Homework 1 Template

Use this template to record your answers for Homework 1. Add your answers using LaTeXand then save your document as a PDF to upload to Gradescope. You are required to use this template to submit your answers. **You should not alter this template in any way** other than to insert your solutions. You must submit all 11 pages of this template to Gradescope. Do not remove the instructions page(s). Altering this template or including your solutions outside of the provided boxes can result in your assignment being graded incorrectly.

You should also export your code as a .py file and upload it to the **separate** Gradescope coding assignment. Remember to mark all teammates on **both** assignment uploads through Gradescope.

Instructions for Specific Problem Types

On this homework, you must fill in blanks for each problem. Please make sure your final answer is fully included in the given space. **Do not change the size of the box provided.** For short answer questions you should **not** include your work in your solution. Only provide an explanation or proof if specifically asked.

Fill in the blank: What is the course number?

10-703

Problem 0: Collaborators

Enter your team members' names and Andrew IDs in the boxes below. If you worked in a team with fewer than three people, leave the extra boxes blank.

Name 1: Madhusha Goonesekera Andrew ID 1: mgoonese

Name 2: Shrudhi Ramesh Shanthi Andrew ID 2: srameshs

Name 3: Siddharth Ghodasara Andrew ID 3: sghodasa

Problem 1: Value Iteration & Policy Iteration (30 pts)

1.1: Contraction Mapping (3 pts)

Solution

1.1.1 FALSE: Although a contraction mapping has a fixed point, the existence of a fixed point does not imply a contraction mapping. The value constant γ would have to hold true for all x, y $\in \mathcal{X}$, which isn't guaranteed by the existance of a fixed point.

1.1.2 TRUE: The Contraction Mapping theorem states that for a γ-contraction F in a complete normed vector space $\mathcal{X}$, there exists a unique fixed point in $\mathcal{X}$

1.1.3 TRUE: Yes, assuming all goes well and iterations of greedification cease due to policy-stability (all actions from previous iteration of $\pi(s)$ match the updated $\pi(s)$ AKA policy is unchanged), then an optimal solution has been found. Since the policy did not change, this also means that $F^{\pi_{k+1}} = F^{\pi_k} = F^{\pi_*}$

1.2.1 Table: Policy Iteration (4 pts)

Environment	**# Policy Improvement Steps**	**Total # Policy Evaluation Steps**
Deterministic-4x4	8	10
Deterministic-8x8	14	23

1.2.2 Optimal Policies for Deterministic-4x4 and 8x8 Maps (2 pts)

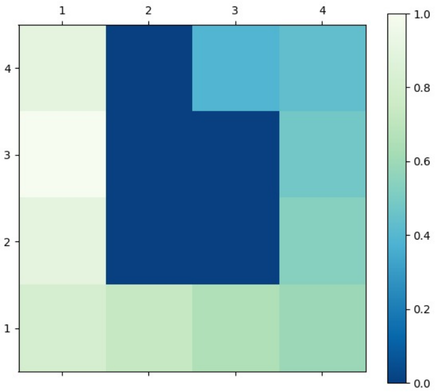
Solution

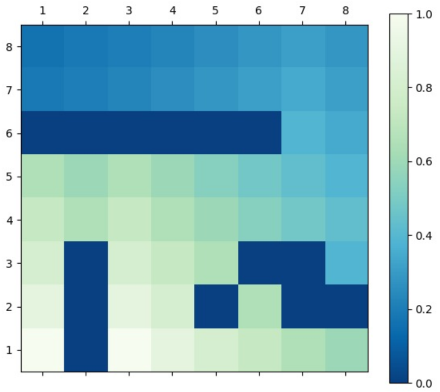
```
(4x4)
DLRD
RLLD
ULLD
ULLL
```

```
(8x8)
DDDDDDDL
RRRRRRDL
LLLLLLDL
DLDLLLLL
DLDLLLLL
DLDLLLLU
DLDLLDLL
RLLLLLLL
```

