Midterm 1 Exam

CSCI 561 Spring 2018: Artificial Intelligence

First Name ______________________________

Last Name ______________________________

Student ID | | | | | | | | | | | |

USC Email | | | | | | | | | | | | @usc.edu

Please print neatly.

Exam Version 3.0 with Solution

Instructions:

1. Date: **2/13/2018 from 5:00 pm 6:20 pm**
2. Maximum credits/points/percentage for this midterm: 100
3. The percentages for each question are indicated in square brackets [ ] near the question.
4. **No books** (or any other material) are allowed.
5. **Write down your name, student ID and USC email address.**
6. **Your exam will be scanned and uploaded online.**
7. **Write within the boxes provided for your answers.**
8. **Do NOT write on the 2D barcode.**
9. **The back of the pages will NOT be graded. You may use it for scratch paper.**
10. No questions during the exam. **If something is unclear to you, write that in your exam.**
11. **Be brief: a few words are often enough if they are precise and use the correct vocabulary studied in class.**
12. When finished, raise completed exam sheets until approached by proctor.
13. **Adhere to the Academic Integrity code.**

Problems	**100 Percent total**
General AI Knowledge	xx
Multiple Choice	xx
Problem Formulation	xx
Intelligent Agents	xx
Uninformed and Informed Search	xx
Min-Max	xx
CSP	xx
Problem Solving	xx

1. [14%] General AI Knowledge

True/False

For each of the statements below, answer **T** if the statement is **always true**, or **F** otherwise.

F	1. An agent is anything that perceives its environment through sensors.
F	2. In CSPs, backtracking is always based on recursion.
F	3. Simulated Annealing (SA) is a probabilistic algorithm that guarantees convergence on a global minimum (or maximum) solution.
F	4. If local beam search with k=1 finds a solution, then local beam search with k=2 and starting from the same state will also find a solution.
T	5. A genetic algorithm with a population of 1 is equivalent to a random walk.
T	6. For a Breadth-First Search algorithm, memory usage poses a bigger concern than execution time.
F	7. Some consistent heuristics are not admissible.
T	8. Alpha-beta pruning never alters the final minimax value of the root of a game search tree.
T	9. A* graph search is optimal when the heuristic is consistent.
F	10. Uniform Cost Search is a special case of Breadth-First Search.

2. Multiple Choice

Each question has one or more correct choices. Circle all correct choices. Please note that there will be no partial credit and you will receive full credit if and only if you choose all of the correct choices and none of the wrong choices.

1. What is the time and space complexity, respectively, of DFS and BFS (goal-check a node when it is generated), where b is the branching factor, d is the depth of the shallowest solution, and m is the maximum depth of the search tree.
   A. $O(b*m)$, $O(b^m)$ for DFS and $O(b^{d+1})$, $O(b^d)$ for BFS
   B. $O(b^m)$, $O(b*m)$ for DFS and $O(b^{d+1})$, $O(b^d)$ for BFS
   C. $O(b^{m+1})$, $O(b*m)$ for DFS and $O(b^d)$, $O(b^d)$ for BFS
   D. $O(b^m)$, $O(b*m)$ for DFS and $O(b^d)$, $O(b^d)$ for BFS

2. Which algorithm is equivalent to simulated annealing with a temperature of 0 at all times (and omitting the termination test).
   A. Hill climbing
   B. Random walk
   C. BFS
   D. A*

3. Which statement is False?
   A. BFS is optimal if the path cost is a decreasing function of the depth of the node.
   B. BFS is optimal if the path cost is a nondecreasing function of the depth of the node.
   C. DFS is optimal if the state space is infinite.
   D. DFS is optimal if the state space is finite.
   E. **Both BFS and DFS are complete and optimal.**

3. [xx%] Problem Formulation

A millennial (M) is in need of an Uber ride from her apartment (A) to a selfie-themed party (P). The millennial has three things, a cat (C), a hamster (H), and a bag of organic kale (K), to bring to the party. The Uber driver, though willing to make multiple trips, is strict about what can be in the car at any one time. In fact, the millennial must be in the car on every trip, and she can take only one pet/item per ride. The millennial is afraid that leaving the hamster alone with the kale (either at her apartment or at the party) is not ideal because the hamster would surely eat the kale. Leaving the cat alone with the hamster, again, would be disastrous, because the cat would eat the hamster. A safe bet is to leave the carnivorous cat with the kale. How does the millennial travel without losing any of them?

