# CSE 415 Winter 2023 Assignment 4

Last name:_____ First name:_____

Due Wednesday night February 8 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.

Do the following four exercises. These are intended to take 25-35 minutes each if you know how to do them. Each is worth 25 points. If any corrections have to be made to this assignment, these will be posted in ED.

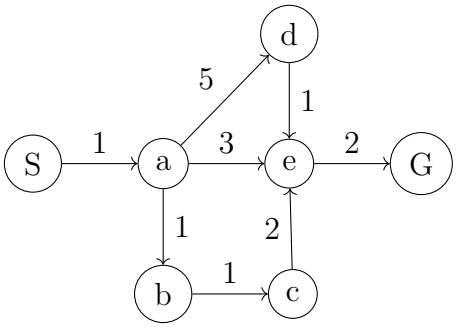
This is an individual-work assignment. Do not collaborate on this assignment.

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.

---

# 1 Heuristic Search (Mingyu)




| state (s) | S | a | b | c | d | e | G |
|---|---|---|---|---|---|---|---|
| heuristic $h_1(s)$ | 6 | 5 | 5 | 4 | 2 | 2 | 0 |
| heuristic $h_2(s)$ | 6 | 5 | 4 | 2 | 1 | 1 | 0 |
| heuristic $h_3(s)$ | 4 | 4 | 3 | 1 | 1 | 1 | 0 |
| heuristic $h_4(s)$ | 7 | 6 | 6 | 3 | 3 | 2 | 0 |

For the following questions, consider three heuristics $h_1$, $h_2$, $h_3$ and $h_4$. The table above indicates the estimated cost to goal, $G$, for each of the heuristics for each node in the search graph.

(a) (5 points) Which heuristics among $\{h_1, h_2, h_3, h_4\}$ shown above are admissible? Identify any violations.

(b) (5 points) Which heuristics among $\{h_1, h_2, h_3, h_4\}$ shown above are consistent? Identify any violations.

(c) (5 points) Show the node expansion order using $A*$ for any one of the heuristics that is consistent and admissible.

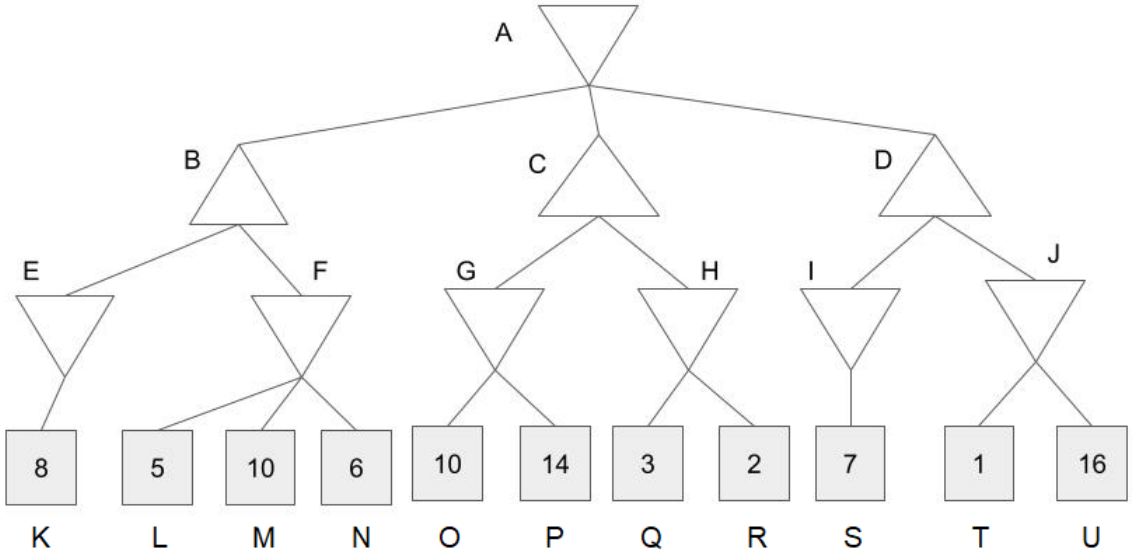
(d) (5 points) With an admissible heuristic, the $A^*$ algorithm continues its expansion process until the goal state is removed from the OPEN list and becomes the current state. Why not terminate as soon as a goal node has been generated (placed onto the OPEN list)? Please illustrate your answer with an example. How should the $A^*$ algorithm be programmed to terminate if the heuristic is not guaranteed to be admissible?

