Algorithms for Decision Support

Online Algorithms (1/3)

Buy-or-Rent

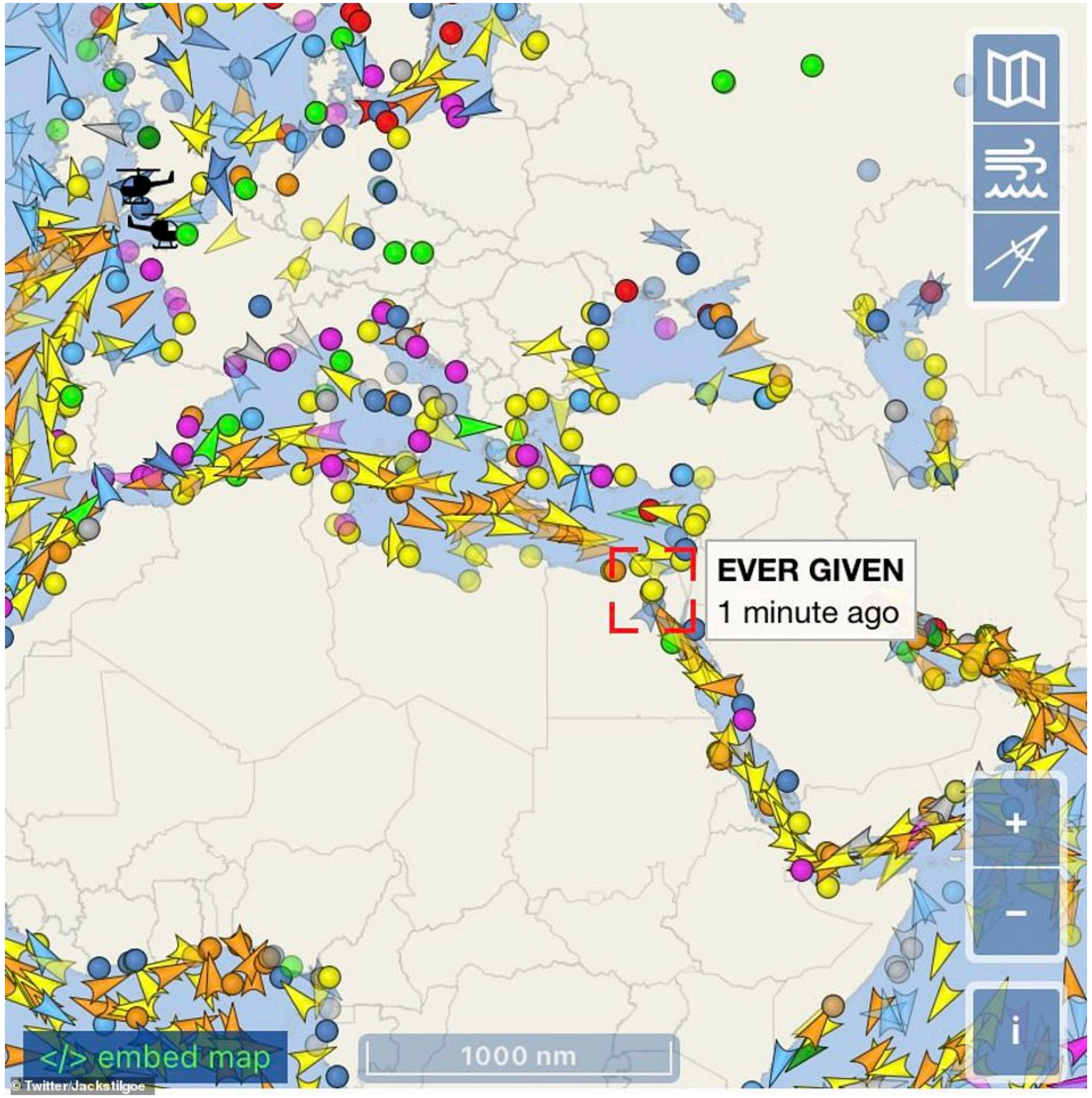
Outline

- **Online problems & online algorithms** — optimization with uncertainty
  - First example: **Ski-rental**
- Measure the performance: **Competitive ratio**
  - How good is an online algorithm?
- **Adversarial game**
  - How bad is an online algorithm?

Ever Given and Suez Canal

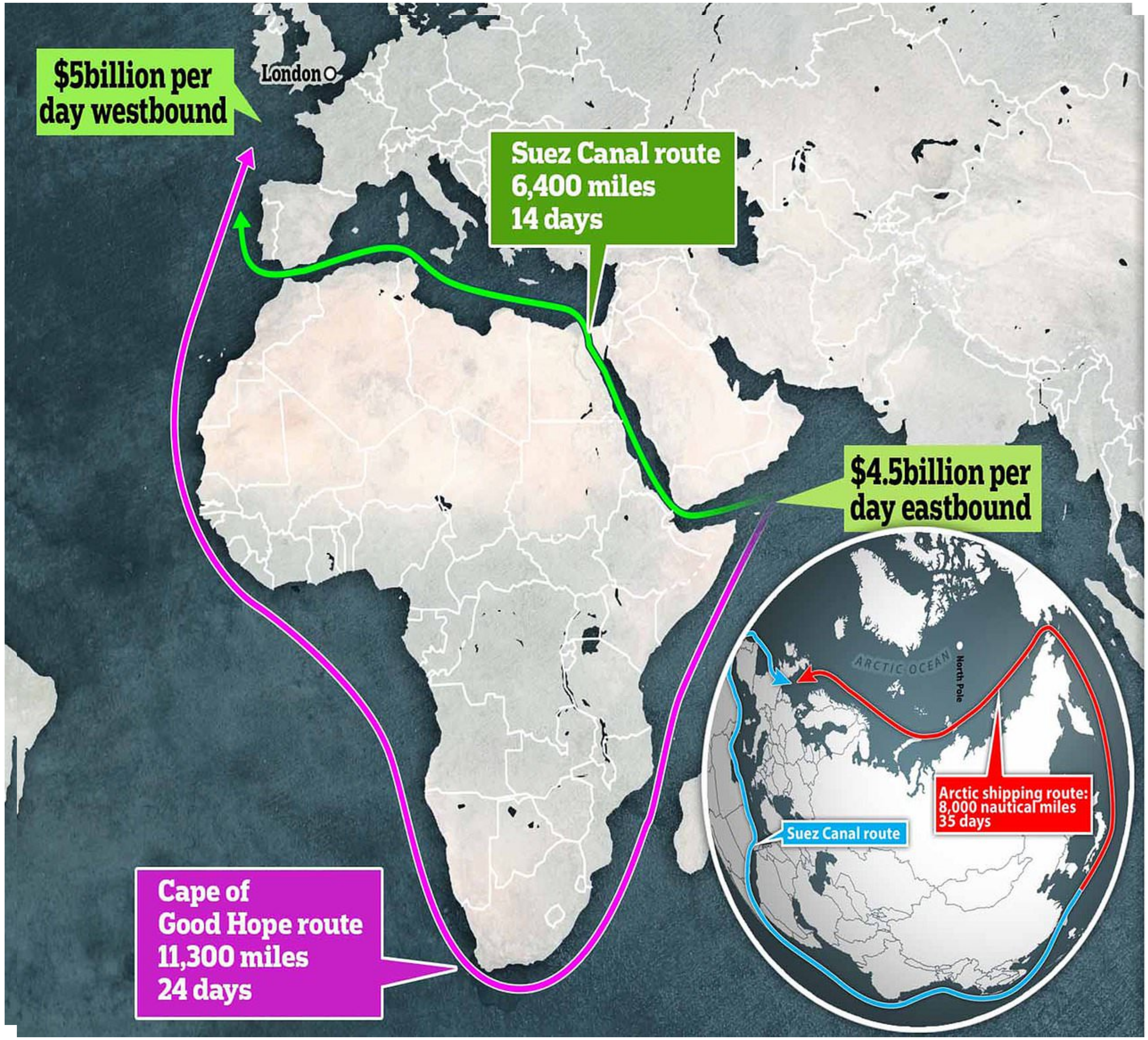
In March 2021, a container ship got stuck in the Suez Canal.

Ever Given and Suez Canal



Ever Given and Suez Canal

- To arrive the destination **as soon as possible**, what will you do?
  - Keep **waiting** until the Ever Given container ship is free, or
    - You may wait for a very long time
  - Turn around and **take the alternate route**, which cost extra 10 days
    - You may start turning around and find the Ever Given is free, and if you turn back again, it will cost you a even longer time



Outline

- **Online problems & online algorithms** — optimization with uncertainty
    - First example: **Ski-rental**
- Measure the performance: **Competitive ratio**
    - How good is an online algorithm?
- **Adversarial game**
    - How bad is an online algorithm?

Online Optimization

- Optimization under *uncertainty*
  - A strategy/algorithm has to respond to each event *without knowledge of future input*
  - A strategy/algorithm *cannot revoke any decision* that is already made
- A good strategy/algorithm should guarantee that even for the worst case, it performs not to bad compared to the optimal solution with hindsight

Online Problems and Online Algorithms

- Offline (optimization) problem:

- Online (optimization) problem:

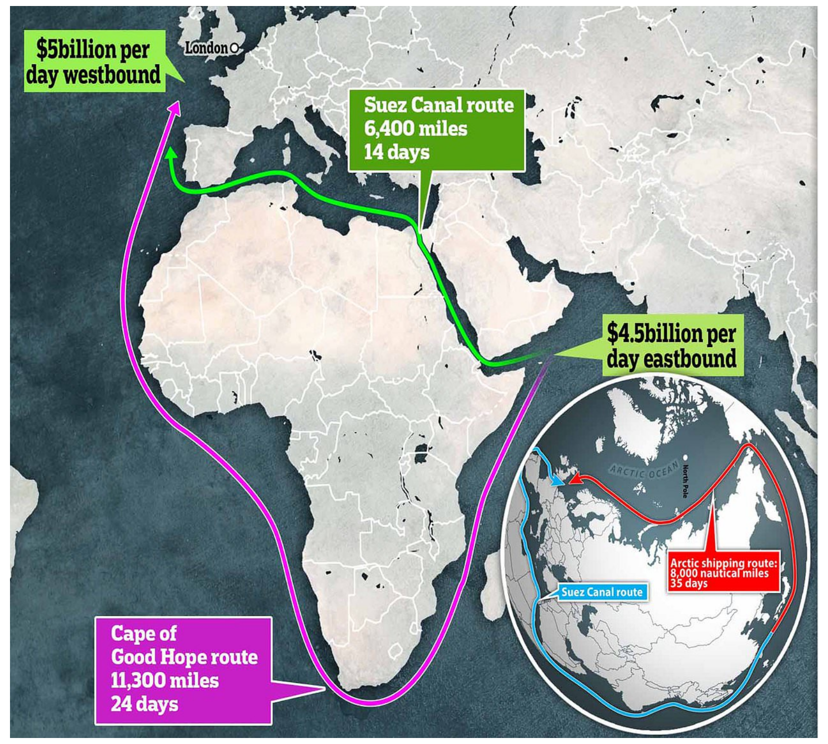
Online Problems and Online Algorithms

- Offline (optimization) problem:
  - The whole *instance* is given from the beginning
- Online (optimization) problem:

Online Problems and Online Algorithms

- Offline (optimization) problem:
  - The whole *instance* is given from the beginning

- Online (optimization) problem:

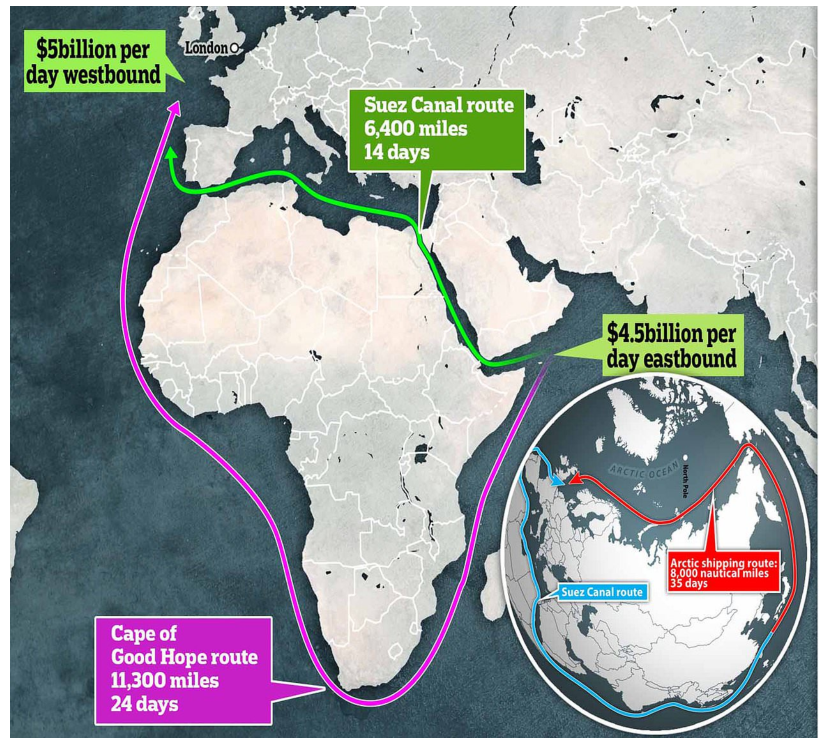


Online Problems and Online Algorithms

Instance: the status of the canal at any time (When will it be free)

- Offline (optimization) problem:
  - The whole *instance* is given from the beginning

- Online (optimization) problem:

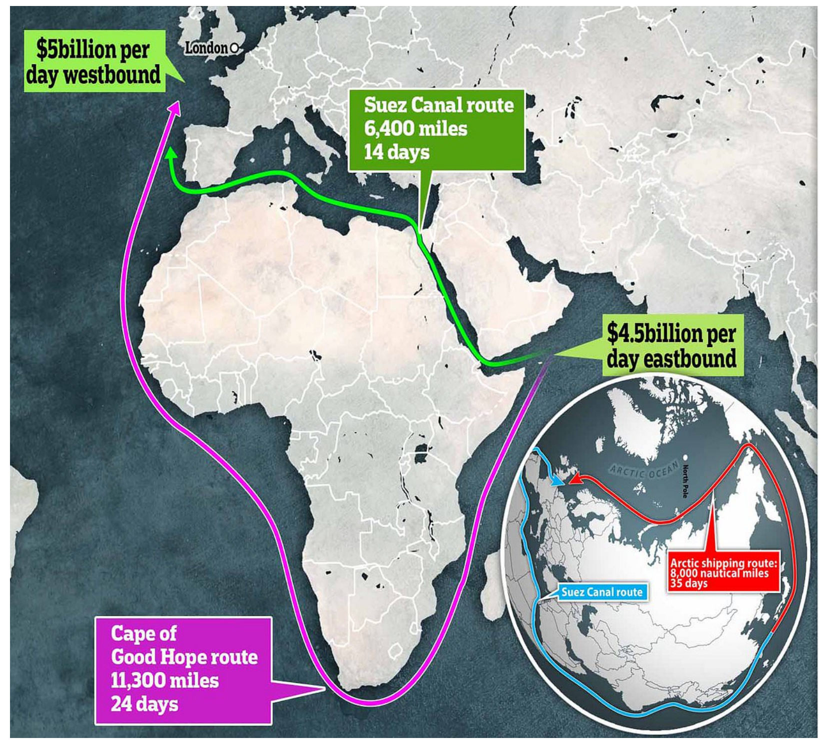


Online Problems and Online Algorithms

- Offline (optimization) problem:
    - The whole *instance* is given from the beginning

Instance: the status of the canal at any time
(When will it be free)

- Online (optimization) problem:
    - The instance is revealed piece-by-piece



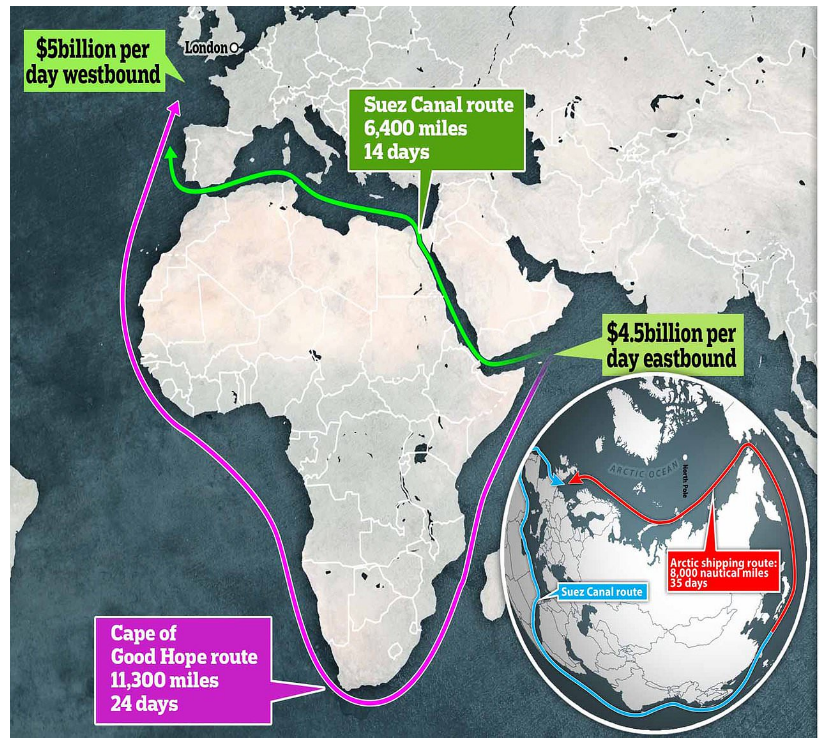
Online Problems and Online Algorithms

Instance: the status of the canal at any time
(When will it be free)

- Offline (optimization) problem:
  - The whole *instance* is given from the beginning

We only know the canal is CURRENTLY free or not

- Online (optimization) problem:
  - The instance is revealed piece-by-piece



Online Problems and Online Algorithms

- Offline (optimization) problem:
  - The whole *instance* is given from the beginning
  - The offline algorithm can make decisions according to the complete data
- Online (optimization) problem:
  - The instance is revealed piece-by-piece

Instance: the status of the canal at any time
(When will it be free)

We only know the canal is CURRENTLY free or not

Online Problems and Online Algorithms

- Offline (optimization) problem:
  - The whole *instance* is given from the beginning
  - The offline algorithm can make decisions according to the complete data
- Online (optimization) problem:
  - The instance is revealed piece-by-piece

Instance: the status of the canal at any time
(When will it be free)

We only know the canal is CURRENTLY free or not

Online Problems and Online Algorithms

- Offline (optimization) problem:
  - The whole *instance* is given from the beginning
  - The offline algorithm can make decisions according to the complete data
- Online (optimization) problem:
  - The instance is revealed piece-by-piece
  - The online algorithm has to respond and make an irrevocable decision once a piece of instance is presented, without knowing the future input

Instance: the status of the canal at any time
(When will it be free)

We only know the canal is CURRENTLY free or not

What are the instances of the project?

