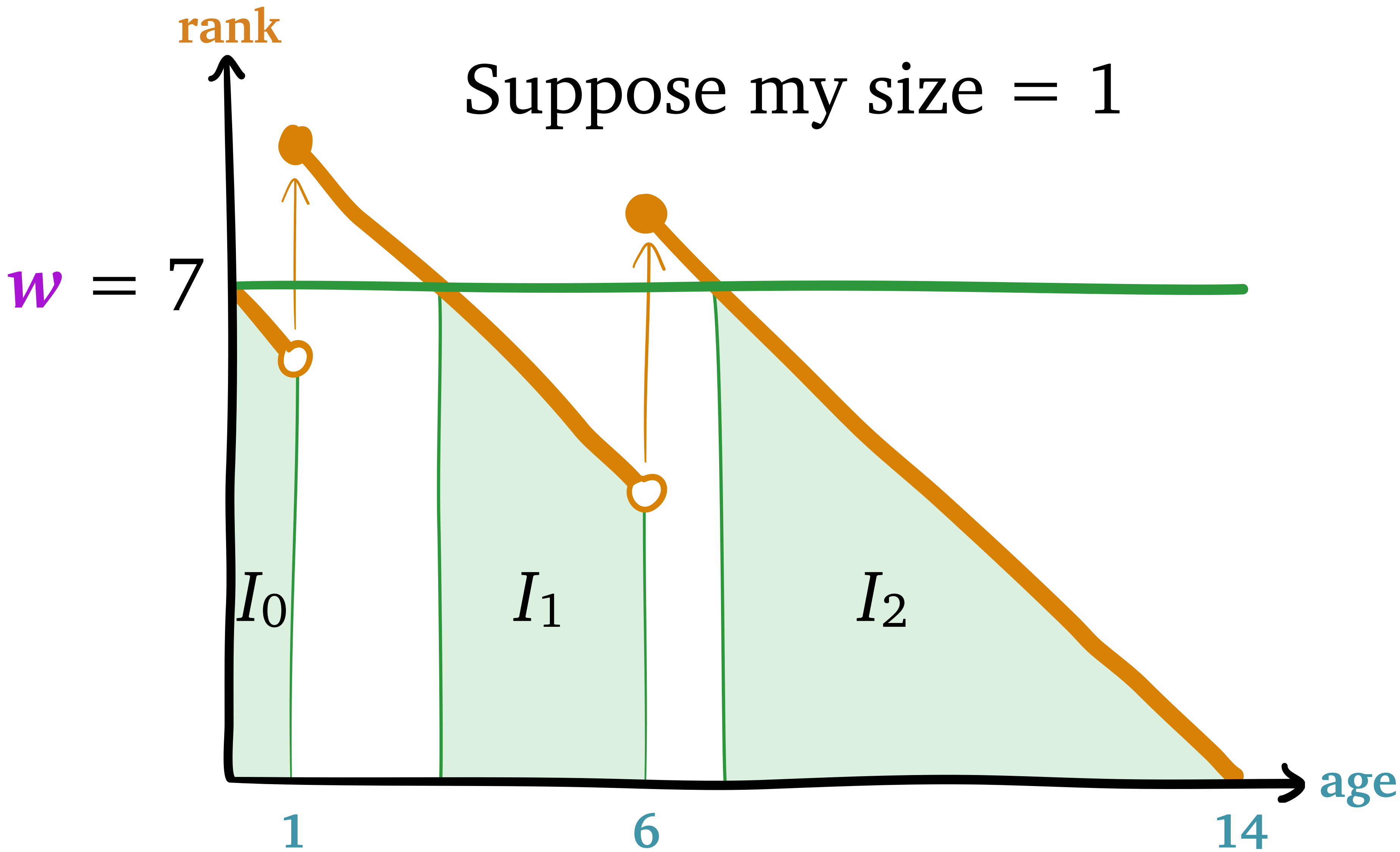
# Relevant Work

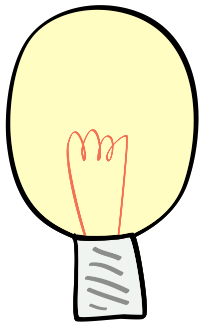



Two caus

- $I_0$: arrivals
- $I_1, I_2$: **recyclings**

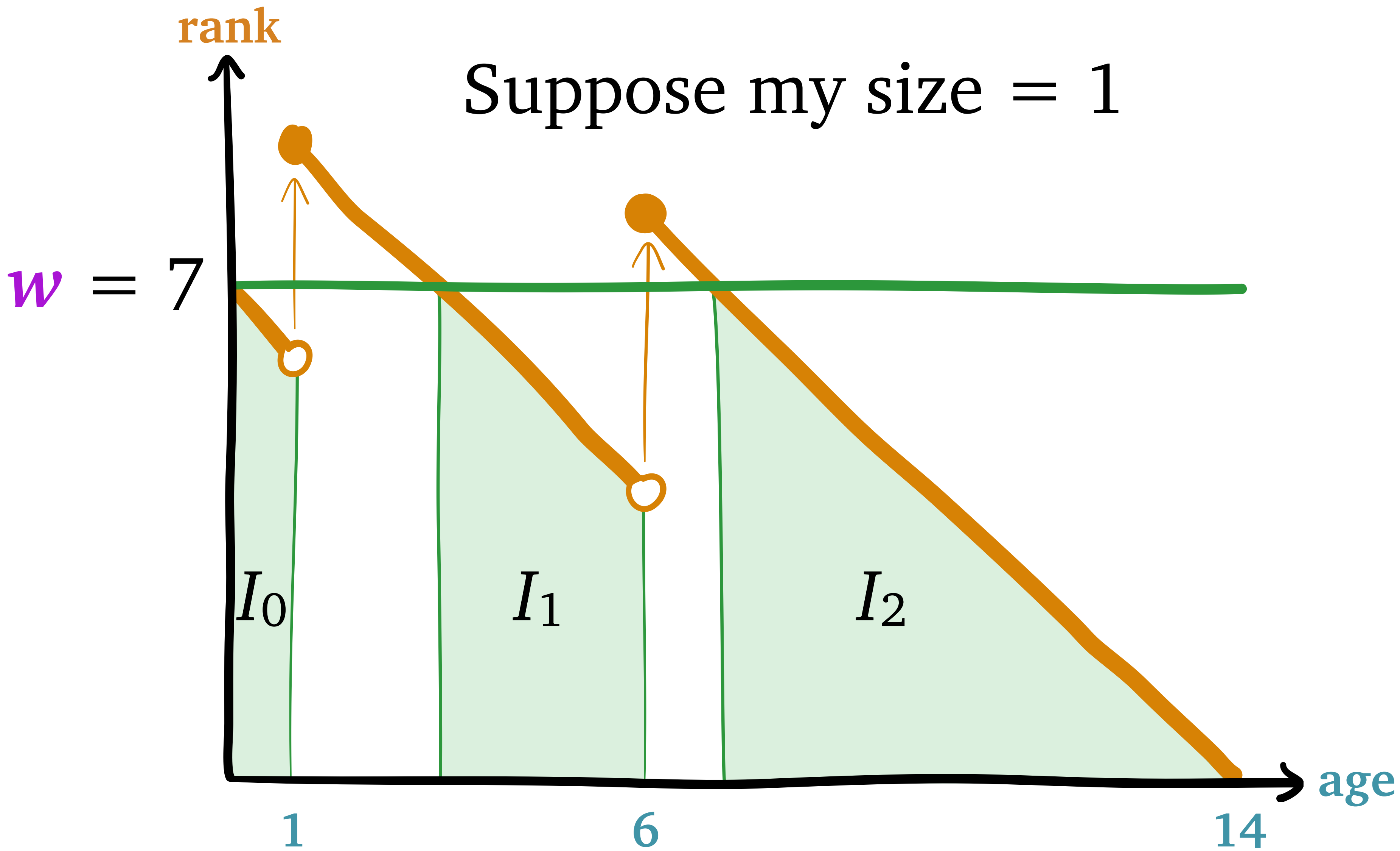
# Relevant Work

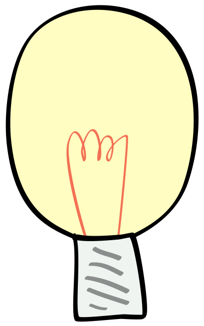


**Observations:**

# Relevant Work

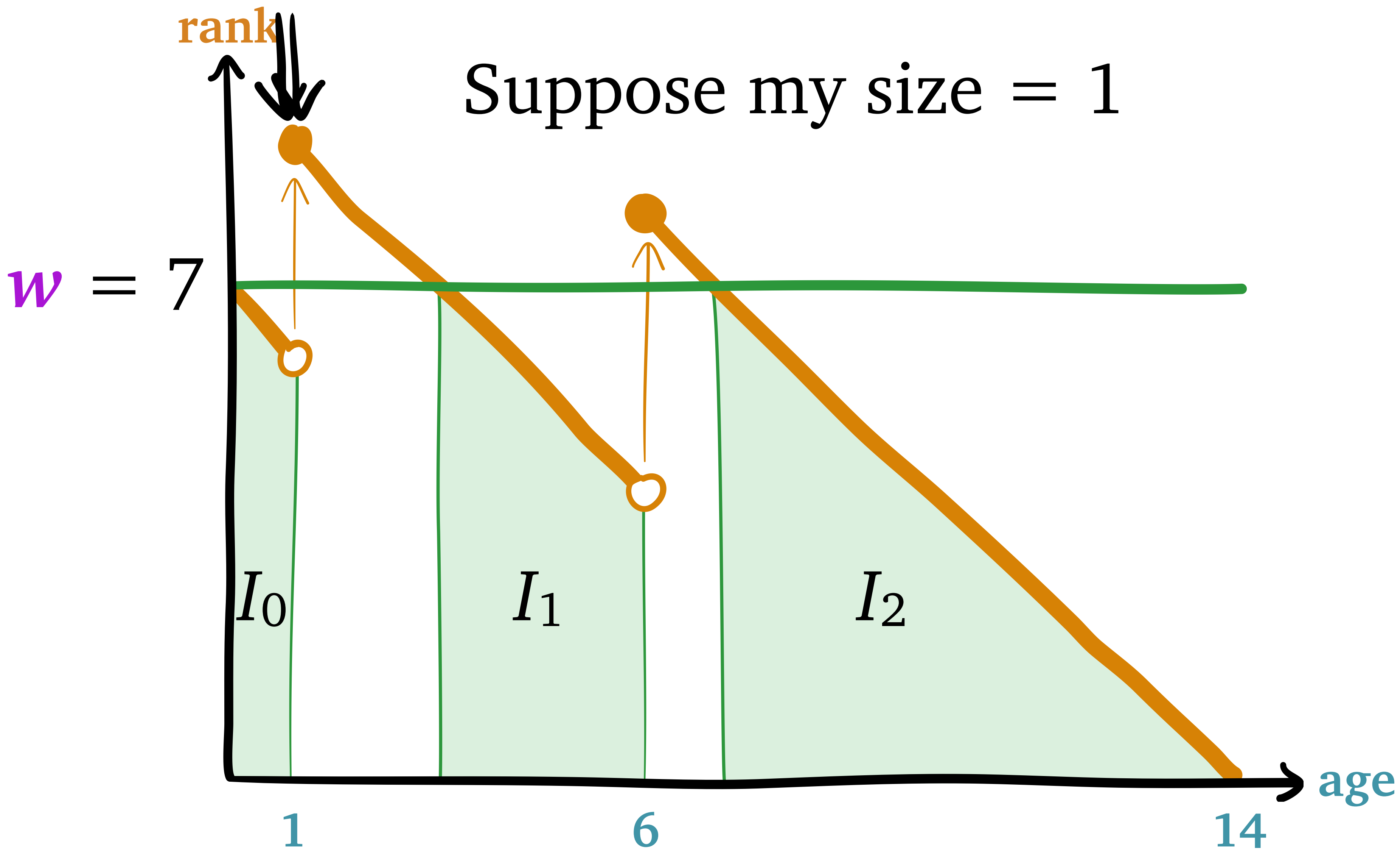


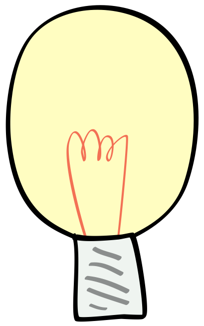



**Observations:**

- at most one **recycled** job at a time

# Relevant Work

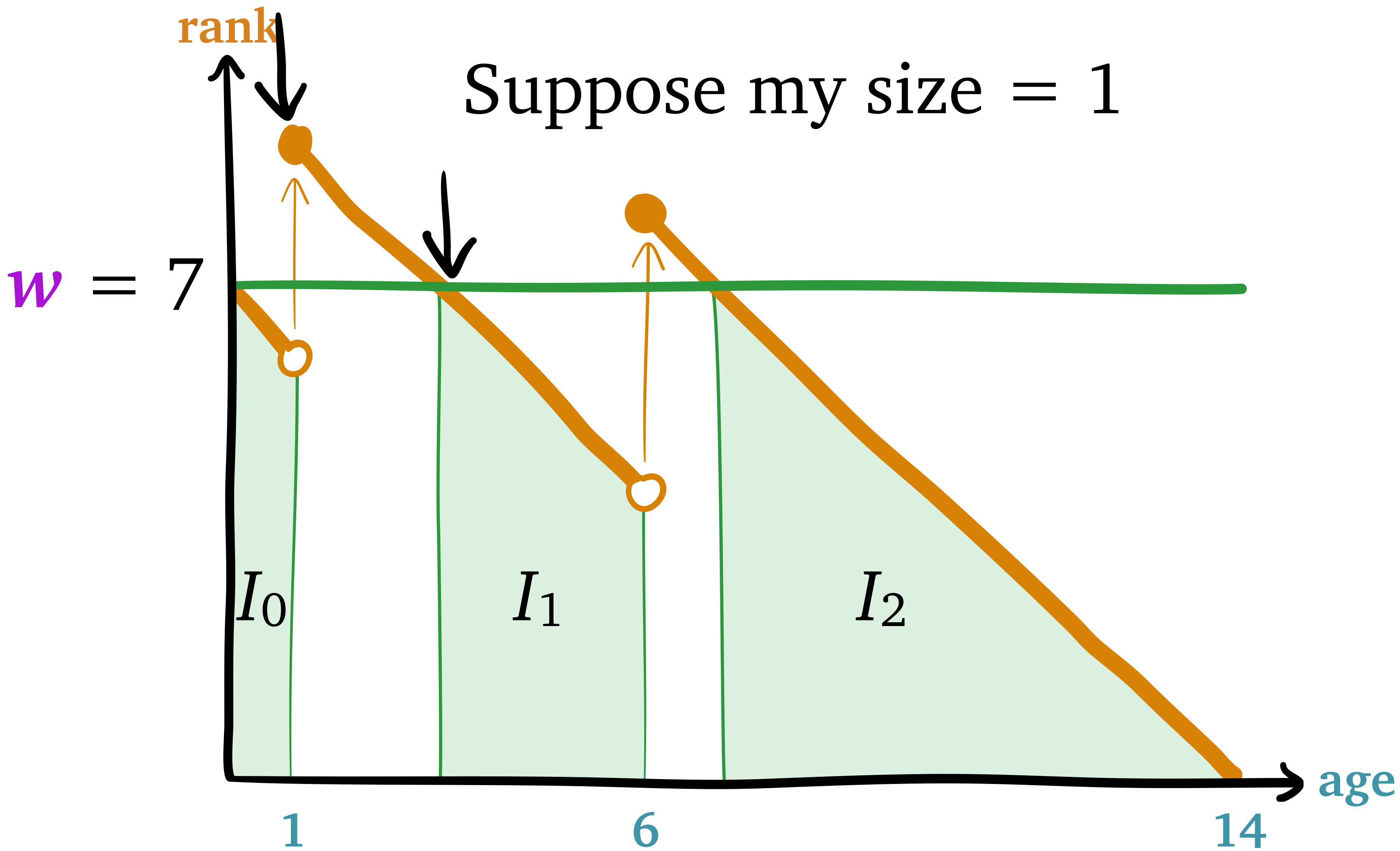


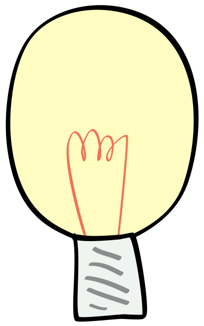



**Observations:**

- at most one **recycled** job at a time

# Relevant Work

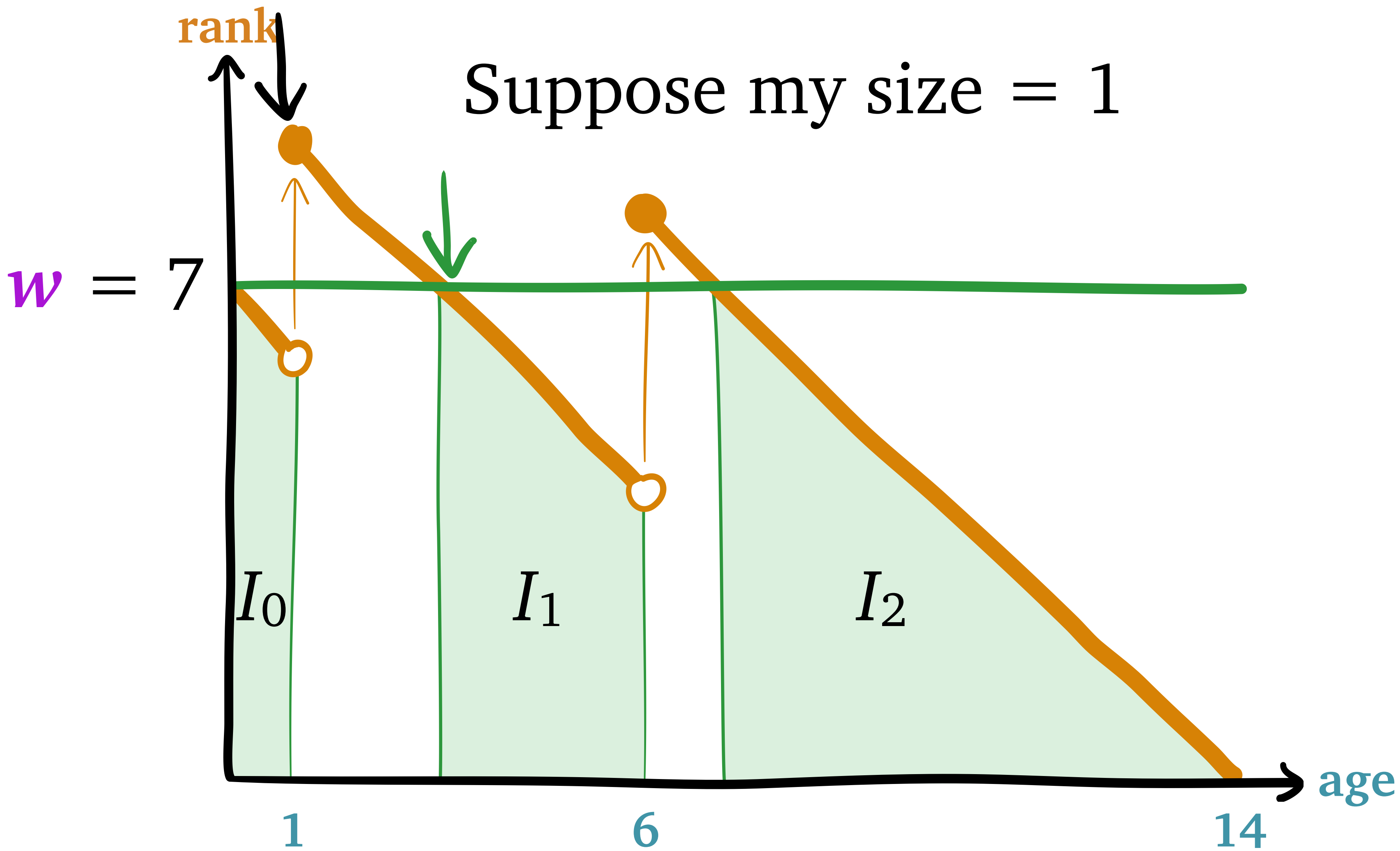


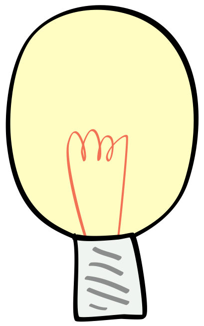



**Observations:**

- at most one **recycled** job at a time

# Relevant Work

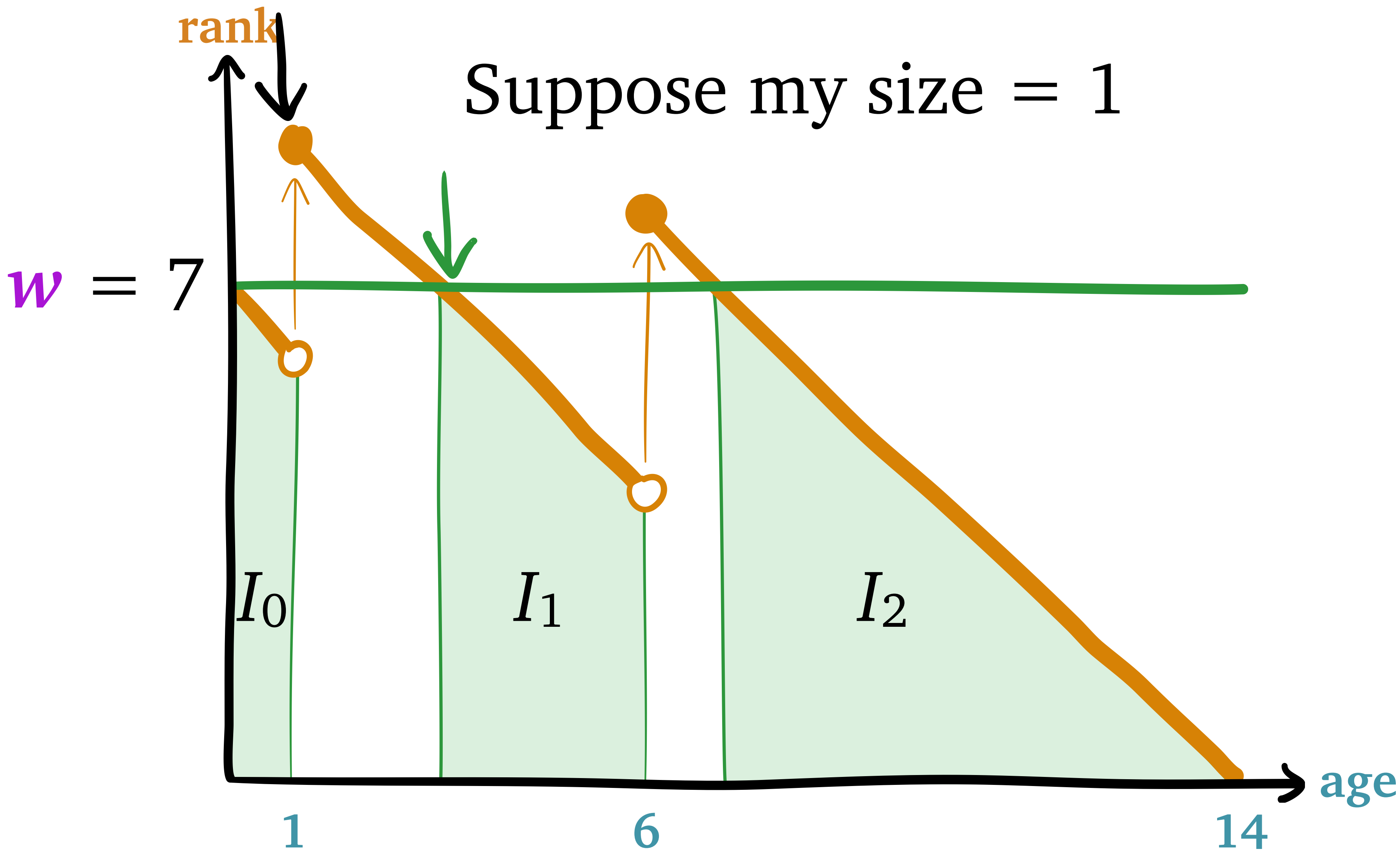


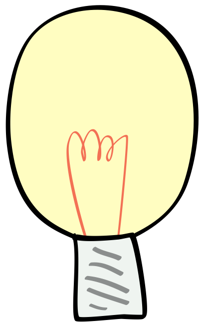



**Observations:**

- at most one **recycled** job at a time

# Relevant Work



**Observations:**

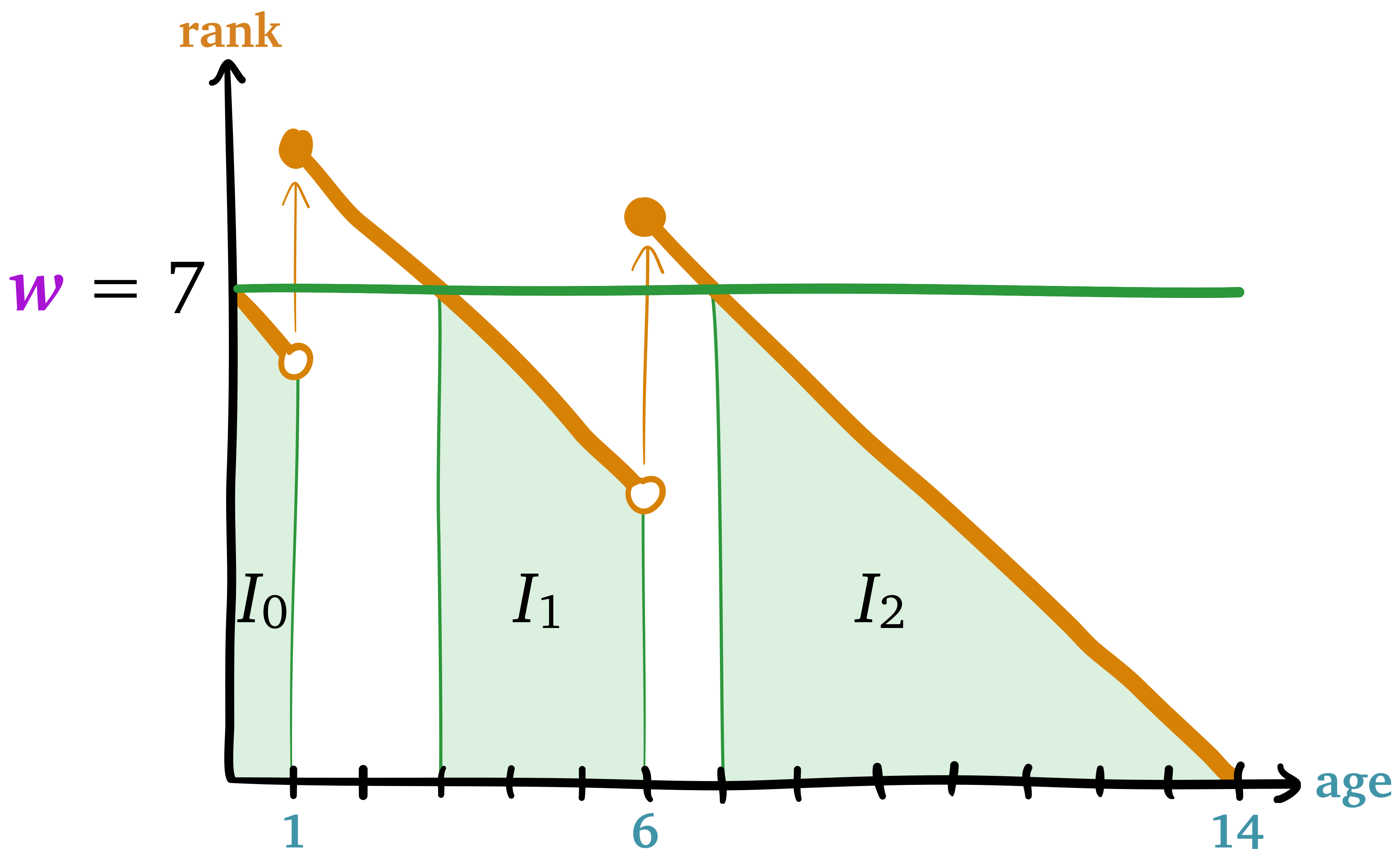
- at most one **recycled** job at a time
- **recyclings** occur only when no relevant work

