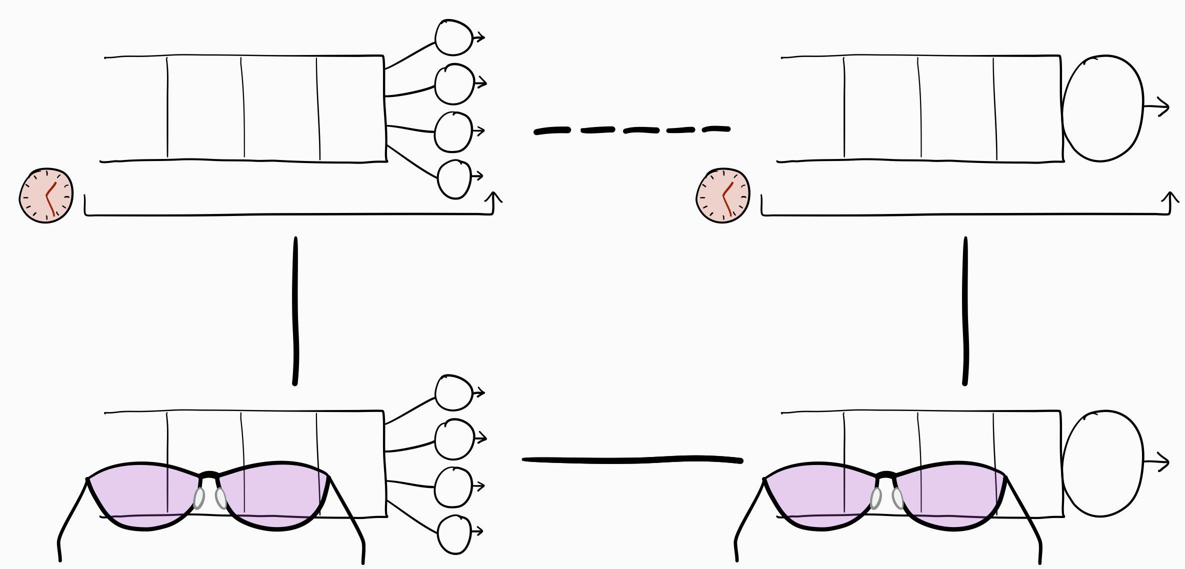
The Gittins Policy is Nearly Optimal in the M/G/k

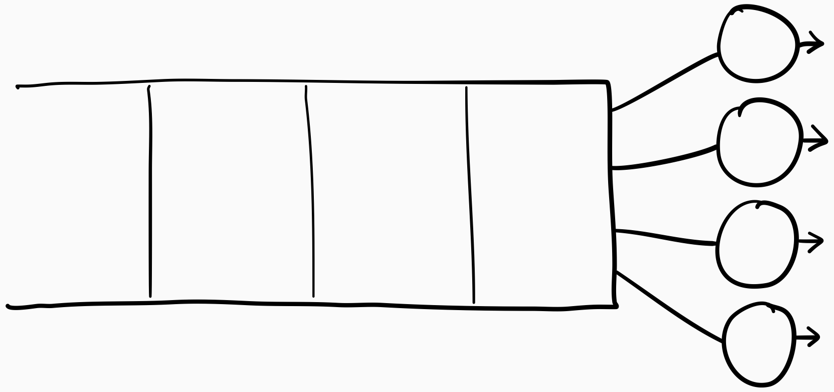
under Extremely General Conditions

Ziv Scully
Isaac Grosof
Mor Harchol-Balter

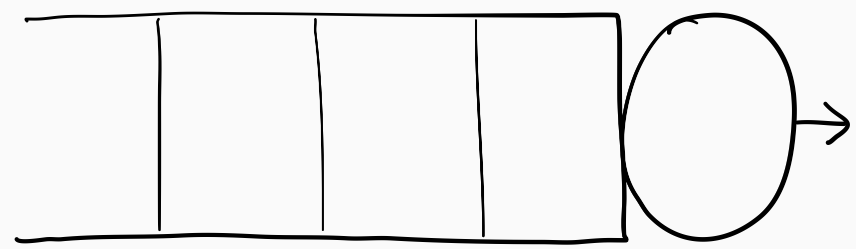
Carnegie Mellon University



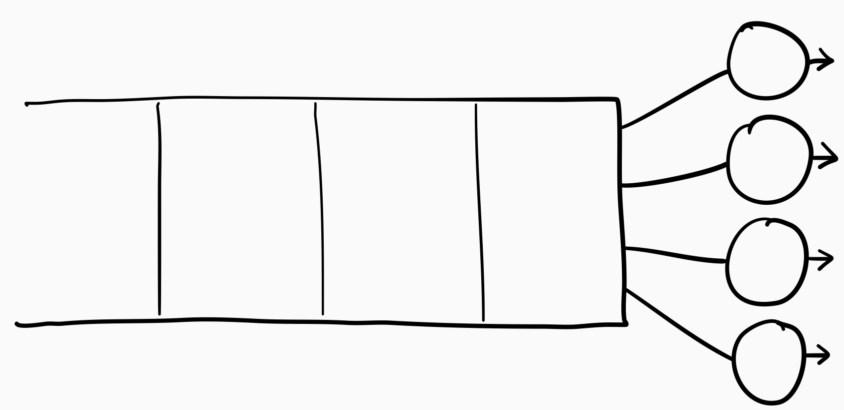
This talk: near-optimal *multiserver* scheduling



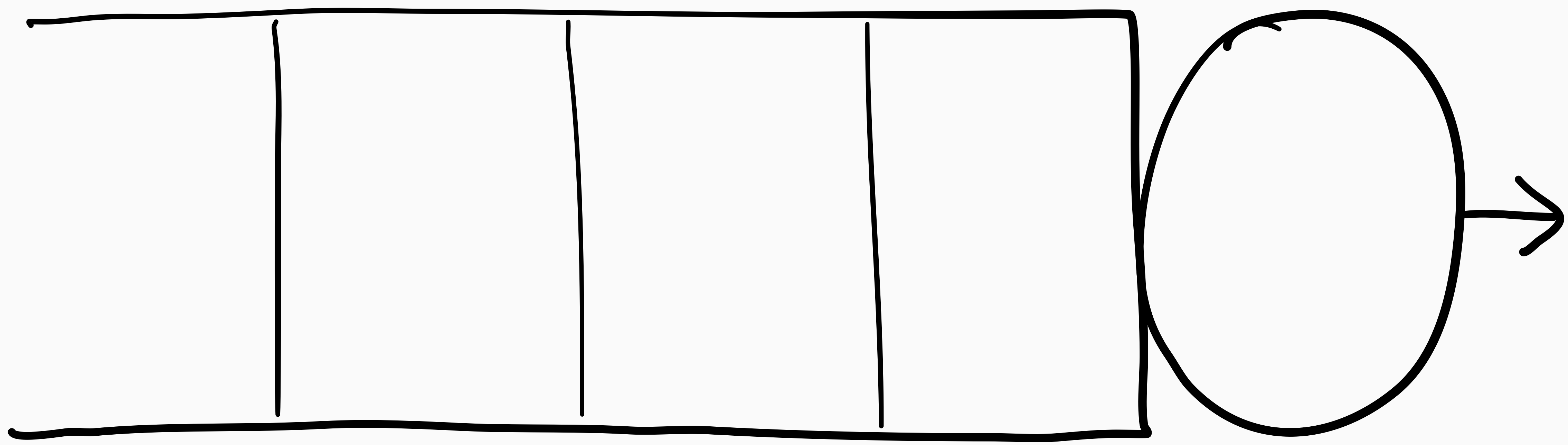
First: background on *single-server* scheduling



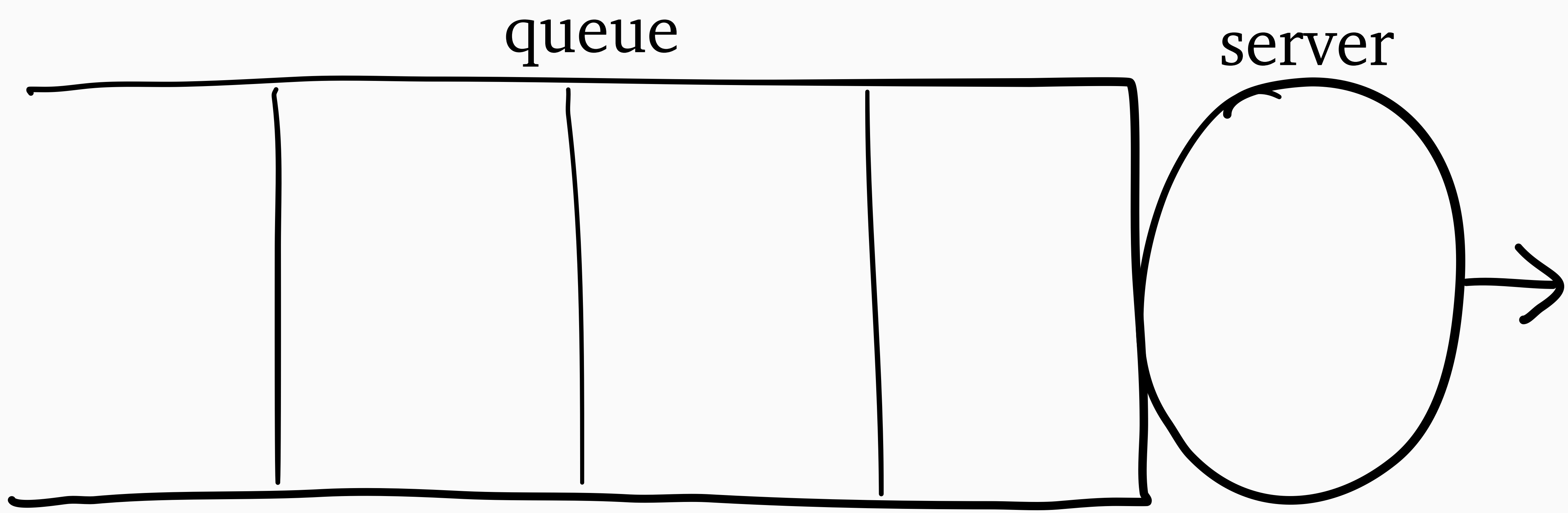
This talk: near-optimal *multiserver* scheduling



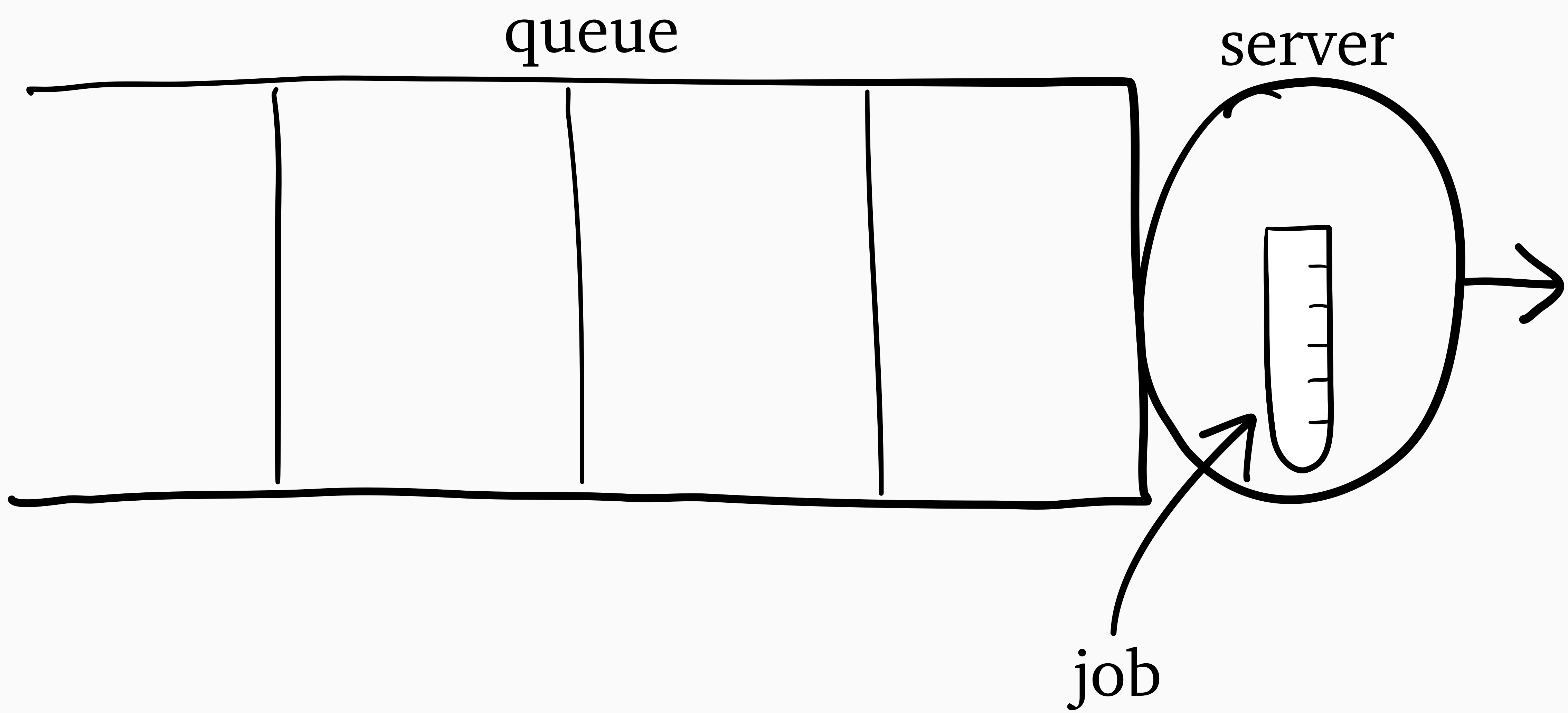
M/G/1 Queue



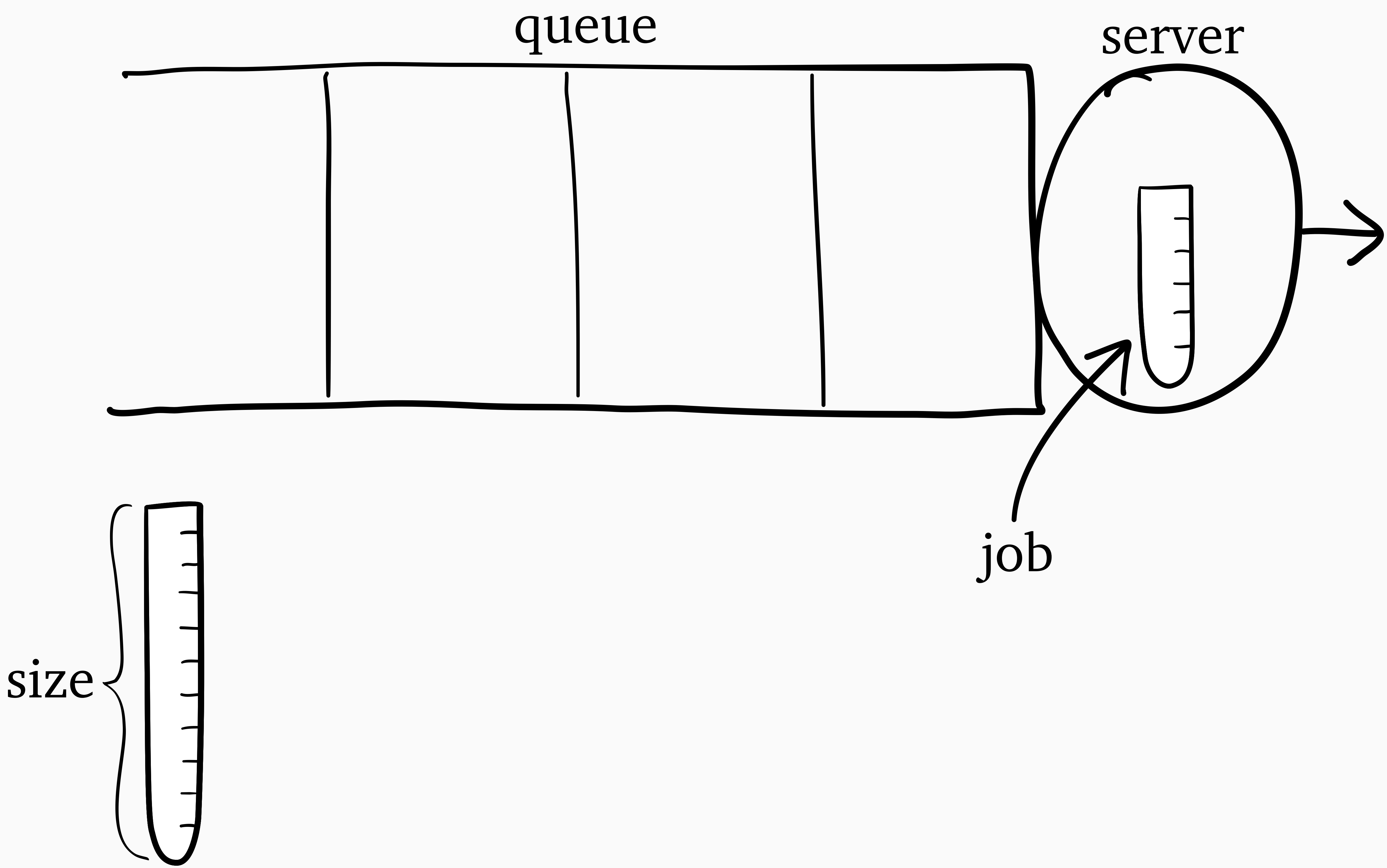
M/G/1 Queue



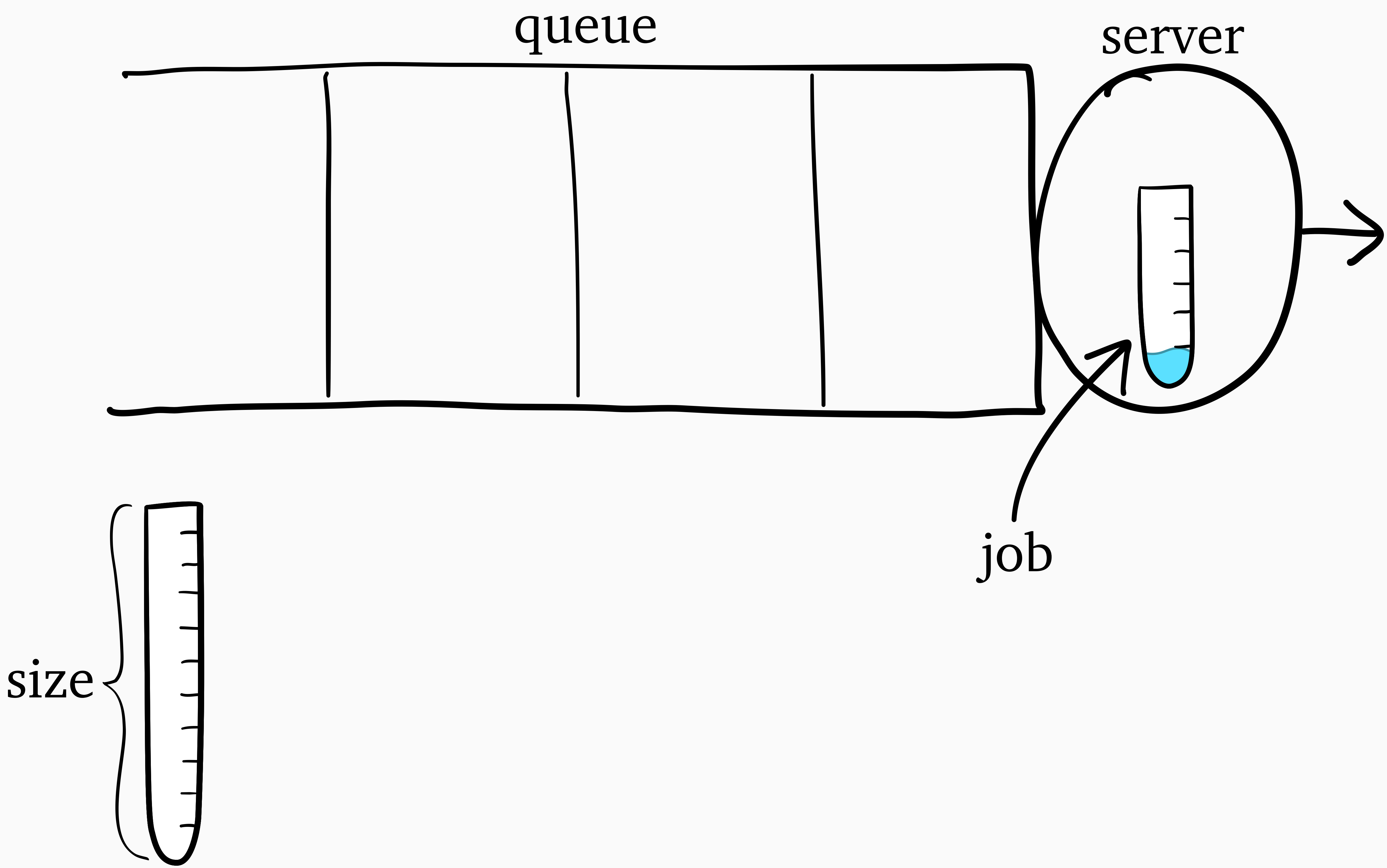
M/G/1 Queue



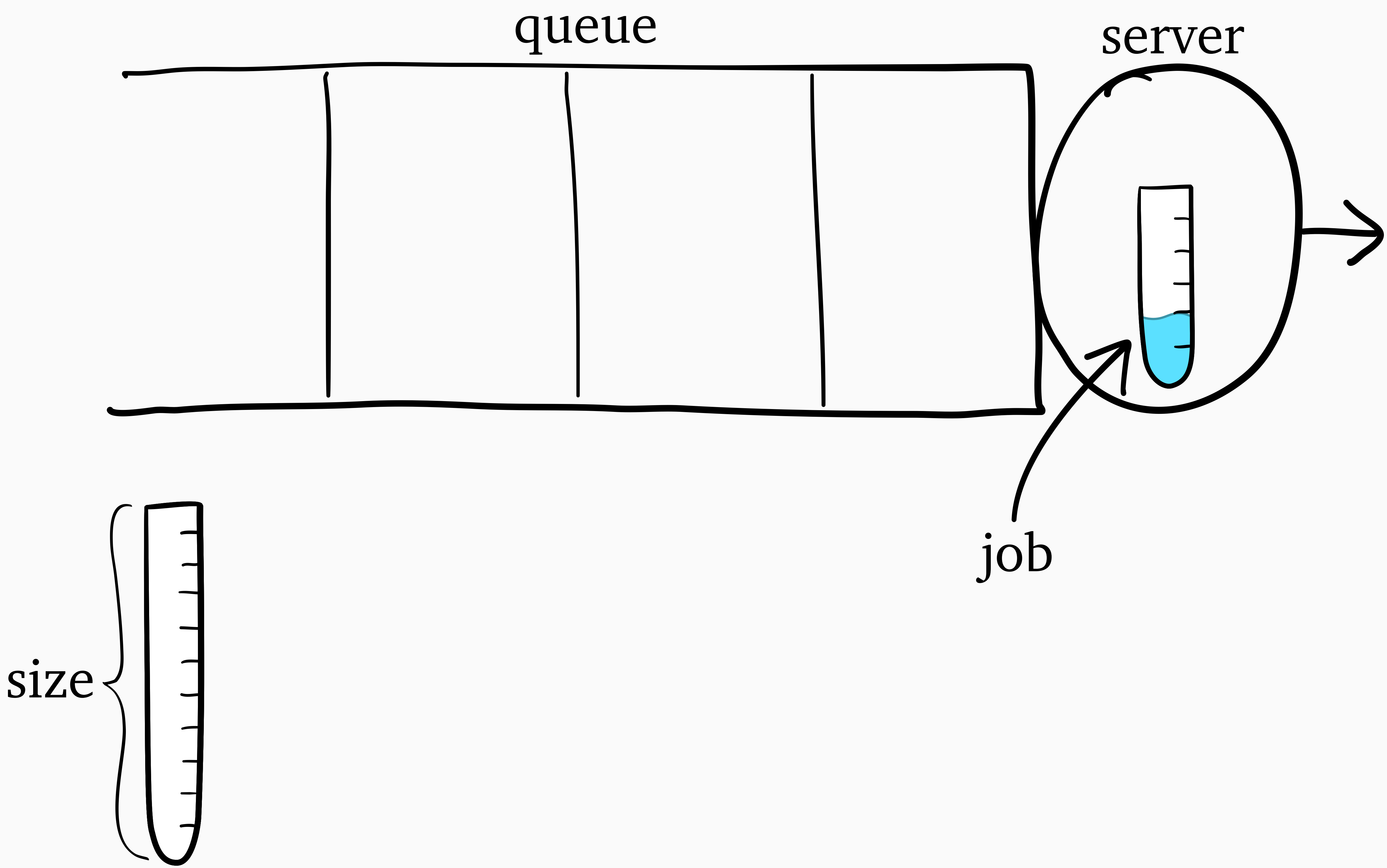
M/G/1 Queue



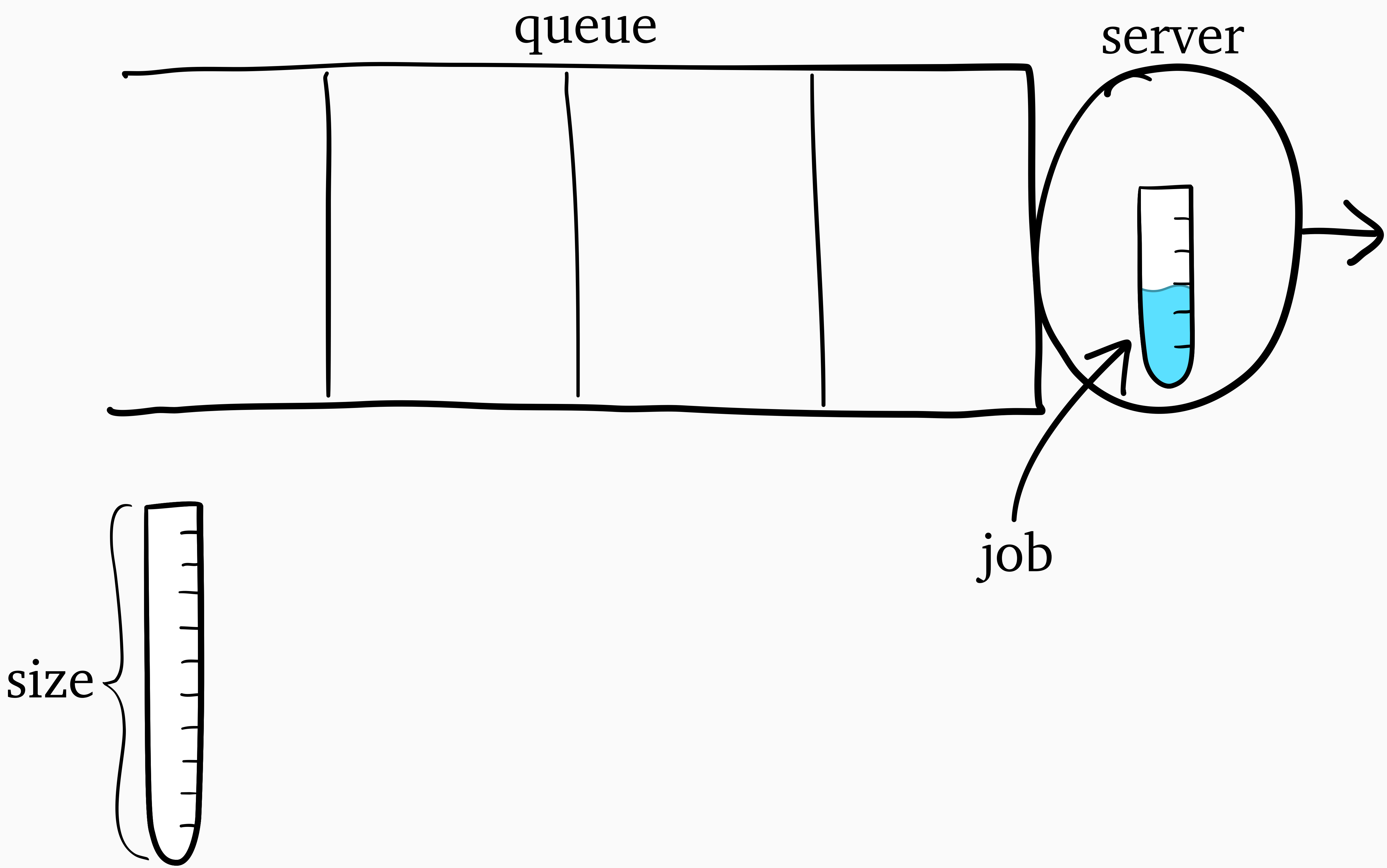
M/G/1 Queue



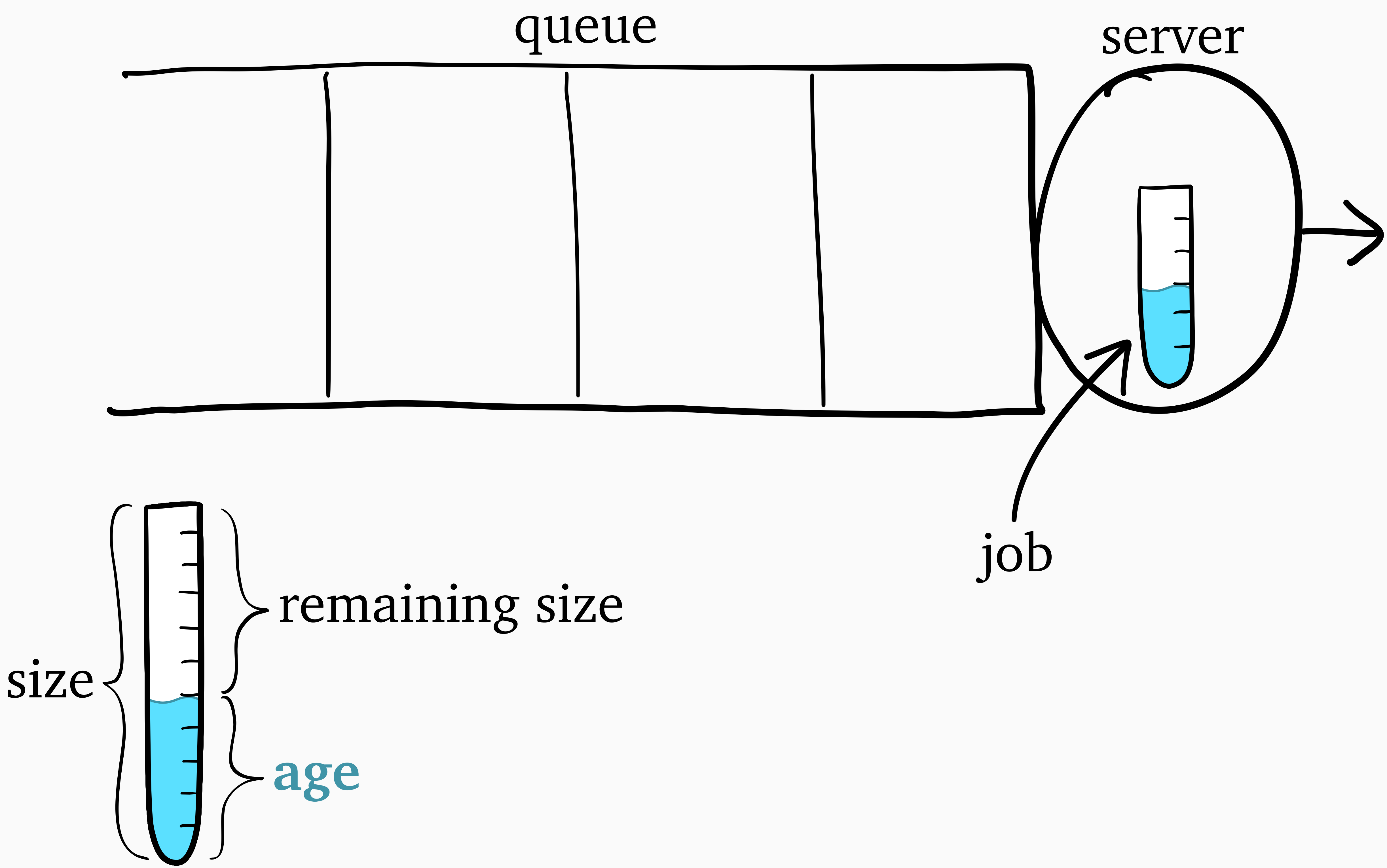
M/G/1 Queue



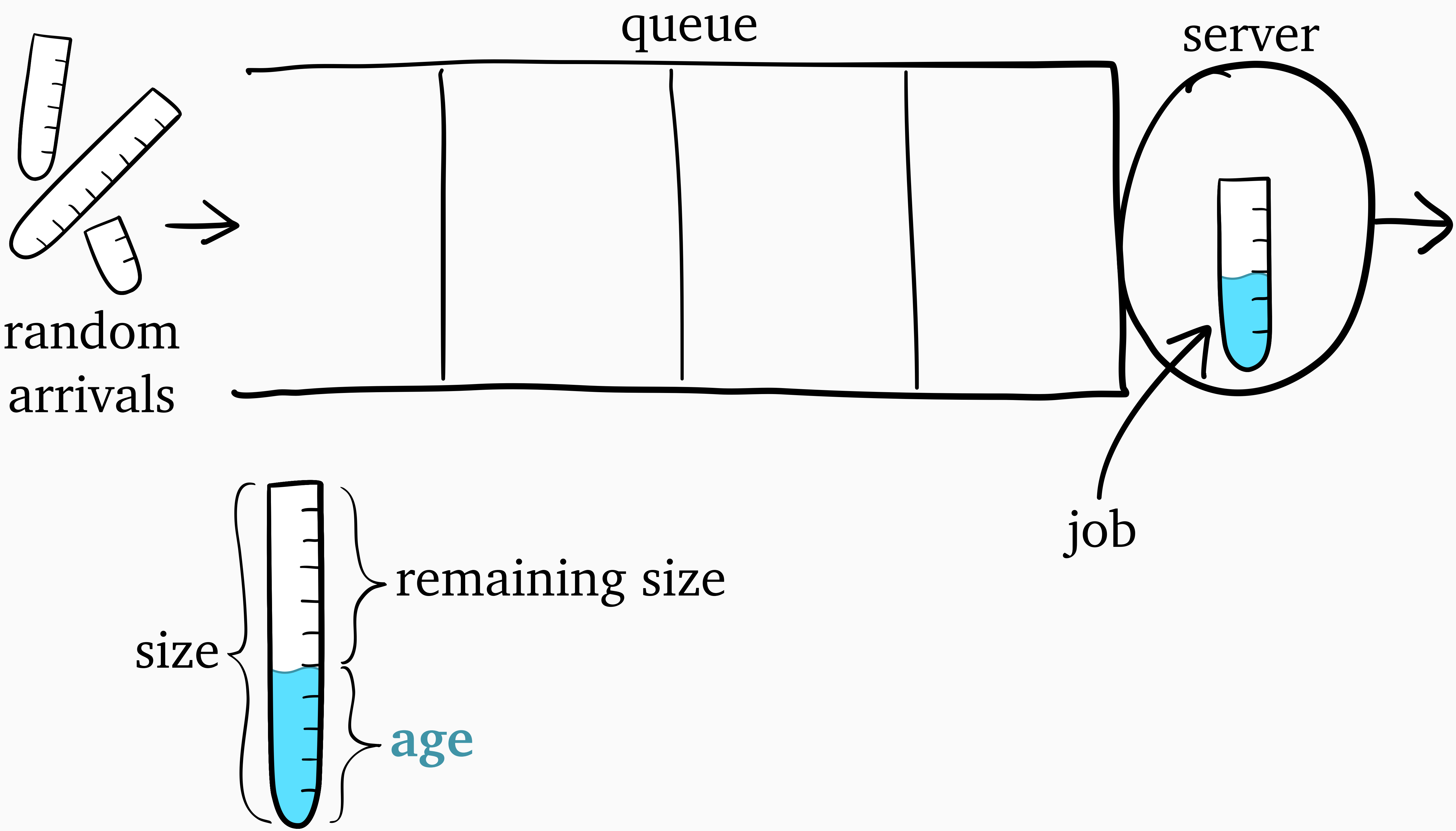
M/G/1 Queue



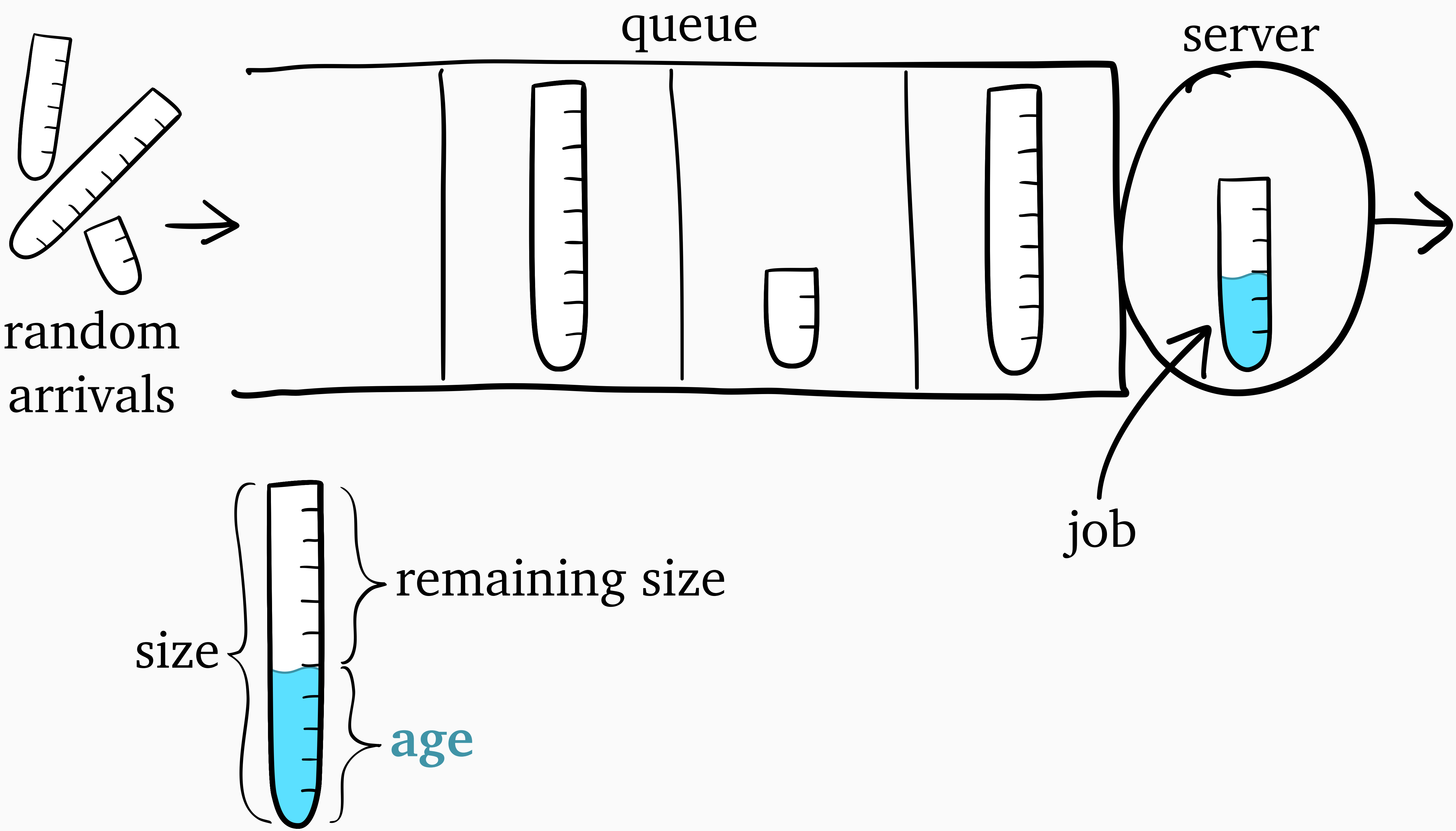
M/G/1 Queue



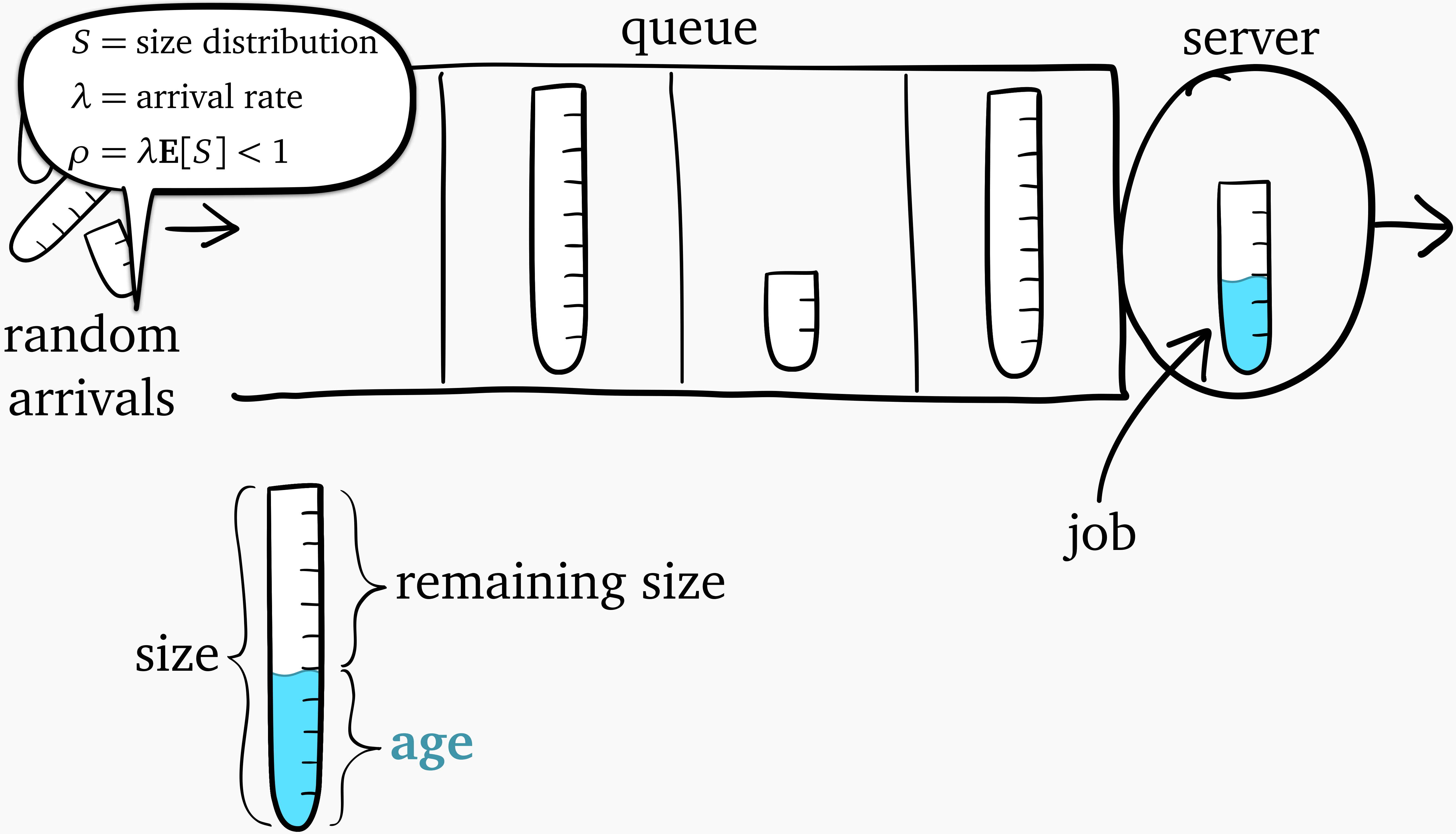
M/G/1 Queue



M/G/1 Queue

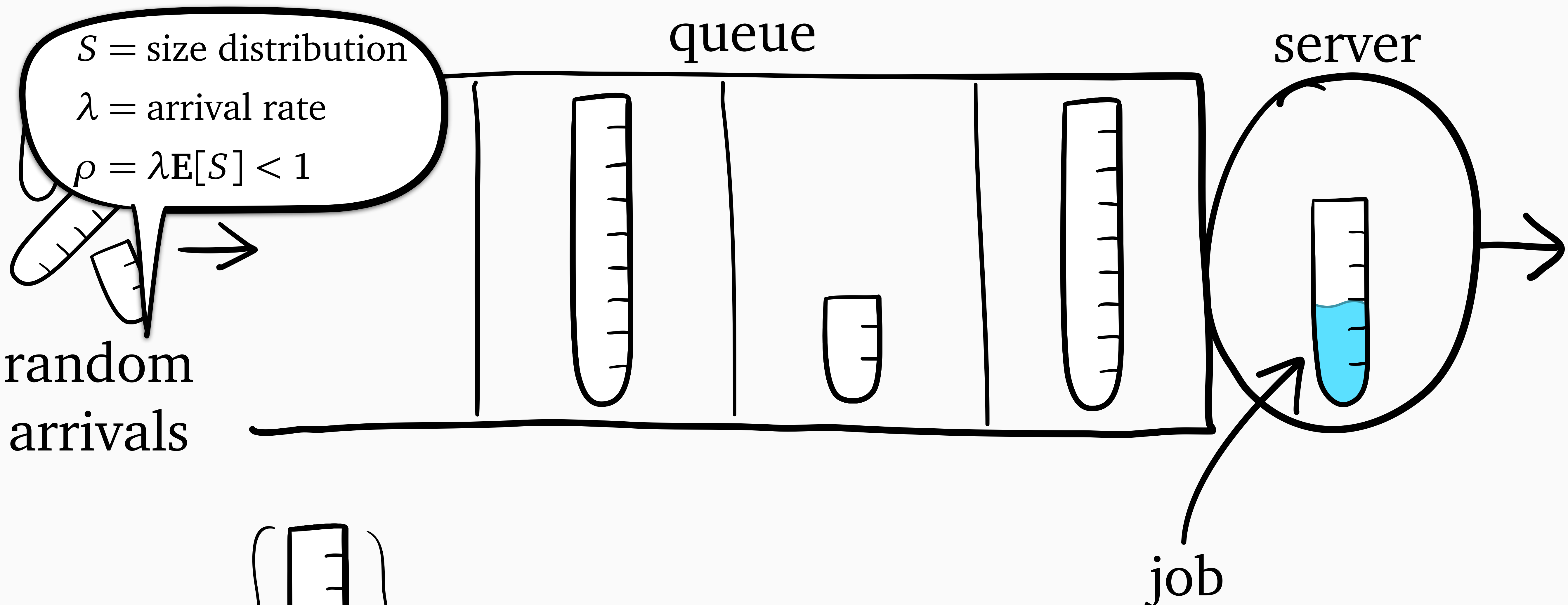


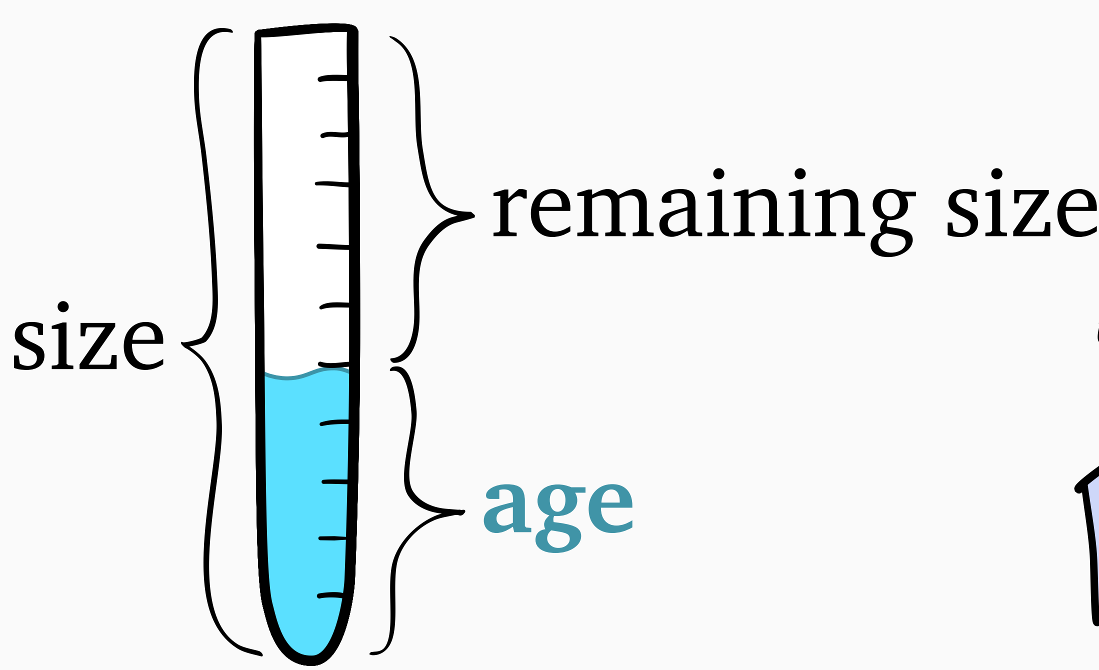
M/G/1 Queue

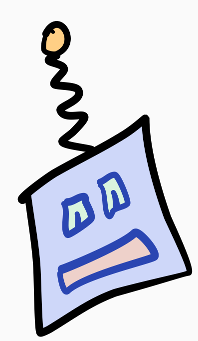


S = size distribution
λ = arrival rate
$\rho = \lambda \mathbf{E}[S] < 1$
random arrivals
queue
server
job
size
remaining size
age