What are the instances of the project?

- Set of images and set of unavailable time intervals
  - $n, s_1, s_2, s_3, \cdots, s_n$
  - $m, t_1, \ell_1, t_2, \ell_2, \cdots, t_m, \ell_m$

What are the instances of the project?

- Set of images and set of unavailable time intervals
  - $n, s_1, s_2, s_3, \cdots, s_n$ $\leftarrow$ Known to the online algorithm
  - $m, t_1, \ell_1, t_2, \ell_2, \cdots, t_m, \ell_m$ $\leftarrow$ Unknown to the online algorithm

Online Algorithms

- Online algorithms:

Online Algorithms

- Online algorithms:
  - The decisions need to be made with *partial knowledge*

Online Algorithms

- Online algorithms:
  - The decisions need to be made with *partial knowledge*
  - The decisions are *irrevocable*

Online Algorithms

- Online algorithms:
  - The decisions need to be made with *partial knowledge*
  - The decisions are *irrevocable*
  - We aim for having an online algorithm that has a cost not too worse than the optimal cost, even for the worst case

Online Algorithms

- Online algorithms:
  - The decisions need to be made with *partial knowledge*
  - The decisions are *irrevocable*
  - We aim for having an online algorithm that has a cost not too worse than the optimal cost, even for the worst case
    - ***Cost***: the objective that we want to minimize or maximize

Online Algorithms

- Online algorithms:
  - The decisions need to be made with *partial knowledge*
  - The decisions are *irrevocable*
  - We aim for having an online algorithm that has a cost not too worse than the optimal cost, even for the worst case
    - ***Cost***: the objective that we want to minimize or maximize
  - Usually we don’t care about the time complexity

Outline

- **Online problems & online algorithms** — optimization with uncertainty
  - First example: **Ski-rental**
- Measure the performance: **Competitive ratio**
  - How good is an online algorithm?
- **Adversarial game**
  - How bad is an online algorithm?

Buy or Rent?

- Imagine that you are having a ski holiday

Buy or Rent?

- Imagine that you are having a ski holiday
  - The price of renting a ski is 1 per day

Buy or Rent?

- Imagine that you are having a ski holiday
  - The price of renting a ski is 1 per day
  - The buying price for ski is B.

Buy or Rent?

- Imagine that you are having a ski holiday
    - The price of renting a ski is 1 per day
    - The buying price for ski is B.
    - Cost: the total money you pay

Buy or Rent?

- Imagine that you are having a ski holiday
  - The price of renting a ski is 1 per day
  - The buying price for ski is B.
  - Cost: the total money you pay
- Suppose you want to spend money as little as possible. Should you buy the ski or rent it?

Buy or Rent?

- Imagine that you are having a ski holiday
  - The price of renting a ski is 1 per day
  - The buying price for ski is B.
  - Cost: the total money you pay
- Suppose you want to spend money as little as possible. Should you buy the ski or rent it?
  - It depends on the number of skiing days during your holiday  

Buy or Rent?

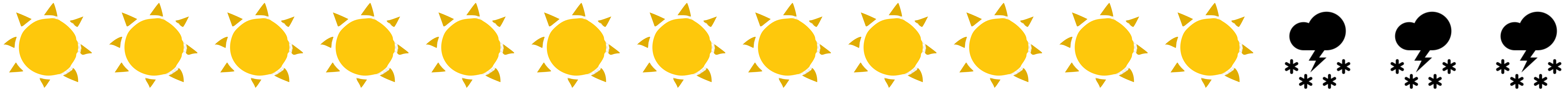
- Imagine that you are having a ski holiday
  - The price of renting a ski is 1 per day
  - The buying price for ski is B.
  - Cost: the total money you pay
- Suppose you want to spend money as little as possible. Should you buy the ski or rent it?
  - It depends on the number of skiing days during your holiday  
- Instance: B and the number of skiing days d

Buy or Rent? — Offline

- Rent: 1
- Buy: $B = 10$

An example

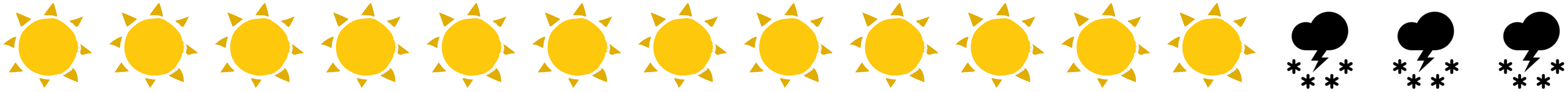
- Goal: minimize the total cost over the sky holiday



At least 10 skiing days

Buy or Rent? — Offline

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday



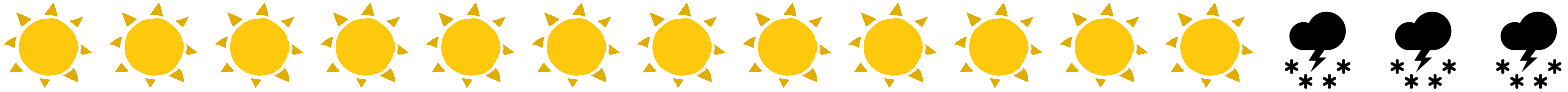
Buy

10

At least 10 skiing days

Buy or Rent? — Offline

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday



OPT: Buy

10 10

At least 10 skiing days

Buy or Rent? — Offline

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

Less than 10 skiing days

Buy or Rent? — Offline

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

Less than 10 skiing days

Buy or Rent? — Offline

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

OPT:	Rent	Rent	Rent
3	1	1	1

Less than 10 skiing days

Buy or Rent? — Offline

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

OPT: Rent Rent Rent

3 1 1 1

Optimal (offline) strategy: Buy the ski iff there are at least B skying days

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

ALG:

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

ALG: Buy

10

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

ALG:

Buy or Rent? — Online

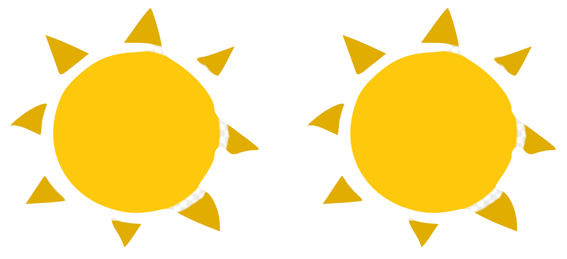
- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

ALG: Rent

1

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday



ALG: Rent

1

Buy or Rent? — Online

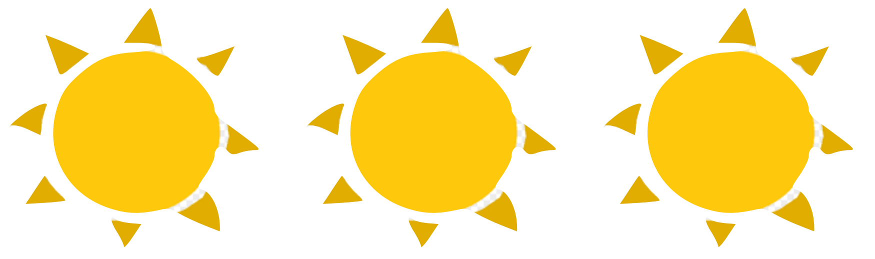
- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

ALG:	Rent	Rent
	1	1

Buy or Rent? — Online

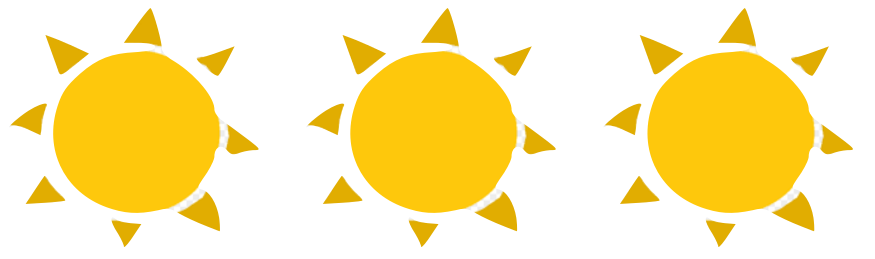
- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday



ALG:	Rent	Rent
	1	1

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

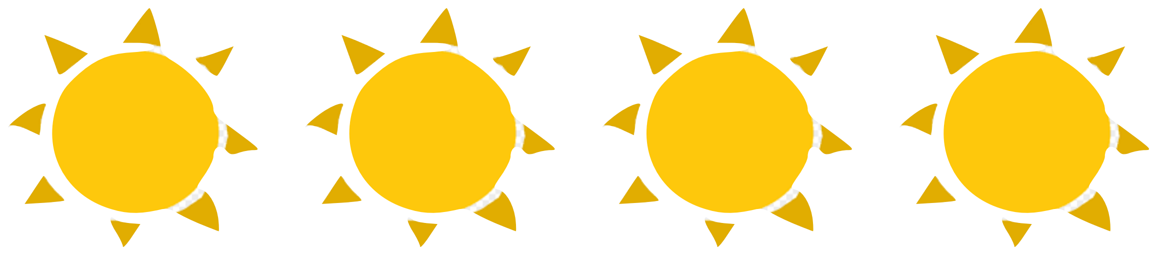


ALG: Rent Rent Rent

$1 \quad 1 \quad 1$

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday



ALG:	Rent	Rent	Rent	Buy
	1	1	1	10

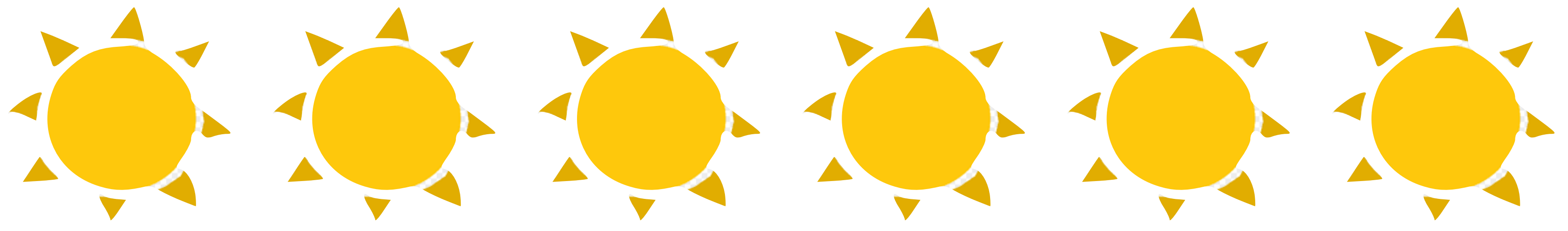
Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

ALG:	Rent	Rent	Rent	Buy
	1	1	1	10

Buy or Rent? — Online

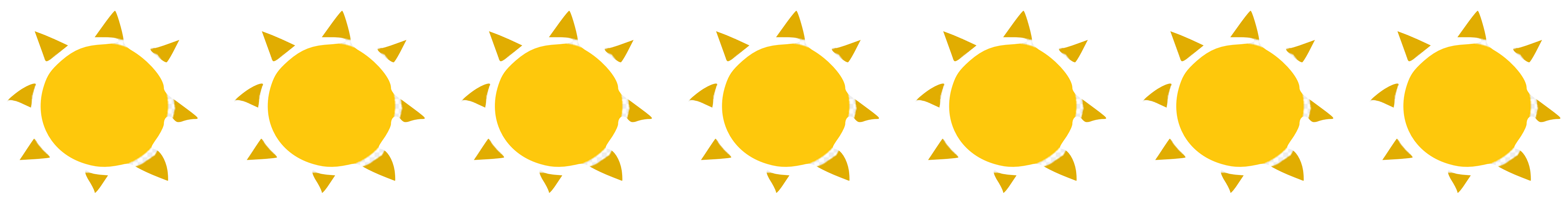
- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday



ALG:	Rent	Rent	Rent	Buy
	1	1	1	10

Buy or Rent? — Online

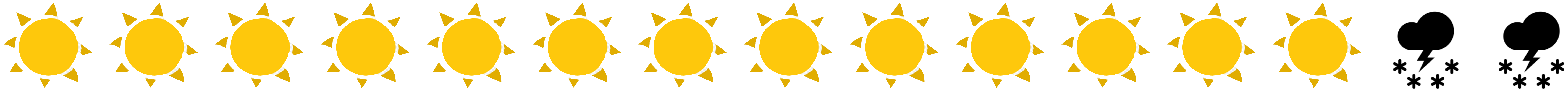
- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday



ALG:	Rent	Rent	Rent	Buy
	1	1	1	10

Buy or Rent? — Online

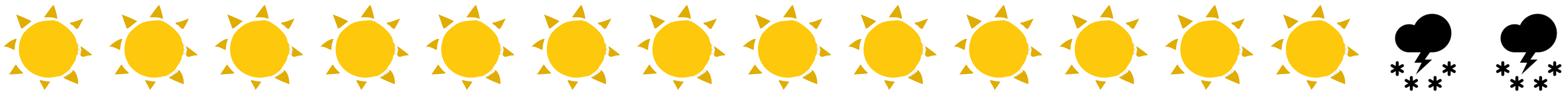
- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday



ALG:	Rent	Rent	Rent	Buy
	1	1	1	10

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

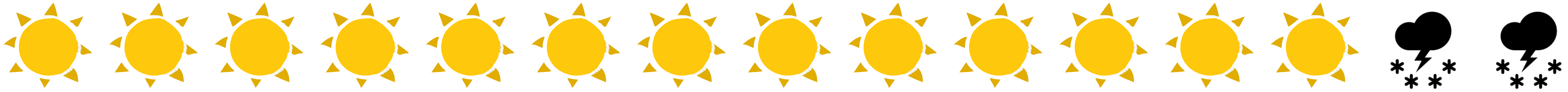


ALG:	Rent	Rent	Rent	Buy
13	1	1	1	10

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

Optimal (offline) strategy: Buy the ski iff there are at least B skying days



ALG:	Rent	Rent	Rent	Buy
13	1	1	1	10
OPT:	Buy			
10	10			

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

ALG:	Rent	Rent	Rent	Buy
13	1	1	1	10

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

ALG:	Rent	Rent	Rent	Buy
13	1	1	1	10
OPT:	Rent	Rent	Rent	Rent
	1	1	1	1

Buy or Rent? — Online

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

Optimal (offline) strategy: Buy the ski iff there are at least B skying days

ALG:	Rent	Rent	Rent	Buy
13	1	1	1	10
OPT:	Rent	Rent	Rent	Rent
4	1	1	1	1

What happened

- In an online problem, not every piece of information is known
- An online algorithm has to make decisions based on partial information
  - The decisions are not revokable
- Since the future input is unknown, the whole instance is uncertain
  - The currently good solution can become bad when more information is revealed

What will you do?

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

On *which day of skiing* will you buy the ski?

Outline

- **Online problems & online algorithms** — optimization with uncertainty
  - First example: **Ski-rental**
- Measure the performance: **Competitive ratio**
  - How good is an online algorithm?
- **Adversarial game**
  - How bad is an online algorithm?

