Prophets & Secretaries

DS592: Randomized Algorithms

Thodoris Tsilivis

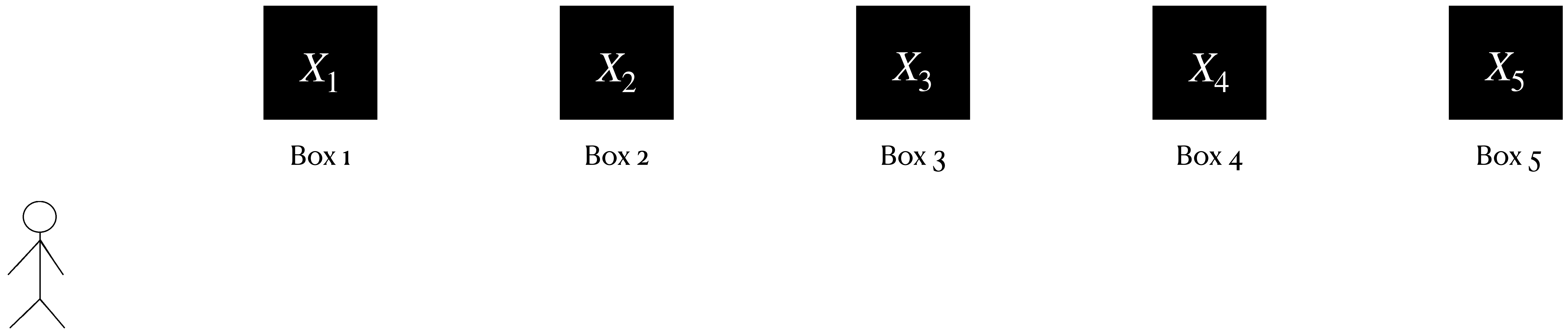
A simple online game

A sequence of boxes each containing a non-negative reward $X_i \geq 0$.

The **objective** is to pick the box with the highest reward i.e. $X_{max} = \max_{i \in [n]} X_i$.

Rules:

- Go through each box i and make an irrevocable decision:
    - A. Commit to box i with reward X_i (the game ends), or
    - B. Lose out on box i and proceed to the next box.

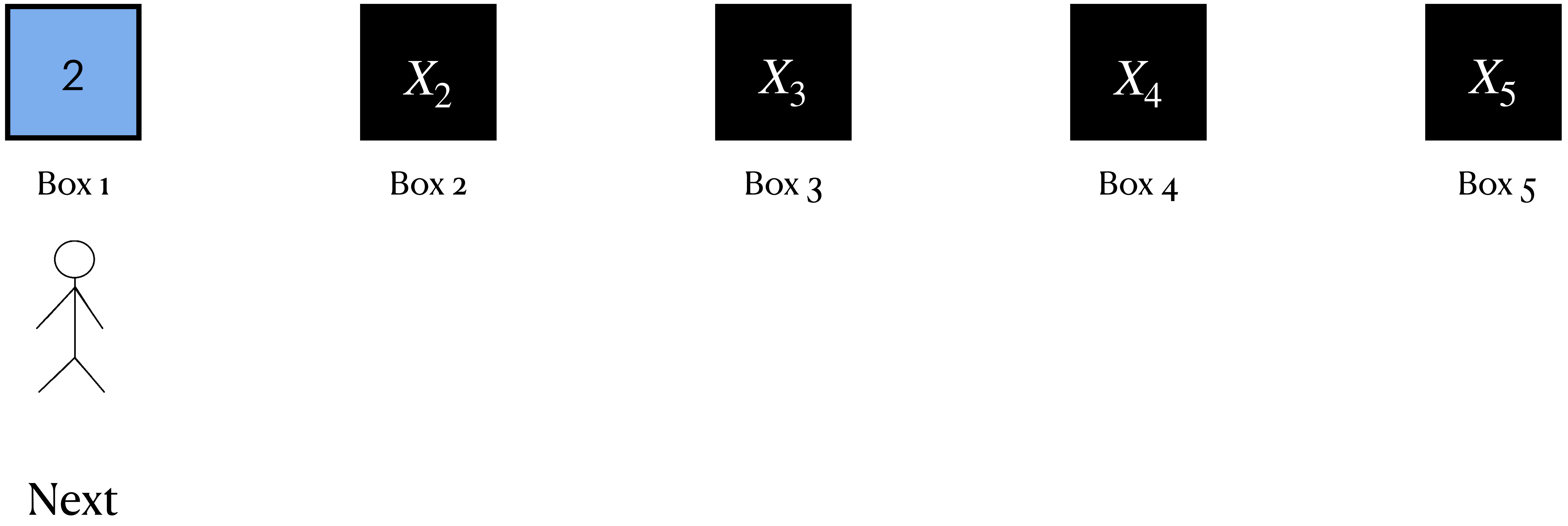


A simple online game

A sequence of boxes each containing a non-negative reward $X_i \geq 0$.
The **objective** is to pick the box with the highest reward i.e. $X_{max} = \max_{i \in [n]} X_i$.

