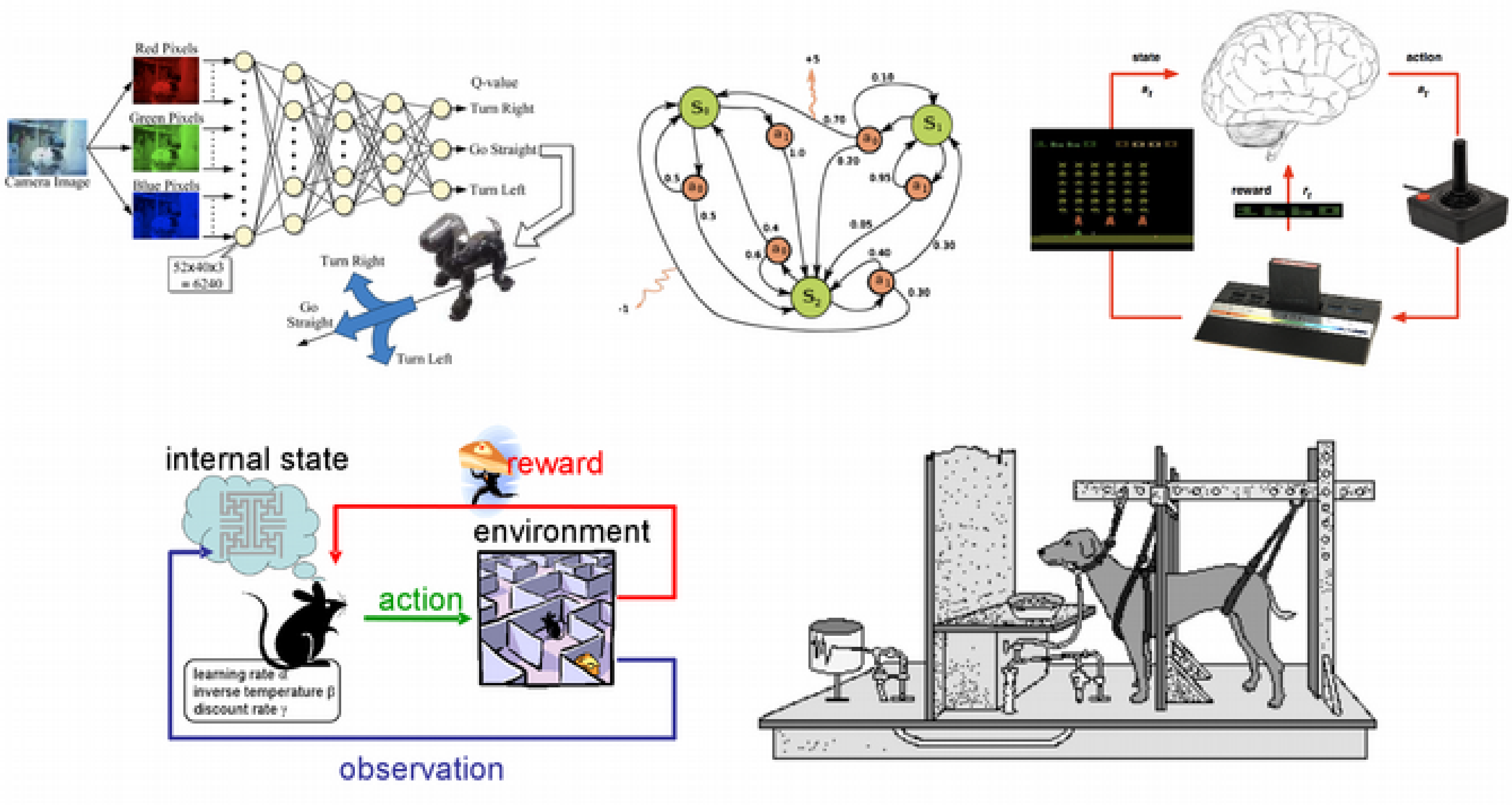
COMP 138: Reinforcement Learning



Instructor: Jivko Sinapov
Webpage: https://www.eecs.tufts.edu/~jsinapov/teaching/comp150_RL_Fall2020/

Announcements

Reading Assignment

- Chapter 7 of Sutton and Barto

Research Article Topics

- Transfer learning
- Learning with human demonstrations and/or advice
- Approximating q-functions with neural networks

Research Paper

- Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., & Thomaz, A. L. (2013). **Policy shaping: Integrating human feedback with reinforcement learning**. In Advances in neural information processing systems (pp. 2625-2633).
- Responses should discuss both readings
- You get extra credit for answering others’ questions!

