Course 2, Module 4
Planning, Learning & Acting

CMPUT 397
Fall 2020

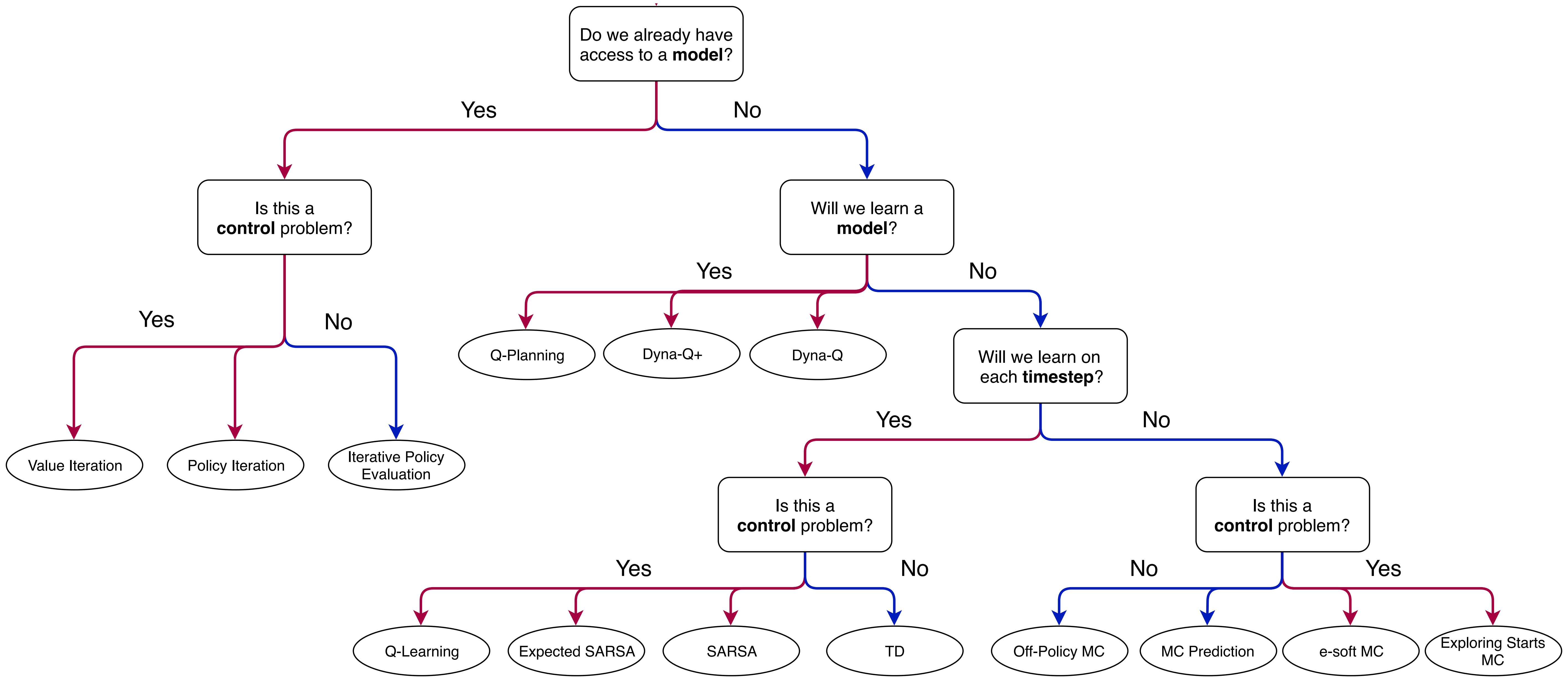
Comments: Oct. 26, 2020

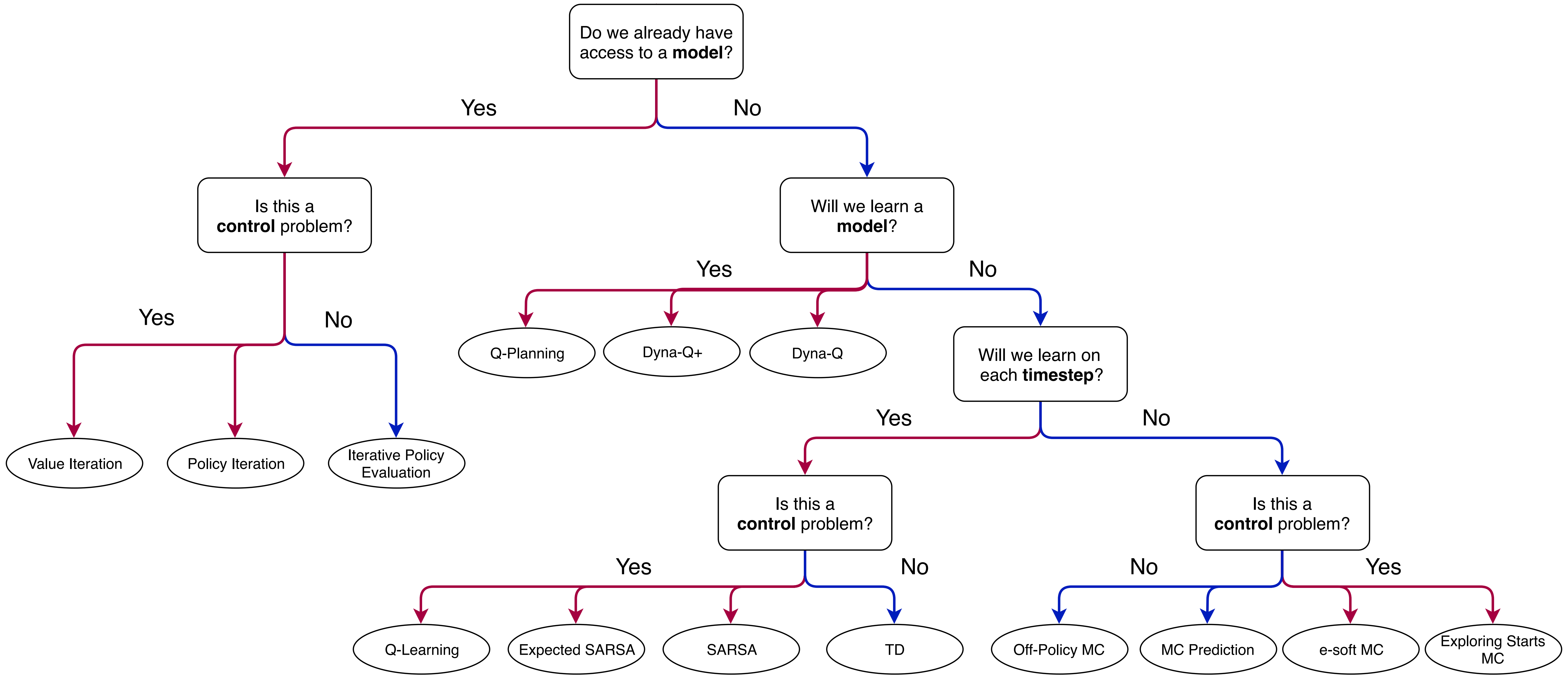
- Course 3 and Course 4 links available
- Do get started on your project soon
- Discussion Session next Wednesday
- Assignment/Quiz review this Friday (come prepared with questions)

So many algorithms! What is a student to do?

