Admin

- We are marking the mini-essay’s now
- Midterm is next month; all questions will be different from previous years
- Practice midterm is on Eclass
  - We will solve it during in class with you

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?”
- Practice question: what part of the Dyna agent makes the model more accurate: real experience or simulated experience?

Problem Setting

- “Dyna-Q is shown with episodic problems. Can we use it with continuous problems?”
    - Does Q-learning work with **continuing** problems.
- “How can we modify the tabular Dyna-Q to solve the stochastic problem?”
    - We will do that today!
- “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
- Practice question: How are continuing problems different from continuous problems?

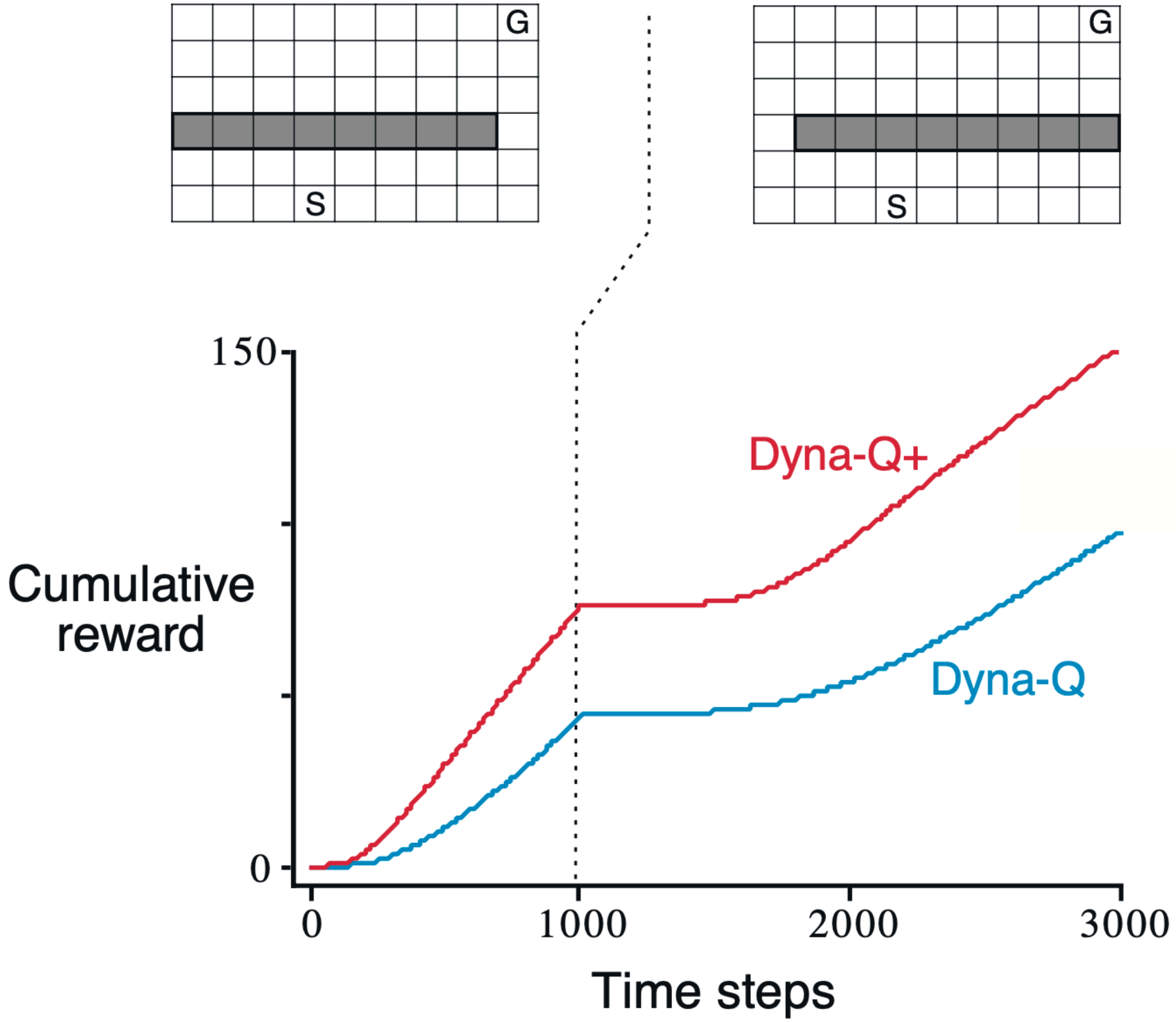
Search control

- “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?”
    - **Check out section 8.4**
- Practice question: can you think of a better way than random?