M/G/1 Queue

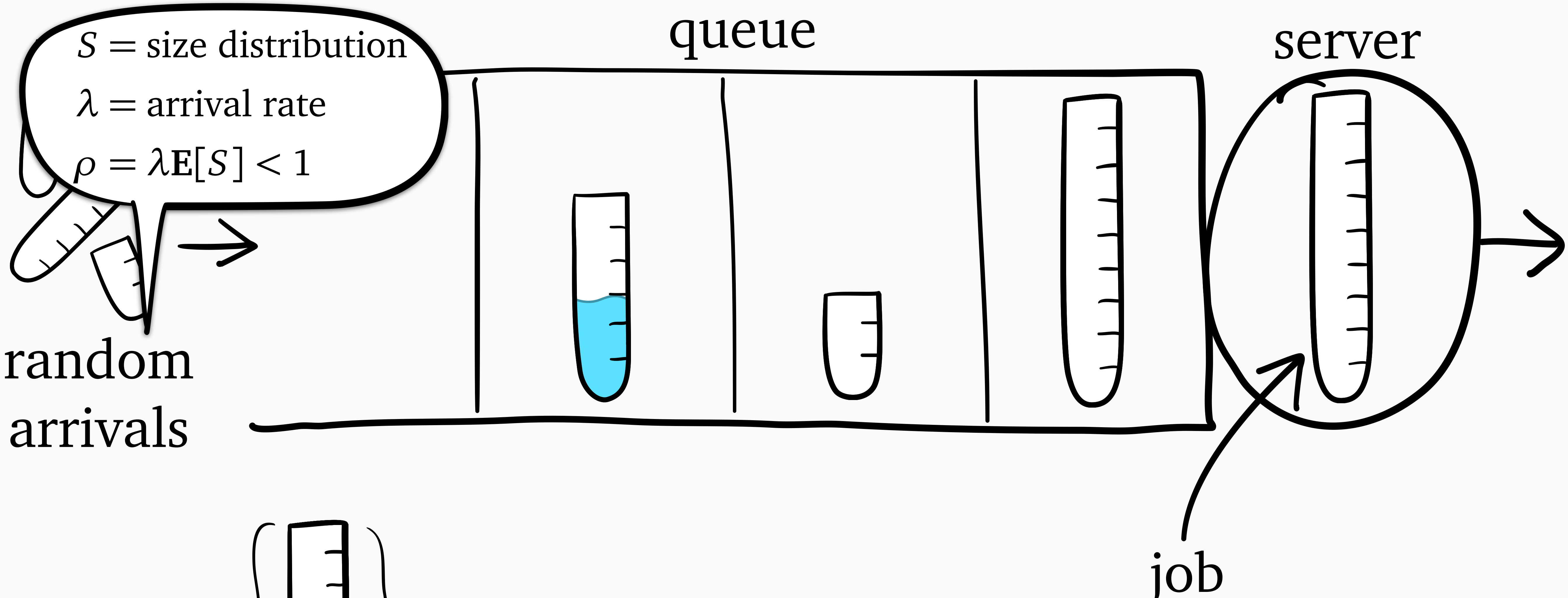


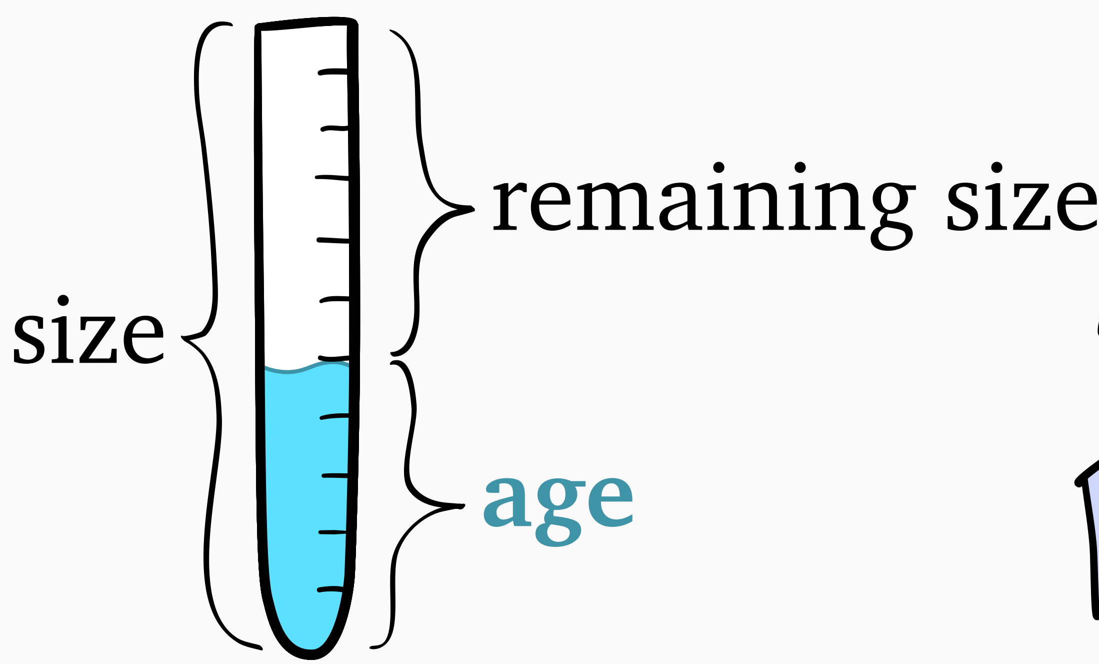


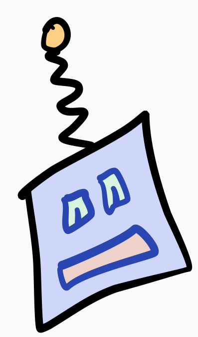


Scheduling policy:
picks which job to serve

M/G/1 Queue

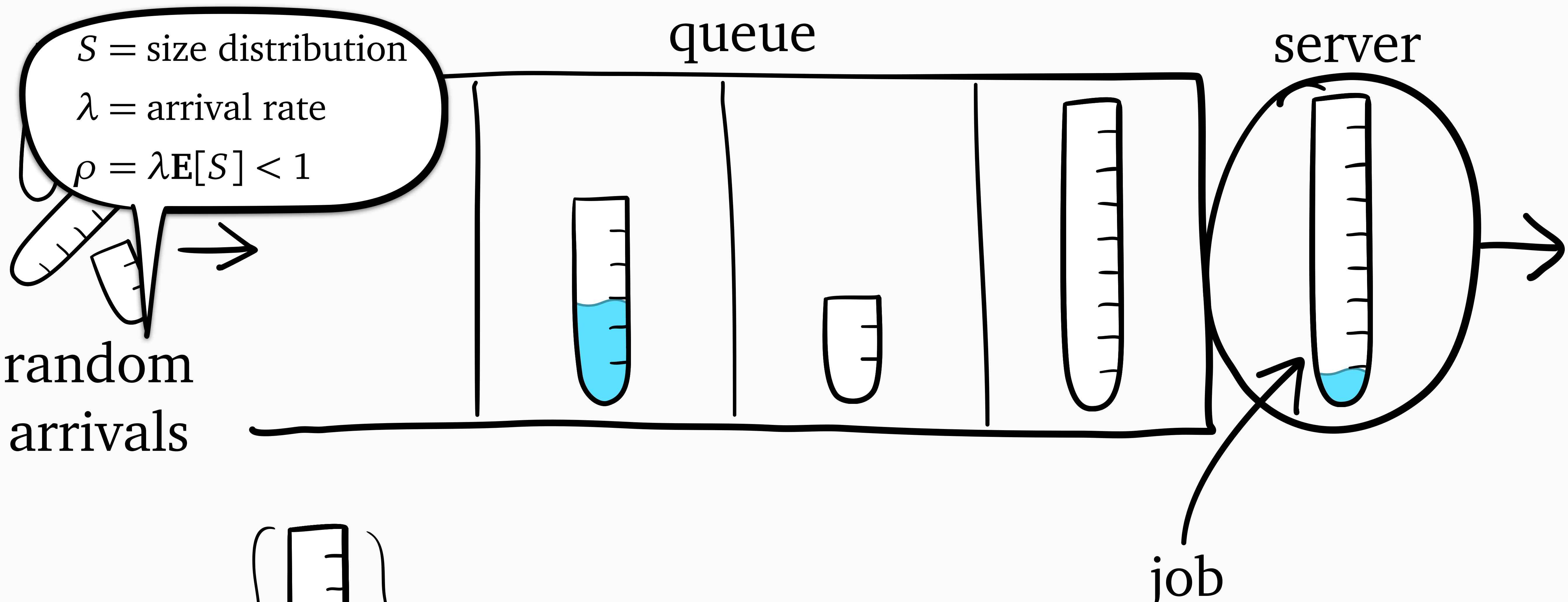


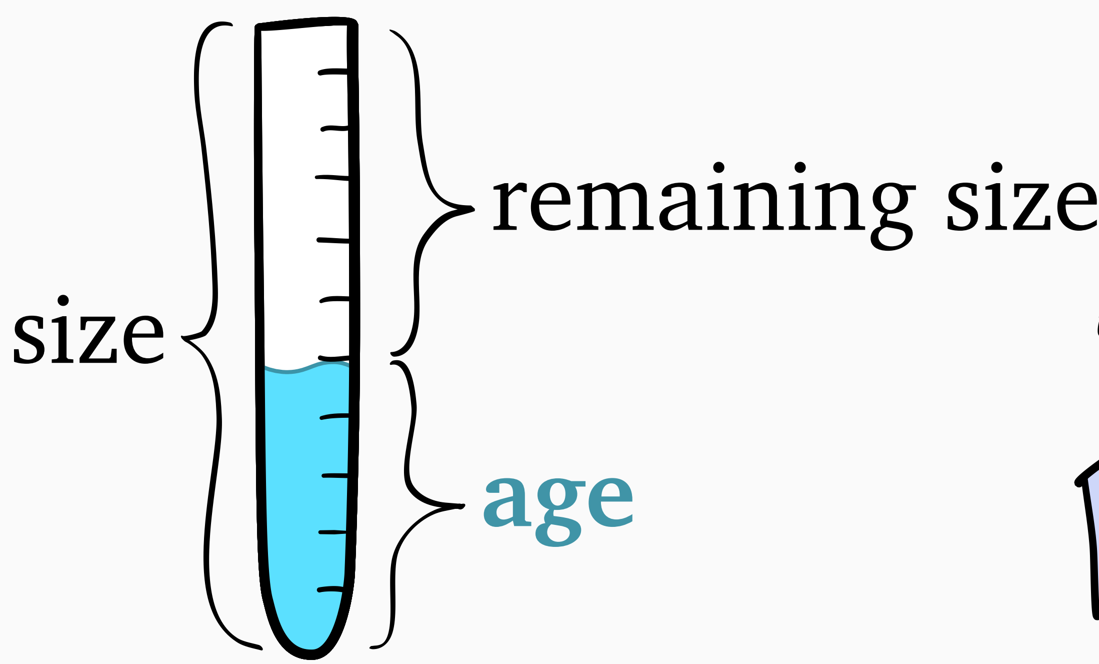


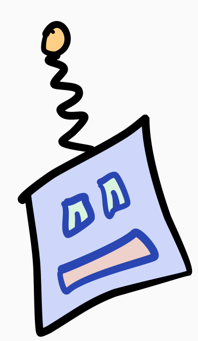


Scheduling policy:
picks which job to serve

M/G/1 Queue

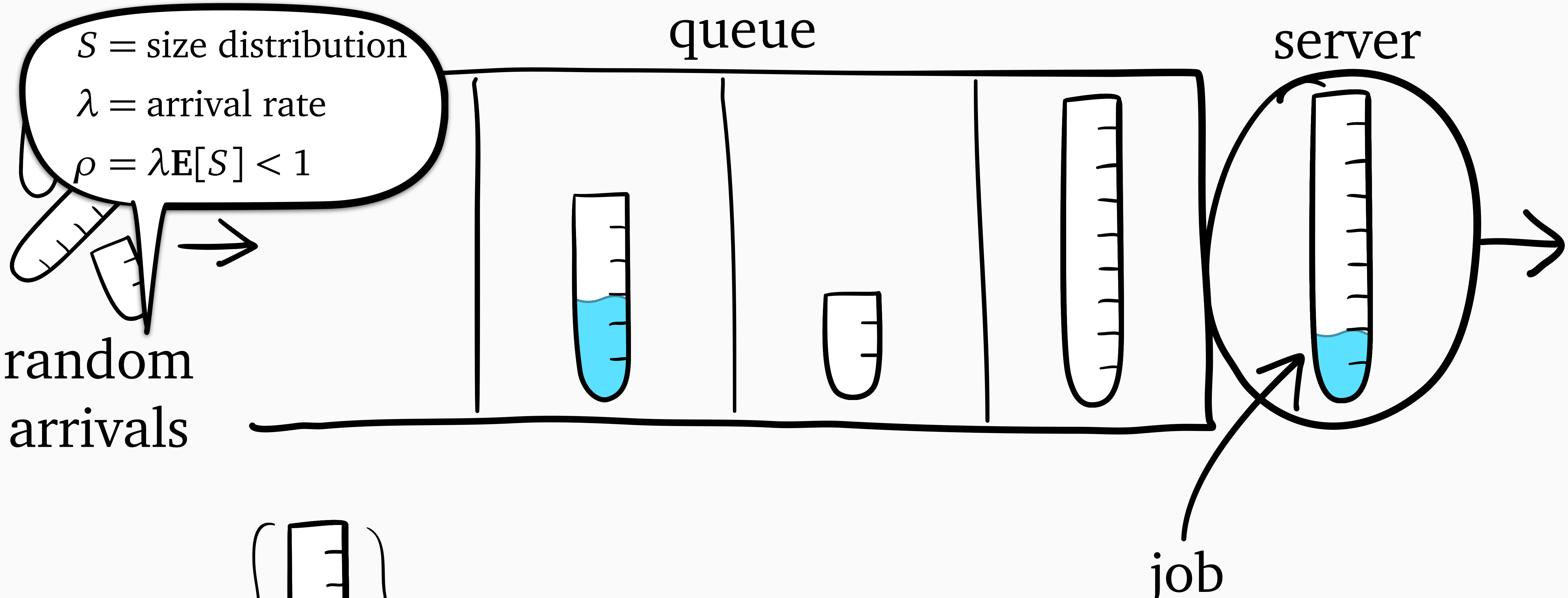


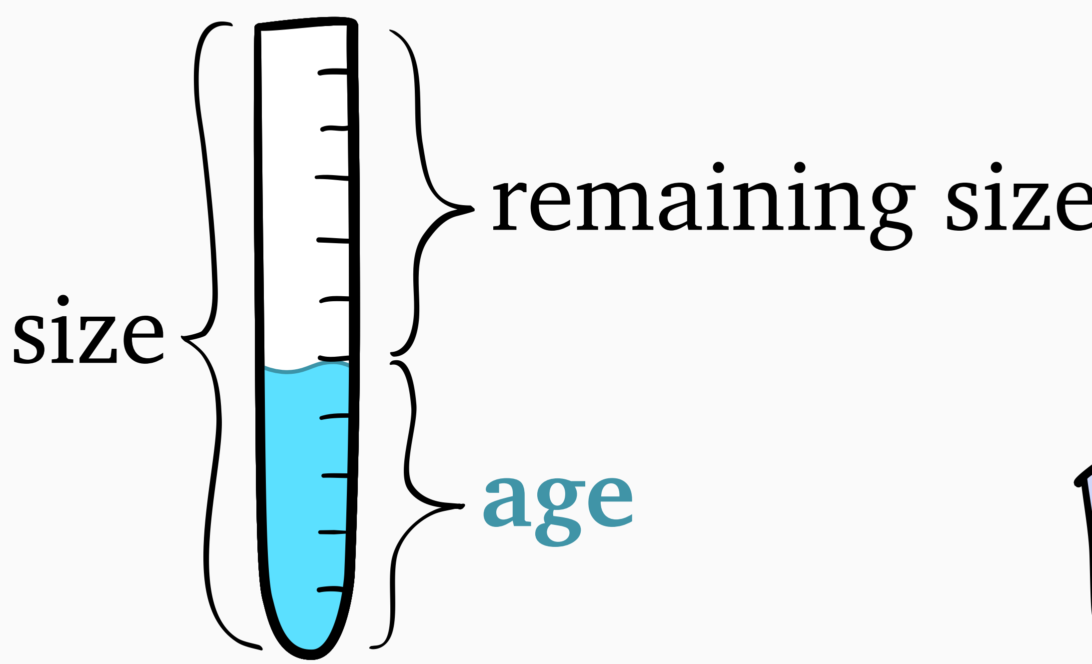


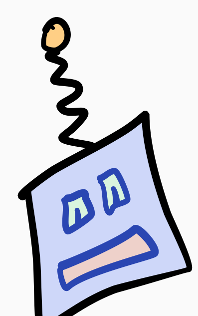


Scheduling policy:
picks which job to serve

M/G/1 Queue

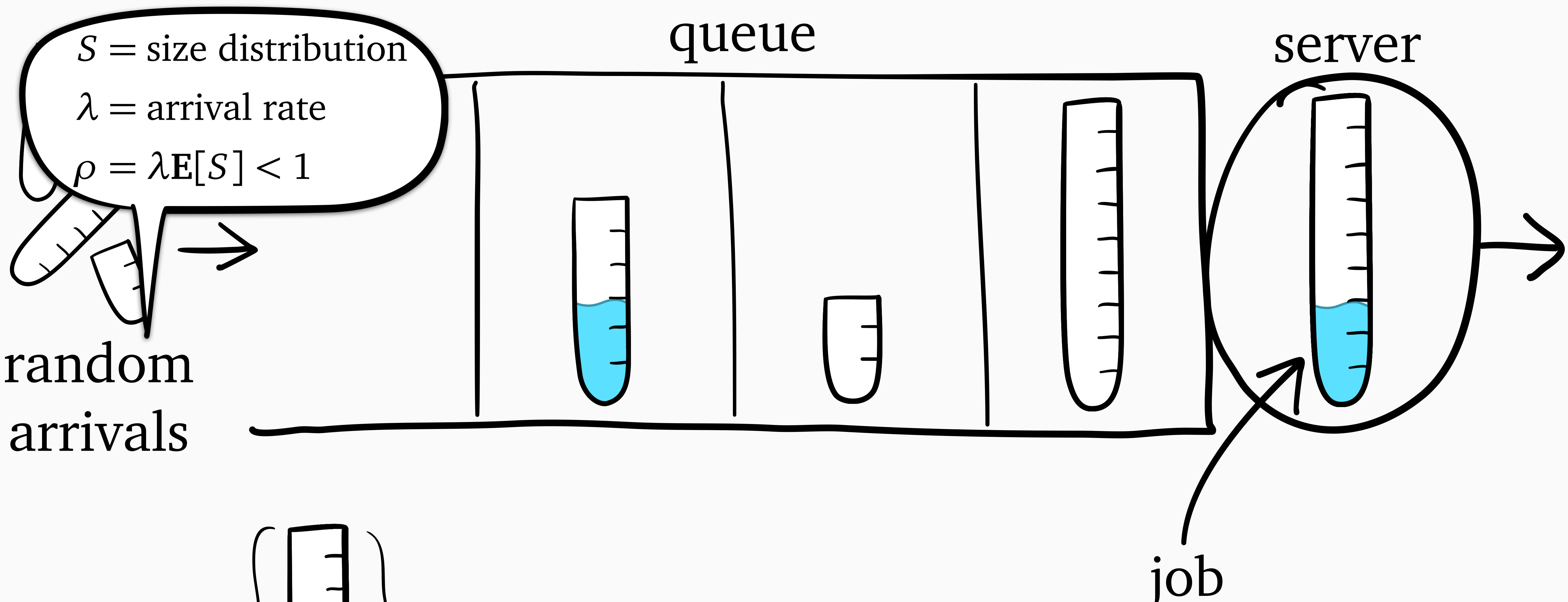


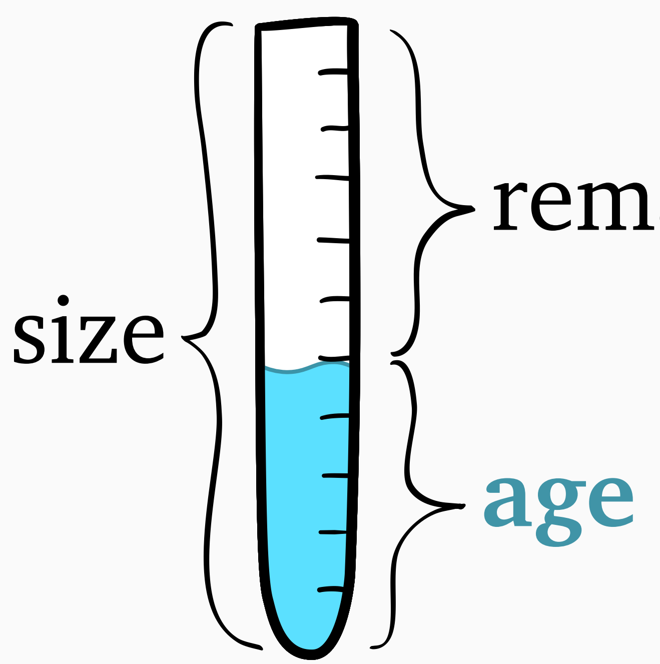


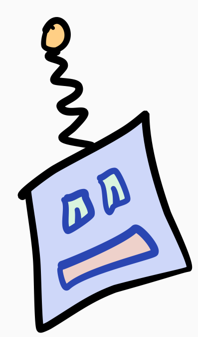


Scheduling policy:
picks which job to serve

M/G/1 Queue

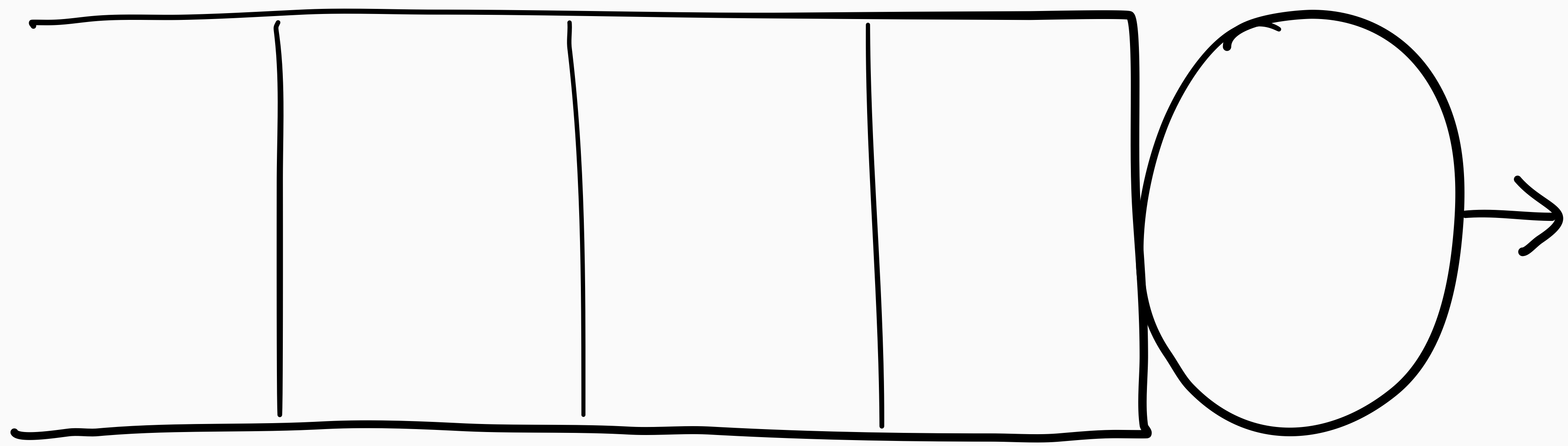




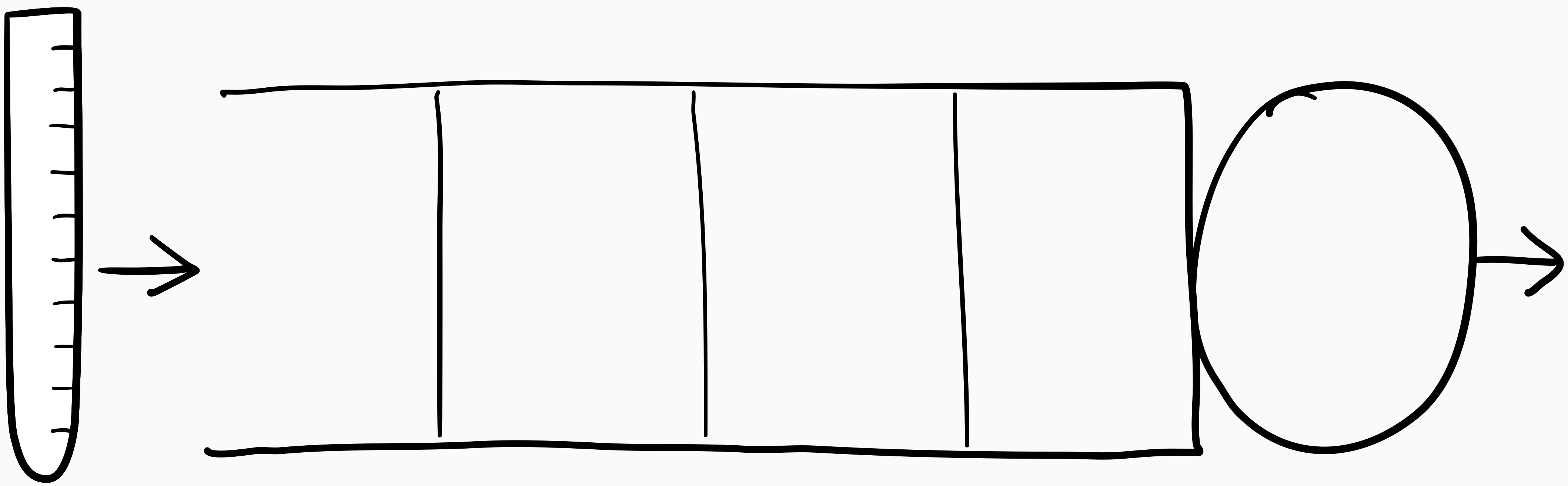


Scheduling policy:
picks which job to serve

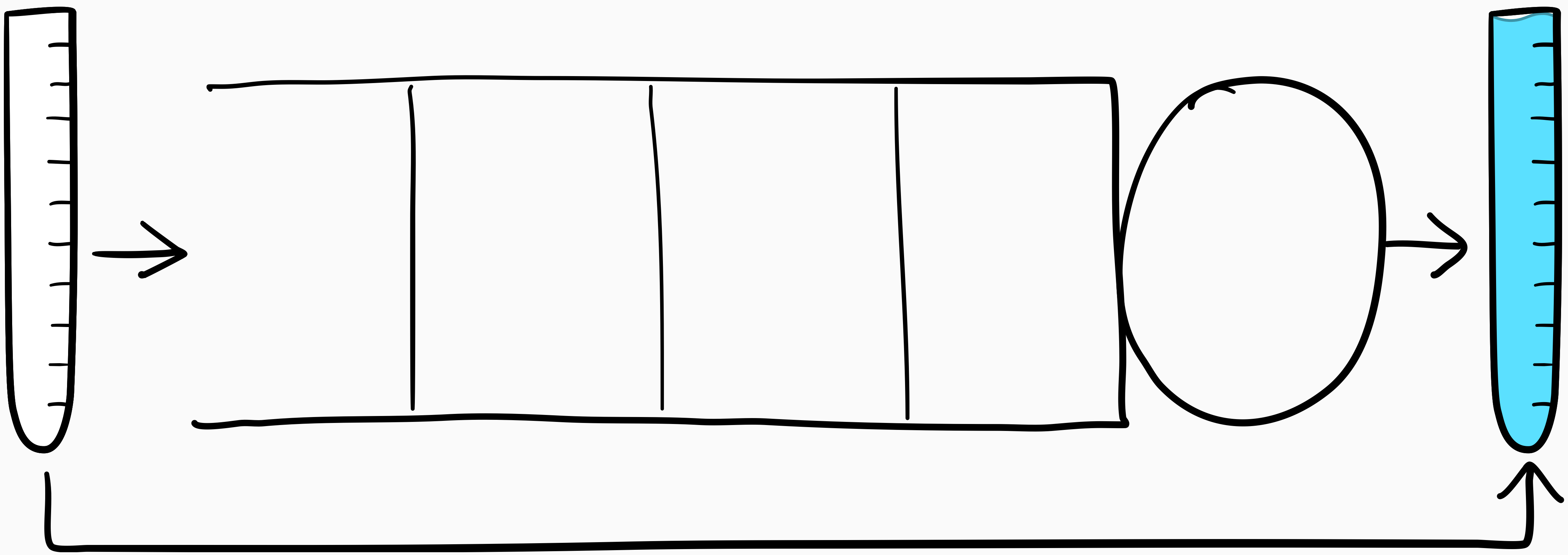
Response Time



Response Time

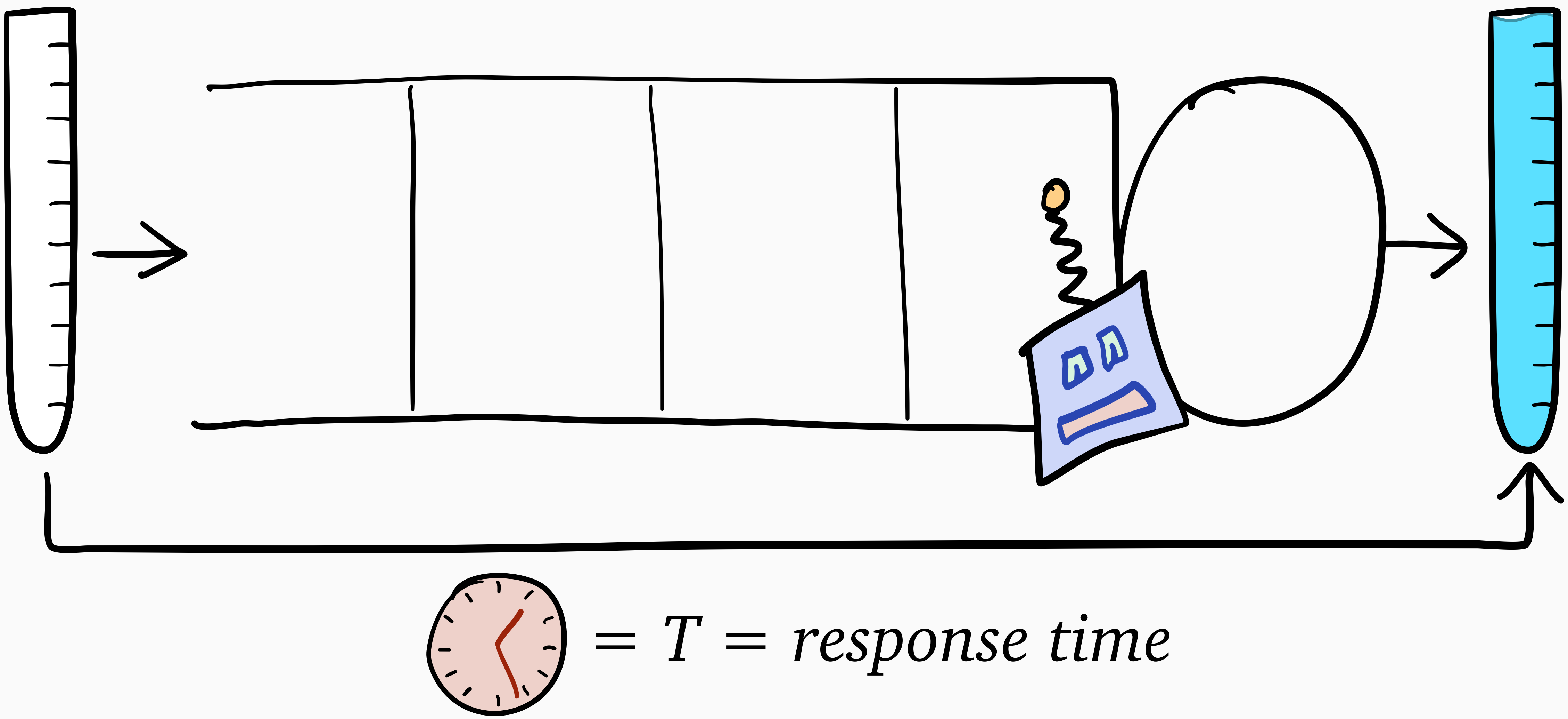


Response Time



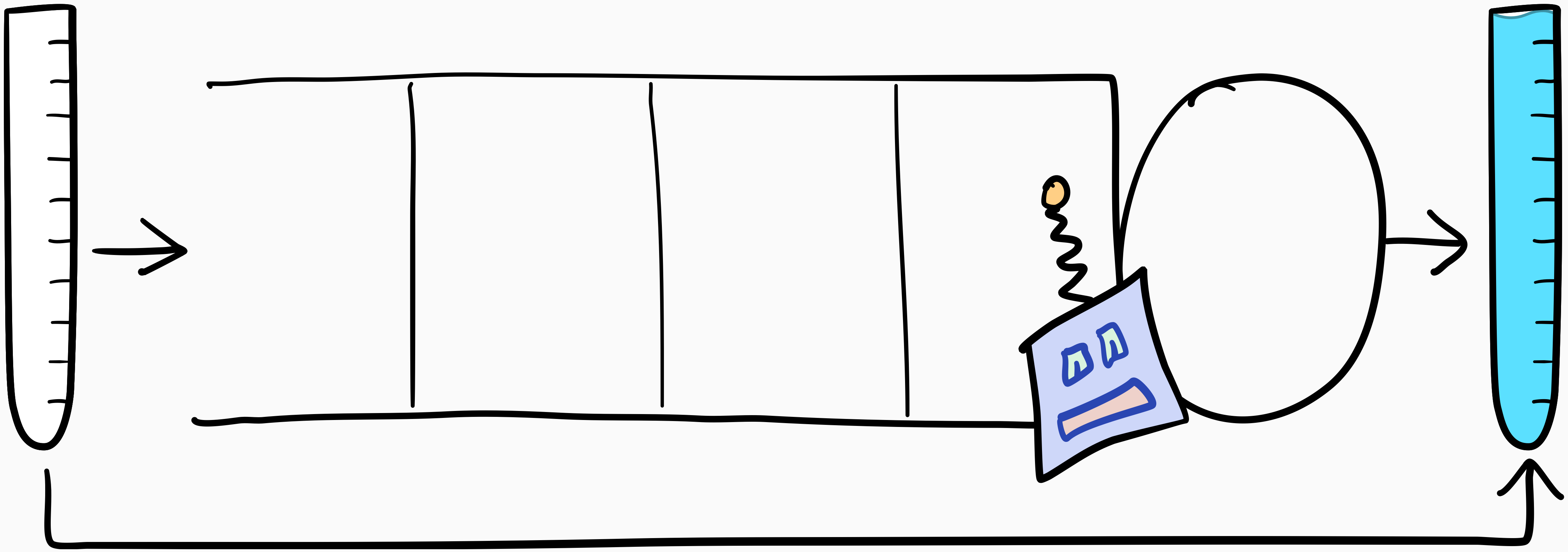
 $= T = \textit{response time}$

Response Time



= T = *response time*

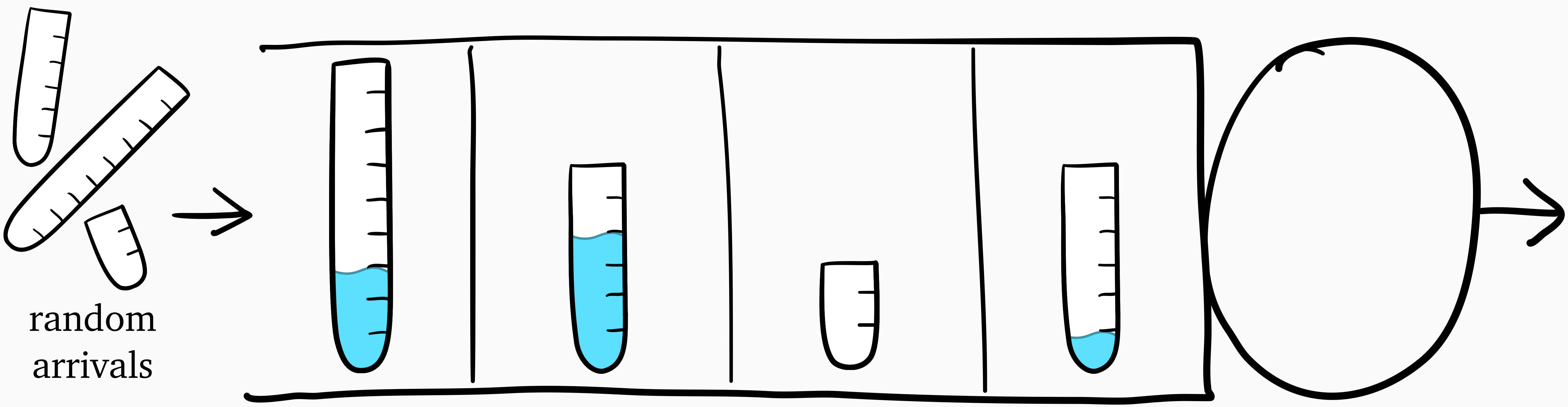
Response Time

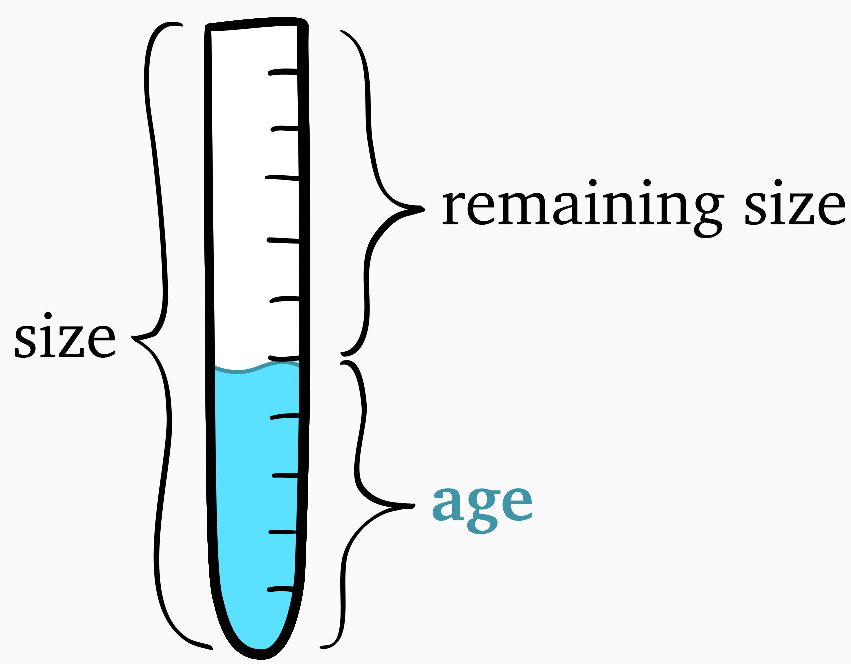


 $= T =$ *response time*

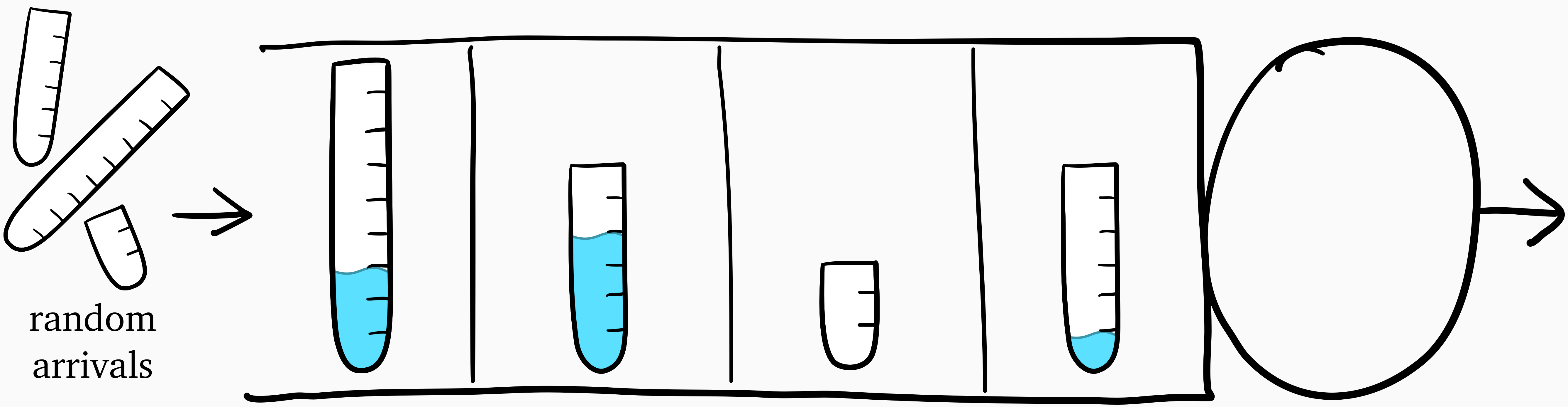
Goal: schedule to minimize *mean response time* $\mathbf{E}[T]$

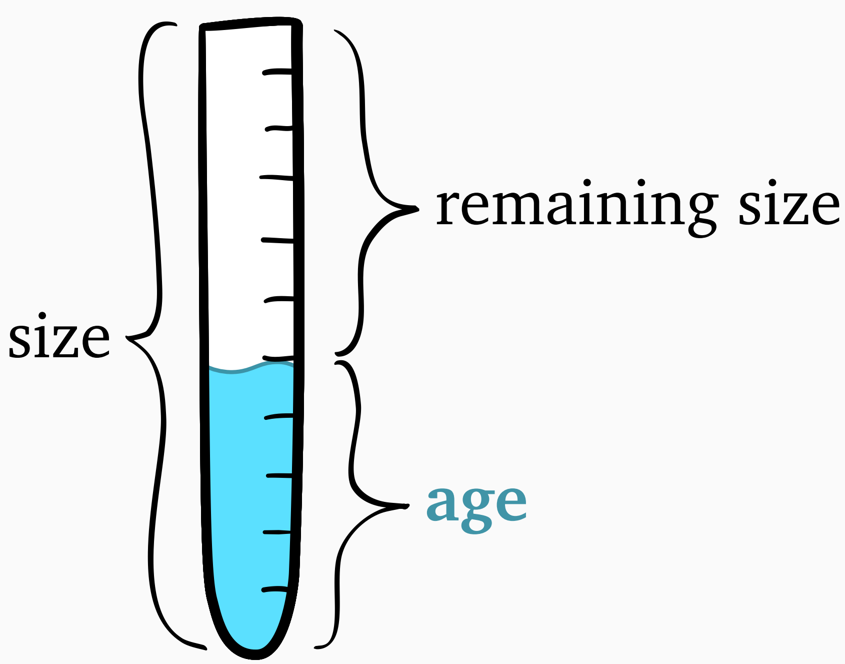
How to Schedule?

