Question Bank

PAGE NO.:
DATE: / /

i] Illustrate task domains of an AI system. Give example for each domain.

Ans: i] Artificial Intelligence (AI) means making machines or computers so they can think, learn & solve problems like humans.

ii] It allows system to do tasks such as understanding speech, recognizing images & etc.

iii] AI Systems can be categorized based on the type of tasks or problems they are designed to solve.

iv] These task domains show the areas where AI technologies are applied.

a] Problem Solving & Reasoning: AI system use logical reasoning & algorithms to solve well-defined problem or puzzles.
They analyze the problem, generate solutions and give best answer.
Ex: Chess Playing game.
Expert systems in medical diagnosis.

b] Natural Language Processing (NLP): This domain involves understanding, interpreting & generate human language so machines can communicate naturally.
Ex: Siri, Alexa, Google Assistant.
Language translation tools such as Google Translate.

Teacher's Signature

c] Computer Vision & AI systems analyze & interpret visual data from the world (images or videos) to recognize objects, faces, activities or scenes.
Ex: Face recognition.

d] Robotics & AI powered robots perceive their environment, plan actions & perform physical task automatically.
Ex: Industrial robots assembling cars.
Autonomous drones.

2] Differentiate betn toy problems & ~~wet~~ real world problems in AI problem representation.

Toy Problems	Real-World problems
i] Small, simple & easy to solve problems.	i] Big, complex, & practical problems.
ii] Use small, clean & complete data.	ii] Use large, messy, & incomplete data.
iii] Work in a controlled environment.	iii] Work in uncertain, changing environments.
iv] Ex: Puzzle solving, tic-tac-toe, 8-puzzle.	iv] Ex: Self driving cars, medical diagnosis.
v] Simple rules & limited choices.	v] Complicated rules & many choices.
vi] Can be solved quickly with less computing power	vi] Need more time, memory, & powerful computers.
vii] Approximate or partial solution is acceptable.	vii] Need high accuracy

Teacher's Signature

viii] Do not need to handle huge data.	viii] Must handle massive data & users.
ix] No real risk or harm if the solution fails.	ix] Failure can cause accidents, financial lose.
x] Mainly used for learning, testing and experimenting.	x] Used to solve actual, practical human problem

3] Define AI & explain it's main goals.

Ans: i] Artificial Intelligence is a branch of Computer Science that aims to create machines or computer programs.

ii] Is capable of performing tasks that normally require human intelligence.

iii] It is used for learning, reasoning, problem-solving, understanding language.

iv] The term AI was first coined by John McCarthy in 1956.

v] And today it is applied in various field like robotics, healthcare, finance, education, etc.

vi] ~~Applied~~ Approach : a] Acting Humanly
b] Think Humanly

vii] Main Goal : a] Automation of Tasks
b] Simulate Human Intelligence
c] Problem - Solving
d] Natural-Language Processing

4] Discuss Breadth-First Search (BFS) with it's algorithm & complexity.

Ans 4) i] BFS is a graph / tree traversal technique

ii] It explores all neighbours (same level) first before moving to the next level.

iii] It uses a queue [FIFO] to keep track of nodes to visit.

iv] To avoid revisiting, each node is marked as visited once it is processed.

Algorithm :- a] Start with the root / start node & mark it as visited.

b] Put (enqueue) it into a queue.

c] While the queue is not empty :-

- Remove (dequeue) the front node.
- Visit all it's unvisited neighbors.
- Mark them as visited & add them to the queue

d] Repeat until all nodes are visited.

Complexity :- Time Complexity :- $O(V+E)$

V = number of vertex, E = number of edges

Space Complexity :- $O(V)$

Teacher's Signature ____________

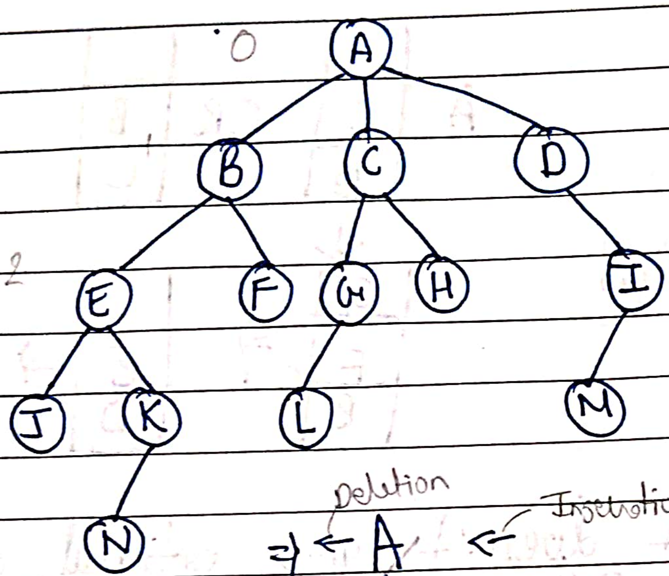
2] Breadth First Search (BFS) :- It is also called blind search

i] Queue [FIFO] is used

ii] Shallowes Node [It is goes level by level].