Rules:

- Go through each box i and make an irrevocable decision:
    A. Commit to box i with reward X_i (the game ends), or
    B. Lose out on box i and proceed to the next box.

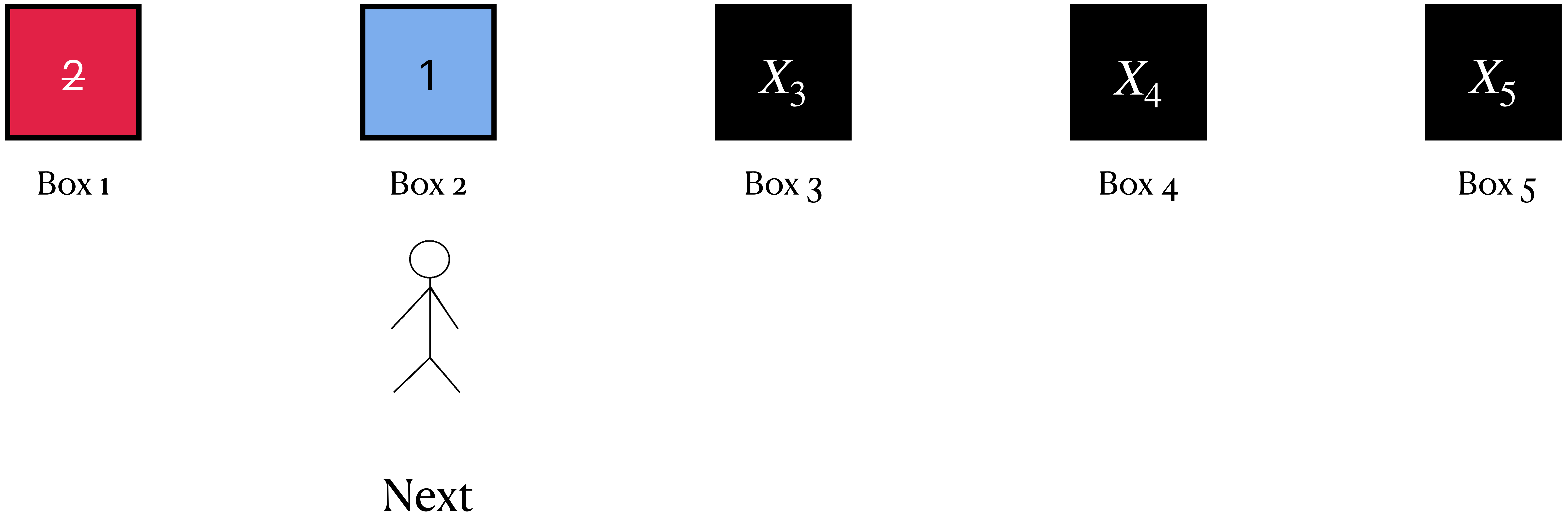


A simple online game

A sequence of boxes each containing a non-negative reward $X_i \geq 0$.
The **objective** is to pick the box with the highest reward i.e. $X_{max} = \max_{i \in [n]} X_i$.

Rules:

- Go through each box i and make an irrevocable decision:
    A. Commit to box i with reward X_i (the game ends), or
    B. Lose out on box i and proceed to the next box.

