Matching on the line admits no $o(\sqrt{\log n})$-competitive algorithm

Enoch Peserico and **Michele Scquizzato**
University of Padova

ICALP 2021

Online Metric Matching

- **Input**:
  - A metric space, with n points designated as *servers*
  - One by one, n *requests* arrive at arbitrary points
- **Task**: Match each request to a yet unmatched server, minimizing the total request-server distance

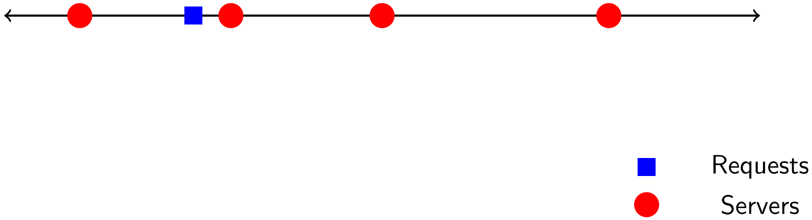
Introduced in 1991 by Khuller, Mitchell, and Vazirani, and independently by Kalyanasundaram and Pruhs

Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height

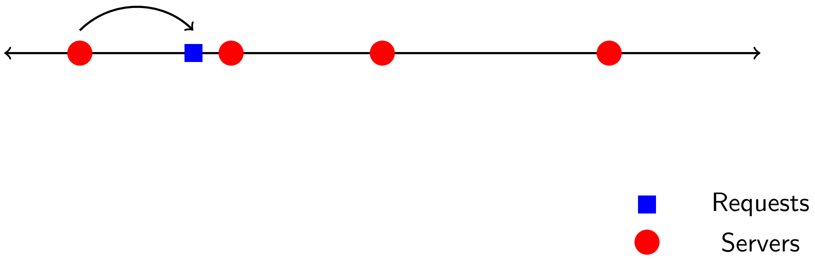
Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height



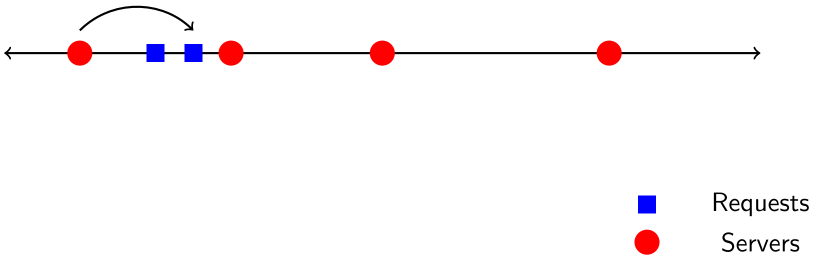
Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height



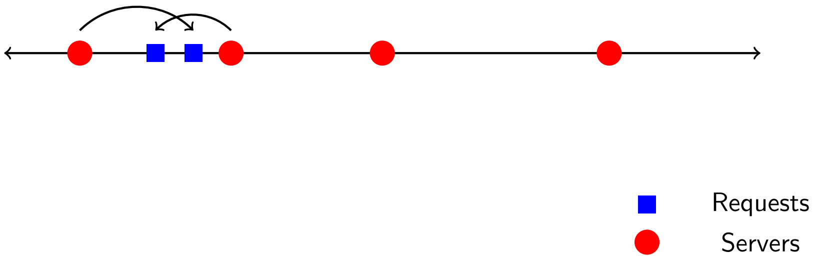
Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height



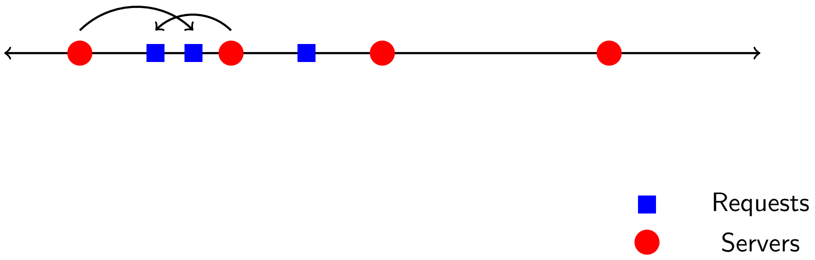
Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height



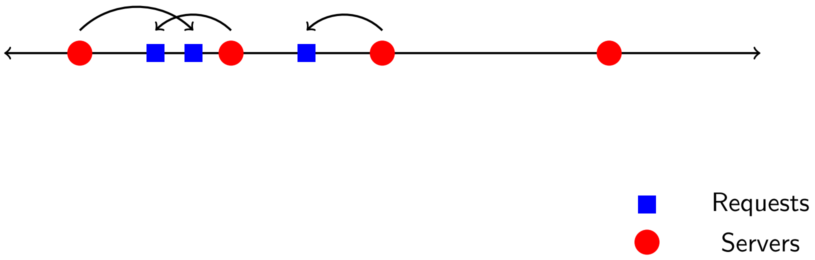
Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height



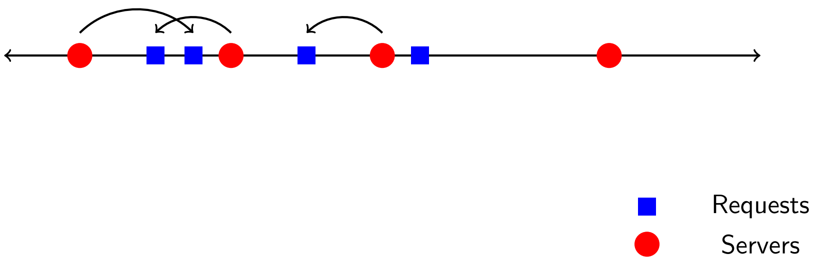
Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height



