Midterm Exam 1

**CSCI 561 Spring 2017: Artificial Intelligence**

| Student ID: | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|

Last Name: ________________________________________

First Name: ________________________________________

| USC E-Mail: | | | | | | | | | | | @usc.edu |
|---|---|---|---|---|---|---|---|---|---|---|---|

## Instructions:

1. Date: **2/14/2017 from 5:00pm – 6:20pm**
2. Maximum credits/points/percentage for this midterm: 100
3. The percentages for each question are indicated in square brackets [ ] near the question.
4. **<u>No books</u>** (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. **Answers must be written in the provided boxed only.** Please make sure NOT to write the answer to one question in the box for another one.
8. **Do NOT write on the 2D barcode.**
9. **The back of the pages will NOT be graded.** You should use the back of the pages only for SCRATCH PAPER, not the actual answers.
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** |
|---|---|
| 1- General AI Knowledge and Applications | 26 |
| 2- Graph Search | 28 |
| 3- Game Playing | 10 |
| 4- Constraint satisfaction | 21 |
| 5- Heuristic design | 15 |

## 1. [26%] General AI Knowledge and Application.

### True or False:

( T ) **1. [2%]** For any graph with the same cost on every arc, we can use breadth-first search to find the shortest path, if one exists.

( F ) **2. [2%]** Min-max algorithm might have different results from alpha-beta pruning on a certain game play.

( F ) **3. [2%]** Breadth-first search, depth-first search, uniform-cost search and greedy best-first search are uninformed search methods.

( F ) **4. [2%]** Breadth-first search is complete and has polynomial space complexity.

( F ) **5. [2%]** The heuristic function for A* needs to be admissible in order to find the optimal solution for graph search.

### Multiple Choice:

(a ) **1. [2%]** Which statement is true:

a. The hill-climbing search with sideways move, an infinite loop might occur.
b. Hill-climbing search is a complete algorithm.
c. The simulated-annealing algorithm is to start by shaking soft and then gradually increase the intensity of the shaking to avoid the local maximum/minimum.
d. Only (a) and (b).
e. Only (b) and (c).
f. Only (a) and (c).
g. All of the above.

(g ) **2. [2%]** Which statement is false?

a. Hill-climbing search is the best algorithm to us for problem domain with randomly distributed goals.
b. Greedy best-first search is complete.
c. A complete search algorithm will always find a correct solution.
d. (a) and (b).
e. (b) and (c).
f. (a) and (c).
g. (a), (b), and (c).

(d ) **3. [2%]** Which of the following is not an operator in genetic algorithm?

a. Crossover.
b. Mutation.
c. Inversion.
d. Schema.
e. None of the above.

(c ) **4. [2%]** The alpha value represents the ___ bound on the value of a ___ node.

a. upper, maximizing.
b. upper, minimizing.
c. lower, maximizing.
d. lower, minimizing.

(d ) **5. [2%]** In CSP, backtracking is based on?

a. Last in first out.
b. First in first out.
c. Recursion.
d. Both (a) & (c).

_Write down the **best** answer for each question in the boxes:

(a) **6. [2%]** What AI agent is most likely to have a speech recognition model?

a. Apple's Siri
b. Google's AlphaGo (the program to play the board game Go)
c. Robots that play soccer
d. All of the above
e. None of the above

(c) **7. [2%]** Assume that you are given two admissible heuristic functions h1(s) and h2(s). Which is also an admissible heuristic function?

a. h1(s) + h2(s)
b. h1(s) * h2(s)
c. min(h1(s), h2(s))
d. All of the above
e. None of the above

(d) **8. [2%]** Which AI application has a partially observed and stochastic environment?