1. [xx%] Write down the definition of a state in this problem. Give the initial and goal states. Hint: you are suggested to use the variables: M, C, H, and K.

The initial state {M,C,H,K} for Millennial, Cat, Hamster, and Kale at the millenial's home. The goal state is {}.

2. [10%] Write down all the legal states (in which any of the three things is not lost).

{M,C,H,K}
{C,K}
{M,C,K}
{C}
{K}
{M,C,H}
{M,H,K}
{H}
{M,H}
{}

Each {*} is worth 1 pt. Deduct 1 pt for incorrect {*} until zero.

An alternative representation is where the variables M, C, H, K can have domain {A,P}. So if the students answers:

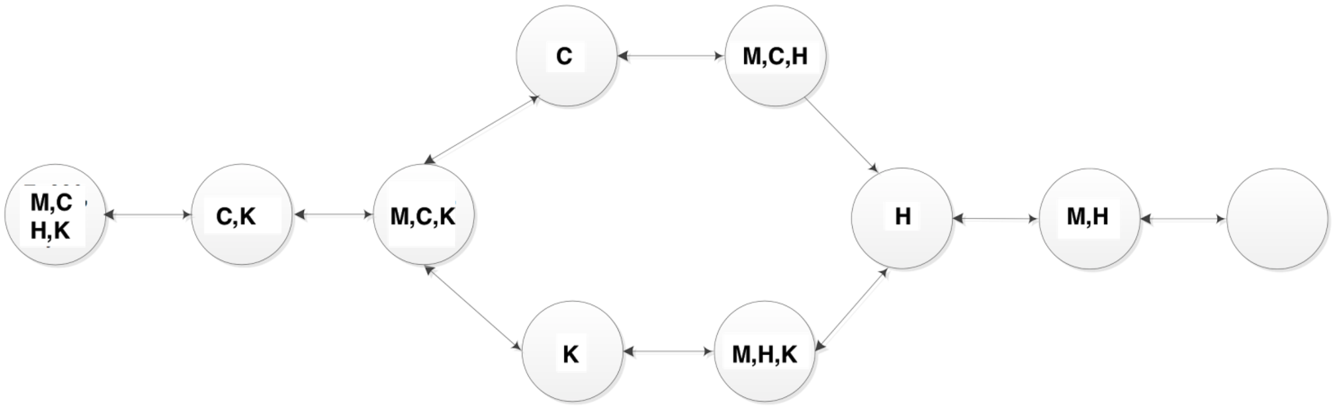
Initial State: M=A, C=A, H=A, K=A
Final State: M=P, C=P, H=P, K=P

Again, each pair is worth 1 pt. Deduct 1pt for incorrect until zero.

3. [xx%] Define the actions in this problem.

Move one thing (or nothing) to the other side. Nothing is statement is correct but not required in the answer.

4. [xx%] The problem can be represented as a graph which contains the states as nodes, and the state transitions as edges. Draw the graph for this problem (you can omit the actions on the edges between nodes).



5. [xx]% Provide one sequence of actions that solves this problem with the fewest trips by M.

M, H ->

<- M

M,K ->

<- M, H

M,C ->

<- M

M,S ->

4. [xx%] Intelligent Agents

1. [xx%] Consider a robot responsible for transporting packages in an Amazon warehouse. Is this robot operating in a deterministic or stochastic environment. Why or why not?

Robot picking parts is a stochastic environment. Why part should include a few of the following keywords:

In a deterministic environment any action has a single guaranteed effect, and no failure or uncertainty. In a non-deterministic environment, the same task performed twice may produce different results or may even fail completely. As much as we would like to think that Amazon warehouse is perfect, errors can occur.

2. [xx%] Consider a spam filtering agent. Is its environment episodic? Why or why not?

Spam filtering mail is episodic if the student does not mention that the agent learns from past filtering, etc. If the student claims that spam filtering is not episodic then the answer must include learning.