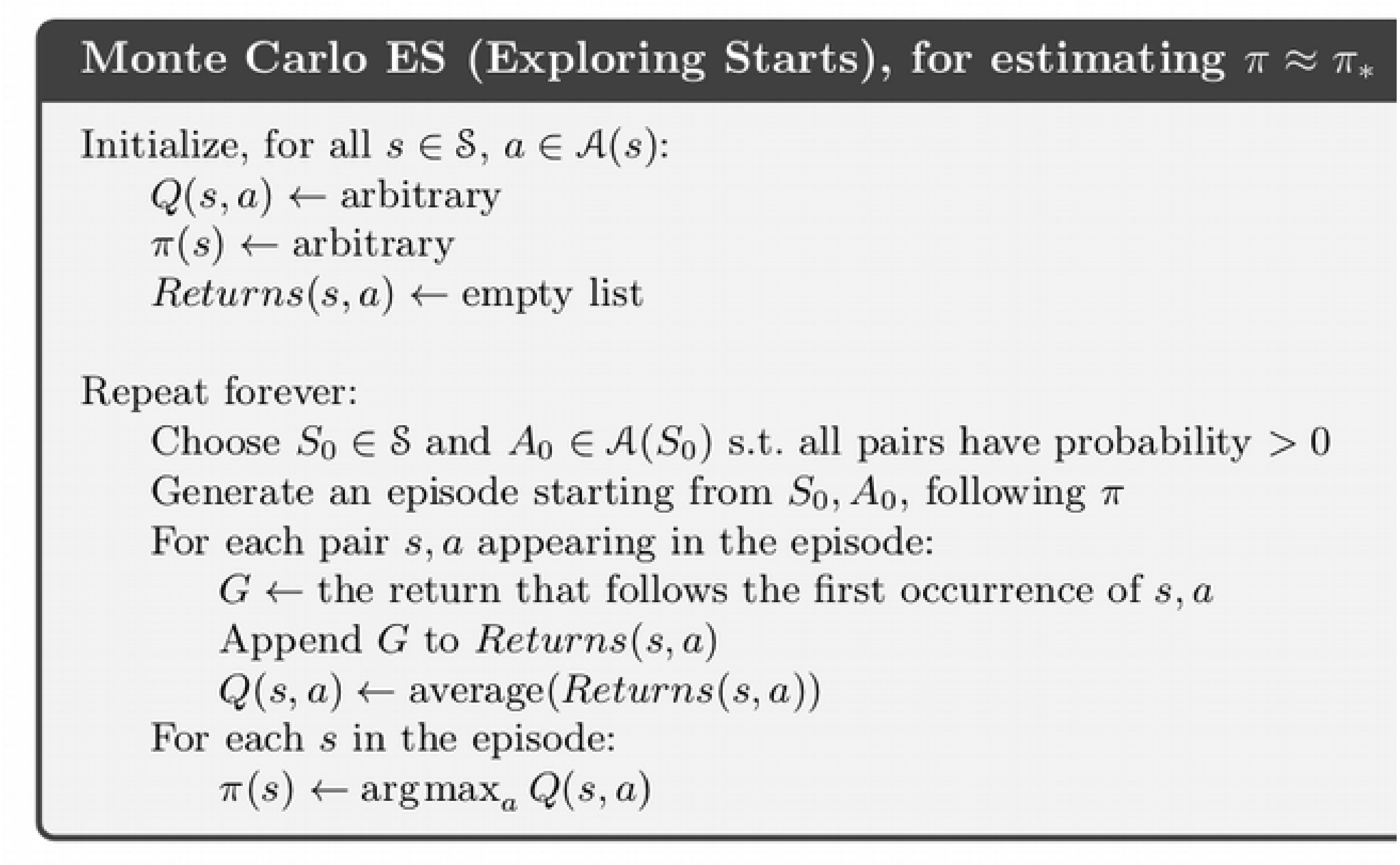
Programming Assignment #2

Programming Assignment #3

- Any programming exercise from Ch. 7 or 8
- You are encouraged to come up with variants of these exercises or try something completely different – just talk to me in advance