- Introducing the **Course Map**

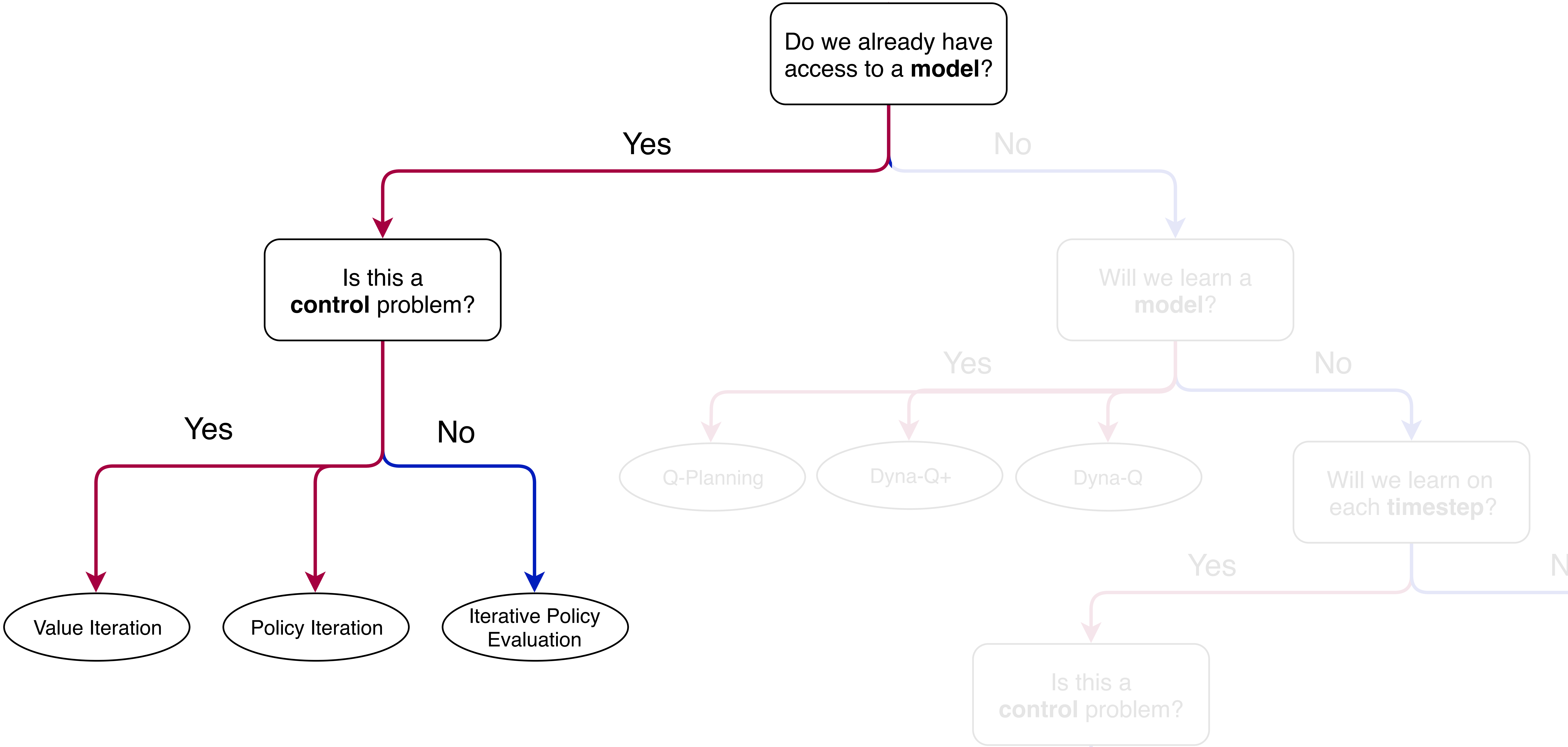

Can we represent the value function with a **table**?
No
Yes
Are we using **average reward**?
Do we already have access to a **model**?
No
Yes
Yes
No
Will we learn on each **timestep**?
Are the actions **continuous**?
Is this a **control** problem?
Will we learn a **model**?
Yes
No
Yes
No
Yes
No
Yes
No
Is this a **control** problem?
Gradient Monte Carlo
Gaussian Actor-Critic
Softmax Actor-Critic
Differential Semi-Gradient SARSA
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each **timestep**?
No
Yes
Semi-Gradient TD
Expected SARSA
Q-Learning
SARSA
Value Iteration
Policy Iteration
Iterative Policy Evaluation
Yes
No
Is this a **control** problem?
Is this a **control** problem?
Yes
No
No
Yes
Q-Learning
Expected SARSA
SARSA
TD
Off-Policy MC
MC Prediction
e-soft MC
Exploring Starts MC


Do we already have access to a **model**?
Yes
No
Is this a **control** problem?
Will we learn a **model**?
Yes
No
Yes
No
Value Iteration
Policy Iteration
Iterative Policy Evaluation
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each **timestep**?
Yes
No
Is this a **control** problem?
Is this a **control** problem?
Yes
No
No
Yes
Q-Learning
Expected SARSA
SARSA
TD
Off-Policy MC
MC Prediction
e-soft MC
Exploring Starts MC



