The Secretary Problem with Independent Sampling

Alexandros Tsigonias-Dimitriadis, TU Munich

Joint work with José Correa, Andrés Cristi (Universidad de Chile),
Laurent Feuilloley (Université Lyon 1), Tim Oosterwijk (Vrije Universiteit Amsterdam)

Preliminary version appeared in SODA 2021

INFORMS 2021 Annual Meeting - Virtual Talk

Technische Universität München

Motivation

Motivation

Sequential Decision Making under uncertainty is a fundamental problem that bridges several areas.

- CS: Online algorithms (traditionally worst-case analysis)
- Applied Probability / Statistics: Optimal Stopping
- MS / OR: Markov Decision Processes
- Game Theory: Stochastic Games (strategic interactions)

Motivation

Sequential Decision Making under uncertainty is a fundamental problem that bridges several areas.

- CS: Online algorithms (traditionally worst-case analysis)
- Applied Probability / Statistics: Optimal Stopping
- MS / OR: Markov Decision Processes
- Game Theory: Stochastic Games (strategic interactions)

Many applications in economics and management:

- Pricing in e-commerce
- Search Theory
- Resource Allocation
- Finance

A very old problem

A very old problem

- The German astronomer **Johannes Kepler** in **1613** tried to describe the decision-making process when he decided to interview eleven candidates to find his second wife.

A very old problem

- The German astronomer **Johannes Kepler** in **1613** tried to describe the decision-making process when he decided to interview eleven candidates to find his second wife.
- **Arthur Cayley** in **1875** posed the following problem:
  There is a lottery with n available tickets, each with a hidden value. A person can draw a ticket, observe the value, and decide whether to draw another ticket. She always receives the value of the last ticket drawn, and she can draw at most k tickets. If the hidden values come from a known probability distribution, what is the expected reward of the player?

A very old problem

- The German astronomer **Johannes Kepler** in **1613** tried to describe the decision-making process when he decided to interview eleven candidates to find his second wife.
- **Arthur Cayley** in **1875** posed the following problem:
  There is a lottery with n available tickets, each with a hidden value. A person can draw a ticket, observe the value, and decide whether to draw another ticket. She always receives the value of the last ticket drawn, and she can draw at most k tickets. If the hidden values come from a known probability distribution, what is the expected reward of the player?

Problem seems to be forgotten until…

A very old problem

- The German astronomer **Johannes Kepler** in **1613** tried to describe the decision-making process when he decided to interview eleven candidates to find his second wife.
- **Arthur Cayley** in **1875** posed the following problem:
  There is a lottery with n available tickets, each with a hidden value. A person can draw a ticket, observe the value, and decide whether to draw another ticket. She always receives the value of the last ticket drawn, and she can draw at most k tickets. If the hidden values come from a known probability distribution, what is the expected reward of the player?

Problem seems to be forgotten until…

- **Moser** in **1956** revisits the problem for the special case of i.i.d. $X_1, X_2, \ldots, X_n \sim U(0,1)$.

A very old problem

- The German astronomer **Johannes Kepler** in **1613** tried to describe the decision-making process when he decided to interview eleven candidates to find his second wife.
- **Arthur Cayley** in **1875** posed the following problem:
  There is a lottery with n available tickets, each with a hidden value. A person can draw a ticket, observe the value, and decide whether to draw another ticket. She always receives the value of the last ticket drawn, and she can draw at most k tickets. If the hidden values come from a known probability distribution, what is the expected reward of the player?

Problem seems to be forgotten until…

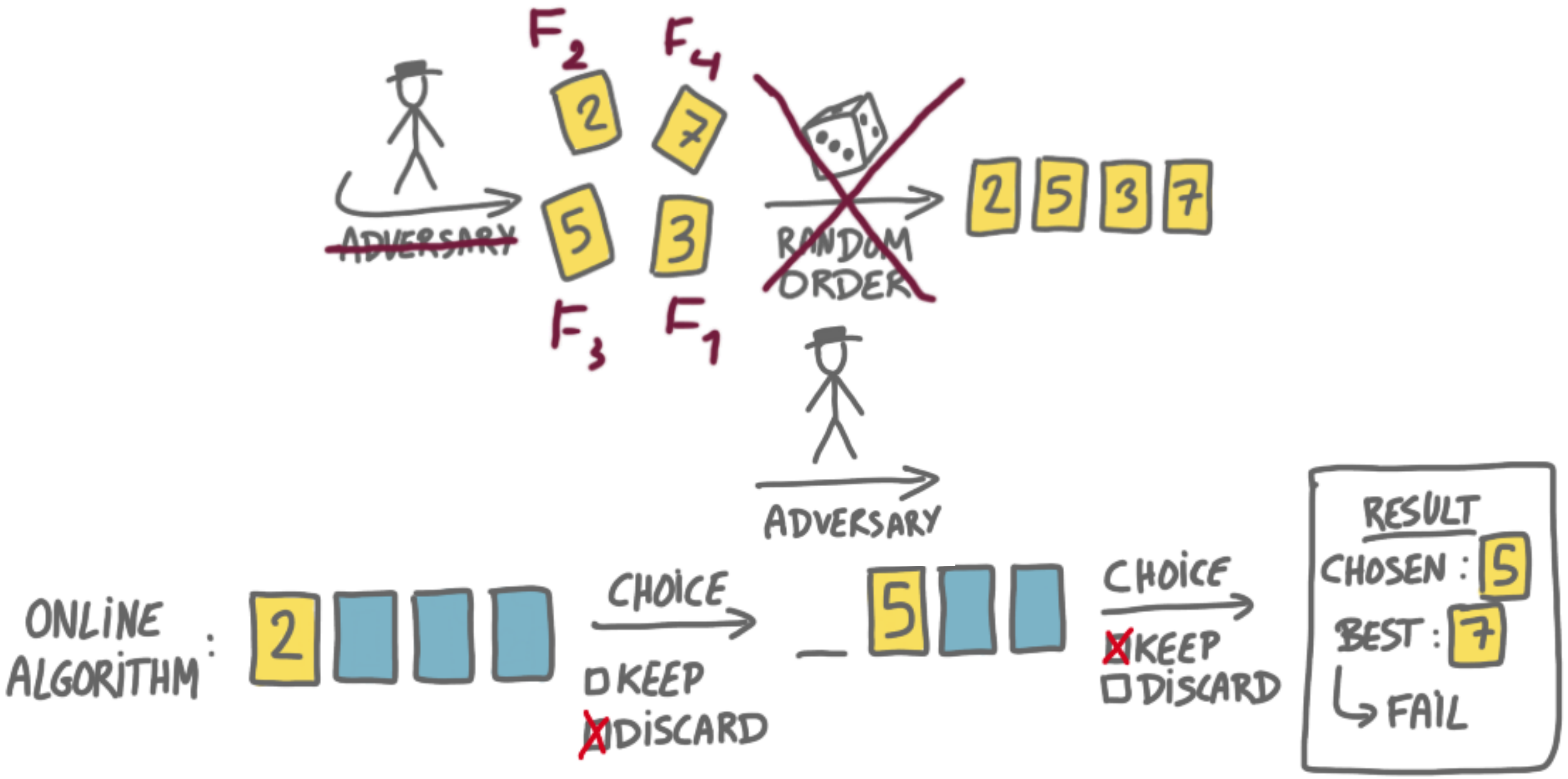
- **Moser** in **1956** revisits the problem for the special case of i.i.d. $X_1, X_2, \ldots, X_n \sim U(0,1)$.
- He solves the limit version for n using dynamic programming.

