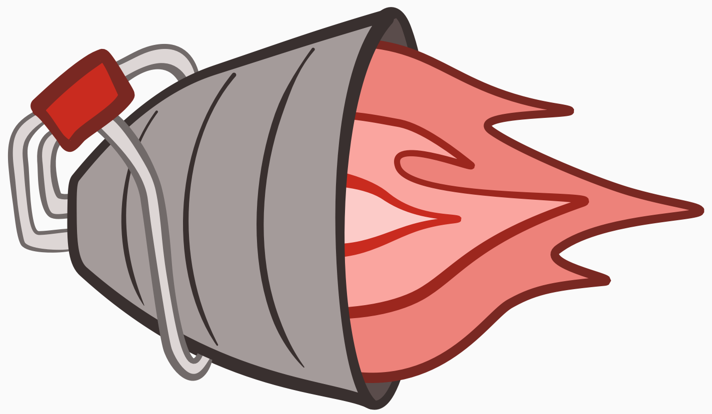
Strongly Tail-Optimal Scheduling *in the Light-Tailed M/G/1*

Ziv Scully Cornell ORIE

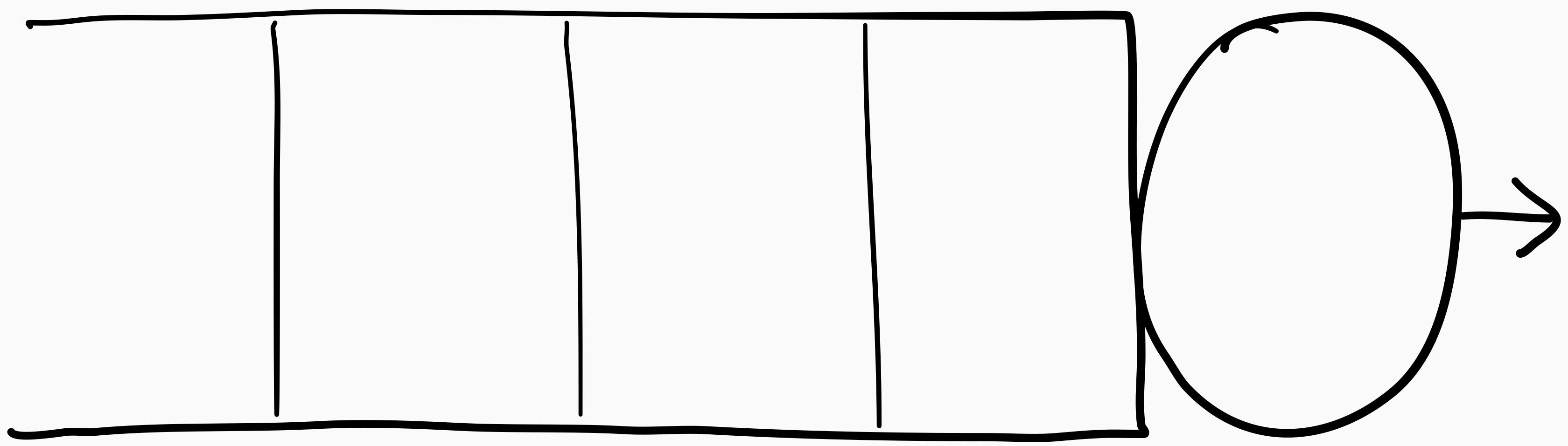
Joint work with

George Yu Cornell ORIE
Amit Harlev Cornell CAM

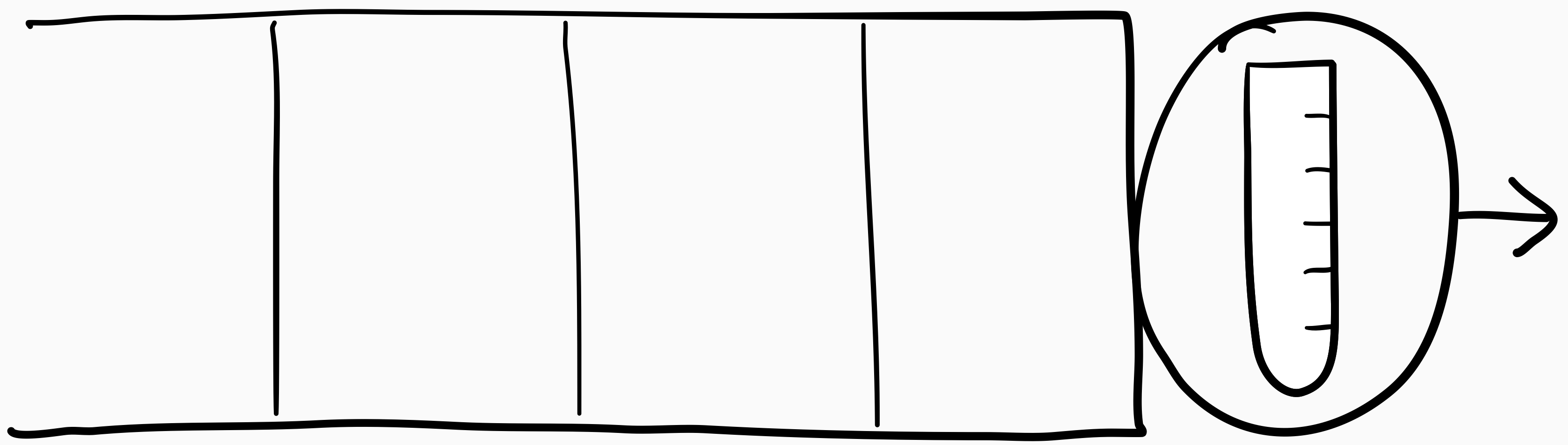


How should we schedule jobs to minimize delay?

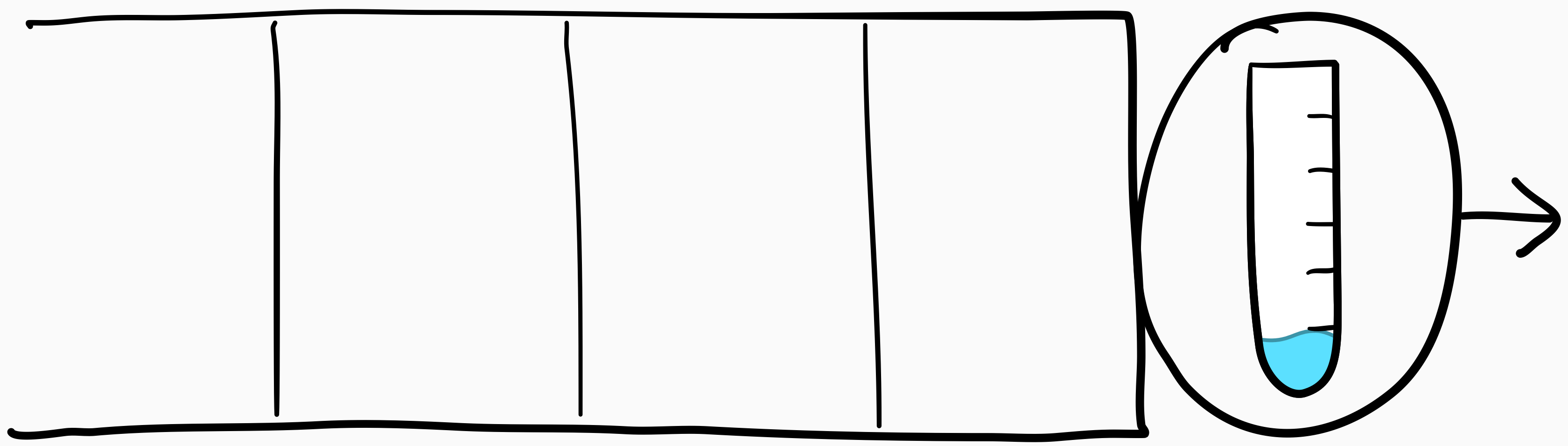
How should we schedule jobs to minimize delay?



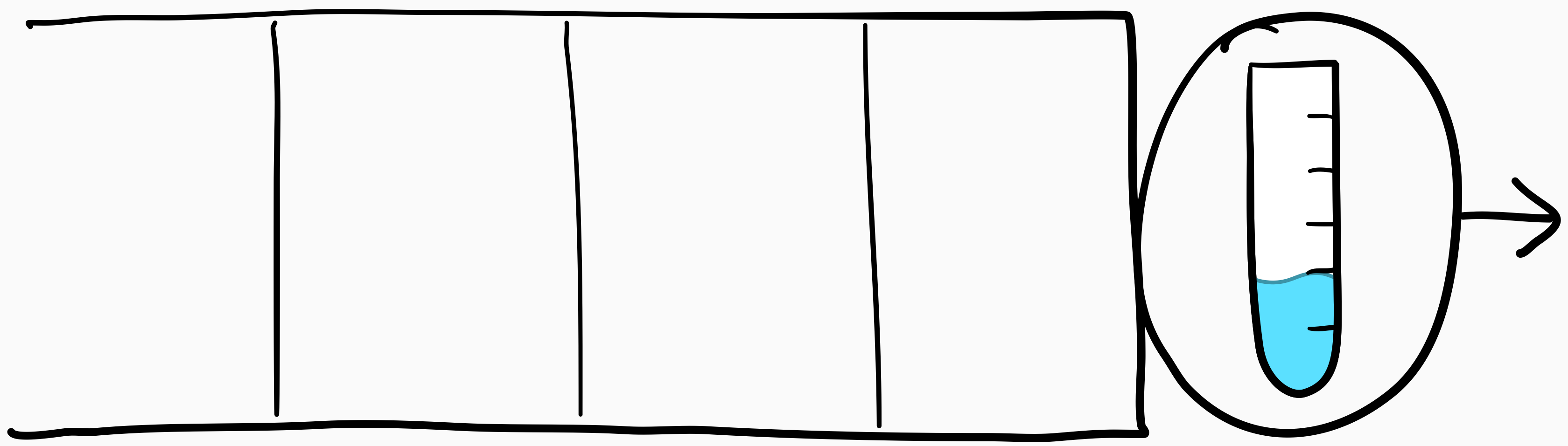
How should we schedule jobs to minimize delay?



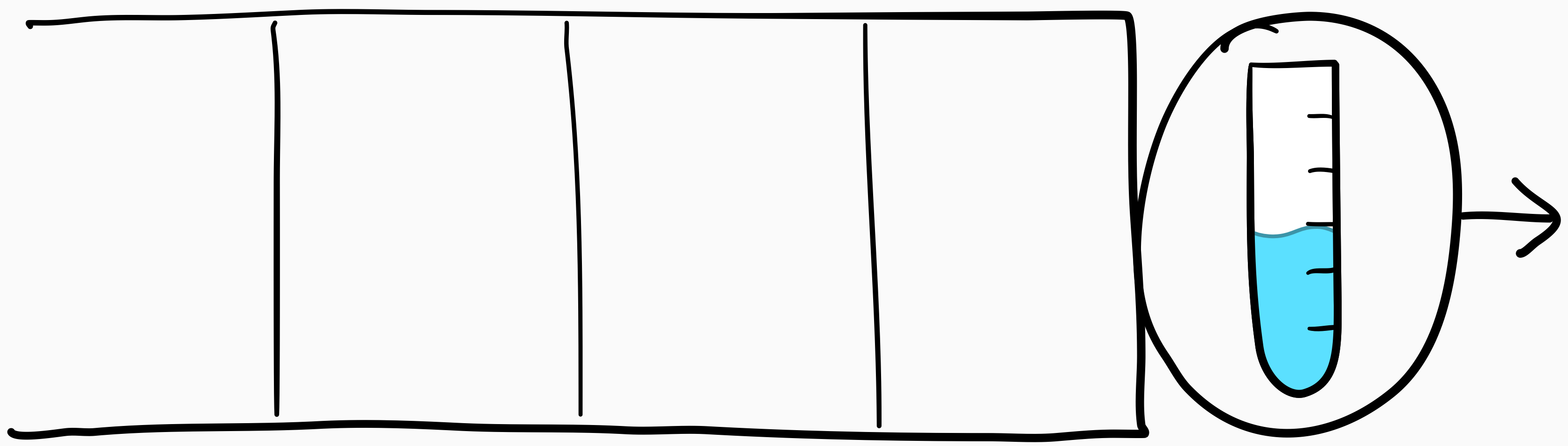
How should we schedule jobs to minimize delay?



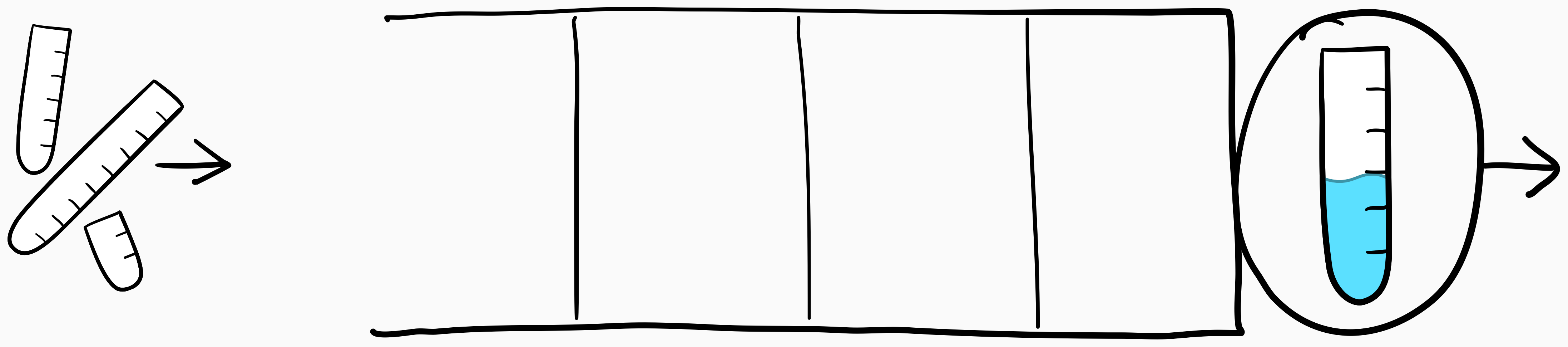
How should we schedule jobs to minimize delay?



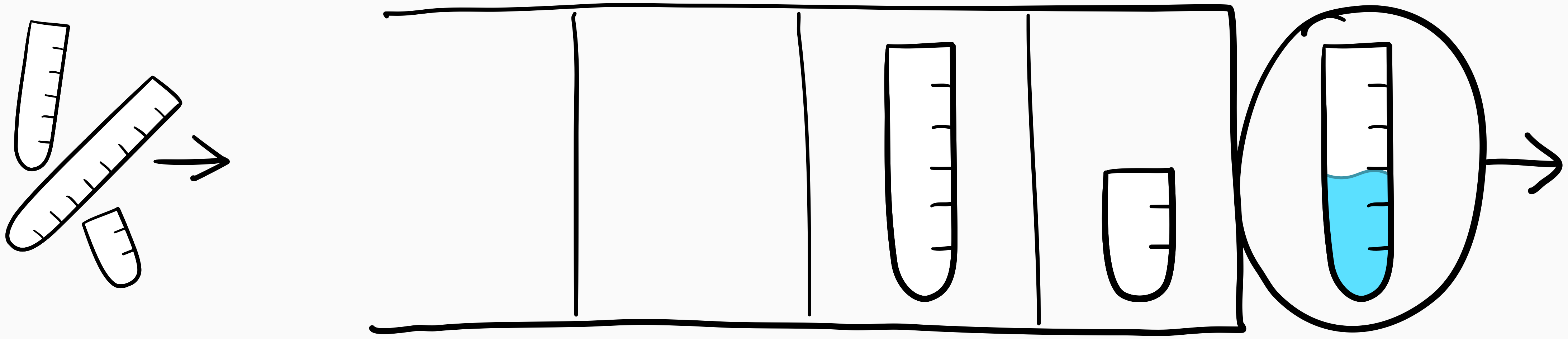
How should we schedule jobs to minimize delay?



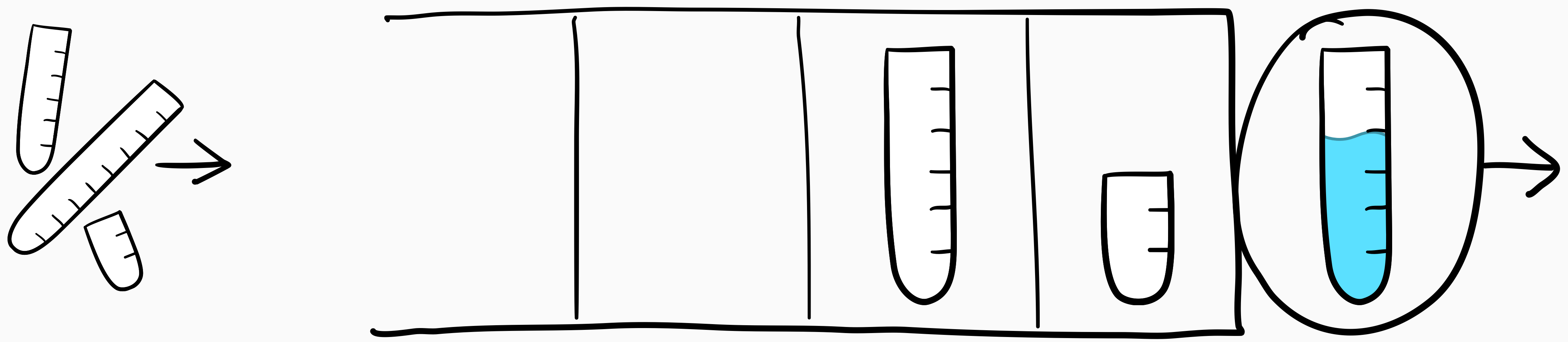
How should we schedule jobs to minimize delay?



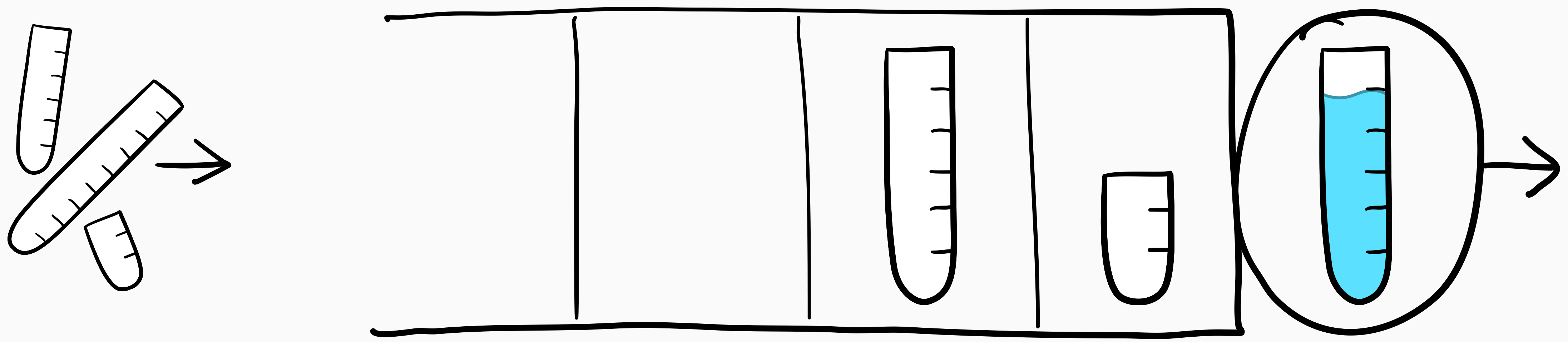
How should we schedule jobs to minimize delay?



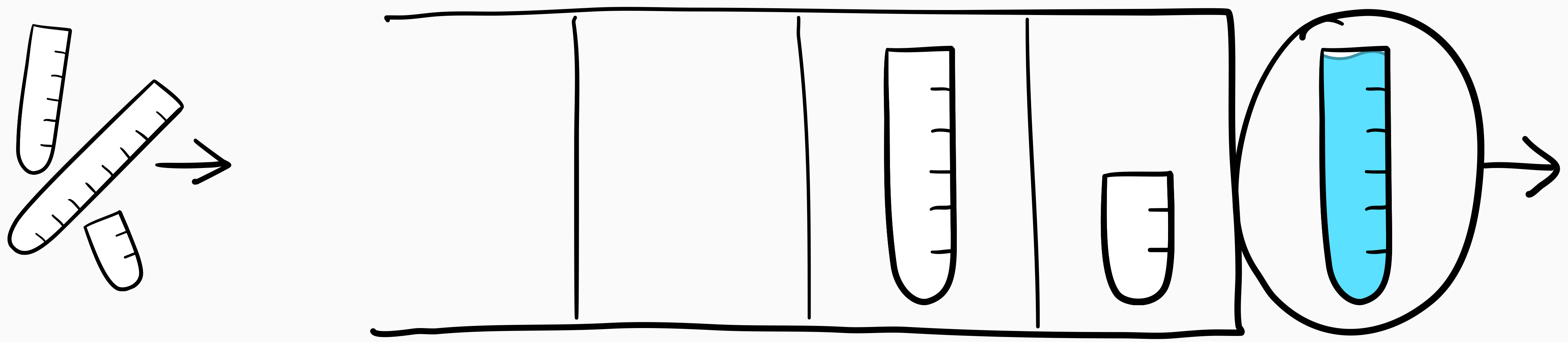
How should we schedule jobs to minimize delay?



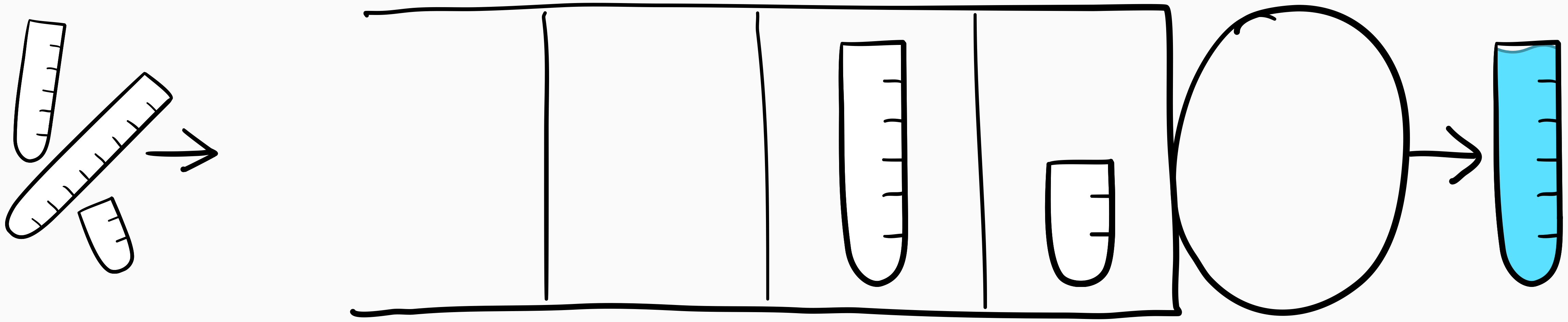
How should we schedule jobs to minimize delay?



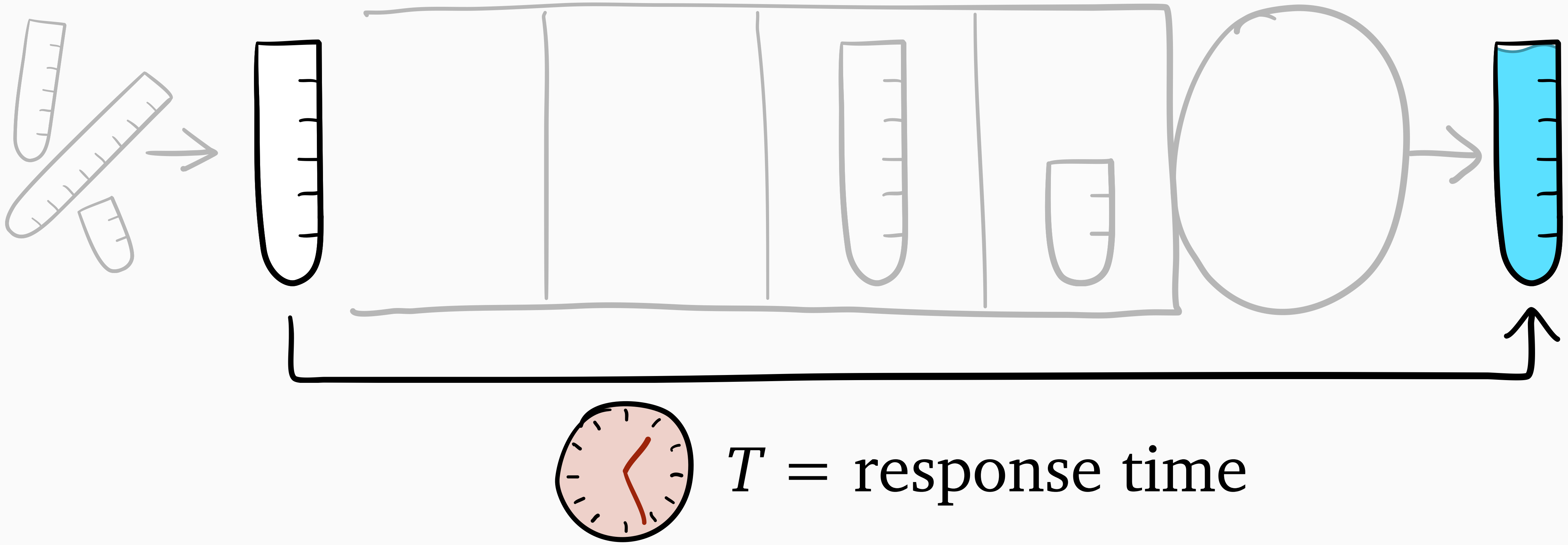
How should we schedule jobs to minimize delay?



How should we schedule jobs to minimize delay?

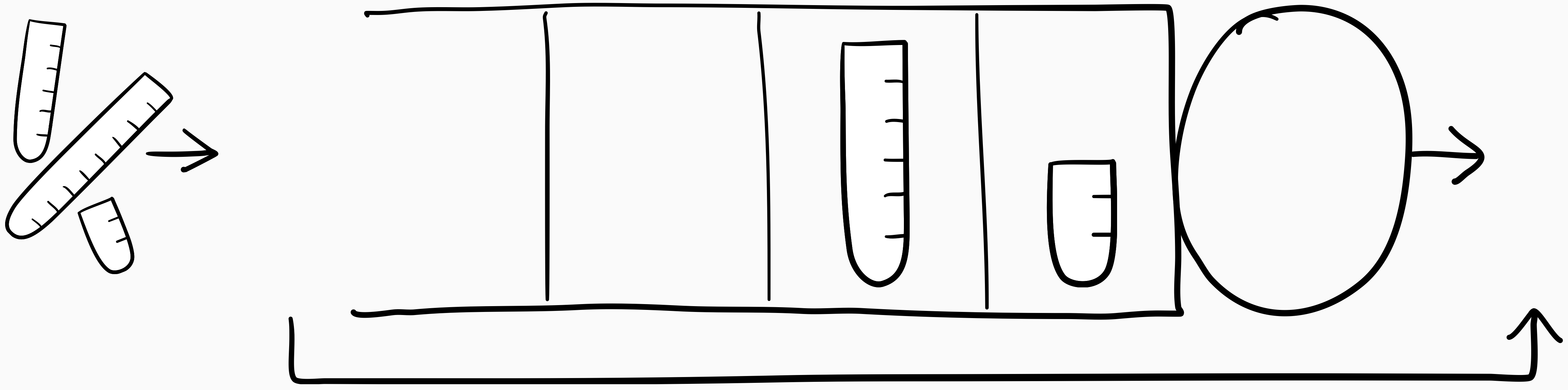


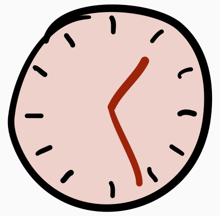
How should we schedule jobs to minimize delay?



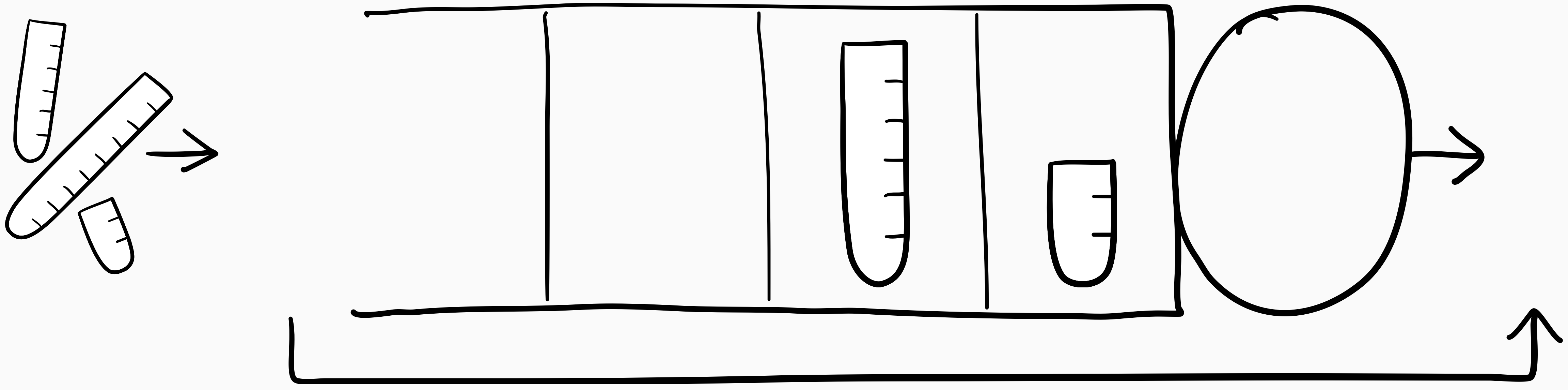
T = response time

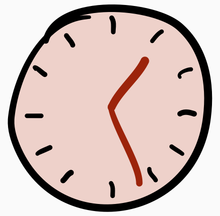
How should we schedule jobs to minimize delay?



 T = response time

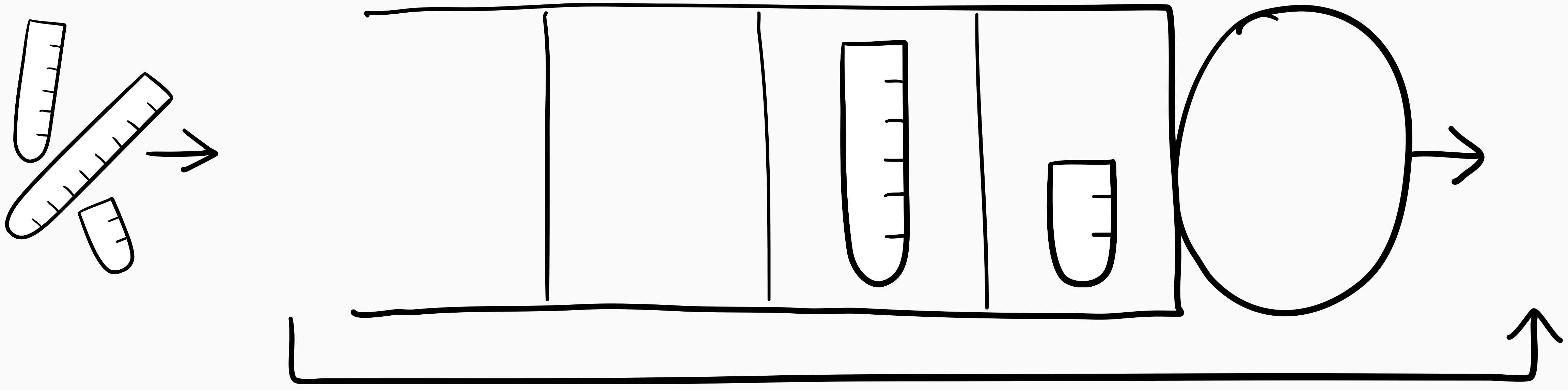
How should we schedule jobs to minimize delay?

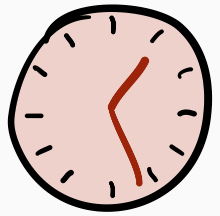


 T = response time

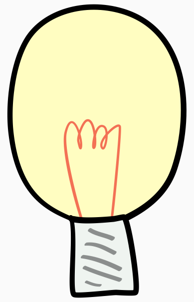
 Minimize $\mathbf{E}[T]$?

How should we schedule jobs to minimize delay?

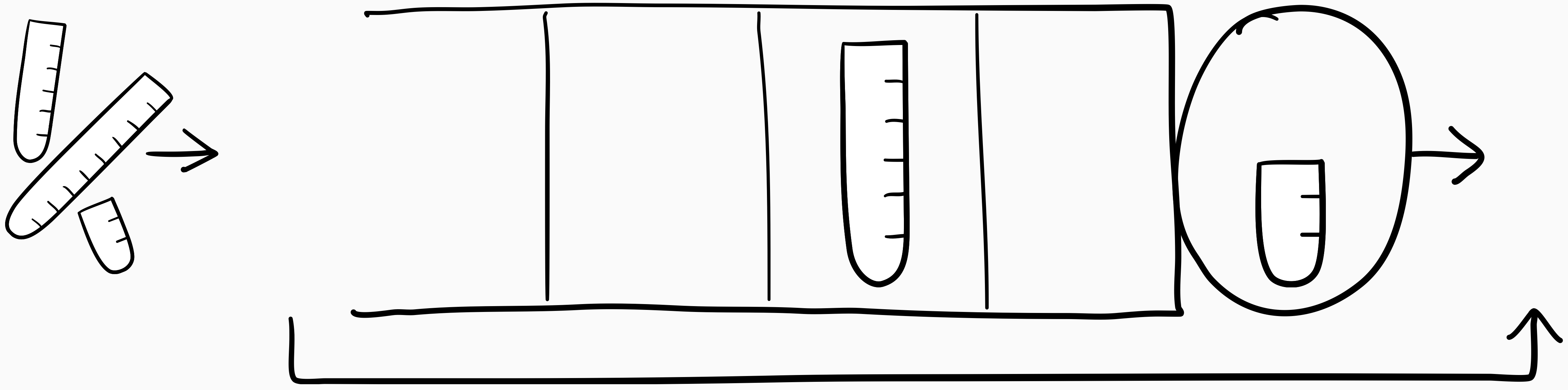


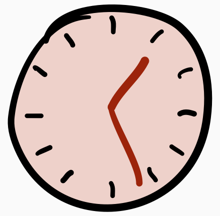
 T = response time

 Minimize $\mathbf{E}[T]$?

 Serve short jobs before long jobs

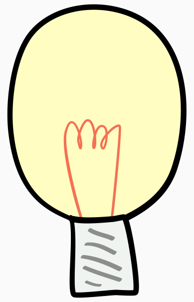
How should we schedule jobs to minimize delay?





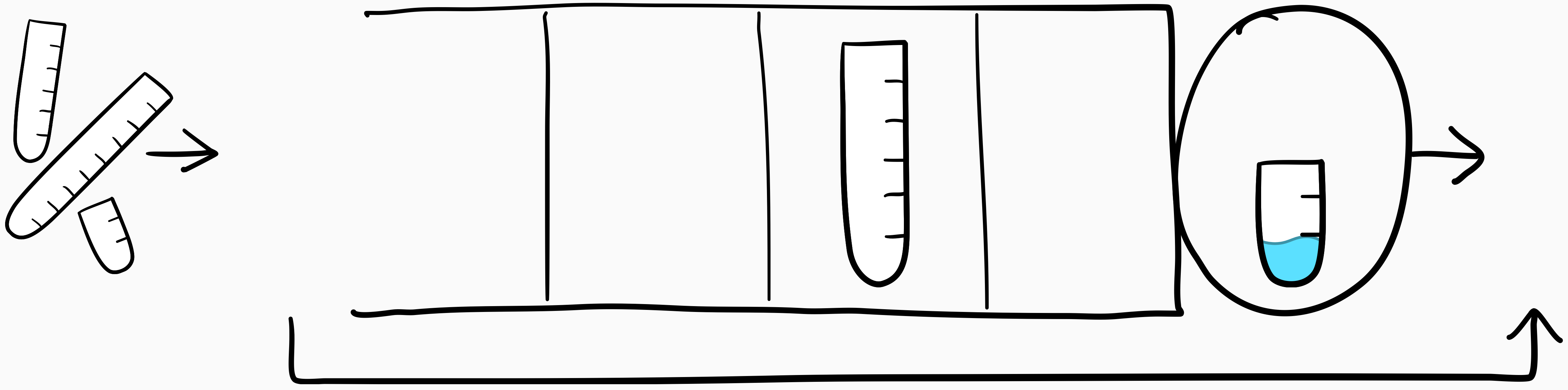
T = response time

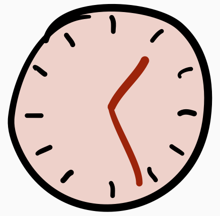
Minimize $\mathbf{E}[T]$?



