CSCI 3202
Midterm Exam
Spring 2018

Name: **Solution**

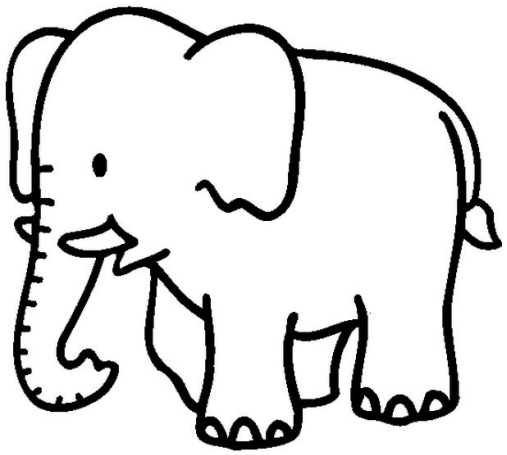
By writing my name here, I agree to the exam rules outlined below as well as in the course syllabus, and the CU Boulder Honor Code pledge:

On my honor, as a University of Colorado Boulder student, I have neither given nor received unauthorized assistance.

Read the following instructions.

- **Write your name** on the provided line above **and on the back of the last page**.
- You may use a calculator provided that it cannot access the internet or store large amounts of data.
- You may **not** use a smartphone in any capacity.
- You must **clearly justify all conclusions** to receive full credit.
  A correct answer with no supporting work will receive little/no credit.
- If you need more space, there are some blank pages at the end of the exam.
  Please clearly indicate what work is associated with which problems.
- If you need to leave the exam room during the exam for any reason, **raise your hand** and show me that you **closed your exam** and **placed your phone** on top of it. Then, go do your thing.
- You have **50 minutes** for this exam.

#	Points possible	Score
1	8	
2	12	
3	25	
4	10	
5	25	
6	20	
Total	100	



1. [8 points]

Consider the task environment of a Roomba in a 3x3 tile room. Each tile is either i) clean or ii) dirty. Roomba can **clean**, **turn**, **move**, and **do nothing**. Assume Roomba must be pointing in the direction it wants to move before that move can be made. An **example** state-of-the-world is given at right. (The arrow denotes the direction Roomba is facing, the circular thing is a Roomba, and the splotchy-looking things are piles of dirt.)

What is the size of the state space? You do not need to simplify your answer.

- Each of 9 tiles is either CLEAN or DIRTY → 2^9 +2
- Roomba in one of 9 tiles → 9 +2
- Roomba facing one of 4 directions (N/S/E/W) → 4 +2

Total: $2^9 \times 9 \times 4$) +2 multiply together

Common mistake:
- 9^2 (no deduction if your reasoning was clear)
- 2×9 (-2)
- 9! (-2)

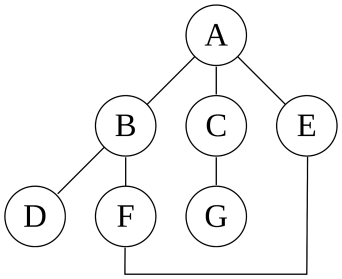
Common mistake: counting actions as characteristic of state space → -1 pt.

2. [12 pts] Suppose we have breadth-first search and depth-first search algorithms wherein states are added to the frontier in alphabetical order. Assume we do not permit any redundant/loopy paths.

Consider the task of finding a path from D to H. Yes, I know that H is not on the graph.

6 pts

a. What is the order in which states will be **added to the frontier** using **breadth-first search**? Which node is the parent of node E?



If you started from A, you'd find:
A, B, C, E, D, F, G ← 3/5 pts
& parent(E) = A ← 1 pt

-1 if expand in wrong order just once

-2 if expand in wrong order consistently

BFS frontier is First-In-First-Out:

1. Start with FRONTIER = {D}
2. EXPLORED {D} // FRONTIER {B} [added to frontier: D, B]
3. EXPLORED {D, B} // FRONTIER {A, F} [added to frontier: D, B, A, F]
4. EXPLORED {D, B, A} // FRONTIER {F, C, E} [added to frontier D, B, A, F, C, E]
5. EXPLORED {D, B, A, F} // FRONTIER {C, E} [added to frontier D, B, A, F, C, E]
6. EXPLORED {D, B, A, F, C} // FRONTIER {E, G} **[added to frontier D, B, A, F, C, E, G]**

Parent of E is A) +1

If done this way in both, consistently, -2 total (don't double-count)

6 pts.

b. What is the order in which states will be **explored** (expanded) using **depth-first search**? Which node is the parent of node E?

