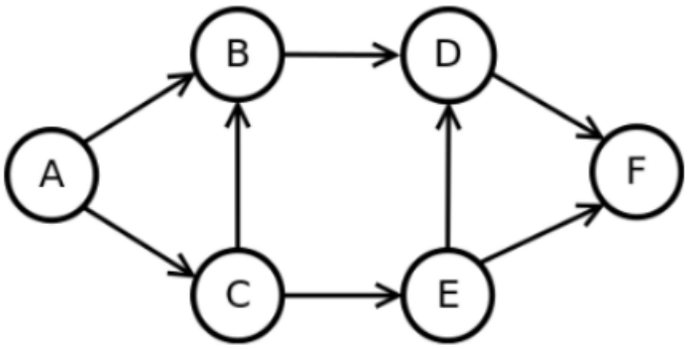
Value Iteration Convergence

We will consider a simple MDP that has six states, A, B, C, D, E, and F. Each state has a single action, *go*. An arrow from a state x to a state y indicates that it is possible to transition from state x to next state y when *go* is taken. If there are multiple arrows leaving a state x, transitioning to each of the next states is equally likely. The state F has no outgoing arrows: once you arrive in F, you stay in F for all future times. The reward is one for all transitions, with one exception: staying in F gets a reward of zero. Assume a discount factor = 0.7. We assume that we initialize the value of each state to 0. (Note: you should not need to explicitly run value iteration to solve this problem.)



After how many iterations of value iteration will the value for state C have become exactly equal to the true optimum? (Enter inf if the values will never become equal to the true optimal but only converge to the true optimal.)

3

How many iterations of value iteration will it take for the values of all states to converge to the true optimal values? (Enter inf if the values will never become equal to the true optimal but only converge to the true optimal.)

4

When value iteration is run on the given MDP, we try to approximate values for each state in the MDP.

Since from F no action is possible, we have the value of F to be 0 which is same as its tru value.

Value of D depends on F, and gets set in the first iteration.

Values of B and E depend on D and hence converge to optimal values in the second iteration.

Value of C depends on B and E, and hence settles in the third iteration.

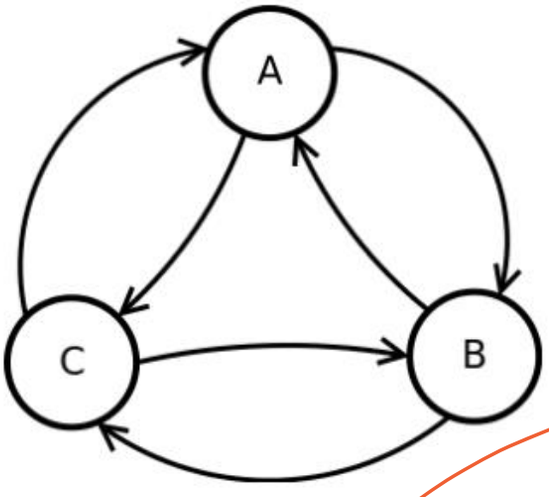
Since the value of A depends on C, it settles in the 4th iteration.

Hence it takes **4 iterations.**

=layer

Consider the following transition diagram, transition function and reward function for an MDP.

Discount Factor, $\gamma = 0.5$



s	a	s'	T(s,a,s')	R(s,a,s')
A	Clockwise	B	0.6	-1.0
A	Clockwise	C	0.4	-1.0
A	Counterclockwise	B	0.2	1.0
A	Counterclockwise	C	0.8	0.0
B	Clockwise	C	1.0	0.0
B	Counterclockwise	A	1.0	-2.0
C	Clockwise	A	1.0	2.0
C	Counterclockwise	A	0.4	-1.0
C	Counterclockwise	B	0.6	-2.0

Suppose we are doing policy evaluation, by following the policy given by the left-hand side table below. Our current estimates (at the end of some iteration of policy evaluation) of the value of states when following the current policy is given in the right-hand side table.

A	B	C
Counterclockwise	Clockwise	Counterclockwise

$V_k^\pi(A)$	$V_k^\pi(B)$	$V_k^\pi(C)$
-0.440	-0.800	-1.560

Answers should keep 3 decimals.

What is $V_{k+1}^\pi(C)$? -1.938

Suppose that policy evaluation converges to the following value function, V_∞^π

$V_\infty^\pi(A)$	$V_\infty^\pi(B)$	$V_\infty^\pi(C)$
-0.724	-1.026	-2.053

What is Q_∞^π(C, clockwise)? 1.638

What is Q_∞^π(C, counterclockwise)? -2.053

What is the updated action for state C? Enter clockwise or counterclockwise. clockwise

Model Free RL

Consider an MDP with 3 states, A, B and C; and 2 actions Clockwise and Counterclockwise. We do not know the transition function or the reward function for the MDP, but instead, we are given with samples of what an agent actually experiences when it interacts with the environment (although, we do know that we do not remain in the same state after taking an action).

In this problem, instead of first estimating the transition and reward functions, we will directly estimate the Q function using Q-learning. Assume, the discount factor, γ is 0.5 and the step size for Q-learning, α is 0.5. Our current Q function, $Q(s,a)$, is as follows.

	A	B	C
Clockwise	0.273	-2.938	-0.557
Counterclockwise	0.344	2.098	0.063

The agent encounters the following samples.

s	a	s'	r
A	Clockwise	B	0.0
B	Counterclockwise	A	2.0
C	**Clockwise**	**A**	**-1.0**

Process the samples given above. Below fill in the Q-values after both samples have been accounted for.

Answers should keep 3 decimals.

Clockwise: A 0.205 0.661 B -2.938 C 0.918 -0.613

Counterclockwise: A 0.334 B 2.574 2.214 C 0.063

$0.5 \times 0.273 + (0.0 + 2.098 \times 0.5) \times 0.5$

clockwise

clockwise

$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[\text{sample}]$

clockwise

$\text{sample} = R(s,a,s') + \gamma \max_{a'} Q(s',a')$

MDPs and RL Part 1

The agent is in a 2×4 gridworld as shown in the figure. We start from square 1 and finish in square 8. When square 8 is reached, we receive a reward of +10 at the game end. For anything else, we receive a constant reward of -1 (you can think of this as a time penalty).