Dyna-Q vs 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?”

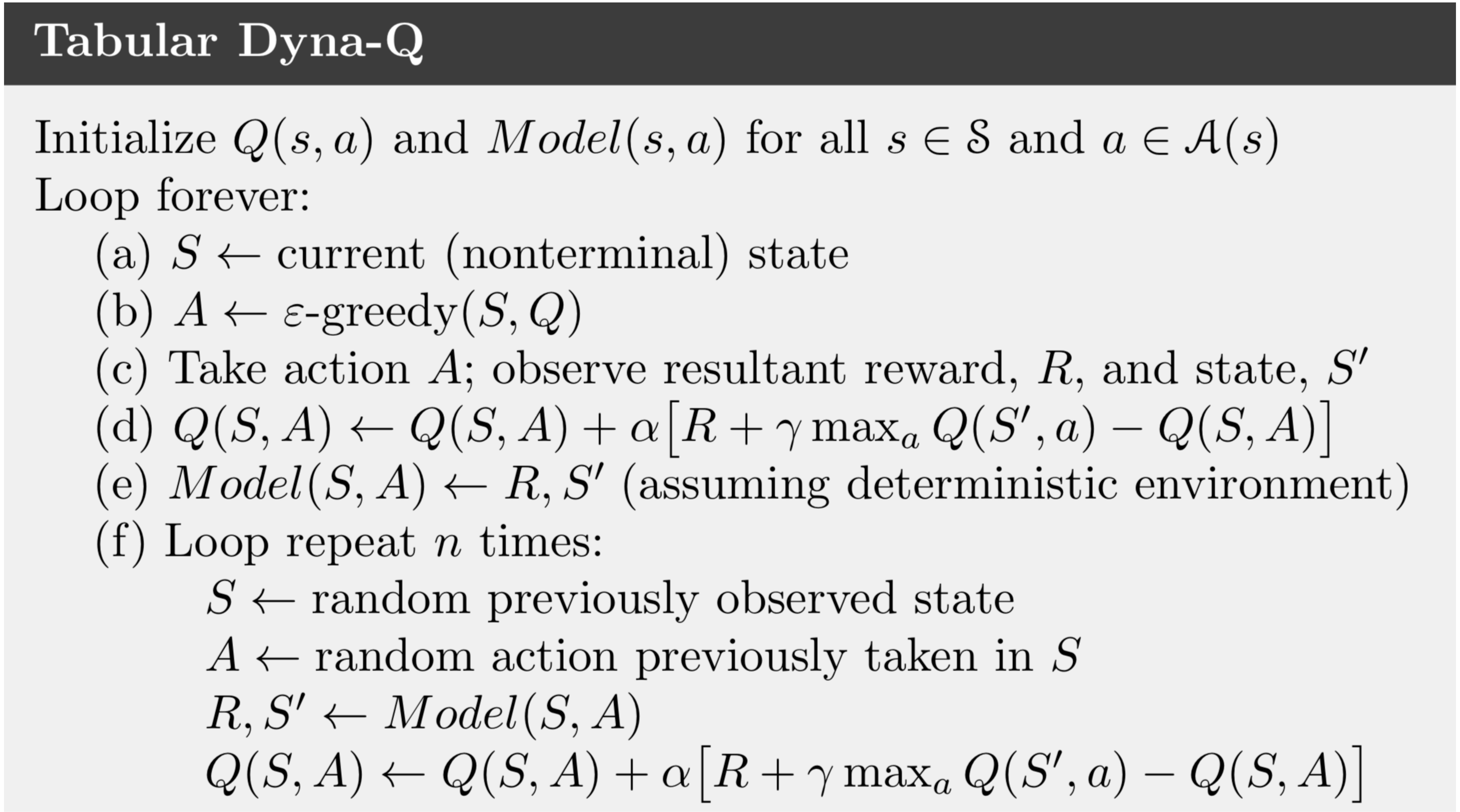


Why does Dyna Q+ encourage exploration?