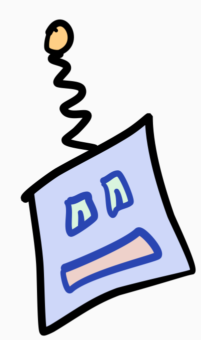




How to Schedule?

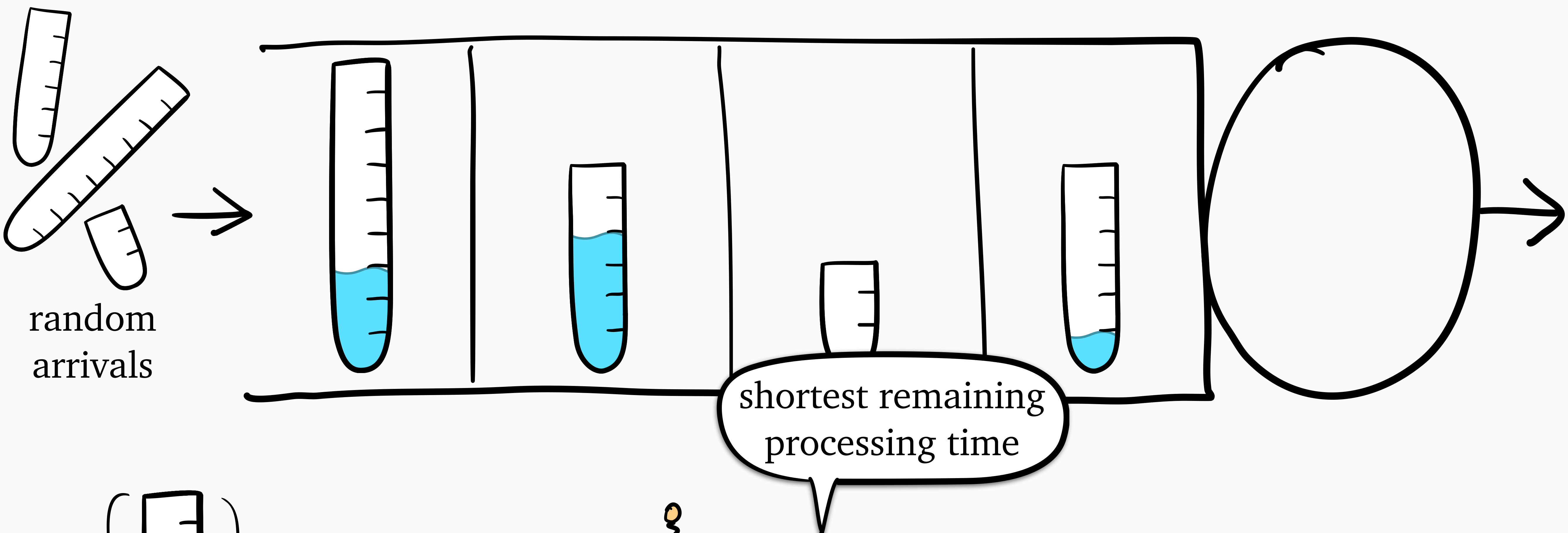


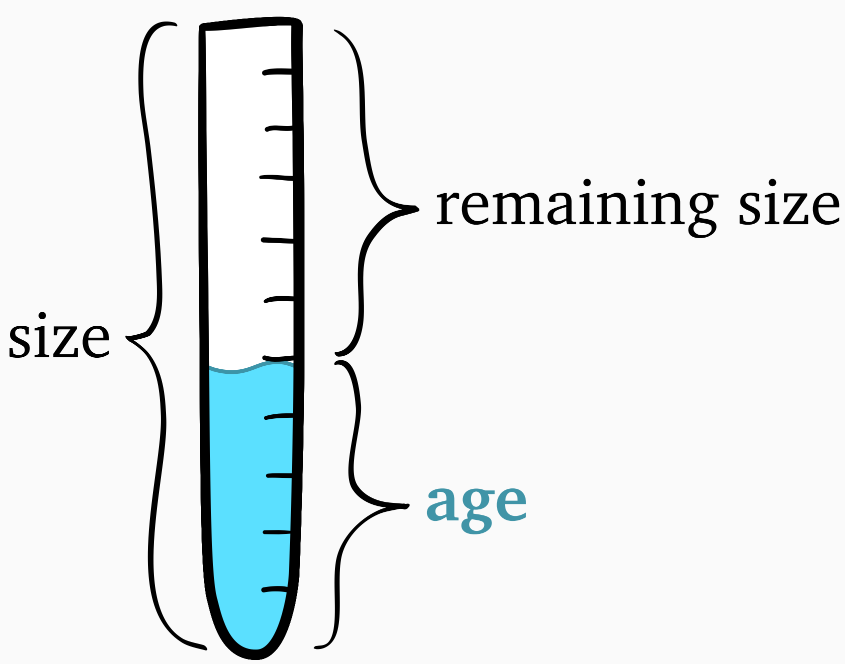


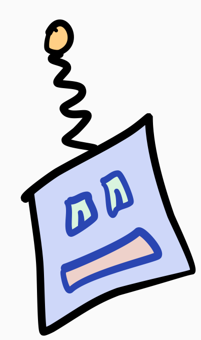


SRPT: always serve job of *least remaining size*

How to Schedule?

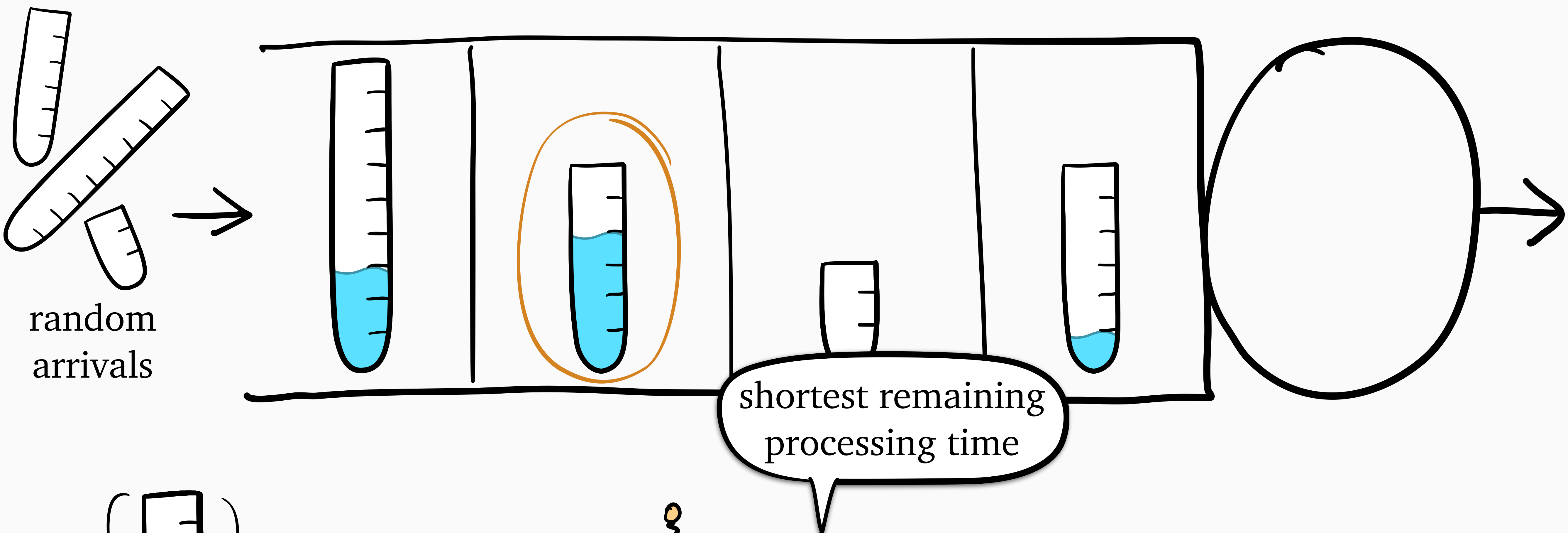


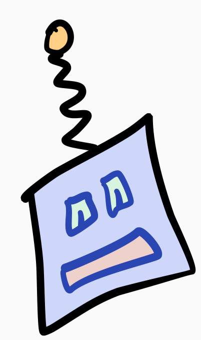




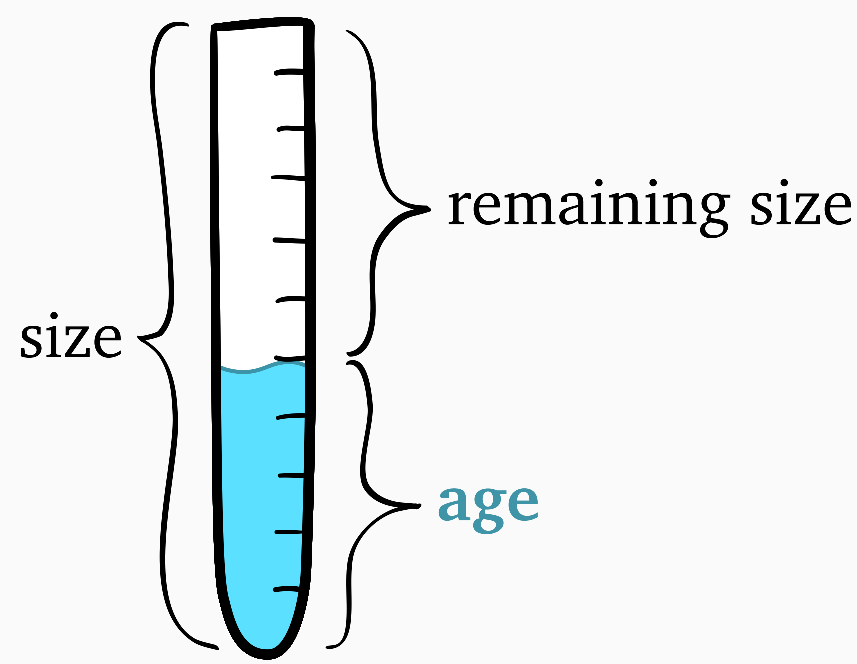
SRPT: always serve job of *least remaining size*

How to Schedule?

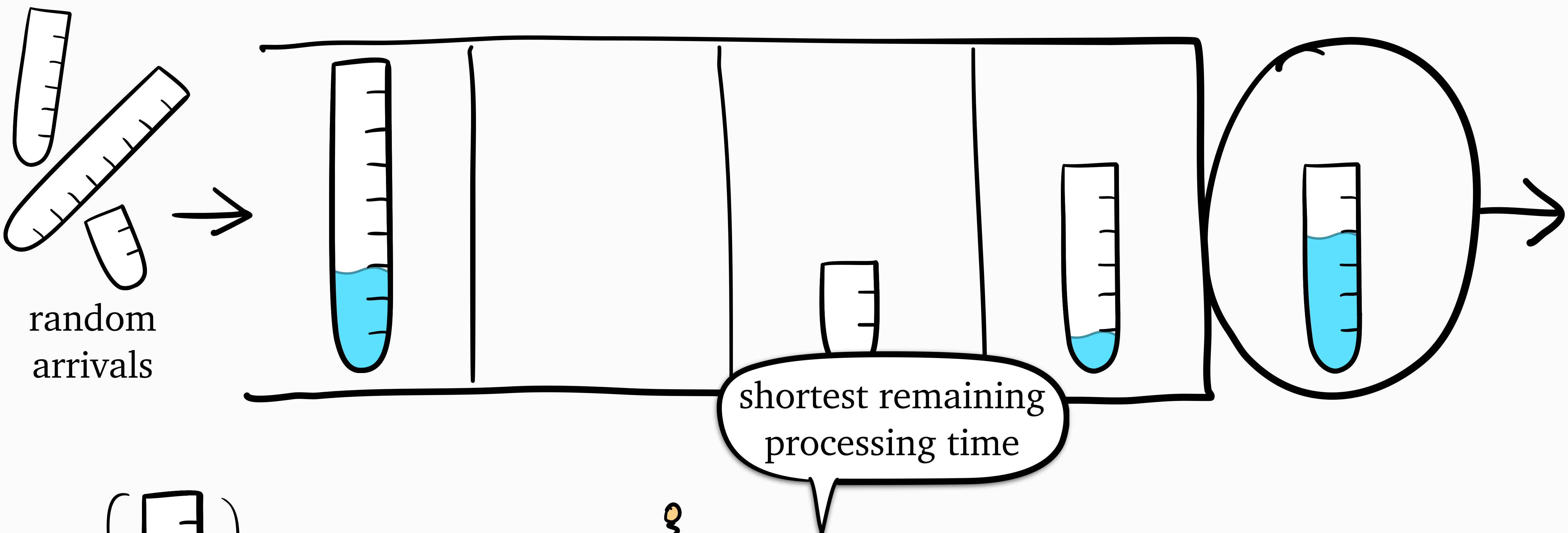


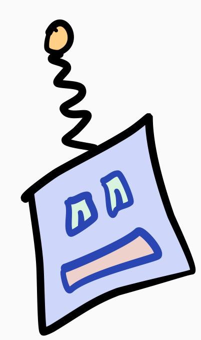


SRPT: always serve job of *least remaining size*

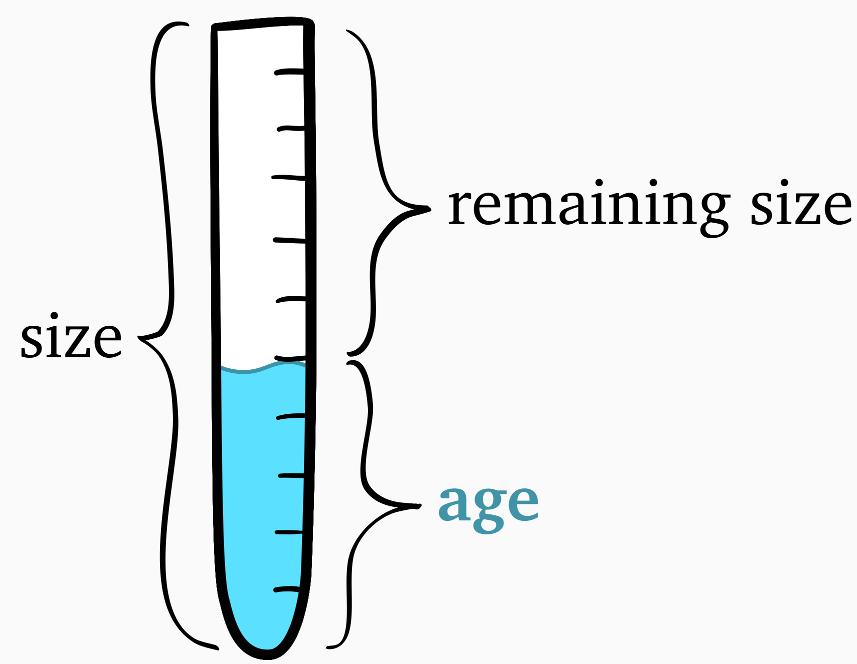


How to Schedule?

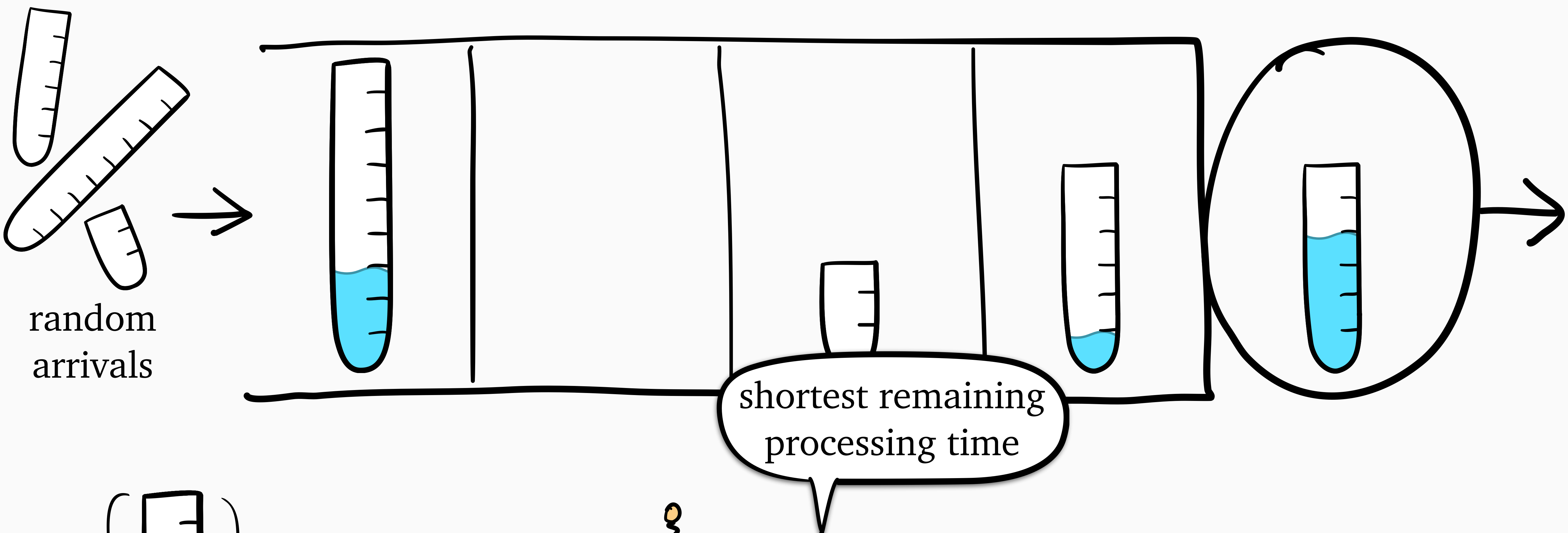


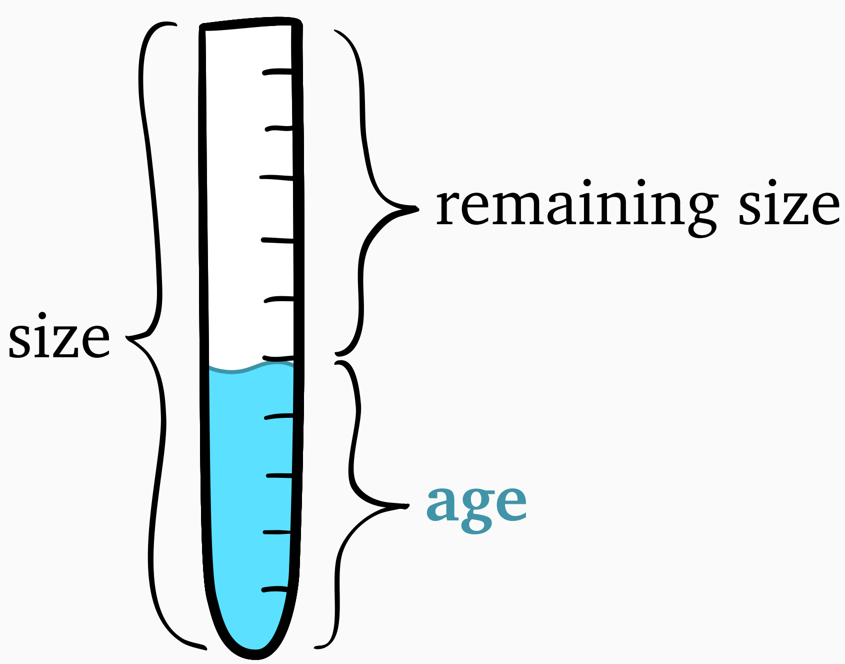


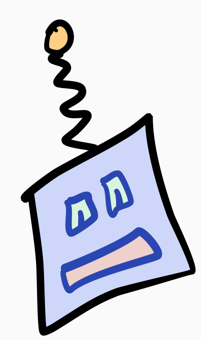
SRPT: always serve job of *least remaining size*



How to Schedule?



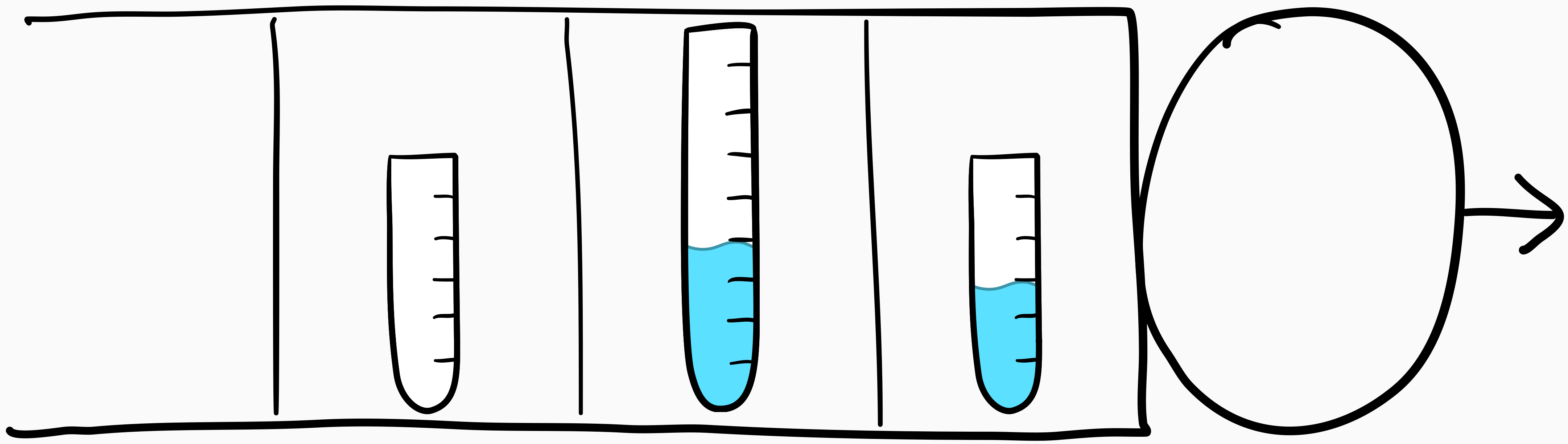




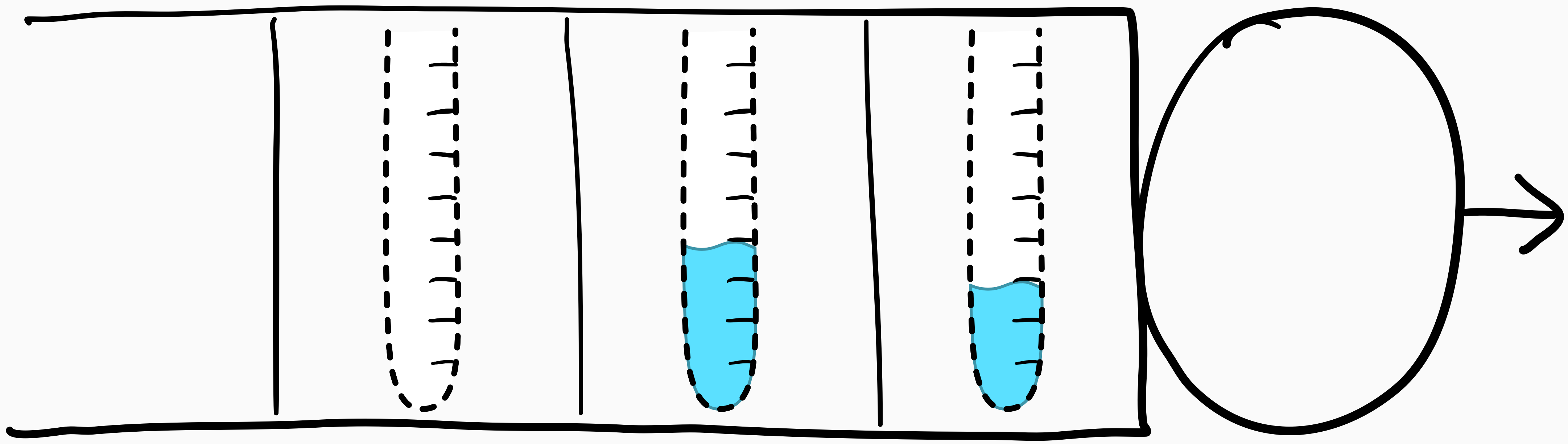
SRPT: always serve job of *least remaining size*

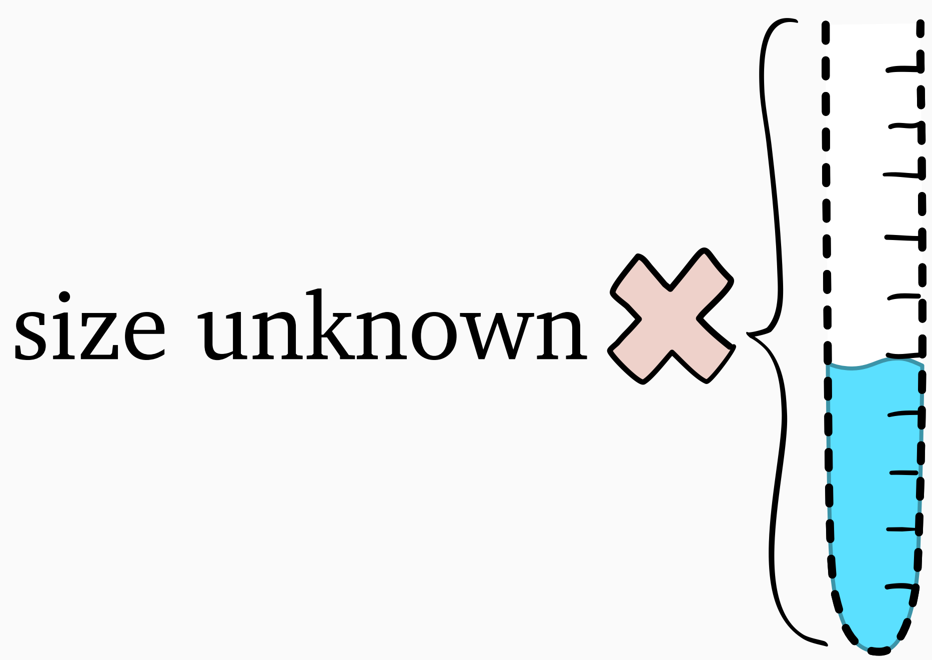
SRPT minimizes $\mathbf{E}[T]$
(Schrage 1968)

Unknown Job Sizes

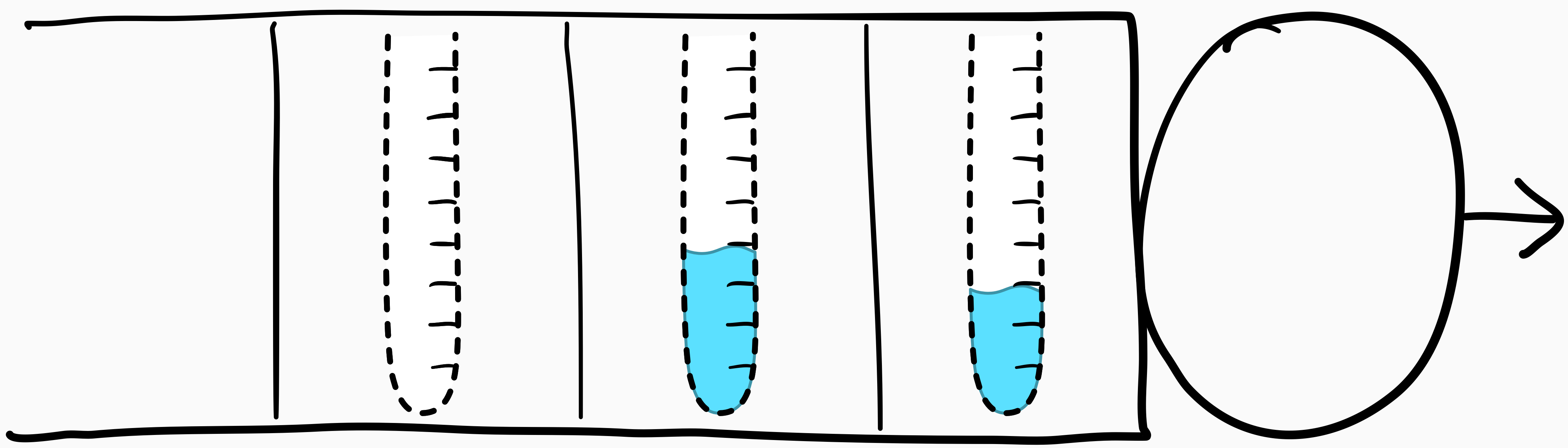


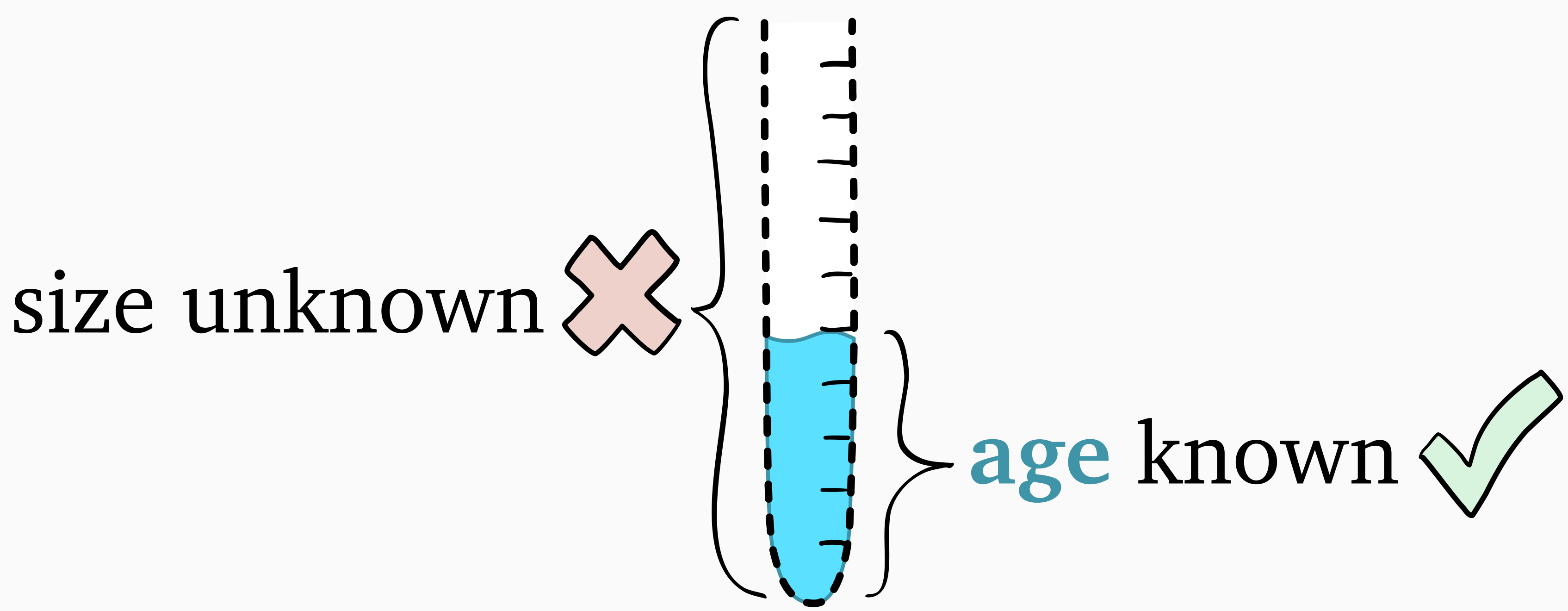
Unknown Job Sizes



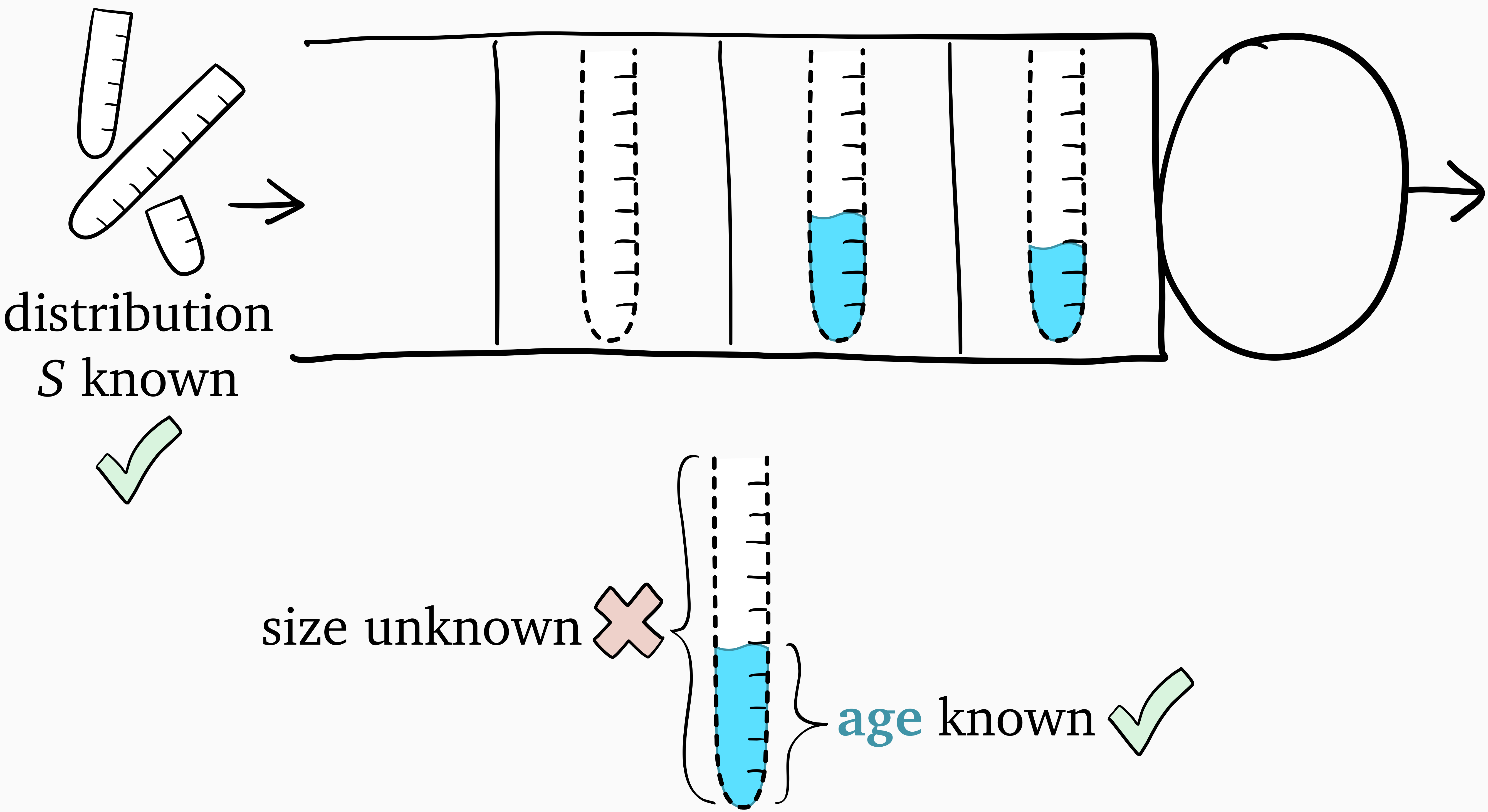


Unknown Job Sizes

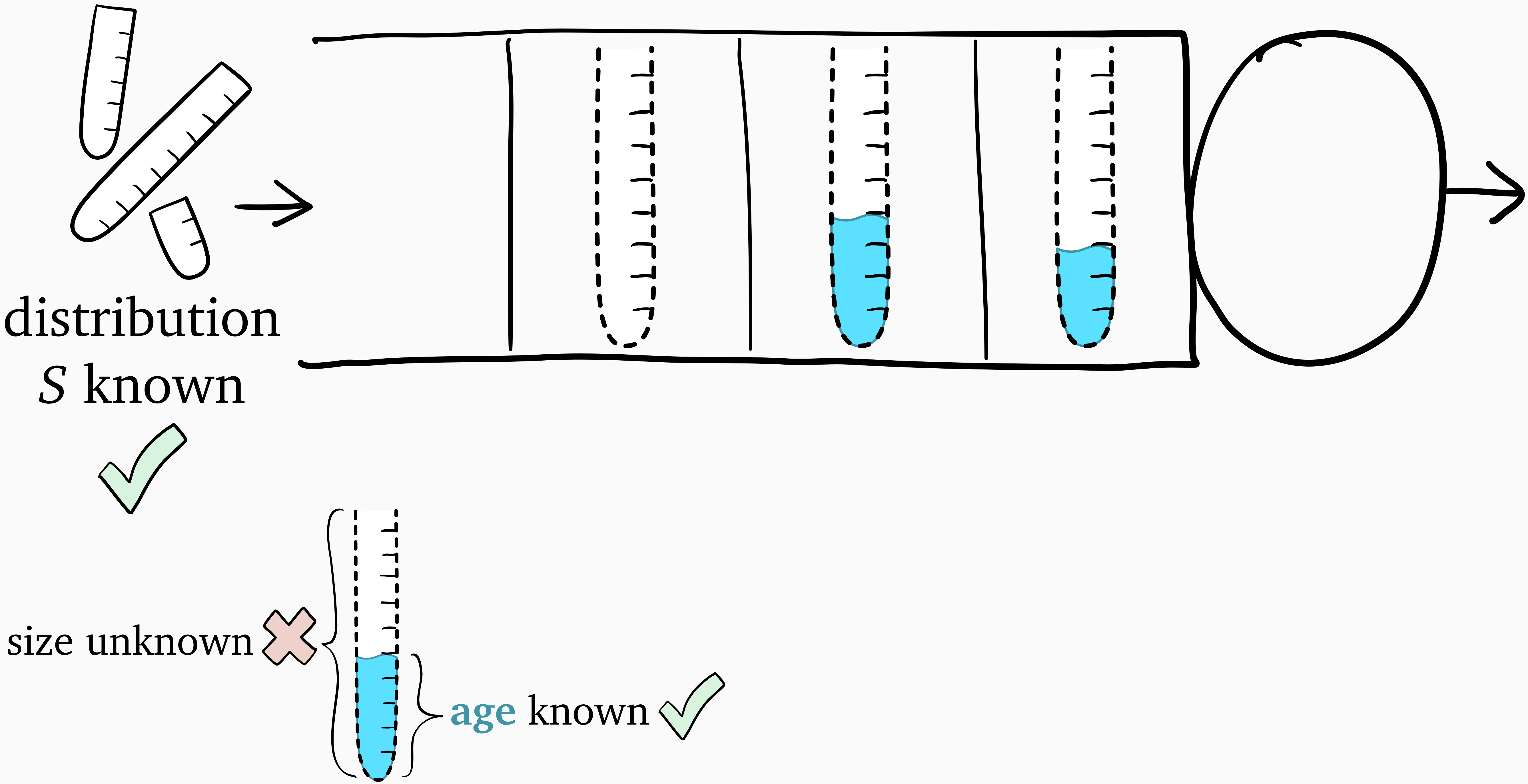




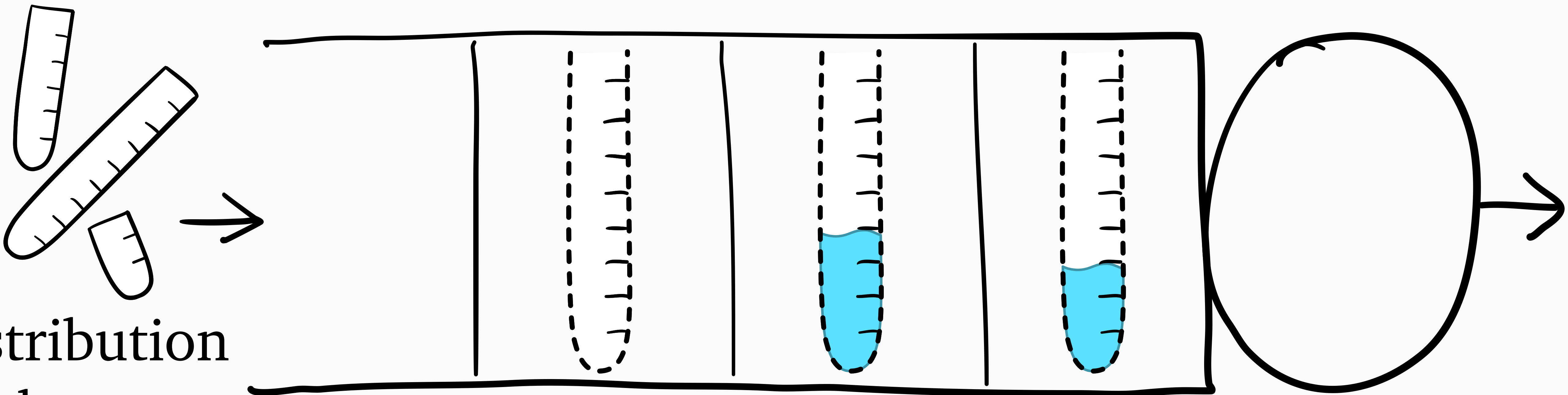
Unknown Job Sizes



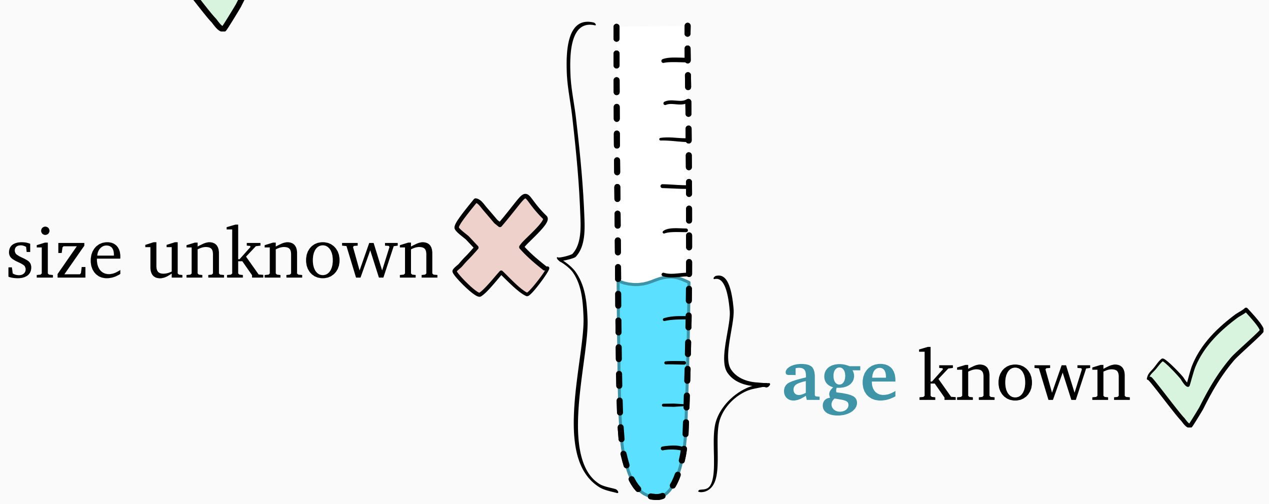
Unknown Job Sizes

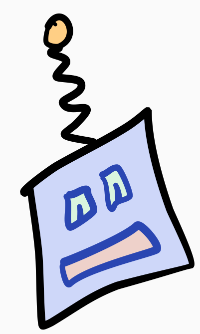


Unknown Job Sizes



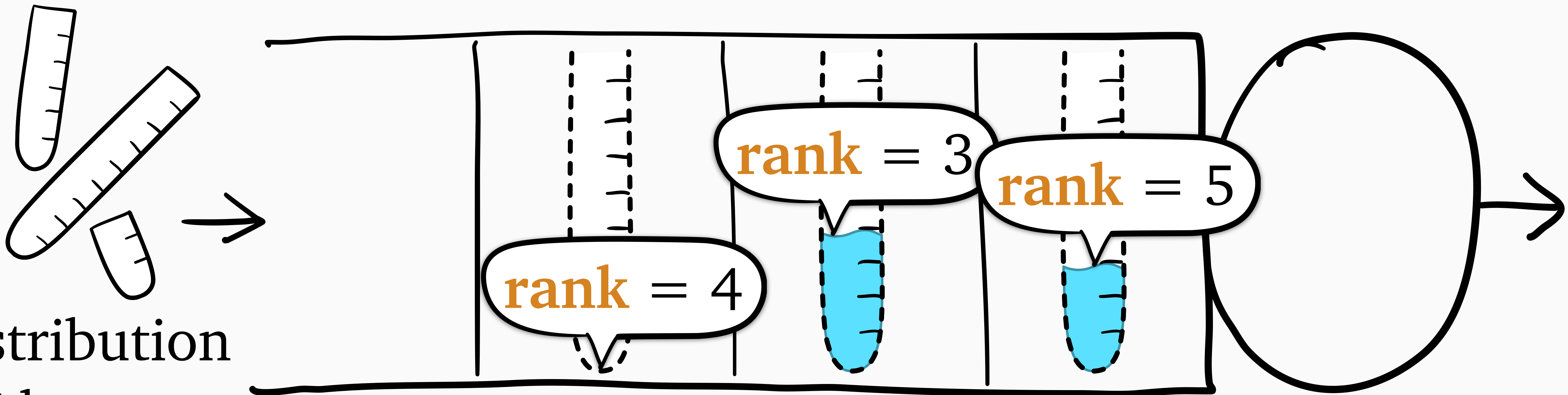
distribution
S known



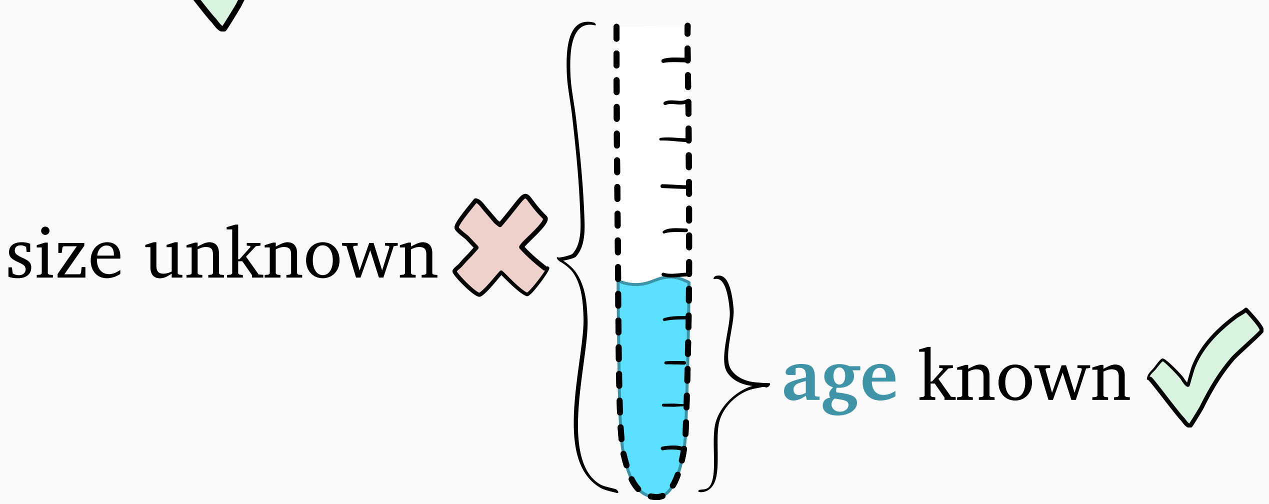


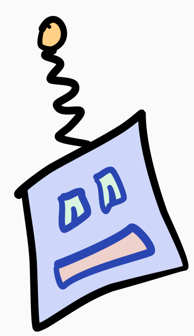
Gittins: assign each job a **rank** based on **age** and S
(lower is better)