If you started from A:
A, E, F, C, G, B, D ← 3/5 pts
& parent(E) = A ← 1 pt.

DFS frontier is Last-In-First-Out:

1. Start with FRONTIER = {D}
2. EXPLORED {D} // FRONTIER {B}
3. EXPLORED {D, B} // FRONTIER {A, F}
4. EXPLORED {D, B, F} // FRONTIER {A, E}
5. EXPLORED {D, B, F, E} // FRONTIER {A}
6. EXPLORED {D, B, F, E, A} // FRONTIER {C}

… can see by now that we will have **EXPLORED {D, B, F, E, A, C, G}**

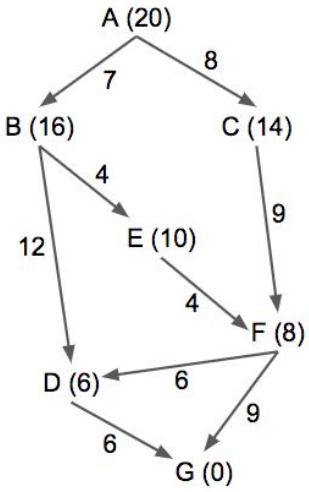
And parent of E is F) +1

-2 for getting ABC LIFO backwards (e.g., D→B→A→C→...)
(don't double-count w/ "wrong order" deductions)

-2 if don't search for H (incomplete search)

3. [25 pts]

Here is a **directed** state space graph. This means, for example, that from A you can only get to B and C, but not the other way around. The values of an admissible heuristic are given in parentheses next to the node names and the step costs to travel between two nodes are given as the edge weights.



15 pts a. In what order would A* search explore the state space to find the solution path from A to G? Include in your answer:

i. the *f*-costs associated with each state as they are **explored**,
ii. the **optimal** path cost, and
iii. what node is the parent of F in the final search tree?

1. Start with A: (f=20) (no deduction for not expanding A)
   EXPLORED {A (f=20)}
   FRONTIER {B (g=7, h=16, f=23), C (g=8, h=14, f=22)}
2. Pop off C (f=22)
   EXPLORED {A (f=20), C (f=22)}
   FRONTIER {B (g=7, h=16, f=23), F (g=17, h=8, f=25)}
3. Pop off B (f=23)
   EXPLORED {A (f=20), C (f=22), B (f=23)}
   FRONTIER {F (g=17, h=8, f=25), E (g=11, h=10, f=21), D (g=19, h=6, f=25)}
4. Pop off E (and replace F on frontier with new path) (f=21)
   EXPLORED {A (f=20), C (f=22), B (f=23), E (f=21)}
   FRONTIER {~~F (g=17, h=8, f=25)~~, D (g=19, h=6, f=25), F (g=15, h=8, f=23)}
5. Pop off F (and reject new possible path to D) (f=23)
   EXPLORED {A (f=20), C (f=22), B (f=23), E (f=21), F (f=23)}
   FRONTIER {D (g=19, h=6, f=25), ~~D (g=21, h=6, f=27)~~, G (g=24, h=0, f=24)}
6. Pop off G … and we're done! (f=24)

9 pts **i. f-costs are above**
- return step costs → +1/9
- return h or g → +2/9 ← b/c this would be greedy / UCS & not really A*
- -1 for each incorrect; if doing A* but w/ wrong heuristic

3 pts **ii. Optimal path is A → B → E → F → G with cost 24**
↳ 2/3 for correct path but no cost (consistent w/ (i))

-1 if give value for D, & D is in their search path bc it is not explored)

3 pts **iii. Final parent of F is E (because of the replacement when we found a shorter path)**
↳ 1/3 for C ; 0 for any other node

5 pts

b. Provide **one** example of a single modification to this graph that would make the heuristic inadmissible. Fully justify your answer.

+2 (explanation)

Many possible answers. Admissible means optimistic, so any change such that the heuristic now overestimates the optimal distance to goal from any given node.

+3 (example)

One possibility is to change the step cost from F to G to anything less than 8.
Or you could change the heuristic value h(F) to anything greater than 9 (for example).

5 pts

c. Is this heuristic **consistent**? Fully justify your answer.

+2

No -- there are a couple examples. Need to satisfy the triangle inequality: $h(n) \leq h(n') + c(n, n')$ for any node n, and n' a successor of n, and c(n, n') the step cost to go from n to n'

+1 for checking nodes

One example: $h(B) = 16 > h(E) + c(B, E) = 10 + 4 = 14$

+2 for working example

+½ if say "yes" & give a "good" reason
0 if just say "yes" (or "no") (like def of cons.)