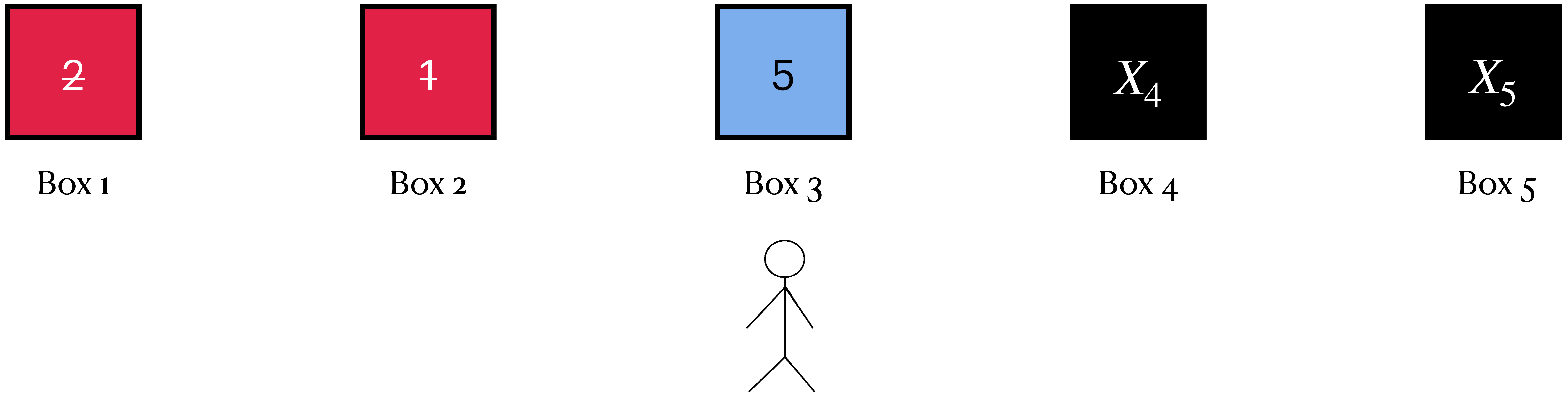


A simple online game

A sequence of boxes each containing a non-negative reward $X_i \geq 0$.
The **objective** is to pick the box with the highest reward i.e. $X_{max} = \max_{i \in [n]} X_i$.

Rules:

- Go through each box i and make an irrevocable decision:
  - A. Commit to box i with reward X_i (the game ends), or
  - B. Lose out on box i and proceed to the next box.



Commit

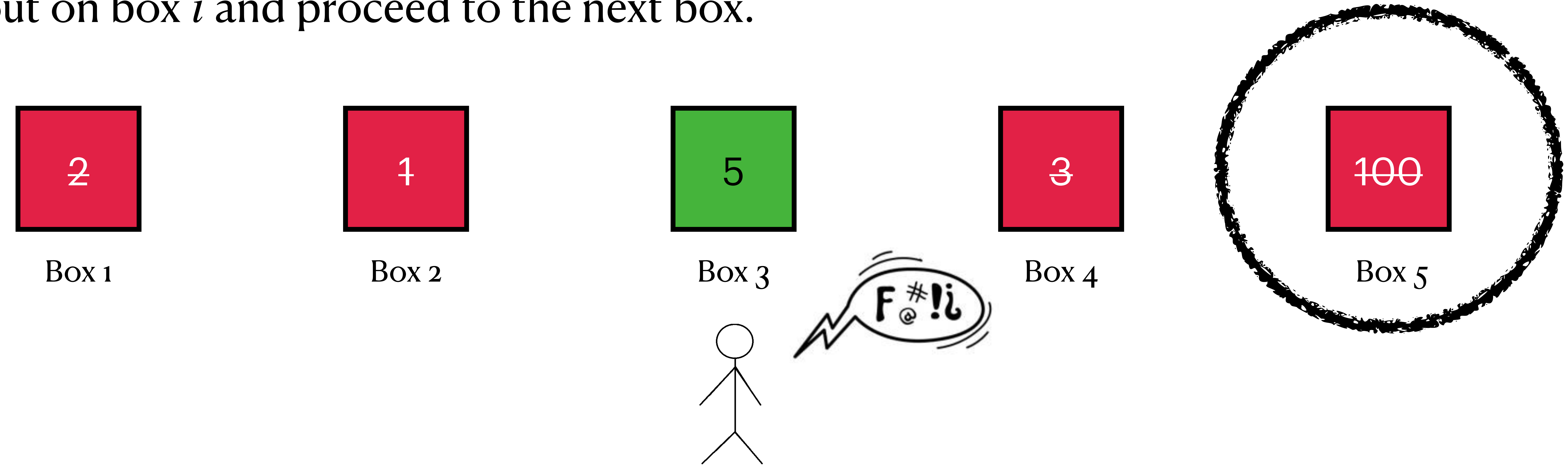
A simple online game

A sequence of boxes each containing a non-negative reward $X_i \geq 0$.

The **objective** is to pick the box with the highest reward i.e. $X_{max} = \max_{i \in [n]} X_i$.

Rules:

- Go through each box i and make an irrevocable decision:
  A. Commit to box i with reward X_i (the game ends), or
  B. Lose out on box i and proceed to the next box.



Difficult instances

	Box 1	Box 2
Instance 1	1	1000
Instance 2	1	$\frac{1}{1000}$

- Deterministic algorithms can be arbitrarily bad.
- Randomized algorithms can only do as well as random guessing.

