Reinforcement Learning Lab
Due Oct 6th Monday 4pm

What to submit:

- FronzenLake Problem Code (40pts)
- Another Problem that you and your partner need to create (**not found in the Internet!** If you can create a reasonable RL coding problem that shows you really know the concept) and its coding work. Provide as many description paragraphs as possible to explain what the problem is about and how you complete the coding work, and of course you shall code it. Use your AI coding editor! (60pts)

OpenAI Gym **Frozen Lake** problem (like a 'Hello World' problem) provides a friendly environment to learn RL. The Frozen Lake problem is a simple, visual game that helps student understand how RL agents quickly adopt to its surroundings and learn to make decisions.

Objectives

- How to interact with the environment using code
- How to visualize the agent's actions and environment state
- The basics of RL concepts: states, actions, rewards, and episodes

1. Setting

- **Your choice of AI coding editor**
- **OpenAI Gym** (or Gymnasium) library installed

To install Gym, run this in your terminal or command prompt:
Use your AI coding editor or,

```
pip install gymnasium
```

2. FrozenLake Description

In the FrozenLake, the agent must find a path from the **Start (S)** to the **Goal (G)**, avoiding **Holes (H)**. The rest of the tiles are **Frozen (F)** and safe to walk on. The agent can move **Left, Down, Right, or Up**. The environment is slippery, so sometimes the agent may not move in the intended direction.

Legend:

- S: Start
- F: Frozen (safe)
- H: Hole (danger)
- G: Goal

Use your AI coding editor to create any map that you like with above information or,

Example 4x4 map:

```
S F F F
F H F H
F F F H
H F F G
```

3. Step-by-Step Instructions

A. Import and Create the Environment

Use your AI coding editor, enter something similar to the following information or,

```
import gymnasium as gym

env = gym.make('FrozenLake-v1', map_name="4x4", is_slippery=True)
```

B. Reset and Render the Environment

This puts the agent at the start and shows the map:
Use your AI coding editor or,

```
state, info = env.reset()
env.render()
```

C. Explore the Action and State Space

See what actions are possible and how many states there are:

Use your AI coding editor or,

```
print("Action space:", env.action_space)      # Discrete(4)
print("Observation space:", env.observation_space)  # Discrete(16)
```

D. Take Random Actions

Let's see what happens when the agent moves randomly:

Use your AI coding editor or,

```
import time

done = False
state, info = env.reset()
env.render()

while not done:
    action = env.action_space.sample()  # Pick a random action
    state, reward, done, truncated, info = env.step(action)
    env.render()
    time.sleep(1)  # Pause to see each move
    if done:
        print(f"Episode finished. Reward: {reward}")
        break
```

E. Understanding the Output

- The agent (red square) moves around the grid.
- If it falls into a hole, the episode ends (reward = 0).
- If it reaches the goal, the episode ends (reward = 1).
- Otherwise, the agent keeps moving.

Questions to think about:

Discuss the following questions with your partner

- Does the agent usually reach the goal?
- How often does the agent succeed?
- Why do you think it fails or succeeds?

Helpful resources

1. Step-by-step video tutorial (YouTube) – Q-Learning with FrozenLake
   https://www.youtube.com/watch?v=fq01-G0xDI8
2. https://zoo.cs.yale.edu/classes/cs470/materials/hws/hw7/FrozenLake.html