Monte Carlo Methods

Overview of Monte Carlo ES



Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

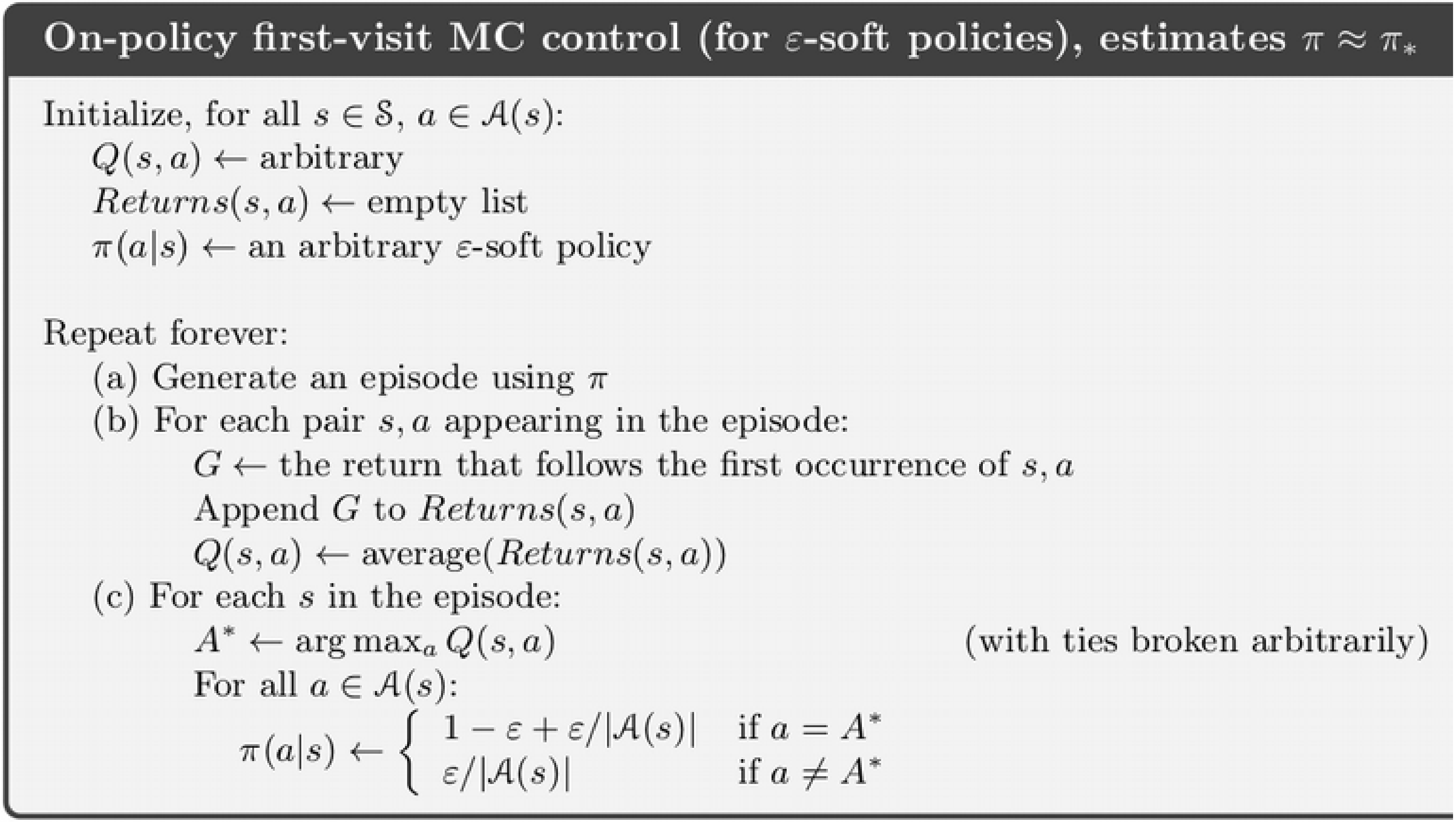
```
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \leftarrow$ arbitrary
    $\pi(s) \leftarrow$ arbitrary
    $Returns(s,a) \leftarrow$ empty list

Repeat forever:
    Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ s.t. all pairs have probability $> 0$
    Generate an episode starting from $S_0, A_0$, following $\pi$
    For each pair $s, a$ appearing in the episode:
        $G \leftarrow$ the return that follows the first occurrence of $s, a$
        Append $G$ to $Returns(s,a)$
        $Q(s,a) \leftarrow$ average$(Returns(s,a))$
    For each $s$ in the episode:
        $\pi(s) \leftarrow \arg\max_a Q(s,a)$
```