1	2	3	4
5	6	7	8

The actions in this MDP include: up, down, left and right. The agent cannot take actions that take them off the board. In the table below, we provide initial non-zero estimates of Q values (Q values for invalid actions are left as blanks):

state\action	up	down	left	right
1		Q(1,down)=4		Q(1,right)=3
2		Q(2, down)=6	Q(2, left)=4	Q(2, right)=5
3		Q(3, down)=8	Q(3, left)=5	Q(3, right)=7
4		Q(4, down)=9	Q(4, left)=6	
5	Q(5, up)=5			Q(5, right)=6
6	Q(6, up)=4		Q(6, left)=5	Q(6, right)=7
7	Q(7, up)=6		Q(7, left)=6	Q(7, right)=8

In this question, please use 3 decimal points as the final answer. E.g. 0.300, 1.524.

(a) Your friend Adam guesses that the actions in this MDP are fully deterministic (e.g. taking down from 2 will land you in 6 with probability 1 and everywhere else with probability 0). Since we have full knowledge of T and R, we can thus use the Bellman equation to improve (i.e., further update) the initial Q estimates.

Adam tells you to use the following update rule for Q values, where he assumes that your policy is greedy and thus does $\max_a Q(s,a)$. The update rule he prescribes is as follows:

$$Q_{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$$

(i) Perform one update of Q(3, left) using the equation above, where $\gamma = 0.9$. You may break ties in any way.

4.400

问题 6

5 分

MDPs and RL Part 2

(ii) Perform one update of Q(3, down) using the equation above, where $\gamma = 0.9$.

6.200

MDPs and RL Part 3

(b) After observing the agent for a while, Adam realized that his assumption of T being deterministic is wrong in one specific way: **when the agent tries to legally move down, it occasionally ends up moving left instead** (except from grid 1 where moving left results in out-of-bound). All other movements are still deterministic.

Suppose we have run the Q updates outlined in the equation above until convergence, to get $Q^*_{wrong}(s,a)$ under the original assumption of the wrong (deterministic) T. Suppose $Q^*_{correct}(s,a)$ denotes the Q values under the new correct T. Note that you don't explicitly know the exact probabilities associated with this new T, but you know that it qualitatively differs in the way described above. As prompted below, list the set of (s,a) pairs where $Q^*_{wrong}(s,a)$ is either an over-estimate or under-estimate of $Q^*_{correct}(s,a)$.

(i) Select all (s,a) where $Q^*_{wrong}(s,a)$ is an over-estimate.

☐ (1, down) ☑ (1, right) ☑ (2, down) ☐ (2, left) ☑ (2, right) ☑ (3, down) ☑ (3, left) ☑ (3, right) ☑ (4, down) ☑ (4, left) ☐ (5, up) ☐ (5, right) ☑ (6, up) ☐ (6, left) ☐ (6, right) ☑ (7, up) ☐ (7, left) ☐ (7, right) ☐ None of the above

问题 8

MDPs and RL Part 4

(ii) Select all (s,a) where $Q^*_{wrong}(s,a)$ is an under-estimate.

☐ (1, down) ☐ (1, right) ☐ (2, down) ☐ (2, left) ☐ (2, right) ☐ (3, down) ☐ (3, left) ☐ (3, right) ☐ (4, down) ☐ (4, left) ☐ (5, up) ☐ (5, right) ☐ (6, up) ☐ (6, left) ☐ (6, right) ☐ (7, up) ☐ (7, left) ☐ (7, right) ☑ None of the above

MDPs and RL Part 5

(c) Suppose that we have a mysterious oracle that can give us either all the correct Q-values $Q(s,a)$ or all the correct state values $V(s)$. Which one do you prefer to be given if you want to use it to find the optimal policy?

- (●) Q-values
- ( ) State values

TD: $V^{\pi}(s) \leftarrow (1-\alpha) V^{\pi}(s) + \alpha[R(s,\pi(s),s') + \gamma V^{\pi}(s')]$

问题 10

MDPs and RL Part 6

(d) Suppose that you perform actions in this grid and observe the following episode: 3, right, 4, down, 8 (terminal).

With learning rate $\alpha = 0.2$, discount $\gamma = 0.9$, perform an update of $Q(3, right)$ and $Q(4, down)$. Note that here, we update Q values based on the sampled actions as in TD learning, rather than the greedy actions.

$Q'(3, right) =$ 7.02

$Q'(4, down) =$ 9.2

$0.8 \times [\quad] + 0.2 \times [-1 + 0.9 \times 8] = 7.02$

3, right　　4, down

问题 11

MDPs and RL Part 7

(e) One way to encourage an agent to perform more exploration in the world is known as the "ϵ-greedy" algorithm. For any given policy $\pi(s)$, this algorithm says to take the original action $a = \pi(s)$ with probability $(1-\epsilon)$, and to take a random action (drawn from a uniform distribution over all legal actions) with probability ϵ. If ϵ can be tuned, would you assign it to be a high or low value at the beginning of the training? What about at the end of the training?

A. ▾ at the beginning of the training

B. ▾ at the end of the training

A. a high value

B. a low value

MDPs and RL Part 8

(f) Instead of using the "ϵ-greedy" algorithm, we will now do some interesting exploration with softmax. We first introduce a new type of policy: A stochastic policy $\pi(a|s)$ represents the probability of action a being prescribed, conditioned on the current state. In other words, the policy is now a distribution over possible actions, rather than a function that outputs a deterministic action.

Let's define a new policy as follows:

$$\pi(a|s) = \frac{e^{Q(s,a)}}{\sum_{a'} e^{Q(s,a')}}$$

(i) Suppose we are at square 3 in the grid and we want to use the originally provided Q values from the table. What is the probability that this policy will tell us to go right? What is the probability that this policy will tell us to go left? Note that the sum over actions prescribed above refers to a sum over legal actions.

The probability that this policy will tell us to go right: 0.259

The probability that this policy will tell us to go left: 0.035

问题 13

MDPs and RL Part 9

(ii) Which of the following statements are correct?

- [x] ϵ-greedy algorithm is **guaranteed** to converge to optimal Q function if every (s,a) is visited infinitely often and α is chosen to decay according to standard stochastic approximation requirements.
- [x] Softmax algorithm is **guaranteed** to converge to optimal Q function if every (s,a) is visited infinitely often and α is chosen to decay according to standard stochastic approximation requirements.
- [ ] In practice, softmax algorithm always shows better performance than ϵ-greedy algorithm and converges faster.
- [ ] None of the above.