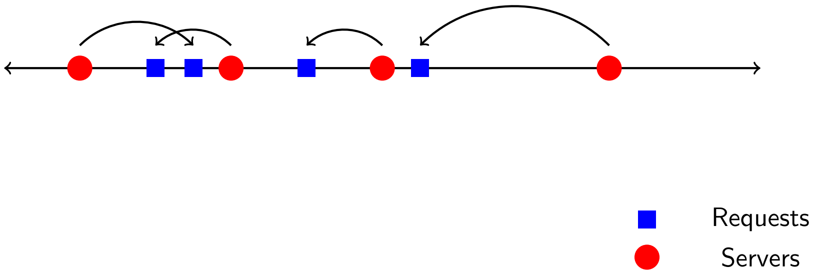
Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height



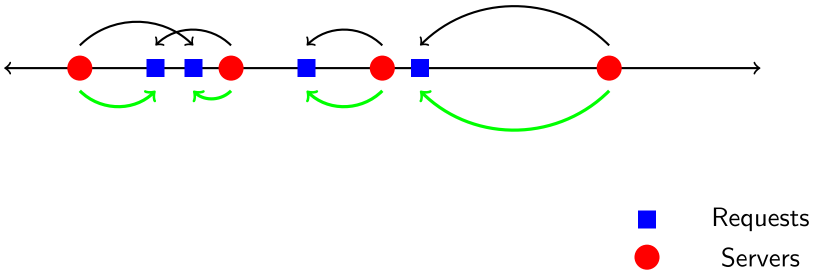
Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height



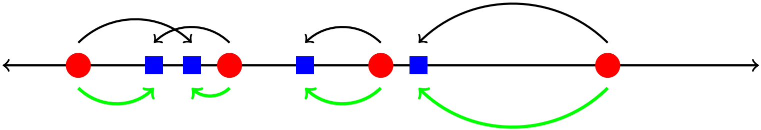
Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height

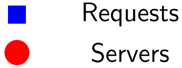


Online Metric Matching on the Line

- Considered the most interesting special case, investigated since 1996
- Example: matching skiers to skis of approximately their height



Competitive ratio: $\max_I \frac{ALG(I)}{OPT(I)}$

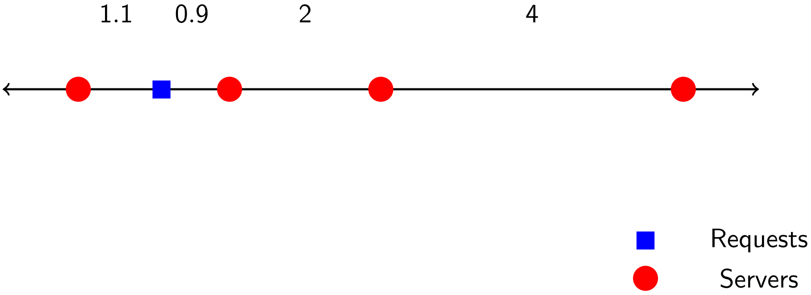


Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server

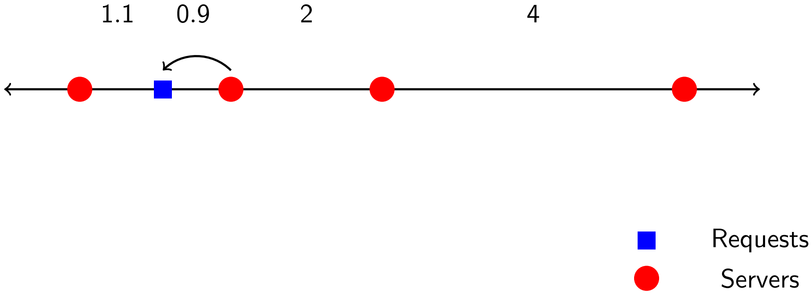
Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server



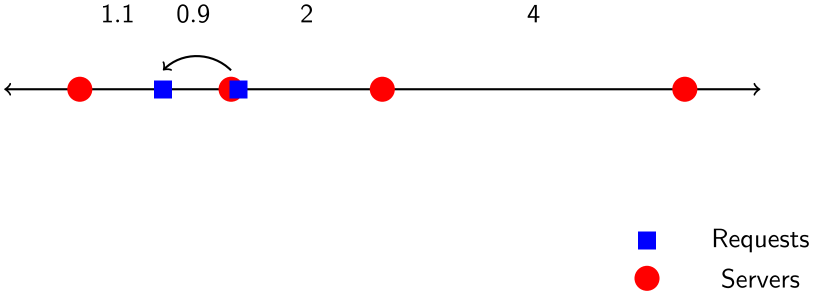
Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server



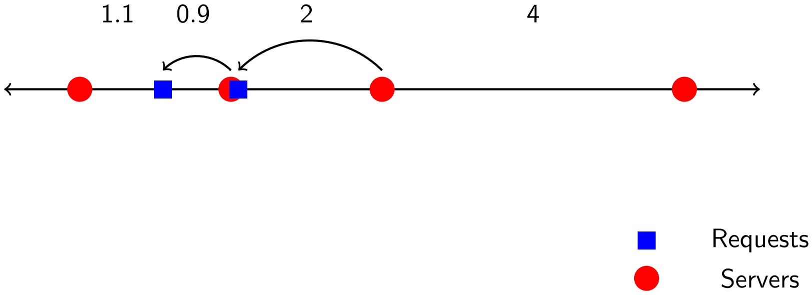
Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server