Quality of Online Algorithms

- In the context of online computation, we ask what we can achieve when we do not know the whole input in advance

Quality of Online Algorithms

- In the context of online computation, we ask what we can achieve when we do not know the whole input in advance
  - To assess the quality of an online algorithm, we compare the cost of a solution computed by an algorithm to the cost of an optimal solution

Quality of Online Algorithms

- In the context of online computation, we ask what we can achieve when we do not know the whole input in advance
  - To assess the quality of an online algorithm, we compare the cost of a solution computed by an algorithm to the cost of an optimal solution
- An online algorithm ALG is c-**competitive** if for all instance I,

$$\text{ALG}(I) \leq c \cdot \text{OPT}(I)$$

Quality of Online Algorithms

- In the context of online computation, we ask what we can achieve when we do not know the whole input in advance
  - To assess the quality of an online algorithm, we compare the cost of a solution computed by an algorithm to the cost of an optimal solution
- An online algorithm ALG is c-**competitive** if for all instance I,

$$\text{ALG}(I) \leq c \cdot \text{OPT}(I)$$

  - ALG attains a **competitive ratio** c

Quality of Online Algorithms

- In the context of online computation, we ask what we can achieve when we do not know the whole input in advance
  - To assess the quality of an online algorithm, we compare the cost of a solution computed by an algorithm to the cost of an optimal solution
- An online algorithm ALG is c**-competitive** if for all instance I,

$$\text{ALG}(I) \leq c \cdot \text{OPT}(I)$$

  - ALG attains a **competitive ratio** c
  - The closer c is to 1, the better ALG is (note that c is always at least 1)

Competitive Ratio (Minimization)

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\text{ALG}(I) \leq c \cdot \text{OPT}(I)$$

Competitive Ratio (Minimization)

- An online algorithm ALG is c**-competitive** if for all instance I,

$$\text{ALG}(I) \leq c \cdot \text{OPT}(I)$$

  - That is, for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

Competitive Ratio (Minimization)

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\text{ALG}(I) \leq c \cdot \text{OPT}(I)$$

  - That is, for all instance I,

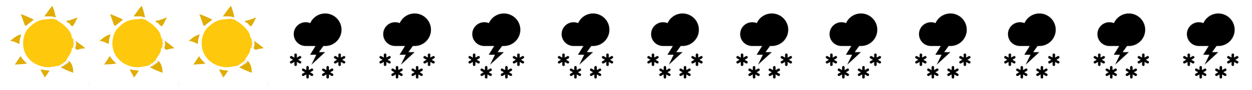
$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

Competitive Ratio (Minimization)

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\text{ALG}(I) \leq c \cdot \text{OPT}(I)$$

- That is, for all instance I,

4, 

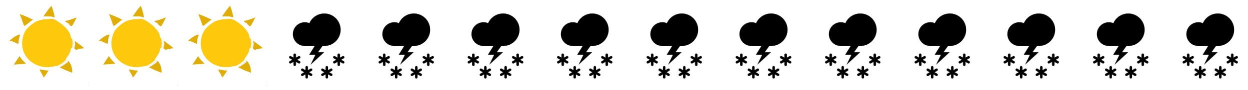
$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

Competitive Ratio (Minimization)

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\mathsf{ALG}(I) \leq c \cdot \mathsf{OPT}(I)$$

- That is, for all instance I,

4, 

4, 

$$\frac{\mathsf{ALG}(I)}{\mathsf{OPT}(I)} \leq c$$

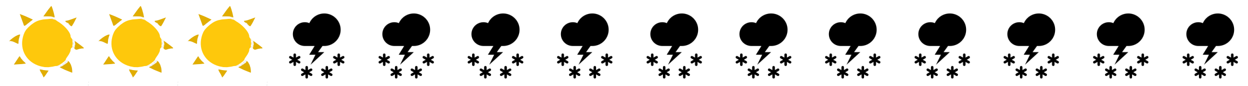
Competitive Ratio (Minimization)

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\text{ALG}(I) \le c \cdot \text{OPT}(I)$$

- That is, for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \le c$$

4, 

4, 

100, 

Competitive Ratio (Minimization)

- An online algorithm ALG is c-**competitive** if for all instance I,

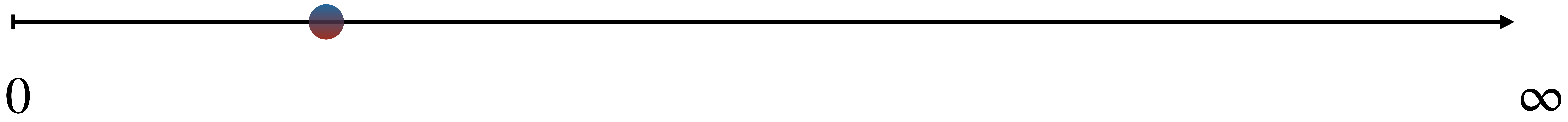
$$\mathrm{ALG}(I) \le c \cdot \mathrm{OPT}(I)$$

  - That is, for all instance I,

$$\frac{\mathrm{ALG}(I)}{\mathrm{OPT}(I)} \le c$$

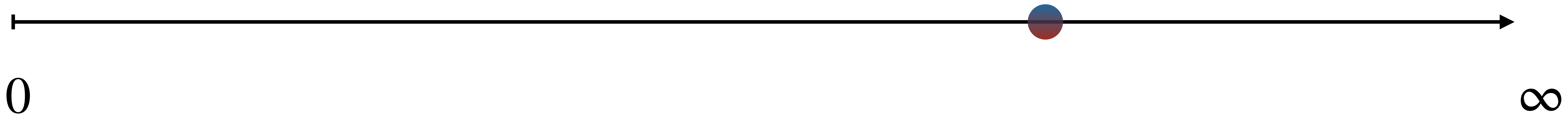
Competitive Ratio (Minimization)

$$\frac{\text{ALG}(B, ☀☀☀⛈⛈⛈⛈⛈⛈⛈⛈⛈⛈⛈)}{\text{OPT}(B, ☀☀☀⛈⛈⛈⛈⛈⛈⛈⛈⛈⛈⛈)}$$



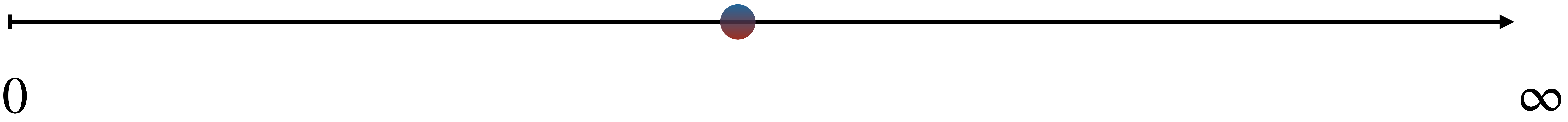
Competitive Ratio (Minimization)

$$\frac{\text{ALG}(B, ☀☀☀☀☀⛈⛈⛈⛈⛈⛈⛈⛈)}{\text{OPT}(B, ☀☀☀☀☀⛈⛈⛈⛈⛈⛈⛈⛈)}$$

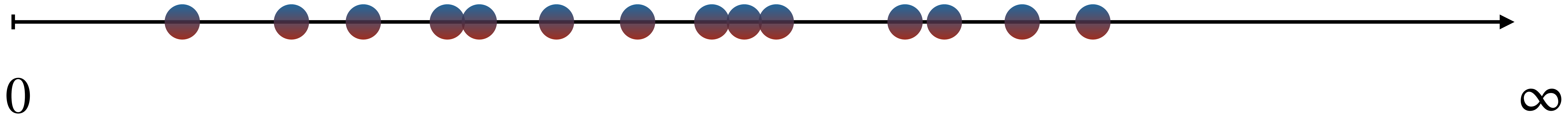


Competitive Ratio (Minimization)

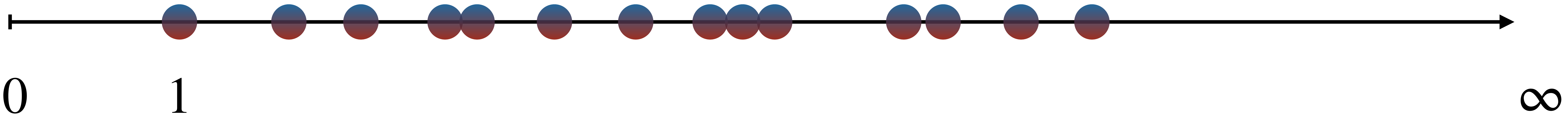
$$\frac{\text{ALG}(B, ☀🌨🌨🌨)}{\text{OPT}(B, ☀🌨🌨🌨)}$$



Competitive Ratio (Minimization)



Competitive Ratio (Minimization)

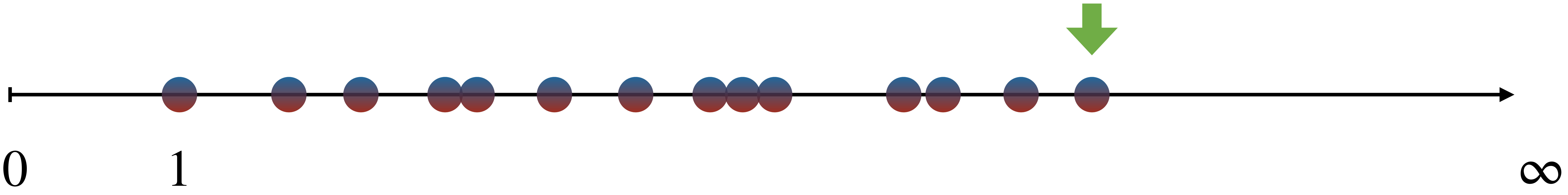


For minimization problems,

$ALG(I) \geq OPT(I)$

Competitive Ratio (Minimization)

For all instance I, $\dfrac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$



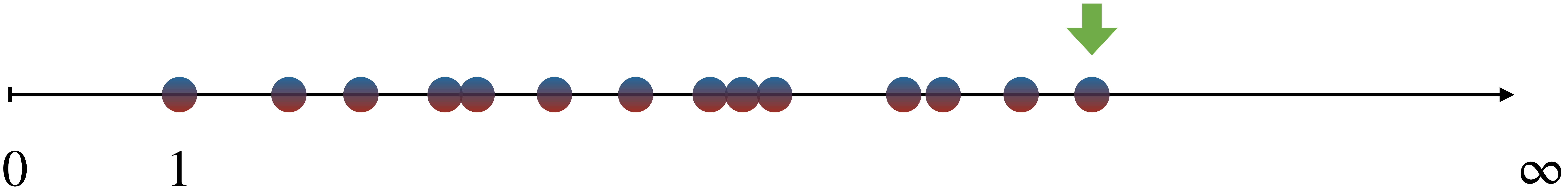
For minimization problems,

$ALG(I) \geq OPT(I)$

Competitive Ratio (Minimization)

For all instance I, $\dfrac{\text{ALG}(I)}{\text{OPT}(I)} \le c$

All the “balls” are at the left or at c



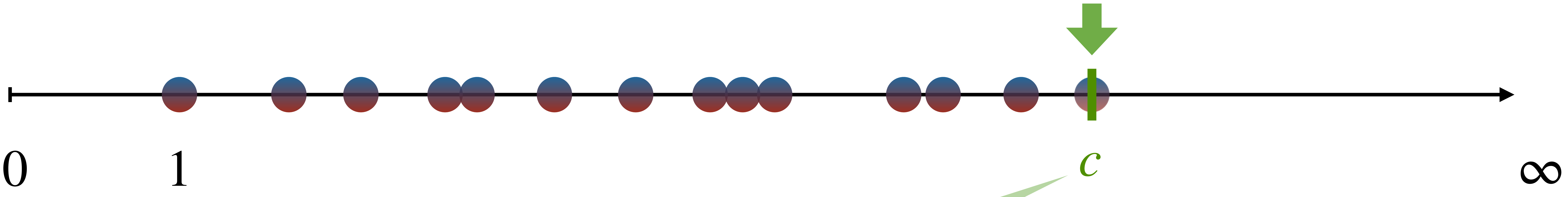
For minimization problems,

$ALG(I) \ge OPT(I)$

Competitive Ratio (Minimization)

For all instance I, $\dfrac{\text{ALG}(I)}{\text{OPT}(I)} \le c$

All the “balls” are at the left or at c



Our goal is
to find this c

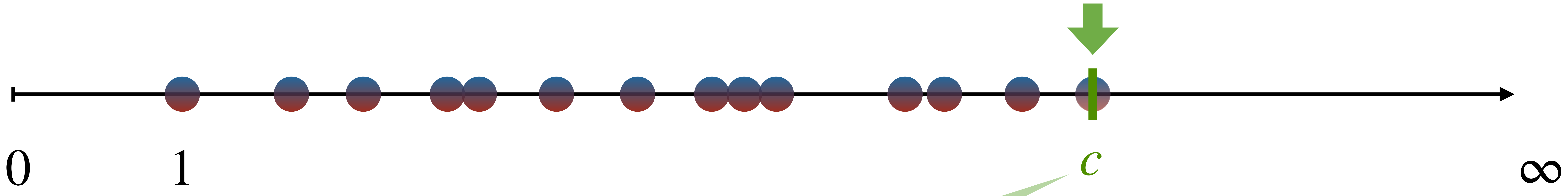
For minimization problems,

$ALG(I) \geq OPT(I)$

Competitive Ratio (Minimization)

For all instance I, $\dfrac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$

All the "balls" are at the left or at c



For minimization problems,

$ALG(I) \geq OPT(I)$

Our goal is
to find this c

Even in the worst case,
we will not pay more than
c times the optimal solution.

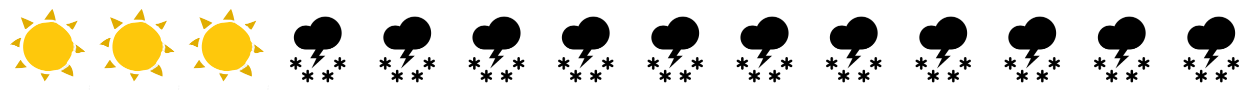
Competitive Ratio (Minimization)

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\text{ALG}(I) \leq c \cdot \text{OPT}(I)$$

  - That is, for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

4, 

4, 

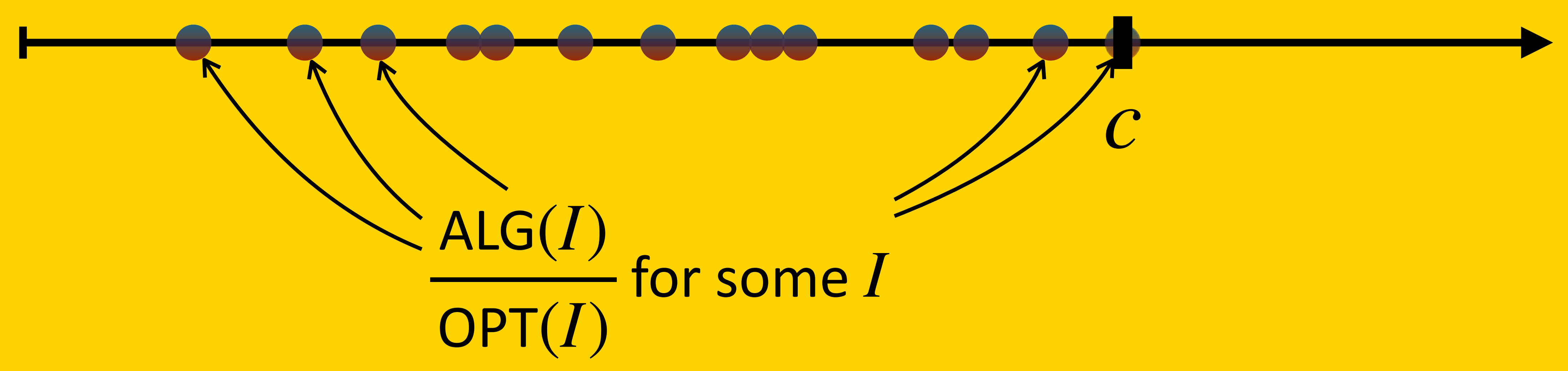
100, 

- The competitive ratio of an online algorithm is its performance guarantee: Even for the worst case, it is not too much worse than the optimal solution

What Happened?

- The performance measurement of an online algorithm is by **competitive analysis**
  - Against an optimal offline algorithm (which knows the future and has unlimited computation power)

- The ultimate goal of competitive analysis on an online algorithm ALG is to show that there is some c such that for any instance, the cost of ALG is no more than c times the optimal cost
  - Algorithms with a smaller c are better



What is the goal for the project

- If you want to have some theoretical results for the online setting, you have to
  - Design an online algorithm
  - Prove that for any set of images and any set of unavailable time intervals, the cost (transmission completed time) of the online algorithm is at most c times of the optimal cost for some c

- **It is okay to target on some special cases!**

Outline

- **Online problems & online algorithms** — optimization with uncertainty
  - First example: **Ski-rental**
- Measure the performance: **Competitive ratio**
  - How good is an online algorithm?
- **Adversarial game**
  - How bad is an online algorithm?

What will you do?

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

On ***which day*** *of skiing* will you buy the ski?

What will you do?

- Rent: 1
- Buy: 10
- Goal: minimize the total cost over the sky holiday

On ***which day*** *of skiing* will you buy the ski?

