Midterm 1 Exam

CSCI 561 Fall 2017: Artificial Intelligence

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

Last Name: ____

First Name: ____

USC email: | | | | | | | | | @usc.edu

Instructions:

1. Date: **9/25/2017 from 8:00 pm – 9:50 pm in**
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**
1- General AI Knowledge	14
2- Search	30
3- Game Playing	16
4- CSP I	19
5- CSP II	15
6- Multiple Choice	6

1. [14%] General AI Knowledge

For each of the statements below, fill in the bubble **T** if the statement is **always and unconditionally true**, or fill in the bubble **F** if it is **always false, sometimes false, or just does not make sense**.

1	T	F
2	T	F
3	T	F
4	T	F
5	T	F
6	T	F
7	T	F

1- Model-based reflex agents select the next action based on condition-action rules. TRUE

2- Doubling a computer's speed allows us to double the depth of a tree search in the same amount of time. FALSE

3- All admissible heuristics are consistent. FALSE

4- Local search algorithms usually use a constant amount of memory. TRUE

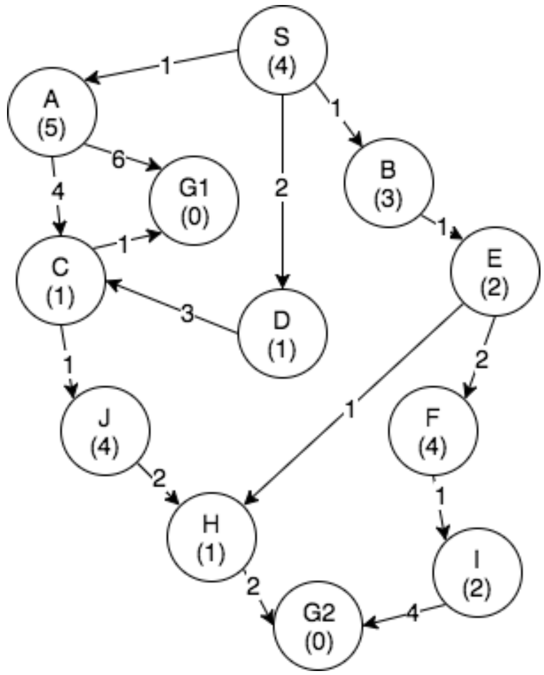
5- The number of nodes pruned by alpha-beta pruning is independent of the order in which the states are examined. FALSE

6- Least-constrained-variable and least-constraining-value heuristics are used to make backtracking search for CSPs more efficient. FALSE

7- Tree search version of A* search is optimal only if the heuristic is consistent. FALSE

2. [30%] Search

Consider the following graph. The start node is S, and the goal nodes are G1 and G2. The cost of each transition is shown on the corresponding edge and the heuristic value of each node is shown within parentheses on that node.



Using graph search, for each of the following search algorithms, show the order in which the nodes are expanded. In case of a tie in frontier, pick the nodes in alphabetical order. Terminate the search once the goal node is reached.

2A. [6%] BFS

[4%] The order of Nodes Expanded	**[2%] Solution Path**
S – A – B – D – C – G1 [Deduce 1 point for each wrong letter, for example SBADCG1 gets only 2 points.] OR **S – A (At this point G1 is added to the frontier.)** **[Deduce 1 point for each wrong letter, for example SB gets only 1 point.]**	S – A – G1 [Deduce 1 point for each wrong letter.]

2B. [6%] DFS

[4%] The order of Nodes Expanded	[2%] Solution Path
S – A – C – G1 [Deduce 1 point for each wrong letter.]	S – A – C – G1 [Deduce 1 point for each wrong letter.]
OR	OR
S – A (At this point G1 is added to the frontier.) **[Deduce 1 point for each wrong letter, for example SB gets only 1 point.]**	**S – A – G1** **[Deduce 1 point for each wrong letter.]**

2C. [6%] UCS

[4%] The order of Nodes Expanded	[2%] Solution Path
S – A – B – D – E – H – F – C – G2 [Deduce 1 point for each wrong letter.]	S – B – E – H – G2 [Deduce 1 point for each wrong letter.]