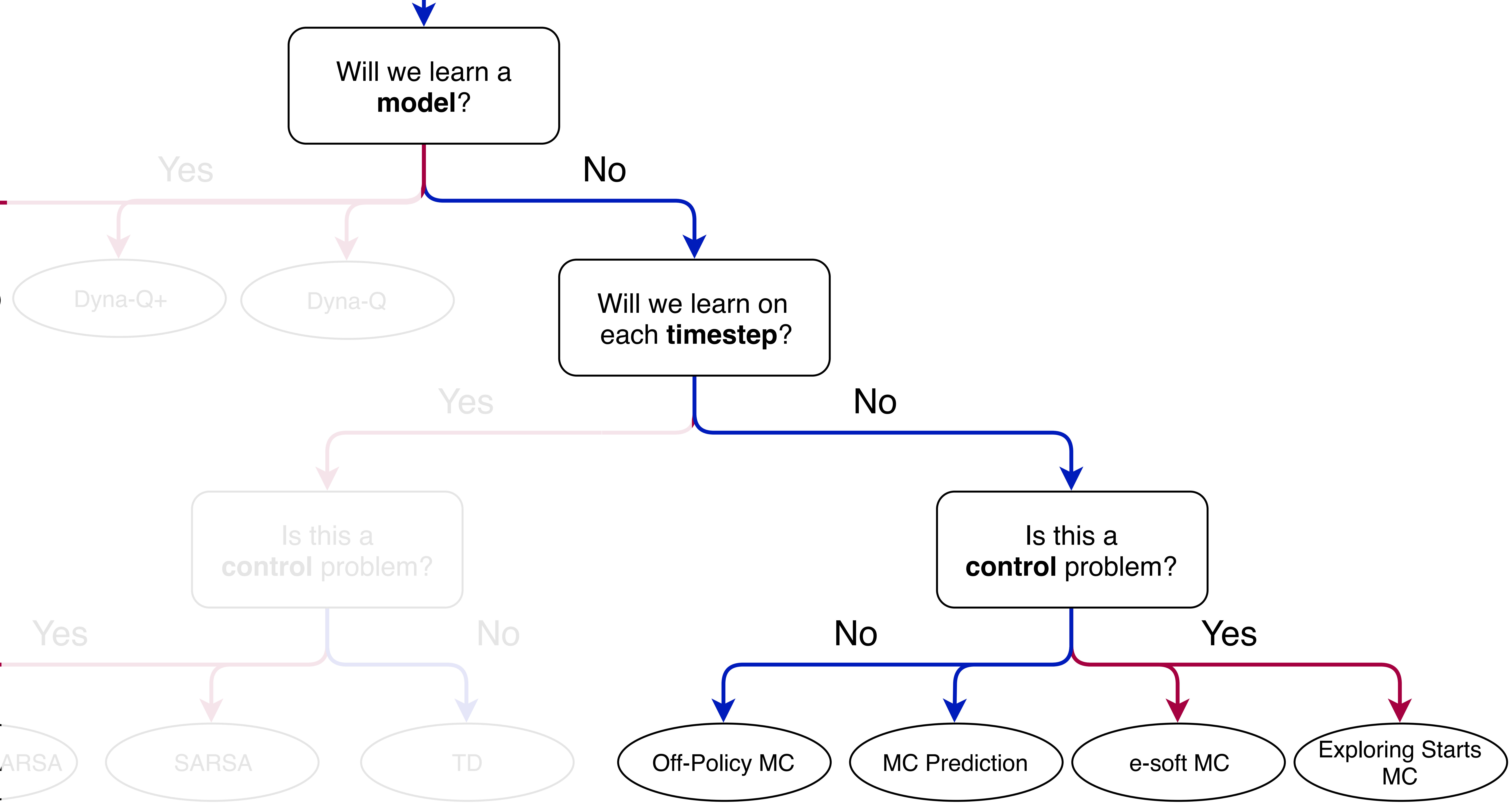
Do we already have access to a **model**?
Yes
No
Is this a **control** problem?
Will we learn a **model**?
Yes
No
Yes
No
Value Iteration
Policy Iteration
Iterative Policy Evaluation
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each **timestep**?
Yes
No
Is this a **control** problem?
Is this a **control** problem?
Yes
No
No
Yes
Q-Learning
Expected SARSA
SARSA
TD
Off-Policy MC
MC Prediction
e-soft MC
Exploring Starts MC


Do we already have access to a **model**?
Yes
No
Is this a **control** problem?
Will we learn a **model**?
Yes
No
Yes
No
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each **timestep**?
Value Iteration
Policy Iteration
Iterative Policy Evaluation
Yes
No
Is this a **control** problem?
Is this a **control** problem?
Yes
No
No
Yes
Q-Learning
Expected SARSA
SARSA
TD
Off-Policy MC
MC Prediction
e-soft MC
Exploring Starts MC



Do we already have access to a **model**?
Yes
No
Is this a **control** problem?
Yes
No
Value Iteration
Policy Iteration
Iterative Policy Evaluation
Will we learn a **model**?
Yes
No
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each **timestep**?
Yes
Is this a **control** problem?


Do we already have access to a **model**?
Yes
No
Is this a **control** problem?
Will we learn a **model**?
Yes
No
Yes
No
Value Iteration
Policy Iteration
Iterative Policy Evaluation
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each **timestep**?
Yes
No
Is this a **control** problem?
Is this a **control** problem?
Yes
No
No
Yes
Q-Learning
Expected SARSA
SARSA
TD
Off-Policy MC
MC Prediction
e-soft MC
Exploring Starts MC


Will we learn a **model**?
Yes
No
Dyna-Q+
Dyna-Q
Will we learn on each **timestep**?
Yes
No
Is this a **control** problem?
Yes
No
ARSA
SARSA
TD
Is this a **control** problem?
No
Yes
Off-Policy MC
MC Prediction
e-soft MC
Exploring Starts MC


Is this a
control problem?
No
icy Iteration
Iterative Policy
Evaluation
Will we learn a
model?
Yes
No
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on
each **timestep**?
Yes
No
Is this a
control problem?
Yes
No
Q-Learning
Expected SARSA
SARSA
TD
Is this a
control problem?
No
Yes
Off-Policy MC
MC Prediction
e-soft MC

Do we already have access to a **model**?
Yes
No
Is this a **control** problem?
Will we learn a **model**?
Yes
No
No
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each **timestep**?
Policy Iteration
Iterative Policy Evaluation
Yes
No
Is this a **control** problem?
Is this a **control** problem?
Yes
No
No
Q-Learning
Expected SARSA
SARSA
TD
Off-Policy MC
MC Prediction

Review of Course 2, Module 4
Learning a Model AND Planning

Video 1: What is a Model?