**SOAP** Insight #2:

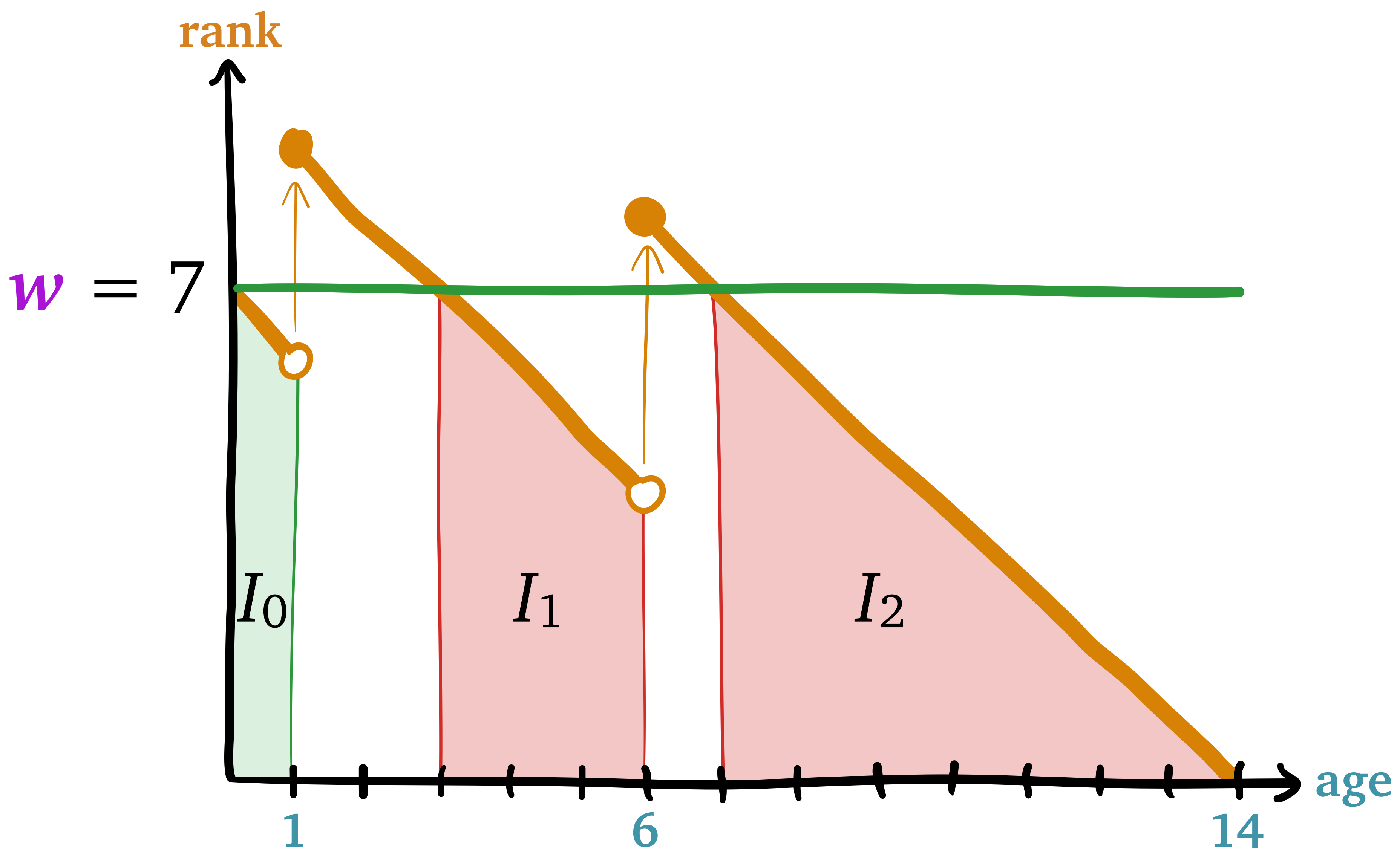
# Vacation Transformation

Replace **recycled** jobs with server **vacations**

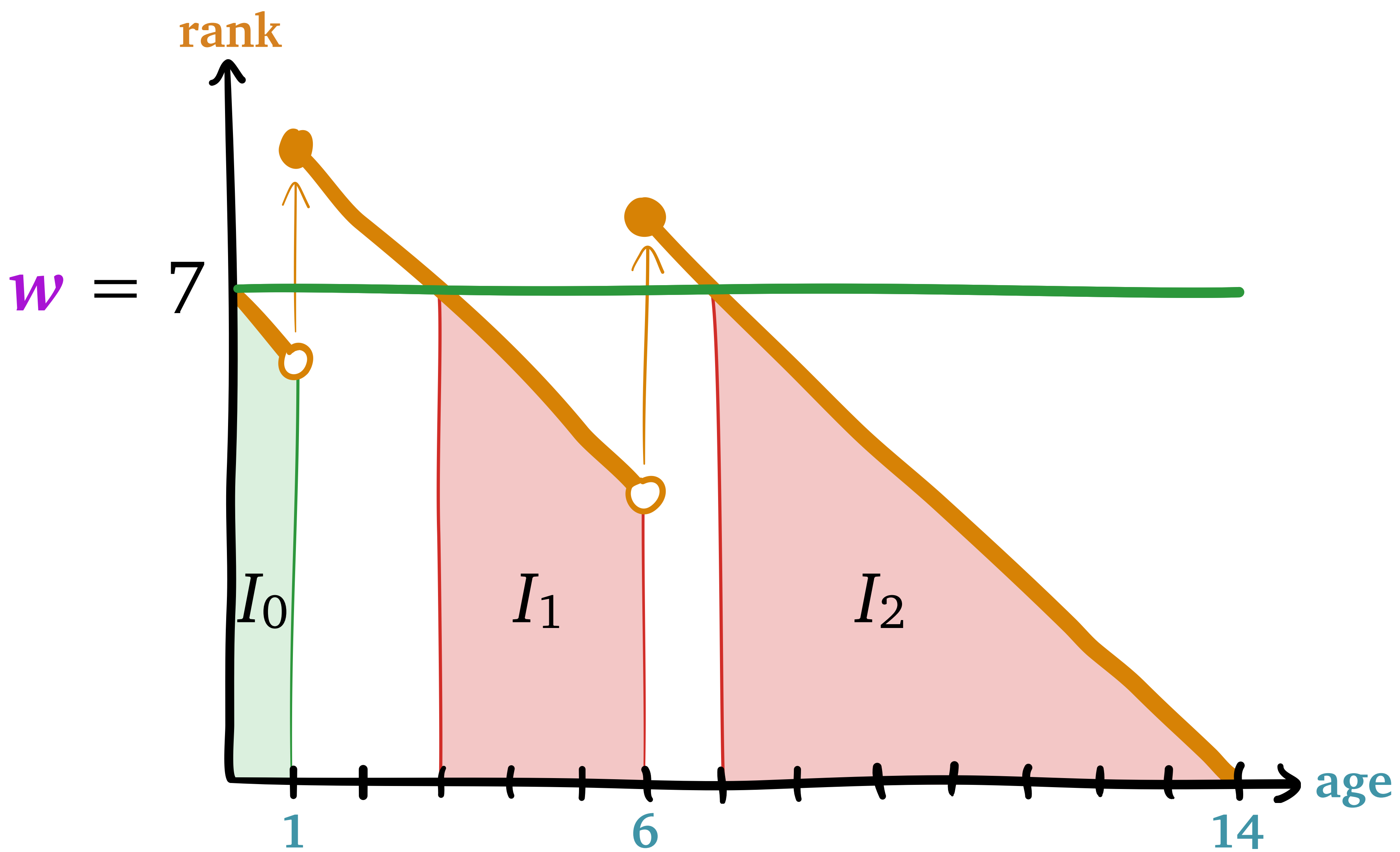
# Vacation Transformation



# Vacation Transformation



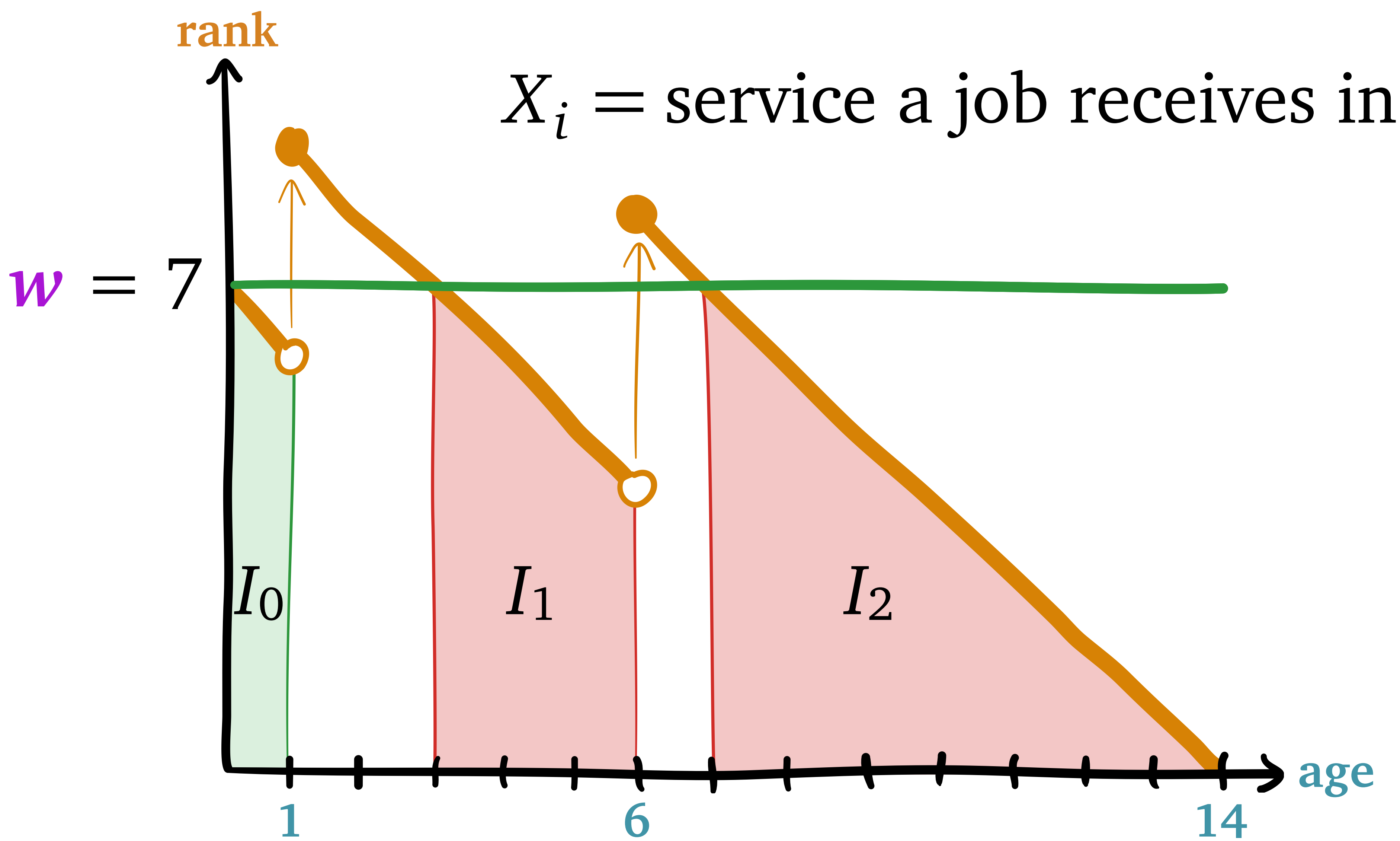
# Vacation Transformation




$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

(Fuhrmann and Cooper, 1985)

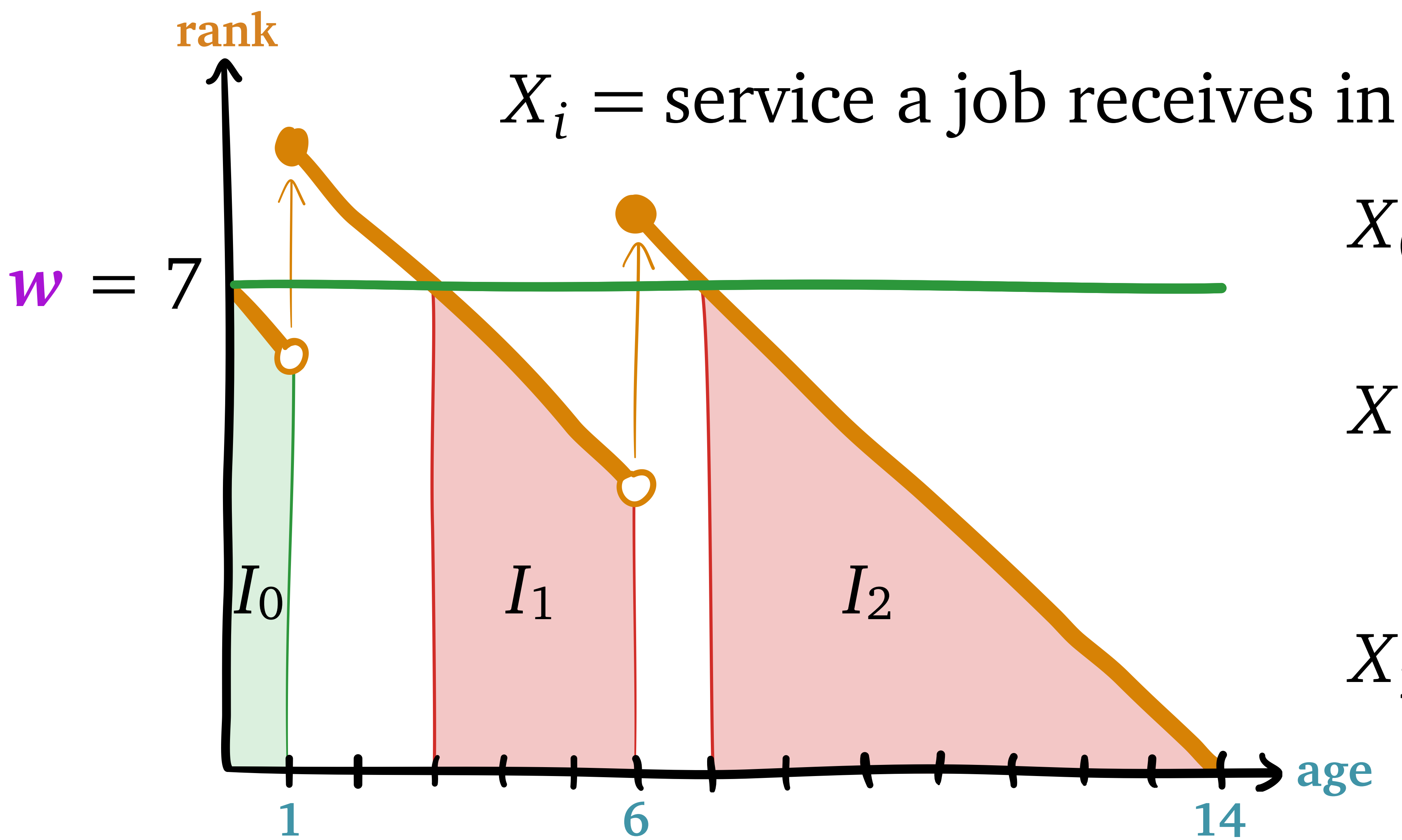
# Vacation Transformation




$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

(Fuhrmann and Cooper, 1985)

# Vacation Transformation




$X_0 =$

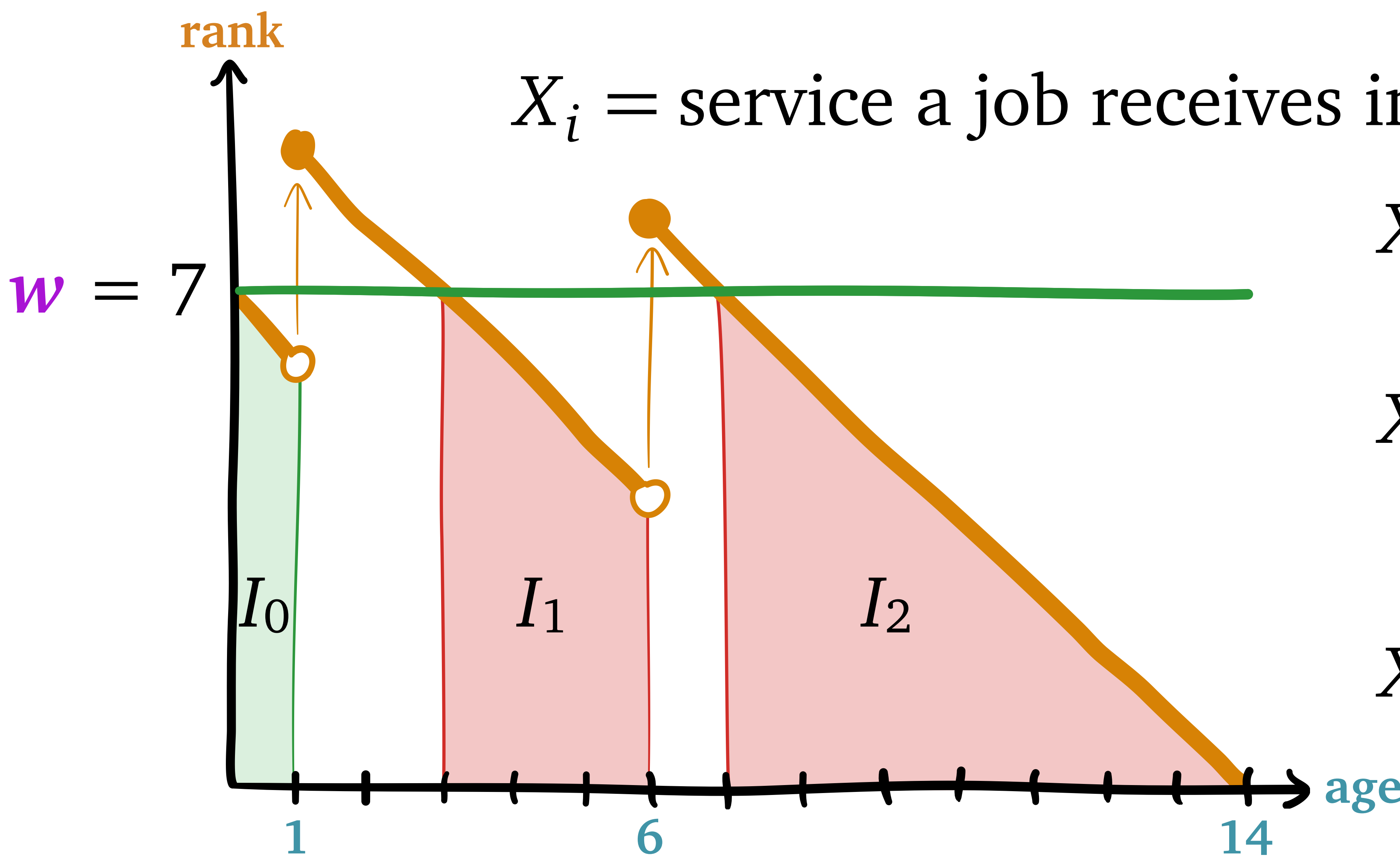
$X_1 =$

$X_2 =$

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

(Fuhrmann and Cooper, 1985)

# Vacation Transformation




$X_i$ = service a job receives in $I_i$

$X_0 = 1$

$X_1 =$

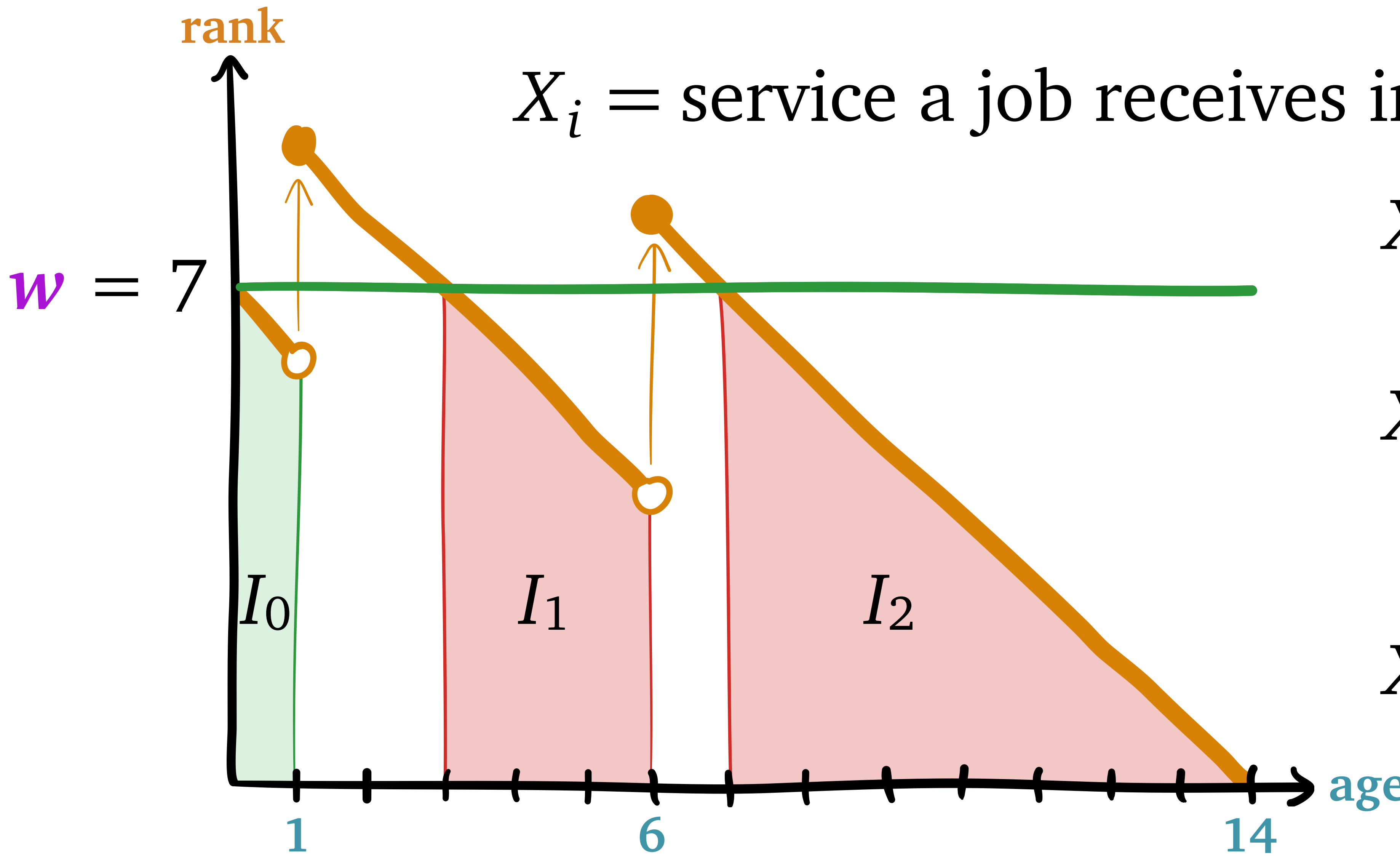
$X_2 =$

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

(Fuhrmann and Cooper, 1985)

# Vacation Transformation

$X_i$ = service a job receives in $I_i$




$$X_0 = 1$$

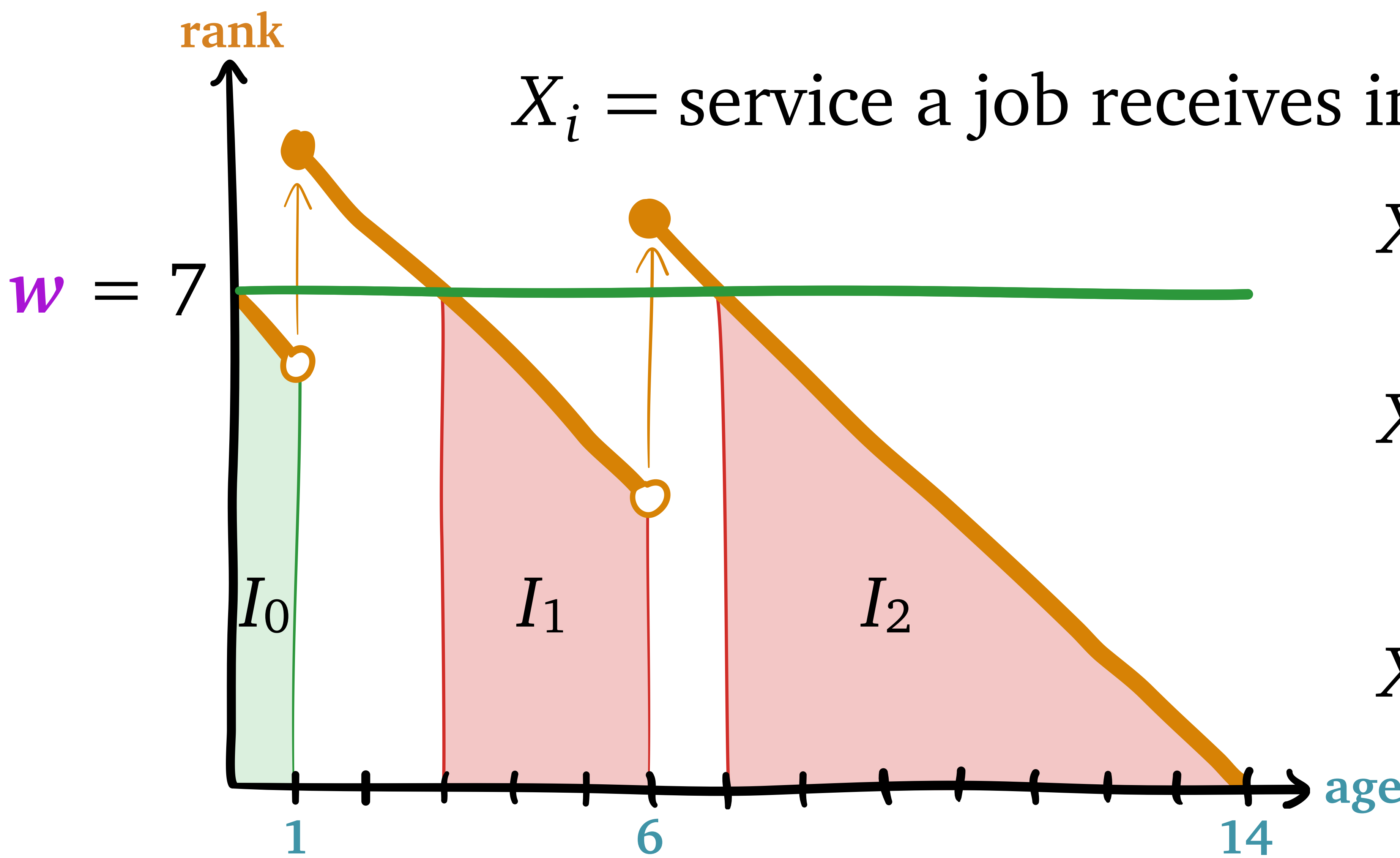
$$X_1 = \begin{cases} 0 & \text{w.p. } \frac{1}{3} \\ 3 & \text{w.p. } \frac{2}{3} \end{cases}$$

$$X_2 =$$

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

(Fuhrmann and Cooper, 1985)

# Vacation Transformation




$X_i$ = service a job receives in $I_i$

$X_0 = 1$

$$X_1 = \begin{cases} 0 & \text{w.p. } \frac{1}{3} \\ 3 & \text{w.p. } \frac{2}{3} \end{cases}$$

$$X_2 = \begin{cases} 0 & \text{w.p. } \frac{2}{3} \\ 7 & \text{w.p. } \frac{1}{3} \end{cases}$$

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

(Fuhrmann and Cooper, 1985)

# Response Time: Size 1

**Relevant work** ($w = 7$):

$$\mathbf{E}[U[7]] = ???$$

# Response Time: Size 1

**Relevant work** ($w = 7$):

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

# Response Time: Size 1

**Relevant work** ($w = 7$):

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

**Waiting time**:

$$\mathbf{E}[Q_1] = \frac{\mathbf{E}[U[7]]}{1 - \rho_{\text{new}}(0)}$$

# Response Time: Size 1

**Relevant work** ($w = 7$):

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

**Waiting time**:

$$\mathbf{E}[Q_1] = \frac{\mathbf{E}[U[7]]}{1 - \rho_{\text{new}}(0)}$$

**Residence time**:

$$\mathbf{E}[R_1] = \int_0^1 \frac{\mathrm{d}a}{1 - \rho_{\text{new}}(a)}$$

# Response Time: Size 1

**Relevant work** ($w = 7$):

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

**Waiting time**:

$$\mathbf{E}[Q_1] = \frac{\mathbf{E}[U[7]]}{1 - \rho_{\text{new}}(0)}$$

**Residence time**:

$$\mathbf{E}[R_1] = \int_0^1 \frac{\mathrm{d}a}{1 - \rho_{\text{new}}(a)}$$

$\rho_{\text{new}}(a) = \lambda \cdot 0$

# Response Time: Size 1

**Relevant work** ($w = 7$):

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

**Waiting time:**

$$\mathbf{E}[Q_1] = \frac{\mathbf{E}[U[7]]}{1 - \rho_{\text{new}}(0)} = \mathbf{E}[U[7]]$$

**Residence time:**

$$\mathbf{E}[R_1] = \int_0^1 \frac{\mathrm{d}a}{1 - \rho_{\text{new}}(a)} = 1$$

$\rho_{\text{new}}(a) = \lambda \cdot 0$

# Response Time: Size 1

**Relevant work** ($w = 7$):

$$\mathbf{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]}$$

**Waiting time:**

$$\mathbf{E}[Q_1] = \frac{\mathbf{E}[U[7]]}{1 - \rho_{\text{new}}(0)} = \mathbf{E}[U[7]]$$

**Residence time:**

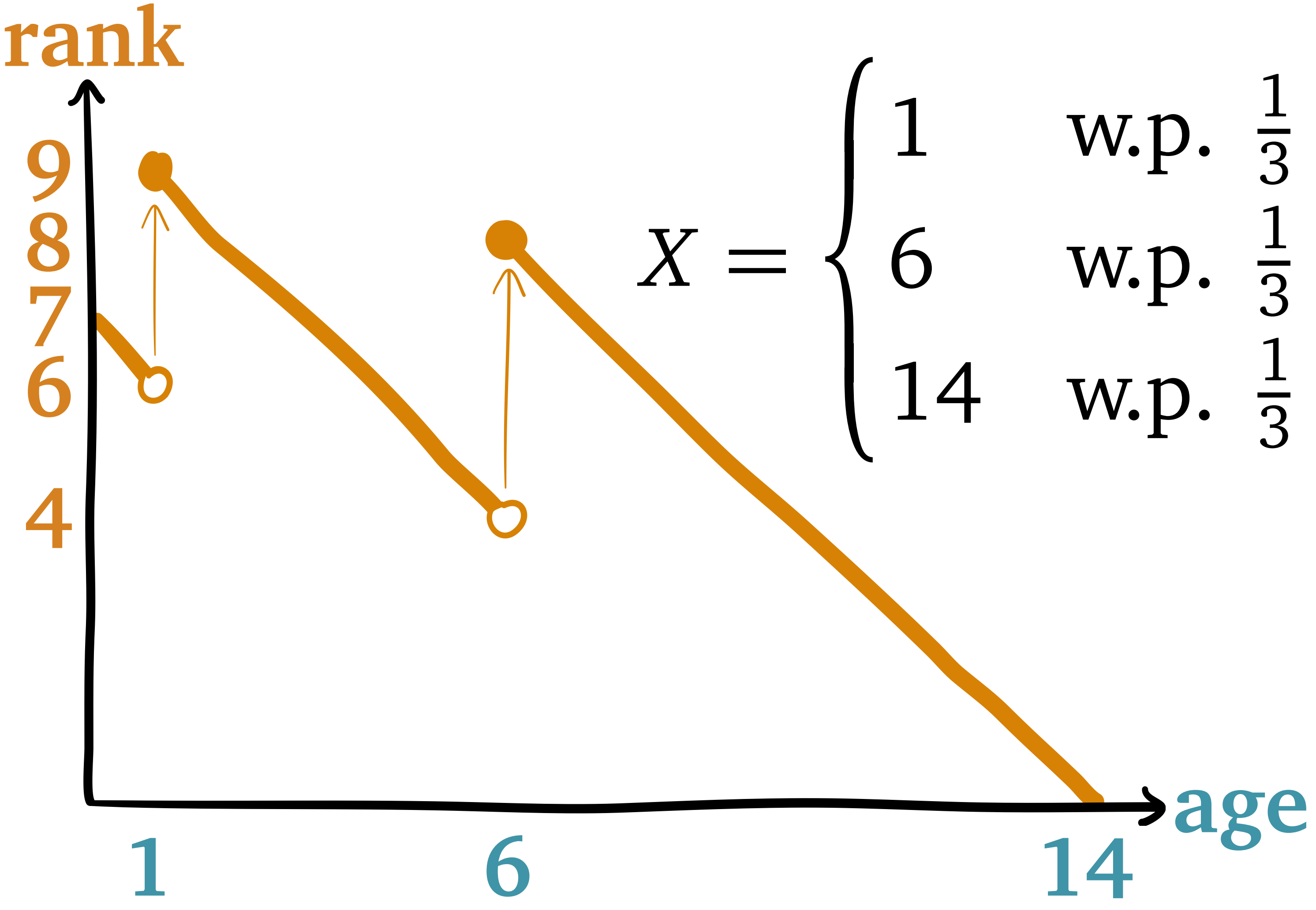
$$\mathbf{E}[R_1] = \int_0^1 \frac{\mathrm{d}a}{1 - \rho_{\text{new}}(a)} = 1$$

$\rho_{\text{new}}(a) = \lambda \cdot 0$

**Response time:**

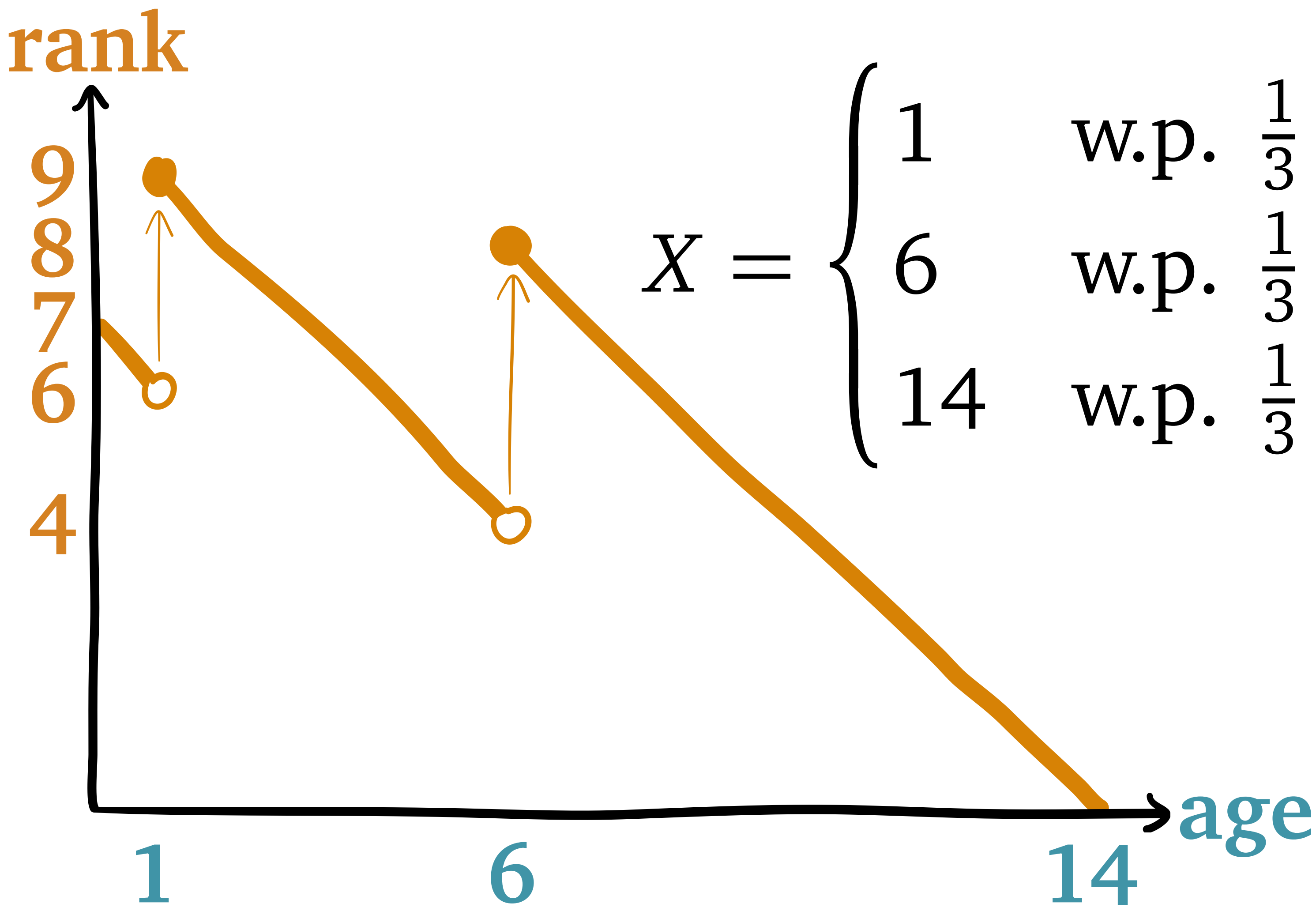
$$\mathbf{E}[T_1] = \mathbf{E}[Q_1] + \mathbf{E}[R_1]$$

