Wprowadzenie do RL 5

Wszystkie zadania dotyczą środowiska **Frozen Lake**

Zadanie 1

Napisz funkcję, która ze znajomości ***V(s)*** dla każdego stanu **s**, pozwala wyliczyć wartości zwrotów dla wszystkich akcji w pewnym stanie ***s***, czyli ***Q(s,a)***, dla każdej akcji ***a*** możliwej do wykonania w stanie ***s***. Wykorzystaj formułę:

$$q_\pi(s,a) \quad = \quad \sum_{s',r} p(s',r\,|\,s,a)\Big[r+\gamma v_\pi(s')\Big]$$

Zapoznaj się z notatnikiem **FrozenLake_3.ipynb** (otwórz go w Colab). Znajdziesz tam wskazówki. Wykonaj podane tam polecenia.

Zadanie 2

Napisz program implementujący poniższy **algorytm iteracji polityki** do środowiska *Frozen Lake* w celu znalezienia optymalnej polityki π_*. Przyjmij, że polityka π jest **deterministyczna**.

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\Delta \leftarrow 0$
   Loop for each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s',r\,|\,s,\pi(s))\big[r+\gamma V(s')\big]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s',r\,|\,s,a)\big[r+\gamma V(s')\big]$
   If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Zapoznaj się z notatnikiem **FrozenLake_4.ipynb** (otwórz go w Colab). Znajdziesz tam wskazówki. Wykonaj podane tam polecenia.