- **All about models.** What they are? How they might be useful? How would we use one if we had it?
- Goals:
  - Describe a model and how it can be **used**.
  - Know the different model types: **distribution** models or **sample** models
  - identify when to use a distribution model or sample model

Video 2: Comparing Sample and Distribution Models

- If you could have either, which one might you **prefer**? It depends ....
- Goals:
  - Describe the **advantages** and **disadvantages** of sample models and distribution models
  - explain why sample models can be represented more **compactly** than distribution models.

Terminology Review

- **Model:** a model of the environment. Anything that can predict how the environment will respond to the agent's actions: M(S,A) --->S',R

- **Planning**: the computational process that takes the model as input and produces or improves the policy

- **Sample Model**: a model that can produce a possible next state and reward, in agreement with the underlying transition probabilities of the world. We need not store all the probabilities to do this (think about epsilon-greedy)

- **Simulate:** sample a transition from the model. Given an S and A, ask the model for a possible next state S' and reward R

- **Simulated Experience:** samples generated by a sample model. Like dreaming or imagining things that could happen

- **Real Experience:** the states, actions, and rewards that are produced when an agent interacts with the real world.

- **Search Control:** the computational process that selects the state and action in the planning loop

Video 3: Random Tabular, Q-planning

- **A simple planning method.** Assumes access to a sample model. Does Q-learning updates
- Goals:
  - **You will be able to explain how planning is used to improve policies**
  - And describe one-step tabular Q-planning

One-step Tabular Q-planning

Random-sample one-step tabular Q-planning

Loop forever:

1. Select a state, $S \in \mathcal{S}$, and an action, $A \in \mathcal{A}(S)$, at random
2. Send S, A to a sample model, and obtain
   a sample next reward, R, and a sample next state, S'
3. Apply one-step tabular Q-learning to S, A, R, S':
   $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$

Question: How does this differ from DP, specifically Value Iteration?

Question: What change makes it more similar to Value Iteration?

Going beyond a given model

- Q-planning does not interact with the world
- It simply computes a policy, by directly querying the given model
- It is an important step since it makes it more clear how to use a sample model
- Next step: do not assume you are given a model

Video 4: The Dyna Architecture

- **Introducing Dyna!** An architecture that mixes (1) learning a model, (2) updating the value function and policy as usual, and (3) planning
- Goals:
  - understand how **simulating experience** from the model **differs** from i**nteracting with the environment.**
  - You will also understand how the **Dyna** architecture **mixes direct RL** updates, and **planning** updates.