On- vs. Off-policy Methods

- On-policy methods attempt to improve a policy that is used for gathering data
- Off-policy methods attempt to improve a different policy from the one used for gathering data

Off-policy exploration in humans

On-policy MC



On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

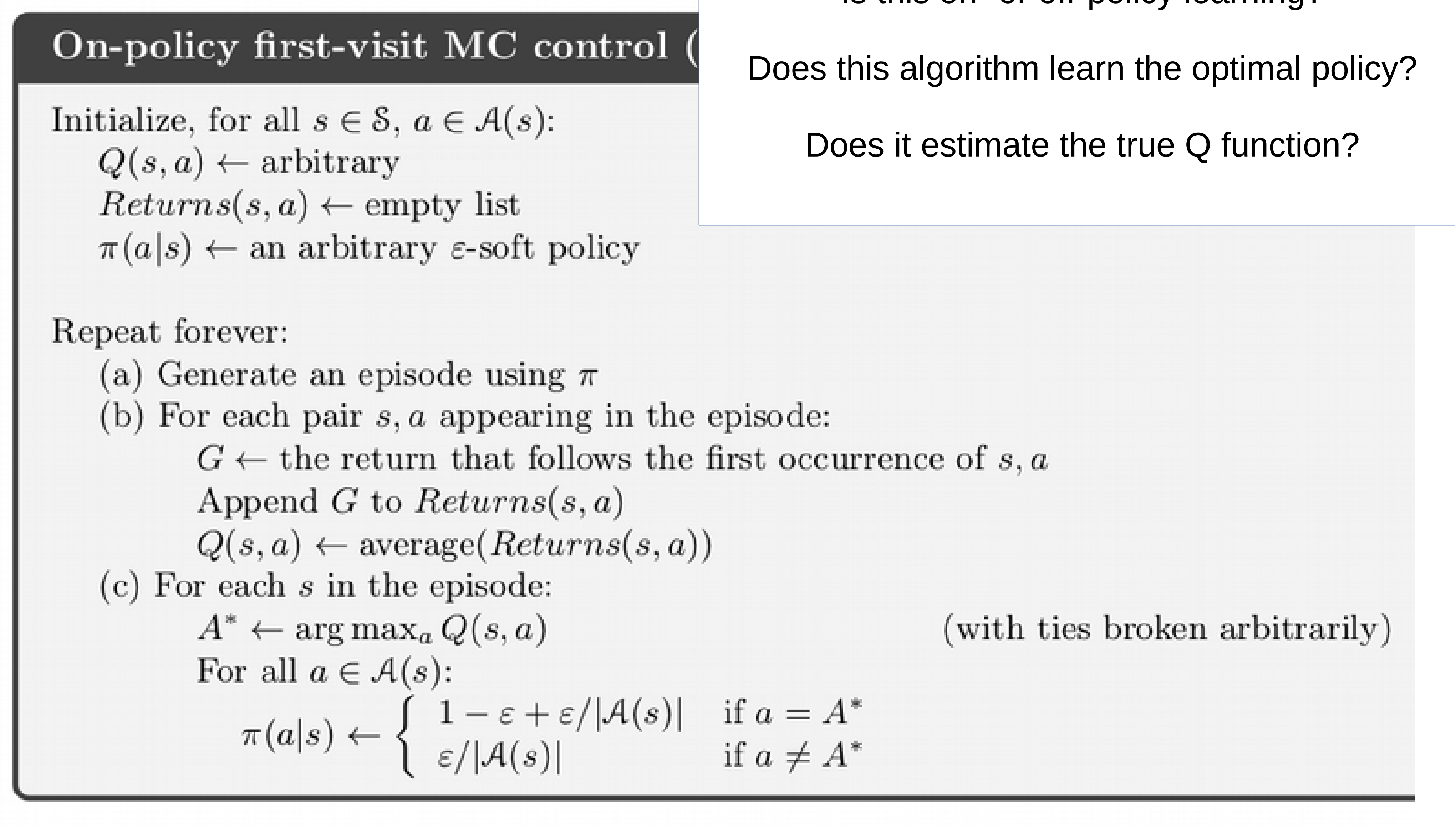
```
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \leftarrow$ arbitrary
    $Returns(s,a) \leftarrow$ empty list
    $\pi(a|s) \leftarrow$ an arbitrary $\varepsilon$-soft policy

Repeat forever:
    (a) Generate an episode using $\pi$
    (b) For each pair $s,a$ appearing in the episode:
            $G \leftarrow$ the return that follows the first occurrence of $s,a$
            Append $G$ to $Returns(s,a)$
            $Q(s,a) \leftarrow$ average($Returns(s,a)$)
    (c) For each $s$ in the episode:
            $A^* \leftarrow \arg\max_a Q(s,a)$                    (with ties broken arbitrarily)
            For all $a \in \mathcal{A}(s)$:
                $\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$
```

