CSE 473, Spring 2023 Assignment 4

Last name:________________ First name:____________________ UWNetID:__________

Due Friday night April 28 via Gradescope at 11:59 PM. You may turn in either of the following types of PDFs: (1) Scans of these pages that include your answers (handwriting is OK, if it's clear), or (2) Documents you create with the answers, saved as PDFs. When you upload to GradeScope, you'll be prompted to identify where in your document your answer to each question lies.

Answer the following seven questions. Each TA on the staff has contributed one question. These are intended to take 15-35 minutes each if you know how to do them. Each is worth 15 points. However, they vary somewhat in difficulty and each has several subquestions. If any corrections have to be made to this assignment, these will be posted in ED.

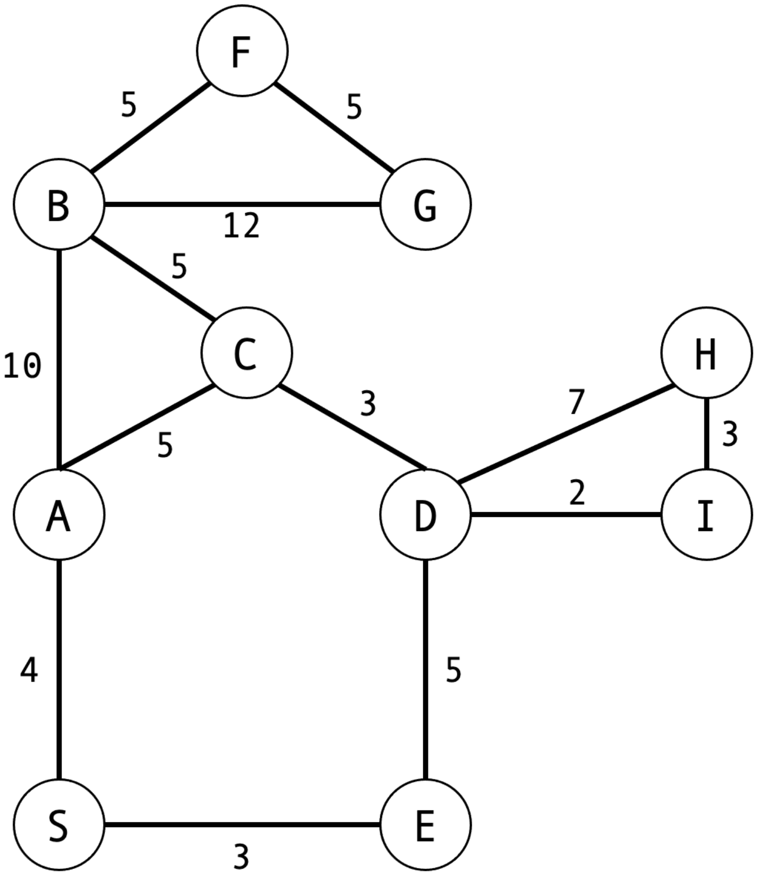
This is an individual-work assignment. Do not collaborate on this assignment.

Prepare your answers in a neat, easy-to-read PDF. Our grading rubric will be set up such that when a question is not easily readable or not correctly tagged or with pages repeated or out of order, then points will be be deducted. However, if all answers are clearly presented, in proper order, and tagged correctly when submitted to Gradescope, we will award a 5-point bonus.

If you choose to typeset your answers in Latex using the template file for this document, please put your answers in blue while leaving the original text black. Using Latex is not required for the bonus points, but it might help in achieving a neat, easy-to-read PDF.

1 A* Search vs Uniform Cost Search

Consider the following graph. Suppose all of the letters represent locations in a rugged, off-road terrain. The higher the cost, the more difficult it is to traverse the terrain between the corresponding locations. To clarify, these costs are independent of the distance between the nodes.



(a) What is the lowest-cost path returned by UCS (Uniform Cost Search) from **S** to **G**? Assume ties are broken alphabetically. Answer using a comma-separated list of letters. (3 pts)

(b) What is the cost of the lowest cost path returned by UCS? (1 pt)

For the remainder of this question, the *expansion time* of a location is the integer representing the order in which the location is removed from the OPEN list. For example, initially only **S** is on the OPEN list. Its expansion time is 1, because it is the first location removed from the OPEN list.