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

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

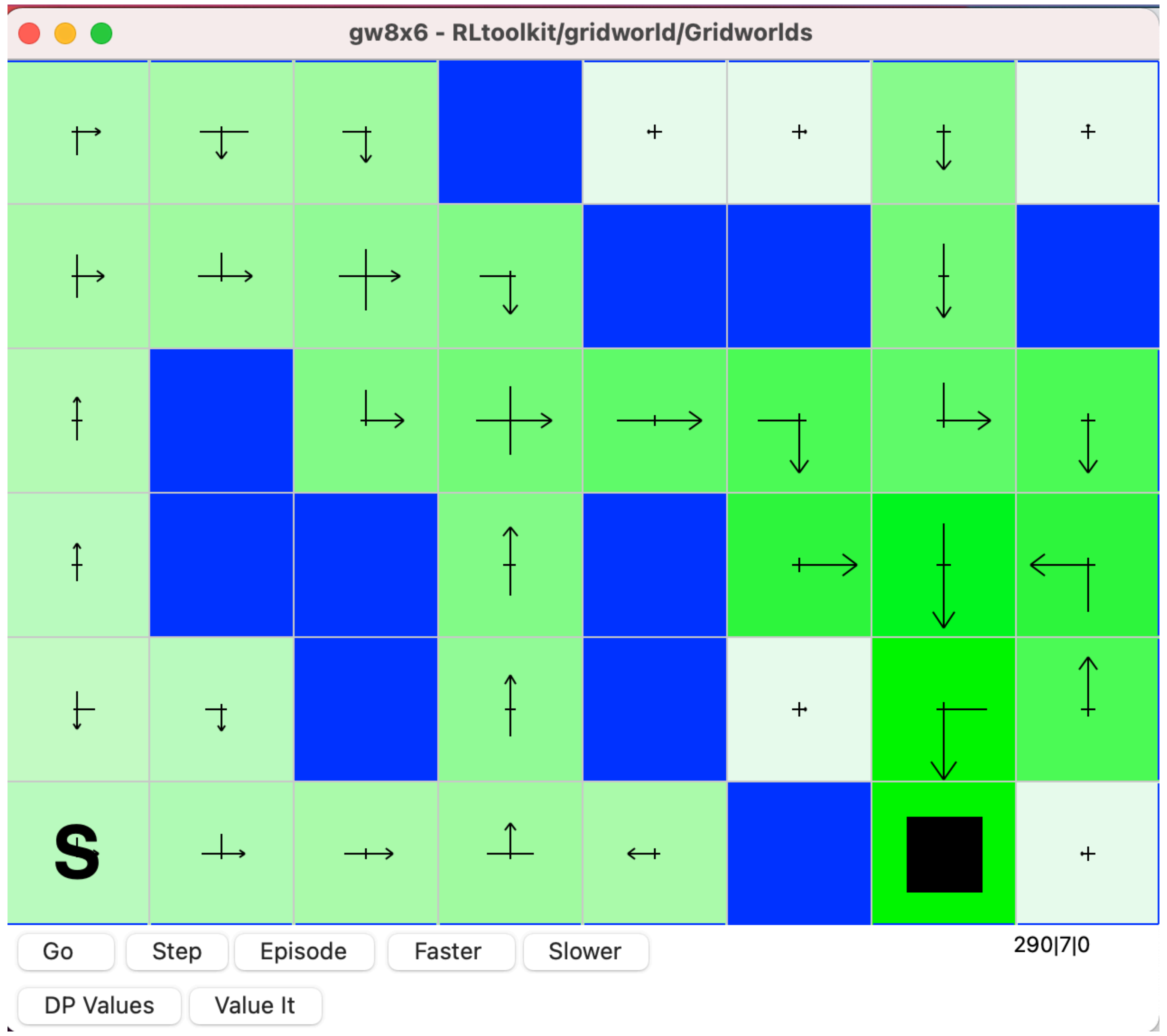
Where would we add this bonus??