Unknown Job Sizes



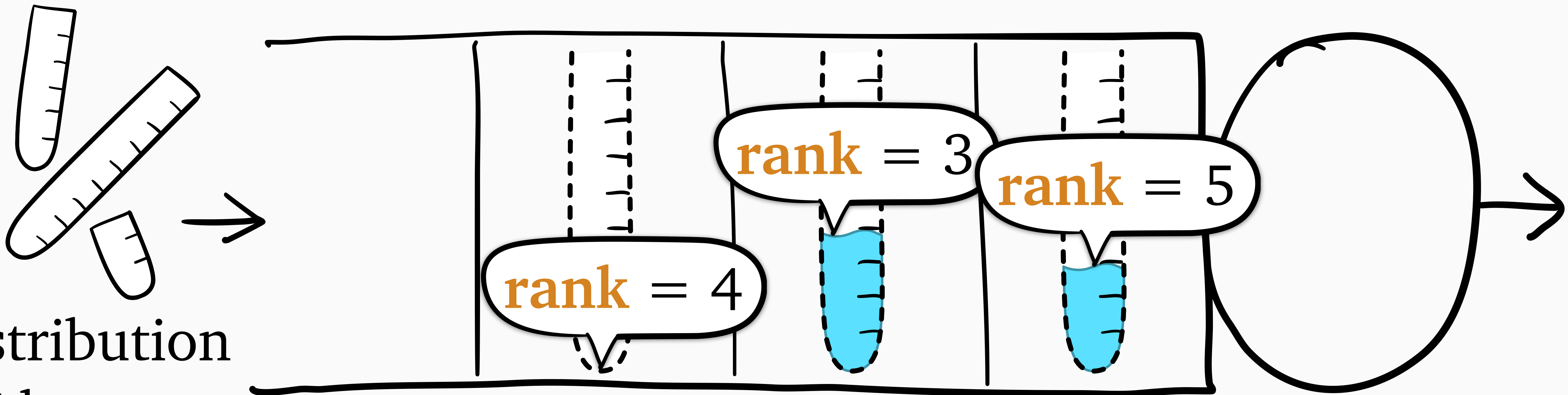
distribution
S known



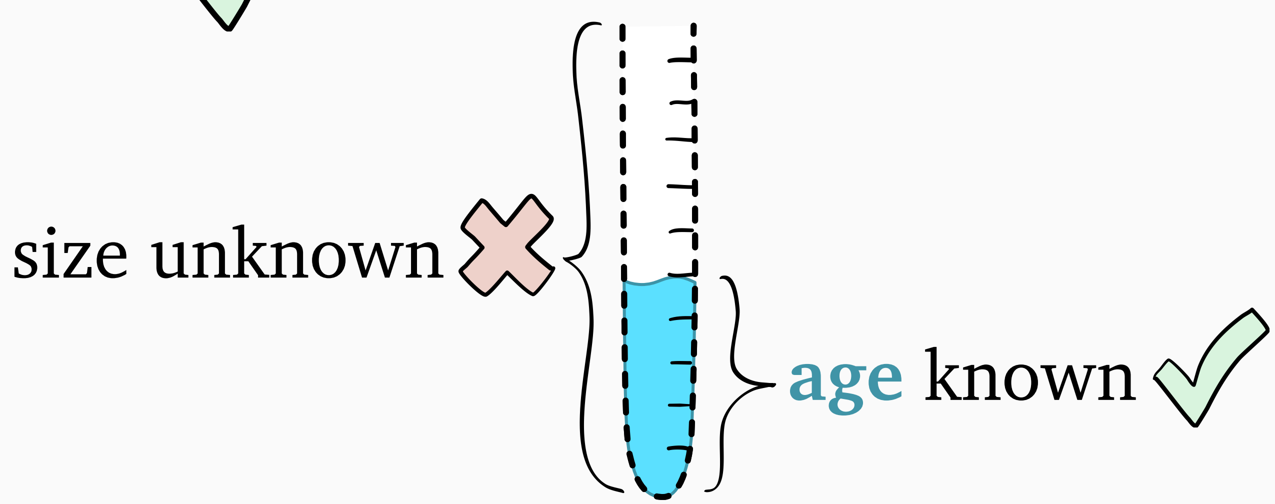


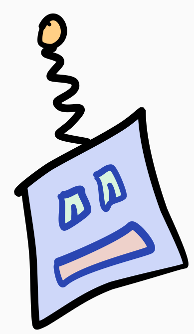
Gittins: assign each job a **rank** based on **age** and S
(lower is better)

Unknown Job Sizes



distribution
S known

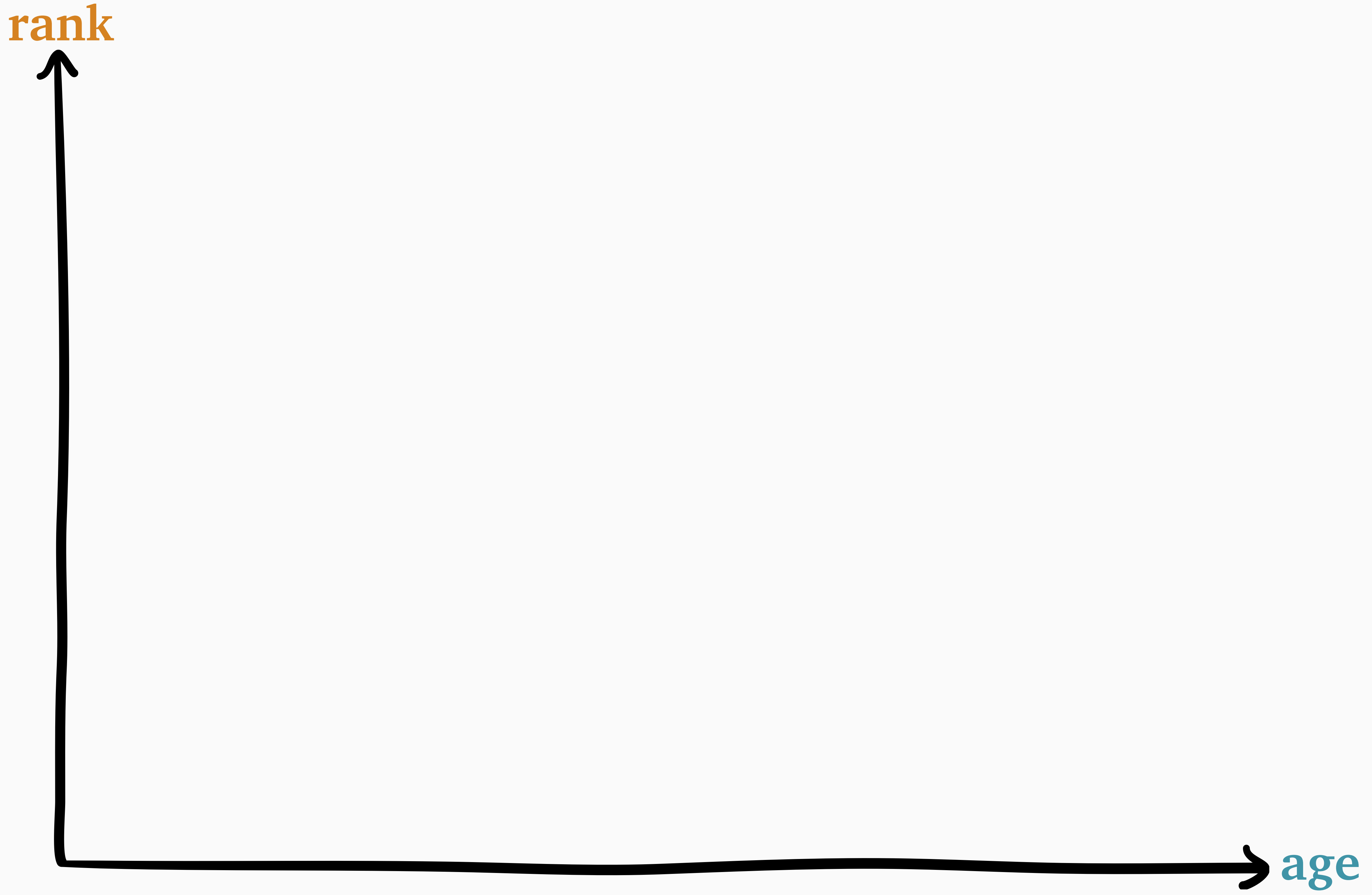




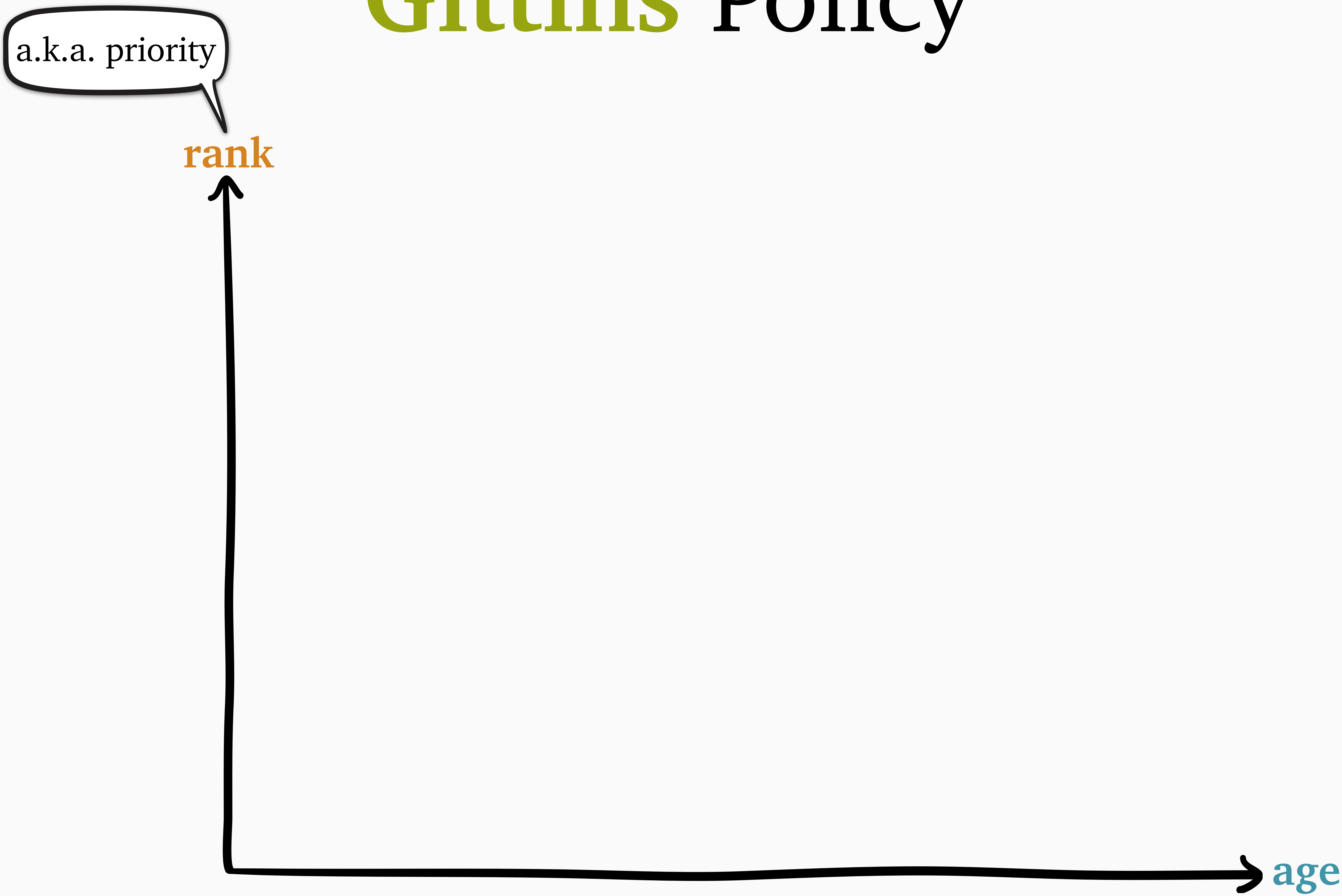
Gittins: assign each job a **rank** based on **age** and S
(lower is better)

Gittins minimizes $\mathbf{E}[T]$
(Gittins 1989)

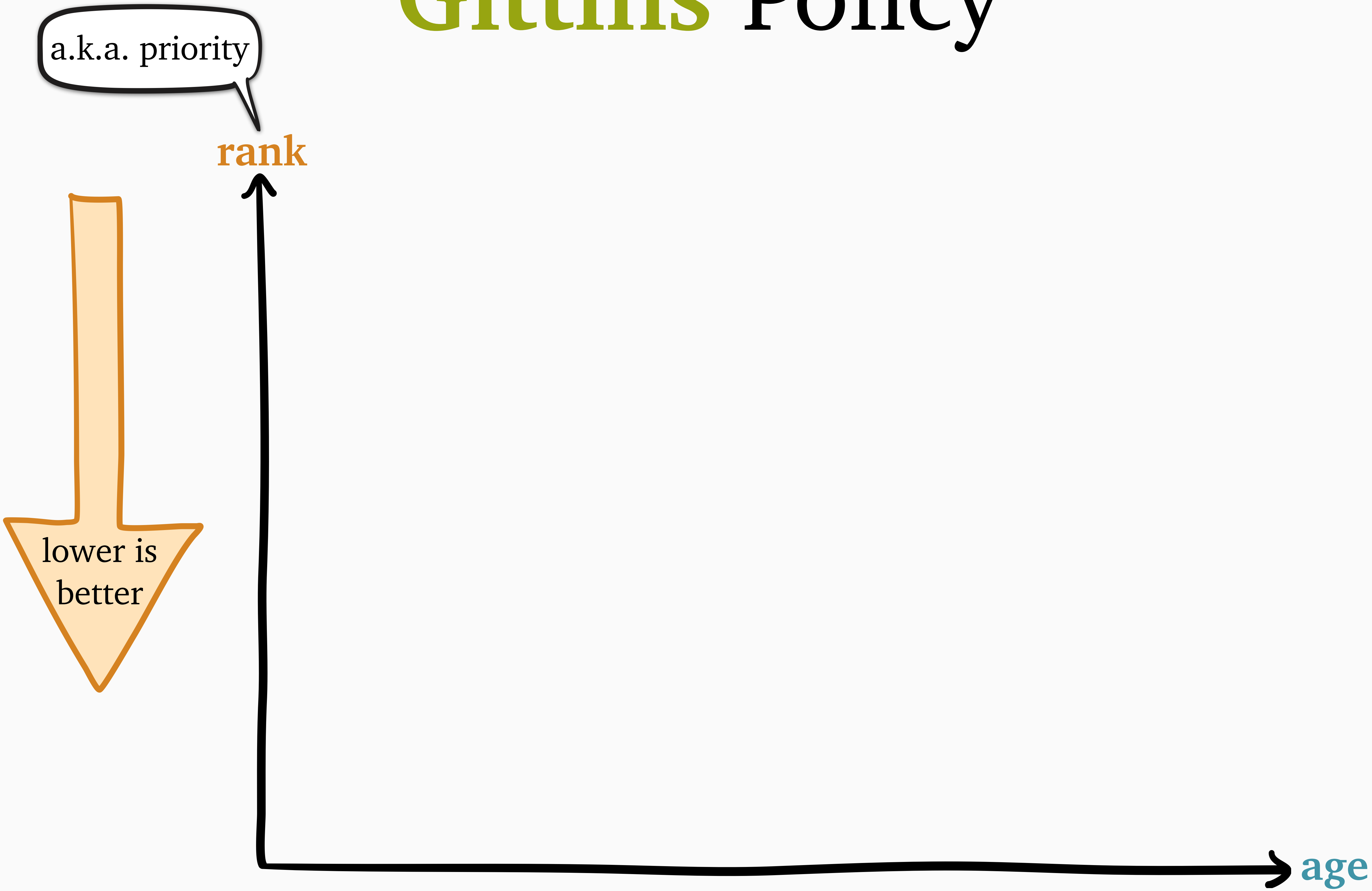
Gittins Policy



Gittins Policy

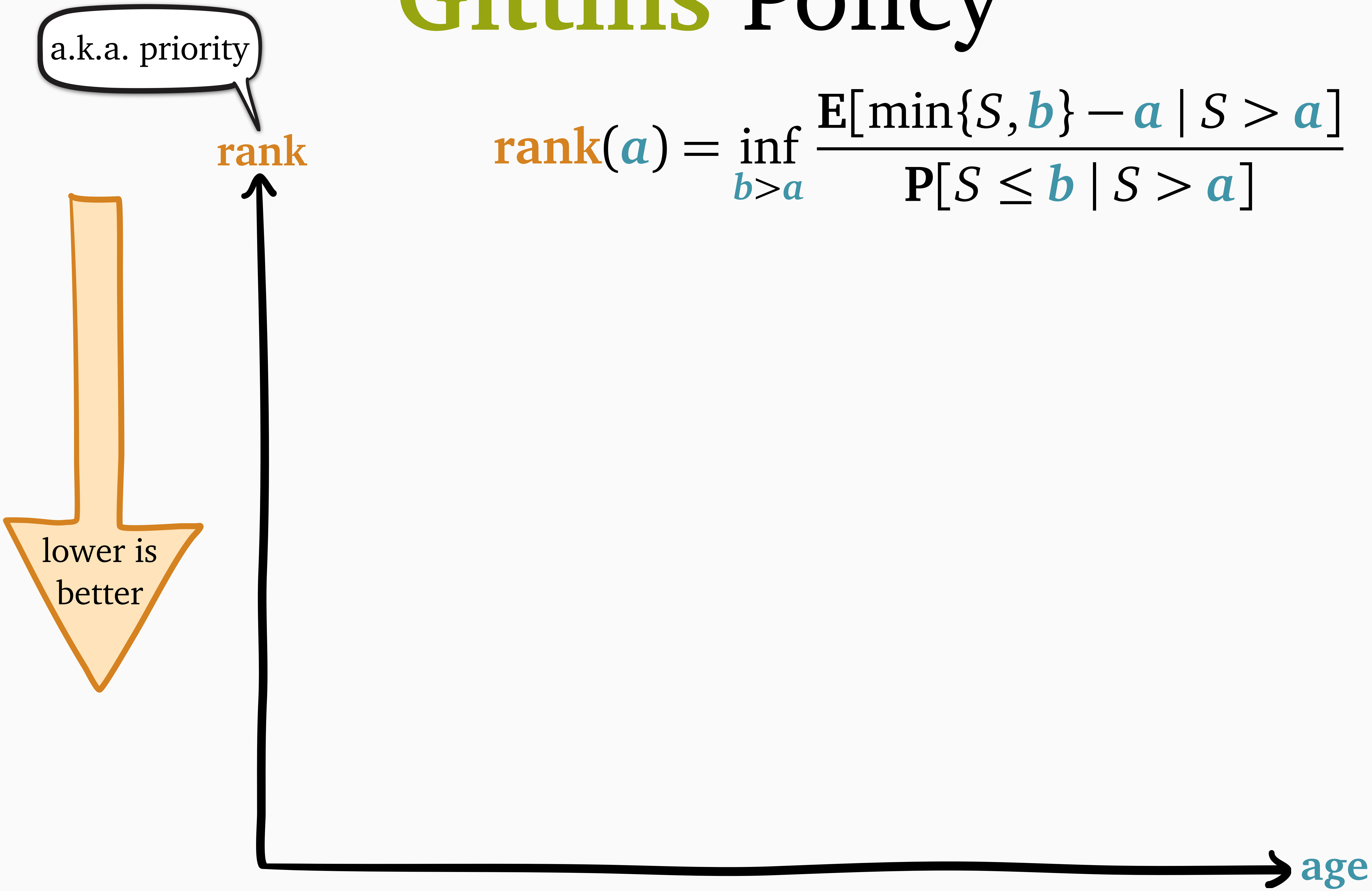


Gittins Policy



Gittins Policy

$$\text{rank}(a) = \inf_{b>a} \frac{\mathbf{E}[\min\{S, b\} - a \mid S > a]}{\mathbf{P}[S \le b \mid S > a]}$$

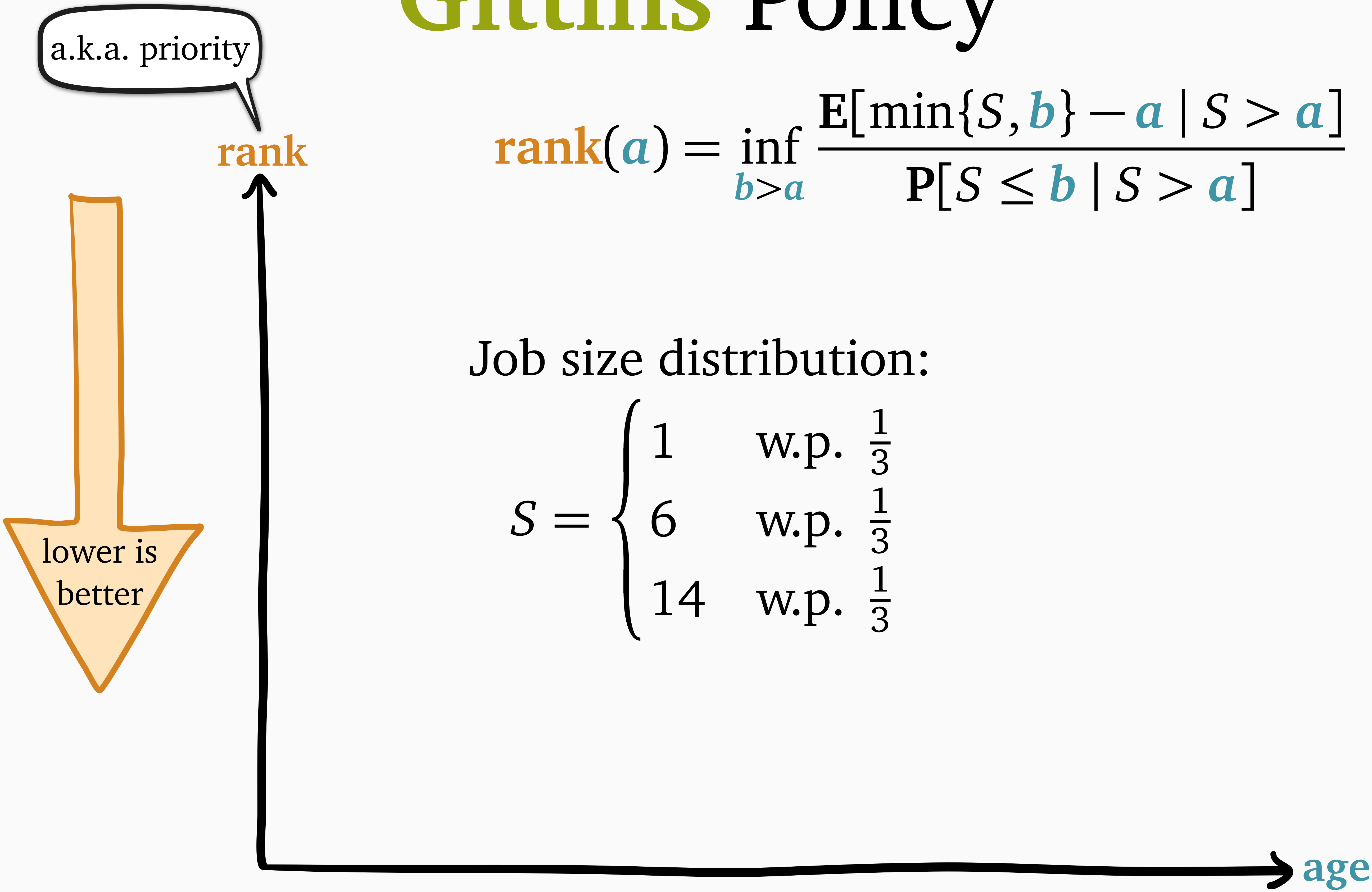


Gittins Policy

$$\text{rank}(a) = \inf_{b>a} \frac{\mathbf{E}[\min\{S, b\} - a \mid S > a]}{\mathbf{P}[S \le b \mid S > a]}$$

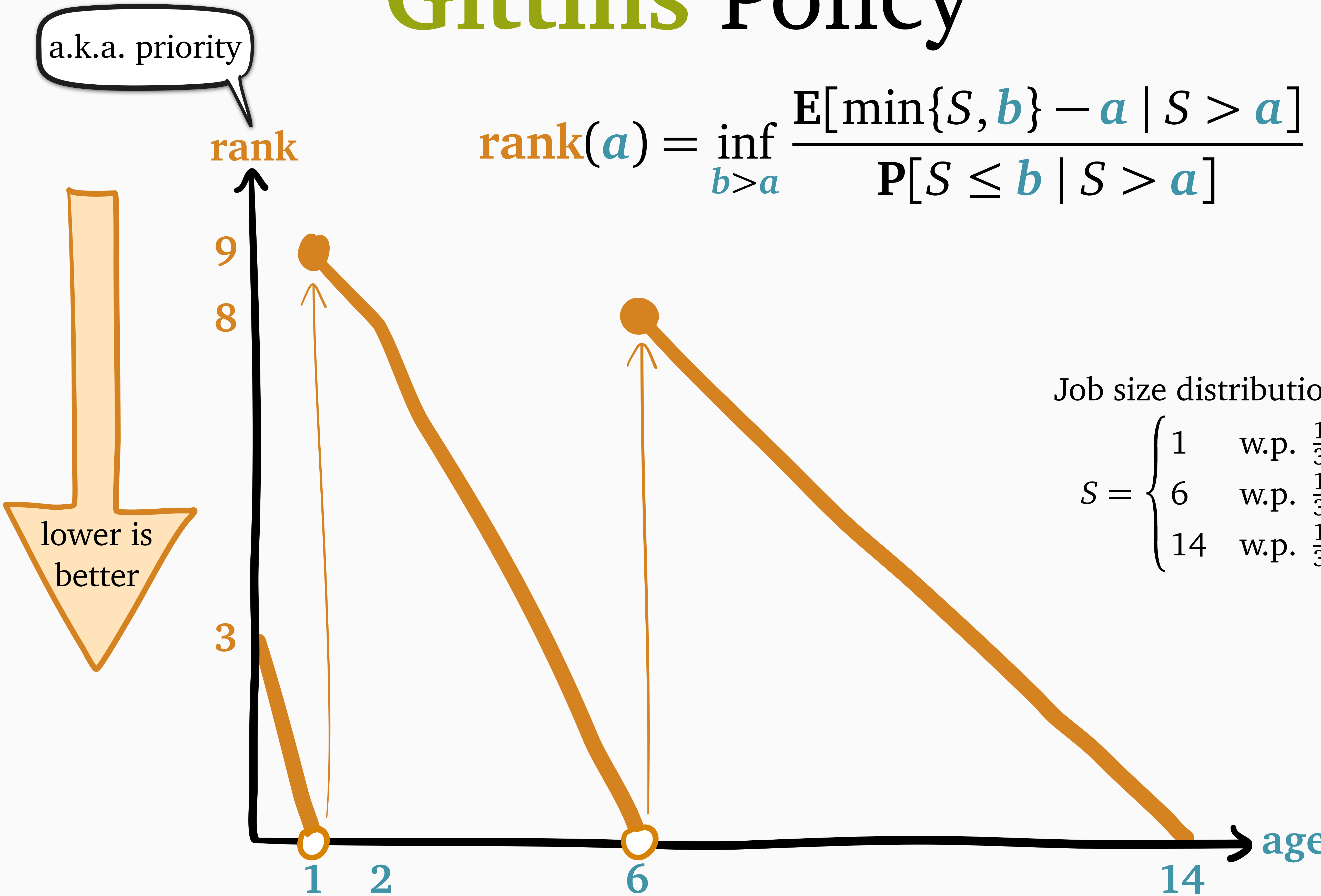
Job size distribution:

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



Gittins Policy

$$\text{rank}(a) = \inf_{b>a} \frac{\mathbf{E}[\min\{S, b\} - a \mid S > a]}{\mathbf{P}[S \le b \mid S > a]}$$



Job size distribution:

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

Optimal Policies

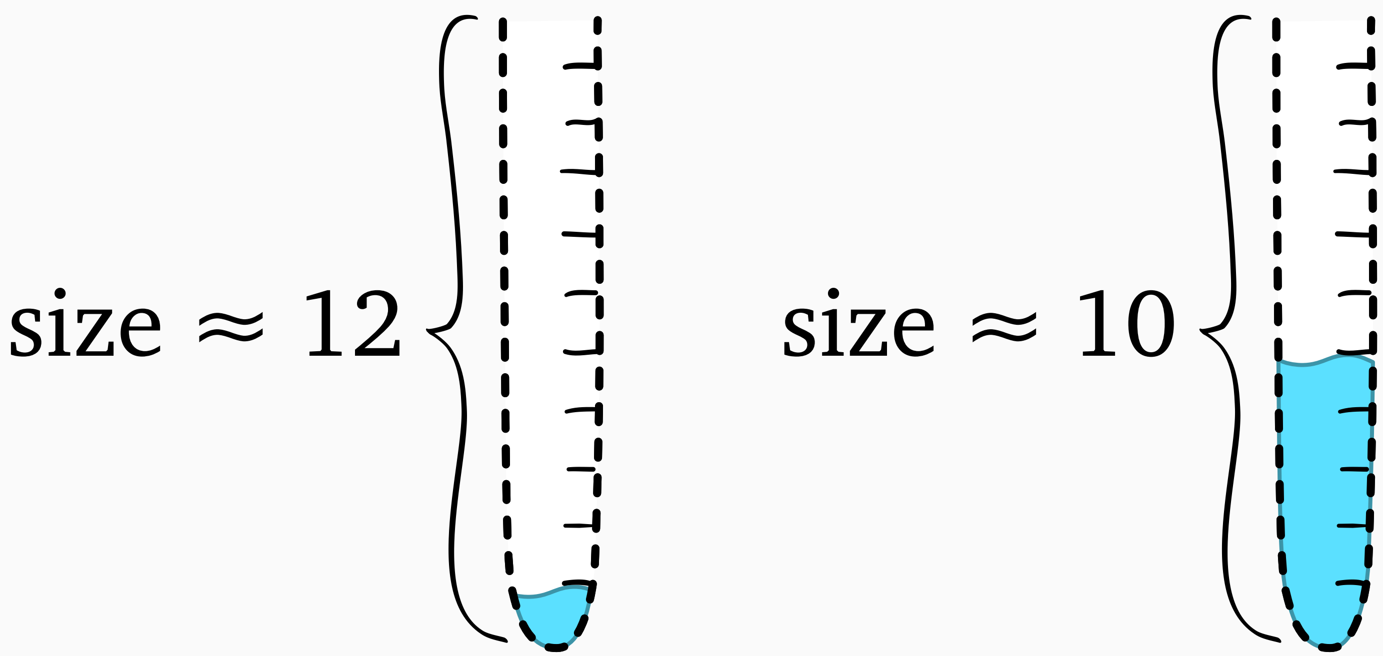
	Known Sizes	Unknown Sizes
M/G/1	**SRPT**	**Gittins**

Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**		**Gittins**

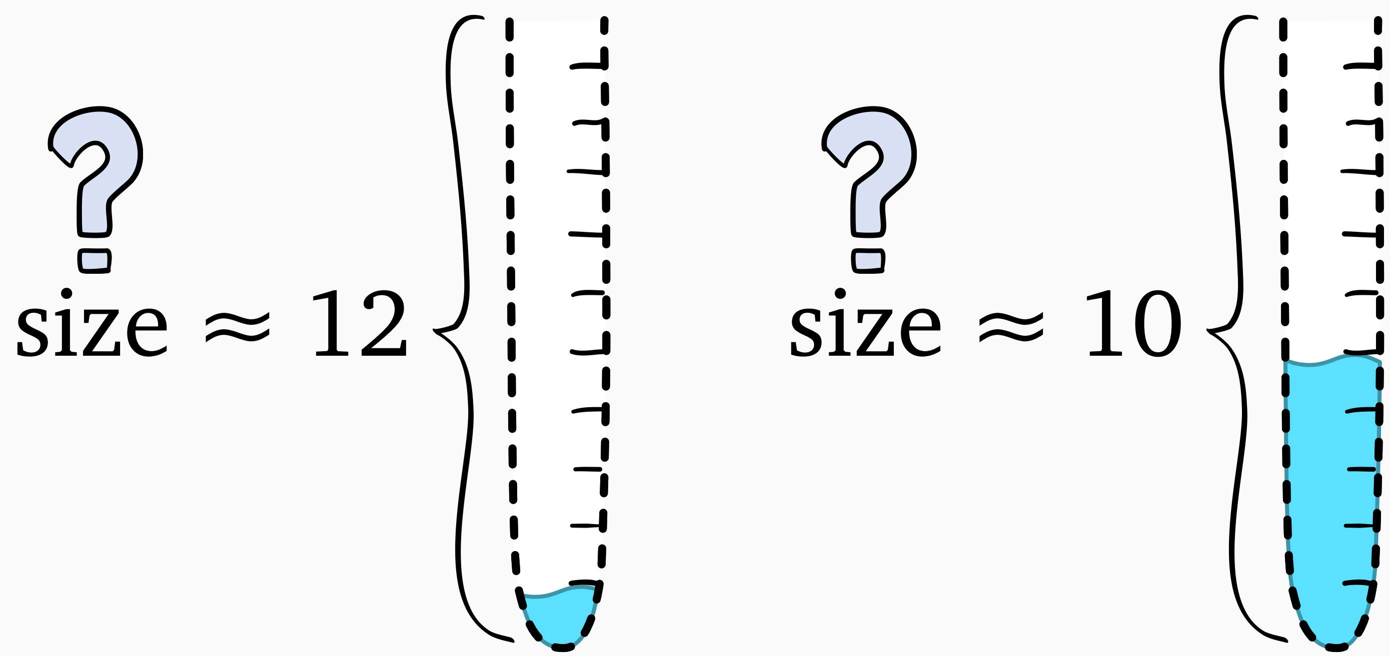
Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**		**Gittins**



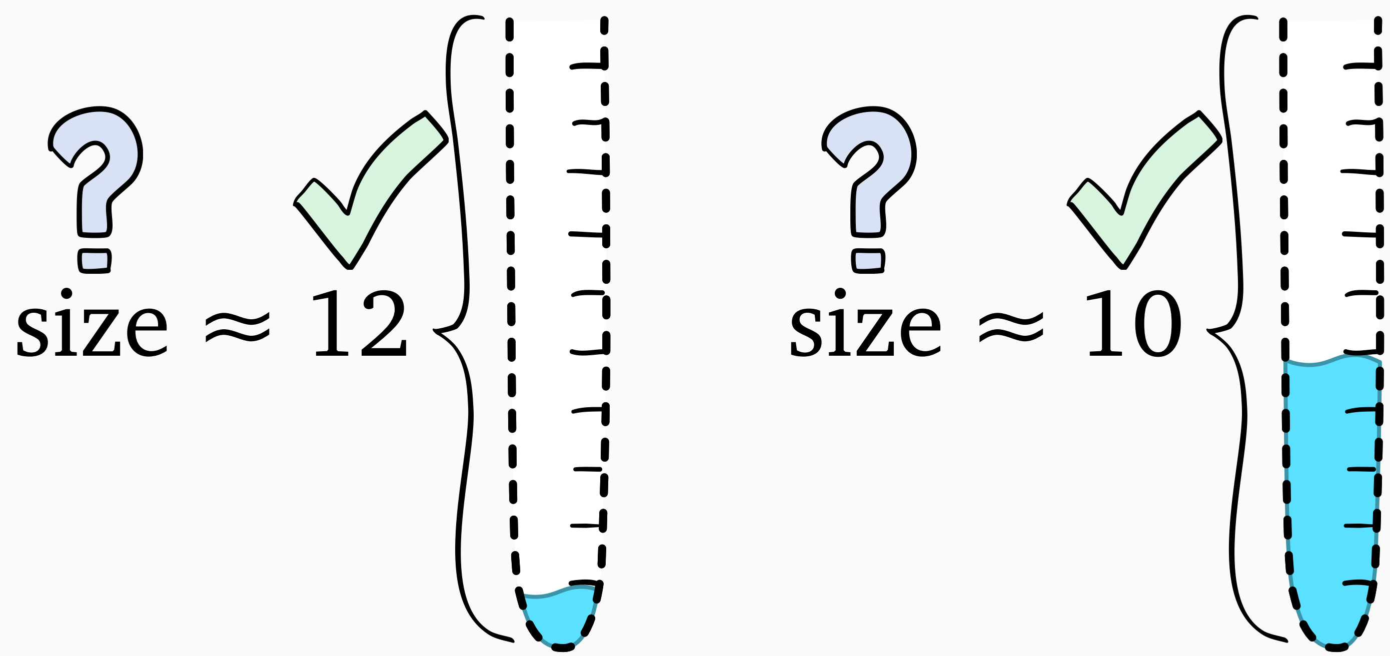
Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**		**Gittins**



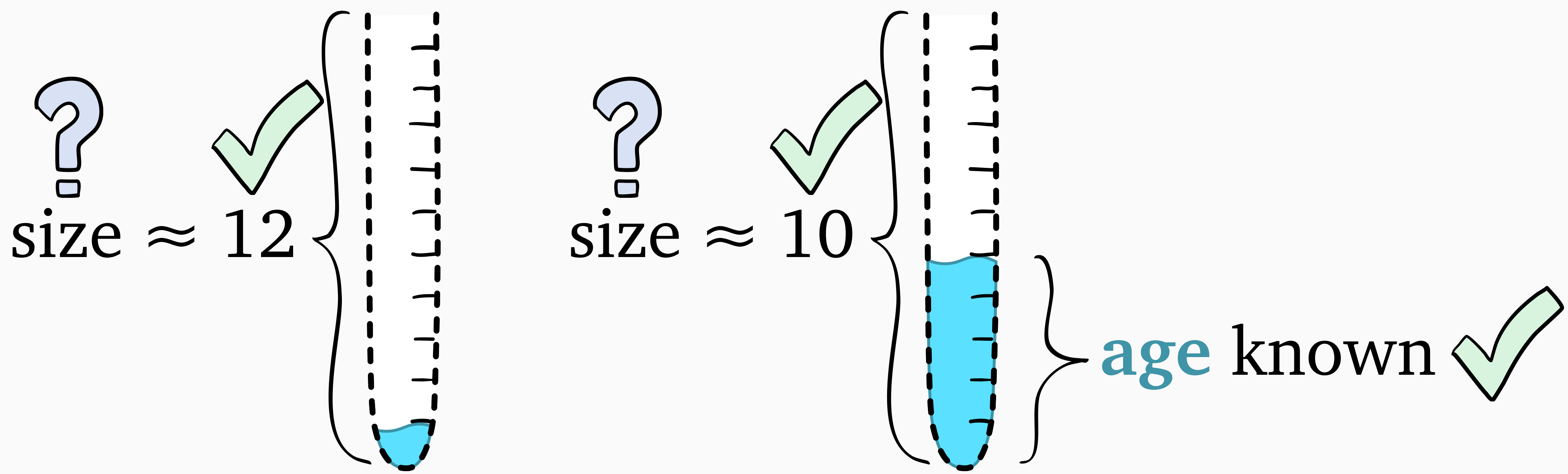
Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	SRPT		Gittins