(c) What is the expansion time for location **G**? (Also, note that this time is determined when it is moved from the OPEN list to the CLOSED list). Your answer should be a number or NEVER if it is never expanded. (2 pts)

(d) What is the maximum size of the OPEN list? If G is expanded, you may assume the algorithm stops immediately after G is removed from OPEN.

(1 pt)

Suppose that we have a function *h*(l)* which returns the following heuristic values for each location:

l	*h*(l)*
S	26
A	20
B	9
C	15
D	20
E	22
F	5
H	24
I	18
G	0

(e) What is the order of location expansions by A*, using *h*(l)* as the heuristic, from **S** to **G**? Assume ties are broken alphabetically. Answer using a comma-separated list of letters. Please note the distinction with part a; this question asks for the order of expanded locations, not the path returned. (3 pts)

(f) When is the expansion time for **G** with A* and heuristic *h*(l)*? Your answer should be a number or NEVER if it is never expanded. (1 pt)

(g) What is the maximum size of the OPEN list with A* and *h*(l)*? If G is expanded, you may assume the algorithm stops after that point. (1 pt)

(h) Is *h*(l)* admissible? If it is, leave the table below blank. If not, enter the the maximum value for each location for which *h*(l)* is not admissible, such that the heuristic with the replaced value(s) would be admissible, into the table below (3 pts).

l	*h*(l)*
S	
A	
B	
C	
D	
E	
F	
H	
I	
G	

2 Design of Heuristics

On a checkered board your agent is at the start **S** tile, with obstacles (zigzag walls) in its way to the end **E** tile. With the ability to only navigate horizontally or vertically in either direction one step at a time (four possible directions similar to a rook's directions in chess) to the end tile while avoiding blockers along the way. (Note: The obstacles only mean your agent cannot move to a tile in the blocked direction. However it can do so from other open directions/angles. For example, moving right from S, where the coordinates are (0, 0, 0, 0), using μ_b isn't possible, therefore to get to C4 at (0, 1, 0, -1) the agent has to navigate around several tiles in any other permissible directions.)
Your move operators $\{\mu_a, \mu_b, \mu_c, \mu_d\}$:

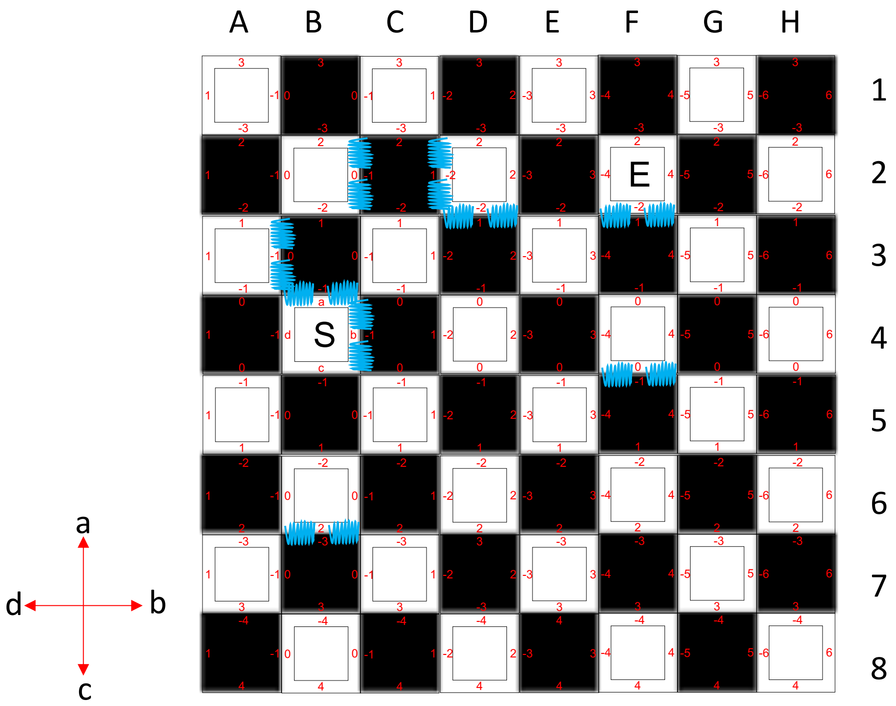
$$\mu_a : a \leftarrow a + 1 \text{ and } c \leftarrow c - 1. \quad \mu_c : a \leftarrow a - 1 \text{ and } c \leftarrow c + 1 \tag{1}$$