On-policy MC

Is this on- or off-policy learning?

Does this algorithm learn the optimal policy?

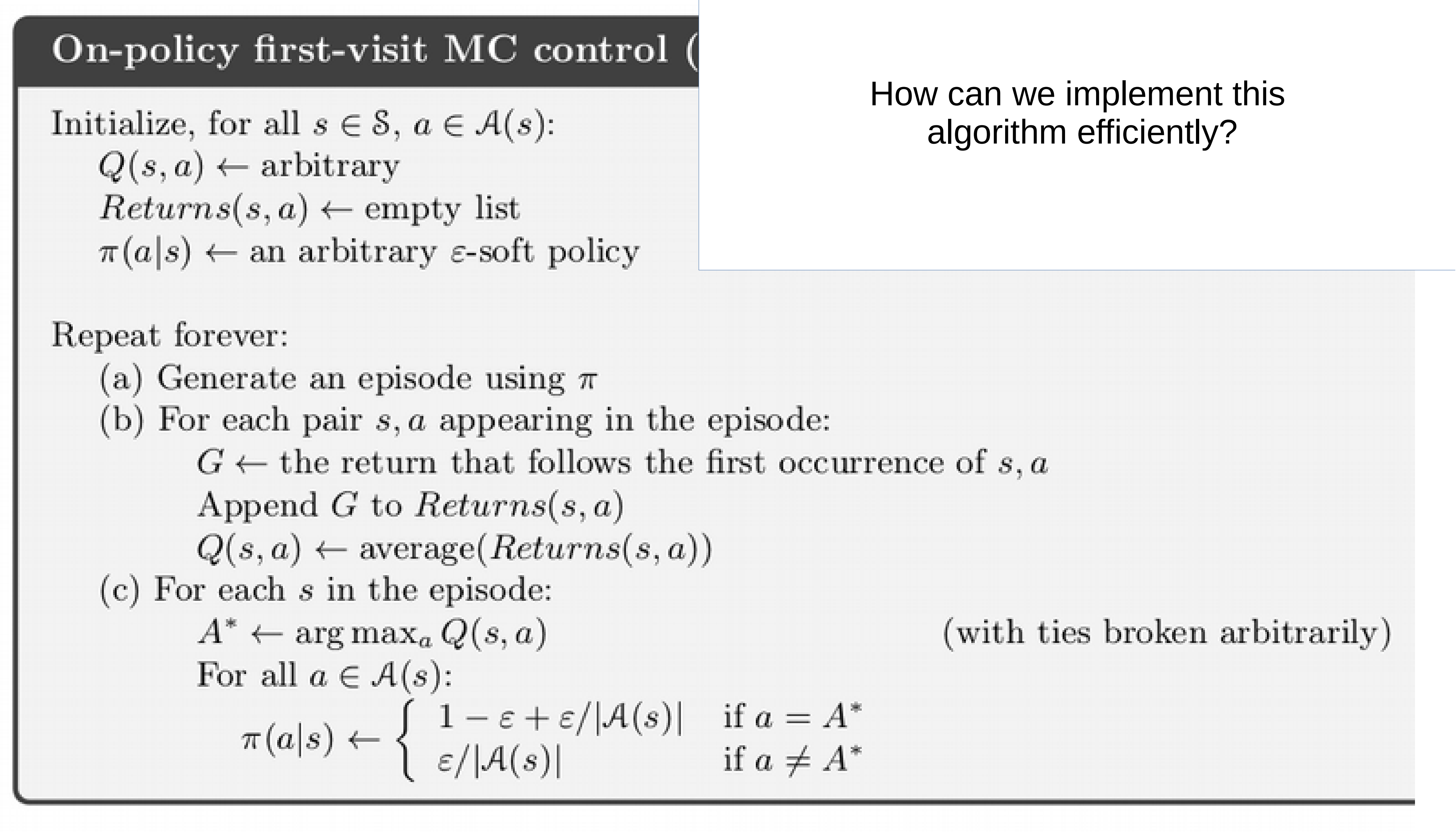
Does it estimate the true Q function?