Q value iter: $Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$

MDPs and RL Part 10

(g) Your friend Cody argues that we could still explicitly calculate Q updates (like Adam's approach in part (a)) even if we don't know the true underlying transition function $T(s,a,s')$, because he believes that our T can be roughly approximated from samples.

Suppose you collect 1,000 transitions from s = 3, a = Down, in the form of (s_{start}, a, s_{end}). Use these samples to compute $T_{approx}(s=3, a=Down, s')$, which is an approximation of the true underlying (unknown) $T(s,a,s')$.

$(s=3, a=Down, s'=6)$	$(s=3, a=Down, s'=7)$
99	901

Perform one step of q-value iteration based on your transition model computed above.

$Q'(3, down) =$ ~~5.4872~~ 6.111

$\frac{99}{1000}(-1 + 8 \times 0.9) + \frac{901}{1000}(4 + 7 \times 0.9)$

(7, max action)

(6, max action)

Feature-Based Representation

Consider the following feature based representation of the Q-function:

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a)$$

with

$f_1(s,a) = 1/(\text{Manhattan distance to nearest dot after having executed action } a \text{ in state } s)$

$f_2(s,a) = (\text{Manhattan distance to nearest ghost after having executed action } a \text{ in state } s)$

Part 1

Assume $w_1 = 1, w_2 = 10$. For the state s shown below, find the following quantities. Assume that the red and blue ghosts are both sitting on top of a dot.

$Q(s, West) =$

31

$Q(s, South) =$

11

) taken as just pacman move, ghost do not.

Part 2

Assume Pac-Man moves West. This results in the state s' shown below.

dot eaten considered.

The reward for this transition is $r = +10 - 1 = 9$ (+10: for food pellet eating, -1 for time passed). Fill in the following quantities. Assume that the red and blue ghosts are both sitting on top of a dot.

$Q(s', West) =$

11

$Q(s', East) =$

11

What is the sample value (assuming $\gamma = 1$)?

$\text{sample} = [r + \gamma \max_{a'} Q(s', a')] =$

20

Part 3

Now let's compute the update to the weights. Let $\alpha = 0.5$.

$\text{difference} = [r + \gamma \max_{a'} Q(s', a')] - Q(s, a) =$

-11

$w_1 \leftarrow w_1 + \alpha (\text{difference}) f_1(s, a) =$

-4.5

$w_2 \leftarrow w_2 + \alpha (\text{difference}) f_2(s, a) =$

-6.5

CS 188 Spring 2020 Introduction to Artificial Intelligence Written HW 2 Sol.

Solutions for HW 2 (Written)

Q1. [40 pts] MDP: Eating Chocolate

We have a chocolate bar of dimensions 1×8, which contains 8 squares. Most of these squares are delicious chocolate squares, but some of them are poison! Although the chocolate and poison squares are visually indistinguishable, someone has told us which ones are which. The layout for our chocolate bar is shown below with P indicating poison squares.

P				P		P	

Eating a chocolate square immediately gives a reward of 1 and eating a poison square immediately gives a reward of -2. **Starting from the right end of the bar**, there are 3 possible actions that we can take (at each step), and all of these actions cause non-deterministic transitions as follows:

- Action b_1: Try to bite 1 square, which will result in you actually eating 0, 1, or 2 squares with equal probability.
- Action b_2: Try to bite 2 squares, which will result in you actually eating 1, 2, or 3 squares with equal probability.
- Action *Stop*: Stop biting, which will end the game definitively and result in no reward.

(a) [10 pts] Formulate this problem as an MDP.

States: Decide (and explain) how you want to represent a "state." Using that representation, list out all of your possible states in this MDP.

1 through 8 indicating the length of the remaining bar, Done state

Actions: bite 1, bite 2, stop

Transitions:

$$T(s, b_1, s') = \frac{1}{3}, s > 2, s' \in \{s, s-1, s-2\}$$

$$T(1, b_1, s') = \begin{cases} 1/3 \text{ if } s' = 1 \\ 2/3 \text{ if } s' = Done \end{cases}$$

$$T(s, b_2, s') = \frac{1}{3}, s > 3, s' \in \{s-1, s-2, s-3\}$$

$$T(2, b_2, s') = \begin{cases} 1/3 \text{ if } s' = 1 \\ 2/3 \text{ if } s' = Done \end{cases}$$

$$T(3, b_2, s') = \frac{1}{3}, s' \in \{1, 2, Done\}$$

$$T(1, b_2, Done) = 1$$

$$T(s, Stop, Done) = 1$$

$$T(s, a, s') = 0 \text{ otherwise}$$

Rewards: Skip the explicit formulation of the rewards for this problem.

Alternatively, the states can be represented as 1 through 8 where the state s indicates that you're on the sth square, starting from the right end of the bar. In this case, all the transitions should be modified accordingly:

$$T(s, b_1, s') = \frac{1}{3}, s < 7, s' \in \{s, s+1, s+2\}$$

$$T(8, b_1, s') = \begin{array}{l} 1/3 \text{ if } s' = 8 \\ 2/3 \text{ if } s' = Done \end{array}$$

$$T(s, b_2, s') = \frac{1}{3}, s < 6, s' \in \{s+1, s+2, s+3\}$$

$$T(7, b_2, s') = \begin{array}{l} 1/3 \text{ if } s' = 8 \\ 2/3 \text{ if } s' = Done \end{array}$$

$$T(6, b_2, s') = \frac{1}{3}, s' \in \{7, 8, Done\}$$

$$T(8, b_2, Done) = 1$$

$$T(s, Stop, Done) = 1$$

$$T(s, a, s') = 0 \text{ otherwise}$$

(b) Value Iteration

(i) [2 pts] Perform value iteration for 3 iterations, using $\gamma = 1$. What are the values for each state, after each of these iterations?

State	1	2	3	4	5	6	7	8
$V_0(s)$	0	0	0	0	0	0	0	0
$V_1(s)$	0	0	1	2	0	0	0	0
$V_2(s)$	0	0	4/3	7/3	0	1	0	0
$V_3(s)$	0	0	14/9	22/9	2/9	11/9	0	1/3

(ii) [1 pt] We consider value iteration to have converged by iteration k if $\forall s, V_{k+1}(s) = V_k(s)$. Did the values converge by iteration 2?

○ Yes ● No

(iii) [1 pt] Given that we know a $V(s)$ for all states, how do we extract a policy $\pi(s)$ from that value function? Your answer should be a one-line math expression which uses some or all of the following: $s, a, s', V, R, T, \gamma, \sum$, argmax, max.

