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 10 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:	Kimberly Nestor	Andrew ID 1:	kimberln
Name 2:		Andrew ID 2:	
Name 3:		Andrew ID 3:	

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

1.1: Contraction Mapping (3 pts)

Solution

1. False, because we only need one use case in the set of possible solutions to break this theory
2. True, because there is guaranteed convergence
3. True, because π_k and π_{k+1} would be the same if the algorithm reached convergence and $\pi_k = \pi_*$

1.2.1 Table: Policy Iteration (4 pts)

Environment	**# Policy Improvement Steps**	**Total # Policy Evaluation Steps**
Deterministic-4x4	9	19
Deterministic-8x8	17	40

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

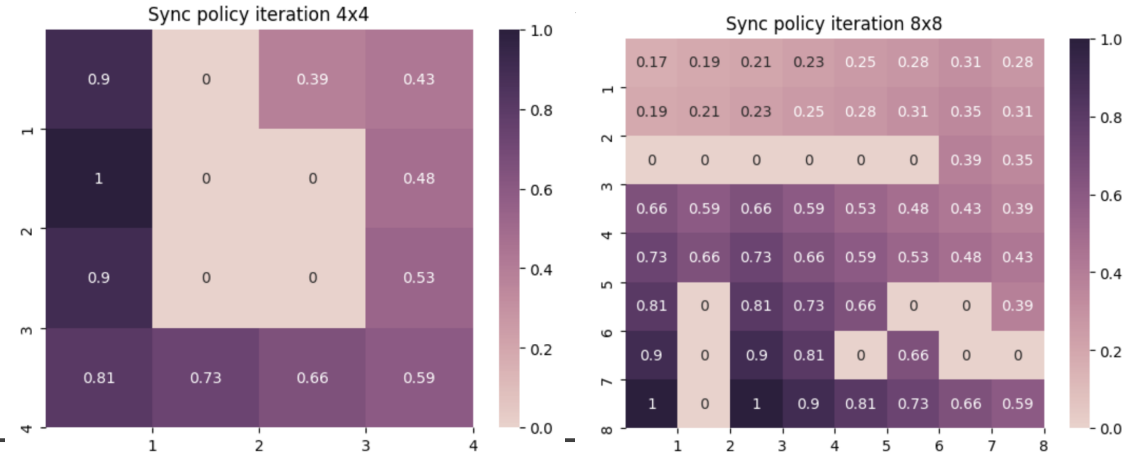
Solution

4x4 grid
DURD
RUUD
UUUD
ULLL

8x8 grid
RRRRRRDD
RRRRRRDD
UUUUUUDD
DRDDDDDD
DRDDDLLL
DUDDLUUU
DUDDUDUU
RULLLLLL

1.2.3 Value Functions of the Optimal Policies (2 pts)

Solution

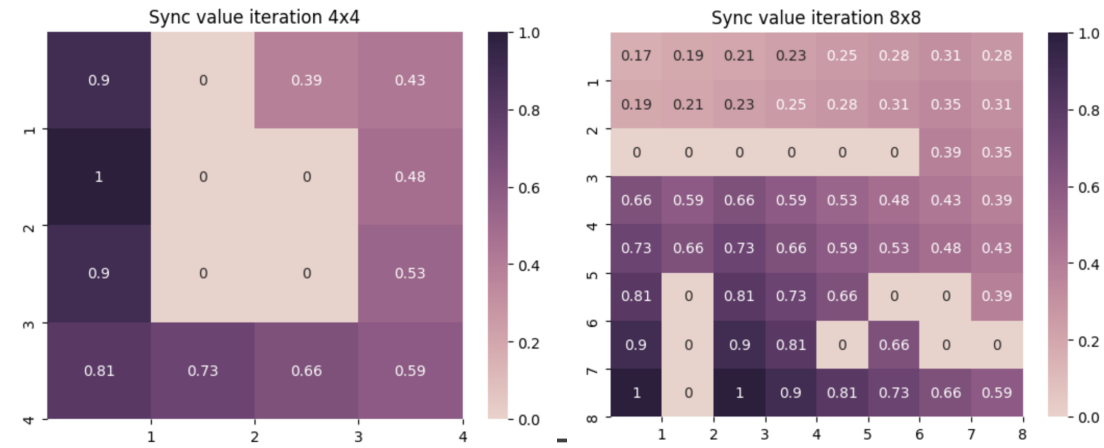


1.3.1 Table: Synchronous Value Iteration (3 pts)

Environment	# Iterations
Deterministic-4x4	6
Deterministic-8x8	15

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 grid
DURD
RUUD
UUUD
ULLL

8x8 grid
RRRRRRDD
RRRRRRDD
UUUUUUDD
DRDDDDDD
DRDDDLLL
DUDDLUUU
DUDDUDUU
RULLLLLL

1.4.1 Table: Asynchronous Policy Iteration (4 pts)

Heuristic	Policy Improvement Steps	Total Policy Evaluation Steps
Ordered	8x8 = 17	8x8 = 40
Randperm	8x8 = 17	8x8 = 40

