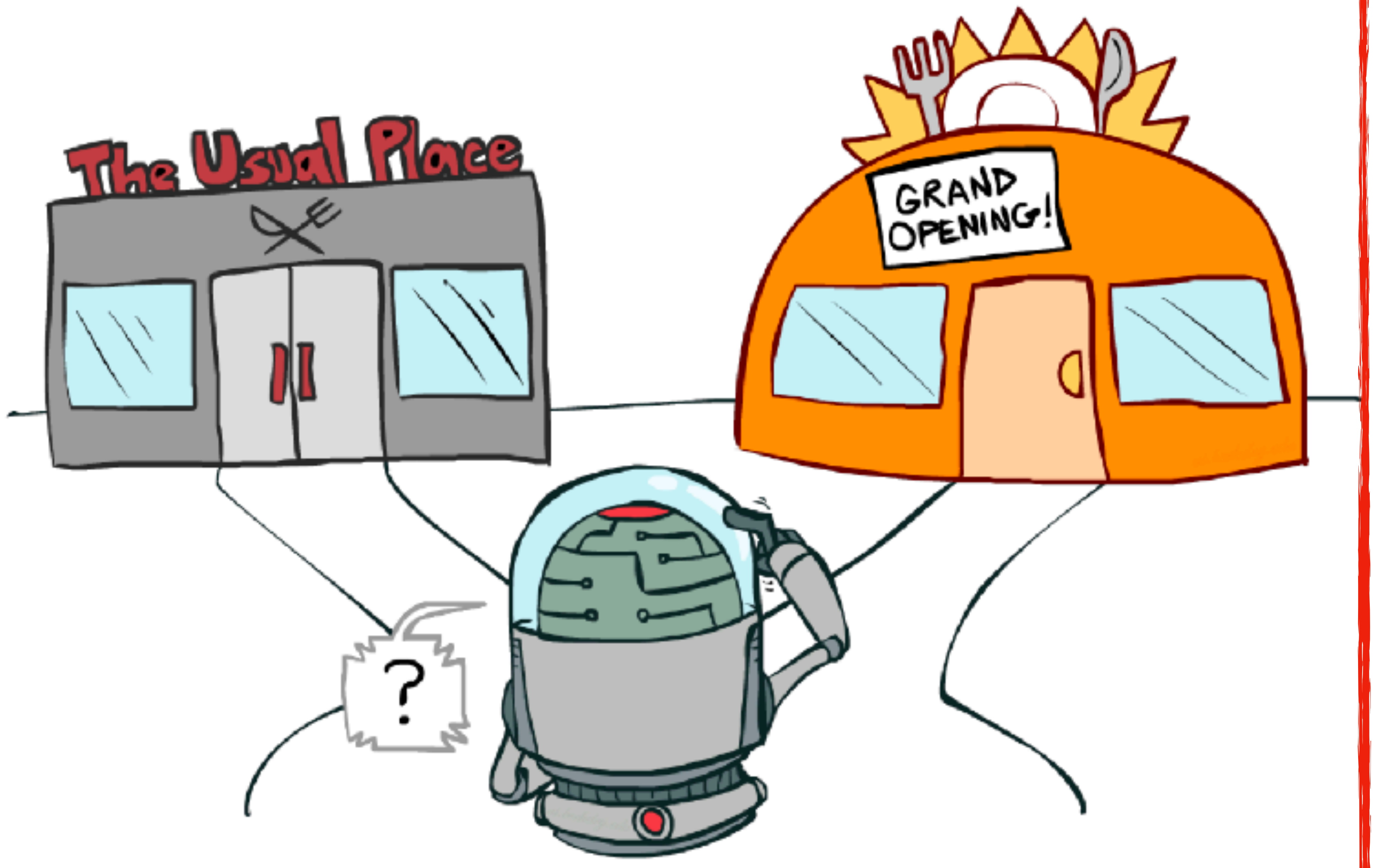
Dealing with Uncertainty

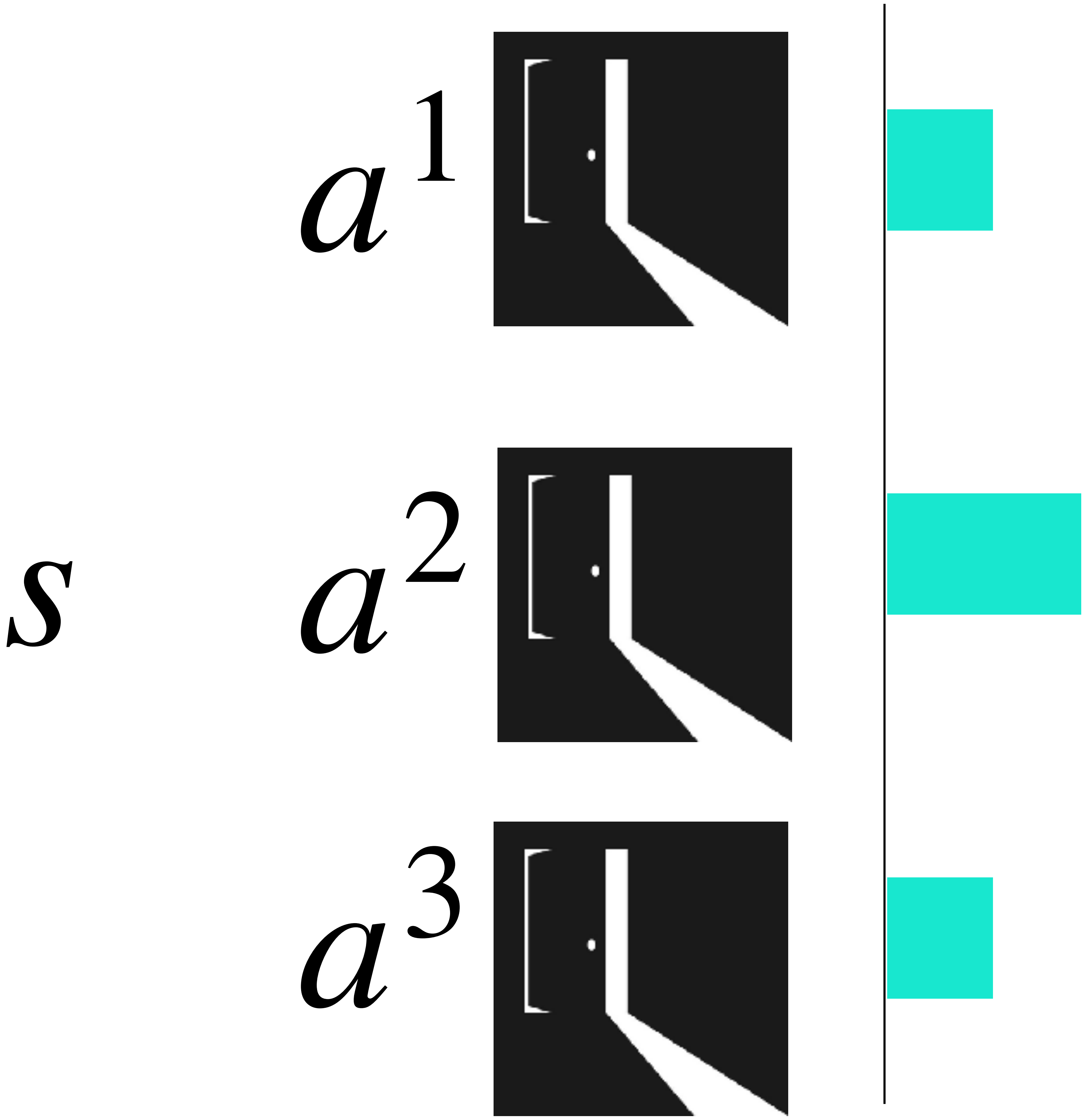
Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Two Ingredients of RL



Exploration Exploitation



Estimate Values $Q(s, a)$

Uncertainty

Types of uncertainty

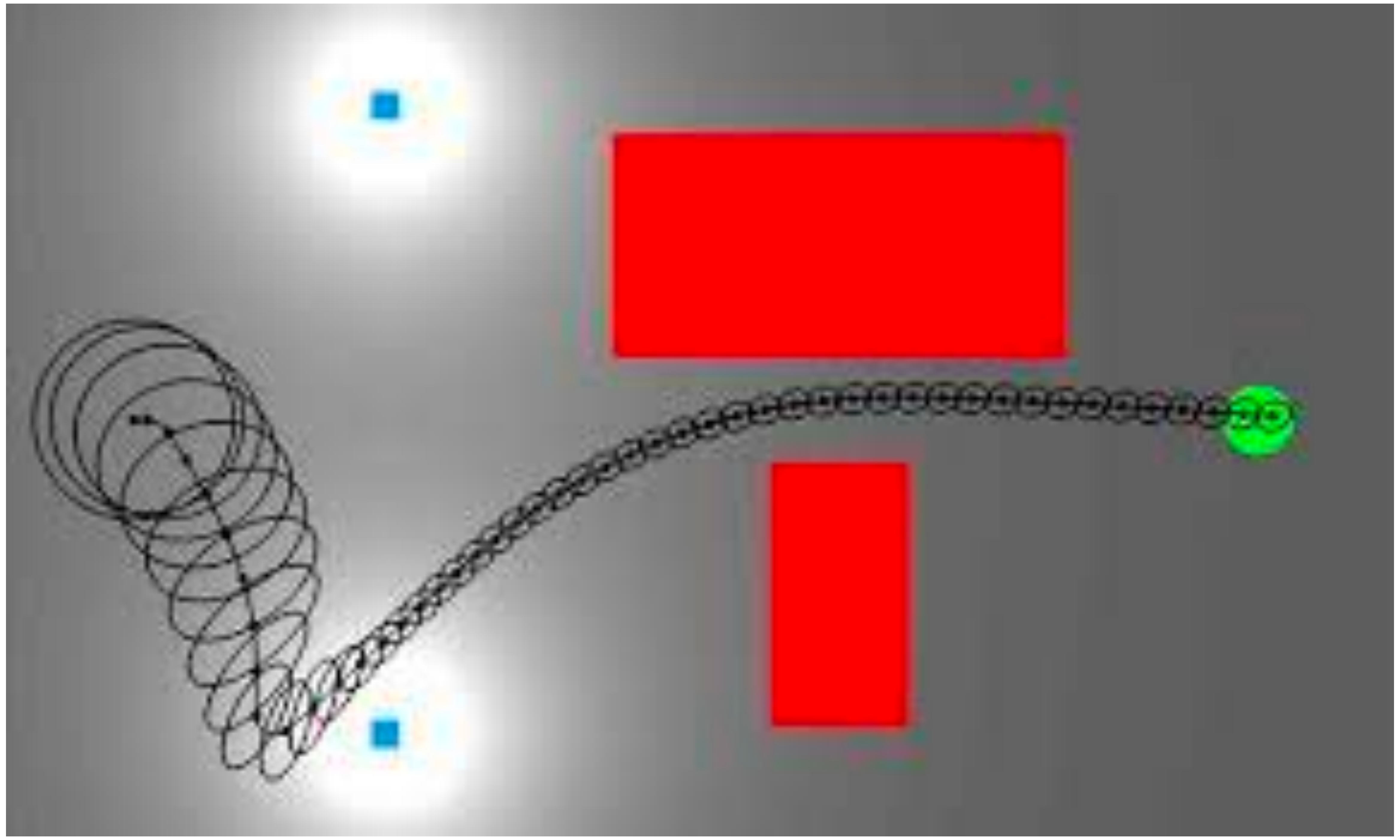
Aleatoric uncertainty

(Can't change this uncertainty)

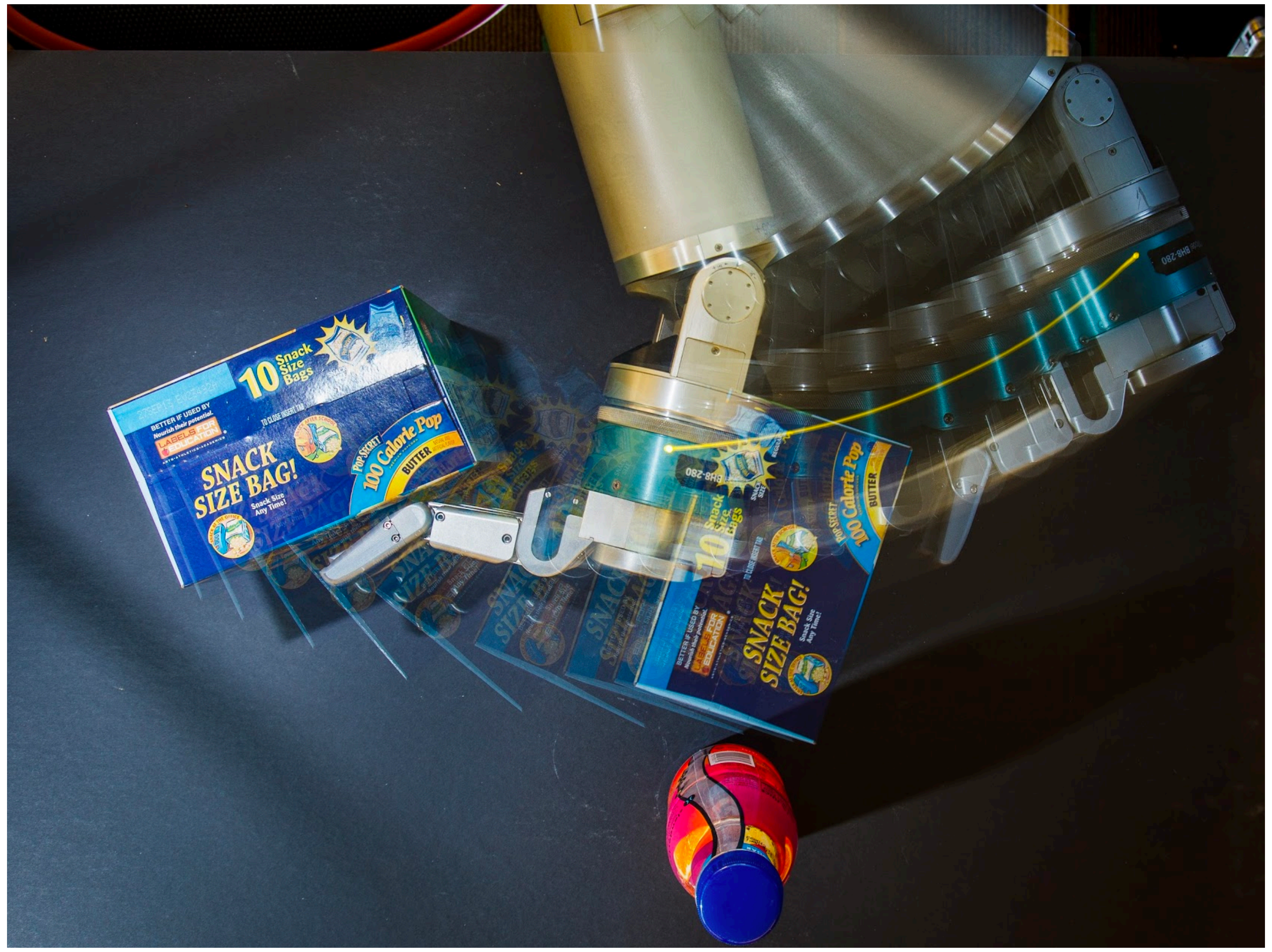
Epistemic uncertainty

(Acquire knowledge!)

Epistemic Uncertainty



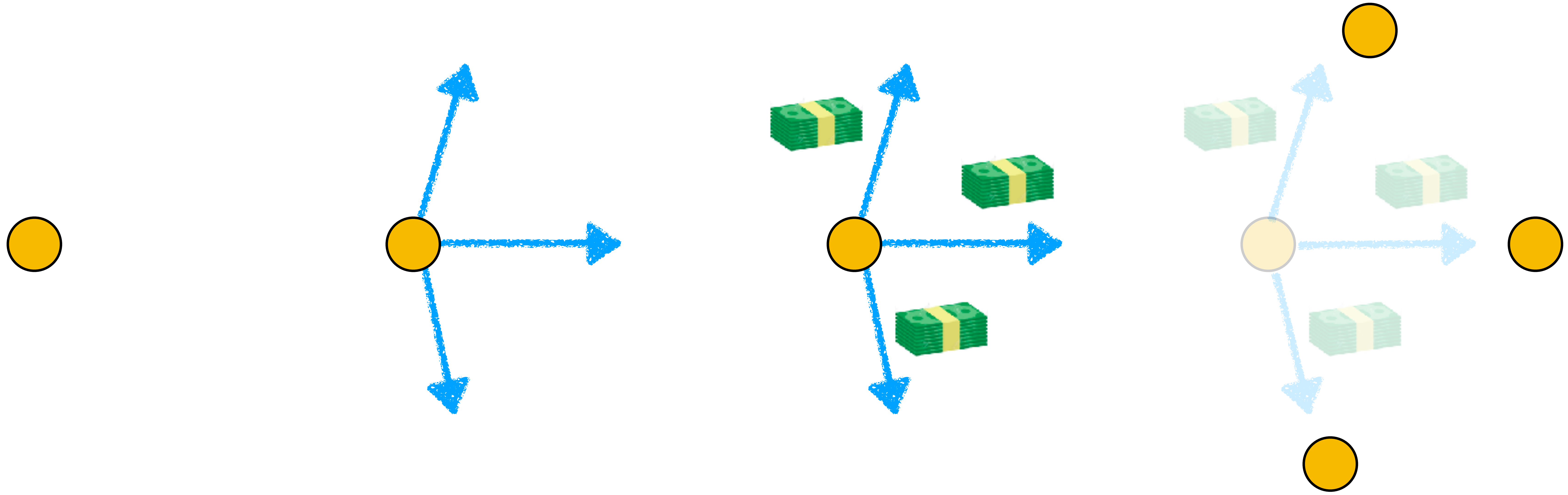
Uncertain about state



Uncertain about transitions

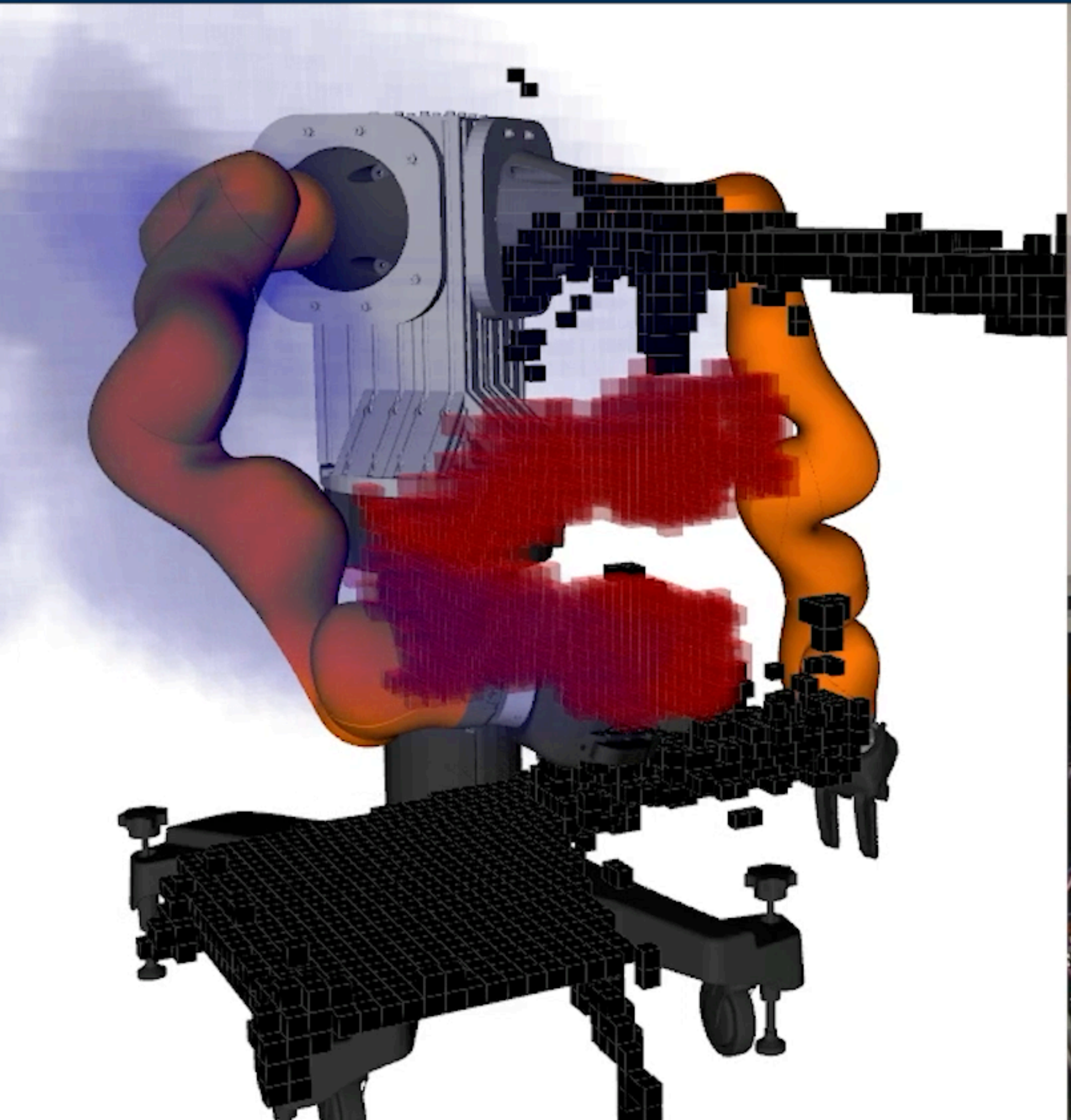
Can be uncertain about any of these things!

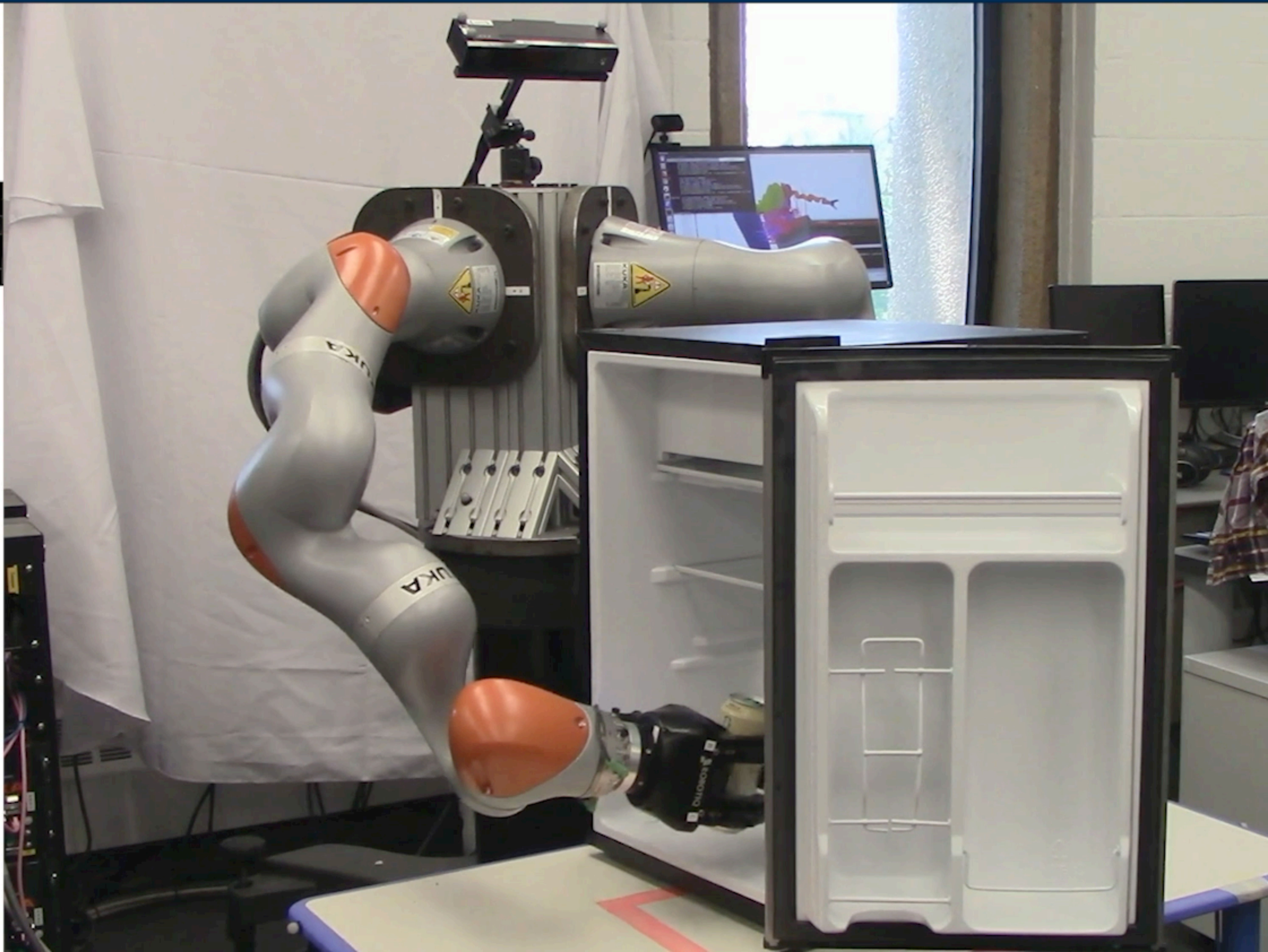
$$< S , A , C , \mathcal{T} >$$



Activity!