(e) (5 points) Identify two of the heuristics $h_a$ and $h_b$ from the four given in the table, such that you can make a change to $h_a$ at just one state (call it $x$) getting a new heuristic $h_{a'}$, resulting in both $h_{a'}$ and $h_b$ being admissible and consistent, and such that one of them dominates the other. What is state $x$ and what is your value for $h_{a'}(x)$ ? Which heuristic dominates the other? Is there anything special about the one that dominates here?

## 2 Adversarial Search (Kenan)

Minimax game-tree search finds the best move under the assumption that both players play rationally to respectively maximize and minimize the utility of the same evaluation function. (It can also do well when those assumptions are relaxed.) The Minimax search, however, typically requires a lot of computation time, especially with a deep and widely branching search tree. Alpha-beta pruning is a method for speeding up Minimax search by skipping any subtrees from the search tree that will not contribute to the outcome. Despite that, Alpha-beta pruning's success is dependent upon the order in which we visit the states. A better order to visit the nodes could result in large difference in performance.

For the following questions, note that the example we are using has a minimizing node at the root, which means our goal is to determine which choice of move at A offers the minimum value.




(a) (6 points) Apply straight **minimax search**, depth-first, left-to-right. Show the backed-up values at each internal node.

(b) (8 points) Apply alpha-beta pruning, left-to-right, on the above-given tree. Mark where cutoffs occur in the tree above.

For each leaf node, put an X in its box below if it has been cutoff (and thus does not need to be visited).

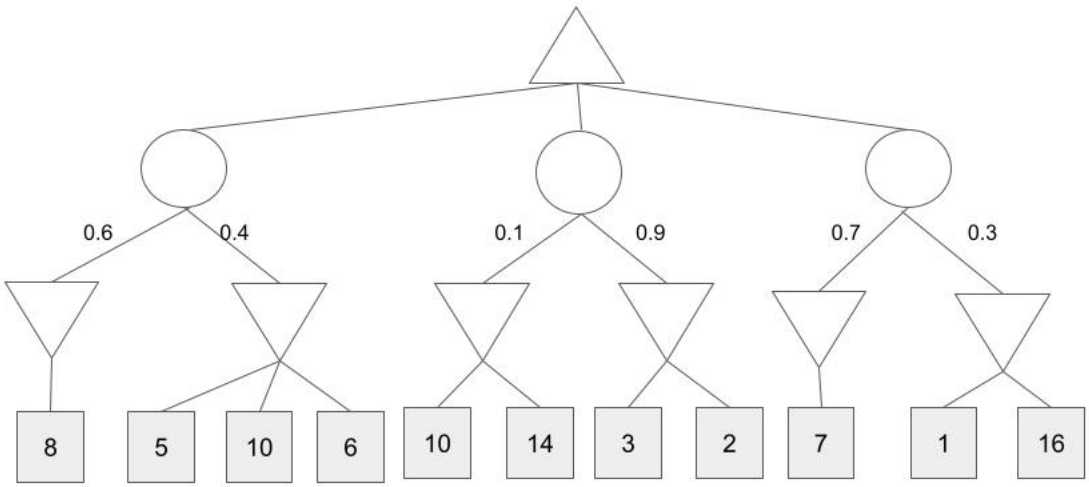
| K | L | M | N | O | P | Q | R | S | T | U |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | |

(c) (4 points) Re-apply alpha-beta search, but operating depth-first *right-to-left* in the tree. Don't mark the tree this time, but put an X in the box below for each leaf node that is cut off in this case.

| K | L | M | N | O | P | Q | R | S | T | U |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | |

(d) (2 points) Which of these two orderings provides for a more efficient search?

