CSE 571 Final Exam Review - Week 7 Module

1. Consider a logistic domain where there are **3 cities, 4 trucks, and 6 packages**. Each truck can be in any of the cities. A package can either be at one of the cities or in one of the trucks. Assuming that a truck can go from any city to any other city, what is the minimum number of states that will be needed to represent this problem (assume we are only interested in the **location of each truck and package**), using **atomic representation**? (Assume a ^ b = a power b / a raised to b)
    a. 3^10
    b. (4^3) * (6 ^ 3)
    c. (3^4) * (7^6)
    d. 10^3

2. Consider a logistic domain where there are **100 cities, 15 trucks, and 20 packages**. Each truck can be at any of the cities. A package can either be at one of the cities or in one of the trucks. Assuming that a truck can go from any city to any other city, what is the minimum number of variables that will be needed to represent this problem (assume we are only interested in the **location of each truck and package**), using **factored representation**? (Assume a ^ b = a power b / a raised to b)
    a. 100
    b. (100 ^ 15) * (115 ^ 20)
    c. 100^35
    d. 35

3. Consider a logistic domain where there are **100 cities, 15 trucks, and 20 packages**. Each truck can be at any of the cities. A package can either be at one of the cities or in one of the trucks. Assuming that a truck can go from any city to any other city, what is the minimum number of variables that will be needed to represent this problem (assume we are only interested in the **location and weight of each truck and package at any given moment in time**), using **factored representation**?

    Your answer: ________

4. **Consider the following PDDL Description:**

```
(:action moveCar
 :parameters(?c - car ?src_loc ?dest_loc - place)
 :precondition(and (car_at ?c ?src_loc) (path ?src_loc ?dest_loc))
 :effect(and (not (car_at ?c ?src_loc))(car_at ?c ?dest_loc))
)

(:action loadBox
 :parameters(?b - box ?c - car ?loc - place)
 :precondition(and (box_at ?b ?loc)(car_at ?c ?loc))
 :effect(and (not (box_at ?b ?loc))(in ?b ?c))
)

(:action unloadBox
 :parameters(?b - box ?c - car ?loc - place)
 :precondition(and (car_at ?c ?loc)(in ?b ?c))
 :effect(and (not (in ?b ?c))(box_at ?b ?loc))
)
```

The Current State is:

```
and ((car_at car_2 place_1)
     (car_at car_1 place_2)
     (box_at box_1 place_2)
     (in box_2 car_2)
     (path place_1 place_2)
     (path place_2 place_1))
```

Based on the given description and current state, which of the following is **NOT** a possible action:

a. moveCar(car_1 place_2 place_1)
b. unloadBox(box_2 car_2 place_2)
c. loadBox(box_1 car_1 place_2)
d. moveCar(car_2 place_1 place_2)

5. Suppose there is a glasses world domain (glasses can be stacked on top of each other) that contains some glasses and a table. A glass can be on top of the table or on another glass. **On** relation specifies which glass is on top of what. **Move** action moves a glass from one location to another. In_Hand relationship specifies that the glass is in the hand. (**Assume that a glass which has other glasses stacked upon it, cannot be moved unless the glasses stacked above it are removed)**

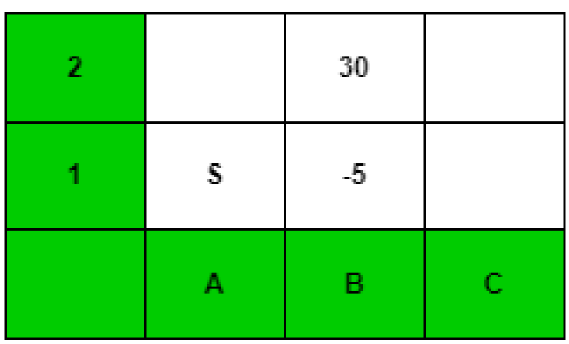
    Assume the current state to be: glass(g1), glass(g2), glass(g3), glass(g4), On(g2, g1), On(g3, g4), On(g1, table), On(g4, table)

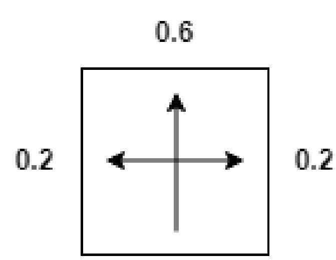
    Goal State: On(g4, g2), On(g1, table)

    For any intermediate state to be a potential landmark, which of the following propositions should be in the state?

    a. In_Hand(g2)
    b. In_Hand(g1)
    c. On(g4, table)
    d. In_Hand(g3)

Instructions for Questions 6 and 7 - Consider the following Markov Decision Process (MDP) - based GridWorld:





(NOTE: Cells in Green just represent Row and Column Numbers - not part of actual grid)

Consider the following:

- Robot at (A, 1) - crossing grid limit causes robot to stay in its original position.
- Terminal Goal States: (B, 1) and (B, 2) with rewards -5 and 30 respectively
- Non-terminal State Reward: 0
- Robot can move UP, DOWN, LEFT, or RIGHT (with a success probability of 0.6, will move perpendicular left/right of given action with a probability 0.2 each)
- Discount Factor, Gamma = 0.9

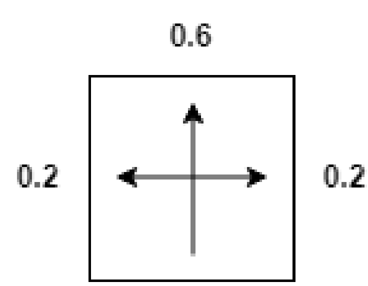
- Assume: $V_0(A, 1) = 0$, $V_0(A, 2) = 0$, $V_0(B, 1) = -5$, $V_0(B, 2) = 30$

6. At time step = 1, what would be the optimal policy for the robot at grid position (A, 1)?
   a. Up
   b. Down
   c. Left
   d. Right

7. At time step = 2, what would be the optimal policy for the robot at grid position (A, 1)?
   a. Up
   b. Down
   c. Left
   d. Right

Instructions for Questions 8 and 9 - Consider the following Markov Decision Process (MDP) - based GridWorld:

2	-2	10	
1		5	
	A	B	C



(NOTE: Cells in Green just represent Row and Column Numbers - not part of actual grid)

Consider the following:

- Crossing the grid limit causes the robot to stay in its original position.
- Terminal Goal States: (B, 1) and (B, 2) with rewards 5 and 10 respectively
- Non-terminal State (A, 2) Reward: -2, rest of Non-Terminal States Reward: 0
- Robot can move UP, DOWN, LEFT, or RIGHT (with a success probability of 0.6, will move perpendicular left/right of given action with a probability 0.2 each)
- Discount Factor, Gamma = 1

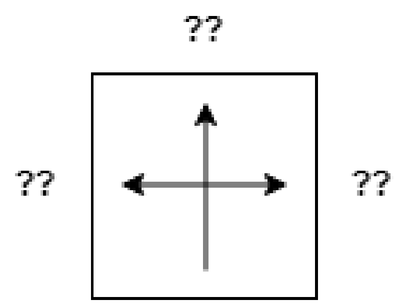
8. Given, $V_1(A, 1) = 3$, $V_1(A, 2) = -4$, $V_1(B, 1) = 5$, $V_1(B, 2) = 10$. What will be the answer of the second round of value iteration for state (A, 1), i.e. $V_2(A, 1)$?

   Your answer: _______

9. Given, $V_1(C, 1) = -2$, $V_1(C, 2) = 0$, $V_1(B, 1) = 5$, $V_1(B, 2) = 10$. What will be the answer of the second round of value iteration for state (C, 2), i.e. $V_2(C, 2)$?
   a. 4.6
   b. 6.2
   c. 5.6
   d. 3.2

Instructions for Questions 10 - Consider the following Markov Decision Process (MDP) - based GridWorld with Reinforcement Learning:

2	-3		6
1	S		
	A	B	C



(NOTE: Cells in Green just represent Row and Column Numbers - not part of actual grid)

Consider the following:

- Robot starts at (A, 1) - Crossing the grid limit causes the robot to stay in its original position.
- Terminal Goal States: (A, 2) and (C, 2) with rewards -3 and 6 respectively
- Non-Terminal States Reward: 0
- Robot can move UP, DOWN, LEFT, or RIGHT
- Discount Factor, Gamma = 1

10. We wish to find a single optimal policy (policy for all states) for the robot and have narrowed it down to two actions: **move right and move up.** We conduct the experiment as follows: for each of the possible policies, we execute 4 trials and calculate its respective Monte Carlo (direct utility) estimate for state (A,1). Given the following trials (or trace) for each policy, **calculate the Monte Carlo estimate - for state (A, 1) for each policy, and decide the optimal policy to be chosen**:

    Policy: **Move right:**

    Trial 1: (A, 1) - (B, 1) - (C, 1) - (C, 2)
    Trial 2: (A, 1) - (B, 1) - (B, 2) - (C, 2)
    Trial 3: (A, 1) - (A, 2)
    Trial 4: (A, 1) - (B, 1) - (B, 2) - (B, 1) - (C, 1) - (C, 2)

    Policy: **Move up:**

    Trial 1: (A, 1) - (A, 2)
    Trial 2: (A, 1) - (B, 1) - (B, 2) - (C, 2)
    Trial 3: (A, 1) - (A, 2)
    Trial 4: (A, 1) - (B, 1) - (B, 2) - (A, 2)

Options arranged as (Monte Carlo Estimate for Move Right action, Monte Carlo Estimate for Move Up action, Optimal Policy)

a. 4.25, -1.25, Move Right
b. 3.75, 1.25, Move Right
c. -1.25, 1.25, Move Up
d. 3.75, -0.75, Move Right