Victor placing item in refrigerator: a Blindfolded Traveler's Problem





Think-Pair-Share

Think (30 sec): Define the MDP <S,A,C,T> for the robot. Which term are you uncertain about?

Pair: Find a partner

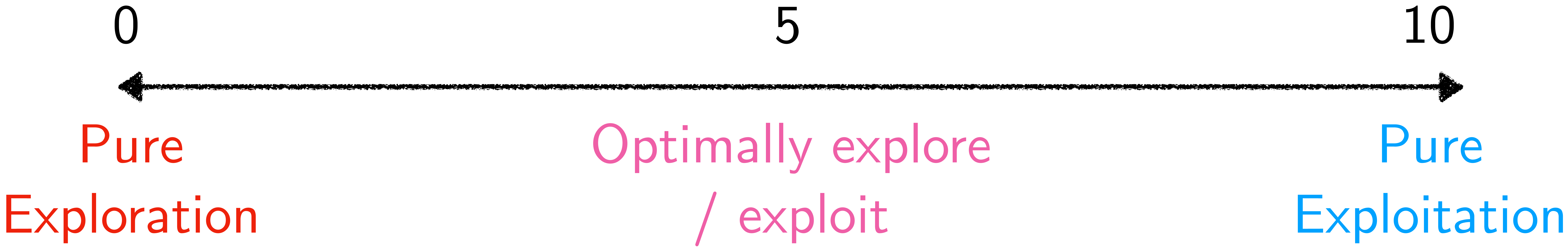
Share (45 sec): Partners exchange ideas

Victor placing item in refrigerator:
a Blindfolded Traveler's Problem

What do we want to do about uncertainty?

Pure Exploration	Optimally explore / exploit	Pure Exploitation
Collapse uncertainty as quickly as possible	Take information gathering steps, but be robust along the way	Be robust against uncertainty
20 questions	Life!	UAV flying in wind

Categorize the following robot applications!



0

5

10

Pure Exploration

Optimally explore / exploit

Pure Exploitation

Self-driving through an intersection

Assistive manipulation via shared autonomy

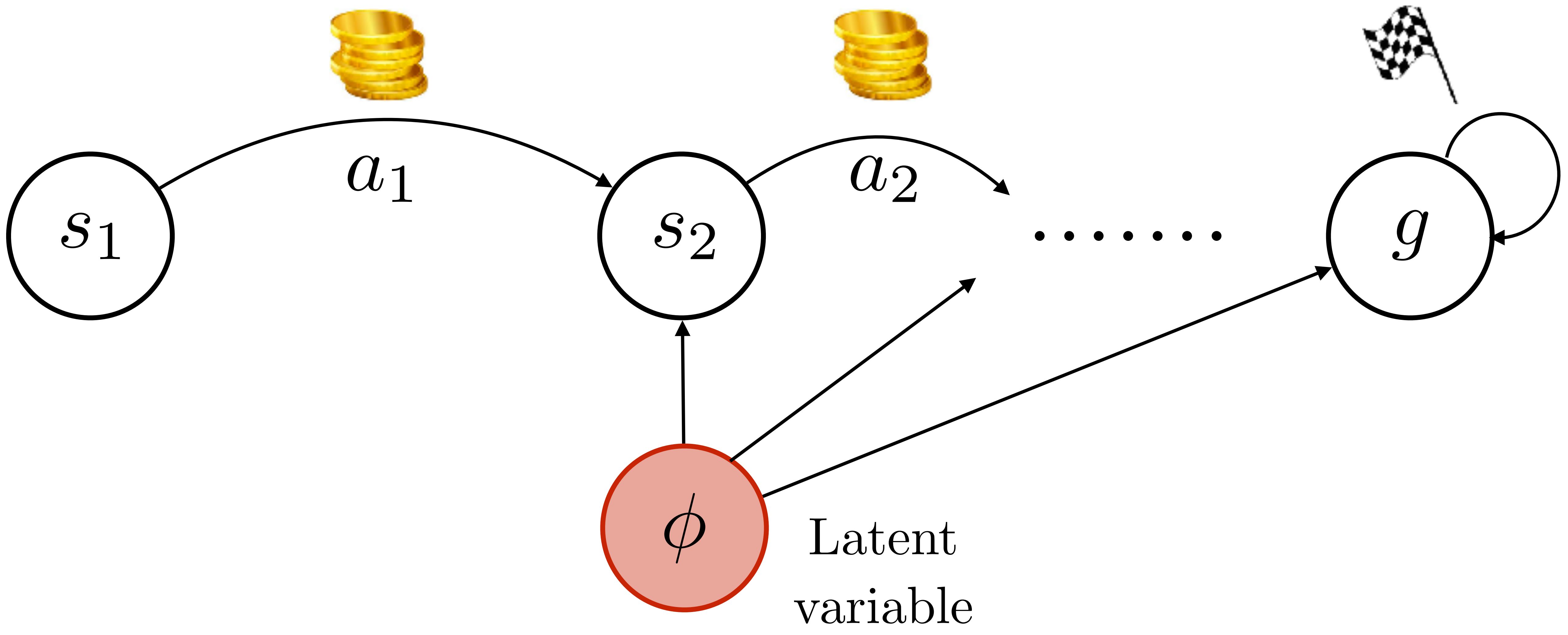
UAV autonomously mapping a building

Grasping an object on the top-shelf

But what is the *optimal* exploration-exploitation algorithm?

Belief Space Planning

Can frame optimal exploration / exploitation as Belief Space Planning



State: $s \in \mathcal{S}$

(fixed latent variable) $\phi \in \Phi$

Transition: $P(s'|s, a, \phi)$

Prior: $P(\phi)$

Bayes Optimality:

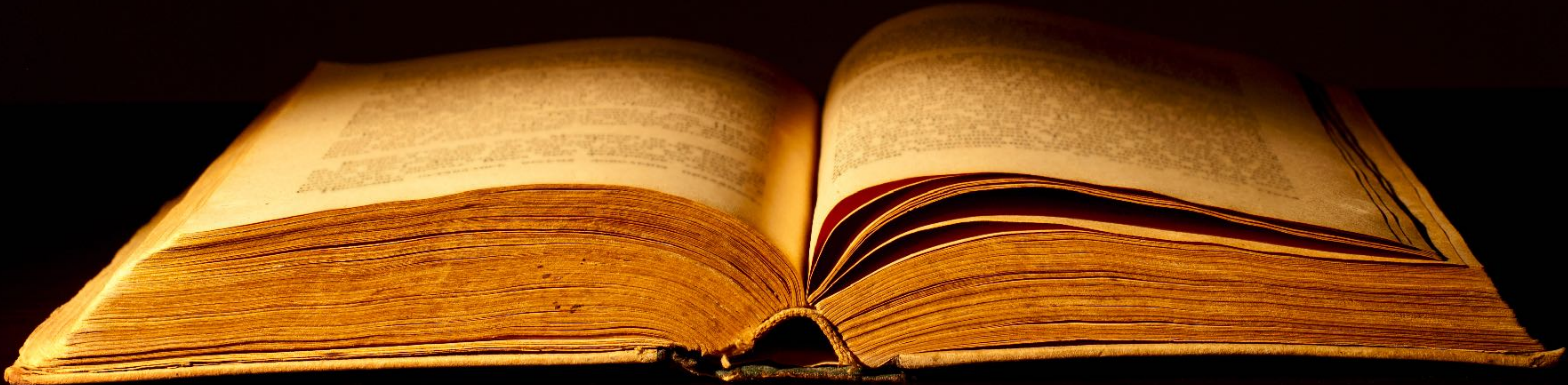
The Holy Grail

GAME OVER!

Belief Space Planning is NP-Hard at best, undecidable at worst

Need to relax our problem!

A Tale of Relaxations



Optimism in the Face of Uncertainty (OFU)

We already know an OFU algorithm!

Can you spot it?

Recap: LazySP!

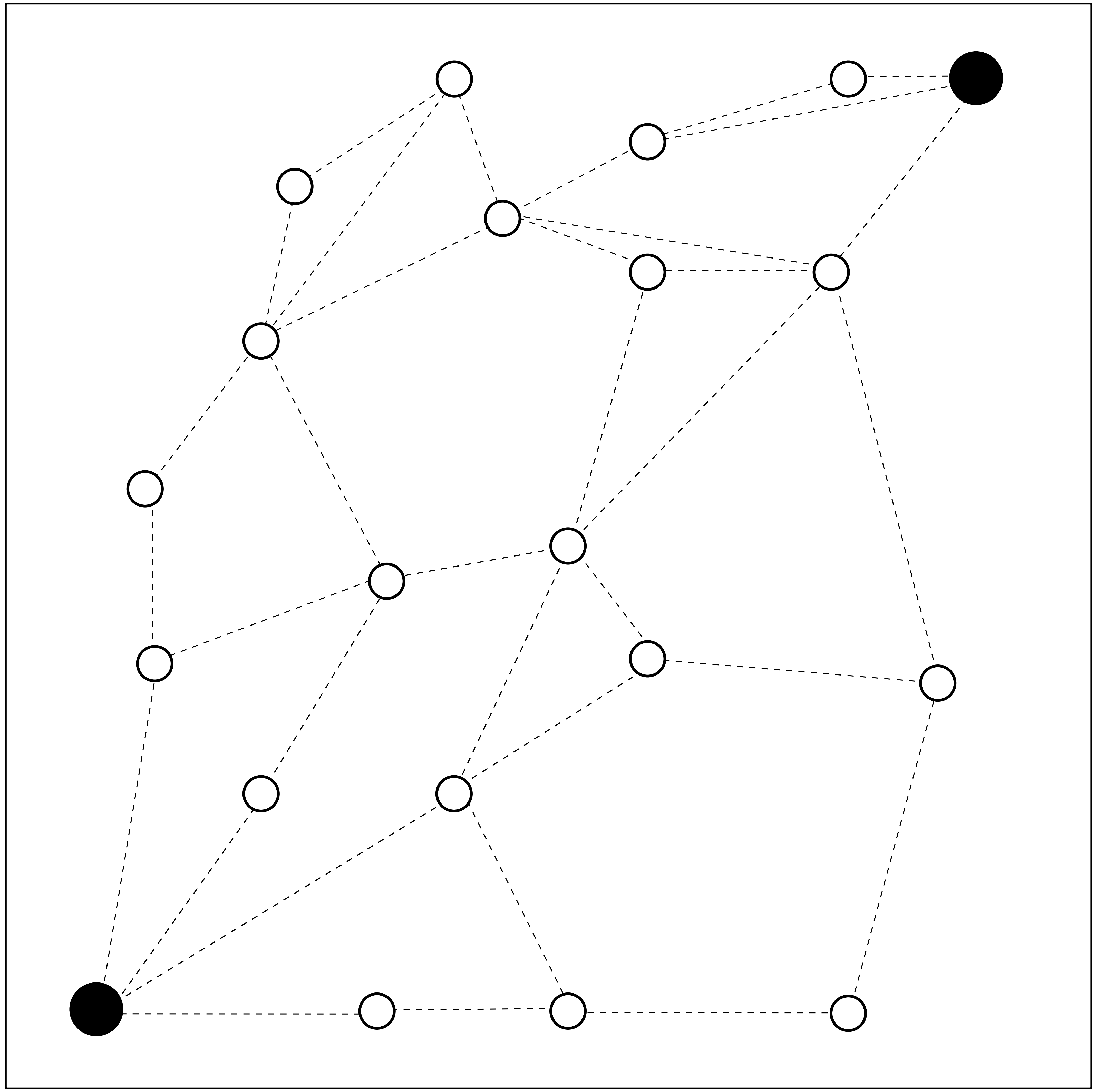
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



Recap: LazySP!

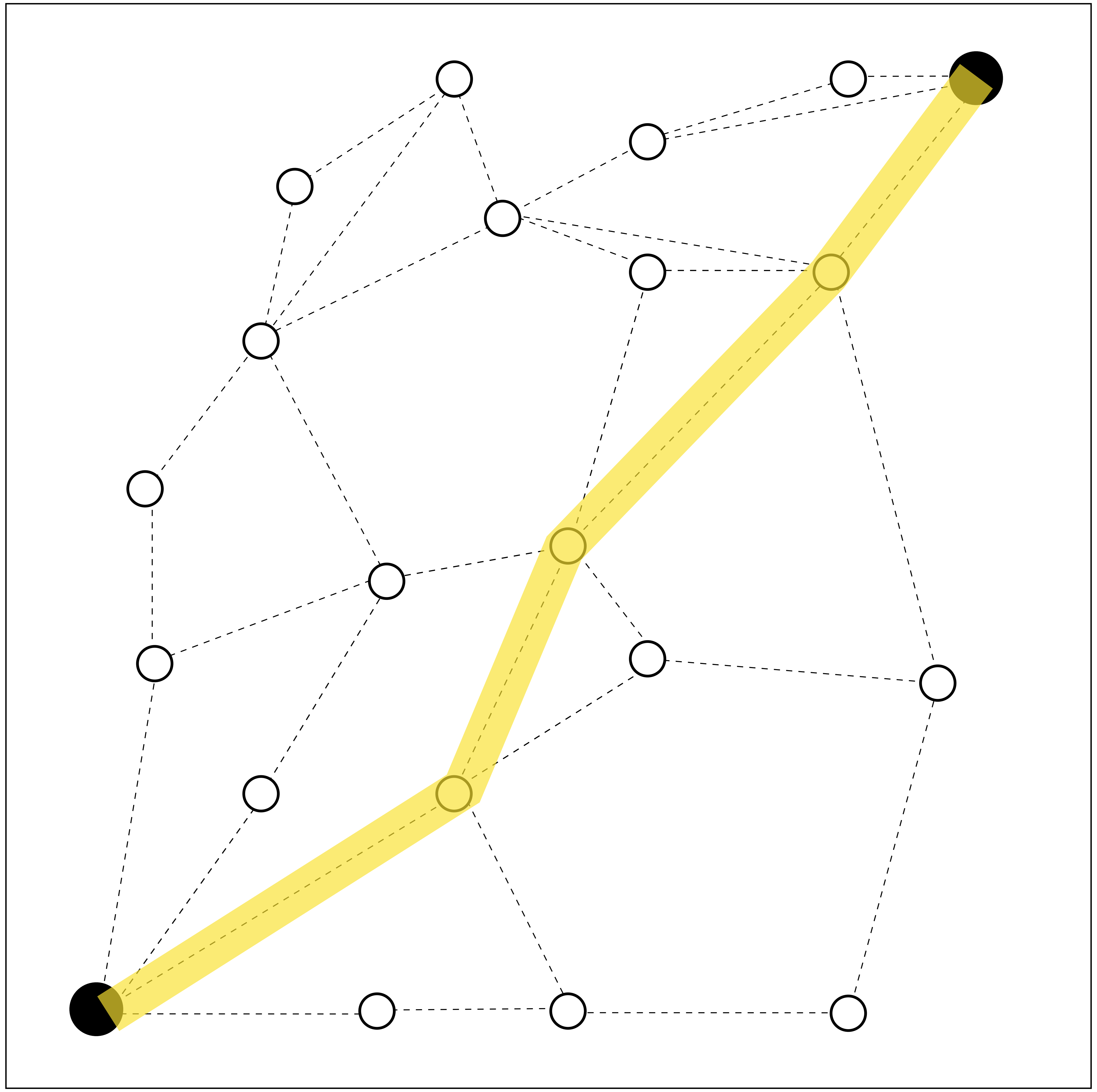
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



Recap: LazySP!

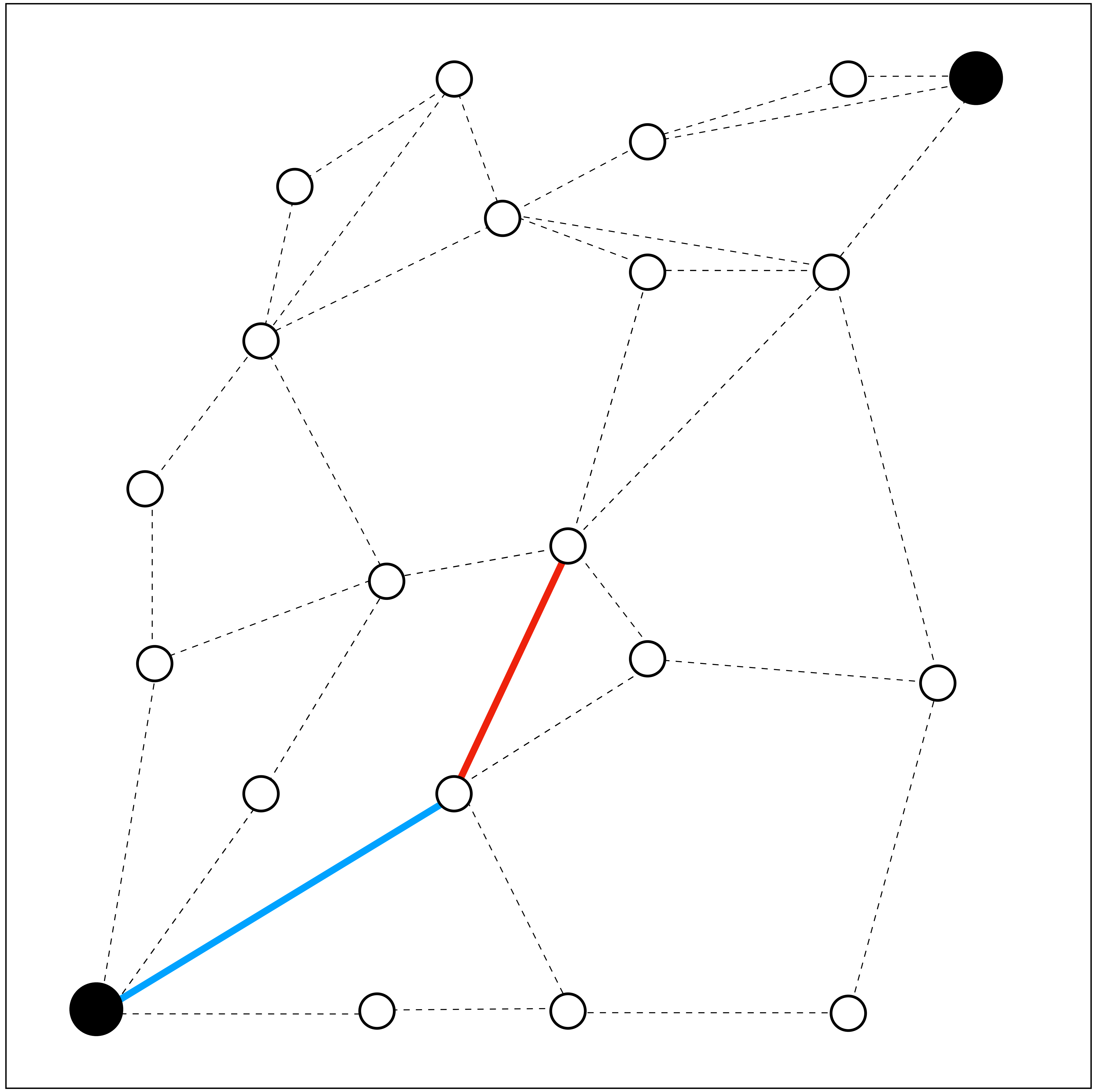
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



Recap: LazySP!

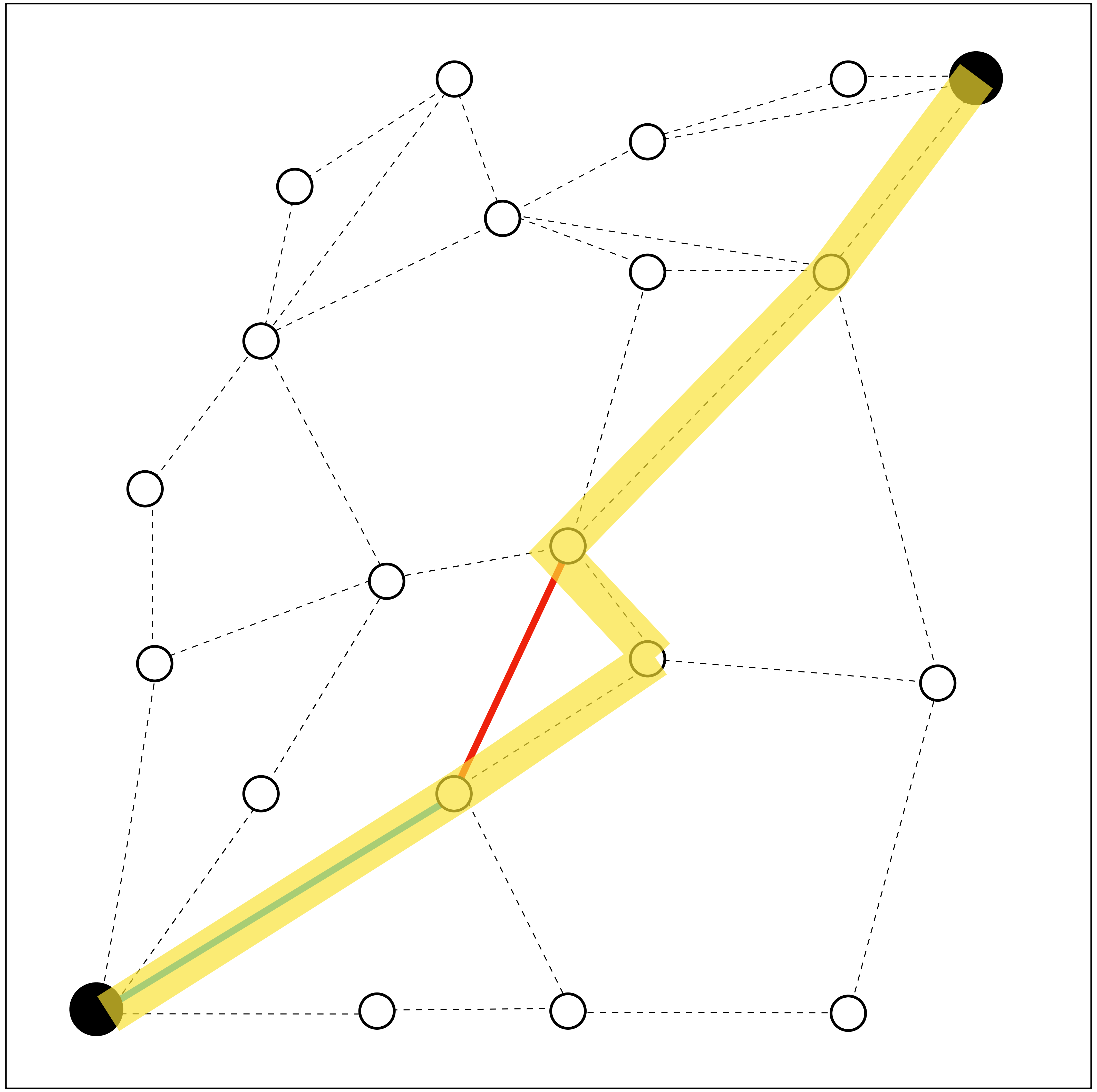
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



Recap: LazySP!