(e) (5 points) Consider the following expectimax game tree. Note that the circle nodes represent expectation nodes and the probability of their successors are denoted on the outgoing edges of these nodes. Fill in the nodes in the tree with the correct values selected by maximizing and minimizing players during the expectimax algorithm.



# 3 Markov Decision Processes, Utilities, Discounting, and Finite Horizons (Emilia)

When we considered deterministic systems, we discussed the concept of moves where if an applicable operator $\phi$ was applied to a state $s$, a certain new state $s'$ (or possibly the same state $s$) would be reached. In a Markov Decision Process (MDP), we are still interested in the transition from state $s$ to a successor state $s'$, but now we have to consider a degree of uncertainty about whether or not the desired transition will take place.

(a) (3 points) In the context of an MDP, describe the transition probability function $T(s, a, s')$ in your own words and specify what its range of possible values can be.

(b) (1 point) Imagine an agent is currently in state $s$. The function $T(s, a, s')$ provides a representation of what is likely to happen from $s$. with an associated transition probability function $T$. Is the value of $T$ affected by the state the agent might have been in *prior to* being in state $s$?

**Yes / No**

(c) (1 point) In an MDP, a transition from state $s$ to a second state $s'$ is associated with some reward $R(s, a, s')$. Mr. R says, "This reward must always be positive." Is that true or false?

**True / False**

(d) (4 points) When modeling something with an MDP, you typically want to maximize the total reward (often determined by taking the sum of the rewards obtained in each state of the process). This is referred to as the utility. However, unless you have a *finite horizon* that limits the number of transitions $n$ in the episode or some other strategy to prevent infinite state visits, this sum could be problematical to calculate. Often, in a sequence of states, rewards for states that follow the initial state are modified using a discount factor $\gamma$. What is the rationale for including this discount factor (give two or more reasons)?

Consider the following scenario that has been modeled as an MDP. You are an artist. You have (essentially) completed a commissioned piece of art for a client. At the current time, you are within the time frame for when you told the client their art should be ready. Seeing how pleased clients are when you given them their art makes you happy. However, you are a perfectionist. For each day that you delay giving the client the art, you could potentially improve it, thus increasing your satisfaction with the piece. Unfortunately, the clients get impatient the longer you delay handing over the art. The longer you take perfecting the work, the more clients complain about the wait, making you less happy. Let's say the rewards are measured in "happiness units" (hu).

The reward associated with the current state $s_{\text{now}}$ and all states that occurred prior to $s_{\text{now}}$ is 0 hu.

The reward associated with handing over the art ($s_{\text{final}}$) is 100 hu, but delivery time uses up one day. The reward for each state (extra days of work, not including the final delivery day) from $s_1$ to $s_{\text{final}} - 1$ is 30 hu.

Assuming a discount factor of 0.9, use the following equation when appropriate, as you determine responses to (e) through (i) below:

$$U(s_{\text{now}}, \ldots, s_{\text{final}}) = R(s_{\text{now}}) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots + \gamma^n R(s_n) + \gamma^{n+1} R(s_{\text{final}})$$

(e) (2 points) What is the total reward if you decide to deliver the art without any extra work days?

(f) (2 points) What if you take one extra work day before turning over the art?

