CSE 415–Autumn 2017 — Midterm Examination

by the Staff of CSE 415, Autumn 2017

INSTRUCTIONS: Write your full name at the top of this cover page. Make sure you have all 8 pages. **Also write your name on top of every page or it is possible that you would not receive credit for that page.** (We will be removing staples from the exams, and if a page of yours gets lost due to not having your name on it, you could miss credit for that page.)

Put your answers inside the framed rectangular boxes that are provided. Write legibly and if you use a pencil, make sure that you write darkly enough that a normal scanner will pick up your writing. If you need more space, use the margins. Do all problems. This is a CLOSED-BOOK, CLOSED-NOTES examination. Do not use any books, notes, calculators, or other electronic devices. There are five problems worth 20 points each for a total of 100. Each problem has multiple parts, and the allocations of points among the parts are as shown on each individual problem.

1. Short Answer (20 points)

(a) (5 points) In state-space search, a set of operators specifies how new states can be produced from existing states. Why is the state-transformation function component of an operator in general only a partial function? (Partial means not necessarily defined on the whole domain. Here our domain is the set of possible states.)

It is often the case that an operator can only be applied to a subset of all the possible states as each operator requires a specific set of preconditions to be met before it can be applied. As a result, the operator is not defined for all the elements of its domain and is thus a partial function

(b) (5 points) Using some of the following symbols, write a formula that expresses the condition under which a heuristic function $h_i(s)$ is admissible, where $h(s)$ is the true shortest distance from s to the nearest goal. Σ represents the set of possible states.

$$h, s, h_i, \leq, \geq, <, >, 0, \in, \Sigma, \forall, \exists, (,)$$

for all s in Sigma, h_i(s) <= h(s)

(c) (5 points) In search using basic hill climbing, what is the main hazard?

we didnt cover this

(d) (5 points) Explain how to determine the size of the state space for a Tower of Hanoi puzzle with n disks. (We assume 3 pegs, as usual.)

Each of the n disks can be on one of 3 pegs, so there are 3^n possible arrangements where the disks are in a valid order (disks decrease in size going up the stack)

2. (20 points) Python

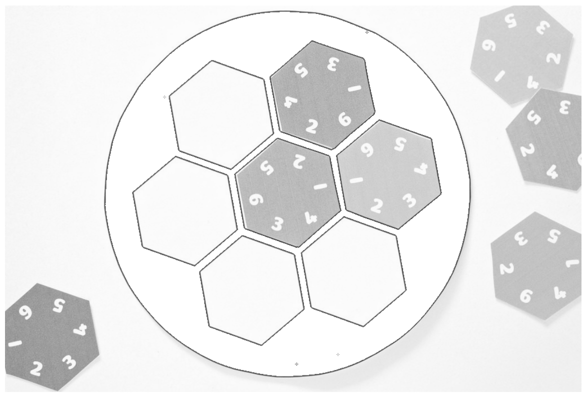
Complete a class definition for a Poker hand, by completing the methods that have stubs provided. The Card class is given for reference and you do not need to change it.

```
class Card:
  def __init__(self, suit, rank)
    self.suit = suit
    self.rank = rank

class PokerHand:
  def __init__(self):  # For creating an empty hand.
    # Initialize a list that will be used to hold the cards.



  def insert(self, card): # Use this to insert a card into the hand.
    # If there are fewer than 5 cards in the list, then
    #    put the card into the list and return True.
    # otherwise return False.



  def copy(self): # create and return a copy of the hand.
    # It is not necessary to make copies of the separate cards,
    # but the list of cards must be copied in the new PokerHand object.



  def isFlush(self):  # If the poker hand is a flush, return True.
    # To be a flush, there must be 5 cards, and all their suits must be
    # equal.
    # Otherwise return False.
```

NO Python on our midterm

3. Determining the Sizes of State Spaces (20 points)

Let us consider the combinatorics of a class of puzzles we will call "Labeled Hexagon" puzzles. In one of these puzzles, seven hexagonal tiles are to be placed into a tray that has a central place for one hexagon and six places around it for the other hexagons. Each hexagon has its sides labeled with numbers from 1 to 6, but the numbers are not necessarily all used or distinct; for example, there could be a hexagon whose sides are all labeled 1 or a hexagon whose sides are alternately labeled 3 and 5. They are like the Painted Squares from the reading in that the available markings for the sides can be arranged in any manner. (Dominoes also have this property.) An example is shown in the figure below.



(a) If all the hexagonal tiles had all their sides labeled with only the number 1, but each of the 7 hexagons used a different color, how many different ways could we fill the tray with the 7 hexagons?

7!

(b) Now let's assume the hexagons still have different colors, but that they each have a pattern of all 6 numbers, as in the figure. In addition to the choices of what tile to put into each place, as in part (a), let's also have a choice for each tile of 6 different rotational orientations for it. Give an expression for the total number of ways to fill the tray. Here we won't require that numbers that meet on the sides of hexagons match each other. (However, they should match if the state is a goal state.)