Video 5: The Dyna Algorithm

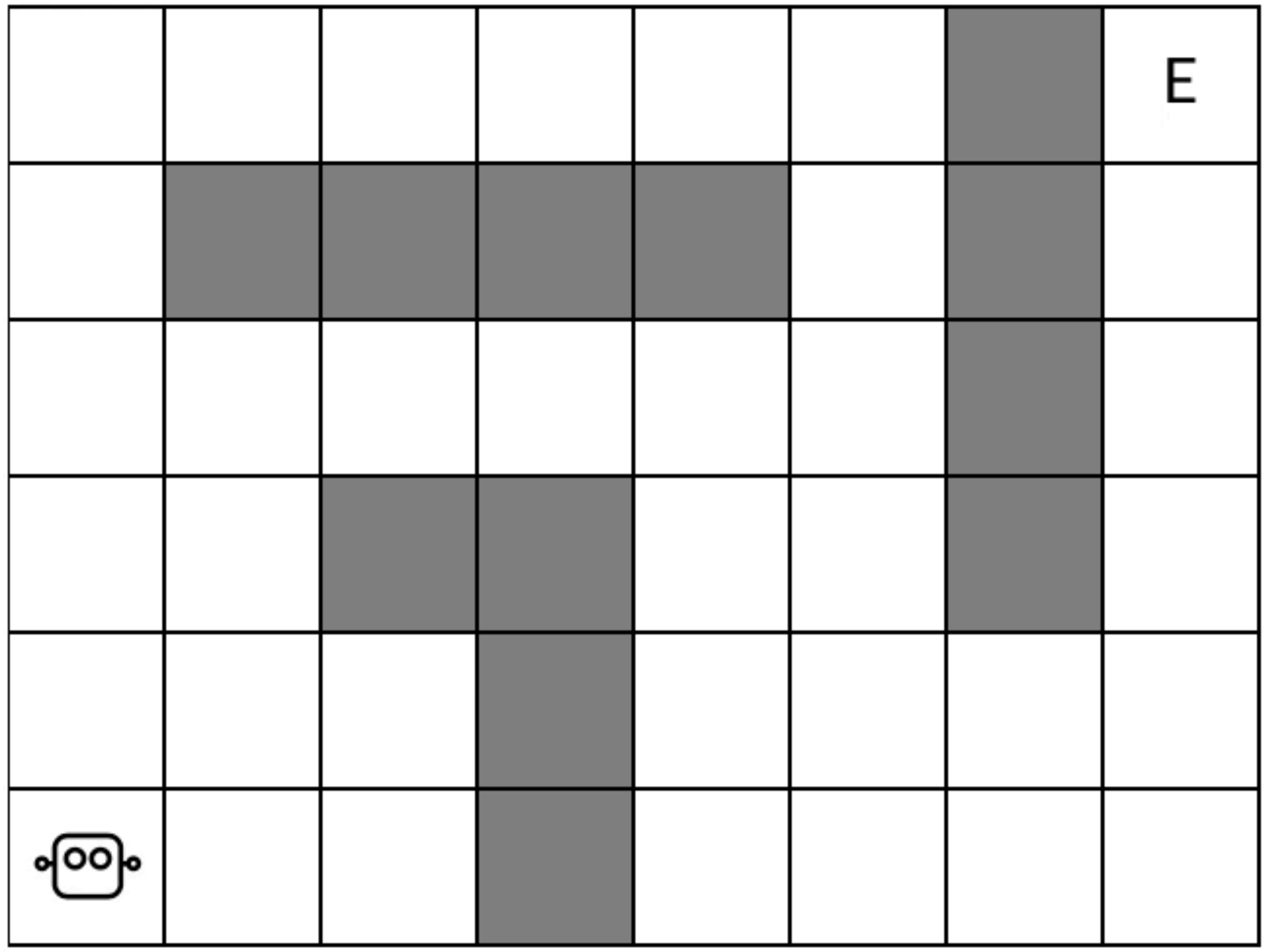
- The details about one implementation of the Dyna Architecture: **Dyna-Q**
- Goals:
  - Describe how Tabular Dyna-Q works.