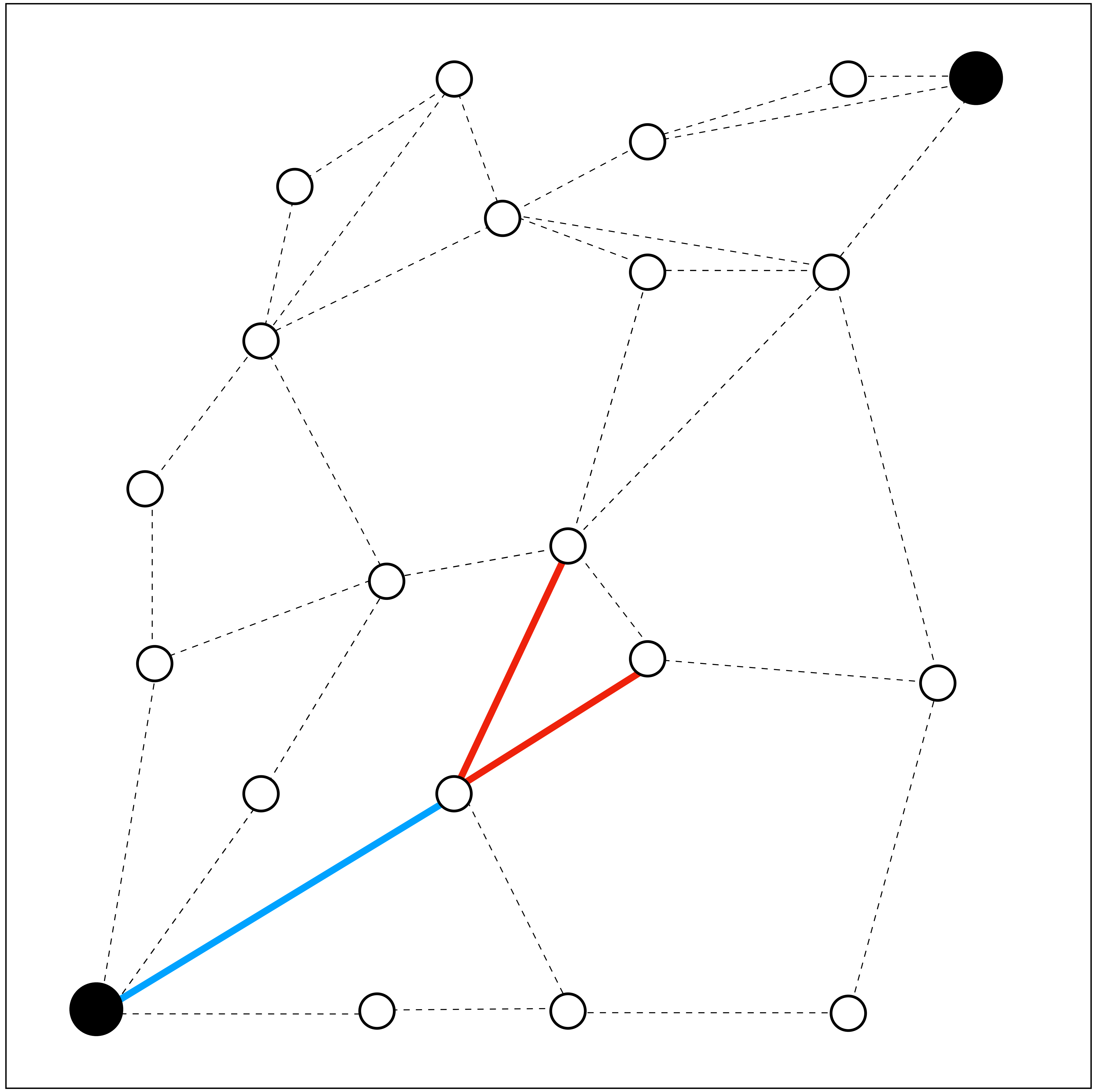
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



Recap: LazySP!

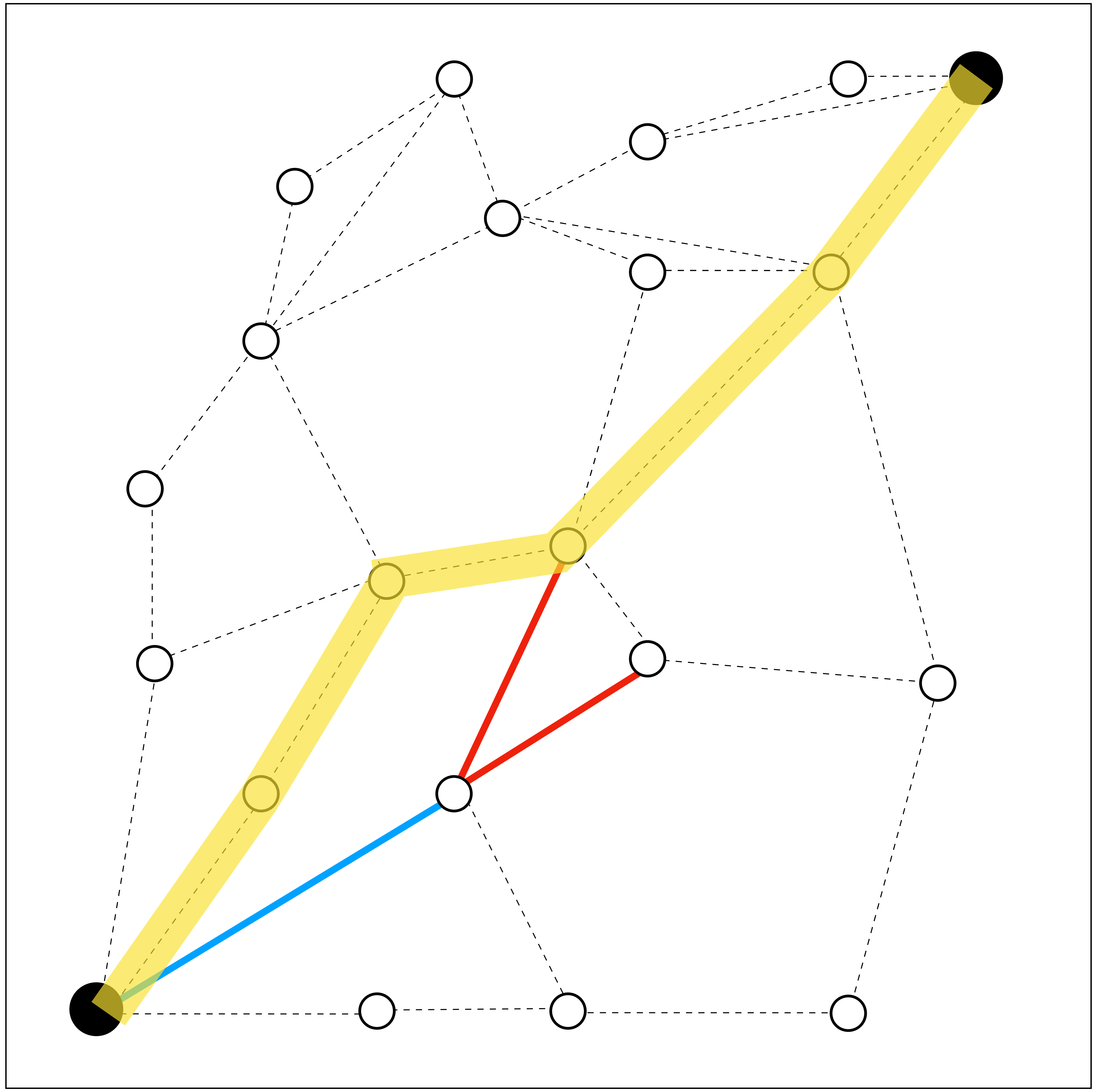
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



Recap: LazySP!

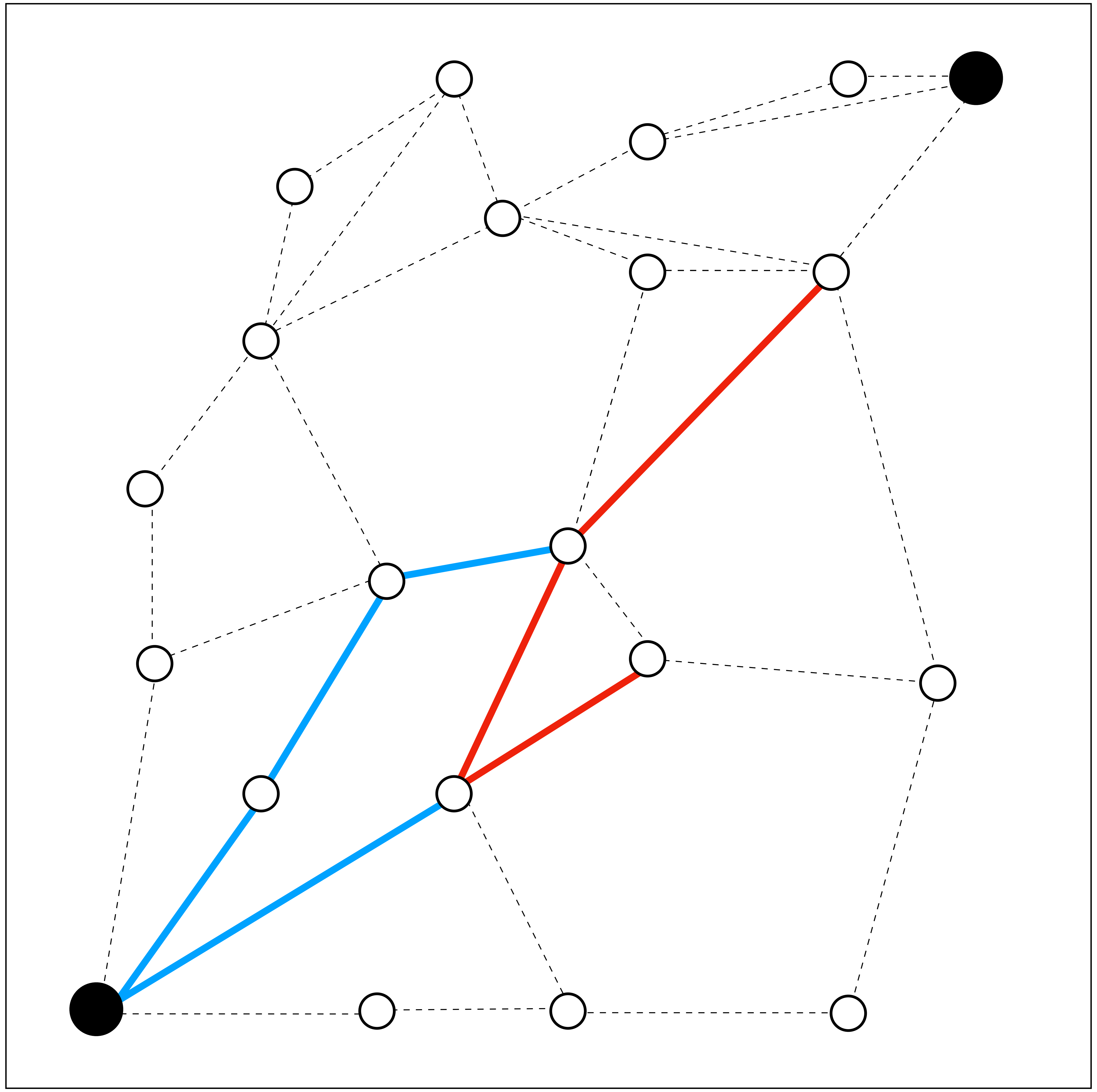
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



Recap: LazySP!

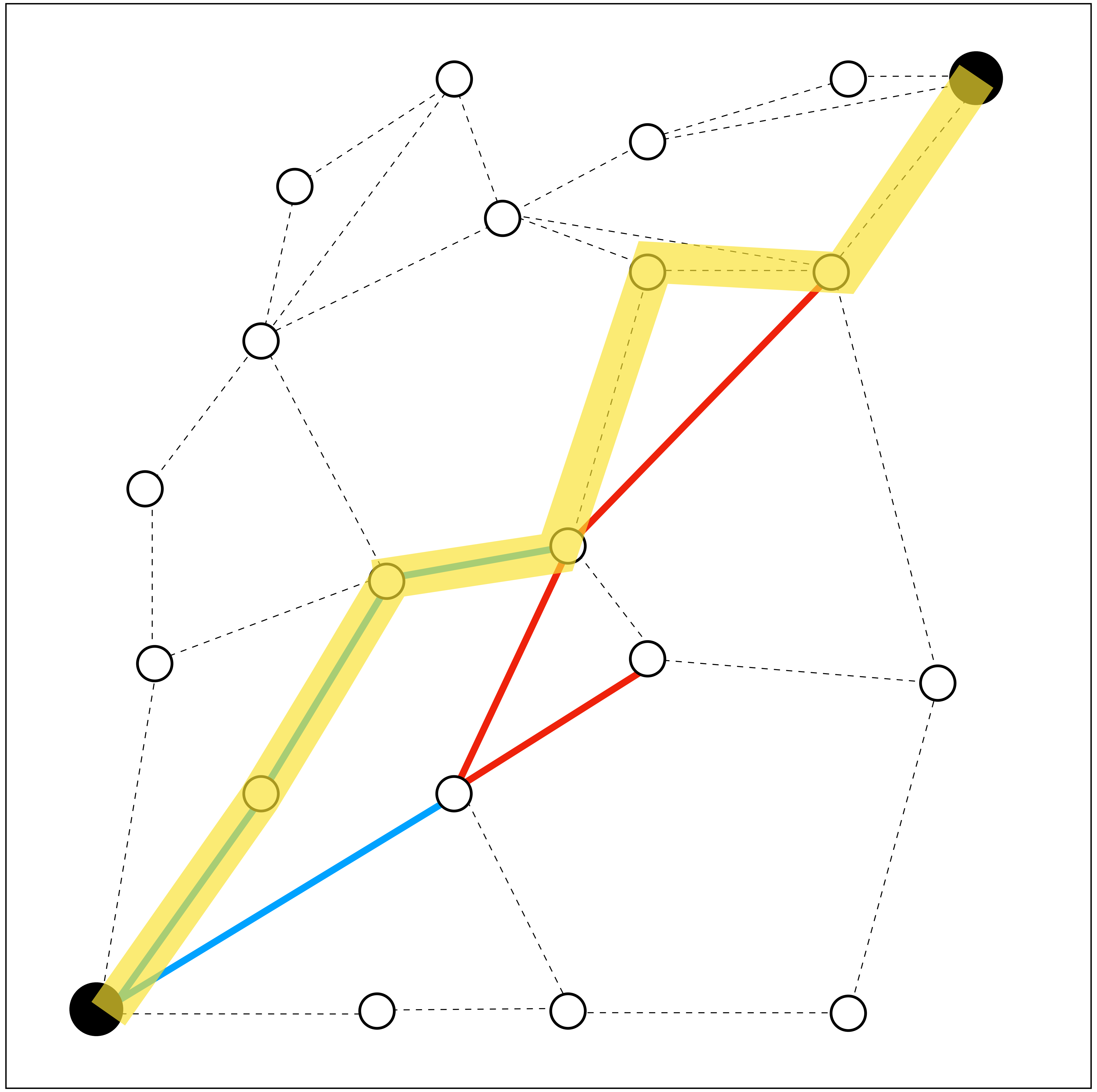
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

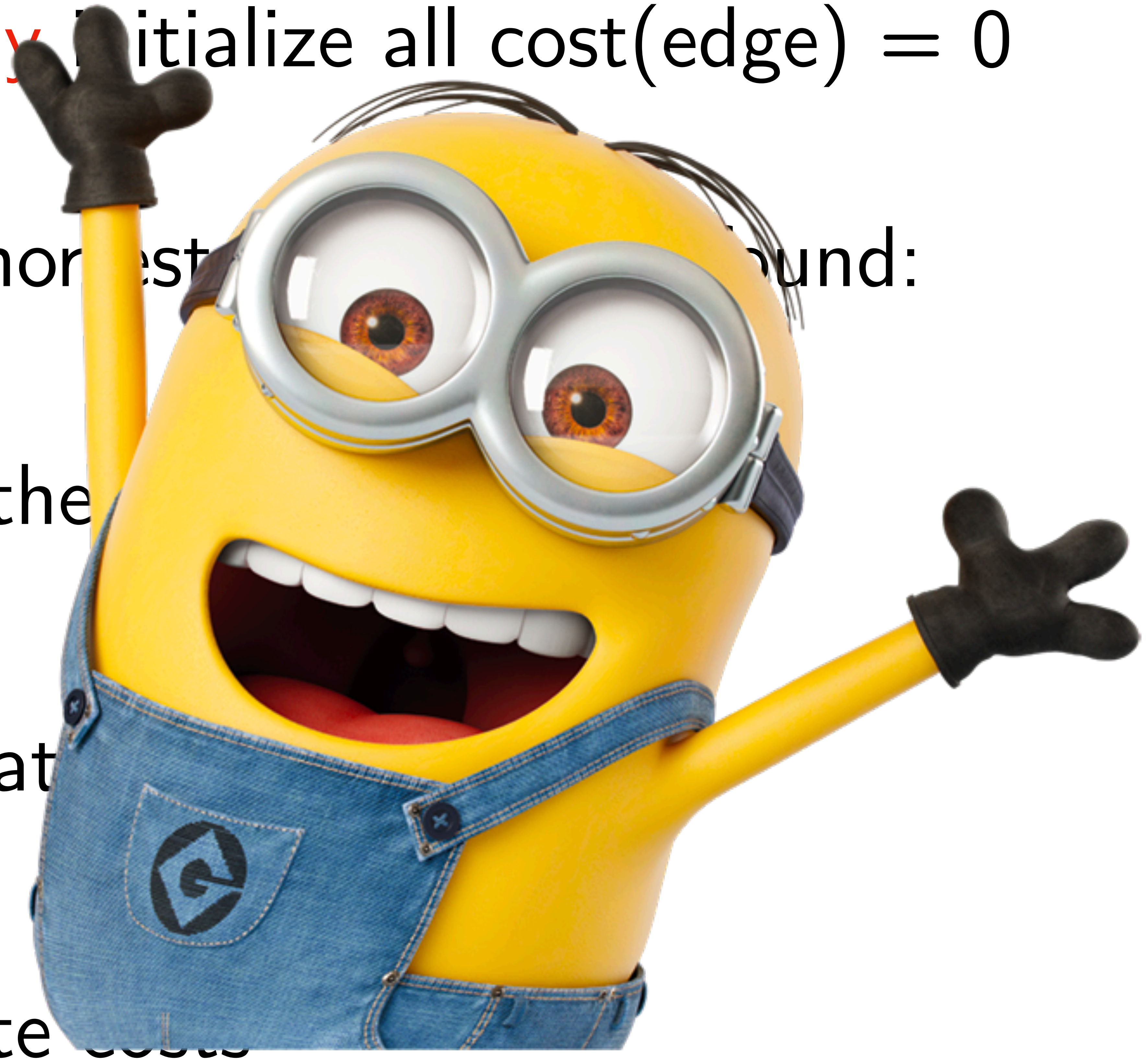
Evaluate shortest path

Update costs



Recap: LazySP!