Why is this problem difficult

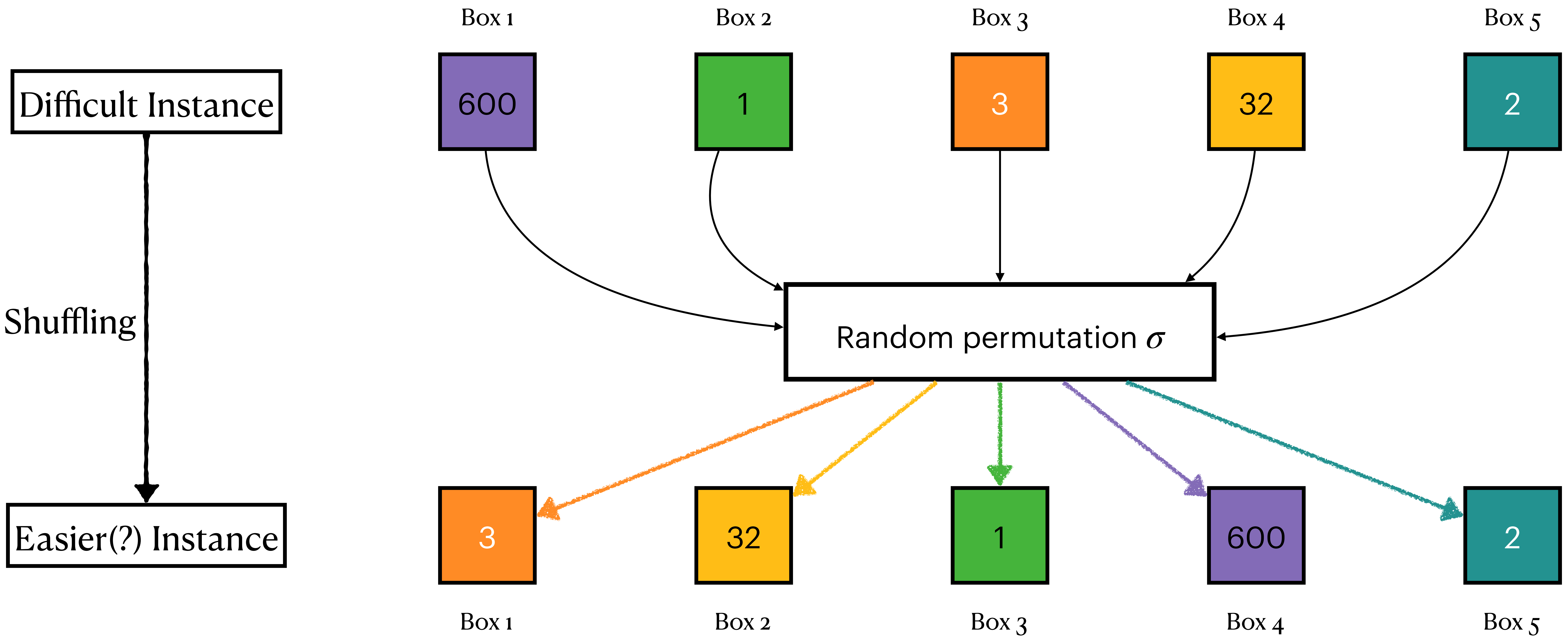
Two components:

1. The ordering.
2. The unknown range of the rewards.

Secretary Problem	**Prophet Inequality**
Assume a relaxed ordering.	Assume structure on the range of the rewards.

Secretary problem - 1950s

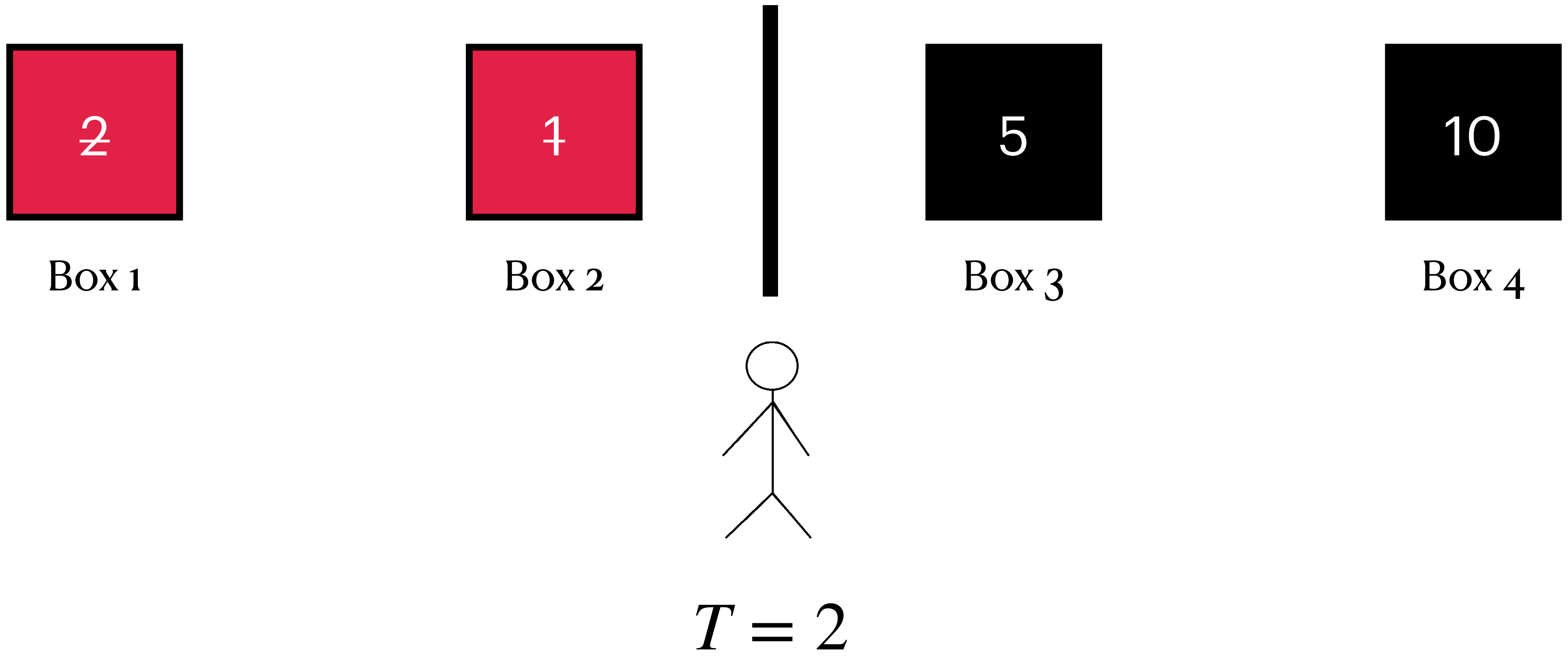
-Makes the problem easier by randomly shuffling the boxes!