Secretary Problem & Prophet Inequality

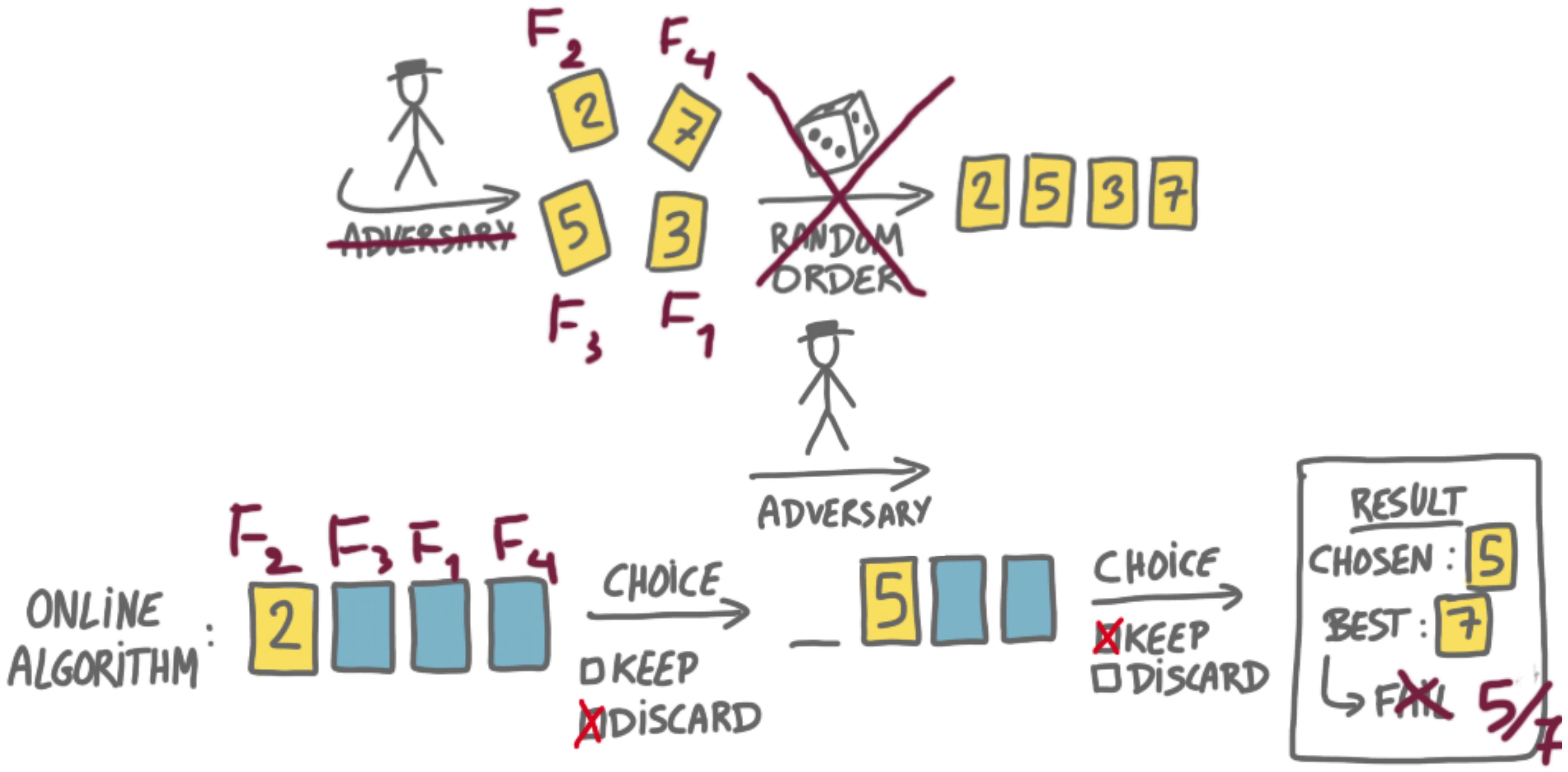
Secretary Problem & Prophet Inequality

Secretary Problem & Prophet Inequality

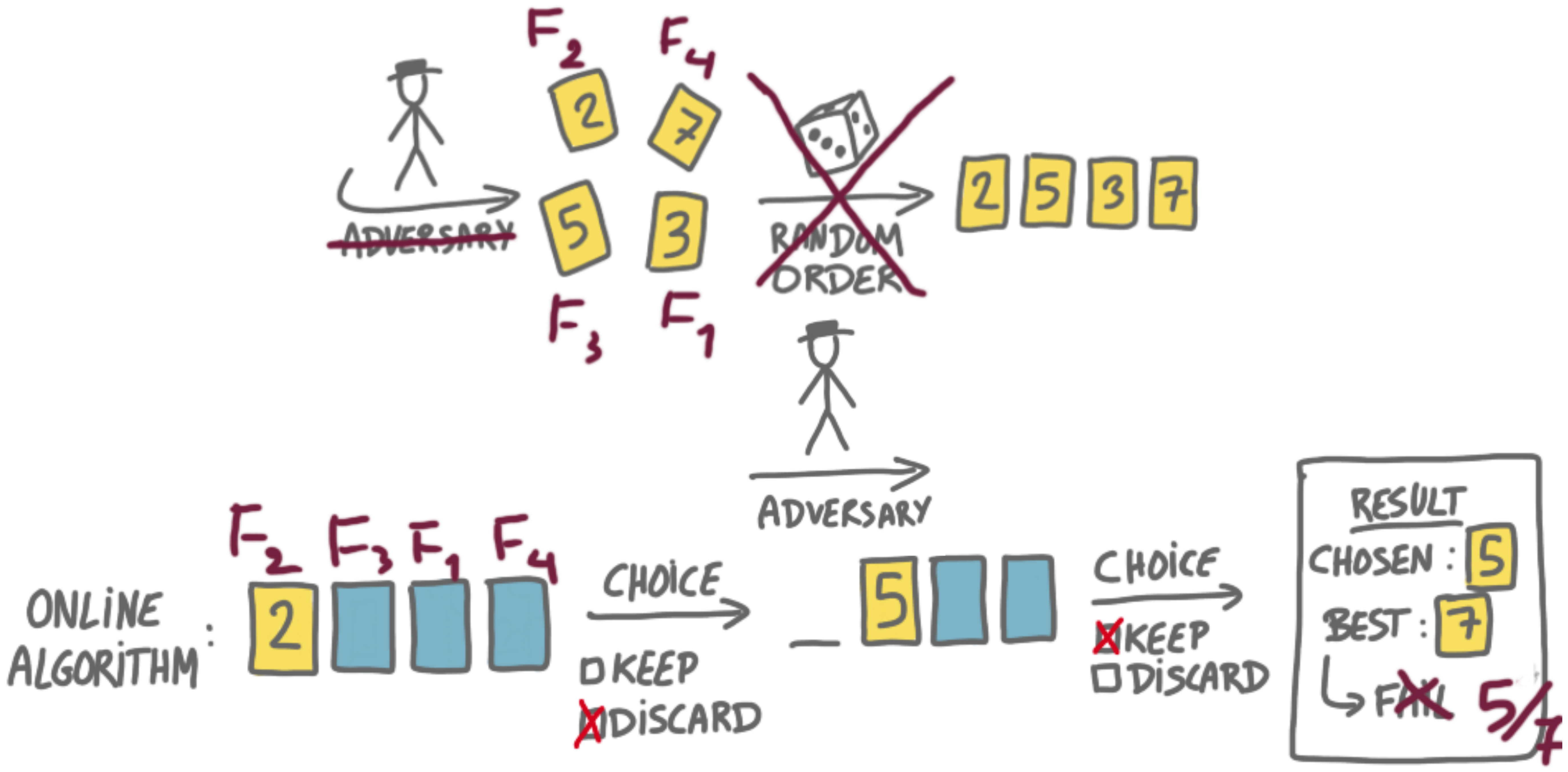
Secretary Problem & Prophet Inequality



Secretary Problem & Prophet Inequality



Secretary Problem & Prophet Inequality



Secretary problem

- Adversarial values
- Random order
- Objective: $\max \Pr[\text{pick the highest value}]$

Prophet inequality

- Values from known distributions
- Adversarial order
- Objective: $\max \mathbb{E}[X_t] \geq c \cdot \mathbb{E}\left[\max X_i\right]$ (stop at t)

Results for Secretary & Prophets

Results for Secretary & Prophets

Theorem (attributed to [Lindley '61] , [Dynkin '63], and others)

Best possible algorithm for the *secretary problem:* Look at first n/e values and then select the first element with the largest value seen so far. Then we pick the maximum with probability $1/e$.

Results for Secretary & Prophets

Theorem (attributed to [Lindley '61] , [Dynkin '63], and others)
Best possible algorithm for the *secretary problem:* Look at first n/e values and then select the first element with the largest value seen so far. Then we pick the maximum with probability $1/e$.