Optimistically Initialize all cost(edge) = 0

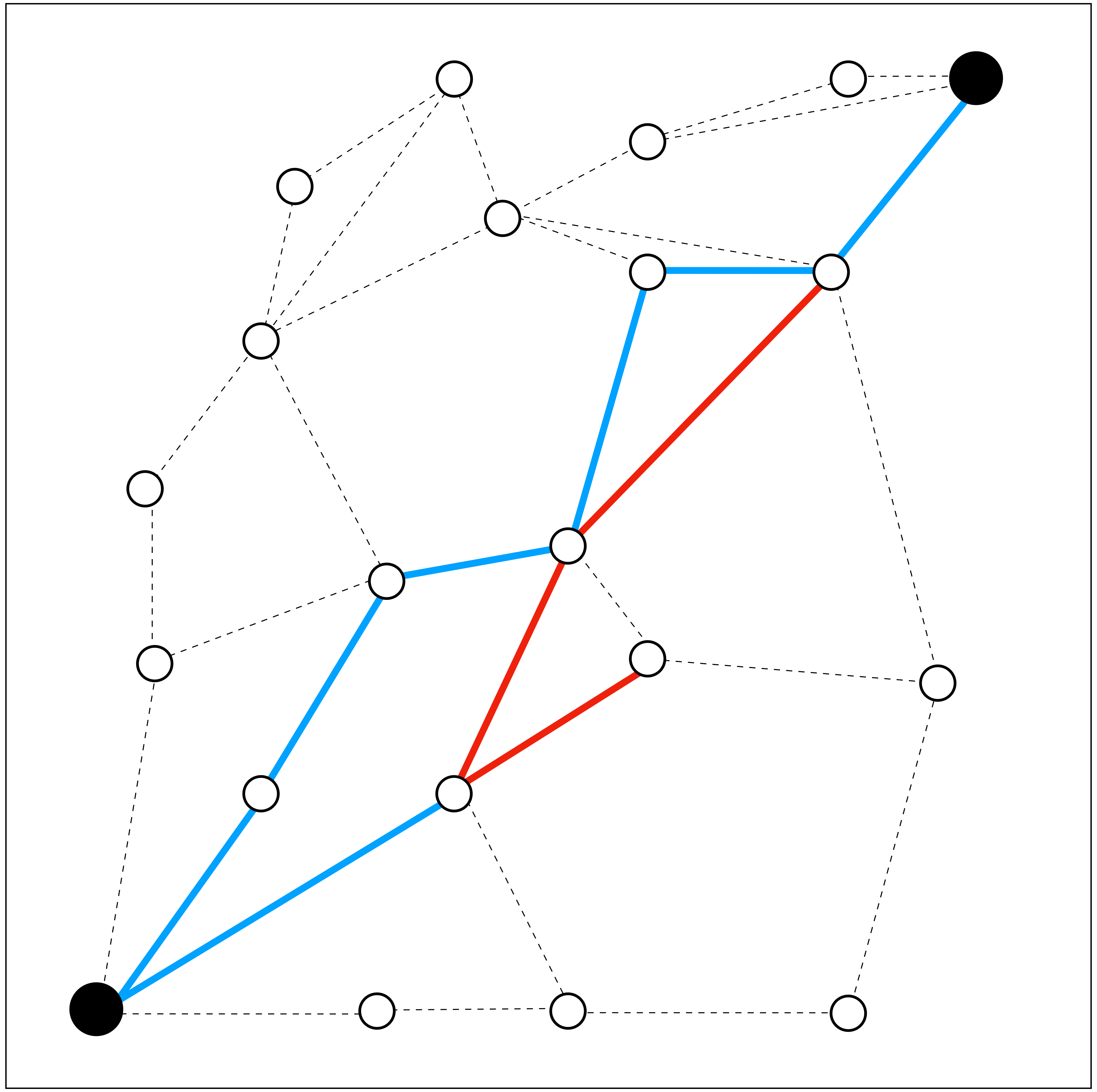


Repeat till shor[illegible]est[illegible]und:

Find the [illegible]

Evaluat[illegible]

Update [illegible]

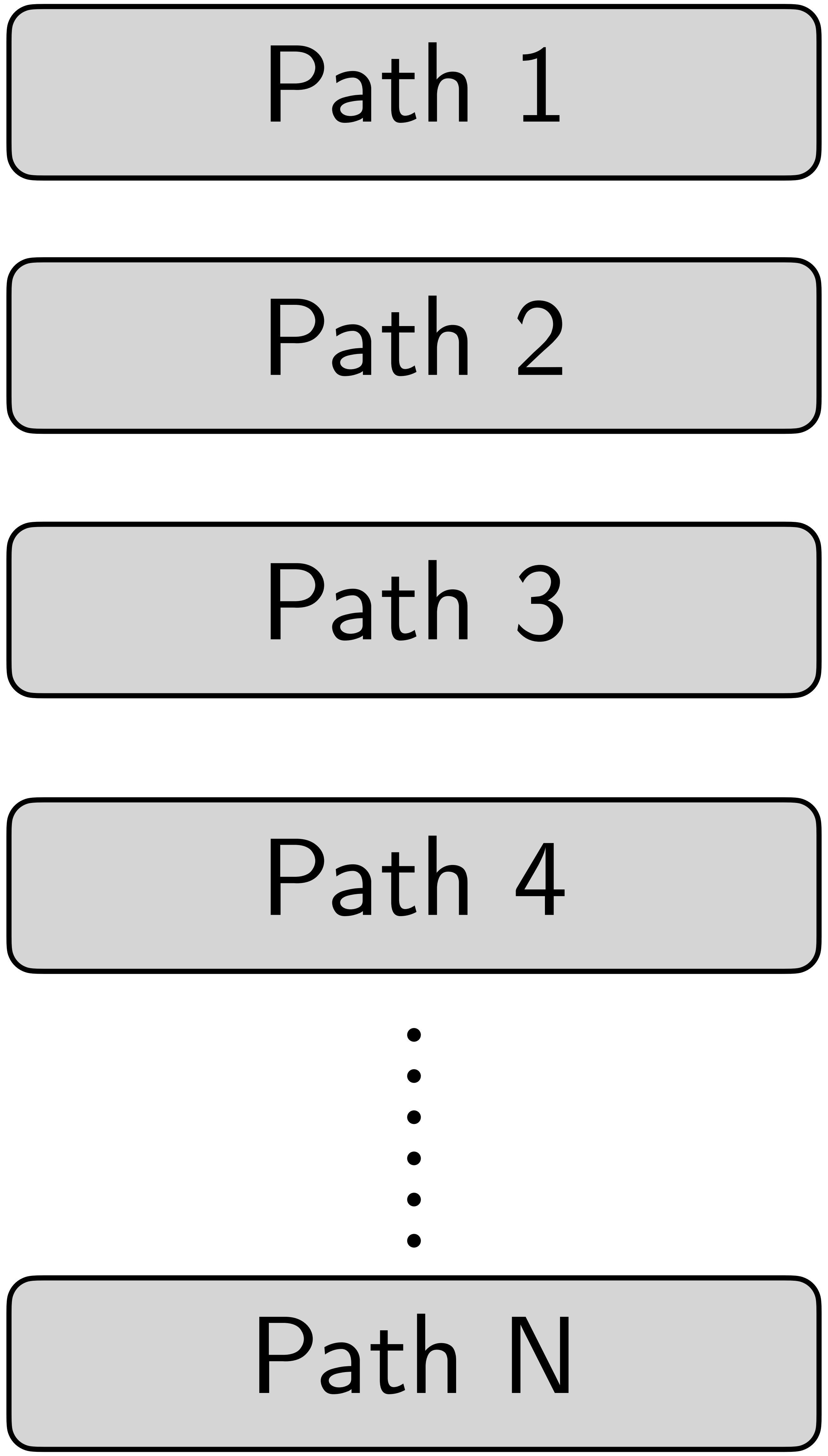


Principle of Optimism in the Face of Uncertainty (OFU)

One of two things will happen:

1. Either we are correct and done!
2. Or we were wrong and eliminated a candidate option

Optimism in the Face of Uncertainty



Path 1

Path 2

Path 3

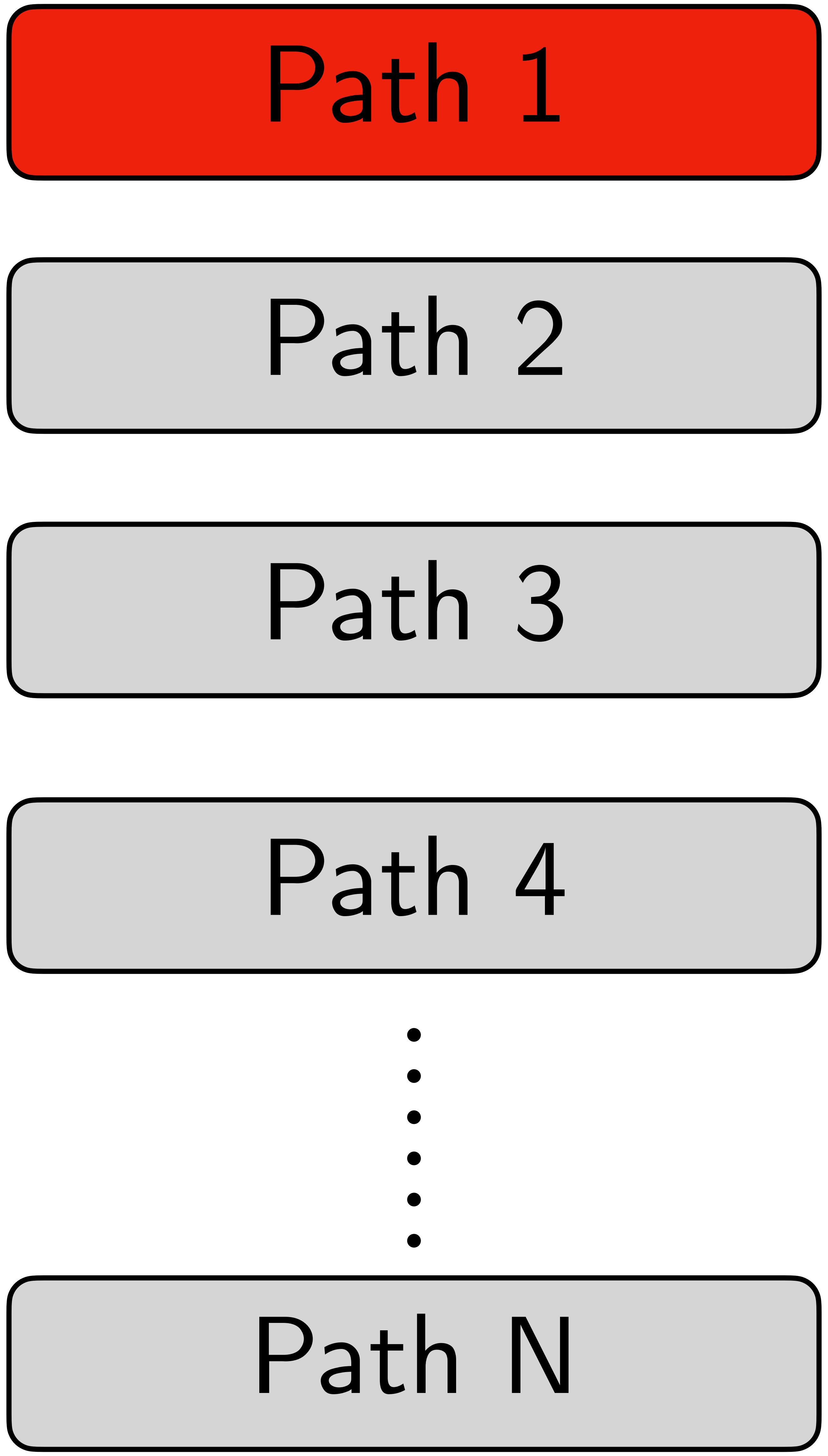
Path 4

⋮

Path N

Sort paths by ascending cost

Optimism in the Face of Uncertainty



Path 1

Path 2

Path 3

Path 4

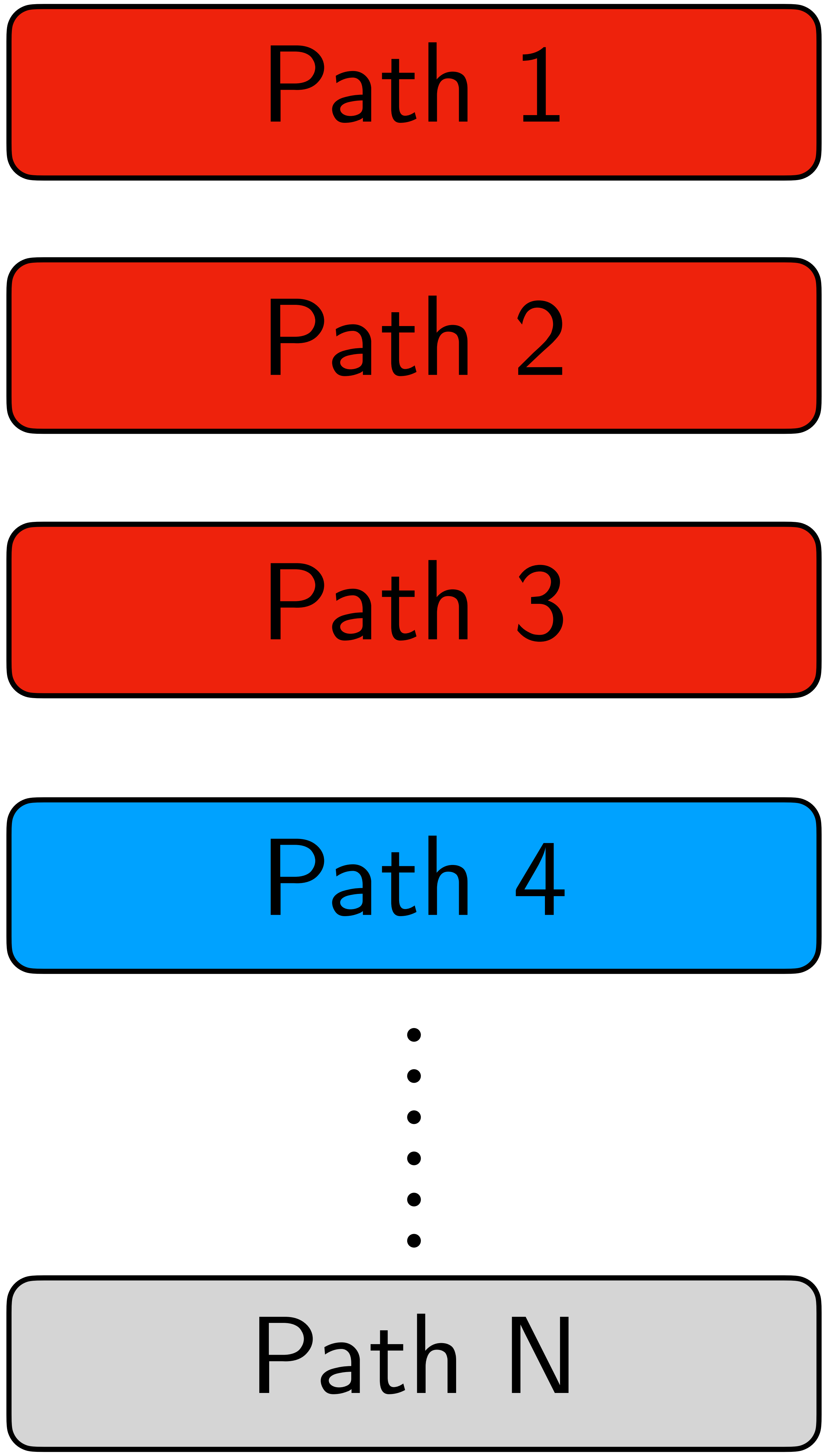
⋮

Path N

Sort paths by ascending cost

Keep checking each path

Optimism in the Face of Uncertainty



Path 1

Path 2

Path 3

Path 4

⋮

Path N

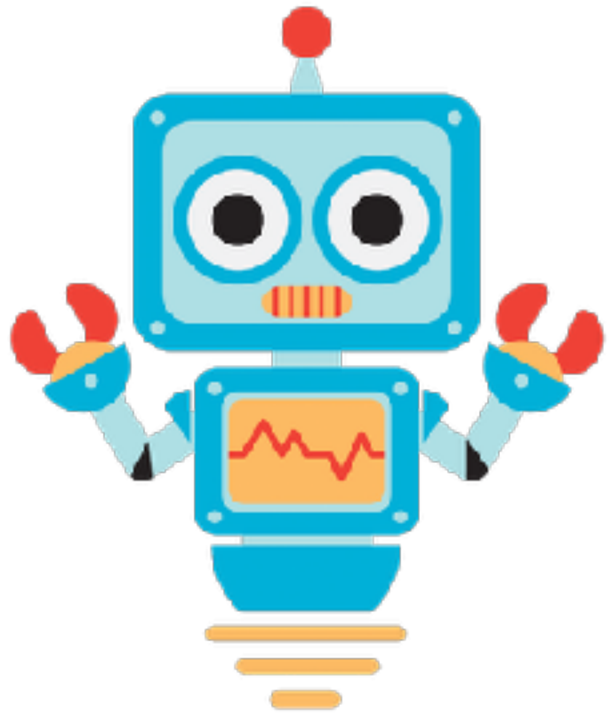
Sort paths by ascending cost

Keep checking each path

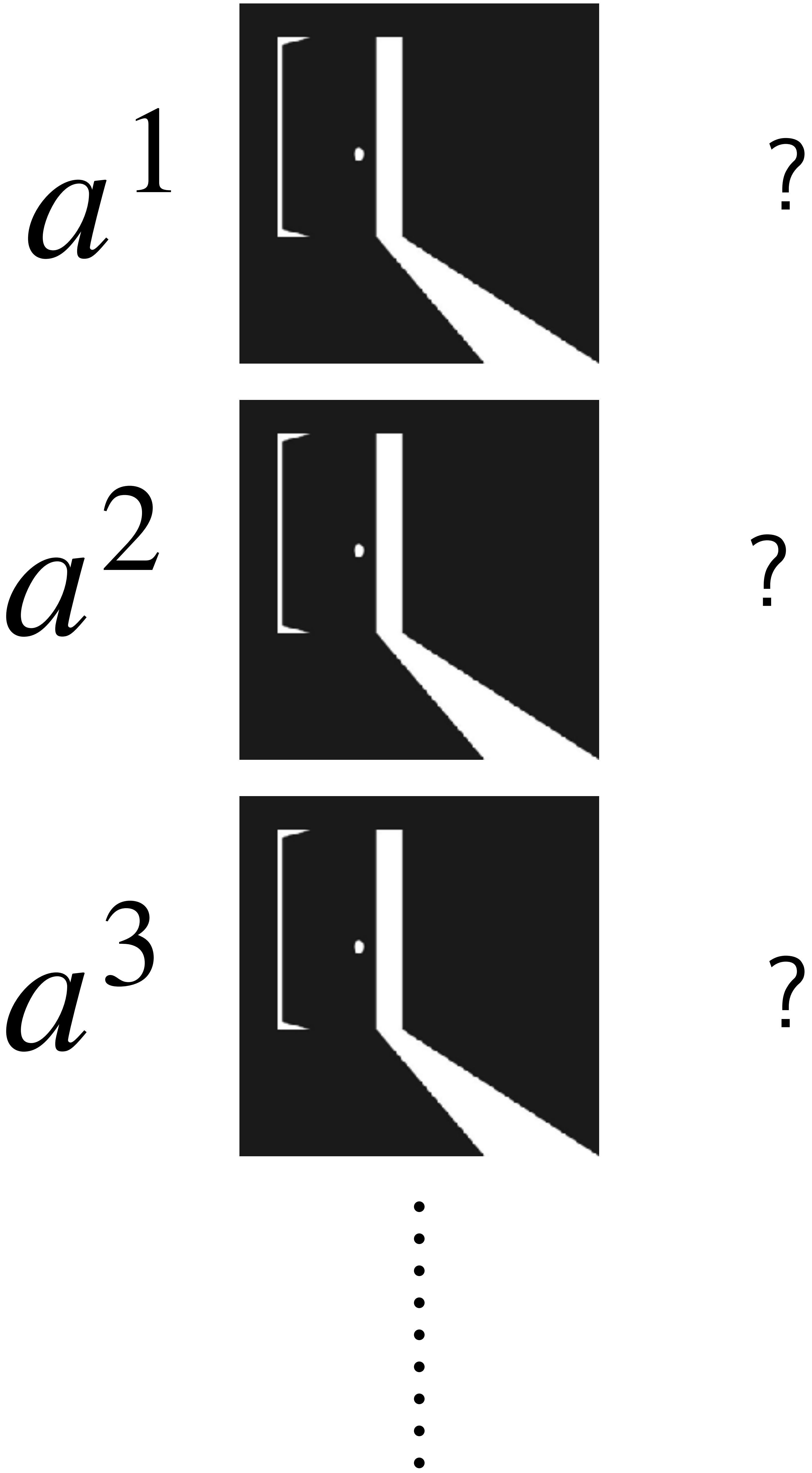
At most check K paths till
you find the shortest one

Optimal strategy given
no other information

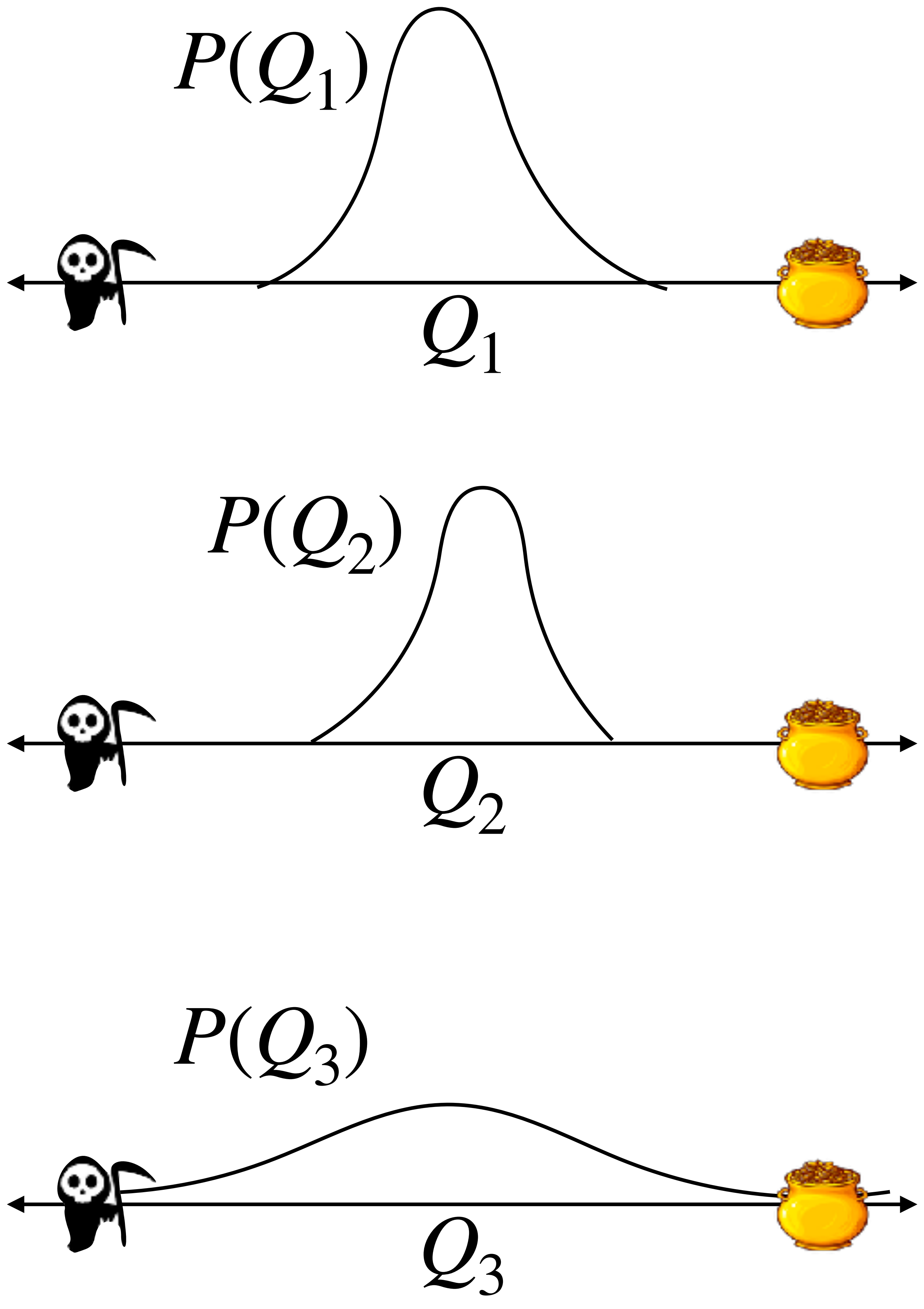
What if each evaluation is stochastic?