a. Interactive AI Tutor

b. Taxi Driving

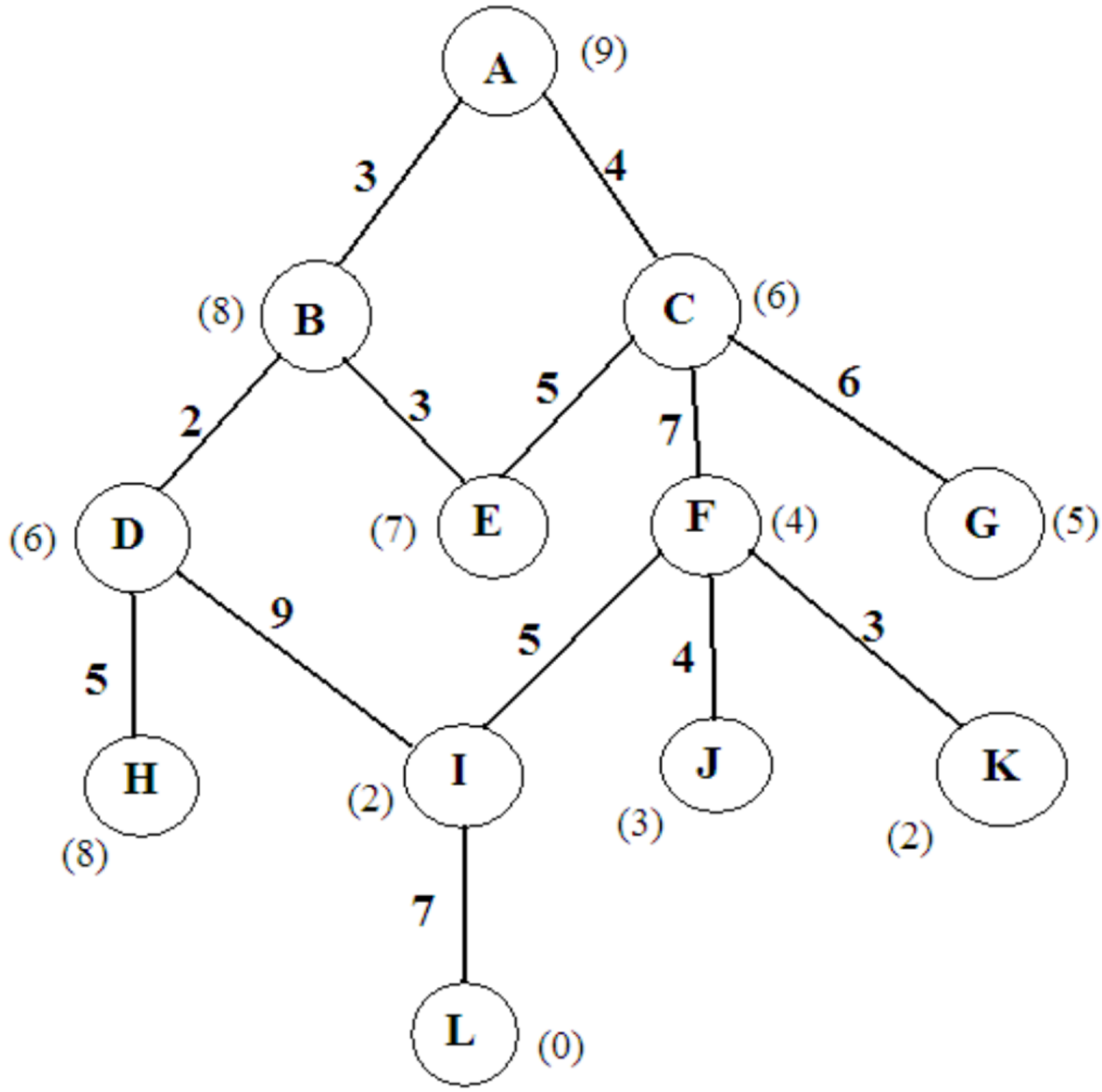
c. Poker

d. All of the above

e. None of the above

## 2.[28%] Search

Consider the following search tree. Each node is labeled with a unique letter. The start node for search is A, and the destination is L. The cost of each edge is shown on the edge. The heuristic value h for the node is shown within parentheses next to that node.