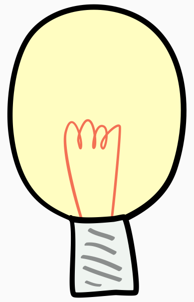
Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?

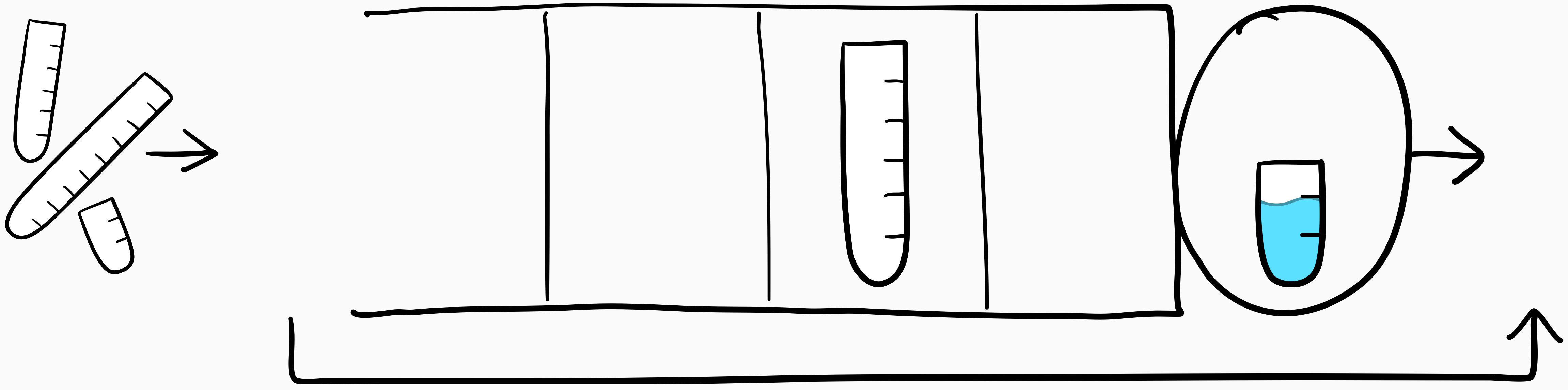


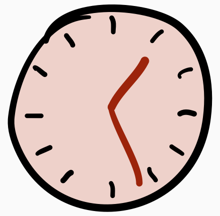
 T = response time

 Minimize $\mathbf{E}[T]$?

 Serve short jobs
before long jobs

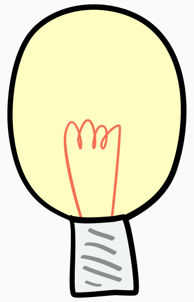
How should we schedule jobs to minimize delay?





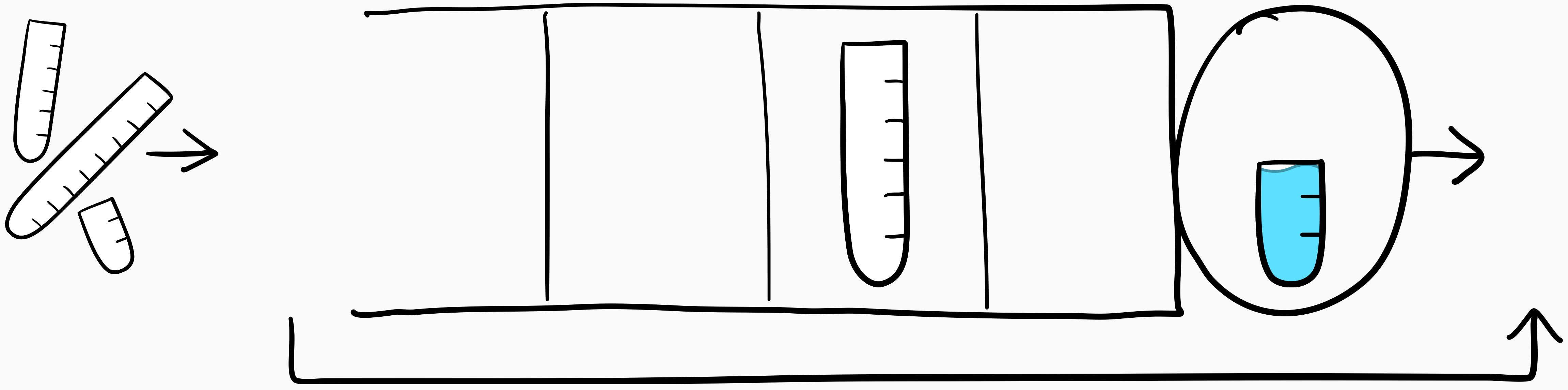
T = response time

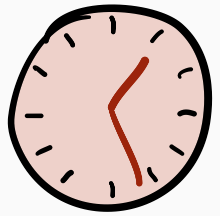
Minimize $\mathbf{E}[T]$?



Serve short jobs
before long jobs

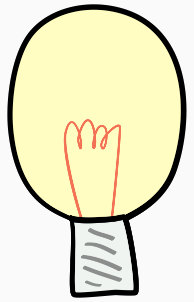
How should we schedule jobs to minimize delay?





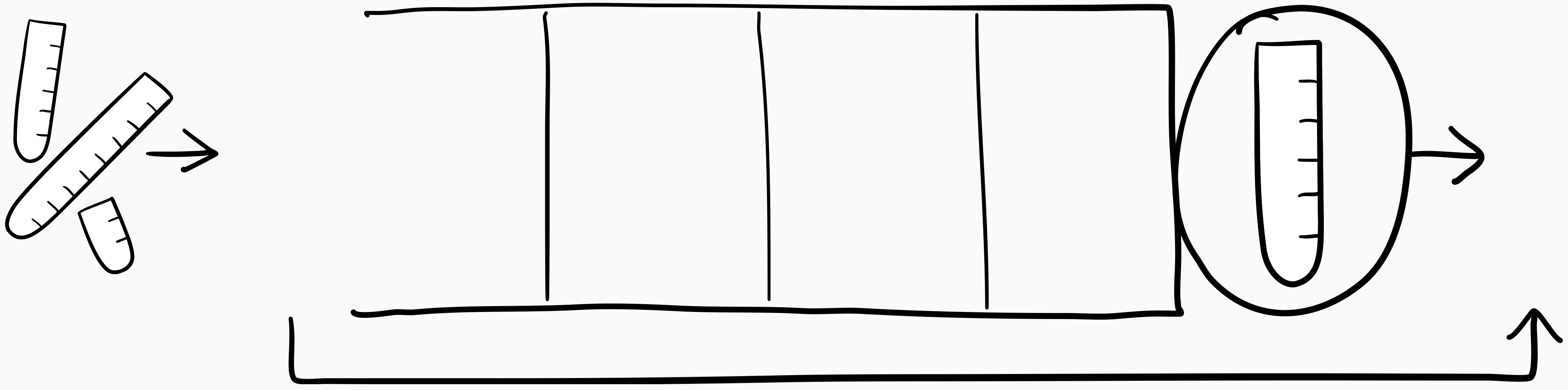
T = response time

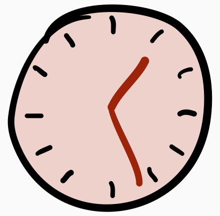
Minimize $\mathbf{E}[T]$?



Serve short jobs
before long jobs

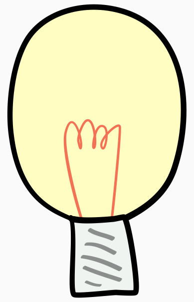
How should we schedule jobs to minimize delay?





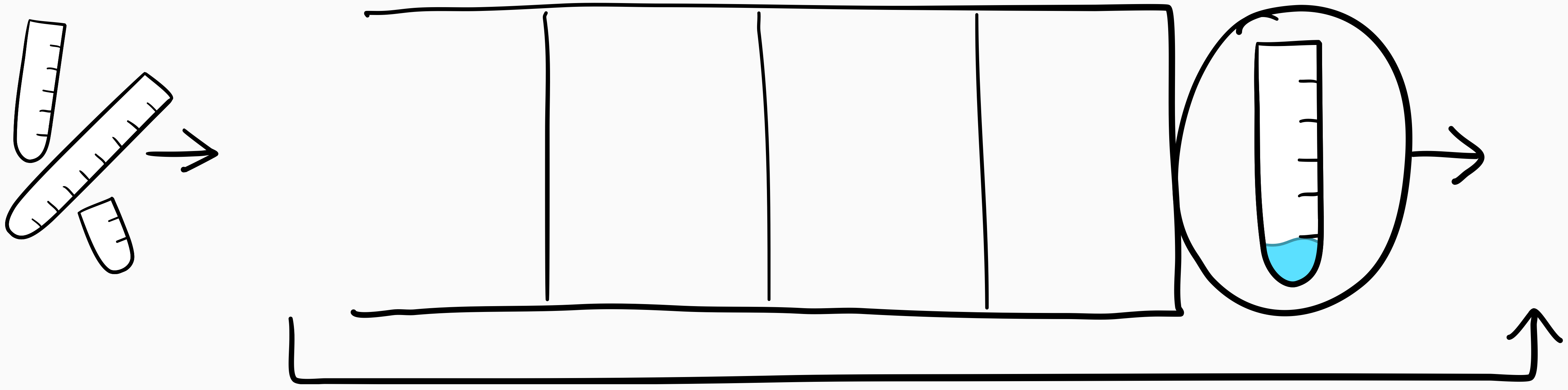
T = response time

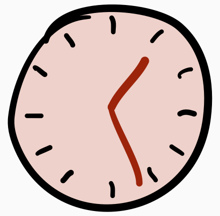
Minimize $\mathbf{E}[T]$?



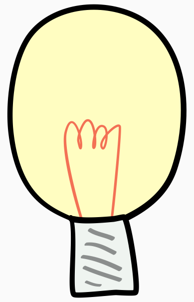
Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?

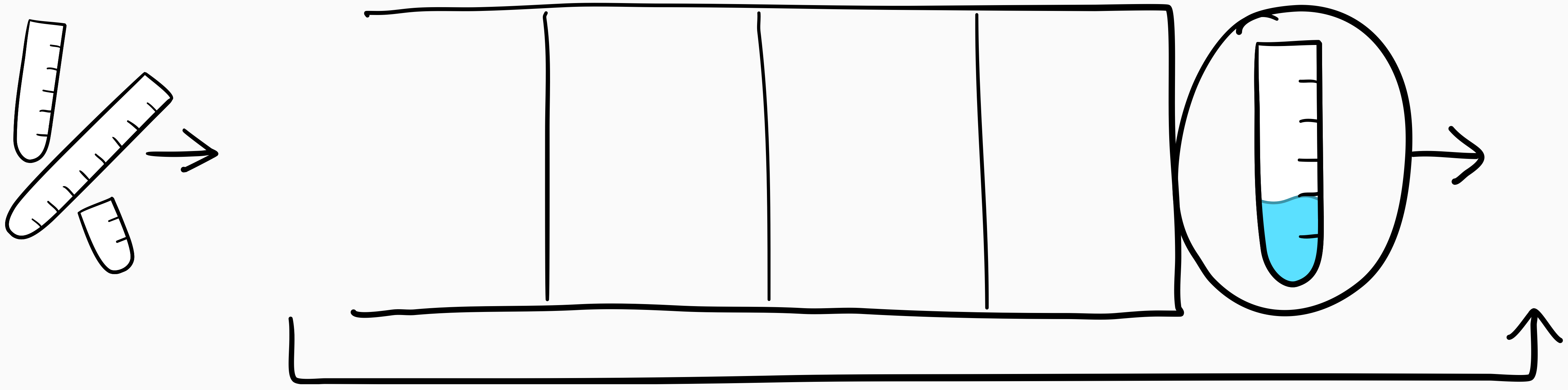


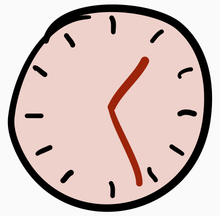
 T = response time

 Minimize $\mathbf{E}[T]$?

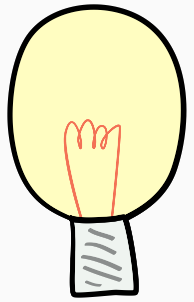
 Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?

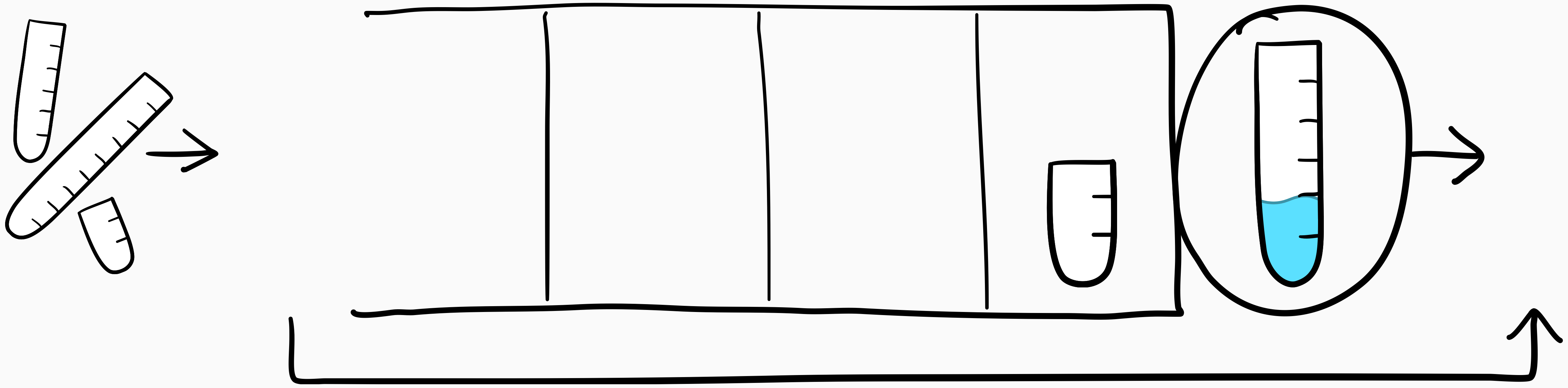


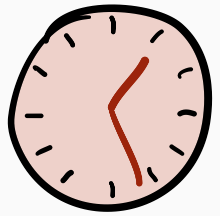
 T = response time

 Minimize $\mathbf{E}[T]$?

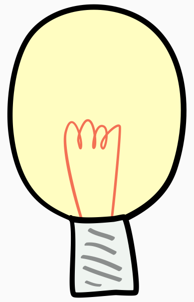
 Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?

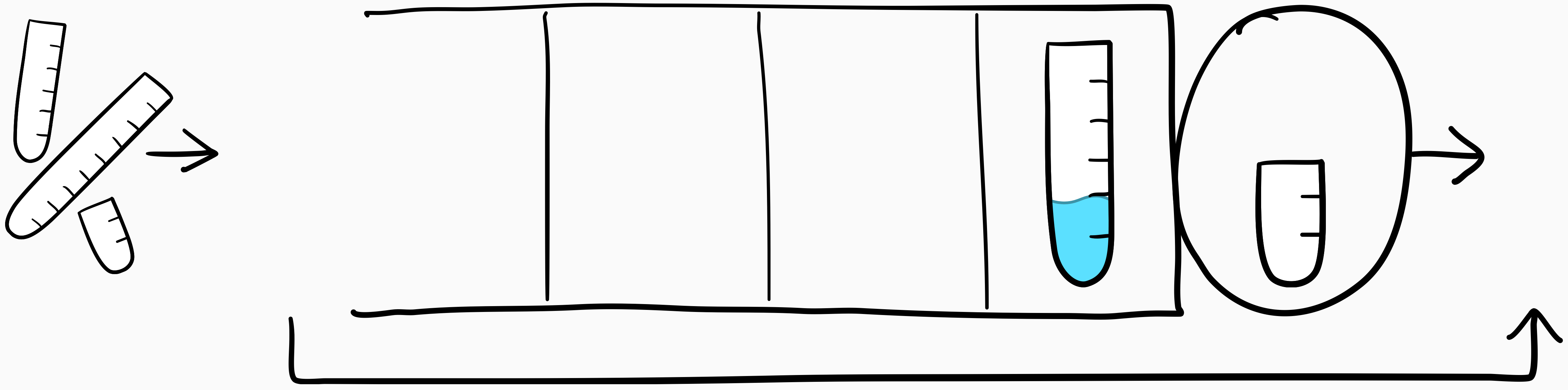


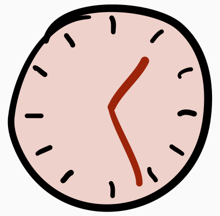

T = response time


Minimize $\mathbf{E}[T]$?

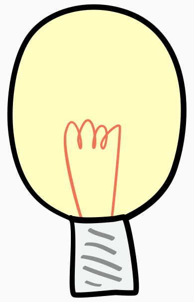

Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?

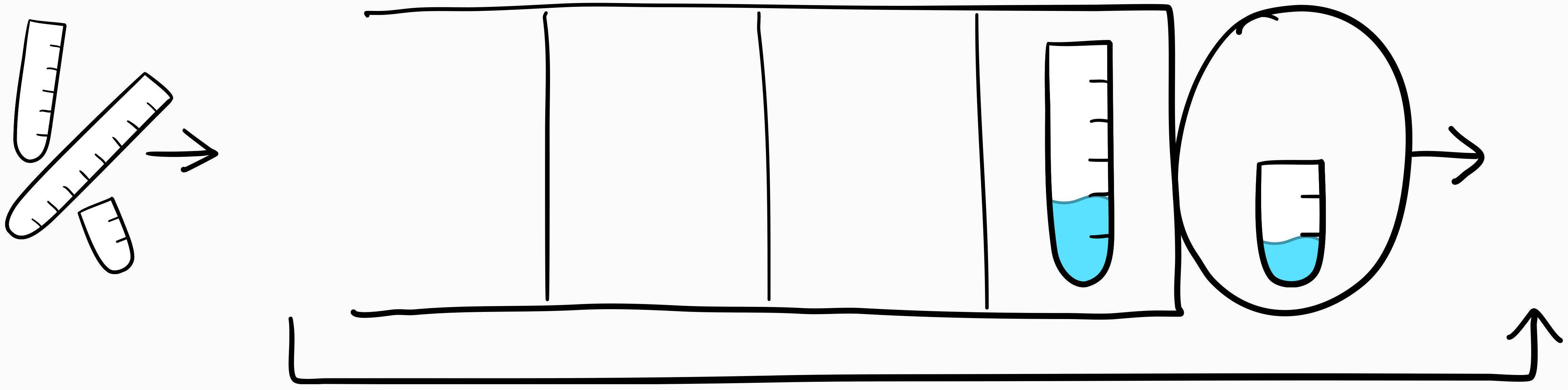


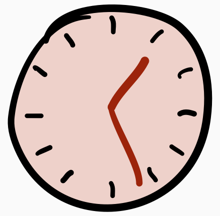
 T = response time

 Minimize $\mathbf{E}[T]$?

 Serve short jobs
before long jobs

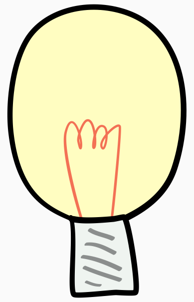
How should we schedule jobs to minimize delay?





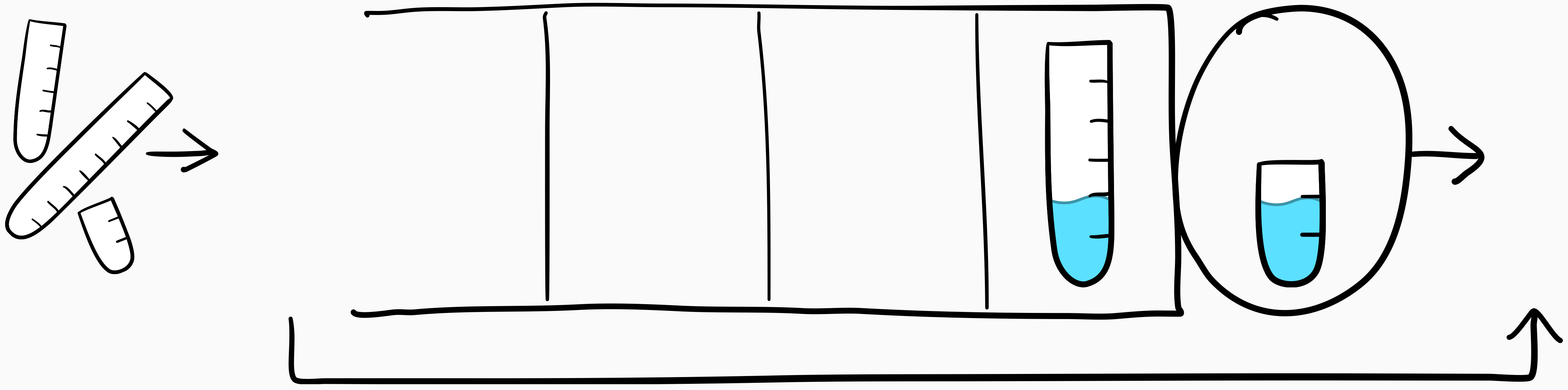
T = response time

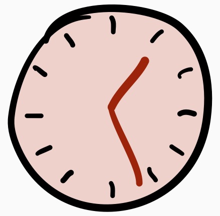
Minimize $\mathbf{E}[T]$?



Serve short jobs
before long jobs

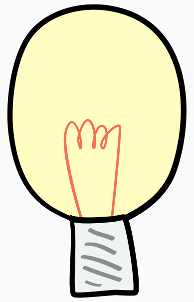
How should we schedule jobs to minimize delay?





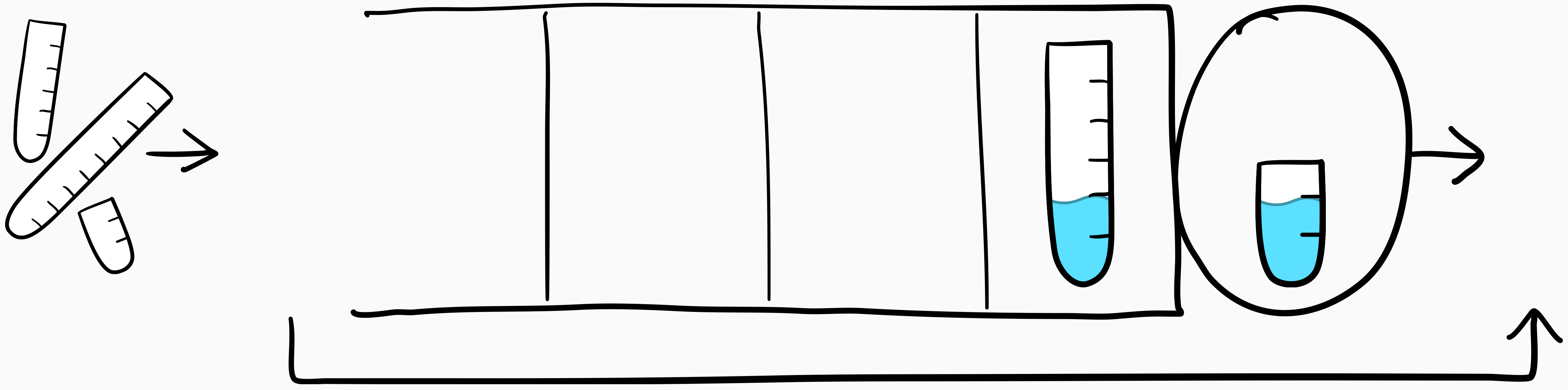
T = response time

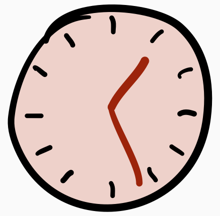
Minimize $\mathbf{E}[T]$?



Serve short jobs
before long jobs

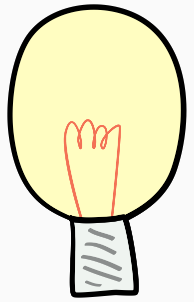
How should we schedule jobs to minimize delay?





T = response time

Minimize $\mathbf{E}[T]$?

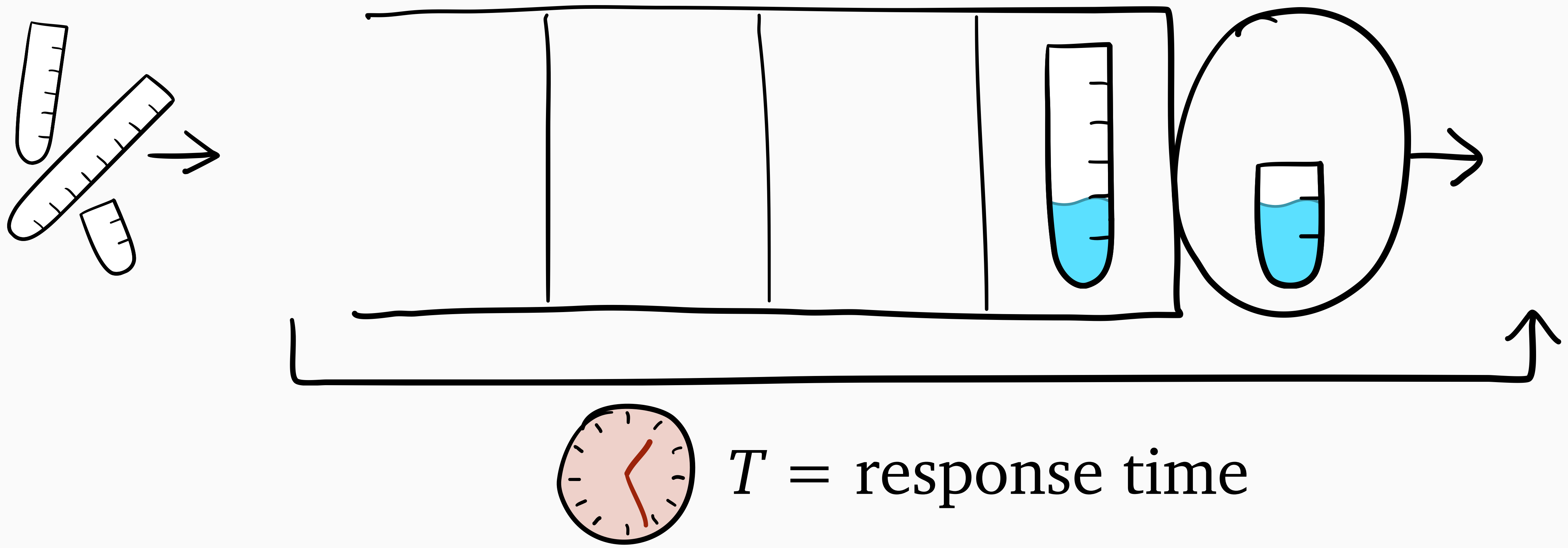


Serve short jobs
before long jobs

shortest remaining
processing time

SRPT: minimizes $\mathbf{E}[T]$

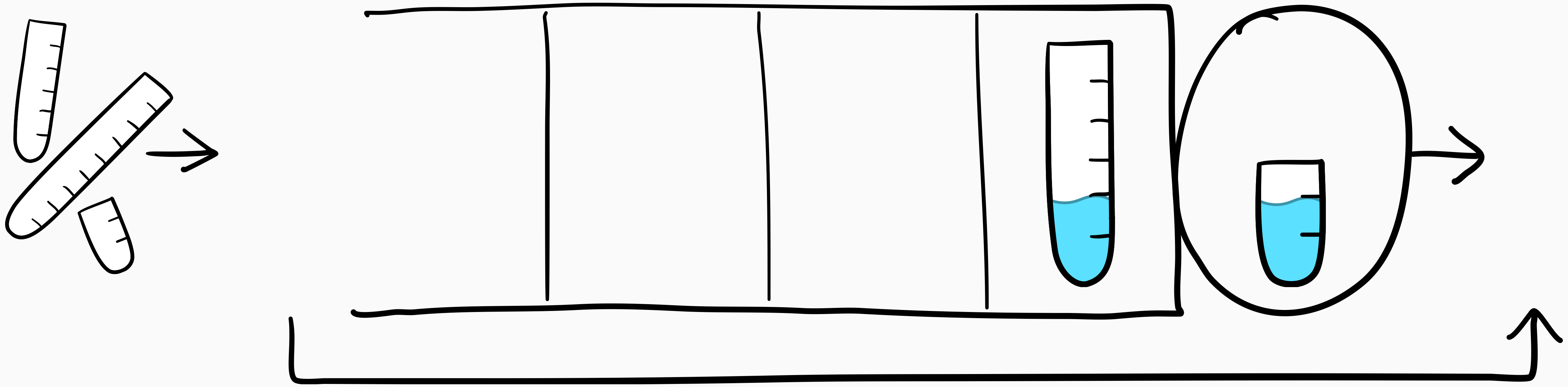
Beyond the mean: tail metrics

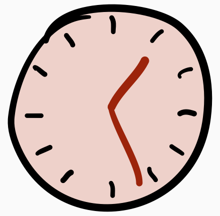


T = response time

Beyond the mean: tail metrics

Minimize $\begin{cases} \mathbf{P}[T > t]? \\ \mathbf{E}[(T - t)^+]? \\ \text{quantiles of } T? \end{cases}$

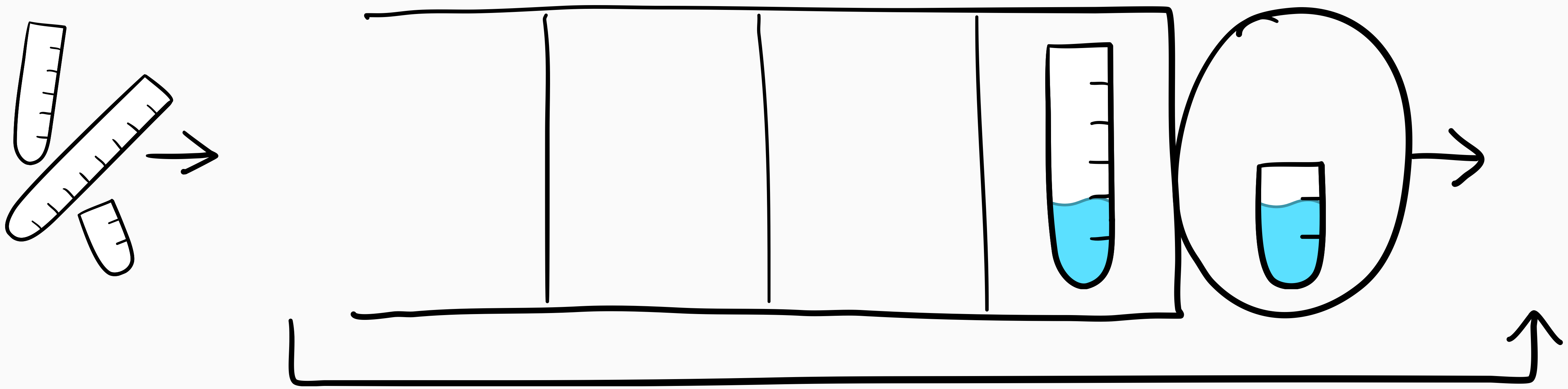


 T = response time

Beyond the mean: tail metrics

Minimize $\begin{cases} \mathbf{P}[T > t]? \\ \mathbf{E}[(T - t)^+]? \\ \text{quantiles of } T? \end{cases}$

Practice: important



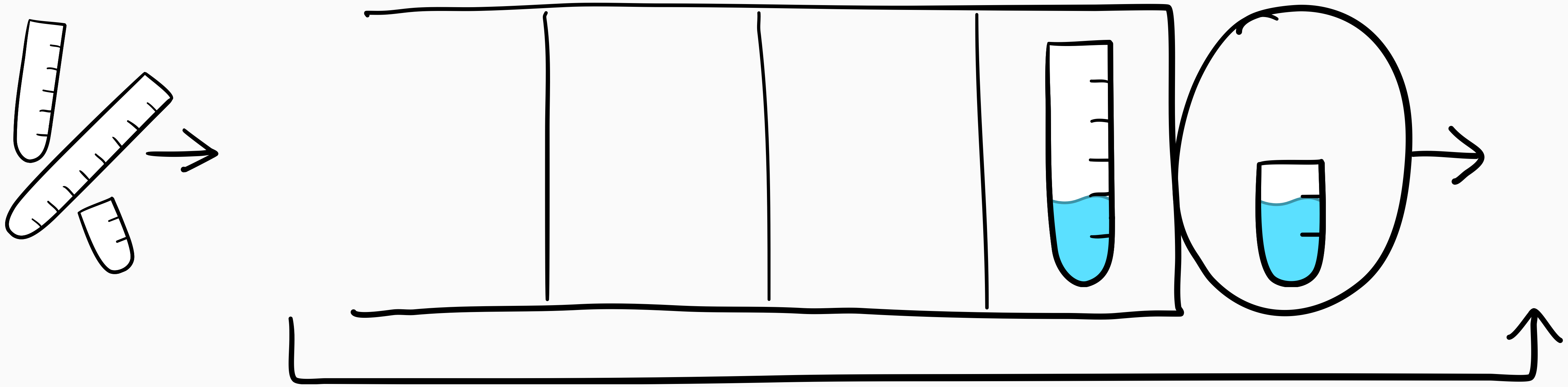
T = response time

Beyond the mean: tail metrics

Minimize $\begin{cases} \mathbf{P}[T > t]? \\ \mathbf{E}[(T - t)^+]? \\ \text{quantiles of } T? \end{cases}$

Practice: important

Theory: very hard



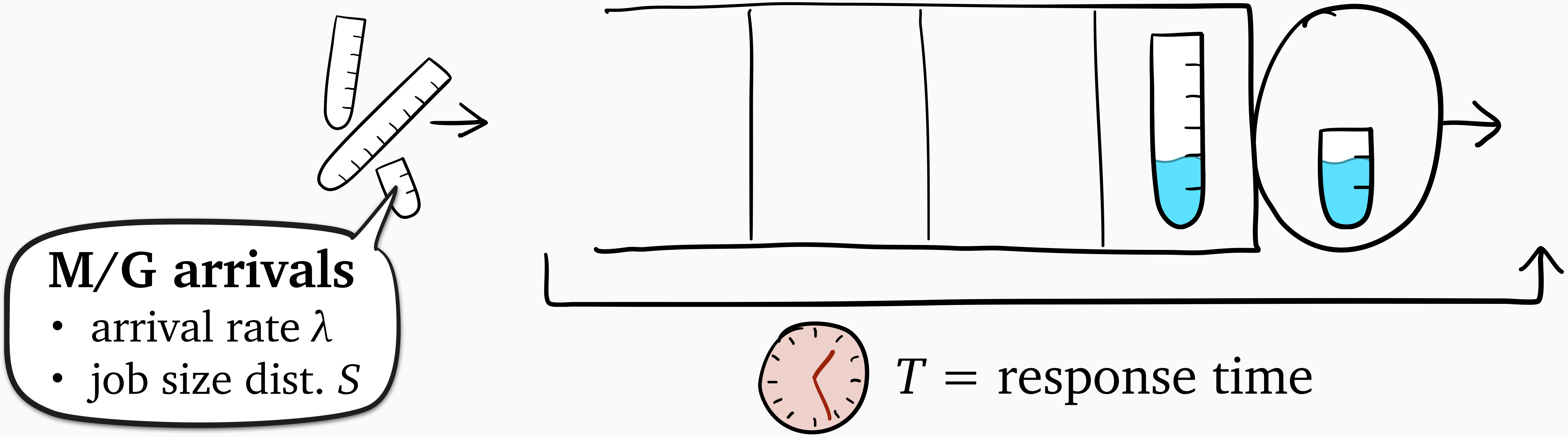
T = response time

Beyond the mean: tail metrics

Minimize $\begin{cases} \mathbf{P}[T > t]? \\ \mathbf{E}[(T - t)^+]? \\ \text{quantiles of } T? \end{cases}$

Practice: important

Theory: very hard



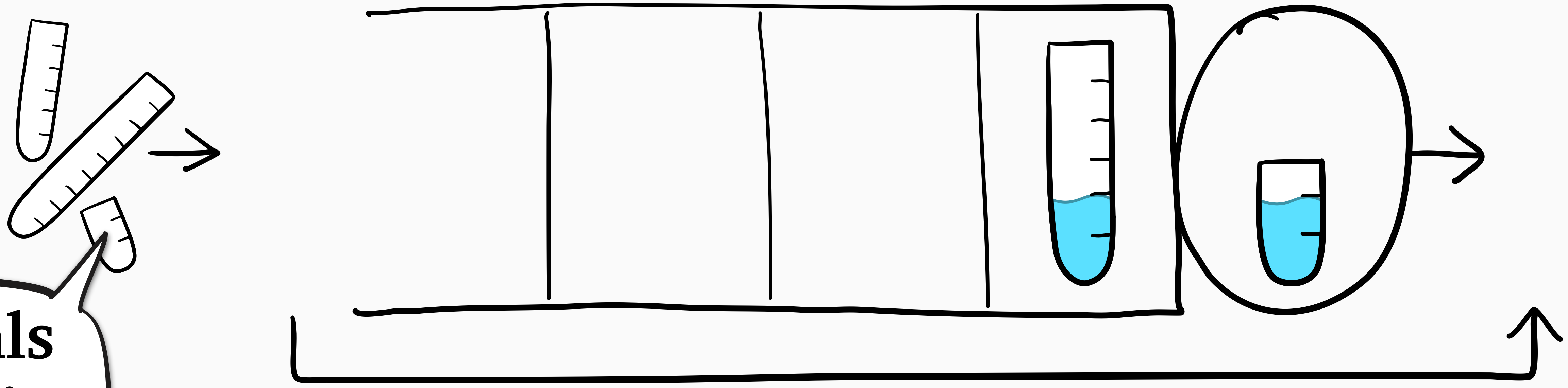
Beyond the mean: tail metrics

Minimize
- $\mathbf{P}[T > t]$?
- $\mathbf{E}[(T - t)^+]$?
- quantiles of T?