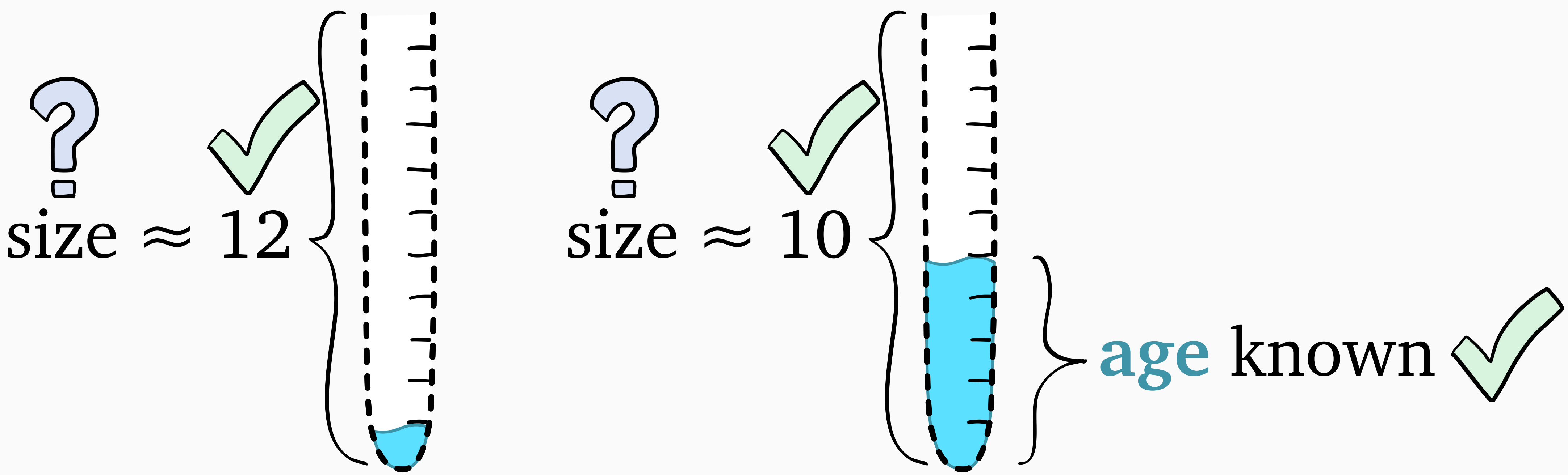
Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	SRPT		Gittins



Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**



Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**

huge variety of scenarios

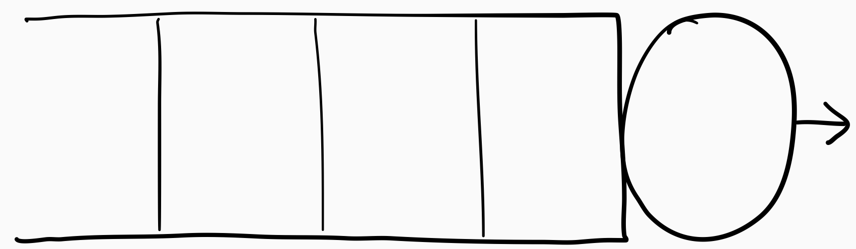
Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	SRPT	Gittins	Gittins

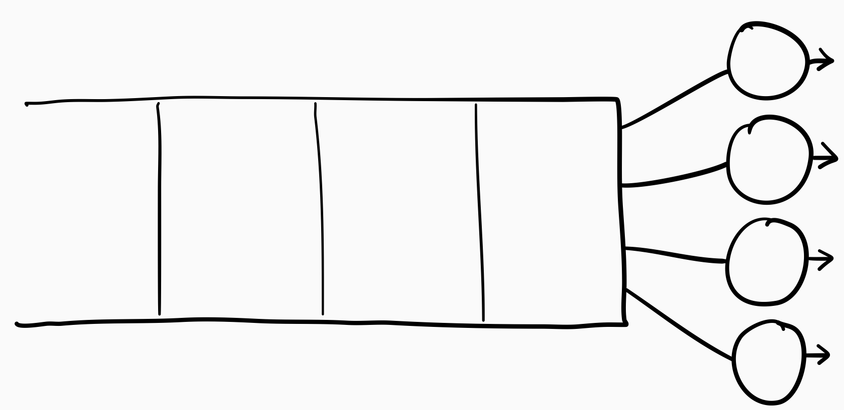
huge variety of scenarios

special case of Gittins: rank = remaining size

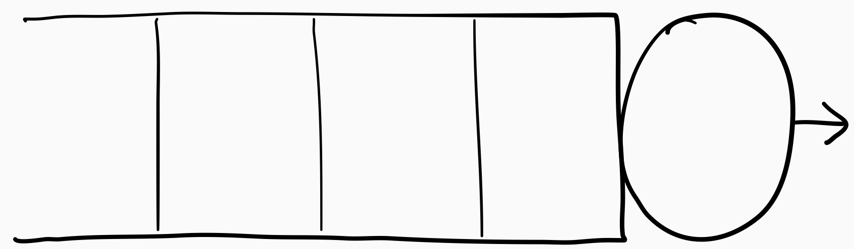
First: background on *single-server* scheduling



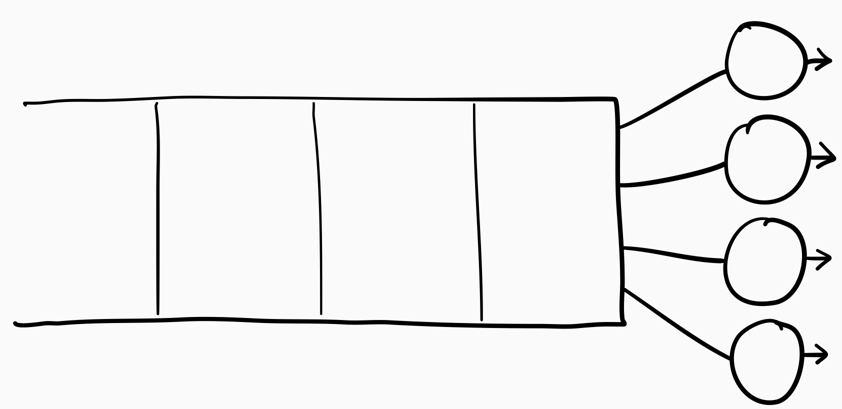
This talk: near-optimal *multiserver* scheduling



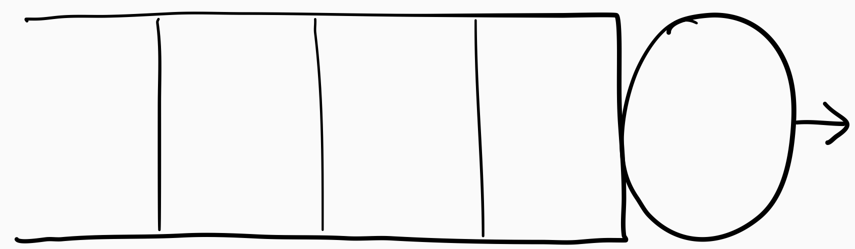
First: background on *single-server* scheduling



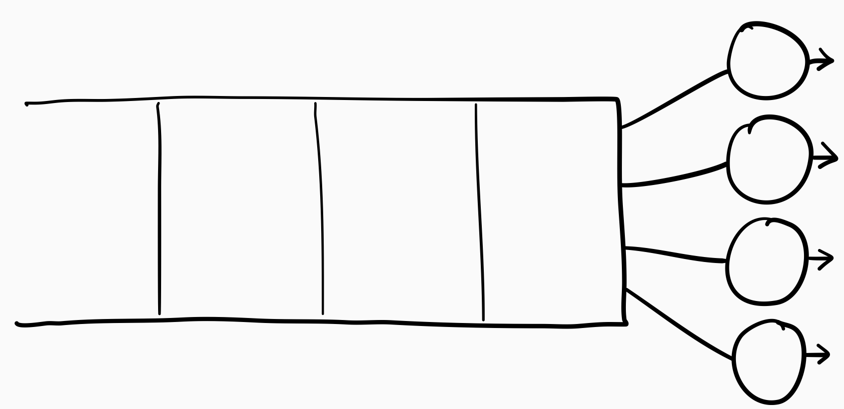
This talk: near-optimal *multiserver* scheduling



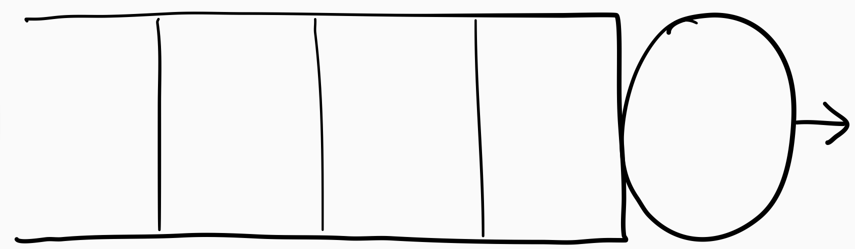
First: background on *single-server* scheduling



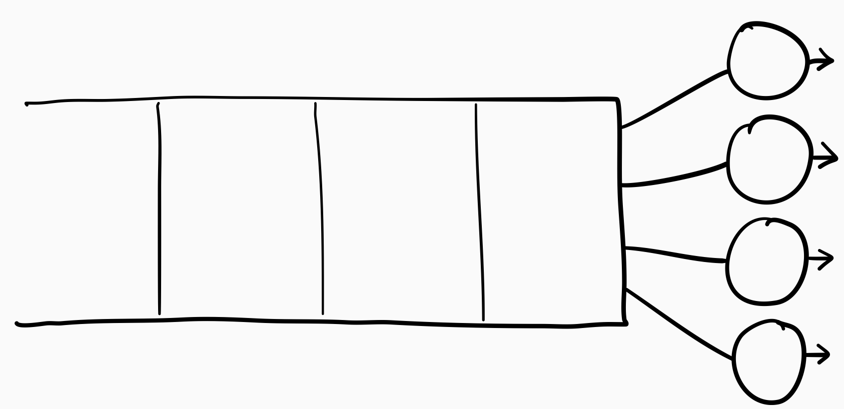
This talk: near-optimal *multiserver* scheduling

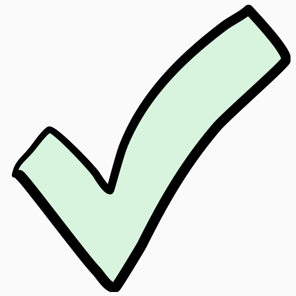


First: background on *single-server* scheduling

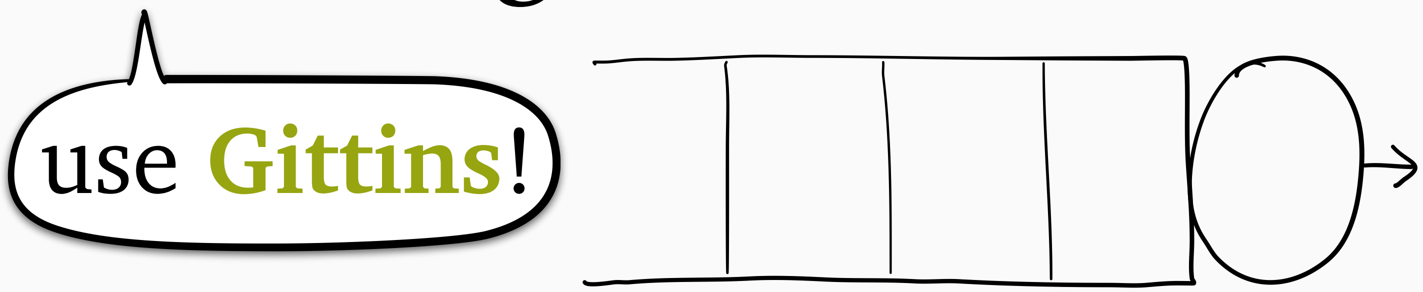


This talk: near-optimal *multiserver* scheduling

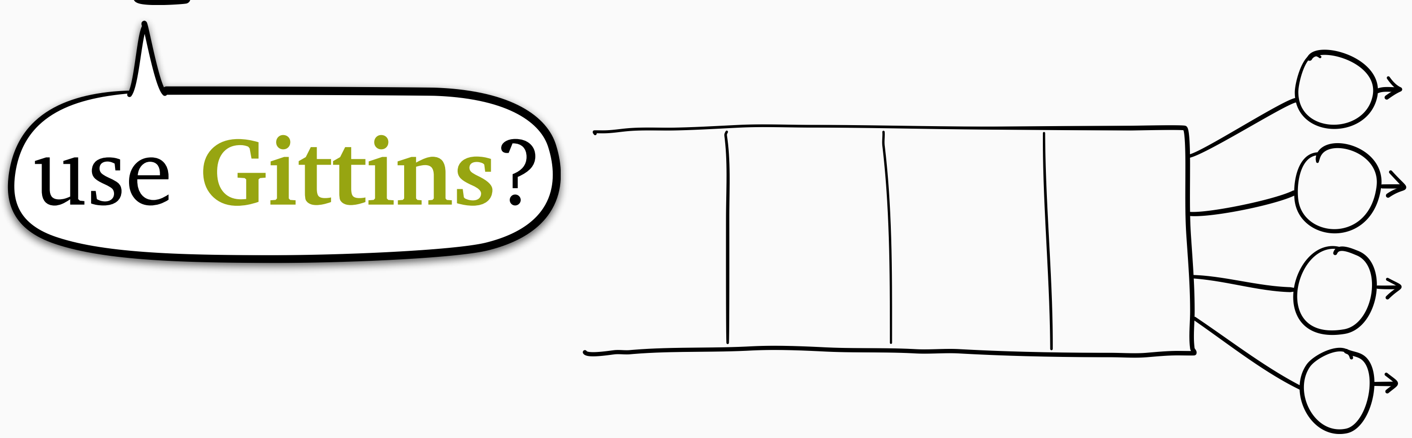




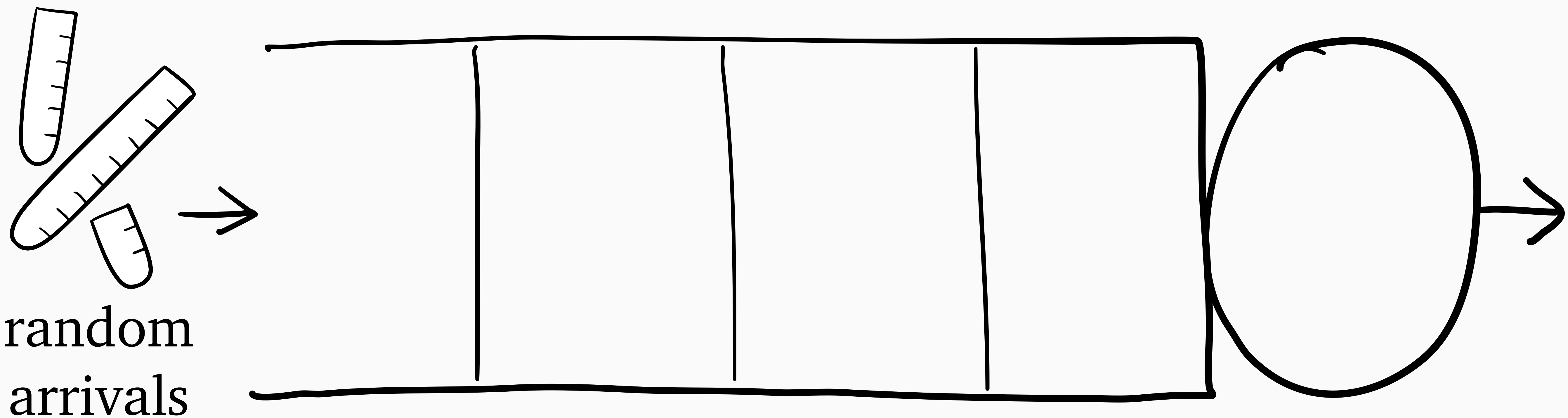
First: background on *single-server* scheduling



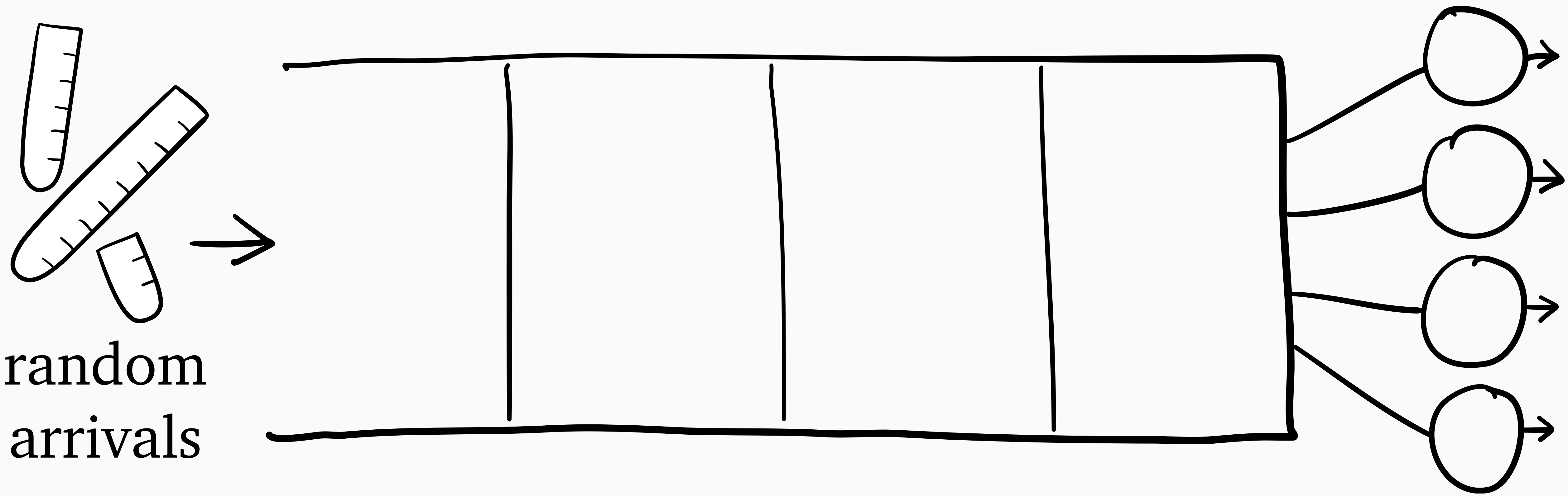
This talk: near-optimal *multiserver* scheduling



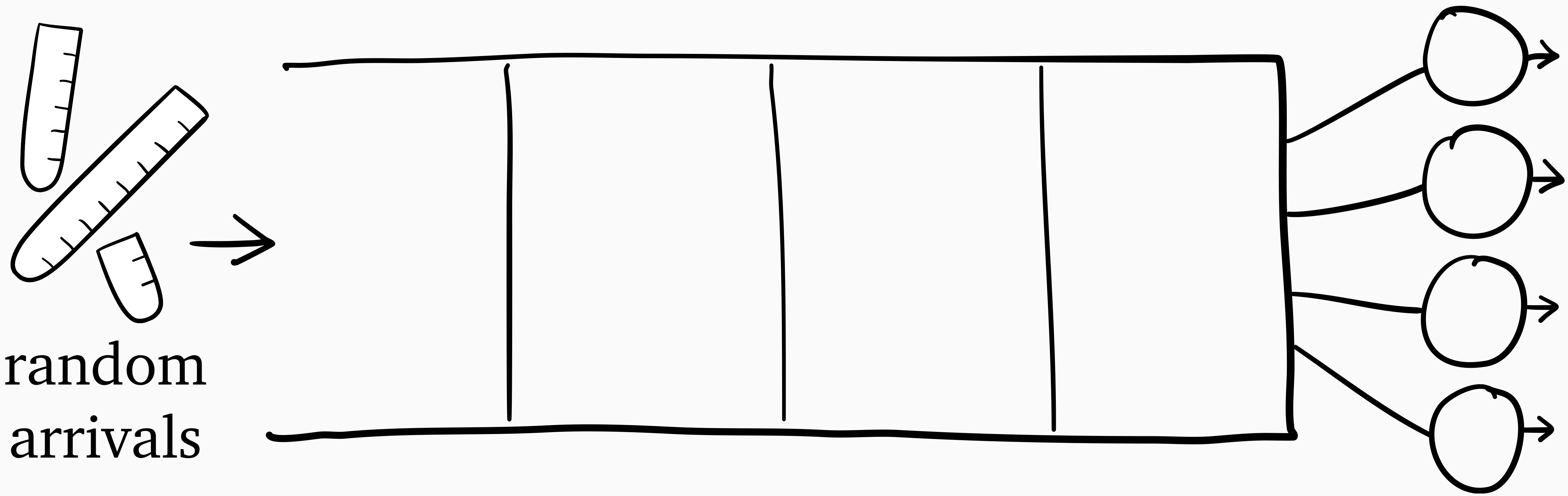
M/G/k Queue



M/G/k Queue

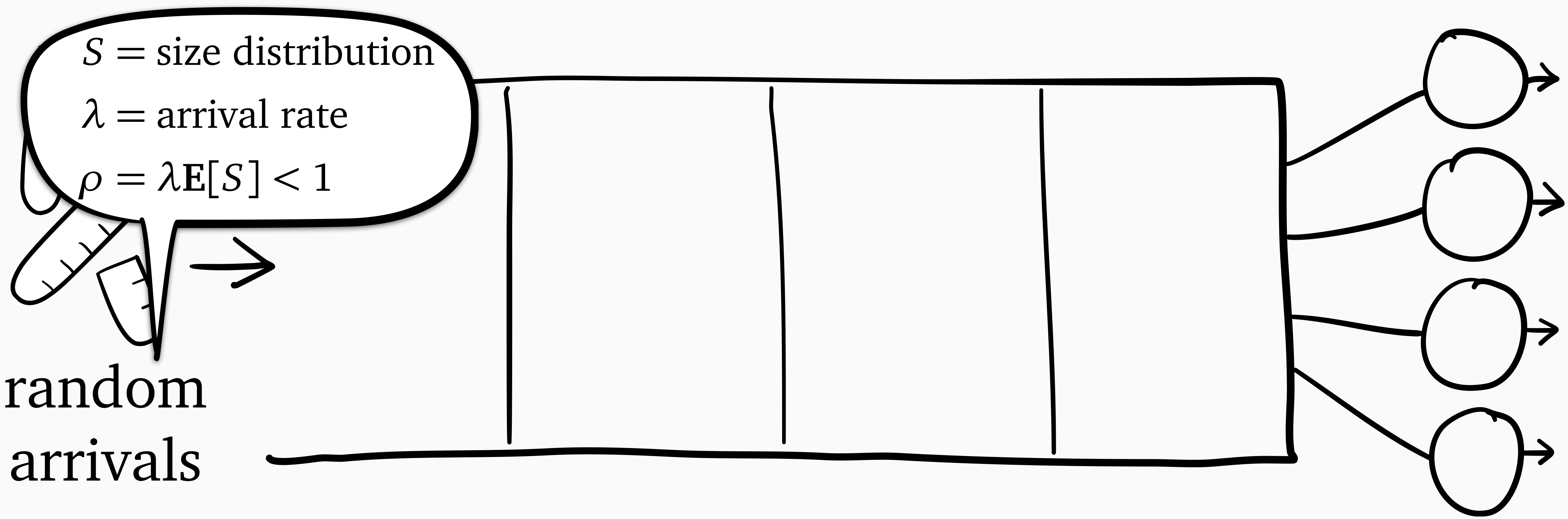


M/G/k Queue



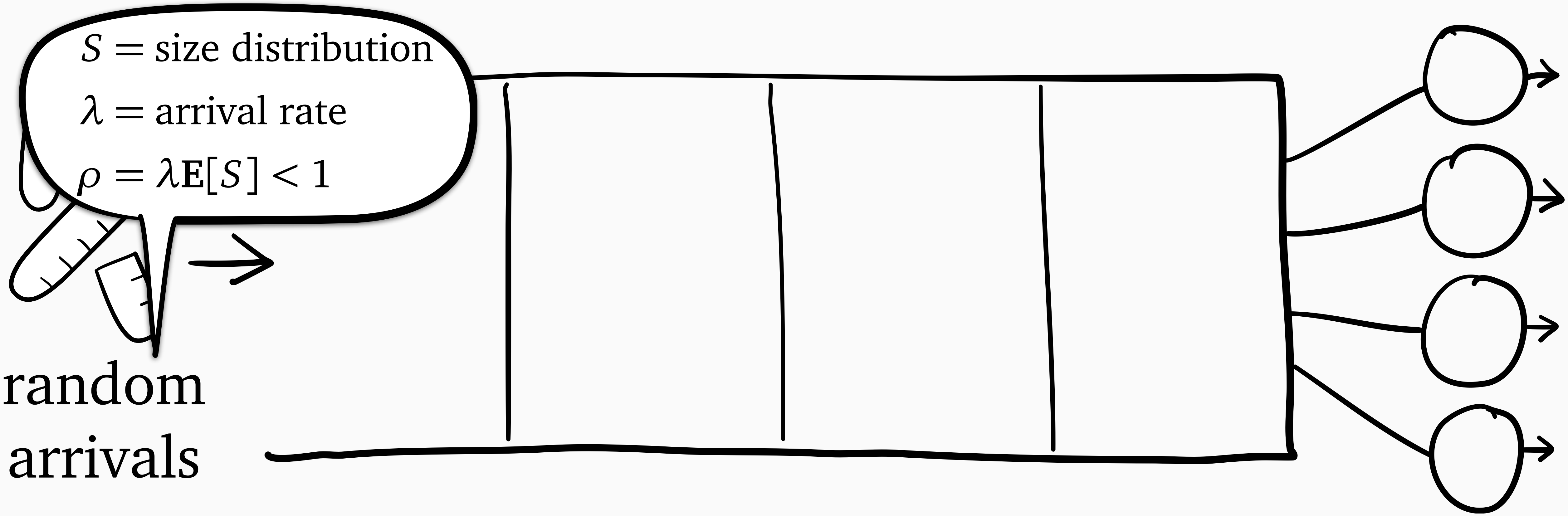
k servers, each speed $1/k$

M/G/k Queue

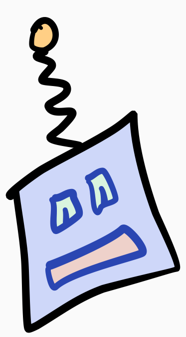


k servers, each speed $1/k$

M/G/k Queue

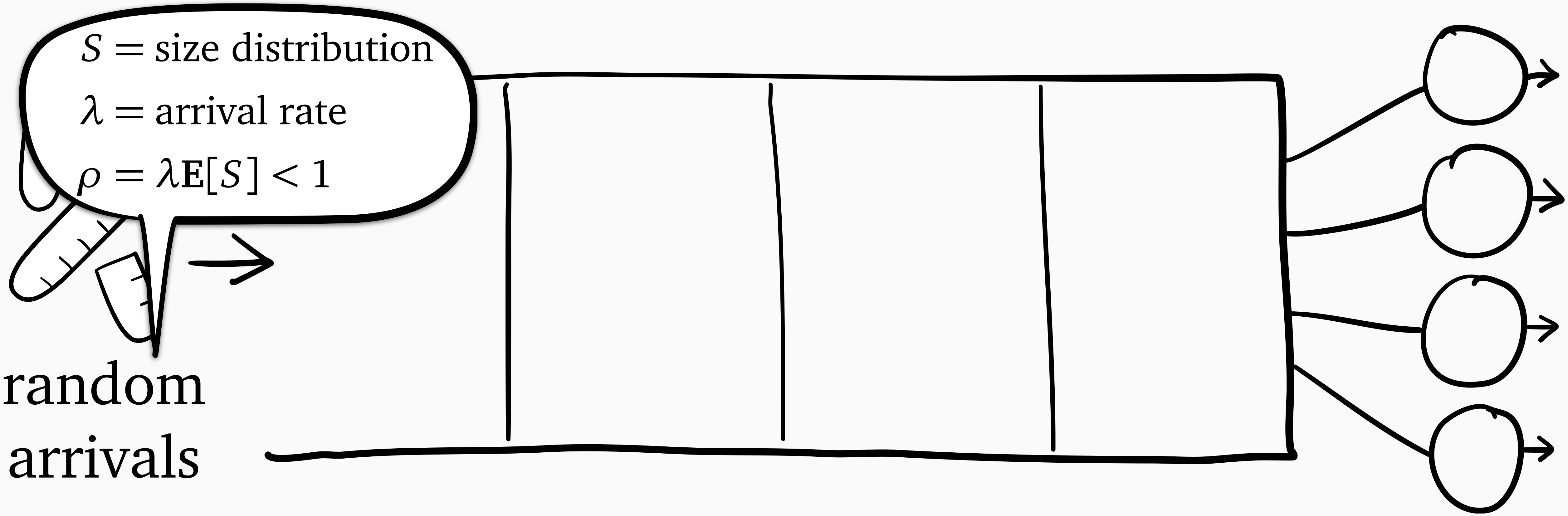


k servers, each speed $1/k$



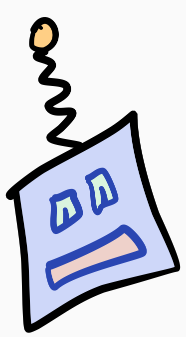
Scheduling policy:
picks which k jobs to serve

M/G/k Queue



k servers, each speed $1/k$

Multiserver Gittins: serves the k jobs with the k lowest **ranks**



Scheduling policy: picks which k jobs to serve

M/G/1 vs. M/G/k

$$\mathbf{E}[T_1^{\min}] \leq \mathbf{E}[T_k^{\min}]$$

M/G/1 vs. M/G/k

$$\mathbf{E}[T_1^{\text{Gittins}}] = \mathbf{E}[T_1^{\text{min}}] \le \mathbf{E}[T_k^{\text{min}}]$$

M/G/1 vs. M/G/k

$$\mathbf{E}[T_1^{\text{Gittins}}] = \mathbf{E}[T_1^{\min}] \le \mathbf{E}[T_k^{\min}] \le \mathbf{E}[T_k^{\text{Gittins}}]$$

M/G/1 vs. M/G/k

$$\mathbf{E}[T_1^{\text{Gittins}}] = \mathbf{E}[T_1^{\text{min}}] \leq \mathbf{E}[T_k^{\text{min}}] \leq \mathbf{E}[T_k^{\text{Gittins}}]$$

Goal 1: "near-optimality" bound

$$\mathbf{E}[T_k^{\text{Gittins}}] \leq \mathbf{E}[T_1^{\text{Gittins}}] + \text{something "small"}$$

M/G/1 vs. M/G/k

$$\mathbf{E}[T_1^{\text{Gittins}}] = \mathbf{E}[T_1^{\min}] \leq \mathbf{E}[T_k^{\min}] \leq \mathbf{E}[T_k^{\text{Gittins}}]$$

Goal 1: "near-optimality" bound

$$\mathbf{E}[T_k^{\text{Gittins}}] \leq \mathbf{E}[T_1^{\text{Gittins}}] + \text{something "small"}$$

Goal 2: heavy-traffic optimality

$$\lim_{\rho \to 1} \frac{\mathbf{E}[T_k^{\text{Gittins}}]}{\mathbf{E}[T_1^{\text{Gittins}}]} = 1$$

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**
M/G/k (prior work)			

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**
M/G/k (prior work)			

Generalize M/G/1
$\mathbf{E}[T]$ analysis

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**
M/G/k (prior work)	**SRPT** Grosof et al. (2018)		

Generalize M/G/1
$\mathbf{E}[T]$ analysis

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**
M/G/k (prior work)	**SRPT** Grosof et al. (2018)		

Generalize M/G/1 $\mathbf{E}[T]$ analysis

$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**
M/G/k (prior work)	**SRPT** Grosof et al. (2018)		

Generalize M/G/1 $\mathbf{E}[T]$ analysis

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**
M/G/k (prior work)	**SRPT** Grosof et al. (2018)	?	?

Generalize M/G/1 $\mathbf{E}[T]$ analysis

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**
M/G/k (prior work)	**SRPT** Grosof et al. (2018)	?	?

Generalize M/G/1 $\mathbf{E}[T]$ analysis

Problem: existing M/G/1-to-M/G/k strategy uses **worst-case** techniques

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	SRPT	Gittins	Gittins
M/G/k (prior work)	SRPT Grosof et al. (2018)	?	?

Generalize M/G/1 $\mathbf{E}[T]$ analysis

Problem: existing M/G/1-to-M/G/k strategy uses **worst-case** techniques

Near-Optimal Policies

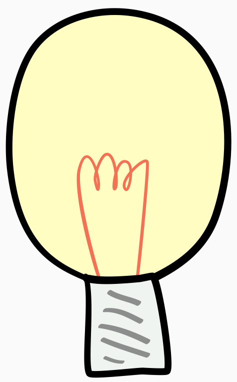
	Known Sizes	Partial Info	Unknown Sizes
M/G/1	SRPT	Gittins	Gittins
M/G/k (prior work)	SRPT Grosof et al. (2018)	?	?

Generalize M/G/1 **E**[T] analysis

Problem: existing M/G/1-to-M/G/k strategy uses **worst-case** techniques

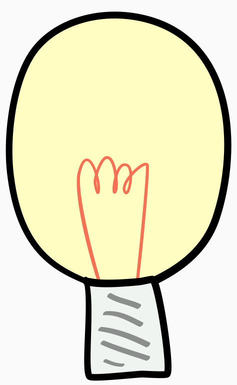
Our contributions:

Our contributions:



Introduce *new techniques* for analyzing $\mathbf{E}[T]$ in the M/G/k

Our contributions:



Introduce *new techniques* for analyzing $\mathbf{E}[T]$ in the M/G/k

Prove that **Gittins** has *near-optimal* $\mathbf{E}[T]$ in the M/G/k

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**
M/G/k (prior work)	**SRPT**	?	?
M/G/k (**new result**)			

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	**SRPT**	**Gittins**	**Gittins**
M/G/k (prior work)	**SRPT**	?	?
M/G/k (**new result**)	**SRPT**	**Gittins**	**Gittins**

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	SRPT	Gittins	Gittins
M/G/k (prior work)	SRPT	?	?
M/G/k (new result)	SRPT	Gittins	Gittins

$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

Near-Optimal Policies

	Known Sizes	Partial Info	Unknown Sizes
M/G/1	SRPT		Gittins
M/G/k (prior work)	SRPT	?	?
M/G/k (new result)	SRPT	Gittins	Gittins

$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

Theorem: under **Gittins**,

$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

Theorem: under **Gittins**,

$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

Theorem: under **Gittins**,

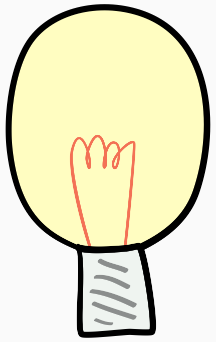
$$\lim_{\rho \to 1} \frac{\mathbf{E}[T_k]}{\mathbf{E}[T_1]} = 1 \qquad \text{if } \mathbf{E}[S^2 (\log S)^+] < \infty$$

Theorem: under **Gittins**,

$$\mathbf{E}[T_k] \le \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

Theorem: under **Gittins**,

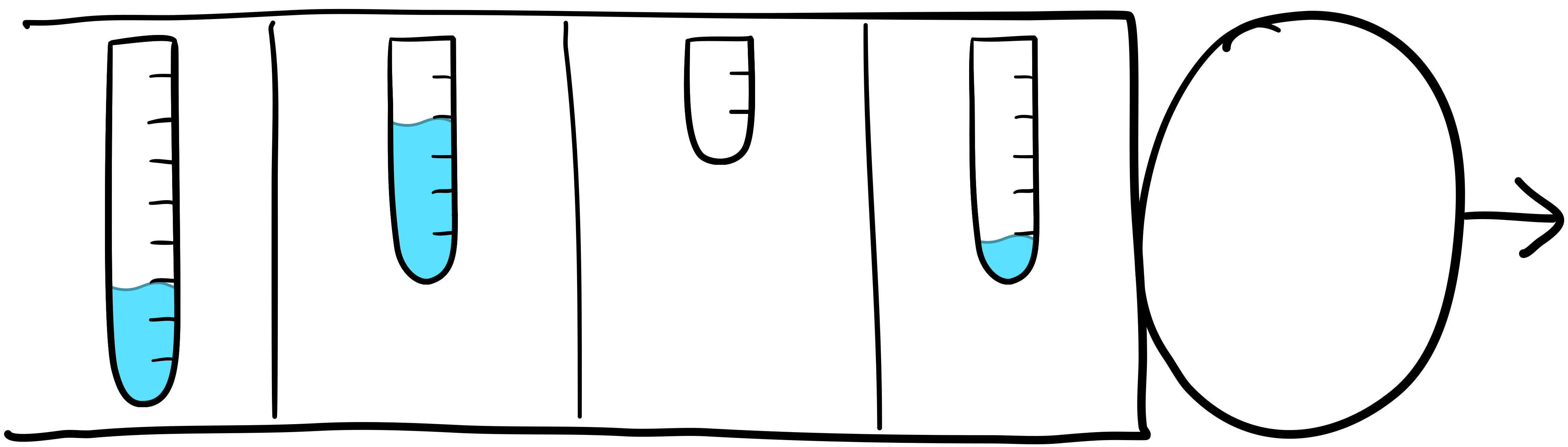
$$\lim_{\rho \to 1} \frac{\mathbf{E}[T_k]}{\mathbf{E}[T_1]} = 1 \qquad \text{if } \mathbf{E}[S^2(\log S)^+] < \infty$$



New concept: *r*-work

What is *r*-Work? (SRPT)

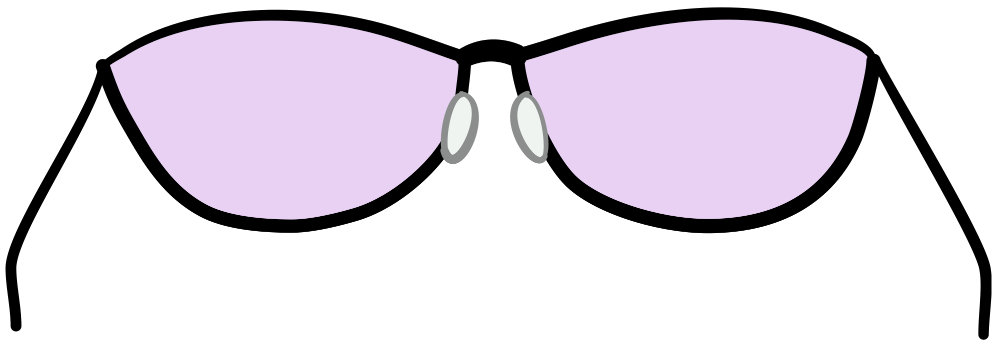
W = work = total remaining size of all jobs

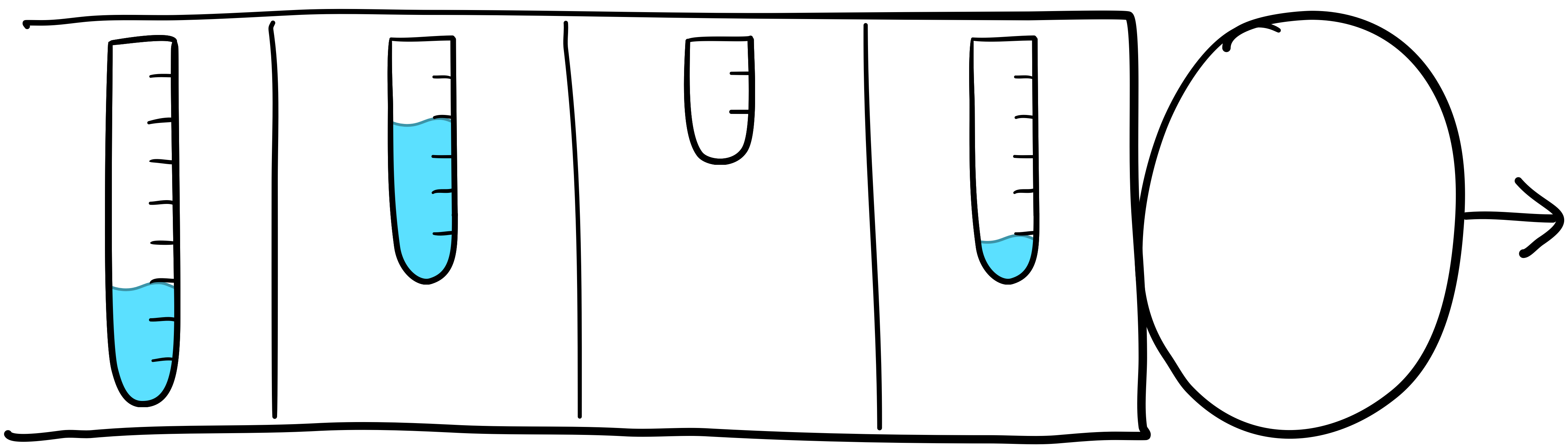


What is *r*-Work? (SRPT)

W = work = total remaining size of all jobs

$W(r)$ = *r*-work = total remaining size of all jobs that have remaining size $\leq r$

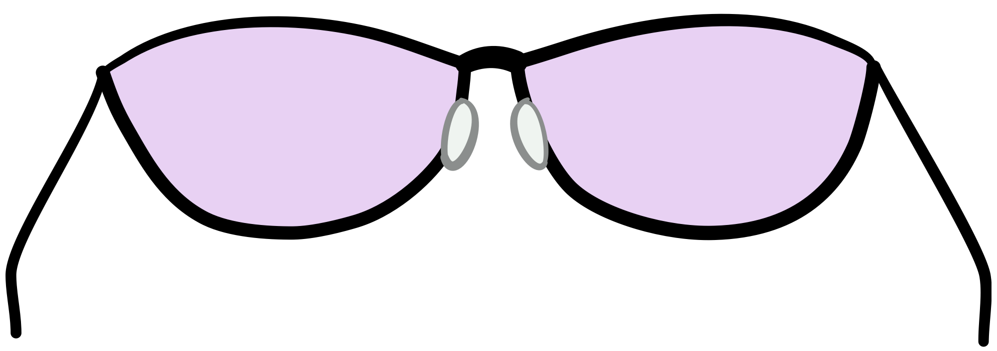


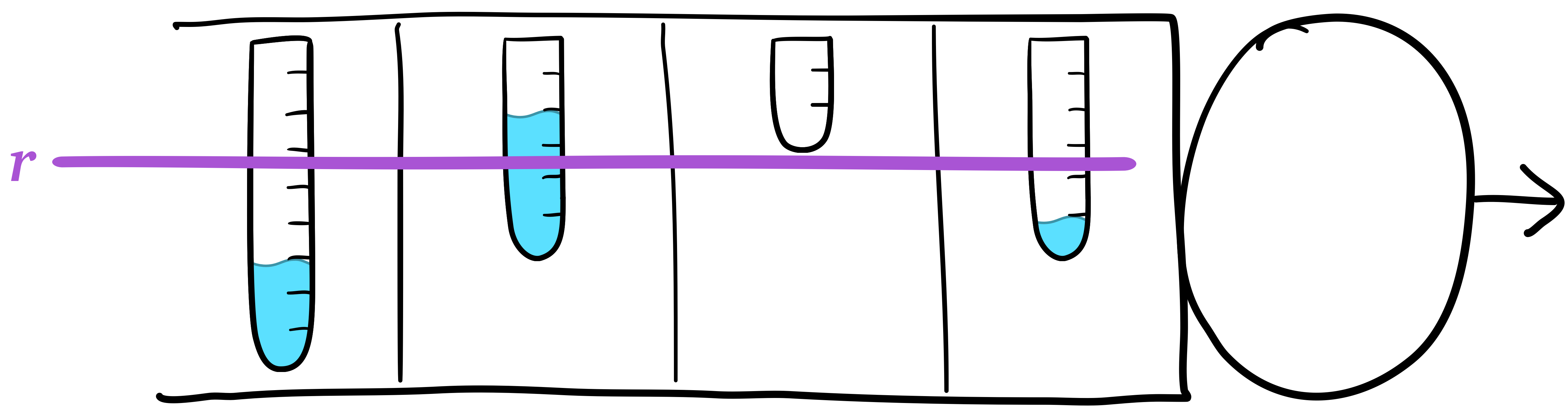


What is *r*-Work? (SRPT)

W = work = total remaining size of all jobs

$W(r)$ = *r*-work = total remaining size of all jobs that have remaining size $\leq r$