$\pi(s) =$ $\text{argmax}_a \sum_{s'} T(s, a, s') * (R(s, a, s') + \gamma V(s'))$

(iv) [1 pt] What is the resulting policy after extracting it from the values $V_2(s)$? If applicable, write all of the valid actions for a given state.

State	1	2	3	4	5	6	7	8
$\pi_3(s)$	S	b_1, S	b_1	b_2	b_1, b_2	b_2	S	b_1, b_2

(v) [5 pts] Conceptual: In general, we consider a policy to have converged if, for all states s, $\pi(s)$ does not change with further calculation steps. Your friend Victor claims that **in any MDP setting, not limited to this game**, $\pi(s)$ can only be considered to be converged if the $V(s)$ that it came from was already converged. Is Victor right?

◯ Victor is right, and I will provide a proof below.
◯ Victor is wrong, and I will provide a counter example below.

Proof/counter example: wrong. TODO write a counter example where the V changes at the next iteration (to the true V) but the corresponding policies of those 2 V's are the same.

For the rest of the problem, assume we have deterministic transitions. This means that taking the b_1 action bites exactly 1 square and taking the b_2 actions bites exactly 2 squares. Note that if you take b_1, then your reward for that step is the reward associated with that one square, and if you take b_2, then your reward for that step is the sum of rewards from the two squares.

(c) **(i)** [6 pts] Let's say that you took a sequence of 3 single bites, so you are now in state $s = 5$. At this point, you can either stop or continue to bite. For what range of discount values γ would you choose to stop versus continue to bite? Show your work and explain your answer.

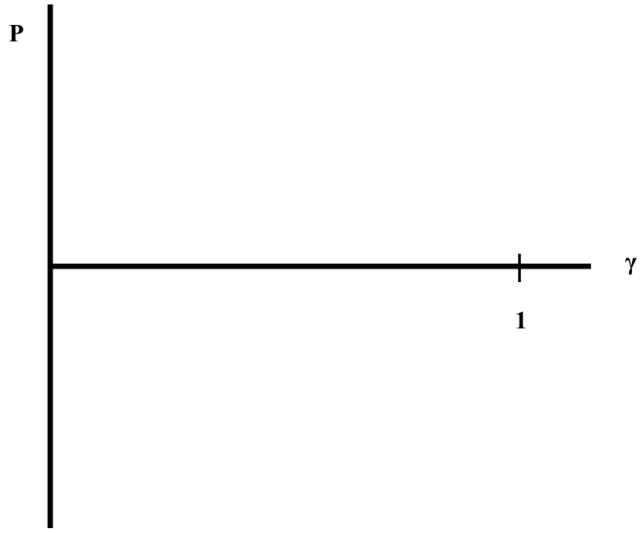
We will choose to stop if the discount is smaller than $\frac{1}{2}$ and continue to bite if it is greater. At $\gamma = \frac{1}{2}$, both produce equal reward, so either action can be taken. In either case, the discount factor must be between 0 and 1.

Explanation: If we stop at $s = 5$, we will receive no additional reward. If we continue to bite, we can receive the most additional reward by taking the b_2 action twice. This will result in the additional reward of: $r_b = (-2 + 1) + (1\gamma + 1\gamma) = -1 + 2\gamma$. Checking for when r_b is greater than 0 (i.e. when we should choose to bite instead of stop), we find that $\gamma \geq \frac{1}{2}$.

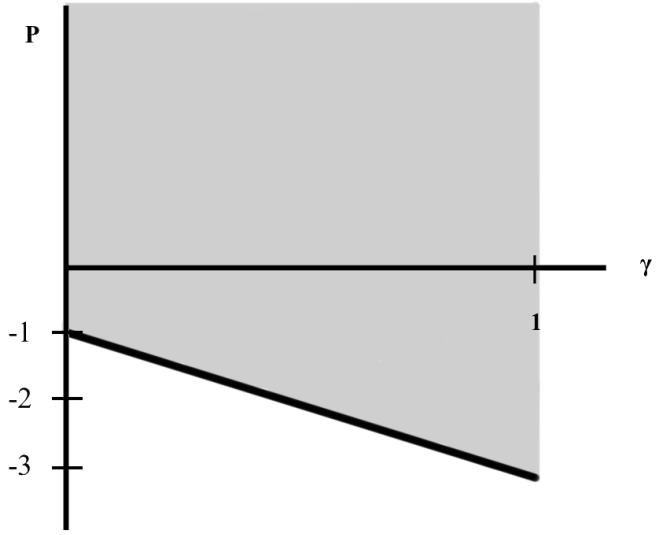
(ii) [4 pts] For this part, assume the same situation as the previous part, where you took 3 single bites already and you are now in state $s = 5$. Think about how the behavior of the optimal policy at this point changes, as we vary (a) the reward associated with eating a poison square and (b) the discount γ from 0 to 1. First, what's the mathematical condition that must hold for us to choose to continue to bite:

$$(P + 1) + \gamma(1 + 1) \geq 0$$

Plot this condition in the plot below by shading in the areas where we will continue to bite. Label important points, including the point where $P = -2$ with the threshold γ value you found in part (c.i).



Solutions:



(d) **(i)** [2 pts] Define/create a policy below such that performing value iteration on that policy would give these values indicated here. Use $\gamma = 1$

State	1	2	3	4	5	6	7	8
$V^{\pi}(s)$	0	-1	2	3	0	2	-1	0
$\pi(s)$	S	b_2	b_2	b_1	S	b_2	b_2	S

(ii) [3 pts] Is the previous policy optimal? Perform one step of policy iteration to justify your answer. Please show your work.

State	1	2	3	4	5	6	7	8
$\pi_{i+1}(s)$	S	b_1	b_2	b_1	b_1, b_2	b_2	b_1, S	b_2

For state 1: $V_{b_1}(1) = V_{b_2}(1) = -2$
For state 2: $V_{b_1}(2) = 1$, $V_S(2) = -1$
For state 3: $V_{b_1}(3) = 0$, $V_S(3) = 0$
For state 4: $V_{b_2}(4) = 1$, $V_S(4) = 0$
For state 5: $V_{b_1}(5) = 1$, $V_{b_2}(5) = 1$
For state 6: $V_{b_1}(6) = 1$, $V_S(6) = 0$
For state 7: $V_{b_1}(7) = 0$, $V_S(7) = 0$
For state 8: $V_{b_1}(8) = 0$, $V_{b_2}(8) = 1$

Was the policy in part (d.i) optimal: ◯ Yes ● No

(iii) [5 pts]

We can build a graph from an MDP as follows. The vertices are the states, and there is an edge from state s to state s' labelled with an action a if $T(s, a, s')$ is nonzero. We can fully represent the MDP and assign a weight of $T(s, a, s')$ and a value of $R(s, a, s')$ to the corresponding edge from s to s' in the graph. Note that there may be more than one edge from s to s' if there is more than one action with a positive transition probability between s and s'. That is, there may be parallel edges in the graph.