Doors

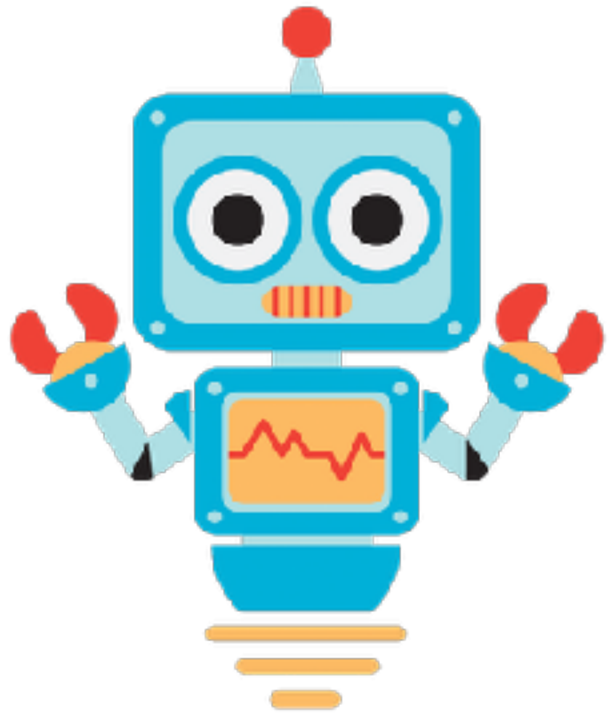


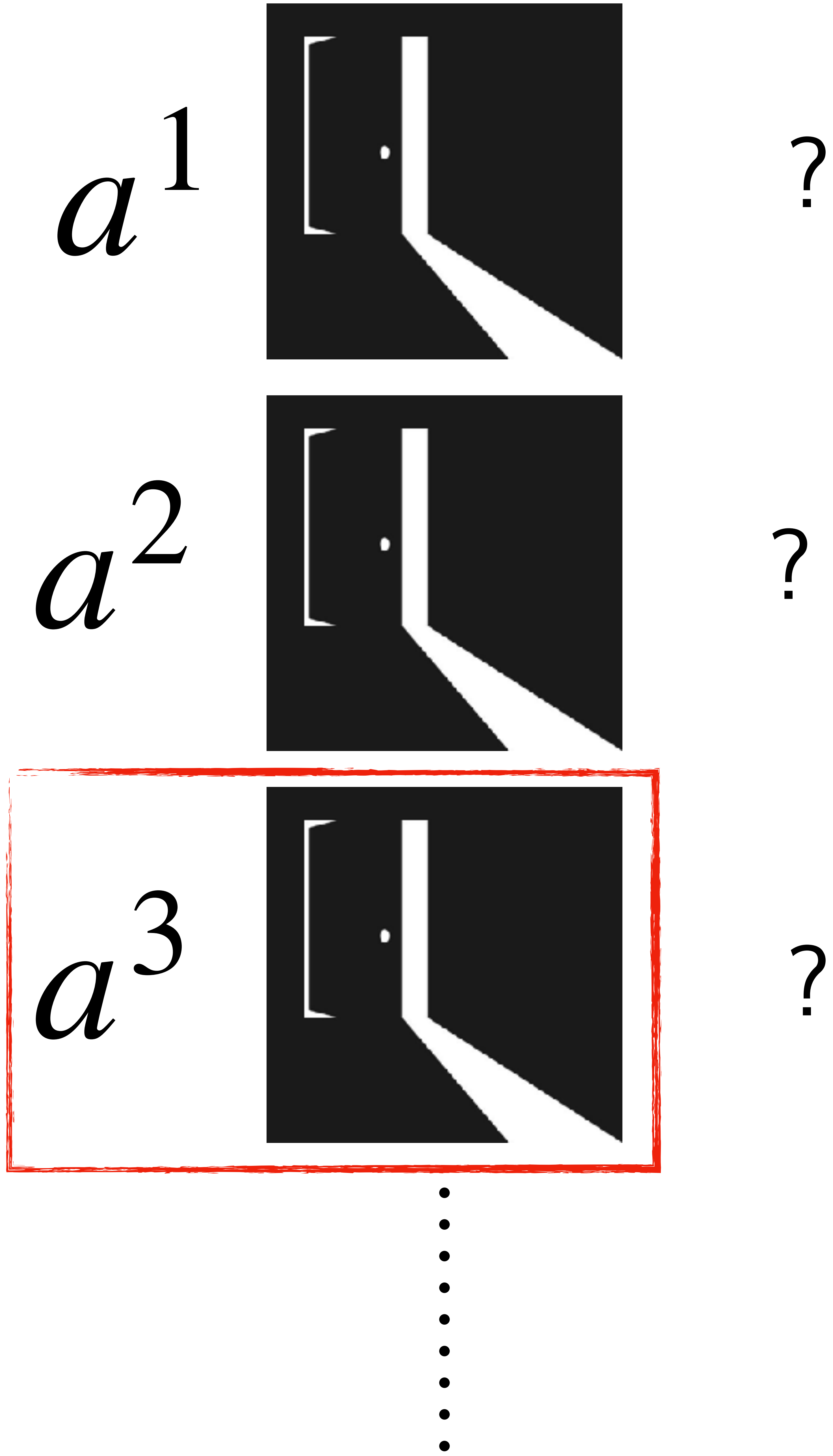
Values

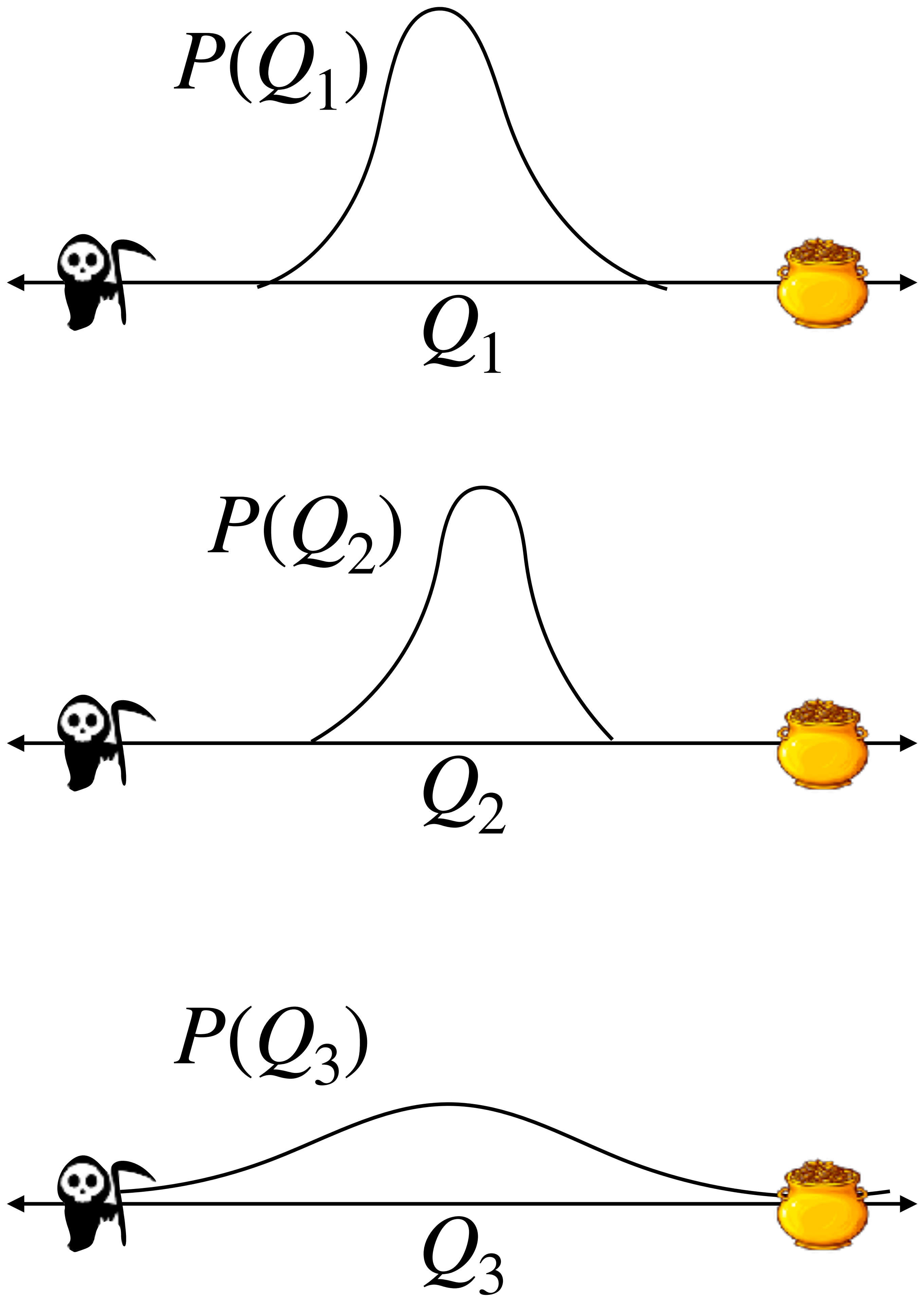


Doors

Values





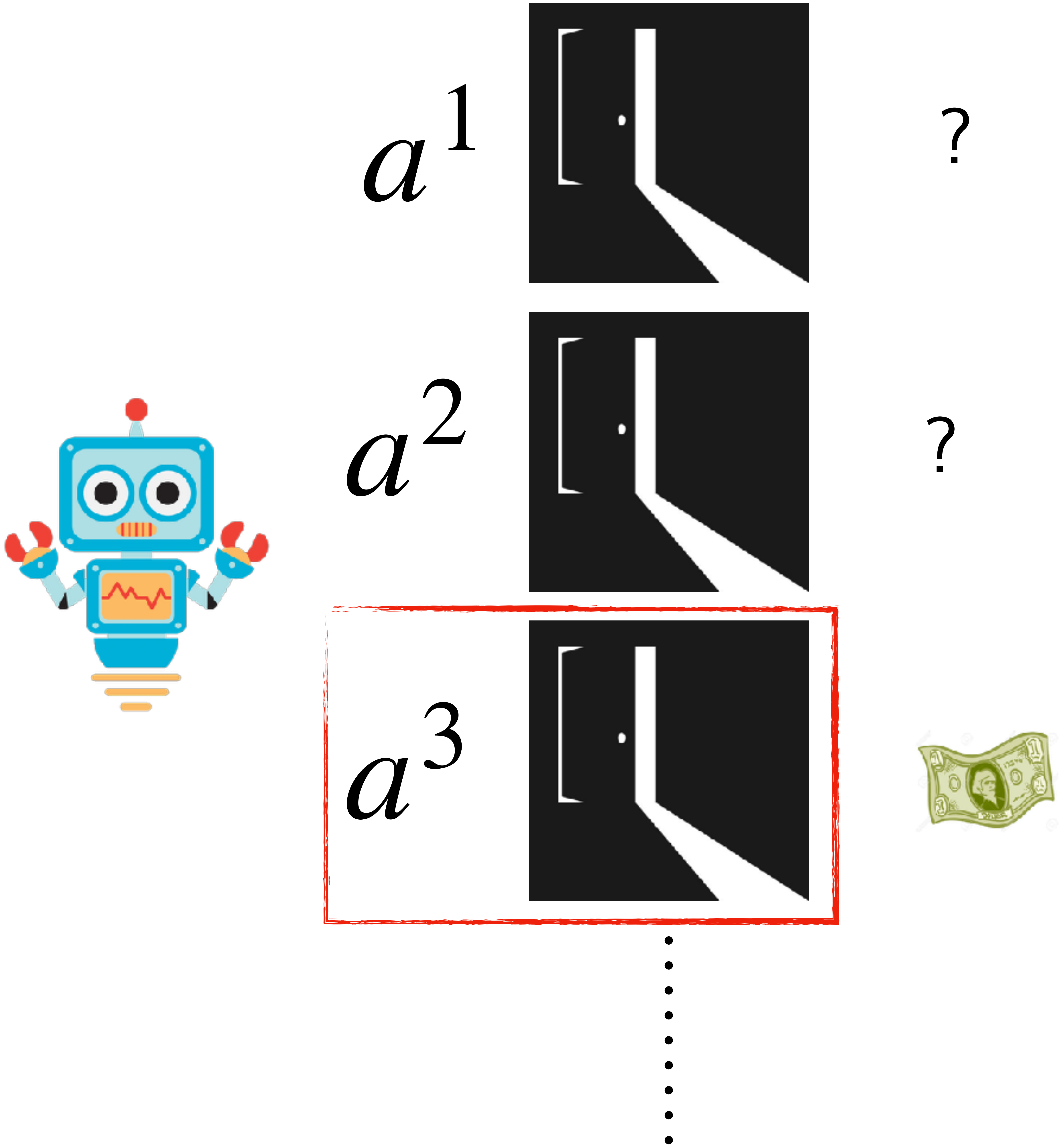


Doors

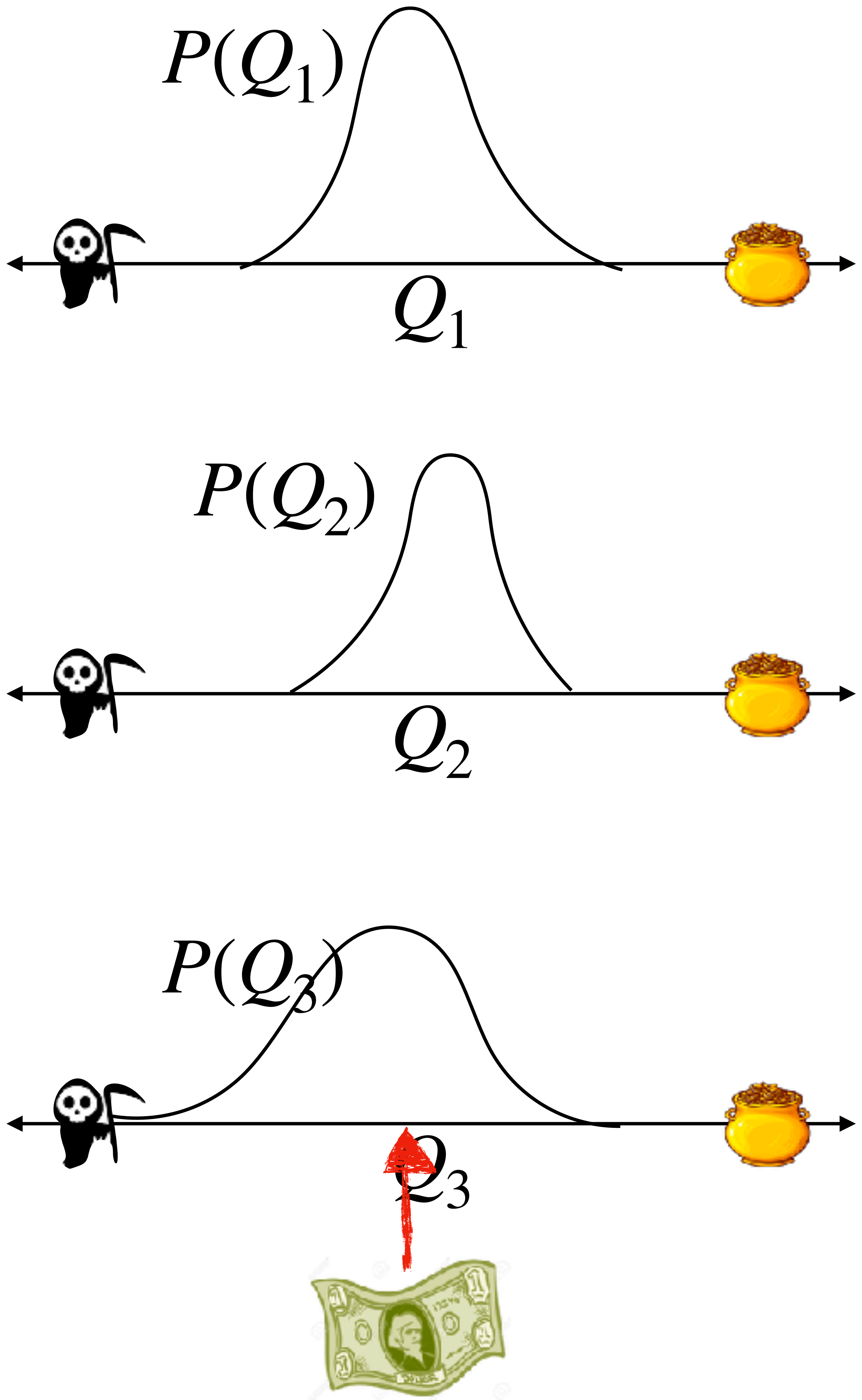
Values

a^1
?
a^2
?
a^3
$P(Q_1)$
Q_1
$P(Q_2)$
Q_2
$P(Q_3)$
Q_3

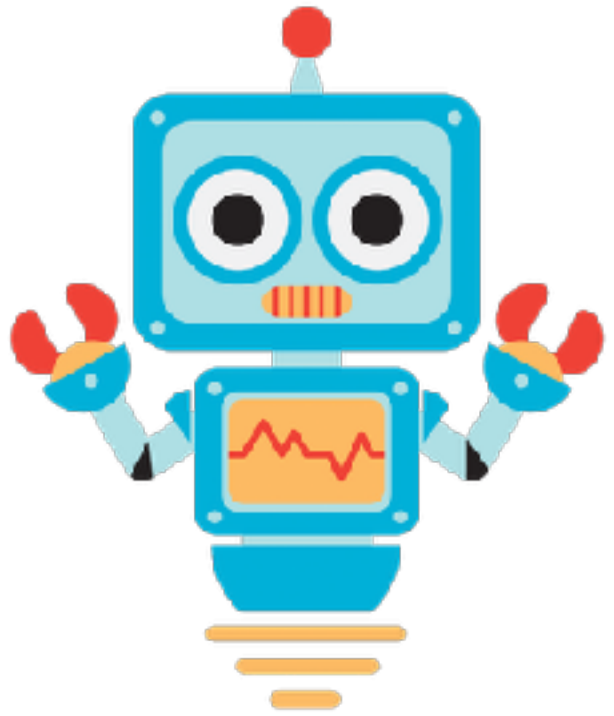
Doors

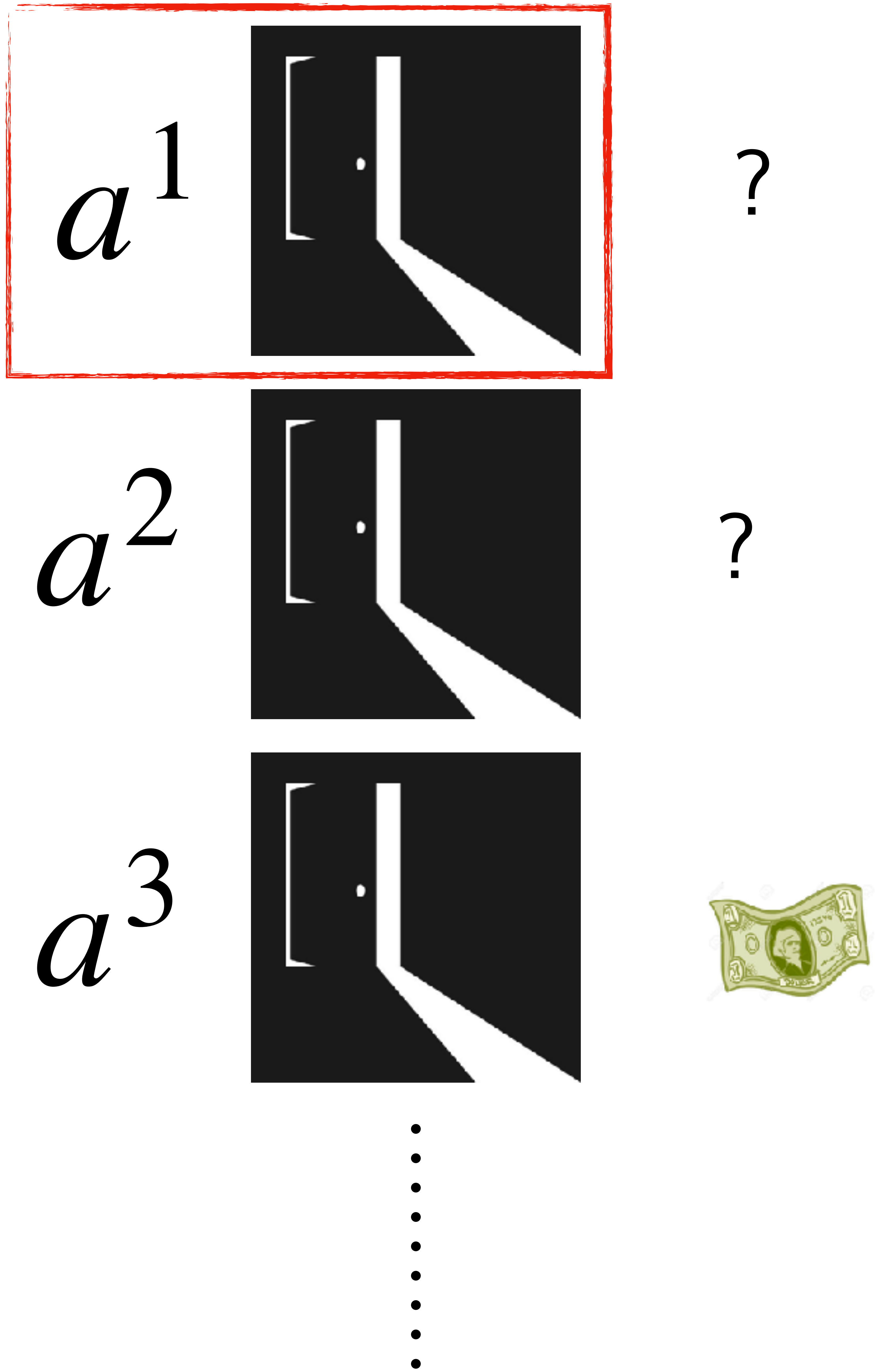


Values

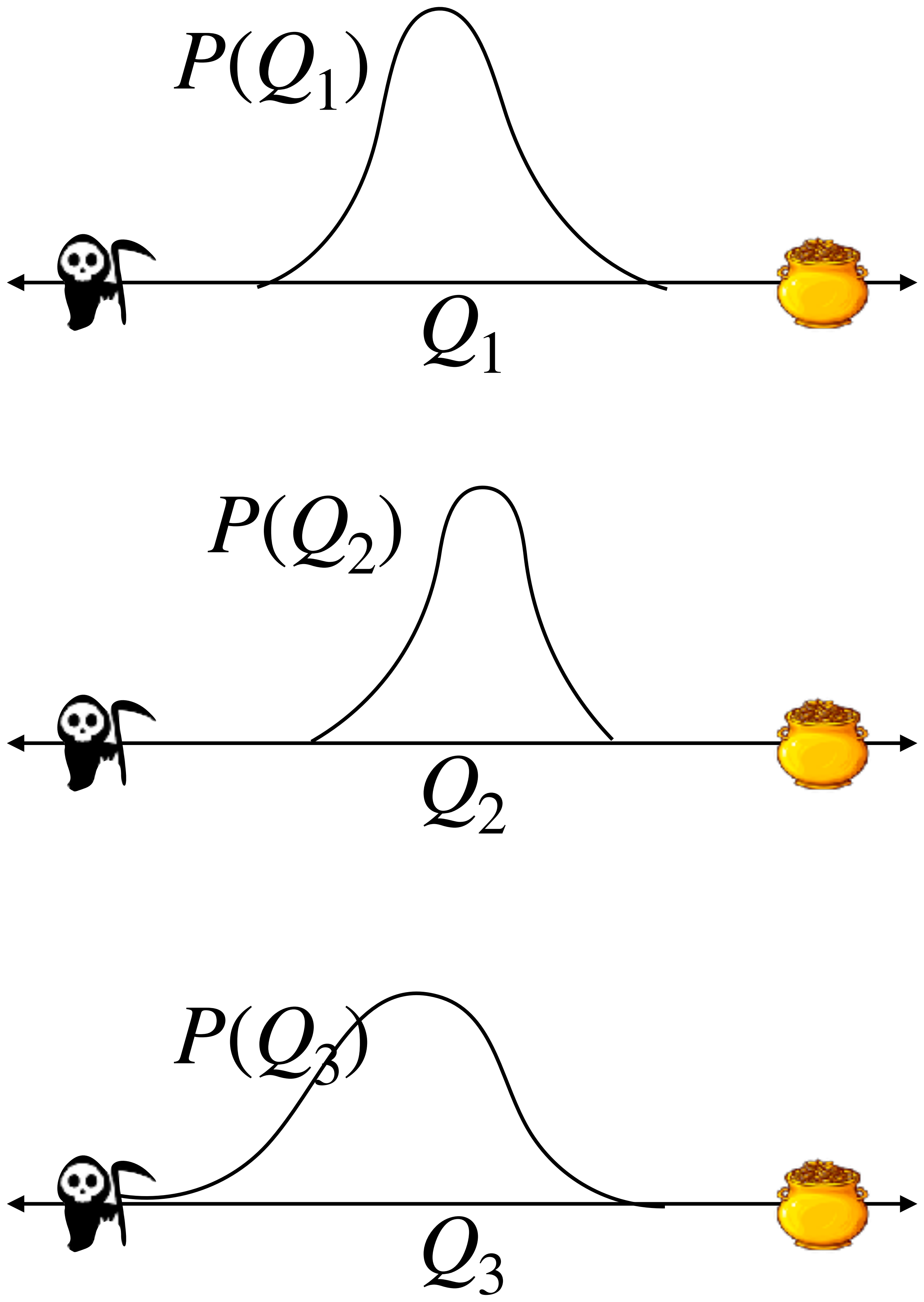


Doors





Values



Doors

Values

a^1
?
$P(Q_1)$
Q_1
a^2
?
?
$P(Q_2)$
Q_2
a^3
?
$P(Q_3)$
Q_3

Upper Confidence Bound

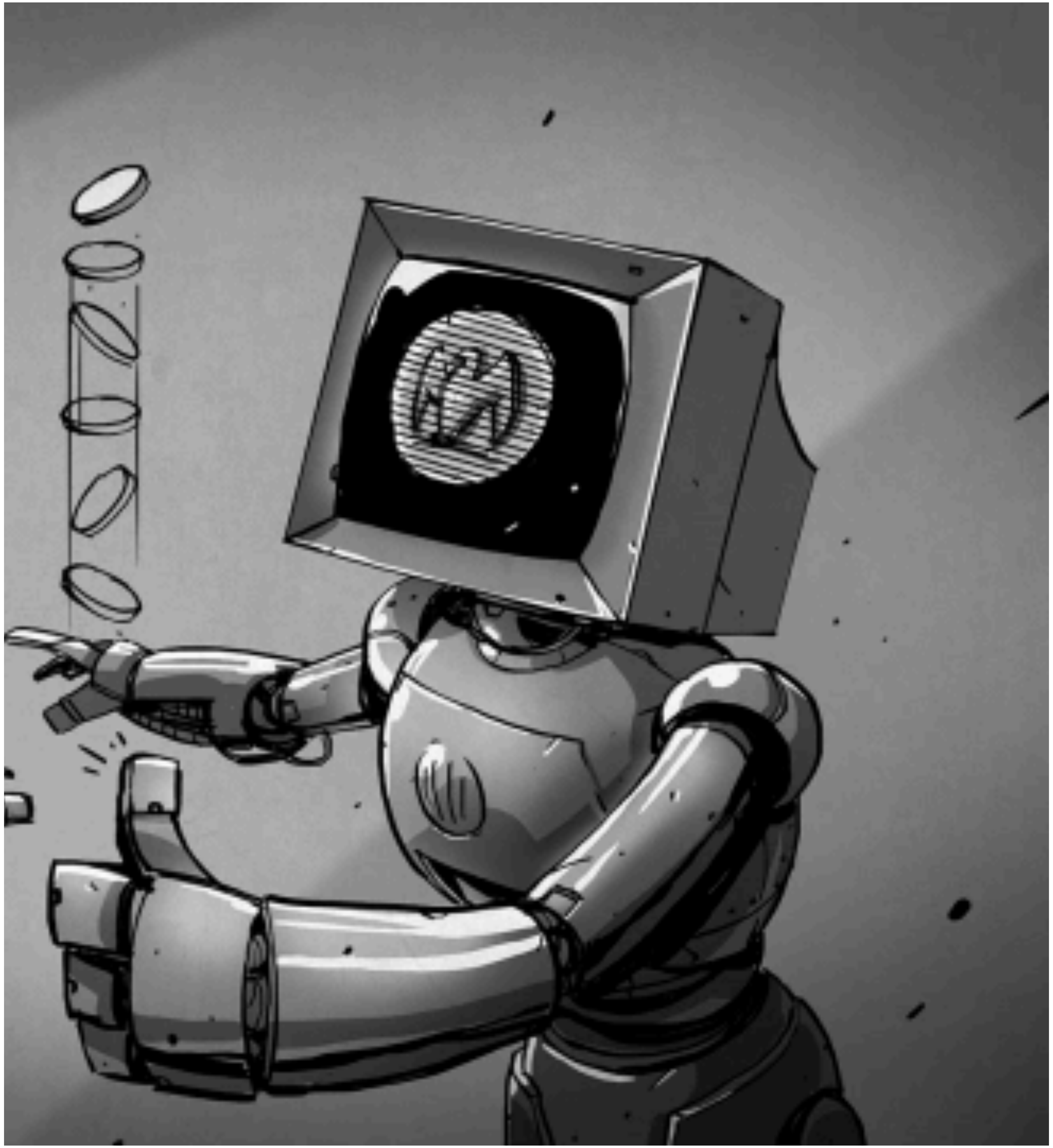
At every time t, for every action a, you need to estimate two things:

- $\hat{Q}_t(a)$: The mean value of an action
- $\hat{U}_t(a)$: The upper confidence of an action

Then select the *most optimistic action*:

$$a_t = \arg\max_a \hat{Q}_t(a) + \hat{U}_t(a)$$

Can OFU explore a bit too much?