$$\mu_b : b \leftarrow b + 1 \text{ and } d \leftarrow d - 1. \quad \mu_d : b \leftarrow b - 1 \text{ and } d \leftarrow d + 1 \tag{2}$$

can be used to navigate the checkered world in search for the optimal route to the END tile. Note that there is (intentionally) some redundancy here. In particular, a + b + c + d = 0. The distance between any two adjacent tile can be calculated as follows:

$$d_{move} = x|a_1 - a_2| + y|c_1 - c_2| + r|b_1 - b_2| + z|d_1 - d_2|$$

with $x = 5, y = 1, r = 2$ and $z = 1$ The cost of navigation with any of the 4 four moves can be calculated, e.g., moving down from S, where the coordinates are (0, 0, 0, 0), using μ_c moves to position (-1, 0, 1, 0) on B5.



1. What's the cost of the shortest path between **S** and **E**? (2 pts)

2. What is the number of states that would be expanded by UCS to find a lowest-cost path? (3 pts)

3. Given the following provided heuristics, write out the heuristic value for each visited state along the lowest-cost path for each heuristic. (2 pts)

4. (4 points) Which heuristics among h_1, h_2, h_3 shown above is **admissible**? Identify any violations if any and what state they occur. (2 pts)

Heuristic	values
$h_1 = 2\|b - b_g\|$	
$h_2 = 4\|b - b_g\| + 2\|a - a_g\|$	
$h_3 = 2\|b - b_g\| + 2\|c - c_g\|$	

5. (4 points) Which heuristics among h_1, h_2, h_3 shown above are **consistent**? Identify any violations if any and what state they occur in. (2 pts)

6. Show the node expansion order using A^* search for any of the heuristics that is consistent and admissible. Mention which heuristic you are using. (2 pts)

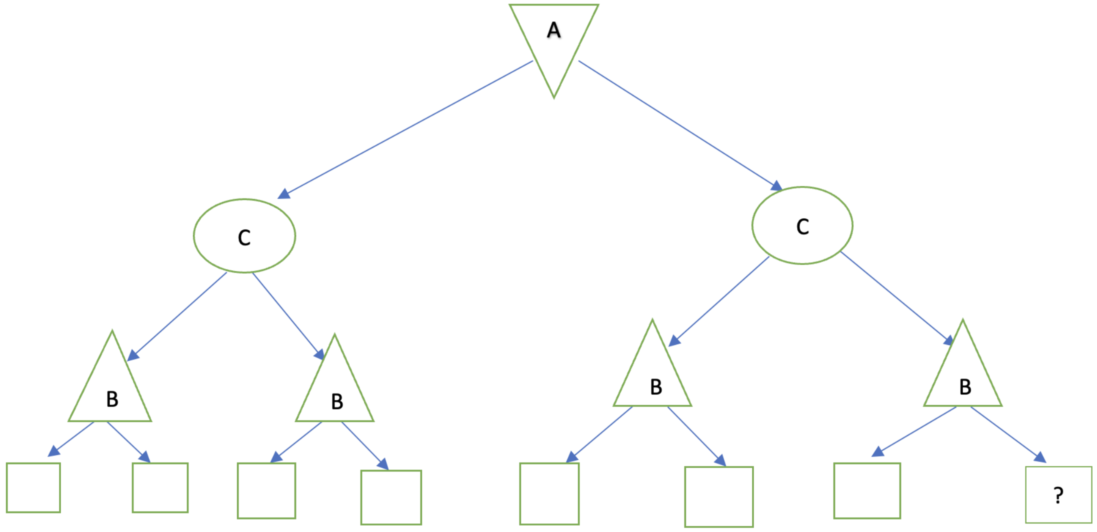
7. Assume now that the heuristic coefficients provided are computationally expensive to estimate. Propose another heuristic h_4 that is admissible, consistent and easier to compute. (Do not propose the exact distance defined above or function that includes it.) Give a formula to define your function. Also state why you believe it will be easier to compute h_4? (2 pts)