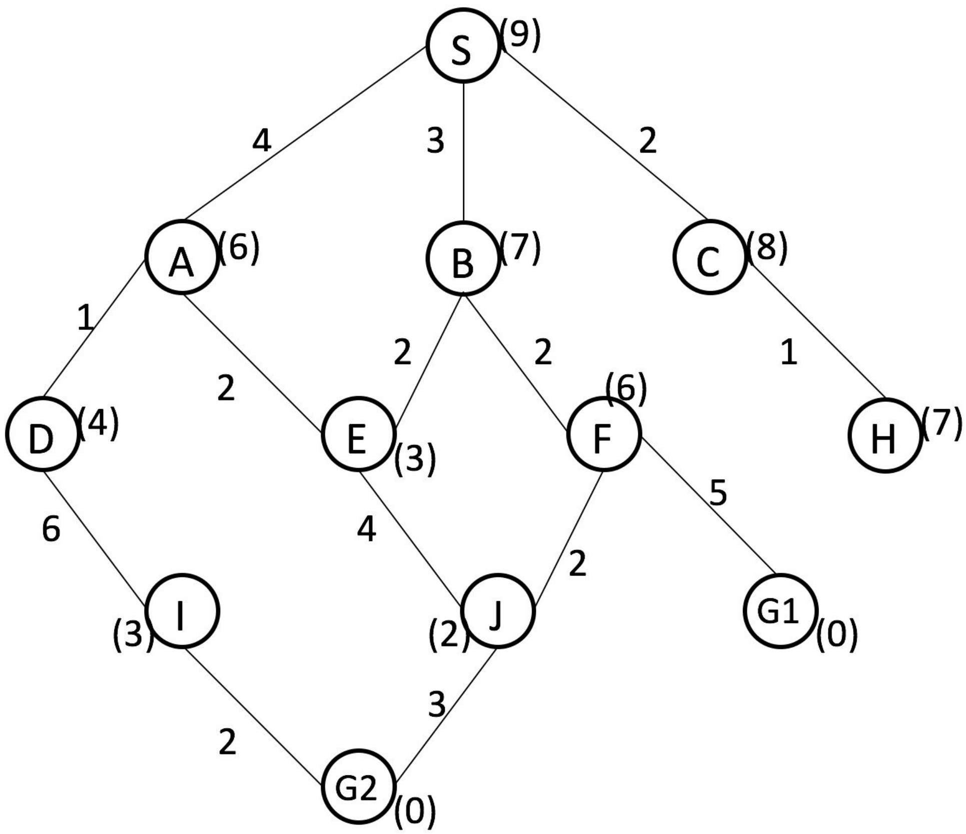
The why part should include a few of the following keywords:

In an episodic environment, each agent's performance is the result of a series of independent tasks performed. There is no link between the agent's performance and other different scenarios. In other words, the agent decides which action is best to take, it will only consider the task at hand and doesn't have to consider the effect it may have on future tasks.

5. [xx%] Classical Search

Consider the following graph. The start node is *S*. The goal nodes are G_1 and G_2. The cost of each transition is shown on the corresponding edge and the heuristic value of each node is shown within parentheses beside that node.

Using graph search, for each of the following search algorithms, show the order in which the nodes are expanded. Also show the path found to the goal. In case of a tie in the frontier, pick the nodes in alphabetical order. G_1 has higher priority than G_2 if the two goal nodes are in tie. Terminate the search if any goal node is chosen from the frontier for expansion.



1. [xx%] BFS

Nodes explored: $SABCDEFHIJG_1$

Path to goal: $SBFG_1$

2. [xx%] DFS

Nodes explored: $SADIG_2$

Path to goal: $SADIG_2$

3.[xx%] Uniform Cost search
Nodes explored: $SCBHADEFJG_1$
path to goal: $SBFG_1$

4. [xx%] Greedy Best-First Search
Nodes explored: $SAEJG_2$
Path to goal: $SAEJG_2$

5. [xx%] A* Search
Nodes explored: $SADEBCHFJG_1$
Path to goal: $SBFG_1$

Each letter is worth 1 point, deduct each wrong answer until zero.