Indeed, the MDP about the going fast and slow in a car in lecture is illustrated as a graph in the slides.

Notice that when viewed as a graph the state transitions of the above MDP forms a directed acyclic graph (DAG). It is useful here to explicitly add a terminal state named 0. The terminal state is reached from any state using the S action and every other action from any state leads to a lower numbered state or the terminal state. (To be sure, this would be true as well for the probabilistic version if action b_1 had zero probability of eating nothing.)

Sketch an algorithm to produce the optimal policy for $\gamma = 1$ in time linear in the number of states and edges in the associated graph in an MDP where the transitions form a DAG. You may assume there is a single terminal state with no outgoing transitions. (Hint: the value of the terminal state is 0.) Can you also produce the optimal policy for an arbitrary $\gamma < 1$?

Since a state with smaller number can never transition into a state with larger number, the optimal value of / policy at a state with smaller number does not depend on those of a state with larger number.

Therefore, we can build the table for value / policy at the state with the smallest number (terminal state, equivalent to state 0), which has definite reward of 0, and we can only choose stop. Then we can get the value of / policy at state 1, which only depend on those of state 0. Then we can get the value of / policy at state 2, which only depend on those of state 0 and 1. etc. We can do this to construct the value of / policy at all states, starting from the states with smaller number and propagate up to the starting state (state 8)

Note that your algorithm should work with deterministic and probabilistic transition dynamics T. If your answer relys on T being deterministic, the answer is not completely correct.

Q2. [30 pts] MDPs and RL

The agent is in a 2×4 gridworld as shown in the figure. We start from square 1 and finish in square 8. When square 8 is reached, we receive a reward of $+10$ at the game end. For anything else, we receive a constant reward of -1 (you can think of this as a time penalty).

1	2	3	4
5	6	7	8

The actions in this MDP include: up, down, left and right. The agent cannot take actions that take them off the board. In the table below, we provide initial non-zero estimates of Q values (Q values for invalid actions are left as blanks):

Table 1

	action=up	action=down	action=left	action=right
state=1		Q(1, down)=4		Q(1, right)=3
state=2		Q(2, down)=6	Q(2, left)=4	Q(2, right)=5
state=3		Q(3, down)=8	Q(3, left)=5	Q(3, right)=7
state=4		Q(4, down)=9	Q(4, left)=6	
state=5	Q(5, up)=5			Q(5, right)=6
state=6	Q(6, up)=4		Q(6, left)=5	Q(6, right)=7
state=7	Q(7, up)=6		Q(7, left)=6	Q(7, right)=8

(a) Your friend Adam guesses that the actions in this MDP are fully deterministic (e.g. taking down from 2 will land you in 6 with probability 1 and everywhere else with probability 0). Since we have full knowledge of T and R, we can thus use the Bellman equation to improve (i.e., further update) the initial Q estimates.

Adam tells you to use the following update rule for Q values, where he assumes that your policy is greedy and thus does $\max_a Q(s,a)$. The update rule he prescribes is as follows:

$$Q_{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$$

(i) [1 pt] Perform one update of $Q(3, \text{left})$ using the equation above, where $\gamma = 0.9$. You may break ties in any way.

$-1 + 0.9 \times (1 \times 6)$ because we are in a deterministic grid world with a greedy policy.

(ii) [3 pts] For the Q update rule prescribed above, how does $Q_{k+1}(s,a)$ depend on $Q_k(s,a)$? How is this different from the normal Q learning update that we saw in lecture, which is $Q_{k+1}(s,a) = (1-\alpha)Q_k(s,a) + \alpha * \text{sample}$?

The Q estimate update does not use a learning rate, and thus does not directly depend on $Q_k(s,a)$

(b) After observing the agent for a while, Adam realized that his assumption of T being deterministic is wrong in one specific way: **when the agent tries to legally move down, it occasionally ends up moving left instead** (except from grid 1 where moving left results in out-of-bound). All other movements are still deterministic.

Suppose we have run the Q updates outlined in the equation above until convergence, to get $Q^*_{wrong}(s,a)$ under the original assumption of the wrong (deterministic) T. Suppose $Q^*_{correct}(s,a)$ denotes the Q values under the new correct T. Note that you don't explicitly know the exact probabilities associated with this new T, but you know that it qualitatively differs in the way described above. As prompted below, list the set of (s,a) pairs where $Q^*_{wrong}(s,a)$ is either an over-estimate or under-estimate of $Q^*_{correct}(s,a)$.

(i) [3 pts] List of (s,a) where $Q^*_{wrong}(s,a)$ is an over-estimate (and why):

All the (s, a) pair that has non-zero probability of landing in $s' = 2$, 3, or 4

This is because $V(2)$, $V(3)$, $V(4)$ will all end up being overestimation. So all (s, a) pair that uses 2,3,4 as s' will be overestimation

Side note: state 1, 5, 6, 7, 8 are not affected because the optimal value of those state can be obtained through a sequence of nodes without having to take a "down" action that is affected by this noisy failure

(ii) [3 pts] List of (s, a) where $Q^*_{wrong}(s, a)$ is an under-estimate (and why):

None

(c) [2 pts] Suppose that we have a mysterious oracle that can give us either all the correct Q-values $Q(s, a)$ or all the correct state values $V(s)$. Which one do you prefer to be given if you want to use it to find a greedy policy, and why?

Q. When you are using Q, you only need the value itself to determine a good action. However, with value function, you also need the transition function to determine this.

(d) [2 pts] Suppose that you perform actions in this grid and observe the following episode: 3, right, 4, down, 8 (terminal).

With learning rate $\alpha = 0.2$, discount $\gamma = 0.9$, perform an update of $Q(3, right)$ and $Q(4, down)$. Note that here, we update Q values based on the sampled actions as in TD learning, rather than the greedy actions.

