National University
of Computer & Emerging Sciences Peshawar Campus

Student Name: ______________________ Roll No: ___________________

Program: BS (CS-19)
Semester: SPRING – 2022
Time Allowed: 3:00 hours
Course: **Artificial Intelligence (AI-2002)**

Examination: **Final**
Total Marks**: 100** Weightage**: 50**
Date: **24-06-2022**
Instructor: **Dr. Hafeez Ur Rehman**

NOTE: Attempt all questions in the **GIVEN ORDER**. Unordered attempts shall have **negative (-2)** penalty.

Question # 01: **[Marks: 10]**

Give at least **two examples** (only names) of the AI environments that have the following properties:

a) Fully Observable, Stochastic, Sequential, and Static

b) Deterministic, Sequential, and Semi-Dynamic

c) Stochastic, Sequential, Dynamic, and Continuous

d) Stochastic, Sequential, Dynamic and Discrete

e) Fully observable, Unknown, continuous

Question # 02: **[Marks: 10 + 10+10]**

a) Differentiate between informed and uninformed search strategies? Also write *completeness* and *optimality* conditions for Iterative Deepening Search & Uniform Cost search strategies?

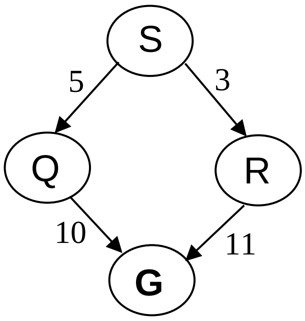
b) Consider an infinite depth tree with branching factor of 50 and depth of the shallowest goal state at 10^{th} level (assuming the worst case). What is Time and Space complexity (in terms of # of nodes) of Depth First Search and Iterative deepening search strategies for this tree (**don't write asymptotic notation but actual number of nodes**)?

c) Draw an **example state space graph** with path costs and heuristic function values in which the heuristic is admissible but A* is not optimal.

Question # 03: **[Marks: 3+3+7+7]**

Consider the following search space (S is the initial state and G is the goal state) with actual path costs *g(n)* along the edges and relative heuristic function *h(n)* values as shown in the table below:

Heuristic *h(n)*	value
h(S)	17
h(Q)	05
h(R)	08
h(G)	00



Now, answer the following questions:

a. Write heuristic admissibility condition. Is *h(n)* admissible?
b. Write heuristic consistency condition. Is *h(n)* consistent?

c. Will **Uniform Cost Search (UCS) algorithm** give an optimal solution for this problem? If yes, give the order of node expansion. If no, then suggest changes (in **search space**) such that **UCS becomes optimal**.

d. Will **A* search algorithm** give an optimal solution using this heuristic? If yes, give the order of node expansion. If no, then suggest changes (in **heuristic or search space**) such that **A* becomes optimal**.

Question # 04: [**Marks: 5+5+10**]

1. In **simulated annealing**, what is the effect of **temperature** and **worst move (ΔE)** on probability of accepting locally bad move? Explain using an example.
2. Explain with an example the difference between Stochastic Hill Climbing, First choice Hill Climbing, Random Restart Hill Climbing?
3. Consider the 5-Queen problem that you would like to solve using Genetic Algorithms. Each queen can only move in its column. The idea is to find a configuration in which no queen attacks the other. A random configuration of the problem is shown below:

[**Marks Distribution: 2+2+4+1+1**]

			Q4	
		Q3		
Q1				**Q5**
	Q2			

In the above context answer the following:

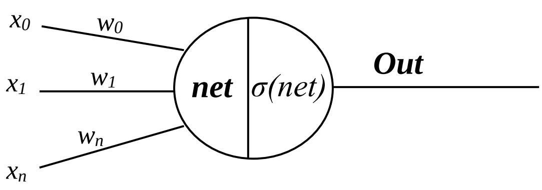
a. How will you turn it into a **minimization** problem? Write objective function.
b. What will be the **best fitness value** that your algorithm will try to achieve?
c. Start with a random population of **two individuals** and list ALL the steps involved using Genetic Algorithm (allowed modification operators are crossover and mutation) in generating the first generation of states?
d. What will happen if the mutation probability is very high i.e., 0.99?
e. What will happen if we do crossover with a low probability i.e., 0.001?

Question # 05: **[Marks: 10 + 10]**

a. Let us consider a perceptron that you want to train for the **NOR function**. Initialize the weights of the perceptron with values (0.1, 0.1, 0.1). You have to train the perceptron to learn the NOR function. **Give values after the first two training iterations** of the perceptron learning algorithm. Assume learning rate η=0.1, and incremental weight updates.

b. **Derive** the formula for ΔW_i using ***gradient descent training rule*** for the following unit with inputs $x_0, x_1, ..., x_n$ (including bias) as well as output ***Out*** which is a **sigmoid** function $\sigma(x)$.

Whereas, $\sigma'(x) = \sigma(x)(1-\sigma(x))$ and ***net*** = $\mathbf{w_0x_0+w_0x_0^2+w_1x_1+w_1x_1^2+.......+\ w_nx_n+w_nx_n^2}$



----- Good Luck! -----