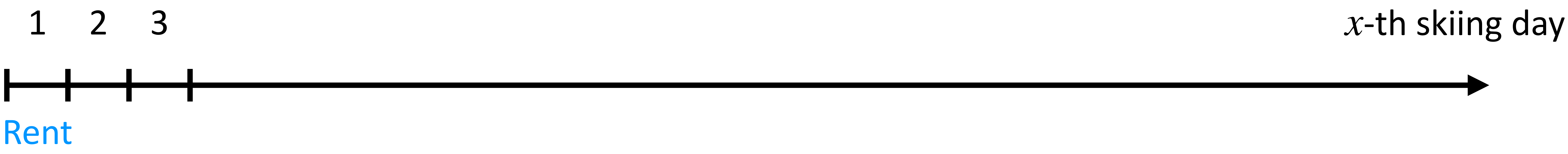
Keep renting the ski until the B-th skiing day

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day

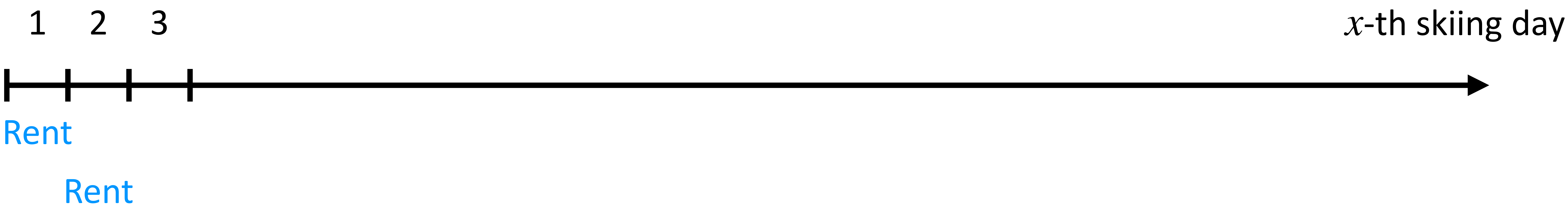
A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



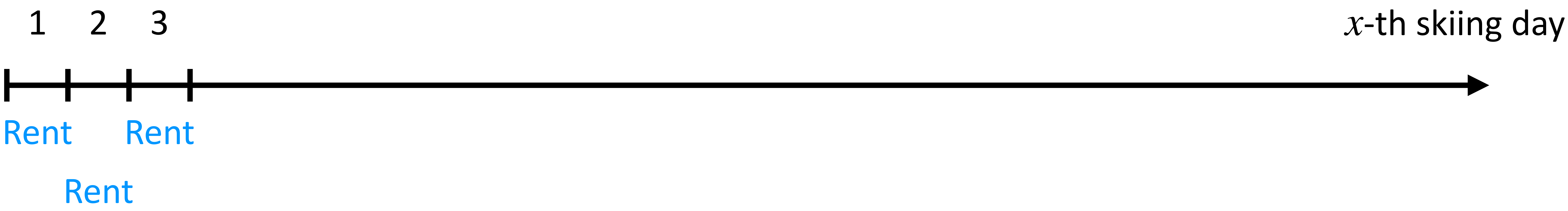
A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



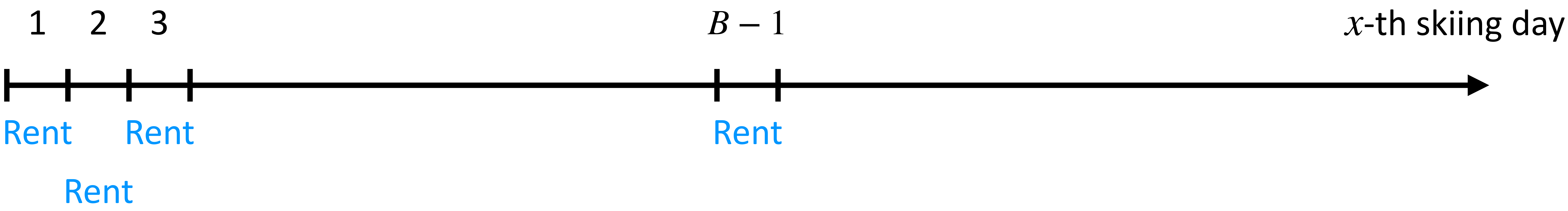
A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



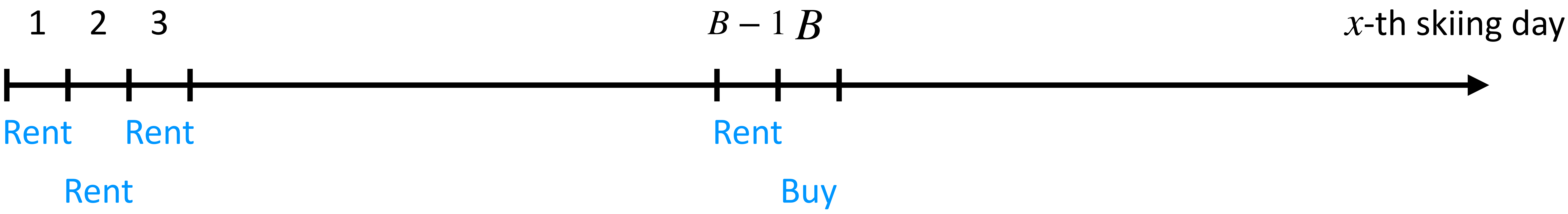
A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day

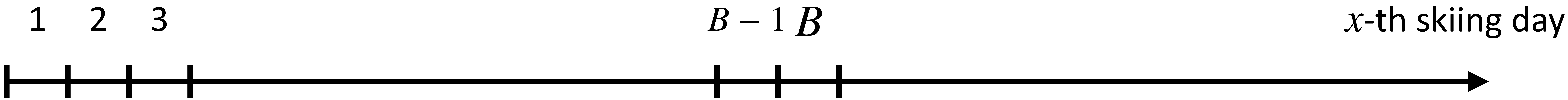
- Theorem: For the Buy-or-Rent problem, this algorithm is $(2 - \frac{1}{B})$-competitive.

<Proof idea>

We prove this theorem by showing that no matter how many skiing days there are, the algorithm cost is no more than twice the optimal cost.

A 2-Competitive Online Algorithm

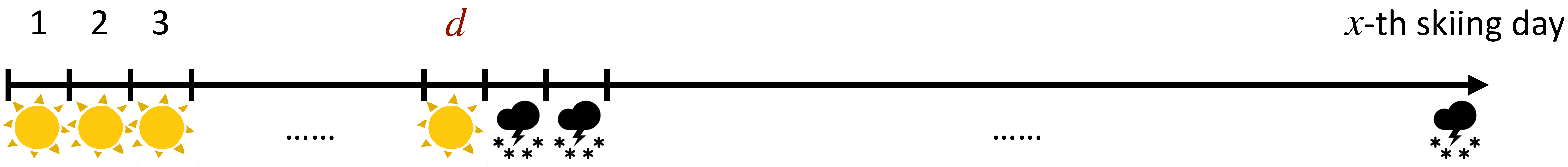
Keep renting the ski until the B-th skiing day



d: the number of actual total skiing days

A 2-Competitive Online Algorithm

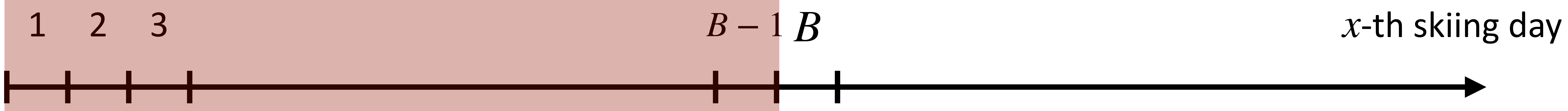
Keep renting the ski until the B-th skiing day



d: the number of actual total skiing days

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



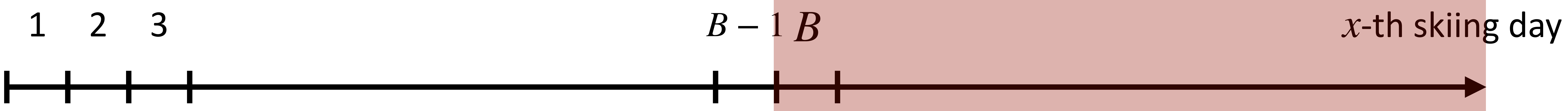
d: the number of actual total skiing days

If d is in this range, the optimal strategy is to buy the ski for each of the d days.

OPT = d

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



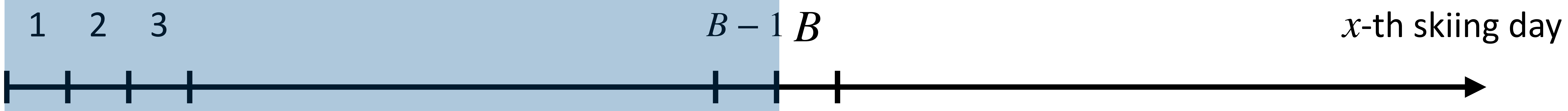
d: the number of actual total skiing days

If d is in this range, the optimal strategy is to buy the ski on the first day.

OPT = B

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



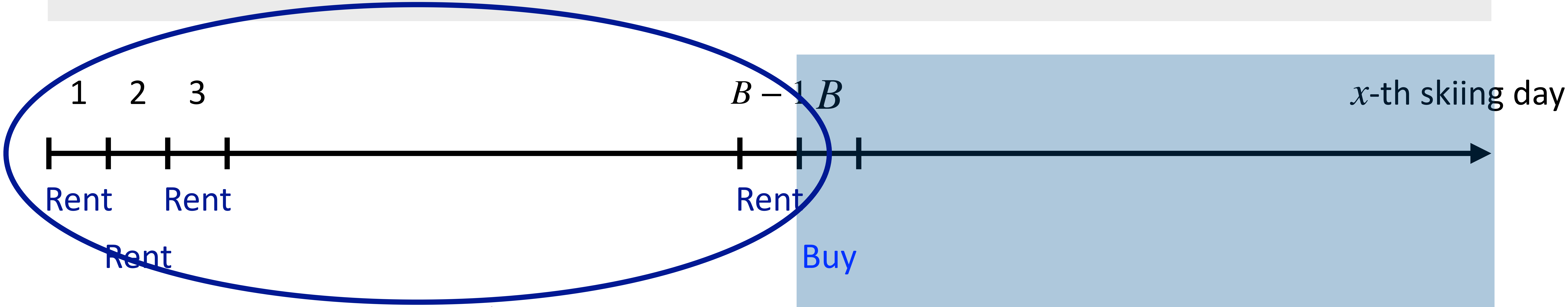
d: the number of actual total skiing days

If d is in this range, the algorithm rents for each of the d days.

ALG = d

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



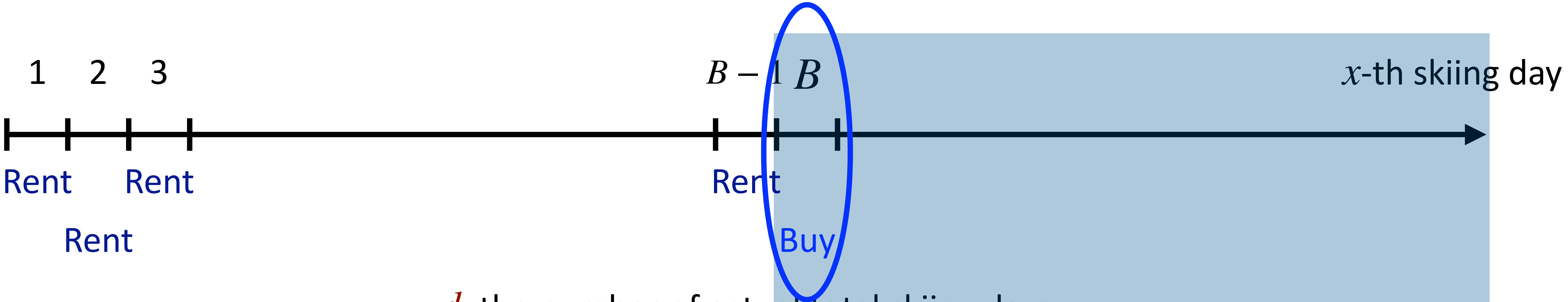
d: the number of actual total skiing days

If d is in this range, the algorithm buys the ski on the B-th day.

ALG $= (B-1) \cdot 1 + B$

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



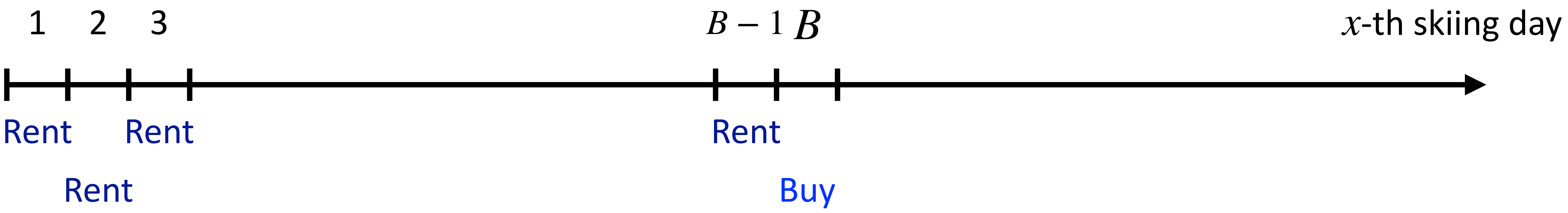
d: the number of actual total skiing days

If d is in this range, the algorithm buys the ski on the B-th day.

ALG = $(B-1)\cdot 1 + B$

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

If $d < B$

OPT = d

ALG = d

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} = \frac{d}{d}$$

If $d \geq B$

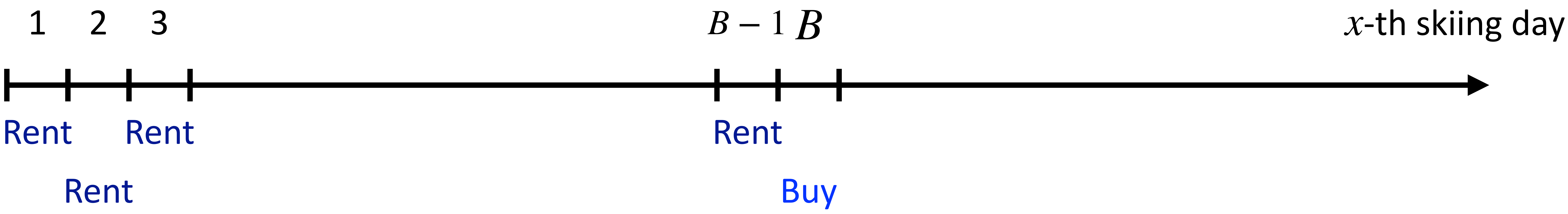
OPT = B

ALG = $(B-1)+B$

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} = \frac{2B-1}{B}$$

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \le c$$

If $d < B$

OPT = d

ALG = d

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} = \frac{d}{d}$$

If $d \ge B$

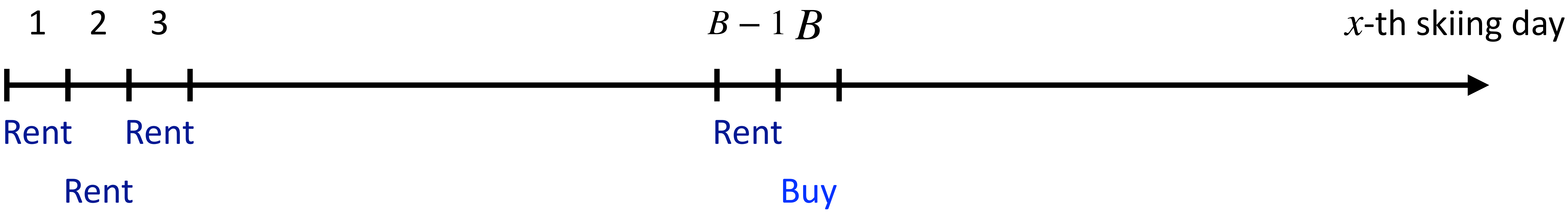
OPT = B

ALG = $(B-1)+B$

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} = \frac{2B-1}{B}$$

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \le c$$

If $d < B$

OPT = d

ALG = d

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} = \frac{d}{d}$$

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} = \max\left\{\frac{d}{d}, \frac{2B-1}{B}\right\}$$

