CS 188
Fall 2019

Introduction to
Artificial Intelligence

# Written HW 1

**Due:** Friday 9/27/2019 at 11:59pm (submit via Gradescope).

**Policy:** Can be solved in groups (acknowledge collaborators) but must be written up individually

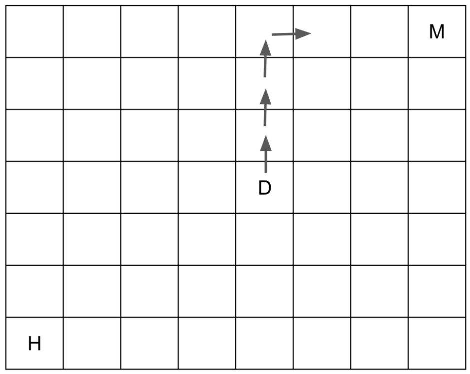
**Submission:** Your submission should be a PDF that matches this template. Each page of the PDF should align with the corresponding page of the template (page 1 has name/collaborators, question 1 begins on page 2, etc.). **Do not reorder, split, combine, or add extra pages.** The intention is that you print out the template, write on the page in pen/pencil, and then scan or take pictures of the pages to make your submission. You may also fill out this template digitally (e.g. using a tablet.)

| | |
|---|---|
| First name | |
| Last name | |
| SID | |
| Collaborators | |

# Q1. Uninformed Search

We are in a world where a disease is slowly taking over, and you have miraculously discovered a part of the cure. While fighting off the disease, you're trying to reach the other medical center with the final piece of the cure. You can imagine the world modeled as the following, where **H** is your Home base, **D** is the starting node of the disease, and **M** is the other Medical Center.

Disease **D** infects 1 extra square during each time step from its current location, exploring in the direction cycling between North, East, South, West, and only changes direction when it is not able to infect the next square. If the disease has no available adjacent square to infect, it does not move. Any square that the disease has been in at any time will be considered infected, and the disease will not move into an infected square.




The cost of exiting a non-infected square is 1, and the cost of exiting an infected square is 3.

For each of the following questions, give a quick justification of your answer.

**(a)** What would be the **minimum** state space representation and its size? Successor function? Goal test?

**State space representation:**

**State space size:**

**Successor function:**

**Goal test:**

**(b)** Assuming that ties are broken in the order East, North, West, South, what is the cost of this path using **BFS Tree Search** if you start at **H**, want to move to **M**?

**(c)** Lets say you have a power to stun the disease so it does not progress for an additional time unit. You can use the power when you share the square where the disease head is, or you are adjacent to that square, before taking a step as you usually would (meaning you can both move and stun within a time step). What is the new minimal state space? What is it's size? How will your successor function and goal test be affected, if at all?

**State space representation:**

**State space size:**

**Successor function:**

**Goal test:**

**(d)** Would we want to use game trees in this scenario? Why or why not?

# Q2. Trolley Transport Problems

**Five Animal Crossing** An eccentric billionaire has tasked you with transporting a collection of animals (and a rare flower) from the base of its mountain to its peak using a trolley. Thankfully, the trolley is currently at the base of the mountain, so your quest has some chance of succeeding.

Specifically, you will need to transport **(1) a rare orchid, (2) an aggressive plant-devouring grasshopper, (3) an insect-eating shrew, (4) a rodent-swallowing grass snake, and (5) a snake-killing mongoose.** If left unattended (at either the base of the mountain or its peak), the mongoose will kill the snake, the snake will swallow the shrew, the shrew will eat the grasshopper, and the grasshopper will devour the rare orchid. However, the animals (and orchid) will leave each other alone otherwise. Needless to say, the billionaire will be extremely angry if any of these organism die, an outcome you will try to avoid at any cost.

As you only have two hands, you will only be able to safely transport two animals at a time in the trolley.