1.5.1 Table: Asynchronous Value Iteration (4 pts)

Heuristic	# Iterations
Ordered	8x8 = 15
Randperm	8x8 = 13

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

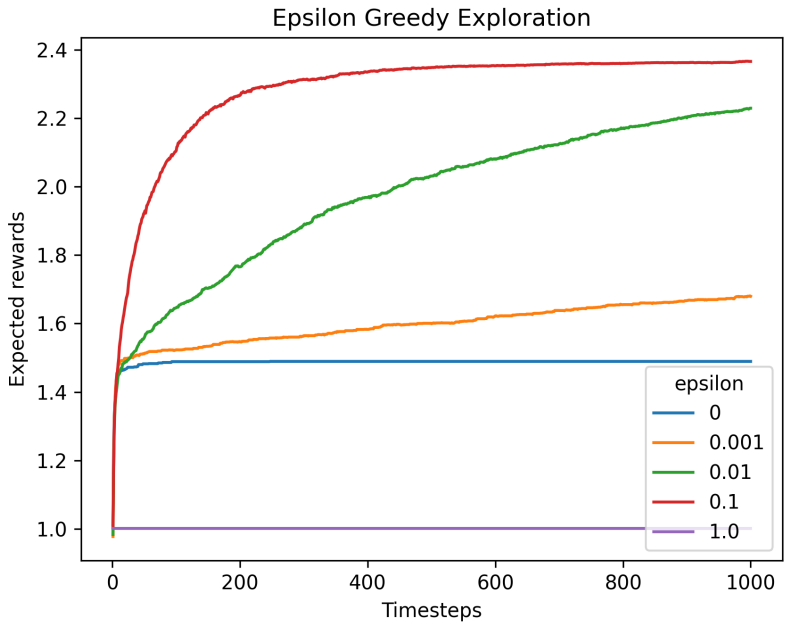
Solution

Env	# Iterations
Deterministic-4x4	6
Deterministic-8x8	14

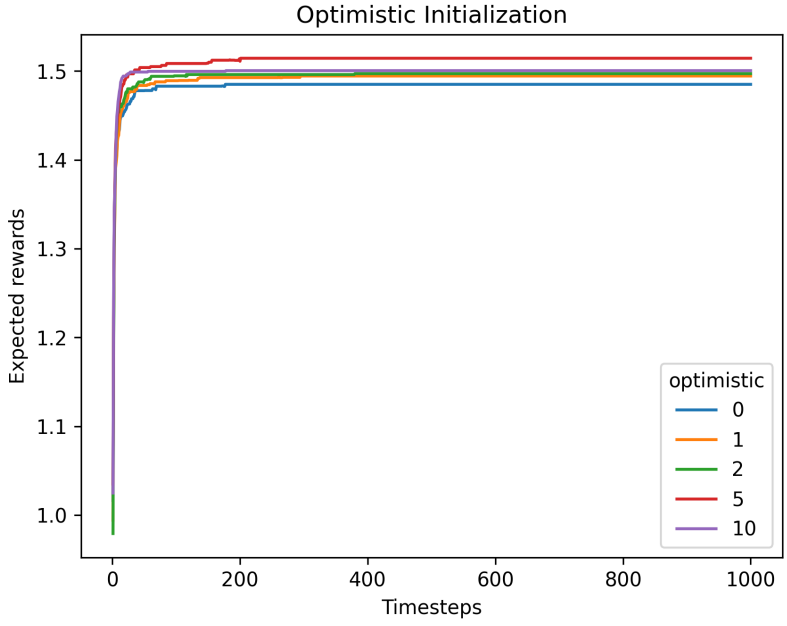
This custom heuristic that uses Manhattan distance to determine update order for states based on distance to goal would be best in a setting where the agent is given a limit number of time or timesteps (i.e. movement actions) in order to solve the goal of the game or reach the end target. In this sense only states that are closest to the goal and most essential will have value updates first and the algorithm can converge faster i.e. in fewer timesteps.

Problem 2: Bandits (36 pts)

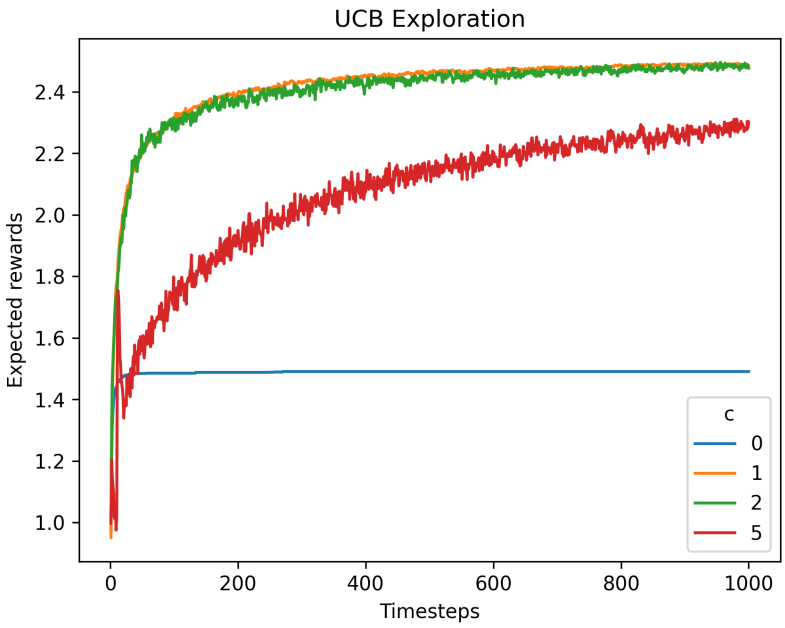
2.1 ϵ-Greedy Plot (8 pts)