1.2.3 Value Functions of the Optimal Policies (2 pts)

Solution



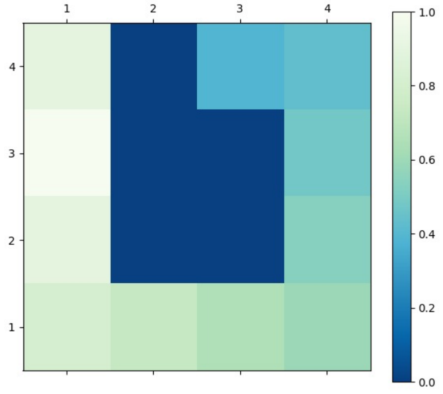


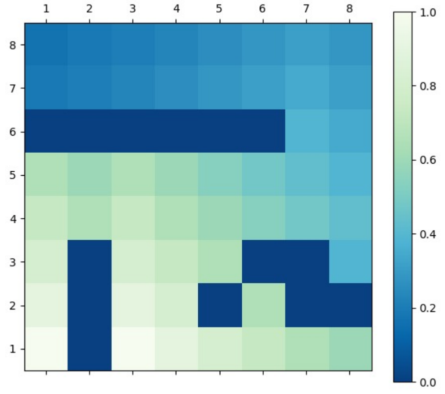
1.3.1 Table: Synchronous Value Iteration (3 pts)

Environment	# Iterations
Deterministic-4x4	10
Deterministic-8x8	18

1.3.2 Value Functions from Synchronous Value Iteration (2 pts)

Solution





1.3.3 Optimal Policies from Synchronous Value Iteration (2 pts)

Solution

```
(4x4)
DLRD
RLLD
ULLD
ULLL
```

```
(8x8)
DDDDDDDL
RRRRRRDL
LLLLLLDL
DLDLLLLL
DLDLLLLL
DLDLLLLU
DLDLLDLL
RLLLLLLL
```

1.4.1 Table: Asynchronous Policy Iteration (4 pts)

Heuristic	**Policy Improvement Steps**	**Total Policy Evaluation Steps**
Ordered	8	7992
Randperm	14	13986

1.5.1 Table: Asynchronous Value Iteration (4 pts)

Heuristic	**# Iterations**
Ordered	14
Randperm	8

1.5.2 Asynchronous VI with Domain-specific Heuristic (4 pts)

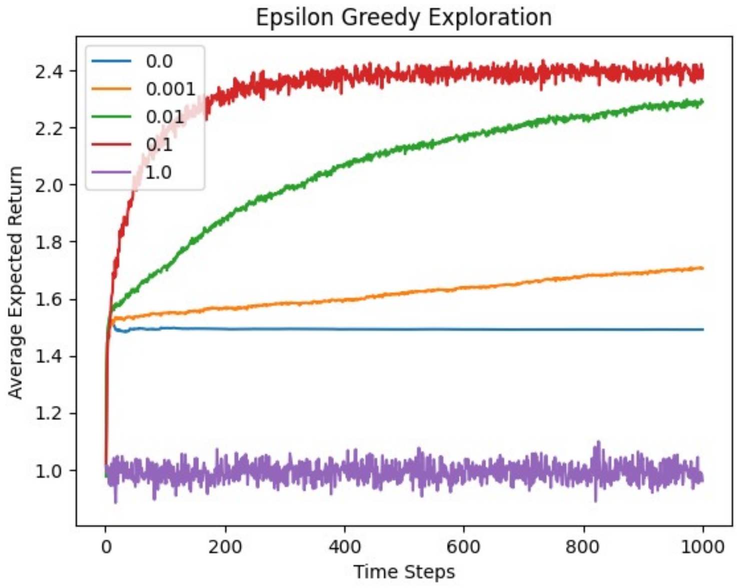
Solution

Env	**# Iterations**
Deterministic-4x4	5
Deterministic-8x8	6

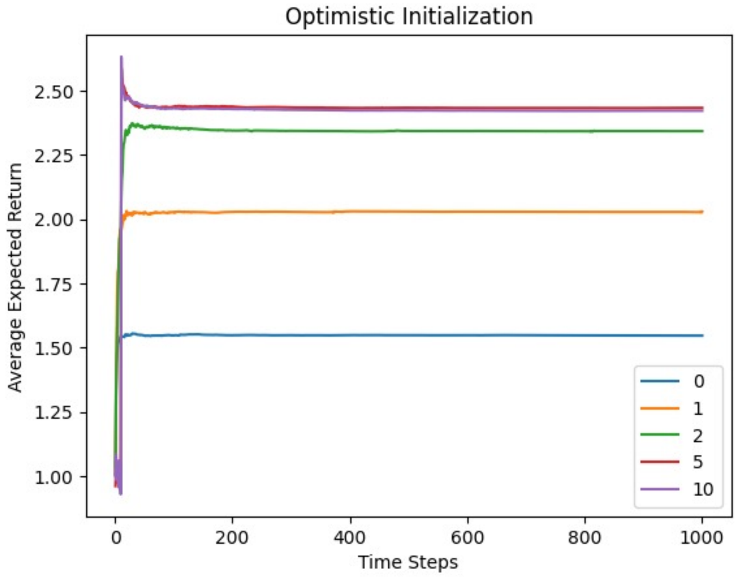
1.5.2 (b) One case where a ”goal based” heuristic can be used is when the search is required to be carried out in a bigger map (larger grid) and where the start & goal are relatively close to each other. This heuristic works well because of this robot’s absence of diagonal movement. This leads to fewer unnecessary state expansions and faster convergence in the value iteration process.