# Running example:
# **SERPT**



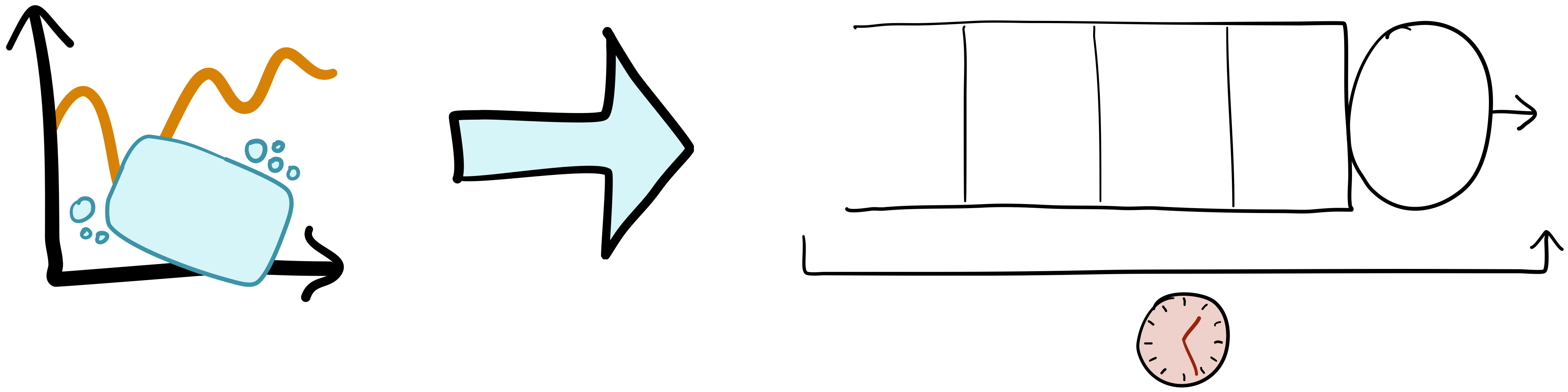

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Running example: **SERPT**




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# $\mathbf{E}[T]$ of *any* **SOAP** Policy



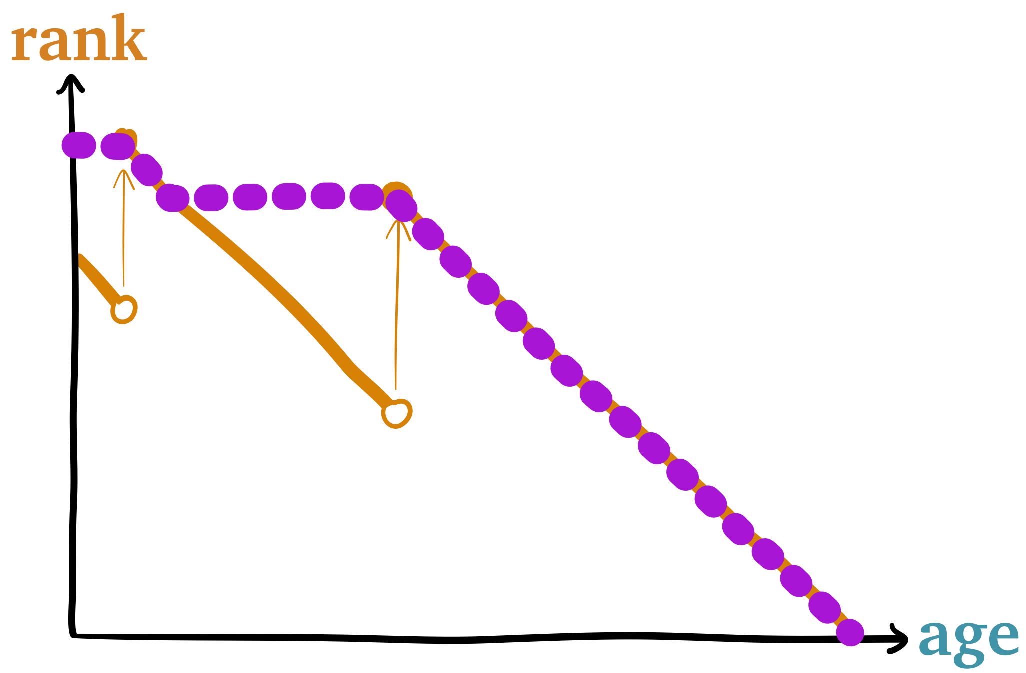
# **Worst** Future Rank

# **Worst** Future Rank

$$w_x(a) = \sup_{a \le b < x} r(b)$$

# Worst Future Rank

$$w_x(a) = \sup_{a \le b < x} r(b)$$



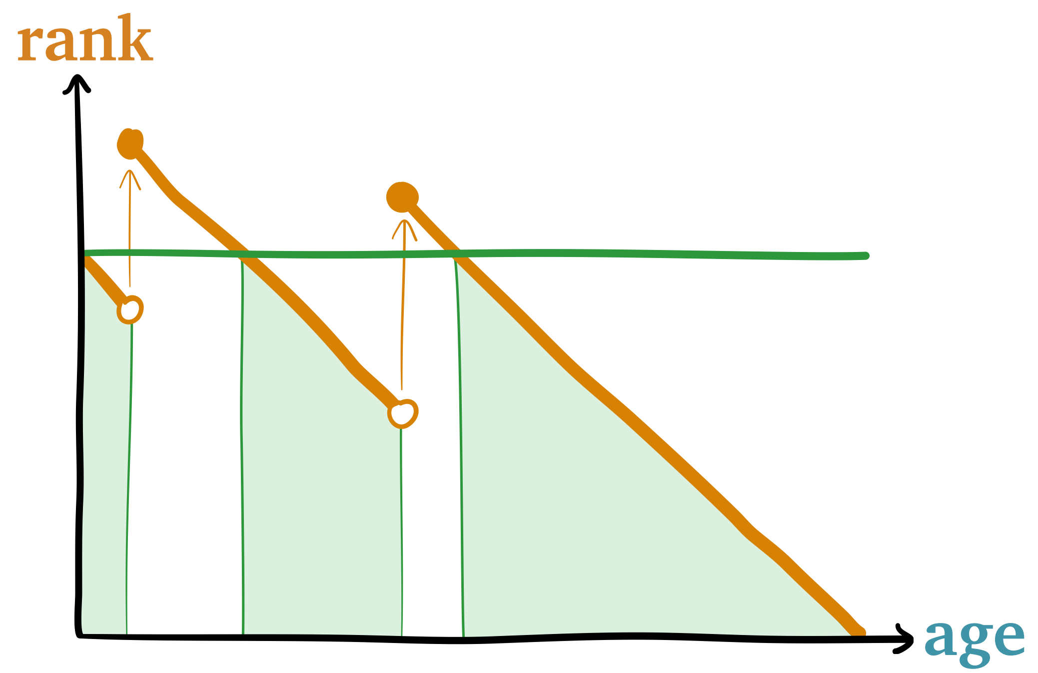
# **Relevant** Intervals

# Relevant Intervals

$I_i[w] = i\text{th interval when } r(a) \leq w$

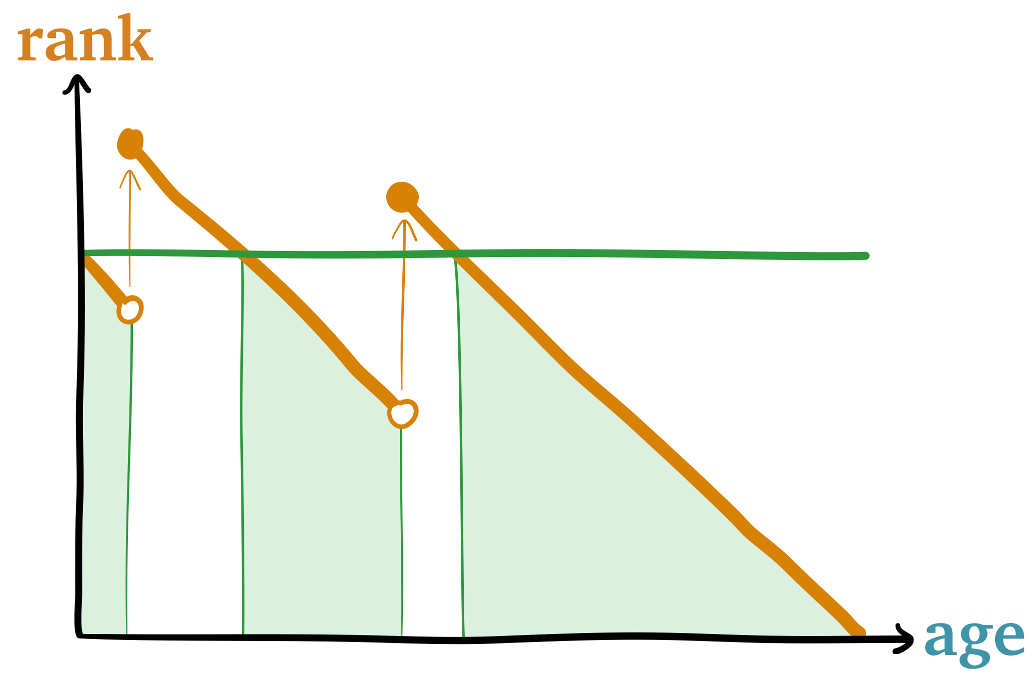
# Relevant Intervals

$$I_i[w] = i\text{th interval when } r(a) \le w$$



# Relevant Intervals

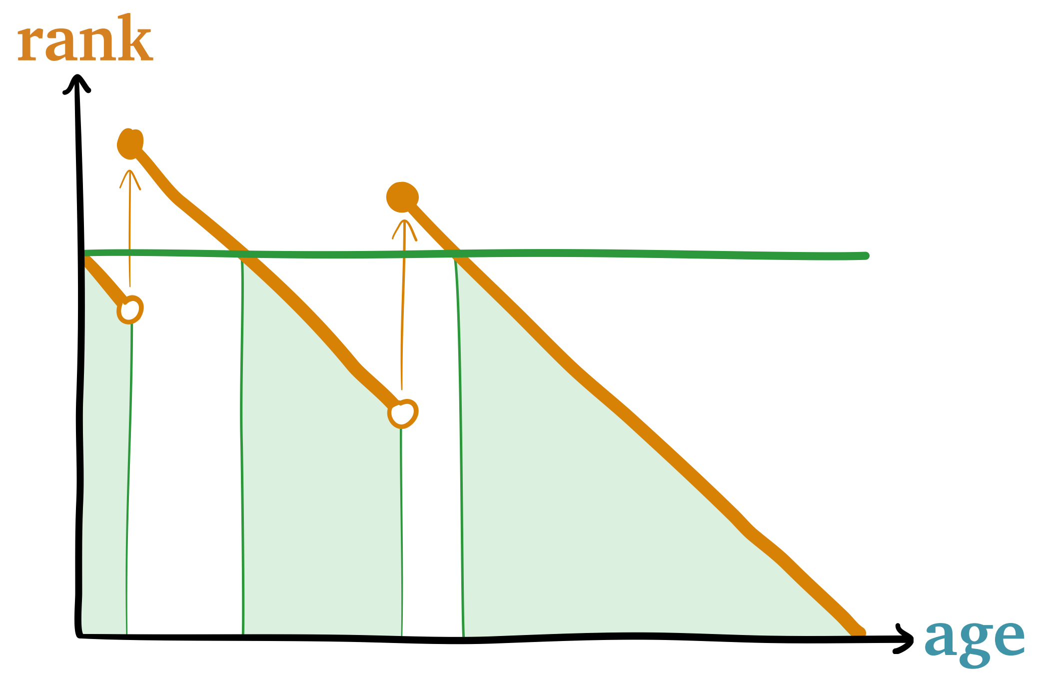
$$I_i[w] = i\text{th interval when } r(a) \le w$$




**Detail**: start with $i = 0$ iff first interval contains age 0, else start with $i = 1$

# Relevant Intervals

$$I_i[w] = i\text{th interval when } r(a) \leq w$$




**Detail**: start with $i = 0$ iff first interval contains age 0, else start with $i = 1$

**Detail**: interval can be empty

# **SOAP** Analysis: One **Descriptor**

# SOAP Analysis: One Descriptor

**Worst Future Rank**

$$w_x(a) = \sup_{a \le b < x} r(b)$$

**Relevant Intervals**

$$I_i[w] = i\text{th interval when } r(a) \le w$$

# SOAP Analysis: One Descriptor

**Worst Future Rank**

$$w_x(a) = \sup_{a \le b < x} r(b)$$

$$\mathbf{E}[T_x] = \frac{\frac{\lambda}{2}\sum_{i=0}^{\infty} \mathbf{E}[X_i[w_x]^2]}{(1-\rho_0[w_x])(1-\rho_{\text{new}}[w_x])} + \int_0^x \frac{\mathrm{d}a}{1-\rho_{\text{new}}[w_x(a)]}$$

**Relevant Intervals**

$$I_i[w] = i\text{th interval when } r(a) \le w$$

# SOAP Analysis: One Descriptor

**Worst Future Rank**

$$w_x(a) = \sup_{a \leq b < x} r(b)$$

$$w_x = w_x(0)$$

$$\mathbf{E}[T_x] = \frac{\frac{\lambda}{2} \sum_{i=0}^{\infty} \mathbf{E}[X_i[w_x]^2]}{(1 - \rho_0[w_x])(1 - \rho_{\text{new}}[w_x])} + \int_0^x \frac{\mathrm{d}a}{1 - \rho_{\text{new}}[w_x(a)]}$$

**Relevant Intervals**

$$I_i[w] = i\text{th interval when } r(a) \leq w$$

# SOAP Analysis: One Descriptor

**Worst Future Rank**

$$w_x(a) = \sup_{a \le b < x} r(b)$$

$$w_x = w_x(0)$$

$$\mathbf{E}[T_x] = \frac{\frac{\lambda}{2}\sum_{i=0}^{\infty} \mathbf{E}[X_i[w_x]^2]}{(1-\rho_0[w_x])(1-\rho_{\text{new}}[w_x])} + \int_0^x \frac{\mathrm{d}a}{1-\rho_{\text{new}}[w_x(a)]}$$

**Relevant Intervals**

$$I_i[w] = i\text{th interval when } r(a) \le w$$

$$X_i[w] = \text{service a job receives in } I_i[w]$$

# SOAP Analysis: One Descriptor

**Worst Future Rank**

$$w_x(a) = \sup_{a \le b < x} r(b)$$

$$w_x = w_x(0)$$

$$\mathbf{E}[T_x] = \frac{\frac{\lambda}{2}\sum_{i=0}^{\infty}\mathbf{E}[X_i[w_x]^2]}{(1-\rho_0[w_x])(1-\rho_{\text{new}}[w_x])} + \int_0^x \frac{\mathrm{d}a}{1-\rho_{\text{new}}[w_x(a)]}$$

**Relevant Intervals**

$$I_i[w] = i\text{th interval when } r(a) \le w$$

$$X_i[w] = \text{service a job receives in } I_i[w]$$

$$\rho_0[w] = \lambda\mathbf{E}[X_0[w]]$$

# SOAP Analysis: One Descriptor

**Worst Future Rank**

$$w_x(a) = \sup_{a \le b < x} r(b)$$

$$w_x = w_x(0)$$

$$\mathbf{E}[T_x] = \frac{\frac{\lambda}{2}\sum_{i=0}^{\infty} \mathbf{E}[X_i[w_x]^2]}{(1-\rho_0[w_x])(1-\rho_{\text{new}}[w_x])} + \int_0^x \frac{\mathrm{d}a}{1-\rho_{\text{new}}[w_x(a)]}$$

**Relevant Intervals**

$$I_i[w] = i\text{th interval when } r(a) \le w$$

$$X_i[w] = \text{service a job receives in } I_i[w]$$

$$\rho_0[w] = \lambda \mathbf{E}[X_0[w]]$$

$$\rho_{\text{new}}[w] = \lambda \mathbf{E}[X_0[w-]]$$

# SOAP Analysis: Complete

## Worst Future Rank

$$w_x(a) = \sup_{a \le b < x} r(b)$$

## Relevant Intervals

$$I_i[w] = i\text{th interval when } r(a) \le w$$

# SOAP Analysis: Complete

**Worst Future Rank**

$$w_{d,x}(a) = \sup_{a \le b < x} r_d(b)$$

**Relevant Intervals**

$$I_{i,d}[w] = i\text{th interval when } r_d(a) \le w$$

# SOAP Analysis: Complete

**Worst Future Rank**

$$w_{d,x}(a) = \sup_{a \le b < x} r_d(b)$$

**Relevant Intervals**

$$I_{i,d}[w] = i\text{th interval when } r_d(a) \le w$$

$$X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w]$$

# SOAP Analysis: Complete

**Worst Future Rank**

$$w_{d,x}(a) = \sup_{a \le b < x} r_d(b)$$

**Relevant Intervals**

$X_d$ = size distribution for descriptor $d$

$$I_{i,d}[w] = i\text{th interval when } r_d(a) \le w$$

$$X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w]$$

# SOAP Analysis: Complete

**Worst Future Rank**

$$w_{d,x}(a) = \sup_{a \le b < x} r_d(b)$$

**Relevant Intervals**

$X_d$ = size distribution for descriptor $d$

$$I_{i,d}[w] = i\text{th interval when } r_d(a) \le w$$

$$X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w]$$

$$X_i[w] = X_{i,D}[w]$$

# SOAP Analysis: Complete

**Worst Future Rank**

$$w_{d,x}(a) = \sup_{a \le b < x} r_d(b)$$

**Relevant Intervals**

$X_d$ = size distribution for descriptor $d$

$$I_{i,d}[w] = i\text{th interval when } r_d(a) \le w$$

$$X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w]$$

$$X_i[w] = X_{i,D}[w]$$

$D$ = descriptor distribution

# SOAP Analysis: Complete

**Worst Future Rank**

$$w_{d,x}(a) = \sup_{a \le b < x} r_d(b)$$

$$w_{d,x} = w_{d,x}(0)$$

**Relevant Intervals**

$X_d$ = size distribution for descriptor $d$

$$I_{i,d}[w] = i\text{th interval when } r_d(a) \le w$$

$$X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w]$$

$$X_i[w] = X_{i,D}[w]$$

$D$ = descriptor distribution

$$\rho_0[w] = \lambda \mathbf{E}[X_0[w]]$$

$$\rho_{\text{new}}[w] = \lambda \mathbf{E}[X_0[w-]]$$

# SOAP Analysis: Complete

**Worst Future Rank**

$$w_{d,x}(a) = \sup_{a \le b < x} r_d(b)$$

$$w_{d,x} = w_{d,x}(0)$$

$$\mathbf{E}[T_{d,x}] = \frac{\frac{\lambda}{2}\sum_{i=0}^{\infty} \mathbf{E}[X_i[w_{d,x}]^2]}{(1-\rho_0[w_{d,x}])(1-\rho_{\text{new}}[w_{d,x}])} + \int_0^x \frac{\mathrm{d}a}{1-\rho_{\text{new}}[w_{d,x}(a)]}$$

**Relevant Intervals**

$X_d$ = size distribution for descriptor $d$

$I_{i,d}[w] = i$th interval when $r_d(a) \le w$

$X_{i,d}[w] =$ service a job of descriptor $d$ receives in $I_{i,d}[w]$

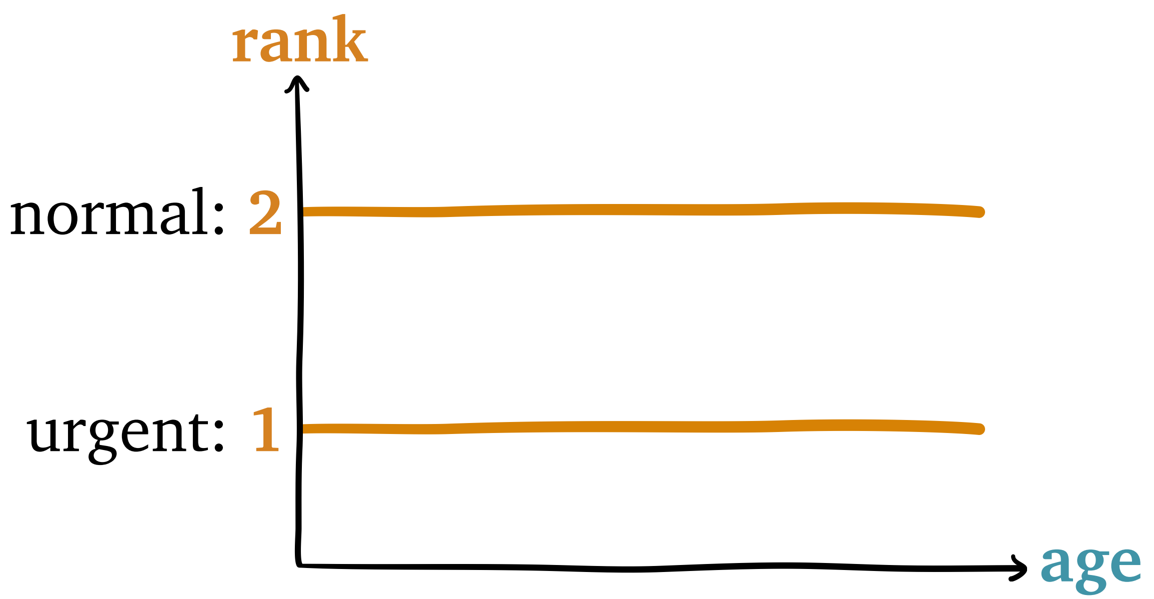
$X_i[w] = X_{i,D}[w]$

$D$ = descriptor distribution

$\rho_0[w] = \lambda \mathbf{E}[X_0[w]]$

$\rho_{\text{new}}[w] = \lambda \mathbf{E}[X_0[w-]]$

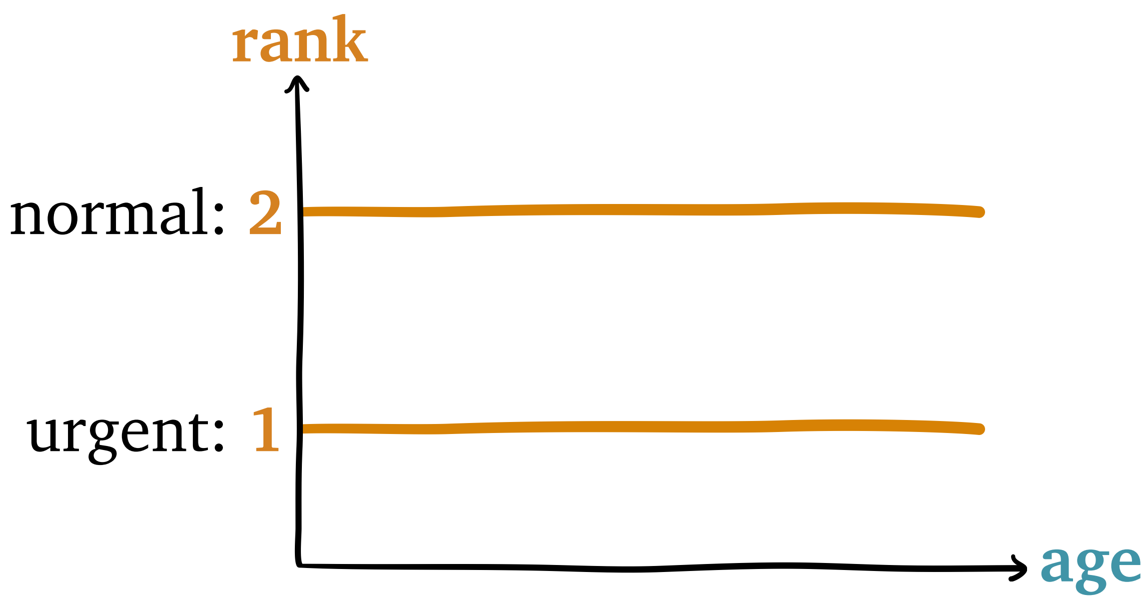
# Example: Preemptive Priority



# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

**Normal** ($d$ = N, $r$ = 2)

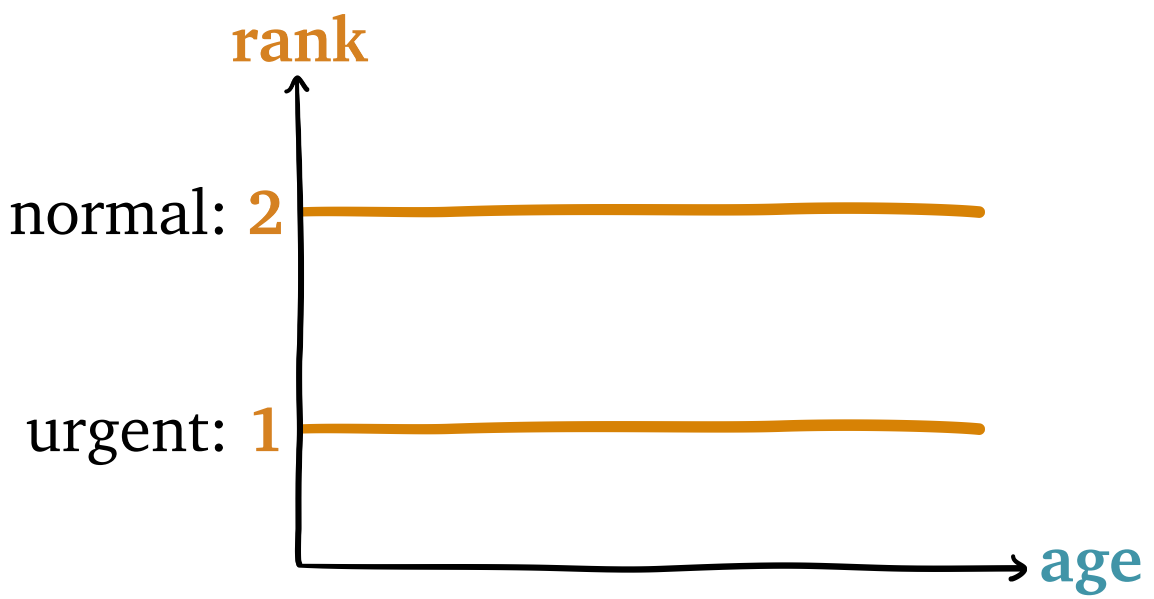


# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs

**Normal** ($d$ = N, $r$ = 2)

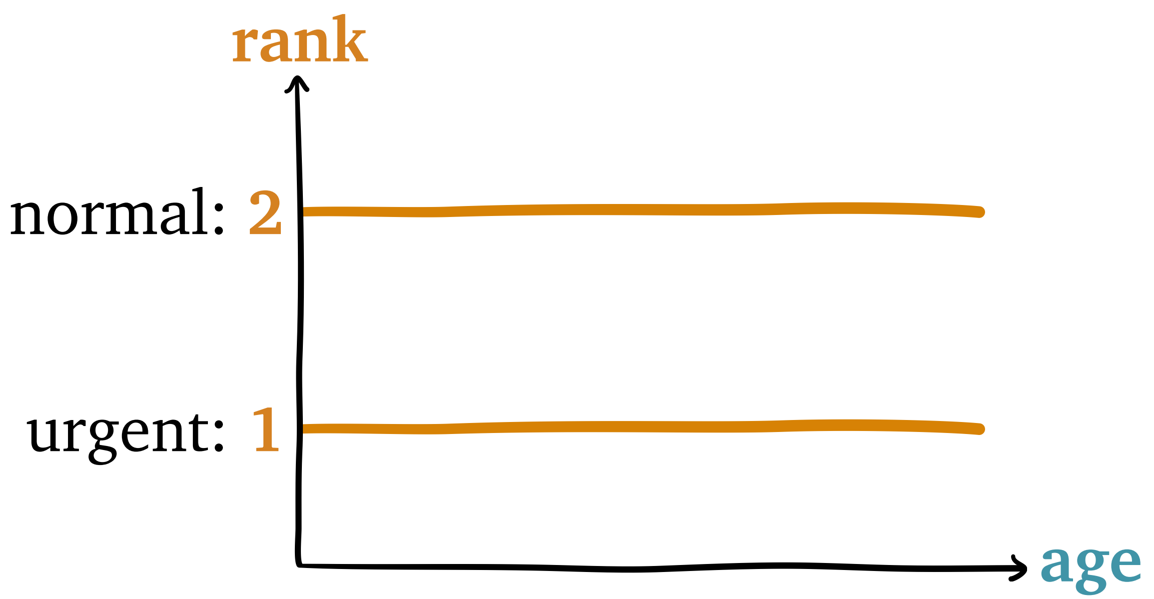


# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_U$

**Normal** ($d$ = N, $r$ = 2)



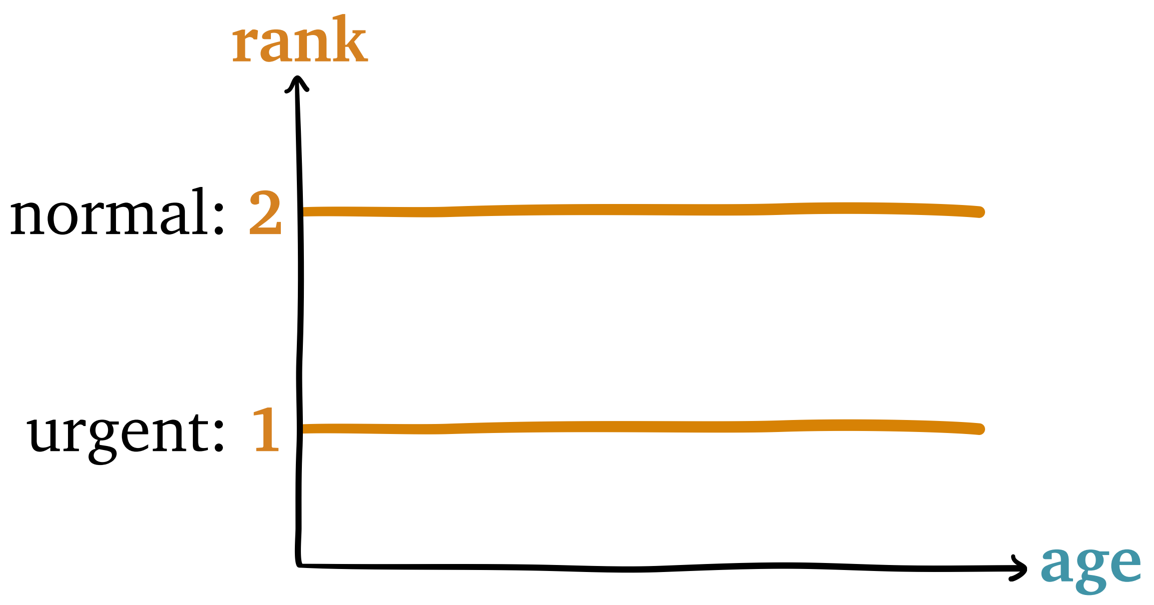
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_U$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs



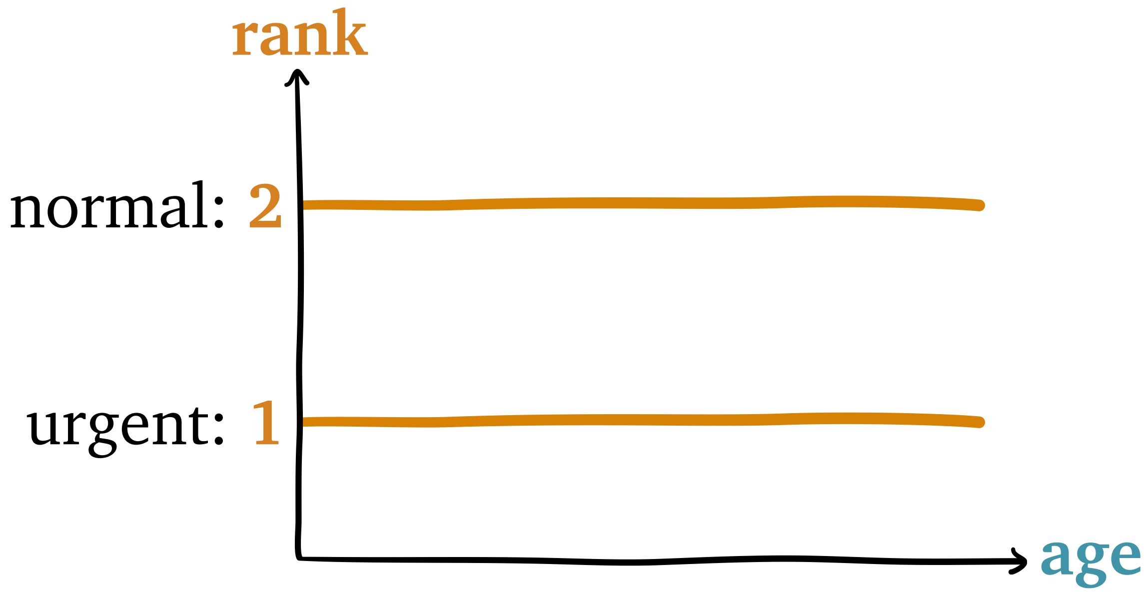
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_U$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_N$



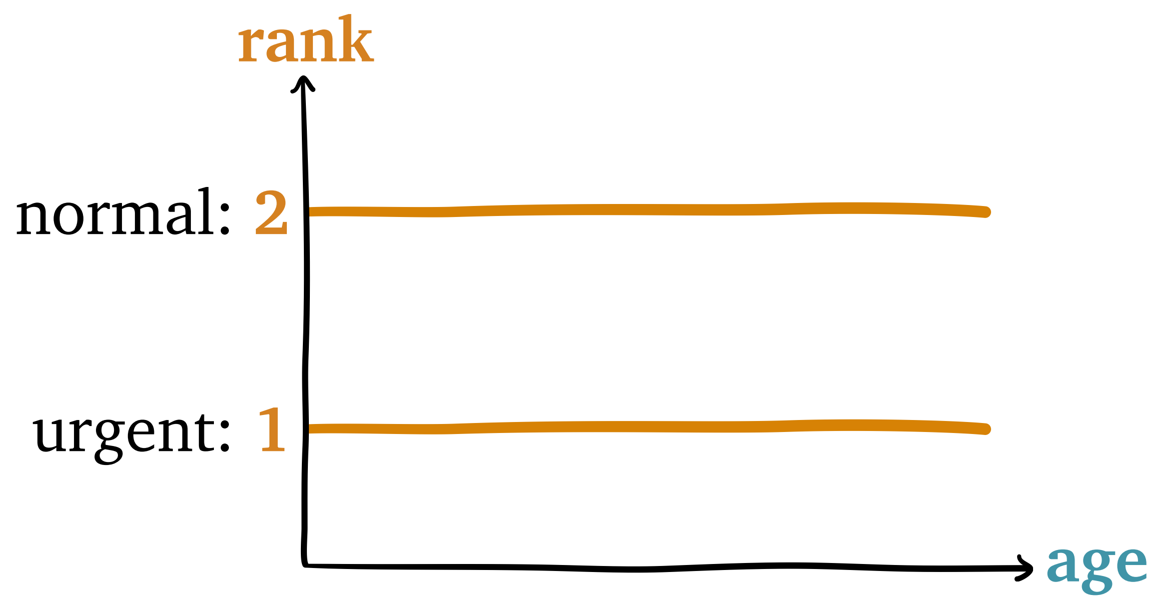
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] =$$
$$I_{0,\mathsf{U}}[1] =$$
$$I_{0,\mathsf{U}}[2-] =$$
$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] =$$
$$I_{0,\mathsf{N}}[1] =$$
$$I_{0,\mathsf{N}}[2-] =$$
$$I_{0,\mathsf{N}}[2] =$$

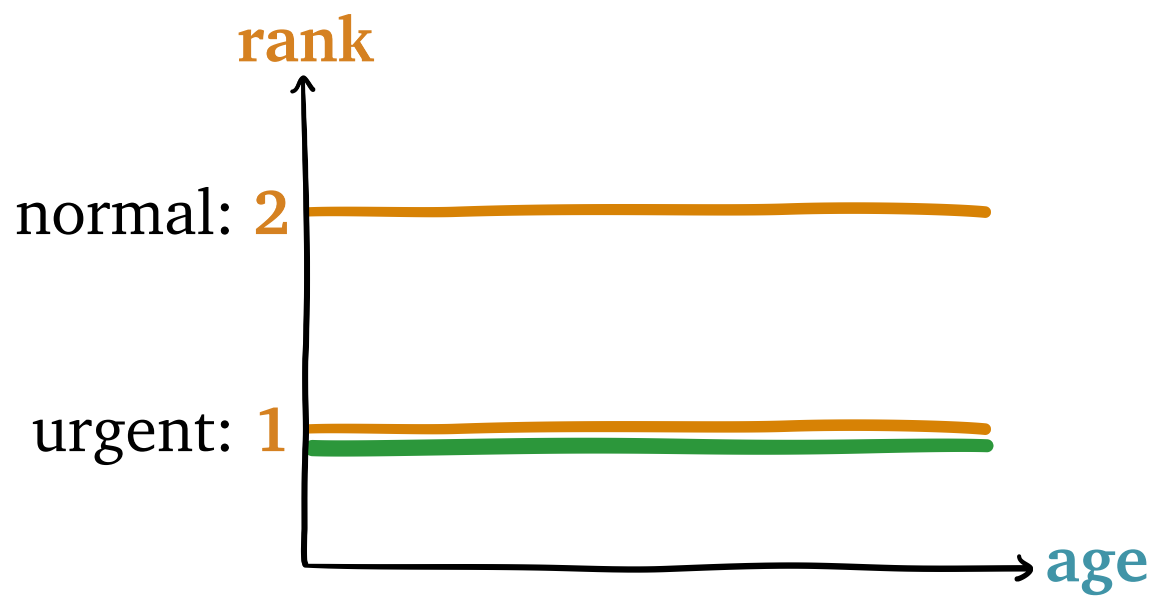
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_\mathsf{U}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_\mathsf{N}$




$$I_{0,\mathsf{U}}[1-] =$$
$$I_{0,\mathsf{U}}[1] =$$
$$I_{0,\mathsf{U}}[2-] =$$
$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] =$$
$$I_{0,\mathsf{N}}[1] =$$
$$I_{0,\mathsf{N}}[2-] =$$
$$I_{0,\mathsf{N}}[2] =$$

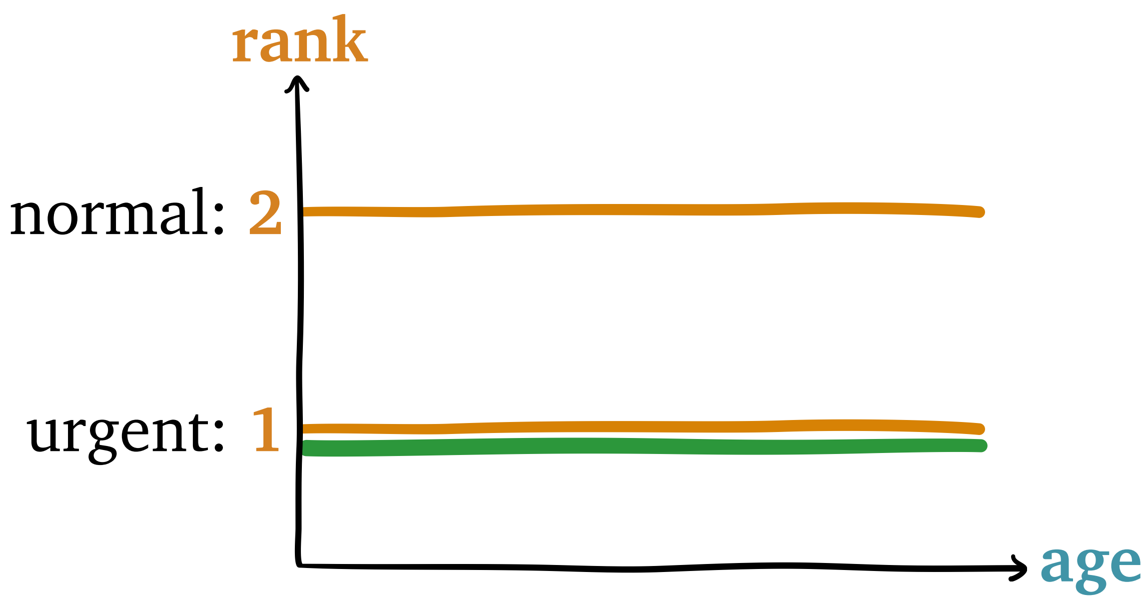
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_\mathsf{U}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_\mathsf{N}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$

$$I_{0,\mathsf{U}}[1] =$$

$$I_{0,\mathsf{U}}[2-] =$$

$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] =$$

$$I_{0,\mathsf{N}}[1] =$$

$$I_{0,\mathsf{N}}[2-] =$$

$$I_{0,\mathsf{N}}[2] =$$

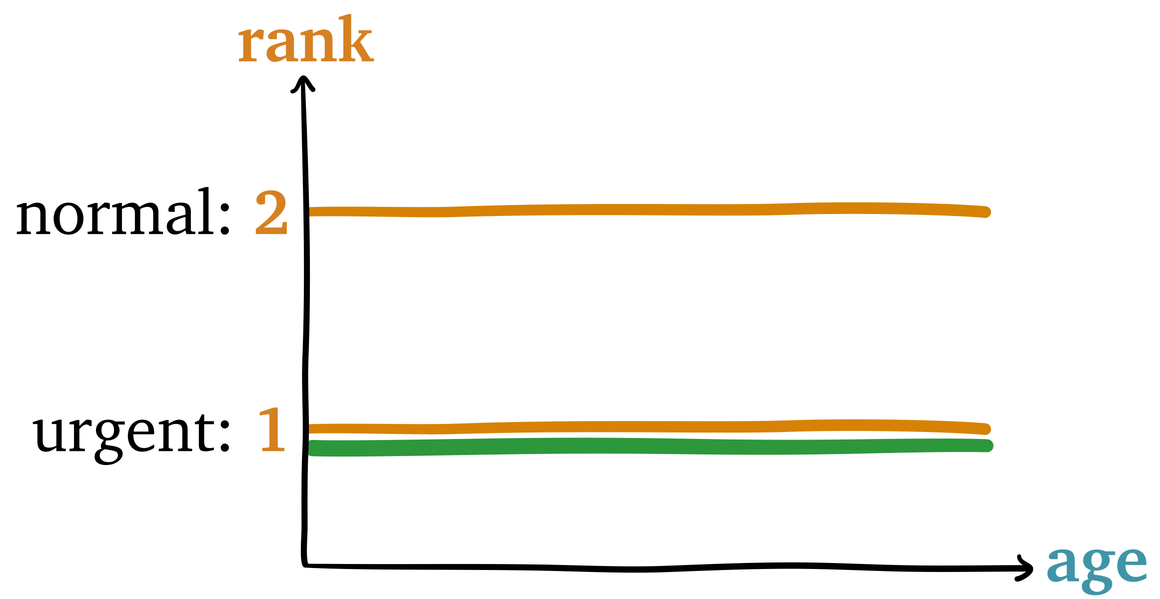
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$
$$I_{0,\mathsf{U}}[1] =$$
$$I_{0,\mathsf{U}}[2-] =$$
$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$
$$I_{0,\mathsf{N}}[1] =$$
$$I_{0,\mathsf{N}}[2-] =$$
$$I_{0,\mathsf{N}}[2] =$$

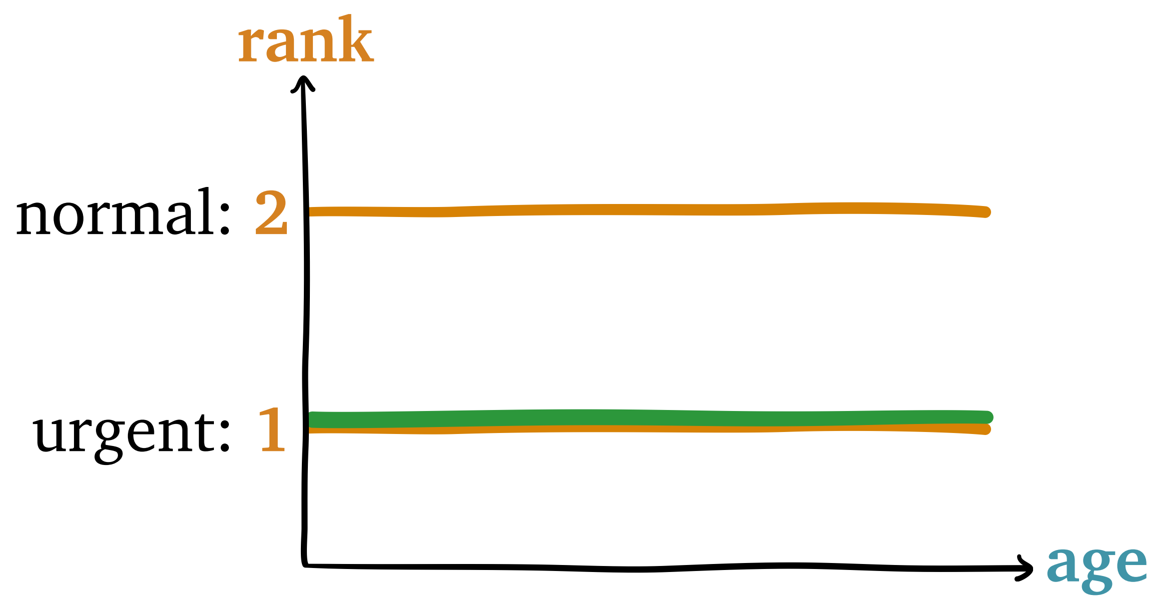
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$
$$I_{0,\mathsf{U}}[1] =$$
$$I_{0,\mathsf{U}}[2-] =$$
$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$
$$I_{0,\mathsf{N}}[1] =$$
$$I_{0,\mathsf{N}}[2-] =$$
$$I_{0,\mathsf{N}}[2] =$$

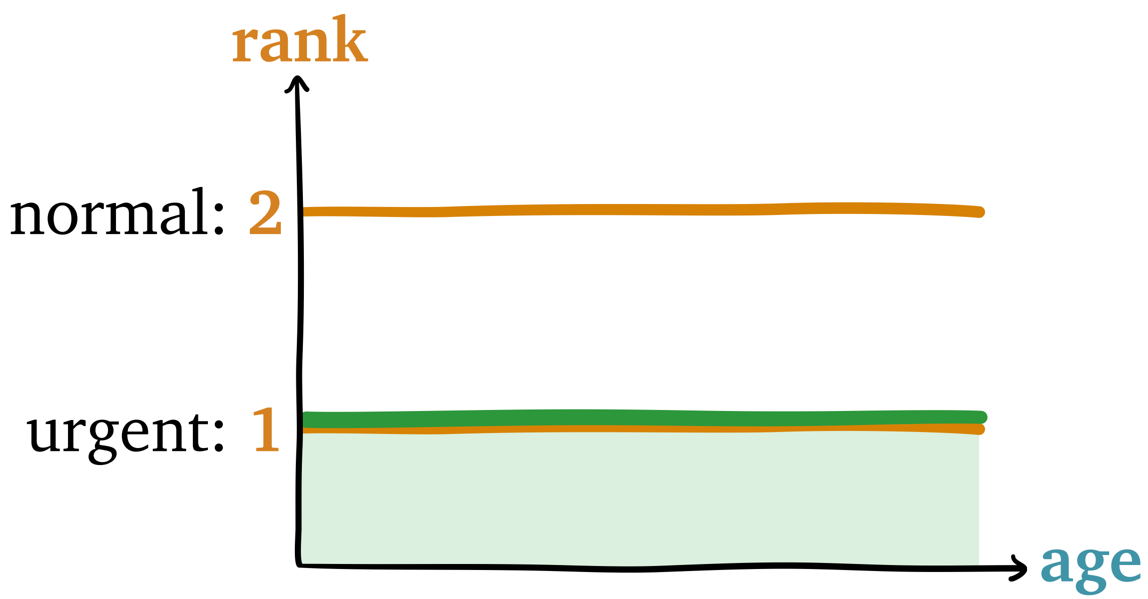
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$

$$I_{0,\mathsf{U}}[1] = [0, \infty)$$

$$I_{0,\mathsf{U}}[2-] =$$

$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$

$$I_{0,\mathsf{N}}[1] =$$

$$I_{0,\mathsf{N}}[2-] =$$

$$I_{0,\mathsf{N}}[2] =$$

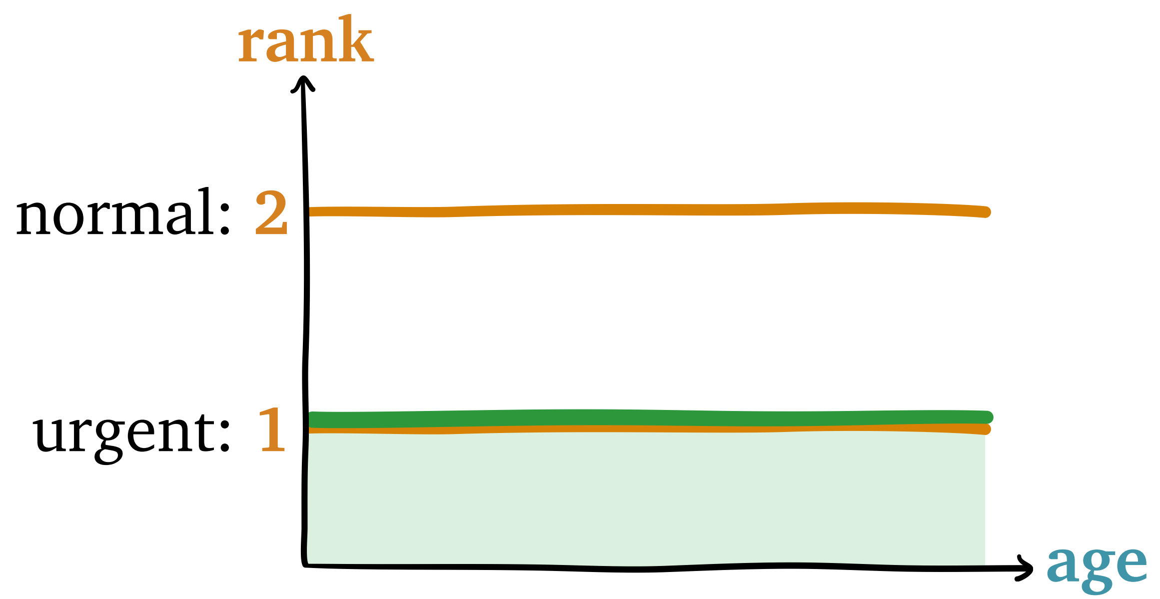
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$
$$I_{0,\mathsf{U}}[1] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2-] =$$
$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$
$$I_{0,\mathsf{N}}[1] = \emptyset$$
$$I_{0,\mathsf{N}}[2-] =$$
$$I_{0,\mathsf{N}}[2] =$$

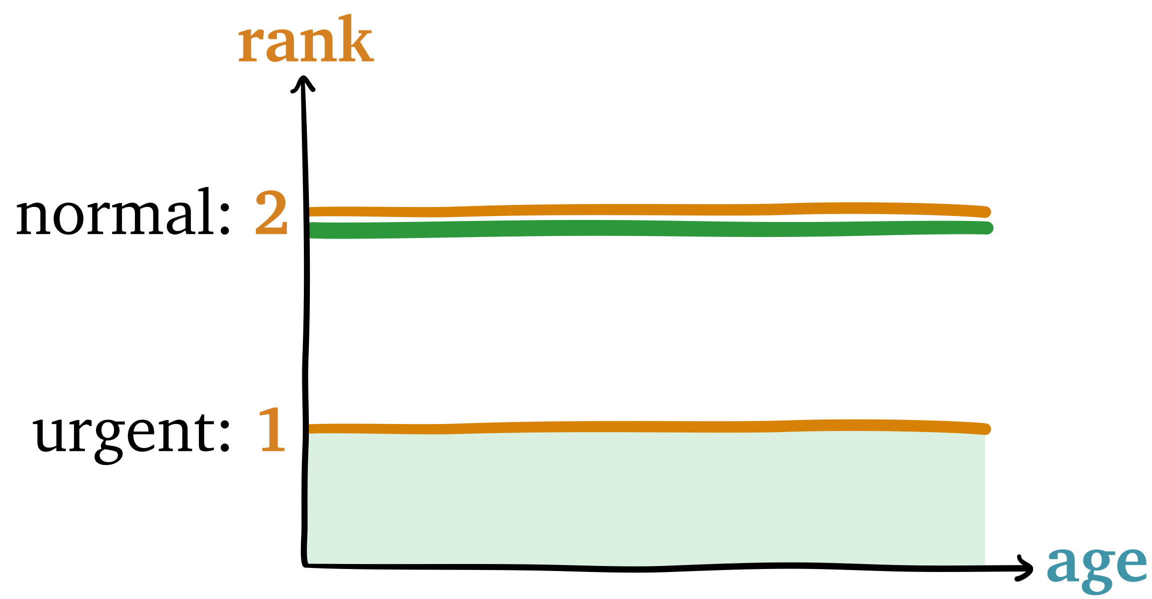
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$



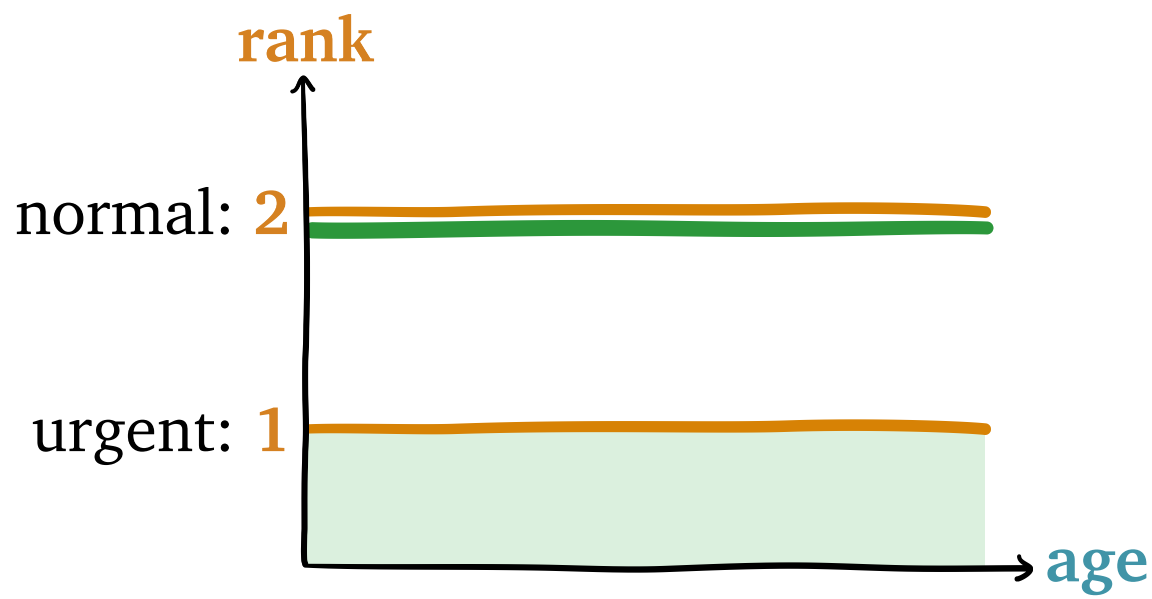

$$I_{0,\mathsf{U}}[1-] = \emptyset$$
$$I_{0,\mathsf{U}}[1] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2-] =$$
$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$
$$I_{0,\mathsf{N}}[1] = \emptyset$$
$$I_{0,\mathsf{N}}[2-] =$$
$$I_{0,\mathsf{N}}[2] =$$

# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)
- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)
- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$
$$I_{0,\mathsf{U}}[1] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2-] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$
$$I_{0,\mathsf{N}}[1] = \emptyset$$
$$I_{0,\mathsf{N}}[2-] =$$
$$I_{0,\mathsf{N}}[2] =$$

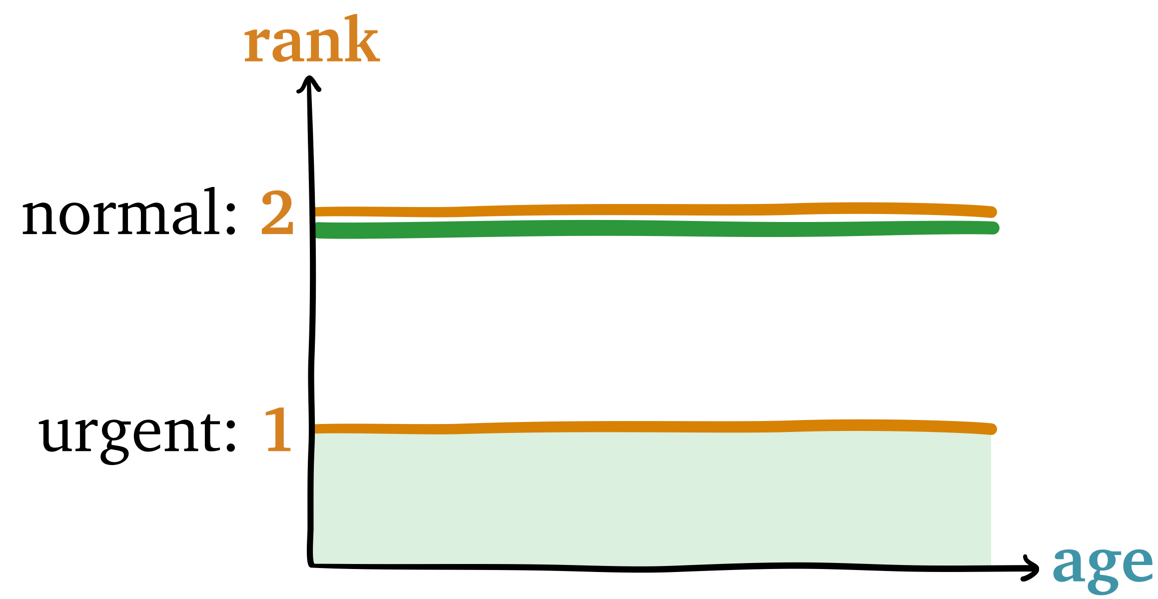
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$
$$I_{0,\mathsf{U}}[1] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2-] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$
$$I_{0,\mathsf{N}}[1] = \emptyset$$
$$I_{0,\mathsf{N}}[2-] = \emptyset$$
$$I_{0,\mathsf{N}}[2] =$$

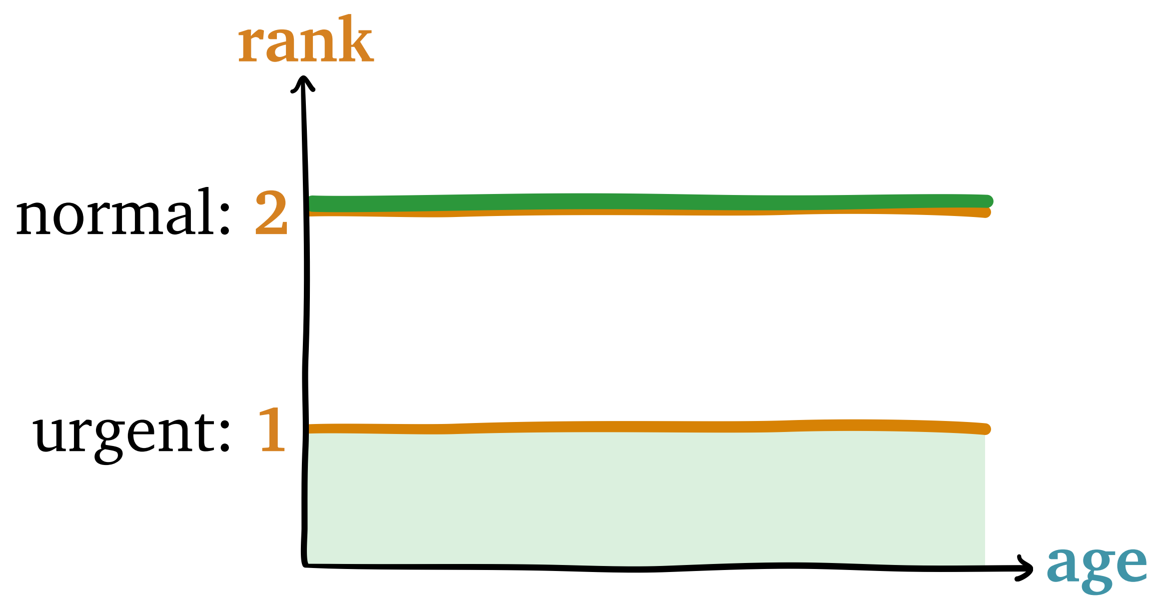
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$
$$I_{0,\mathsf{U}}[1] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2-] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2] =$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$
$$I_{0,\mathsf{N}}[1] = \emptyset$$
$$I_{0,\mathsf{N}}[2-] = \emptyset$$
$$I_{0,\mathsf{N}}[2] =$$

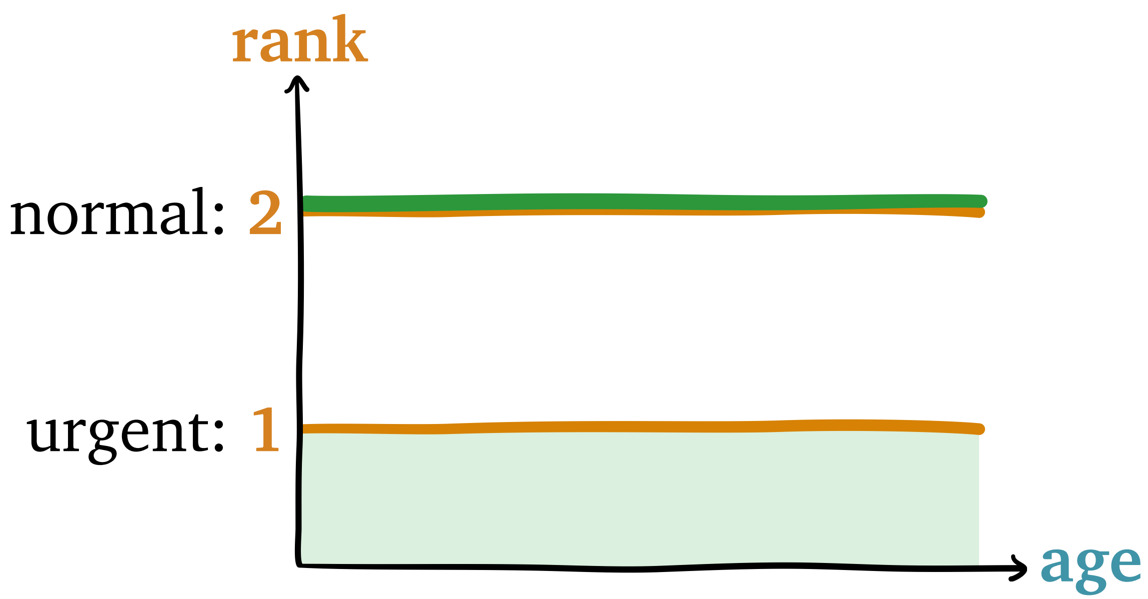
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$
$$I_{0,\mathsf{U}}[1] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2-] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2] = [0, \infty)$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$
$$I_{0,\mathsf{N}}[1] = \emptyset$$
$$I_{0,\mathsf{N}}[2-] = \emptyset$$
$$I_{0,\mathsf{N}}[2] =$$