Tabular Dyna-Q

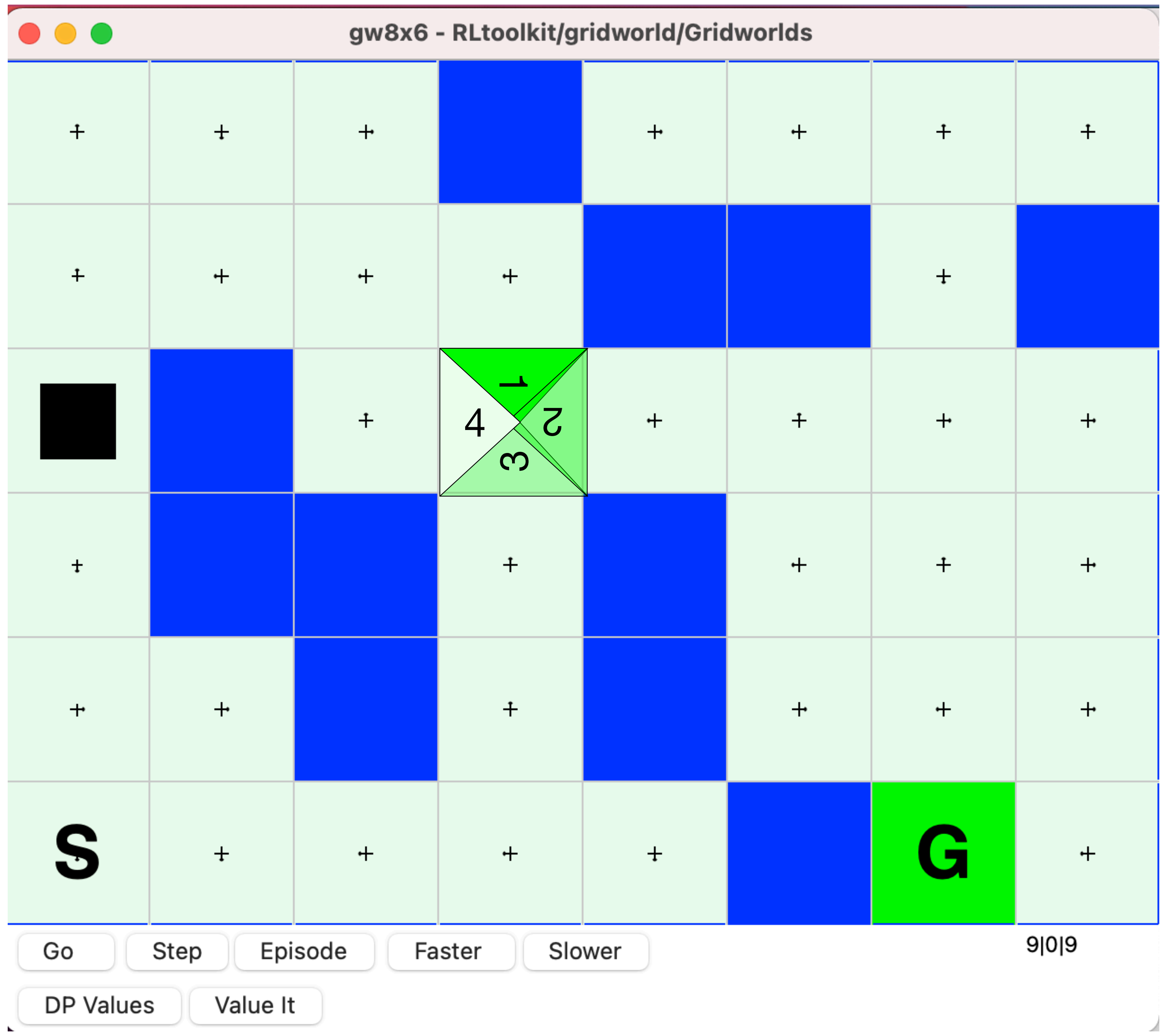
```
Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:
    (a) $S \leftarrow$ current (nonterminal) state
    (b) $A \leftarrow \varepsilon$-greedy$(S, Q)$
    (c) Take action $A$; observe resultant reward, $R$, and state, $S'$
    (d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
    (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 $S$
            $R, S' \leftarrow Model(S,A)$
            $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
```

The bonus propagates through the value function via planning


gw8x6 - RLtoolkit/gridworld/Gridworlds

290|7|0

Vs


gw8x6 - RLtoolkit/gridworld/Gridworlds

9|0|9

Exploration bonus does not change value function

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?”
  - Yes!
- Practice question: what idea/technology from machine learning could be used instead of a table make a model? Imagine the states were continuous.

