CS 495 & 540 Fall 2018 Artificial Intelligence Final Exam

- I leave plenty of space for each problem. Please write your solution on the exam itself. Two blank sheets are attached at the back of the exam to serve as scratch paper for you. DO NOT detach the sheets. You only need to turn in one copy per group.
- Good luck and thank you for taking this class.

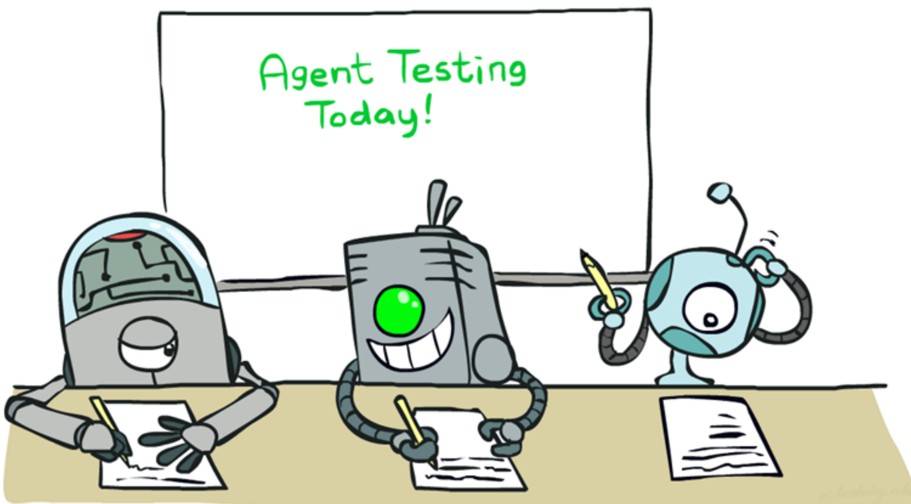
First name	
Last name	
Group ID	

For staff use only:

Q1.	Warm-Up	/5
Q2.	Reservoir Computing	/30
Q3.	BFS/DFS/Iterative Deepening Revisited	/20
Q4.	A Star Search	/15
Q5.	Baysian Network: Independence	/10
Q6.	Offline MDP Simulation	/10
Q7.	NLP: Extreme Case Smoothing	/10
Q8.	Heuristics: Application & Design	/20
Q9.	Hidden Markov Model: Computation	/20
Q10.	Inference: Enumeration and Variable Elimination	/15
Q11.	Limiting Distribution	/10
Q12.	Scratch paper: Do Not Detach	/0
Q13.	Scratch paper: Do Not Detach	/0
	Total	/165

Q1. [5 pts] Warm-Up

Circle the AI mascot that would respesent your behavior in this final exam if it was in class test and it was not being proctored ...



Q2. [30 pts] Reservoir Computing

Reservoir computing is another machine learning approach. It is similar to parallel computing that somehow different. Please read the following article Reservoir Computing and Extreme Learning Machines using Pairs of Cellular Automata Rules at https://ieeexplore.ieee.org/document/7966151and write a 2-page summary (similar to our PS1) in latex. Please attach this summary to the end of the final exam. Be reminded, any summary that are more than 60 percent similar, it would be considered plagiarism.

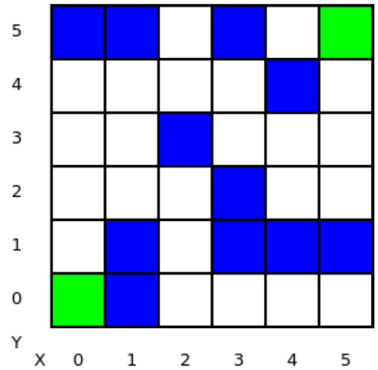
Q3. [20 pts] BFS/DFS/Iterative Deepening Revisited

Given a balanced tree with branching factor = b and height = h. Let suppose the goal is hidden at level k (**now let us assume** ***root*** **is at level** h **and leaves are at level 0**). It turns out this question is not that easy as you could imagine because there is another factor t (how far the target node is away from the **leftmost** node in level k, assuming we label the nodes from 1, 2, ..., t, ... , b^k, for level k). Let us assume that those search algorithms start from **right most, instead of leftmost**. So, in order to compare Iterative deepening with DFS, you must count the complexity. With the given condition above, please compute (**Please show step by step calculation since this problem has appeared couple times ... I will check the computation seriously**):
(a) Complexity of Iterative Deepening [Hint: you might want to use ceiling function for this and you have to be careful with how many time level 0 till $k-1$ are counted from iteration 1 till $k-1$. And in the very last iteration (the k level), not all nodes in level 1 to $k-1$ are visited]

(b) Compute the complexity of DFS

Q4. [15 pts] A Star Search

In this problem we are given the following configuration that was seen in your homework. But here is some twist:
(I)[3]How many paths (a cell can be at most visited once) are there to go from (0,0) to (5,5), given the blue cells are the barriers?