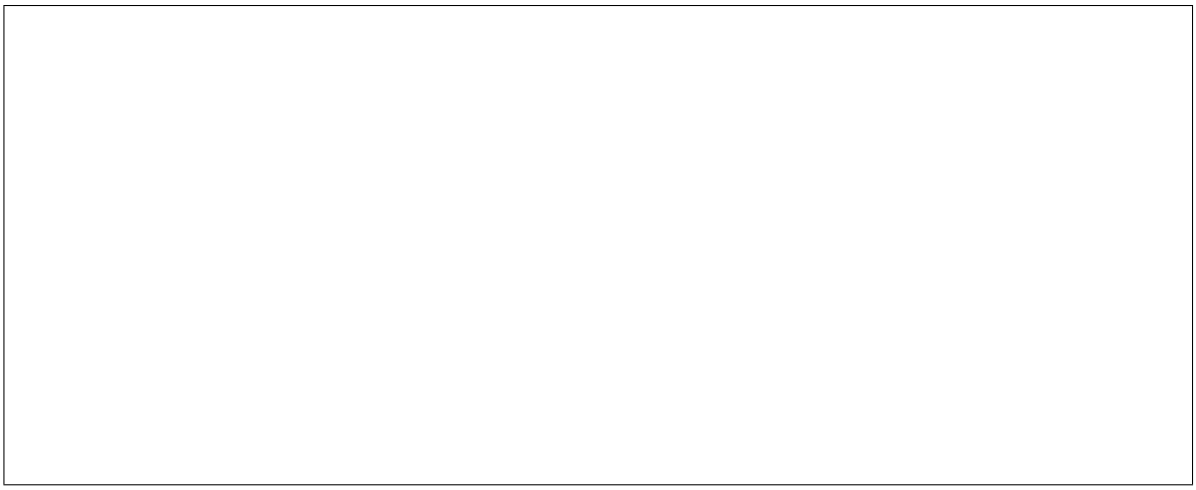
(g) (3 points) (When would you give your clients the art if you wanted to maximize the total reward?

(h) (1 point) Assuming a discount factor $\gamma$ very close to 0, when would you give the clients the art?

(i) (1 point) If the discount factor $\gamma$ were 1.0 (i.e. no discounting), when would your clients get their art?

Now, let's include some uncertainty. In addition to being a perfectionist, you are easily distracted when out walking and don't always complete errands (such as delivering completed art) successfully. Assume that on any given day, there is a 75% chance you will successfully deliver the completed art to your clients.

(j) (4 points) Draw a simple state transition graph (see Lecture 09-MDPs, slides 13-14) to represent the complete scenario.