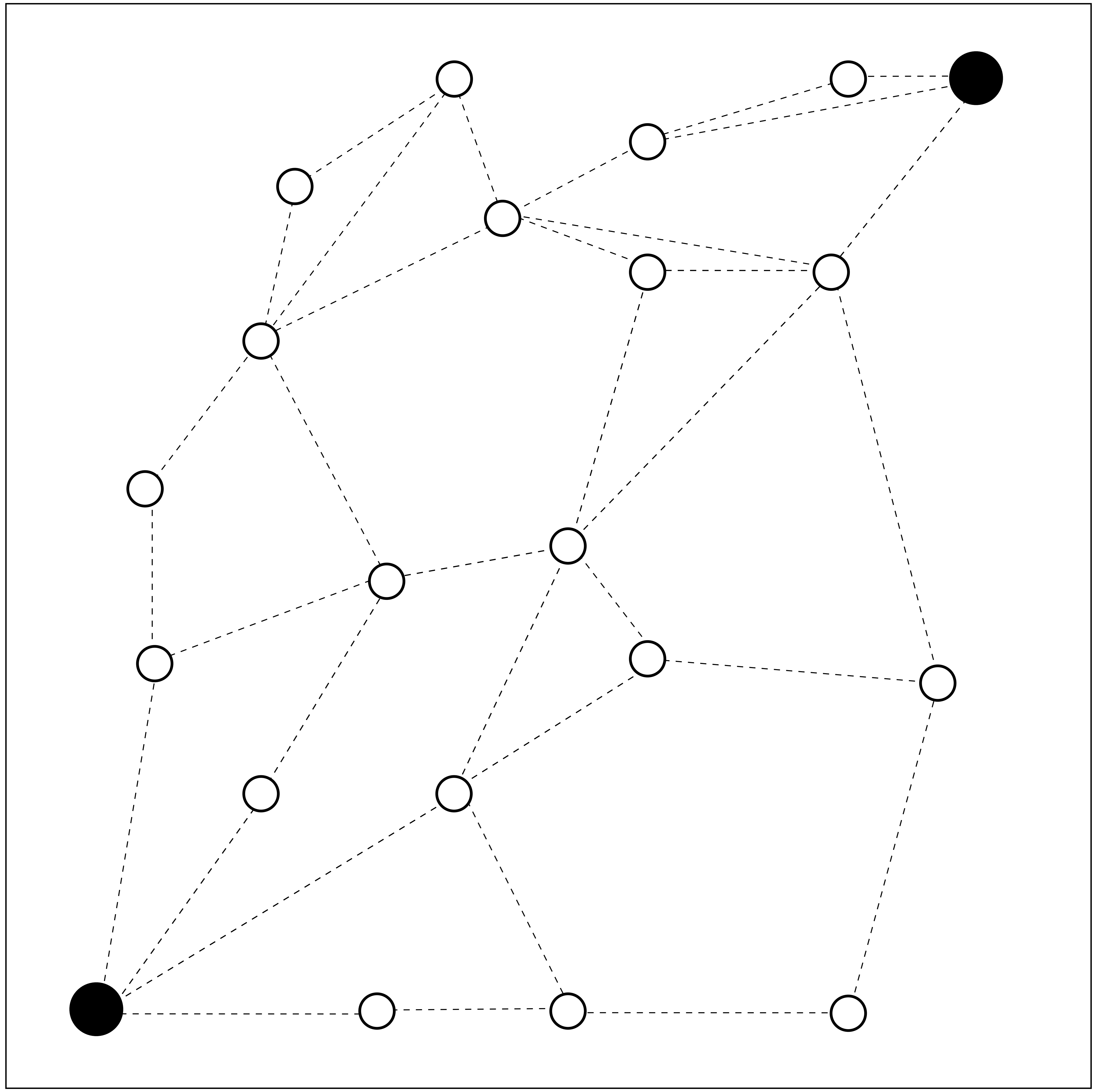
Posterior Sampling

The Online Shortest Path Problem

You just moved to Cornell and are traveling from office to home.

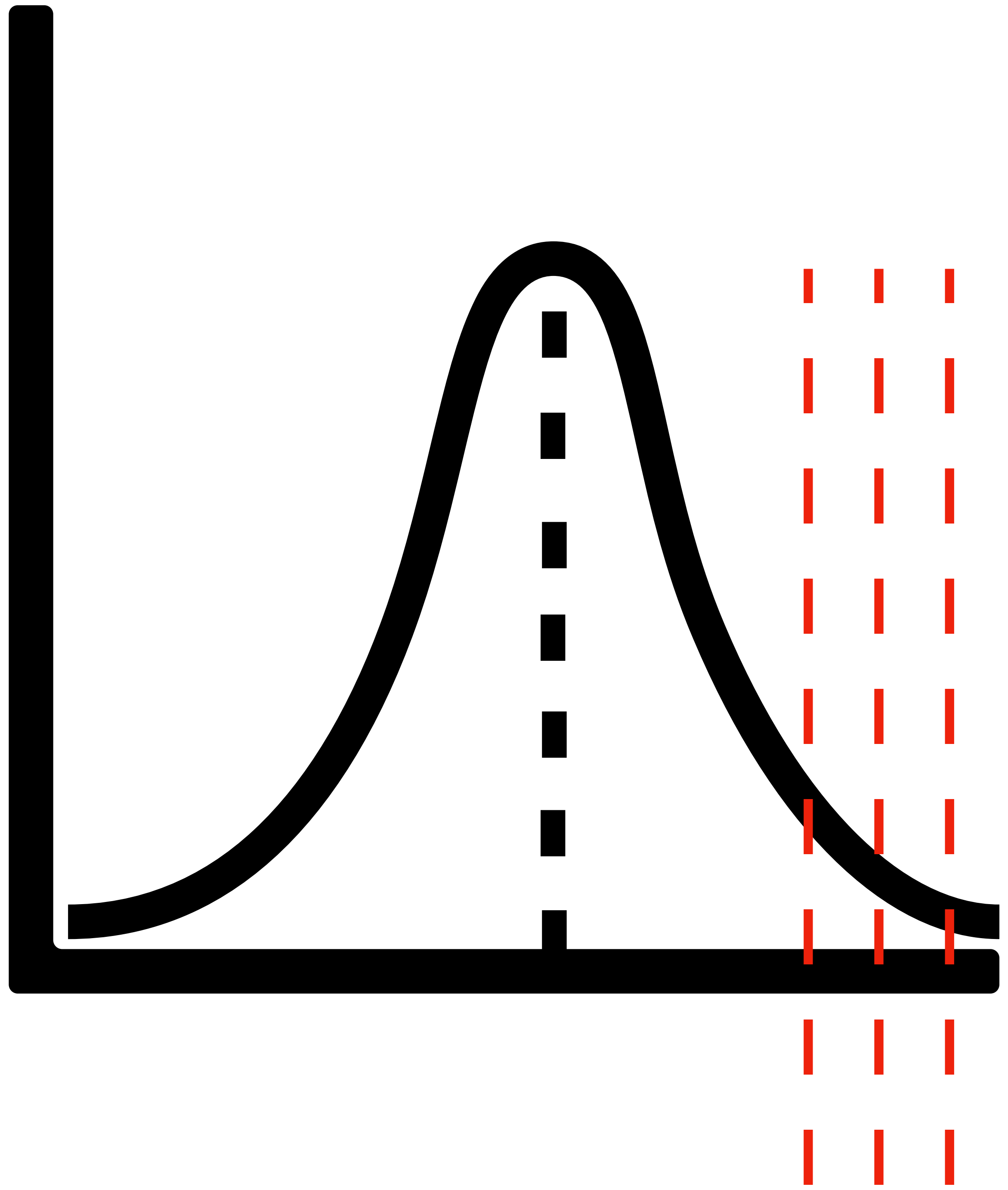
You would like to get home quickly but you are uncertain about travel times along each edge

Suppose we had a prior on travel time for each edge (Mean θ_e, Var σ_e)

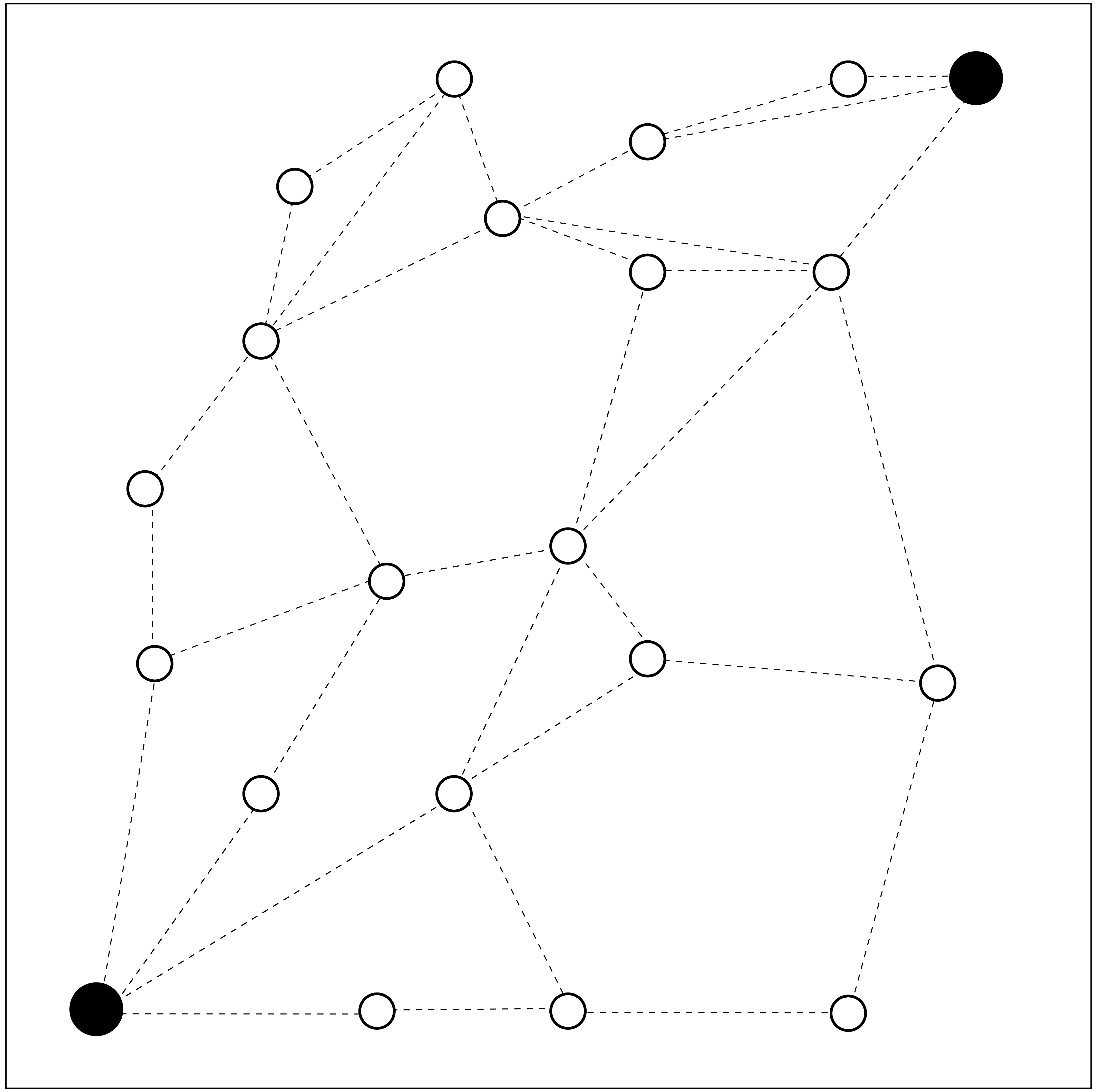


Can we apply UCB?

For each edge e we have to compute an upper confidence bound (Let's say negative of travel time)



Which one do you choose?



What if …

… we just sampled travel times from our prior and solved the shortest path?

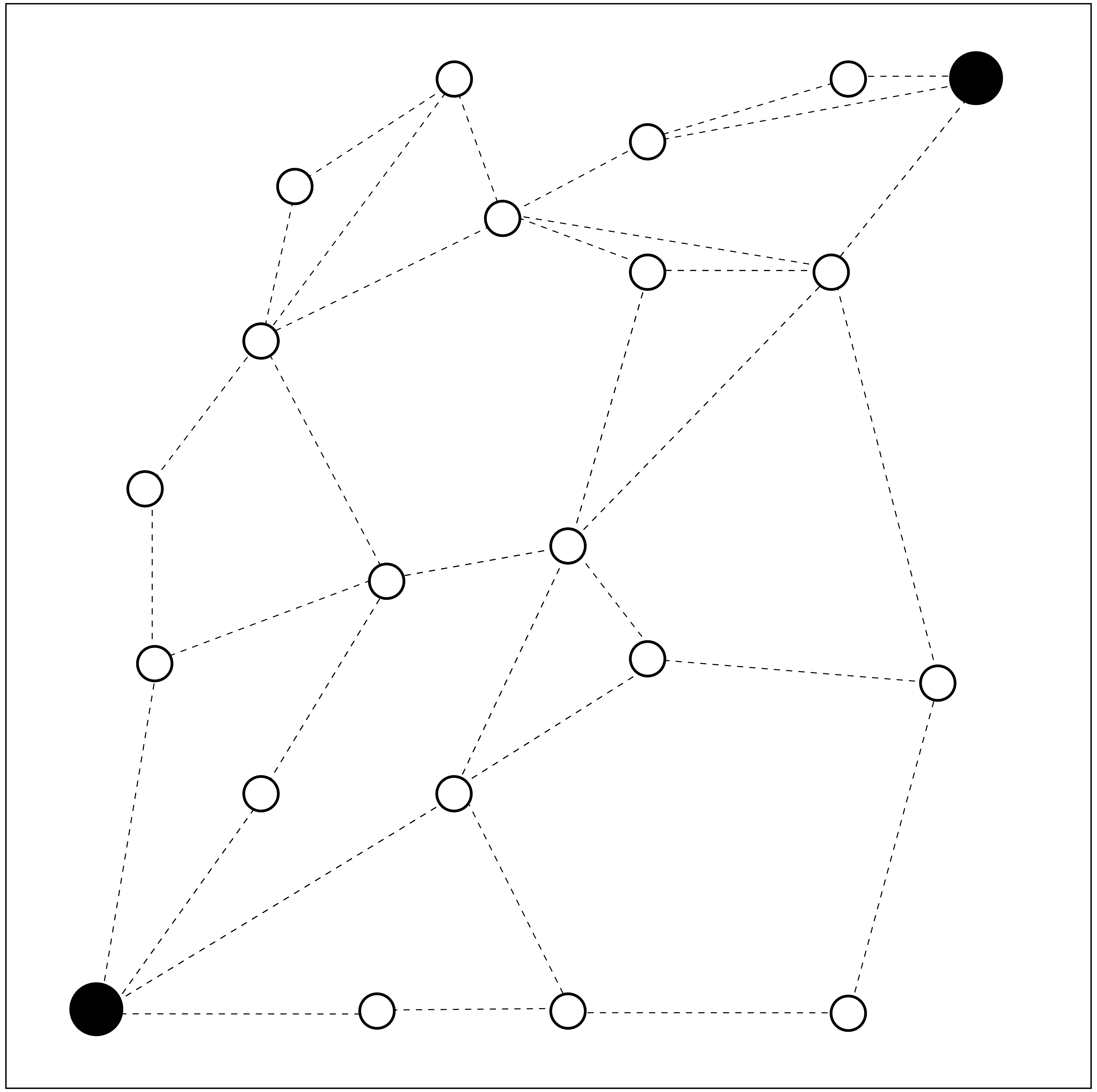
A suspiciously simple algorithm

Repeat forever:

Sample edge times from posterior

Compute shortest path

Travel along path, and update posterior



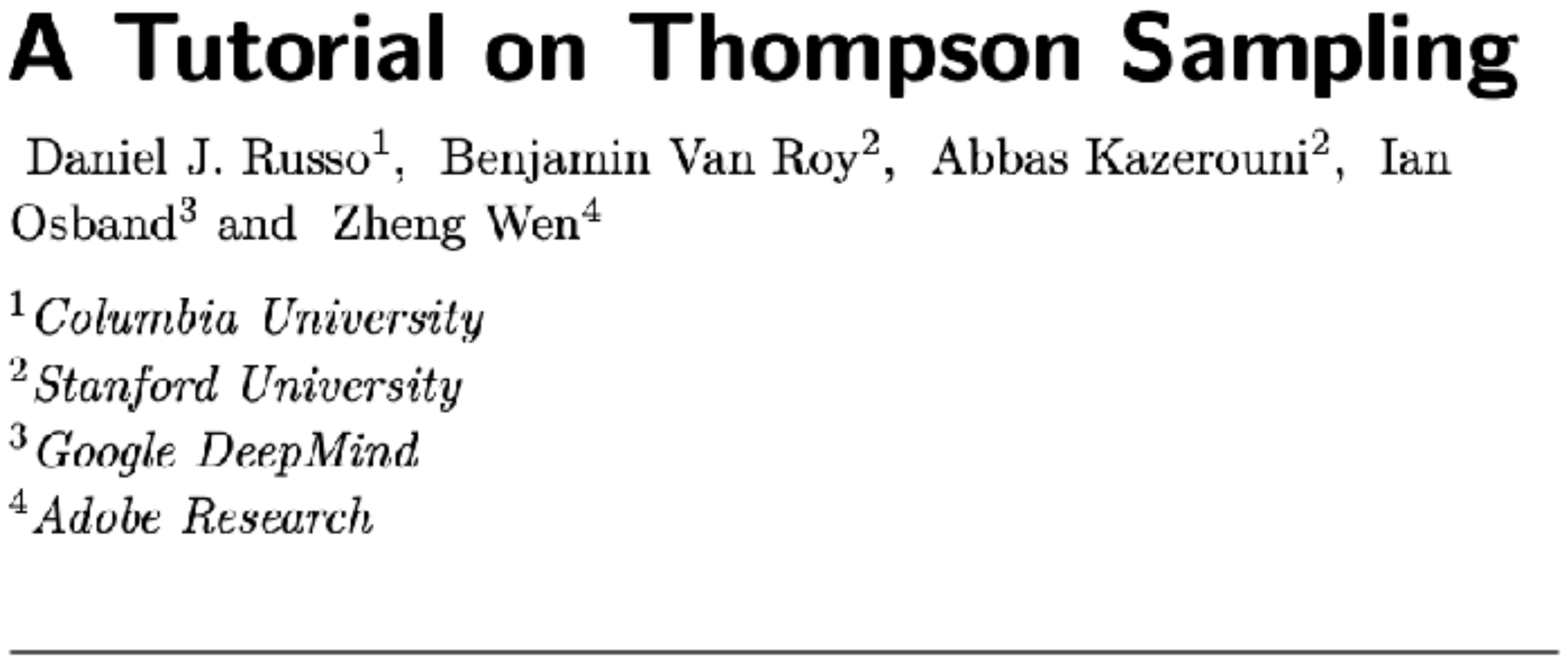
A suspiciously simple algorithm

Repeat forever:

    Sample model from posterior

    Compute optimal policy

    Execute policy, observe s,a,s',
    Update model

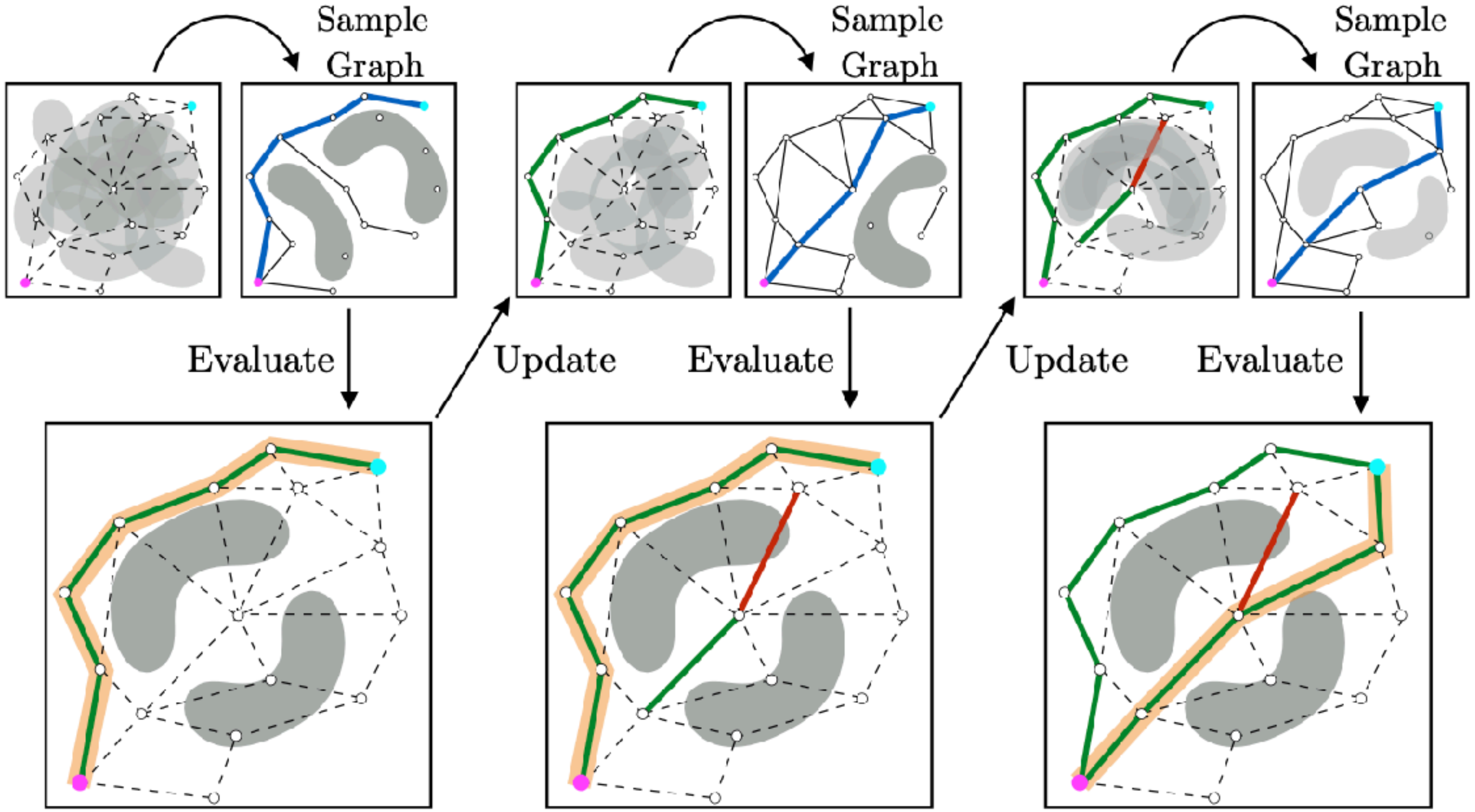


A Tutorial on Thompson Sampling

Daniel J. Russo[1], Benjamin Van Roy[2], Abbas Kazerouni[2], Ian Osband[3] and Zheng Wen[4]

[1] *Columbia University*
[2] *Stanford University*
[3] *Google DeepMind*
[4] *Adobe Research*

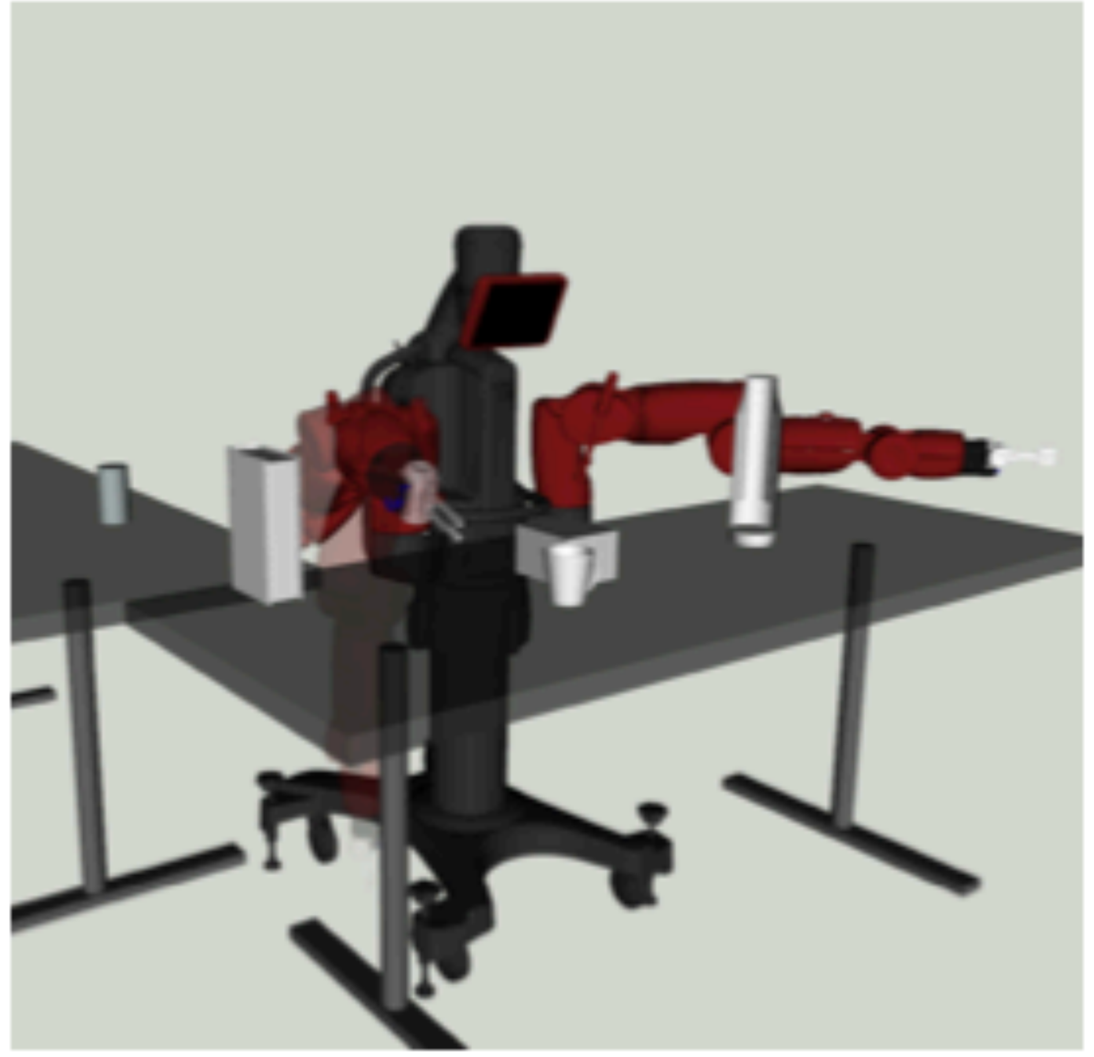
Posterior Sampling for Motion Planning

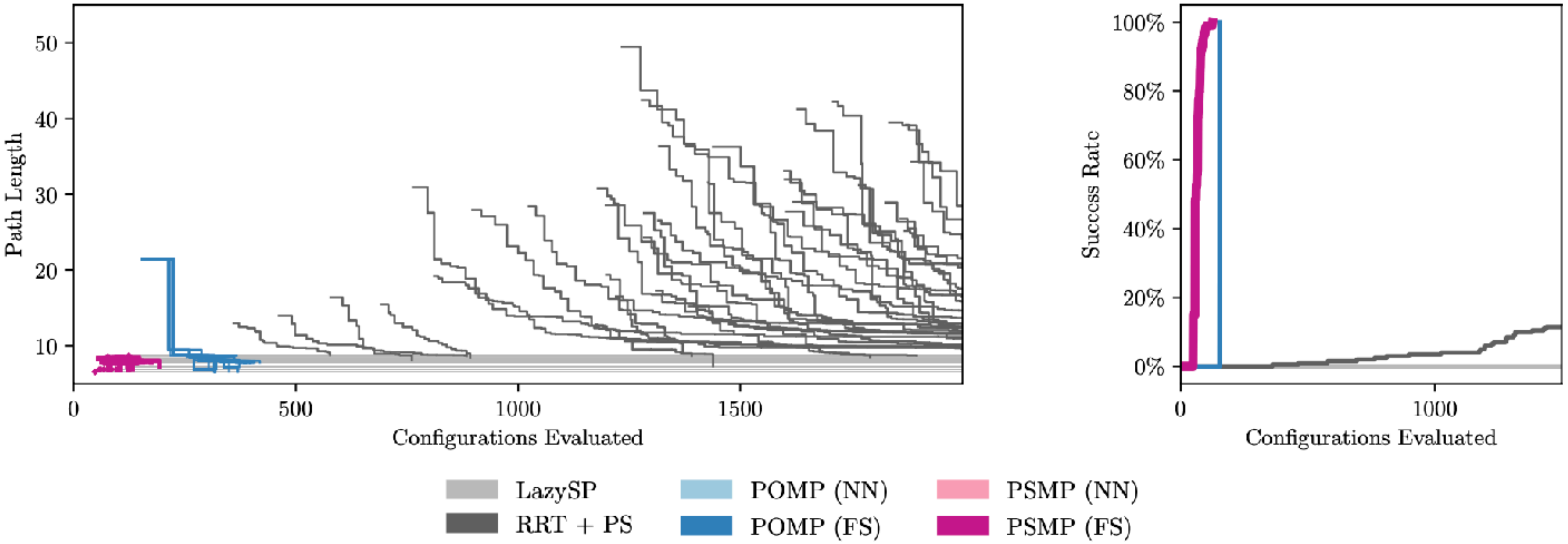


Posterior Sampling for Anytime Motion Planning on Graphs with Expensive-to-Evaluate Edges

Brian Hou, Sanjiban Choudhury, Gilwoo Lee, Aditya Mandalika, and Siddhartha S. Srinivasa

Posterior Sampling for Motion Planning

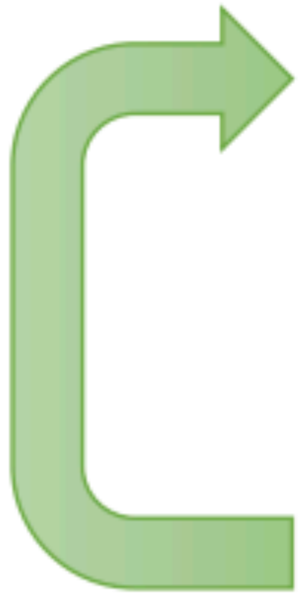




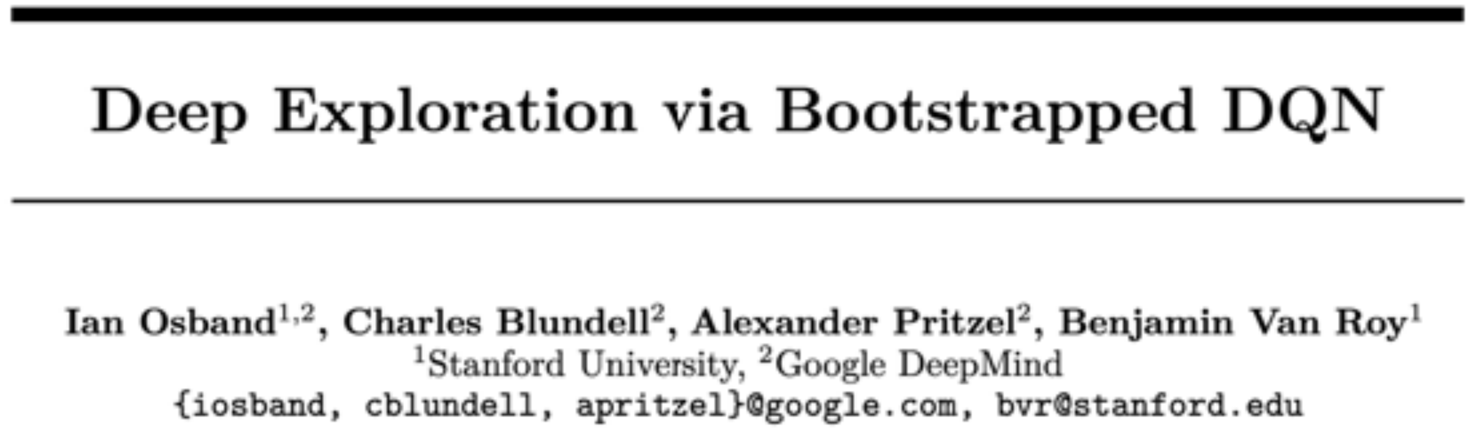
Posterior Sampling for Anytime Motion Planning
on Graphs with Expensive-to-Evaluate Edges

Brian Hou, Sanjiban Choudhury, Gilwoo Lee, Aditya Mandalika, and Siddhartha S. Srinivasa

Posterior Sampling for Reinforcement Learning

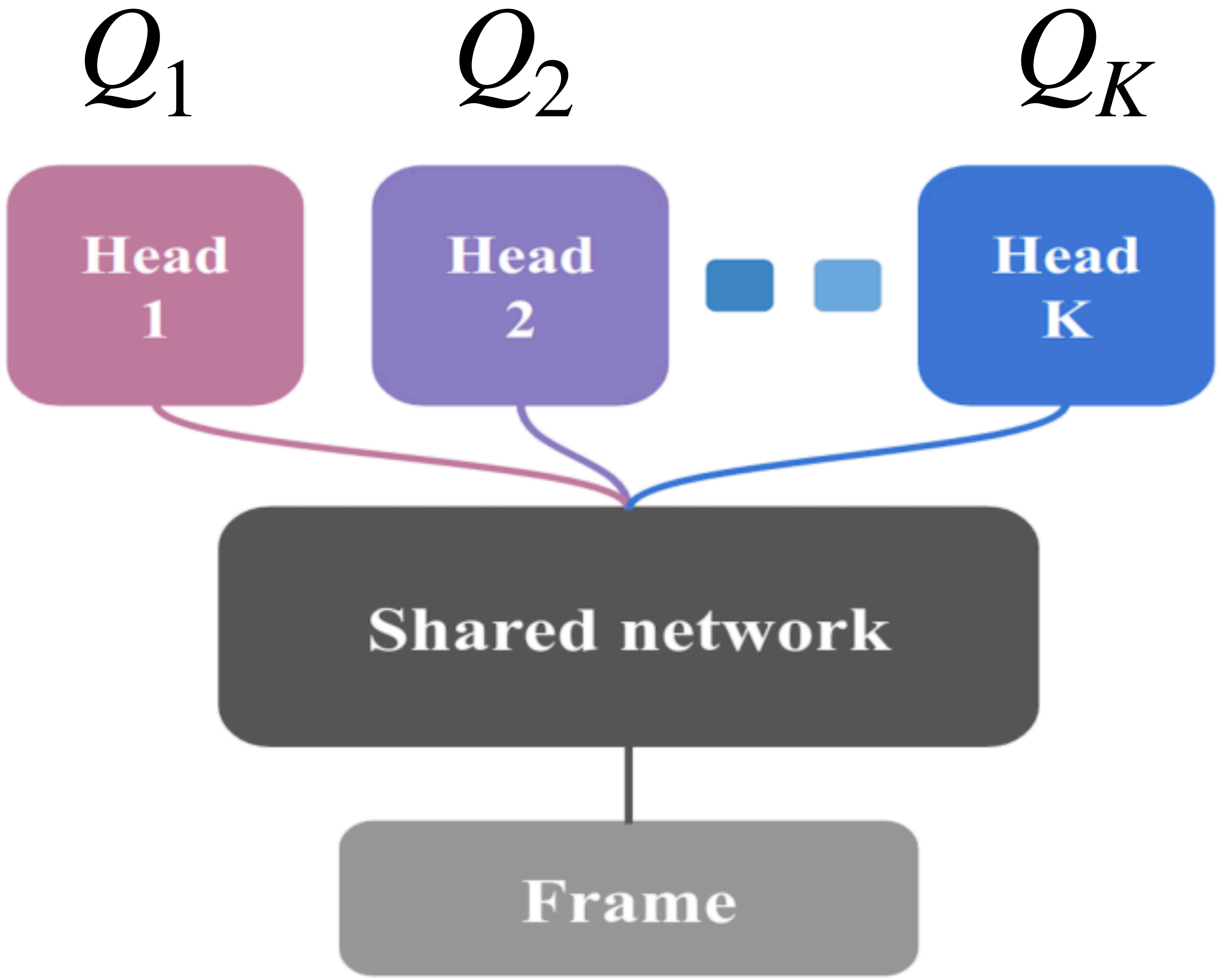


1. sample Q-function Q from $p(Q)$
2. act according to Q for one episode
3. update $p(Q)$



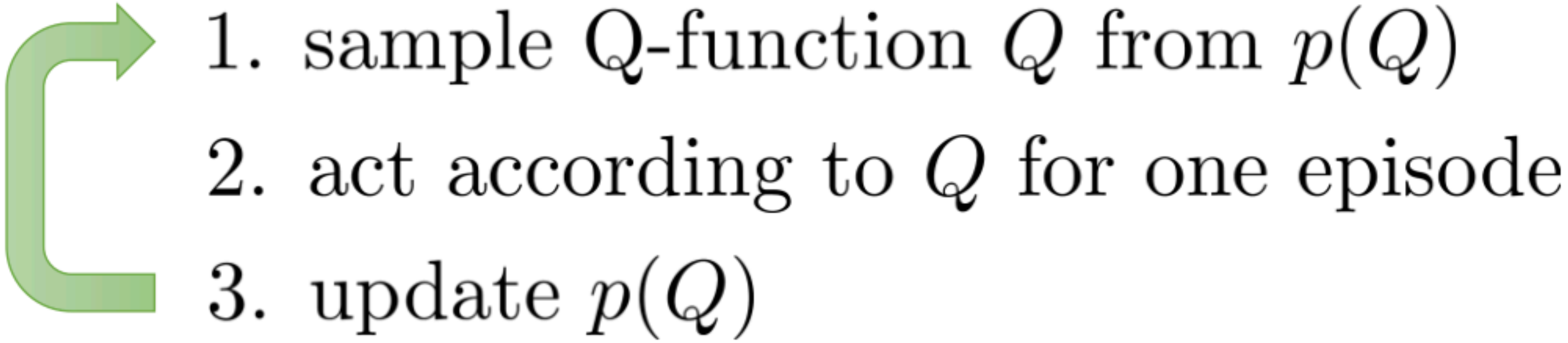
Deep Exploration via Bootstrapped DQN

Ian Osband[1,2], Charles Blundell[2], Alexander Pritzel[2], Benjamin Van Roy[1]
[1]Stanford University, [2]Google DeepMind
{iosband, cblundell, apritzel}@google.com, bvr@stanford.edu



Bootstrapped Q Network

Posterior Sampling for Reinforcement Learning



1. sample Q-function Q from $p(Q)$
2. act according to Q for one episode
3. update $p(Q)$

Atari

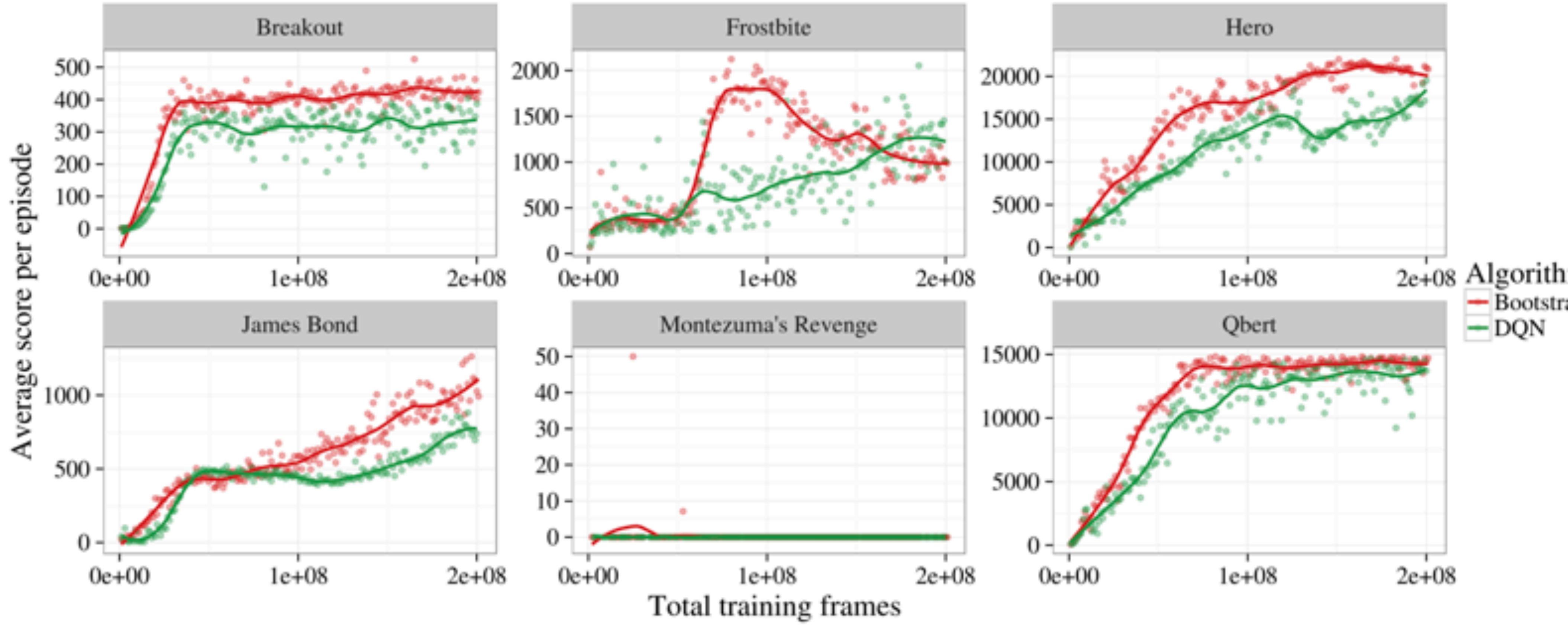
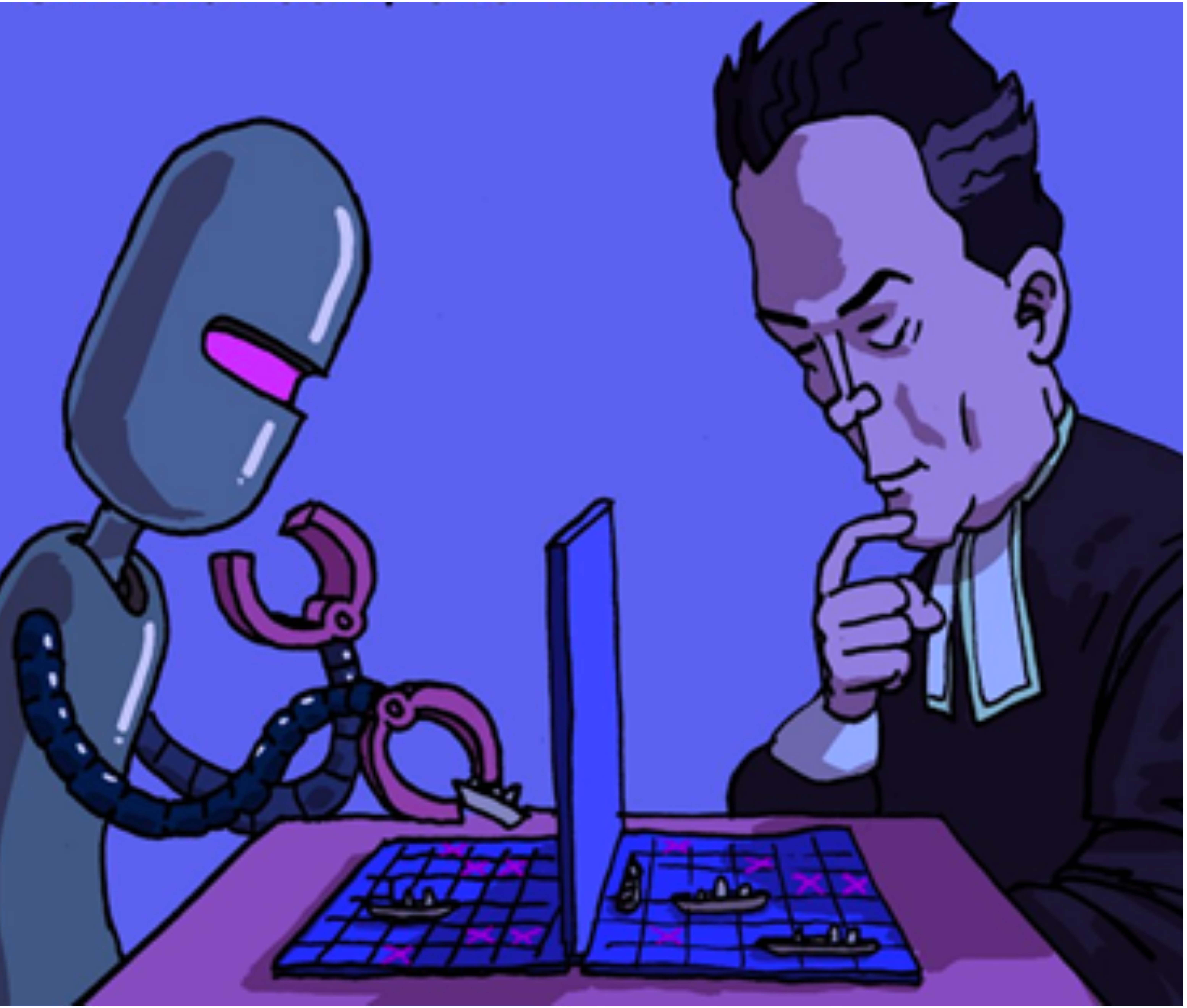


Figure 6: Bootstrapped DQN drives more efficient exploration.