Practice: important

Theory: very hard

Tractable: study $t \to \infty$ asymptotics



M/G arrivals
- arrival rate λ
- job size dist. S

T = response time

Optimizing tail asymptotics

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

Optimizing tail asymptotics

Asymptotic tail ratio: $$R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$$

Weakly optimal

$R_\pi < \infty$

Strongly optimal

$R_\pi = 1$

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$		
Strongly optimal $R_\pi = 1$		

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS SRPT PS (processor sharing) LAS (least attained service)	
Strongly optimal $R_\pi = 1$		

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	Light-tailed sizes
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

newish Corollary of prior work
[Wierman & Zwart, 2012]

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	Light-tailed sizes
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

newish Corollary of prior work [Wierman & Zwart, 2012]

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes ✓	**Light-tailed sizes** ?
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served)
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \dfrac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes ✓	Light-tailed sizes ?
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served)
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	FCFS? [Wierman & Zwart, 2012]

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	FCFS? [Wierman & Zwart, 2012]

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	Light-tailed sizes
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	FCFS? [Wierma

Theorem: $R_{\text{Nudge}} < R_{\text{FCFS}}$

Optimizing tail asymptotics

Asymptotic tail ratio: $$R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$$

	Heavy-tailed sizes	Light-tailed sizes
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	FCFS? [Wierma

Theorem: $R_{\text{Nudge}} < R_{\text{FCFS}}$

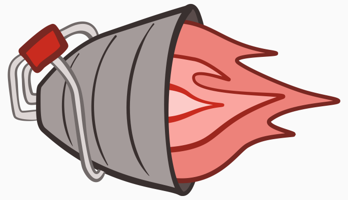
Optimizing tail asymptotics

Asymptotic tail ratio: $$R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$$

	Heavy-tailed sizes	Light-tailed sizes
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

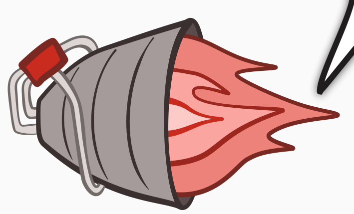
Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes 	Light-tailed sizes 
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	 **Boost** 

Optimizing tail asymptotics

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

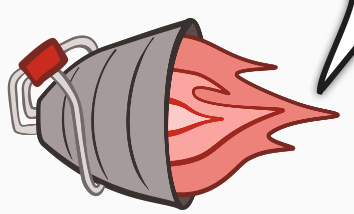
	Heavy-tailed sizes	Light-tailed sizes
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	NEW! **Boost** 

Theorem: $R_{\text{Boost}} = 1$

Optimizing tail asymptotics

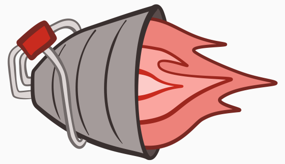
Asymptotic tail ratio: $$R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$$

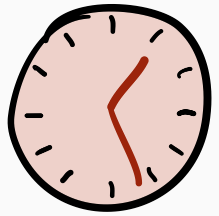
	Heavy-tailed sizes ✓	Light-tailed sizes ✓ NEW!
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	NEW! **Boost**



Theorem: $R_{\text{Boost}} = 1$

Our contributions:

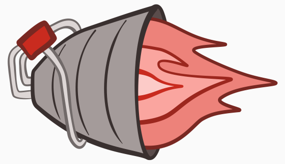
 Design the **Boost** scheduling policy

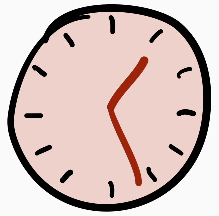
 Analyze **Boost**'s performance

 Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

Our contributions:

actually a *family* of many policies

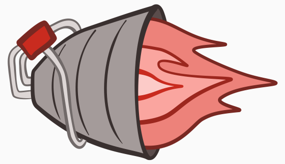

Design the **Boost** scheduling policy


Analyze **Boost**'s performance


Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

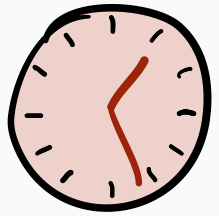
Our contributions:

actually a *family* of many policies



- Design the **Boost** scheduling policy

all instances



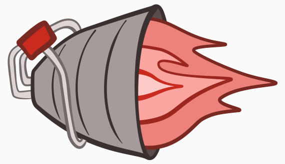
- Analyze **Boost**'s performance

- Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

Our contributions:

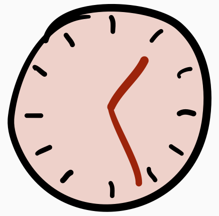
- Design the **Boost** scheduling policy
  - (actually a *family* of many policies)
- Analyze **Boost**'s performance
  - (all instances)
- Prove **Boost** is *strongly tail-optimal* for light-tailed sizes
  - (specific instance called **γ-Boost**)

Our contributions:



Design the **Boost** scheduling policy

actually a *family* of many policies

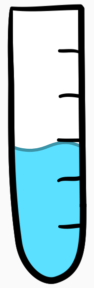


Analyze **Boost**'s performance

all instances

Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

specific instance called **γ-Boost**

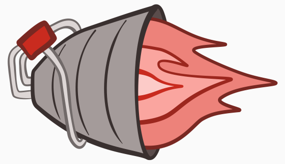


Known job sizes

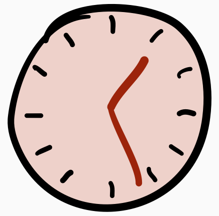
Yu & Scully. *Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1.* SIGMETRICS 2024.

Our contributions:

actually a *family* of many policies

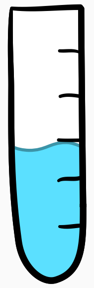
 Design the **Boost** scheduling policy

all instances

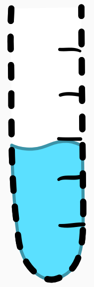
 Analyze **Boost**'s performance

specific instance called **γ-Boost**

 Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

 **Known job sizes**

Yu & Scully. *Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1.* SIGMETRICS 2024.

 **Unknown job sizes**

Harlev, Yu, & Scully. *A Gittins Policy for Optimizing Tail Latency.* MAMA 2024.

Boost

Boost

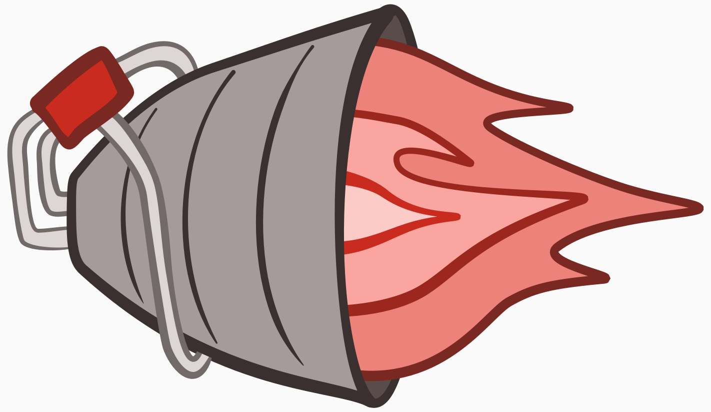
How does the **Boost** policy family work?

Boost

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost



Why is achieving strong tail optimality hard?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost

Why is achieving strong tail optimality hard?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Heavy-tailed sizes

“S Pareto-ish” (regularly varying)

$$\mathbf{P}[S > s] \sim A s^{-\alpha}$$

Light-tailed sizes

“S exponential-ish or lighter” (class I)

$$\mathbf{P}[S > s] \sim A e^{-\alpha s}$$

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim As^{-\alpha}$$

$$\Downarrow$$

$$\mathbf{P}[T > t] \sim Ct^{-\gamma}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim Ae^{-\alpha s}$$

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim As^{-\alpha}$$

$$\mathbf{P}[T > t] \sim Ct^{-\gamma}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim Ae^{-\alpha s}$$

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t}$$

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim A s^{-\alpha}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi t^{-\gamma_\pi}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim A e^{-\alpha s}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi e^{-\gamma_\pi t}$$

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim As^{-\alpha}$$

⇓

$$\mathbf{P}[T_\pi > t] \sim C_\pi t^{-\gamma_\pi}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim Ae^{-\alpha s}$$

⇓

$$\mathbf{P}[T_\pi > t] \sim C_\pi e^{-\gamma_\pi t}$$

$\gamma_\pi =$ *decay rate of* π

$C_\pi =$ *tail constant of* π

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim As^{-\alpha}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi t^{-\gamma_\pi}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim Ae^{-\alpha s}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi e^{-\gamma_\pi t}$$

γ_π = *decay rate* of π

C_π = *tail constant* of π

Weak optimality:
maximize γ_π

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim A s^{-\alpha}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi t^{-\gamma_\pi}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim A e^{-\alpha s}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi e^{-\gamma_\pi t}$$

$\gamma_\pi = \textit{decay rate}$ of π

$C_\pi = \textit{tail constant}$ of π

Weak optimality:
maximize γ_π

Strong optimality:
maximize γ_π, minimize C_π

Heavy-tailed sizes

“S Pareto-ish” (regularly varying)

$$\mathbf{P}[S > s] \sim A s^{-\alpha}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi t^{-\gamma_\pi}$$

Light-tailed sizes

“S exponential-ish or lighter” (class I)

$$\mathbf{P}[S > s] \sim A e^{-\alpha s}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi e^{-\gamma_\pi t}$$

$\gamma_\pi =$ *decay rate* of π

$C_\pi =$ *tail constant* of π

Weak optimality:
maximize γ_π

Strong optimality:
maximize γ_π, minimize C_π

$$R_\pi = \frac{C_\pi}{\inf_{\pi'} C_{\pi'}}$$

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.		
FCFS		

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$	
FCFS		optimal γ

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
Main cause of large T?		

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
Main cause of large T?	**“Catastrophe”** one giant job	

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$ I'm the giant job	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
Main cause of large T?	**"Catastrophe"** one giant job	

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$ I'm the giant job	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$ I'm stuck behind the giant job	optimal γ
Main cause of large T?	**"Catastrophe"** one giant job	

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$ I'm the giant job	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$ I'm stuck behind the giant job	optimal γ
? Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$ I'm the giant job	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$ I'm stuck behind the giant job	optimal γ I see lots of *work* when I arrive
Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$ I'm the giant job	pessimal γ I'm a very big job, lots of smaller jobs are passing me
FCFS	pessimal $\gamma = \alpha - 1$ I'm stuck behind the giant job	optimal γ I see lots of *work* when I arrive
? Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
Main cause of large T?	**“Catastrophe”** one giant job	**“Conspiracy”** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
SRPT or LAS with just two buckets		
Main cause of large T?	**“Catastrophe”** one giant job	**“Conspiracy”** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
SRPT or LAS with just two buckets	pessimal $\gamma = \alpha - 1$	
Main cause of large T?	**“Catastrophe”** one giant job	**“Conspiracy”** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	**optimal** $\gamma = \alpha$	**pessimal** γ
FCFS	**pessimal** $\gamma = \alpha - 1$	**optimal** γ
SRPT or LAS with just two buckets	**pessimal** $\gamma = \alpha - 1$	**intermediate** γ
Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc.	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
SRPT or LAS with just two buckets	pessimal $\gamma = \alpha - 1$	intermediate γ
Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

I'm in bucket 2, lots of bucket 1 jobs are passing me

Background on decay rates

	Heavy-tailed sizes	Light-taile
SRPT, LAS, etc.	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
SRPT or LAS with just two buckets	pessimal $\gamma = \alpha - 1$	intermediate γ
Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

I'm a very big job, lots of smaller jobs are passing me

I'm in bucket 2, lots of bucket 1 jobs are passing me

Background on decay rates

	Light-taile
SRPT, LAS, etc.	pessimal γ
FCFS	optimal γ
SRPT or LAS with just two buckets	intermediate γ
Main cause of large T?	**“Conspiracy”** lots of biggish jobs

I’m a very big job, lots of smaller jobs are passing me

I’m in bucket 2, lots of bucket 1 jobs are passing me

Background on decay rates

	Light-tailed
SRPT, LAS, etc.	pessimal γ
FCFS	optimal γ
SRPT or LAS with just two buckets	intermediate γ
Main cause of large T?	**"Conspiracy"** lots of biggish jobs

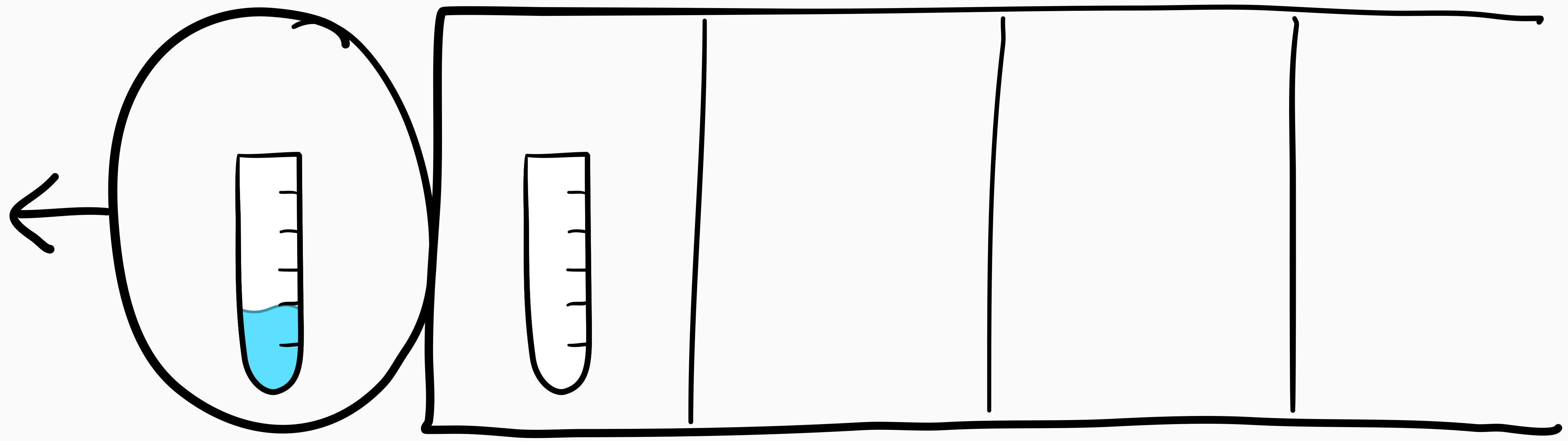
I'm a very big job, lots of smaller jobs are passing me

I'm in bucket 2, lots of bucket 1 jobs are passing me

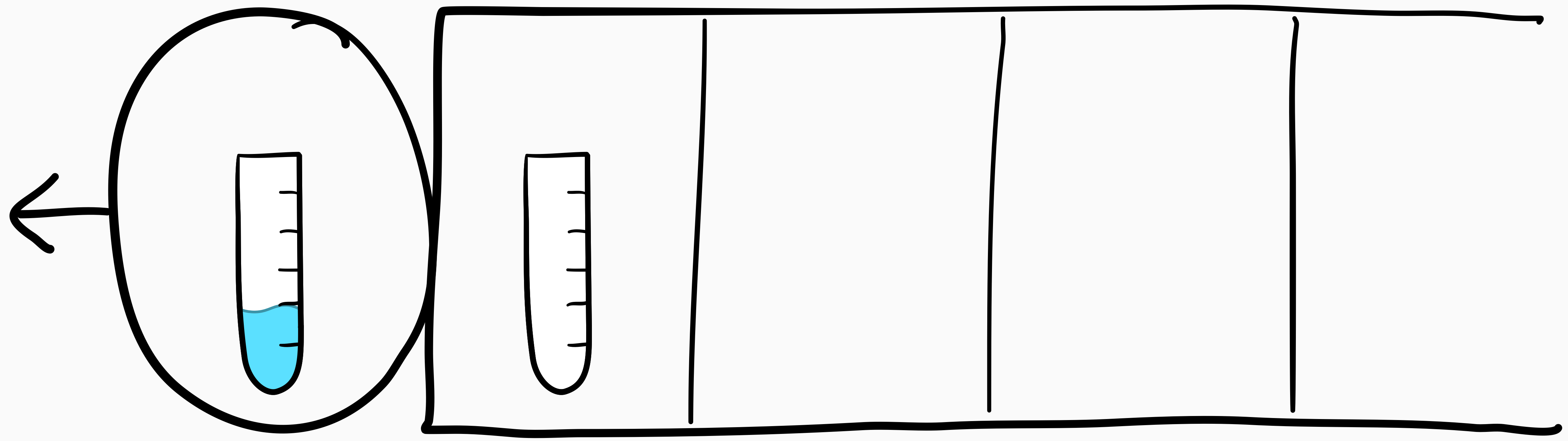
Takeaway: for optimality, must *avoid strict priorities*

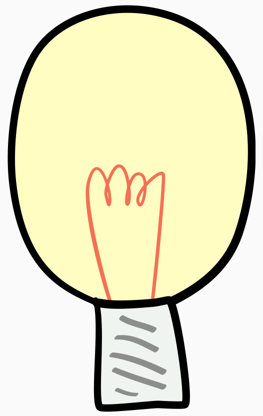
Can we beat FCFS?

Can we beat FCFS?



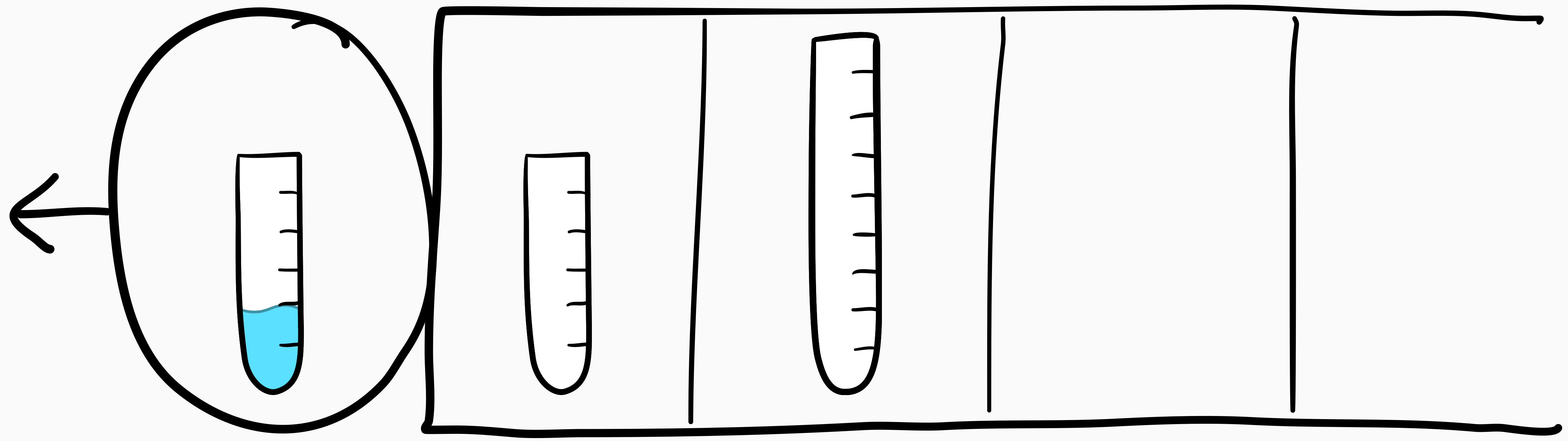
Can we beat FCFS?

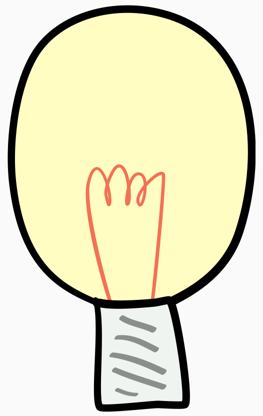




Nudge [Grosof et al., 2021]

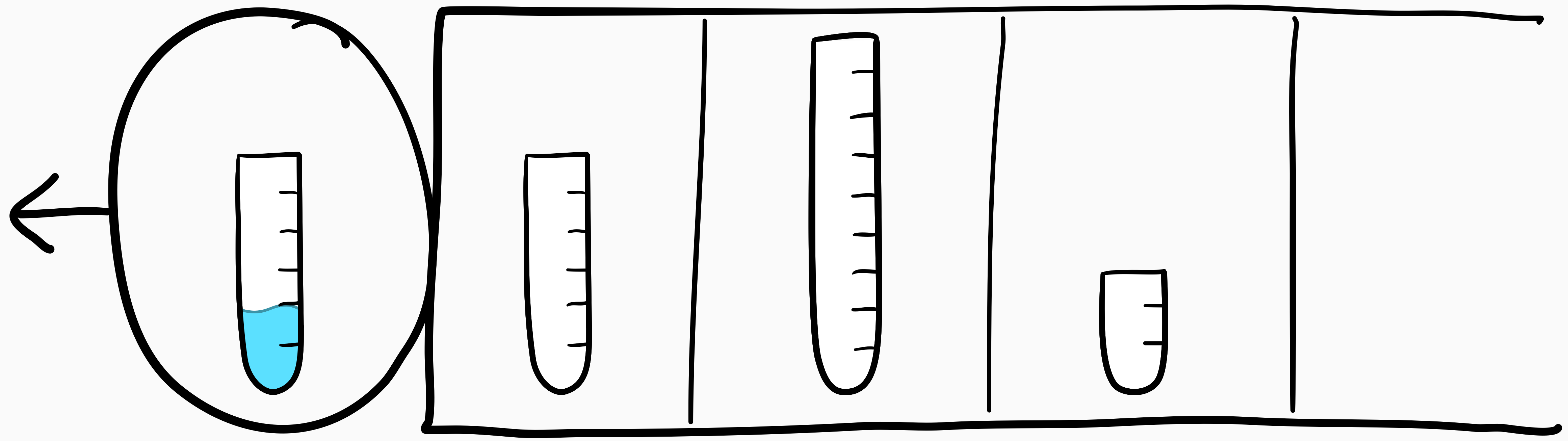
Can we beat FCFS?

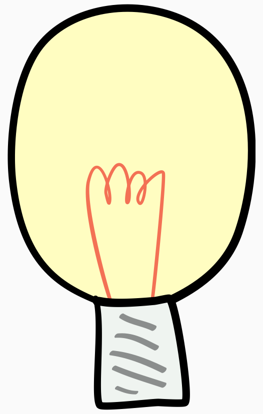




Nudge [Grosof et al., 2021]

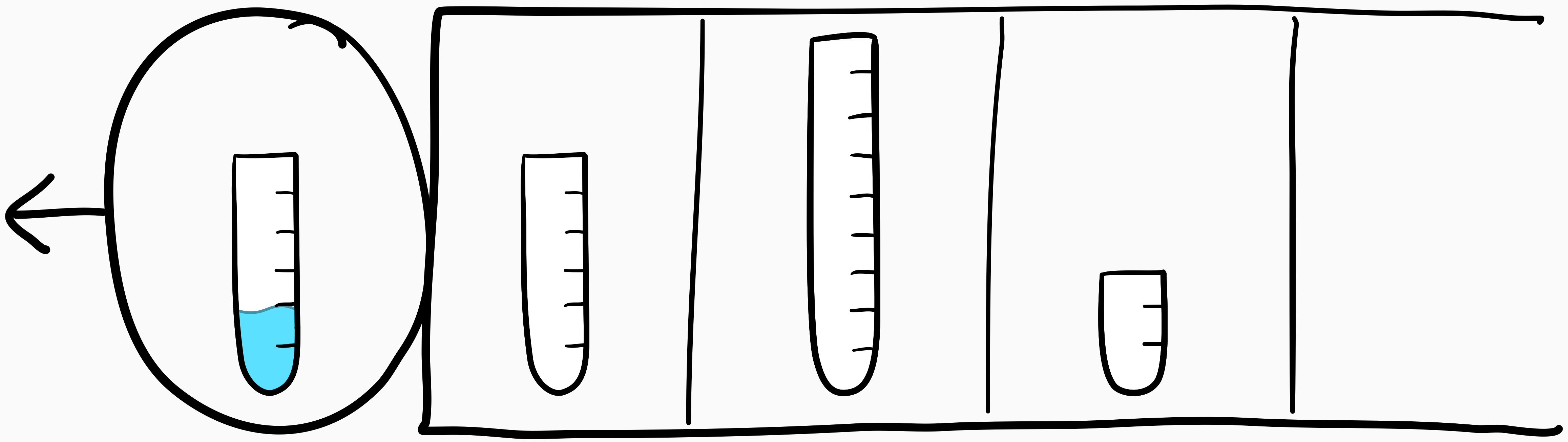
Can we beat FCFS?

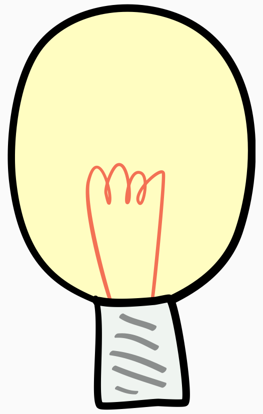




Nudge [Grosof et al., 2021]

Can we beat FCFS?

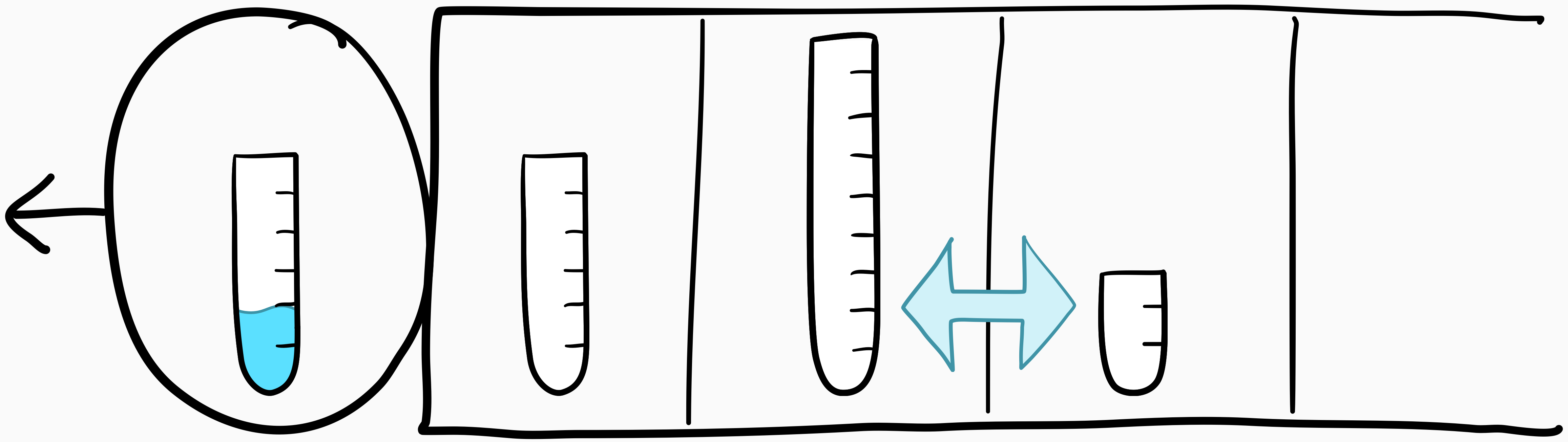


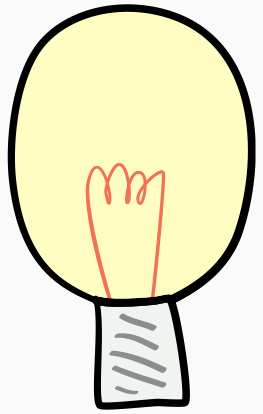


Nudge [Grosof et al., 2021]

- small job can pass one large job

Can we beat FCFS?

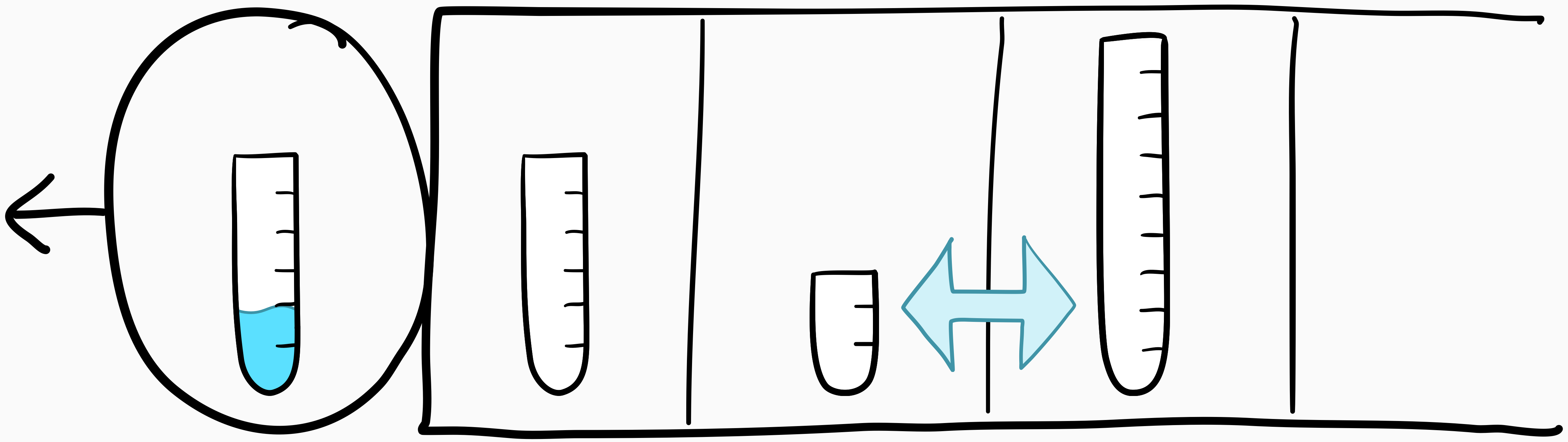


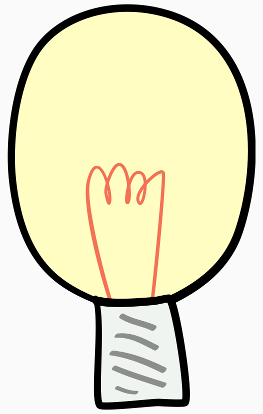


Nudge [Grosof et al., 2021]

- small job can pass one large job

Can we beat FCFS?

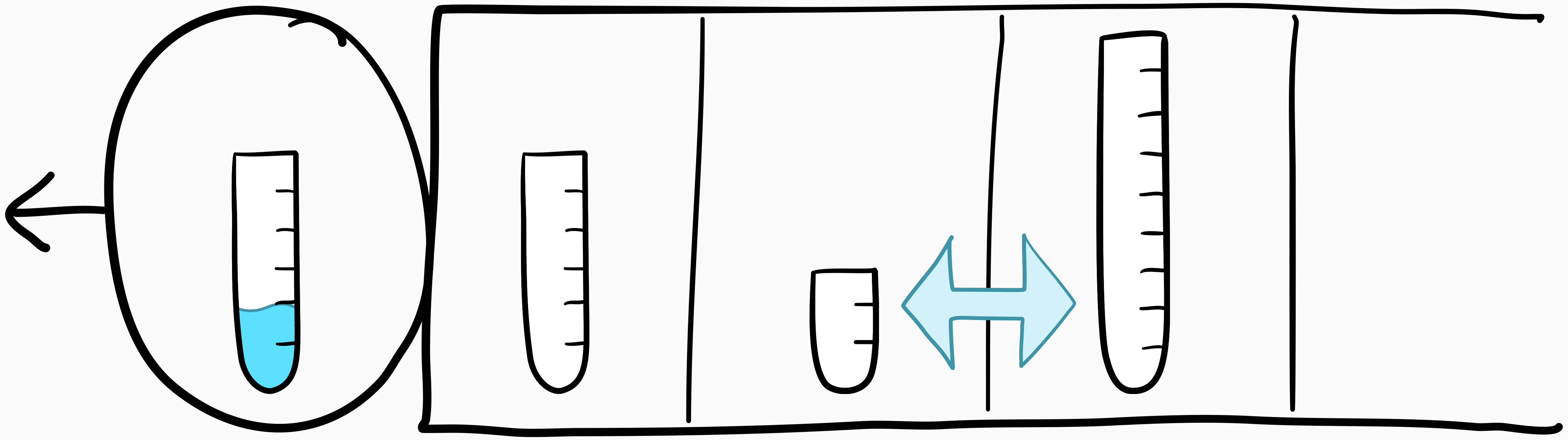


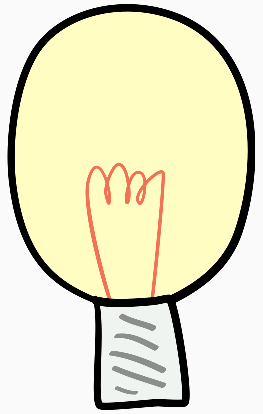


Nudge [Grosof et al., 2021]

- small job can pass one large job

Can we beat FCFS?

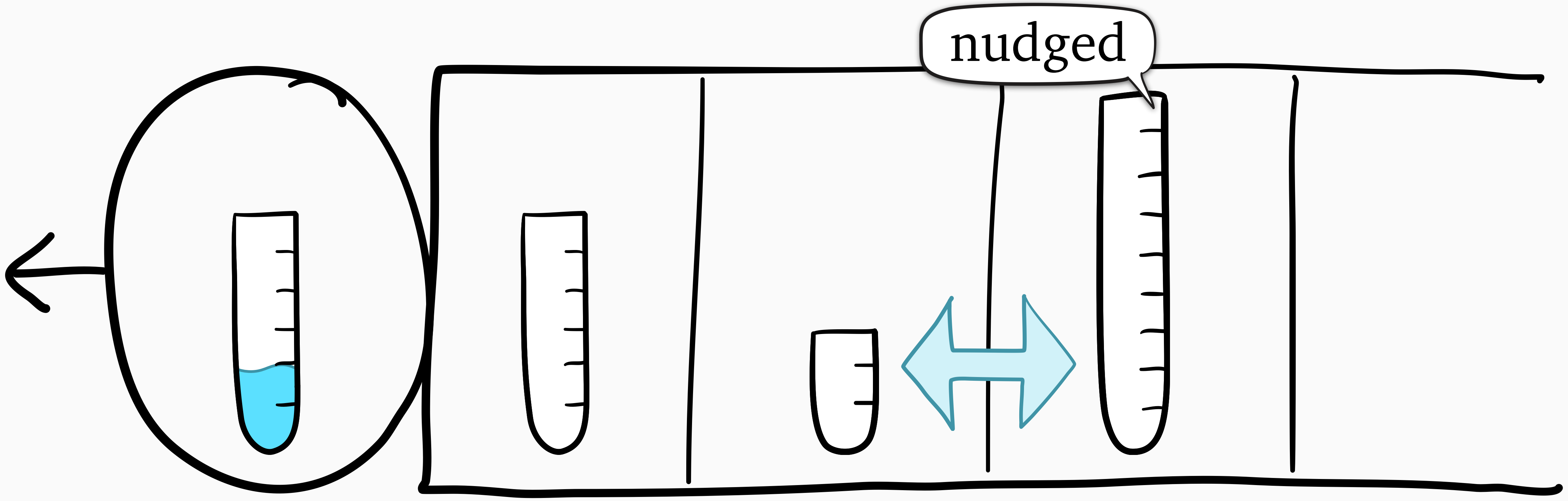


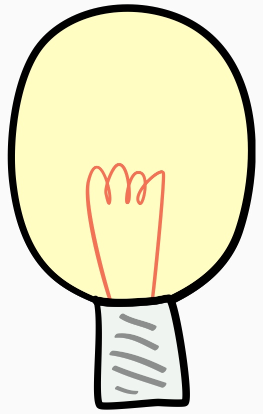


Nudge [Grosof et al., 2021]

- small job can pass one large job
- large job can't be passed twice

Can we beat FCFS?