Problem 2: Bandits (36 pts)

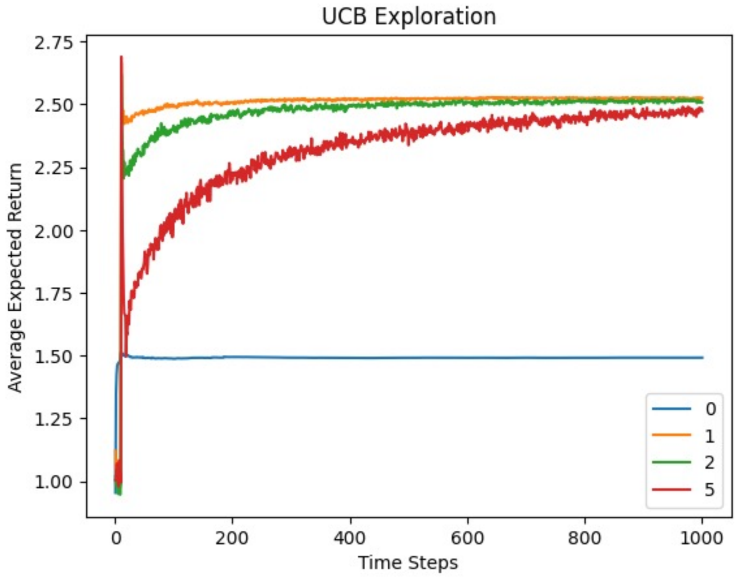
2.1 ϵ-Greedy Plot (8 pts)



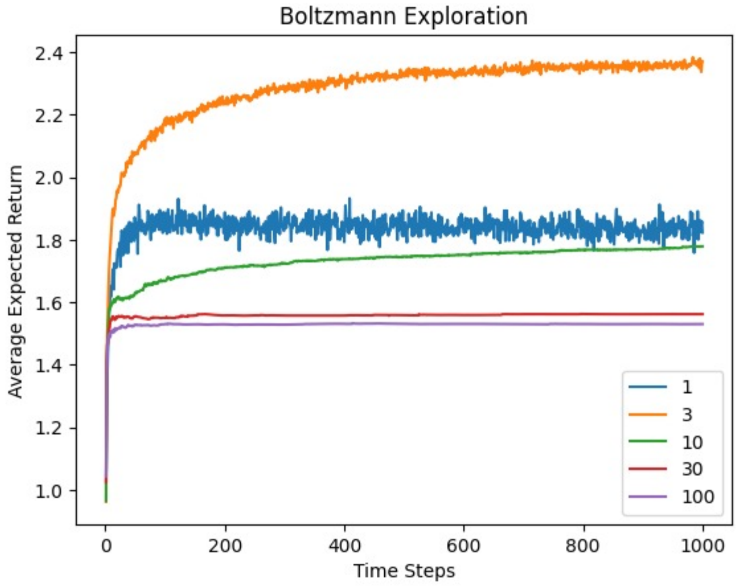
2.2 Optimistic Initialization Plot (8 pts)



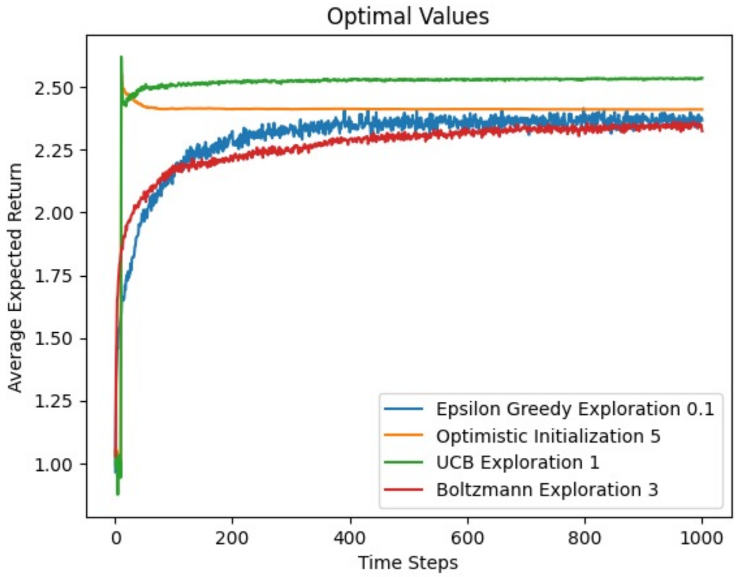
2.3 UCB Exploration Plot (8 pts)