We are interested in going from A to L in the search tree above. For each of the following search algorithms, show the order in which the nodes are examined. If there is ambiguity or choice about which node goes next, pick the node that is leftmost in the tree. Terminate the search once the goal node is reached.

## 2A. [7%] Breadth-First Search

| [5%]Nodes Explored<br>**AB C DE F GHI J KL**<br><br>**Each wrong letter got -1**<br>**Ex:**<br>**If the student's answer: ACBDEFGHIJKL**<br>**Then -2**<br>**If the student's answer: ABCKJIH**<br>**Then -5** | [2%]Path to goal<br>**A B D I L** |
|---|---|

## 2B. [7%] Depth-First Search

| [5%]Nodes Explored<br>**AB DHI L** | [2%]Path to goal<br>**A B D I L** |
|---|---|

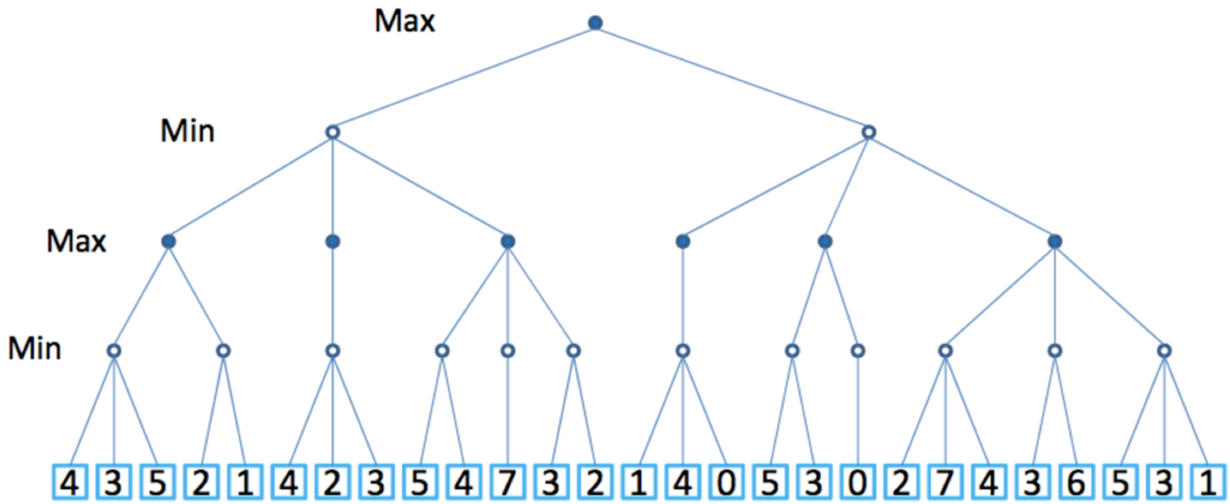
## 2C. [7%] Greedy Search

| [5%]Nodes Explored | [2%]Path to goal |
|---|---|
| AC F I L | AC F I L |

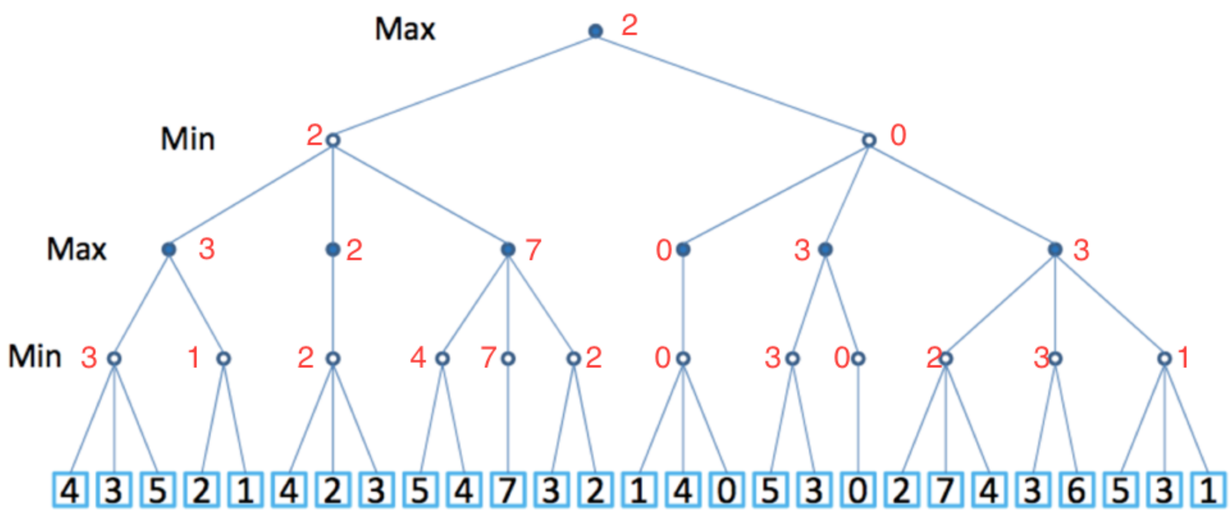
**2D. [7%] A* Search**

| [5%]Nodes Explored | [2%]Path to goal |
|---|---|
| A C B D E F G I KH J L | A B D I L |

## 3.[10%] Game Playing

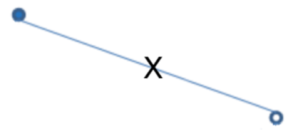


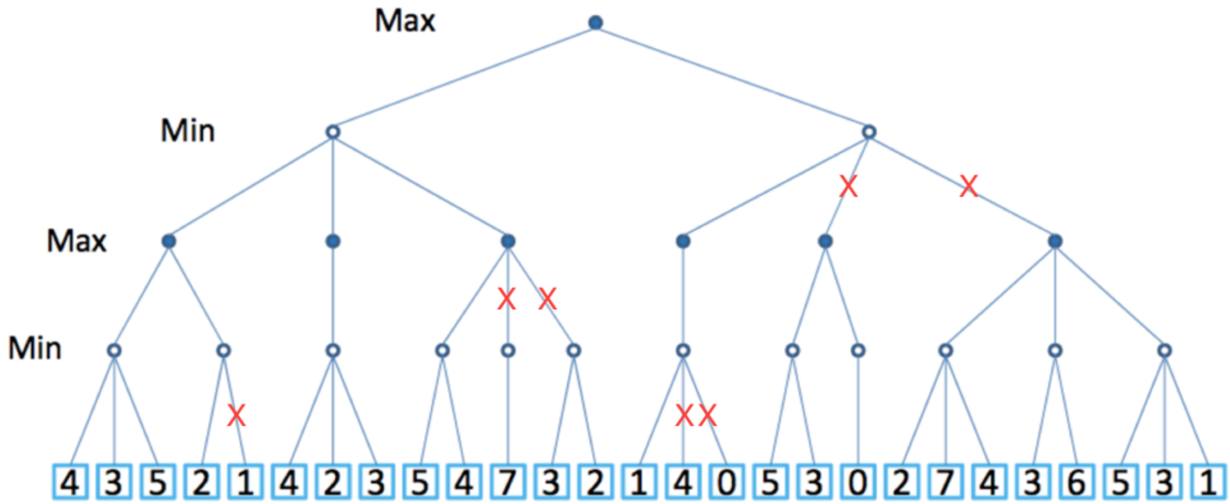

**3A. [5%] Perform the minimax algorithm on the game tree, without any pruning. Show the final min/max values at each node in the tree.**




**3B. [5%] Now with αβ pruning, please show which edges are pruned by marking X on the game tree above.**

**EX:**






## 4.[21%] Constraint Satisfaction

Suppose you have a small wedding to plan, and want to arrange the wedding seating for a certain number of guests in a hall. The hall has 3 tables, **T1, T2, T3** for seating. Some pairs of guests are couples or close Friends (**F**), and want to sit together at the same table. Some other pairs of guests are Enemies (**E**), and must be separated into different tables. The rest pairs are Indifferent (**I**) with each other, and do not mind sitting together or not. However, each pair of guests can ***only*** have one relationship, (F), (E) or (I). You should find a seating arrangement that satisfies all the constraints. The relationships of the guests are represented in the following table. The best man **Ben** has been assigned in **T1**, and the best woman **Emma** has been assigned in **T2**.