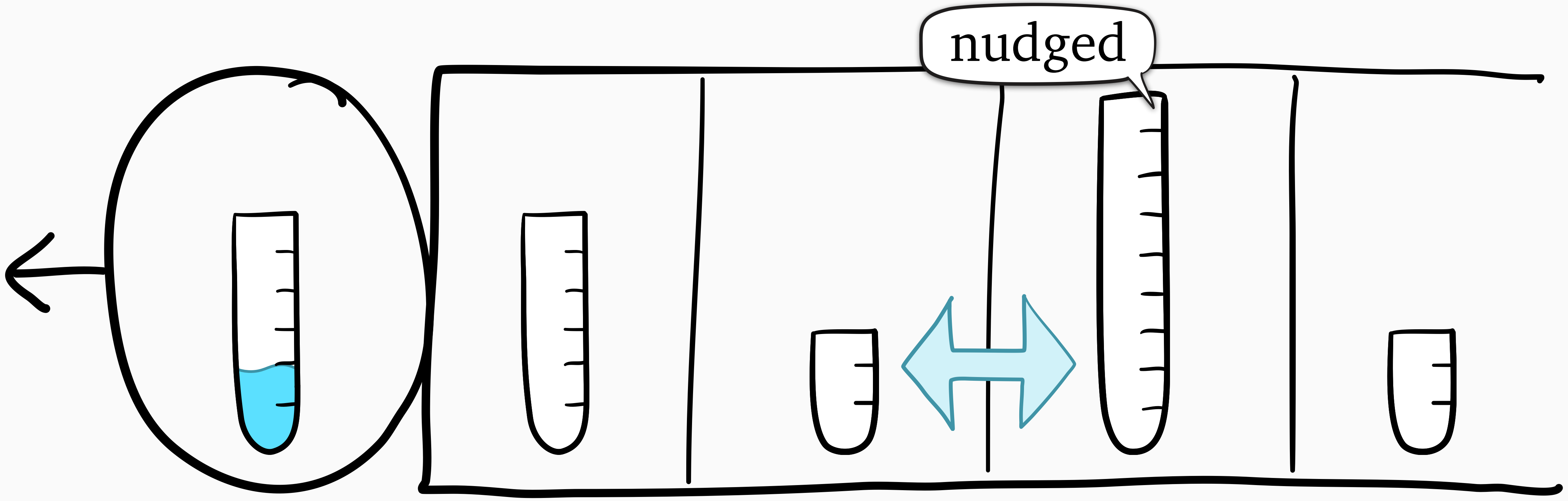


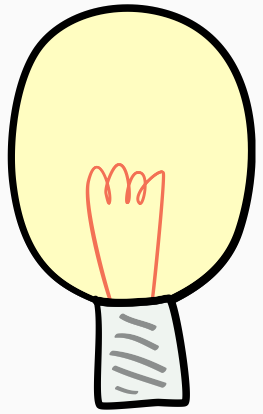


Nudge [Grosof et al., 2021]

- small job can pass one large job
- large job can't be passed twice

Can we beat FCFS?

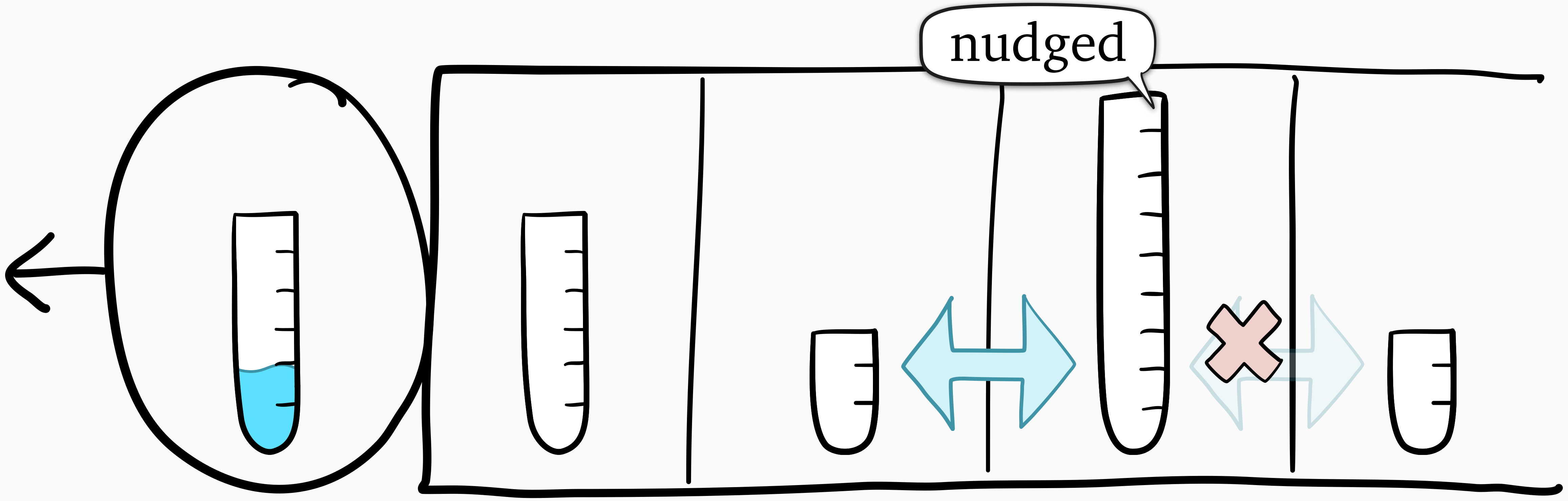


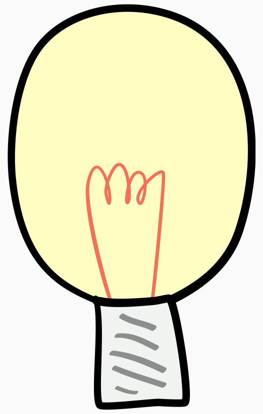


Nudge [Grosof et al., 2021]

- small job can pass one large job
- large job can't be passed twice

Can we beat FCFS?

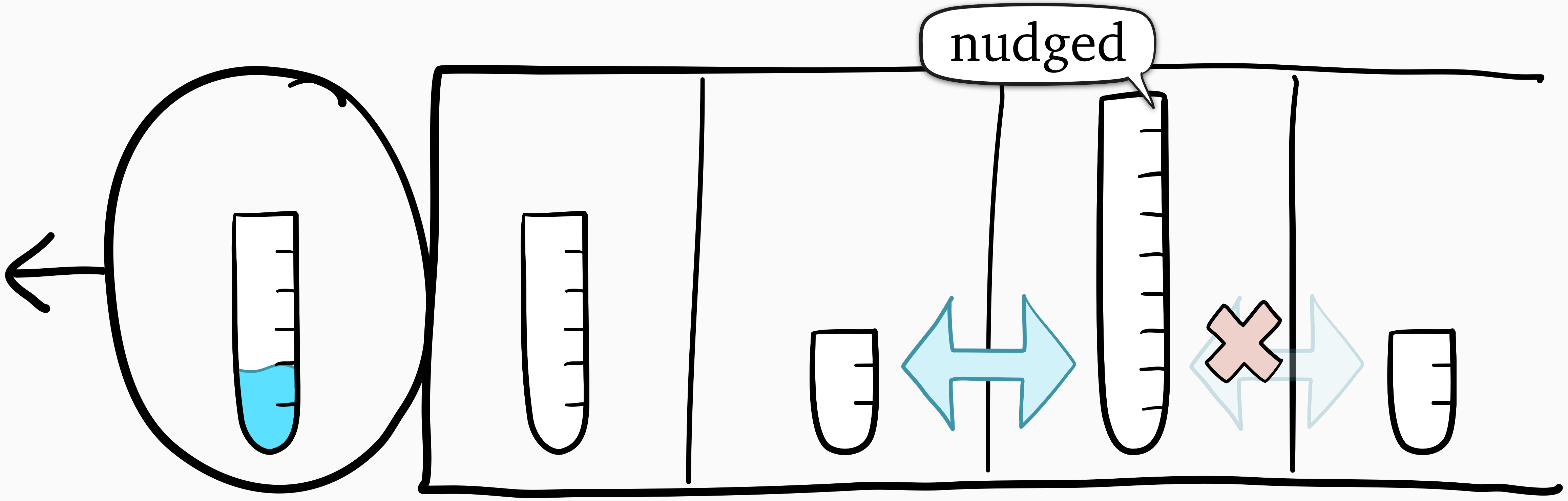


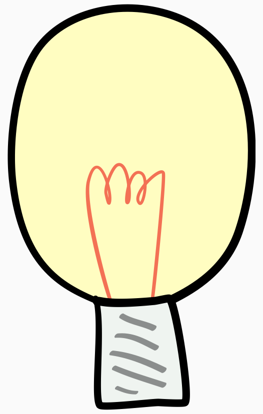


Nudge [Grosof et al., 2021]

- small job can pass one large job
- large job can't be passed twice

Can we beat FCFS?





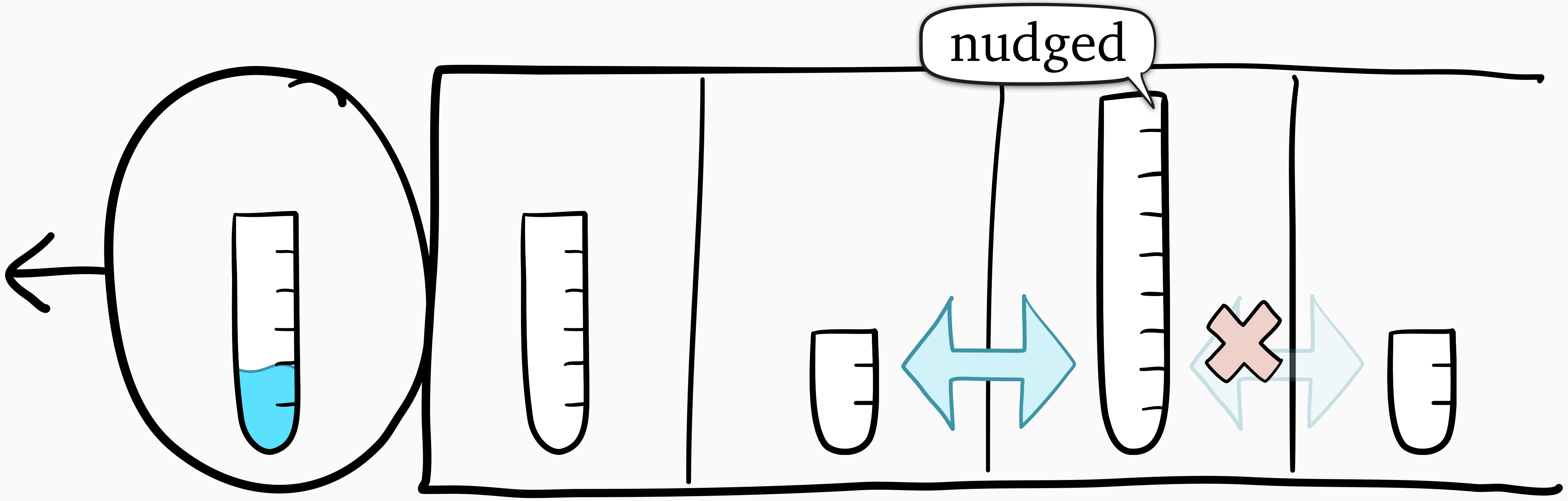
Nudge [Grosof et al., 2021]

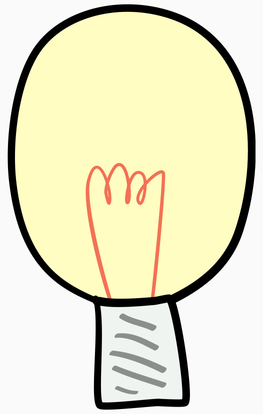
- small job can pass one large job
- large job can't be passed twice

Theorem:

$C_{\text{Nudge}} < C_{\text{FCFS}}$

Can we beat FCFS?





Nudge [Grosof et al., 2021]

- small job can pass one large job
- large job can't be passed twice

Theorem:

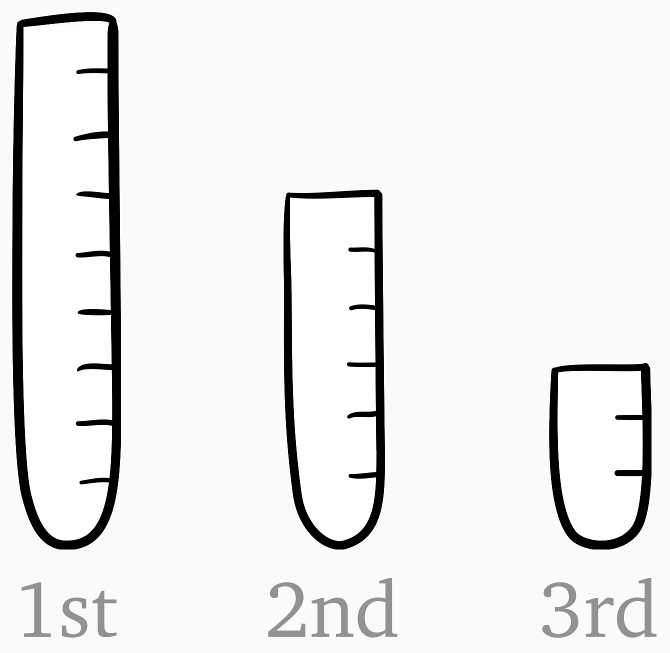
$C_{\text{Nudge}} < C_{\text{FCFS}}$

More complex variants get even lower C

[Van Houdt, 2022; Charlet & Van Houdt, 2024]

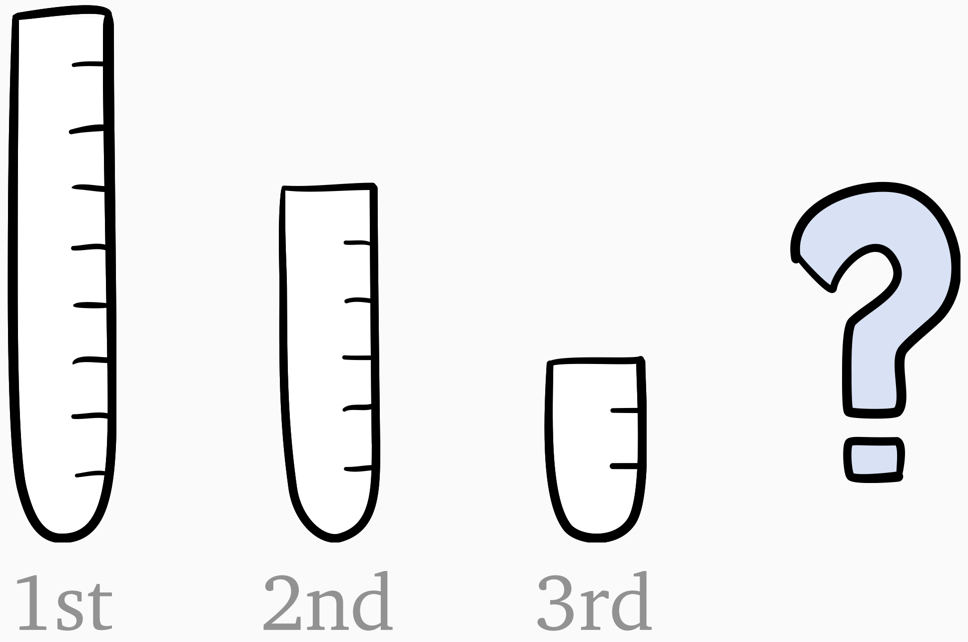
Can we beat Nudge?

Can we beat Nudge?



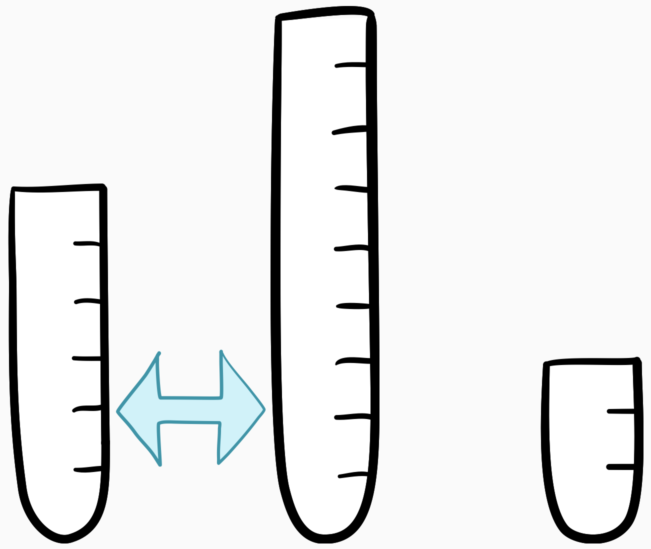
How to handle range of sizes?

Can we beat Nudge?

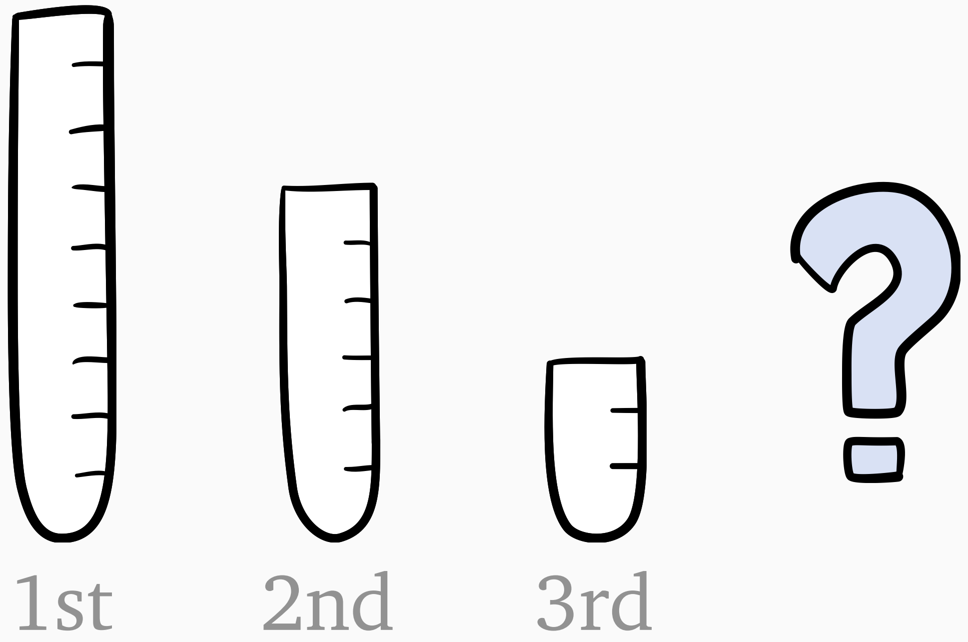


How to handle range of sizes?

medium = small?

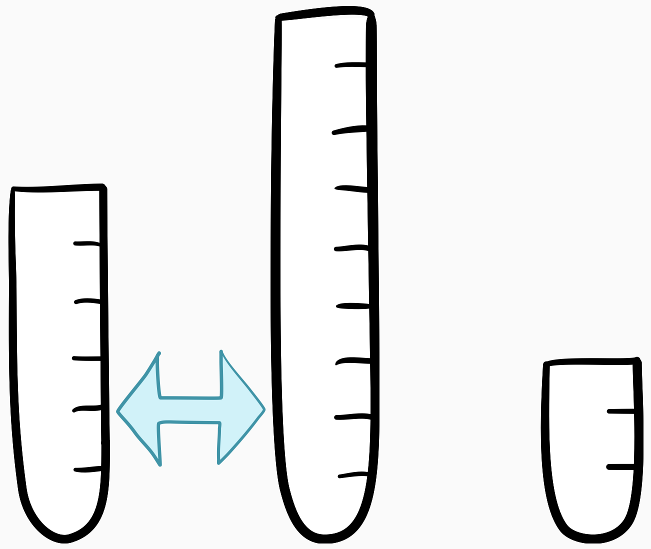


Can we beat Nudge?

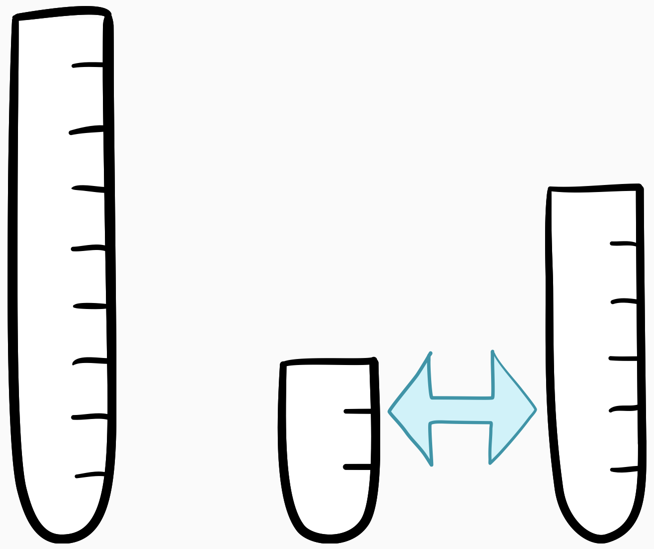


How to handle range of sizes?

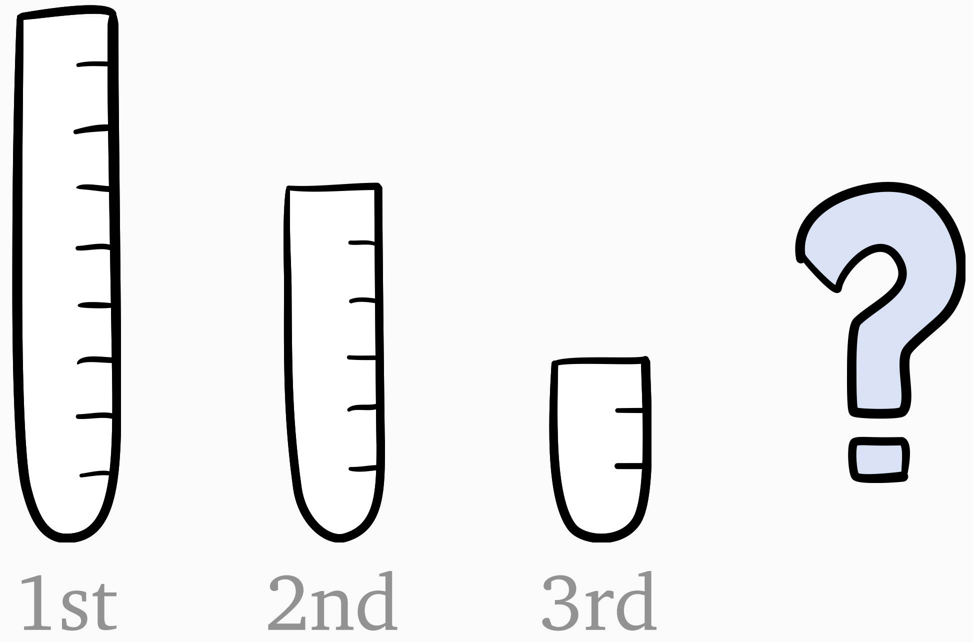
medium = small?



medium = large?

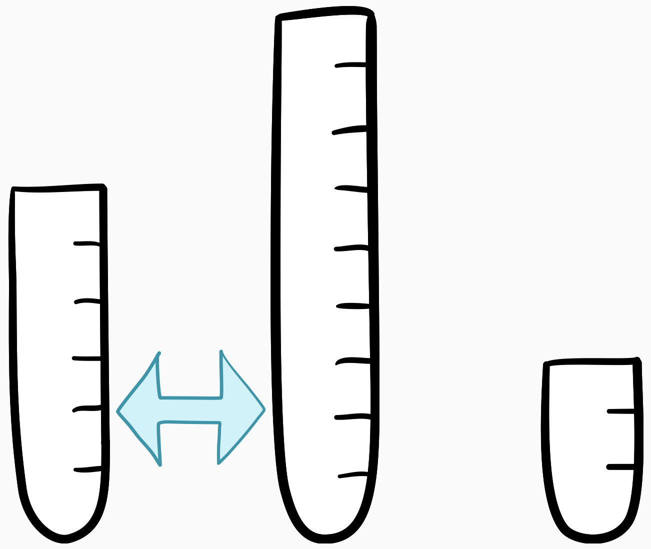


Can we beat Nudge?

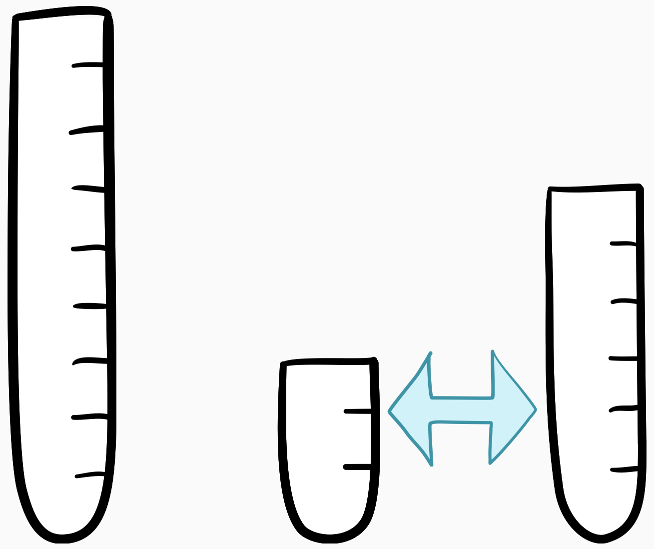


How to handle range of sizes?

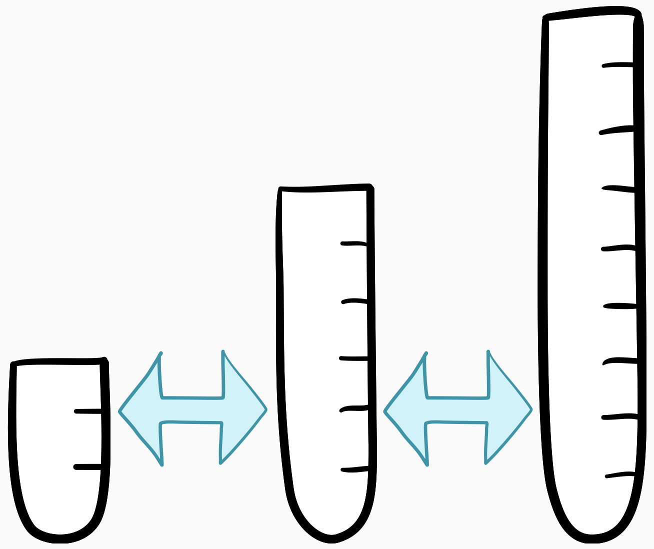
medium = small?



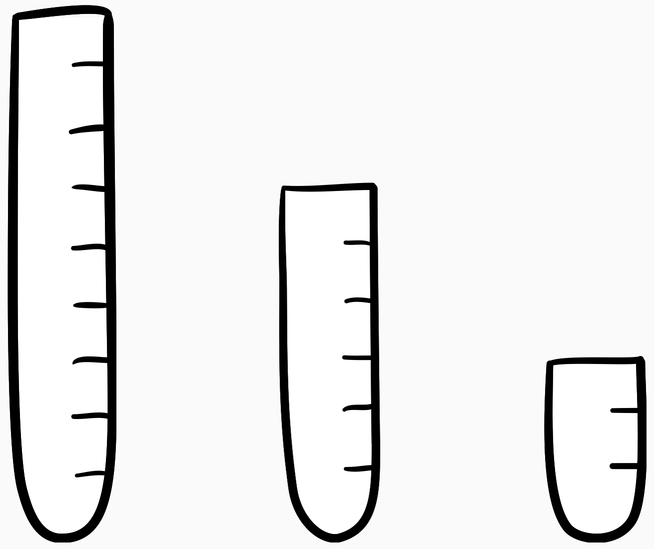
medium = large?



something else?



Can we beat Nudge?

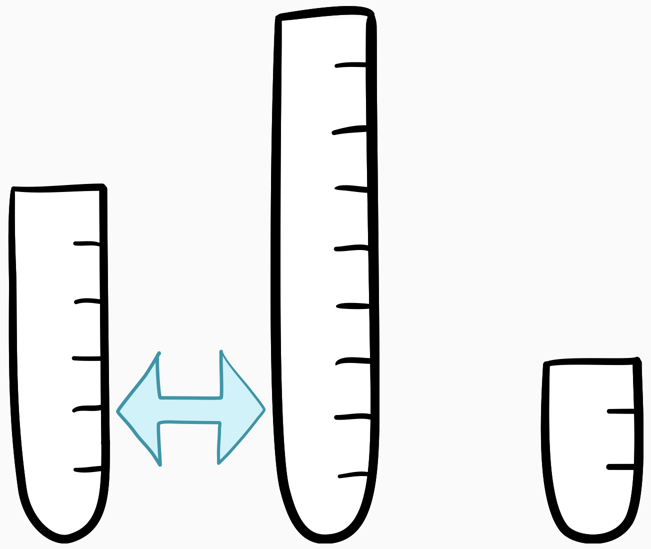


1st 2nd 3rd

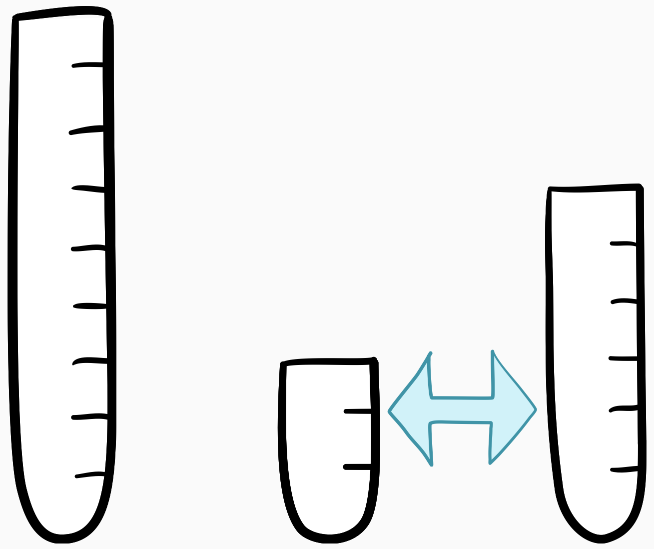
How to handle range of sizes?

What info could help us decide?

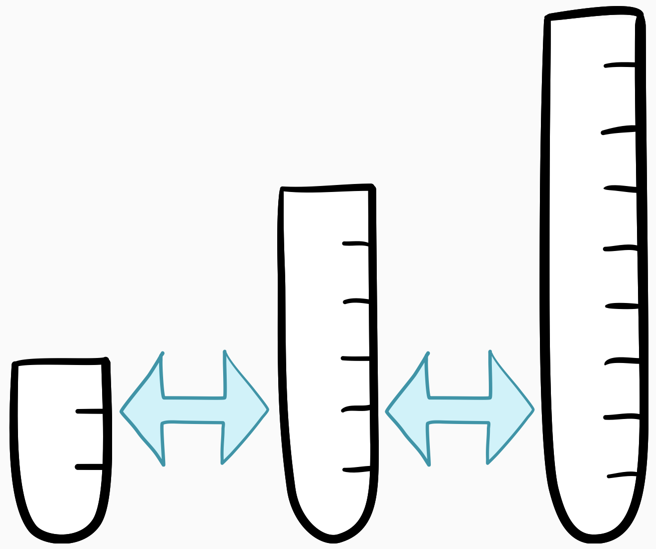
medium = small?



medium = large?



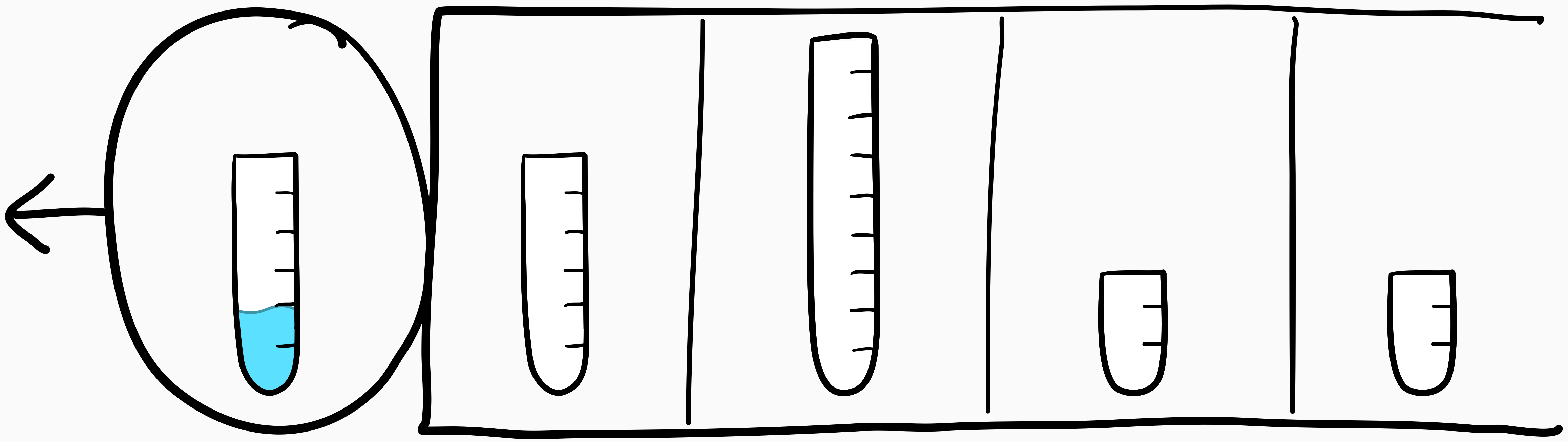
something else?



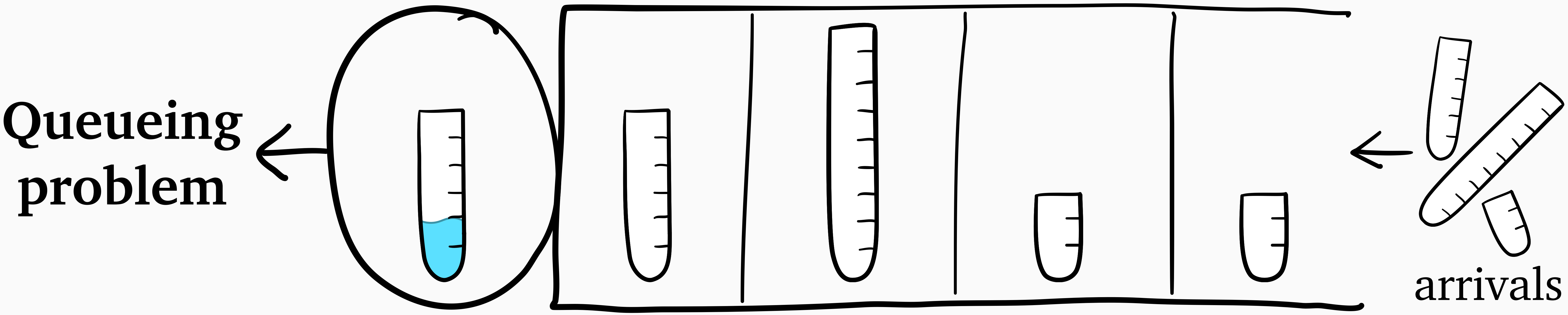
Where do optimal policies come from?