iv] Complete [It give clear Answer]

Ex :-



Deletion ← A ← Insertion

B C D

CDEF

DEFGH

EFGHI

FGHIJK

GHIJK

HIJKL → IJKL → JKLM →

KLMN → ~~L~~MN

→ ~~M~~N → ~~N~~ − $\boxed{}$

v] Optimal [It give shortest result]

vi] Time Complexity :- $O(V+E)$ OR $O(b^d)$

$\because$ b = branch factor

$G = 3^2 = 9$ d = depth

Teacher's Signature ____________

5] Differentiate betn Hill-Climbing & Simulated Annealing in local search techniques

Ans:

Hill-Climbing	Simulated Annealing
i] Always moves to a better neighbour.	i] Sometimes allows worse move to escape local traps.
ii] Explores only upward	ii] Explores both upward & downward
iii] Can get stuck in local optima.	iii] Can escape local optimal by accepting worse move.
iv] Almost deterministic (depends only on neighbors).	iv] Uses randomness in decision-making.
v] No extra parameter.	v] Uses temperature, which controls acceptance of bad moves
vi] Faster	vi] Slower
vii] Not guaranteed to find the global optimum.	vii] Higher chance of finding global optimum.
viii] Good for simple, smooth search spaces.	viii] Good for complex, rugged search spaces.
ix] Used in optimization, pathfinding, game AI ~~is used~~.	ix] Used in scheduling, circuit design, traveling salesmen problem.
x] Stops when no better neighbor exists.	x] Stops when the temperature reaches zero or after fixed iteration.

Teacher's Signature ____________

Q] Elaborate Depth-First Search (DFS) with it's advantages & disadvantages.

Ans:- DFS is graph / tree traversal method where we go as deep as possible along one path before backtracking to explore other paths.

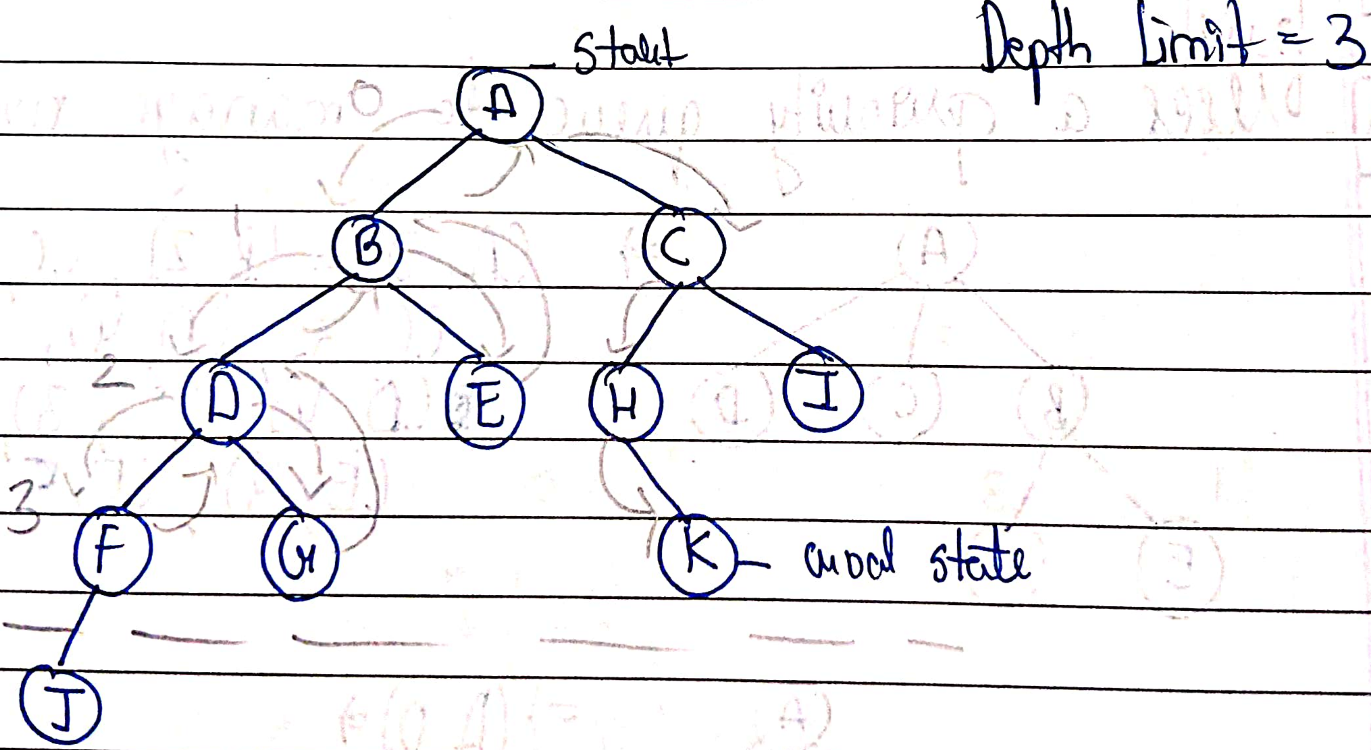
Works:- • Start from a source node
• Visited the first unvisited neighbor and go deeper.
• If no neighbor left, backtrack & explore other branches.
• Continue untill all nodes are visited.