Worksheet Q’s

Q1

1. An agent observes the following two episodes from an MDP,

$$S_0 = 0, A_0 = 1, R_1 = 1, S_1 = 1, A_1 = 1, R_2 = 1$$

$$S_0 = 0, A_0 = 0, R_1 = 0, S_1 = 0, A_1 = 1, R_2 = 1, S_2 = 1, A_2 = 1, R_3 = 1$$

and updates its deterministic model accordingly. What would the model output for the following queries:

(a) Model$(S = 0, A = 0)$:

(b) Model$(S = 0, A = 1)$:

(c) Model$(S = 1, A = 0)$:

(d) Model$(S = 1, A = 1)$:

Q2

2. An agent is in a 4-state MDP, $\mathcal{S} = \{1, 2, 3, 4\}$, where each state has two actions $\mathcal{A} = \{1, 2\}$. Assume the agent saw the following trajectory,

$$\begin{aligned}
&S_0 = 1, A_0 = 2, R_1 = -1, \\
&S_1 = 1, A_1 = 1, R_2 = 1, \\
&S_2 = 2, A_2 = 2, R_3 = -1, \\
&S_3 = 2, A_3 = 1, R_4 = 1, \\
&S_4 = 3, A_4 = 1, R_5 = 100, \\
&S_5 = 4
\end{aligned}$$

and uses Tabular Dyna-Q with 5 planning steps for each interaction with the environment.

(a) Once the agent sees S_5, how many Q-learning updates has it done with **real experience**? How many updates has it done with **simulated experience**?

Q2

2. An agent is in a 4-state MDP, $\mathcal{S} = \{1, 2, 3, 4\}$, where each state has two actions $\mathcal{A} = \{1, 2\}$. Assume the agent saw the following trajectory,

$$\begin{aligned}
&S_0 = 1, A_0 = 2, R_1 = -1, \\
&S_1 = 1, A_1 = 1, R_2 = 1, \\
&S_2 = 2, A_2 = 2, R_3 = -1, \\
&S_3 = 2, A_3 = 1, R_4 = 1, \\
&S_4 = 3, A_4 = 1, R_5 = 100, \\
&S_5 = 4
\end{aligned}$$

(b) Which of the following are possible (or not possible) simulated transitions $\{S, A, R, S'\}$ given the above observed trajectory with a deterministic model and random search control?