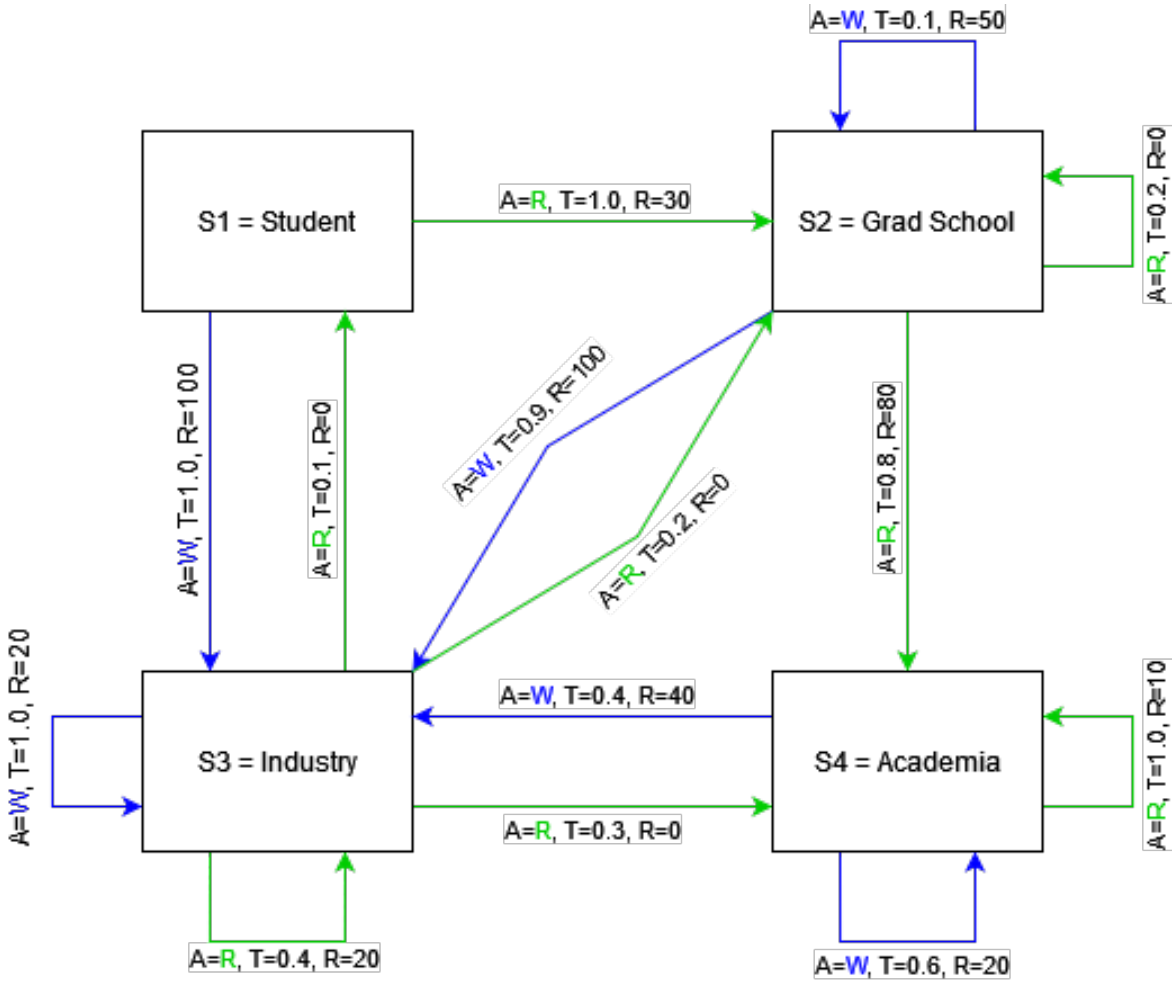


(k) (3 points) Use the following equation to determine the utility (sum of discounted rewards) for $s_{\text{now}}$ (Note that $\gamma$ has been pulled out in front of the second term – this is not another discount factor.):

$U(s) = R(s) + \gamma \Sigma s' T(s, a, s') U(s')$, where $s = s_{\text{now}}$ and $s'$ could be either $s_{\text{final}}$ or $s_1$.

# 4 Computing MDP State Values and Q-Values with Value Iteration (Phuong)

(25 points) An undergraduate student is trying to navigate the next steps in their journey beyond university. They have talked to other students, professors, and alumni in the industry to determine whether they should go into industry or pursue graduate school for research. They plan out the potential paths they can take as a undergrad student, which is represented as a Markov Decision Process below. There are two potential "actions" (outcomes) that can happen at each state: $W$ for work, and $R$ for research. Each action $A$, is associated with some transition probability $T$, and some reward value $R$, which is labeled for each arrow coming out of each state.



(a) Compute 3 iterations of Value iteration (including $V_0(s)$) to calculate the value of each state $s = \{S1, S2, S3, S4\}$, for all actions $a = \{W, R\}$, using the Bellman update formulas, for discount value $\gamma = 0.9$. (*Note*: Write out your calculations of the Bellman update formulas at every step for full credit)

$V_2(S1) =$ _____

$V_2(S2) =$ _____

$V_2(S3) =$ _____

$V_2(S4) =$ _____

(b) Compute the following Q-values for each state $s$ and action pair $a$, on the 4th iteration, $Q_3(s, a)$ (*Note*: Write out the expressions for each one, for full credit.)

$Q_3(S1, W) =$ _____

$Q_3(S2, W) =$ _____

$Q_3(S3, W) =$ _____

$Q_3(S4, W) =$ _____