Where do optimal policies come from?

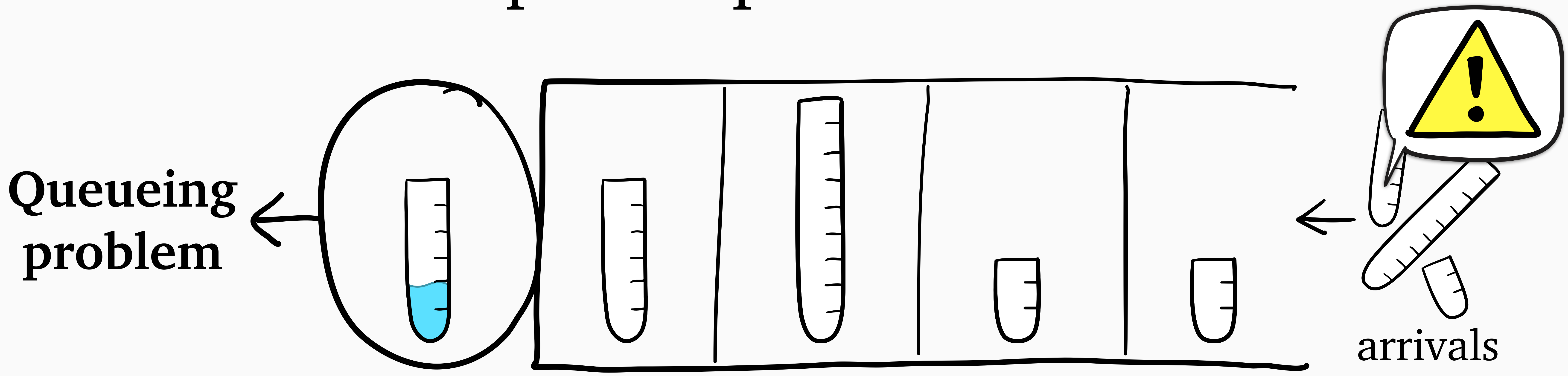
Queueing problem



Where do optimal policies come from?

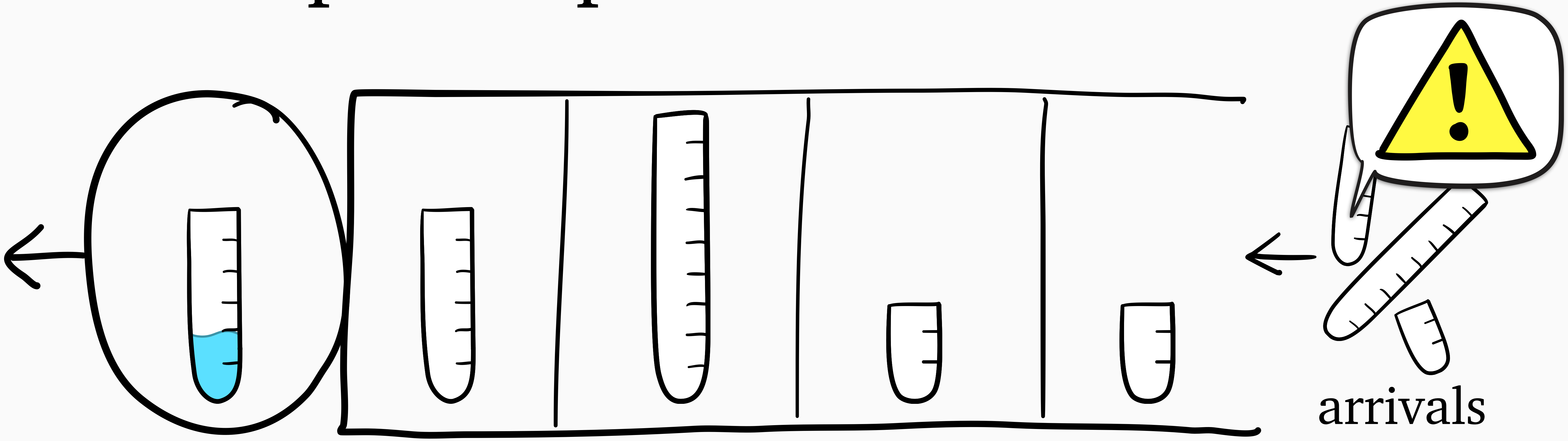


Where do optimal policies come from?

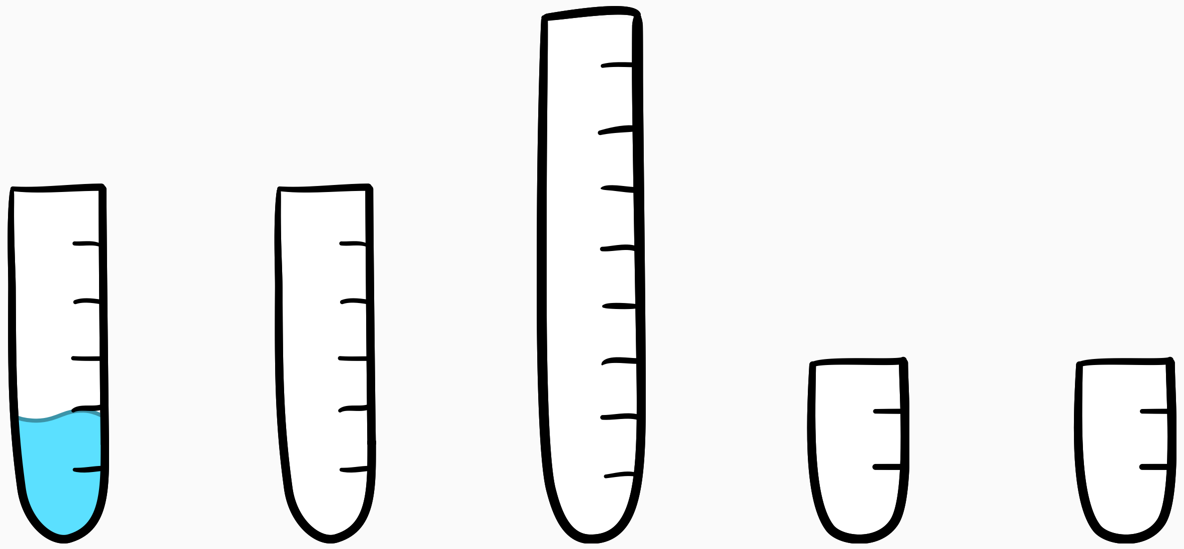


Where do optimal policies come from?

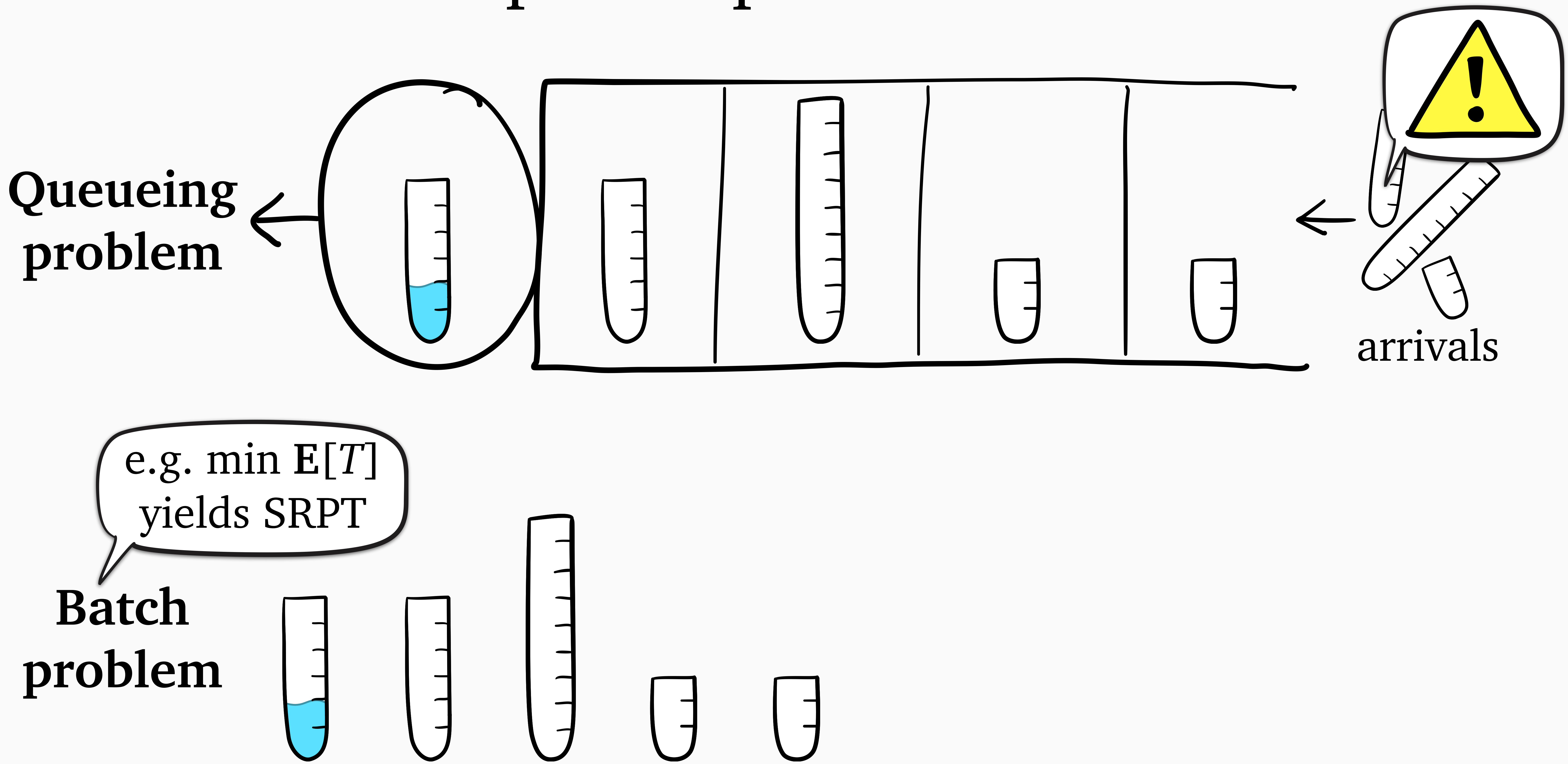
Queueing problem



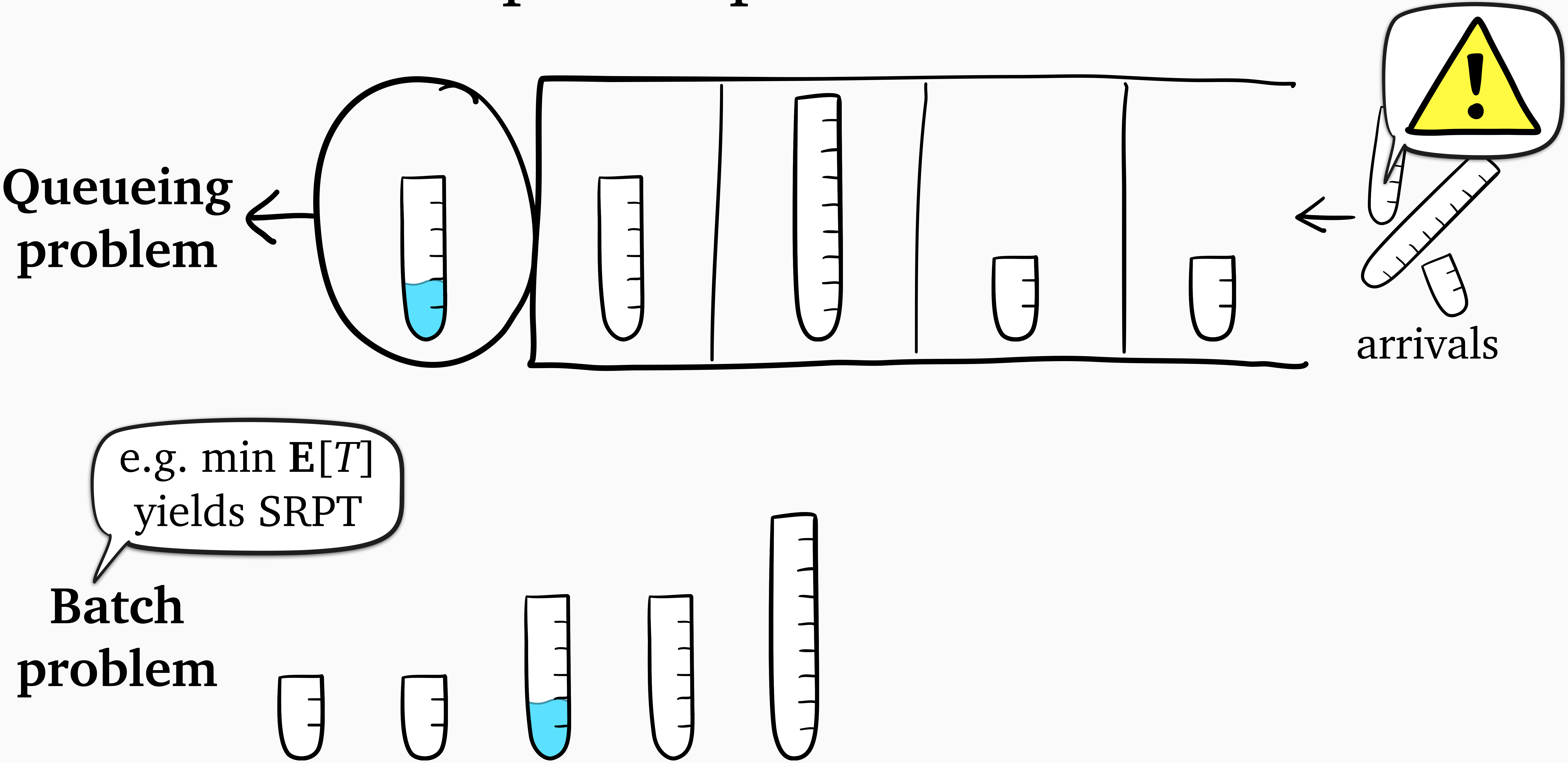
Batch problem



Where do optimal policies come from?

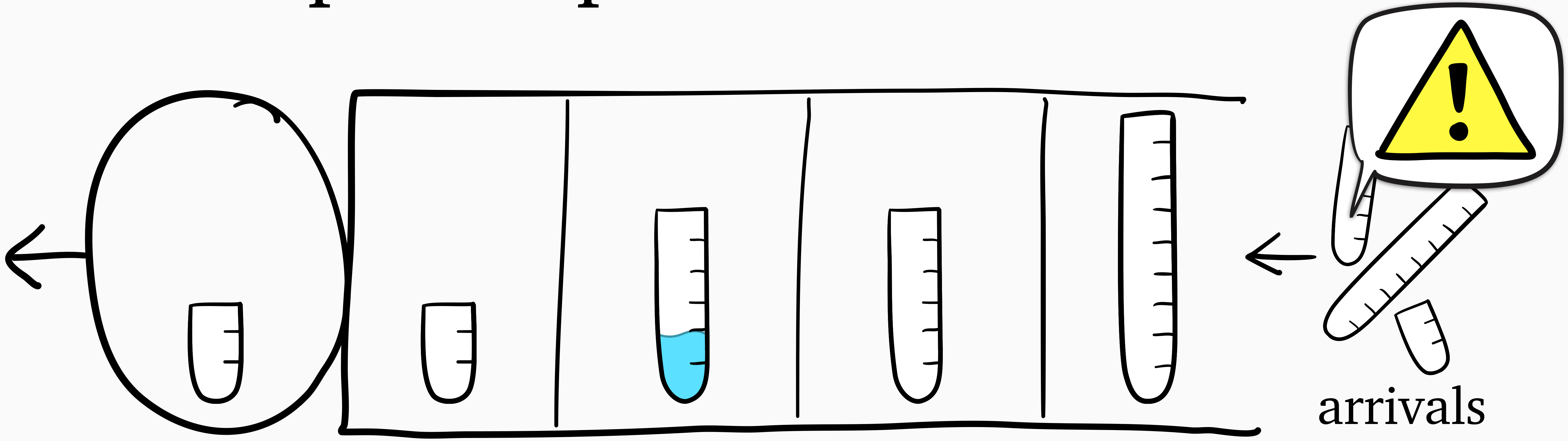


Where do optimal policies come from?



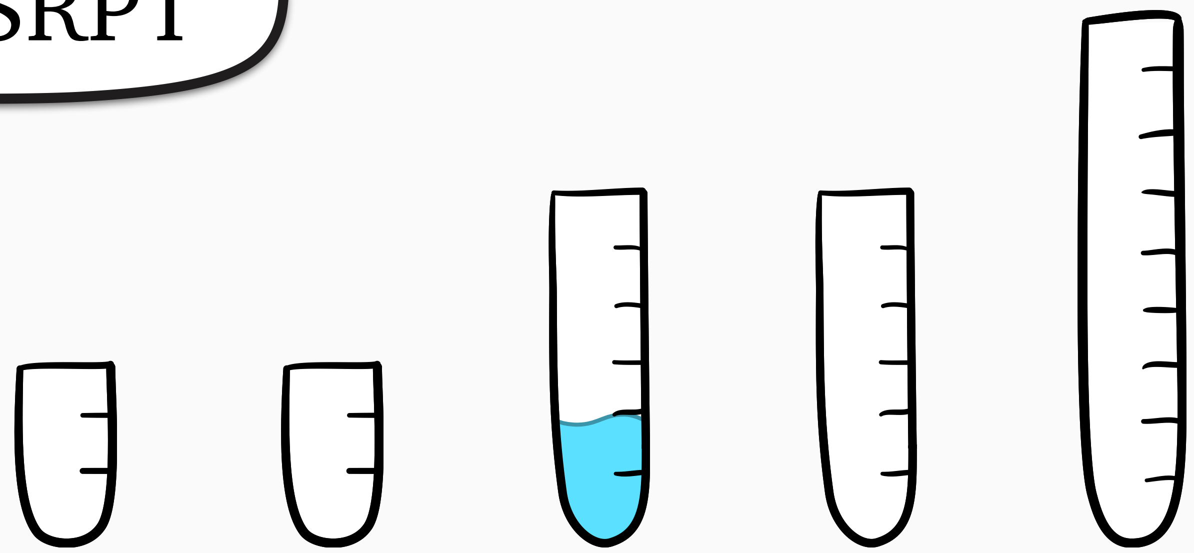
Where do optimal policies come from?

Queueing problem



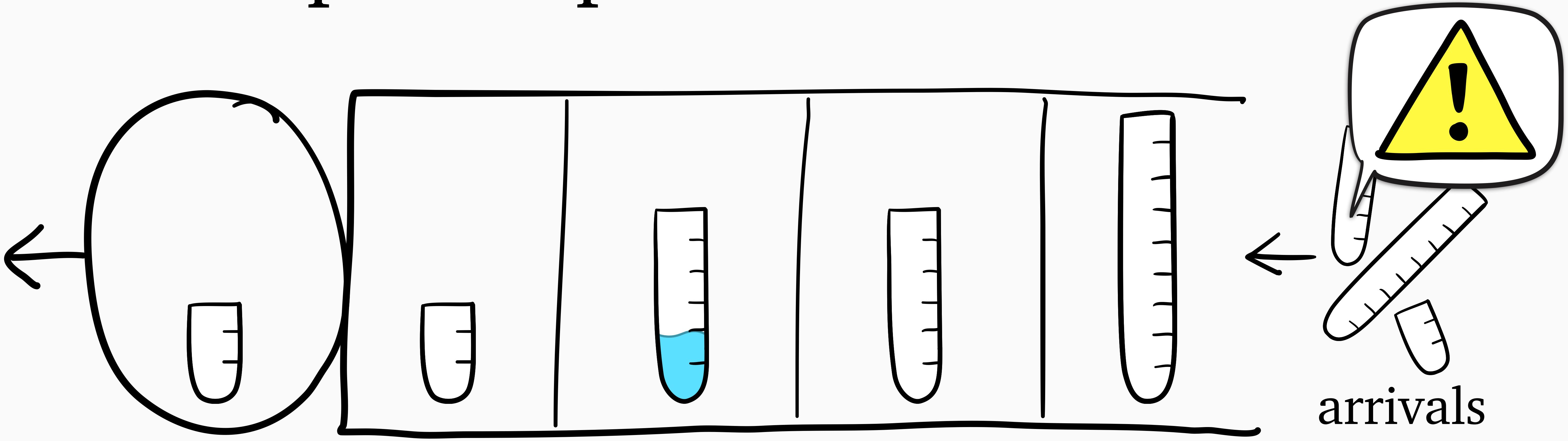
e.g. min $\mathbf{E}[T]$ yields SRPT

Batch problem



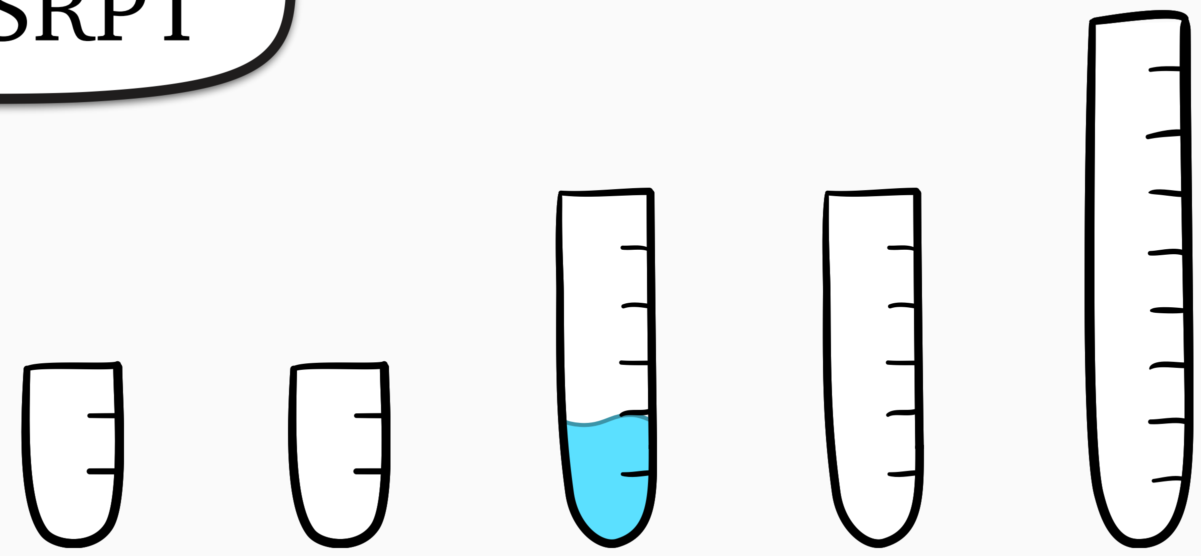
Where do optimal policies come from?

Queueing problem



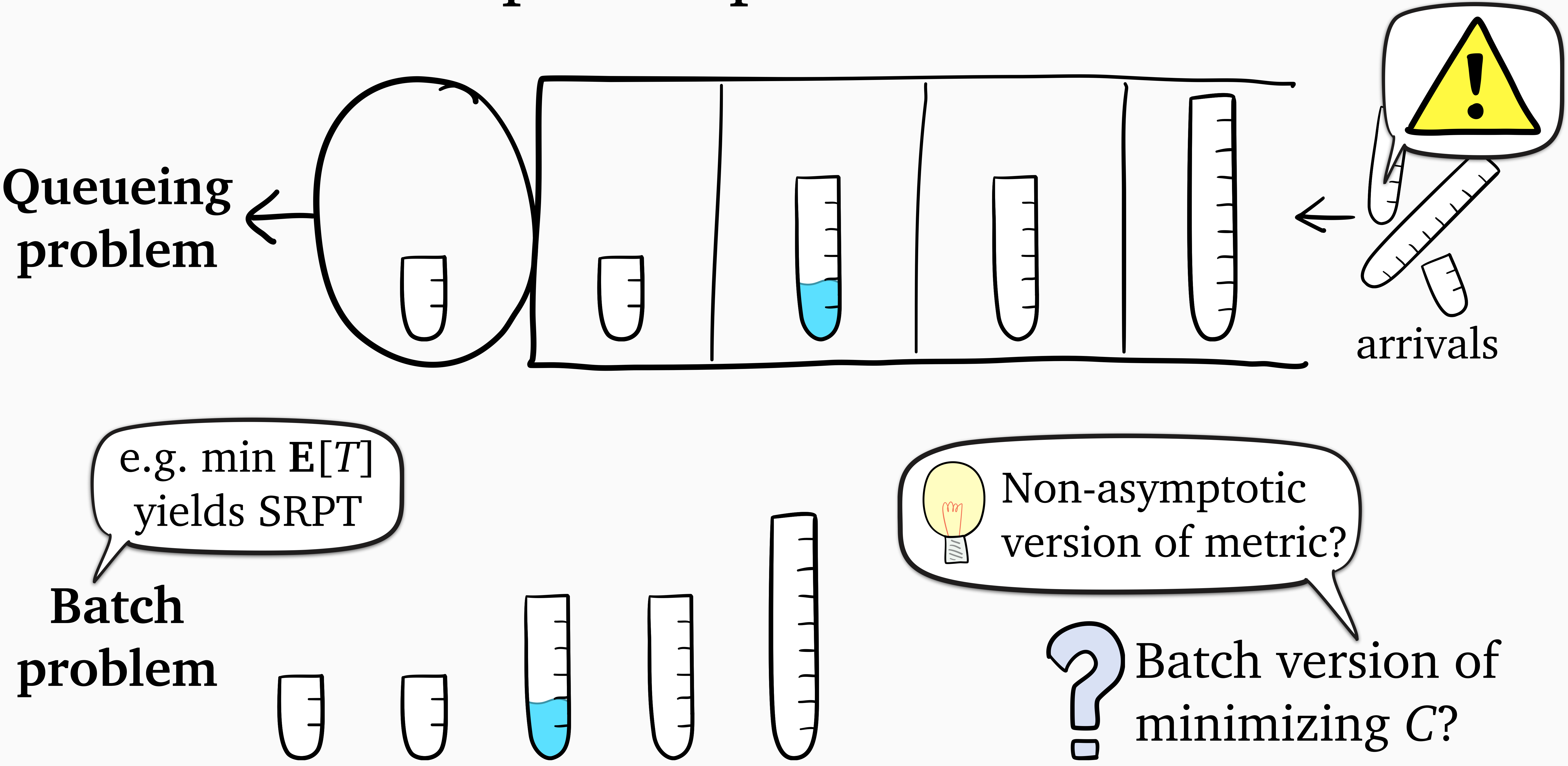
e.g. min $\mathbf{E}[T]$
yields SRPT

Batch problem

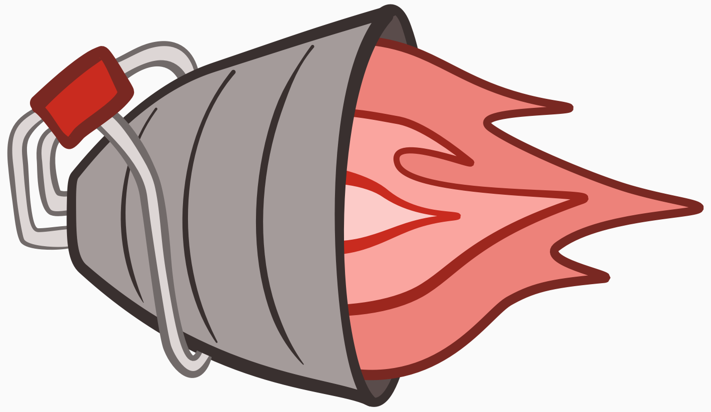


Batch version of minimizing C?

Where do optimal policies come from?



Boost

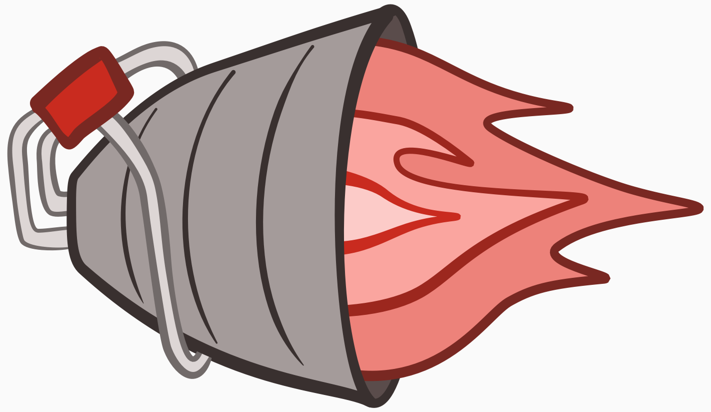


Why is achieving strong tail optimality hard?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost



Why is achieving strong tail optimality hard?

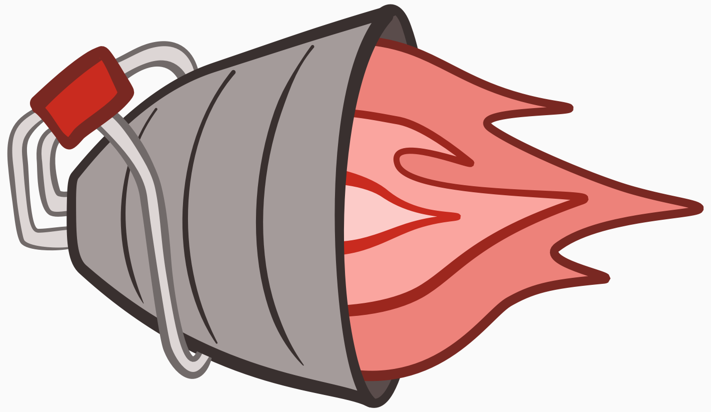
How to handle range of sizes?

Batch version of minimizing C?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost



How to handle range of sizes?

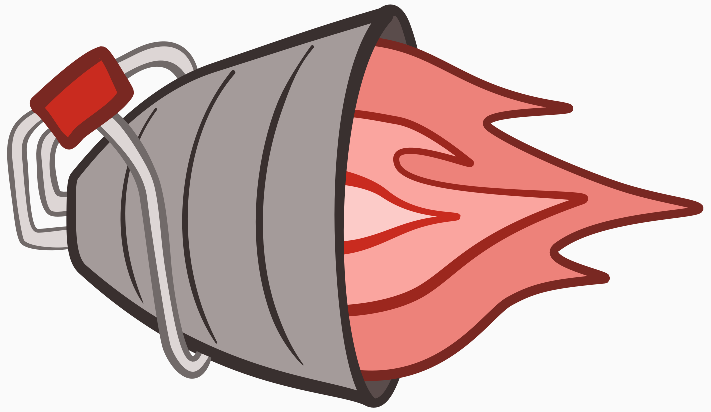
Why is achieving strong tail optimality hard?

Batch version of minimizing C?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost



How to handle range of sizes?

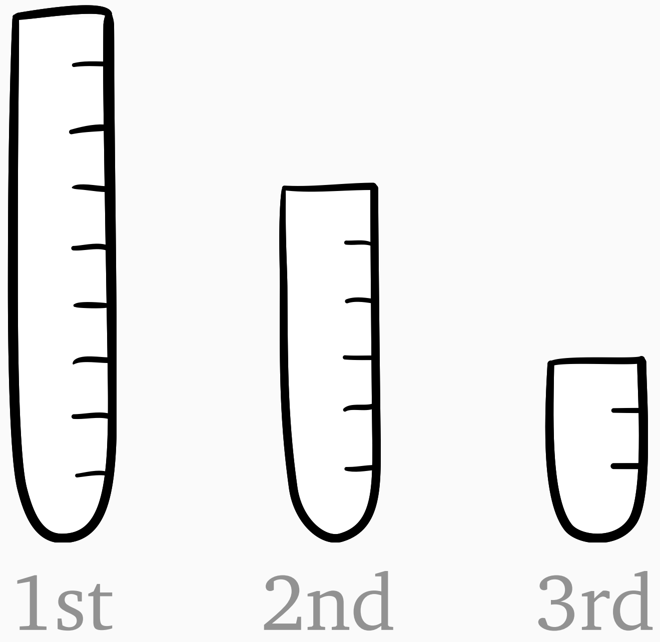
Why is achieving strong tail optimality hard?

Batch version of minimizing C?

How does the **Boost** policy family work?

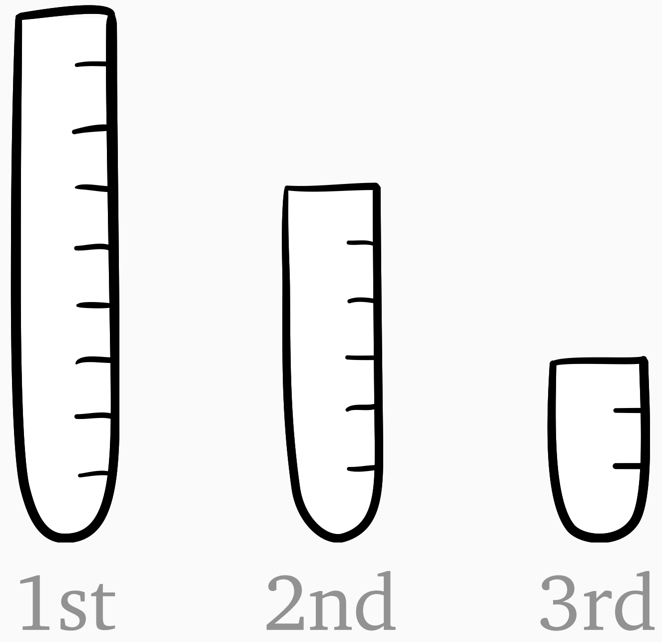
How do we achieve strong tail optimality?

Key information:



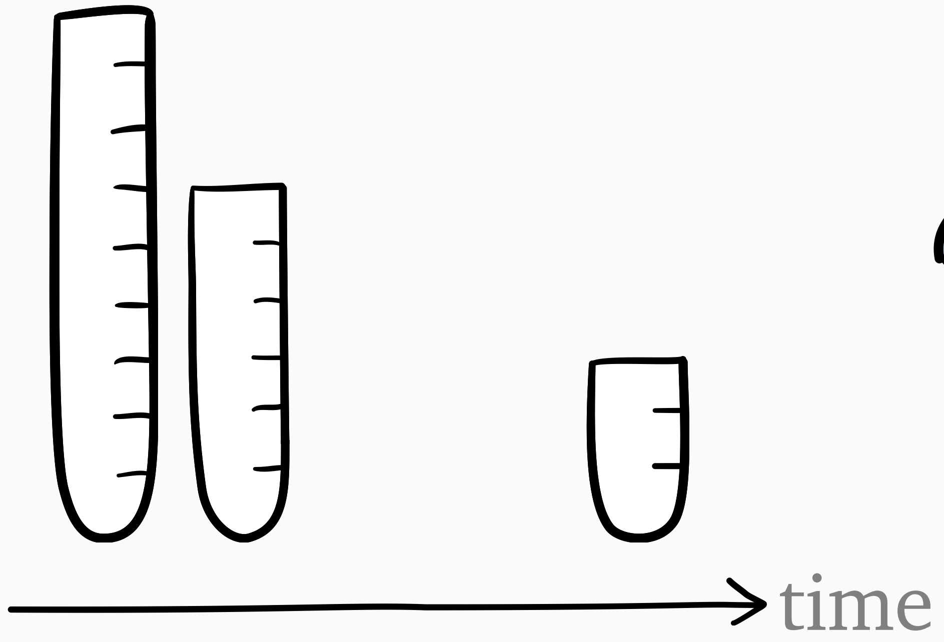
How to handle range of sizes?

