North East University Bangladesh
Department of Computer Science and Engineering
Program: BSc(Engg) in CSE
Mid Semester Examination, Summer - 2022

Course Code: CSE 411 Course Title: Artificial Intelligence
Time: $1\frac{1}{2}$ hours Maximum Marks: 30

Square bracketed numbers in the margin indicate marks for each part of a question.
Boxed numbers in the margin indicate total marks of the question.

(Answer all of the following questions)

Question 1 .... Total Marks on Question 1 is: 9

Answer **any 3 questions** from the following:

(a) Draw the block diagram of an agent's system architecture. [3]

(b) Complete the 2^{nd} column of the following table by chosing any one from the 4 options — "c = complete but not optimal", "o = optimal but not complete", "b = both complete and optimal", or "n = neither complete nor optimal". [Hint: An algorithm is complete if it returns the solution if the solution is in a finite depth of the state space graph. An optimal algorithm returns the least-cost path from the starting state to the goal state.] [3]

Algorithm	Completeness and Optimality
Uniform-cost Search (UCS)	
Iterative Deepening Search (IDS)	
Greedy Best First Search (GBFS)	

(c) **A CSP**: Consider a constraint satisfaction problem with variables $V = \{V_1, V_2, V_3, V_4, V_5, V_6\}$, each variable has the same domain $dom(V_i) = \{1, 2, 3, 4\}$, and there are 10 constraints $c = \{V_1 < V_3, V_1 < V_4, V_3 > V_4, V_2 < V_5, V_2 < V_6, V_5 < V_6, V_3 < 4, V_6 > 2, V_3 \neq V_6, V_1 + V_2 < V_5\}$. [3]

Evaluation Function and Optimization: The evaluation function for this CSP is defined as "the number of conflicts". A conflict means one constraint got violated. The optimization goal for this problem is to minimize the number of conflicts.

Problem: Consider a local search algorithm starts with the following assignment:

$$\langle V_1 = 1, V_2 = 1, V_3 = 3, V_4 = 1, V_5 = 3, V_6 = 1 \rangle$$

The evaluation value for this assignment is 4, since it has 4 conflicts with violated constraints: $V_1 < V_4, V_2 < V_6, V_5 < V_6, V_6 > 2$.

Now your task is to:

(i) Generate a successor so that it has evaluation value **3**, and mention which constraints are violated by this successor

(ii) Generate a successor so that it has evaluation value **1**, and mention which constraints are violated by this successor

[Hint: Remember that, at one step only one variable-value pair can be changed; multple assignment in one successor is not supported in a local search algorithm].

(d) True or False. If false, give the correct answer:

i. "Simple Hill Climbing" is a complete algorithm. [1]

ii. The "Random Sampling" algorithm assigns different random values to each variable in every iteration. [1]

iii. The "Random Walk" algorithm is better than the "Random Sampling" algorithm if the landscape has a complex structure. [1]

Question 2 .... Total Marks on Question 2 is: 12

Problem Description #1: A Search Problem

States, Initial State, and Successors:
Consider a state space where the **start-state** is the number **1** and the successor of any state x returns three states: $3x$, $3x+1$, and $3x+2$.

Goal State, Step-cost:
The goal-states of this state space graph are $\{13, 28, 31\}$. Consider also that the step-cost (cost given in an edge) from any parent-state to any child-state is 1.

Heuristic:
Definition #1: In a tree, the height of a node is defined as the number of edges from that node to its deepest leaf (for example, in the tree $A \to B \to C$ the height of A is 2, the height of B is 1, and the height of C is 0).
Definition #2: The height of a tree is the height of its root.
The Heuristic: A heuristic function has been defined on the given state space as: $h(n) = treeHeight - nodeHeight$, where n is any node, $treeHeight$ is the height of the tree, and $nodeHeight$ is the height of the node n.

The following questions are based on this **problem description #1**.

Answer **question #2.a**, and **any 1** from **question #2.b – question #2.c**, and **any 3** from **question #2.d – question #2.g**.