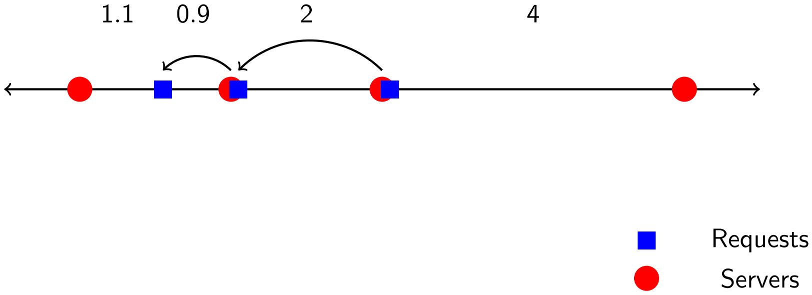
Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server



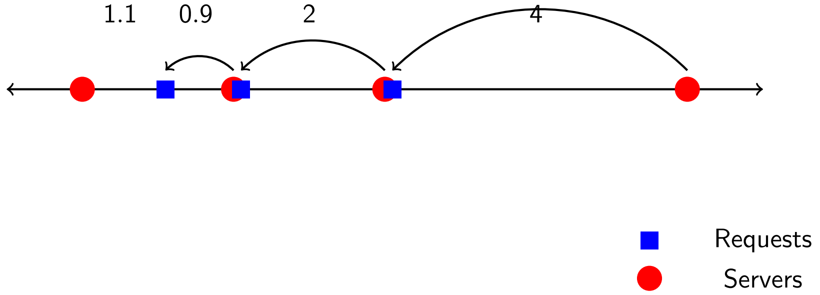
Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server



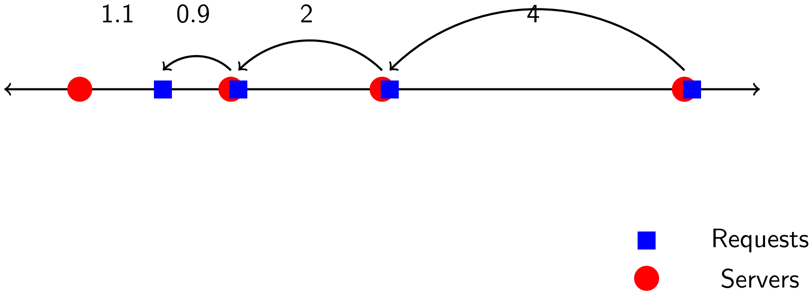
Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server



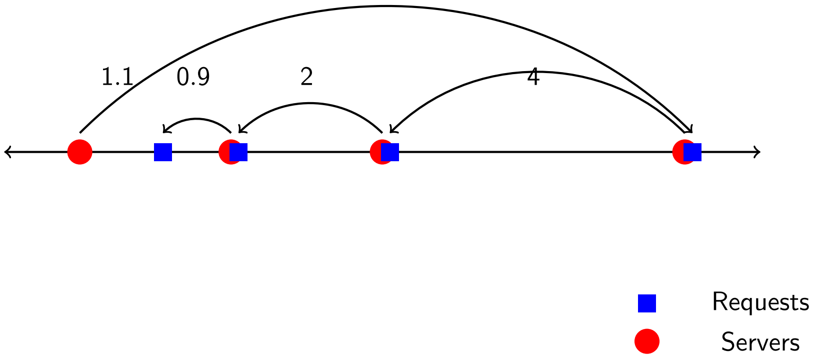
Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server



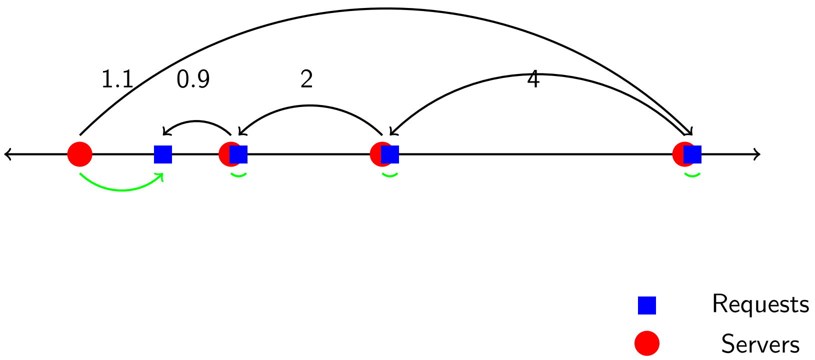
Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server



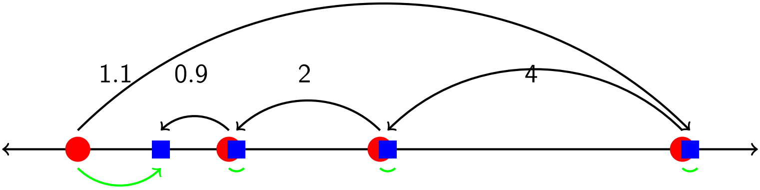
Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server

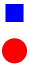


Matching on the Line: How Difficult can it Be?