Key information: *arrival times*



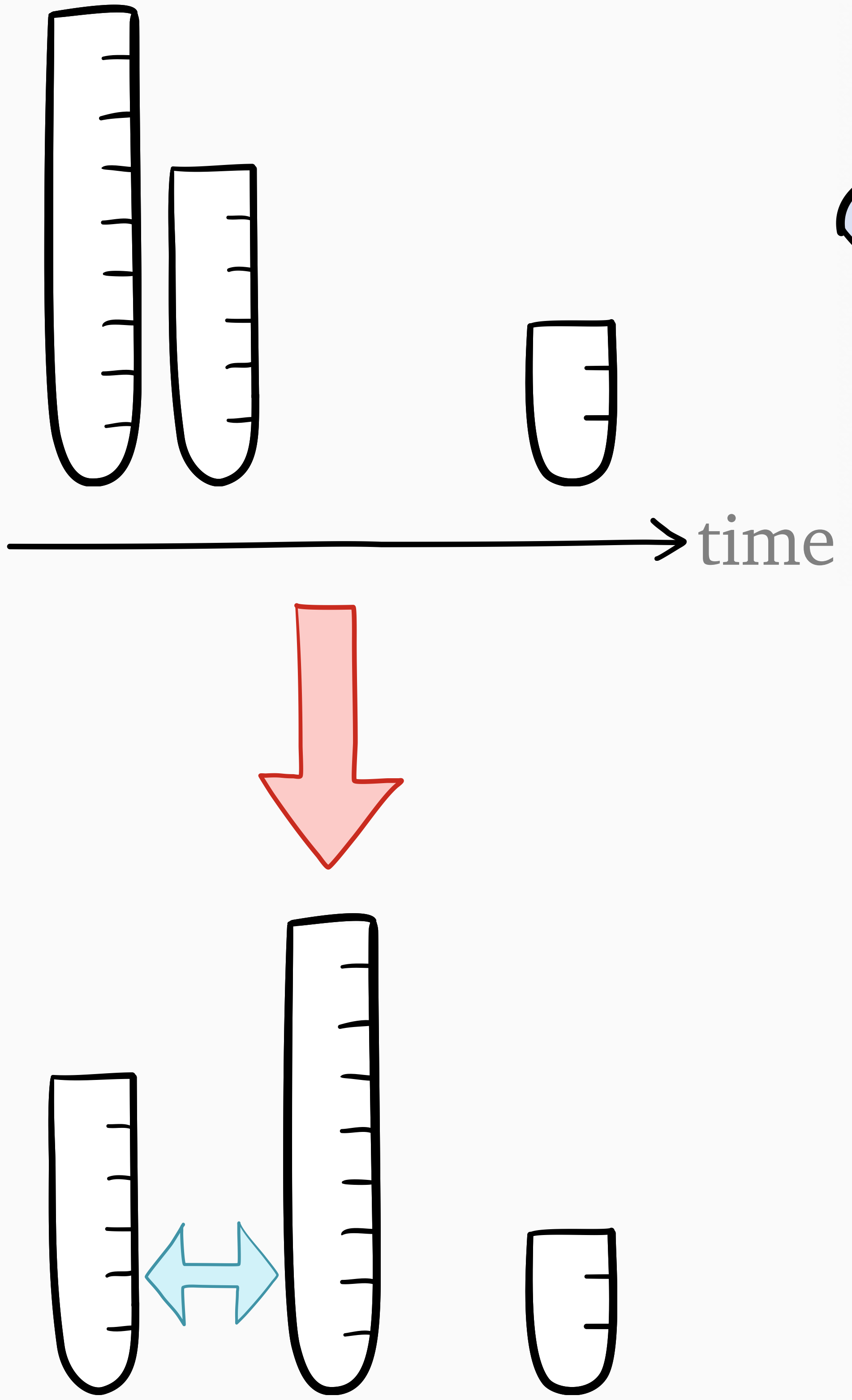
How to handle range of sizes?

Key information: *arrival times*



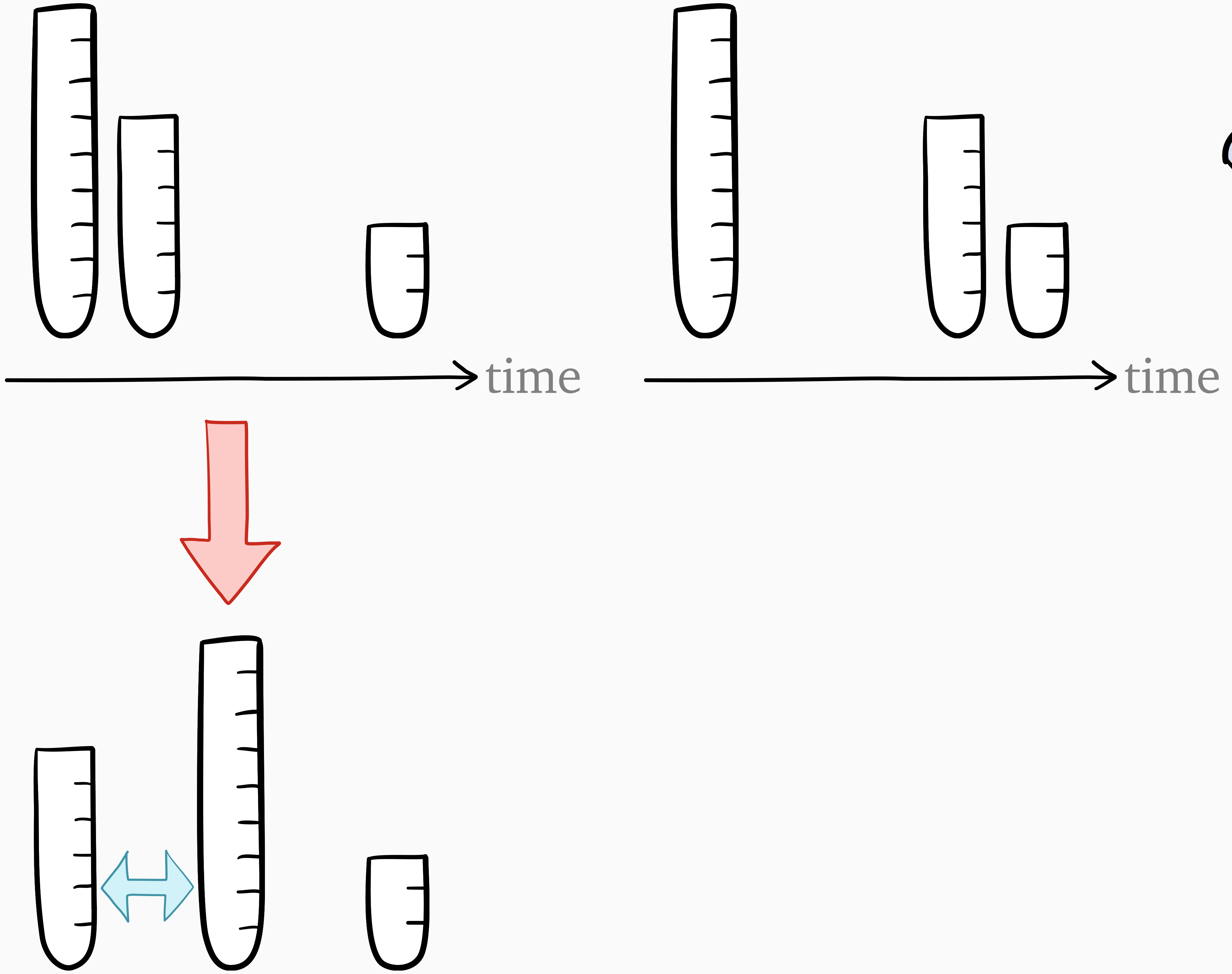
How to handle range of sizes?

Key information: *arrival times*



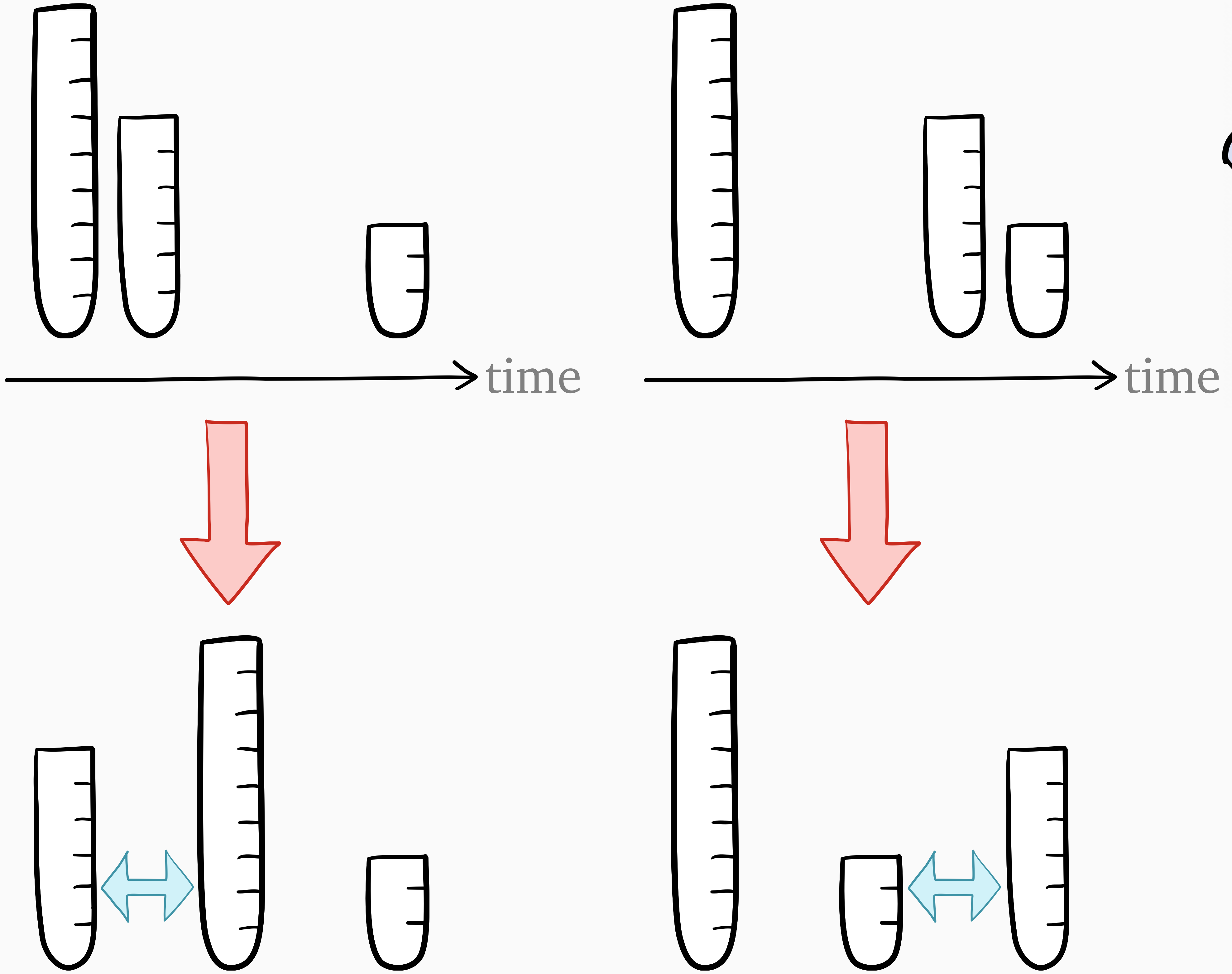
How to handle range of sizes?

Key information: *arrival times*



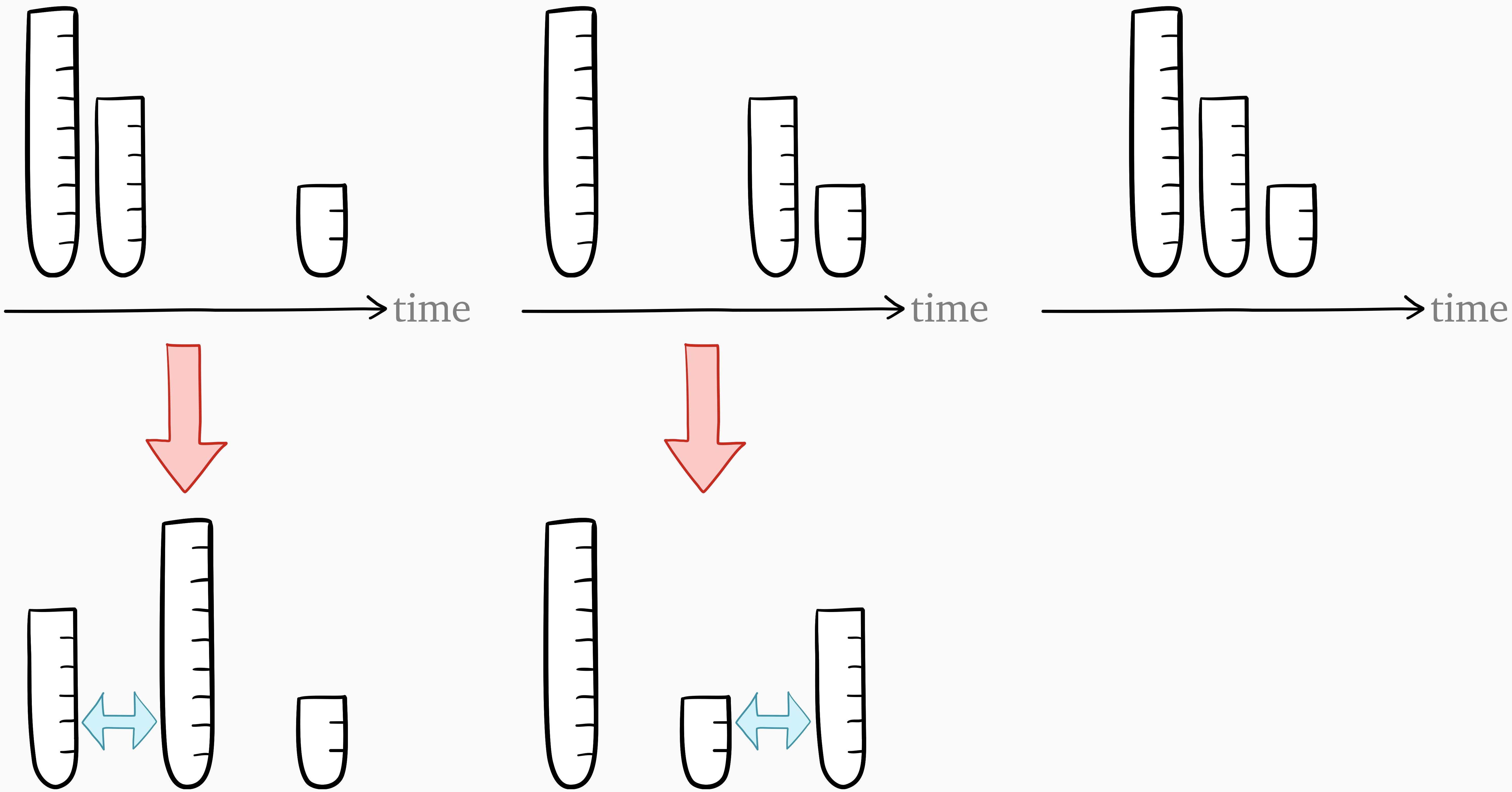
How to handle range of sizes?

Key information: *arrival times*

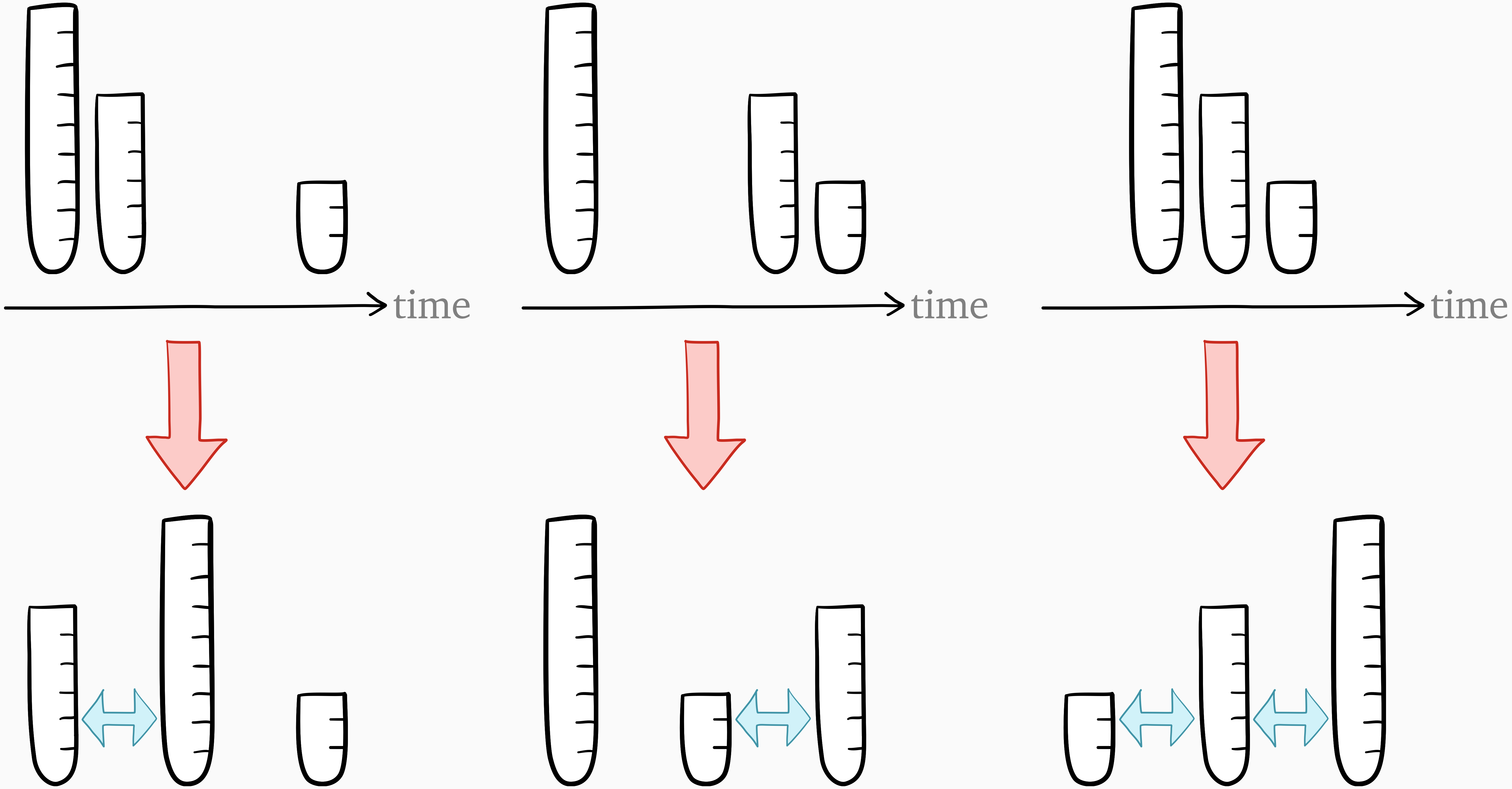


How to handle range of sizes?

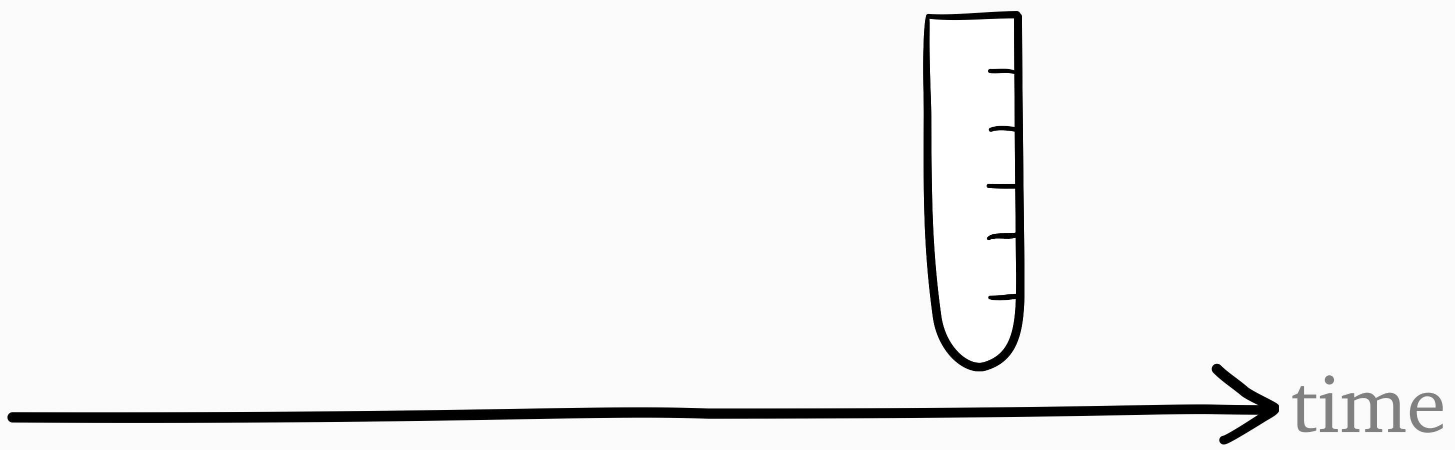
Key information: *arrival times*



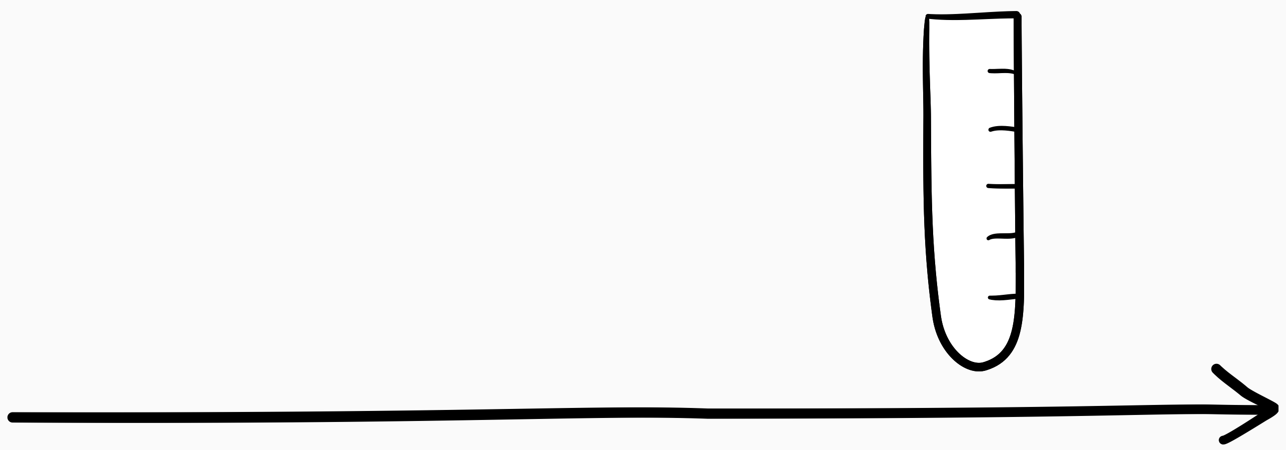
Key information: *arrival times*



Combining arrival time and size

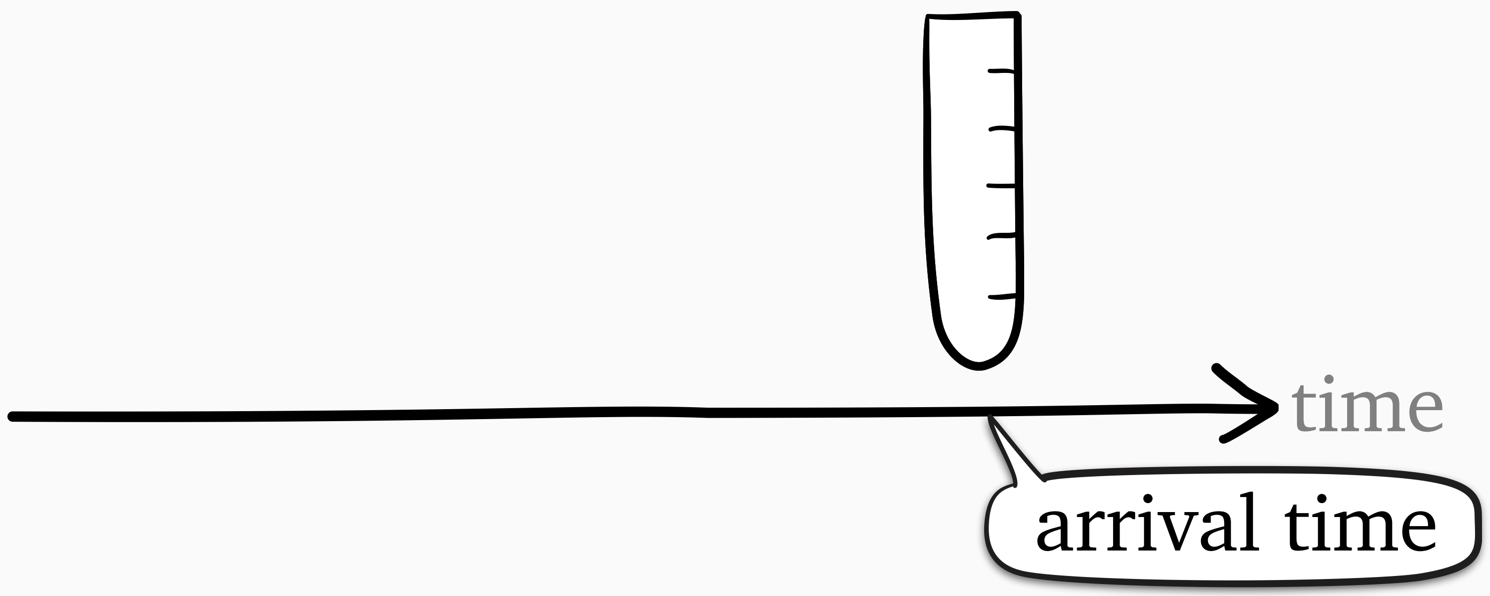


Combining arrival time and size



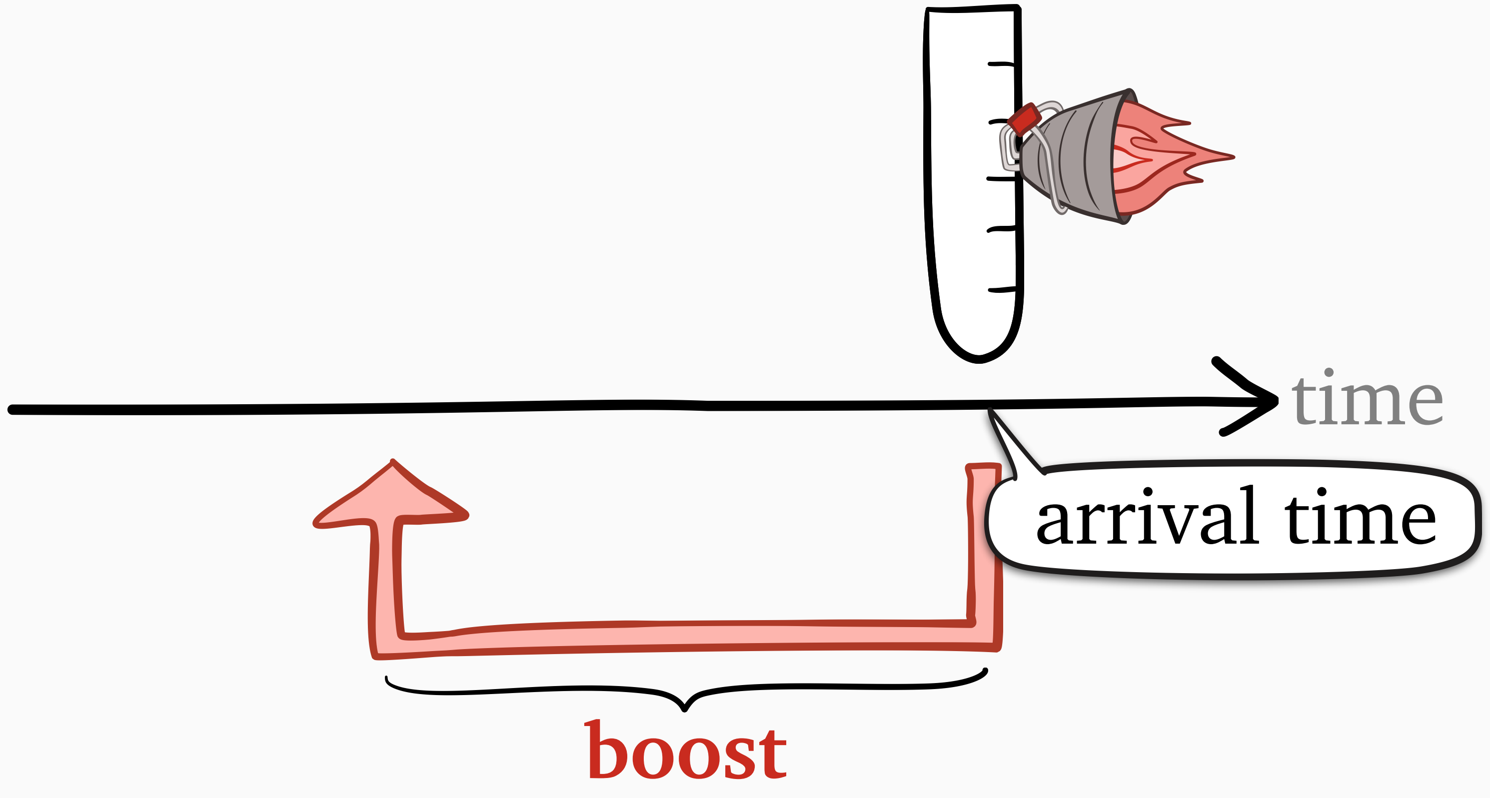
boosted arrival time
= arrival time − **boost**(size)

Combining arrival time and size



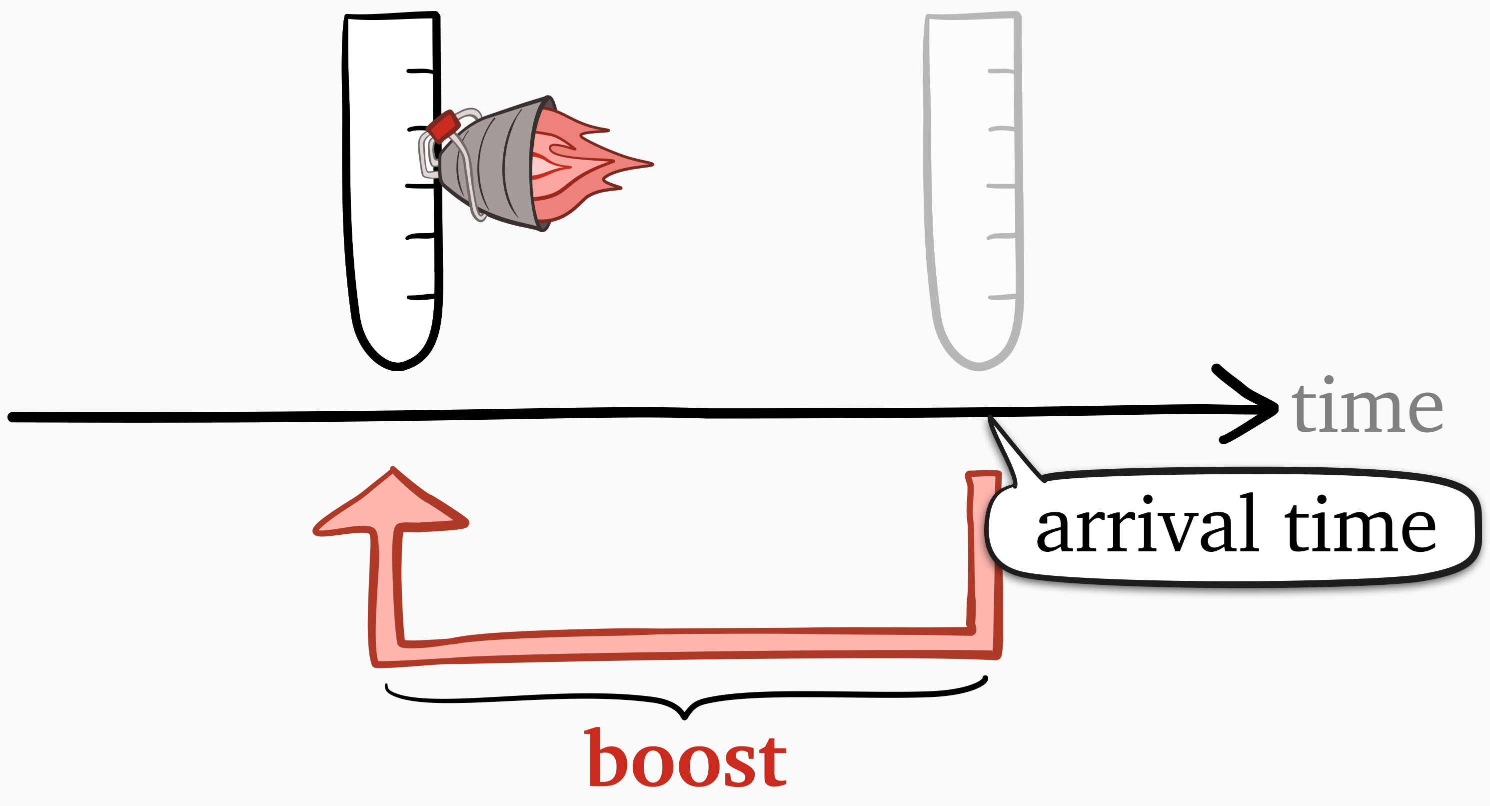
boosted arrival time
= arrival time − **boost**(size)

Combining arrival time and size



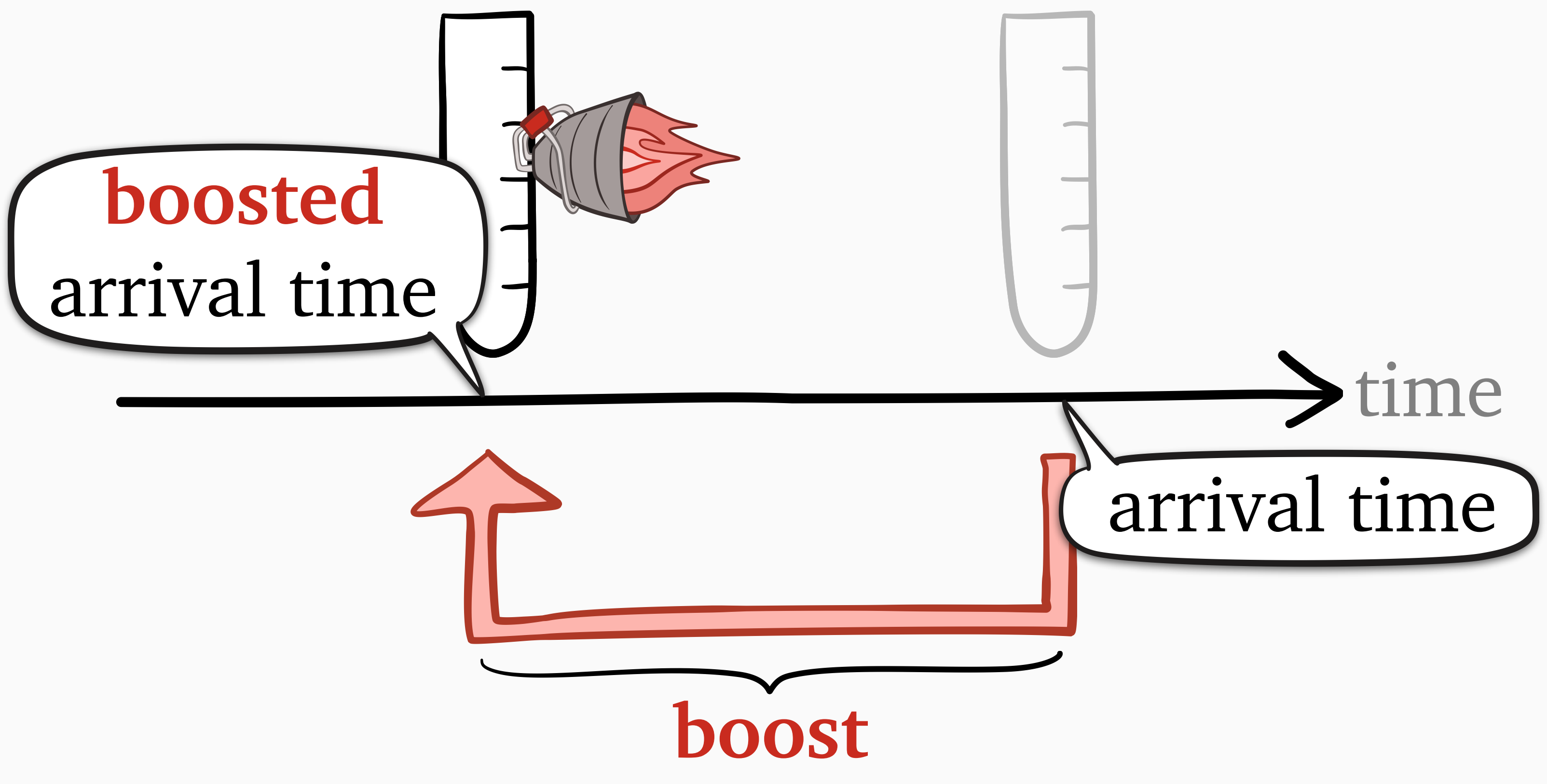
boosted arrival time
= arrival time − **boost**(size)