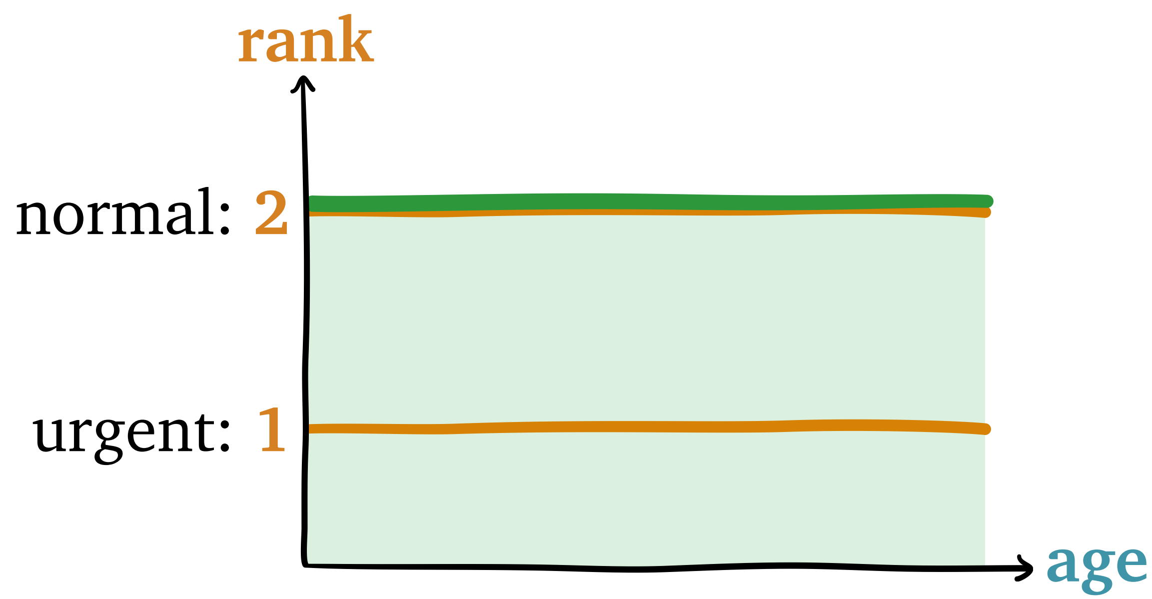
# Example: Preemptive Priority

**Urgent** ($d = \mathsf{U}$, $r = 1$)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d = \mathsf{N}$, $r = 2$)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$I_{0,\mathsf{U}}[1-] = \emptyset$$
$$I_{0,\mathsf{U}}[1] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2-] = [0, \infty)$$
$$I_{0,\mathsf{U}}[2] = [0, \infty)$$

$$I_{0,\mathsf{N}}[1-] = \emptyset$$
$$I_{0,\mathsf{N}}[1] = \emptyset$$
$$I_{0,\mathsf{N}}[2-] = \emptyset$$
$$I_{0,\mathsf{N}}[2] = [0, \infty)$$

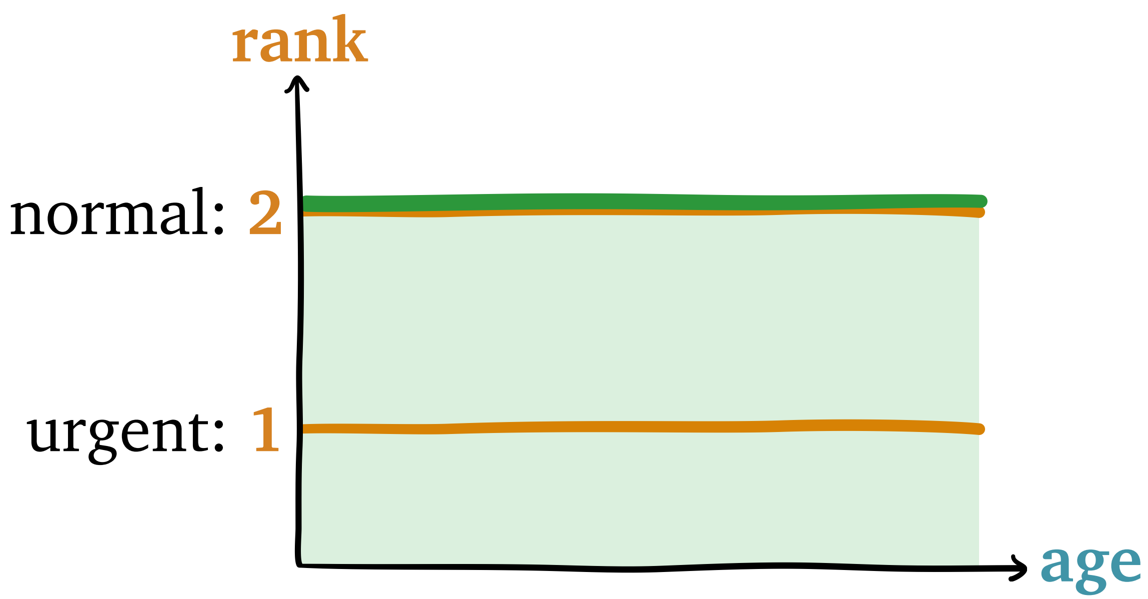
# Example: Preemptive Priority

**Urgent** ($d$ = U, $r$ = 1)

- 1/4 of all jobs
- Size distribution $X_{\mathsf{U}}$

**Normal** ($d$ = N, $r$ = 2)

- 3/4 of all jobs
- Size distribution $X_{\mathsf{N}}$




$$X_{0,\mathsf{U}}[1-] = 0$$
$$X_{0,\mathsf{U}}[1] = X_{\mathsf{U}}$$
$$X_{0,\mathsf{U}}[2-] = X_{\mathsf{U}}$$
$$X_{0,\mathsf{U}}[2] = X_{\mathsf{U}}$$

$$X_{0,\mathsf{N}}[1-] = 0$$
$$X_{0,\mathsf{N}}[1] = 0$$
$$X_{0,\mathsf{N}}[2-] = 0$$
$$X_{0,\mathsf{N}}[2] = X_{\mathsf{N}}$$

# Example: Preemptive Priority

$$X_0[1-] = 0$$

$$X_0[1] = \begin{cases} X_\mathsf{U} & \text{w.p. } \frac{1}{4} \\ 0 & \text{w.p. } \frac{3}{4} \end{cases}$$

$$X_0[2-] = \begin{cases} X_\mathsf{U} & \text{w.p. } \frac{1}{4} \\ 0 & \text{w.p. } \frac{3}{4} \end{cases}$$

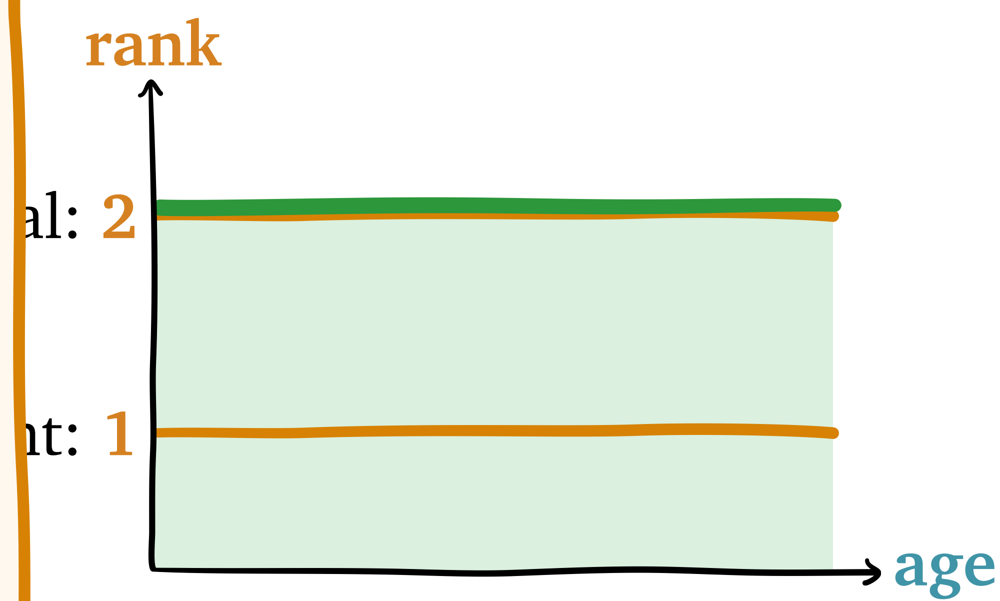
$$X_0[2] = \begin{cases} X_\mathsf{U} & \text{w.p. } \frac{1}{4} \\ X_\mathsf{N} & \text{w.p. } \frac{3}{4} \end{cases}$$

**Urge**

- 1/4
- Siz

**Norn**

- 3/4
- Siz




$X_{0,\mathsf{U}}$

$X_0$

$$X_{0,\mathsf{U}}[2-] = X_\mathsf{U}$$

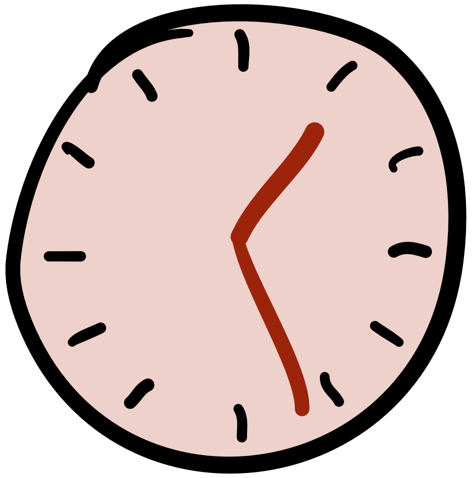
$$X_{0,\mathsf{U}}[2] = X_\mathsf{U}$$

$$X_{0,\mathsf{N}}[1-] = 0$$

$$X_{0,\mathsf{N}}[1] = 0$$

$$X_{0,\mathsf{N}}[2-] = 0$$

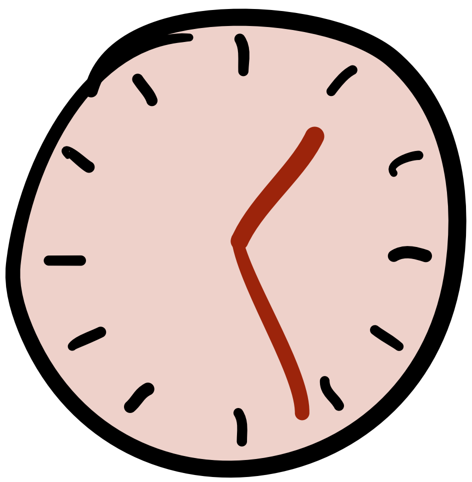
$$X_{0,\mathsf{N}}[2] = X_\mathsf{N}$$



# Part 2:

*analyzing* **SOAP** policies

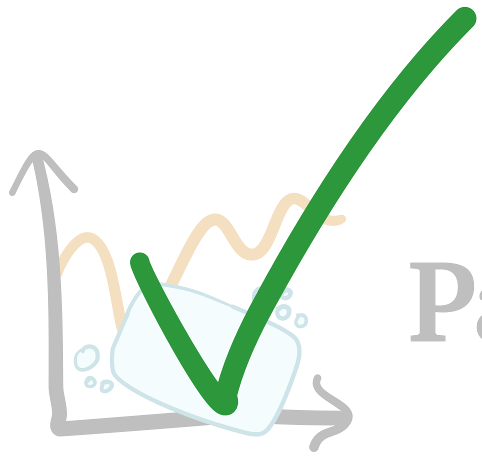
Practice!



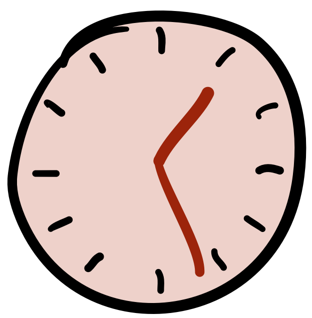
# Part 2:

*analyzing* **SOAP** policies

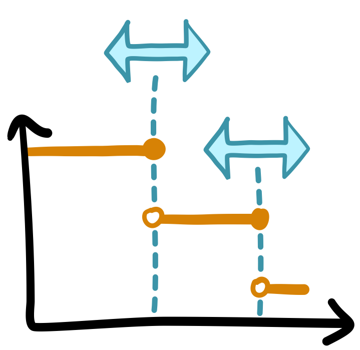
# Outline



**Part 1**: *defining* **SOAP** policies



**Part 2**: *analyzing* **SOAP** policies



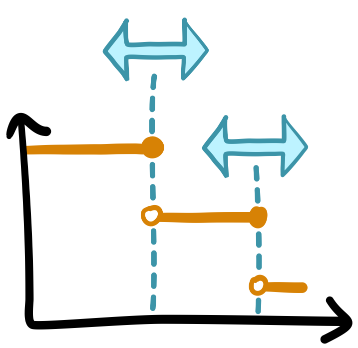
**Part 3**: *policy design* with **SOAP**

**Part 4**: *optimality proofs* with **SOAP**

# Outline

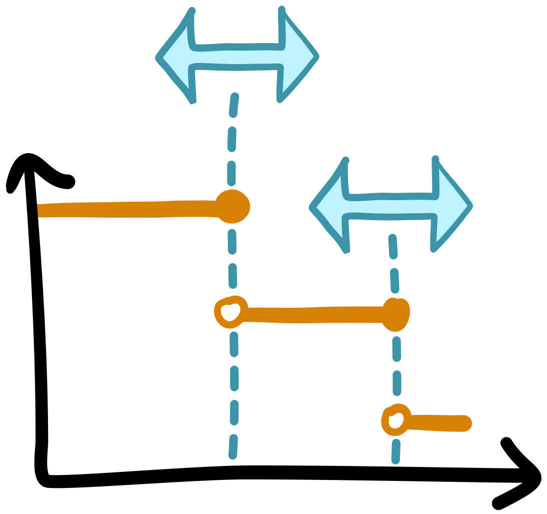
**Part 1**: *defining* **SOAP** policies

**Part 2**: *analyzing* **SOAP** policies



**Part 3**: *policy design* with **SOAP**

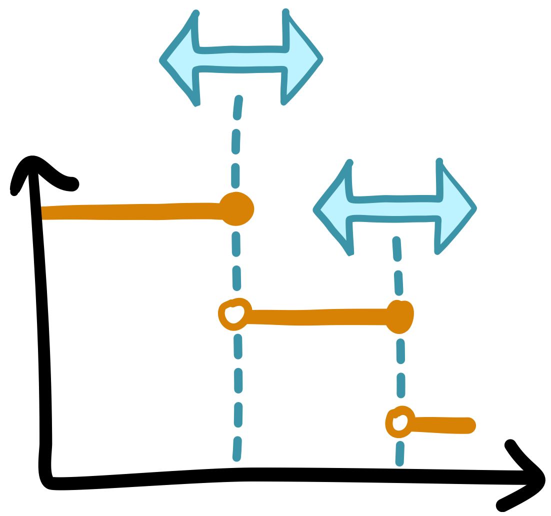
**Part 4**: *optimality proofs* with **SOAP**



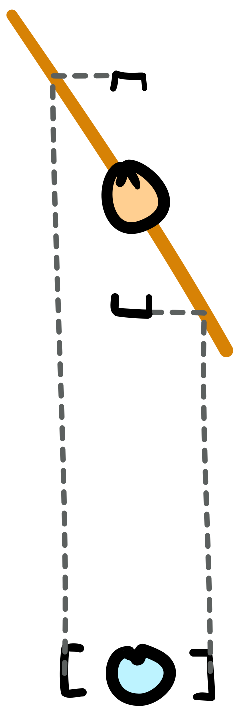
# Part 3:

*policy design* with **SOAP**

# Two Design Problems

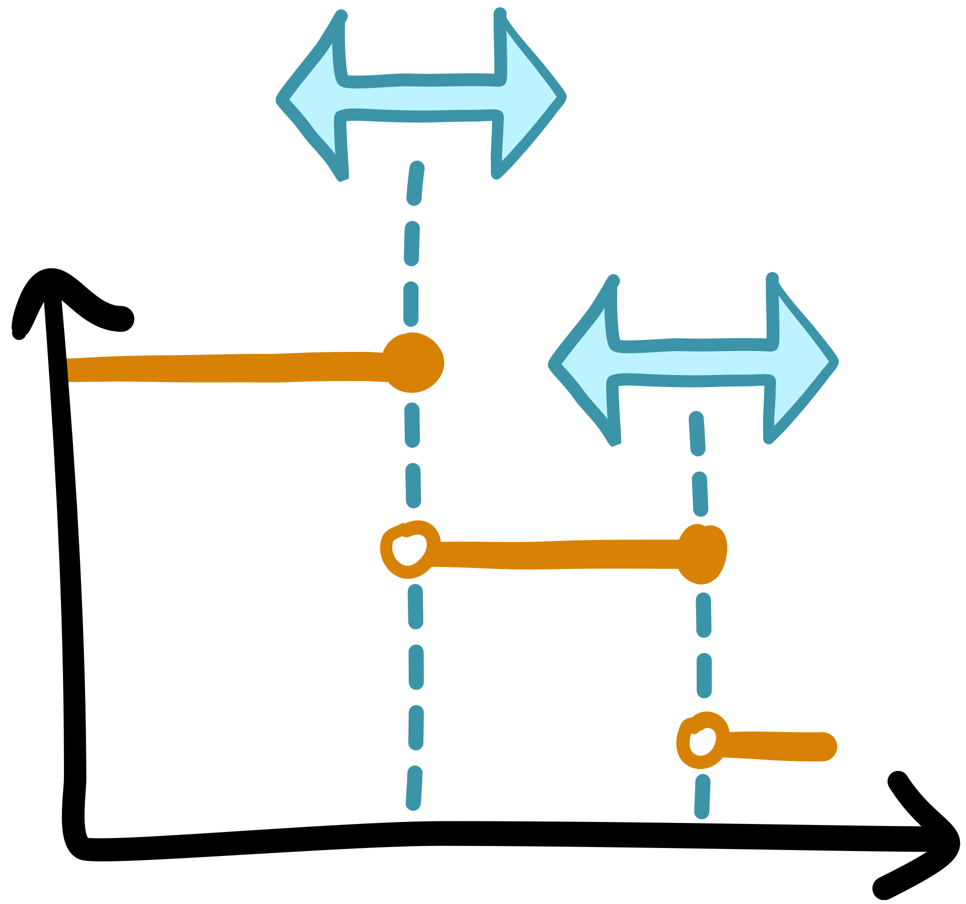


**Bucketed SRPT**



**Noisy Systems**

# Bucketed SRPT



**Question**: given number of priority levels, which job sizes go in which size buckets?

# Two Buckets

$X =$ bounded Pareto on $[1, 10^6]$ with $\alpha = 1$

# Two Buckets

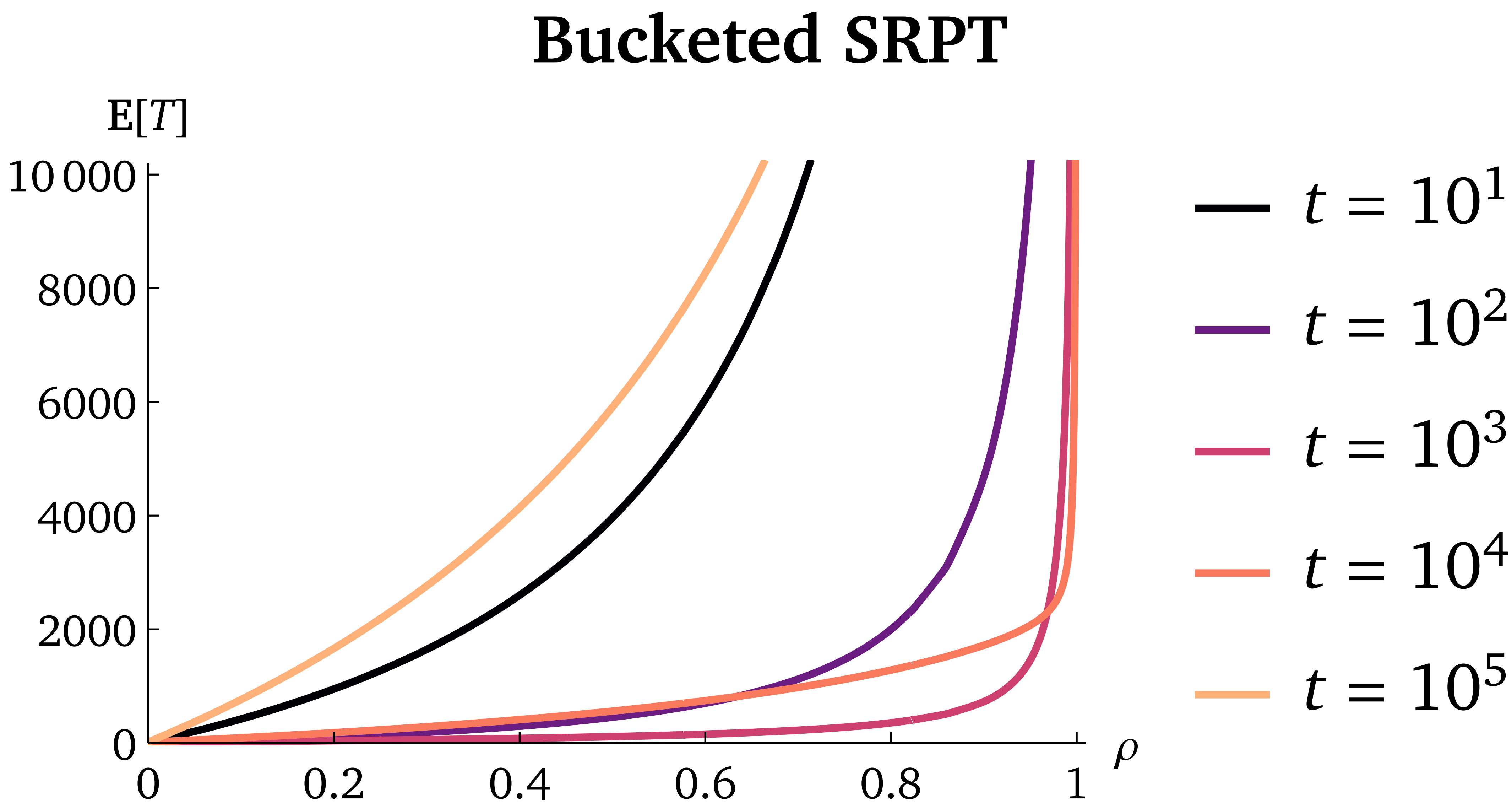
$X =$ bounded Pareto on $[1, 10^6]$ with $\alpha = 1$