Use your knowledge of CS 188 to safely move all the organisms to the peak using the trolley, and complete the billionaire's quest as fast as possible. Assume that using the trolley will incur a cost of 1.

**(a)** Assume that the trolley will only move with you on it.

**(i)** How many different states are there where the trolley is docked at either the base or peak of the mountain?

**(ii)** How many legal states are there where no animals (or orchids) will be harmed?

**(b)** Being a savvy CS 188 student, you decide to use $A^*$ search. Your first idea is to use the heuristic of "the number of animals at the base". Is the heuristic:

Admissible? Yes ◯ No ◯
Explain:

Consistent? Yes ◯ No ◯
Explain:

**(c)** A friend proposed another heuristic, which is simply 1 if the trolley is at the base and 0 if the trolley is at the peak. Is the heuristic:

Admissible? Yes ◯ No ◯
Explain:

Consistent? Yes ◯ No ◯
Explain:

**(d)** Consider the heuristic which is half the animals at the base, rounded up, plus 1 if the trolley is at the peak. Is the heuristic:

Admissible? Yes ◯ No ◯
Explain:

Consistent? Yes ◯ No ◯
Explain:

**(e)** Finally, consider the heuristic which is simply the number of animals at the base, minus 1 if the trolley is at the bottom. Is the heuristic:

Admissible? Yes ◯ No ◯
Explain:

Consistent? Yes ◯ No ◯
Explain:

**(f)** Of the four heuristics mentioned above, is there an admissible heuristic that dominates all other ones on all legal states? If not, why not?

# Q3. The Office Hour Assignment Problem

Eight Teaching Assistants (TA) are assigned to eight office hour slots. Each TA will take only one slot and each slot can only be taken by one TA. Before the assignment is conducted, the head TA has kindly asked every TA to indicate their preferences toward the slots (e.g. TA 1 cannot take Slot 8, or TA 2 prefers to take the slot earlier than TA3, etc.). The head TA would then assign the slots according to the preferences provided by all the TAs. This assignment problem can be formulated into a CSP.

**(a)** If all the TAs are available for all the slots and there are no preferences indicated by any TA, is this CSP equivalent to the eight-queen problem introduced in the class? To answer this question, the head TA asked you to help to formulate this into a CSP with regard to *Variables*, *Domain*, and *Constraints*. **Do not use the same formulation as part (b)**.

Variables: ______________________________. A total of _____ variables.

Domain: ______________________________. A total of _____ elements in the domain.

Constraints: ______________________________.

This problem is ______________ ("the same as" or "different from") the eight-queens problem

because: ______________________________

**(b)** The head TA instead proposed to formulate this CSP in the following way: if we represent the 8 TAs numbered from 1 to 8, and the 8 slots numbered from 1 to 8, we have

- Variables: $\{X_j\}_{j=1}^{8}$ where $X_j$ represents the slot ID of the j-th TA.
- Domain: $\{1, 2, 3, 4, 5, 6, 7, 8\}$.
- Constraints:
  1. $X_i \neq X_j \quad \forall i \neq j$
  2. $X_{3k} = 3X_k \quad \forall k \in \{1, 2\}$
  3. $X_{4k} = 4X_k \quad \forall k \in \{1, 2\}$
  4. $X_5 > X_6 > X_7$

The constraints above are all binary, and reflect the preferences of all the TAs towards all the slots.

Is the solution to the CSP unique? ______ (yes or no). Do we arrive at a valid solution by **only** enforcing arc consistency (are the domains for all variables size 1)? ______ (yes or no). Write down the remaining values for each variable after enforcing arc consistency.

$X_1$ : _________, $X_2$ : _________, $X_3$ : _________, $X_4$ : _________,

$X_5$ : _________, $X_6$ : _________, $X_7$ : _________, $X_8$ : _________.

**(c)** To solve the CSP in Question (b) via backtracking search, after the filtering step in Question (b), we wish to pick a variable to assign values. Which variable(s) would be the most reasonable choice based on **domain size**? ______. What is this selection process called? ________________________________________________.