2D. [6%] Greedy Best-first Search

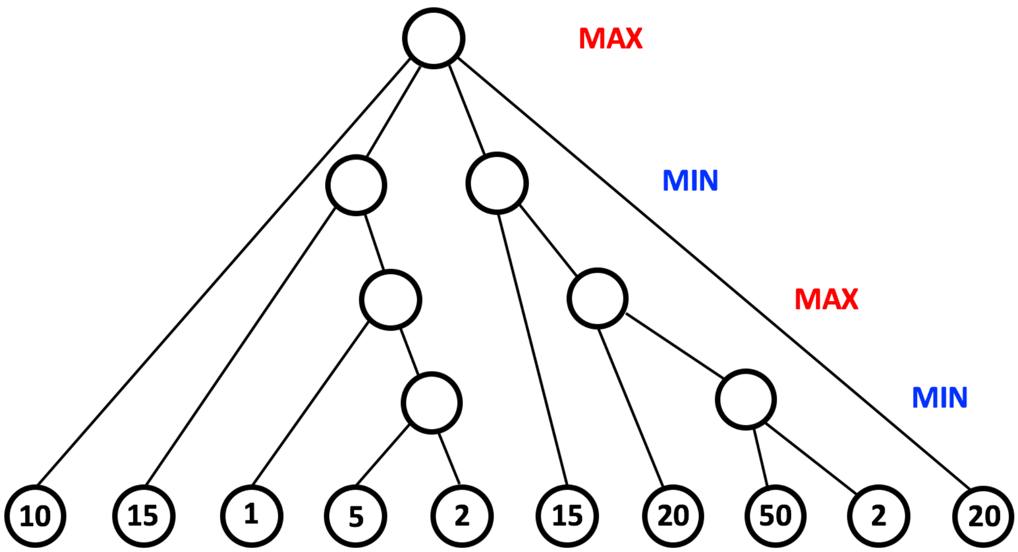
[4%] The order of Nodes Expanded	[2%] Solution Path
S – D – C – G1 [Deduce 1 point for each wrong letter.]	S – D – C – G1 [Deduce 1 point for each wrong letter.]

2E. [6%] A* Search

[4%] The order of Nodes Expanded	[2%] Solution Path
S – D – B – E – H – G2 [Deduce 1 point for each wrong letter.]	S – B – E – H – G2 [Deduce 1 point for each wrong letter.]

3. [16%] Game Playing

Consider the following game tree in which the evaluation function values are shown below each leaf node. Assume that the root node corresponds to the maximizing player. Assume that the search always visits children left-to-right.



(a) [4%] Compute the backed-up values using the minimax algorithm. **Show your answer by writing values next to the appropriate nodes in the above tree.**

20
2 15
2 20
2 2
Each error -1, until they got 0 points.
(b) [4%] Partial alpha-beta pseudocode is given below (only the min-value recursion), please fill in the 2 empty boxes to complete the pseudocode.

/*
alpha = the value of the best (i.e., highest-value) choice we have found so far at any choice point along the path for MAX.
beta = the value of the best (i.e., lowest-value) choice we have found so far at any choice point along the path for MIN.
*/
Function MIN-VALUE(state, alpha, beta) returns a utility value
If TERMINAL-TEST(state) then return UTILITY(state)
v ← +∞
for each a in ACTIONS(state) do
v ←MIN(v, MAX-VALUE(RESULT(s, a), alpha, beta))

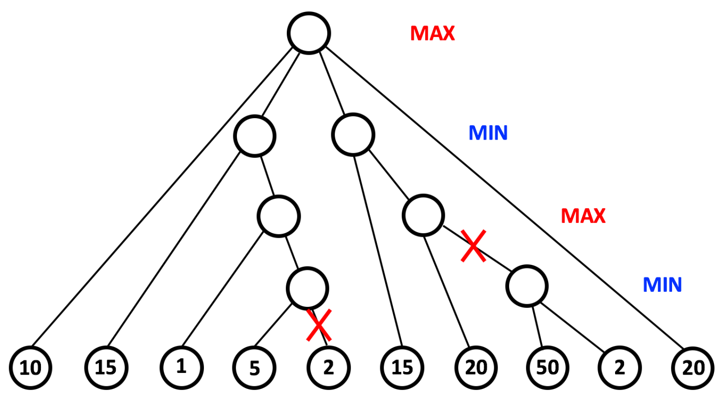
if v <= alpha then return v

beta <- MIN(beta, v)

return v

Each blank 2 points.
Any error in the blank -2.

(c) [4%] Which nodes will not be examined by the alpha-beta pruning algorithm? **Show your answer by crossing out the nodes that will be pruned.**



Each error -1, until they got 0 points.

(d) [4%] Propose a general, practical method for ordering children of nodes which will tend to increase the opportunities for pruning. You should be concise, but clearly state both what to do about min nodes and max nodes.

In general, we would want to use an evaluation function to estimate the values of the children and then order them so that at max nodes we order the children with larger estimated values first and at min nodes we order the children with larger estimated values first.