A constant approximation

The 50-50 Algorithm

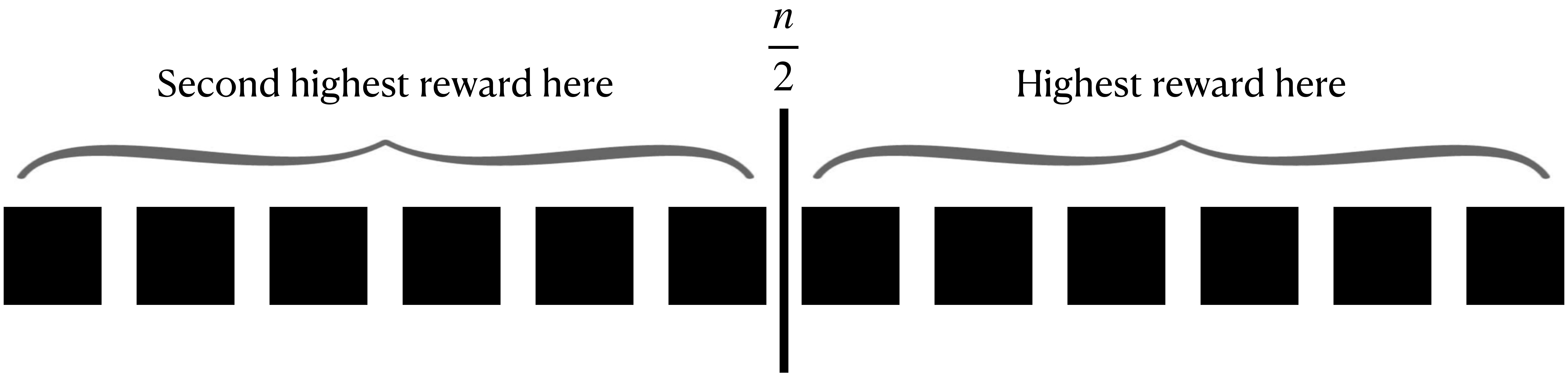
1. Observe (without committing) all boxes up to index $\frac{n}{2}$.
2. Set threshold $T = \max_{i \in [1, \frac{n}{2}]} X_i$.
3. Out of the remaining boxes commit to the first box that has $X_i > T$.



What does this algorithm achieve?

Easily **bound** the **probability** of selecting the **highest reward**.

A **sufficient condition:**



Guarantee : $\mathbb{E}[Reward] \geq \frac{X_{max}}{4}$

Can we do better?

- **Observe first** m boxes.

- Compute the **probability** of getting the **highest reward** as a function of m.

- Optimize over m. Turns out that $m^* \approx \lceil \frac{n}{e} \rceil$ which yields:

$$\mathbb{E}[Reward] \geq \frac{X_{max}}{e}$$

Prophet Inequality - 1978

- Makes the problem easier by assuming that **rewards** are drawn **independently** from **distributions**!
- Our objective is to have a guarantee comparable to $\mathbb{E}[\max_{i \in [n]} X_i] = \mathbb{E}[X_{max}]$.

