DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY (ISM), DHANBAD
MID SEMESTER EXAMINATION, MONSOON SESSION 2023-24

Subject: *Artificial Intelligence* **Time:** *02 Hours*
Code: *CSO303* **Total Marks:** *60*

MODEL SOLUTION

1. (a) Are reflex actions (such as flinching from a hot stove) rational? Are these intelligent? Justify your answers. Characterize the environment of a Poker-playing agent. **[1+1+2]**

Solution: Reflex actions, such as flinching from a hot stove, are rational, because it is the right thing to do. However, flinching is not intelligent, because we do it without thinking/understanding. Intelligence requires at least some amount of thought for understanding.

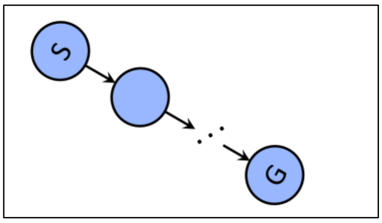
Poker-playing agent environment characteristics: (any of the following four carry total 2 marks)

- Partially observable
- Stochastic
- Sequential
- Static
- Discrete
- Multi-agent

(b) What is a state space? Describe a state space in which iterative deepening search performs much worse than depth first search. **[1+2]**

Solution: A state space is the set of all states reachable from the initial state by any sequence of actions. The state space forms a directed network or graph in which the nodes are states and the links between nodes are actions.

Consider a domain in which every state has a single successor, and there is a single goal at depth n. Then depth-first search will find the goal in n steps, whereas iterative deepening search will take $1 + 2 + 3 + \cdots + n = O(n^2)$ steps.



[Other appropriate examples with adequate descriptions would also be fine.]

(c) Compare and contrast Hill-Climbing and Simulated Annealing. Use *Manhattan distance* as the heuristic to solve the following 8-puzzle problem using steepest ascent Hill-Climbing (moving to best possible state that is one move away). Show the state space and comment on result. **[2+6+1]**

Solution: **[Both similarity and difference are expected here.]**

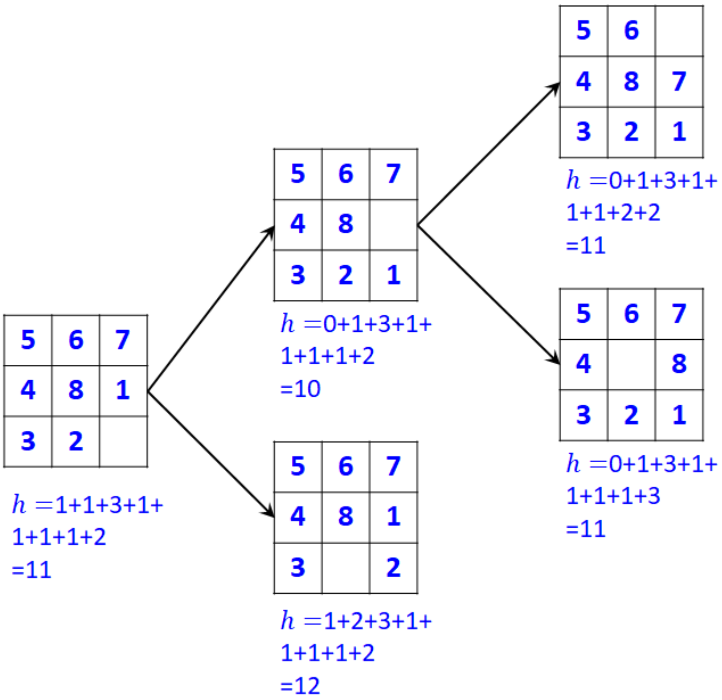
	Hill-Climbing	Simulated Annealing
Comparison (Similarity)	• Local search technique • Memory-efficient	• Local search technique • Memory-efficient
Contrast (Difference)	• Only updates when it finds a better solution • Can easily get stuck in a local optimum and miss the global optimum.	• Has a mechanism to escape from local optimum by accepting worst solutions with a given probability • Has a higher probability of finding the global optimum

Initial State:

5	6	7
4	8	1
3	2	

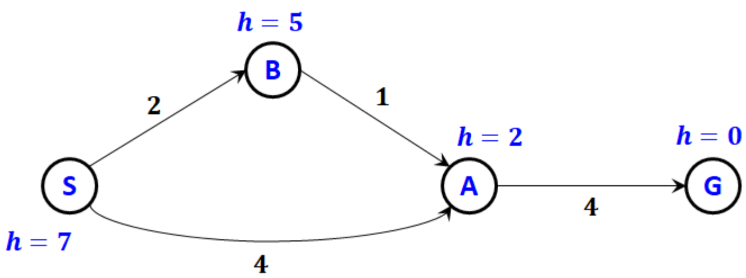
Goal State:

8	7	6
5	4	3
2		1



Stuck at local optima.

