Exercise I: Introductory Lectures

Part A:

Ex 3.7. The problem could be lack of exploration, because the agent might not be exploring enough to discover the optimal path. The rewards could be too sparse and make the robot hard to learn and update the robot policy learning. Insufficient state representation of the robot, so the relevant information is not captured by the robot.

Ex. 3.8.

G5 = 0

G4 = R5 + gamma \* G5 = 2 + 0.5 \* 0 = 2

G3 = R4 + gamma \* G4 = 3 + 0.5 \* 2 = 4

G2 = R3 + gamma \* G3 = 6 + 0.5 \* 4 = 8

G1 = R2 + gamma \* G2 = 2 + 0.5 \* 8 = 6

G0 = R1 + gamma \* G1 = -1 + 0.5 \* 6 = 2

Ex 3.9.

G1=

G0=R1+gamma\*G1=2+0.9\*70=65

Ex 3.12.

vπ(s) = Σπ(a|s) \* qπ(s, a)

Ex 3.18

Ex 3.19

Part B.

1. The RL agent in the code is solving the task of playing Tic-Tac-Toe. It learns to make optimal moves by updating its value estimations based on the outcomes of the game.

The code includes the necessary components for an RL agent, such as a policy, value function estimation, learning algorithm, and exploration-exploitation trade-off.

Overall, the code demonstrates how the concepts of RL, such as policies, value estimation, and learning algorithms, are implemented in a specific domain (Tic-Tac-Toe). It showcases how an RL agent can learn and improve its performance through interactions with the environment.

2.