If the order is **adversarial** not much we can do. No algorithm can achieve a constant success guarantee.

Results for Secretary & Prophets

Theorem (attributed to [Lindley '61] , [Dynkin '63], and others)
Best possible algorithm for the *secretary problem:* Look at first n/e values and then select the first element with the largest value seen so far. Then we pick the maximum with probability $1/e$.

If the order is **adversarial** not much we can do. No algorithm can achieve a constant success guarantee.

Theorem ([Krengel, Sucheston, Garling '77])
For the prophet inequality, there exists a stopping rule t such that

$$\max_{t \text{ stopping time}} \mathbb{E}[X_t] \geq \frac{1}{2} \cdot \mathbb{E}\left[\max_{i \in [n]} X_i\right].$$

Moreover, the factor $1/2$ is best possible.

Results for Secretary & Prophets

Theorem (attributed to [Lindley '61] , [Dynkin '63], and others)
Best possible algorithm for the *secretary problem:* Look at first n/e values and then select the first element with the largest value seen so far. Then we pick the maximum with probability $1/e$.

If the order is **adversarial** not much we can do. No algorithm can achieve a constant success guarantee.

Theorem ([Krengel, Sucheston, Garling '77])
For the prophet inequality, there exists a stopping rule t such that

$$\max_{t \text{ stopping time}} \mathbb{E}[X_t] \geq \frac{1}{2} \cdot \mathbb{E}\left[\max_{i \in [n]} X_i\right].$$

Moreover, the factor $1/2$ is best possible.

It can be achieved by setting a single threshold T and accepting the first value that exceeds it [Samuel-Cahn '84] (also [Kleinberg, Weinberg '12]).

Online algorithms with samples

Online algorithms with samples

Prophet inequality

Online algorithms with samples

Prophet inequality

- Initiated by [Azar, Kleinberg, Weinberg '14].

Online algorithms with samples

Prophet inequality

- Initiated by [Azar, Kleinberg, Weinberg '14].
- If the distributions are unknown, just **one sample** from each distribution suffices to obtain the optimal guarantee of $1/2$! [Rubinstein, Wang, Weinberg '20]

Online algorithms with samples

Prophet inequality

- Initiated by [Azar, Kleinberg, Weinberg '14].
- If the distributions are unknown, just **one sample** from each distribution suffices to obtain the optimal guarantee of $1/2$! [Rubinstein, Wang, Weinberg '20]
- Values drawn i.i.d. from an unknown distribution [Correa, Dütting, Fischer, Schewior '19; RWW '20]
- Random order ("prophet secretary"), one sample from each distribution [Correa, Cristi, Epstein, Soto '20; Kaplan, Naori, Raz '20].

Online algorithms with samples

Prophet inequality

- Initiated by [Azar, Kleinberg, Weinberg '14].
- If the distributions are unknown, just **one sample** from each distribution suffices to obtain the optimal guarantee of $1/2$! [Rubinstein, Wang, Weinberg '20]
- Values drawn i.i.d. from an unknown distribution [Correa, Dütting, Fischer, Schewior '19; RWW '20]
- Random order ("prophet secretary"), one sample from each distribution [Correa, Cristi, Epstein, Soto '20; Kaplan, Naori, Raz '20].

Secretary (or secretary-like)

- A fraction h of the values is sampled [Kaplan, Naori, Raz '20].
- General model that also captures secretary with samples [Dütting, Lattanzi, Paes Leme, Vassilvitskii '21].

Secretary - Distributional information helps

Secretary - Distributional information helps

- Random order, values i.i.d. from a ***known*** distribution: Algorithm that picks the max with prob $\gamma \approx 0.5801$ and this is best possible. [Gilbert, Mosteller '66]

Secretary - Distributional information helps

- Random order, values i.i.d. from a ***known*** distribution: Algorithm that picks the max with prob $\gamma \approx 0.5801$ and this is best possible. [Gilbert, Mosteller '66]
- Random order, independent values from known distributions: Multi-threshold strategy that picks the max with prob *at least* γ. [Nuti '20; Esfandiari, Hajiaghayi, Lucier, Mitzenmacher '20]

Secretary - Distributional information helps

- Random order, values i.i.d. from a ***known*** distribution: Algorithm that picks the max with prob $\gamma \approx 0.5801$ and this is best possible. [Gilbert, Mosteller '66]
- Random order, independent values from known distributions: Multi-threshold strategy that picks the max with prob *at least* γ. [Nuti '20; Esfandiari, Hajiaghayi, Lucier, Mitzenmacher '20]
- Adversarial order, again independent values again from known distributions: Single-threshold strategy that picks the max with probability $1/e$ and this is tight. [Allart, Islas '15]

Secretary - Distributional information helps

- Random order, values i.i.d. from a ***known*** distribution: Algorithm that picks the max with prob $\gamma \approx 0.5801$ and this is best possible. [Gilbert, Mosteller '66]
- Random order, independent values from known distributions: Multi-threshold strategy that picks the max with prob *at least* γ. [Nuti '20; Esfandiari, Hajiaghayi, Lucier, Mitzenmacher '20]
- Adversarial order, again independent values again from known distributions: Single-threshold strategy that picks the max with probability $1/e$ and this is tight. [Allart, Islas '15]

Question: Can we design a model which nicely interpolates between the classic secretary (where there is no additional information) and drawing values from fully known distribution(s)?

Our model - Secretary with independent sampling

Our model - Secretary with independent sampling

Resembles the classic secretary but has some additional information in the form of samples.

Our model - Secretary with independent sampling

Resembles the classic secretary but has some additional information in the form of samples.

- n adversarial values $v_1, v_2, \cdots, v_n$
- Each value is sampled independently with probability p.
- We get the set of samples S and the sampling probability p.
- The set of non-sampled values V is presented online in the order dictated by σ.
- Objective: Max the prob of picking the **maximum value in** V.

Our model - Secretary with independent sampling

Resembles the classic secretary but has some additional information in the form of samples.

- n adversarial values $v_1, v_2, \cdots, v_n$
- Each value is sampled independently with probability p.
- We get the set of samples S and the sampling probability p.
- The set of non-sampled values V is presented online in the order dictated by σ.
- Objective: Max the prob of picking the **maximum value in** V.

The problem comes in two versions:

AOSp: σ is adversarial

ROSp: σ is a uniform random permutation

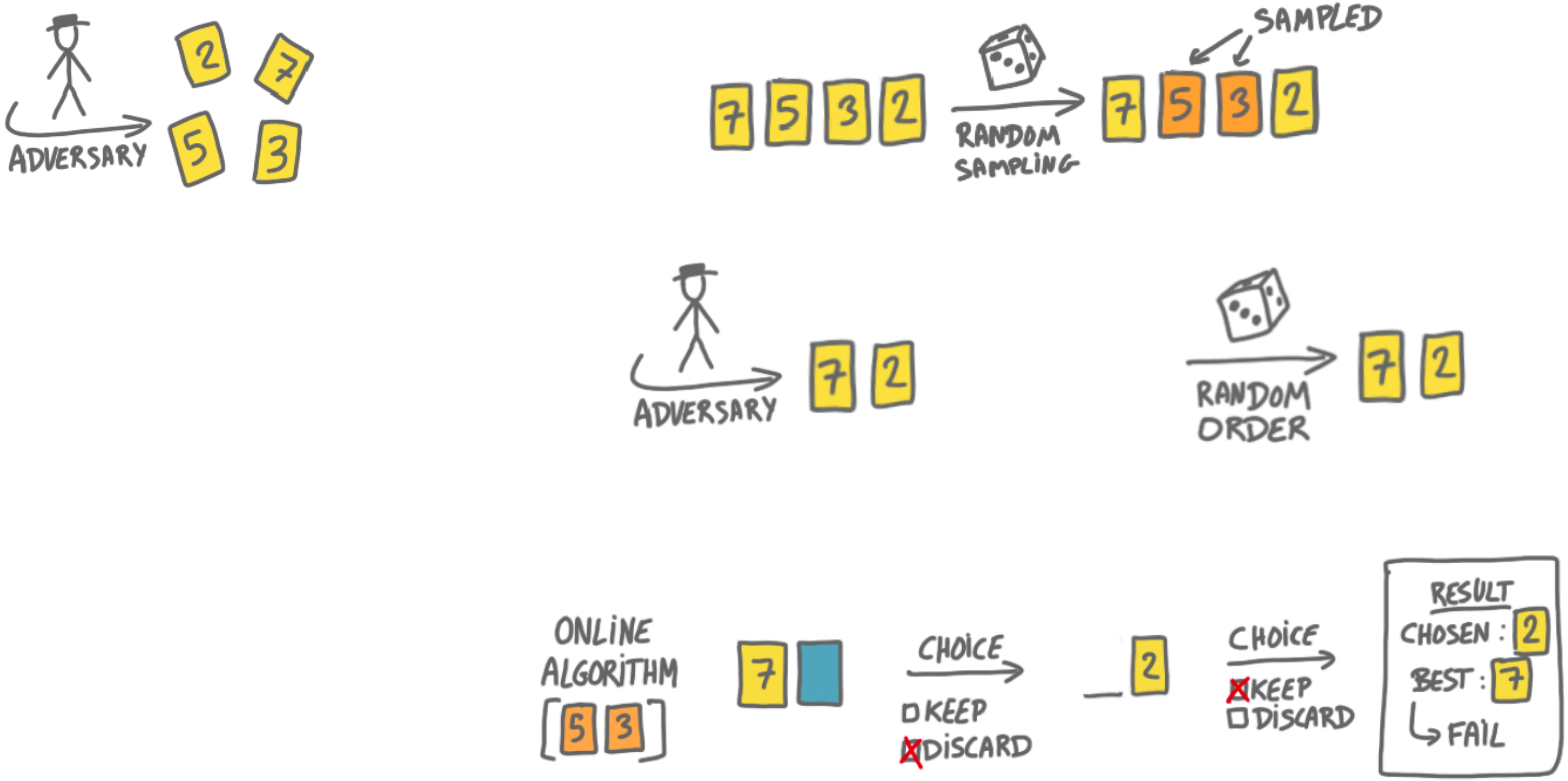
Our model - Secretary with independent sampling

Our model - Secretary with independent sampling

Our model - Secretary with independent sampling

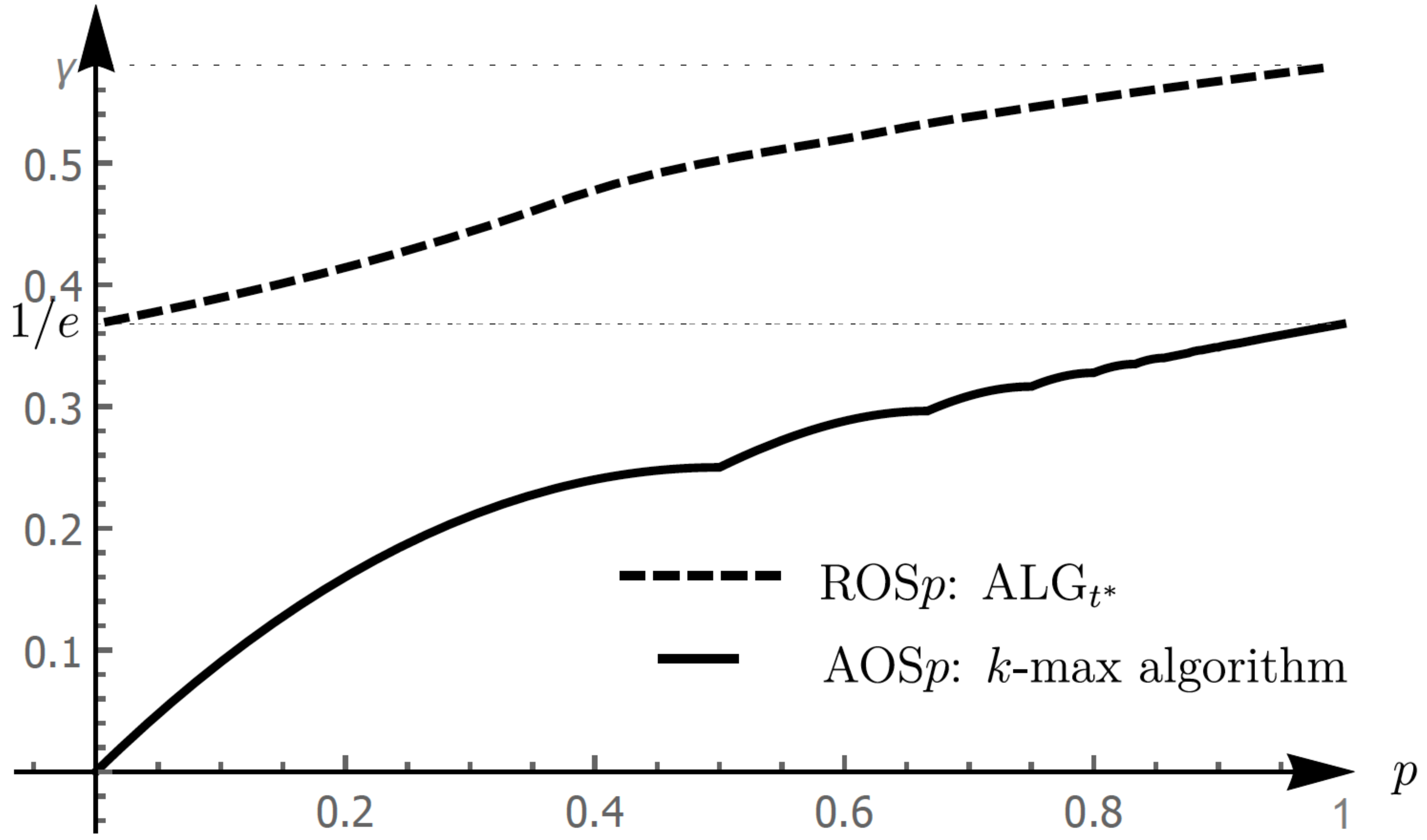
Our model - Secretary with independent sampling

Our model - Secretary with independent sampling



Our results

We obtain best possible algorithms for **AOSp** and **ROSp** for ***any*** value of p.



Results for AOSp

Results for AOSp

Theorem **(k-max algorithm is best possible)**
Consider the following family of algorithms: Set the k-th largest sample as the threshold and accept the first value in V that exceeds this threshold.

Algorithm: Intuitively k should increase as $p \to 1$. Set $k(p)$ to be $\left\lfloor \frac{1}{1-p} \right\rfloor$.

No algorithm can achieve a better success guarantee.

Results for AOSp

Theorem (**k-max algorithm is best possible**)
Consider the following family of algorithms: Set the k-th largest sample as the threshold and accept the first value in V that exceeds this threshold.

Algorithm: Intuitively k should increase as $p \to 1$. Set $k(p)$ to be $\left\lfloor \frac{1}{1-p} \right\rfloor$.

No algorithm can achieve a better success guarantee.

Results for AOSp

Theorem (**k-max algorithm is best possible**)
Consider the following family of algorithms: Set the k-th largest sample as the threshold and accept the first value in V that exceeds this threshold.

Algorithm: Intuitively k should increase as $p \to 1$. Set $k(p)$ to be $\left\lfloor \frac{1}{1-p} \right\rfloor$.

No algorithm can achieve a better success guarantee.

Bound for **deterministic** algos
for $p = 1/2$.
Increasing seq, unknown length n,
actual values irrelevant.

Results for AOSp

Theorem (k-max algorithm is best possible)
Consider the following family of algorithms: Set the k-th largest sample as the threshold and accept the first value in V that exceeds this threshold.

Algorithm: Intuitively k should increase as $p \to 1$. Set $k(p)$ to be $\left\lfloor \frac{1}{1-p} \right\rfloor$.

No algorithm can achieve a better success guarantee.

Bound for **deterministic** algos for $p = 1/2$.
Increasing seq, unknown length n, actual values irrelevant.