  - **Identify** the direct RL, planning, model learning, and search control parts of Dyna-Q.

Tabular Dyna-Q

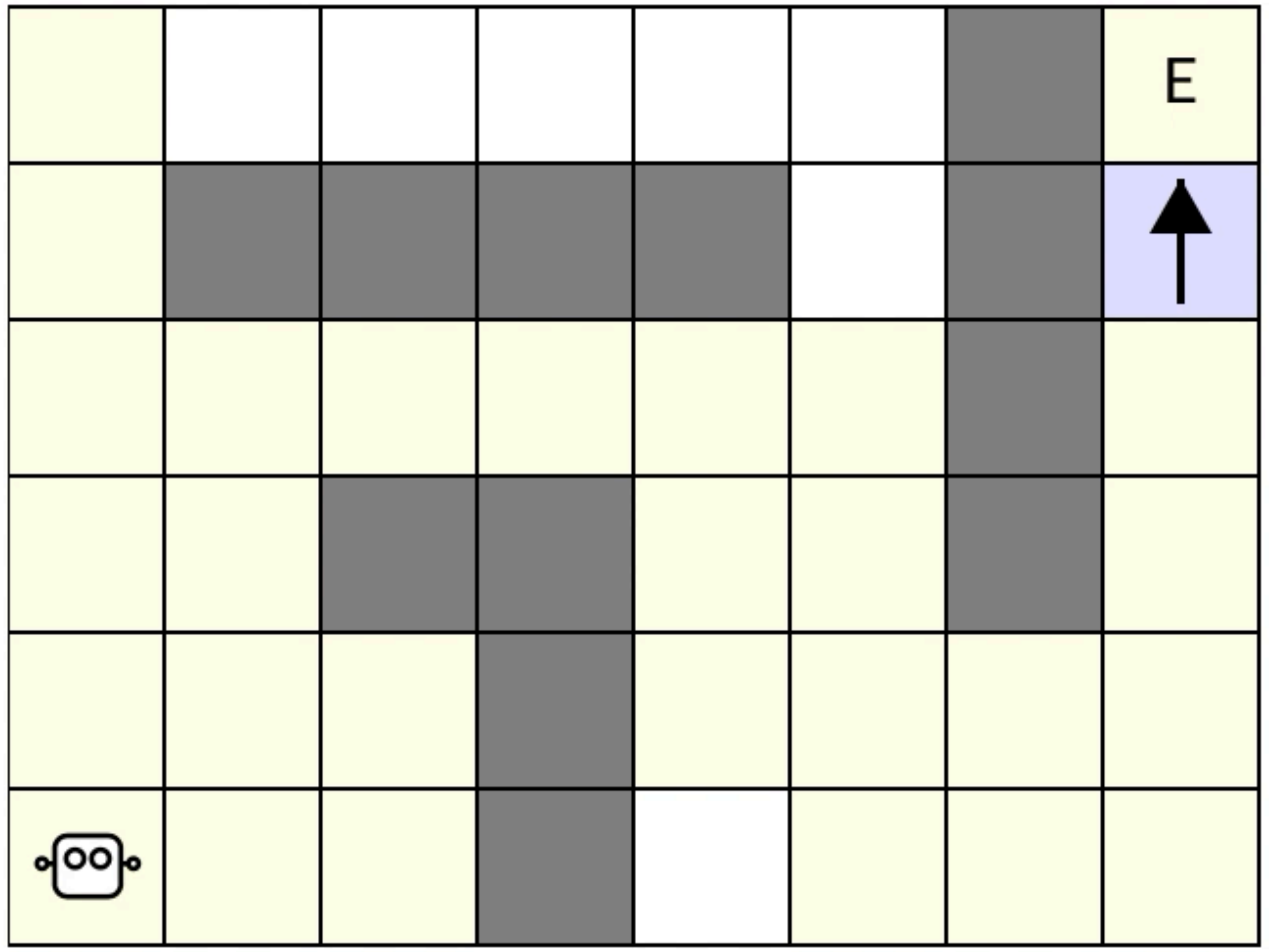
Video 6: Dyna & Q-learning in a Simple Maze