Greedy algorithm: match each request to the closest server

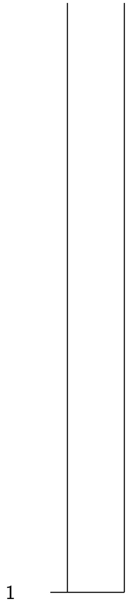


Competitive ratio: $\Omega(2^n)$

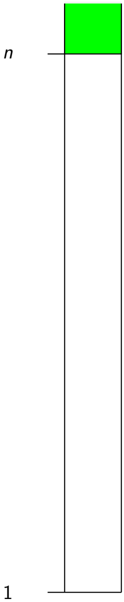


The Story so Far

Randomized **Deterministic**

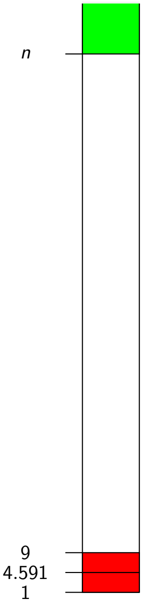


The Story so Far



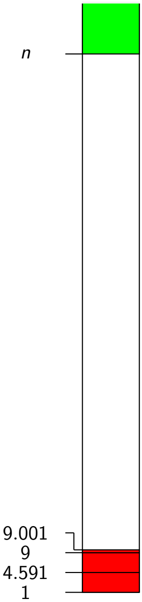
Randomized	**Deterministic**
	Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991

The Story so Far



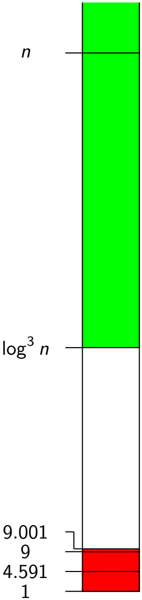
Randomized	Deterministic
	Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
Kalyanasundaram and Pruhs, 1996	Kalyanasundaram and Pruhs, 1996

The Story so Far



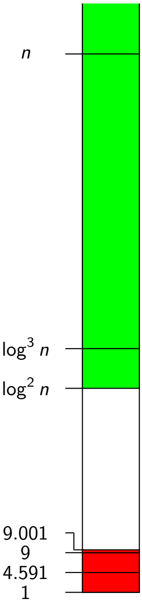
Randomized	**Deterministic**
	Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
	Fuchs et al., 2005 Kalyanasundaram and Pruhs, 1996
Kalyanasundaram and Pruhs, 1996	

The Story so Far



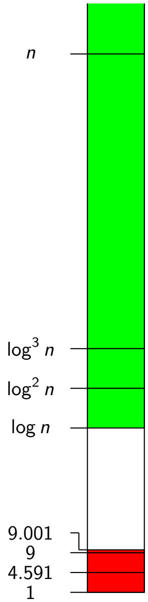
	Randomized	Deterministic
n		Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
$\log^3 n$	Meyerson et al., 2006	
9.001		Fuchs et al., 2005
9		Kalyanasundaram and Pruhs, 1996
4.591	Kalyanasundaram and Pruhs, 1996	

The Story so Far



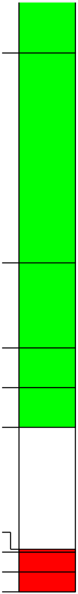
Bound	Randomized	Deterministic
n		Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
$\log^3 n$	Meyerson et al., 2006	
$\log^2 n$	Bansal et al., 2007	
9.001		Fuchs et al., 2005
9		Kalyanasundaram and Pruhs, 1996
4.591	Kalyanasundaram and Pruhs, 1996	

The Story so Far



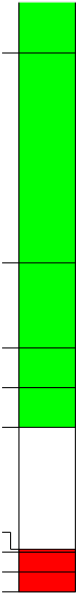
	Randomized	Deterministic
n		Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
$\log^3 n$	Meyerson et al., 2006	
$\log^2 n$	Bansal et al., 2007	
$\log n$	Gupta and Lewi, 2012	
9.001		Fuchs et al., 2005
9		Kalyanasundaram and Pruhs, 1996
4.591	Kalyanasundaram and Pruhs, 1996	

The Story so Far



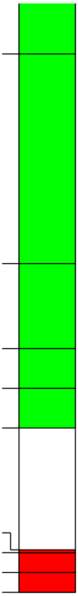
	Randomized	Deterministic
n		Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
$n^{0.58}$		Antoniadis et al., 2014
$\log^3 n$	Meyerson et al., 2006	
$\log^2 n$	Bansal et al., 2007	
$\log n$	Gupta and Lewi, 2012	
9.001		Fuchs et al., 2005
9		Kalyanasundaram and Pruhs, 1996
4.591	Kalyanasundaram and Pruhs, 1996	
1		

The Story so Far



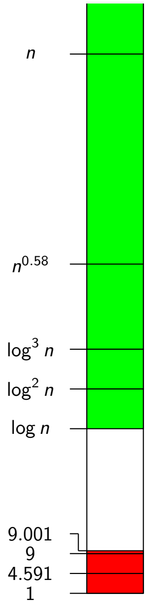
	Randomized	Deterministic
n		Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
$n^{0.58}$		Antoniadis et al., 2014
$\log^3 n$	Meyerson et al., 2006	
$\log^2 n$	Bansal et al., 2007	Nayyar and Raghvendra, 2017
$\log n$	Gupta and Lewi, 2012	
9.001		Fuchs et al., 2005
9		Kalyanasundaram and Pruhs, 1996
4.591	Kalyanasundaram and Pruhs, 1996	
1		

The Story so Far



Bound	Randomized	Deterministic
n		Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
$n^{0.58}$		Antoniadis et al., 2014
$\log^3 n$	Meyerson et al., 2006	
$\log^2 n$	Bansal et al., 2007	Nayyar and Raghvendra, 2017
$\log n$	Gupta and Lewi, 2012	Raghvendra, 2018
9.001		Fuchs et al., 2005
9		Kalyanasundaram and Pruhs, 1996
4.591	Kalyanasundaram and Pruhs, 1996	
1		

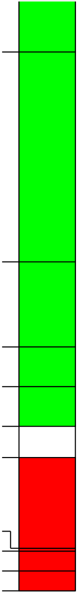
The Story so Far



Bound	Randomized	Deterministic
n		Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
$n^{0.58}$		Antoniadis et al., 2014
$\log^3 n$	Meyerson et al., 2006	
$\log^2 n$	Bansal et al., 2007	Nayyar and Raghvendra, 2017
$\log n$	Gupta and Lewi, 2012	Raghvendra, 2018
9		Fuchs et al., 2005 Kalyanasundaram and Pruhs, 1996
4.591	Kalyanasundaram and Pruhs, 1996	

Is there an $O(1)$-competitive online algorithm for the line?