→

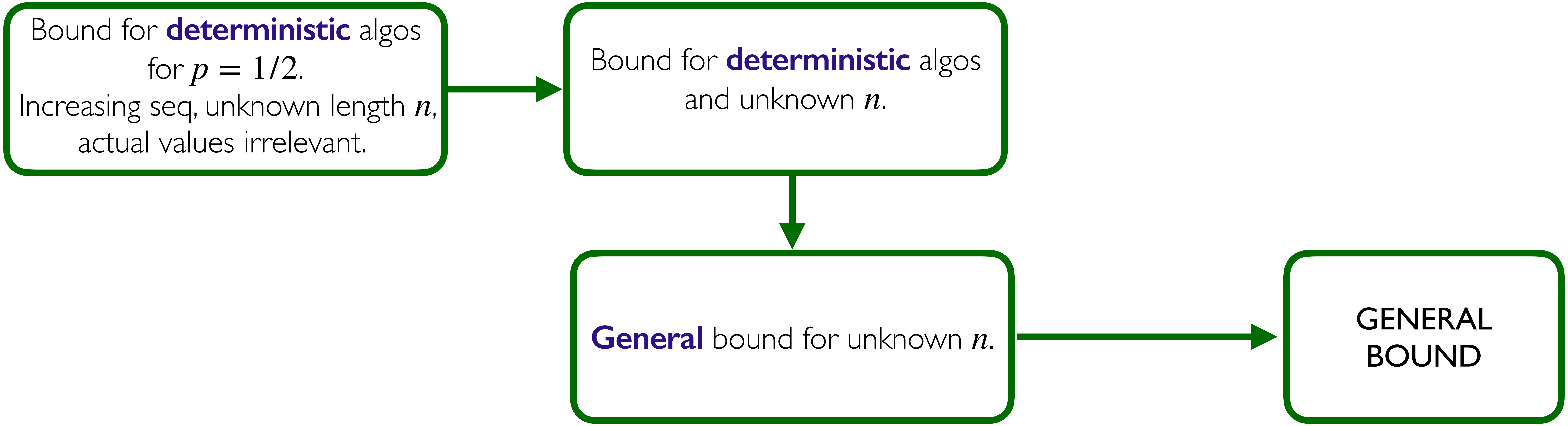
Bound for **deterministic** algos and unknown n.

Results for AOSp

Theorem **(k-max algorithm is best possible)**
Consider the following family of algorithms: Set the k-th largest sample as the threshold and accept the first value in V that exceeds this threshold.

Algorithm: Intuitively k should increase as $p \to 1$. Set $k(p)$ to be $\left\lfloor \frac{1}{1-p} \right\rfloor$.

No algorithm can achieve a better success guarantee.



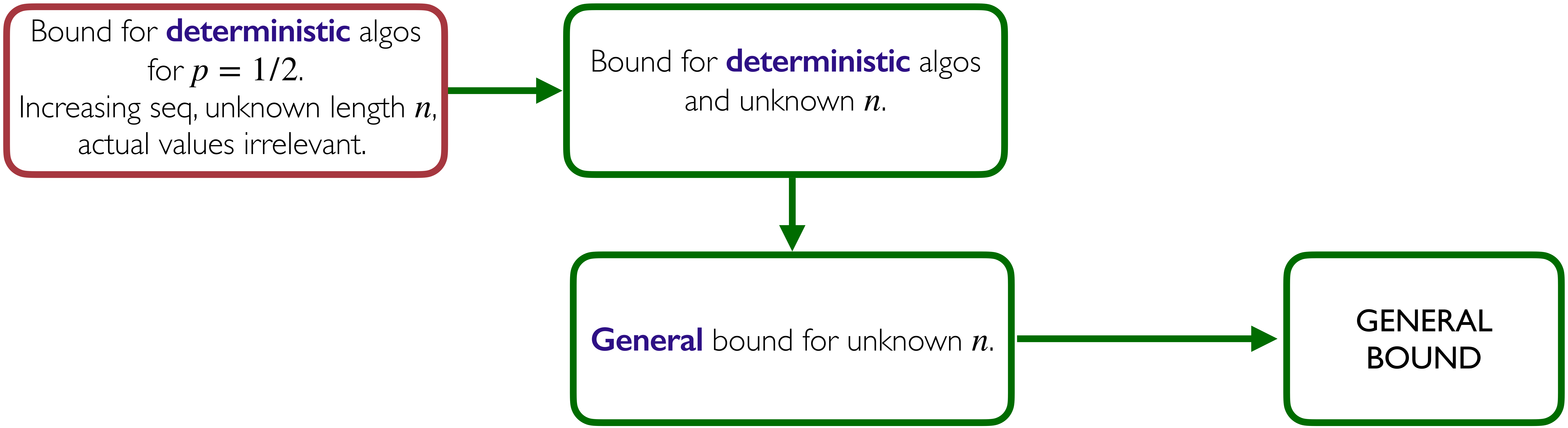
Results for AOSp

Theorem (k-max algorithm is best possible)

Consider the following family of algorithms: Set the k-th largest sample as the threshold and accept the first value in V that exceeds this threshold.

Algorithm: Intuitively k should increase as $p \to 1$. Set $k(p)$ to be $\left\lfloor \frac{1}{1-p} \right\rfloor$.

No algorithm can achieve a better success guarantee.



Last zero problem

Last zero problem

- Adversary chooses an integer n, unknown to us.

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1-p$.

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1 - p$.
- Referee tells us the total number of 1s and the probability p.

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1 - p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1-p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1-p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

$p = 1/2 \quad 6 \times$ 

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1 - p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

$p = 1/2 \quad 6 \times$ 

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1 - p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

$p = 1/2 \quad 6 \times$ 

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1-p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

$p = 1/2 \quad 6 \times$ 1

1 0 0 0 1 1 0 1 1

Last zero problem

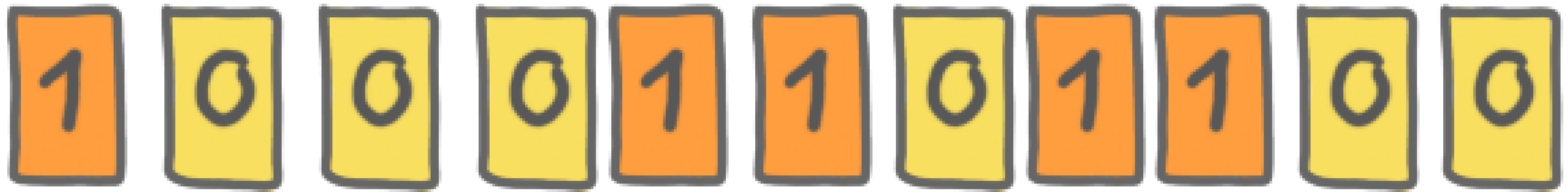
- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1-p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

$p = 1/2 \quad 6 \times$ 

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1 - p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

$p = 1/2 \quad 6 \times$ 



Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1 - p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

$p = 1/2 \quad 6 \times$ 

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1 - p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

$p = 1/2 \quad 6 \times$ 

END

Last zero problem

- Adversary chooses an integer n, unknown to us.
- Referee writes on each of the n cards 1 w.p. p and 0 w.p. $1-p$.
- Referee tells us the total number of 1s and the probability p.
- Cards are revealed to us one by one in the order of the deck.
- Goal: We win if we stop at the last 0.

$p = 1/2 \quad 6 \times$ 

END

We will show that we cannot do better than $1/4$ for ***any*** length n.

