Birla Institute of Technology & Science, Pilani
Work-Integrated Learning Programmes Division
Second Semester 2018-19
Mid Semester Examination(Make-Up)

Course No.	: DSECL ZG557	
Course Title	: Artificial & Computational Intelligence	
Nature of Exam :	Closed Book	
Weightage	: 30%	
Duration	: 1.5 Hours	
Date of Exam	: 18 / 08 / 2019, AN	Time of Exam: 2:00 PM to 3:30 PM

No. of Pages	= 3
No. of Questions	*= 4*

Note:

1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. ***Questions must be answered only in the page numbers mentioned against each question. In case you require more pages to answer a question, you can continue from page #21***
4. Assumptions made if any, should be stated clearly at the beginning of your answer.

Answer all the questions

Question -1 [3+3 = 6M] **[Page 03 - 05]**

(1-a) Explain how a model based agent maintains an internal state, reflecting relevant aspects of the world. What are possible (any two) representations of this internal state?

(1-b) Give a complete problem formulation for the stated problem. Choose a formulation that is precise enough to be implemented.

"A 3-foot-tall monkey is in a room where some bananas are suspended from the 8-foot ceiling. He would like to get the bananas. The room contains two stackable, movable, climbable 3-foot high crates"

Question -2 [4+3+1 = 8 M] **[Page 06 - 11]**

(2-a) Imagine an agent is confronted with a very large state space. Assume that the states-space has on the order of b^d nodes, where b is the average branching factor and d is the average length of a path between two randomly chosen nodes in the state-space graph. We wish to compare two design strategies for designing an agent able to achieve an arbitrary initial state.

(i) First strategy is to plan a path at run time using IDA*, given an initial and goal state.

(ii) The second strategy is to precompute and store all the possible paths, for all goal states and for all the start states.

Compare the strategies (i) and (ii) on time and space complexities. The following table summarizes the aspects expected from your answer.

	Strategy-1	Strategy-2	Two Comments, indicating the relative advantages/ shortcomings
Time Complexity			
Space Complexity			

(2-b) Formally argue the optimality of A* Algorithm. [*Note: Casual/ informal / incorrect statements will be penalized*]

(2-c) What is the impact of zero step cost in A* algorithm. Explain. [Note: Clearly articulate your answer in not more than two unambiguous statements]

Question -3 [2 + 3 + 2 +1 = 8 M] **[<u>Page 12 - 16</u>]**

(3-a) Write any two possible approaches which hill-climbing uses to escape a plateau.

To answer questions (3-b), (3-c) consider the context given below.

Assume your agent uses genetic algorithm for certain optimization task. The description of a chromosome (or state) is of the form

a b c d

in which a, b, c and d can take any value from 0 to 9. The fitness function f(x) is as below:

$$f(x) = (a + b) + (c-d)$$

The initial population x_1, x_2, x_3 and x_4 are as below

x_1 = 2 9 1 3

x_2 = 1 2 5 8

x_3 = 6 2 8 4

x_3 = 1 5 9 7

The agent attempts to find a,b,c and d such that it maximize f(x)

(3-b) Use a simple cross over operation (as discussed in the class) to generate offsprings O_1, O_2, O_3 and O_4 as follows:

Generate O_1, O_2, using first and third fittest individuals (among x_1, x_2, x_3 and x_4)

Generate O_3, O_4, using second and fourth fittest individuals (among x_1, x_2, x_3 and x_4)

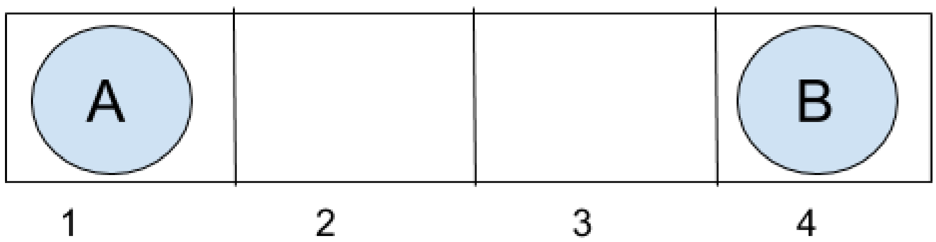
Comment on the overall fitness of O_1, O_2, O_3 and O_4 in comparison to x_1, x_2, x_3 and x_4. Show all the necessary computations.

(3-c) Is it possible to reach the optimal solution without a mutation operation? Why or Why not? [Note: Answer in at most two well articulated statements]

(3-d) Hill climbing with random restarts is guaranteed to find the global optimum. Explain the condition in which this statement is true. [Note: Answer in at most two well articulated statements]

Question -4 [2+2+2+2 = 8 M] **[<u>Page 17 - 20</u>]**

Consider a simple game described below.



The starting position of a simple game is shown in the above picture. It is a two player game (player A and B). Player A moves first. The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1.) The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game to A is +1; if player B reaches space 1 first, then the value of the game to A is −1.

(4-a) Draw the complete game tree, using the following conventions:

- Write each state as (s_A , s_B), where s_A and s_B denote the token locations.
- Put each terminal state in a square box and write its utility value in a circle.
- Put loop/repeated states (states that already appear on the path to the root) in double square boxes. Since their value is unclear, annotate the utility of each of these states with a "?" in a circle.

(4-b) Mark each node with its backed-up minimax value (also in a circle). Explain how you handled the "?" values and why. [no need to show the traces of the algorithm, you can directly use the minimax recurrence here.]

(4-c) Do you see any challenges in applying minimax algorithm to this game tree. Explain.

(4-d) What is the time and space complexity of minimax for deciding the best move for MAX player?