**(d)** If we wish to use local search to solve the CSP in Question (b), with the initial value assignments to all the variables as $X_k = k, \forall k = 1, ..., 8$, which variables have conflicting constraints? ______________. What is the minimum number of iterations to refine the initial assignment into the correct solution? ________.

# Q4. Go Bears

The historical game of Go is played on a 19x19 grid by two adversaries, black and white, who alternately place pieces at the intersections of the grid lines. While rules and strategies for this game are abound, the only information required for this problem is that the game ends when no more valid moves exist for either player (e.g. when the board is full), and that the player with the most pieces on the board wins.

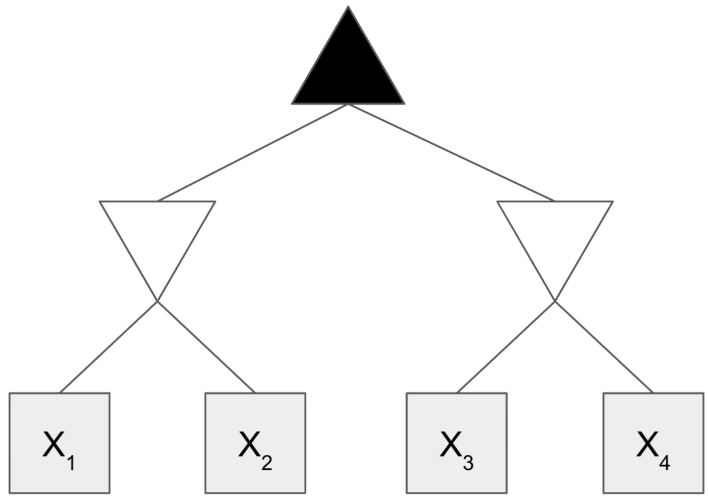
**(a)** We would like to tackle the problem of finding the optimal Go playing strategy using the minimax algorithm.

**(i)** What properties of this game make it amenable to analysis via minimax?

**(ii)** What properties of this game make it difficult to analyze via minimax?

**(b)** Because the search tree is so unwieldy, we will employ the strategy of cutting off our search at a certain depth and evaluating the utility of the leaf nodes. One way to evaluate the utilities is to start the game from that state and simulate many games, with both black and white randomly choosing moves. The utility of the leaf nodes for each player can then be approximated as the average number of games they won.

Let's say that we are playing as black, and we have employed this strategy that results in the game tree below (which is simplified for the purposes of this problem). Each $X_i$ is the number of games that black won in 10 simulated games, and we will estimate the utility of the leaf nodes as $X_i/n$.




Let $X_i \sim Binomial(n = 10, p = p_i)$ where $p_1 = 0.3, p_2 = 0.8, p_3 = 0.1, p_4 = 0.5$. All games are simulated independently of each other.

**(i)** In expectation, what move should the black player take, and what is their expected win rate?

**(ii)** What is the probability that the black and white players both choose their optimal moves in expectation (i.e. that black chooses their move from part (i) and white chooses their optimal move in expectation after that)? In the case of a tie, assume the left move is chosen. You may leave your answer in terms of the parameters specified (but you are encouraged to compute a numerical answer, for which you may need a computer).

**(c)** In the previous question, we considered the white player to be a deterministic minimizing agent. Now consider the white player to take actions with some randomness. Instead of minimax search, we should now employ expectimax search to discover the optimal strategy in expectation.

Our tree from part (b) was highly simplified with respect to the problem of Go. In reality, even if we did a depth limited search, we would still have many branches to search through given the huge branching factor of the Go tree. Therefore to save time we would like to prune branches that are unnecessary to explore.

**(i)** In general, we cannot expect to prune anything in expectimax tree search. Why not?

**(ii)** However, in our problem we can expect pruning to make our computation more efficient. Why?