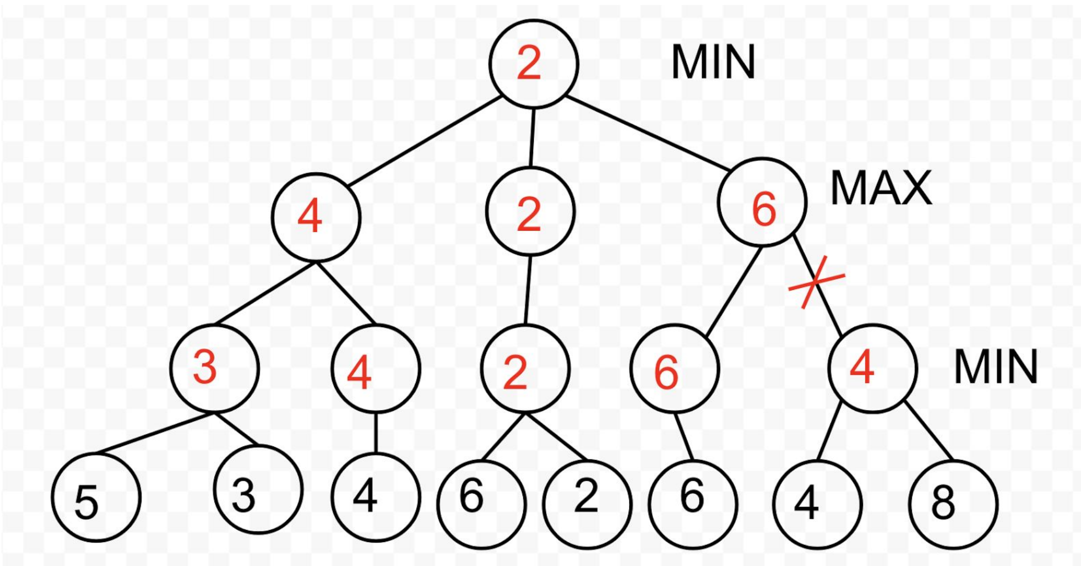
6. [xx%] Minimax

a. [xx%] Given the game tree below, write each node's minimax value beside it.

Deduct 1 point for each error until zero.

b. [xx%] On the same graph, cross out any edge that will not be explored when doing alpha-beta pruning and exploring nodes from left to right.

2 points if solution identifies one and only one correct edge, 1 point if the solution include the correct edge and some other incorrect edge(s), 0 point if the solution does not include the correct edge.

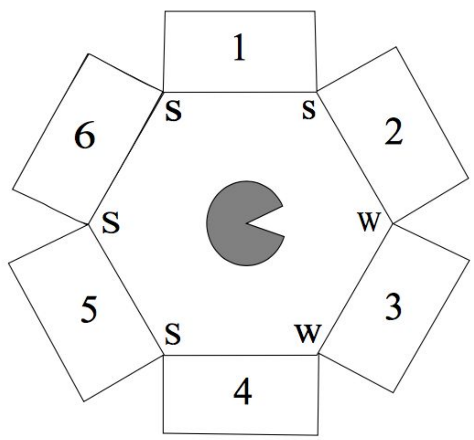


7. [xx%] CSP

Pacman is trapped! He is surrounded by mysterious corridors, each of which leads to either a pit (P), a ghost (G), or an exit (E). In order to escape, he needs to figure out which corridors, if any, lead to an exit and freedom, rather than the certain doom of a pit or a ghost.

The one sign of what lies behind the corridors is the wind: a pit produces a strong breeze (S) and an exit produces a weak breeze (W), while a ghost doesn't produce any breeze at all. Unfortunately, Pacman cannot measure the strength of the breeze at a specific corridor. Instead, he can stand between two adjacent corridors and feel the max of the two breezes. For example, if he stands between a pit and an exit he will sense a strong (S) breeze, while if he stands between an exit and a ghost, he will sense a weak (W) breeze. The measurements for all intersections are shown in the figure below.

Also, while the total number of exits might be zero, one, or more, Pacman knows that two neighboring squares will not both be exits.



Pacman models this problem using variables X_i for each corridor i and domains P, G, and E.

1. [xx%] State the binary and unary constraints for this CSP (either implicitly or explicitly).

