Problem 1 (10)
In the context of the Tic-Tac-Toe game discussed in class, what if instead of playing against a fixed opponent, the reinforcement learning (RL) agent would play against itself (i.e. another instantiation of the RL agent constitutes the opponent). What do you think would happen? How would such a change impact the learning process? Explain your answer in detail.

Problem 2 (20)
Assume a finite state set MDP with a known model and a stochastic policy, \( \pi(s,a) \). Find an expression for the mean transition probabilities between the states. **Note:** you’ll need to “transform” the MDP into a Markov Chain.

Problem 3 (20)
Imagine that you are designing a robot to run in a maze. You decide to give it a reward of +1 for escaping from the maze and a reward of 0 at all other times. The task seems to break down naturally into episodes – the successive runs through the maze – so you decide to treat it as an episodic task, where the goal is to maximize expected total reward. After running the learning agent for a while, you find that it is showing no improvement in escaping from the maze. What is going wrong? What are the changes that can be made to improve the performance of the robot?

Problem 4 (25)
(a) In class, we discussed the notion of defining the return as the sum of (exponentially) discounted rewards. Describe two reasons for using discounted rewards vs. a simple summation of the rewards.

(b) We further mentioned that despite the many advantages of using sum of discounted rewards to form the Return, it might not be an optimal construct/representation. Provide an intuitive example of a case whereby discounting rewards exponentially is improper or counter-productive.

(c) Propose an alternative formulation for the return and discuss your proposition with regard to the two advantages discussed in (a) as well as how it addresses the limitation raised in (b).
Problem 5 (25)

Consider a discrete-time, two-state Markov-modulated process shown below.

![Diagram of a two-state Markov process]

Assume that the above model corresponds to a packet generator which generates a single packet for each time slot that the process is in the ON state, while no packets are generated when the process is in the OFF state. Let a burst be defined as a sequence of continuous packet generations. One way of interpreting this model is that it produces bursts of packets separated by “idle bursts”. Let the traffic load be defined as the mean share of the time slots in which packets are being generated.

(a) Find closed-form expressions for the mean burst length and the traffic load.
(b) Extend this two-state model such that each burst will have a minimal length of $k$ time slots followed by a random duration, as would be the case in the two-state model.