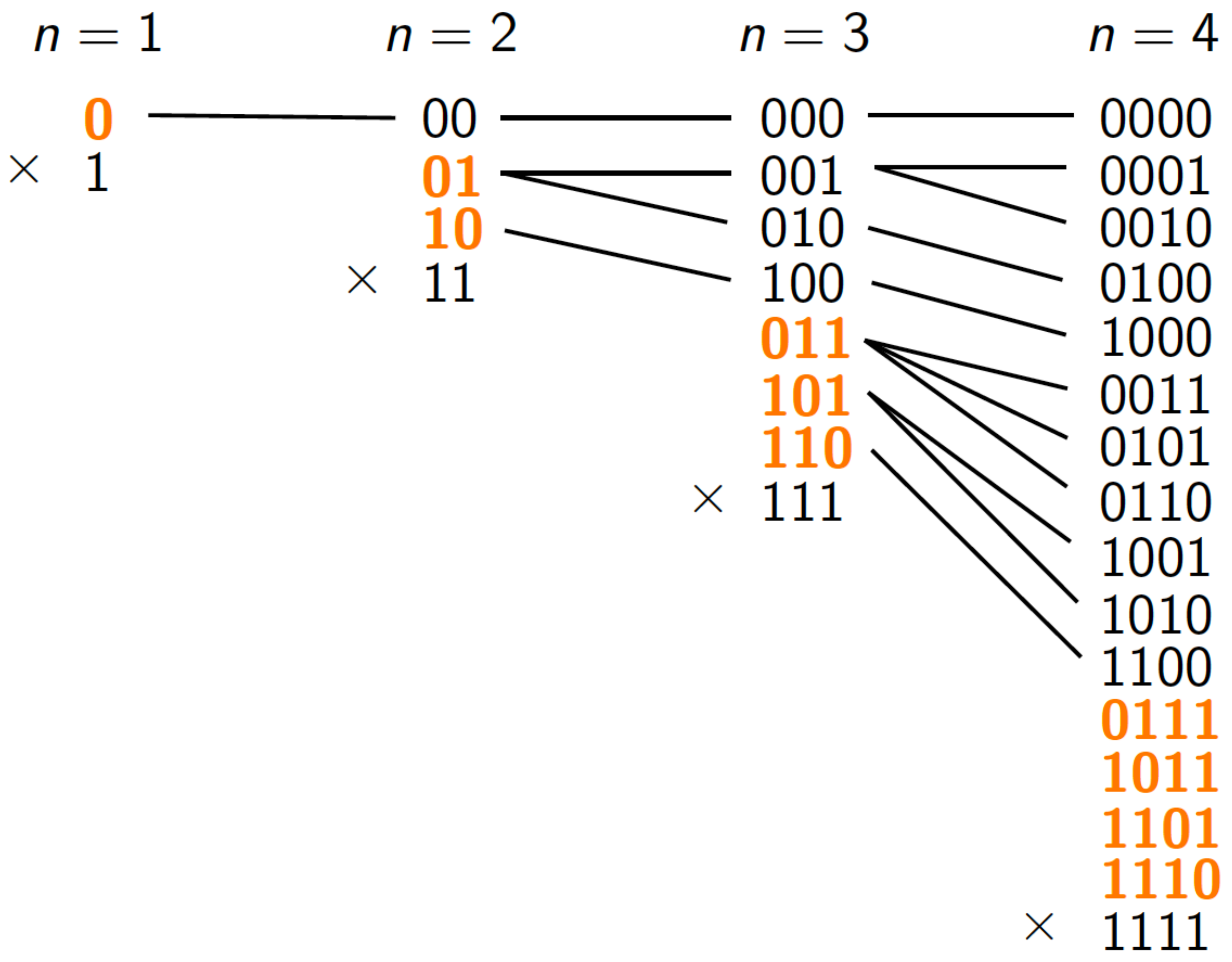
Conflict graph

Conflict graph

Observation: If we stop in some sequence, we also need to stop in other sequences with the same "prefix".

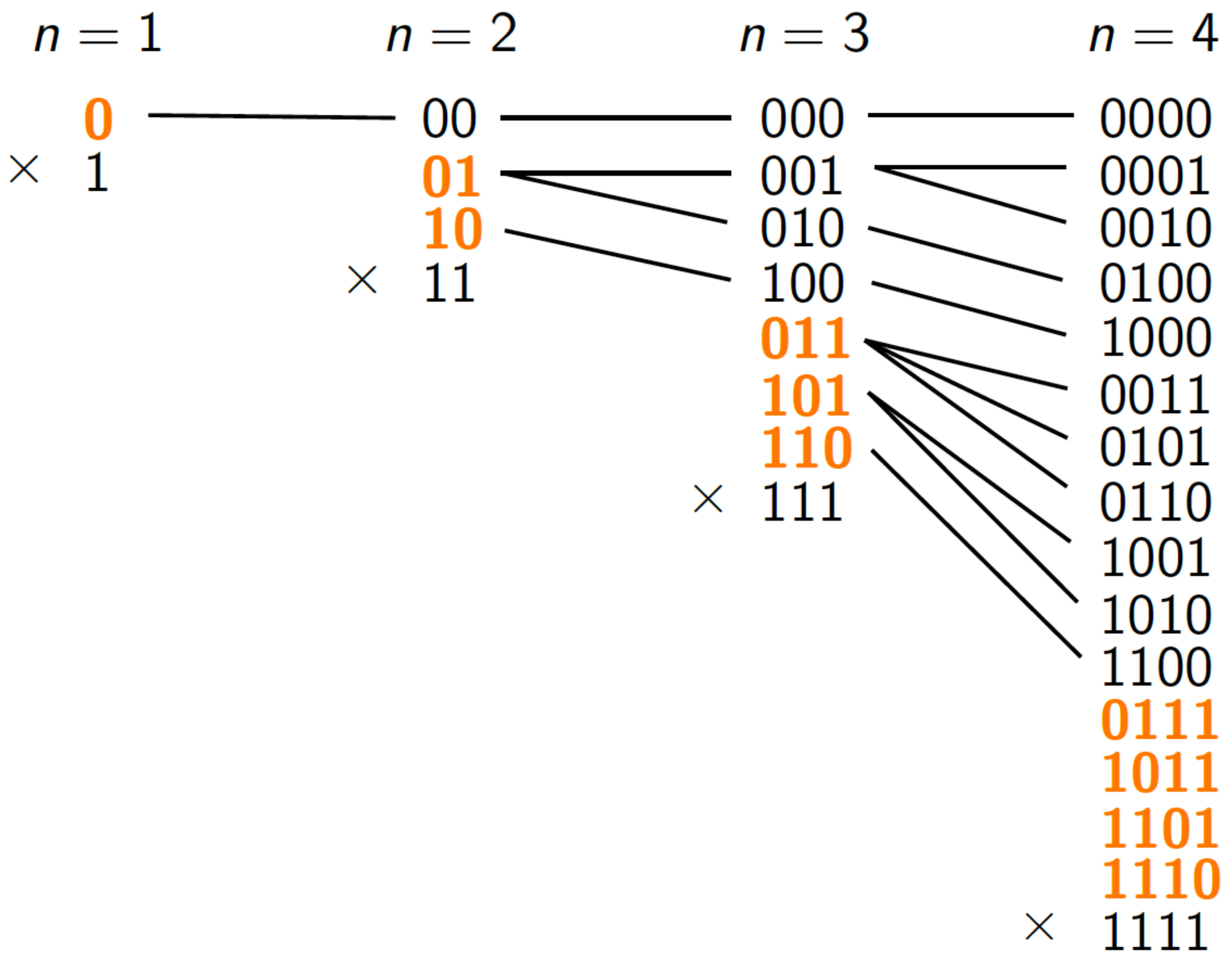
Conflict graph

Observation: If we stop in some sequence, we also need to stop in other sequences with the same “prefix”.