$Q(3, right)$: 7*(1-0.2)+0.2*(-1+0.9*9)
$Q(4, down)$: 9*(1-0.2)+0.2*(10)

Please note that the -1 and 10 comes from the first paragraph of the problem statement (which describes the R(s, a, s')

(e) [2 pts] One way to encourage an agent to perform more exploration in the world is known as the "ϵ-greedy" algorithm. For any given policy $\pi(s)$, this algorithm says to take the original action $a = \pi(s)$ with probability $(1-\epsilon)$, and to take a random action (drawn from a uniform distribution over all legal actions) with probability ϵ. If ϵ can be tuned, would you assign it to be a high or low value at the beginning of training? What about at the end of the training? Please answer both questions and justify your choices.

Higher at beginning because want more exploration. We should use that exploration to converge to more optimal strategies at the end, but near the end of training, lower epsilon so we can exploit instead of explore.

(f) Instead of using the "ϵ-greedy" algorithm, we will now do some interesting exploration with softmax. We first introduce a new type of policy: A stochastic policy $\pi(a|s)$ represents the probability of action a being prescribed, conditioned on the current state. In other words, the policy is a now a distribution over possible actions, rather than a function that outputs a deterministic action.

Let's define a new policy as follows:

$$\pi(a|s) = \frac{e^{Q(s,a)}}{\sum_{a'} e^{Q(s,a')}}$$

(i) [1 pt] Suppose we are at square 3 in the grid and we want to use the originally provided Q values from the table. What is the probability that this policy will tell us to go right? Note that the sum over actions prescribed above refers to a sum over legal actions.

$\pi(3, right) = \frac{e^7}{e^8+e^5+e^7}$

(ii) [3 pts] What is the advantage of this exploration strategy, compared with "ϵ-greedy"? How are they qualitatively different?

This exploration is guided by Q value rather than purely random, so you can explore while still taking some amount of goodness (value) into account.

(g) [10 pts] Your friend Cody argues that we could still explicitly calculate Q updates (like Adam's approach in part (a)) even if we don't know the true underlying transition function $T(s, a, s')$, because he believes that our T can be roughly approximated from samples.

(i) [5 pts] Consider that you are given a moderate amount of (maybe 100,000) of transitions, in the form of (s_{start}, a, s_{end}). In a few sentences, describe **a sequence of steps** that you can take to compute $T_{approx}(s, a, s')$, which is an approximation of the true underlying (unknown) $T(s, a, s')$.

From lecture note 5 (https://inst.eecs.berkeley.edu/ cs188/sp20/assets/notes/n5.pdf):

In model-based learning an agent generates an approximation of the transition function, $\hat{T}(s, a, s)$, by keeping counts of the number of times it arrives in each state s after entering each q-state (s,a). The agent can then generate the the approximate transition function $\hat{T}$ upon request by normalizing the counts it has collected - dividing the count for each observed tuple (s, a, s) by the sum over the counts for all instances where the agent was in q-state (s, a). Normalization of counts scales them such that they sum to one, allowing them to be interpreted as probabilities.

Quick Note: If you didn't answer with the lecture note 5 approach, please discuss under the solutions thread on Piazza. The staff will monitor the debate over alternative answers.

(ii) [5 pts] Now, consider a case where Cody does the following: He (1) uses samples to compute $T_{approx}(s, a, s')$ in the way that you described in the previous part, and then (2) uses that $T_{approx}(s, a, s')$ to perform Q updates with Adam's equations. What could be one potential problem with this approach?
Hint: If your $T_{approx}(s, a, s')$ is not exactly correct (i.e., you didn't get infinite data samples, so it's probably not exactly correct), what could go wrong?

The main problems is that if the true world dynamics makes it extremely hard to reach a subset of "secluded" states, then we cannot guarantee to be able to collect any data about those secluded state-action pairs unless we have infinite data.

One example for secluded state is that, imagine a glitchy world where the all "down" actions are overwritten by "right" actions with probability 0.99999, and only actually move "down" with probability 0.00001 (these are parameters of the real world which you won't have direct access to). You can then imagine that the states at the bottom-left corners are "secluded" because it is very very hard to reach them in the first place.

If we don't have the data for those "secluded states", then we will encounter a 0/0 problem with $T_{approx}(s, a, s')$ coming out of those states. You may address the division by zero by assuming some type of transition, but those will be "hacks" and potentially very inaccurate..

The problem with the iterative Q estimate update is that even if a small subset of $T(s, a, s')$ is wrong, the Q estimate update for a lot of states will be inaccurate (partly demonstrated by the error propagation in part (b).

Quick Note: if you didn't answer with the lecture note 5 approach in part (i), please discuss under the solution thread on Piazza.

Q3. [30 pts] Probability: Flowers

Your friend Ethan wants to find the biggest flower in the field. In the field, there are n flowers where the value of n is unknown. The size of all the flowers can be ranked unambiguously if all are seen. Ethan will walk across this field, with each order of flowers being equally likely. Ethan decides not to walk one meter back because he is too tired. He also only gives him one chance to pick the flower. That means whenever he picks a flower, he is done with the whole sequence of flowers. He wants to have the highest probability of selecting the actual biggest flower.

(a) **(i)** [3 pts] Ethan thinks he can do the following: He will check out $r-1$ flowers without touching them so that he can have a good calibration. M is the biggest flower in the $r-1$ flowers. He then selects the first subsequent flowers that is better than M (and the last flower if nothing was picked till then). For an arbitrary cutoff r, can you show the **probability that the biggest flower is actually selected**.

The probability

$$P(r) = \sum_{i=1}^{n} P(\text{flower } i \text{ is selected and it is the biggest}) \tag{1}$$

$$= \sum_{i=1}^{n} P(\text{flower } i \text{ is selected}|i \text{ is the best}) P(\text{flower } i \text{ is the best}) \tag{2}$$

$$= (\sum_{i=1}^{r-1} P(\text{flower } i \text{ is selected}|i \text{ is the best}) + \sum_{i=r}^{n} P(\text{flower } i \text{ is selected}|i \text{ is the best})) P(\text{flower } i \text{ is the best}) \tag{3}$$

Following this, can you derive the probability formulation for the probability that the biggest flower is actually selected.

Hint: Ethan will stop at any flower F larger than the calibration one. What's the probability of this one (F) to be the actual biggest one? Maybe all the flowers before F is smaller than M.