2.4 Boltzmann Exploration Plot (8 pts)



2.5 Comparison Plot (8 pts)



2.6 Why not use the best-performing exploration strategy? (2-3 sentences) (4 pts)

The following highlight the reasons one should not the UCB algorithm:

- In this case, the rewards are bounded within a known range & Hoeffding's inequality is highly dependent on this characteristic
- Initially, sub optimal arms may be explored more frequently leading to larger exploration overhead
- This algorithm is not very scalable to an environment with a large number of actions

Problem 3: Feedback

Feedback: You can help the course staff improve the course by providing feedback. What was the most confusing part of this homework, and what would have made it less confusing?

For our team of 3, working on the assignment together allowed us to clarify most of the blockers as a team effort.

- As a general blocker, it was harder for two group members to get used to working with gymnasium as they did not have prior experience. This was not of a huge concern as the third teammate was able to pitch in with their experience in working with gymnasium, allowing the other two to learn and gain experience through practice. Although, it would be nice to get some input prior to the assignment on the different tools that one can get familiarized with.
- For question-1, one of the biggest blockers faced was trying to understand how to iterate between states when starting out with synchronous policy evaluation. After going through the documentation, we were able to understand that this can be done through a *transient dynamic* matrix.
- For the second question, one of the major blockers faced was trying to understand how to implement action selection for the UCB Exploration algorithm. We were stumped with how to go ahead as we were always getting a divided by zero error. We were able to solve this by adding a very small value to n (counts for each arm)

Collaboration: Detail the work division amongst your group in detail below.

Question-1:

- Contraction Mapping (theory question): Madhusha, Shrudhi, Siddharth
  - The answers were ideated and discussed by the team
  - The answers were filled out into the assignment by Madhusha
- General problem breakdown and approach handling: Shrudhi, Madhusha
  - This effort involved team members understanding the ”ask” of the problem. Writing out the pseudo-code of policy evaluation
  - Team members began to start getting code in by understanding how the gymnasium API works
- Initial code for synchronous policy evaluation: Madhusha, Siddharth, Shrudhi
  - This effort involved getting the basic implementation of the synchronous policy evaluation in place
  - As a team, we were able to get that done by understanding the different API calls that can be made through gymnasium and getting the pre-ideated code into the implementation
- Implemented by team member:
  - Synchronous Value and Policy iteration: Siddharth
  - Asynchronous Policy Iteration - Ordered & Randperm: Siddharth
  - Asynchronous Value Iteration - Ordered & Randperm: Shrudhi, Madhusha
- Aysnchronous Value Iteration - Custom: Shrudhi, Madhusha, Siddharth
  - Formulated the logic for this task and implemented the code through pair-programming

Question-2:

- Understanding the problem and getting a fair understanding of how to implement: Madhusha, Shrudhi, Siddharth
  - This effort involved re-visiting how exploration works with bandits and mapping out a plan of action for this question of the assignment
- ϵ-greedy exploration, optimistic initialization, UCB exploration, Boltzmann exploration: Shrudhi, Madhusha, Siddharth
  - Team worked together to understand the theory behind each algorithm and had supported each other during implementation
  - Madhusha had implemented -greedy exploration, optimistic initialization
  - Shrudhi had implemented UCB exploration, Boltzmann exploration
  - Siddharth had worked on supporting both of the above implementations by ensuring it aligns with the theory & the right coding practices were involved
  - Team members had worked together to implement the plotting function and running at different instances to get the required plots for the assignment
  - Siddharth had worked on cleaning up the code and Madhusha had worked on optimising the *plotAlgorithms* function

Time Spent: How many hours did you spend working on this assignment? Your answer will not affect your grade.

Alone	0 hours
With teammates	25 hours
With other classmates	0 hours
At office hours	0 hours