4. [10 pts] Suppose you are using **simulated annealing** to find a global optimum of some objective function. But, gosh darn it, you keep getting stuck in what you know to be a local optimum! What are two options for escaping or diagnosing a local optimum versus a global one?

   Be **specific** in your answer, and indicate how exactly each option will help. For example, a response of "use a different function" would receive 0 points.

   1. Could alter the temperature updating schedule, so that we accept more suboptimal moves in order to escape the local optimum
   2. Could use random restarts, to see how many/which restarts end up in each possible optimum
   3. Could change the step size, so that we propose more extreme moves (this may or may not be applicable, depending on the application)
   4. Try a beam search approach, where many "beams" are started and you do simulated annealing on each, and take the best *k* simulations each iteration

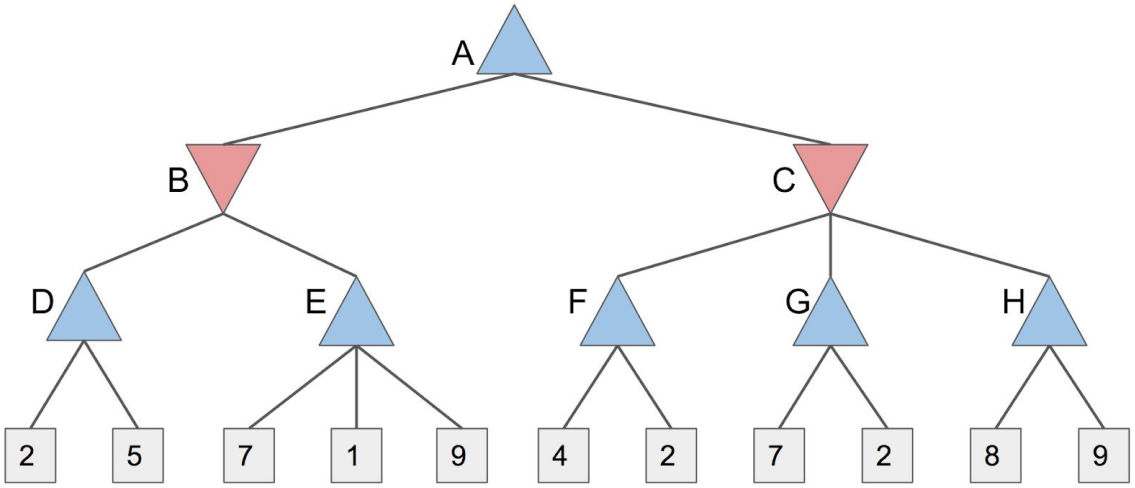
5 pts for each example

3 pts: partial explanation

2 pts: something SA already does

5 pts: fully specified what to do/how to help

5. [25 pts] Below is a Max/Min game tree, as seen in class. The values in the boxes denote utility to Max, the upward-pointing triangles are Max nodes and the downward-pointing triangles are Min nodes.



6 pts. a. What are the minimax values associated with each node? Do **not** consider any pruning at this point. Write your answers in the table below.

-1 for each incorrect, up to max. deduction of all 6 pts

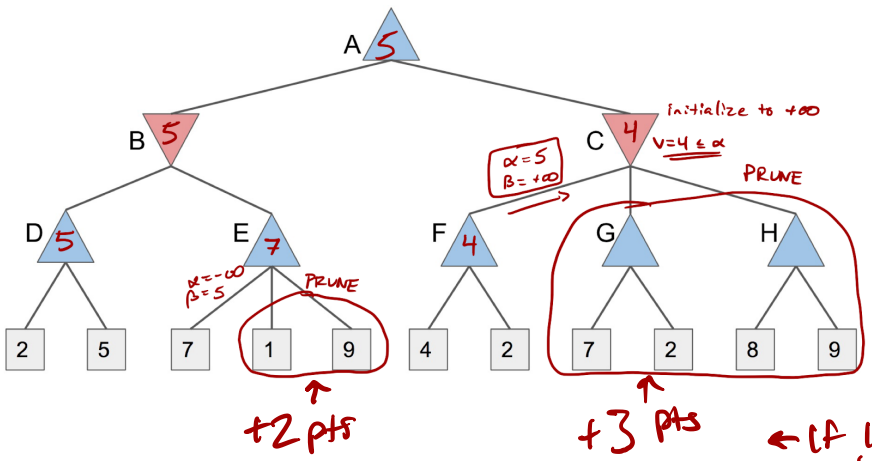
If have max/min as opposites, award 3/6

		A. 5		
	B. 5		C. 4	
D. 5	E. 9	F. 4	G. 7	H. 9

7 pts. b. **Indicate clearly** in the above figure which branches/leaves are pruned when alpha-beta pruning is applied to this game tree. Assume that nodes are expanded from left-to-right at each layer.