On-policy first-visit MC control (

```
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s, a) \leftarrow$ arbitrary
    $Returns(s, a) \leftarrow$ empty list
    $\pi(a|s) \leftarrow$ an arbitrary $\varepsilon$-soft policy

Repeat forever:
    (a) Generate an episode using $\pi$
    (b) For each pair $s, a$ appearing in the episode:
            $G \leftarrow$ the return that follows the first occurrence of $s, a$
            Append $G$ to $Returns(s, a)$
            $Q(s, a) \leftarrow$ average($Returns(s, a)$)
    (c) For each $s$ in the episode:
            $A^* \leftarrow \arg\max_a Q(s, a)$                    (with ties broken arbitrarily)
            For all $a \in \mathcal{A}(s)$:
                $\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$
```

On-policy MC



On-policy first-visit MC control (

```
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \leftarrow$ arbitrary
    $Returns(s,a) \leftarrow$ empty list
    $\pi(a|s) \leftarrow$ an arbitrary $\varepsilon$-soft policy

Repeat forever:
    (a) Generate an episode using $\pi$
    (b) For each pair $s, a$ appearing in the episode:
            $G \leftarrow$ the return that follows the first occurrence of $s, a$
            Append $G$ to $Returns(s,a)$
            $Q(s,a) \leftarrow \text{average}(Returns(s,a))$
    (c) For each $s$ in the episode:
            $A^* \leftarrow \arg\max_a Q(s,a)$                    (with ties broken arbitrarily)
            For all $a \in \mathcal{A}(s)$:
                $\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$
```

How can we implement this algorithm efficiently?

Off-policy learning and importance sampling

- The prediction problem: given data generated using policy *b*, what is the value function for policy *π?*
- Off-policy prediction and control