$X_1 \sim D_1$ $X_2 \sim D_2$ $X_3 \sim D_3$ $X_4 \sim D_4$ $X_5 \sim D_5$

The optimal algorithm - KW12

1. Set threshold $T = \frac{1}{2}\mathbb{E}[\max_{i \in [n]} X_i]$.
2. Accept the first box that has $X_i > T$.

Why does it work?

- Balances out probability of accepting **at least one** box **and** also that the accepted box has **high enough reward**.
- This algorithm yields:

$$\mathbb{E}[Reward] \geq \frac{1}{2}\mathbb{E}[\max_{i \in [n]} X_i]$$

Prophet Inequality beyond distributions

- Access to distributions may be **unrealistic**.
- Computing $T = \frac{1}{2}\mathbb{E}[\max_{i \in [n]} X_i]$ can be **computationally challenging** even for simple distributions.

- Can we relax these requirements, without severely deteriorating the performance of the proposed algorithm?
- **Yes**, lets see how.

Prophet Inequality - Sample access - RWW18

Access to 1 sample $X'_1, X'_2, \ldots, X'_n$ from each of the distributions $D_1, D_2, \ldots, D_n$.

Single Sample Algorithm

1. Set threshold $T = \max_{i \in [n]} X'_i$.
2. Accept the first box i that has rewards $X_i > T$.

We will prove that the **Single Sample Algorithm** guarantees:

$$\mathbb{E}[Reward] \geq \frac{1}{2}\mathbb{E}[\max_{i \in [n]} X_i]$$

A different view

Instead of thinking of samples X_i' and rewards X_i as different entities, we will use a Deferred Decision procedure to group both types of draws together.

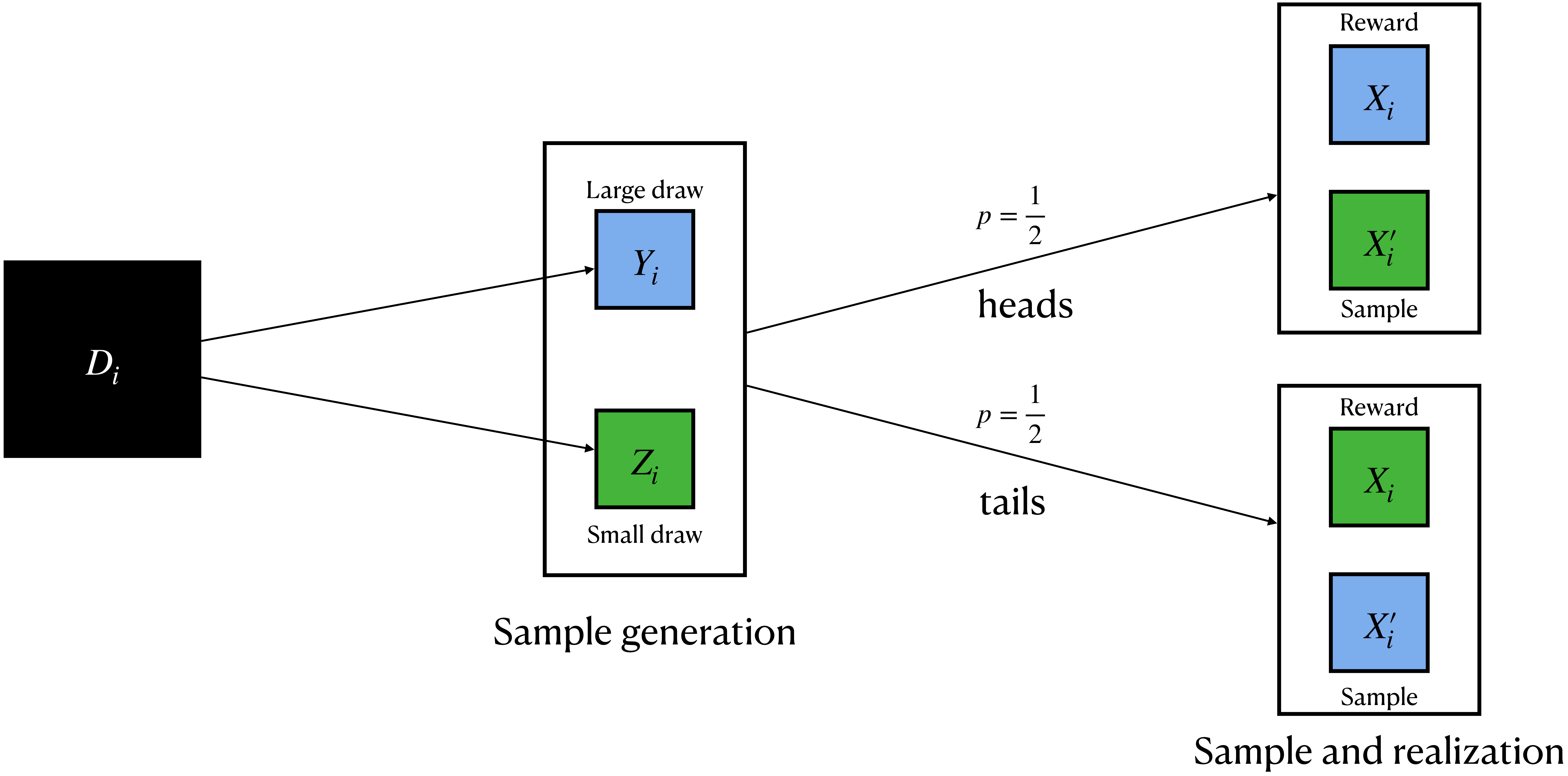
Principle of Deferred-Decisions:

1. Draw **two samples** from each of $D_1, \ldots, D_n$ **independently**.

2. Denote the **large draw** of D_i as Y_i and the **small draw** of D_i as Z_i (**note** that $Y_i > Z_i$).

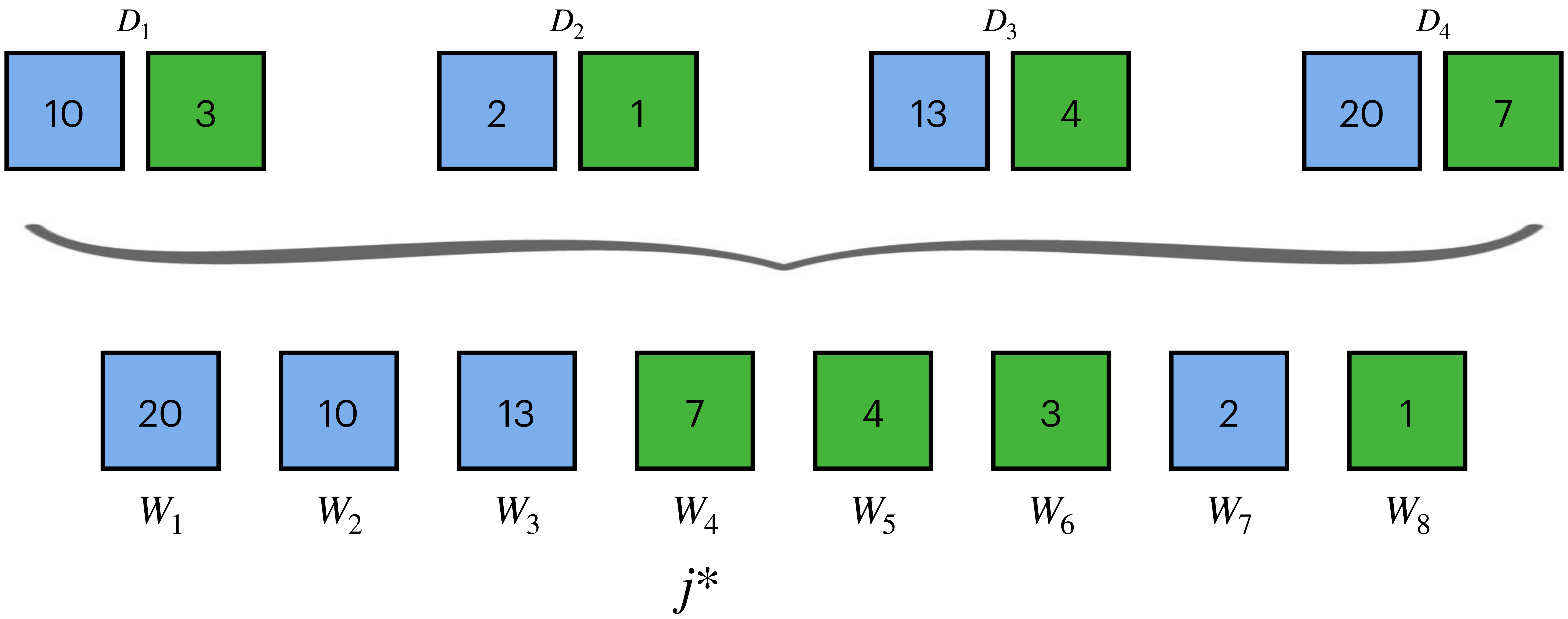
3. **Flip** a **fair coin independently** for each D_i . If coin i is heads, set $X_i = Y_i$ and $X_i' = Z_i$. Otherwise, set $X_i = Z_i$ and $X_i' = Y_i$.