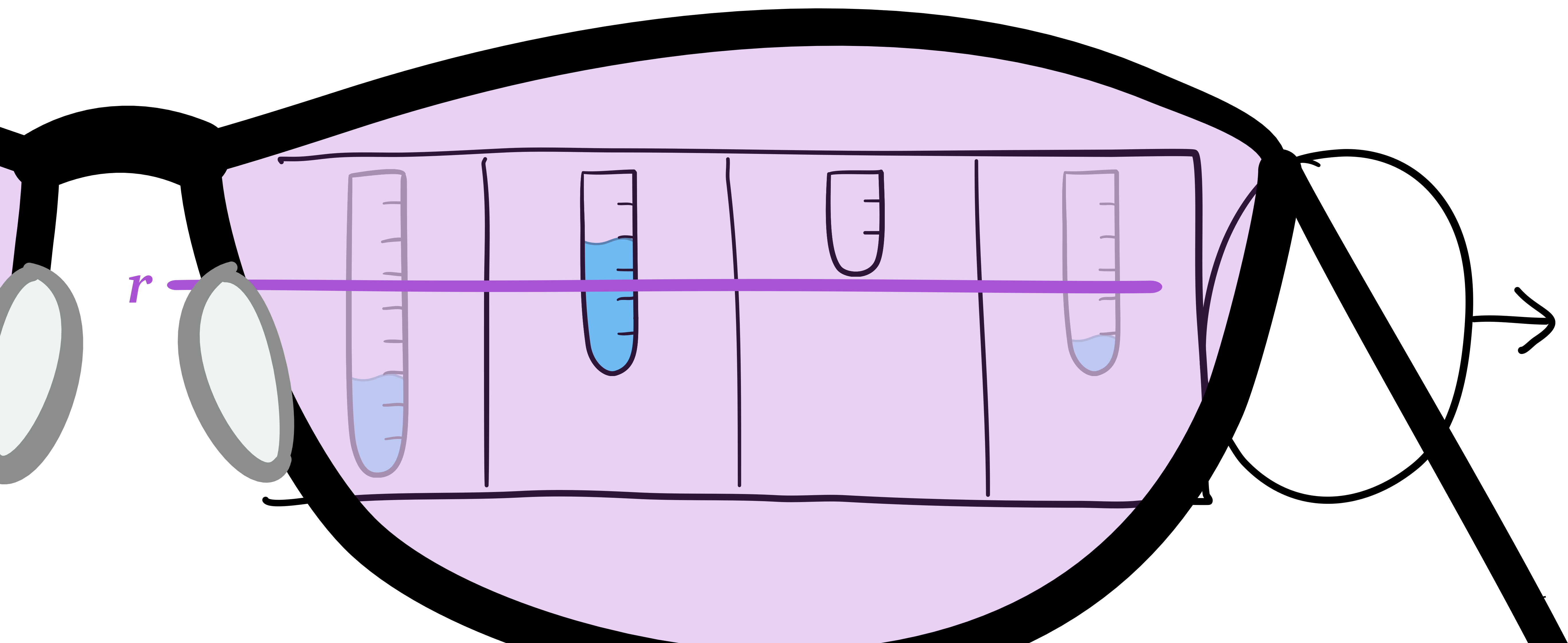




What is *r*-Work? (SRPT)

W = work = total remaining size of all jobs

$W(r)$ = *r*-work = total remaining size of all jobs that have remaining size $\le r$



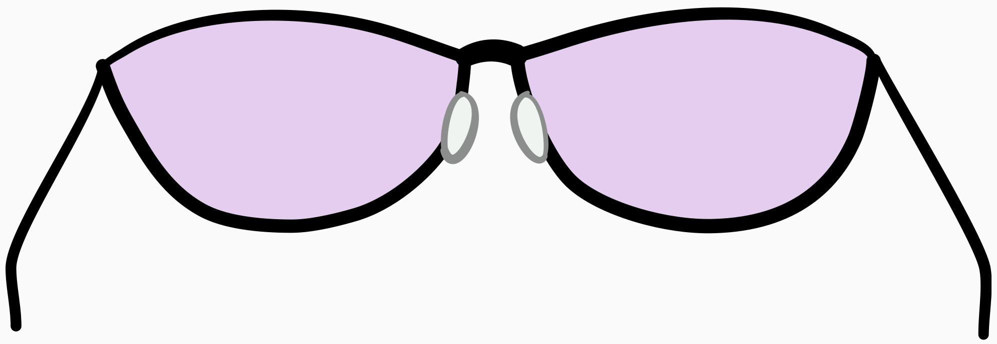
What is *r*-**Work**? (**Gittins**)

W = work = total remaining size of all jobs

What is *r*-**Work**? (**Gittins**)

W = work = total remaining size of all jobs

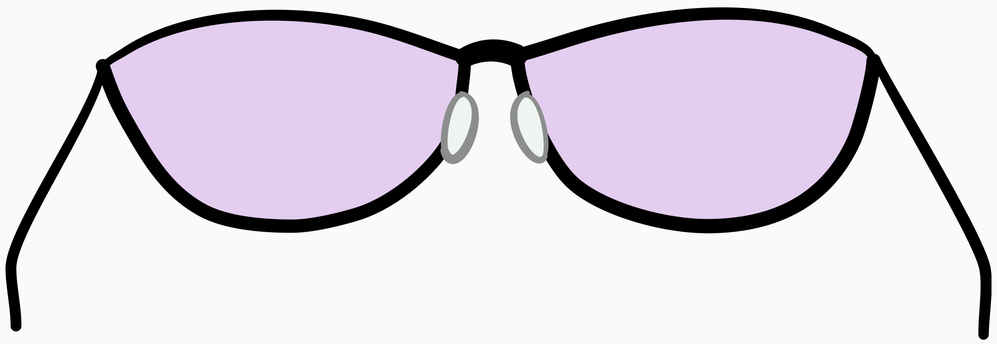
$W(r)$ = ***r*-work** = work "relevant" to **rank *r***



What is r-Work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = r-work = work "relevant" to rank r

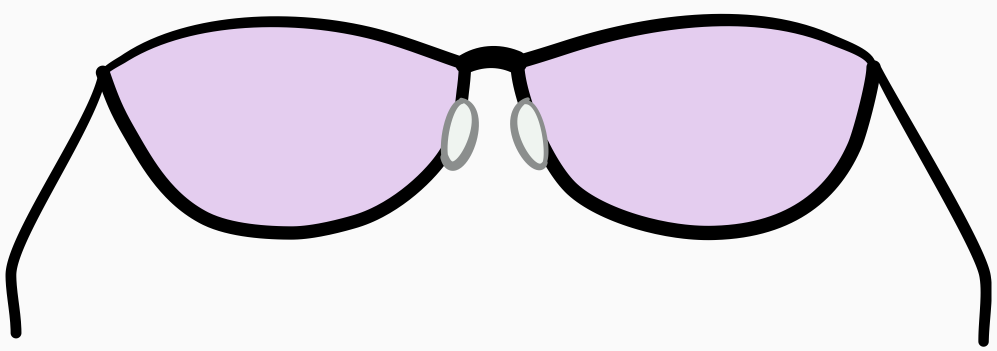


until job completes
or exceeds rank r

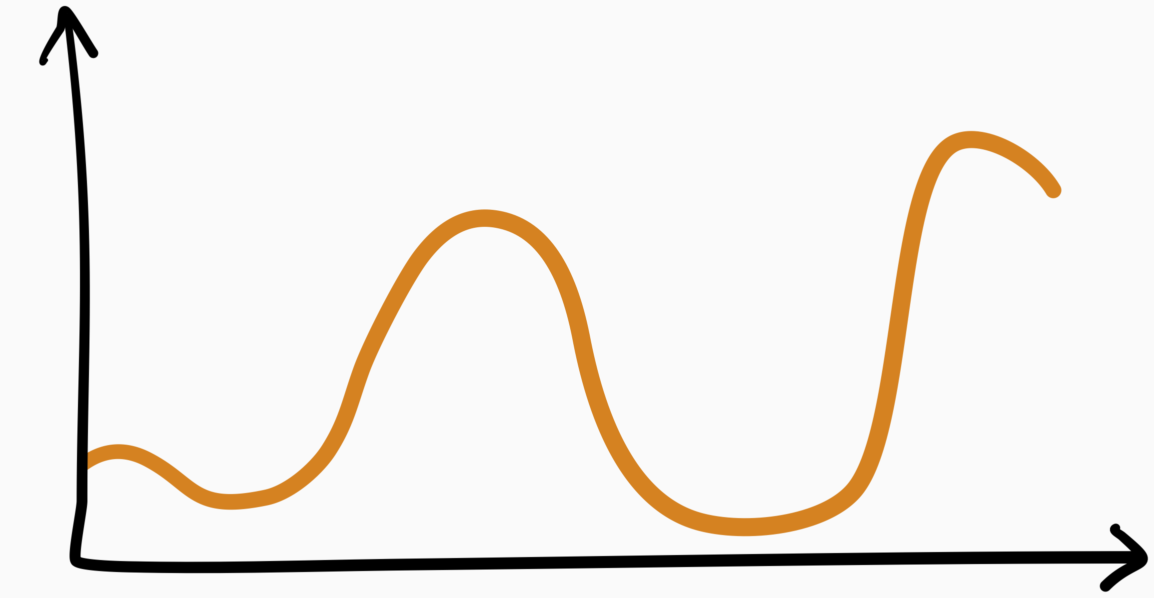
What is *r*-Work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = ***r*-work** = work "relevant" to **rank *r***



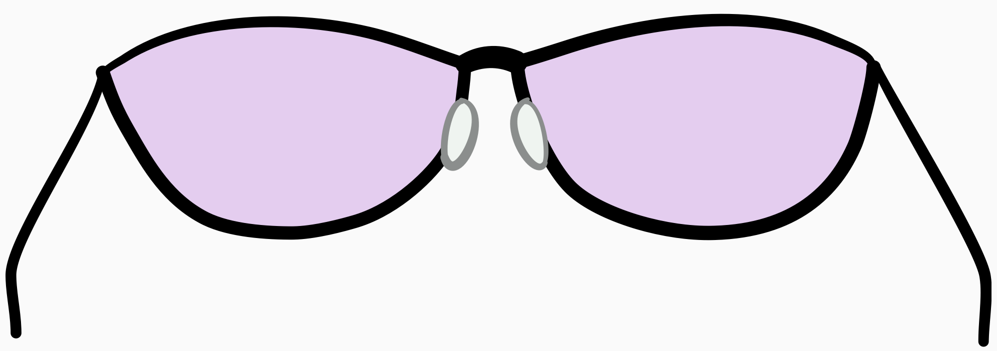
until job completes
or exceeds **rank *r***



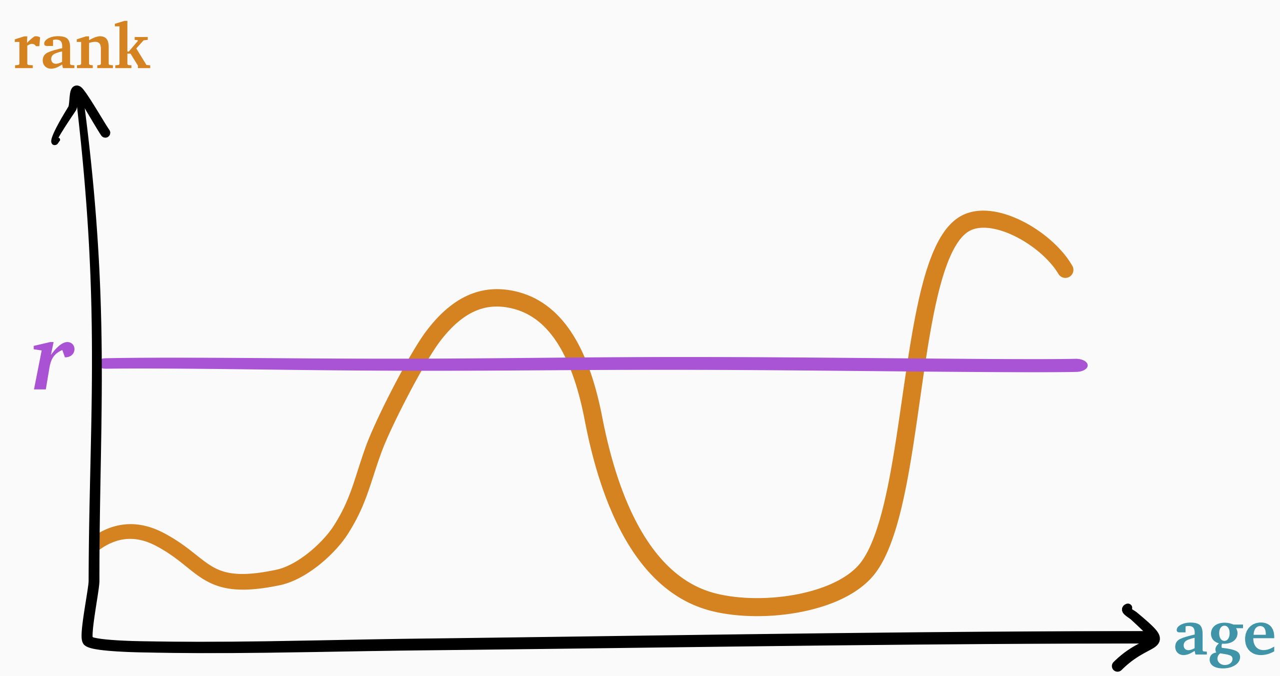
What is r-Work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = r-work = work "relevant" to rank r



until job completes
or exceeds rank r

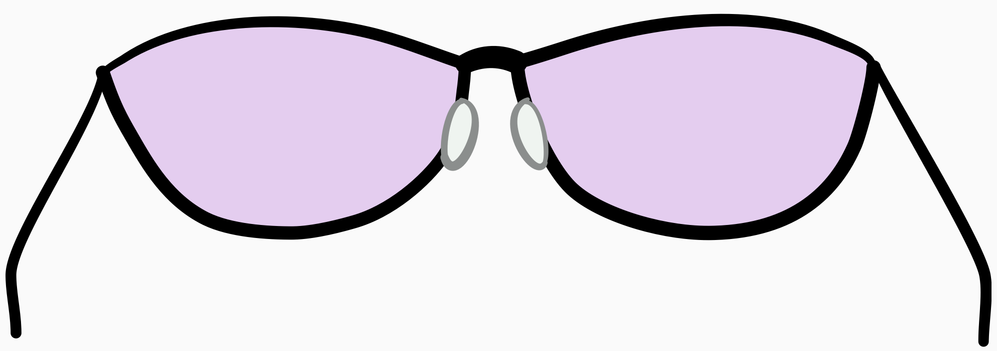


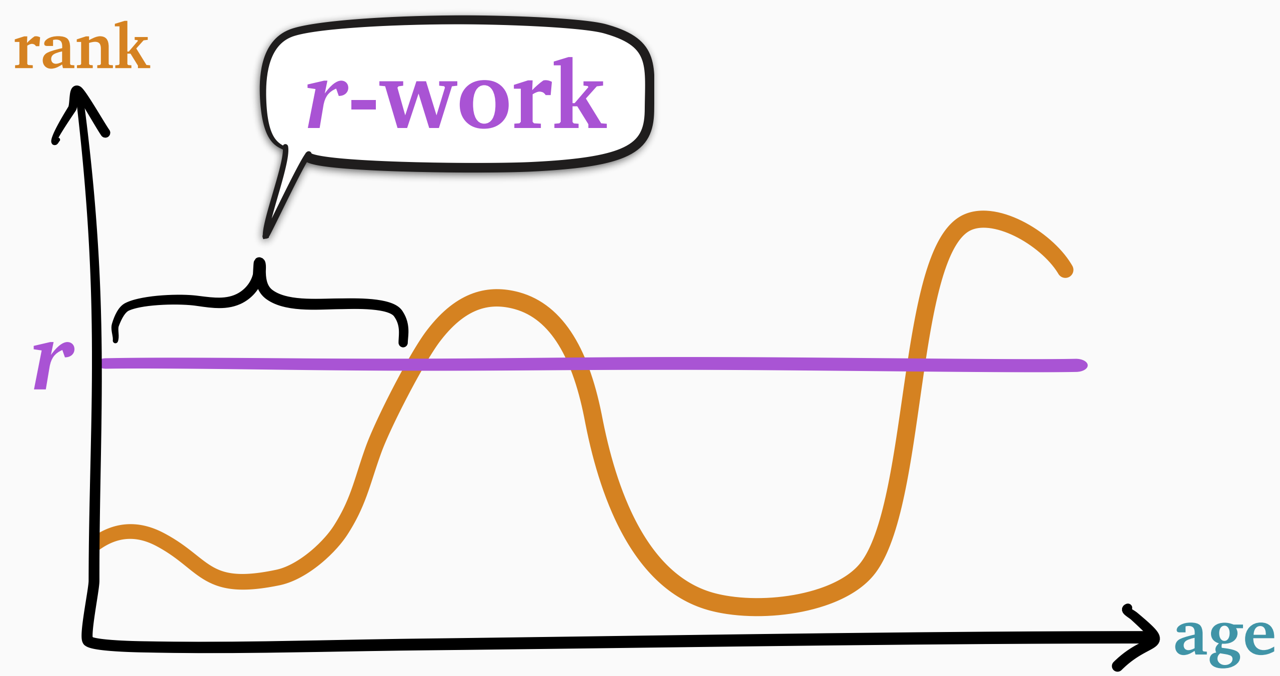
What is *r*-Work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = *r*-work = work "relevant" to rank *r*

until job completes or exceeds rank *r*

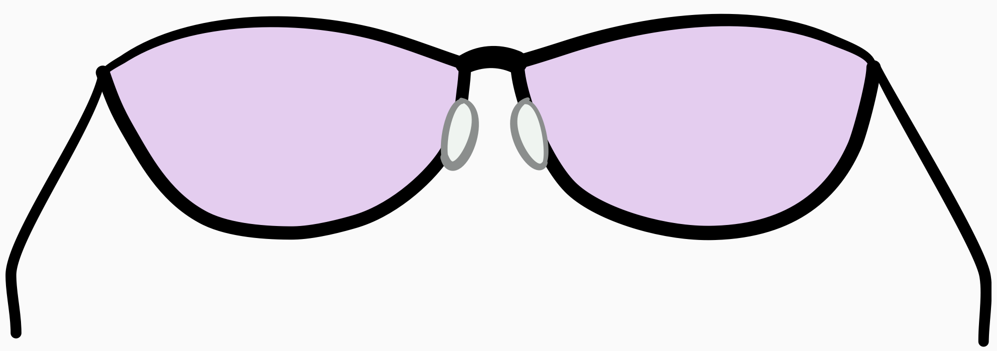




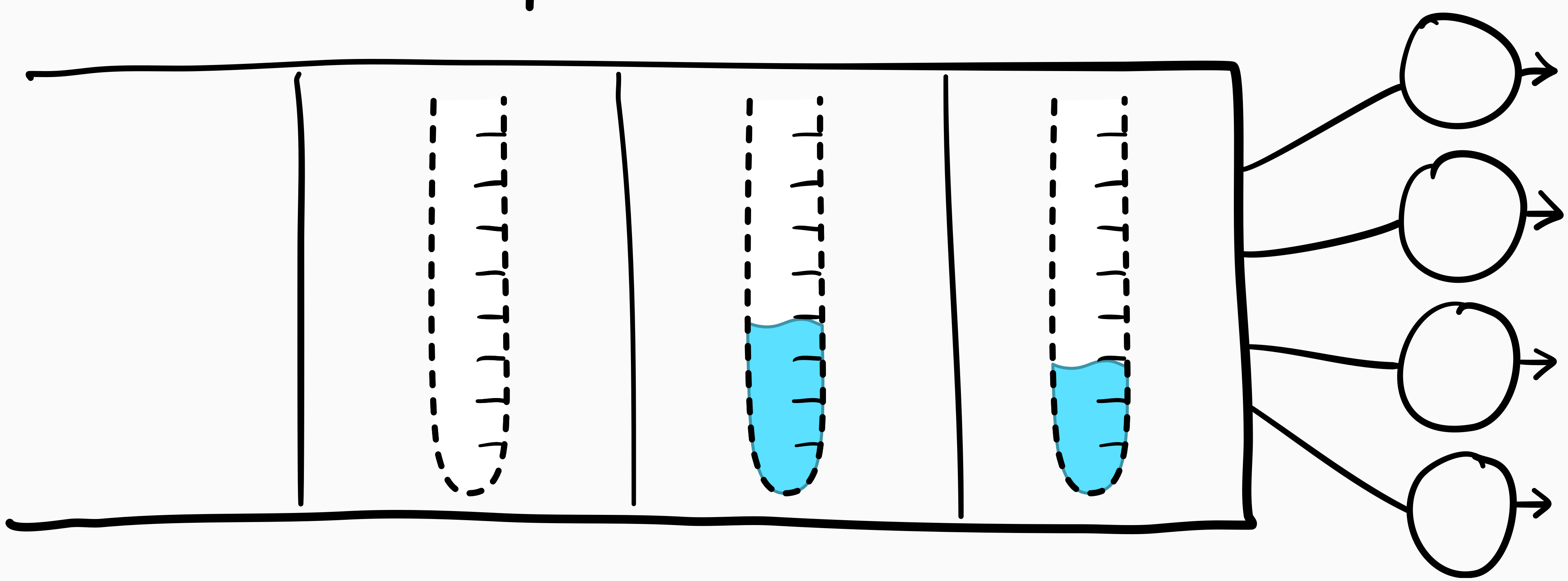
What is *r*-Work? (Gittins)

W = work = total remaining size of all jobs

$W(r)$ = *r*-work = work "relevant" to **rank** r



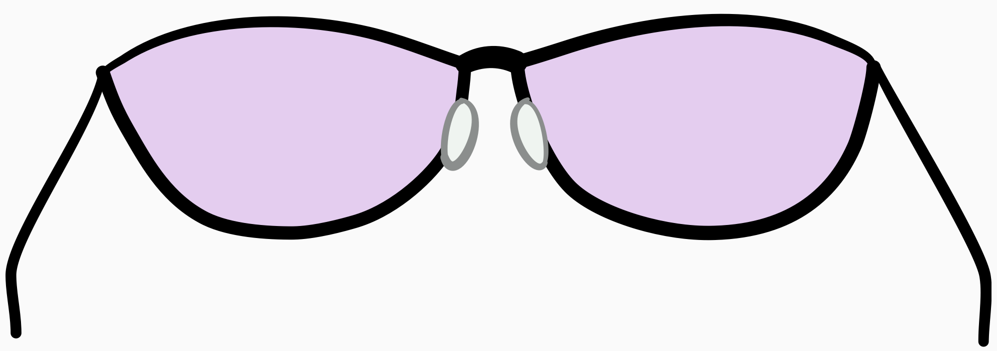
until job completes
or exceeds **rank** r



What is *r*-Work? (Gittins)

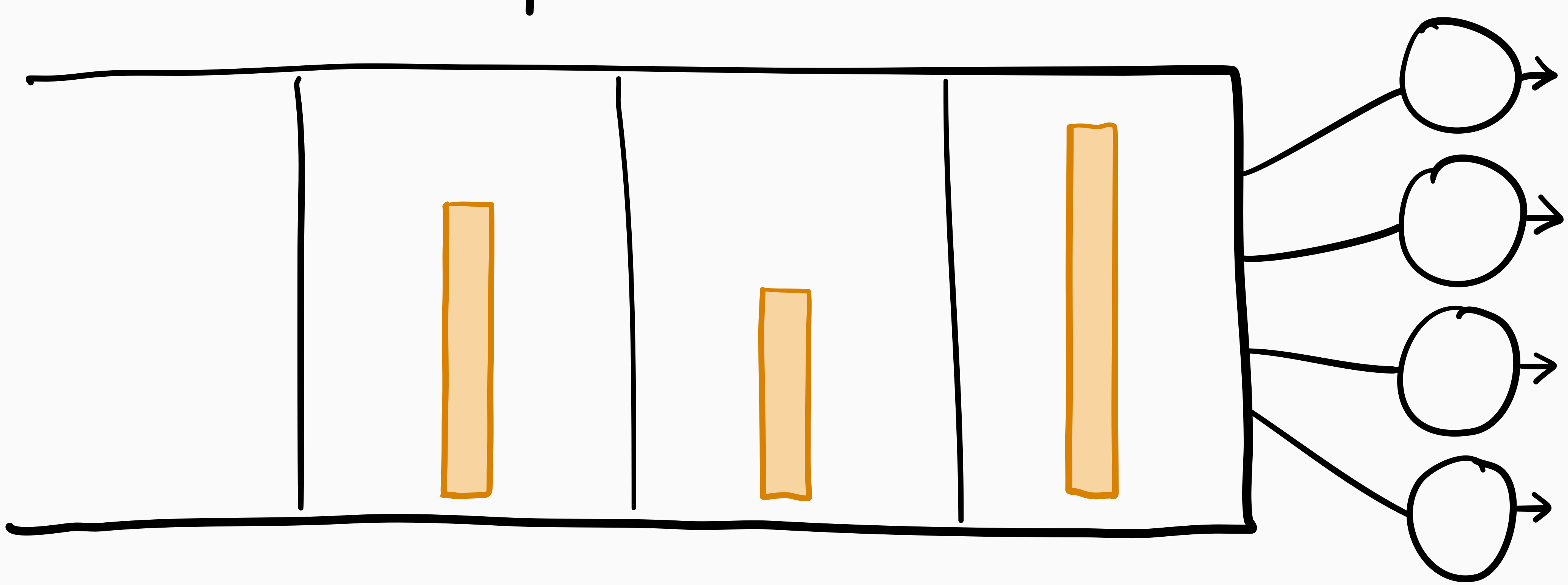
W = work = total remaining size of all jobs

$W(r)$ = ***r*-work** = work “relevant” to **rank *r***



until job completes
or exceeds **rank *r***

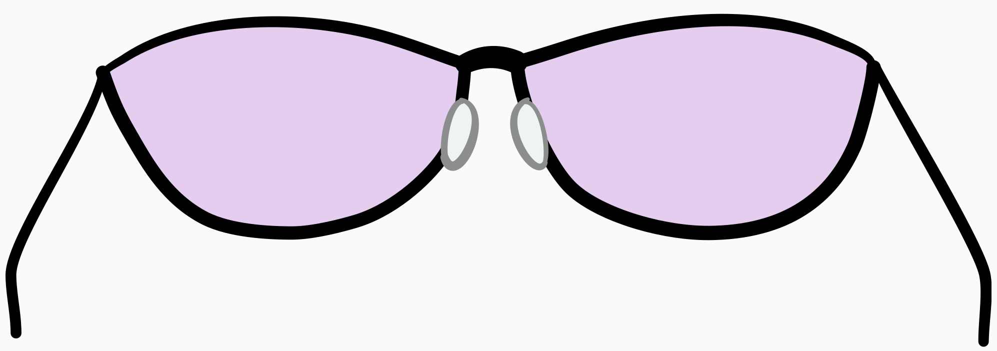
= **rank**
(lower is better)



What is *r*-Work? (Gittins)

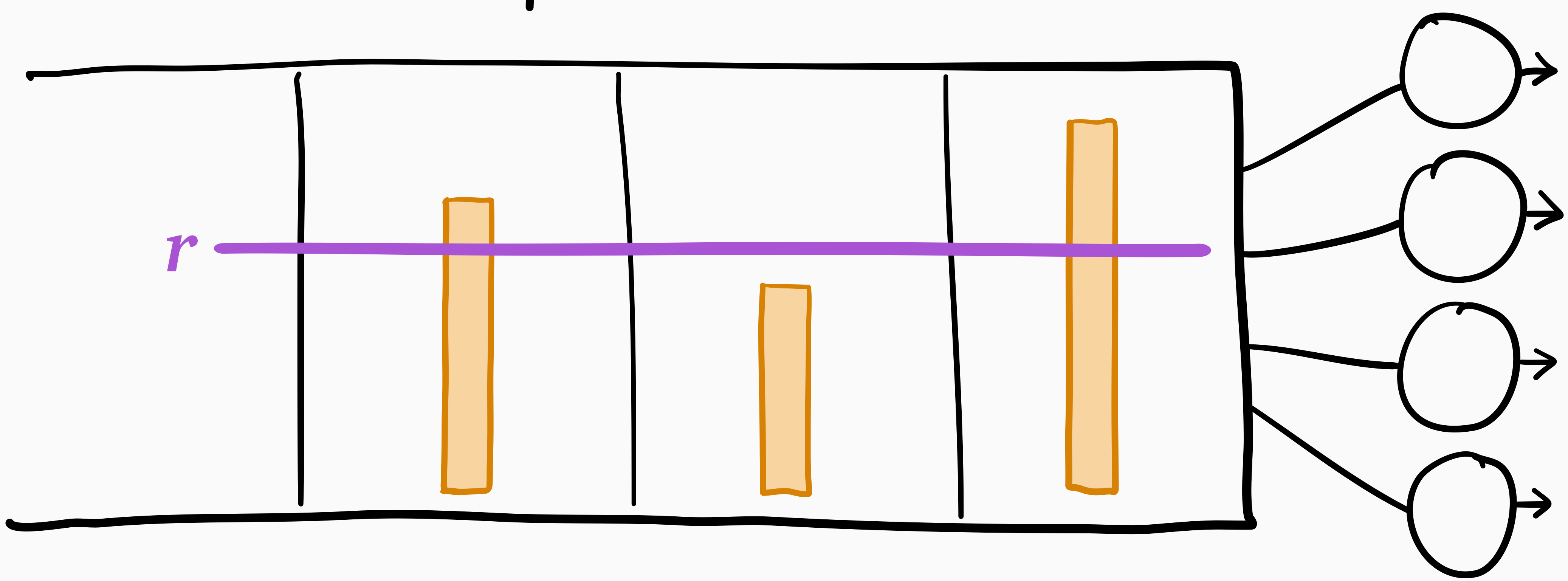
W = work = total remaining size of all jobs

$W(r)$ = *r*-work = work "relevant" to **rank** r



until job completes
or exceeds **rank** r

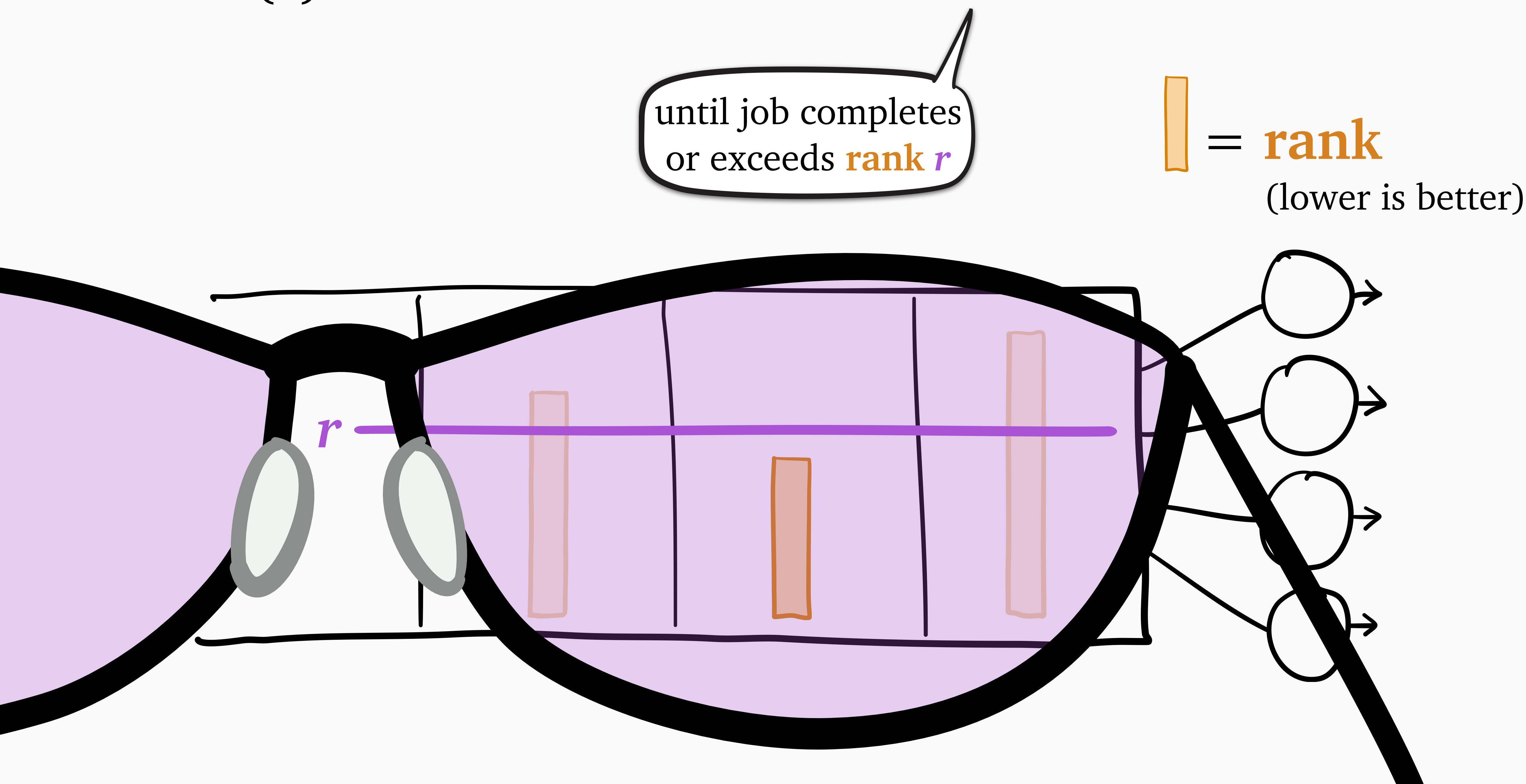
= **rank**
(lower is better)



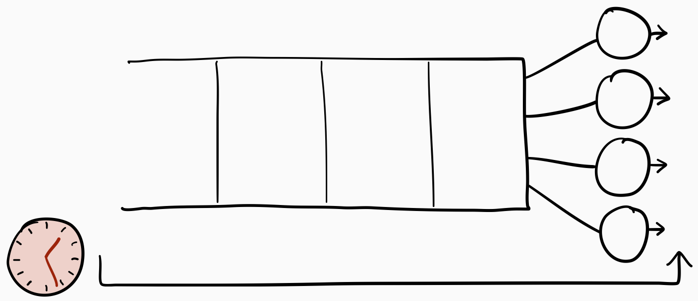
What is *r*-Work? (Gittins)

W = work = total remaining size of all jobs

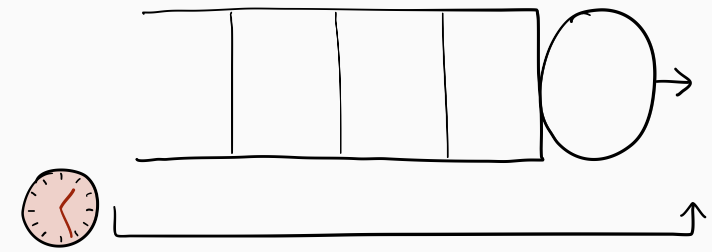
$W(r)$ = r-work = work "relevant" to rank r



Response Time via *r*-Work

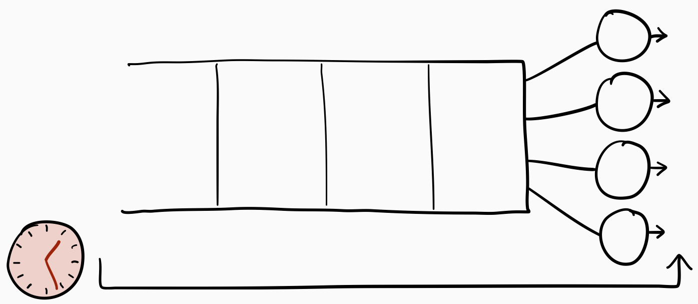


mean response
time in M/G/*k*

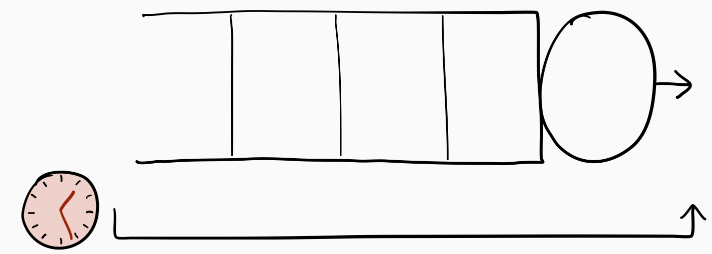


mean response
time in M/G/1

Response Time via *r*-Work

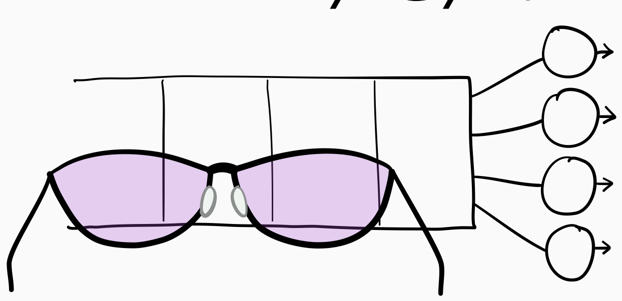


mean response time in M/G/k

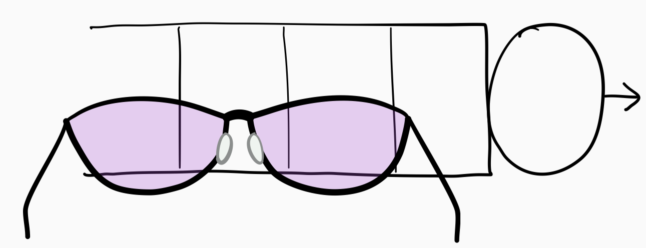


mean response time in M/G/1

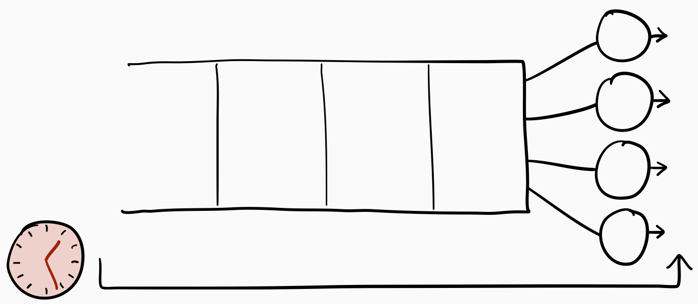
mean *r*-work in M/G/k



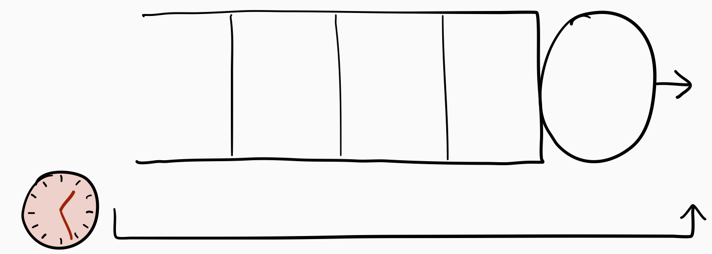
mean *r*-work in M/G/1



Response Time via *r*-Work

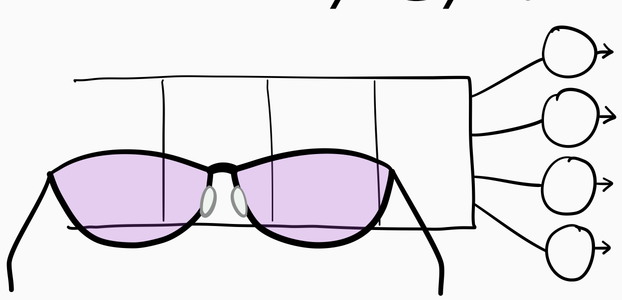


mean response time in M/G/k

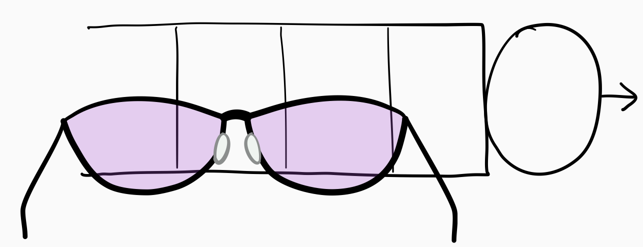


mean response time in M/G/1

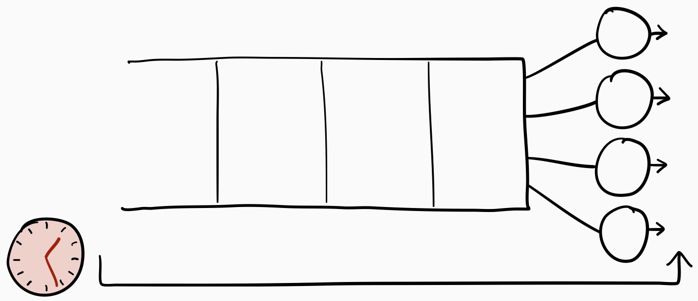
mean *r*-work in M/G/k

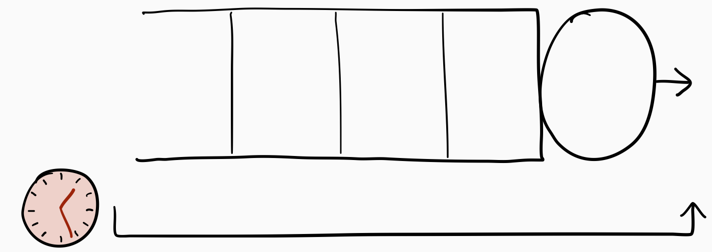


mean *r*-work in M/G/1



Response Time via *r*-Work



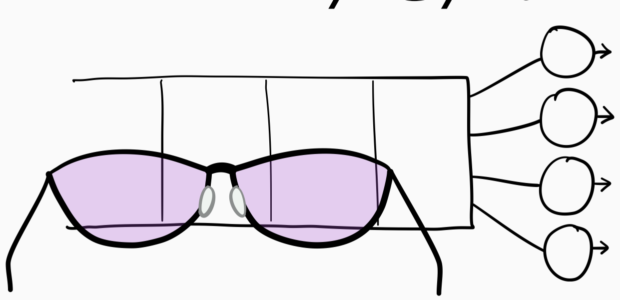


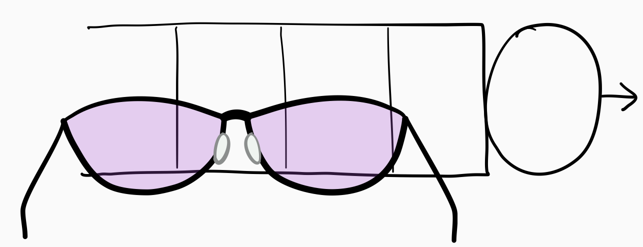
mean response time in M/G/*k*

mean response time in M/G/1

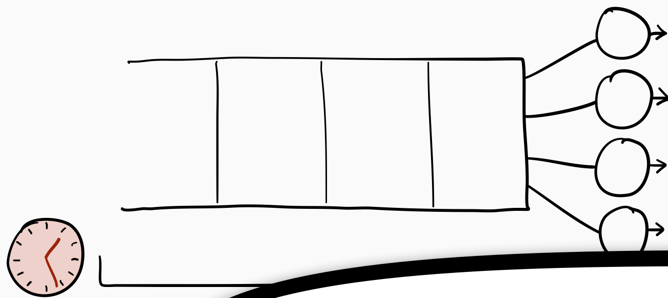
mean *r*-work in M/G/*k*

mean *r*-work in M/G/1



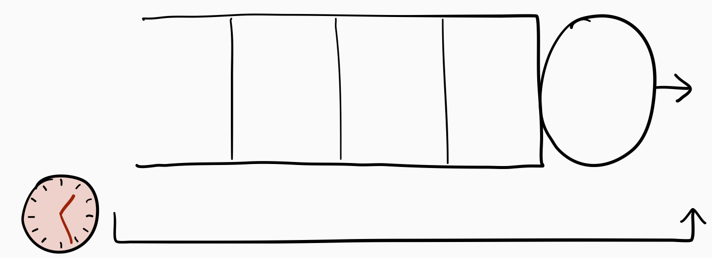


Response Time via *r*-Work



m

t



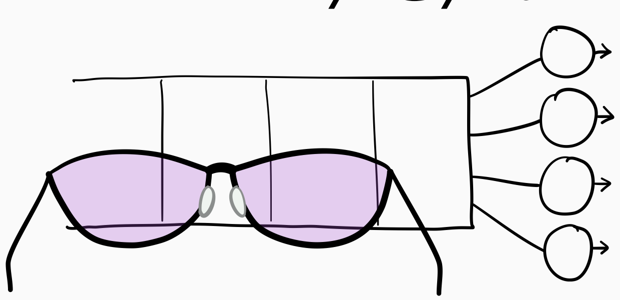
mean response time in M/G/1

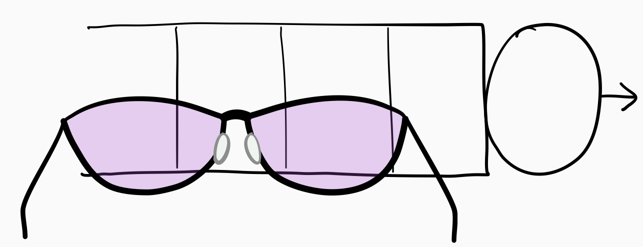
Theorem:

$$\mathbf{E}[T] = \frac{1}{\lambda} \int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\, \mathrm{d}r$$