Binary: 1 possible solution:

$< (X1, X2), [(P, G), (P, E), (P, P), (G, P), (E, P)] >$
$< (X2, X3), [(E, G), (E, P), (P, E), (G, E)] >$
$< (X3, X4), [(E, G), (E, P), (P, E), (G, E)] >$
$< (X4, X5), [(P, G), (P, E), (P, P), (G, P), (E, P)] >$
$< (X5, X6), [(P, G), (P, E), (P, P), (G, P), (E, P)] >$
$< (X1, X6), [(P, G), (P, E), (P, P), (G, P), (E, P)] >$

Binary: 1 possible solution:
$(1) X_1 = P \ or \ X_2 = P,$
$(2) X_2 = E \ or \ X_3 = E,$
$(3) X_3 = E \ or \ X_4 = E,$
$(4) X_4 = P \ or \ X_5 = P,$
$(5) X_5 = P \ or \ X_6 = P,$
$(6) X_1 = P \ or \ X_6 = P,$
(7) $\forall i,j \ s.t. \ Adj(i,j) \ \neg(X_i = E \ and \ X_j = E)$, any other solutions that can represent "any two neighboring X will not both be E" are correct.
e.g. $\{X_1 \neq E \ or X_2 \neq E \ , \ X_2 \neq E \ or X_3 \neq E \ , X_3 \neq E \ or X_4 \neq E \ , X_4 \neq E \ or X_5 \neq E \ , \ X_5 \neq E \ or X_6 \neq E, X_1 \neq E \ or X_6 \neq E\}.$
e.g. $\forall i,j \ s.t. \ Adj(i,j) \ , X_i \neq E \ or \ X_j \neq E$
e.g. $\nexists i,j \ s.t. \ Adj(i,j) \ , X_i = E \ and \ X_j = E$

Unary:
$(1) X_2 \neq P,$
$(2) X_3 \neq P \ ,$
$(3) X_4 \neq P$

Deduct 1 point for each wrong answer until zero

2. [xx%] Cross out the values from the domains of the variables that will be deleted in enforcing both node and arc consistency.

X_1	P	G	E
X_2	P	G	E
X_3	P	G	E

X_4	P	G	E
X_5	P	G	E
X_6	P	G	E

Solution:

X_1	P		
X_2		G	E
X_3		G	E
X_4		G	E
X_5	P		
X_6	P	G	E

3. [xx%] According to MRV (Minimum Remaining Values), which variable(s) could the solver assign first?

X_1 or X_5 (tie breaking)

4. [xx%] Assume that Pacman knows that $X_6 = G$. List all the solutions of this CSP or write *none* if no solutions exist.

(P, E, G, E, P, G)

(P, G, E, G, P, G)

8. [xx%] Problem Solving

Consider a sliding tile puzzle with six tiles (three black and three white) in a linear tray which can hold seven tiles. The following depicts the initial state:

[B][B][B][W][W][W][E]

where [B] represents a black tile, [W] a white one and an empty cell [E]. The puzzle has the following two legal moves:

(1) A tile may move into an adjacent empty cell with unit cost;
(2) A tile may hop over one or two tiles into the empty cell with cost equal to the number of tiles hopped over.

Thus, the initial state has the following three immediate successors:

[B][B][B][W][W][E][W] (cost = 1)
[B][B][B][W][E][W][W] (cost = 1)
[B][B][B][E][W][W][W] (cost = 2)

The goal is to have all of the white tiles to the left of all of the black ones. It is unimportant where the empty cell is. So the possible goal states are:

[W][W][W][B][B][B][E]
[W][W][W][B][B][E][B]
[W][W][W][B][E][B][B]
[W][W][W][E][B][B][B]
[W][W][E][W][B][B][B]
[W][E][W][W][B][B][B]
[E][W][W][W][B][B][B]

1. [XX%] Calculate the size of the state space by assuming the tiles of a given color are interchangeable or equivalent.

7! /(3! 3!) = 140.

2. [XX%] What is the maximum number of successors that a state can have, by considering all possible positions of the empty cell? What's the minimum? Calculate the branching factor (that is the average number of successors).

Max: six;
Min: three;
Branching factor: (3+4+5+6+5+4+3)/7 or ~ 4.3 or 30/7

3. [xx%] Let *h*(*s*) be the following heuristic: *h*(*s*) = the number of tiles that would have to be moved (by any number of spaces) for a state "*s*" to become a goal state. For example:

> h(BBXBWWW) = 5, because every piece but the rightmost black one must move to make a goal state.
> h(WWXBBWB) = 1, because only the rightmost white piece needs to be moved to make a goal state.
> h(WWWXBBB) = 0, because this is one of the goal states.

What are the *h* values for the three immediate successors of the initial state?

h(BBBXWWW) = 6, the other two equal to 5.

4. [xx%] Is this heuristic admissible?

Yes, because the heuristic is less than or equal to the cost.