Max nodes: 2 points
Min nodes: 2 points

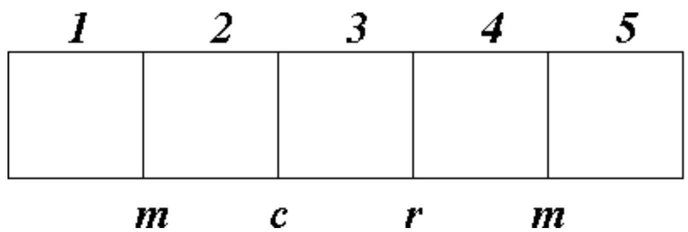
(e) [4%] Consider the same game tree but every minimizer node is replaced with a chance node (with an arbitrary but known probability distribution). If we apply alpha-beta pruning algorithm on it, will the pruning results be the same as (c)? Please briefly explain why no matter you answer YES or NO.

NO, once the chance node exists, we need to evaluate every terminal node in order to compute the value for the chance node. Since the terminal node might have arbitrary large or small value which will affect the result of minmax algorithm.

Yes/No: 2 point
Reason: 2 point

4. [19%] CSP I

You are a detective in charge of bringing down drug dealers (D). A tip has led you to a small apartment complex where you believe one or more D might be hiding out. There are five apartments in a row. Each apartment could contain a drug dealer D or could contain innocent people: adults (A), families with babies (B), or with teenagers (T). Before you break down a door, you need to be absolutely sure that a dealer D is inside, otherwise you could get sued for police suboptimality.



To help you narrow down where drug dealers D might be (if any are there at all!), you use the fact that different people make different noises. Every time you walk between two apartments, you can hear the louder of the two noises that are being made in those apartments. The loudest people are teenagers T, who blast music (m), the next loudest are babies B who cry (c), the next loudest is the drug dealer D, who makes a rustling sound (r), and the quietest people are adults A, who are totally silent (s). For example, if there were a baby in one house and a teenager in next, you would hear music (m) when standing between those apartments. Walking by the five apartments, you hear the noises shown in the diagram above. You decide to try solving this problem as a CSP.

a) [3%] What are the variables and domains in this CSP? (Do not use a formulation where the noises are variables; they'll show up as constraints.)

Variables:
1, 2, 3, 4, 5: apartments 1 through 5
Domains:
{A, B, T, D}: the kind of person in each apartment.

Each error -1, until they got 0 points.
If the Student wrote:
Variables: ABTD
Domains: 12345
Then -2.

b) [3%] Write down the binary and unary constraints implied by the noises shown above at the apartment boundaries.

Unary:
2 != T
3 != T
3 != B
4 != T
4 != B

Binary:
1 = T or 2 = T
2 = B or 3 = B
3 = D or 4 = D
4 = T or 5 = T

Each error -1, until they got 0 points.
Miss one constrain -1, until they got 0 points.

c) [3%] You decide to narrow down your domains by enforcing arc-consistency. What are the remaining domains of each variable after arc consistency is enforced?

$$
\begin{array}{l}
1\ [\quad\quad\quad\quad\ \mathrm{T}] \\
2\ [\quad\quad\quad \mathrm{B}\quad\ ] \\
3\ [\,\mathrm{D}\quad \mathrm{A}\quad\quad\ ] \\
4\ [\,\mathrm{D}\quad \mathrm{A}\quad\quad\ ] \\
5\ [\quad\quad\quad\quad\ \mathrm{T}]
\end{array}
$$

Each error row -1, until they got 0 points.
Miss one row -1, until they got 0 points.
Ex:
The student wrote:
1 [D]
2 [B]
3 [D T]
4 [D A]
5 [T]
Then -2

d) [3%] If you wanted to find a solution at this point, what variables could the MRV (minimum remaining values) ordering heuristic tell you to assign first?

1, 2, or 5.

Each error -1, until they got 0 points.

e) [3%] List all solutions to this CSP or state that none exist.

1	2	3	4	5
T	B	D	A	T
T	B	D	D	T
T	B	A	D	T

Each error row -1, until they got 0 points.

f) [2%] Is it ok for you to break into apartment 4? Justify your answer.

No, there could be an an adult (A) inside.

Yes/No: 1 point
Reason: 1 point

g) [2%] Imagine that you cannot actually solve CSPs yourself in the field. However, you can formulate them and give them to the police department's CSP detector. A CSP detector takes in your variables, domains, and unary and binary constraints, and outputs true if at least one solution exists. Using a detector like this, describe how you could prove that some apartment k was safe to storm into?

Add a constraint that k != D and submit to the detector. If the CSP detector returns false, it is safe to storm in because it means that there is no way to satisfy the configuration of sounds other than there being a drug dealer inside apartment k.