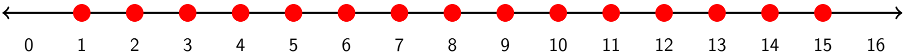
The Story so Far



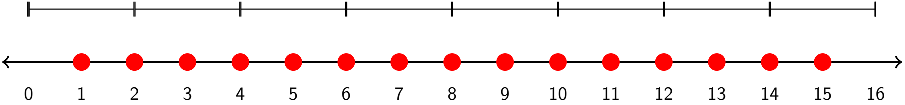
Bound	Randomized	Deterministic
n		Kalyanasundaram and Pruhs, 1991 Khuller et al., 1991
$n^{0.58}$		Antoniadis et al., 2014
$\log^3 n$	Meyerson et al., 2006	
$\log^2 n$	Bansal et al., 2007	Nayyar and Raghvendra, 2017
$\log n$	Gupta and Lewi, 2012	Raghvendra, 2018
$\boldsymbol{\sqrt{\log n}}$	**Peserico and Scquizzato, 2021**	
9.001		Fuchs et al., 2005
9		Kalyanasundaram and Pruhs, 1996
4.591	Kalyanasundaram and Pruhs, 1996	
1		

Is there an $O(1)$-competitive online algorithm for the line?

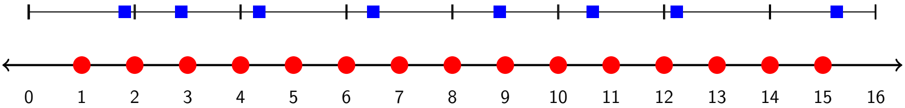
An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



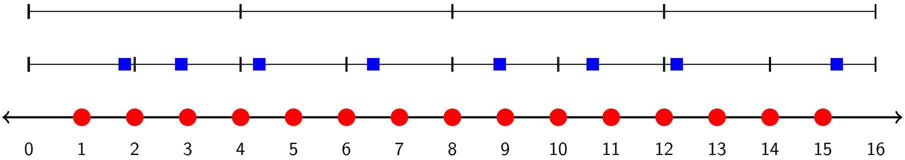
An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



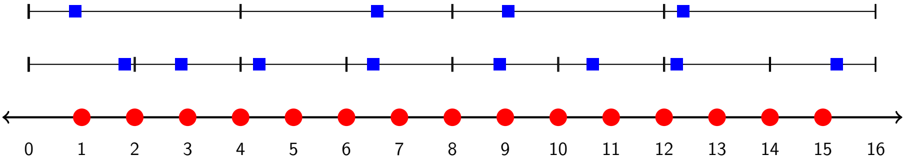
An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



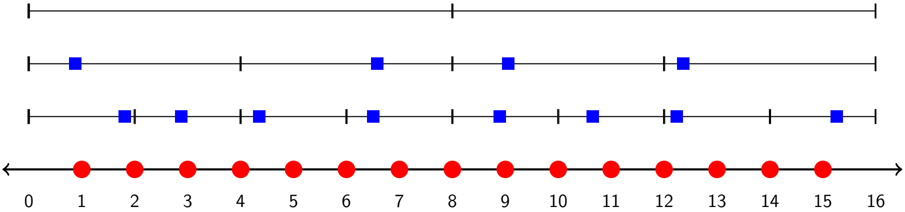
An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



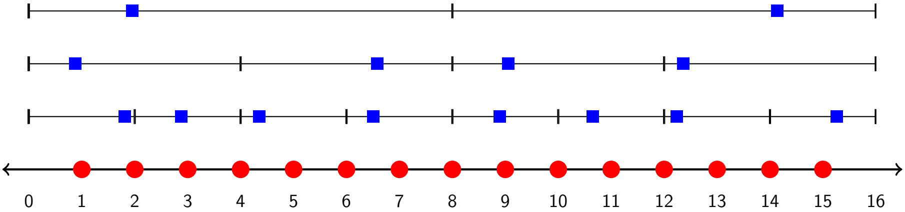
An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



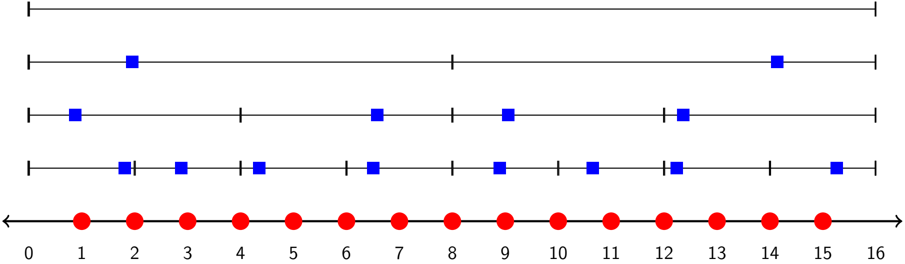
An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