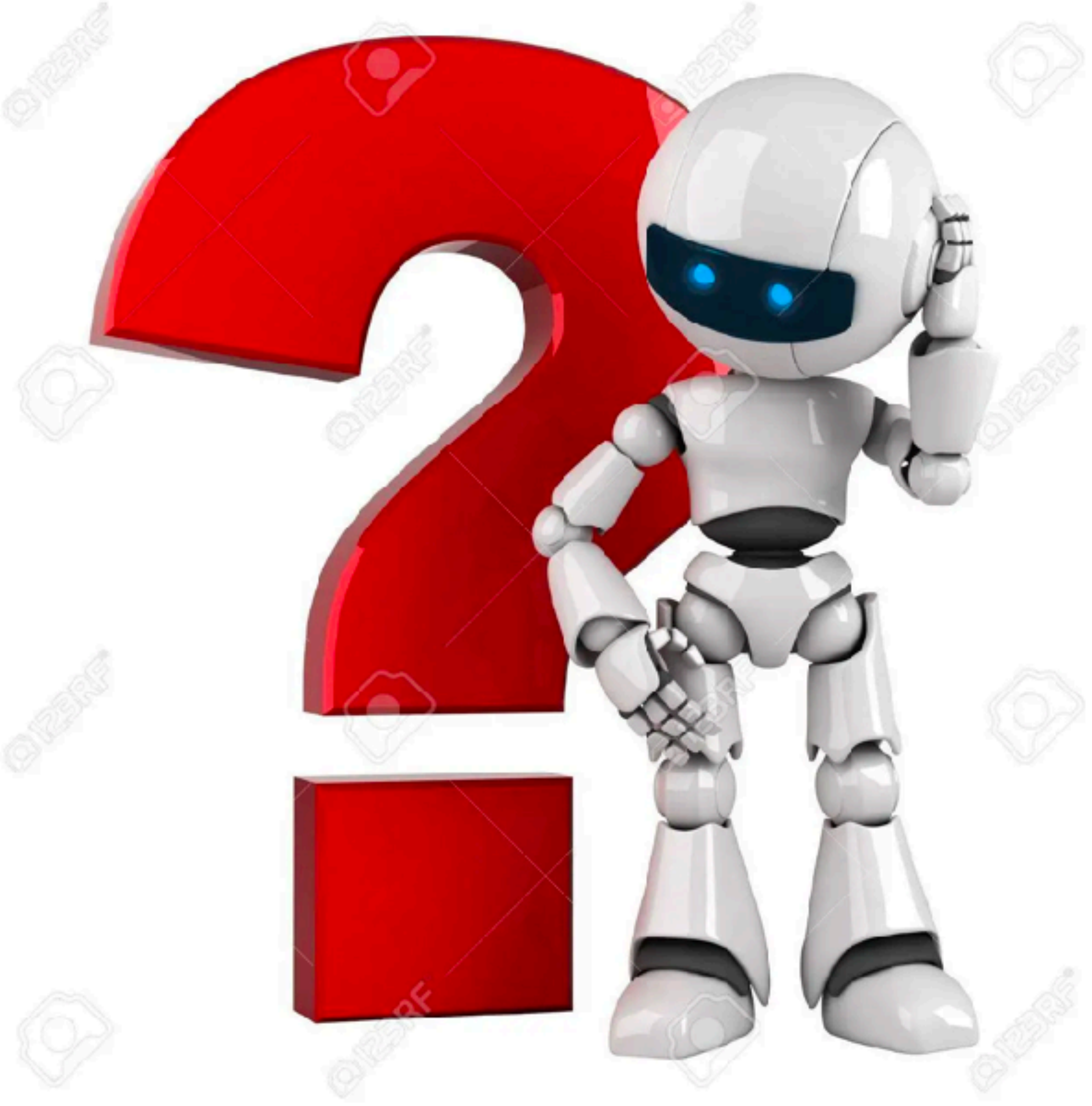
Why does work better than taking random actions?

What if we wanted to explore as optimally as possible using prior information?



Information Gain

20 Questions



Let's say you have a set of hypotheses

$$\{\theta_1, \theta_2, \ldots, \theta_n\}$$

and a set of tests

$$\{t_1, t_2, \ldots, t_n\}$$

Given a prior over hypotheses $P(\theta)$

Find the minimal number of tests to identify hypothesis

20 Questions

Let's say you have a set of hypotheses

$$\{\theta_1, \theta_2, \ldots, \theta_n\}$$

and a set of tests

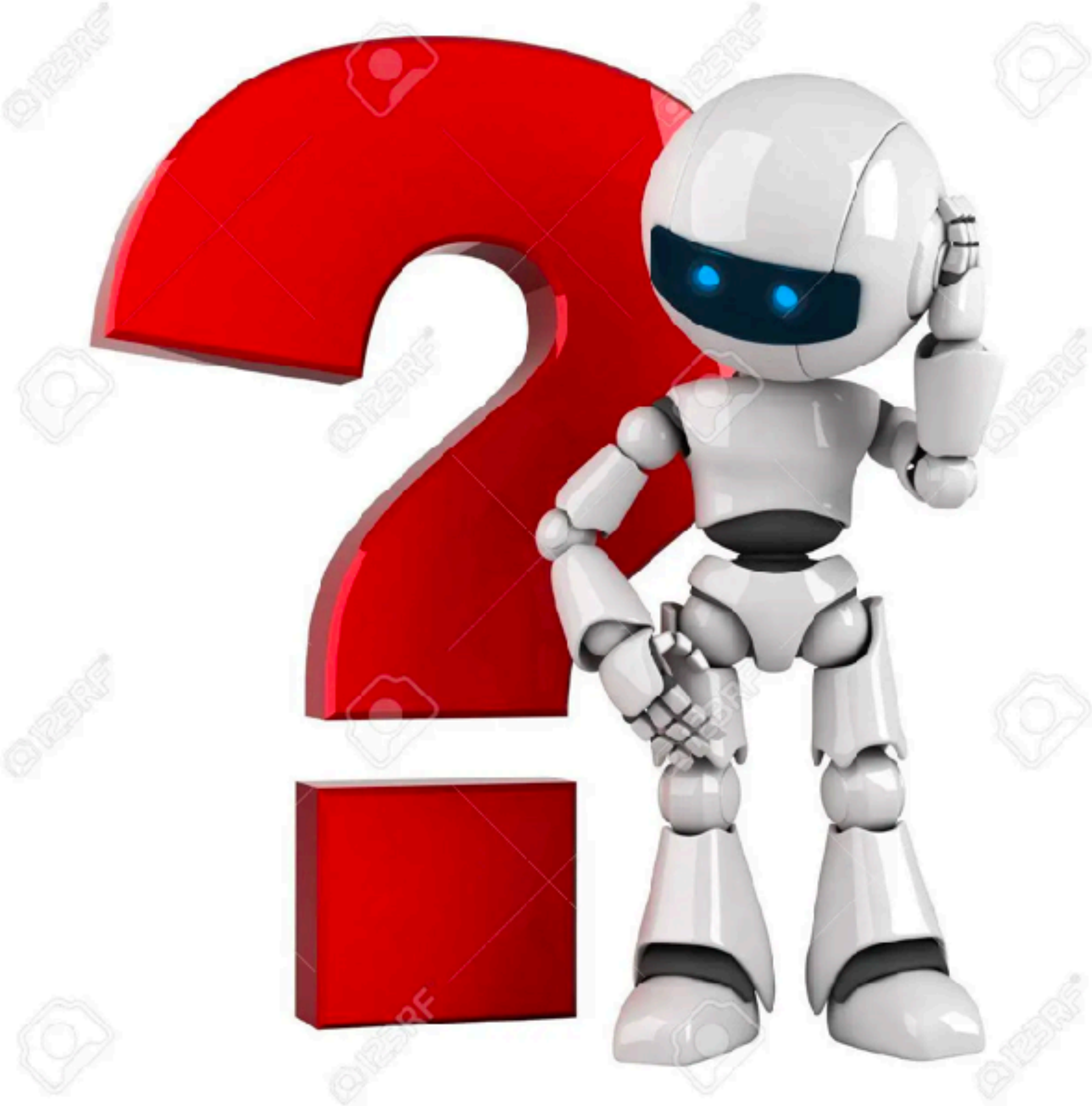
$$\mathcal{T} = \{1, \ldots, N\}$$

NP-HARD

Given a prior over hypotheses $P(\theta)$

Find the minimal number of tests to identify hypothesis

A simple algorithm



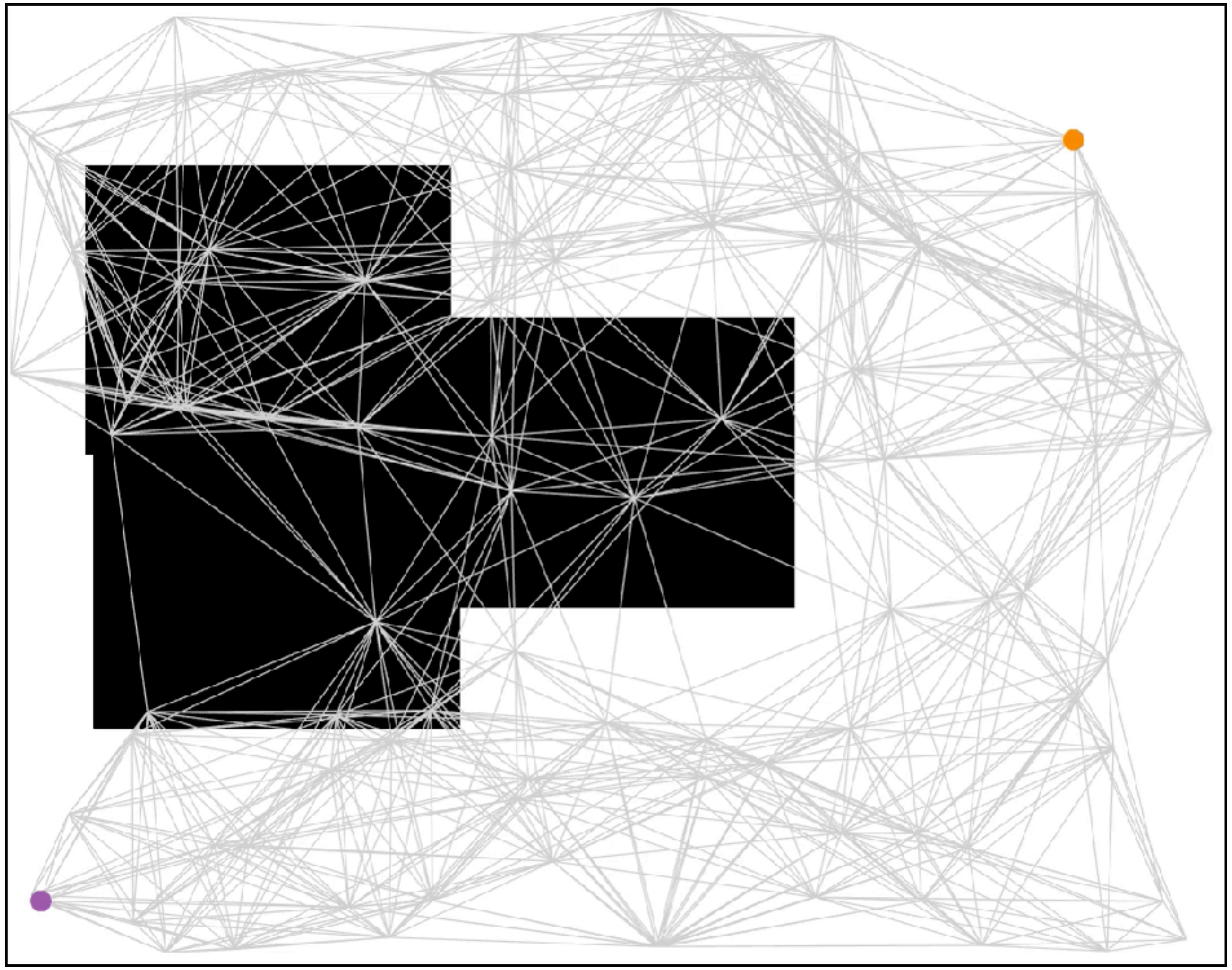
Greedily pick the test that maximizes information gain

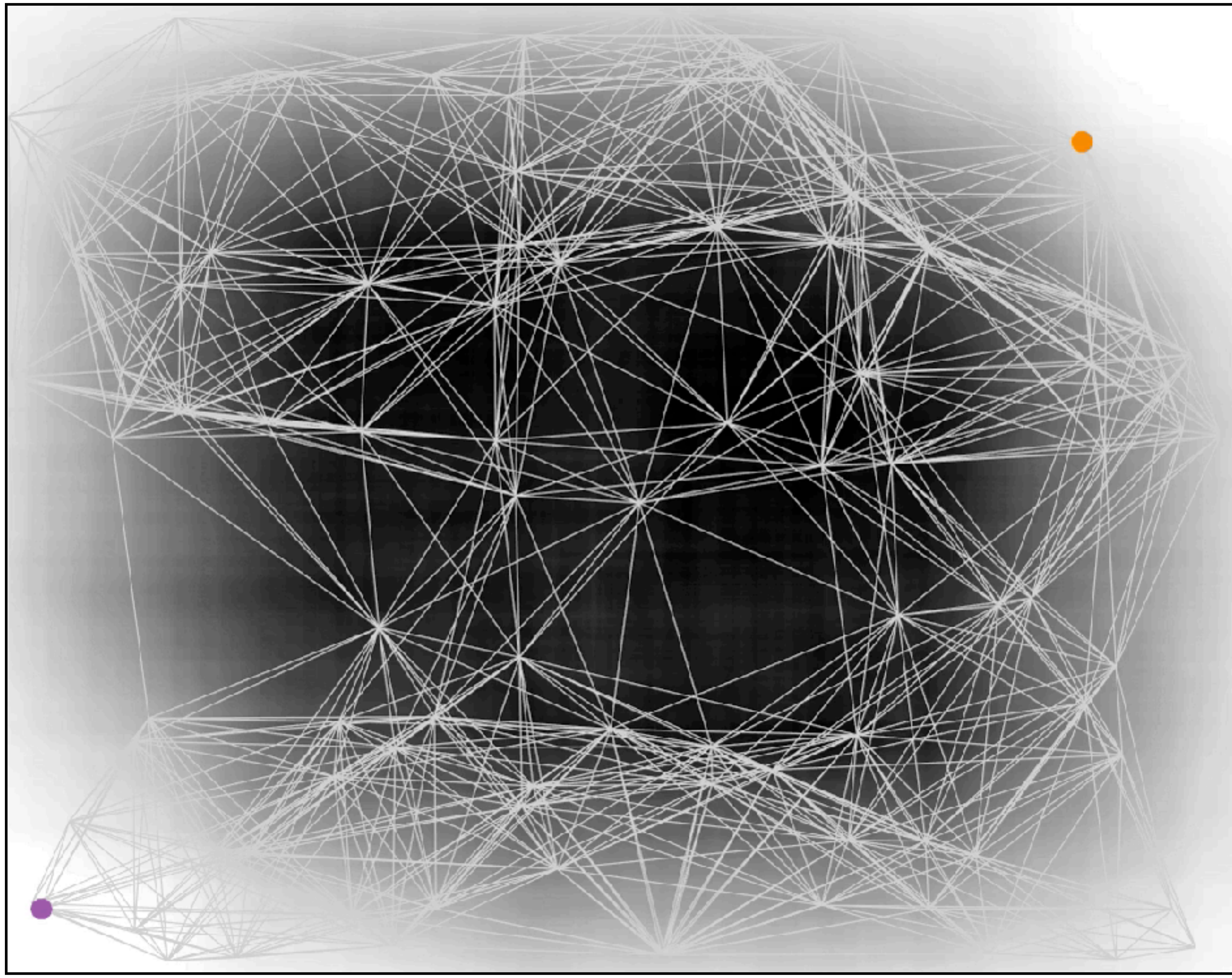
$$\underset{t}{\max}\, \underbrace{H(\theta)}_{\text{Entropy}} - \underbrace{\mathbb{E}_o H(\theta \mid t, o)}_{\text{Posterior entropy}}$$

This is near-optimal!

Optimal edge evaluation for shortest path

[CJS+ NeurIPS'17] [CSS IJCAI'18]

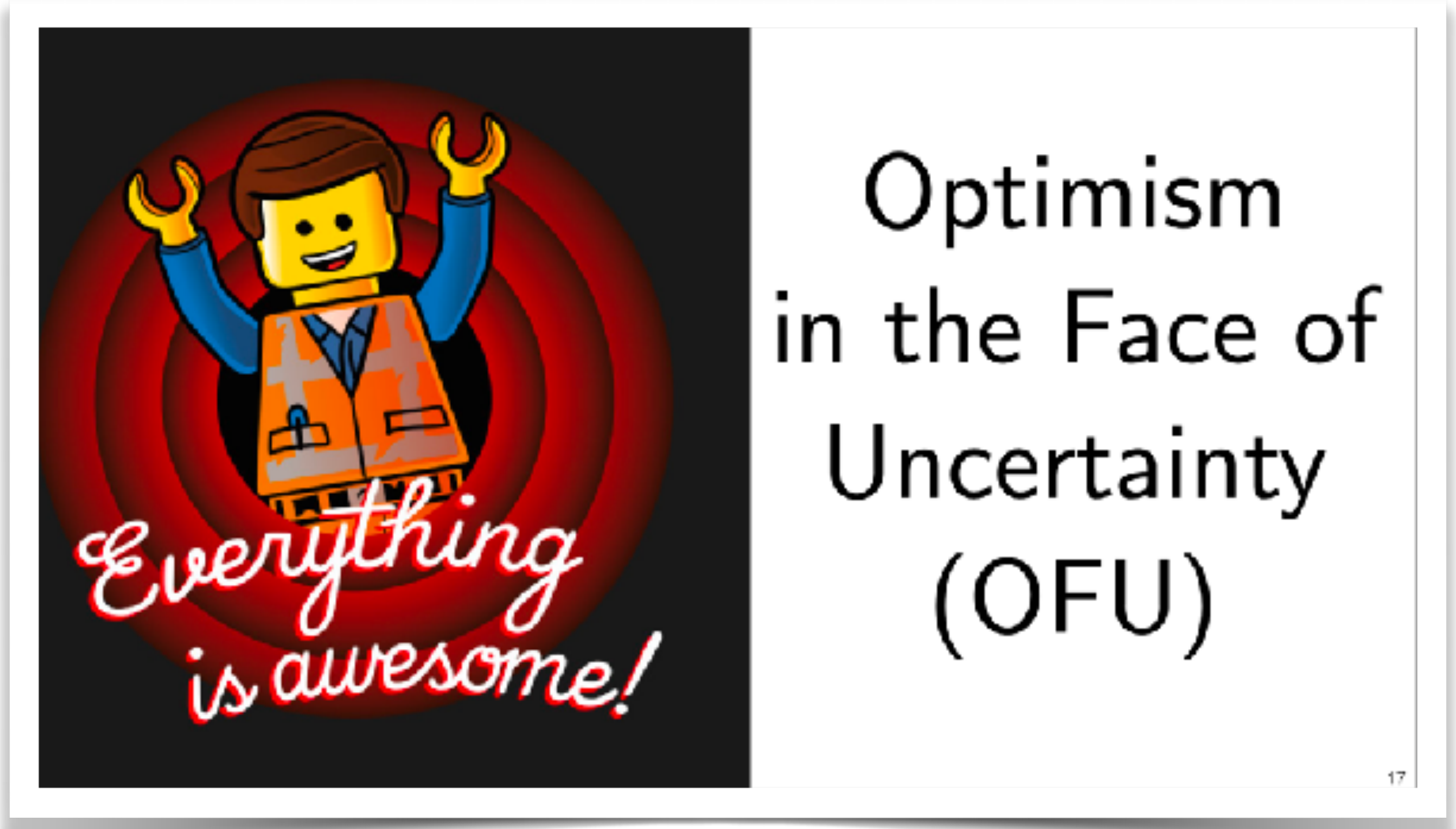




tl;dr

Belief Space Planning is NP-Hard at best, undecidable at worst

Need to relax our problem!



Optimism in the Face of Uncertainty (OFU)

Posterior Sampling

Information Gain