Solutions to Taxi-v1, Cartpole-v1, and MountainCar-v0 (along with many others) are also ranked according to the number of episodes before the solution is found. Towards this objective, it makes sense to design an algorithm that learns the optimal policy π_* as quickly as possible.

Exploration-Exploitation Dilemma

Recall that the environment's dynamics are initially unknown to the agent. Towards maximizing return, the agent must learn about the environment through interaction.

At every time step, when the agent selects an action, it bases its decision on past experience with the environment. And, towards minimizing the number of episodes needed to solve environments in OpenAI Gym, our first instinct could be to devise a strategy where the agent always selects the action that it believes (*based on its past experience*) will maximize return. With this in mind, the agent could follow the policy that is greedy with respect to the action-value function estimate. We examined this approach in a previous video and saw that it can easily lead to convergence to a sub-optimal policy.

To see why this is the case, note that in early episodes, the agent's knowledge is quite limited (and potentially flawed). So, it is highly likely that actions *estimated* to be non-greedy by the agent are in fact better than the *estimated* greedy action.

With this in mind, a successful RL agent cannot act greedily at every time step (*that is*, it cannot always **exploit** its knowledge); instead, in order to discover the optimal policy, it has to continue to refine the estimated return for all state-action pairs (*in other words*, it has to continue to **explore** the range of possibilities by visiting every state-action pair). That said, the agent should always act *somewhat greedily*, towards its goal of maximizing return *as quickly as possible*. This motivated the idea of an ϵ-greedy policy.

We refer to the need to balance these two competing requirements as the **Exploration-Exploitation Dilemma**. One potential solution to this dilemma is implemented by gradually modifying the value of ϵ when constructing ϵ-greedy policies.

Setting the Value of ϵ, in Theory