$t =$ threshold between buckets

# Two Buckets

$X =$ bounded Pareto on $[1, 10^6]$ with $\alpha = 1$

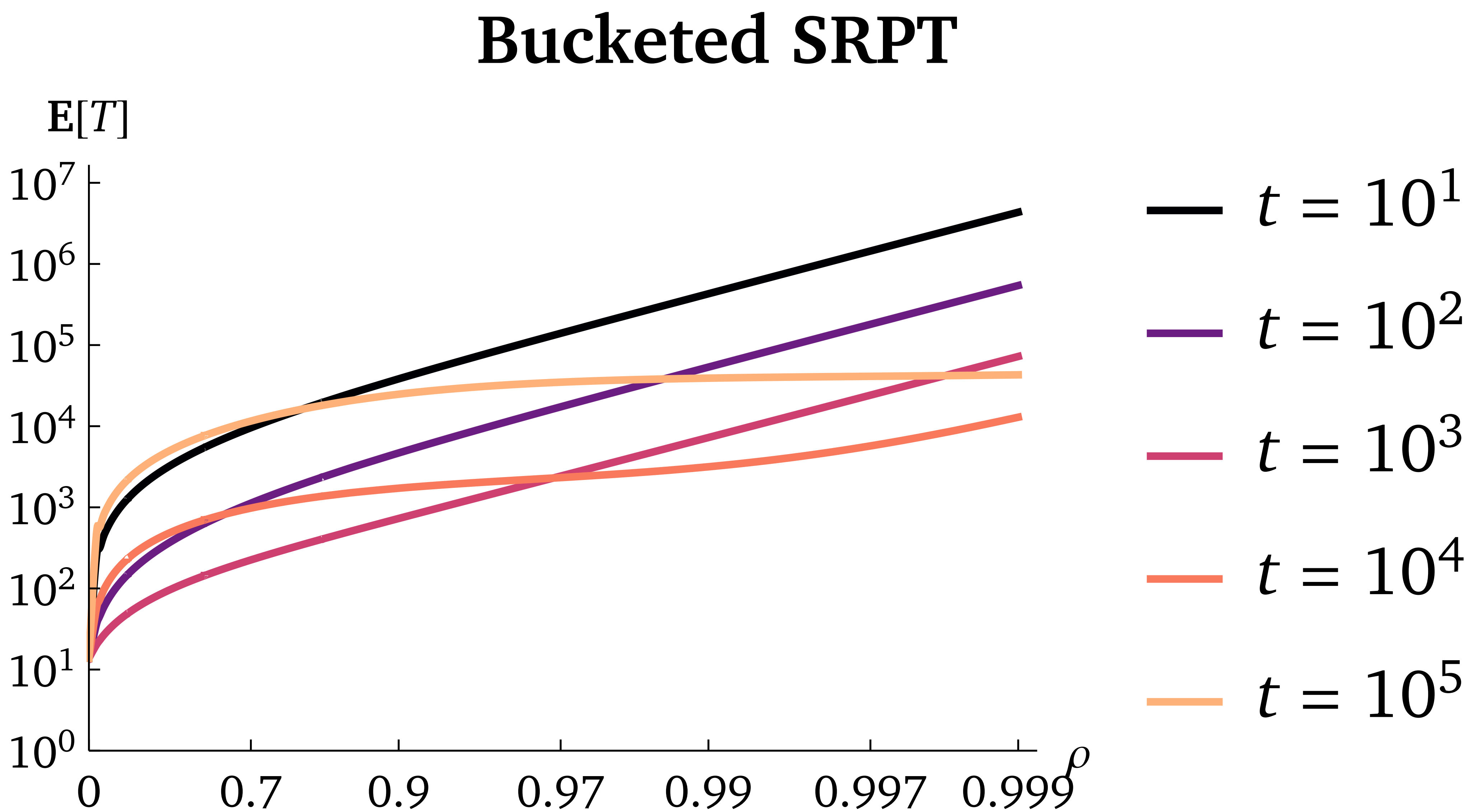
$t =$ threshold between buckets



# Two Buckets

$X$ = bounded Pareto on $[1, 10^6]$ with $\alpha = 1$

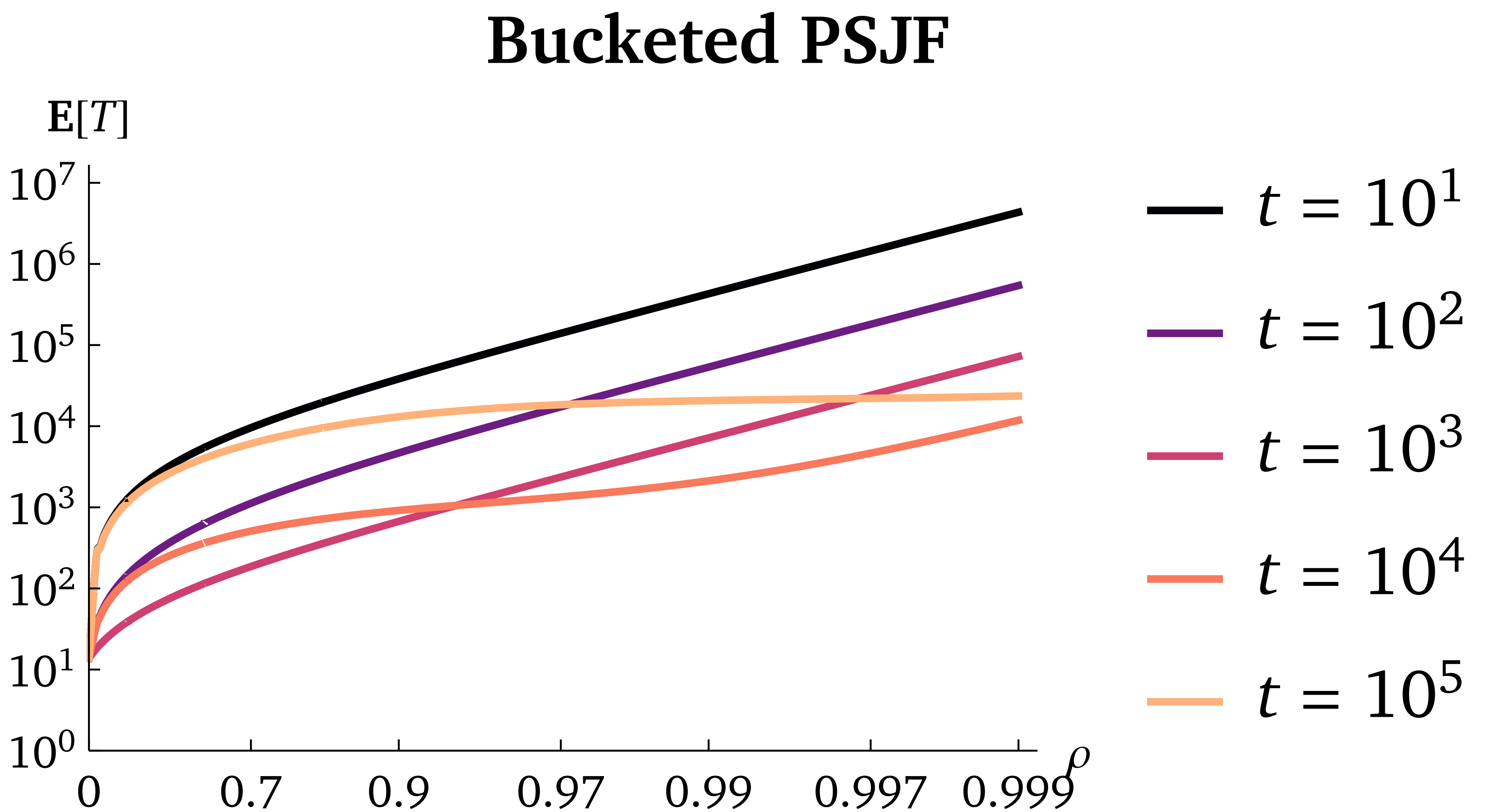
$t$ = threshold between buckets



# Two Buckets

$X$ = bounded Pareto on $[1, 10^6]$ with $\alpha = 1$

$t$ = threshold between buckets

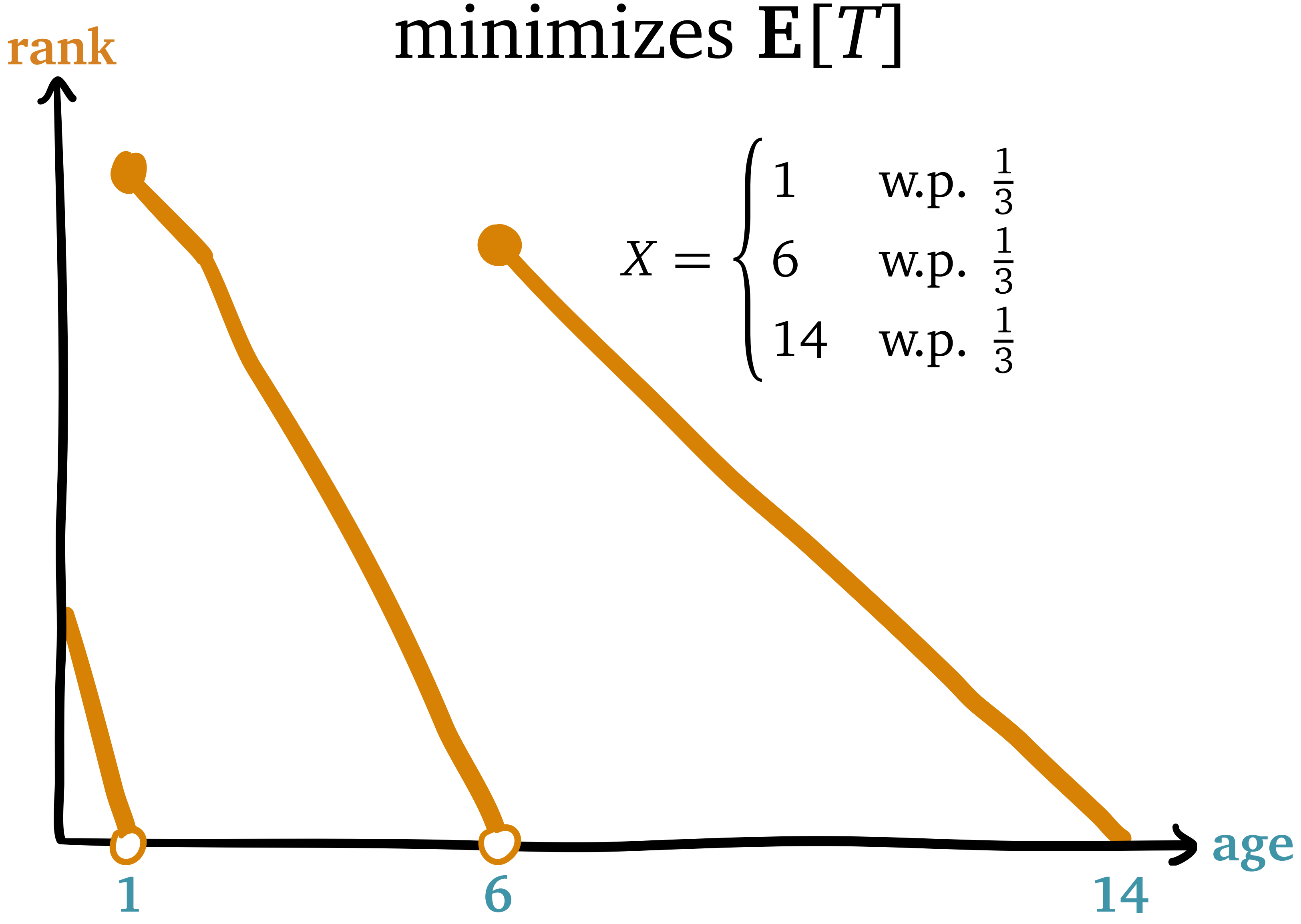


# Noisy System

# Noisy System

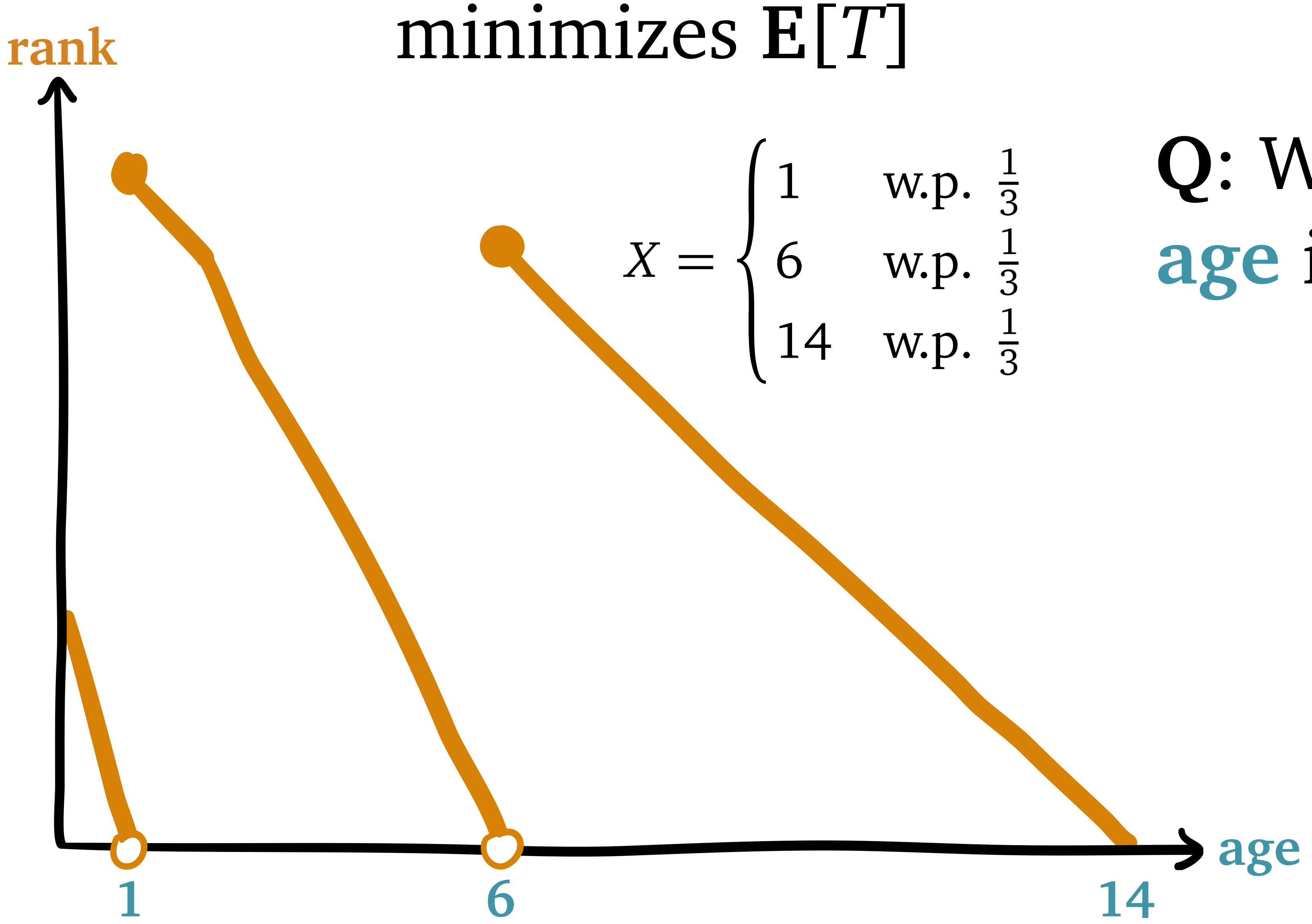
**Gittins**
minimizes $\mathbf{E}[T]$

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$



# Noisy System

**Gittins**
minimizes $\mathbf{E}[T]$



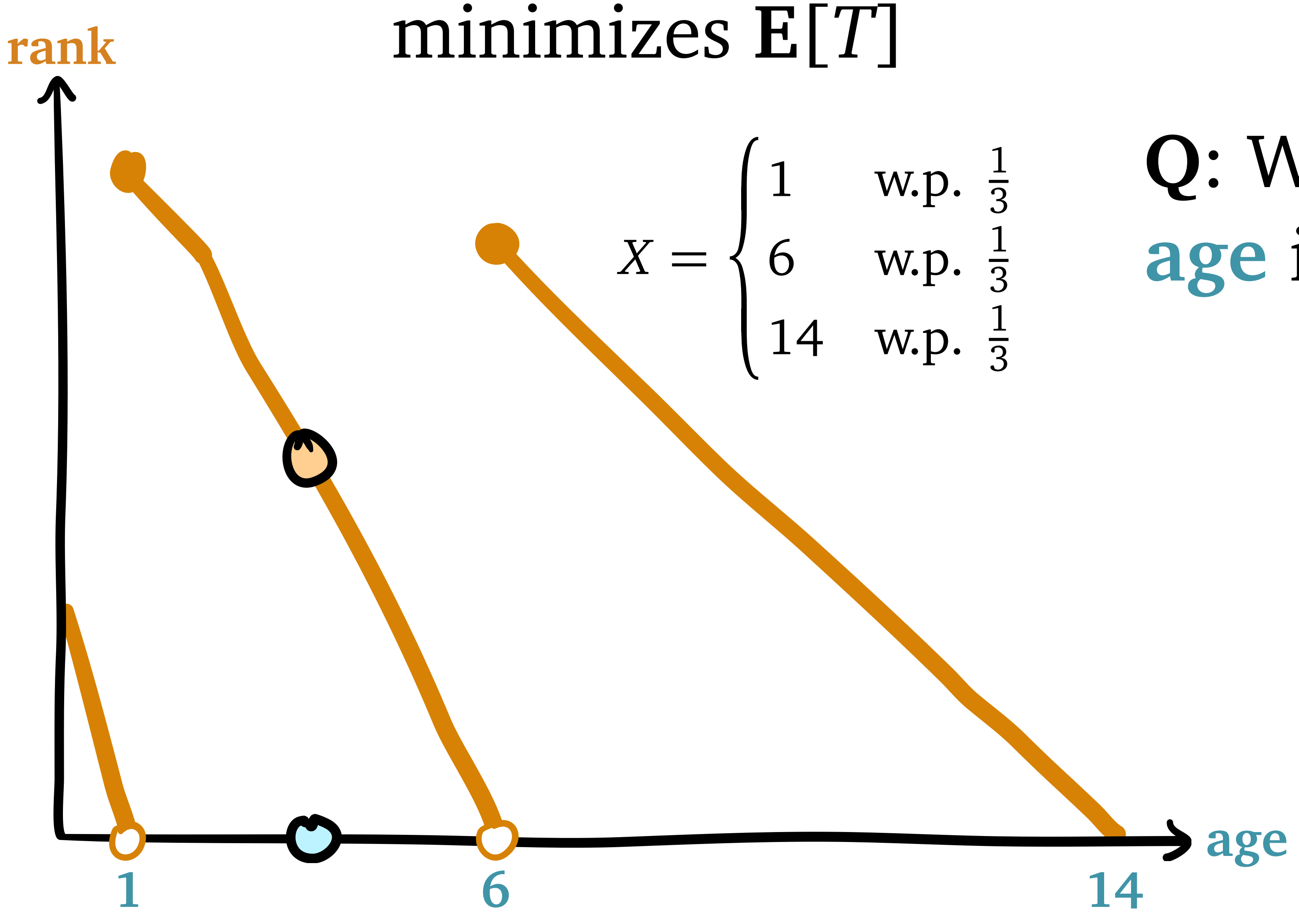

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

**Q**: What if we have noisy **age** information?

# Noisy System

**Gittins**
minimizes $\mathbf{E}[T]$

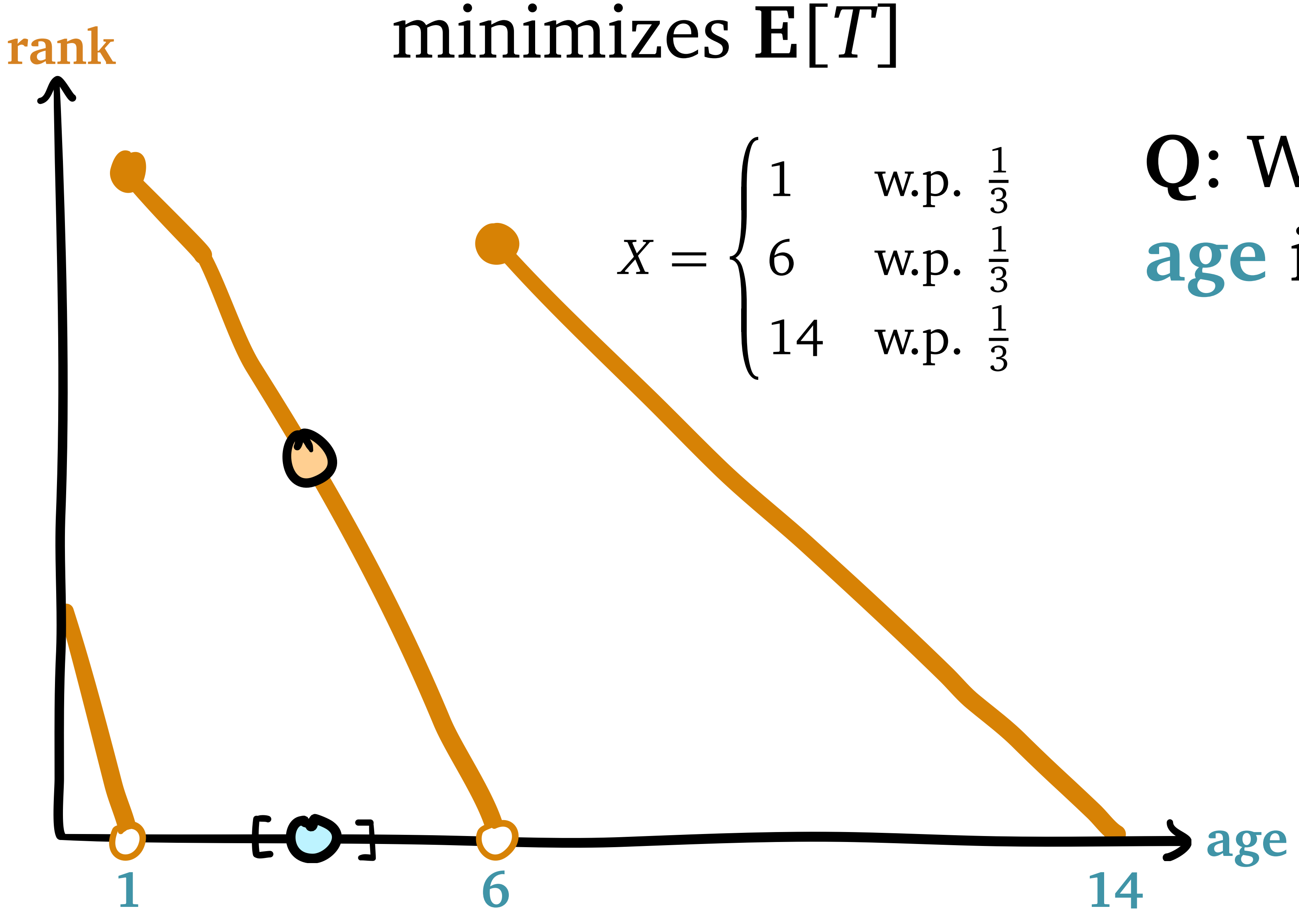
$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$




**Q**: What if we have noisy **age** information?

# Noisy System

**Gittins**
minimizes $\mathbf{E}[T]$

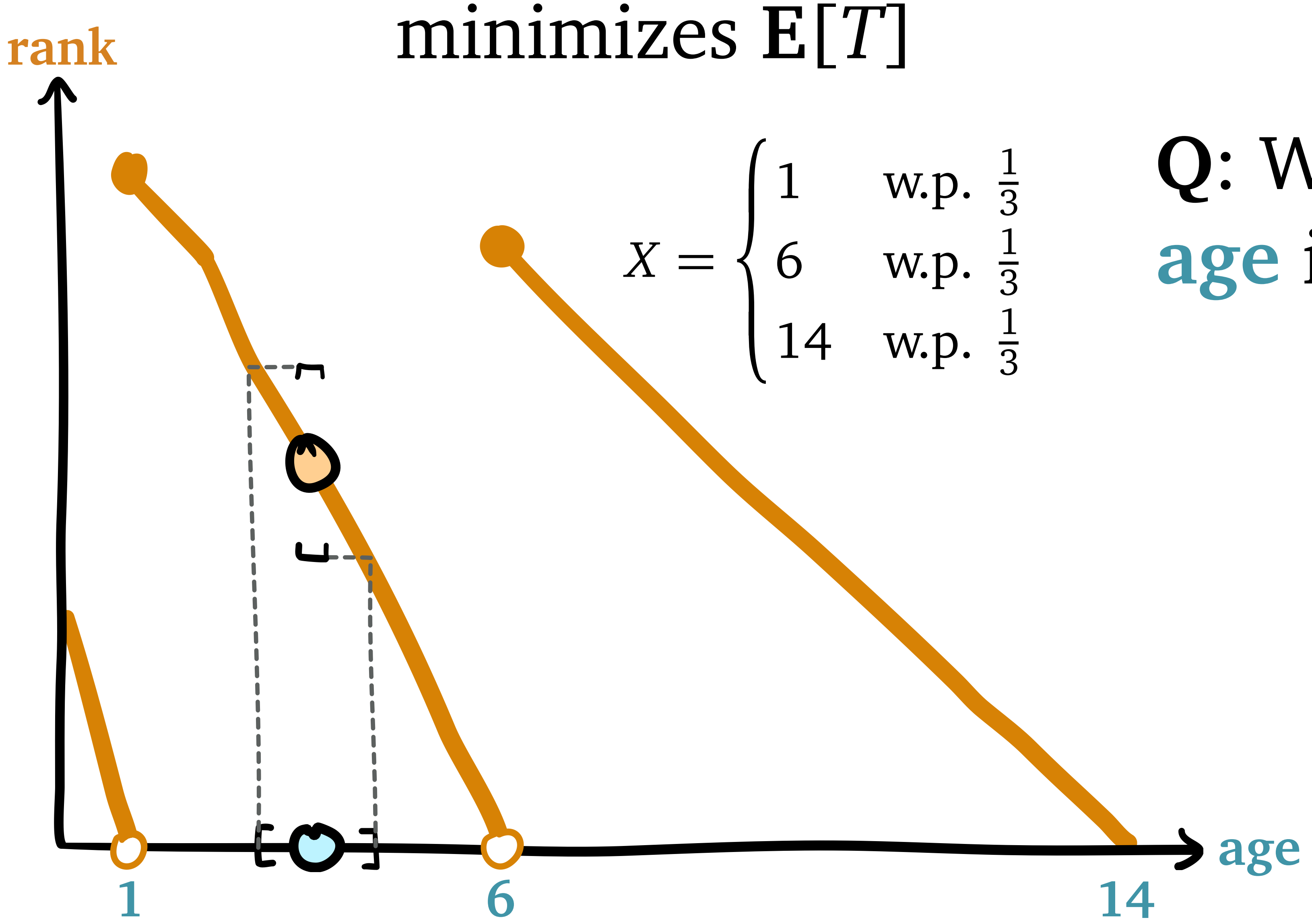



$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

**Q**: What if we have noisy **age** information?

# Noisy System

**Gittins**
minimizes $\mathbf{E}[T]$

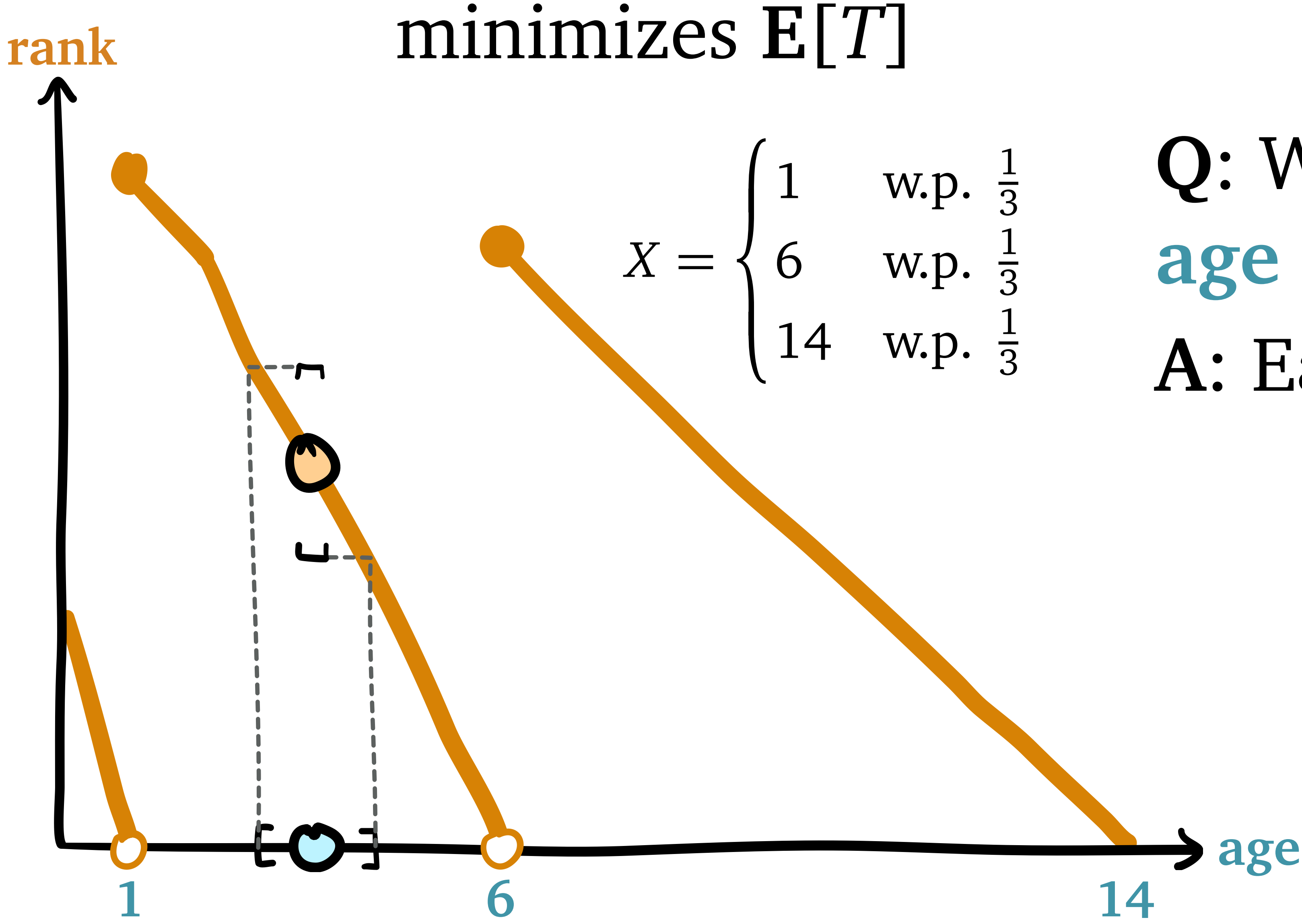



$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

**Q**: What if we have noisy **age** information?

# Noisy System

**Gittins**
minimizes $\mathbf{E}[T]$



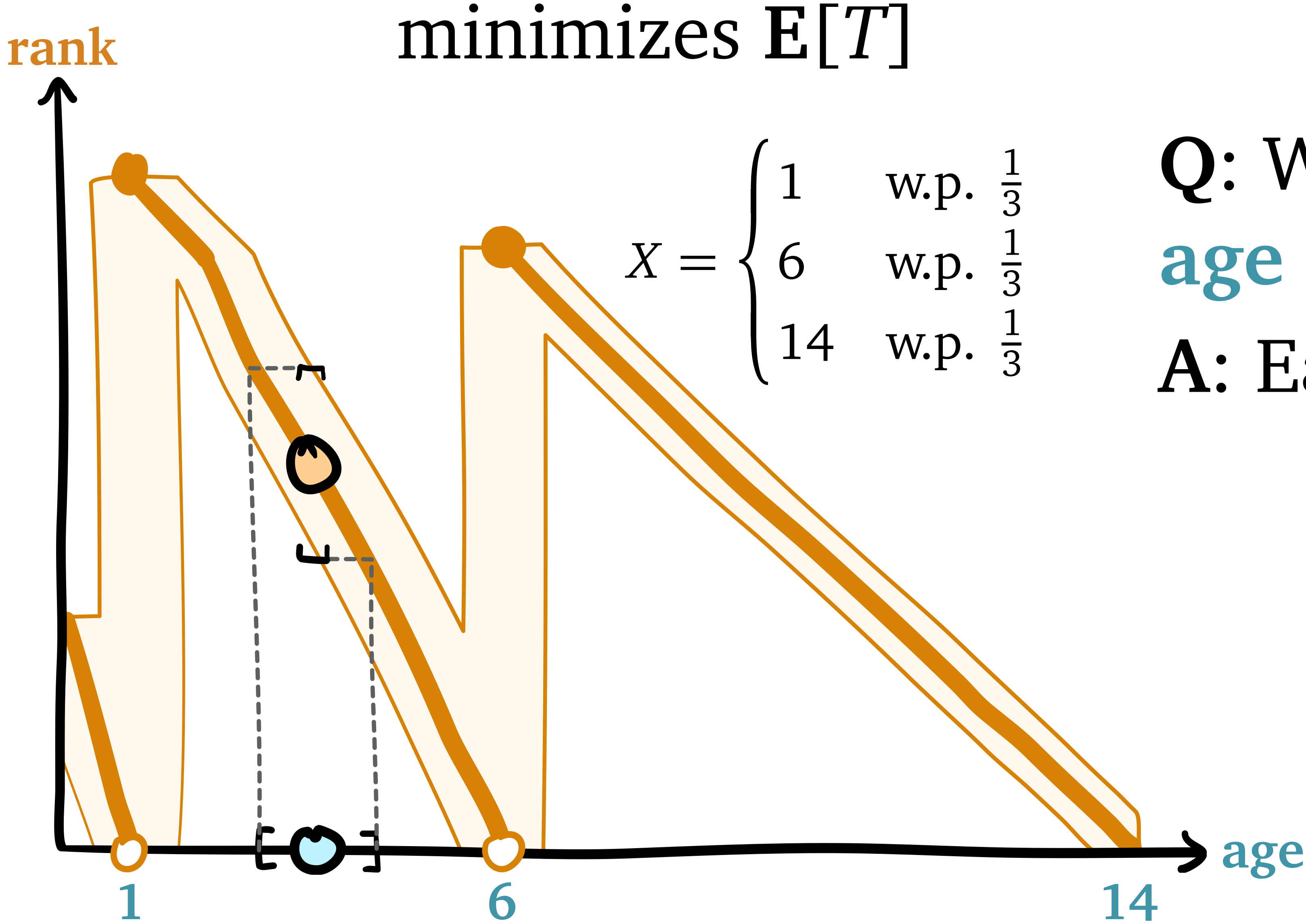

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

**Q**: What if we have noisy **age** information?

**A**: Each **age** has ***rank*** *range*

# Noisy System

**Gittins**
minimizes $\mathbf{E}[T]$




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

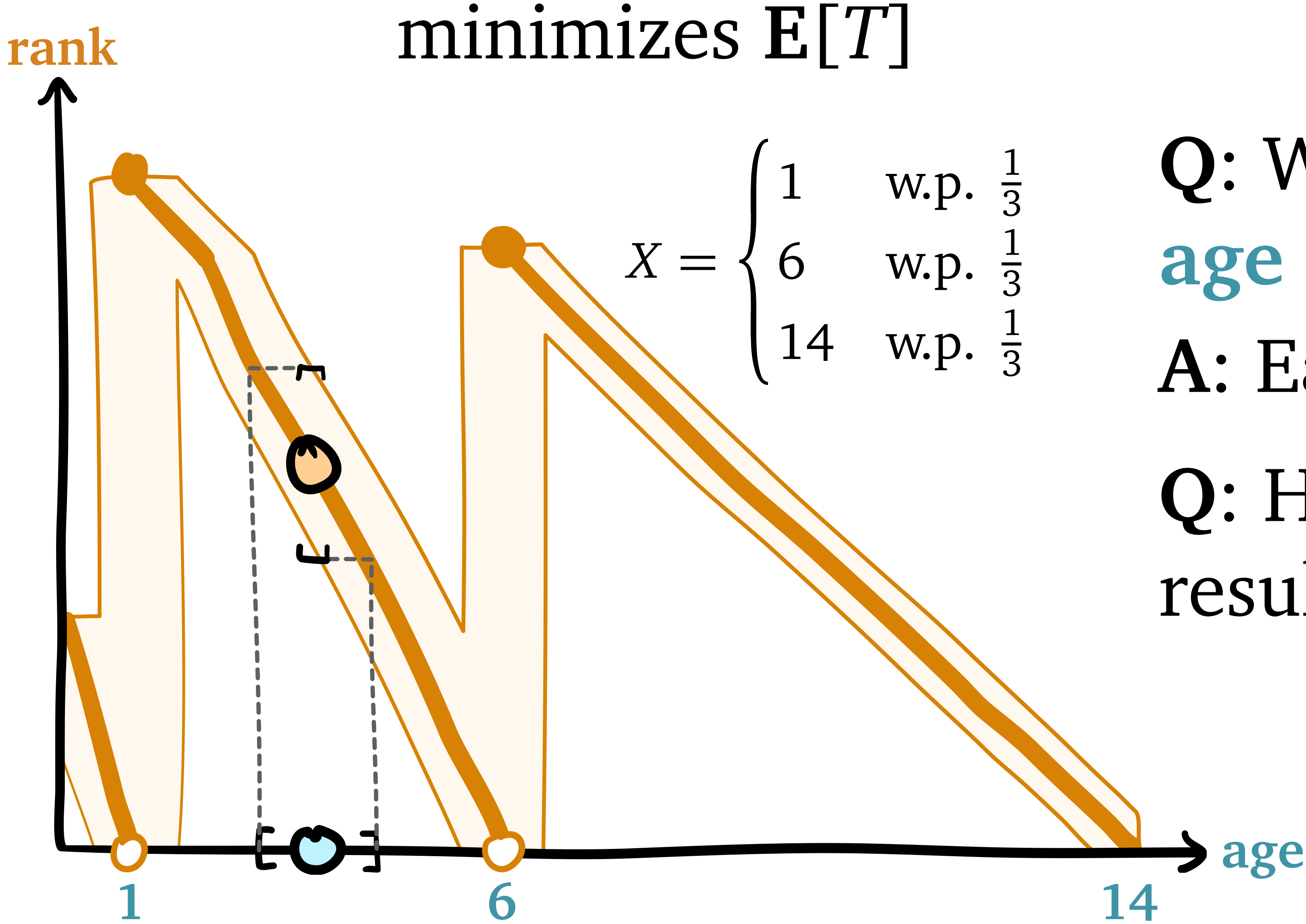
**Q**: What if we have noisy **age** information?

**A**: Each **age** has ***rank*** *range*

# Noisy System

**Gittins**
minimizes $\mathbf{E}[T]$

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$




**Q**: What if we have noisy **age** information?

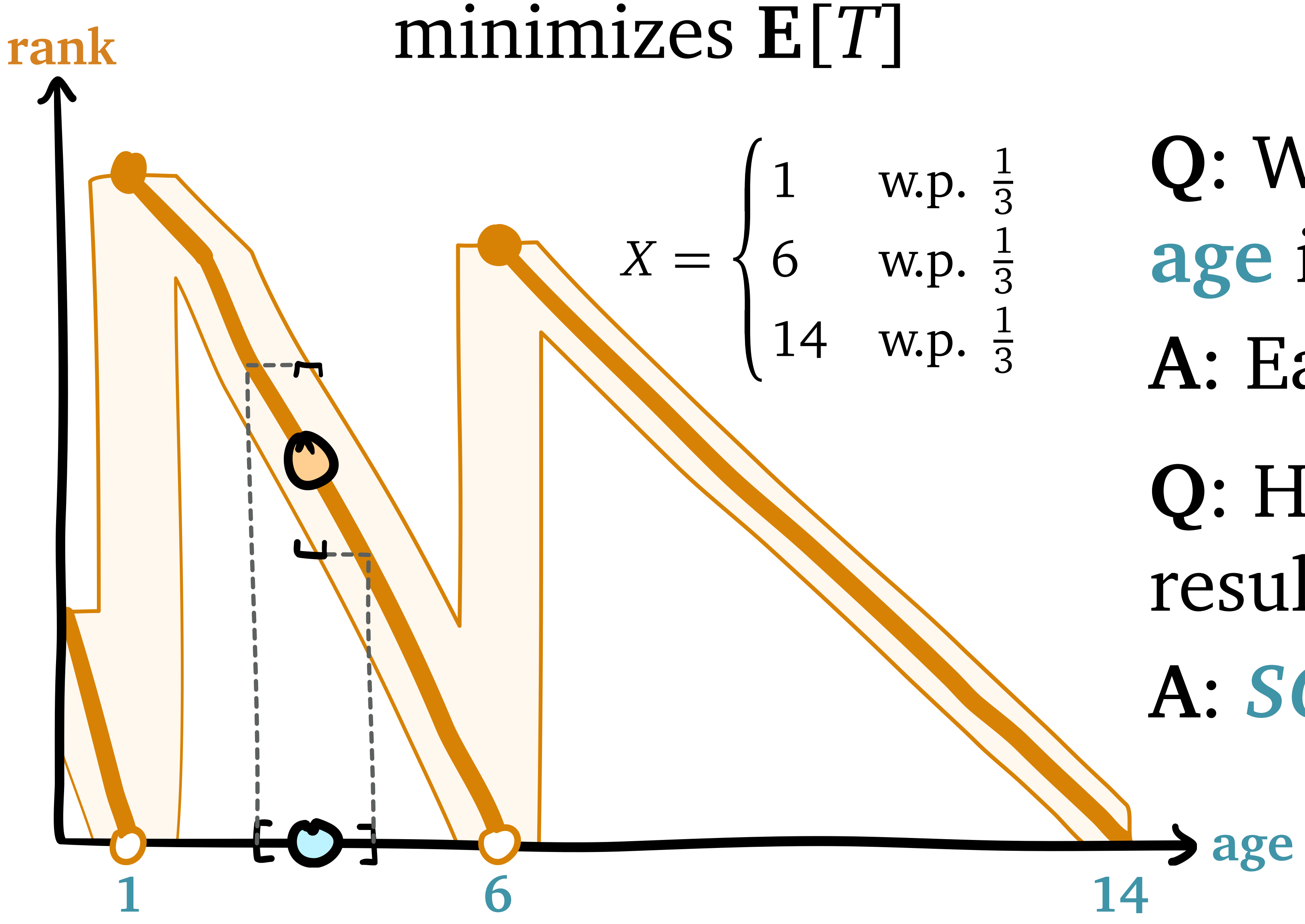
**A**: Each **age** has ***rank*** *range*

**Q**: How do we analyze resulting scheduling policy?

# Noisy System

**Gittins**

minimizes $\mathbf{E}[T]$




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

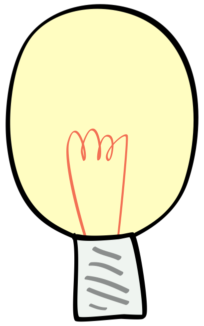
**Q**: What if we have noisy **age** information?