If $d \geq B$

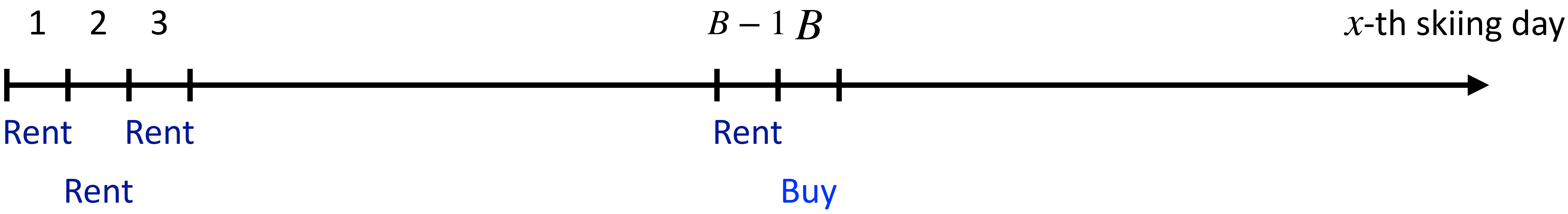
OPT = B

ALG = $(B-1)+B$

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} = \frac{2B-1}{B}$$

A 2-Competitive Online Algorithm

Keep renting the ski until the B-th skiing day



An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

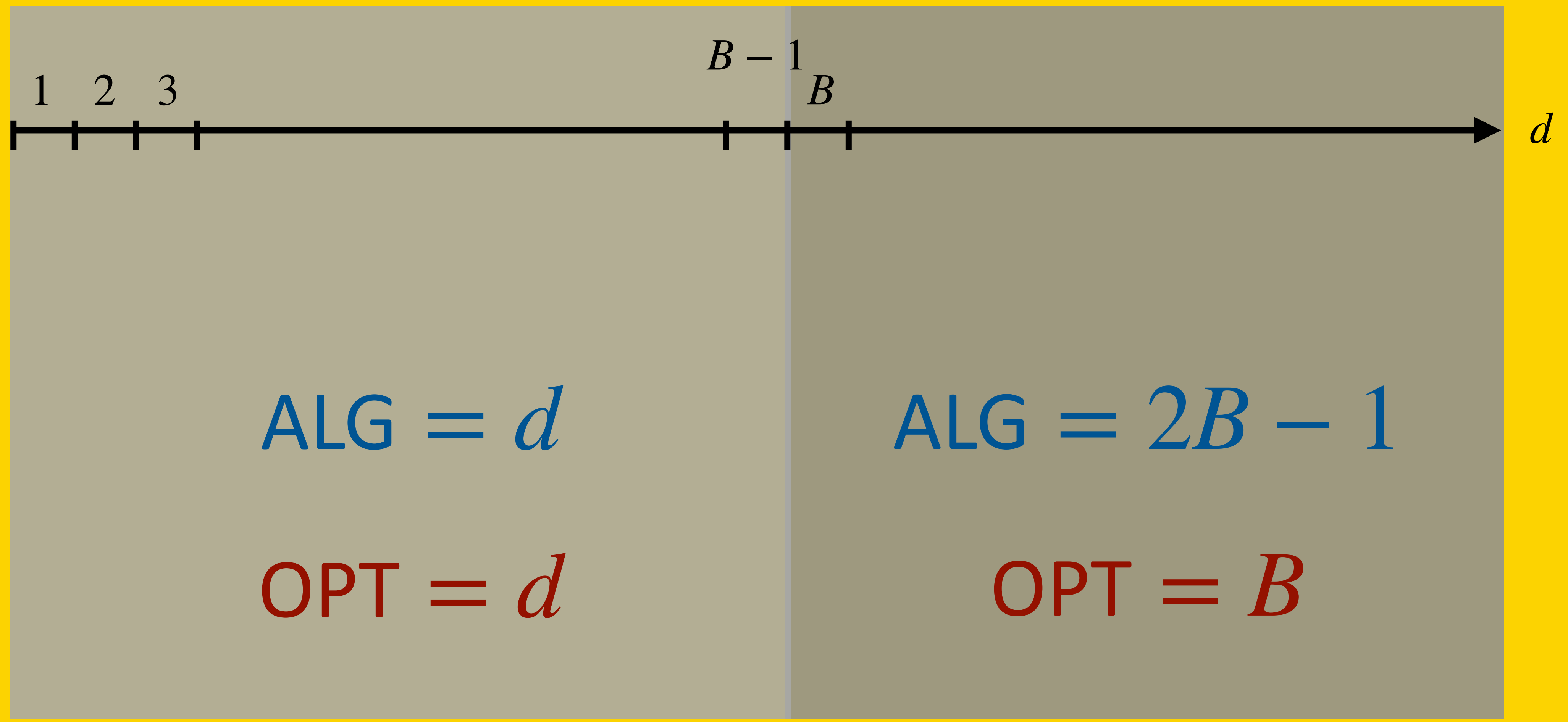
$$\frac{\text{ALG}(I)}{\text{OPT}(I)} = \max\left\{\frac{d}{d}, \frac{2B-1}{B}\right\}$$

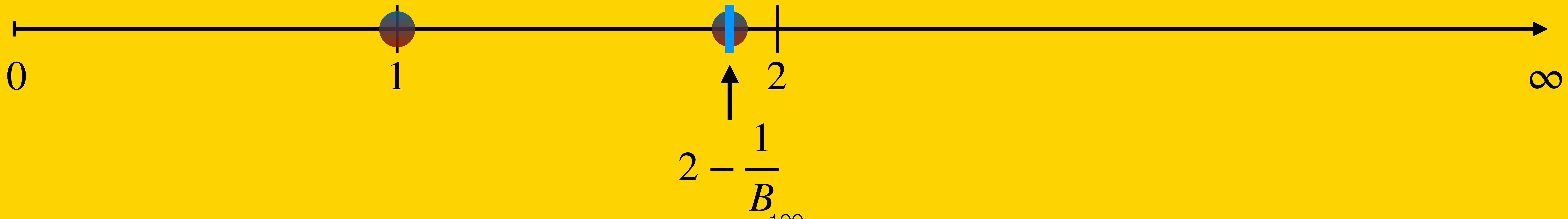
$$= \frac{2B-1}{B} = 2 - \frac{1}{B}$$

Competitive Ratio Proof Review

- We partition the set of instances into to cases: $d \leq B - 1$ or $d \geq B$.
  - For the case $d \leq B - 1$, $\dfrac{\text{ALG}(B, d)}{\text{OPT}(B, d)} = 1$
  - For the case $d \geq B$, $\dfrac{\text{ALG}(B, d)}{\text{OPT}(B, d)} = 2 - \dfrac{1}{B}$

➡ The algorithm **ALG** is $(2 - \dfrac{1}{B})$-competitive





Outline

- **Online problems & online algorithms** — optimization with uncertainty
  - First example: **Ski-rental**
- Measure the performance: **Competitive ratio**
  - How good is an online algorithm?
- **Adversarial game**
  - How bad is an online algorithm?

Online Player - Adversary Game

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

Online algorithm

Online Player - Adversary Game

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \le c$$

Online algorithm

I want to guarantee a small c for any I!

Online Player - Adversary Game

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

Bad person

Online algorithm

I want to guarantee a small c for any I!

Online Player - Adversary Game

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\mathsf{ALG}(I)}{\mathsf{OPT}(I)} \leq c$$

Bad person

I want to make the algorithm fail
to guarantee a small c!

Online algorithm

I want to guarantee a small c
for any I!

Online Player - Adversary Game

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \le c$$

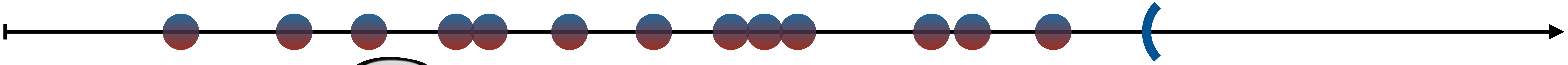
Bad person

Online algorithm

I want to find some instance I' such that $\frac{\text{ALG}(I')}{\text{OPT}(I')} > c$!

I want to guarantee that for any I, $\frac{\text{ALG}(I)}{\text{OPT}(I)} \le c$!

Online Player - Adversary Game



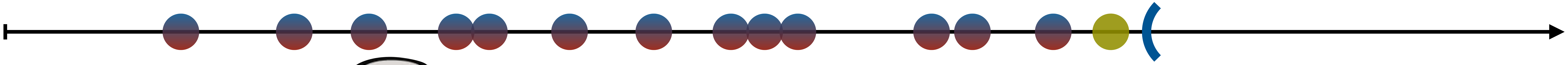
Bad person

Online algorithm

I want to find some instance I' such that $\dfrac{\text{ALG}(I')}{\text{OPT}(I')} > c$!

I want to guarantee that for any I, $\dfrac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$!

Online Player - Adversary Game



Bad person

Online algorithm

I want to find some instance I' such that $\dfrac{\text{ALG}(I')}{\text{OPT}(I')} > c$!

I want to guarantee that for any I, $\dfrac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$!

Online Player - Adversary Game

- One way to view the problem of analyzing online algorithms is to view it as a game between an **online player** (algorithm) and a malicious **adversary**

Online Player - Adversary Game

- One way to view the problem of analyzing online algorithms is to view it as a game between an **online player** (algorithm) and a malicious **adversary**

  - The online player runs its online algorithm on an input created by the adversary

Online Player - Adversary Game

- One way to view the problem of analyzing online algorithms is to view it as a game between an **online player** (algorithm) and a malicious **adversary**

  - The online player runs its online algorithm on an input created by the adversary

  - The adversary, based on the knowledge of the algorithm used by the online player, constructs the worst possible input so as to maximize the competitive ratio

Adversary

ALG_1: Buy the ski on the first day 

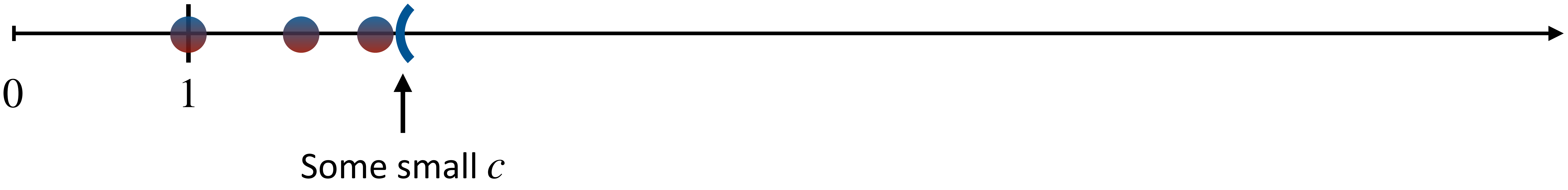
Adversary

ALG_1: Buy the ski on the first day

Adversary

ALG$_1$: Buy the ski on the first day

It starts raining since
the 4-th day!



Adversary

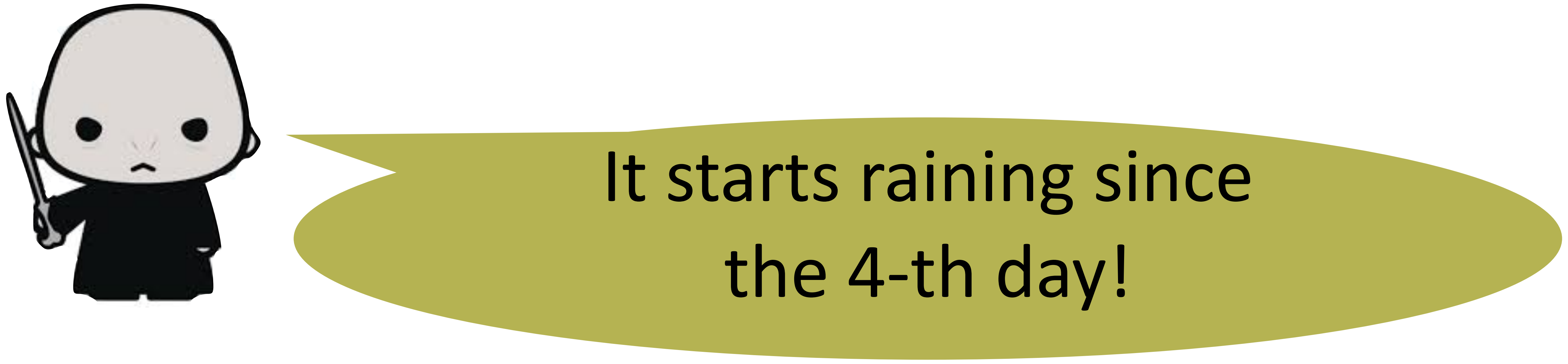
ALG$_1$: Buy the ski on the first day

It starts raining since the 4-th day!

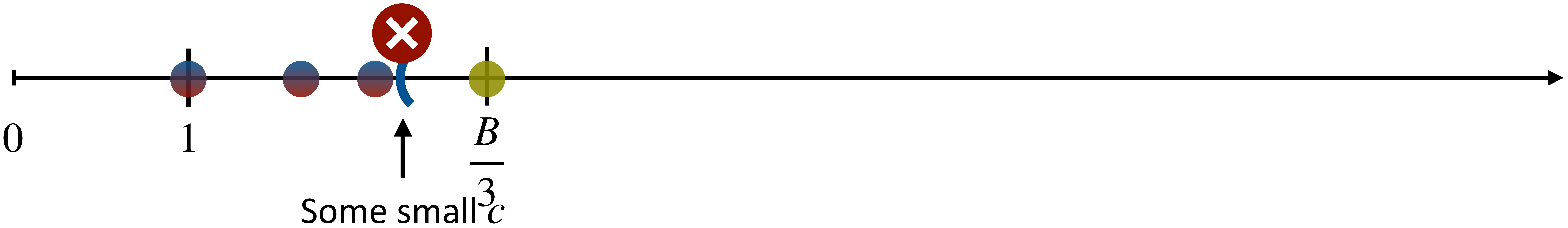
$$\frac{\text{ALG}_1(d=3)}{\text{OPT}(d=3)} = \frac{B}{3}$$

Adversary

ALG_1: Buy the ski on the first day



$$\frac{\text{ALG}_1(d=3)}{\text{OPT}(d=3)} = \frac{B}{3}$$



Adversary

ALG$_1$: Buy the ski on the first day

It starts raining since the 4-th day!

$$\frac{\text{ALG}_1(d=3)}{\text{OPT}(d=3)} = \frac{B}{3}$$

ALG$_1$ is **at least** $\frac{B}{2}$-competitive

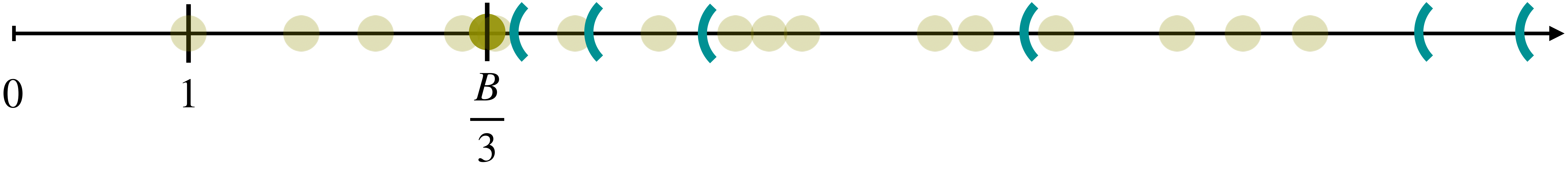
The competitive ratio (the point where all balls are at the left) can not be smaller than $\frac{B}{2}$

Adversary

ALG_1: Buy the ski on the first day

It starts raining since the 4-th day!

$$\frac{\mathrm{ALG}_1(d=3)}{\mathrm{OPT}(d=3)} = \frac{B}{3}$$



Adversary

ALG$_1$: Buy the ski on the first day

It starts raining since
the second day!

Adversary

ALG_1: Buy the ski on the first day

It starts raining since
the second day!

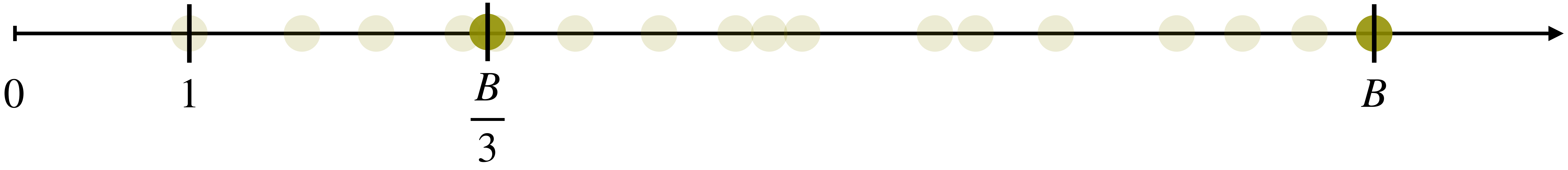
$$\frac{\text{ALG}_1(d = 1)}{\text{OPT}(d = 1)} = \frac{B}{1} = B$$

Adversary

ALG_1: Buy the ski on the first day

It starts raining since the second day!

$$\frac{\text{ALG}_1(d=1)}{\text{OPT}(d=1)} = \frac{B}{1} = B$$



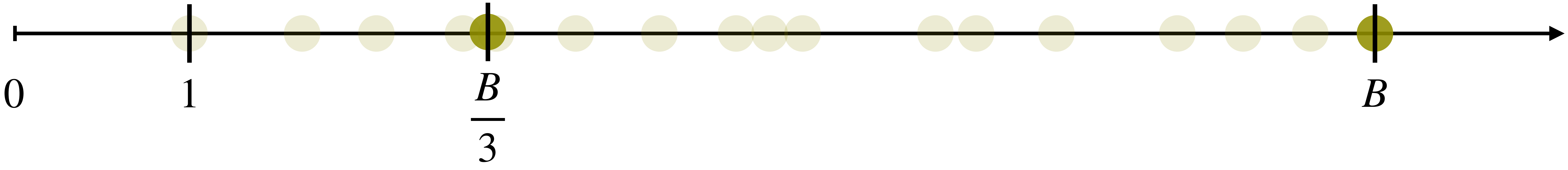
Adversary

ALG$_1$: Buy the ski on the first day

It starts raining since the second day!

