Koç University
COMP341
Introduction to Artificial Intelligence
Assignment 5

Instructor: Barış Akgün
Due Date: January 24 2020, 10:00 (in the morning, no late submissions)
Submission Through: Blackboard
Make sure you read and understand every part of this document
Important: Download your submission to make sure it is not corrupted and it has your latest report/code. You are only going to be graded by your blackboard submission.

This programming assignment will test your knowledge and your implementation abilities of what you have learned in the reinforcement learning part of the class.You are asked to complete a coding part and answer a few questions about how it runs. The coding part of the homework will follow the Berkeley CS188 Fall 2018 pacman project P3: Reinforcement Learning at `https://inst.eecs.berkeley.edu/~cs188/fa18/project3.html`. The questions for the report part are given in this document.

This homework must be completed individually. Discussion about algorithms, algorithm properties, code structure, and Python is allowed but group work is not. Coming up with the same approach and talking about the ways of implementation leads to very similar code which is treated as plagiarism! Furthermore, do not discuss the answers directly as it will lead to similar sentences which is treated as plagiarism. If you are unsure, you should not discuss. Any academic dishonesty, will not be tolerated. **By submitting your assignment, you agree to abide by the Koç University codes of conduct.**

You may find yourself having trouble implementing the coding part. In this case, we are going to let you use someone else's code to answer the given questions, as long as you credit the person or the website you take the code from. If you chose this option, we are only going to grade your report.

Grading

You are given two options about submitting your homework: (1) both code and report, and (2) only report. The second option is given to you in case you are not able to implement the programming part. These options will be graded differently:

- **Code and Report:** The code part and the report will have 2:1 weight ratio in your submission (programming 2/3, report 1/3).
- **Only Report:** The report part will be treated as 1/2 of the total grade. You **must** credit the code you used and **must not** submit a code part.

The solution code for the homeworks can be found online. We are going to compare your submission with these sources. We are also going to compare your code to previous submissions of Koç students. If your code's similarity level is above a certain threshold, your code will be scrutinized. If we see any plagiarism, you will lose points in the best case and disciplinary action will be taken against you in the worst.

Part 1: Programming

You are going to do the 10 programming questions about reinforcement learning given in the website. You are only required to change *valueIterationAgents.py*, *qlearningAgents.py* and *analysis.py*. If you have any issues with other parts of the code let your instructor or TA know ASAP, even if you manage to solve your problem. Use the data structures in *util.py* for the autograder to work properly. Some of these are really useful as well! If the you think you have the right answer but the autograder is not giving you any points, try to run it on individual questions (examples on how to do this is given in the website).

The website of the homework and the comments in the code have very useful tips and explanations. Do not skip them!

Hints

The questions are fairly straightforward. We have not covered Q4 and Q5 in class, but the website has enough information to implement them. Feel free to complete the other questions and get back to these later.

util.Counter()
The `util.Counter()` data structure is very useful. It is a modified dictionary, returning 0 for keys that are not in the dictionary. Use this to your advantage.

Calculating the max and the argmax for Q-Values
I recommend that you write a separate function, which returns both the maximum Q-value given a state and corresponding action (argmax). This will be useful in many parts of your code. Note that you can return two values from a function in Python. You should do this both for value iteration and q-learning agents.

Terminal States
You do not need to calculate anything for the terminal states, do not forget to handle them accordingly! There is a function to test whether a state is terminal or not.

Asynchronous Value Iteration
You need to iterate over the states in a cyclic manner for the given number of iterations. Use the modulo operator to do this.

Part 2: Report

This part includes answering the following questions based on your code's output. You are expected to answer the questions concisely. Five sentences is more than enough for most of them. Limit yourself to 300 words on average (1 question may require more). It is okay if you over-generalize, as long as your direction is clear and correct.
If you get different outputs then the provided ones, feel free to use them in your answers. Make sure you include them in your report!

Create a PDF file named *report.pdf* containing your answers for submission. **Write your name and your number on the report as well!**

Written Q1:
Why are all the agents trained in this homework called "Reflex Agents"? Why do we call the procedure of value iteration "offline planning"?

Written Q2:
For the programming Question 2, did you change the discount factor or the noise parameter and why?

Written Q3:
For each part of question 3, write down your parameter selections and explain how you decided on them. If one or more parametres do not matter, you do not need to talk about them.

Written Q4:
Run the following commands, preferably on two separate command lines to be able look at them side-by-side (or take screen shots).

```
python gridworld.py -a value -i 100 -k 100
python gridworld.py -a q -k 100
```

Press the "q" button when the new window is open to view the q-values or contact us if this does not happen. Compare the calculated q-values between these commands. Can you explain the reasons of the

main differences?

Written Q5:
Explain your answer to question 8.

Written Q6:
Run the following, wait for training (might take a while) to be done and observe the pacman behavior.

```
python pacman.py -p PacmanQAgent -x 5000 -n 5010 -l mediumGrid
```

Why does tabularized q-learning work for samller grids but not the larger one?

Written Q7:
Look at the `SimpleExtractor` under the *featureExtractors.py* file. Then run your q-learning solution and observe pacman. Discuss pacman's behavior based on the selected features.
For personal satisfaction, I recommend you add your features from homework 2 and watch pacman learn!

Submission

You are going to submit a compressed archive through the blackboard site. The file should extract to a folder with your student ID without the leading zeros. This folder should only contain *report.pdf*, *valueIterationAgents.py*, *qlearningAgents.py* and *analysis.py*. Other files will be deleted and/or overwritten. Do not submit any code if you only want us to grade your report.
Important: Download your submission to make sure it is not corrupted and it has your latest report/code. You are only going to be graded by your blackboard submission.

- **Important:** Download your submission to make sure it is not corrupted and it has your latest report/code. You are only going to be graded by your blackboard submission. We are not going to accept a version on your harddrive or cloud storage
- You are going to submit a **single** compressed archive through the blackboard site. The file can have *zip*, *rar*, *tar*, *tar.gz* or *7z* format.
- You are fine as long as the compressed archive has the required files (within 4 folder levels).
- Code that does not run (e.g. due to syntax errors) or that does not terminate (e.g. due to infinite loops) will not receive any credits.
- Once you are sure about your assignment and the compressed file, submit it through Blackboard.
- Do not submit code that does not terminate or that blows up the memory.

Best of luck and happy coding!