**A**: Each **age** has ***rank*** *range*

**Q**: How do we analyze resulting scheduling policy?

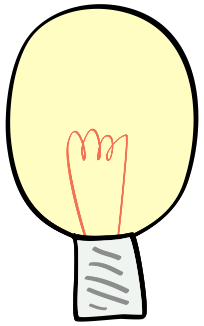
**A**: ***SOAP*** *Bubble* analysis

# SOAP Bubble Analysis



**Idea**: do tagged job analysis, but…

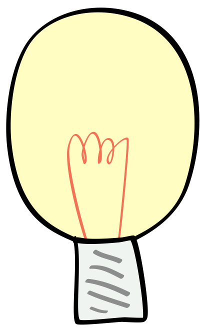
# SOAP Bubble Analysis



**Idea**: do tagged job analysis, but…

- I get *worst* possible **rank**

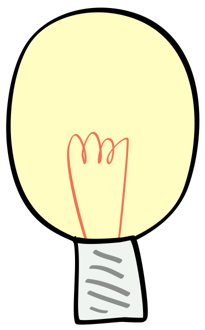
# **SOAP** Bubble Analysis



**Idea**: do tagged job analysis, but…

- I get *worst* possible **rank**
- Everyone else gets *best* possible **rank**

# **SOAP** Bubble Analysis

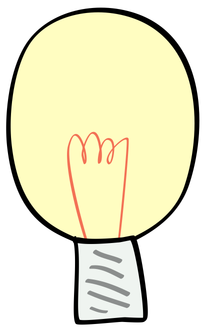


**Idea**: do tagged job analysis, but…

- I get *worst* possible **rank**
- Everyone else gets *best* possible **rank**

**Theorem**: this *always* gives an upper bound on $\mathbf{E}[T]$

# SOAP Bubble Analysis



**Idea**: do tagged job analysis, but…

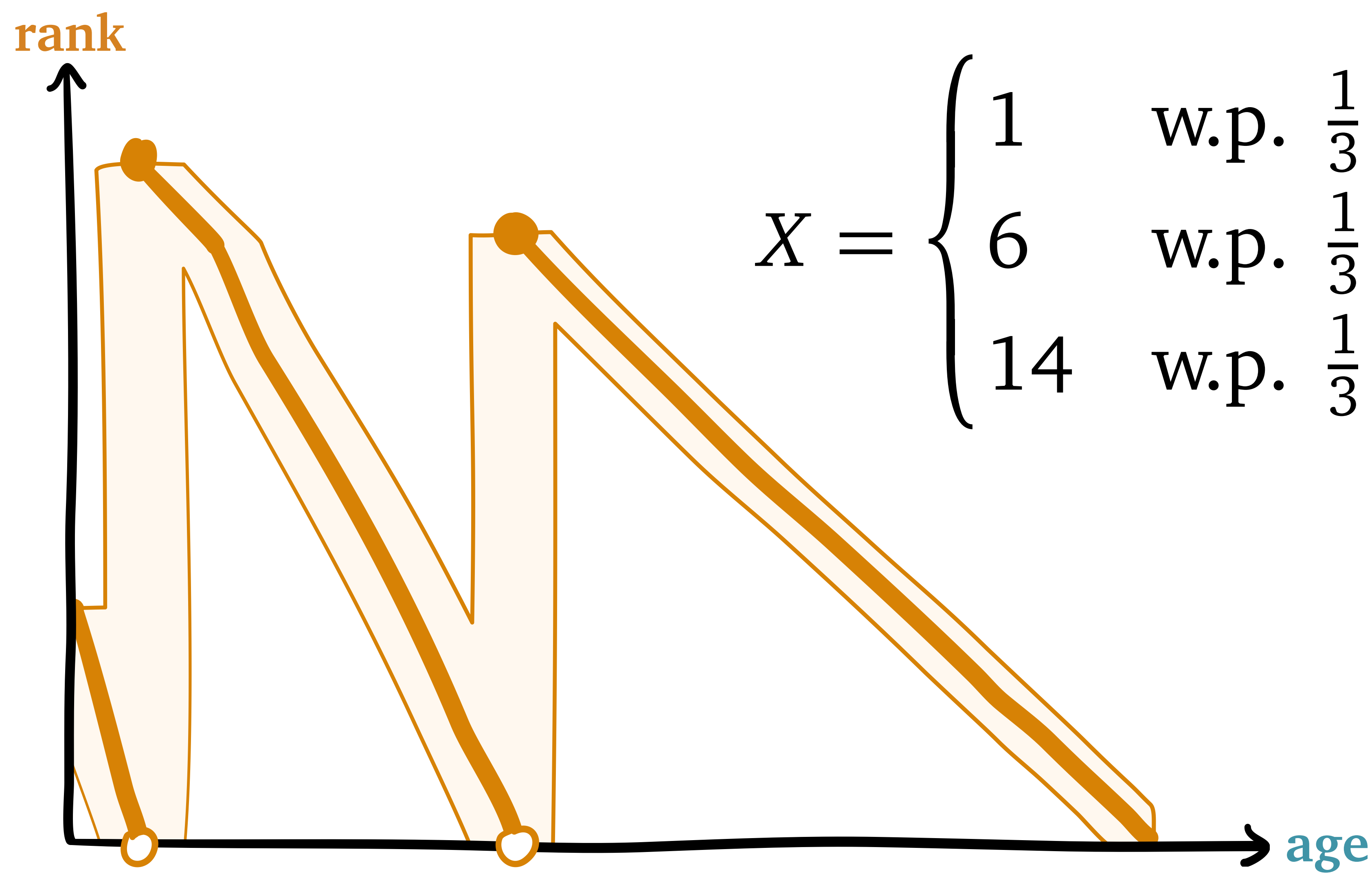
- I get *worst* possible **rank**
- Everyone else gets *best* possible **rank**

Noise could be adversarial!

**Theorem**: this *always* gives an upper bound on $\mathbf{E}[T]$

# Designing for Noisy Systems

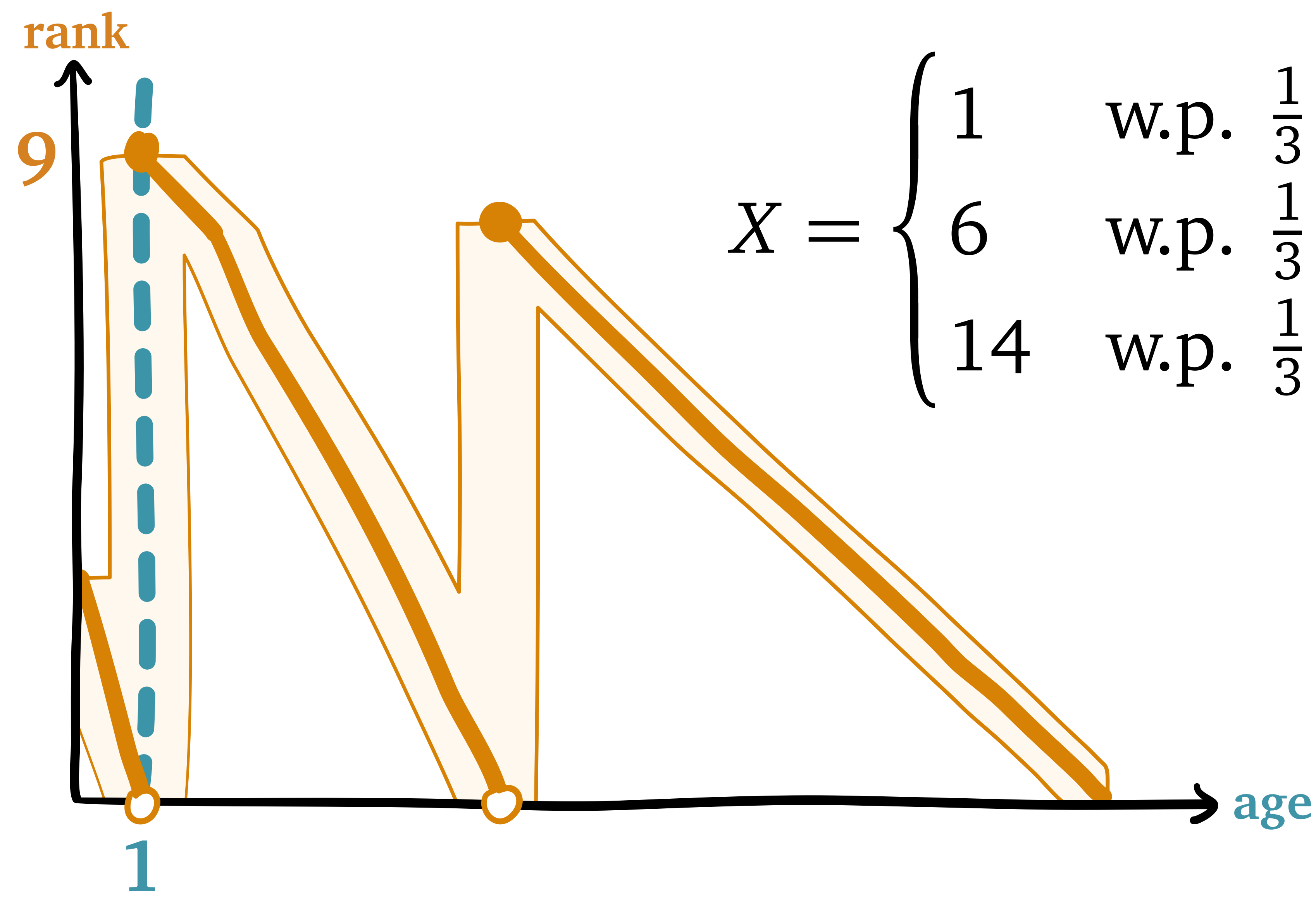
**Gittins**




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Designing for Noisy Systems

**Gittins**

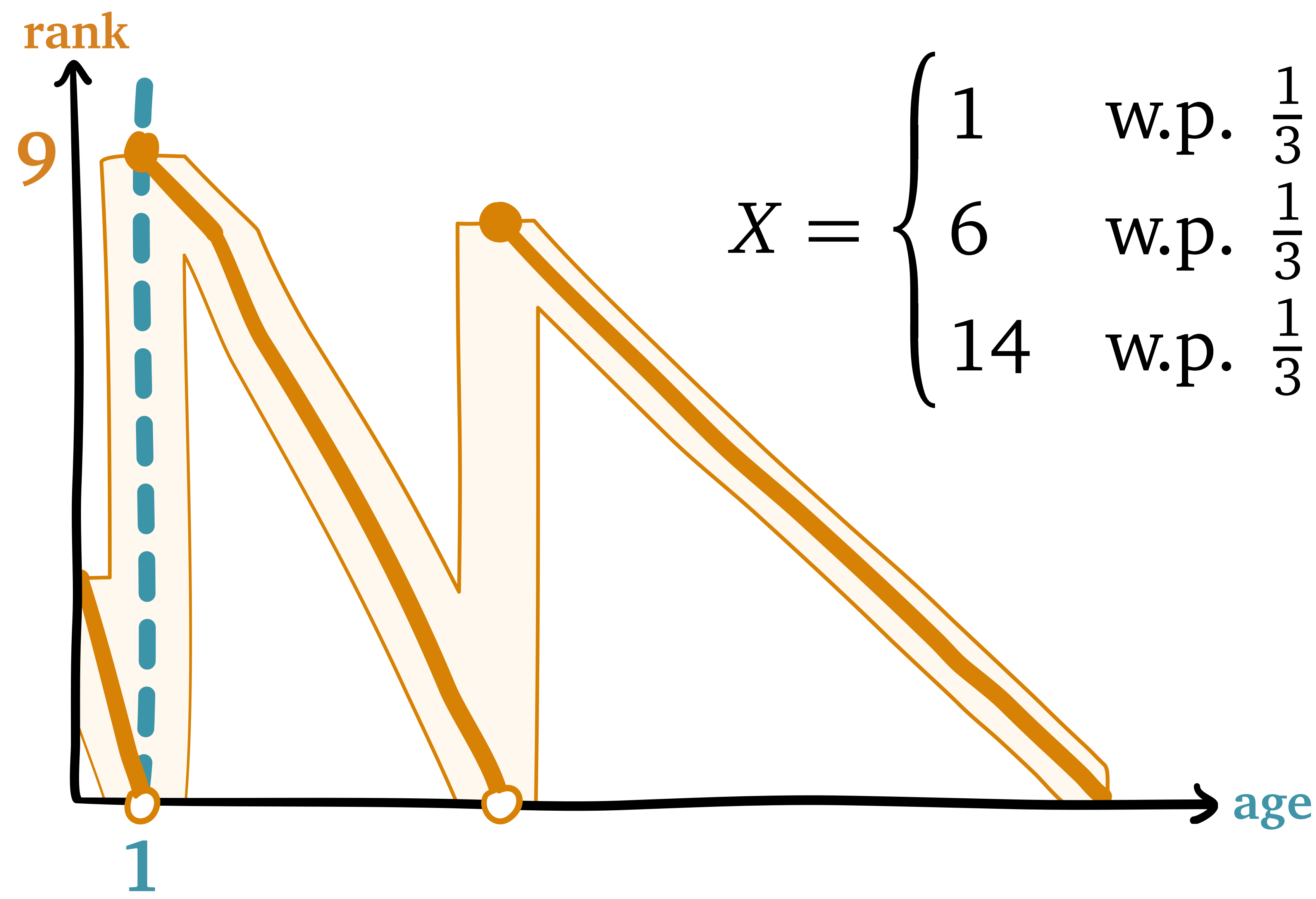



$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Designing for Noisy Systems

**Problem**:
I can jump up to rank 9 before age 1

**Gittins**



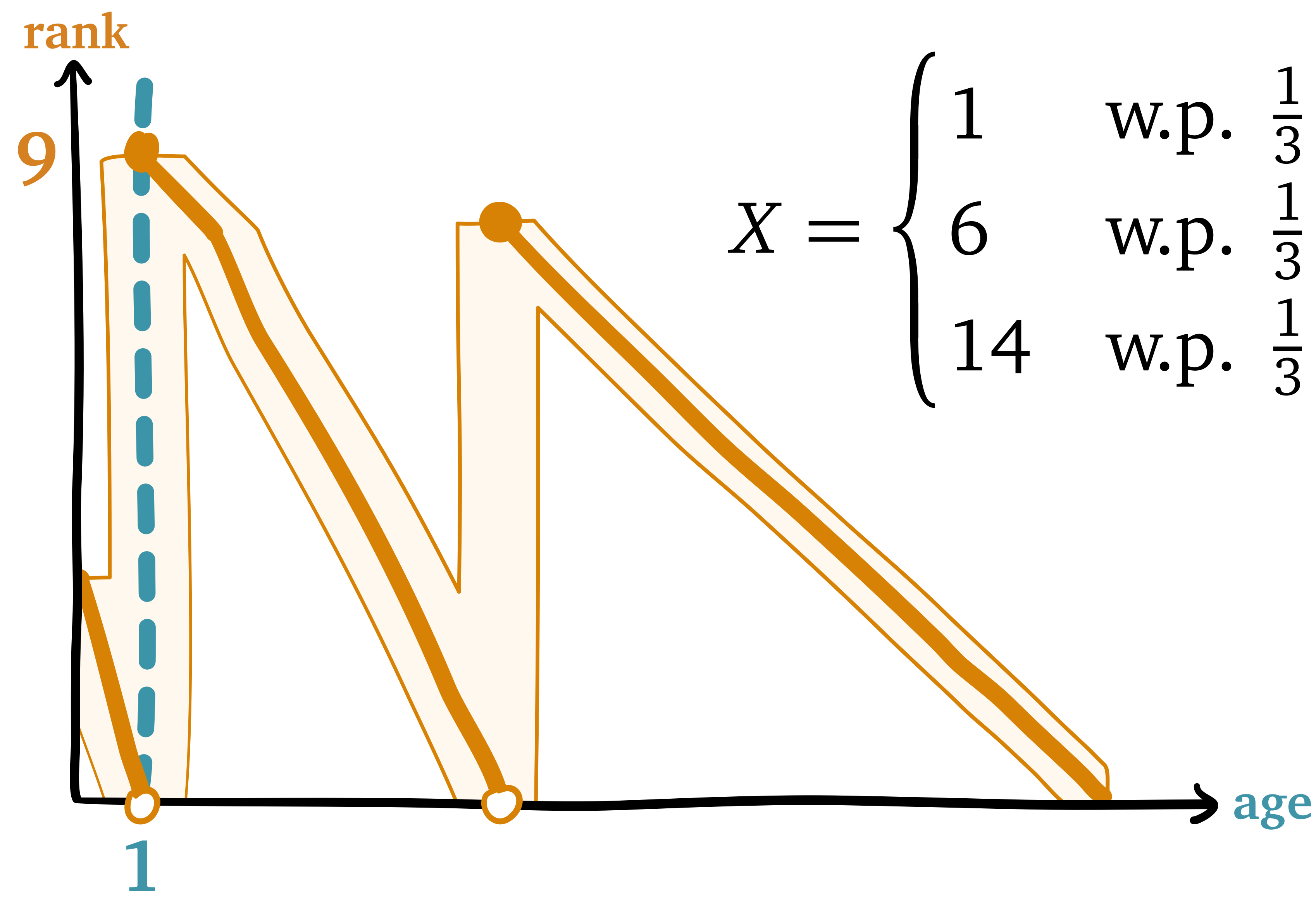

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Designing for Noisy Systems

**Gittins**

**Problem**:
I can jump up to rank 9 before age 1

**Solution**: *shift*



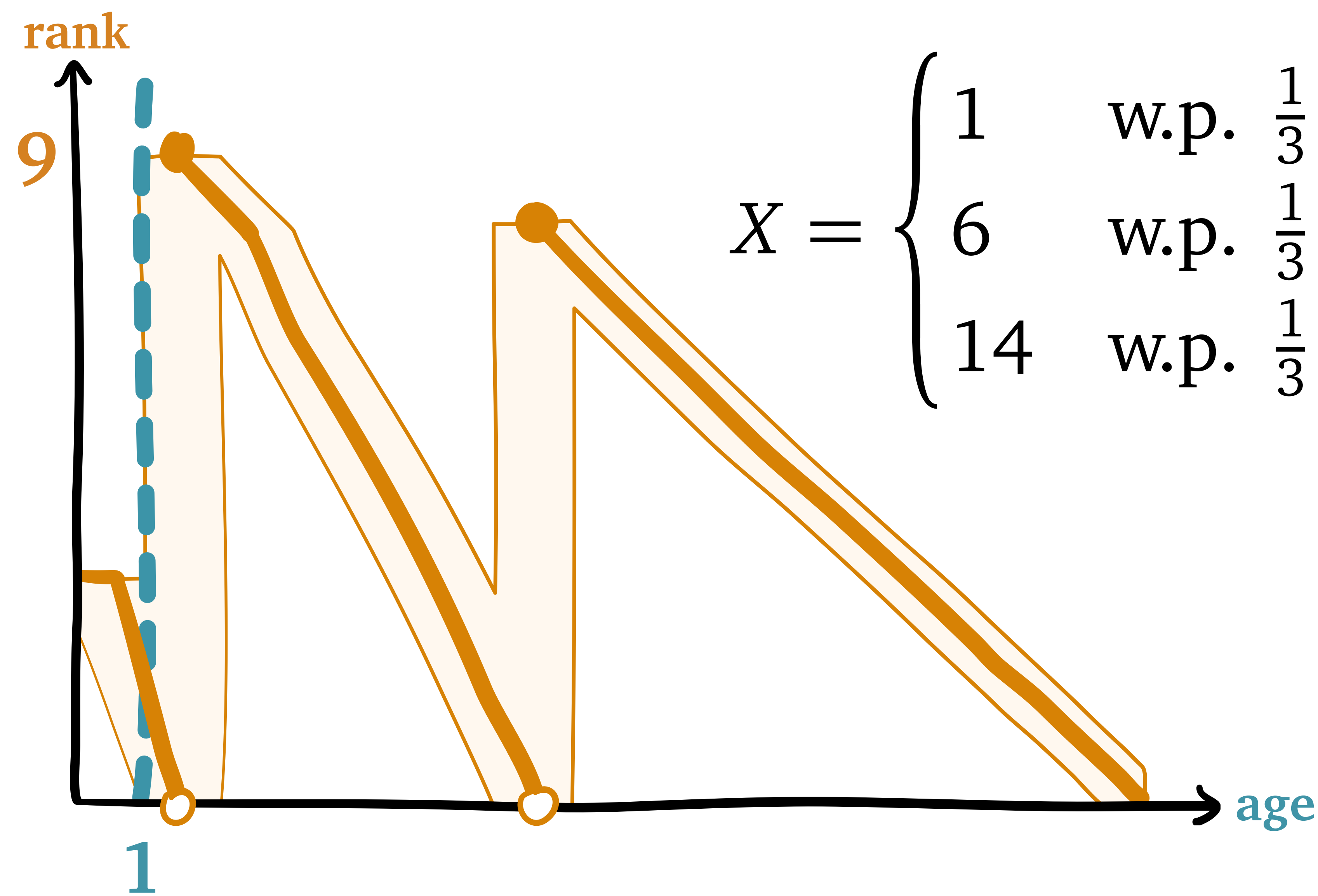

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Designing for Noisy Systems

**Problem**:
I can jump up to rank 9 before age 1

**Solution**: *shift*

**Shift Gittins**




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

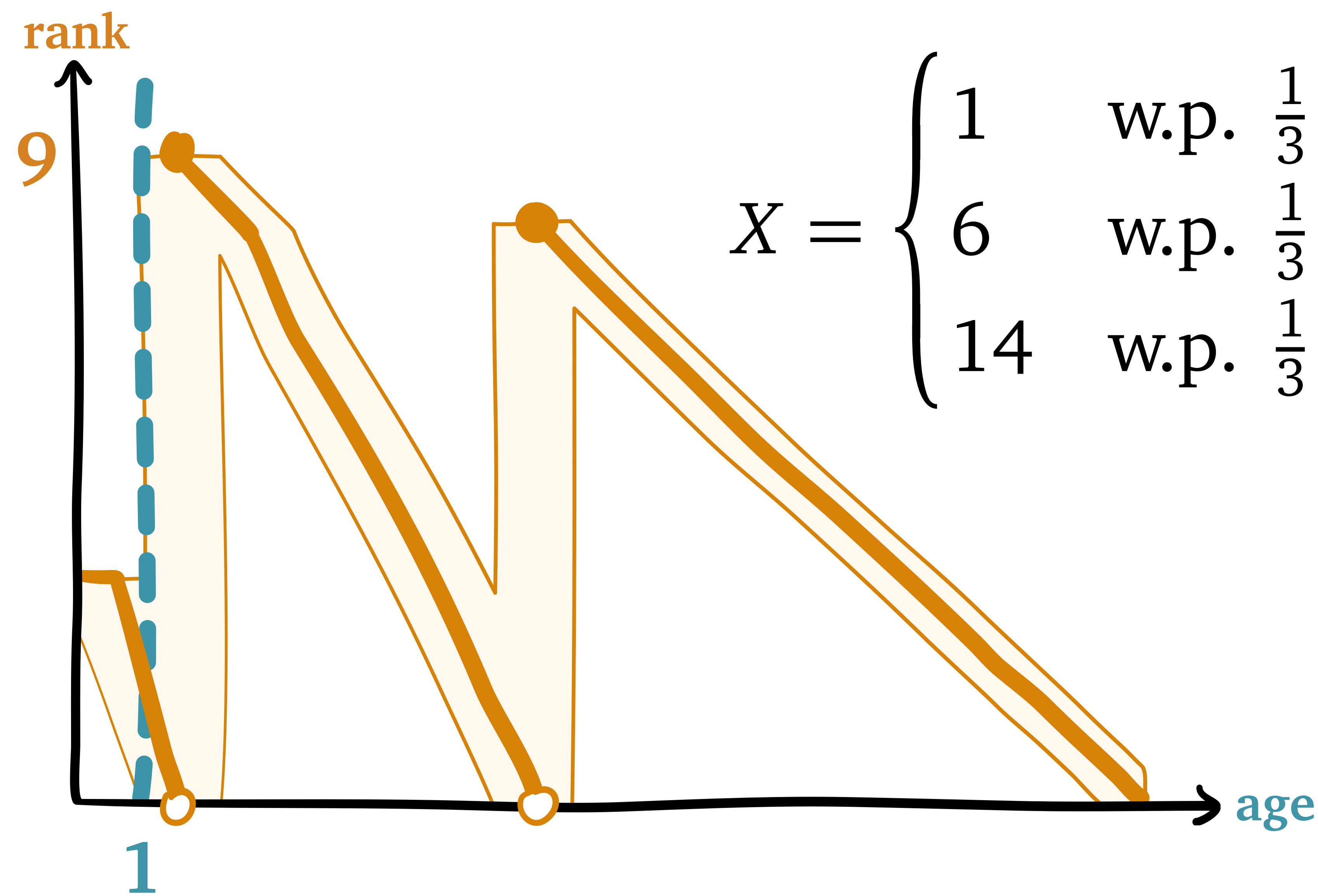
# Designing for Noisy Systems

**Problem**:
I can jump up to rank 9 before age 1

**Solution**: *shift*

**Problem**:
other jobs might not reach rank 9

**Shift Gittins**




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Designing for Noisy Systems

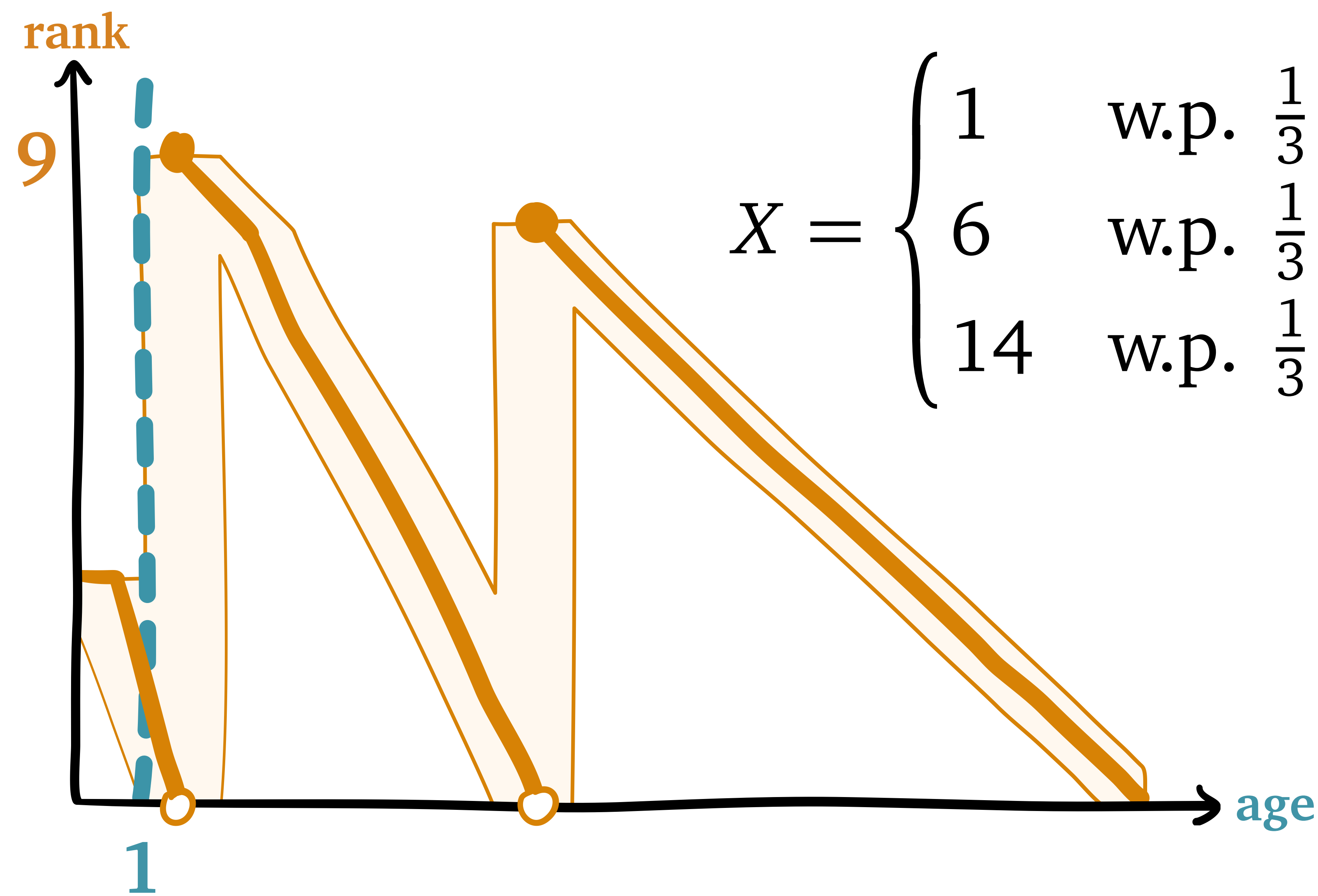
**Problem**:
I can jump up to rank 9 before age 1

**Solution**: *shift*

**Problem**:
other jobs might not reach rank 9

**Solution**: *flatten*

**Shift Gittins**




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Designing for Noisy Systems

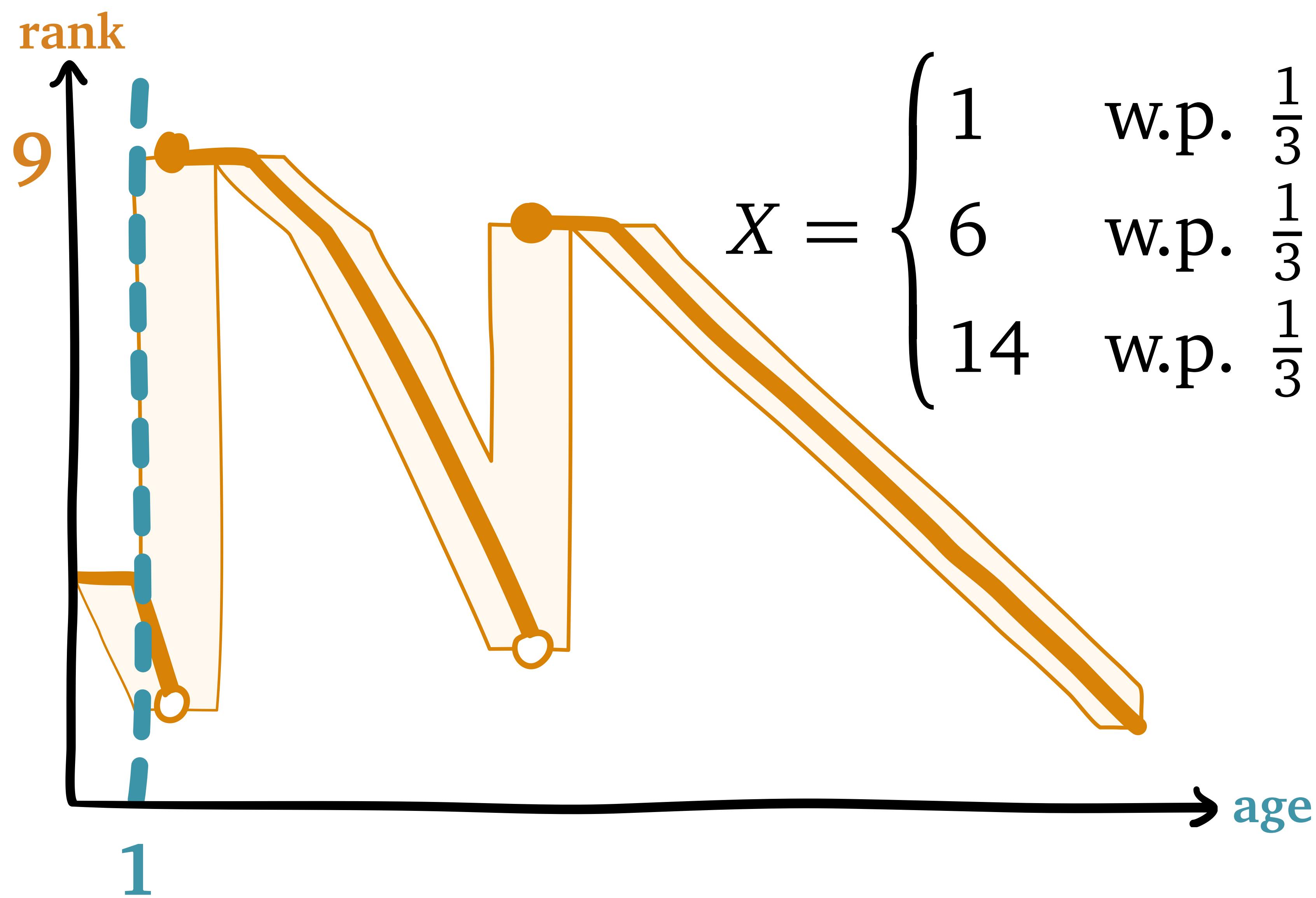
**Problem**:
I can jump up to rank 9 before age 1