i. $\{S = 1, A = 1, R = 1, S' = 2\}$

ii. $\{S = 2, A = 1, R = -1, S' = 3\}$

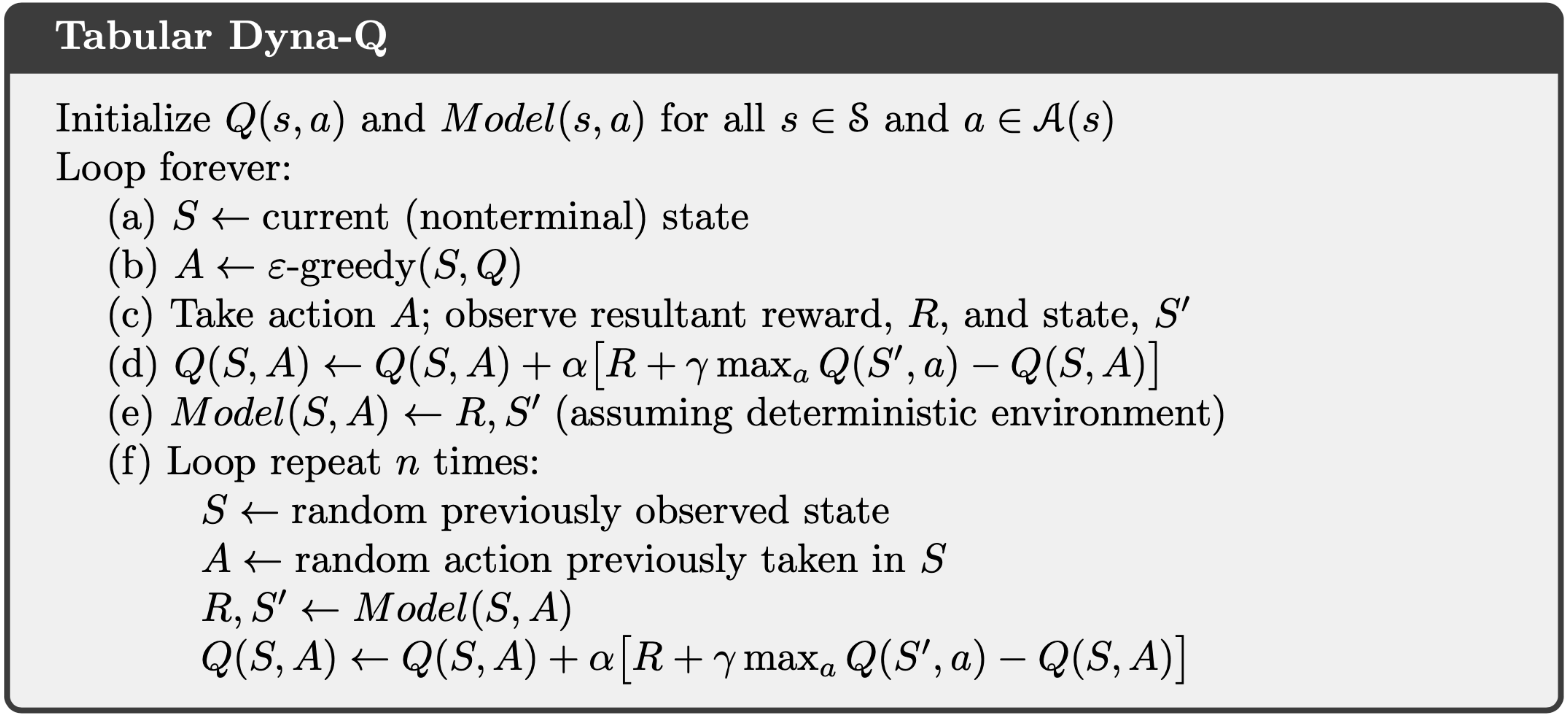
iii. $\{S = 2, A = 2, R = -1, S' = 2\}$

iv. $\{S = 1, A = 2, R = -1, S' = 1\}$

v. $\{S = 3, A = 1, R = 100, S' = 5\}$

Q3

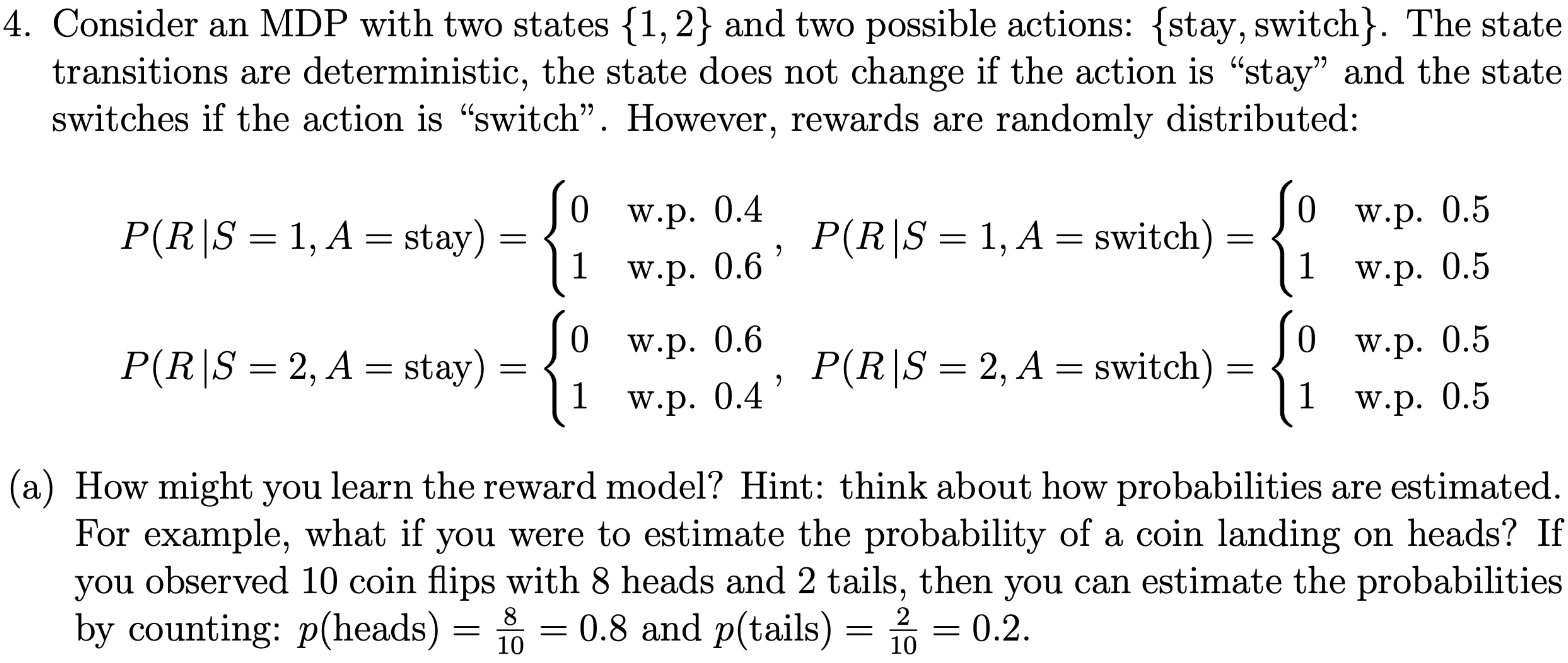
3. Modify the Tabular Dyna-Q algorithm so that it uses Expected Sarsa instead of Q-learning. Assume that the target policy is ϵ-greedy. What should we call this algorithm?