2. (a) Devise a state space in which A∗ using GRAPH-SEARCH returns a suboptimal solution with an $h(n)$ function that is admissible but inconsistent. Justify your answer. **[2+2]**



Solution: With respect to the above-depicted state space, the A* with Graph-Search version will work as follows:

Frontier: {S(0+7)}
Explored set: {}

Frontier: {A(4+2), B(2+5)}
Explored set: {S}

Frontier: {B(2+5), G(8+0)}
Explored set: {S, A}

Frontier: {G(8+0)}
Explored set: {S, A, B}

In the next iteration, the Goal state is found.
So, the path returned by A* here is: S→A→G with path cost of 8.
However, the optimal path would be S→B→A→G with path cost 7.
This happens to be so, because here the heuristic function $h(n)$ is admissible but inconsistent.

In the above example, the heuristic function $h(n)$ is **admissible, because** for each state/node n, the $h(n)$ does not over-estimate the actual cost of reaching the goal from that state.
However, $h(n)$ is **inconsistent, because** for the node $n = B$, $h(B) = 5$ and $c(B, A) + h(A) = 1 + 2 = 3 < h(B)$ which violates the condition for consistent heuristic.

[Other appropriate examples with adequate justifications would also be fine.]

(b) Prove that if a heuristic is consistent, it must be admissible. **[5]**

Solution:

We can prove that consistency implies admissibility through induction.

Recall that consistency is defined such that $h(n) \leq c(n, n+1) + h(n+1)$.

Base Case: We begin by considering the $n-1$th node in any path where n denotes the goal state.

$$h(n-1) \leq c(n-1, n) + h(n) \tag{1}$$

Because n is the goal state, by definition, $h(n) = h^*(n)$. Therefore, we can rewrite the above as

$$h(n-1) \leq c(n-1, n) + h^*(n)$$

and given that $c(n-1, n) + h^*(n) = h^*(n-1)$, we can see:

$$h(n-1) \leq h^*(n-1)$$

which is the definition of admissibility!

Inductive Step: To see if this is always the case, we consider the $n-2$nd node in any of the paths we considered above (e.g. where there is precisely one node between it and the goal state). The cost to get from this node to the goal state can be written as

$$h(n-2) \leq c(n-2, n-1) + h(n-1)$$

From our base case above, we know that

$$h(n-2) \leq c(n-2, n-1) + h(n-1) \leq c(n-2, n-1) + h^*(n-1)$$

$$h(n-2) \leq c(n-2, n-1) + h^*(n-1)$$

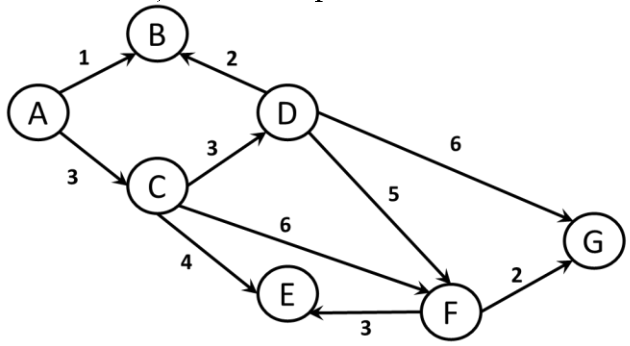
And again, we know that $c(n-2, n-1) + h^*(n-1) = h^*(n-2)$, so we can see:

$$h(n-2) \leq h^*(n-2)$$

By the inductive hypothesis, this holds for all nodes, proving that consistency does imply admissibility!

[Complete and correct proofs other than using induction approach would also be fine.]

(c) Assume that the heuristic function value is 0 for all the nodes in the following graph. Now, apply A* search (graph-search version) to find the path to reach node G from node A.



Show the status of fringe/frontier and closed/explored sets in each step. Which path is returned as the solution? Can you name another search strategy, which will work exactly in a similar way as in the present case? **[5+1+1]**

Solution:

Frontier: {A(0+0)}
Explored set: {}

Frontier: {B(1+0), C(3+0)}
Explored set: {A}

Frontier: {C(3+0)}
Explored set: {A, B}

Frontier: {D(6+0), E(7+0), F(9+0)}
Explored set: {A, B, C}

Frontier: {E(7+0), F(9+0), G(12+0)}
Explored set: {A, B, C, D}

Frontier: {F(9+0), G(12+0)}
Explored set: {A, B, C, D, E}

Frontier: {G(11+0)}
Explored set: {A, B, C, D, E, F}

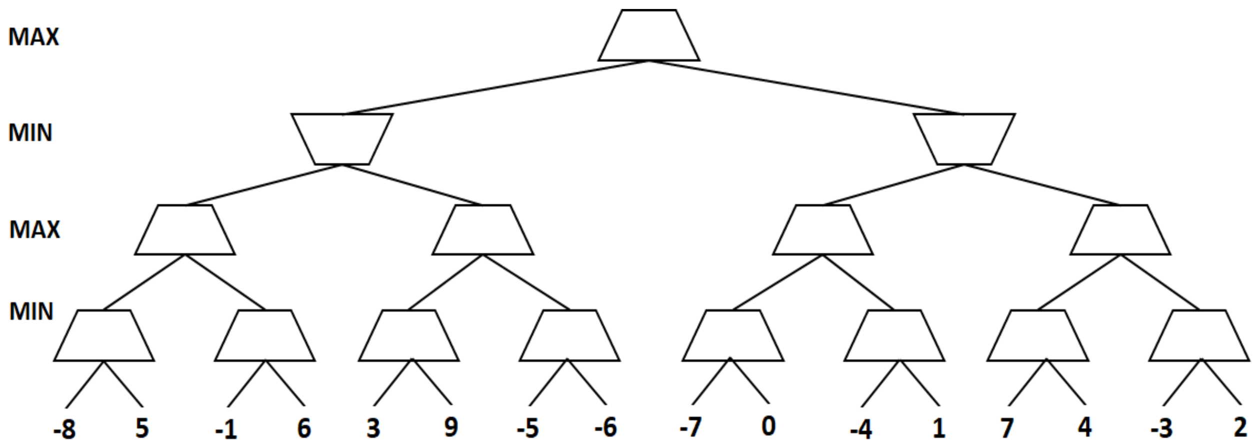
Solution path returned: A→ C→ F→ G (cost 11)

This is equivalent to the **Uniform Cost Search**.

3. (a) Define MINIMAX function as used for optimal decision making in games. Comment on the time complexity and space complexity of minimax algorithm. **[2+2]**

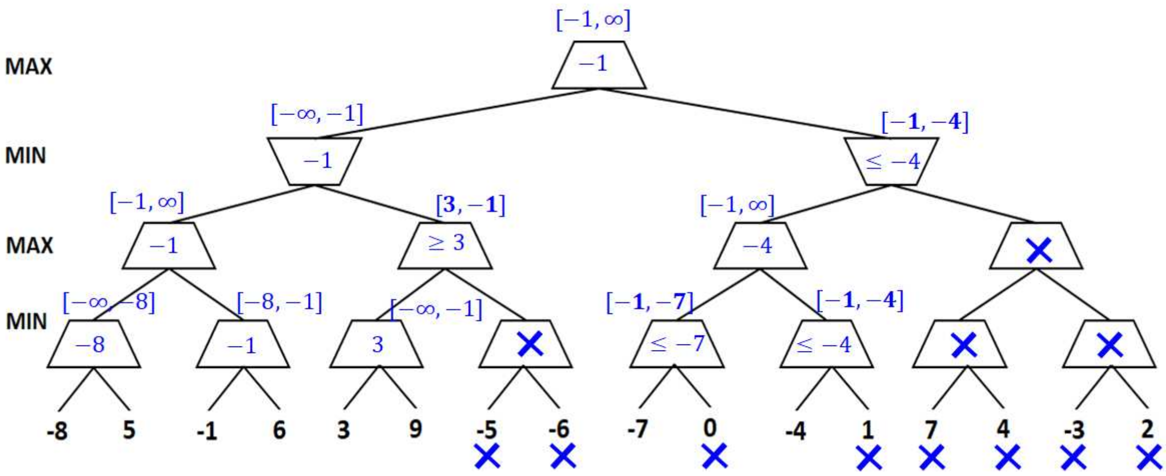
$$\text{MINIMAX}(s) = \begin{cases} \text{UTILITY}(s) & \text{if TERMINAL-TEST}(s) \\ \max_{a \in Actions(s)} \text{MINIMAX}(\text{RESULT}(s,a)) & \text{if PLAYER}(s) = \text{MAX} \\ \min_{a \in Actions(s)} \text{MINIMAX}(\text{RESULT}(s,a)) & \text{if PLAYER}(s) = \text{MIN} \end{cases}$$

The minimax algorithm performs a complete depth-first exploration of the game tree. If the maximum depth of the tree is m and there are b legal moves at each point, then the time complexity of the minimax algorithm is $O(b^m)$. The space complexity is $O(bm)$ for an algorithm that generates all actions at once, or $O(m)$ for an algorithm that generates actions one at a time.