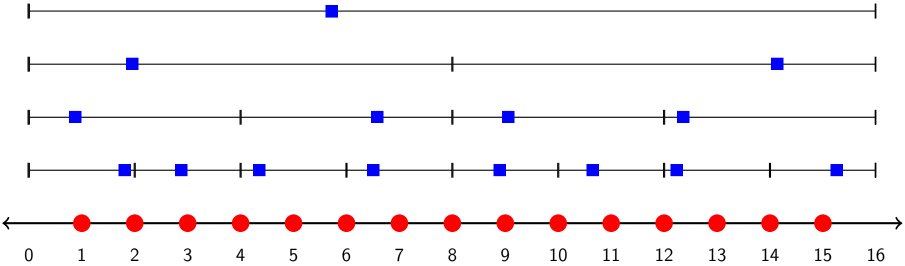
An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



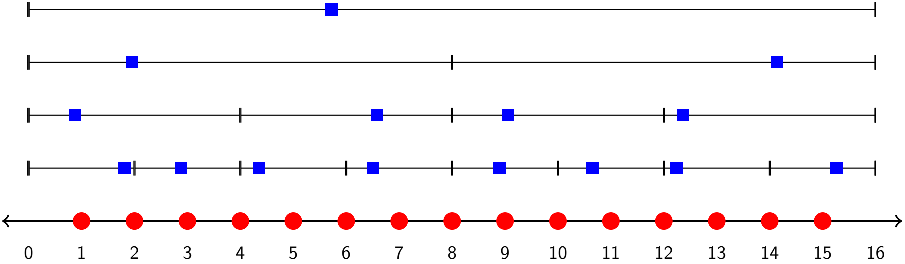
An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



An $\Omega(\sqrt{\log n})$ Lower Bound - The Adversarial Input



Note: the adversarial input does *not* depend on ALG!

An $\Omega(\sqrt{\log n})$ Lower Bound - Roadmap

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

An $\Omega(\sqrt{\log n})$ Lower Bound - Roadmap

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

An $\Omega(\sqrt{\log n})$ Lower Bound - Roadmap

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Then:

$$\frac{\mathrm{E}[ALG]}{\mathrm{E}[OPT]} = \frac{\Omega(n \log n)}{O(n\sqrt{\log n})} = \Omega(\sqrt{\log n})$$

An $\Omega(\sqrt{\log n})$ Lower Bound - Roadmap

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Then:

$$\frac{\mathrm{E}[ALG]}{\mathrm{E}[OPT]} = \frac{\Omega(n \log n)}{O(n\sqrt{\log n})} = \Omega(\sqrt{\log n})$$

Theorem

The competitive ratio of any randomized online matching algorithm for the line exceeds $\sqrt{\log_2(n+1)}/15$ for all $n = 2^i - 1 : i \in \mathbb{N}$.

An $\Omega(\sqrt{\log n})$ Lower Bound - ALG

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

An $\Omega(\sqrt{\log n})$ Lower Bound - ALG

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

Proof sketch:

If unmatched servers are roughly equidistant

An $\Omega(\sqrt{\log n})$ Lower Bound - ALG

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

Proof sketch:

If unmatched servers are roughly equidistant

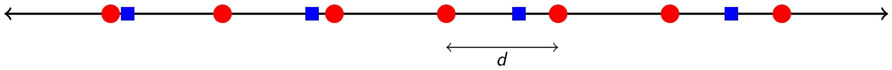
An $\Omega(\sqrt{\log n})$ Lower Bound - ALG

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

Proof sketch:

If unmatched servers are roughly equidistant



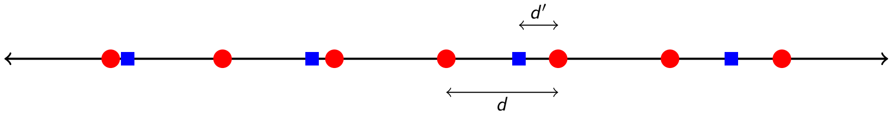
An $\Omega(\sqrt{\log n})$ Lower Bound - ALG

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

Proof sketch:

If unmatched servers are roughly equidistant



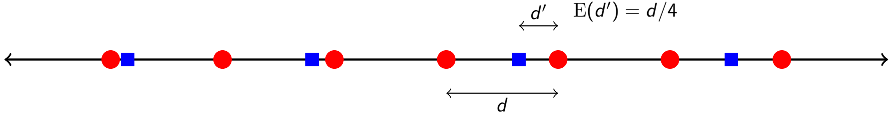
An $\Omega(\sqrt{\log n})$ Lower Bound - ALG

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

Proof sketch:

If unmatched servers are roughly equidistant



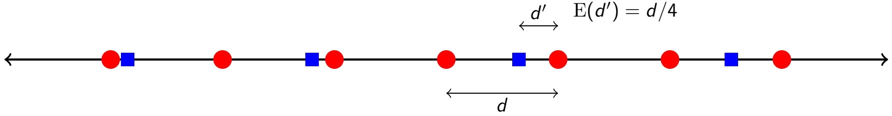
An $\Omega(\sqrt{\log n})$ Lower Bound - ALG

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

Proof sketch:

If unmatched servers are roughly equidistant



$$\mathrm{E}[\mathrm{cost}] = \Omega\left(d\right) \cdot \Omega\left(\frac{n}{d}\right) = \Omega(n)$$

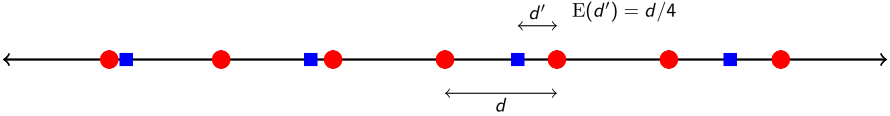
An $\Omega(\sqrt{\log n})$ Lower Bound - ALG

