Your Tasks

1. ***(TURN THIS IN, 5 points)*** *First, read the assignment specification and* ***estimate how long you think it will take you*** *and write it down.*
2. *Task 1: Understand the updates to the Q-value table via a toy MDP:*
    A. Investigate QLearningAgent.py's implementation to understand how Q-learning updates the table, as well as how it uses the table to select actions.
    B. Run RlLabTest1 in main.py to see the updates of the Q-value table.
    C. Understand the policy (what action the agent selects in each state) and the Q-value table (the value of taking a particular action in a particular state, and then following the policy). You may find it helpful to run RlLabTest2 to see a table that is randomly initialized to see each update.
    D. **(TURN THIS IN, 10 points)** Provide a written interpretation comparing and contrasting the learned values and policy after the 1st trajectory with the final learned values and policy (after the 2nd trajectory).
3. *Task 2: Understand behavior of Q-learning agent on a parking MDP :*
    A. Run RlLabTest3 in main.py to understand the the steps for training of the Q-learning on a small Parking MDP
    B. **(TURN THIS IN, 10 points)** Provide a written interpretation comparing and contrasting the learned values and policy after the 1st trajectory with the final learned values and policy (after the second trajectory). Be sure to discuss differences and similarities with what you observed in Task 1 part D.
4. *Task 3: Understand behavior of Q-learning agent on a random MDP:*
    A. Run RlLabTest4 to see a more full training a Q-learning Agent on a random MDP. This test actually draws enough samples to hope to converge the values.
    B. **(TURN THIS IN, 10 points)** Provide a written interpretation of the output you see, in terms of the trend in reward, as well as variability in rewards (standard deviation is printed in your console, but does not appear on the learning curve).
5. *Task 4: Investigate the impacts of the hyperparameters on the performance of Q-learning:*
    A. Run RlLabTest5 to run Q-learning algorithms with three different probabilities of making a greedy action choice (probGreed
    B. Run RlLabTest6 to see how Q-learning behaves with three different learningRate values.
    C. **(TURN THIS IN, 15 points)** Provide a written interpretation comparing and contrasting what you observe in your learning curves where we vary greed and learning rate hyperparameters. As

you answer, consider how the picture from Jeremy Jordan in the slidesLinks to an external site. will manifest in a learning curve.

6. *Task 5: Test the Q-learning on different MDPs (previously we had held the MDP fixed):*
    A. Run RlLabTest7 to see how a Q-learning agent performs on different Parking MDPs of varying difficulty (note that the difference between test7 and test4 is that test7 varies BOTH agent and MDP.)
    B. **(TURN THIS IN, 15 points)** Provide a written interpretation about what you see in your charts. Be sure to discuss how parameters defining the MDP affect the Q-learning as well as how hyperparameters controlling the Q-Learning Agent affect performance on different MDPs.
7. **(TURN THIS IN**) *Task 6: Leading toward Line and Grid Search:* We have now seen how to vary MDP parameters, as well as test multiple agents of varying types. Perform a more comprehensive evaluation by doing programming-by-example to create two of your own test functions based on RlLabTest8 that compare the Q-Learning agents with different hyperparameters on different MDPs (Note: you will not need to change much). You should use at least 3 different combinations of values for each parameter while varying each of the following:
    - (15 points) The agent's discount factor (this parameter manages the tradeoff between instant gratification and delayed gratification)
    - (15 points) The reward structure (this is defined by the 3 variables named parkedReward, crashPenalty, waitingPenalty). Be sure to elicit different behaviors from your agent (e.g., find a reward structure where the agent should prefer to crash, or one where it should prefer to not wait)
    - Note that this should generate ~6 graphs, each with 3 lines, though there may be some clever ways to combine them. Please submit your graphs in addition to interpretations of what you see in them.
8. **(TURN THIS IN, 5 points)** Upon completing the lab, determine how long you actually spent on the lab, and report that timeframe in addition to your estimate beforehand.

Submit

A file that is readable (pdf, docx, etc) containing your charts, explanations, and test function(s).