(a) Draw the state space graph for states **1** to **32**. [3]

(b) Which states (including the goal-state) will be explored by the breadth first search (BFS) algorithm, if applied in this state space? [3]

(c) Which states will remain in the *frontier* when the depth first search (DFS) algorithm explores a goal-state, if successors are added by DFS to its *frontier* in decreasing order so that among these successors the one gets removed earlier from the *frontier* is the smallest one? [3]

(d) In which order the successors of a state needs to be inserted to the frontier of the depth first search (DFS) algorithm, if both DFS and breadth first search (BFS) algorithms need to return the same goal? [2]

(e) What will be the heuristic value of nodes: **4**, **9**, **11**, and **28**? [2]

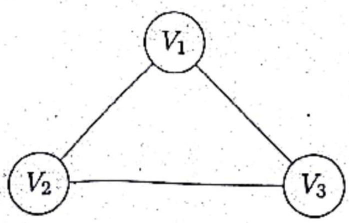
(f) Do you think the given heuristic is admissible? If yes, why admissible? If no, why not admissible? [2]

(g) Using the heuristics given, if A* search explores the state space, which goal will the A* search algorithm find? [2]

Question 3 .... Total Marks on Question 3 is: 9

Problem Description #2: A CSP

The Constraint Network:
Consider you are given the following constraint network:



CSP Formulation:
This CSP has 3 variables $\{V_1, V_2, V_3\}$. The domain for each variable is given as: $dom(V_1) = \{R, G, B\}$, $dom(V_2) = \{R, G\}$, and $dom(V_3) = \{G\}$ where $R = red, G = green, B = blue$ colors. Each edge of the constraint network represents a binary constraint between connecting variables. If two variables V_i and V_j are connected by an edge in the network, then

the constraint is $color(V_i) \neq color(V_j)$.

The following questions are based on this **problem description #2. Answer question #3.a** and **any 3 from question #3.b – question #3.e.**

(a) Write all of the arcs created for the *to_do* list by the "arc consistency" algorithm. [3]

(b) Consider at certain stage of applying the "arc consistency" algorithm to this constraint network, the *to_do* becomes:

$to\_do = \{ \langle V_3, color(V_1) \neq color(V_3) \rangle, \langle V_2, color(V_2) \neq color(V_3) \rangle, \langle V_3, color(V_2) \neq color(V_3) \rangle \}$

During this stage, also consider that the domain of all three variables are: $dom(V_1) = \{R, B\}$, $dom(V_2) = \{R, G\}$, and $dom(V_3) = \{G\}$.
Now answer the following questions:

i. If the first arc from the *to_do* list is removed, does any of the variable gets its domain updated? If yes, which variable and what is the updated domain. [1]

ii. For the same case, is it needed to insert any arc back to the *to_do* list? If yes, which arcs gets inserted? [1]

(c) Consider at certain stage of applying the "arc consistency" algorithm to this constraint network, the *to_do* becomes:

$to\_do = \{ \langle V_1, color(V_1) \neq color(V_2) \rangle, \langle V_3, color(V_2) \neq color(V_3) \rangle \}$

During this stage, also consider that the domain of all three variables are: $dom(V_1) = \{R, B\}$, $dom(V_2) = \{R\}$, and $dom(V_3) = \{G\}$.
Now answer the following questions:

i. If the first arc from the *to_do* list is removed, does any of the variable gets its domain updated? If yes, which variable and what is the updated domain. [1]

ii. For the same case, is it needed to insert any arc back to the *to_do* list? If yes, which arcs gets inserted? [1]

(d) True or False. If false, give the correct answer. "For this constraint network, arc consistency alone is not sufficient to find a unique solution, backtracking search will also be needed". [2]

(e) Consider applying only the backtracking search algorithm in this contraint network (no arc consistency or domain splitting will be applied anywhere). If applied independently, which heuristic with the backtracking search will create a better state space — "Backtracking Search with Minimum Remaining Value" or "Backtracking Search with Degree Heuristics". [2]