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

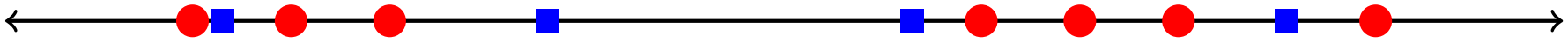
Proof sketch:

If unmatched servers are roughly equidistant



$$\mathrm{E}[\mathrm{cost}] = \Omega\left(d\right) \cdot \Omega\left(\frac{n}{d}\right) = \Omega(n)$$

If not



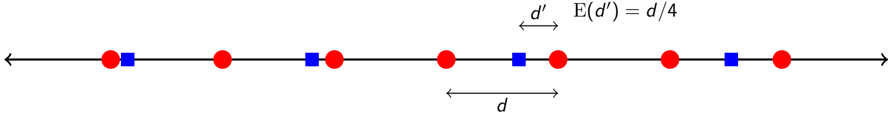
An $\Omega(\sqrt{\log n})$ Lower Bound - ALG

Lemma

Any randomized online matching algorithm ALG incurs an expected $\Omega(n)$ cost in each round.

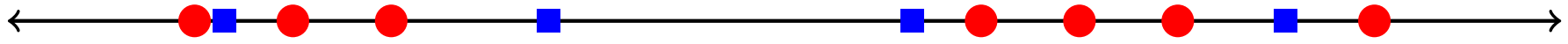
Proof sketch:

If unmatched servers are roughly equidistant



$$\mathrm{E}[\mathrm{cost}] = \Omega\left(d\right) \cdot \Omega\left(\frac{n}{d}\right) = \Omega(n)$$

If not



$\mathrm{E}[\mathrm{cost}]$ can only be higher

An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Proof sketch:

- $d =$ distance between ℓ-th leftmost request and ℓ-th leftmost server
- $r_\ell =$ number of requests to the left of the ℓ-th leftmost server

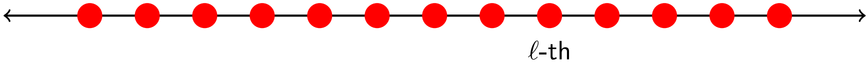
An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Proof sketch:

- d = distance between ℓ-th leftmost request and ℓ-th leftmost server
- r_ℓ = number of requests to the left of the ℓ-th leftmost server
- Observe that $d = O(|r_\ell - \ell|)$



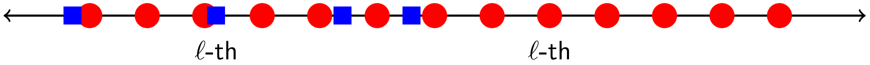
An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Proof sketch:

- $d =$ distance between ℓ-th leftmost request and ℓ-th leftmost server
- $r_\ell =$ number of requests to the left of the ℓ-th leftmost server
- Observe that $d = O(|r_\ell - \ell|)$



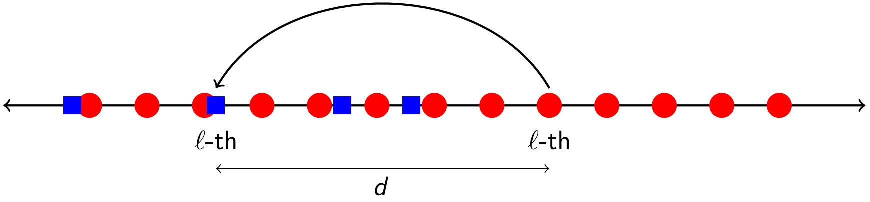
An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Proof sketch:

- $d =$ distance between ℓ-th leftmost request and ℓ-th leftmost server
- $r_\ell =$ number of requests to the left of the ℓ-th leftmost server
- Observe that $d = O(|r_\ell - \ell|)$



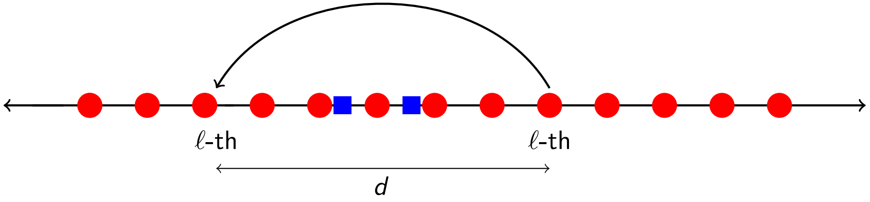
An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Proof sketch:

- $d =$ distance between ℓ-th leftmost request and ℓ-th leftmost server
- $r_\ell =$ number of requests to the left of the ℓ-th leftmost server
- Observe that $d = O(|r_\ell - \ell|)$



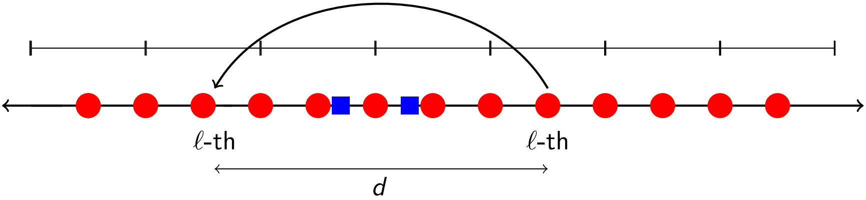
An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Proof sketch:

- $d =$ distance between ℓ-th leftmost request and ℓ-th leftmost server
- $r_\ell =$ number of requests to the left of the ℓ-th leftmost server
- Observe that $d = O(|r_\ell - \ell|)$