Temporal Difference Learning

- Overview of Section 6.1

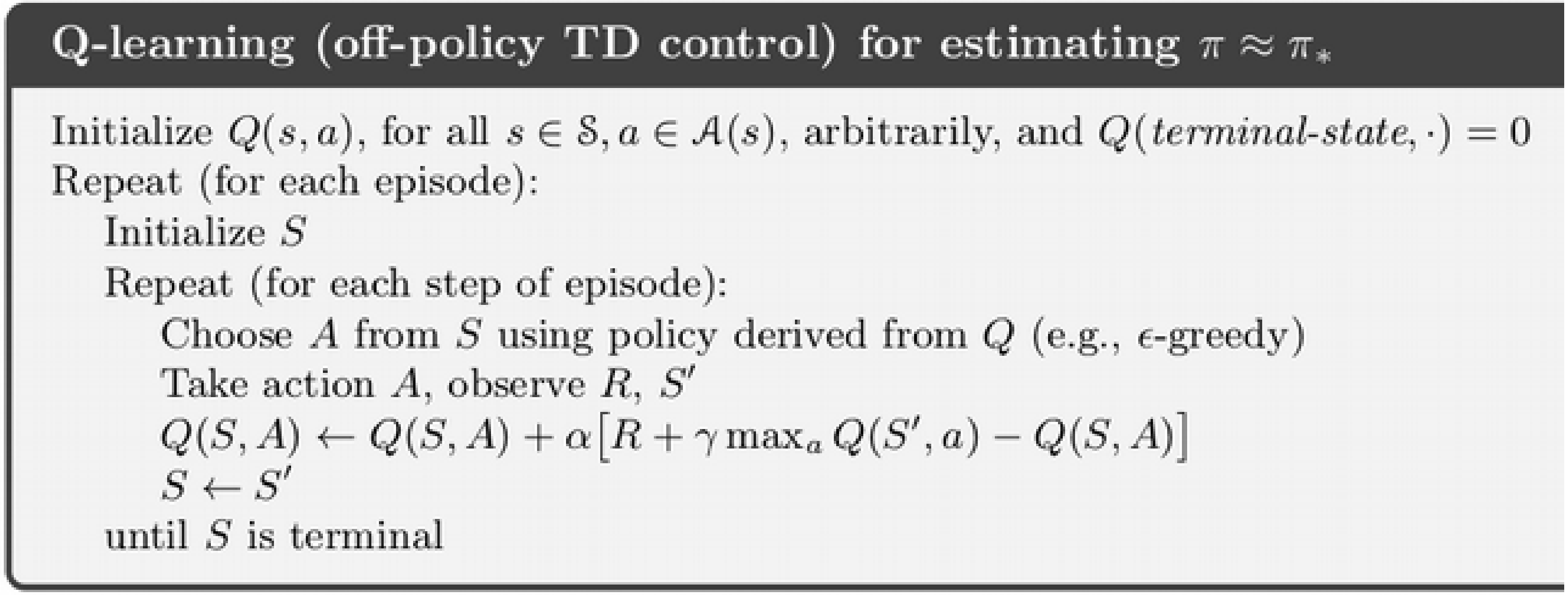
Temporal Difference Learning

- Example 6.1

Temporal Difference Learning

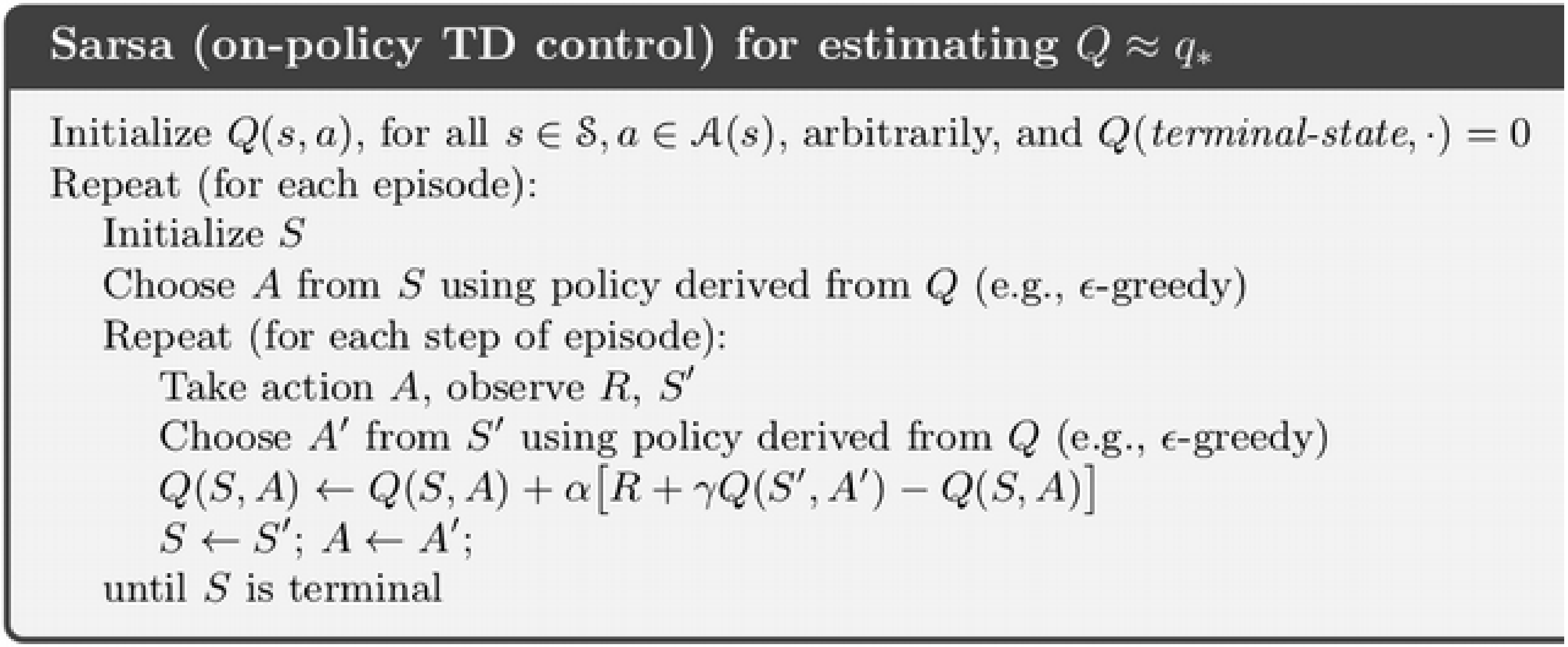
- Small group activity: Exercise 6.2

Q-Learning and Sarsa



Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

```
Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize $S$
    Repeat (for each step of episode):
        Choose $A$ from $S$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
        Take action $A$, observe $R$, $S'$
        $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
        $S \leftarrow S'$
    until $S$ is terminal
```