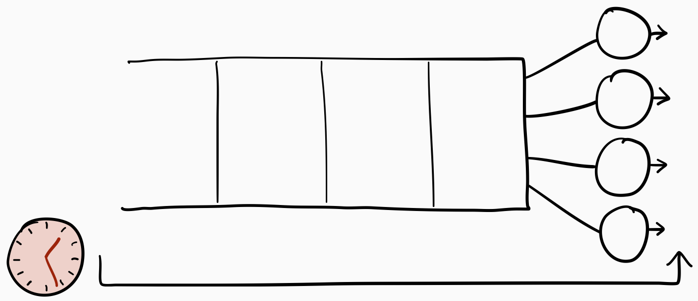
mean *r*-work in M/G/k

mean *r*-work in M/G/1

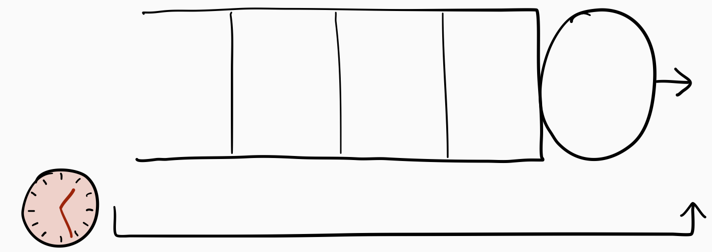




Response Time via *r*-Work



mean response time in M/G/*k*

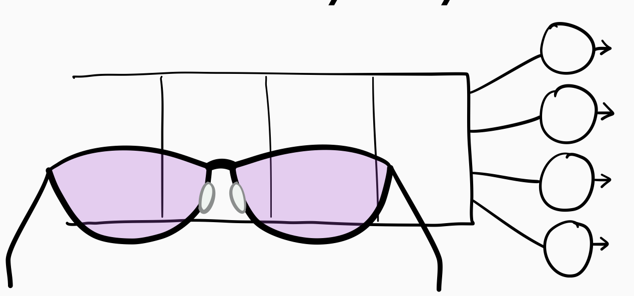


mean response time in M/G/1

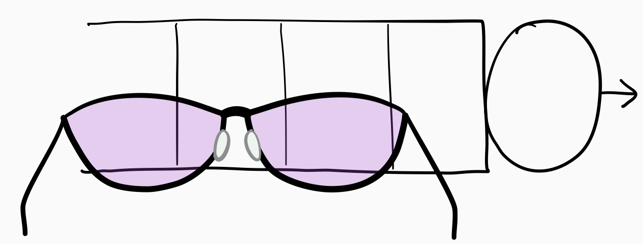
Theorem:

$$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

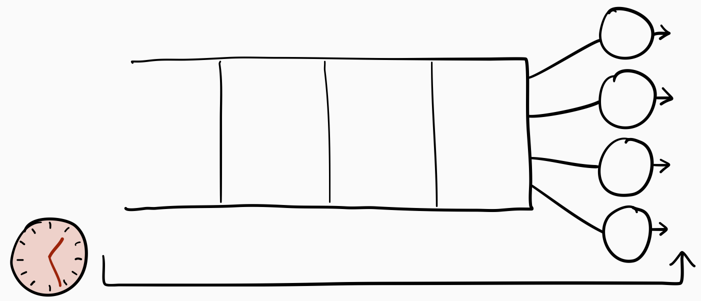
mean *r*-work in M/G/*k*



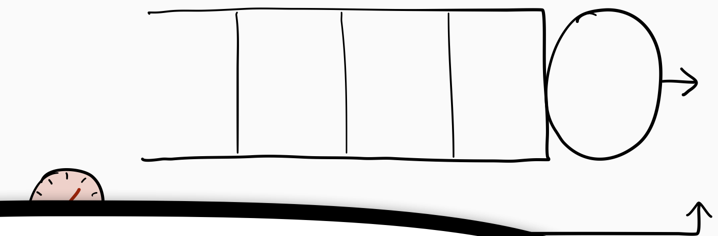
mean *r*-work in M/G/1



Response Time via *r*-Work



mean response
time in M/G/k



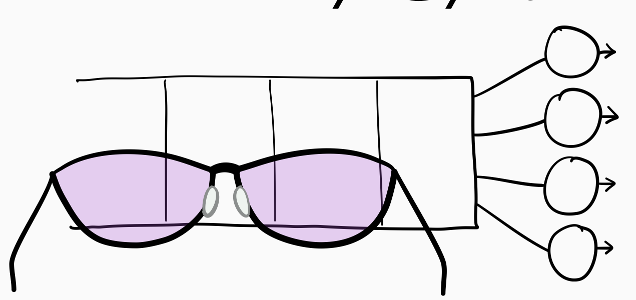
se
1

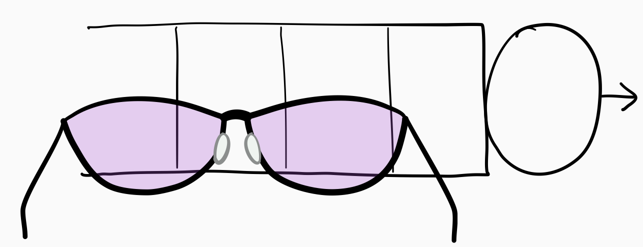
Theorem:
$$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

Theorem:
$$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

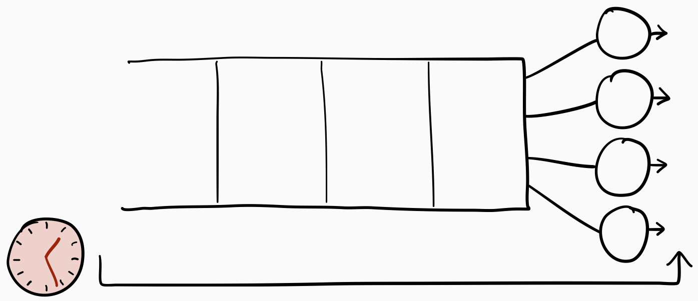
mean *r*-work
in M/G/k

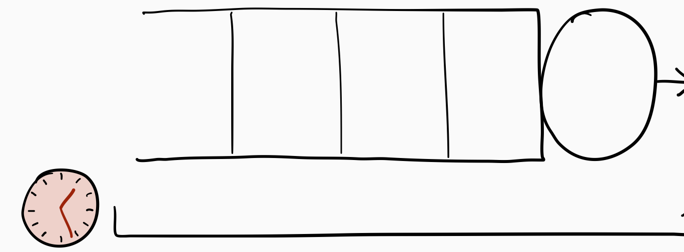
mean *r*-work
in M/G/1





Response Time via *r*-Work





mean response time in M/G/k

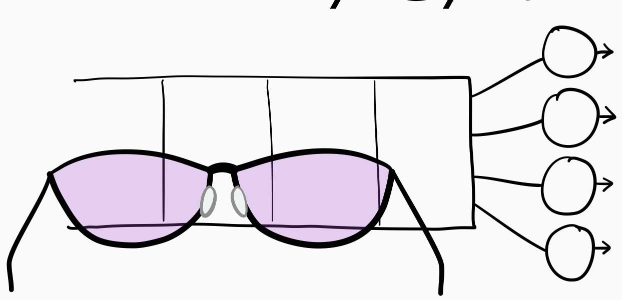
mean response time in M/G/1

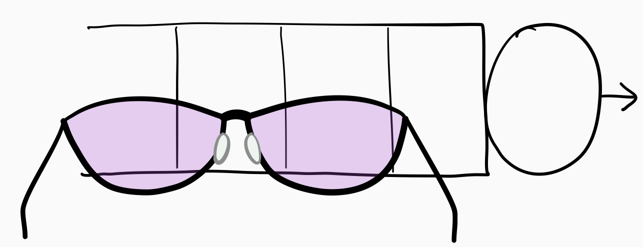
Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$

Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$

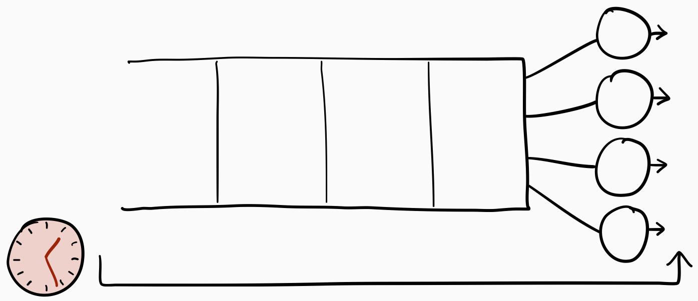
mean *r*-work in M/G/k

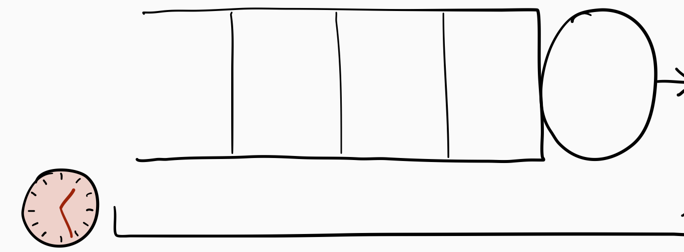
mean *r*-work in M/G/1





Response Time via *r*-Work





mean response time in M/G/k

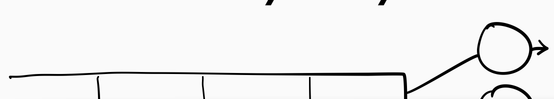
mean response time in M/G/1

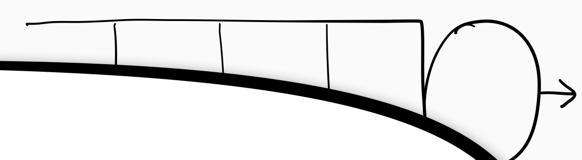
Theorem: $\mathbf{E}[T] = \frac{1}{\lambda} \int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2} \, \mathrm{d}r$

Theorem: $\mathbf{E}[T] = \frac{1}{\lambda} \int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2} \, \mathrm{d}r$

mean ***r*-work** in M/G/k

mean ***r*-work** in M/G/1

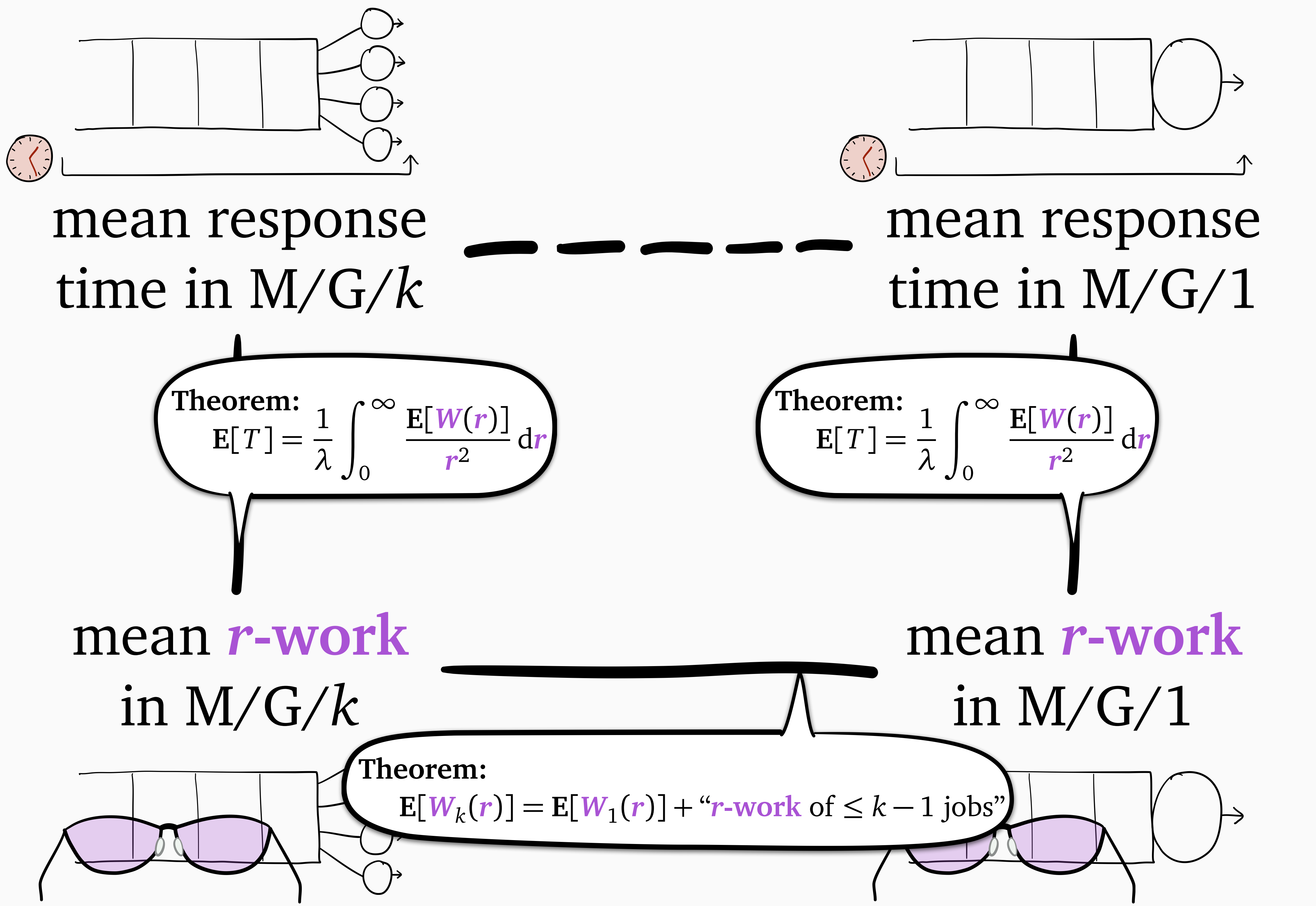




Theorem:

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}r\text{-work of} \le k-1 \text{ jobs”}$$

Response Time via *r*-Work



New $\mathbf{E}[T]$ Formula

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda} \int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2} \, \mathrm{d}r$$

New $\mathbf{E}[T]$ Formula

mean *r*-work

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda} \int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2} \, \mathrm{d}r$$

New $\mathbf{E}[T]$ Formula

mean r-work

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda} \int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2} \, \mathrm{d}r$$

Holds for *any* queueing system:
M/G/k, G/G/k, load-balancing, …

New $\mathbf{E}[T]$ Formula

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

New $\mathbf{E}[T]$ Formula

Theorem: $$\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

Proof:

New $\mathbf{E}[T]$ Formula

Th

for **SRPT** case, in which
rank = remaining size

$$\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

Proof:

New $\mathbf{E}[T]$ Formula

Th for **SRPT** case, in which **rank** = remaining size

$$\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

Proof:

$$\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \int_0^\infty \frac{\mathbf{E}\left[\sum_{i=1}^{N} \text{job } i\text{'s } r\text{-work}\right]}{r^2}\,\mathrm{d}r$$

New $\mathbf{E}[T]$ Formula

Th[eorem:] $\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$

for **SRPT** case, in which
rank = remaining size

Proof:

N = # jobs

$$\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \int_0^\infty \frac{\mathbf{E}\left[\sum_{i=1}^{N} \text{job } i\text{'s } r\text{-work}\right]}{r^2}\,\mathrm{d}r$$

New $\mathbf{E}[T]$ Formula

Th … $\dfrac{1}{\lambda}\displaystyle\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$

for **SRPT** case, in which **rank** = remaining size

N = # jobs

Proof:

$$\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \int_0^\infty \frac{\mathbf{E}\left[\sum_{i=1}^{N} \text{job } i\text{'s } r\text{-work}\right]}{r^2}\,\mathrm{d}r$$

$$= \mathbf{E}\left[\sum_{i=1}^{N} \int_0^\infty \frac{\text{job } i\text{'s } r\text{-work}}{r^2}\,\mathrm{d}r\right]$$

New $\mathbf{E}[T]$ Formula

Th

for **SRPT** case, in which **rank** = remaining size

$$\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

N = # jobs

Proof:

$$\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \int_0^\infty \frac{\mathbf{E}\left[\sum_{i=1}^N \text{job } i\text{'s } r\text{-work}\right]}{r^2}\,\mathrm{d}r$$

$$= \mathbf{E}\left[\sum_{i=1}^N \int_0^\infty \frac{\text{job } i\text{'s } r\text{-work}}{r^2}\,\mathrm{d}r\right]$$

$$= \mathbf{E}\left[\sum_{i=1}^N \int_0^\infty \frac{X_i\,\mathbb{1}(X_i \le r)}{r^2}\,\mathrm{d}r\right]$$

New $\mathbf{E}[T]$ Formula

Th

for SRPT case, in which
rank = remaining size

$$\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

Proof:

N = # jobs

$$\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \int_0^\infty \frac{\mathbf{E}\left[\sum_{i=1}^N \text{job } i\text{'s } r\text{-work}\right]}{r^2}\,\mathrm{d}r$$

$$= \mathbf{E}\left[\sum_{i=1}^N \int_0^\infty \frac{\text{job } i\text{'s } r\text{-work}}{} \right]$$

X_i = remaining size of job i

$$= \mathbf{E}\left[\sum_{i=1}^N \int_0^\infty \frac{X_i\,\mathbb{1}(X_i \le r)}{r^2}\,\mathrm{d}r\right]$$

New $\mathbf{E}[T]$ Formula

Th

for **SRPT** case, in which
rank = remaining size

$$\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$$

Proof:

$N = \#$ jobs

$$\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r = \int_0^\infty \frac{\mathbf{E}\left[\sum_{i=1}^N \text{job } i\text{'s } r\text{-work}\right]}{r^2}\,\mathrm{d}r$$

$$= \mathbf{E}\left[\sum_{i=1}^N \int_0^\infty \frac{\text{job } i\text{'s } r\text{-work}}{r^2}\,\mathrm{d}r\right]$$

X_i = remaining size of job i

$$= \mathbf{E}\left[\sum_{i=1}^N \int_0^\infty \frac{X_i\,\mathbb{1}(X_i \le r)}{r^2}\,\mathrm{d}r\right]$$

$$= \mathbf{E}[N] = \lambda\mathbf{E}[T]$$

New $\mathbf{E}[T]$ Formula

Th

for **SRPT** case, in which
rank = remaining size

$$\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,dr$$

Proof:

N = # jobs

$$\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,dr = \int_0^\infty \frac{\mathbf{E}\left[\sum_{i=1}^{N} \text{job } i\text{'s } r\text{-work}\right]}{r^2}\,dr$$

$$= \mathbf{E}\left[\sum_{i=1}^{N}\int_0^\infty \frac{\text{job } i\text{'s } r\text{-work}}{r^2}\,dr\right]$$

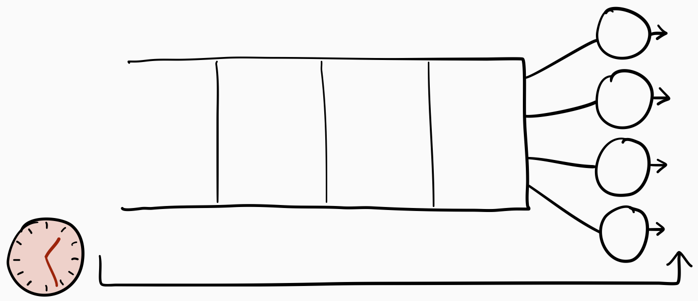
X_i = remaining size of job i

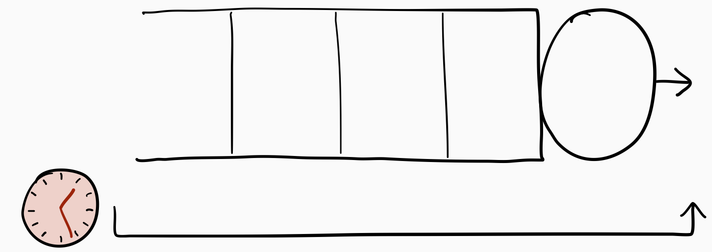
$$= \mathbf{E}\left[\sum_{i=1}^{N}\int_0^\infty \frac{X_i\,\mathbb{1}(X_i \le r)}{r^2}\,dr\right]$$

$$= \mathbf{E}[N] = \lambda\mathbf{E}[T]$$

Little's law

Response Time via *r*-Work





mean response time in M/G/k

mean response time in M/G/1

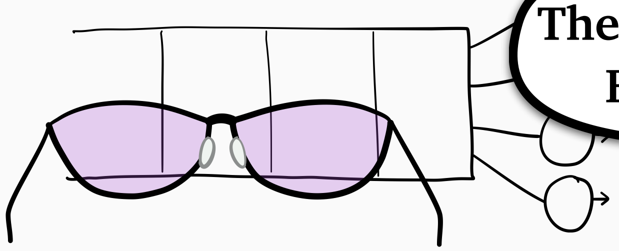
Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$

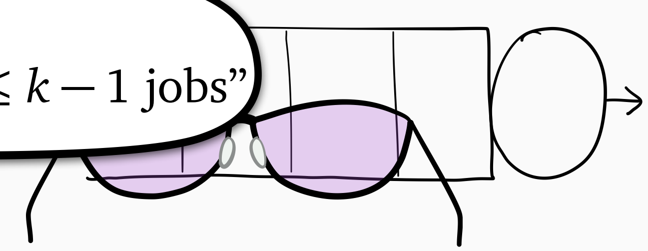
Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$

mean *r*-work in M/G/k

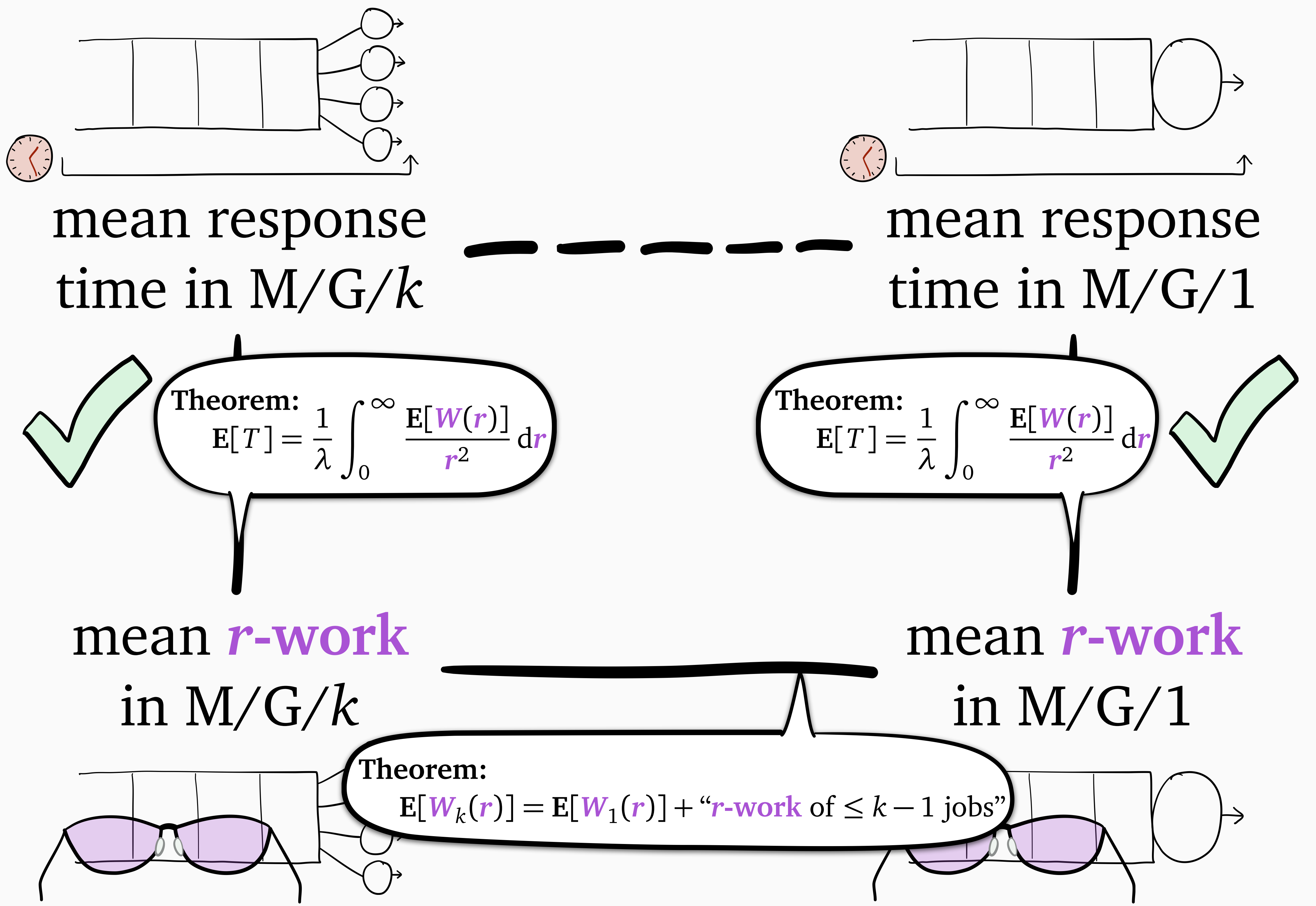
mean *r*-work in M/G/1

Theorem:
$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] +$ "*r*-work of $\leq k-1$ jobs"





Response Time via *r*-Work



Work in M/G/k vs. M/G/1

Work in M/G/k vs. M/G/1

Theorem: $$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[I_k W_k]}{1-\rho}$$

Work in M/G/k vs. M/G/1

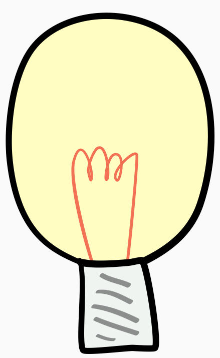
$$I_k = \frac{\#\text{ idle servers}}{k}$$

Theorem: $$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[I_k W_k]}{1-\rho}$$

Work in M/G/k vs. M/G/1

$$I_k = \frac{\#\text{ idle servers}}{k}$$

Theorem: $$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[I_k W_k]}{1-\rho}$$

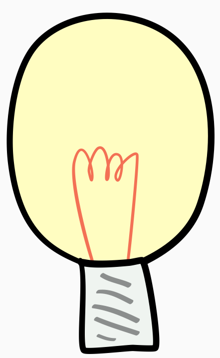


Whenever $I_k > 0$, a server is idle,
so system has at most k – 1 jobs

Work in M/G/k vs. M/G/1

$$I_k = \frac{\#\text{ idle servers}}{k}$$

Theorem: $$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \underbrace{\frac{\mathbf{E}[I_k W_k]}{1-\rho}}_{\text{“work of } \leq k-1 \text{ jobs”}}$$



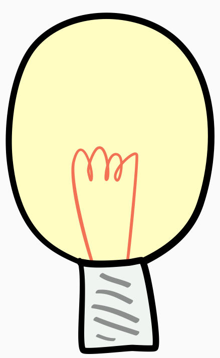
Whenever $I_k > 0$, a server is idle,
so system has at most $k - 1$ jobs

Work in M/G/k vs. M/G/1

$I_k = \frac{\#\text{ idle servers}}{k}$

Theorem: $\mathbf{E}[W_k] = \mathbf{E}[W_1] + \underbrace{\frac{\mathbf{E}[I_k W_k]}{1-\rho}}$

"work of ≤ k – 1 jobs"



Whenever $I_k > 0$, a server is idle,
so system has at most k – 1 jobs

Can generalize to *any* system
with Poisson arrivals

r-**Work** in M/G/k vs. M/G/1

Theorem:

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \text{“work of} \leq k-1 \text{ jobs”}$$

r-Work in M/G/k vs. M/G/1

Theorem:

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \text{“work of} \leq k-1 \text{ jobs”}$$

Theorem:

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}\boldsymbol{r}\textbf{-work} \text{ of} \leq k-1 \text{ jobs”}$$

r-Work in M/G/*k* vs. M/G/1

Theorem:

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \text{“work of} \leq k-1 \text{ jobs”}$$

Theorem:

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}\boldsymbol{r}\textbf{-work} \text{ of} \leq k-1 \text{ jobs”}$$

$$\leq \mathbf{E}[W_1(r)] + (k-1)r$$

r-Work in M/G/k vs. M/G/1

Theorem:

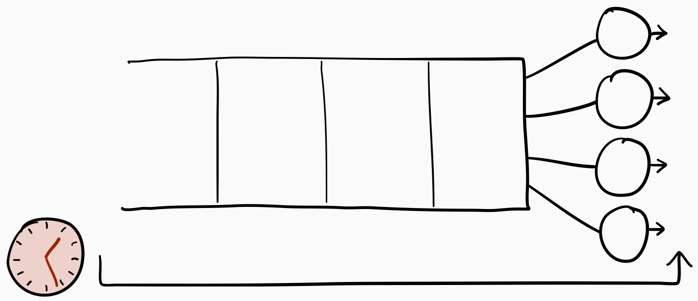
$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \text{“work of} \le k-1 \text{ jobs”}$$

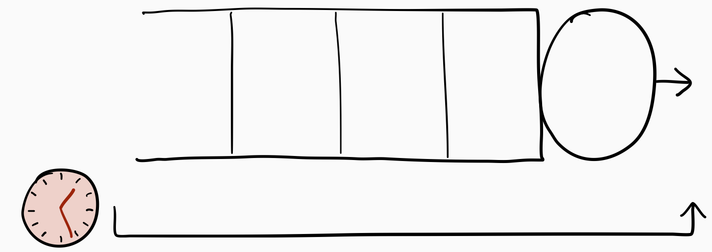
Theorem:

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}r\text{-work of} \le k-1 \text{ jobs”}$$
$$\le \mathbf{E}[W_1(r)] + (k-1)r$$

paper uses better bound

Response Time via *r*-Work





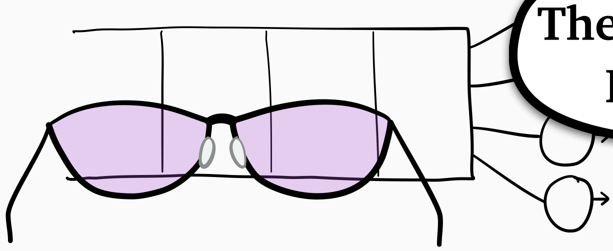
mean response time in M/G/k — mean response time in M/G/1

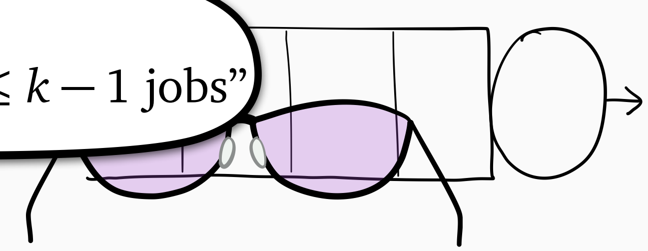
Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$

Theorem: $\mathbf{E}[T] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W(r)]}{r^2}\,\mathrm{d}r$

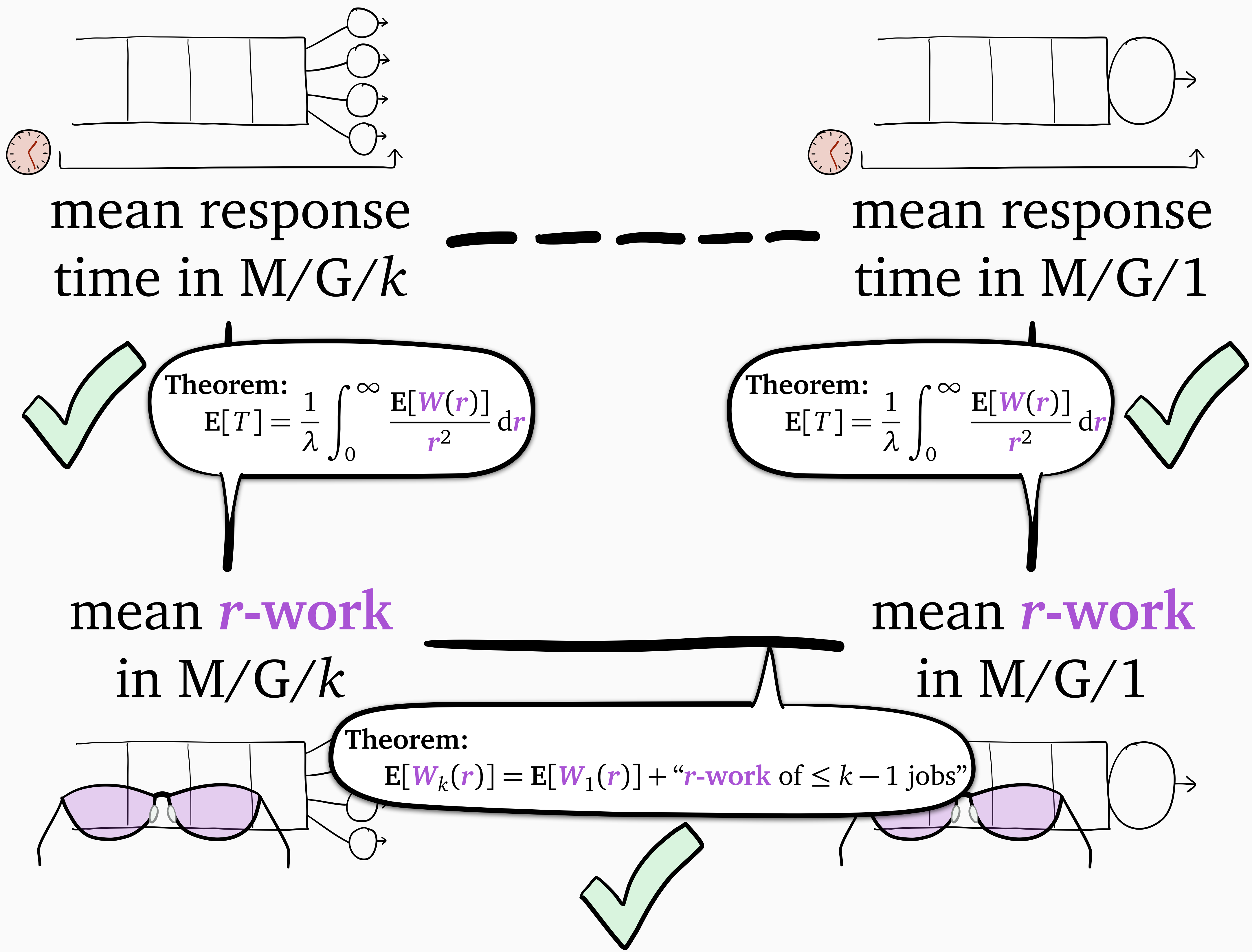
mean *r*-work in M/G/k — mean *r*-work in M/G/1

Theorem:
$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] +$ "r-work of $\leq k-1$ jobs"

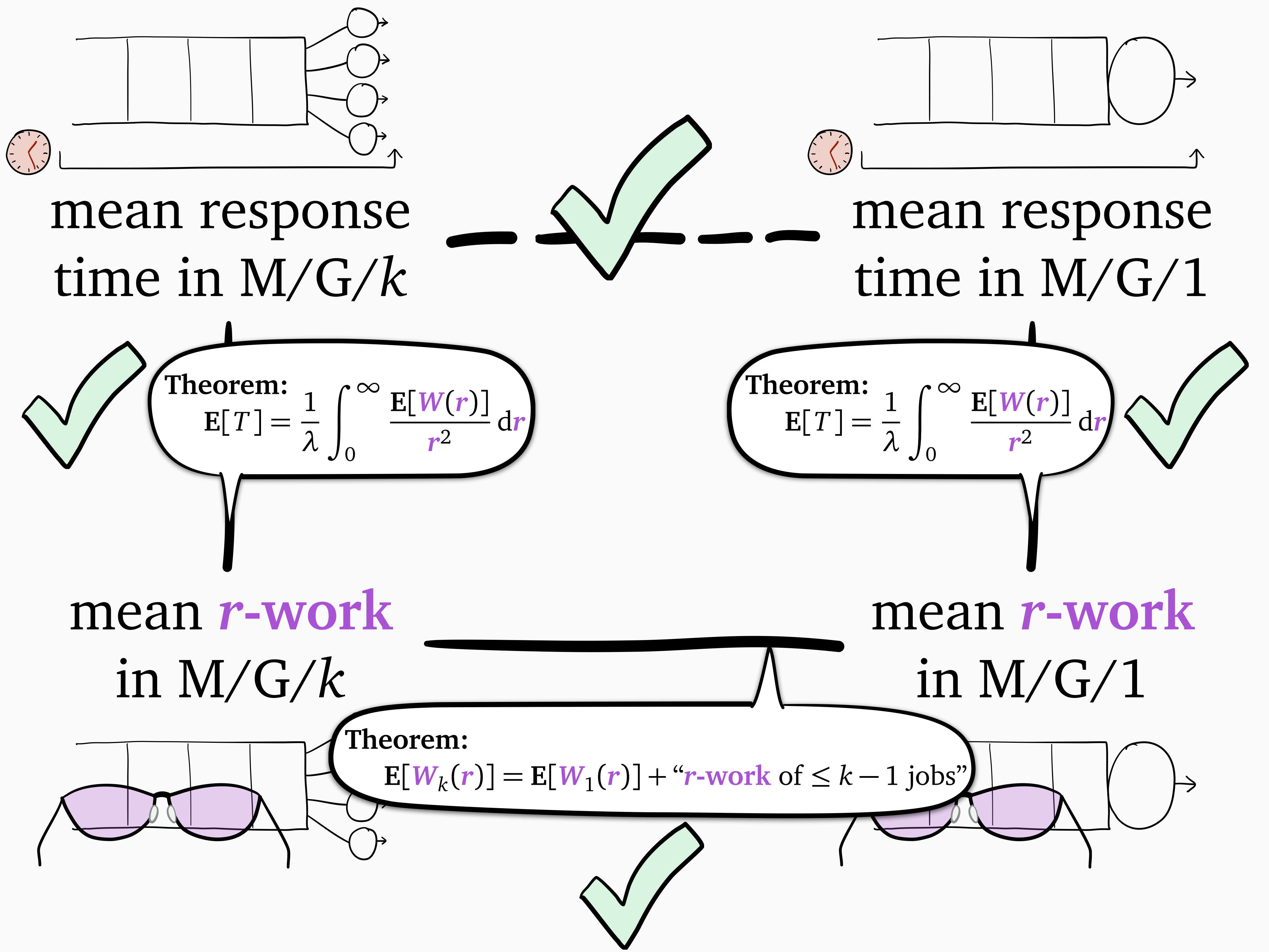




Response Time via *r*-Work



Response Time via *r*-Work



Summary

Minimize $\mathbf{E}[T]$ in M/G/k
without known job sizes

Summary

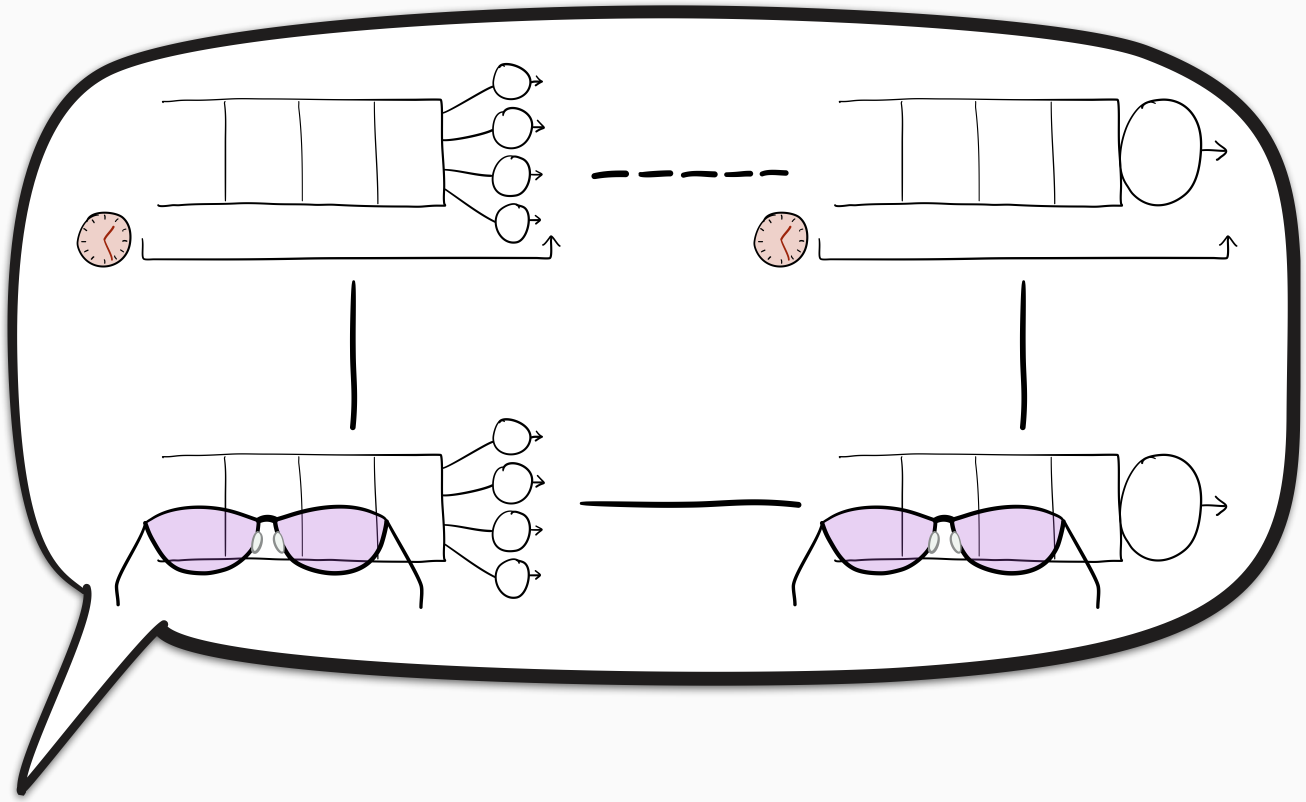
Minimize $\mathbf{E}[T]$ in M/G/k without known job sizes

Prior M/G/k techniques need known job sizes

Summary

Minimize $\mathbf{E}[T]$ in M/G/k without known job sizes

Prior M/G/k techniques need known job sizes

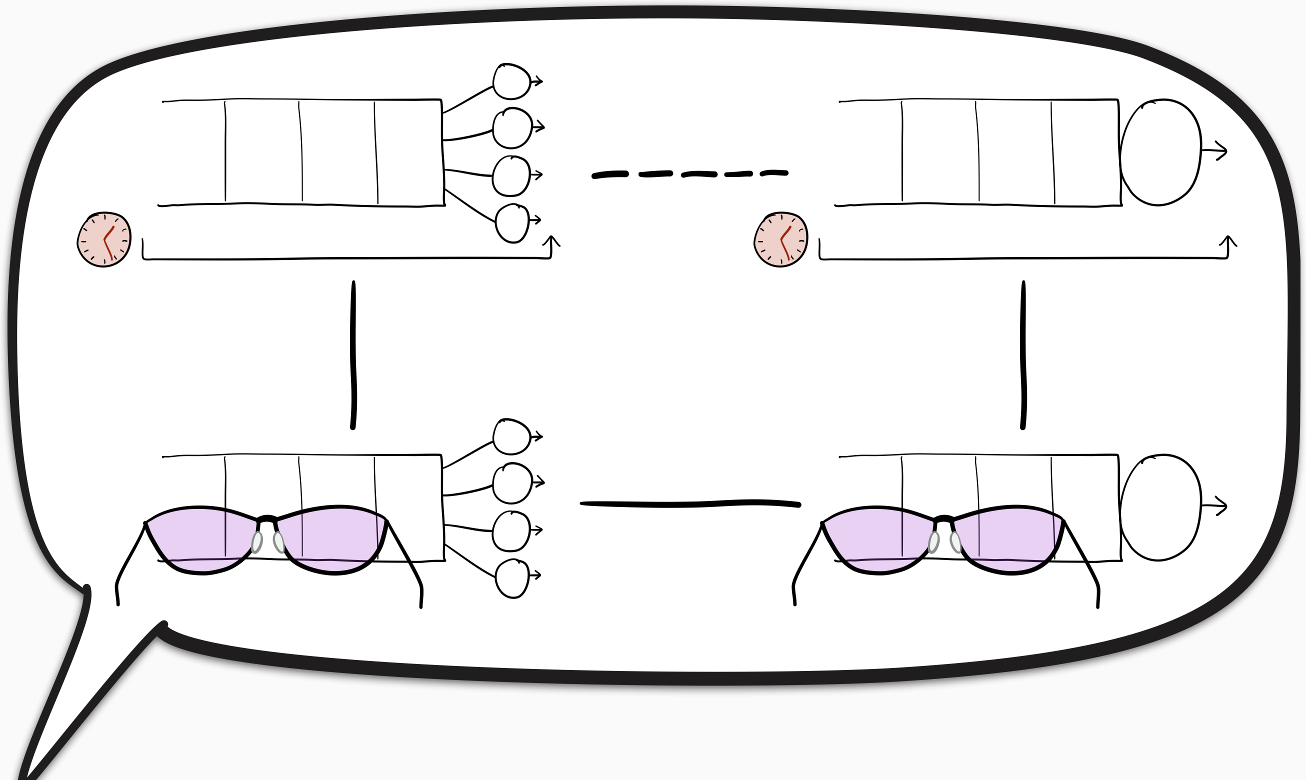


New technique based on relating $\mathbf{E}[T]$ to ***r*-work**

Summary

Minimize $\mathbf{E}[T]$ in M/G/k without known job sizes

Prior M/G/k techniques need known job sizes



$$\mathbf{E}[T_k] \le \mathbf{E}[T_1] + (k-1)\cdot O\left(\log \frac{1}{1-\rho}\right)$$