Deferred Decision

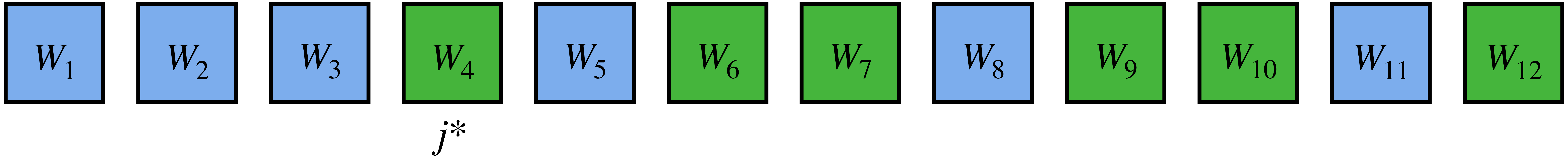


Analysis setup

- Sort all draws $Y_1, \ldots, Y_n, Z_1, \ldots, Z_n$ in **descending order** and relabel them as $W_1, \ldots, W_{2n}$.
- Define j^* to be the smallest index that comes from a Z_i.



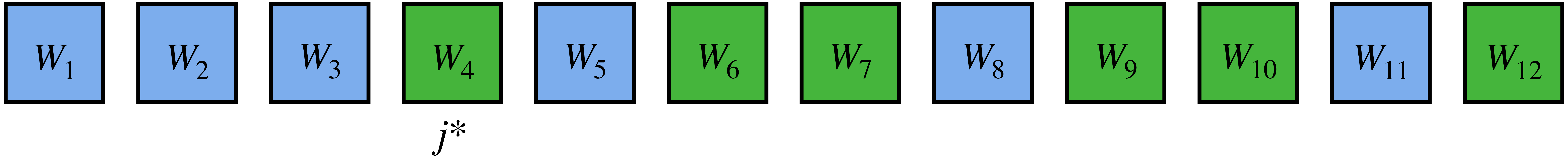
Expected maximum (Prophet's payoff)



$$\max_{i \in n} X_i = \begin{cases} W_j & \text{w.p. } \frac{1}{2^j} \text{ for } j < j^*, \\ W_{j^*} & \text{w.p. } \frac{1}{2^{j^*-1}} \end{cases}$$

$$\mathbb{E}[\max_{i \in n} X_i] = \sum_{j=1}^{j^*-1} \frac{W_j}{2^j} + \frac{W_{j^*}}{2^{j^*-1}}$$

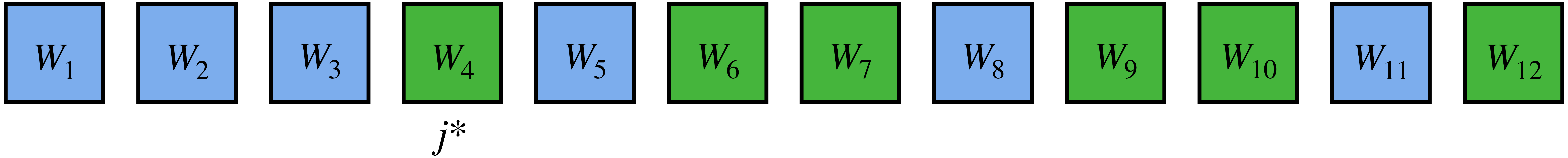
Expected algorithms reward



For $j < j^* - 1$:

- The **probability** that "all indices up to j are rewards and index $j + 1$ is a sample" is **equal** to $\frac{1}{2^{j+1}}$.
- In this case the algorithm **yields reward** at least W_j.

Expected algorithms reward



Special case $j^* - 1$:

- The **probability** that "all indices up to $j^* - 1$ are rewards and index j^* is a sample" is **equal** to $\frac{1}{2^{j^*-1}}$.
- In this case the algorithm **yields reward** at least W_{j^*-1}.

Expected algorithms reward

$$\begin{aligned}
\mathbb{E}[Reward] &\geq \sum_{j=1}^{j^*-2} \frac{W_j}{2^{j+1}} + \frac{W_{j^*-1}}{2^{j^*-1}} \\
&= \sum_{j=1}^{j^*-2} \frac{W_j}{2^{j+1}} + \frac{1}{4} \cdot \frac{W_{j^*-1}}{2^{j^*-1}} + \frac{3}{4} \cdot \frac{W_{j^*-1}}{2^{j^*-1}} \\
&\geq \sum_{j=1}^{j^*-1} \frac{W_j}{2^{j+1}} + \frac{W_{j^*}}{2^{j^*}} \\
&= \frac{1}{2}\left(\sum_{j=1}^{j^*-1} \frac{W_j}{2^j} + \frac{W_{j^*}}{2^{j^*-1}} \right) \\
&= \frac{1}{2}\mathbb{E}[X_{max}]
\end{aligned}$$

Conclusions

- The **secretary problem** and the **prophet inequality** setting are two sides of the same coin.
- Everything boils down to **setting** an appropriate **threshold**.
- **Tight** algorithms exist for both problems.

- We can solve the **prophet inequality** setting **optimally** using only **samples**!

Thank you!