- First notice $\sum_{i=1}^{r-1} P(\text{flower } i \text{ is selected}|i \text{ is the best}) = 0$ because Ethan will never select a flower from the first group of $r-1$ according to the problem setup.
- Then notice $P(\text{flower } i \text{ is the best}) = \frac{1}{n}$ because given no other information about the arrangement of the flowers, each of the n flowers is equally likely to be the best.
- So now we tackle the hardest part. $\sum_{i=r}^{n} P(\text{flower } i \text{ is selected}|i \text{ is the best})$. We'll attempt to solve $P(\text{flower } i \text{ is selected}|i \text{ is the best})$.
- Notice this is equal to $P(\text{2nd best flower of the first } i \text{ is within the first } r-1|i \text{ is the best})$
- The way we solve this is to realize that given that flower i is the best, there are $i-1$ possible flowers that can all be the 2nd best of the first i flowers.
- Thus each of the $i-1$ flowers has a $\frac{1}{i-1}$ chance of being the 2nd best flower of the first i flowers (because we are given no other information about the arrangement of the remaining $i-1$ flowers; we only know that the *ith* flower is the best). We are interested in the probability that the first $r-1$ flowers contain the 2nd best. Since each of the $r-1$ flowers has $\frac{1}{i-1}$ probability of being the 2nd best flower out of the first i flowers, we find that $P(\text{2nd best flower of the first } i \text{ is within the first } r-1|i \text{ is the best}) = (r-1)(\frac{1}{i-1})$.
- Substituting this into the sum, we get $\sum_{i=r}^{n} P(\text{flower } i \text{ is selected}|i \text{ is the best}) = \sum_{i=r}^{n} \frac{r-1}{i-1}$

$$P(r) = (0 + \sum_{i=r}^{n} \frac{r-1}{i-1})\frac{1}{n} \tag{4}$$

$$= \frac{r-1}{n} \sum_{i=r}^{n} \frac{1}{i-1} \tag{5}$$

(ii) [4 pts] Now compute the probability when n goes to ∞.

Hint: you might need to do a bit of calculus here. For example, when $n \to \infty$, $\frac{1}{n}$ can be dt in an integral. Here t is a new variable we introduce.

Solution 1:
Using the hint, we have $dt = \frac{1}{n}di$, so we can guess $t = \frac{i-1}{n}$. Let $x = \frac{(r-1)}{n}$. Then we have $x\int_x^1 \frac{1}{t}dt = -xlnx$.
Solution 2:
Using the hint from Mesut's piazza post @466, we have $\sum_{i=r}^{n} f(i) = \int_r^n f(i)di$ as $n \to \infty$
So let's plug and chug.

$$\frac{r-1}{n}\sum_{i=r}^{n}\frac{1}{i-1} = \frac{r-1}{n}\int_r^n \frac{1}{i-1}di \text{ as } n \to \infty \tag{6}$$

$$= \frac{r-1}{n}\int_{r-1}^{n-1}\frac{1}{u}du \text{ if we use } u = i-1 \tag{7}$$

$$= \frac{r-1}{n}[\ln u]_{r-1}^{n-1} \tag{8}$$

$$= \frac{r-1}{n}[\ln(n-1) - \ln(r-1)] \tag{9}$$

$$= \frac{r-1}{n}\ln\frac{n-1}{r-1} \tag{10}$$

$$= \frac{r-1}{n}\ln\frac{n}{r-1} \text{ since } n \to n-1 \text{ as } n \to \infty \tag{11}$$

Solution 3: You could have used the fact that $\sum_{i=1}^{\infty}\frac{1}{x} = \ln x$ and avoided integrals all together. Though this would require breaking the sum into two and is altogether not as intuitive as Solution 2: $\sum_{i=r}^{n}\frac{1}{i-1} = (\sum_{i=1}^{n}\frac{1}{i-1}) - (\sum_{i=1}^{r-1}\frac{1}{i-1})$

(iii) [3 pts] Take the derivative with respect to your. expression from (ii) and tell Ethan when should he stop calibration. (what is r?) Please show all steps of your work

Use product rule and then chain rule to take a derivative and set to 0. $\frac{d(\frac{r-1}{n}\ln\frac{n}{r-1})}{dr} = 0$
Solving for this, you get $r = \frac{n}{e}$
If you substitute this optimal $r = \frac{n}{e}$ into your equation for $P(r)$ from a(i), you will find that $P(r) = \frac{1}{e}$ though this was not required for a full-credit answer for this problem.

(b) **(i)** [1 pt] If it is a sunny day (+s), then Ethan will be happy (+h) with a high probability. If Ethan is happy (+h), he will execute his plan to pick a flower with a high probability (+p). Otherwise, he will execute his plan with a low probability (+p'). Sunny day (S), Ethan's happiness (H), and pick a flower (P) are three variables. Draw out the bayes net with those three variables.
$S \to H \to P$

(ii) [2 pts] If condition on Ethan's happiness, is the weather and flower picking independent? Explain your answer intuitively **without referring to the active/inactive triples**.
yes.

(iii) [7 pts] Now we find more concrete information about this activity. For weather, it can be sunny and rainy. For happiness, it can be happy and unhappy. For flower pick, it can be pick or not pick. $Pr(sunny) = 0.4$, $Pr(happy|sunny) = 0.8$, $Pr(happy|rainy) = 0.5$, $Pr(pick|happy) = 0.8$, $Pr(pick|unhappy) = 0.2$. Can you compute $Pr(sunny|pick)$? Please show all steps of your work.

$$P(sunny|pick) = \frac{P(sunny, pick)}{P(pick)} \tag{12}$$

$$= \frac{\sum_{Happiness} P(sunny, Happiness, pick)}{\sum_{Happiness, Sunniness} P(Sunniness, Happiness, pick)} \tag{13}$$

Note that $P(S, H, P) = P(S)P(H|S)P(P|H)$ from the joint probability distribution of the Bayes net. So

we can write the above into values we know.

$$= \frac{P(+s)P(+h|+s)P(+p|+h) + P(+s)P(-h|+s)P(+p|-h)}{NUMERATOR + P(-s)P(+h|-s)P(+p|+h) + P(-s)P(-h|-s)P(+p|-h)} \tag{14}$$

$$= \frac{0.4 \cdot 0.8 \cdot 0.8 + 0.4 \cdot (1-0.8) \cdot 0.2}{NUMERATOR + (1-0.4) \cdot 0.5 \cdot 0.8 + (1-0.4) \cdot (1-0.5) \cdot 0.2} \tag{15}$$

$$= \frac{0.272}{NUMERATOR + 0.3} \tag{16}$$

$$= \frac{0.272}{0.272 + 0.3} \tag{17}$$

$$= 0.476 \tag{18}$$

(c) Consider a modified setting where Ethan and his friends Fiona are **competing to pick a bigger flower**. Ethan and Fiona are walking in two separate fields with flowers of the same set of sizes $[s_1, s_2, s_3, s_4, ..., s_n]$, which in turn have ranks $[1, 2, 3, 4, ..., n]$ (where higher ranks means bigger sizes), but arranged in (possibly) different order. At each discrete time-step t, both Ethan and Fiona will see one more flower in their respective field.