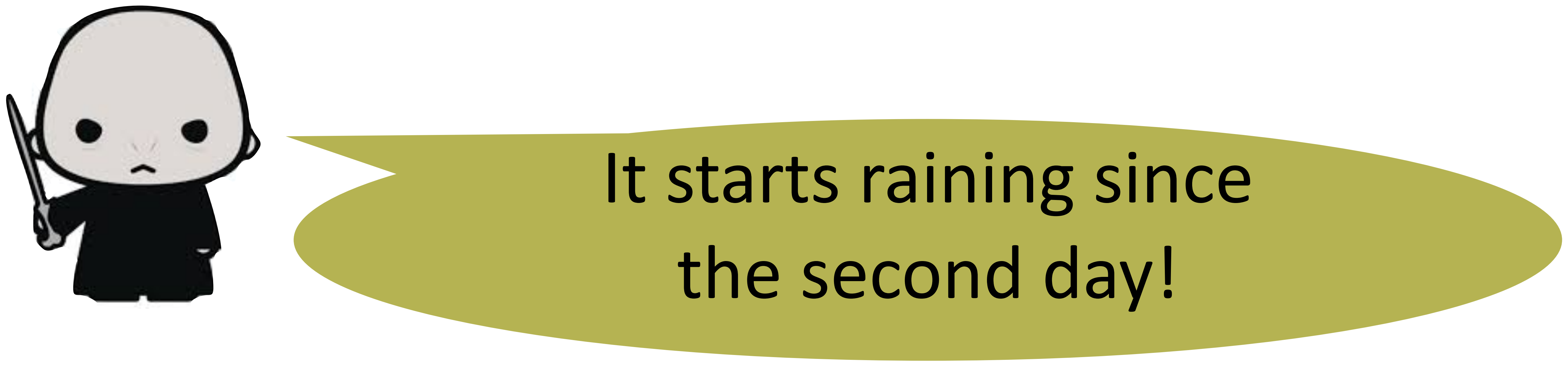
$$\frac{\text{ALG}_1(d=1)}{\text{OPT}(d=1)} = \frac{B}{1} = B$$

ALG$_1$ is **at least** B-competitive



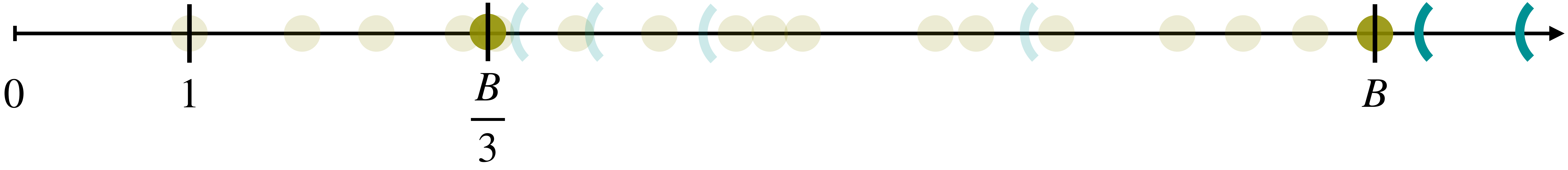
Adversary

ALG_1: Buy the ski on the first day



$$\frac{\text{ALG}_1(d=1)}{\text{OPT}(d=1)} = \frac{B}{1} = B$$

ALG_1 is **at least** B-competitive



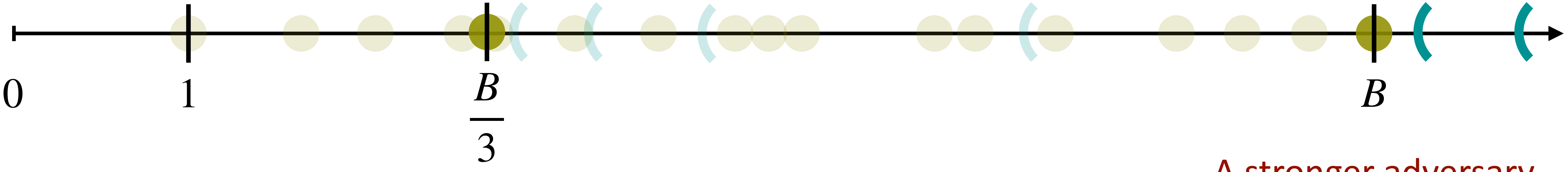
Adversary

ALG$_1$: Buy the ski on the first day

It starts raining since the second day!

$$\frac{\text{ALG}_1(d=1)}{\text{OPT}(d=1)} = \frac{B}{1} = B$$

ALG$_1$ is **at least** B-competitive



A stronger adversary

Adversary

- By designing an adversarial input for an algorithm ALG, one can find the *lower bound* of the competitive ratio of ALG

Adversary

- By designing an adversarial input for an algorithm ALG, one can find the *lower bound* of the competitive ratio of ALG

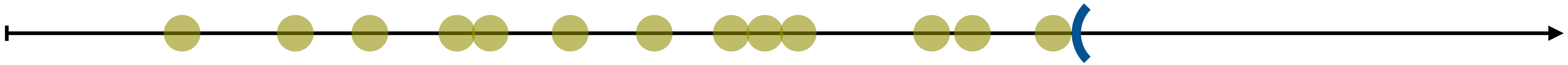
- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

Adversary

- By designing an adversarial input for an algorithm ALG, one can find the *lower bound* of the competitive ratio of ALG
  - There is an input I' such that $\mathrm{ALG}(I') \geq c' \cdot \mathrm{OPT}(I')$

  $\iff$ NOT $\left\{ \text{for all instance } I,\ \mathrm{ALG}(I) \leq c \cdot \mathrm{OPT}(I) \right\}$ for some $c < c'$



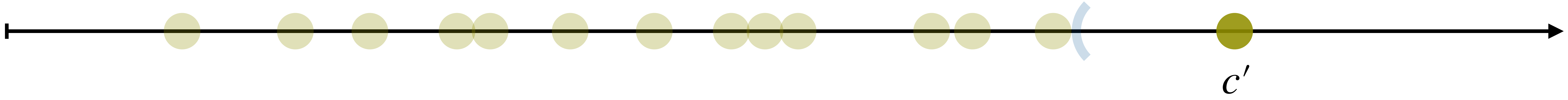
- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\mathrm{ALG}(I)}{\mathrm{OPT}(I)} \leq c$$

Adversary

- By designing an adversarial input for an algorithm ALG, one can find the *lower bound* of the competitive ratio of ALG
  - There is an input I' such that ALG$(I') \geq c' \cdot$ OPT(I')

$\iff$ NOT $\left\{ \text{for all instance } I,\ \text{ALG}(I) \leq c \cdot \text{OPT}(I) \right\}$ for some $c < c'$



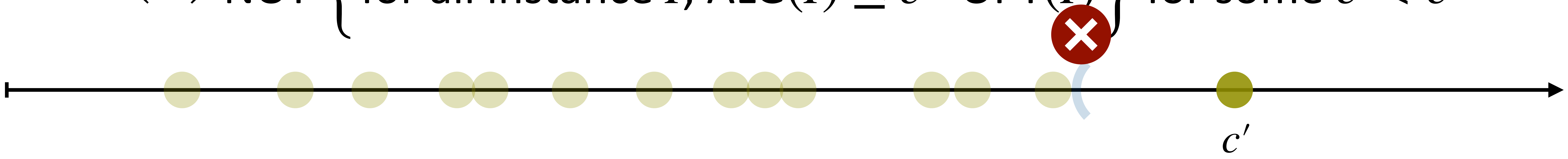
- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

Adversary

- By designing an adversarial input for an algorithm ALG, one can find the *lower bound* of the competitive ratio of ALG
  - There is an input I' such that ALG$(I') \geq c' \cdot$ OPT(I')

$\iff$ NOT $\left\{ \text{for all instance } I, \text{ALG}(I) \leq c \cdot \text{OPT}(I) \right\}$ for some $c < c'$



- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

Adversary

- By designing an adversarial input for an algorithm ALG, one can find the *lower bound* of the competitive ratio of ALG
  - There is an input I' such that ALG$(I') \geq c' \cdot$ OPT(I')

    $\iff$ NOT $\left\{ \text{for all instance } I, \text{ALG}(I) \leq c \cdot \text{OPT}(I) \right\}$ for some $c < c'$
  - A *stronger* adversary leads a bigger lower bound

- An online algorithm ALG is c-**competitive** if for all instance I,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$$

Adversary

- By designing an adversarial input for an algorithm ALG, one can find the *lower bound* of the competitive ratio of ALG
    - A *stronger* adversary leads a bigger lower bound

ALG$_B$: Buy the ski on the B-th skiing day

Adversary

- By designing an adversarial input for an algorithm ALG, one can find the *lower bound* of the competitive ratio of ALG
  - A *stronger* adversary leads a bigger lower bound

ALG$_B$: Buy the ski on the B-th skiing day

- Which is/are the *strongest* adversary?

① $d = B - 1$

② $d = B$

③ $d = 2B$

What Happened

- The analysis of online algorithms can be seen as a game between the online algorithms and an adversary

- The adversary designs the next input according to the previous decisions of the online algorithm
  - The adversary tries its best to torture the online algorithm, punishes it for everything it does

- An adversary provides a *lower bound* of the competitive ratio

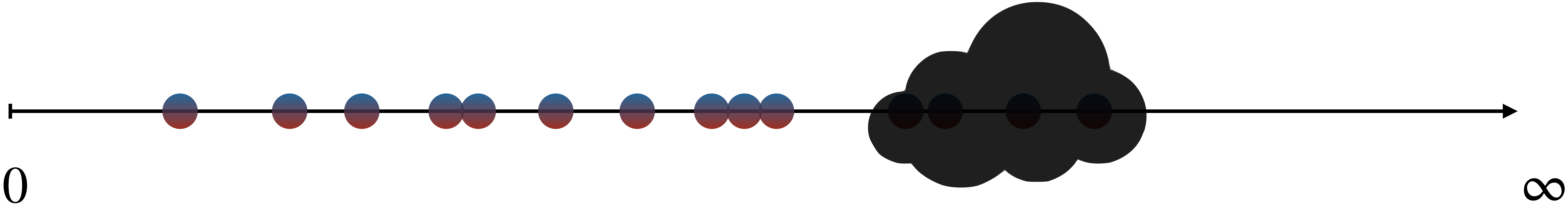
What is the goal for the project

- If you want to have some theoretical results for the online setting, you have to
  - Design an online algorithm
  - Prove that for any set of images and any set of unavailable time intervals, the cost (transmission completed time) of the online algorithm is at most c times of the optimal cost for some c

- You can start with finding the competitive ratio lower bound by designing an online algorithm and finding an adversary against it

Lower Bound and Tight Analysis

- In many cases, we don't know the optimal solution cost
  - There may be infinite instances
  - Even there are finite instances, we may not know the optimal solution behavior of all the instances

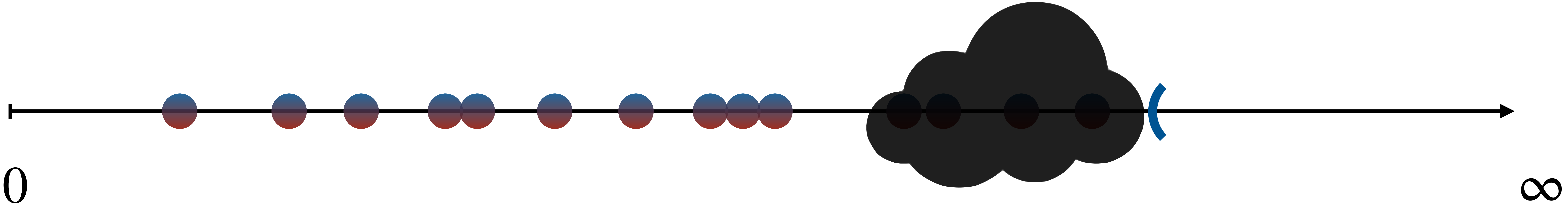


Lower Bound and Tight Analysis

- Sometimes we can only *upper-bound* the competitive ratio:

Argue that: For any instance I, ALG$(I) \leq x$ and OPT$(I) \geq y$. Therefore,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq \frac{x}{y}$$



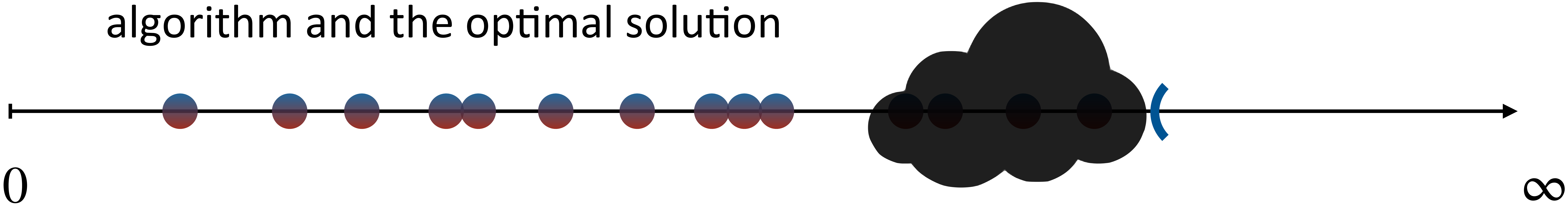
Lower Bound and Tight Analysis

- Sometimes we can only *upper-bound* the competitive ratio:

Argue that: For any instance I, ALG$(I) \leq x$ and OPT$(I) \geq y$. Therefore,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq \frac{x}{y}$$

- The upper bound of the competitive ratio relies on the bounds x and y, which need observations/arguments on the behavior of the online algorithm and the optimal solution



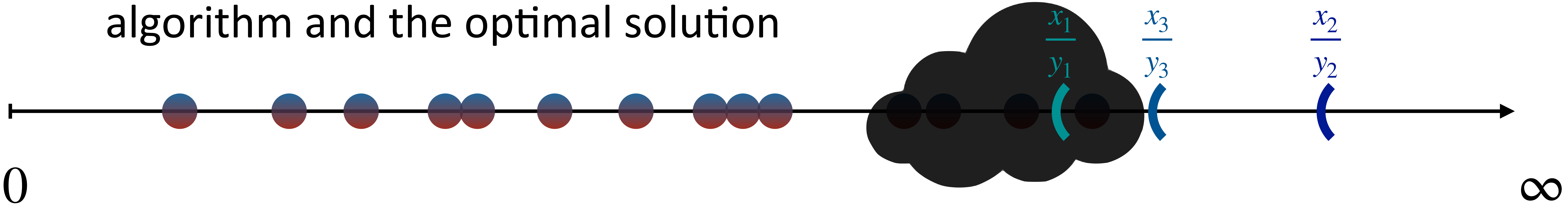
Lower Bound and Tight Analysis

- Sometimes we can only *upper-bound* the competitive ratio:

Argue that: For any instance I, ALG$(I) \leq x$ and OPT$(I) \geq y$. Therefore,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq \frac{x}{y}$$

- The upper bound of the competitive ratio relies on the bounds x and y, which need observations/arguments on the behavior of the online algorithm and the optimal solution

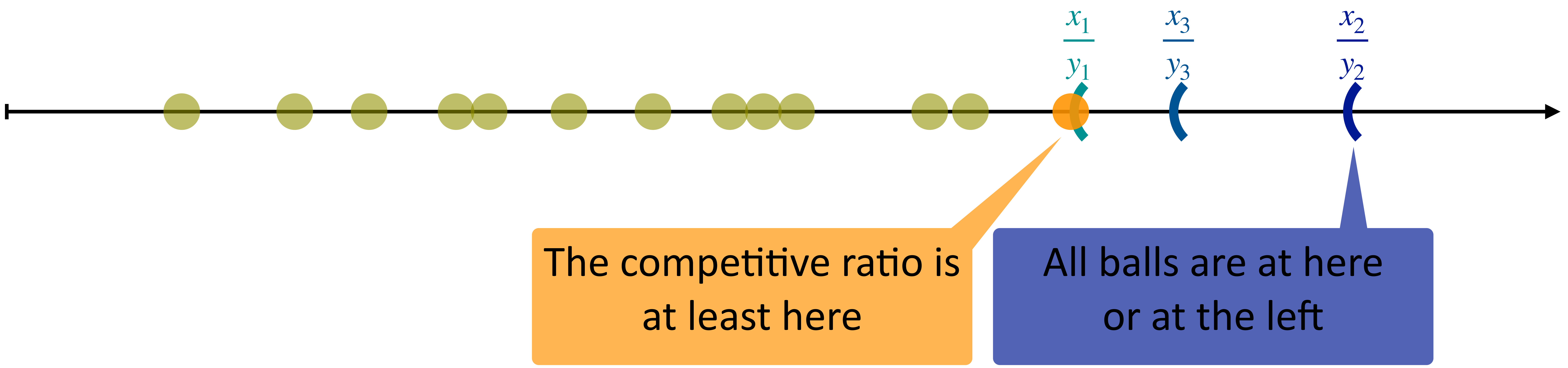


- Different x's and y's provide different upper bounds

Lower Bound and Tight Analysis

- An upper bound c is *tight* if

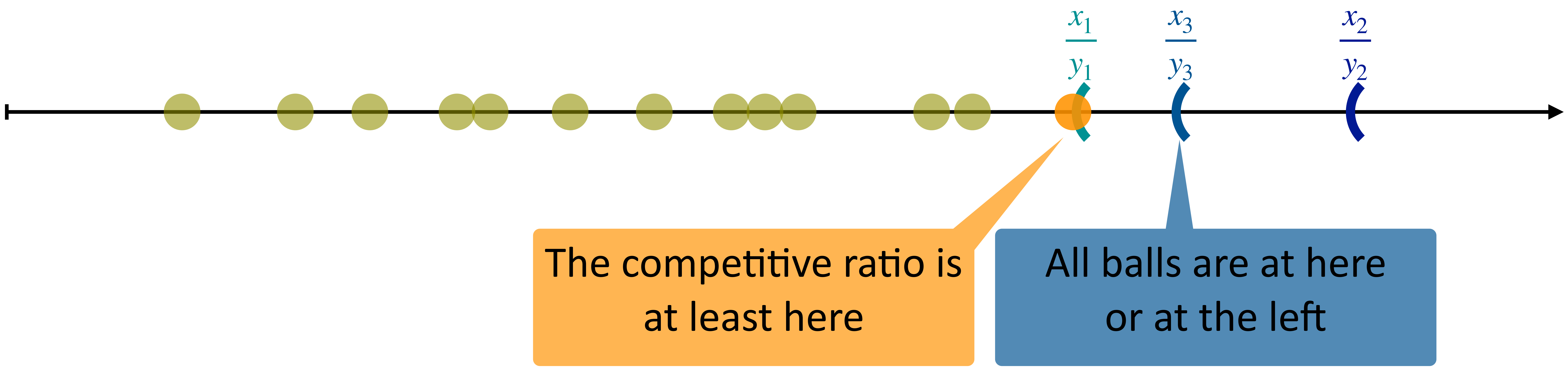
there exists an instance I^* such that $\dfrac{\mathrm{ALG}(I^*)}{\mathrm{OPT}(I^*)} = c$