7! * 6^7

(c) Finally, consider that an algorithm for finding a suitable arrangement of the tiles, starting from an empty tray will deal with many partially-filled tray arrangements. The initial state corresponds to an empty tray, and successors of the initial state each have one tile placed somewhere in the tray. Give an expression for the total number of states in the state space for this problem. (Assume operators are available to choose any unplaced piece to go in any vacant space, and to rotate it to any of its 6 orientations.) Briefly explain your formula.

suppose we have 0 <= k <= 7 tiles placed. There are (7 choose k) ways to choose which tiles these will be, (7 choose k)*k! ways to order these tiles into the slots on the board, and 6^k ways to choose their arrangements. Thus, overall there are

sum k = 0 to 7 of (7 choose k)^2 6^k total states in the state space.

4. Heuristic Search (20 points)

In a fast-action video game, one of the enemy agents (the "bot") uses the A* algorithm to find a path to its goal. The game takes place in a 2D grid world. The bot has operators that correspond to 1-step moves to go either north, east, west, or south. The cells marked B are blocked. There are obstacles in those cells, and the bot is not allowed to go into them, but must travel around them in order to get to G. It uses the following heuristic function in its search. Here $x(s)$ is the x-coordinate of the state s, and $y(s)$ is its y-coordinate.

$$h(s) = 0.8 \cdot |x(G) - x(s)| + 0.1 \cdot |y(G) - y(s)|$$

A simulation of the A* algorithm has been started. The first iteration of the algorithm has already been performed for you, and you can see that the initial state is the first state visited (V = 1); and the g values of each successor are 1, because those successors are one step from I. The h value of the north successor of I is 4 + 0, because the difference in x coordinates between it and the goal is 5, and multiplying that by 0.8 gives us 4, and the difference between the y coordinates is 0. Among the three successors of I, the one with lowest f is the one to the east having f = 4.3. So that is chosen at the beginning of the second iteration, and it has V=2 indicated.

(a) (5 points) Is h admissible? Why or why not? What is the significance of this?

Yes. Since the bot can only move up/down and left/right, it changes its x or y coordinate by 1 each move. Thus, its distance to the goal is at least |x(G) - x(s)| + |y(G) - y(s)| and our heuristic is certainly less than that. This implies that when our search finds a goal state, the path we took to reach there is an optimal one.

(b) (15 points) Finish simulating the A* algorithm to find a path from I to G. Whenever a successor is generated by the algorithm, write the g, h, and f values for that state as shown. Show the visitation order of the states (the order in which states come off the OPEN priority queue via the DELETEMIN operation.) Also, draw the backpointer links that the algorithm uses to keep track of the parentage of each state generated.

0.8 dx + 0.1 dy

g = 3 h = 4 + 0.2 f = 7.2 V = 17	g = 4 h = 3.2 + 0.2 f = 7.4 V = 18	g = 5 h = 2.4 + 0.2 f = 7.6 V = 19	g = 6 h = 1.6 + 0.2 f = 7.8 V = 20	g = 7 h = 0.8 + 0.2 f = 8	
g = 2 h = 4 + 0.1 f = 6.1 V = 13	g = 3 h = 3.2+0.1 f = 6.3 V = 15	**B**	g = 7 h = 1.6 + 0.2 f = 8.8		
g = 1 h = 4 + 0 f = 5 V = 5	g=2 h = 3.2+0 f = 5.2 V = 7	g = 3 h = 2.4 + 0 f = 5.4 V = 9	**B**		g = 8 h = 0 f = 8 **G** V = 21
I V = 1	g = 1 h = 3.2 + 0.1 f = 4.3 V = 2	g=2 h = 2.4 + 0.1 f = 4.5 V = 3	g = 3 h = 1.6 + 0.1 f = 4.7 V = 4	**B**	g = 7 h = 0 + 0.1 f = 7.1 V = 16
g = 1 h = 4 + 0.2 f = 5.2 V = 6	g=2 h = 3.2+0.2 f = 5.4 V = 8	g = 3 h = 2.4+0.2 f = 5.6 V = 10	g = 4 h = 1.6 + 0.2 f = 5.8 V = 11	g = 5 h = 0.8 + 0.2 f = 6 V = 12	g = 6 h = 0 + 0.2 f = 6.2 V = 14

5. (20 points)

Given $D = \{a, b, c, d, e, f, g\}$ and
$R = \{(a, a), (a, b), (a, d), (b, c), (c, c), (d, b), (d, d), (e, e), (e, f), (e, g), (f, f), (f, g), (g, e), (g, g)\}$

(a) (4 points) Draw a graph of R:

(b) (2 points) We can make the relation reflexive over D by adding one pair. What is it?

(b,b)

(c) (3 points) We can make the relation antisymmetric by *removing* one pair. What is it?

(e,g) or (g,e)

(d) (3 points) We can make the relation transitive by removing one pair. What is it?

(b,c)

not true… consider (g,e)(e,f)

(e) (8 points) Explain why it makes sense to argue (i) below, but not (ii):

(i) A whale is a mammal. Mammals bear live young. Therefore whales bear live young.

(ii) There are two types of sub-atomic particles. A boson is a sub-atomic particle. Therefore, there are two types of bosons.

Your UW Student Number: _______________

I have neither given nor received assistance during this examination:

(Sign here): __________________

I certify that I received all 8 pages of this test.

(Sign here): __________________