Conflict graph

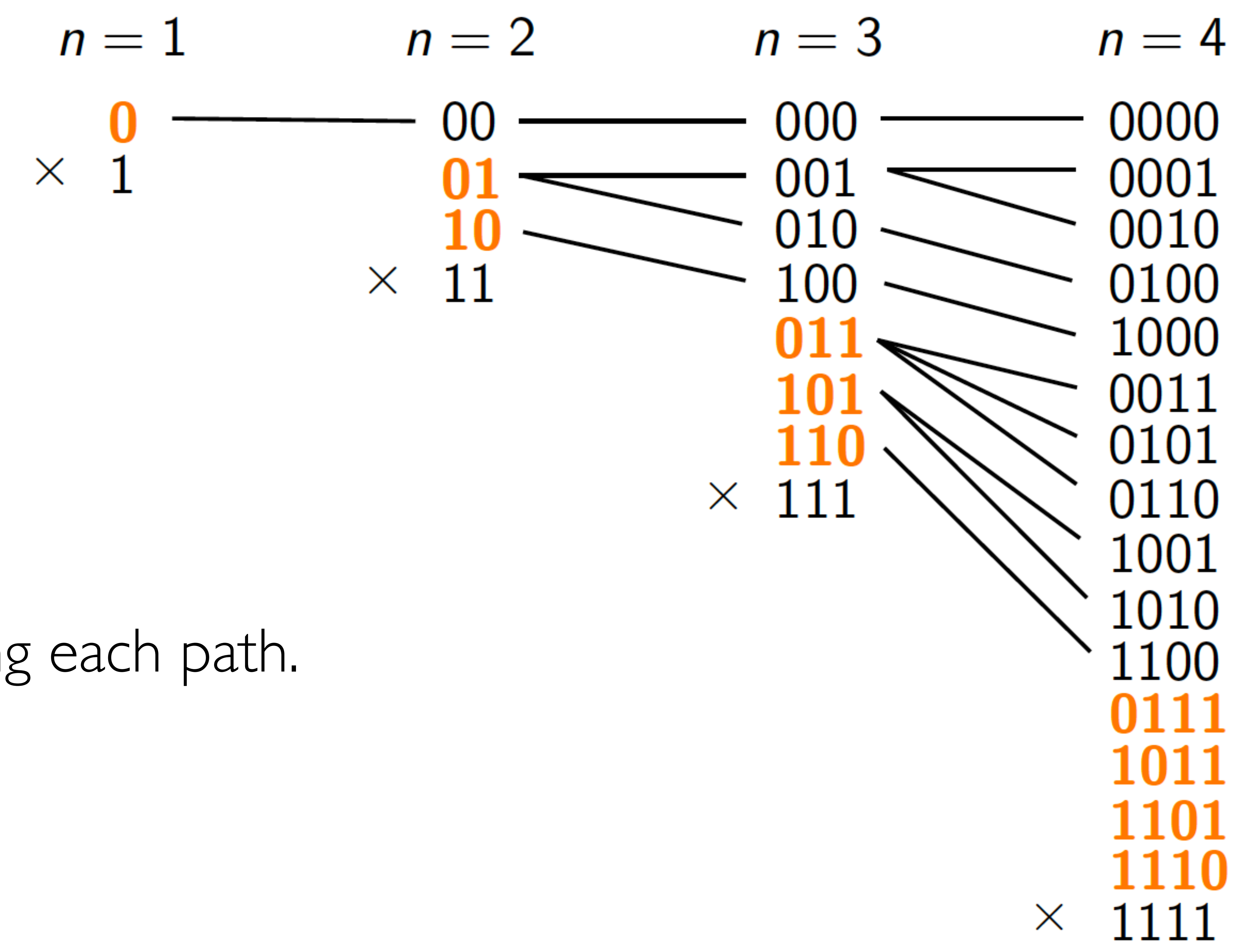
Observation: If we stop in some sequence, we also need to stop in other sequences with the same "prefix".



- Algorithm: Select sequences to win.

Conflict graph

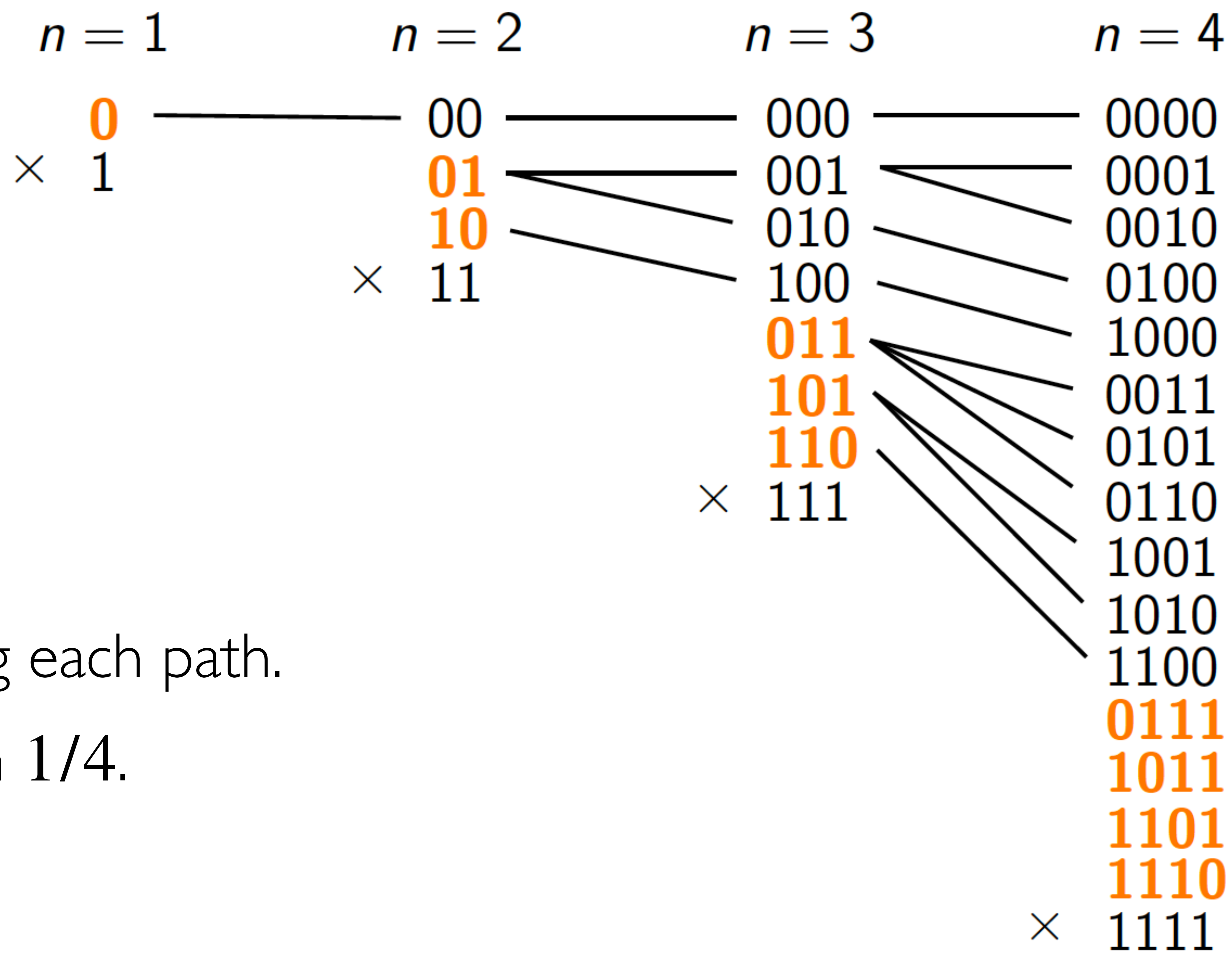
Observation: If we stop in some sequence, we also need to stop in other sequences with the same "prefix".



- Algorithm: Select sequences to win.
- Algo can select only one sequence along each path.

Conflict graph

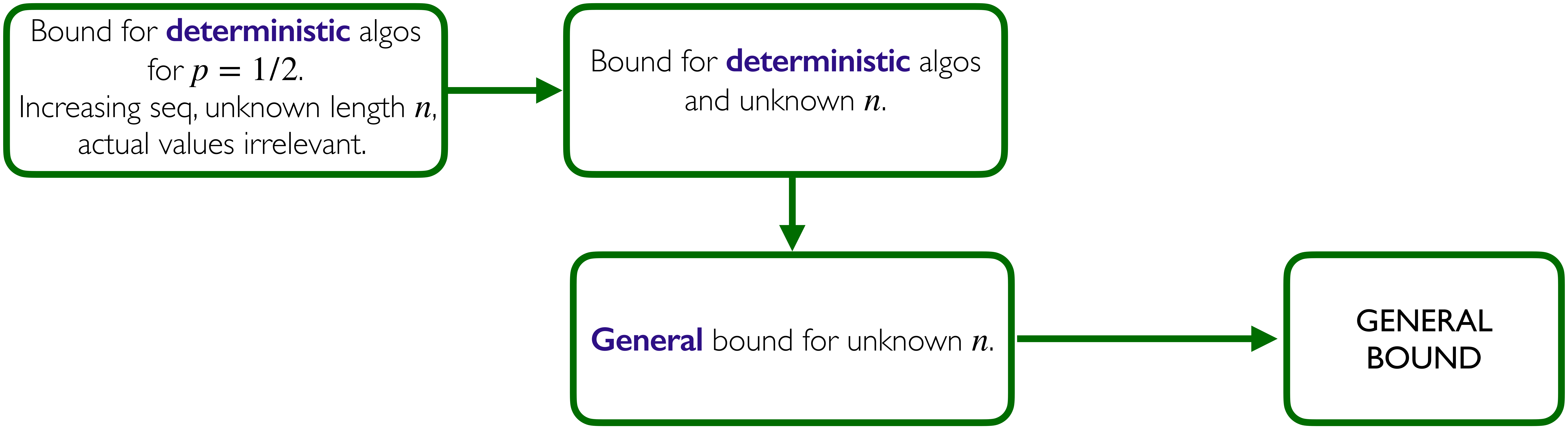
Observation: If we stop in some sequence, we also need to stop in other sequences with the same "prefix".