- Use a small gridworld to compare Tabular Dyna-Q and model-free Q-learning. **We run an experiment!**
- Goals:
  - describe how learning from both real and model experience impacts performance
  - explain how a model allows the agent to learn from **fewer interactions** with the environment.

Number of actions taken: 0

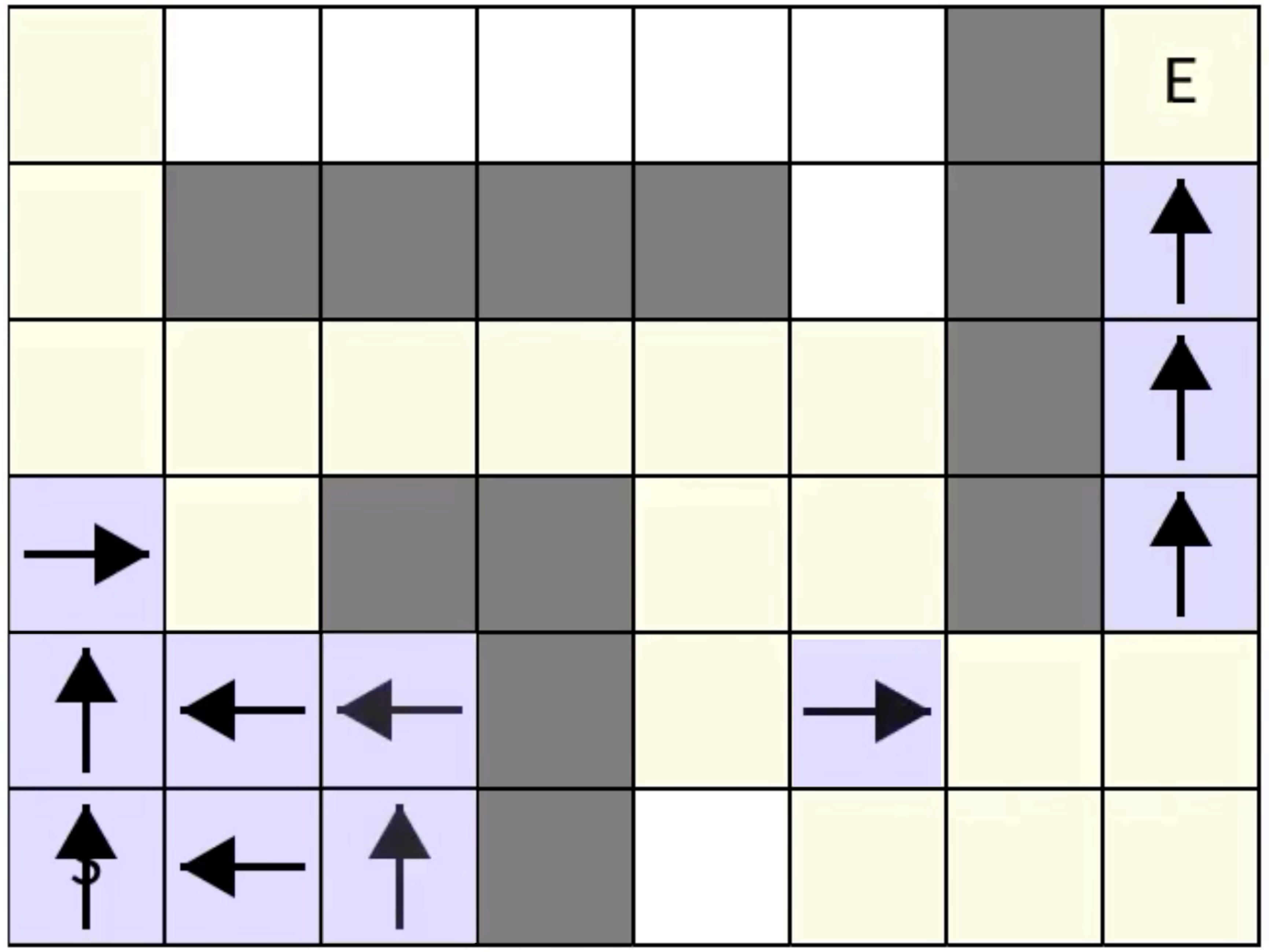


Number of actions taken: 184



Number of steps planned: 100

Number of actions taken: 185



Video 7: What if the model is inaccurate?

- How do we handle if the model is **wrong in some way**? How could that happen? What would be the impact of trying to plan with an inaccurate model?
- Goals:
  - Identify **ways** in which models can be inaccurate,
  - Explain the **effects** of planning with an inaccurate model
  - Describe how Dyna can plan successfully with an incomplete model

Video 8: In-depth with changing environments

- We focus on a specific way the model can be inaccurate: **the world changes** and the model is out of date. **New Algorithm!**
- Goals:
  - Explain how model inaccuracies produce **another exploration-exploitation trade-off**
  - Describe how **Dyna-Q+** addresses this trade-off.

Slido Q about Dyna-Q vs Dyna-Q+

- “Does Dyna-Q+ always have better performance than Dyna-Q? If not can you give an example that we would prefer to use Dyna-Q over Dyna-Q+?”

Slido Q: Algorithm Choices

- “How would we determine the optimal amount of steps to take during the planning phase for a given problem?”
  - Related: “In practical applications what usually limits the amount of planning steps that an agent can take? Is the number of planning steps usually preset or does it just go until the agent performs it's next action?”
- “Why we use kappa*sqrt(tau) in Dyna-Q+ instead of simply using kappa*tau? In other word, what is the advantage of using square root in it?”
  - Square root number of

Slido Q: Problem Setting

- “Dyna-Q is shown with episodic problems. Can we use it with continuous problems?”

- “How can we modify the tabular Dyna-Q to solve the stochastic problem?”

    - Worksheet question on this

- “When an environment exists entirely within our computer (so we're not limited by slow "real world" actions), is there any benefit to Dyna-Q over Q-Learning? I'm thinking it could be more computationally efficient to simply generate another episode.” —> This comes down to computational efficiency due to search control

Slido Qs: Dyna Q+

- “Can you go over how the bonus reward encourages the agent to return to previous states?”