Lower Bound and Tight Analysis

- An upper bound c is *tight* if

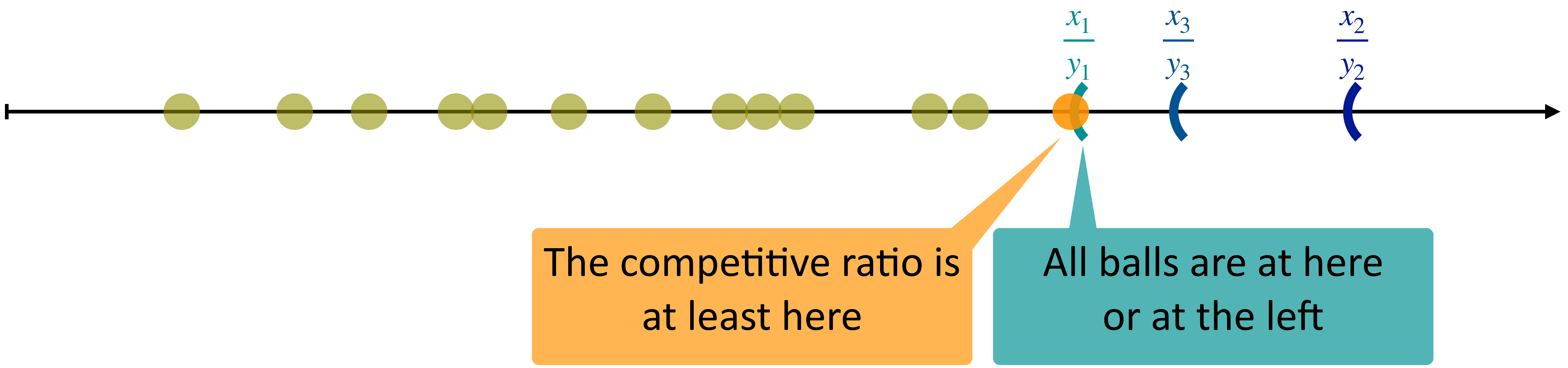
$$\text{there exists an instance } I^* \text{ such that } \frac{\mathrm{ALG}(I^*)}{\mathrm{OPT}(I^*)} = c$$



Lower Bound and Tight Analysis

- An upper bound c is *tight* if

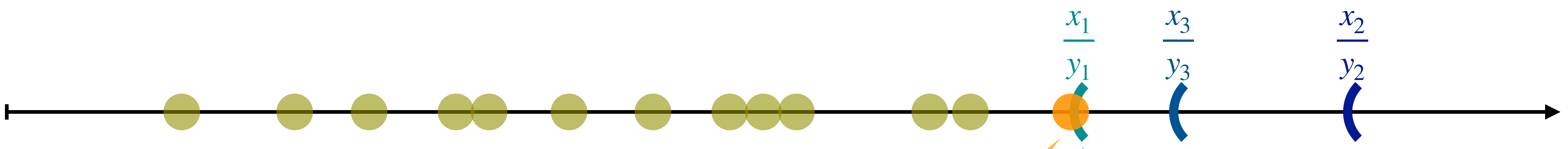
$$\text{there exists an instance } I^* \text{ such that } \frac{\mathsf{ALG}(I^*)}{\mathsf{OPT}(I^*)} = c$$



Lower Bound and Tight Analysis

- An upper bound c is *tight* if

$$\text{there exists an instance } I^* \text{ such that } \frac{\mathrm{ALG}(I^*)}{\mathrm{OPT}(I^*)} = c$$



The competitive ratio is at least here

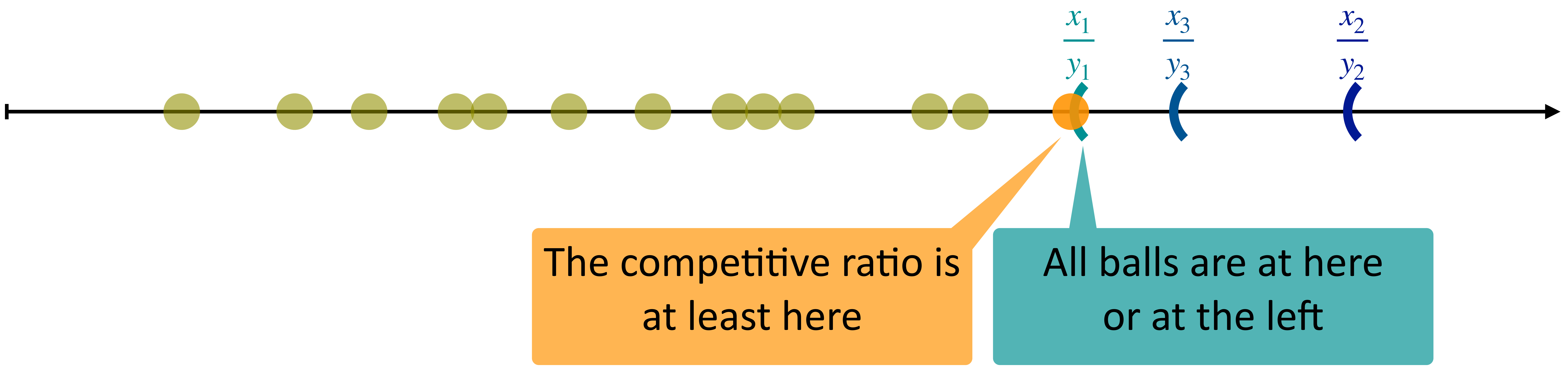
All balls are at here or at the left

The upper bound hits a "real ball" and cannot be pushed left further

Lower Bound and Tight Analysis

- An upper bound c is *tight* if

$$\text{there exists an instance } I^* \text{ such that } \frac{\mathrm{ALG}(I^*)}{\mathrm{OPT}(I^*)} = c$$

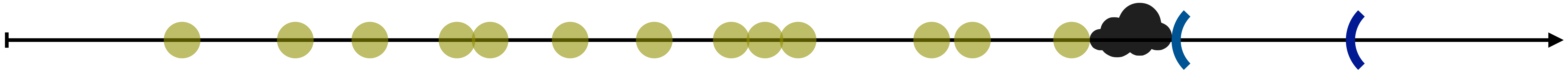


The upper bound hits a "real ball" and cannot be pushed left further

Lower Bound and Tight Analysis

- An upper bound c is *tight* if

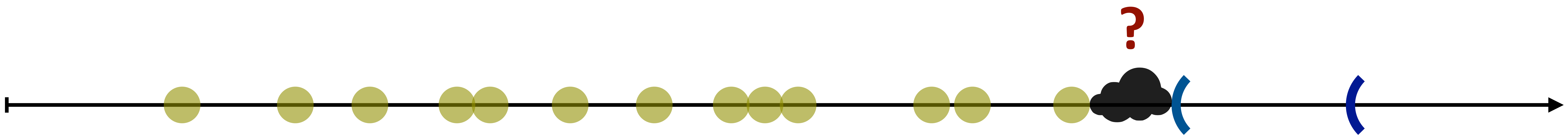
there exists an instance I^* such that $\dfrac{\text{ALG}(I^*)}{\text{OPT}(I^*)} = c$



Lower Bound and Tight Analysis

- An upper bound c is *tight* if

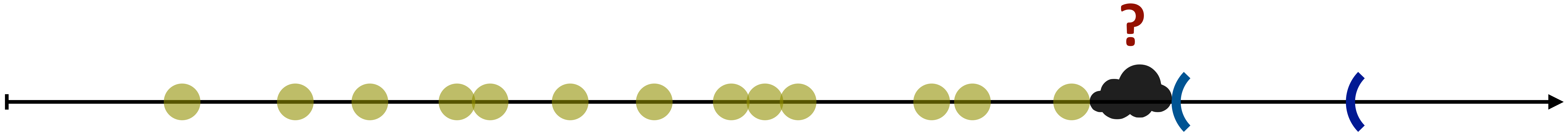
$$\text{there exists an instance } I^* \text{ such that } \frac{\text{ALG}(I^*)}{\text{OPT}(I^*)} = c$$



Lower Bound and Tight Analysis

- An upper bound c is *tight* if

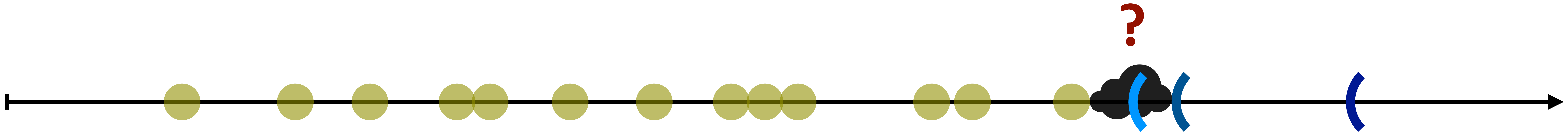
$$\text{there exists an instance } I^* \text{ such that } \frac{\mathsf{ALG}(I^*)}{\mathsf{OPT}(I^*)} = c$$



Lower Bound and Tight Analysis

- An upper bound c is *tight* if

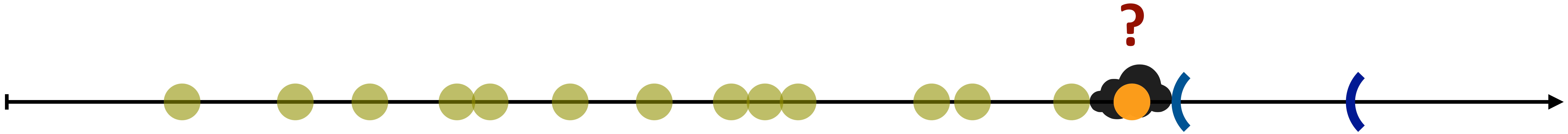
$$\text{there exists an instance } I^* \text{ such that } \frac{\text{ALG}(I^*)}{\text{OPT}(I^*)} = c$$



Lower Bound and Tight Analysis

- An upper bound c is *tight* if

there exists an instance I^* such that $\dfrac{\text{ALG}(I^*)}{\text{OPT}(I^*)} = c$