(b) If we apply left-to-right alpha-beta pruning on the above game tree, then which of the leaf nodes would never be visited? Show the alpha and beta values for each node in the tree as you obtain in this process. **[4+3]**

The leaf nodes which would not be visited due to alpha-beta pruning are:
-5, -6, 0, 1, 7, 4, -3, 2



(c) Write the major steps of Genetic Algorithm (GA). Given the following two parent chromosomes corresponding to a Travelling Salesman Problem with 8 cities, what would be the children after applying order-based crossover considering 2nd , 4th and 7th position? **[3+2]**

P1:	3	1	2	7	5	8	4	6
P2:	1	4	8	2	6	7	3	5

Step-1: Start with a randomly generated population of individuals.
The population size is n.
Step-2: Calculate the fitness of each individual in the population.
Step-3: Repeat the following steps until n children or offspring are created.
- Select pair of parents from the population
- With probability P_c perform crossover on these selected parents to generate two offspring
- Mutate offspring with probability P_m

Step-4: Replace the old population with this new population
Step-5: Go to step 2 until the termination condition is reached.

After applying order-based crossover considering 2nd, 4th, and 7th position, the children will be as follow:

C1:	1	7	8	2	6	4	3	5
C2:	4	1	2	7	5	8	3	6

4. (a) Explain why it is a good heuristic to choose the variable that is most constrained but the value that is least constraining, in a CSP search. **[2]**

It is a good heuristic to choose the most constrained variable because it is the variable that is ***most likely to cause a failure***, and it is more efficient to fail as early as possible (thereby pruning large parts of the search space).
The least constraining value heuristic is also good because it allows ***the most chances for future assignments*** to avoid conflict.

(b) What is the worst-case time complexity of running Arc Consistency (AC-3) on a tree-structured CSP and why? **[2]**

On a tree-structured graph, no arc will be considered more than once, so the worst-case time complexity of AC-3 algorithm is $O(nd^2)$, where n is the number of nodes and d is the size of the largest domain.
[Any tree with n nodes has $n-1$ arcs, so we can make it directed arc-consistent in $O(n)$ steps, each of which must compare up to d possible domain values for two variables, for a total time of $O(nd^2)$.]

(c) Solve the following cryptarithmetic puzzle using constraint satisfaction approach. **[8]**

BASE + BALL = GAMES

The constraints are follows:

(i) There should be a unique digit to be replaced with a unique alphabet.

(ii) The result should satisfy the predefined arithmetic rules, i.e., 2+2 =4, nothing else.

(iii) Digits should be from 0-9 only.

(iv) No leading zeroes are allowed.

Show the solution steps in detail.

Formally, the constraints can be expressed as follows:

$E + L = S + 10.\ X1$

$S + L + X1 = E + 10.\ X2$

$A + A + X2 = M + 10.\ X3$

$B + B + X3 = A + 10.\ X4$

$X4 = G$

$G \neq 0,\ B \neq 0$

$Alldiff(A, B, S, L, E, G, M)$

X1	X2	X3	X4	G	A	B	E	L	M	S
{0, 1}	{0, 1}	{0, 1}	{1}	{1}	{0,1, 2,3 4, 5, 6, 7,8, 9}	{2,3, 4, 5, 6, 7,8, 9}	{2,3, 4, 5, 6, 7,8, 9}	{1, 2,3, 4, 5, 6, 7,8, 9}	{1, 2,3, 4, 5, 6, 7,8, 9}	{1, 2, 3, 4, 5, 6, 7,8, 9}
{0, 1}	{0, 1}	{0, 1}	1	1	{0,2,3 4, 5, 6, 7,8, 9}	{5, 6, 7,8, 9}	{0, 2,3, 4, 5, 6, 7,8, 9}	{0,2,3, 4, 5, 6, 7,8, 9}	{0, 2, 3, 4, 5, 6, 7,8, 9}	{0,2, 3, 4, 5, 6, 7,8, 9}
{0, 1}	{0, 1}	0	1	1	{0,2,4}	{5, 6, 7}	{0,2,3, 4, 5, 6, 7,8, 9}	{0,2,3, 4, 5, 6, 7,8, 9}	{0, 4,5,8,9}	{0,2, 3, 4, 5, 6, 7,8, 9}
{0}	1	0	1	1	{2,4}	{6, 7}	{0,2,3, 4}	{5}	{5,9}	{5,7,8, 9}
0	1	0	1	1	{4}	{7}	{2,3, 4}	5	{9}	{7,8, 9}
0	1	0	1	1	4	7	{3}	5	9	{8}
0	1	0	1	1	4	7	3	5	9	8

Hence,

A = 4

B = 7

E = 3

G = 1

L = 5

M = 9

S = 8

7483+7455=14938

*** * ***