Ethan and Fiona cannot observe the other person's field directly (and thus cannot see what the other person has seen so far).

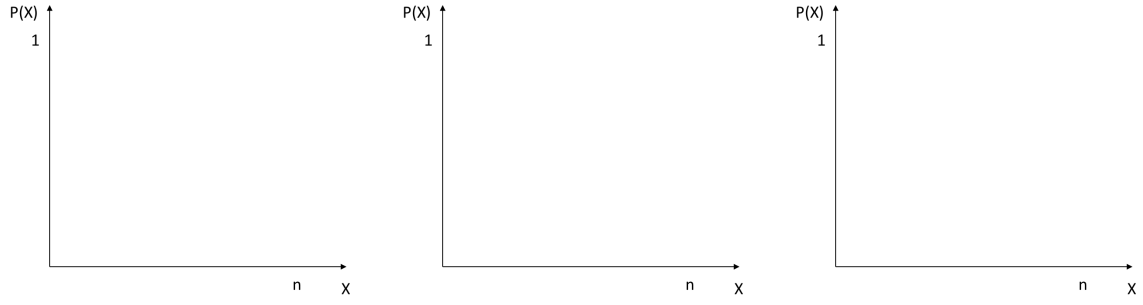
If Fiona knows that Ethan is committed to the strategy described in part (a.i),and the number of flowers Ethan is checking out is exactly r_e. What is Fiona's optimal r_f to get a bigger flower than Ethan?

(i) [1 pt] Consider the rank of the flower Ethan ended up picking as random variable X, and the rank of the flower Fiona ended up picking as random variable Y. What is the condition for Fiona strictly beating Ethan in the competition, in term of X and Y

$Y > X$

(ii) [3 pts] How does the probability distribution for $P(X = x)$ for $x = 1, 2, 3, ..., n$ depend on r_e? Please provide a sketch for $P(X)$ parametrized by $r_e = n - 1$, $r_e = 1$, $r_e = \lfloor \frac{n}{2} \rfloor$. Note that the function will be discrete.

You do not have to derive the expression for $P(X = x)$ in terms of r_e. Reasoning the shape should be adequate



$r_e = n - 1$ $\qquad$ $r_e = 1$ $\qquad$ $r_e = \lfloor \frac{n}{2} \rfloor$

Quick note for grading: We accept two interpretations of this problem. (1) You understood the strategy to check out r instead of $r - 1$ flowers (since this problem said Ethan checks out r_e and Fiona checks out r_f flowers. (2) You understood the strategy in the context of part a, in which case you did this problem assuming Ethan checks out $r_e - 1$ flowers and Fiona checks out $r_f - 1$ flowers.

Justification for the shape of $r_e = n - 1$:

(1) **Should be uniform $P(X) = \frac{1}{n}$ everywhere.** We are selecting the last flower for sure, and each flower rank has equal probability of being selected. Thus uniform everywhere.

(2) **Slightly increasing function, with smallest probability at $x = 1$**

Justification for the shape of $r_e = 1$:

(1) **Slightly increasing function, with smallest probability at $x = 1$.** In expectation, our calibration flower is of rank $\frac{n}{2}$ (the average sized flower). Thus in expectation, we pick a flower rank $\geq \frac{n}{2}$. Thus

higher ranks are more likely. Why is the probability of picking the smallest flower $x = 1$ not zero? It is possible, if all the flowers are arranged in largest to smallest size, that we end up forced to pick the last flower (which is the smallest).

(2) **Should be uniform $P(X) = \frac{1}{n}$ everywhere.**

Justification for the shape of $r_e = \lfloor \frac{n}{2} \rfloor$:

(1 and 2) **Small constant probability from $x = 1$ to $\frac{n}{2}$. Mildly increasing from $\frac{n}{2}$ to n.** In expectation, our calibration flower is roughly of rank $\frac{3n}{4}$ (75th percentile), because the worst calibration flower is average $x = \frac{n}{2}$ and the best calibration flower possible is the best flower on the field $x = n$. Thus in expectation, we pick a flower rank $\geq \frac{3n}{4}$. Thus higher ranks are more likely. Why is the probability of picking below average flowers $x \leq \frac{n}{2}$ not zero? Say all the above average flowers were the first $\frac{n}{2}$. Then our calibration flower is the best flower of rank $x = n$. Then in the latter half of the field when we start picking, we are stuck with all the below average flowers. Then, because we never see a flower that is better than the best flower, we will be forced to pick the last flower, which can be any rank from the smallest to the average-sized flower ($x = 1$ to $x = \frac{n}{2}$) with uniform probability.

(iii) [2 pts] Does r_f parametize $P(Y)$ the same way r_e paramatize $P(X)$?

● Yes ○ No

Explanation: Because Ethan and Fiona follows the same procedure.

(iv) [2 pts] What is the probability of $P(Y > X)$? Your answer should include two summations, and you are not required to simplify $P(X)$, $P(Y)$.

$P(Y > X) =$

$\sum_{x=1}^{x=n} \sum_{y=x+1}^{y=n} P(x, y) = \sum_{x=1}^{x=n} \sum_{y=x+1}^{y=n} P(x)P(y)$ because X and Y are independent

(v) [2 pts] Suppose you have the full expression for $P(X)$ and $P(Y)$ in terms of r_e, r_f respectively. How would you find the optimal r_f that would maximize the chance for Fiona to beat Ethan in this competition?

Explanation:

Write the quantity above as a function of r_f and r_e: $f(r_f, r_e) = \sum_{x=1}^{x=n} \sum_{y=x+1}^{y=n} P(x)P(y)$ (because all other variables in the expression should be fixed

But realize that r_e is fixed because that is given to us.

Then we can take the first derivative of the function with respect to r_f, set it to zero, and solve for r_f. We can verify it is a maximum by checking the second derivative is indeed positive.