| | Ammy | Ben | Claire | David | Emma | Frank | Helen | Ian | Jane |
|---|---|---|---|---|---|---|---|---|---|
| Ammy | ------- | ------- | ------- | ------- | ------- | ------- | ------- | ------- | ------- |
| Ben | E | ------- | ------- | ------- | ------- | ------- | ------- | ------- | ------- |
| Claire | F | | ------- | ------- | ------- | ------- | ------- | ------- | ------- |
| David | | F | | ------- | ------- | ------- | ------- | ------- | ------- |
| Emma | | | | | ------- | ------- | ------- | ------- | ------- |
| Frank | | | | | F | ------- | ------- | ------- | ------- |
| Helen | | | E | | | | ------- | ------- | ------- |
| Ian | F | | | | | | | ------- | ------- |
| Jane | | | | | | F | | E | ------- |

**4A. [12%] Using the names of guests as variables, formulate this problem as a CSP with variables, domains and constraints. Constraints should be specified formally and precisely.**

**Variables {A, B, C, D, E, F, H, I, J} Domain {T1, T2, T3} [2pts each]**
**Constraints**

**A != B [1pts each]**
**A = C**
**B = D**
**E = F**
**C != H**
**A = I**
**F = J**
**I != J**
**B = T1 (can be ignored)**
**E = T2 (can be ignored)**

**4B.[5%] Show the domains of the variables after running arc-consistency and enforcing any unary constraints.**

**A {T3} [0.5pts]**
**B {T1} ..**
**C {T3}**
**D {T1}**
**E {T2}**
**F {T2}**
**H {T1, T2} [1pts]**
**I {T3}**
**J {T2}**

**4C. [4%] Give all the solutions to this CSP.**

Solution 1: [2pts]
A, C, I = T3
B, D = T1
E, F, J, H = T2

Solution 2: [2pts]

A, C, I = T3

B, D, H = T1

E, F, J= T2

**<u>5.[15%] Heuristic design</u>**

**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 | |

**where B represents a black tile, W a white one, and ☐ an empty cell . 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 | | W | **(cost = 1)** |
| B | B | B | W | | W | W | **(cost = 1)** |
| B | B | B | | 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 goal states are:**

| | | | | | | |
|---|---|---|---|---|---|---|
| W | W | W | B | B | B | |
| W | W | W | B | B | | B |
| W | W | W | B | | B | B |
| W | W | W | | B | B | B |

| | | | | | | |
|---|---|---|---|---|---|---|
| W | W | | W | B | B | B |
| W | | W | W | B | B | B |
| | W | W | W | B | B | B |

**We are analyzing the complexity of this problem.**

**<u>5A. [2%]</u> Calculate the size of the state space by assuming the tiles of a given color are interchangeable or equivalent.**

7! /(3! 3!) = 140.   [2pts]

**<u>5B.</u> <u>[7%]</u> Consider all possible positions of the empty cell, what is the maximum number of successors that a state can have? What's the minimum? Calculate the branching factor (that is the average number of successors). It's ok to make a move that undoes the previous move.**

Max = 6,
Min=3 [1 pt each]
Branching factor: (3+4+5+6+5+4+3)/7 or ~ 4.3   [0.5pt each branch, 1.5pt for the final result]

**<u>5C. [6%]</u> 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 achieve a goal state. For example:**

**h(B B   B W W W) = 5, since 1 black could remain where it is in a goal state.**

**h(W W   B B W B) = 1, since only 1 white needs to be moved.**

**h(W W W   B B B) = 0, since this is one of the goal states.**

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

h(BBBEWWW) = 6, [2pts]

the other two equal to 5.  [2pts each]