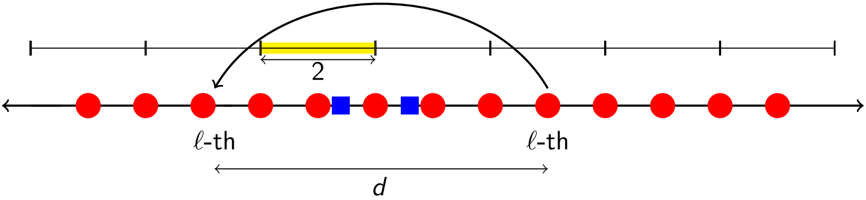
An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Proof sketch:

- $d =$ distance between ℓ-th leftmost request and ℓ-th leftmost server
- $r_\ell =$ number of requests to the left of the ℓ-th leftmost server
- Observe that $d = O(|r_\ell - \ell|)$



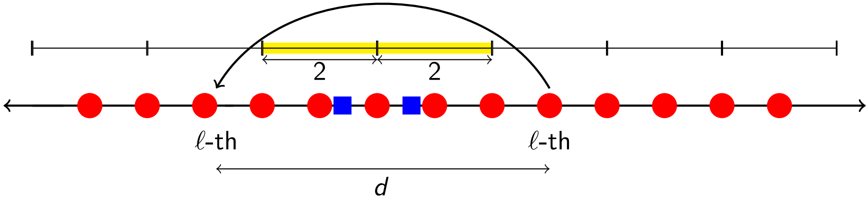
An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Proof sketch:

- $d =$ distance between ℓ-th leftmost request and ℓ-th leftmost server
- $r_\ell =$ number of requests to the left of the ℓ-th leftmost server
- Observe that $d = O(|r_\ell - \ell|)$



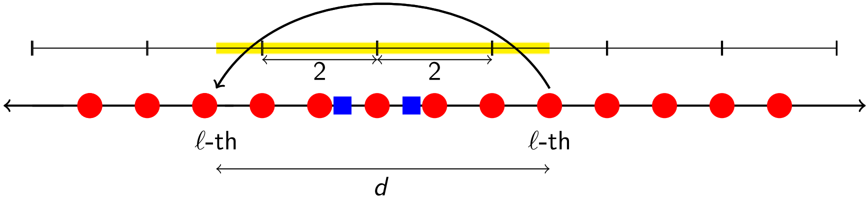
An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

Lemma

The expected distance between the ℓ-th leftmost request and the ℓ-th leftmost server is $O(\sqrt{\log n})$.

Proof sketch:

- $d =$ distance between ℓ-th leftmost request and ℓ-th leftmost server
- $r_\ell =$ number of requests to the left of the ℓ-th leftmost server
- Observe that $d = O(|r_\ell - \ell|)$



An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

- What's the value of r_ℓ?

$$r_\ell = \sum_{i=1}^{n} X_i \qquad X_i = \begin{cases} 1 & i\text{-th request is left of } \ell\text{-th leftmost server} \\ 0 & \text{otherwise} \end{cases}$$

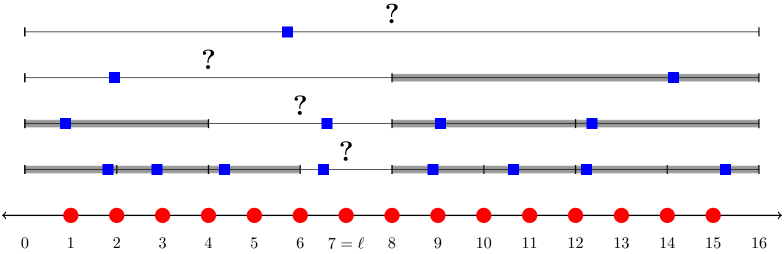
- $\mathrm{E}[r_\ell] \approx \ell$, hence $d = O(|r_\ell - \mathrm{E}[r_\ell]|)$

An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

- What's the value of r_ℓ?

$$r_\ell = \sum_{i=1}^{n} X_i \qquad X_i = \begin{cases} 1 & i\text{-th request is left of } \ell\text{-th leftmost server} \\ 0 & \text{otherwise} \end{cases}$$

- $\mathrm{E}[r_\ell] \approx \ell$, hence $d = O(|r_\ell - \mathrm{E}[r_\ell]|)$
- Only $\log_2(n+1)$ X_i's (one per round) are "truly" random variables

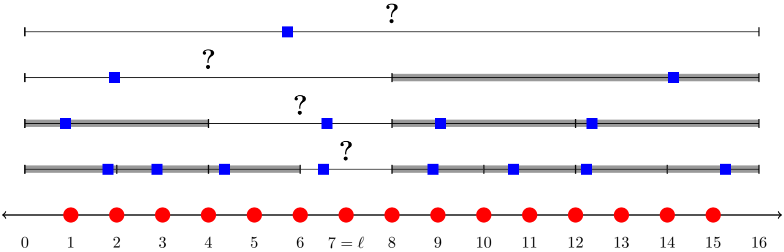


An $\Omega(\sqrt{\log n})$ Lower Bound - OPT

- What's the value of r_ℓ?

$$r_\ell = \sum_{i=1}^{n} X_i \qquad X_i = \begin{cases} 1 & i\text{-th request is left of } \ell\text{-th leftmost server} \\ 0 & \text{otherwise} \end{cases}$$

- $\mathrm{E}[r_\ell] \approx \ell$, hence $d = O(|r_\ell - \mathrm{E}[r_\ell]|)$
- Only $\log_2(n+1)$ X_i's (one per round) are "truly" random variables



- Thus $|r_\ell - \mathrm{E}[r_\ell]|$ is $O(\sqrt{\log n})$ in expectation