Below, briefly justify how you know those branches/leaves can be pruned. Vague responses along the lines of simply saying "alpha-beta pruning algorithm" will receive 0 points.

Note: This figure is reproduced in the extra pages at the end of this exam, just in case your figure gets too messy and you need a do-over. Be sure to indicate which version you would like graded!



Specifically, we prune the 1 and the 9 because the value for E, 7, is greater than the current value for beta (5). So Min would never let the game go there.

And we prune G and H subtrees because the value for C, 4, is less than alpha (5). So Max would never let the game go there.

+2 justify

+2 pts

+3 pts

← If leave a node or small subtree out of pruning, just -1 pt each time

4 pts

c. Consider alpha-beta pruning, as in part b. What are the values of α and β (alpha and beta) as we proceed from C to F?

alpha = 5, beta = +infinity

+2 +2

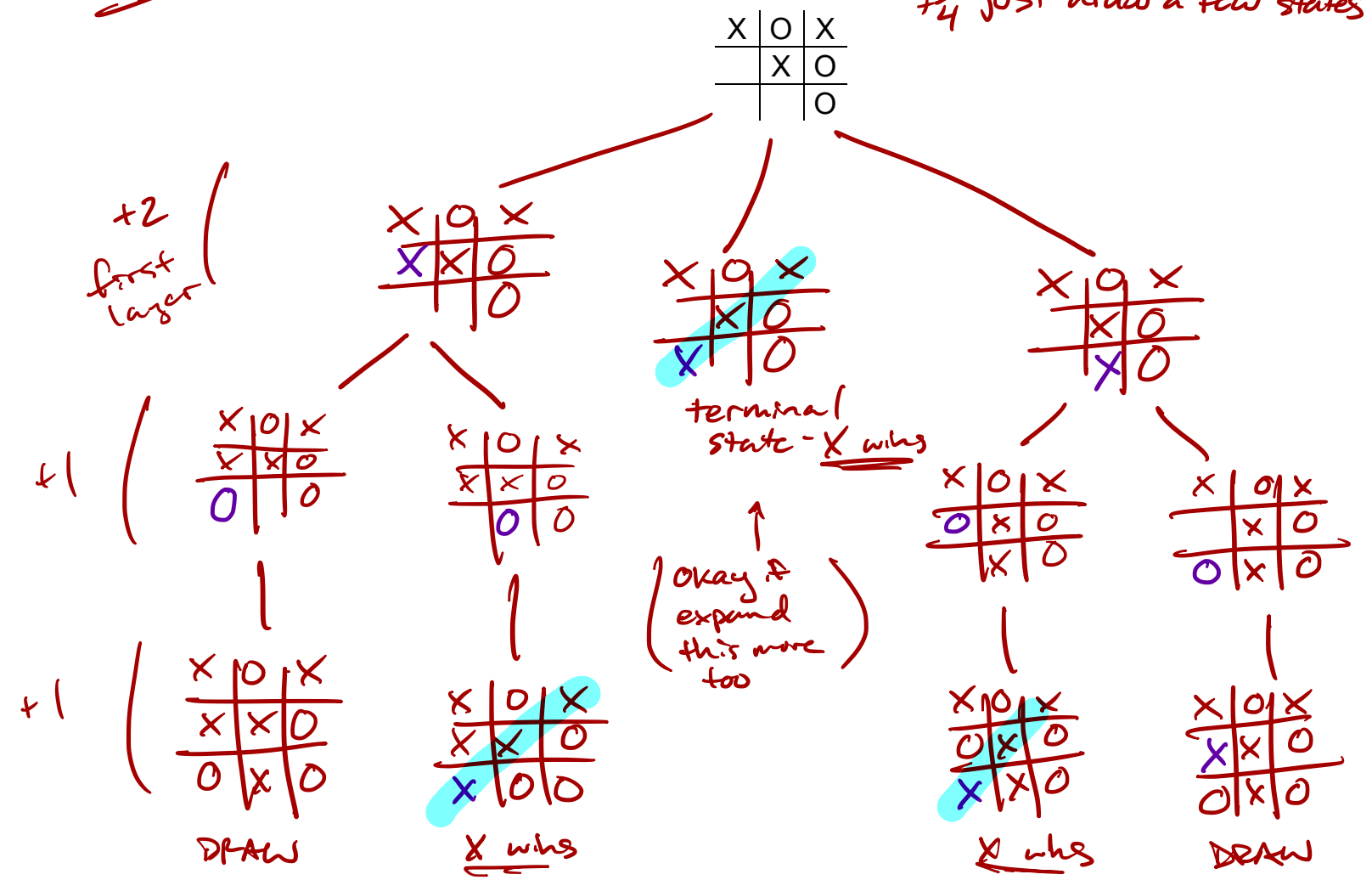
if swapped, 2/4

& okay as long as consistent w/ (a) & (b)