3 Game Tree Pruning

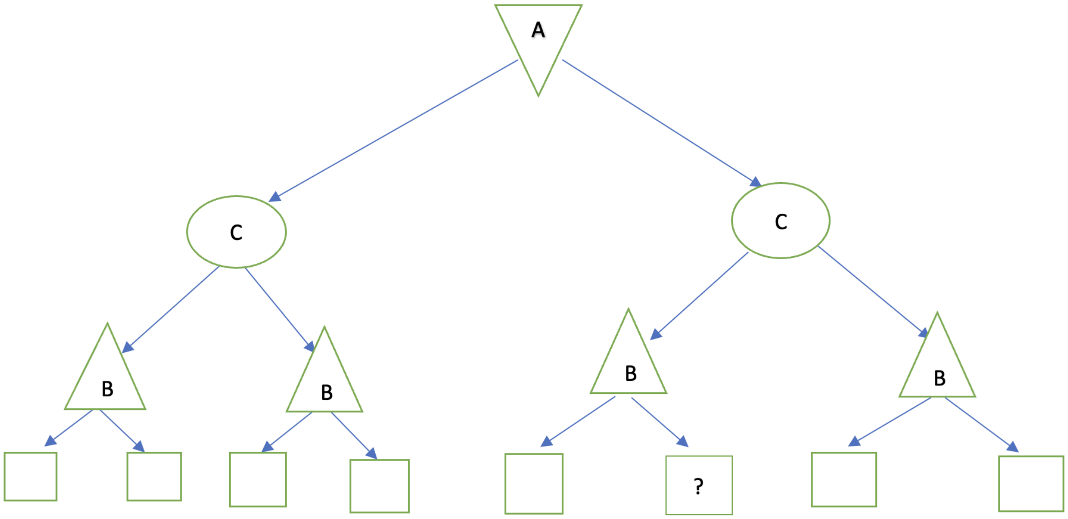
In this question, you will explore how to prune a game tree that contains not only max and min nodes, but also chance nodes. In this problem, a chance node is a node where all of its children are equally likely. A is a minimizer node, the B nodes are maximizer nodes, and the C nodes are chance nodes. In each of these questions, determine whether values exist for the terminal nodes such that the one marked "?" can be pruned (i.e., not examined), and if they exist, give values for them that work. If the values exist, write "can be pruned" and if not then write "cannot be pruned". The values you consider must be finite and you should assume that children of a node are visited from left to right. If there exist terminal values of the leaf nodes, write them in the squares, and give an explanation for the values you've chosen.

Note: Alpha-beta pruning is a technique that applies to minimax search but does not apply, in general, to trees that contain chance nodes. This problem asks you to reason about trees that do contain chance nodes. Here n can be pruned if its value cannot affect the the value at the top of the tree. Note that when terminal node values exist that make this possible, there might be many possible sequences of values that work.

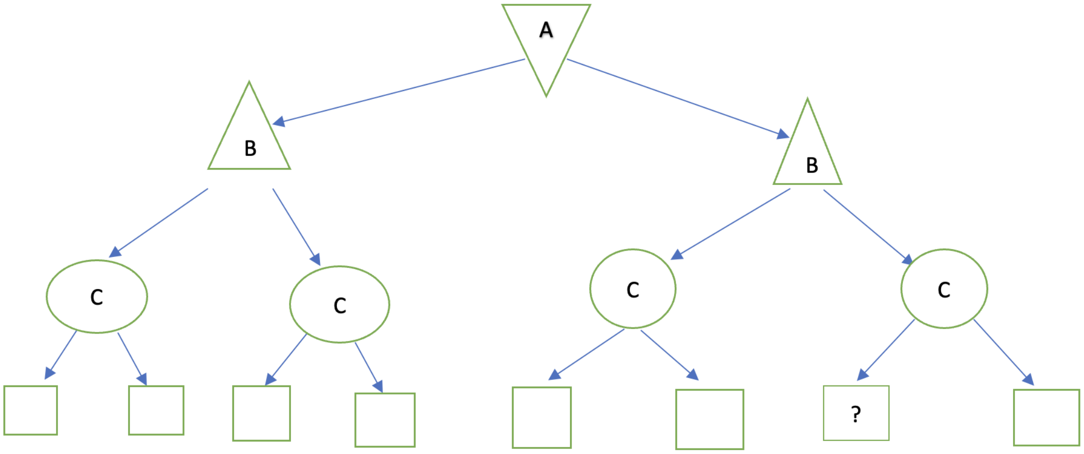
(a)



(b)