- “What is the benefit of Dyna-Q+ trying transitions that have not been done in a long time if they are only being tested on simulated experiences? How does this allow the model to improve if the transition is not tested on real experience and can not capture changes in the real environment?”
- “Doesn't including an exploratory reward in Dyna-Q+ go against the idea that one shouldn't encode behaviors into the reward function?”

Why does Dyna Q+ encourage exploration?

What if instead we just used the bonus in action selection?

$$Q(S_t, a) + \kappa\sqrt{\tau(S_t, a)}$$

Greedy wrt to this value+bonus in step (b)

Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:
(a) $S \leftarrow$ current (nonterminal) state
(b) $A \leftarrow \varepsilon$-greedy(S, Q)
(c) Take action A; observe resultant reward, R, and state, S'
(d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
(e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
(f) Loop repeat n times:
$S \leftarrow$ random previously observed state
$A \leftarrow$ random action previously taken in S
$R, S' \leftarrow Model(S,A)$
$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

Slido Qs: Search control

- “For the maze example, our method of search control was randomly selecting a state-action pair. Would it not be more efficient to focus on state-action pairs near those where there was a nonzero reward? Is there an optimal search control algorithm for this example?”

Slido Qs: Misc/Advanced

- “Are these called tabular methods because the model is a table of s, a, s' and r? Or what does tabular mean/refer to?”
  - The action values are stored as a table, and so is the model
- “How do you prevent the use of inaccurate models in Dyna-Q? Or is it rather better to experience it's use?”
- “Can we use something MC to do our model updates in dynaQ? I ask because MC can be faster and give us better results with more data since we're simulating steps already can MC give better performance than Q-learning when doing the model updates?”

Slido Qs: Issues with Dyna

- “Is there ever a danger of Dyna-Q having such a large state space that it explores constantly, and takes a long time to explore, that it's effectiveness is reduced? Especially if the environment changes very quickly, it seems like Dyna-Q would end up either without a valid model, or always exploring”
- “What are the risks of r + kappa * √tau providing a larger reward than the goal state? Is the point to draw the agent away from the goal state is favour of exploration?”