Combining arrival time and size



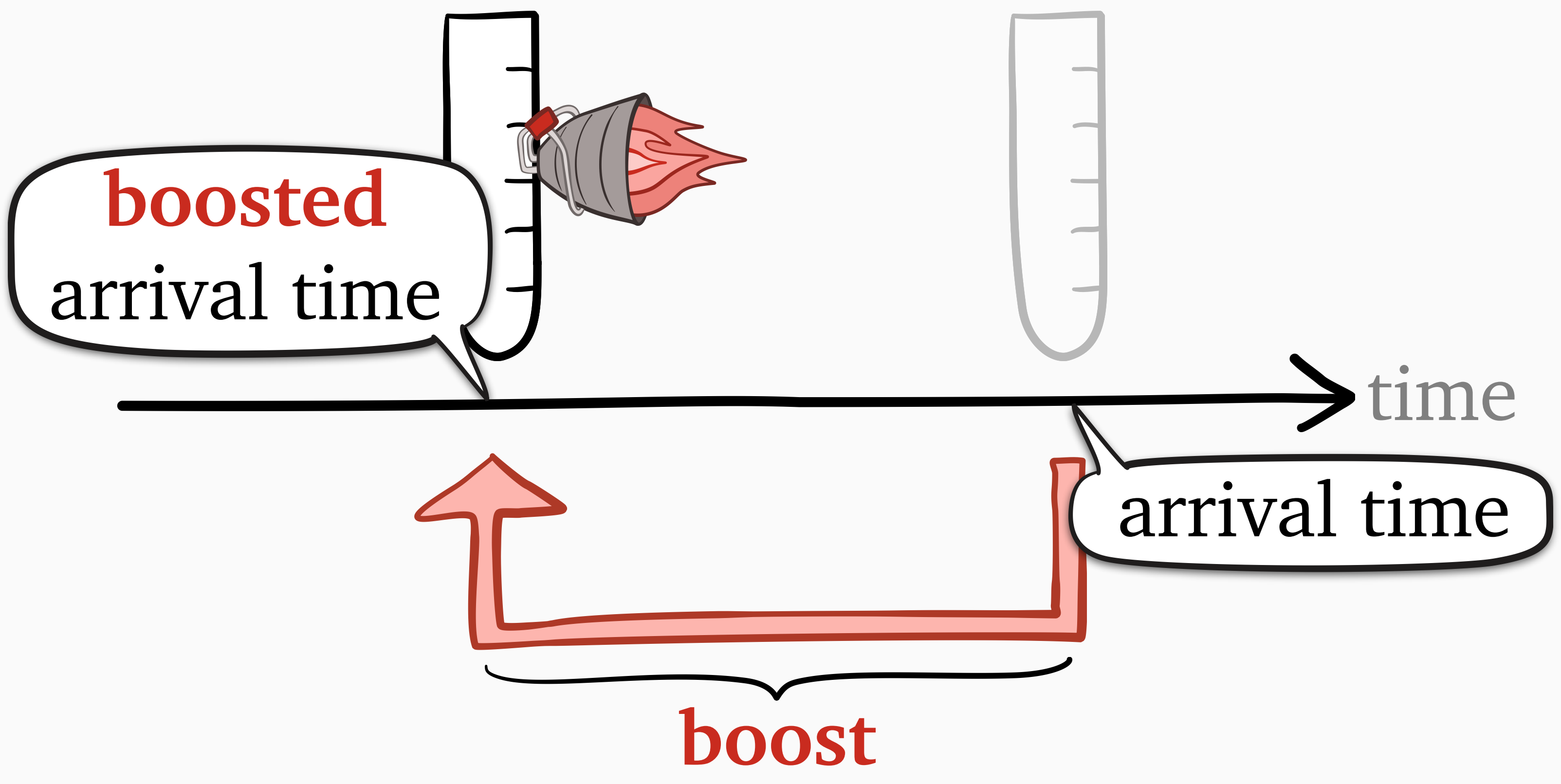
boosted arrival time
= arrival time − **boost**(size)

Combining arrival time and size



boosted arrival time
= arrival time − **boost**(size)

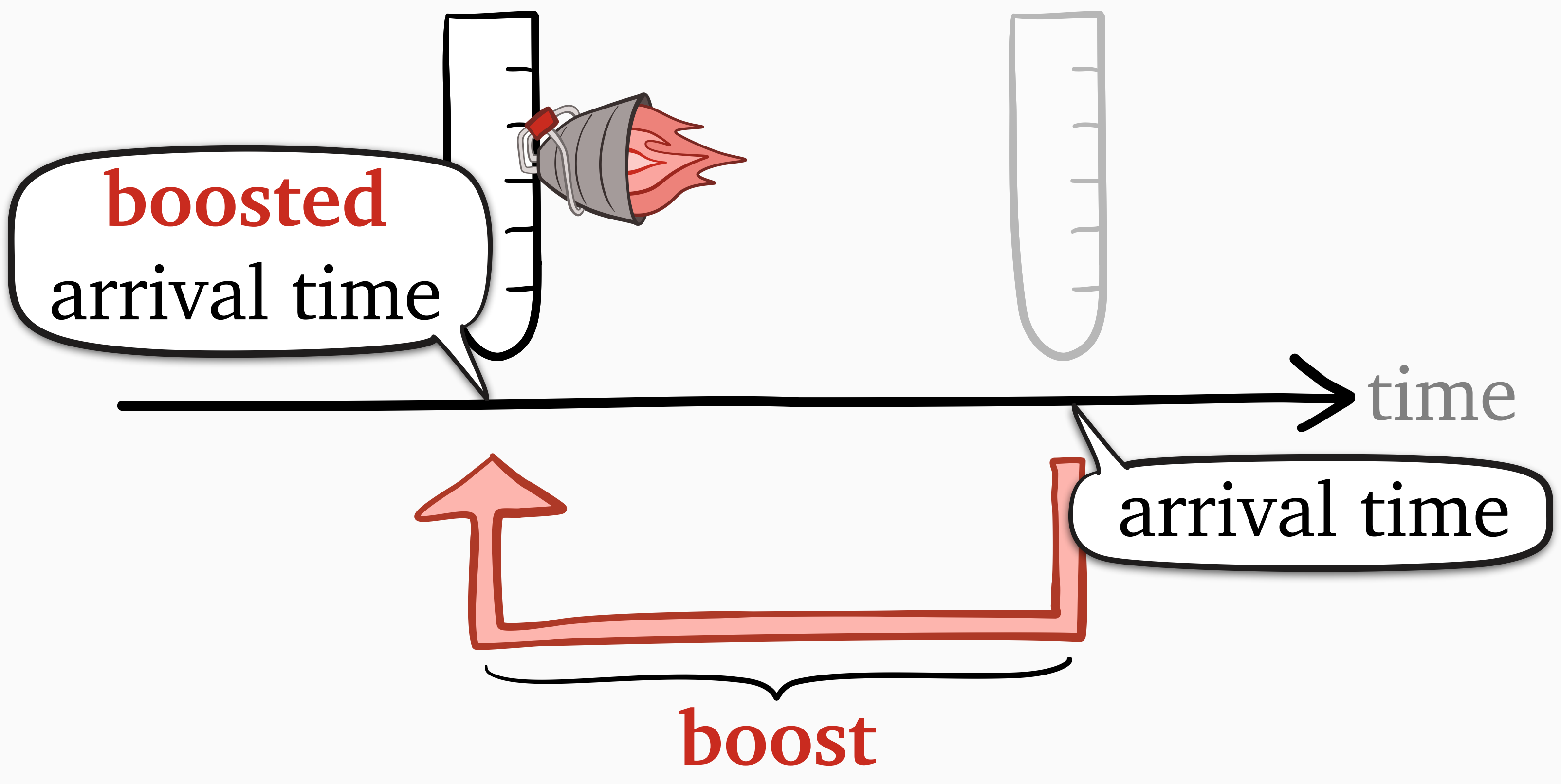
Combining arrival time and size



boosted arrival time
= arrival time − **boost**(size)

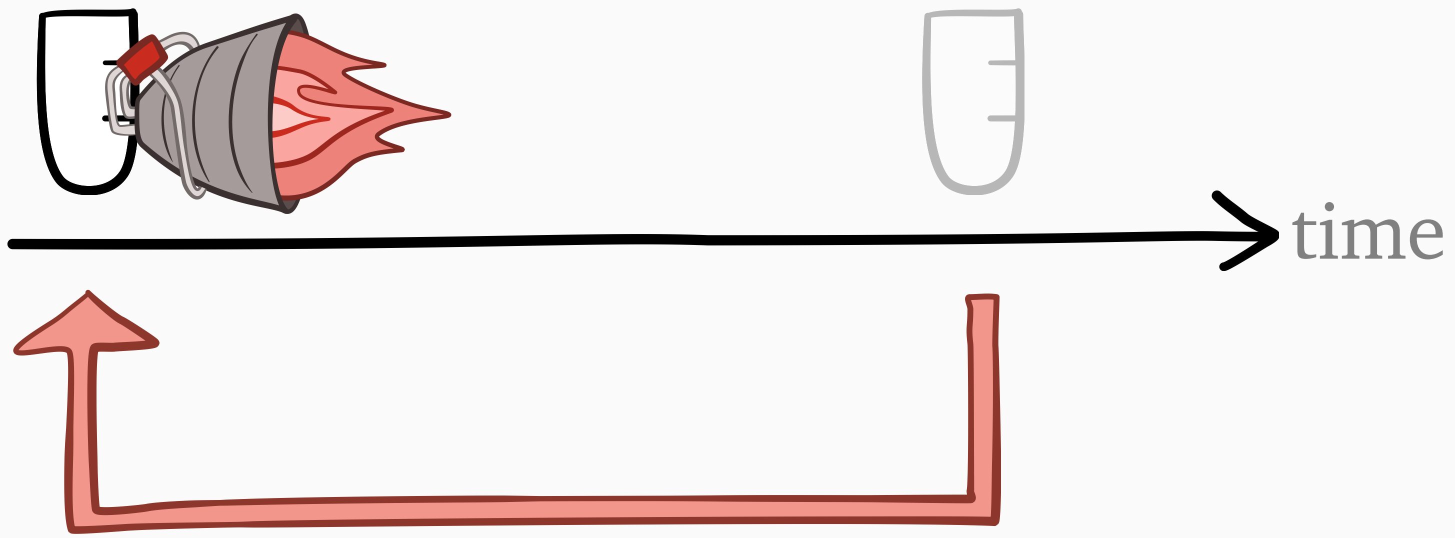
smaller sizes get bigger boosts

Combining arrival time and size

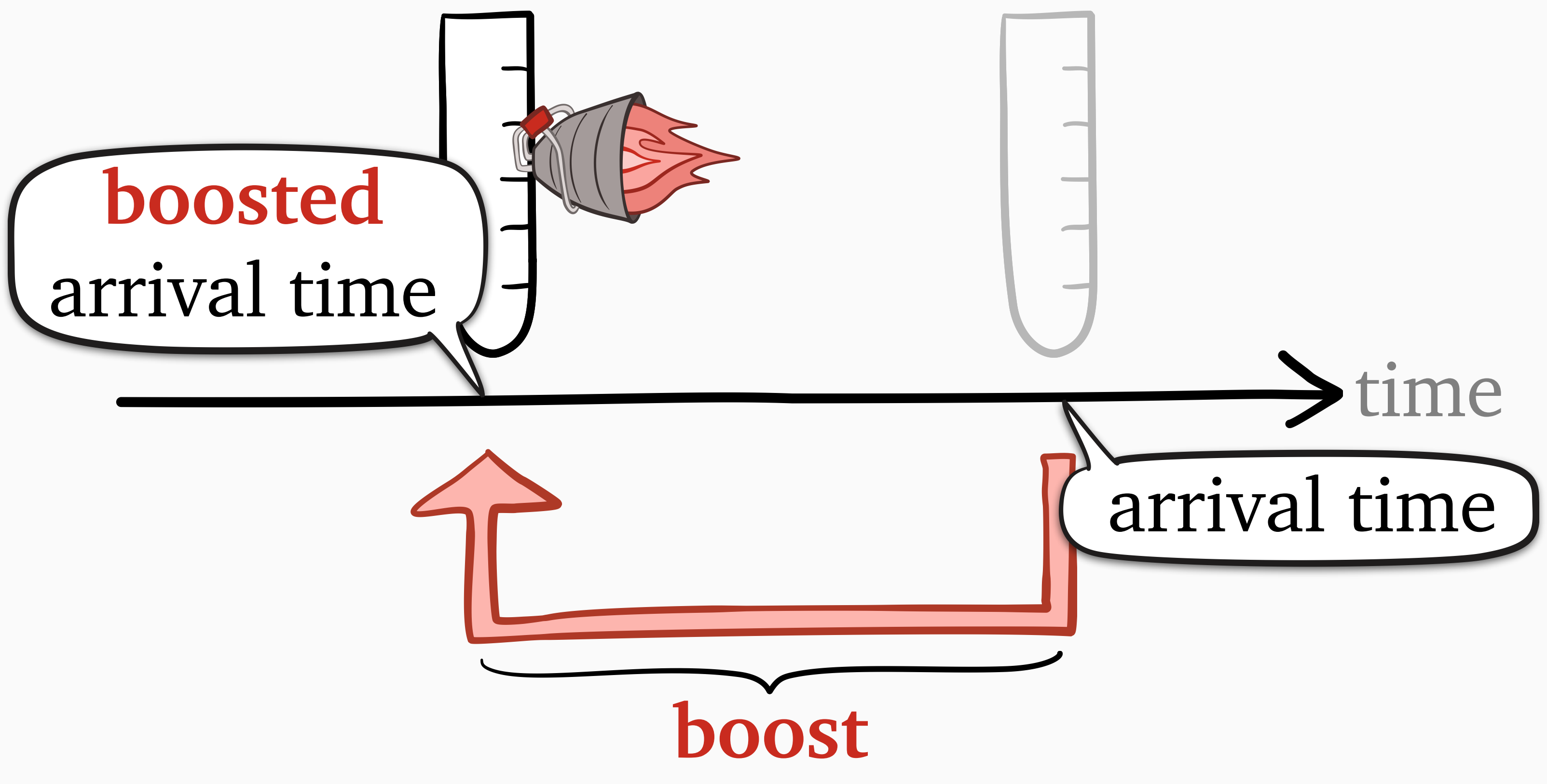


boosted arrival time
= arrival time − **boost**(size)

smaller sizes get bigger boosts

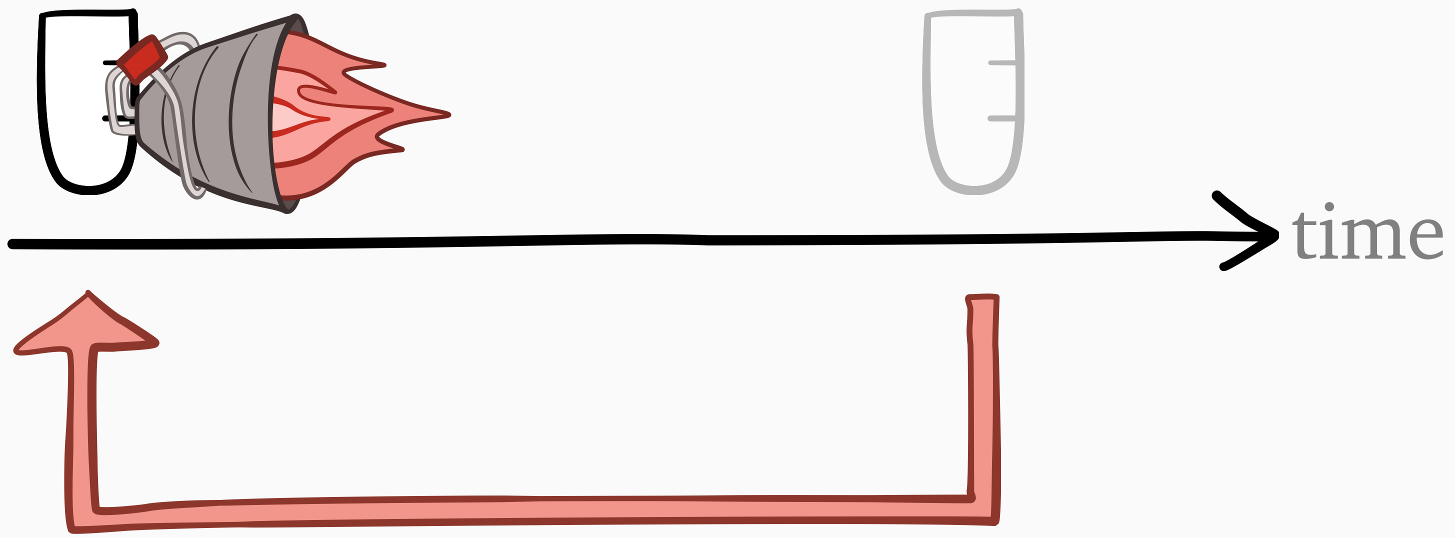


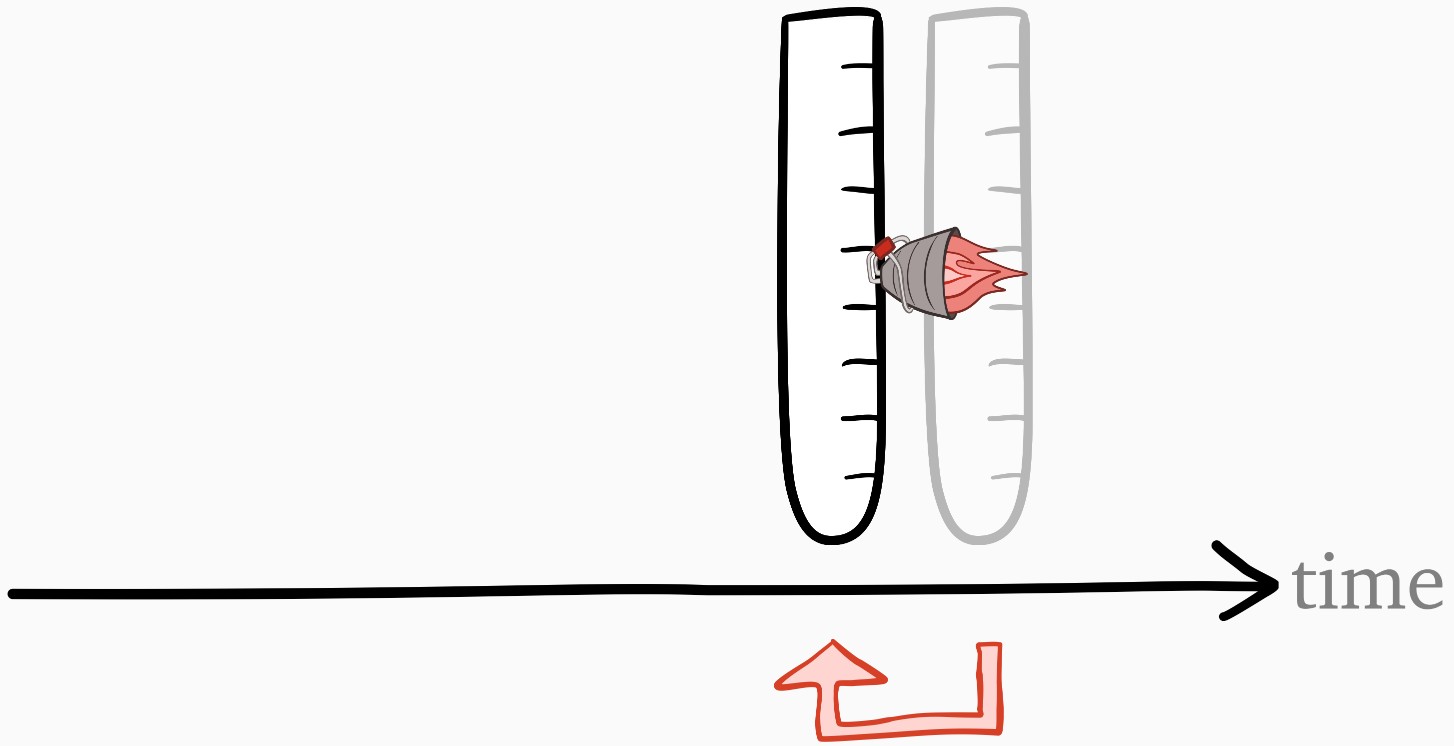
Combining arrival time and size



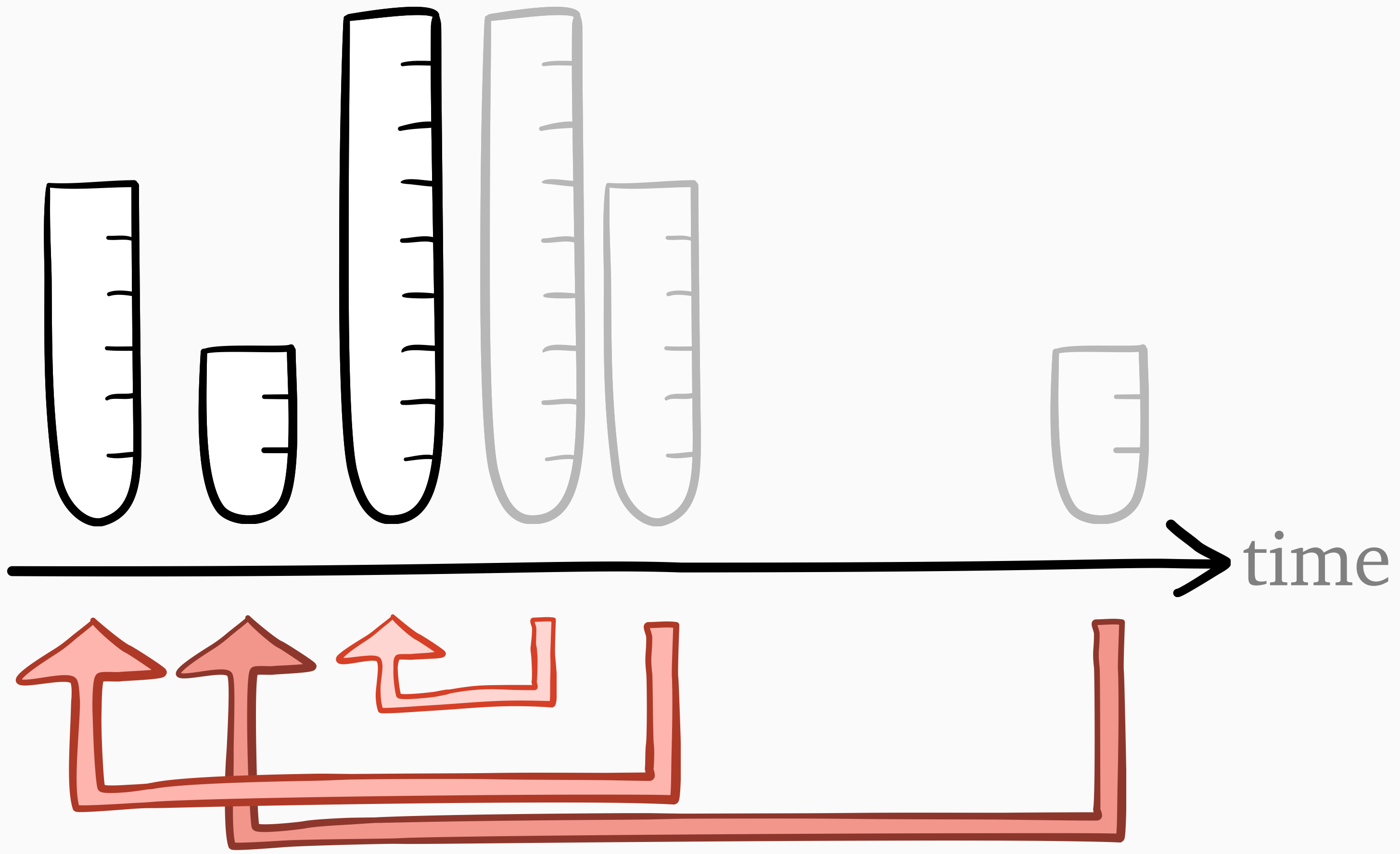
boosted arrival time
= arrival time − **boost**(size)

smaller sizes get bigger boosts



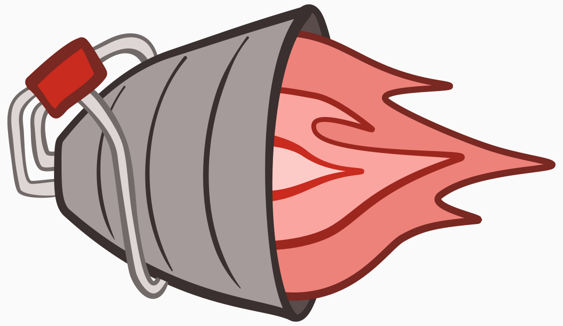


Boost policies

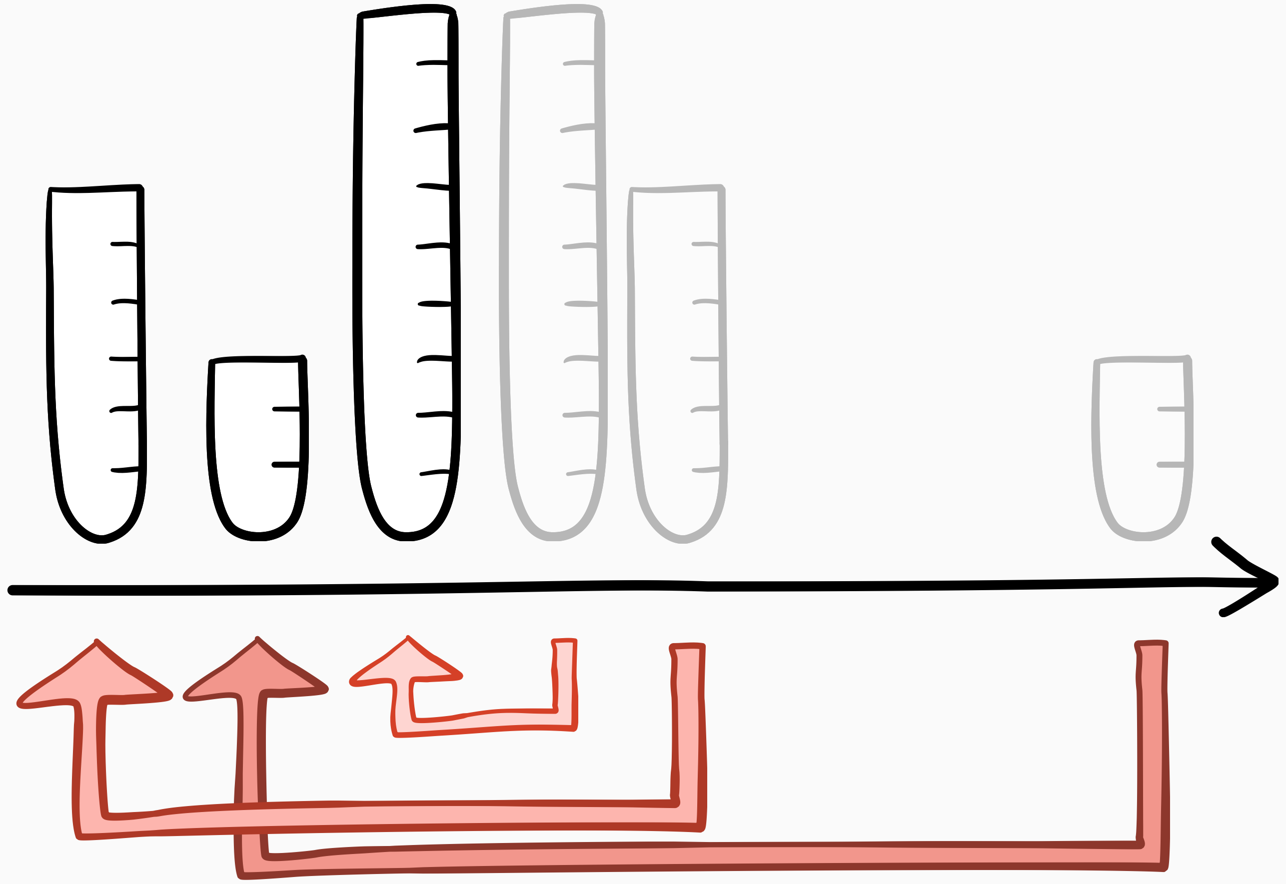


boosted arrival time
= arrival time − **boost**(size)

Boost policies

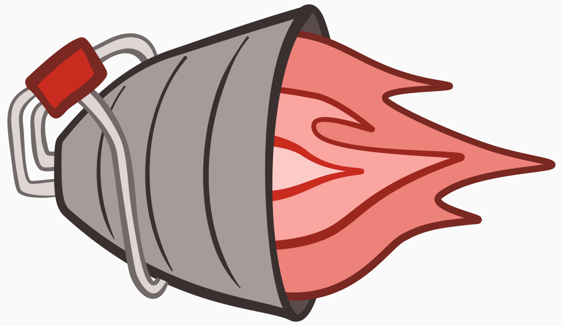


Scheduling rule: always serve job of *minimum **boosted** arrival time*

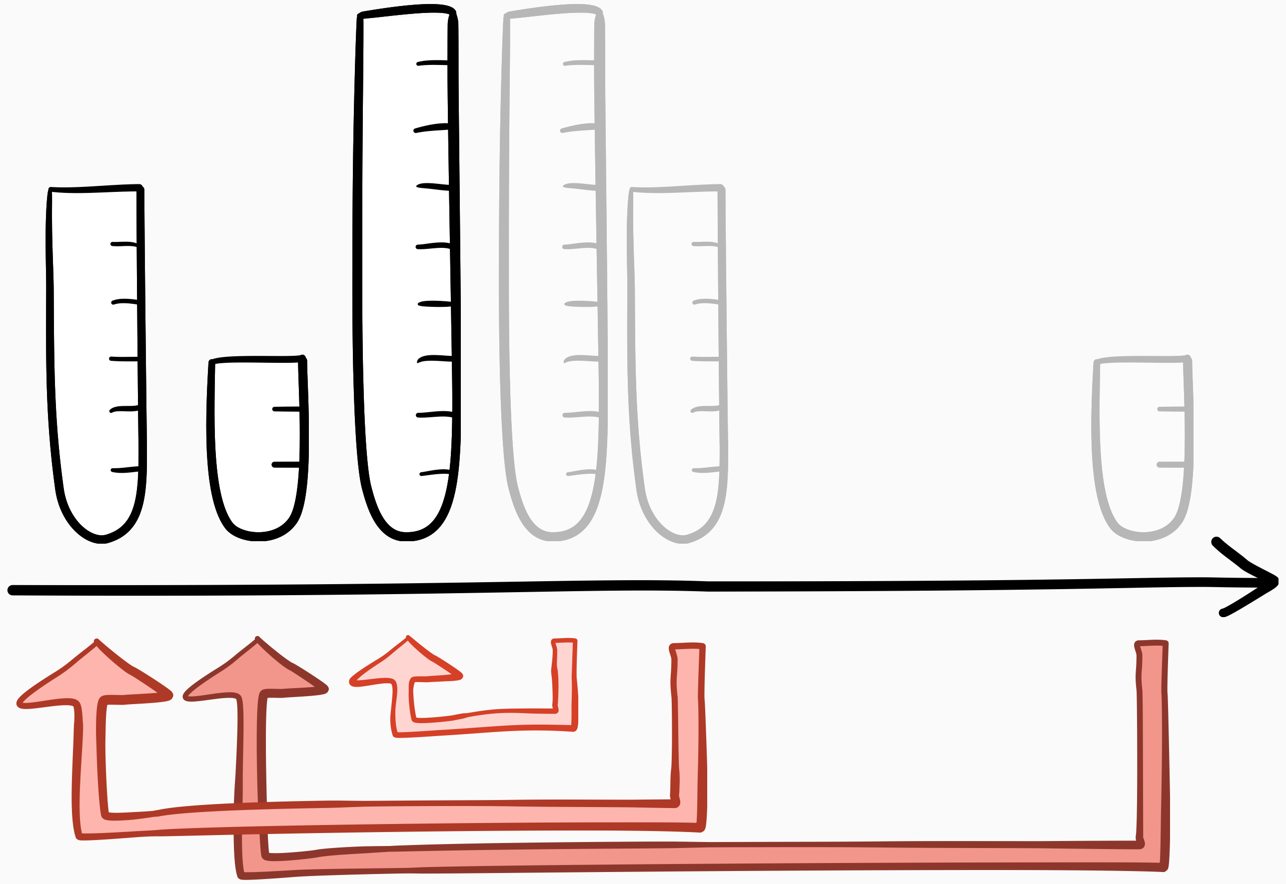


boosted arrival time
= arrival time − **boost**(size)

Boost policies



Scheduling rule: always serve job of
*minimum **boosted** arrival time*

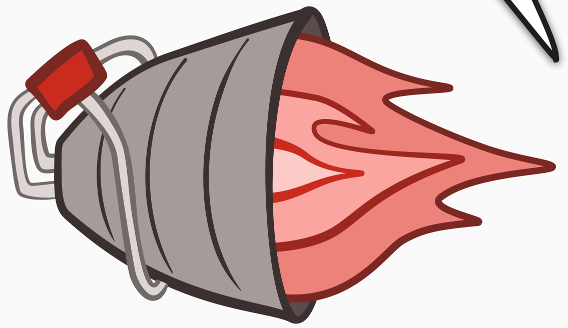


boosted arrival time
= arrival time − **boost**(size)

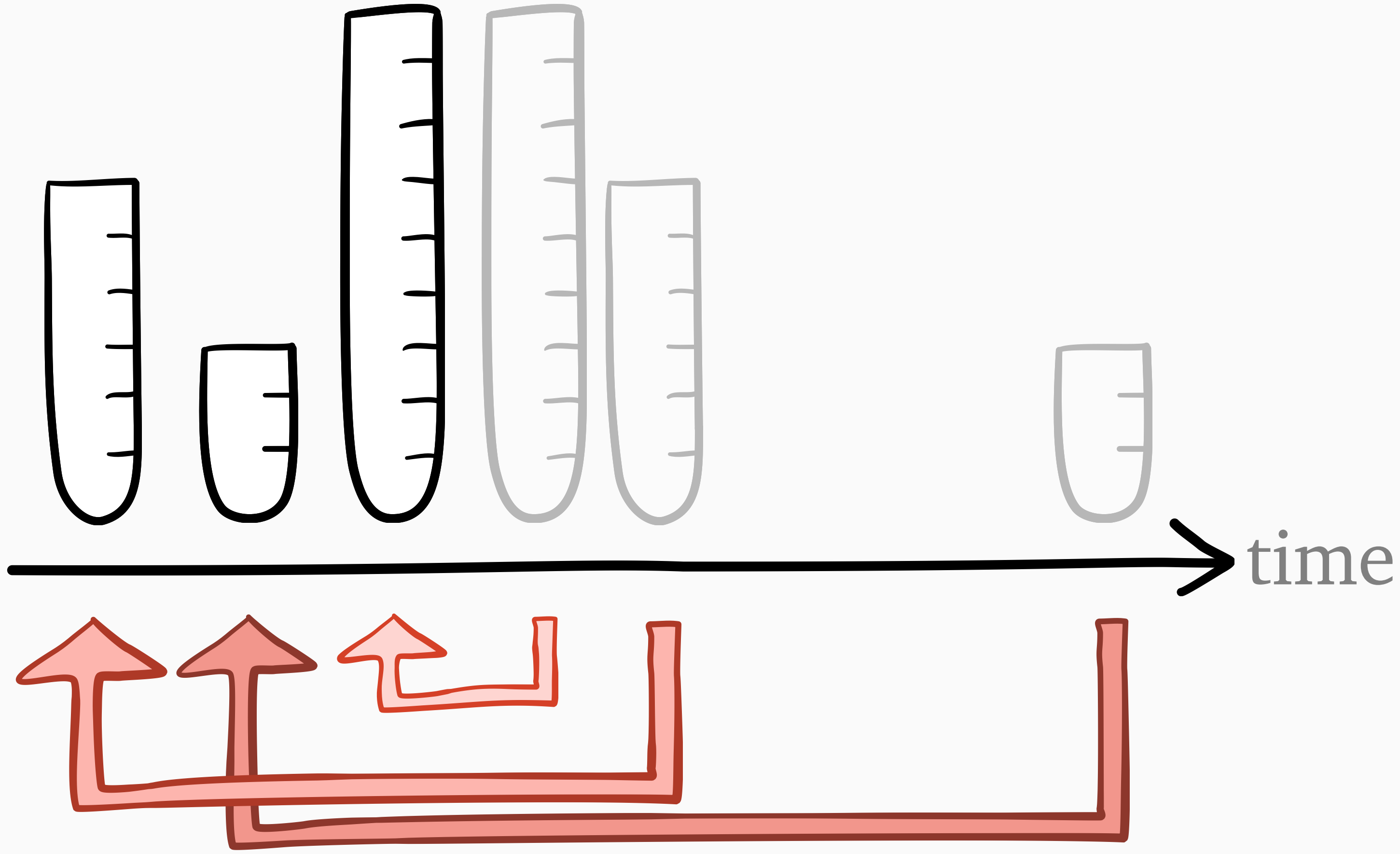
can vary choice of
boost function

Boost policies

can be preemptive or nonpreemptive



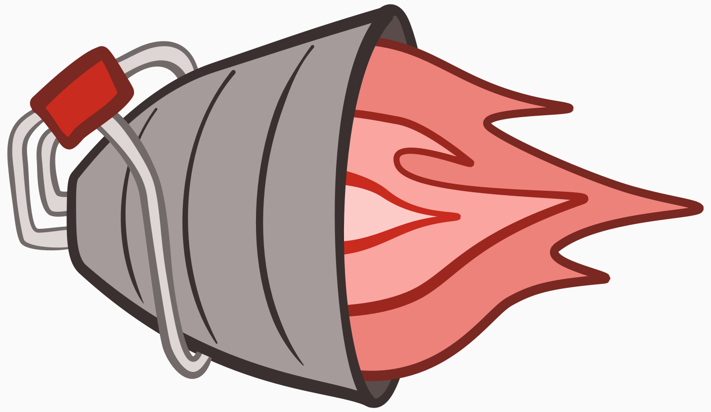
Scheduling rule: always serve job of *minimum boosted arrival time*



boosted arrival time
= arrival time − boost(size)

can vary choice of boost function

Boost

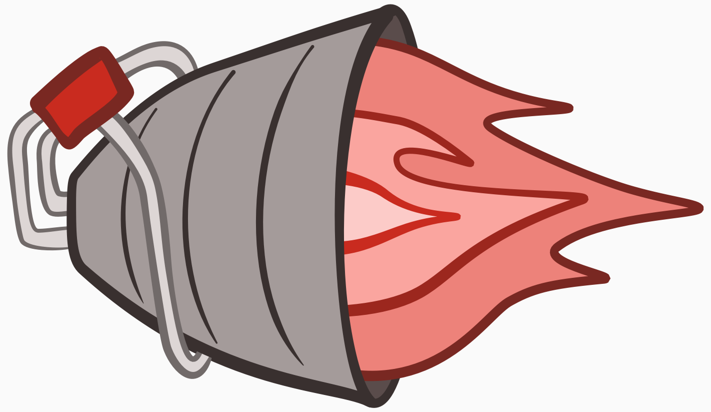


Why is achieving strong tail optimality hard?