**Solution**: *shift*

**Problem**:
other jobs might not reach rank 9

**Solution**: *flatten*

**Shift-Flat Gittins**




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

# Designing for Noisy Systems

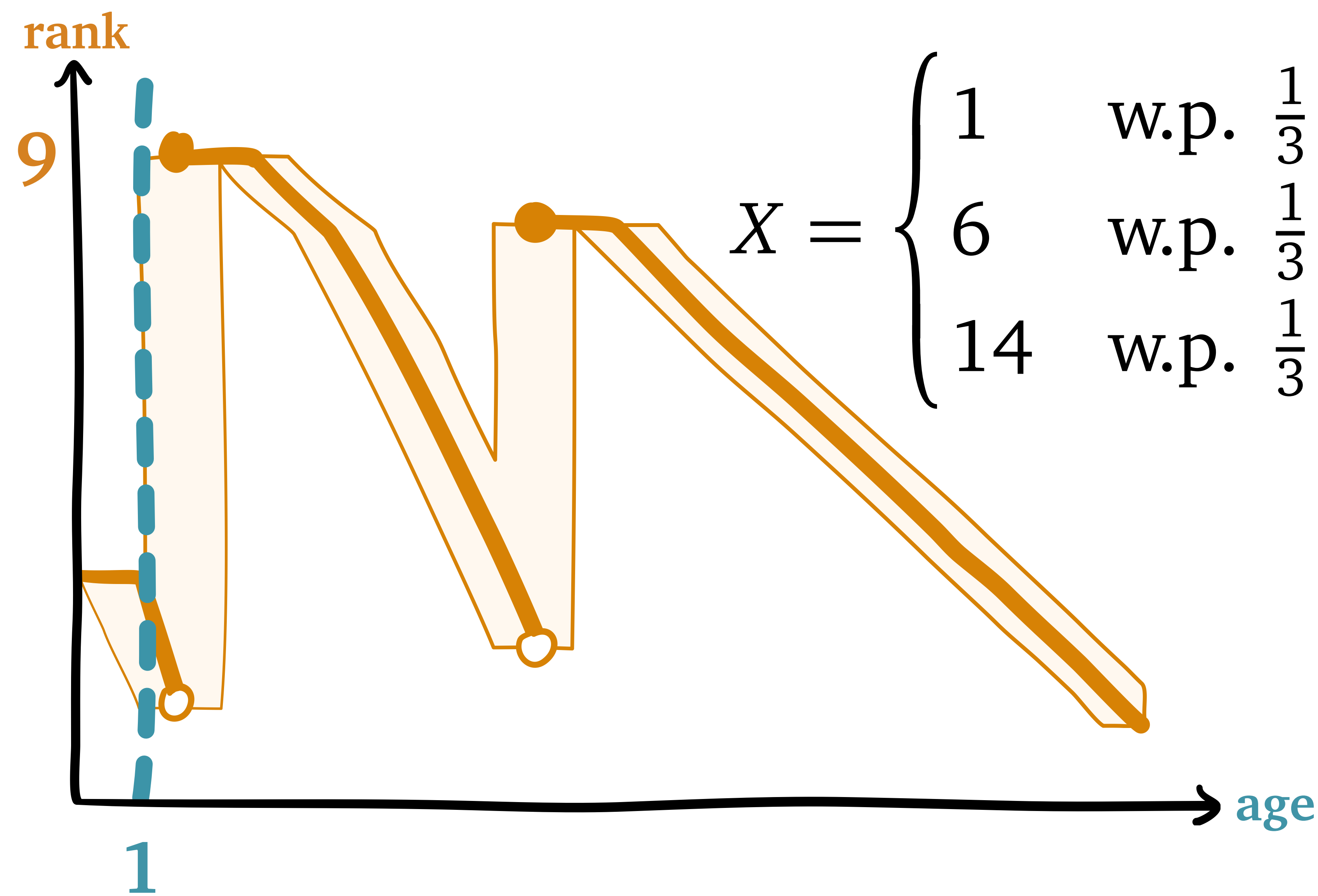
**Problem**:
I can jump up to rank 9 before age 1

**Solution**: *shift*

**Problem**:
other jobs might not reach rank 9

**Solution**: *flatten*

**Shift-Flat Gittins**



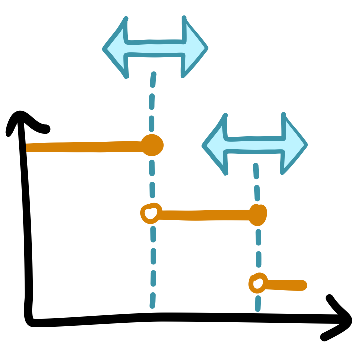

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

**Theorem**: $\mathbf{E}[T \text{ of Shift-Flat Gittins with noise } \Delta]$
$= \mathbf{E}[T \text{ of Gittins } \textit{without} \text{ noise}] + O(\Delta)$

# Outline

**Part 1**: *defining* **SOAP** policies

**Part 2**: *analyzing* **SOAP** policies



**Part 3**: *policy design* with **SOAP**

**Part 4**: *optimality proofs* with **SOAP**

# Outline

**Part 1**: *defining* **SOAP** policies

**Part 2**: *analyzing* **SOAP** policies

**Part 3**: *policy design* with **SOAP**

**Part 4**: *optimality proofs* with **SOAP**

# Part 4:

*optimality proofs* with **SOAP**

# Gittins vs. SERPT

# Gittins vs. SERPT

**Gittins**

$$r(a) = \sup_{\Delta > 0} \frac{\mathbf{E}[\min\{X - a, \Delta\} \mid X > a]}{\mathbf{P}[X - a \leq \Delta \mid X > a]}$$

# Gittins vs. SERPT

**Gittins**

$$r(a) = \sup_{\Delta > 0} \frac{\mathbf{E}[\min\{X - a, \Delta\} \mid X > a]}{\mathbf{P}[X - a \leq \Delta \mid X > a]}$$

**SERPT**

$$r(a) = \mathbf{E}[X - a \mid X > a]$$

# Gittins vs. SERPT

**Gittins**

$$r(a) = \sup_{\Delta > 0} \frac{\mathbf{E}[\min\{X - a, \Delta\} \mid X > a]}{\mathbf{P}[X - a \leq \Delta \mid X > a]}$$

Minimizes $\mathbf{E}[T]$, but can be intractable

**SERPT**

$$r(a) = \mathbf{E}[X - a \mid X > a]$$

# Gittins vs. SERPT

**Gittins**

$$r(a) = \sup_{\Delta > 0} \frac{\mathbf{E}[\min\{X - a, \Delta\} \mid X > a]}{\mathbf{P}[X - a \leq \Delta \mid X > a]}$$

⚠ Minimizes $\mathbf{E}[T]$, but can be intractable

**SERPT**

$$r(a) = \mathbf{E}[X - a \mid X > a]$$

⚠ Simple, but no $\mathbf{E}[T]$ guarantee

# Gittins vs. SERPT

**Gittins**

$$r(a) = \sup_{\Delta > 0} \frac{\mathbf{E}[\min\{X - a, \Delta\} \mid X > a]}{\mathbf{P}[X - a \leq \Delta \mid X > a]}$$

Minimizes $\mathbf{E}[T]$, but can be intractable

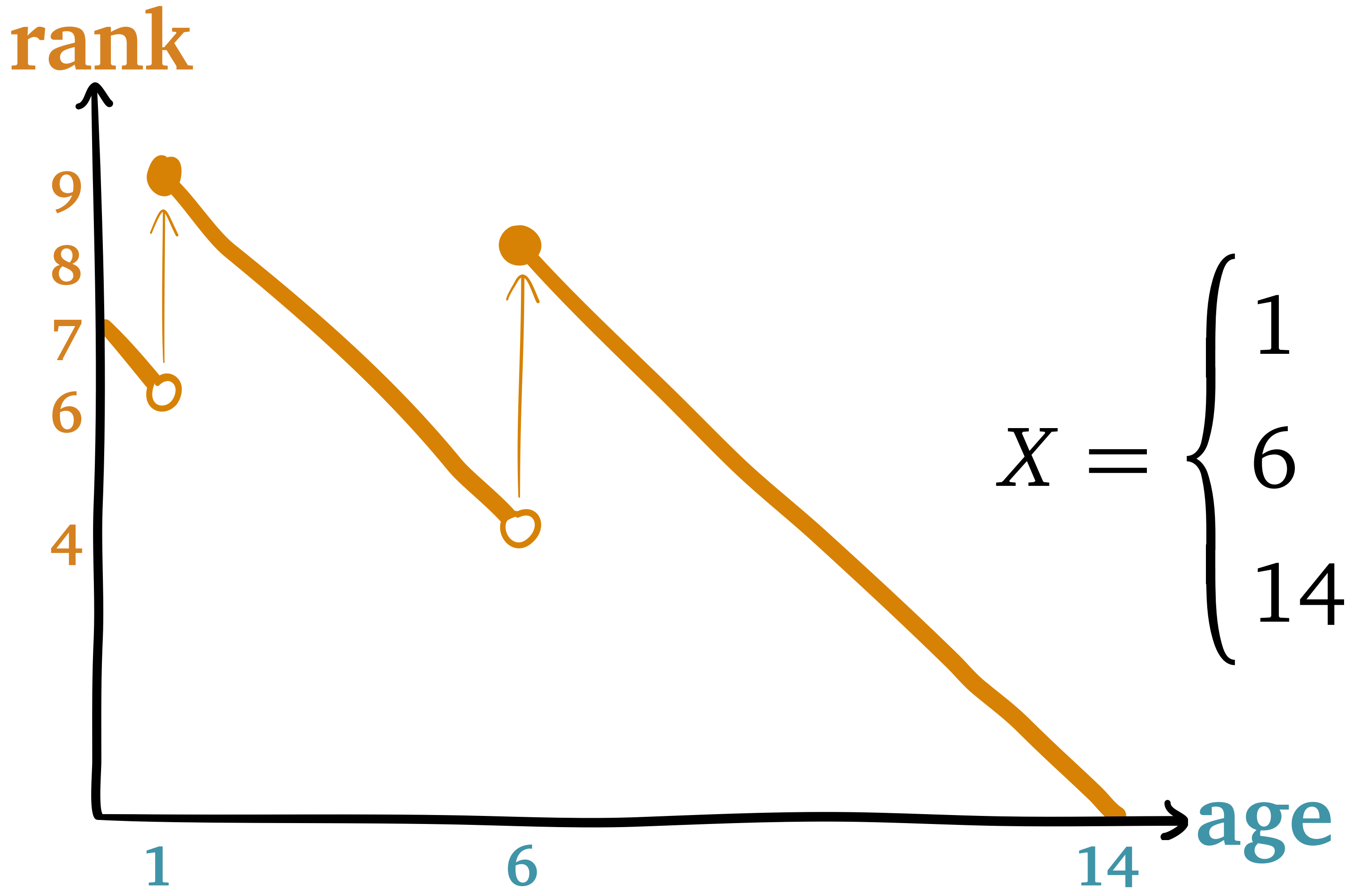
**SERPT**

$$r(a) = \mathbf{E}[X - a \mid X > a]$$

Simple, but no $\mathbf{E}[T]$ guarantee

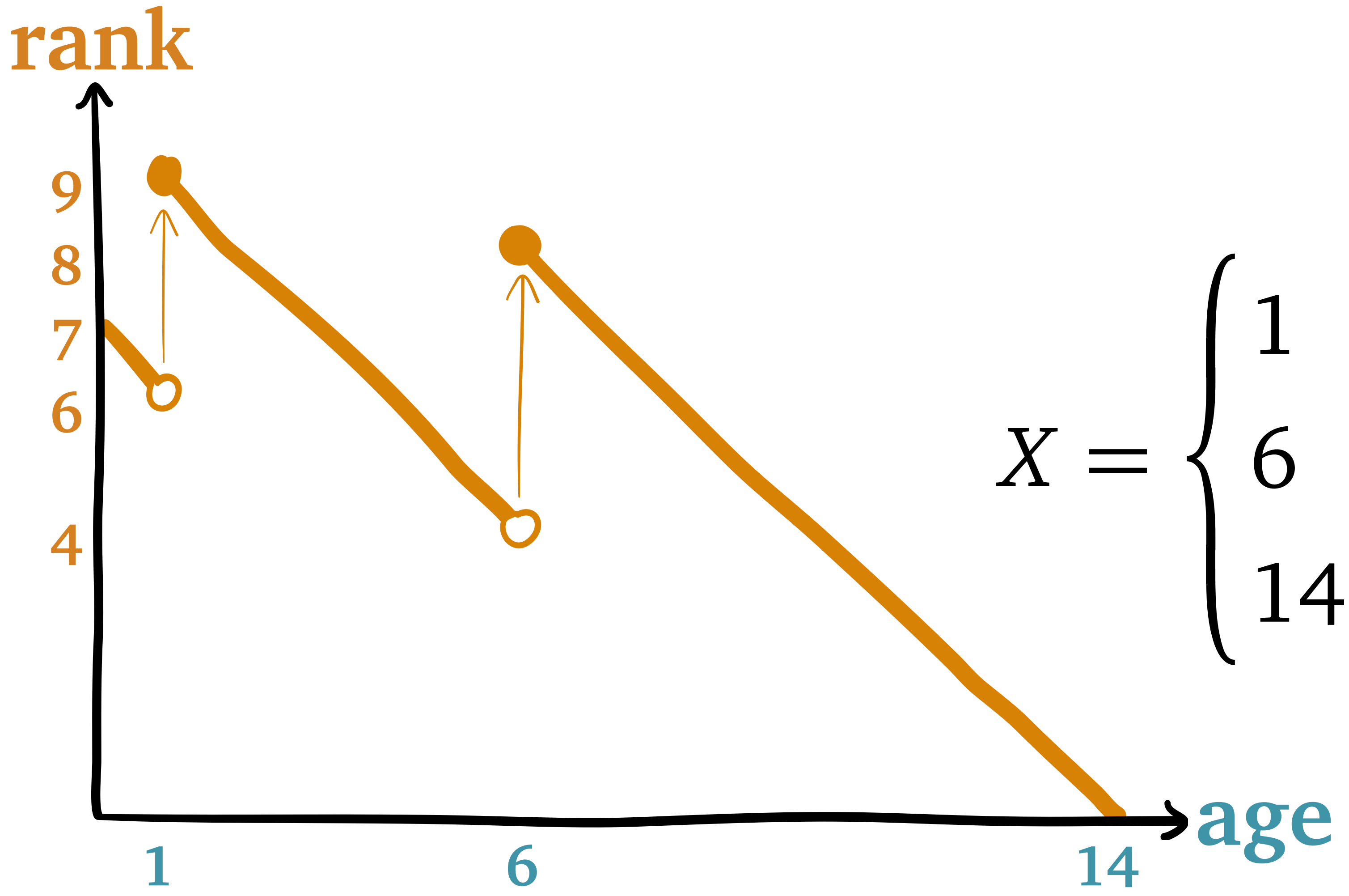
**Question**: is there a *simple* policy with *near-optimal* $\mathbf{E}[T]$?

# Monotonic SERPT




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

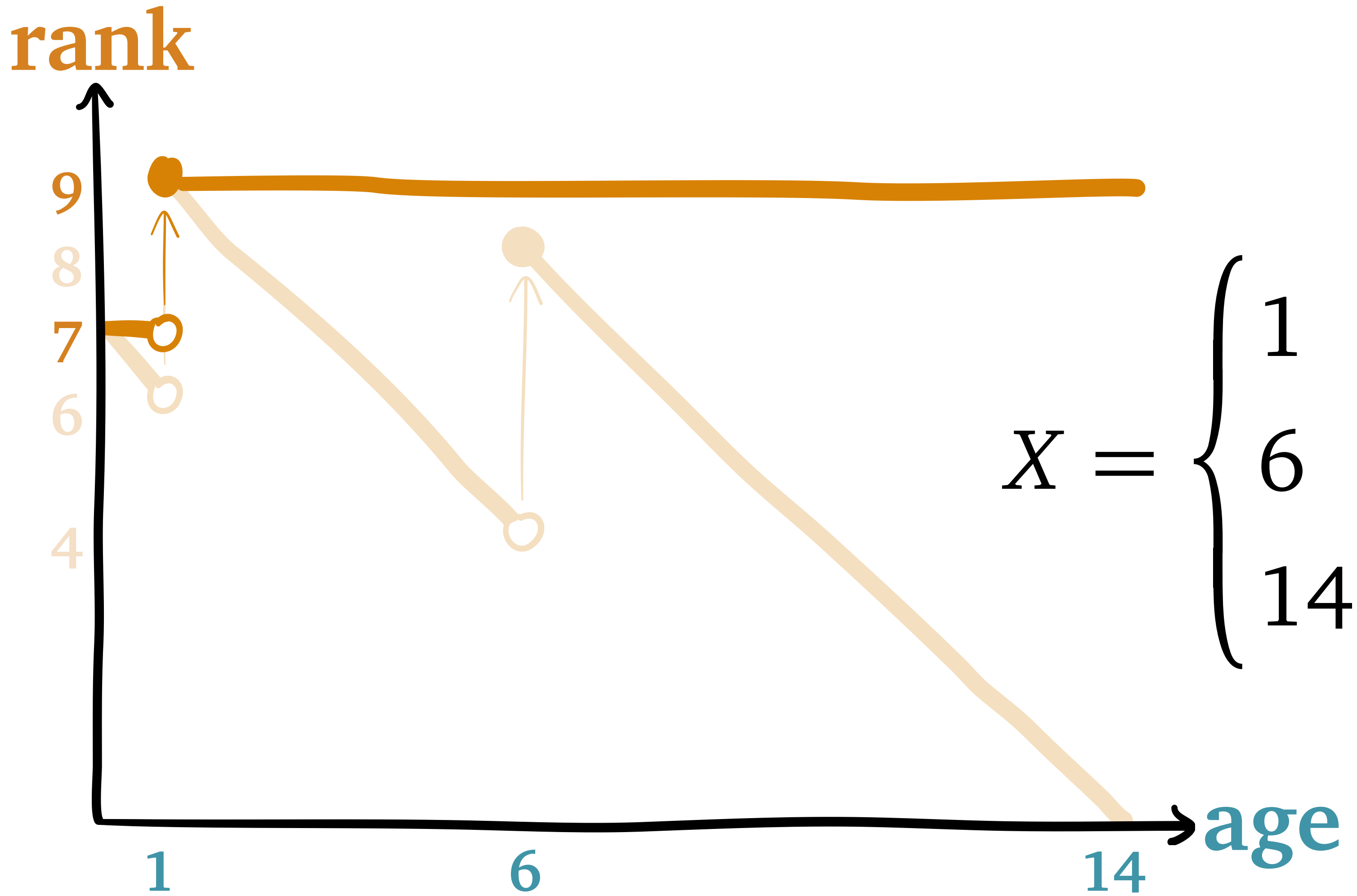
# Monotonic SERPT




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

M-SERPT is like SERPT,
but **rank** *never goes down*

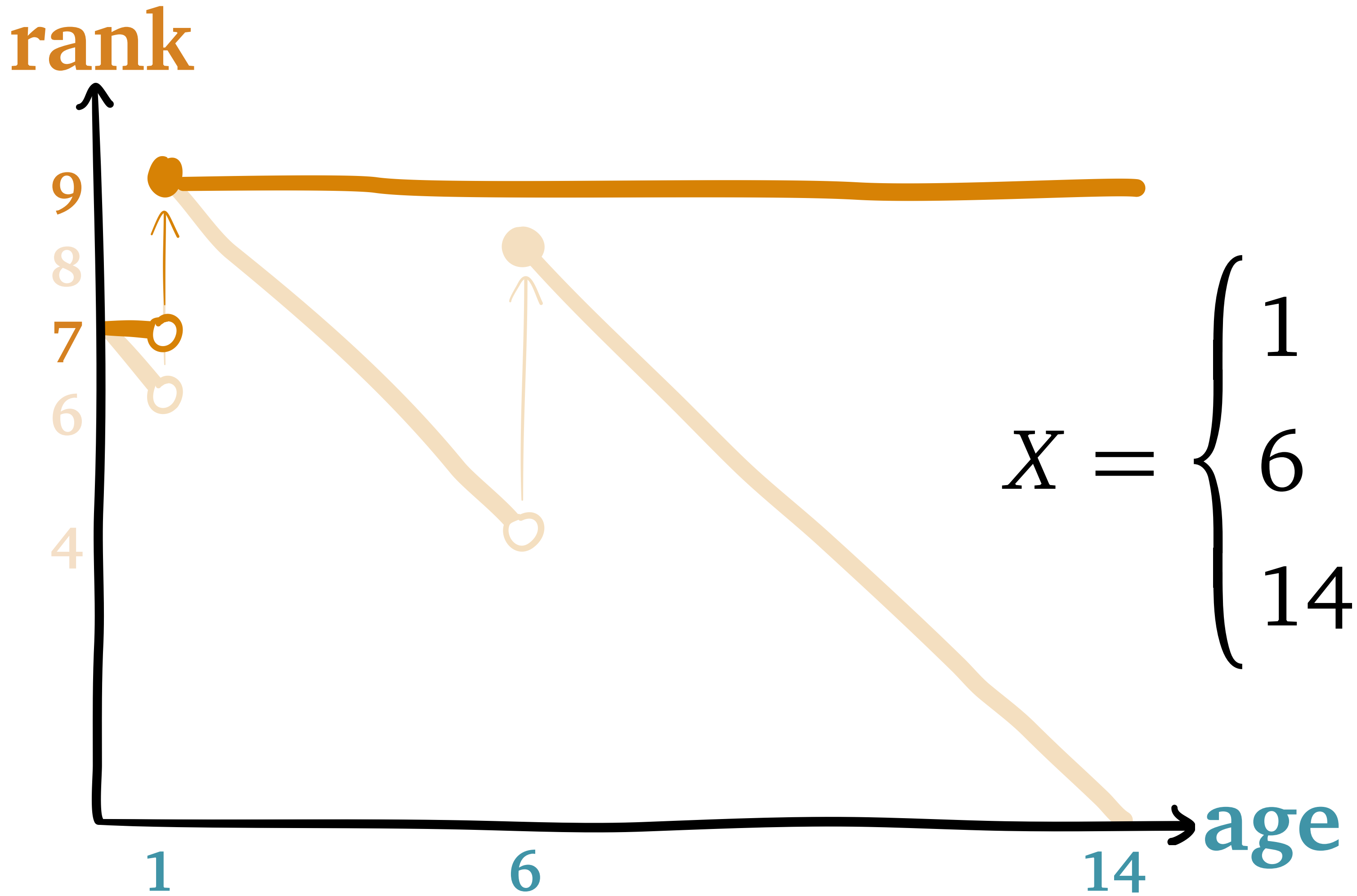
# Monotonic SERPT




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

M-SERPT is like SERPT,
but ***rank*** *never goes down*

# Monotonic SERPT




$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$

M-SERPT is like SERPT,
but ***rank*** *never goes down*

**Theorem:**

$$\frac{\mathbf{E}[T \text{ of M-SERPT}]}{\mathbf{E}[T \text{ of Gittins}]} \leq 5$$

# Outline

**Part 1**: *defining* **SOAP** policies

**Part 2**: *analyzing* **SOAP** policies

**Part 3**: *policy design* with **SOAP**

**Part 4**: *optimality proofs* with **SOAP**

# Outline

**Part 1:** *defining* **SOAP** policies

**Part 2:** *analyzing* **SOAP** policies

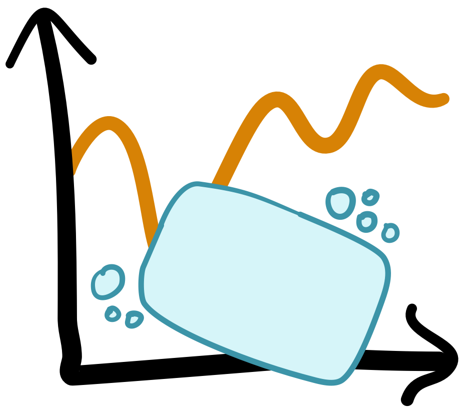
**Part 3:** *policy design* with **SOAP**

**Part 4:** *optimality proofs* with **SOAP**

# SOAP Summary

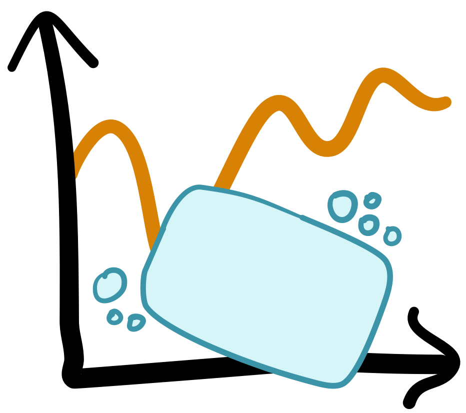
# SOAP Summary

**Idea**: schedule with **rank** functions

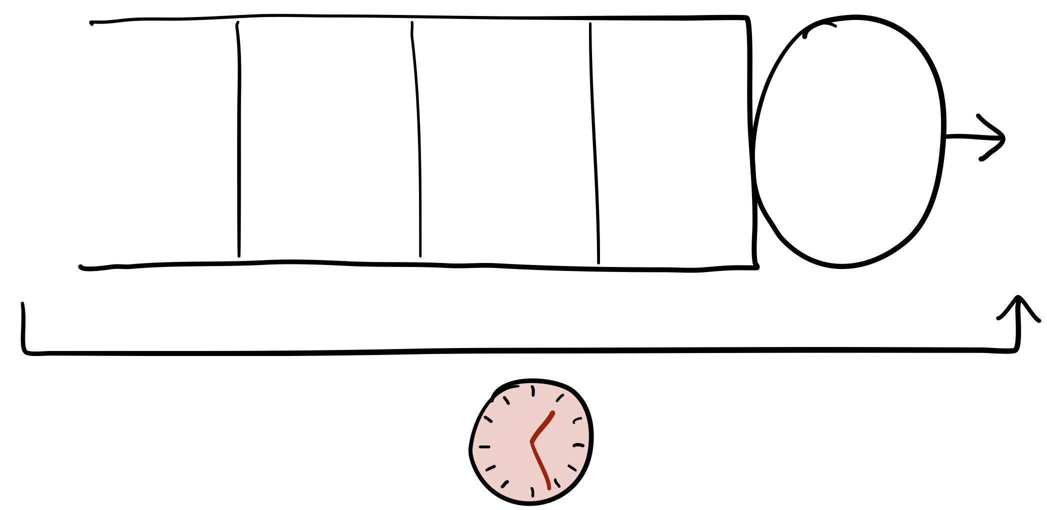


# SOAP Summary

**Idea**: schedule with **rank** functions

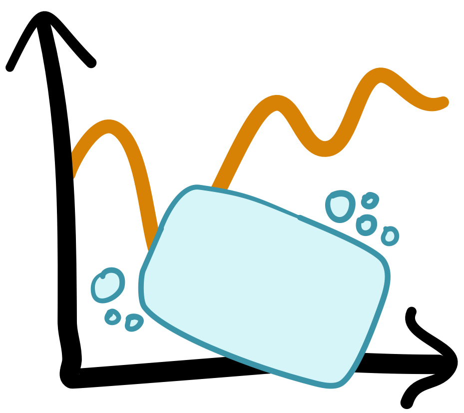


**Result:** universal response time analysis

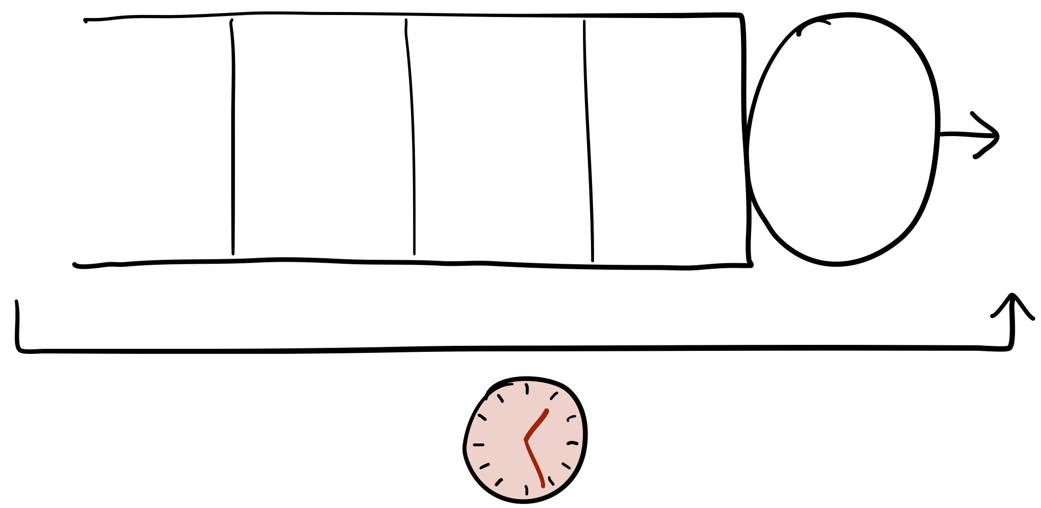


# SOAP Summary

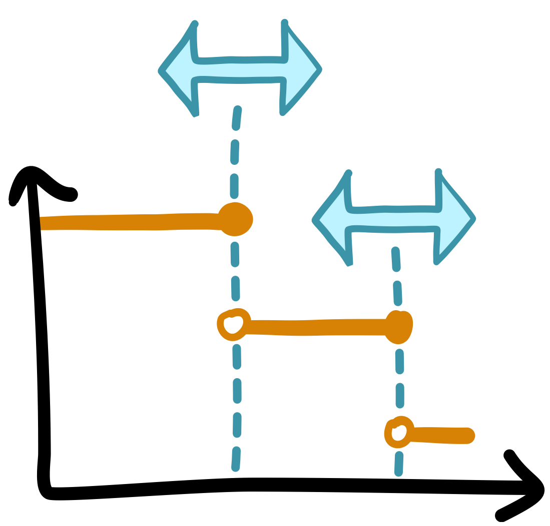
**Idea**: schedule with rank functions



**Result:** universal response time analysis



**Impact**: optimize and prove guarantees



# References: SOAP


Z. Scully, M. Harchol-Balter, and A. Scheller-Wolf (2018). **SOAP: One Clean Analysis of All Age-Based Scheduling Policies**. *Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS)*, 2(1), 16. Presented at SIGMETRICS 2018.

Z. Scully and M. Harchol-Balter (2018). **SOAP Bubbles: Robust Scheduling Under Adversarial Noise**. In *56th Annual Allerton Conference on Communication, Control, and Computing* (pp. 144–154). IEEE.

Z. Scully, M. Harchol-Balter, and A. Scheller-Wolf (2019). **Simple Near-Optimal Scheduling for the M/G/1**. *ACM SIGMETRICS Performance Evaluation Review*, to appear. Presenting at MAMA 2019 this Friday!

# References: Analyzing $\mathbf{E}[T]$


L. Kleinrock and R. R. Muntz (1972). **Processor sharing queueing models of mixed scheduling disciplines for time shared system**. *Journal of the ACM (JACM)*, 19(3), 464–482.

S. W. Furhmann and R. B. Cooper (1985). **Stochastic Decompositions in the M/G/1 Queue with Generalized Vacations**. *Operations Research*, 33(5), 1117–1129.

M. Harchol-Balter (2013). ***Performance Modeling and Design of Computer Systems: Queueing Theory in Action***. Cambridge University Press.

# References: Possible Applications

M. Harchol-Balter, Schroeder, B., Bansal, N., and Agrawal, M. (2003). **Size-based scheduling to improve web performance**. *ACM Transactions on Computer Systems (TOCS)*, 21(2), 207–233.

B. Montazeri, Y. Li, M. Alizadeh, and J. Ousterhout (2018). **Homa: A receiver-driven low-latency transport protocol using network priorities**. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (pp. 221–235). ACM.

S. Emadi, R. Ibrahim, and S. Kesavan (2019). **Can "very noisy" information go a long way? An exploratory analysis of personalized scheduling in service systems**. Working paper.

M. Mitzenmacher (2019). **Scheduling with Predictions and the Price of Misprediction**. Preprint, *arXiv:1902.00732*.

B. Kamphorst (2018). ***Heavy-traffic behaviour of scheduling policies in queues*** (Doctoral dissertation, Technische Universiteit Eindhoven).

Y. Chen and J. Dong (2019). **The Power of Two in Queue Scheduling**. Working paper.