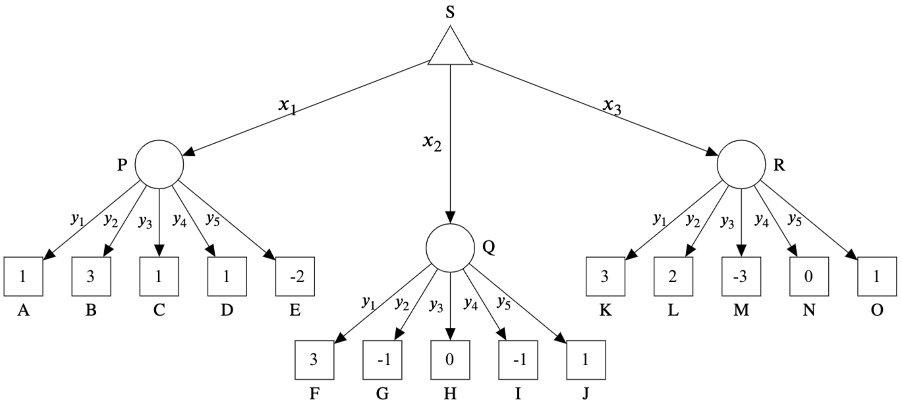


(c)



4 Expectimax

Alice and Bob are bidding in an auction at the UW for a bike. The auction has some unusual features. The bids are secret but each will choose from their own alternative options known to both of them. Alice will either bid x_1, x_2, or x_3 for the bike. She knows that Bob will bid y_1, y_2, y_3, y_4, or y_5, but she does not know which. Before bidding, Bob wants to examine the paint, cleanliness, lubrication, etc. of the bike, which is why he might bid differently, as given by y_i. Those attributes also affect the net value of the bike to Alice if she wins the auction, and so she takes these attributes into consideration in her assessment of the payoffs. Bob and Alice do not know what the other one bids. Alice wants to maximize her payoff given by the expectimax tree below. The leaf nodes show Alice's payoff possibilities as she has determined them (though her method might not seem consistent to us).. The nodes are labeled by letters, and the edges are labeled by the bid values x_i and y_i. The maximization node S represents Alice, and the branches below it represent each of her possible bids: x_1, x_2, x_3. The chance nodes P, Q, R represent Bob, and the branches below them represent each of his bids: y_1, y_2, y_3, y_4, y_5.



(a) Suppose that Alice believes that Bob would bid any of his bid options with equal probability. What are the values of the chance (circle) and maximization (triangle) nodes? (10 pts)

(i) Node P: □

(ii) Node Q: □

(iii) Node R: □

(iv) Node S: □

(b) Based on part (a), how much should Alice bid for the bike? (3 pts)
□ x_1 □ x_2 □ x_3

(c) What if node S, which represents Alice, is instead a minimizer node (what were payoffs are now expected net costs including repairs and bid payments)? (2 pts)
□ x_1 □ x_2 □ x_3

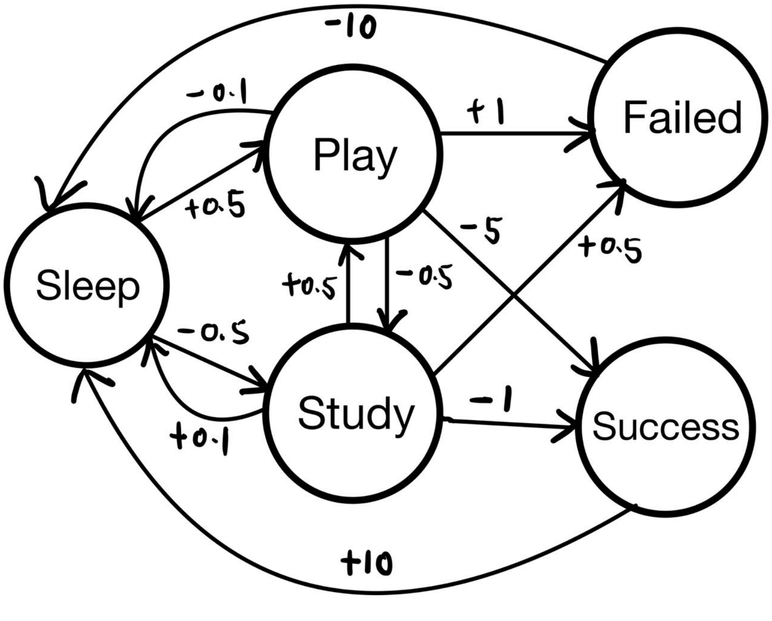
5 MDP Definitions

A student named Conan is preparing for CSE 473 midterm. He follows a crazy schedule in that he typically studies all day and maybe practices a problem set at the end of the day. However, he likes to play video games as well so he may also be attracted by them and play a long time.

