Mini-Project 1: Sheep & Wolves (Spring 2023)

In this mini-project, you'll implement an agent that can solve the Sheep and Wolves problem for an arbitrary number of initial wolves and sheep. You will submit the code for solving the problem to the Mini-Project 1 assignment in Gradescope. You will also submit a report describing your agent to Canvas. Your grade will be based on a combination of your report (50%) and your agent's performance (50%).

About the Project

The Sheep and Wolves problem is identical to the Guards & Prisoners problem from the lecture, except that it makes more semantic sense why the wolves can be alone (they have no sheep to eat). Ignore for a moment the absurdity of wolves needing to outnumber sheep in order to overpower them. Maybe it's baby wolves vs. adult rams.

As a reminder, the problem goes like this: you are a shepherd tasked with getting sheep and wolves across a river for some reason. If the wolves ever outnumber the sheep on either side of the river, the wolves will overpower and eat the sheep. You have a boat, which can only take one or two animals in it at a time, and must have at least one animal in it because you'll get lonely (and because the problem is trivial otherwise). How do you move all the animals from one side of the river to the other?

In the original Sheep & Wolves (or Guards & Prisoners) problem, we specified there were 3 sheep and 3 wolves; here, though, your agent should be able to solve the problem for an arbitrary number of initial sheep and wolves. You may assume that the initial state of the problem will follow those rules (e.g. we won't give you more wolves than sheep to start). However, not every initial state will be solvable; there may be combinations of sheep and wolves that cannot be solved.

You will return a list of moves that will solve the problem, or an empty list if the problem is unsolvable based on the initial set of Sheep and Wolves. You will also submit a brief report describing your approach.

Your Agent

To write your agent, download the starter code below. Complete the `solve()` method, then upload it to Gradescope to test it against the autograder. Before the deadline, make sure to select your best performance in Gradescope as your submission to be graded.

Starter Code

Here is your starter code: SemanticNetsAgent.zip. You may also obtain it from the course Github repository.

The starter code contains two files: SemanticNetsAgent.py and main.py. You will write your agent in SemanticNetsAgent.py. You may test your agent by running main.py. You will only submit SemanticNetsAgent.py; you may modify main.py to test your agent with different inputs.

In SemanticNetsAgent.py, your `solve()` method will have two parameters: the number of sheep and the number of wolves. For example, for the original Sheep & Wolves problem from the lectures, we would call your agent with `your_agent.solve(3, 3)`. You may assume that the initial state is valid (there will not be more Wolves than Sheep in the initial state).

Returning Your Solution

Your `solve()` method should return a list of moves that will result in the successful solving of the problem. These are *only* the moves your agent ultimately selected to be performed, not the entire web of possible moves. Each item in the list should be a 2-tuple where each value is an integer representing the number of sheep (the first integer) or wolves (the second integer) to be moved; we assume the moves are alternating. So, if your first move is (1, 1), that means you're moving one sheep and one wolf to the right. If your second move is (0, 1), that means you're moving one wolf to the left.

For example, one possible solution to the test case of 3 sheep and 3 wolves would be:

```
[(1, 1), (1, 0), (0, 2), (0, 1), (2, 0), (1, 1), (2, 0), (0, 1), (0, 2), (0, 1), (0, 2)
```


The result of running the moves in order should be (a) that all animals are successfully moved from left to right, and (b) that all intermediate states along the way are valid (wolves never outnumber sheep in any state).

Submitting Your Solution

To submit your agent, go to the course in Canvas and click Gradescope on the left side. Then, select CS7637 if need be.

You will see an assignment named Mini-Project 1. Select this project, then drag your SemanticNetsAgent.py file into the autograder. If you have multiple files, add them to a zip file and drag that zip file into the autograder.

When your submission is done running, you'll see your results.

How You Will Be Graded

Your agent will be run against 20 initial configurations of sheep and wolves. 7 of these will be the same every time your agent is tested: (1, 1), (2, 2), (3, 3), (5, 3), (6, 3), (7, 3), and (5, 5). The other 13 will be semi-randomly selected, up to 25 of each type of animal, with sheep always greater than or equal to the number of wolves.

You can earn up to 40 points. You will earn 1 point for each of the 20 configurations you solve correctly (meaning that your solution does in fact move all the animals to the right side), and an additional point for each of the 20 configurations you solve *optimally* (in the minimum number of moves). For every case that you correctly label as unsolvable (by returning an empty list), you will receive 2 points as well.

You may submit up to 40 times prior to the deadline. The large majority of students do not need nearly that many submissions, so do not feel like you should use all 40; this cap is in place primarily

to prevent brute force methods for farming information about patterns in hidden test cases or submitting highly random agents hoping for a lucky submission. Note that Gradescope has no way for us to increase your individual number of submissions, so we cannot return submissions to you in the case of errors or other issues, but you should have more than enough submissions to handle errors if they arise.

You **must** select which of your submissions you want to count for a grade prior to the deadline. Note that by default, Gradescope marks your last submission as your submission to be graded. We cannot automatically select your best submission. Your agent score is worth 50% of your overall mini-project grade.

Your Report

In addition to submitting your agent to Gradescope, you should also write up a short report describing your agent's design and performance. Your report may be up to 4 pages, and should answer the following questions:

- How does your agent work? How does it generate new states, and how does it test them?
- How well does your agent perform? Does it struggle on any particular cases?
- How efficient is your agent? How does its performance change as the number of animals rises?
- Does your agent do anything particularly clever to try to arrive at an answer more efficiently?
- How does your agent compare to a human? Does your agent solve the problem the same way you would?

You are encouraged but not required to include visuals and diagrams in your four page report. The primary goal of the report is to share with your classmates your approach, and to let you see your classmates' approaches. You may include code snippits if you think they are particularly novel, but please do not include the entirety of your code.

> **Tip:** Remember, we want to see how you put the content of this class into action when designing your agent. You don't need to use the principles and methods from the lectures precisely, but we want to see your knowledge of the content reflected in your terminology and your reflection.

Submission Instructions

Complete your assignment using JDF, then save your submission as a PDF. Assignments should be submitted to the corresponding assignment submission page in Canvas. You should submit a **single** PDF for this assignment. This PDF will be ported over to Peer Feedback for peer review by your classmates. If your assignment involves things (like videos, working prototypes, etc.) that cannot be provided in PDF, you should provide them separately (through OneDrive, Google Drive, Dropbox, etc.) and submit a PDF that links to or otherwise describes how to access that material.

This is an individual assignment. All work you submit should be your own. Make sure to cite any sources you reference, and use quotes and in-line citations to mark any direct quotes.

Late work is not accepted without advanced agreement except in cases of medical or family emergencies. In the case of such an emergency, please contact the Dean of Students.

Grading Information

Your report is worth 50% of your mini-project grade. As such, your report will be graded on a 40-point scale coinciding with a rubric designed to mirror the questions above. Make sure to answer those questions; if any of the questions are irrelevant to the design of your agent, explain why.

Peer Review

After submission, your assignment will be ported to Peer Feedback for review by your classmates. Grading is *not* the primary function of this peer review process; the primary function is simply to give you the opportunity to read and comment on your classmates' ideas, and receive additional feedback on your own. All grades will come from the graders alone. See the course participation policy for full details about how points are awarded for completing peer reviews.