Loop of Online Algorithms Design

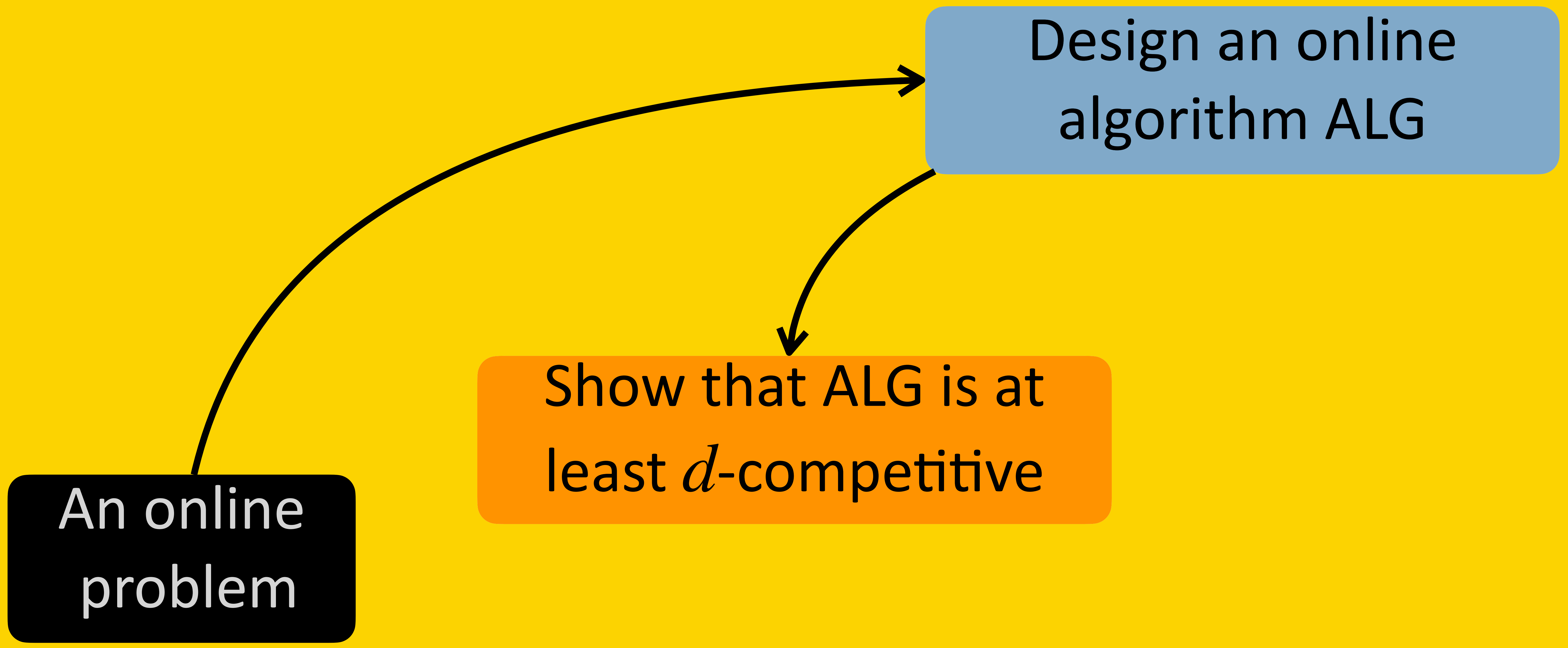
An online
problem

Loop of Online Algorithms Design

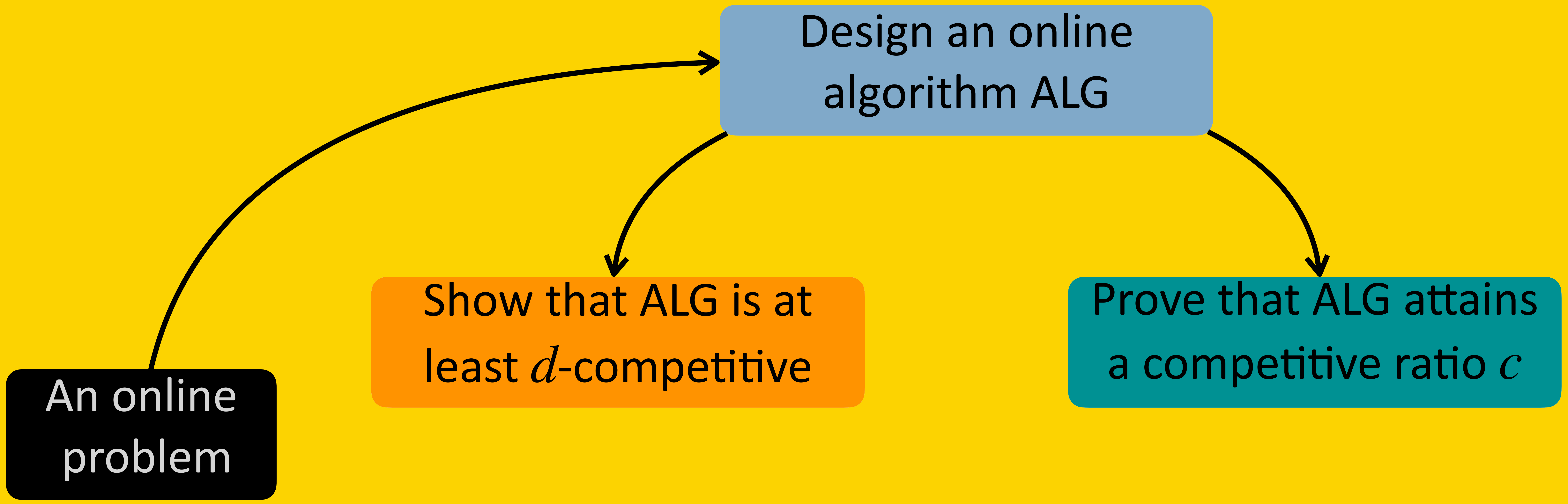
Design an online algorithm ALG

An online problem

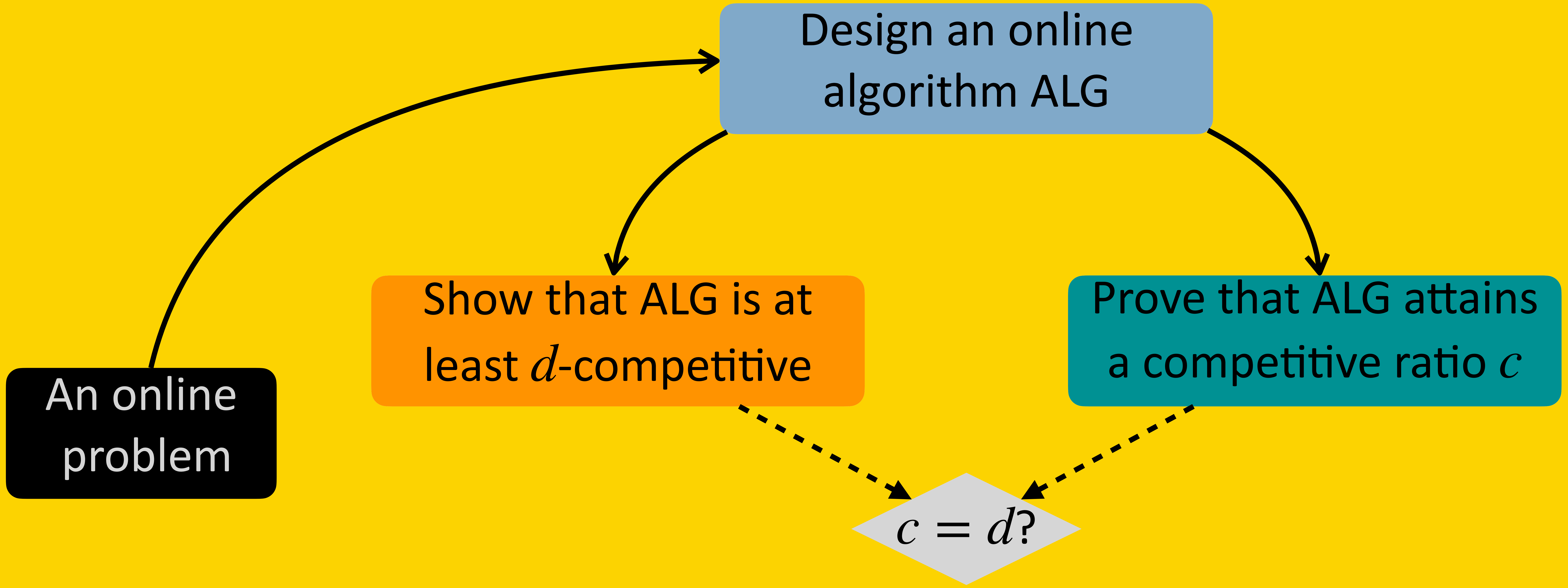
Loop of Online Algorithms Design



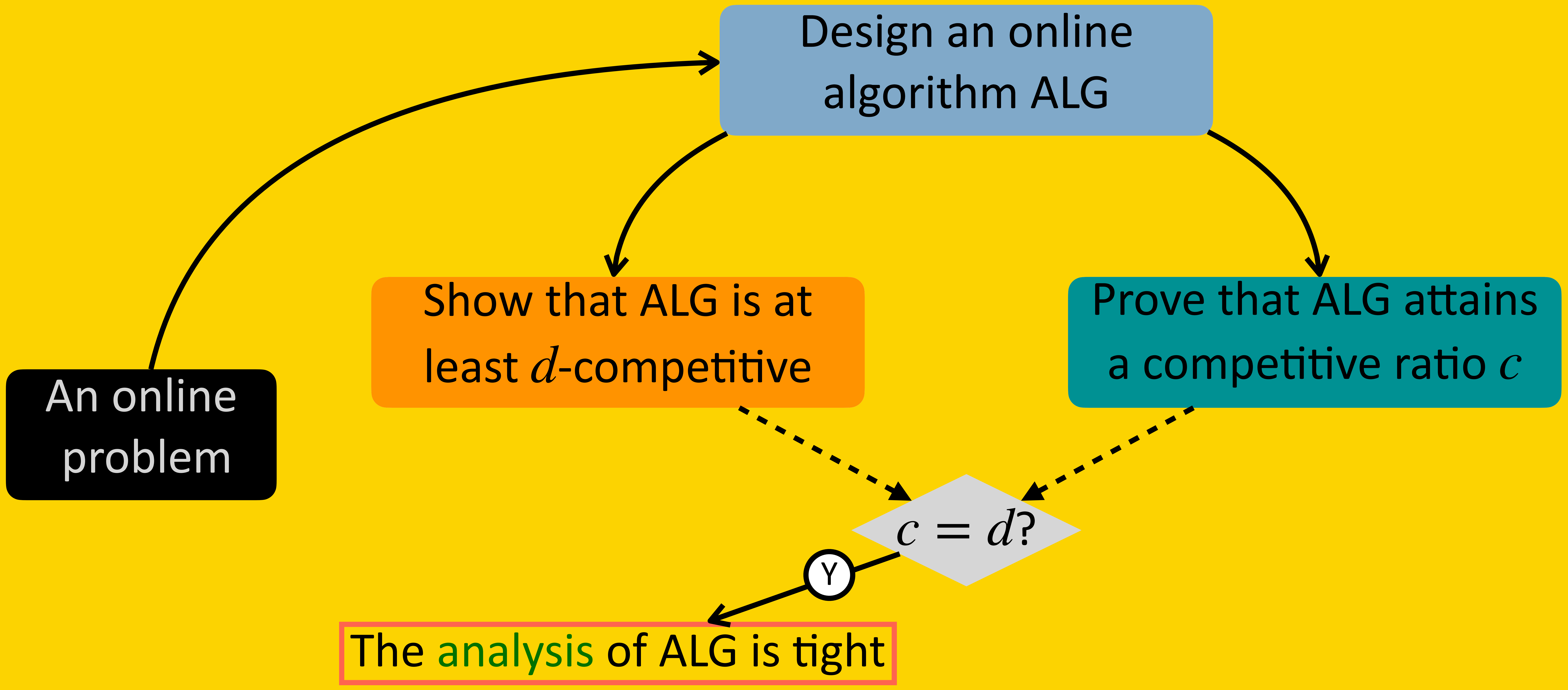
Loop of Online Algorithms Design



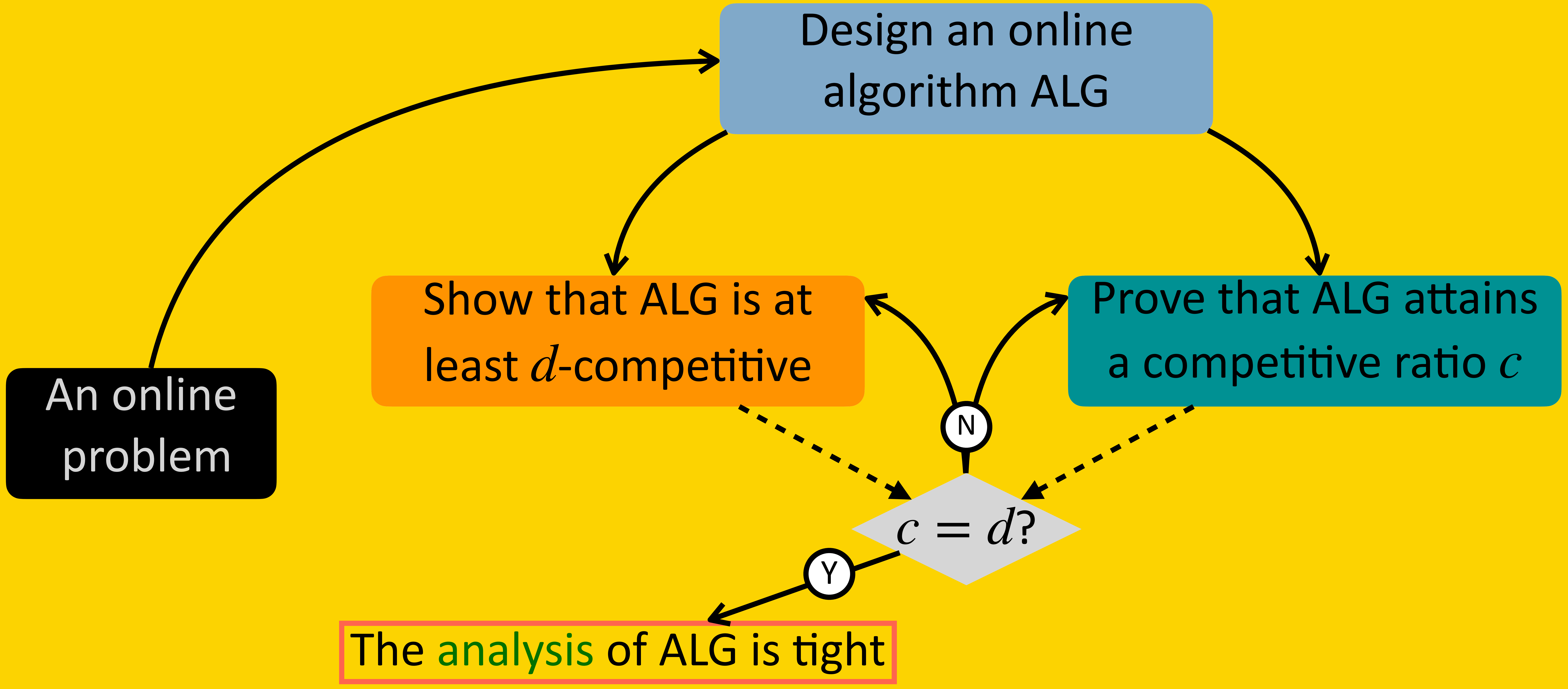
Loop of Online Algorithms Design



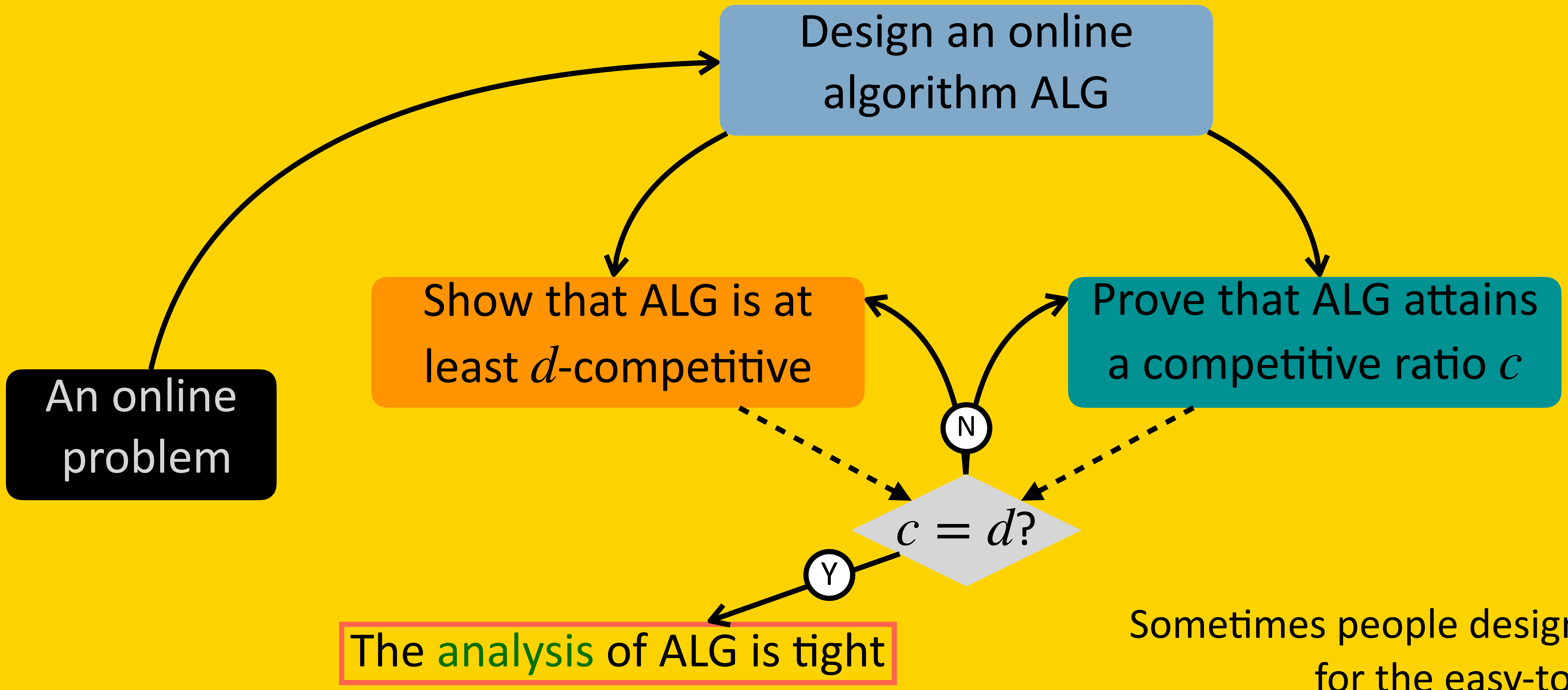
Loop of Online Algorithms Design



Loop of Online Algorithms Design



Loop of Online Algorithms Design



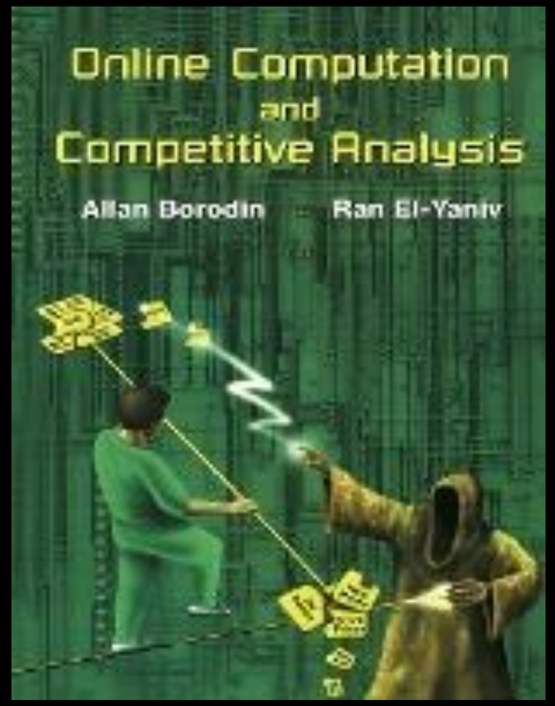
Sometimes people design online algorithms for the easy-to-analyze

Summary

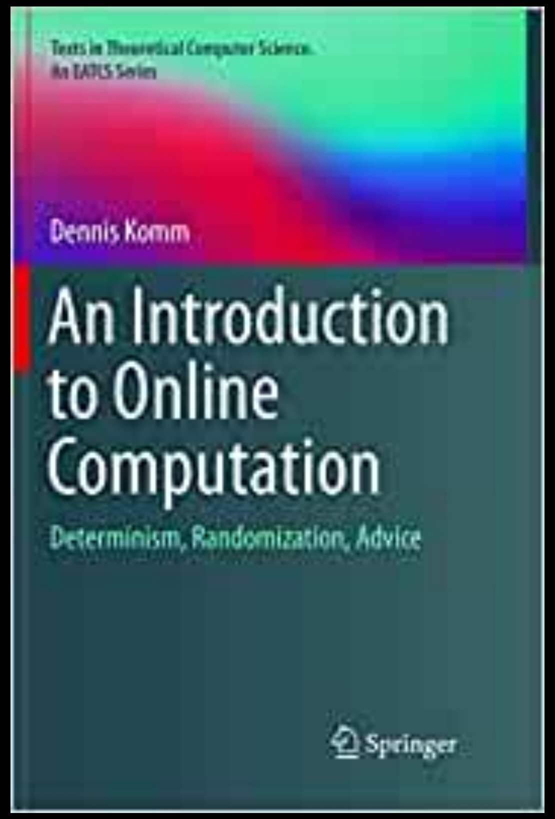
- Online optimization
- Measure the performance: **Competitive ratio**
  - How good is an online algorithm?
    - Show that the algorithm is (*at most*) c-competitive. (For all instance I, $\dfrac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$)
  - How bad is an online algorithm?
    - Adversary game
    - Find an adversary for the algorithm and prove that it cannot be better than c-competitive
      - That is, it is *at least* c-competitive
  - **Tight analysis**: Find an adversary I' for the algorithm such that $\dfrac{\text{ALG}(I')}{\text{OPT}(I')}$ meets the lower bound.

Online Algorithms Books

- *Online Computation and Competitive Analysis Paperback English*
  by Allan Borodin and Ran El-Yaniv



- *An Introduction to Online Computation: Determinism, Randomization, Advice*
  by Dennis Komm



- *Online Algorithms: The State Of The Art*
  by Amos Fiat and Gerhard J. Woeginger