Sarsa (on-policy TD control) for estimating $Q \approx q_*$

```
Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize $S$
    Choose $A$ from $S$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
    Repeat (for each step of episode):
        Take action $A$, observe $R$, $S'$
        Choose $A'$ from $S'$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
        $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma Q(S',A') - Q(S,A)\big]$
        $S \leftarrow S'$; $A \leftarrow A'$;
    until $S$ is terminal
```

Cliff-walking example

Reading Discussion

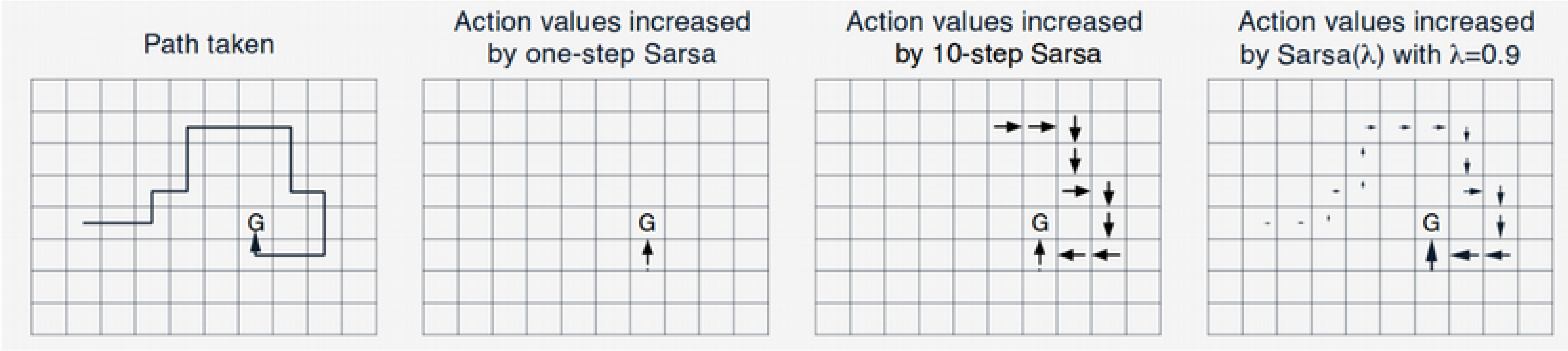
“The exact concept of TD learning confused me at first, but Example 6.1 and Figure 6.1 are very insightful. From what I understand, instead of waiting for an episode to terminate, learning is done after reward is received. Seems simple, but for something like games where the reward is 0 except on a terminal state, does this really apply? It would seem like there would need to be some cascading effect over multiple episodes to update state values near the start; not exactly "learning as you go".”

Reading Discussion

“How would one choose an algorithm, Sarsa, Q-learning, or Expected Sarsa, when tackling a particular problem?”

– Catherine

Reading Discussion


Path taken
Action values increased by one-step Sarsa
Action values increased by 10-step Sarsa
Action values increased by Sarsa(λ) with λ=0.9
G
G
G
G

Reading Discussion

“The article mentioned that "**it is almost certainly possible to find pairs of tasks for which no ρ exists, where transfer would provide no benefit or even hinder learning"**. I have two questions on this: are there ways to identify these pairs (other than experimentally)? Additionally, how typical are these occurrences? I would expect that the start and end domains in which transfer learning is taking place would be similar enough for this to be infrequent, but I'm wondering if that's the case.”

– Mike

Reading Discussion

“How can one determine the optimal amount of training in the source task automatically based on task characteristics?”

– Sai

Reading Discussion

“How to understand the sentence **“Instead of finding similarities between different states, we focus on exploiting similarities between different tasks”** from the paper? Shouldn't we think that a state refers to a specific task?”

– Pandong

Reading Discussion

“If expected Sarsa often finds more optimal policies faster than both Sarsa and Q-learning, why is it less popular than Q-learning?”

– Michael

Open Questions about Transfer Learning

- What are some unanswered research questions posed by the article?

Further Reading on Transfer Learning

THE END