How does the **Boost** policy family work?

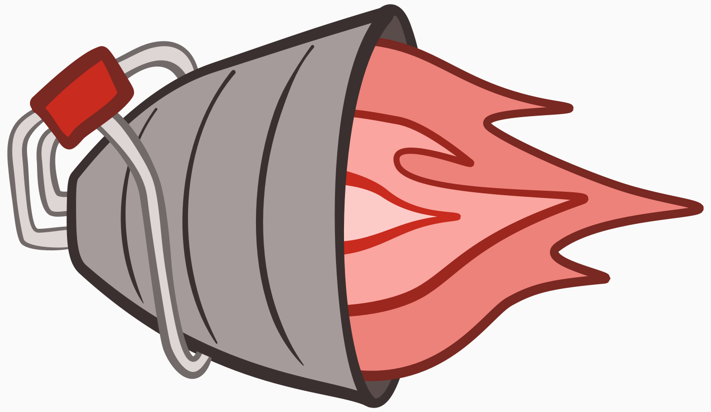
How do we achieve strong tail optimality?

Boost



- Why is achieving strong tail optimality hard?
- How does the **Boost** policy family work?
- How do we achieve strong tail optimality?

Boost



Why is achieving strong tail optimality hard?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Is **Boost** *weakly* tail optimal?

Is **Boost** *weakly* tail optimal?

$\mathbf{P}[T > t] \sim Ce^{-\gamma t}$

Is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t}$$

$$C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

Is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

Is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

Is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t \to \infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

Is Boost *weakly* tail optimal?

$\mathbf{P}[T > t] \sim Ce^{-\gamma t}$ ⇒ $C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W > t]$

Is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$$\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W > t]$$

Boost

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

Is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W > t]$

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

Is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W > t]$

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

Is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W > t]$

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S - b(S))}]\,\mathbf{E}[e^{\gamma V}]$$

Is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Crossing work

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

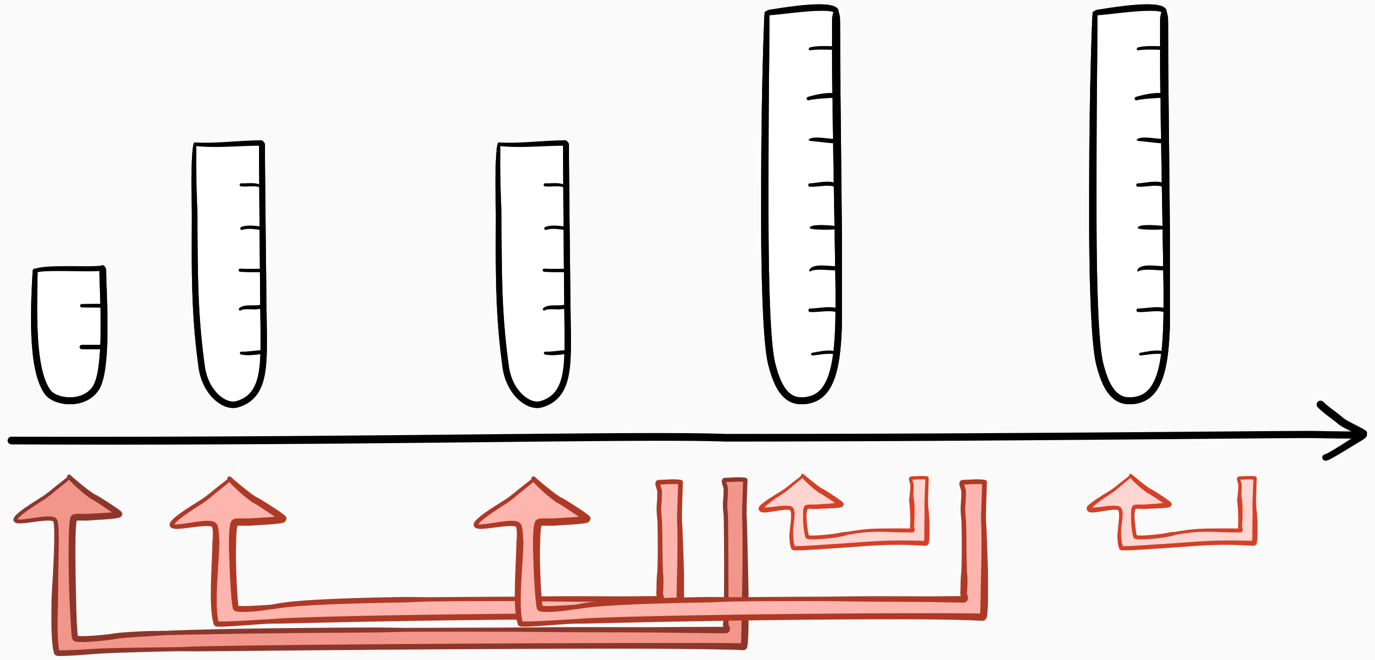
$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\,\mathbf{E}[e^{\gamma V}]$$

Is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Crossing work



Boost boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

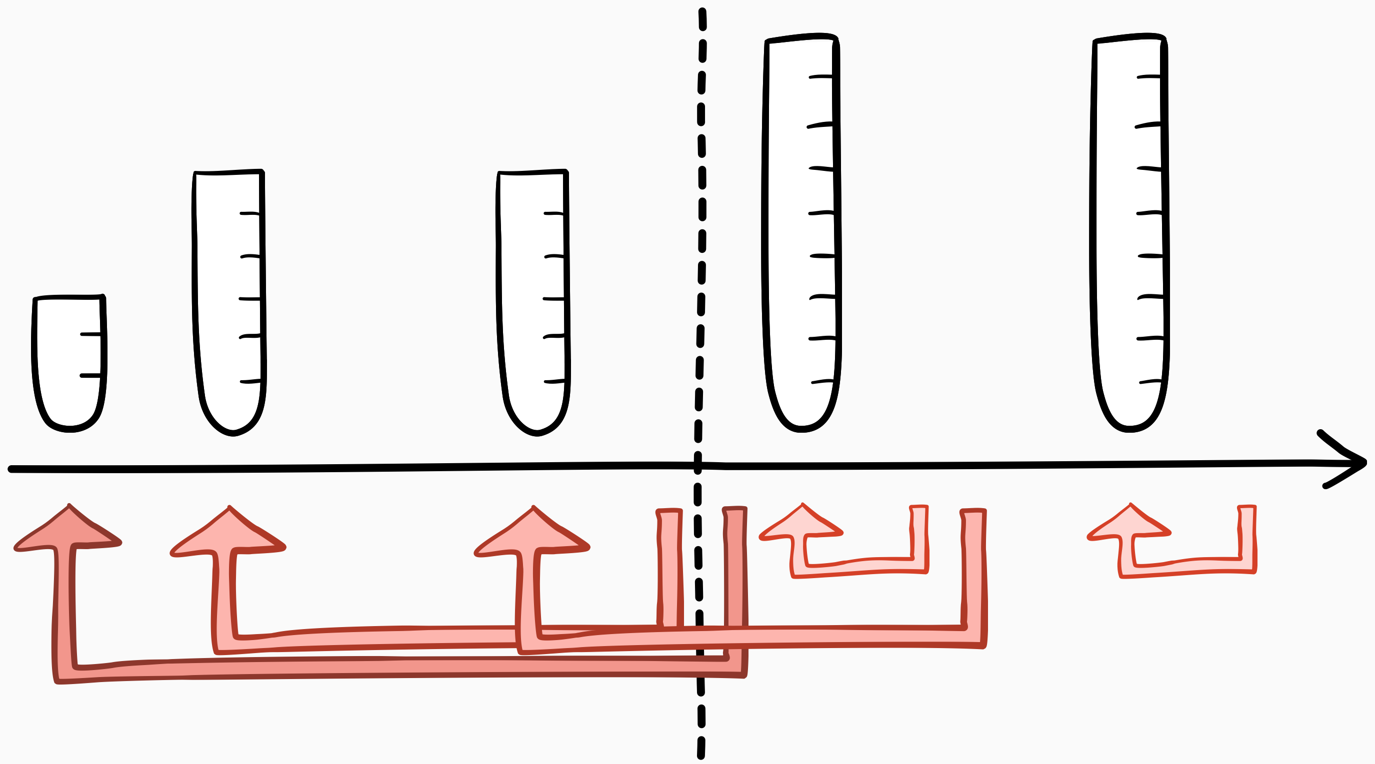
$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\,\mathbf{E}[e^{\gamma V}]$$

Is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Crossing work



Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

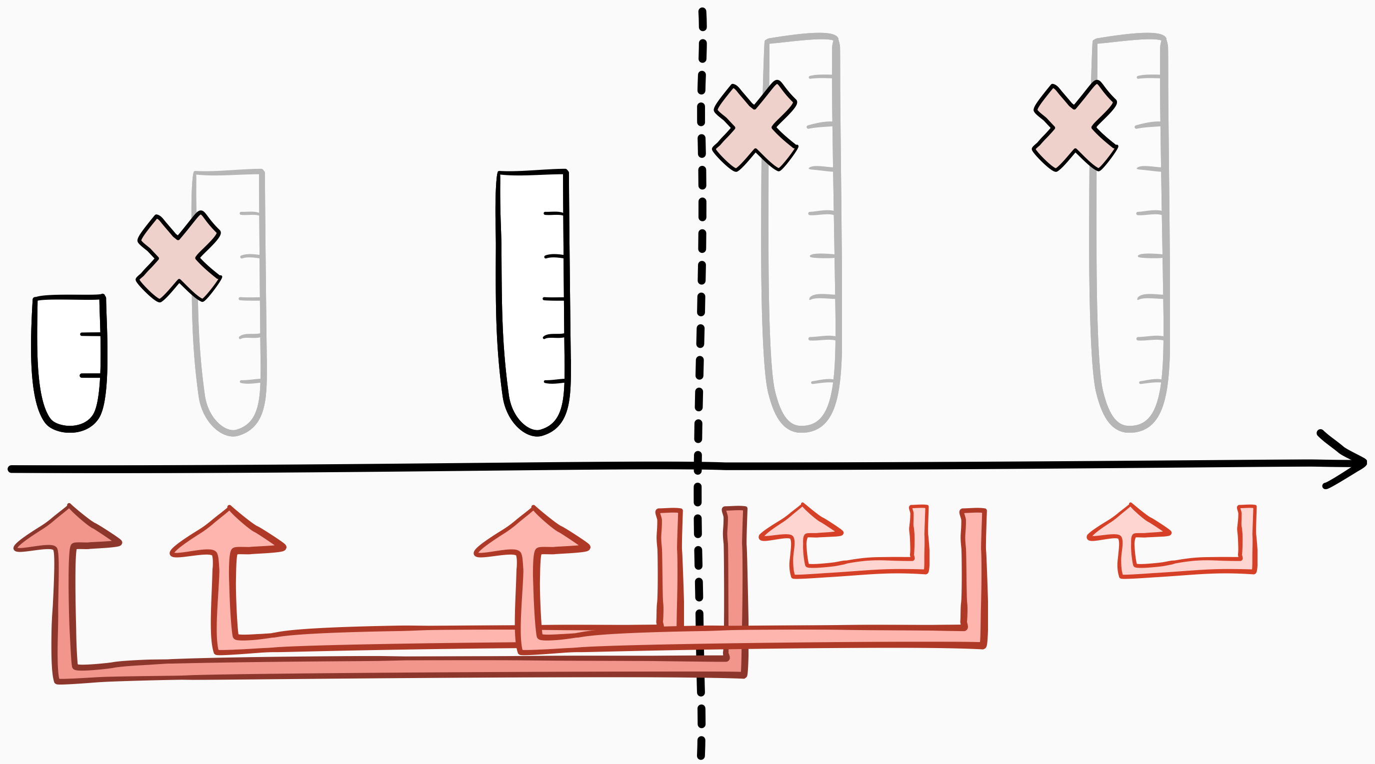
$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\,\mathbf{E}[e^{\gamma V}]$$

Is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Crossing work



Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

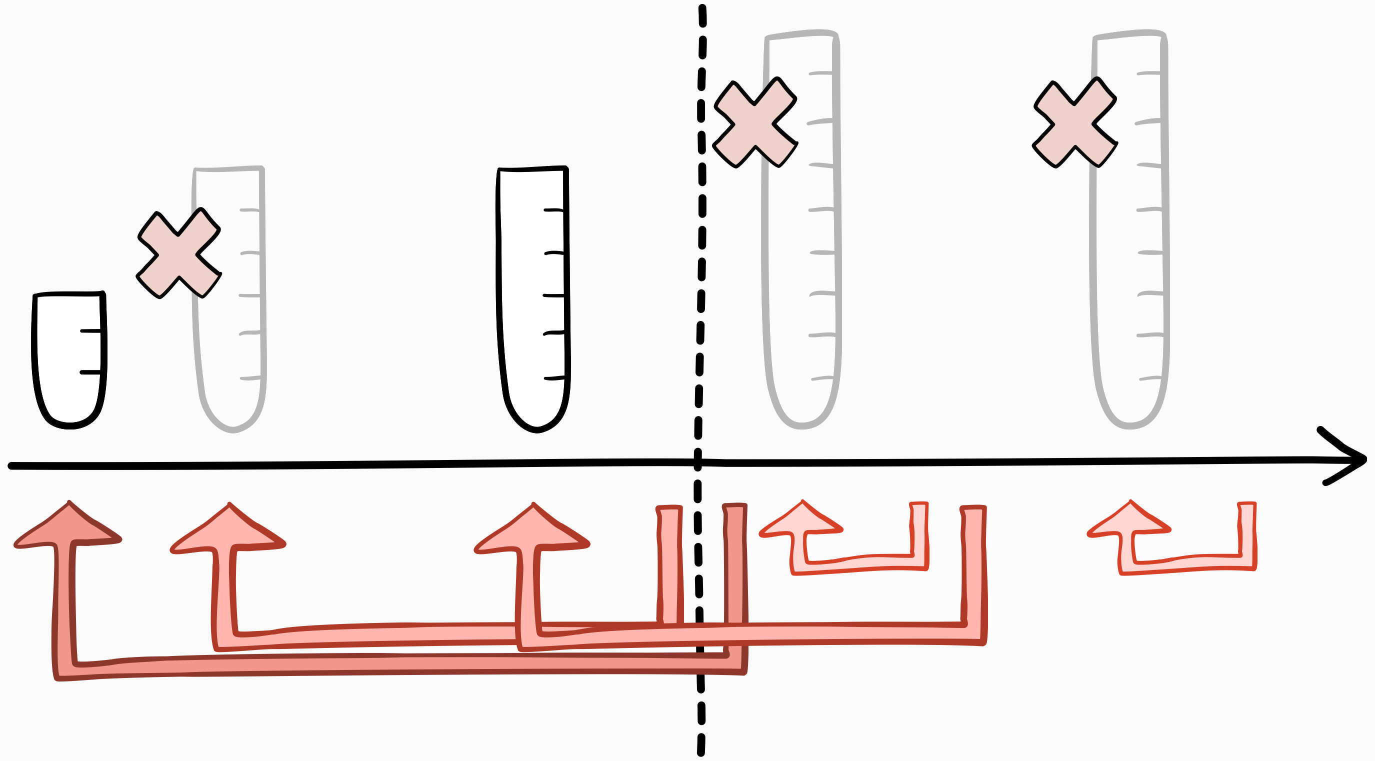
$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\,\mathbf{E}[e^{\gamma V}]$$

Is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

final value theorem

Crossing work



Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S - b(S))}] \mathbf{E}[e^{\gamma V}]$$

Lemma: finite if $b(s) = O(1/s)$

A path to strong tail optimality

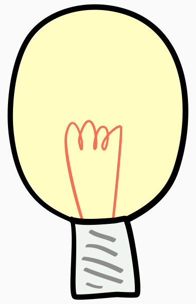
$$C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

A path to strong tail optimality

$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \underbrace{\lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]}_{\text{“}0 \cdot \mathbf{E}[e^{\gamma T}]\text{”}}$$

A path to strong tail optimality

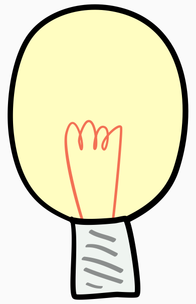
$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \underbrace{\lim_{\theta\to\gamma}\frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]}_{\text{“}0\cdot\mathbf{E}[e^{\gamma T}]\text{”}}$$



Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$...

A path to strong tail optimality

$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \underbrace{\lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]}_{\text{“}0 \cdot \mathbf{E}[e^{\gamma T}]\text{”}}$$

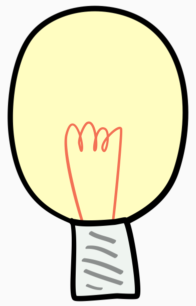


Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

A path to strong tail optimality

$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \underbrace{\lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]}_{\text{“}0\cdot\mathbf{E}[e^{\gamma T}]\text{”}}$$



Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

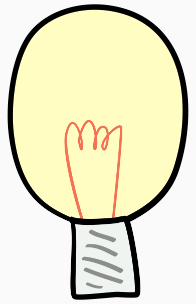
$t_i = d_i - a_i$

a_i = arrival time of job i

d_i = departure time of job i

A path to strong tail optimality

$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \underbrace{\lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]}_{\text{“}0\cdot\mathbf{E}[e^{\gamma T}]\text{”}}$$



Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

$t_i = d_i - a_i$

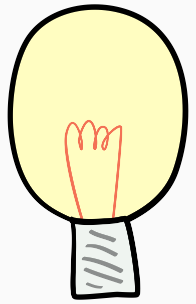
$a_i =$ arrival time of job i

$d_i =$ departure time of job i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i}e^{\gamma d_i}$$

A path to strong tail optimality

$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \underbrace{\lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]}_{\text{“}0\cdot\mathbf{E}[e^{\gamma T}]\text{”}}$$



Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

$t_i = d_i - a_i$

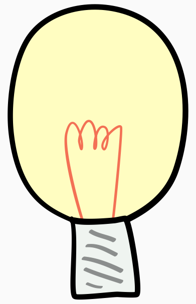
a_i = arrival time of job i

d_i = departure time of job i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

almost classic problem

Almost classic



Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

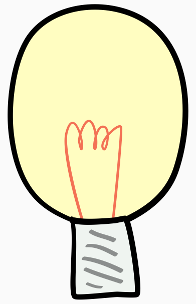
$t_i = d_i - a_i$

$a_i =$ arrival time of job i

$d_i =$ departure time of job i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

Almost classic



Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$...

... which we can turn into a finite batch problem!

$t_i = d_i - a_i$

$a_i =$ arrival time of job i

$d_i =$ departure time of job i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

Mean weighted discounted departure time: $\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$

Almost classic

Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

$t_i = d_i - a_i$

a_i = arrival time of job i

d_i = departure time of job i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

Mean weighted discounted departure time: $\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$

Almost classic

Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

$t_i = d_i - a_i$

a_i = arrival time of job i

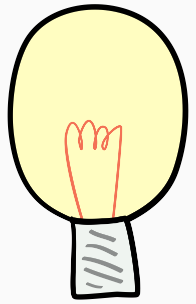
d_i = departure time of job i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

Mean weighted discounted departure time: $\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$

$-\theta < 0$

Almost classic



Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

$t_i = d_i - a_i$

a_i = arrival time of job i

d_i = departure time of job i

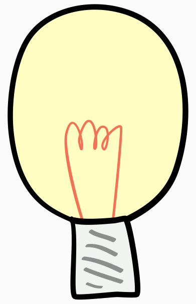
$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

$\gamma > 0$

Mean weighted discounted departure time: $\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$

$-\theta < 0$

Almost classic



Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

can't start i before a_i

$$t_i = d_i - a_i$$

a_i = arrival time of job i

d_i = departure time of job i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

$\gamma > 0$

Mean weighted discounted departure time: $\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$

$-\theta < 0$

Almost classic

Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

can't start i before a_i

$t_i = d_i - a_i$
$a_i =$ arrival time of job i
$d_i =$ departure time of job i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

$\gamma > 0$

Relaxation solved by WDSPT, which is **Boost** with

$$b(s) = \frac{1}{\gamma}\log\frac{1}{1-e^{-\gamma s}}$$

Mean weighted discounted departure time: $\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$

$-\theta < 0$

Almost classic

Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

can't start i before a_i

$t_i = d_i - a_i$
$a_i =$ arrival time of job i
$d_i =$ departure time of job i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

$\gamma > 0$

Relaxation solved by WDSPT, which is **Boost** with

$$b(s) = \frac{1}{\gamma}\log\frac{1}{1 - e^{-\gamma s}}$$

***γ*-Boost**

Mean weighted discounted departure time: $\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$

$-\theta < 0$

Almost classic

Minimizing C is like minimizing $\mathbf{E}[e^{\gamma T}]$…

… which we can turn into a finite batch problem!

can't start i before a_i

$t_i = d_i - a_i$

$a_i =$ arriva

$d_i =$ depar

Unknown sizes:
swap WDSPT for Gittins

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

$\gamma > 0$

Relaxation solved by WDSPT, which is **Boost** with

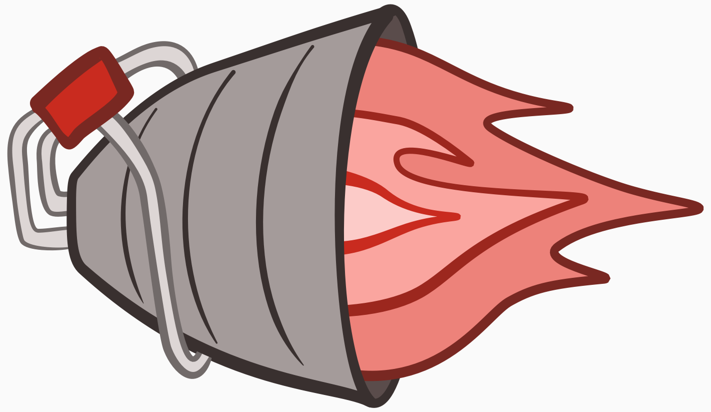
$$b(s) = \frac{1}{\gamma}\log\frac{1}{1 - e^{-\gamma s}}$$

γ-Boost

Mean weighted discounted departure time: $\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$

$-\theta < 0$

Boost



Why is achieving strong tail optimality hard?

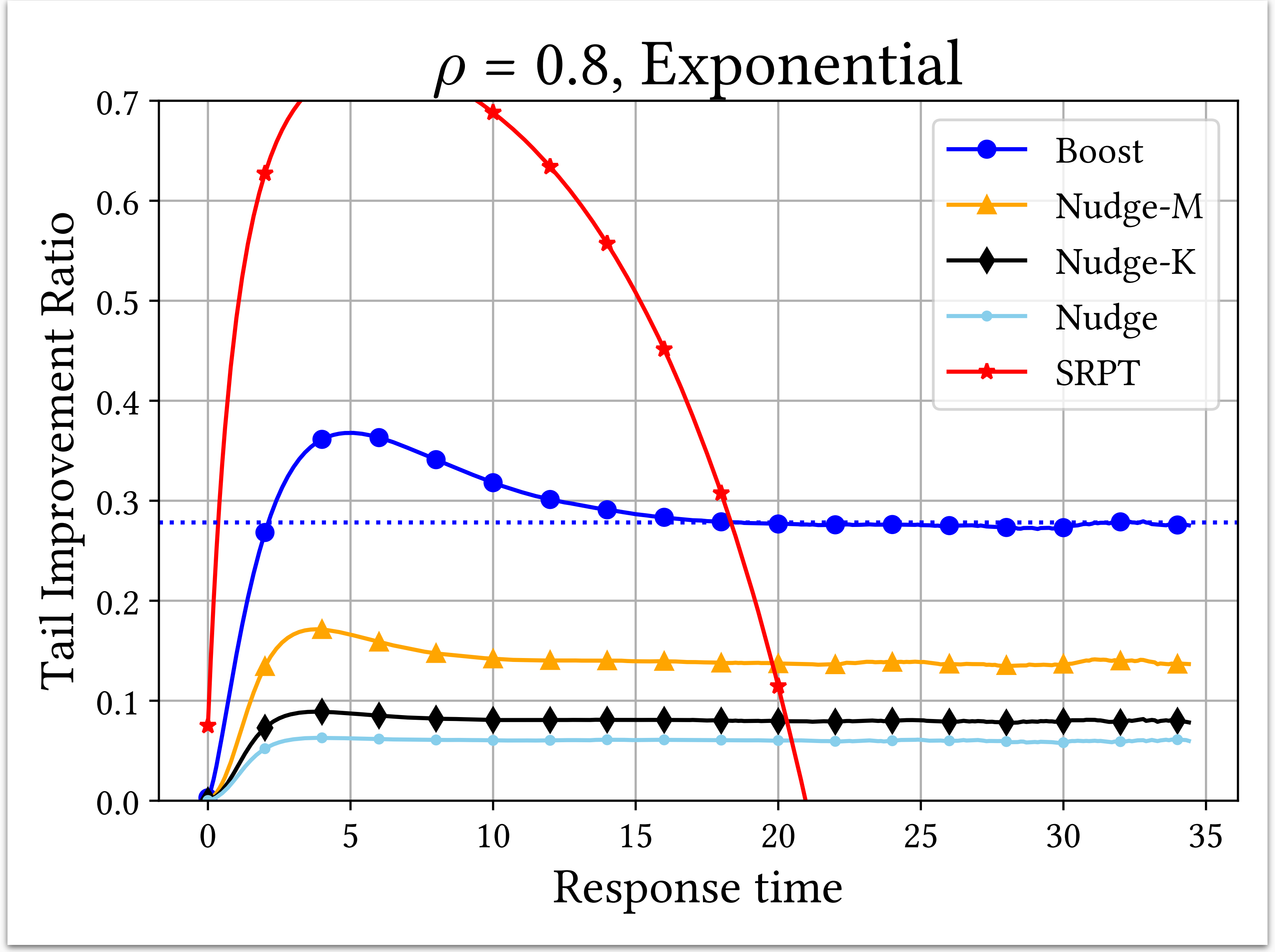
How does the **Boost** policy family work?

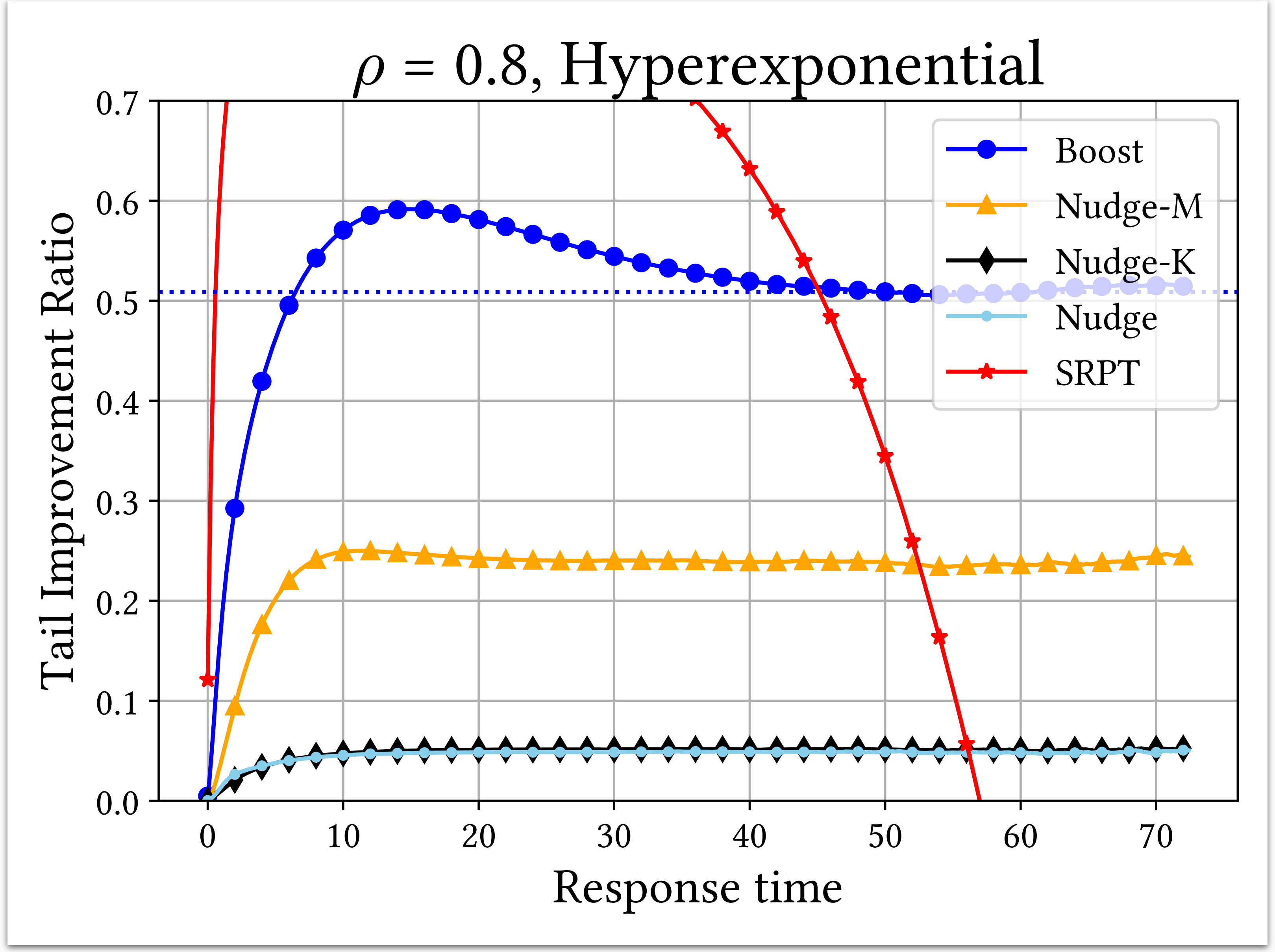
How do we achieve strong tail optimality?

Boost

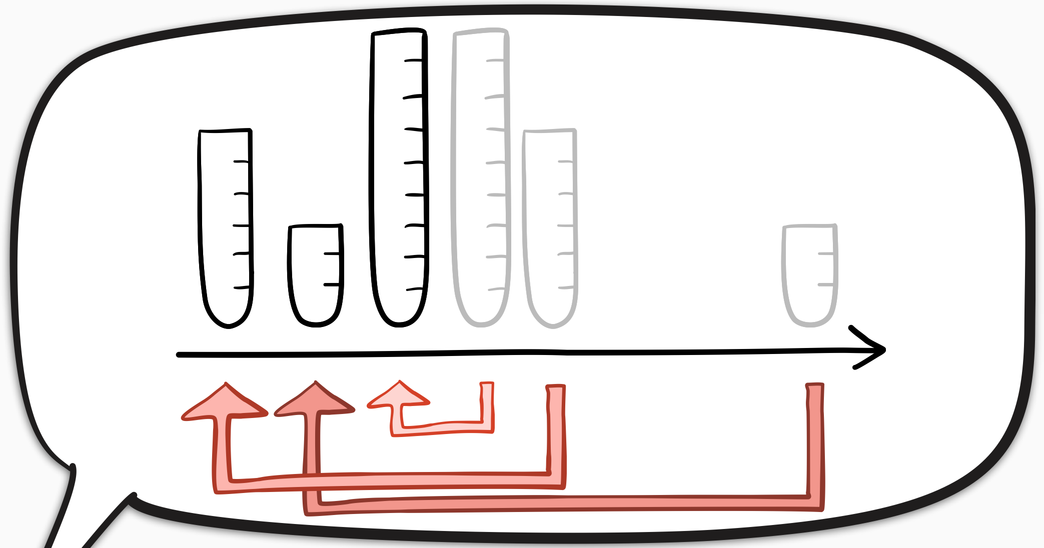
- Why is achieving strong tail optimality hard?
- How does the **Boost** policy family work?
- How do we achieve strong tail optimality?

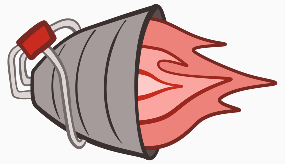
Empirical performance

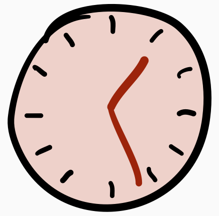




Our contributions:



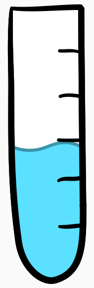
 Design the **Boost** scheduling policy

 Analyze **Boost**’s performance

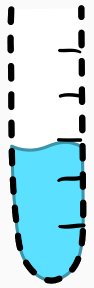
all instances

specific instance called **γ-Boost**

 Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

 **Known job sizes**

Yu & Scully. *Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1.* SIGMETRICS 2024.

 **Unknown job sizes**

Harlev, Yu, & Scully. *A Gittins Policy for Optimizing Tail Latency.* MAMA 2024.