His everyday activity could be considered as an MDP with rewards as below. There are five states, “Sleep”, “Play”, “Study”, “Failed” and “Success” in total. There are also five actions “Sl”, “Pl”, “St”, “Fa”, “Su”, where each of the actions means taking a step to the corresponding state (For example, action Pl could happen at state Sleep and Study, and it would cause the next state to be Play).

However, for three of the actions, there’s transition noise of 0.2 that would affect transitions. And it would only affect the actions Sl, Pl, St. For example, Conan is currently at Study, and he takes an action Pl, there would be a 0.8 transition probability for him to go to the state Play, and 0.2 probability to go to the state Sleep. However, if he takes the action Su to succeed in the problem set, the probability for him to be at Success for the next stage is 1.0.

Also, there are some rewards for each action Conan takes. Note that the other rewards are either effort-consuming (Study hard to succeed in the problem set for rewards of -0.5 or -1) or somewhat easier, playing games and being relaxed (for reward of 0.5). You may also notice that Failed or Success in the problem set affects Conan’s next day a lot with a reward $+10$ or -10.



(a) (5 points) Read the description above carefully and create a table representing the transition function $T(s, a, s')$ for this MDP. In this table, you do not have to provide an explicit entry for any $T(s, a, s')$ whose value is 0.
Hint: there should be 18 entries in your table.

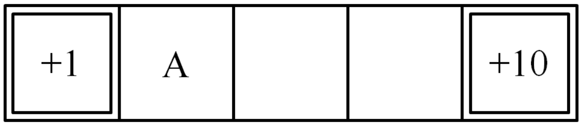
(b) (3 points) Create a table representing the reward function $R(s, a, s')$ for this MDP. There should be one explicit entry here for each explicit entry you had in part (a).

(c) (5 points) Suppose state values $V_0(s)$ are all 0, and we use a discounting $\gamma = 0.9$. Create a table representing the state values $V_2(s)$ after the second iteration of Value Iteration. Also, show the progress for calculating out the value $V_2(\text{Play})$.

(d) (2 points) What could you conclude about Conan's schedule if these values were updated many times? What will happen if $\gamma = 0.0$?

6 MDP: Linear Grid World

Consider the following linear grid world MDP, where A is the start state and the double-rectangle states are the exit states. From an exit state, the only action available is *Exit*, which results in the listed reward and ends the game. From non-exit states, the agent can choose either *Left* or *Right* actions, which move the agent in the corresponding direction. Assume that value iteration begins with initial values $V_0(s) = 0$ for all states s.



For the following questions, assume that: there is no living reward, the discount is $\gamma = 1$, and legal movement actions will always succeed (and so the state transition function is deterministic).

(a) What is the optimal value $V^*(A)$?

(b) When running value iteration, remember that we start with $V_0(s) = 0$ for all s. What is the first iteration k for which $V_k(A)$ will be non-zero?

(c) What will $V_k(A)$ be when it is first non-zero?

(d) After how many iterations k will we have $V_k(A) = V^*(A)$? If they will never become equal, write *never*.

Now, the situation is as before, but the discount value γ is less than 1.

(e) If $\gamma = 0.5$, what is the optimal value $V^*(A)$?

(f) For what range of values γ of the discount will it be optimal to go *Right* from A? Remember that $0 \leq \gamma \leq 1$. Write *all* or *none* if all or no legal values of γ have this property.

Now, the discount value γ is 1, but there is a living reward of 1.

(g) What is the optimal value $V^*(A)$?

(h) After how many iterations k will we have $V_k(A) = V^*(A)$? If they will never become equal, write *never*.

7 Value Iteration — MDP Values & Q-Values

Consider the grid world MDP shown below. It is similar to the one you saw in lecture, except that it is slightly larger and has two possible 'good' exit states. From all three exit states, the only available action is *Exit*, resulting in an earned reward and game termination. From non-exit states, the agent may move Straight (the intended direction), or as a result of 'noise' in the system, to the *Left* or to the *Right* (the latter two being from the agent's point of view based on the intended direction of movement). As in the example you saw in class, the probabilities of going right, left, or your intended directions are 0.1, 0.1, and 0.8, respectively. However, when the agent intentionally or unintentionally tries to go into a wall (i.e., to a non-existent state), it ends up staying put in that turn. In this exercise, use a discount factor γ of 0.9.

