Overview: The following tutorial questions relate to material taught in week 3 of the 2023-24 Reinforcement Learning course. They aim at encouraging engagement with the course material and facilitating a deeper understanding.

For this week, we will look at how the concepts of terminating and absorbing state are not actually compatible, but relate to different formulations of the same problem (or pseudo-code, if you prefer). We then rehash Monte Carlo (MC) prediction and touch upon MC control. We will make use of Temporal Difference (TD) learning for one prediction (policy evaluation) step. The answers (whether delivered by your tutor or read later at home) will provide you also with a somewhat more theoretical consideration relating to TD prediction convergence.

Problem 1 - Modelling & Monte Carlo Control

Consider the simple maze problem in Figure 1 below, comprised of 8 states $s_1, \ldots, s_8$, numbered from the bottom left to the top right. The agent can move from any state to any adjacent state (e.g. from $s_1$ to either $s_4$ or $s_2$), without error. Our goal is to follow the shortest path (from any state) to $s_8$. Upon arrival to a new state, the agent receives a reward dependent only on that state. We assign $s_8$ a reward of 10, and penalise arrival to any other state with $-1$.

The arrows in Figure 1 summarise the policy $\pi_0$ which we will be evaluating in Part b of this question. Essentially, assume a deterministic policy for states $s_2, s_3, s_5, s_6, s_7$, as indicated by the respective arrow. Further assume a 50% chance of moving in either direction for states $s_1, s_4$. 

Part a

- Should $s_8$ be defined as a terminating state? Why?
- Should $s_8$ be defined as an absorbing state? Why?

From here on, assume a discount factor of $\gamma = 1$.

Part b

Assuming the starting state $S_0 = s_1$ and the policy $\pi_0$ outlined above, list the two shortest possible trajectories our agent can follow (stopping at state 8). Further to that, consider the trajectory:

$$(s_1, up), -1, (s_4, down), -1, (s_1, right), -1, (s_2, right), -1, (s_3, up), -1, (s_5, up), +10, (s_8)$$

For each of those trajectories, carry out an iteration of policy evaluation using First-visit Monte Carlo (where it is implied that you average across samples as opposed to using some other learning rate), computing the action value function. Start from an initial evaluation of 0 across state-action pairs and go through the trajectories in any order.
Part c
Perform one step of greedy policy improvement on policy $\pi_0$ (assuming no access to the model), based on the evaluation from Part b.

Problem 2 - TD Prediction
Use the trajectory

$$(s_1, right), -1, (s_2, right), -1, (s_3, up), -1, (s_5, up), +10, (s_8)$$

to run one iteration of Temporal Difference policy evaluation (use the SARSA update rule) on the policy $\pi_1$ you computed for Problem 1c. Assume a step size of $\alpha = 0.1$ (you are assuming that the action that would be taken at each time-step of this trajectory when sampling actions using $\pi_1$ is the one indicated in the trajectory).

References