(II)[3] So, we want to find the solution using A* search and we have decided to use a hybrid heustric $h = \alpha * h_1 + \beta * h_2$ where $\alpha > 0, \beta > 0$, $\alpha + \beta = 1, h_1$ is the Manhattan Distance and h_2 is the Euclidean distance. Is this a valid heuristic? Why?

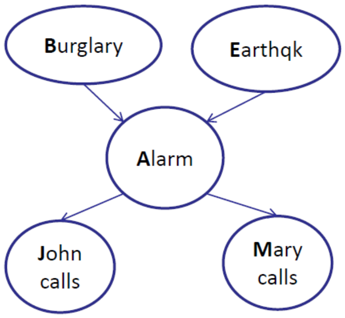
(II)[9] Now let say you decided to use Eclidean Distance only. At $(x, y) = (0, 2)$, your open list should be $[(1, 2), (0, 3), (0, 2)]$ and your closed list is $[(0, 0), (0, 1)]$. Please simluate the next three moves and the corresponding closed lists and open lists. (e.g. $\{(0, 1), [(0, 1)], [(0, 0), (0, 1)]\} \rightarrow \{(0, 2), [(1, 2), (0, 3)], [(0, 0), (0, 1), (0, 2)]\}$

Q5. [10 pts] Baysian Network: Independence

(a) [5pt] If we observe Alarm = true , are Burglary and Earthquake independent? Justify your answer by calculating whether the probabilities involved satisfy the definition of conditional independence

B	P(B)
+b	0.001
-b	0.999

E	P(E)
+e	0.002
-e	0.998



A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

(A, B, E)	P(A|B,E)
(+a, +b, +e)	(1/4)P(-a|+b,+e)
(+a, +b, -e)	4*P(-a|+b,-e)
(+a, -b, +e)	2*P(-a|-b,+e)
(+a, -b, -e)	7*P(-a|-b,-e)

(b) [5pt] Assume B, E are independent. Compute P(+a,-b,+e,-j, -m), P(+a,-b,-e,+j, +m) and P(-a,-b,+e,+j,+m).

Q6. [10 pts] Offline MDP Simulation

Please simulate the MDP using the value iteration for the scenario below and here we will have depreciation rate γ set to **0.6**. Let k be the number of iterations. We know offline MDP has the following formula:

$$V_{k+1}(s) \leftarrow max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

Please simulate and compute that

(a) [5 pts] When $k = 1$

$V_1(Cold) =$

$V_1(Warm) =$

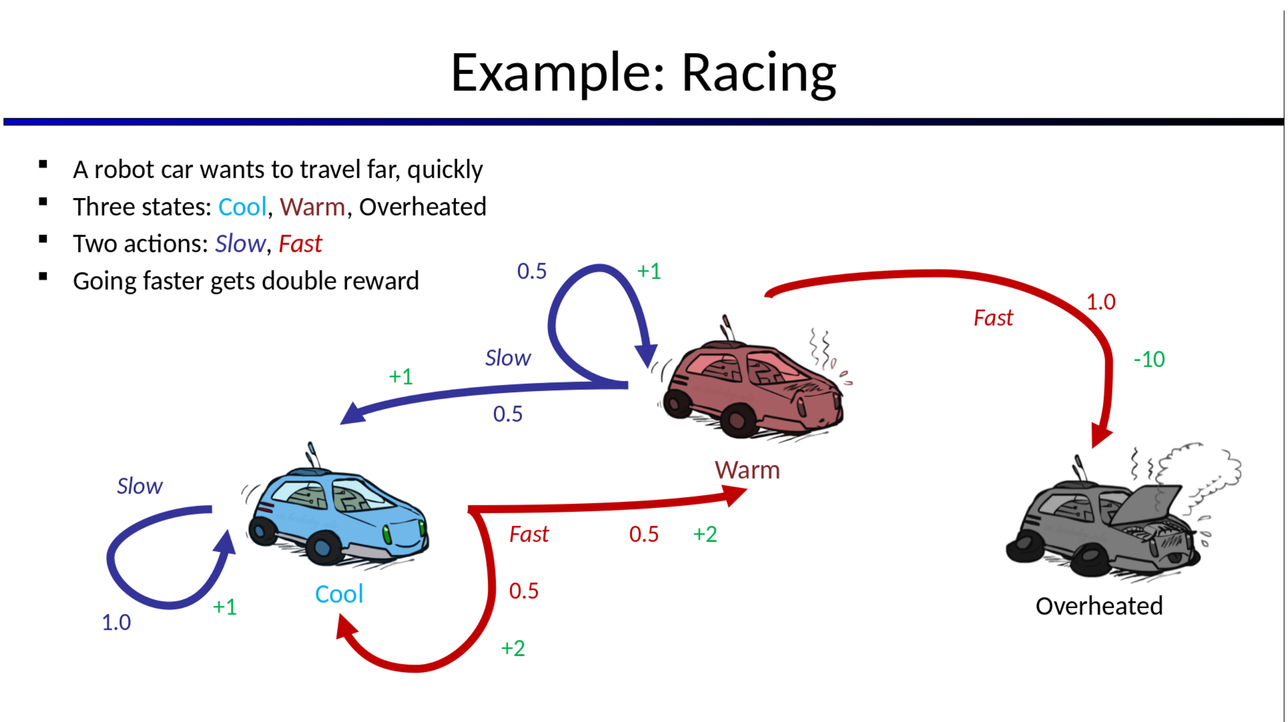
$V_1(Overheated) =$

(b) [5 pts] When $k = 2$

$V_2(Cold) =$

$V_2(Warm) =$

$V_2(Overheated) =$



Q7. [10 pts] NLP: Extreme Case Smoothing

We talked about NLP modeling in the class (see slides regarding langauge modeling at http://web.stanford.edu/̃jurafsky/NLPCourseraSlides.html) One of the common problems in NLP is how well the machine handle unknown words (words not seen in the training data set). We learned about **Laplace Smoothing** which is adding 1 occurence to each word in the vocabulary the recalculate the probability for the predictor. However, this is rather primitive. Let us explore the **Good Turing Smoothing**. An example is as following: you were fishing and you caught 7 carp, 3 perch, 2 whitefish, 1 trout, 1 salmon, 1 eel (total : 15 fish). So, how likely the next fish (next specises) is new? Good Turing will assume it is 1/5 (sum of those you only see once). It is obvious now the probability of seeing a trout in the next catch is definitely less than 1/15. Please describe a method to scale down the probability of seeing a trout from 1/15 to a lower number such that how many trouts you see in previous catches will have an effect in the new probability (prefer Good Turing Smoothing). Please also justify your answer.

Q8. [20 pts] Heuristics: Application & Design

n vehicles occupy squares $(1,1)$ through $(n,1)$ of an $n \times n$ grid. The vehicles must be moved in reverse order, so the vehicle i that starts in $(i,1)$ must end up in $(n-i+1,n)$. On each time step, every one of the n vehicles can move one square up, down, left or right or just stay put; but if a vehicle stays put, one other adjacent vehicle (but not more than one) can hop over it. Two vehicles cannot occupy the same square.

(a) Calculate the size of the state space as a function of n.

(b) Calculate the branching factor as a function of n

(c) Suppose that vehicle i is at (x_i, y_i); write a nontrivial admissible heuristic h_i for the number of moves it will require to get to this goal location $(n-i+1,n)$, assuming no other vehicles are on the grid.

(d) Which of the following heuristics are admissible for the problem of moving all n vehicles to their destinations? **Explainatioin is required. Guessing correctly won't gain any partial credits.**

(i) $\sum_{i=1}^{n} h_i$

(ii) $\max\{h_1, \cdots, h_n\}$

(iii) $\min\{h_1, \cdots, h_n\}$

Q9. [20 pts] Hidden Markov Model: Computation

Hidden Markov Model (HMM) basically is a belief propatation process. It starts with an initially uniform distribution for the unseen random variable X by checking the highly correlated observable evidence variable e. We know that (I) passage of time phase $B'(X_{t+1}) = P(X_{t+1}|e_{1\cdots t})$ and (II) observation phase $B(X_{t+1}) = P(X_{t+1}|e_{1\cdots t+1})$. Follow the weather HMM with the following modification: $P(+r \to +r) = 0.7, P(+r \to -r) = 0.3, P(-r \to +r) = 0.5, P(-r \to -r) = 0.5, P(+r \to +u) = 0.7, P(+r \to -u) = 0.3, P(-r \to +u) = 0.4, P(-r \to -u) = 0.6$. And let assume you live in Utica, therefore your initial belief is $B_0(+r) = 0.35, B_0(-r) = 0.65$. Please compute $B'_i(+r), B'_i(-r), B_i(+r), B_i(-r)$ where $i = 1, 2$.

1.$B'_1(+r) =$

2.$B_1(+r) =$

3.$B'_2(+r) =$

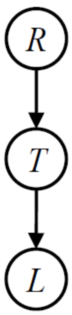
4.$B_2(+r) =$

Q10. [15 pts] Inference: Enumeration and Variable Elimination

The main advantage of variable elimination over enumeration is the complexity reduction. The general idea is that you marginalinze the joint probability space early such that it costs less in next iteration of join operation. Given the following, please compute $P(L)$ using (I)[5] Enumerationg approach (II)[5] Variable Elimination. Finally, (III)[5]please justify if II is better than I by comparing the cost of generating $P(L)$ via those two different approaches .

$P(R)$

+r	0.1
-r	0.9



$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Q11. [10 pts] Limiting Distribution

We talk about stochastic matrices and some of the convergence property.
(a) Please briefly describe the spectrum theorem of a stochastic matrix

(b) Given a matrix G that is an $N \times N$ **column-wise stochastic** matrix and a uniform distribution column vector v, someone claims that v is the limiting (stationary) distribution of G, i.e. $Gv = v$. What extra property can you impose on G such that this is true? Show a concrete example of G (and assume $N = 4$ in the example).

Q12. [0 pts] Scratch paper: Do Not Detach

(a) [0 pts]

Q13. [0 pts] Scratch paper: Do Not Detach

(a) [0 pts]