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 $S$
         $R, S' \leftarrow Model(S,A)$
         $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
```

Q4



4. Consider an MDP with two states $\{1, 2\}$ and two possible actions: $\{\text{stay}, \text{switch}\}$. The state transitions are deterministic, the state does not change if the action is "stay" and the state switches if the action is "switch". However, rewards are randomly distributed:

$$P(R\,|S = 1, A = \text{stay}) = \begin{cases} 0 & \text{w.p. } 0.4 \\ 1 & \text{w.p. } 0.6 \end{cases}, \quad P(R\,|S = 1, A = \text{switch}) = \begin{cases} 0 & \text{w.p. } 0.5 \\ 1 & \text{w.p. } 0.5 \end{cases}$$

$$P(R\,|S = 2, A = \text{stay}) = \begin{cases} 0 & \text{w.p. } 0.6 \\ 1 & \text{w.p. } 0.4 \end{cases}, \quad P(R\,|S = 2, A = \text{switch}) = \begin{cases} 0 & \text{w.p. } 0.5 \\ 1 & \text{w.p. } 0.5 \end{cases}$$

(a) How might you learn the reward model? Hint: think about how probabilities are estimated. For example, what if you were to estimate the probability of a coin landing on heads? If you observed 10 coin flips with 8 heads and 2 tails, then you can estimate the probabilities by counting: $p(\text{heads}) = \frac{8}{10} = 0.8$ and $p(\text{tails}) = \frac{2}{10} = 0.2$.

Q4

- Assume we want to estimate r(S,A,S’): reward as a function of state, action, next state
- If we have a finite set of rewards we could just count the number of times we see each reward in each <S,A,S’>
- Alternatively, we could also use or knowledge of learning rules and estimate the expected reward: E[r(S,A,S’)]
  - Look at the **Monte Carlo policy evaluation algorithm** on page 110
  - Look at the **Simple Bandit algorithm** on page 32

Q4 (b)

- If the state transitions were stochastic, how could we estimate the model?
  - We have already handled the reward part of the model
- Let’s focus on estimating p(s’|s,a):
  - The probability of transitioning from state s’ from state s under action a

Q4 (c)

- Using r(S,A,S’) and mu(S,A) and c(S,A,S’) how can we do an update to the value function in the planning loop?

- **Hint: this would be a full backup, not a sample backup**

  - It will look more like DP

  - e.g., $Q(S,A) \leftarrow \sum_{s' \in \mathcal{S}} p(s' \mid S,A)[r(S,A,s') + \gamma \max_{a' \in \mathcal{A}} Q(S',a)]$

  - How do we estimate p(s’|s,a) and what states do we sum over?