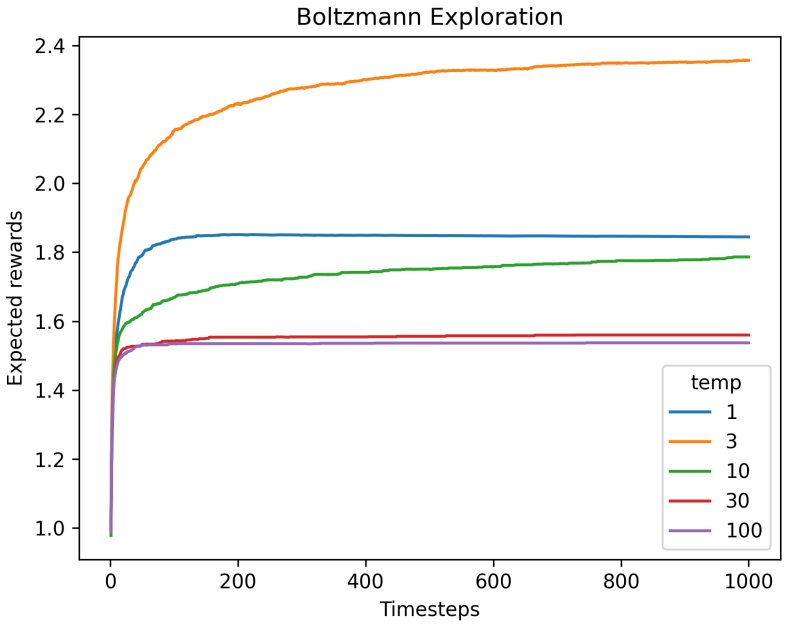
2.2 Optimistic Initialization Plot (8 pts)



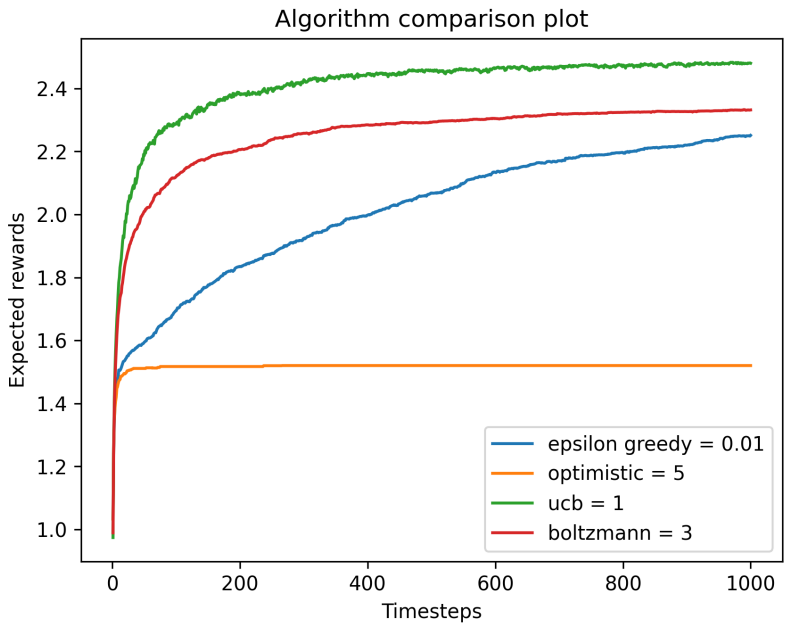
2.3 UCB Exploration Plot (8 pts)



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)

In this experiment UCB converges first, followed by Bolztmann and then epsilon greedy algorithms. The best choice of the four algorithms depends on the task that is to be solved. We could select UCB for the task since it is the best performing algorithm in this set, but in the timesteps greater than 1000 not sampled here the algorithm could hit the upper limit of performance while epsilon greedy, which is steadily increasing could surpass UCB in performance. Additionally the task may require that the algorithm improv over time instead of hitting expert status early on in training for example in learning to play a game.

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 problem 2 it would be helpful to but the formulas for all the algorithms in the assignment like with the Boltzmann question.
For problem 1.5.2 it is a little confusing to understand, maybe needs to be rephrased or more detail is needed.

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

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

Alone	30hrs
With teammates	
With other classmates	
At office hours	