4			+1	▨
3				+1
2		▨		-1
1				
	1	2	3	4

(a) Bellman Equations (1 points): Write down the Bellman equations for V^* and Q^*.

(b) Explain what each equation above is expressing (include a discussion of the difference between states and chance nodes) (2 points).

(c) Value Iteration (10 points): Calculate V values for the following iterations (from $k = 0$ to $k = 7$). For ease of calculations, please round your results to the 2nd decimal place at each iteration. Be strategic; you likely won't need to calculate ALL the Q-values.

k= 0

4				▨
3				
2		▨		
1				
	1	2	3	4

k= 1

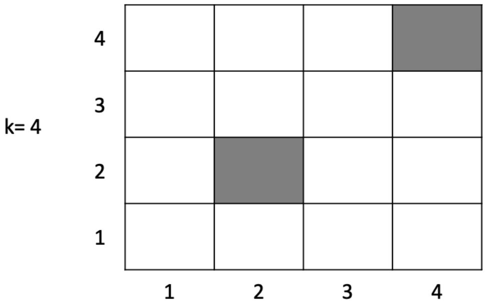
4				▨
3				
2		▨		
1				
	1	2	3	4

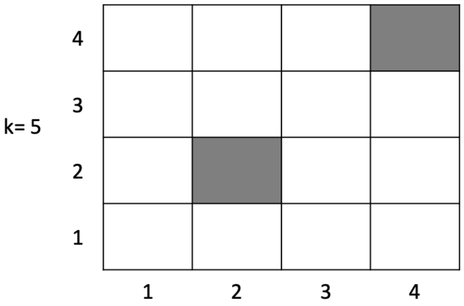
k= 2

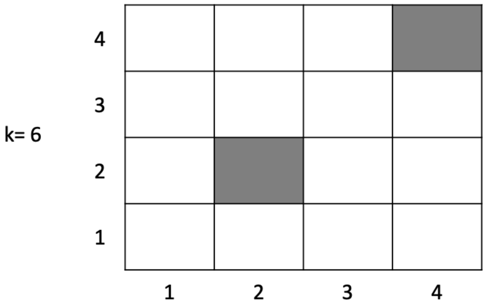
4				▨
3				
2		▨		
1				
	1	2	3	4

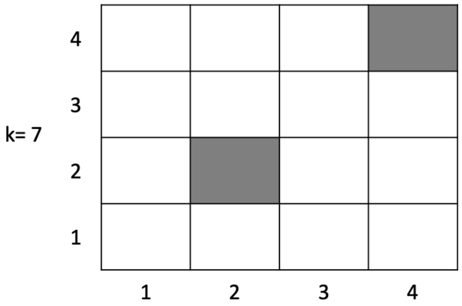
k= 3

4				▨
3				
2		▨		
1				
	1	2	3	4

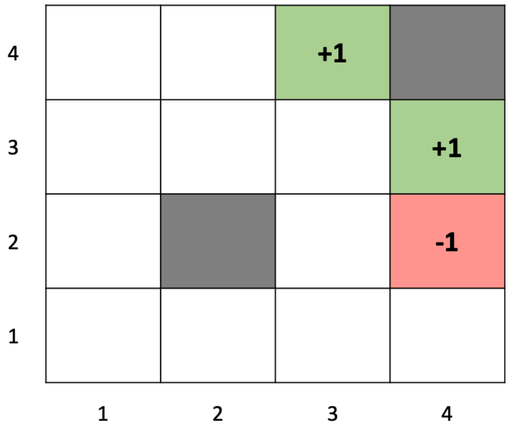








(d) Policy Determination (1 point): Fill in the policy after 7 iterations (k=7), using the diagram below. You may just use arrows to indicate the policy; there's no need to include any V or Q values in the diagram.



(e) Policy Changes (1 point): Without doing any additional calculations, if the discount factor γ were equal to 1, what change might you see in the policy for the state at (column 3, row 2)? This state is in the 3rd column and 2nd row. Describe any change and explain why it might occur.