4 pts

d. Draw the rest of the min/max game tree starting from the Tic-Tac-Toe state given below. Assume X had the first move.

+1/4 just draw a few states



4 pts.

e. If the O player was playing optimally, would the game ever come to this state? Why or why not?

+2

No. If O were playing optimally, they would never find themself in a position where X can win. More to the point, there are no terminal states in this subtree in which O is the winner, so this subtree would have been **pruned**

+2 for justification

+1/4 SOME attempt to reason it out

-1 for just a picture (if it works)

6. [20 points] Parts a-c are unrelated.

a. Suppose you are deciding when to arrive at a party. There is some optimal time to arrive when the loss you feel, as measured by *awkwardness*, is minimized at 0. That is, at some particular time, it is not awkward at all to show up to the party. The awkwardness (loss) increases as you arrive too early or too late relative to this optimal time. What is a suitable loss function, $L(d, x)$, to model this situation? Include definitions for d and x, consistent with the examples from this class. Use this loss function this weekend.

4 pts

d = when you arrive
x = when you should've arrived <-1 no def. for d, x

Many possible answers. Anything that has a min of 0 when d=x and increases as |d-x| increases is okay.

One possibility is quadratic loss: $L(d,x) = (d-x)^2$

Another is absolute value loss: $L(d,x) = |d-x|$

-1 for (d-x)

2 pts: min of 0 at d=x
2 pts: increases as |d-x| increases

b. Suppose we have a situation where loss is given by the function $L(d, x) = 2(d - x)^2$.

8 pts.

Set up, simplify, **but do not evaluate** integral(s) for the expected loss, $E_x[L(d, x)]$, where your prior beliefs regarding x follow the distribution $f(x)$ given at right. You may assume $f(x) = 0$ for values of x outside the interval [0, 3].

$$f(x) = \begin{cases} 1/2 & 0 \le x < 1 \\ 3/8 & 1 \le x < 2 \\ 1/8 & 2 \le x \le 3 \end{cases}$$

$$\begin{aligned} E_x[L(d,x)] &= \int_{-\infty}^{\infty} L(d,x) f(x)\, dx \\ &= \int_0^1 2(d-x)^2 \cdot \frac{1}{2}\, dx + \int_1^2 2(d-x)^2 \cdot \frac{3}{8}\, dx + \int_2^3 2(d-x)^2 \cdot \frac{1}{8}\, dx \\ &= \int_0^1 (d-x)^2\, dx + \frac{3}{4}\int_1^2 (d-x)^2\, dx + \frac{1}{4}\int_2^3 (d-x)^2\, dx \end{aligned}$$

+2 starting integral

+2 +2 +2

5/8 if missing either L or f but otherwise right form

c. Suppose our expected loss is represented by the function $E_x[L(d, x)] = (2 - d)^2 + 2$, and our prior beliefs regarding x are given by the distribution $f(x)$ from part b.

8 pts.

i. Calculate the Bayes' Decision, d_{Bayes}

ii. Calculate the Expected Value of Including Uncertainty, EVIU? Suppose that if we ignore uncertainty, our best guess for what decision to make is the **median** of x (under our prior $f(x)$).

i. Bayes' Decision minimizes expected loss → $d_{Bayes} = 2$

(just reading it off the quadratic is fine, or differentiating and setting equal to 0)

+4

2 pts if only recognize E[L]=0 but don't get anywhere

ii. $EVIU = E_x[L(d_{iu}, x)] - E_x[L(d_{Bayes}, x)]$

and $d_{iu} = \text{median}(x) = 1$ (half the probability mass on either side, can read right off the distribution)

+2

→ $EVIU = E_x[L(d_{iu}, x)] - E_x[L(d_{Bayes}, x)]$

$= [(2 - 1)^2 + 2] - [(2 - 2)^2 + 2]$

$= 1$

+2

If just say EVIU = 0 b/c quad. loss, +2/4 pts