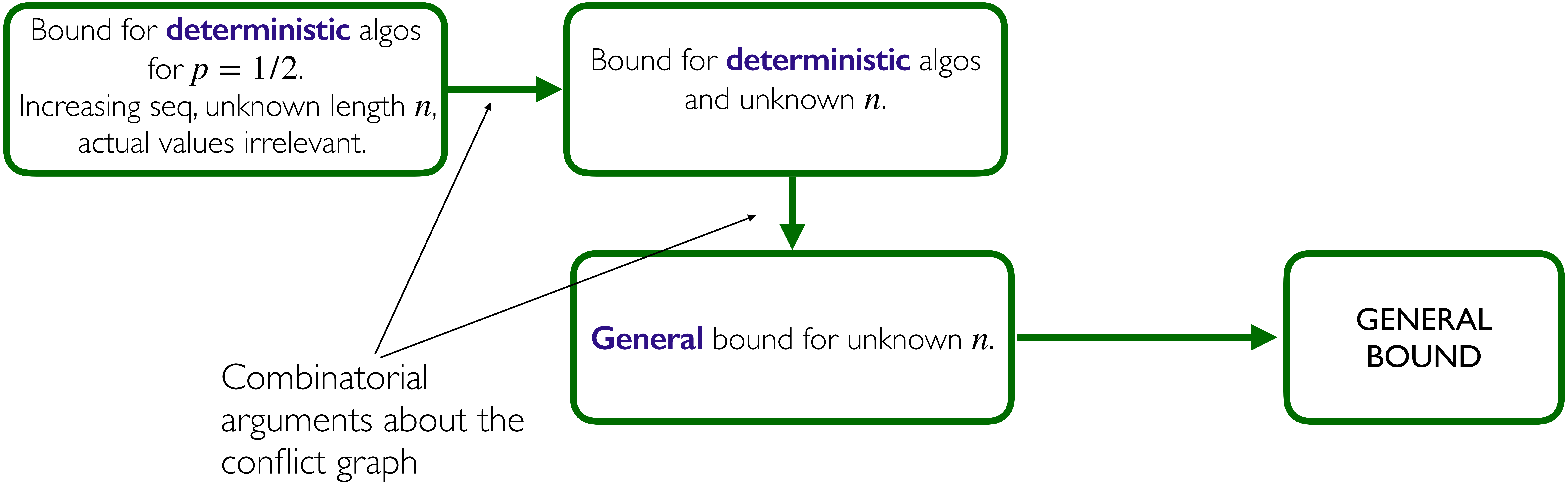
- Algorithm: Select sequences to win.
- Algo can select only one sequence along each path.
- Cannot keep selecting strictly more than $1/4$.

Overview of the proof

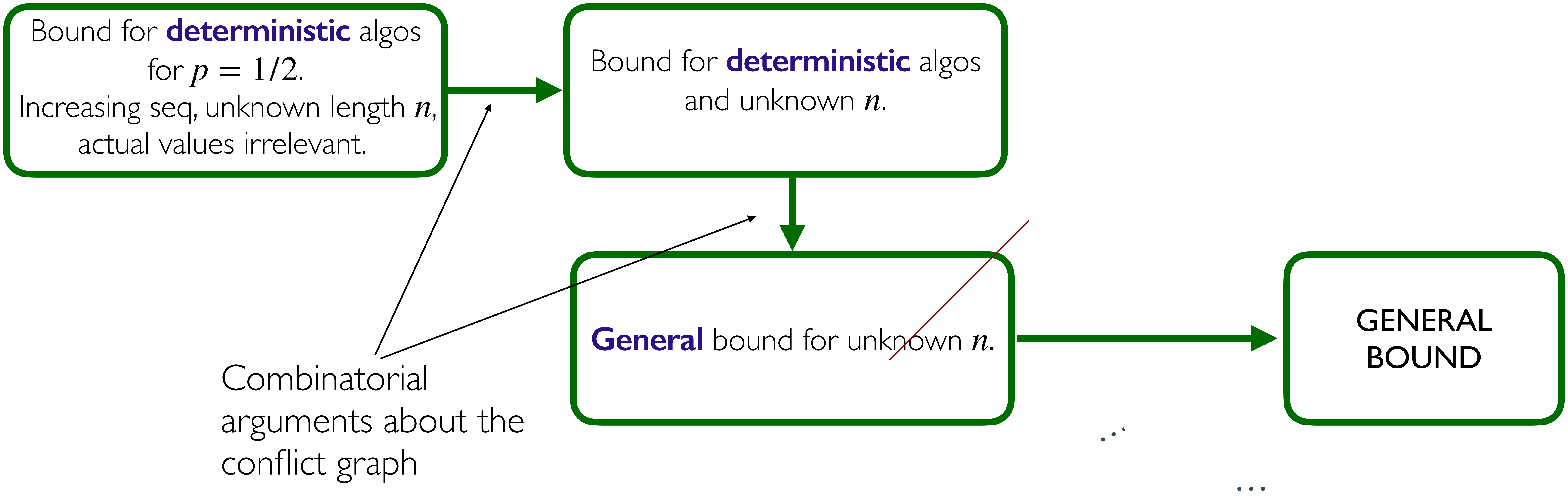
Overview of the proof



Overview of the proof



Overview of the proof



Results for ROS*p*

Results for ROSp

- New perspective:

Results for ROSp

- New perspective:
    1. Assign uniform random times $\tau_i \in [0,1]$.

Results for ROSp

- New perspective:
  1. Assign uniform random times $\tau_i \in [0,1]$.
  2. If $\tau_i < p$ let $v_i \in S$, otherwise let $v_i \in V$.

Results for ROSp

- New perspective:
  1. Assign uniform random times $\tau_i \in [0,1]$.
  2. If $\tau_i < p$ let $v_i \in S$, otherwise let $v_i \in V$.
  3. Values in V are revealed in order of τ_i values.

Results for ROSp

- New perspective:
  1. Assign uniform random times $\tau_i \in [0,1]$.
  2. If $\tau_i < p$ let $v_i \in S$, otherwise let $v_i \in V$.
  3. Values in V are revealed in order of τ_i values.
- ALG_t: Fix decreasing sequence of thresholds. Accept value v_i if largest in V so far and larger than the k-th largest sample if $\tau_i \in [t_k, t_{k+1}]$.

Results for ROSp

- New perspective:
  1. Assign uniform random times $\tau_i \in [0,1]$.
  2. If $\tau_i < p$ let $v_i \in S$, otherwise let $v_i \in V$.
  3. Values in V are revealed in order of τ_i values.
- ALG_t: Fix decreasing sequence of thresholds. Accept value v_i if largest in V so far and larger than the k-th largest sample if $\tau_i \in [t_k, t_{k+1}]$.
- The success guarantee of ALG_t is a separable and concave optimization problem. We can solve for the optimal sequence t^*.

Results for ROSp

- New perspective:
  1. Assign uniform random times $\tau_i \in [0,1]$.
  2. If $\tau_i < p$ let $v_i \in S$, otherwise let $v_i \in V$.
  3. Values in V are revealed in order of τ_i values.
- ALG_t: Fix decreasing sequence of thresholds. Accept value v_i if largest in V so far and larger than the k-th largest sample if $\tau_i \in [t_k, t_{k+1}]$.
- The success guarantee of ALG_t is a separable and concave optimization problem. We can solve for the optimal sequence t^*.
- Idea for proving best possible: Any optimal strategy can be seen as a decreasing sequence of thresholds.

Thank you for watching!