Constrain: 1 point
Reason: 1 point

5. [15%] CSP II

You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time.

The classes are:

- Class 1 - Intro to Programming: meets from 8:00-9:00am
- Class 2 - Intro to Artificial Intelligence: meets from 8:30-9:30am
- Class 3 - Natural Language Processing: meets from 9:00-10:00am
- Class 4 - Computer Vision: meets from 9:00-10:00am
- Class 5 - Machine Learning: meets from 9:30-10:30am

The professors are:

- Professor A, who is available to teach Classes 3 and 4.
- Professor B, who is available to teach Classes 2, 3, 4, and 5.
- Professor C, who is available to teach Classes 1, 2, 3, 4, 5.

a) [4%] Formulate this problem as a CSP problem in which there is one variable per class, stating the domains, and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

Variables	**Domains**
C1	**C**
C2	**B,C**
C3	**A,B,C**
C4	**A,B,C**
C5	**B,C**

Constraints:

C1 != C2

C2 != C3

C3 != C4

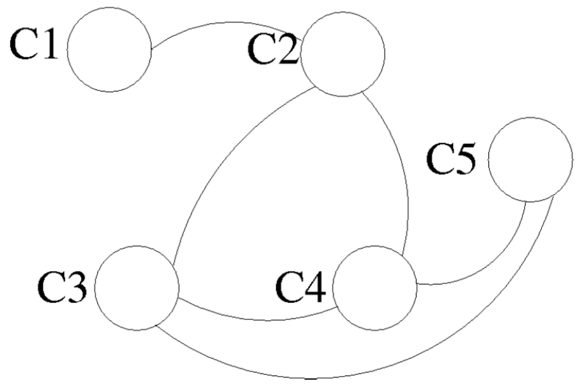
C4 != C5

C2 != C4

C3 != C5

Each error -1, until they got 0 points.
Miss one constrain -1, until they got 0 points.

b) [4%] Draw the constraint graph associated with your CSP.



Each error -1, until they got 0 points.
Miss one constrain -1, until they got 0 points.

c) [4%] Show the domains of the variables after running arc-consistency on this initial graph (after having already enforced any unary constraints).

Variable	**Domain**
C1	**C**
C2	**B**
C3	**A,C**
C4	**A,C**
C5	**B,C**

Note that C5 cannot possibly be C, but arc consistency does not rule it out.

Each error -1, until they got 0 points.
Miss one -1, until they got 0 points.

d) [3%] Give one solution to this CSP.

C1 = C, C2 = B, C3 = C, C4 = A, C5 = B. One other solution is possible (where C3 and C4 are switched).

Each error -1, until they got 0 points.

<u>6. [6%] Multiple Choice</u>

Each question has <u>one or more</u> correct choices. Check the boxes of all correct choices and leave the boxes of wrong choices blank. Please note that there will be no partial credit and you will receive full credit if and only if you choose all the correct choices and none of the wrong choices.

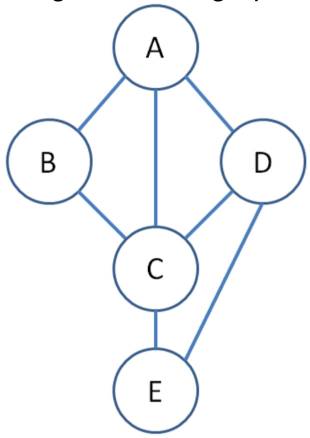
(a) [2%] Assume c_{ij} is the cost to go from node i to node j. Mark all choices for costs d_{ij} that make running Uniform Cost Search algorithm with these costs d_{ij} equivalent to running Uniform Cost Search with the original costs c_{ij}.

- ☐ $d_{ij} = c_{ij}^2$
- • $d_{ij} = \alpha c_{ij}, \alpha > 0$ (correct choice, no partial credit)
- ☐ $d_{ij} = c_{ij} + \alpha, \alpha > 0$
- ☐ $d_{ij} = \alpha c_{ij} + \beta, \alpha, \beta > 0$

(b) [2%] If both h1(s) and h2(s) are admissible heuristics, then which of the following are also guaranteed to be admissible heuristics?

- ☐ h3(s) = h1(s) + h2(s)
- • h3(s) = α.h1(s) + (1-α).h2(s) (0 < α < 1) (correct choice, no partial credit)
- • h3(s) = Max(h1(s), h2(s)) (correct choice, no partial credit)
- ☐ h3(s) = h1(s) - h2(s)

(c) [2%] According to degree heuristic, which variable(s) could be selected first in the CSP problem represented by the following constraint graph?



- ☐ A
- ☐ B
- • C (correct choice, no partial credit)
- ☐ D
- ☐ E