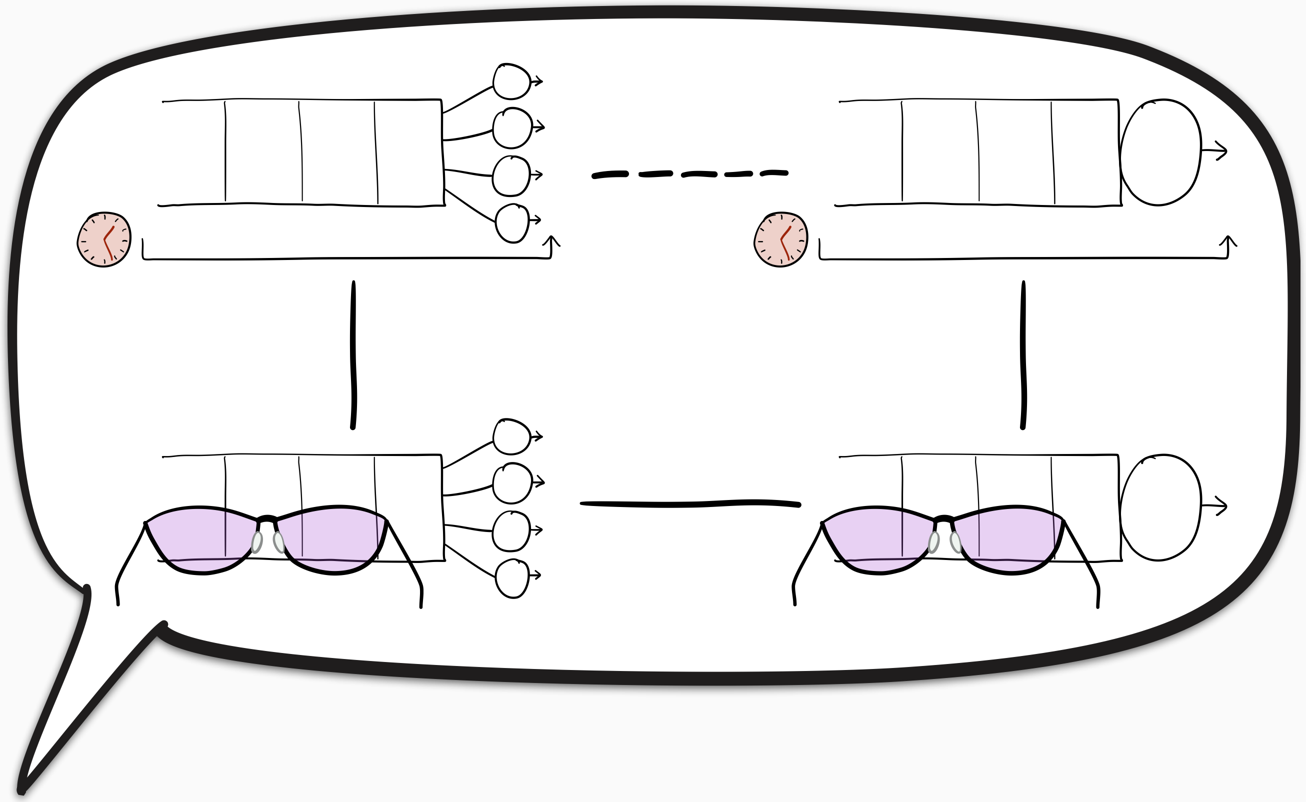
New technique based on relating $\mathbf{E}[T]$ to *r*-**work**

Gittins has near-optimal $\mathbf{E}[T]$ in M/G/k

Summary

Minimize $\mathbf{E}[T]$ in M/G/k without known job sizes

Prior M/G/k techniques need known job sizes



$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + (k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)$$

New technique based on relating $\mathbf{E}[T]$ to *r-work*

Gittins has near-optimal $\mathbf{E}[T]$ in M/G/k

Get in touch: `zscully@cs.cmu.edu`

Bonus Slides

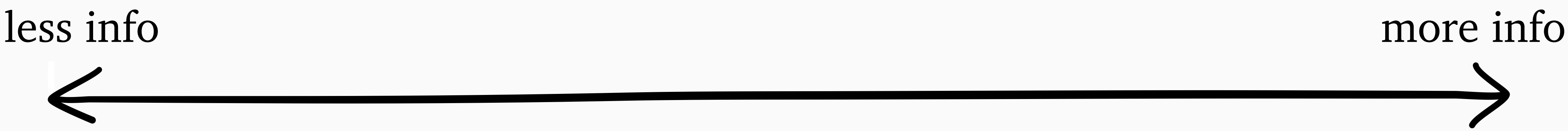
Levels of Size Information

less info — more info

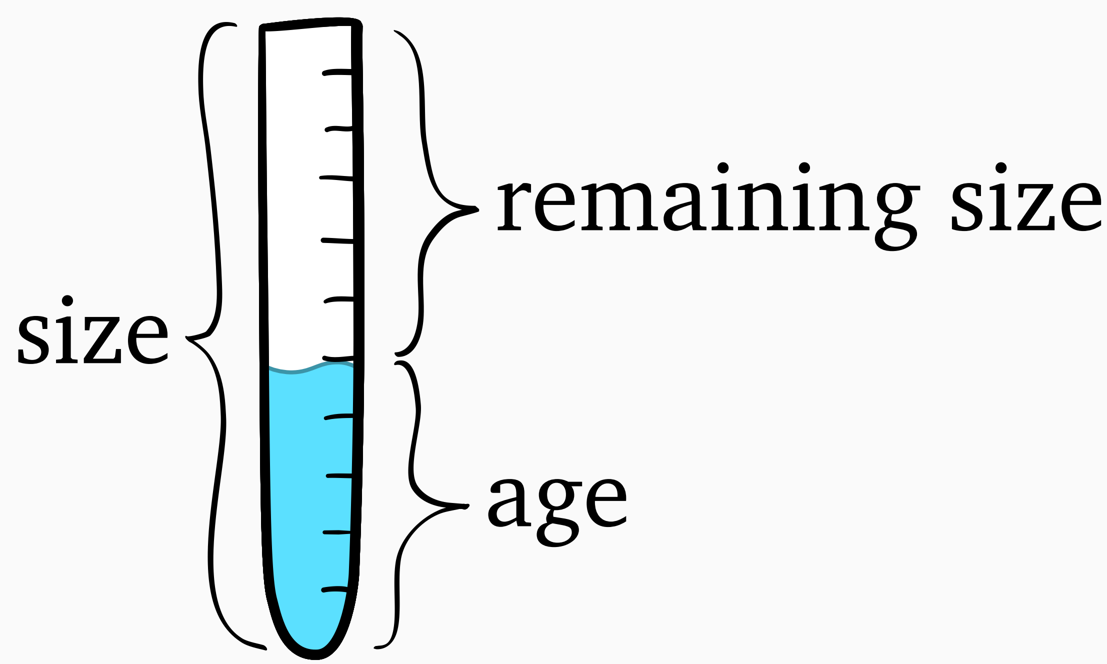
Levels of Size Information

Known Size

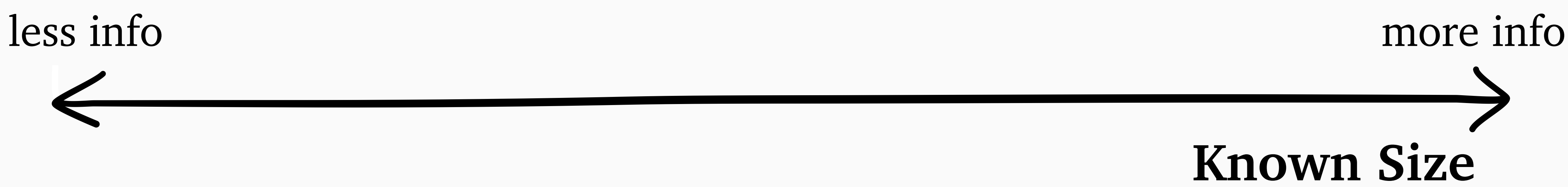
Levels of Size Information



Known Size



Levels of Size Information



Levels of Size Information

less info

more info

Known Size

Blind

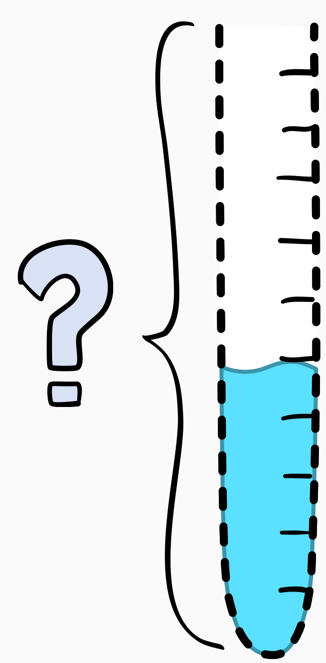
Levels of Size Information

less info — more info

Known Size

Blind

unknown size

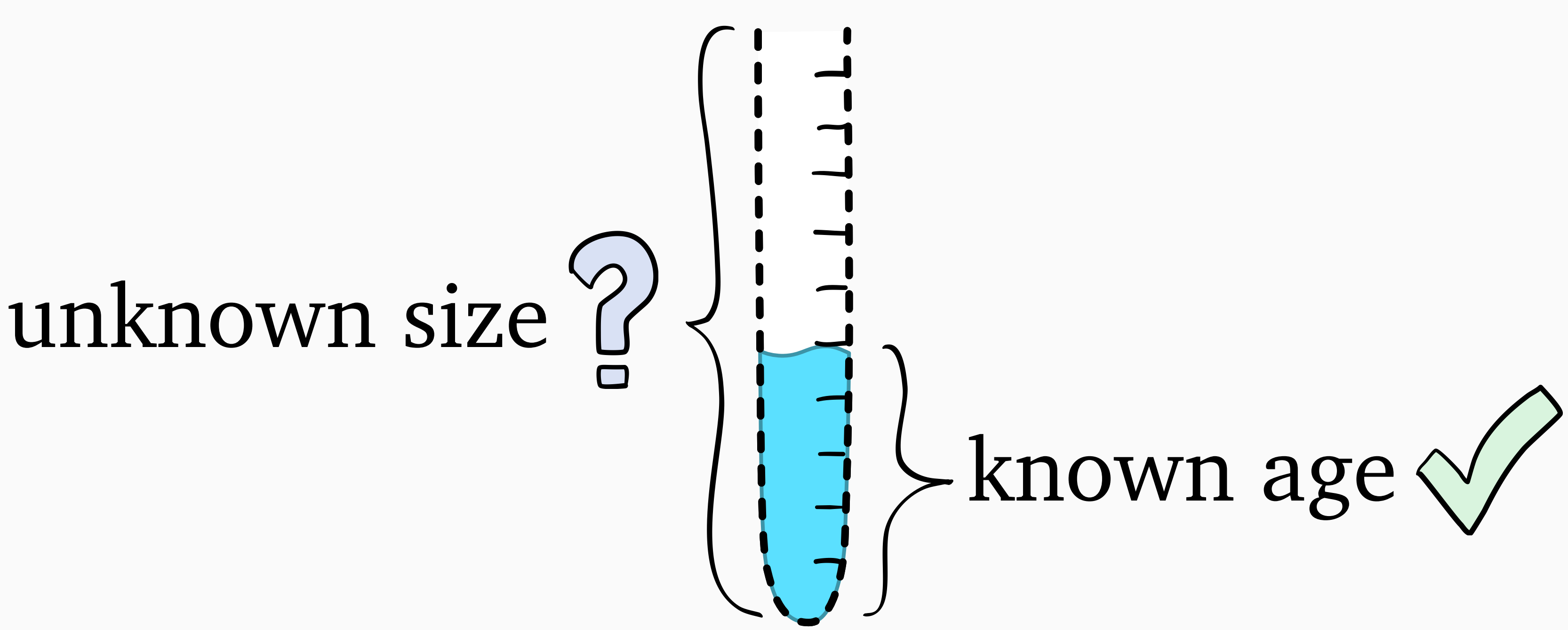


Levels of Size Information

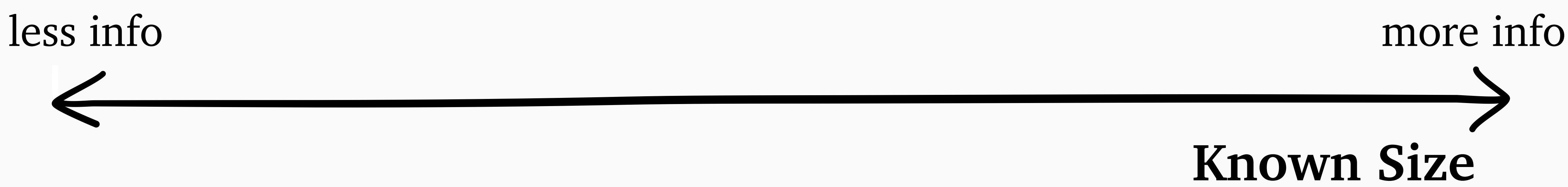
less info ⟷ more info

Known Size

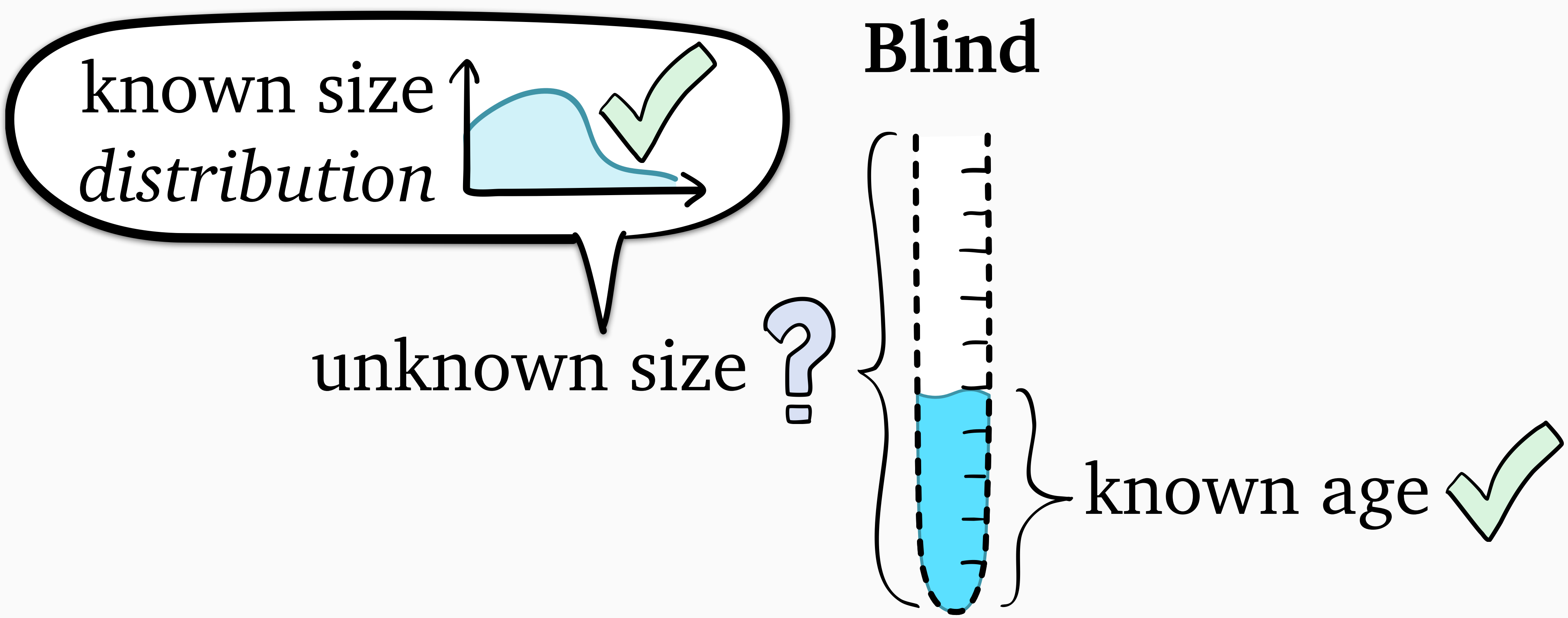
Blind



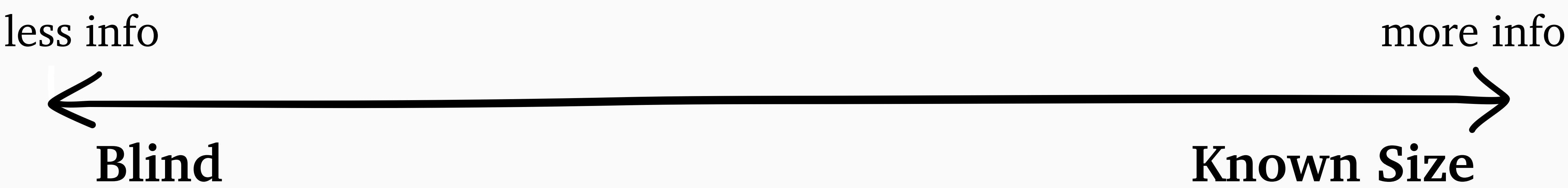
Levels of Size Information



Blind



Levels of Size Information



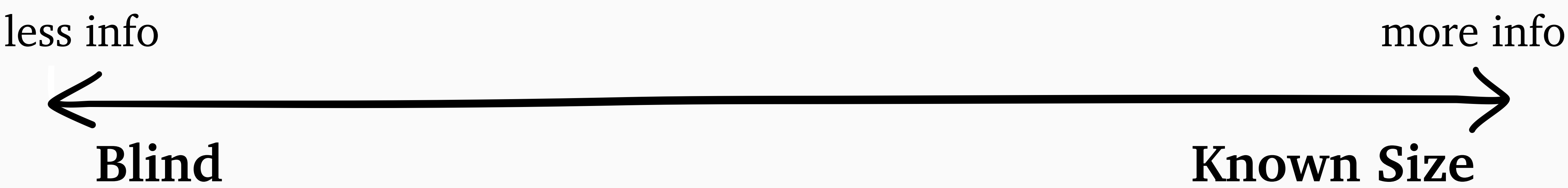
Levels of Size Information

less info — more info

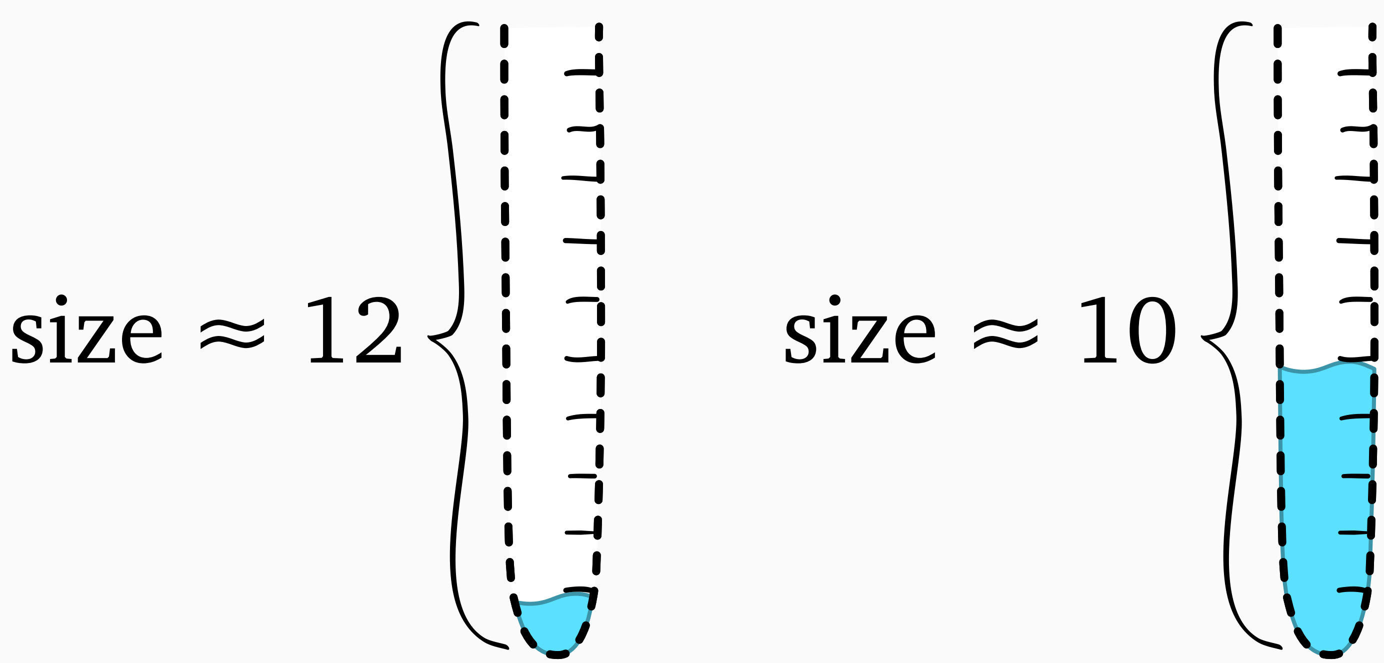
Blind — **Known Size**

Noisy Estimates

Levels of Size Information



Noisy Estimates

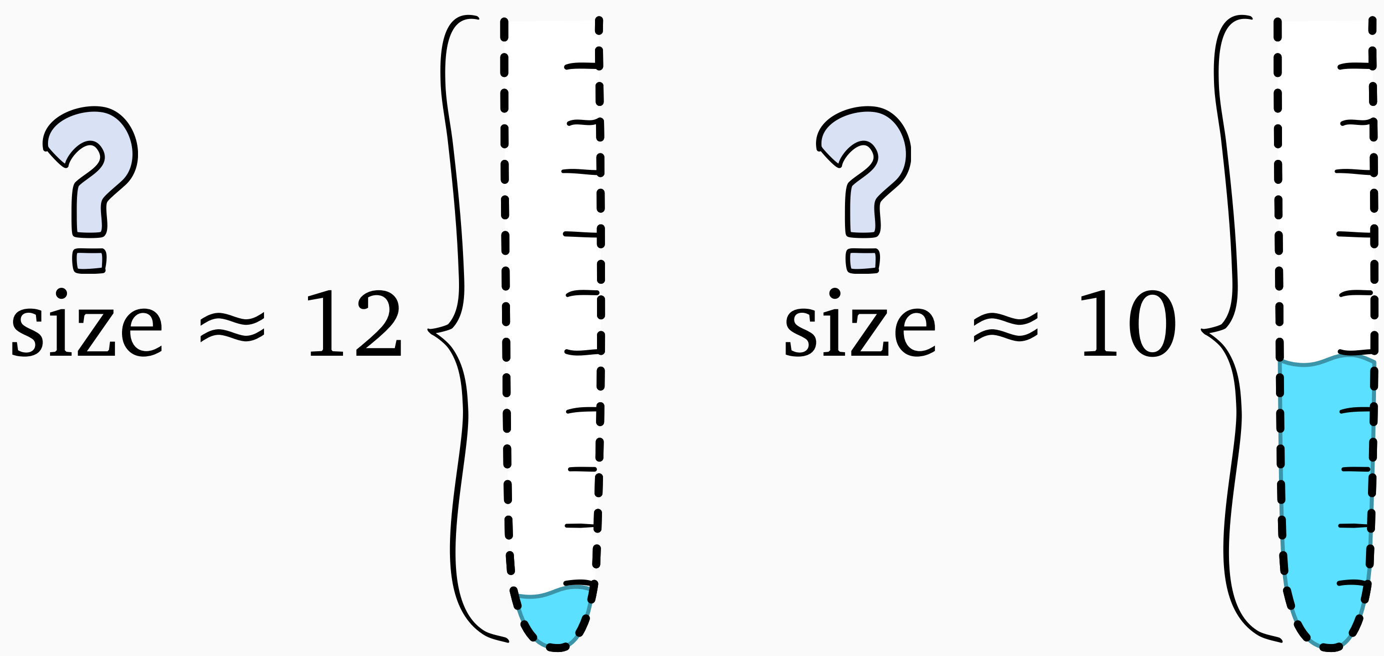


Levels of Size Information

less info — more info

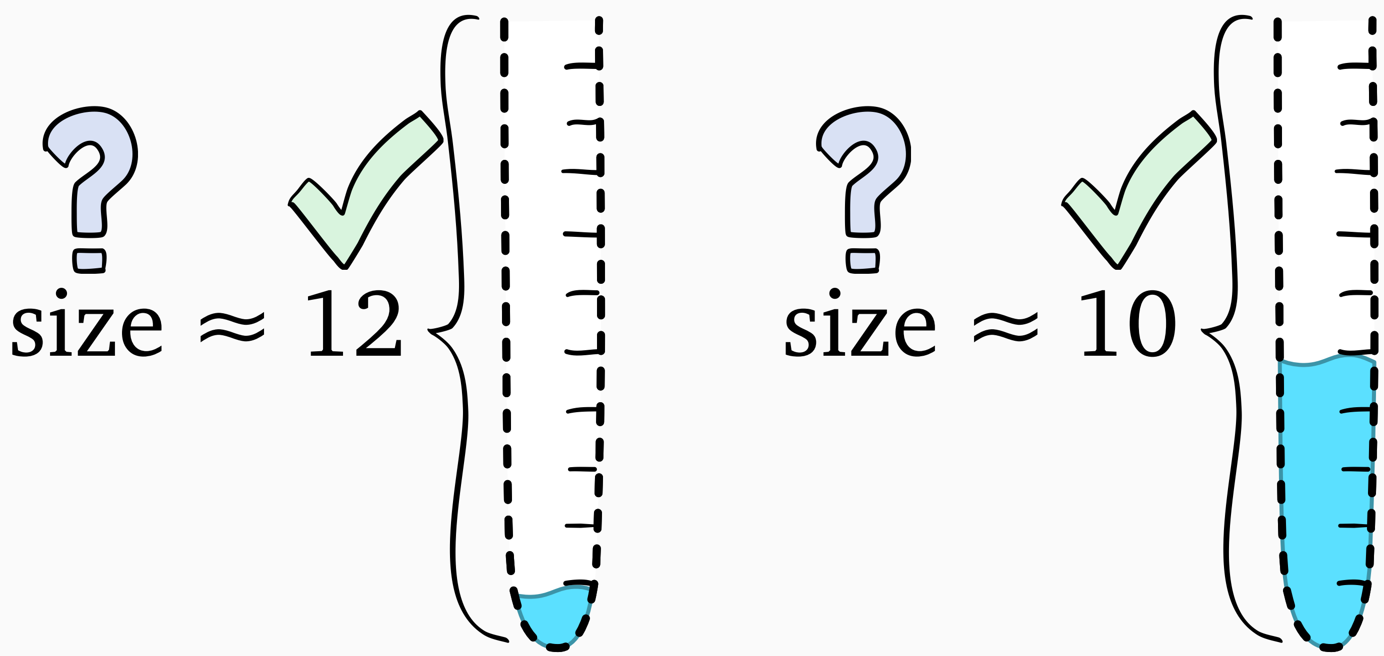
Blind ⟷ **Known Size**

Noisy Estimates



Levels of Size Information

Noisy Estimates

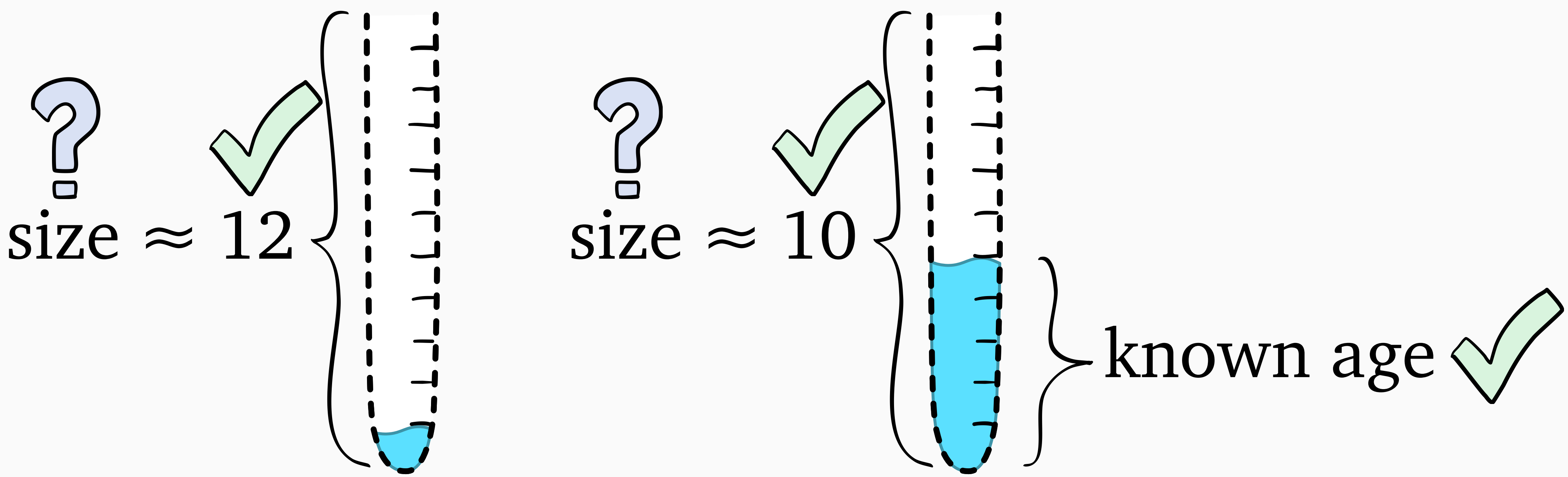


Levels of Size Information

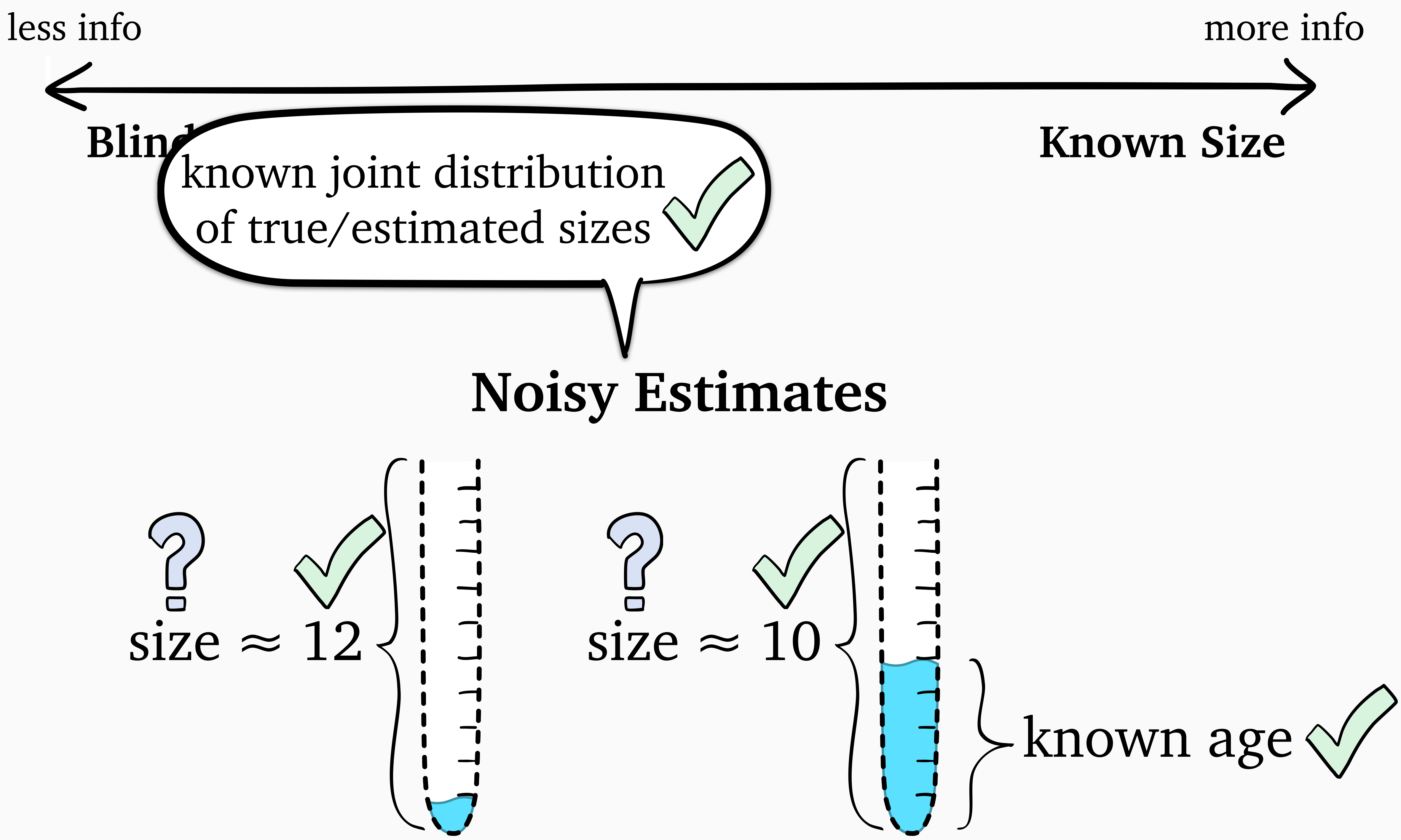
less info ←→ more info

Blind … **Known Size**

Noisy Estimates



Levels of Size Information



Levels of Size Information

less info ⟷ more info

Blind — **Noisy Estimates** — **Known Size**

Levels of Size Information

less info ⟷ more info

Blind — **Noisy Estimates** — **Known Size**

General case: a job is a *Markov process*

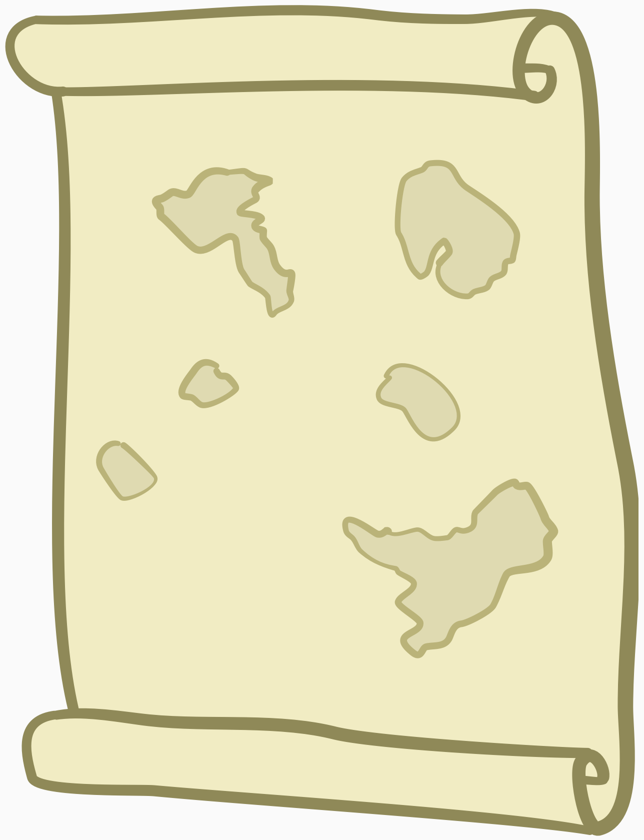
Levels of Size Information

less info ⟷ more info

Blind — **Noisy Estimates** — **Known Size**

General case: a job is a *Markov process*

 general *state space*



Levels of Size Information

less info ⟷ more info

Blind — **Noisy Estimates** — **Known Size**

General case: a job is a *Markov process*

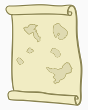
- general *state space*
- job's ***state*** encodes all known info

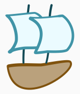
Levels of Size Information

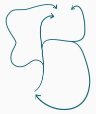
less info ⟷ more info

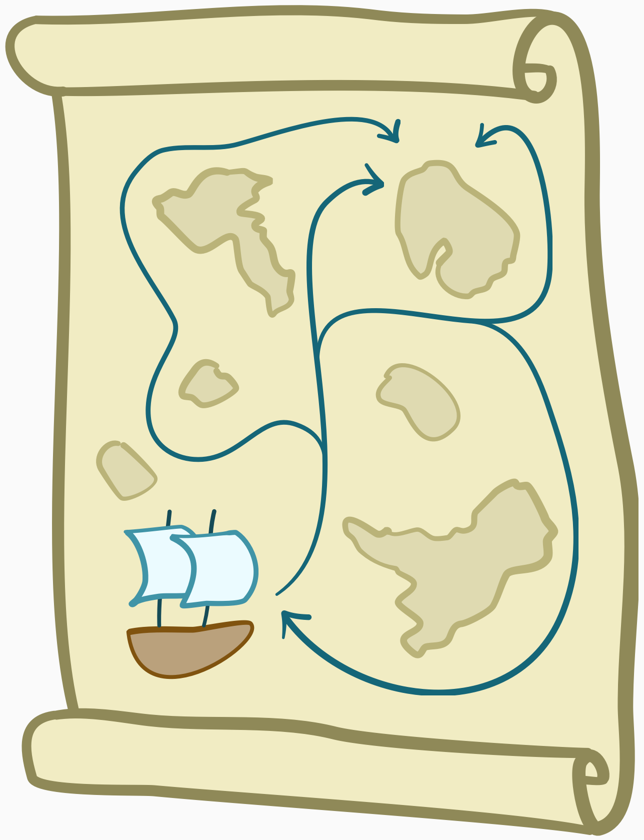
Blind — **Noisy Estimates** — **Known Size**

General case: a job is a *Markov process*

 general *state space*

 job's ***state*** encodes all known info

 **state** *stochastically evolves* with service



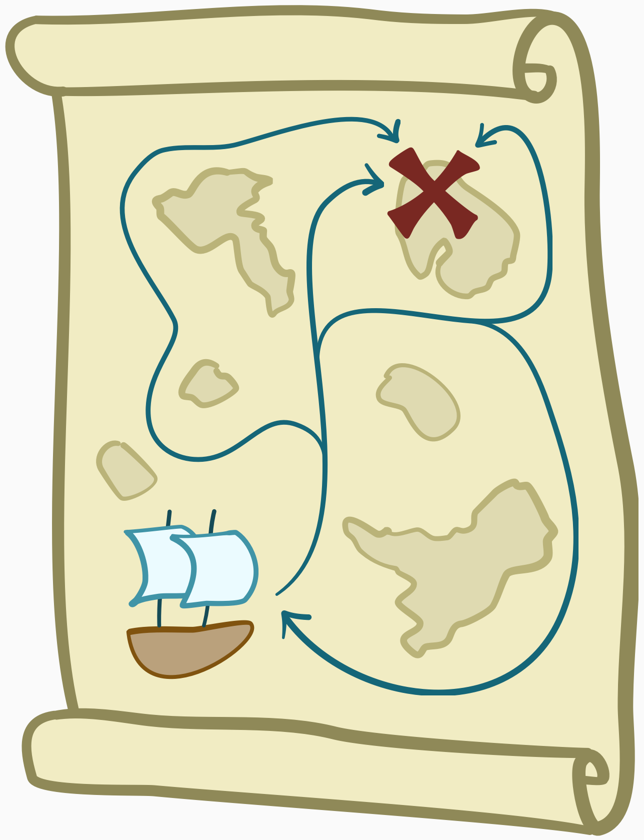
Levels of Size Information

less info ⟵⟶ more info

Blind — **Noisy Estimates** — **Known Size**

General case: a job is a *Markov process*

- general *state space*
- job's ***state*** encodes all known info
- **state** *stochastically evolves* with service
- completes upon entering *goal state*



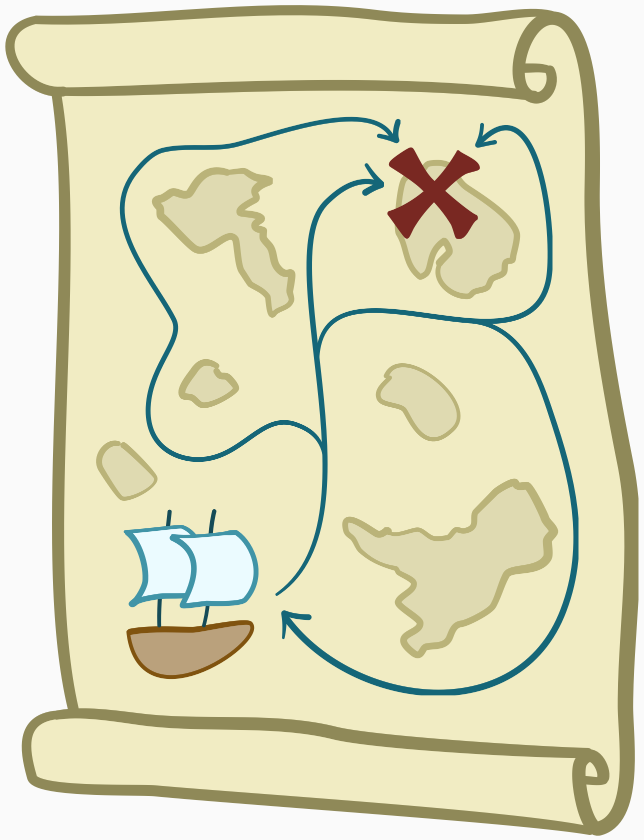
Levels of Size Information

less info ⟷ more info

Blind — **Noisy Estimates** — **Known Size**

General case: a job is a *Markov process*

- ✓ general *state space*
- ✓ job's ***state*** encodes all known info
- **state** *stochastically evolves* with service
- ✓ completes upon entering *goal state*



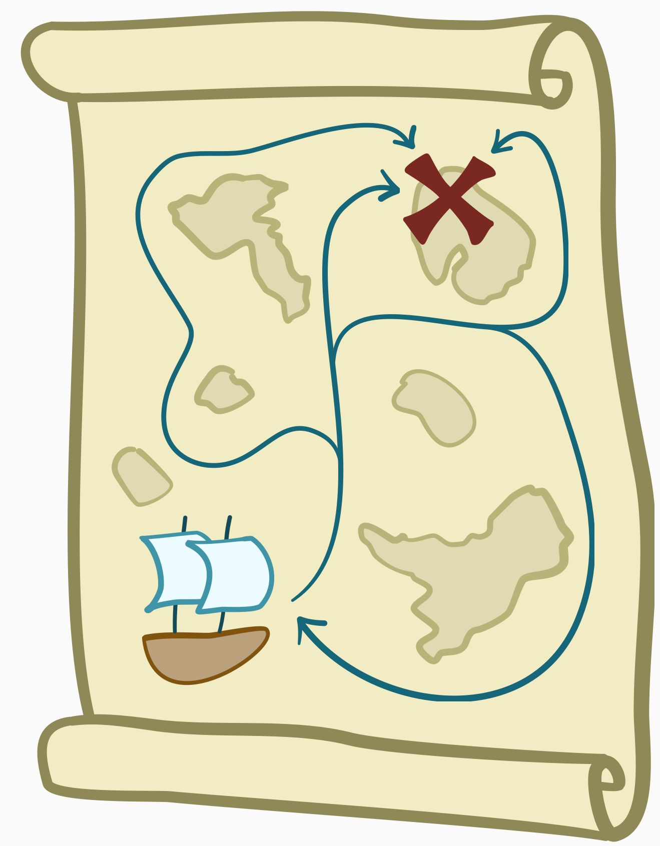
Levels of Size Information

less info ⟷ more info

Blind — **Noisy Estimates** — **Known Size**

General case: a job is a *Markov process*

- general *state space*
- job's ***state*** encodes all known info
- **state** *stochastically evolves* with service
- completes upon entering *goal state*



Levels of Size Information

less info ⟷ more info

Blind **Noisy Estimates** **Known Size**

General case: a job is a *Markov process*

- general *state space*
- job's ***state*** encodes all known info
- **state** *stochastically evolves* with service
- completes upon entering *goal state*

but *probabilities* known