Stack is used [LIFO].
Time Complexity :- $O(V+E)$
Space Complexity :- $O(V)$

Advantages :- Require less memory
Useful for path finding
Can be implemented easily using recursion
Useful in backtracking algorithms.

Disadvantages :- May get stuck in deep path
Does not always find shortest path
Not suitable for real-time problem
Overflow in very deep graph.

4] Depth-Limited Search (DLS) :- DLS is a variant of Depth-First Search (DFS) it imposes a limit on the depth of the search tree.
This limit helps prevent the search from going too deep and potentially getting stuck in an infinite loop. It is particularly useful in problems where the search space is vast or infinite.



Possible 1 :- A→B→D→F→G→E→C→H→K

Possible 2 :- A→C→I→H→K

Possible 3 :- A→C→H→K

7] Explain the Min-Max algorithm with a suitable game tree diagram.

Ans: Min-Max is a decision-making algorithm used in two-player game like Chess & Tic-Tac-Toe) where one player tries to meximize the score & the other tries to minimize it.

Players: Max Player: Tries to get max score
Min Player: Tries to get min score

Working: The algorithm build a game tree of all possible moves.
- It assign values (win, loss, draw) to end states
- Then it backtrack, choosing the best moves that best for current player.

- It optimal for both player.
Min-Max guarantees the best possible result for the player.

Teacher's Signature

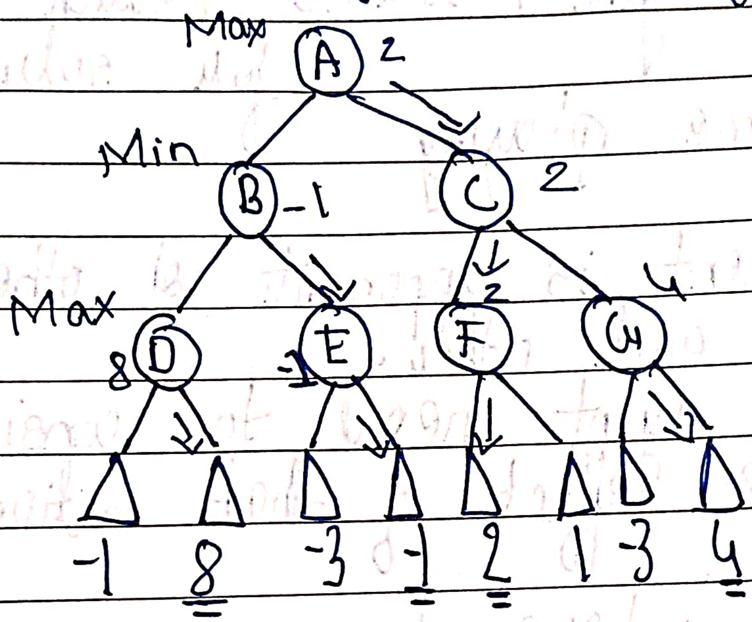
i] Minimax Algorithm :- a] It is a backtracking algorithm

b] Best move strategy used

c] Max will try to maximize its utility (Best move)

d] Min will try to minimize ~~it~~ utility (Worst move)

Ex :-



Game Tree

Time Complexity = $O(b^d)$

DATE : / /

8] Explain Alpha-Beta pruning

Ans:- i] Alpha-Beta Pruning is an optimization technique for the Min-Max algorithm in game playing.
ii] It prunes (cuts off) branches of the game tree that do not need to be explored because they cannot affect the final decision.

Purpose :- Speeds up the Min-Max algorithm.
Reduces the number of nodes evaluated
Gives the same result as Min-Max but faster.

Alpha (α) :- Best value that the Max player can guarantee so far.

Beta (β) :- Best value that the Min player can guarantee so far.

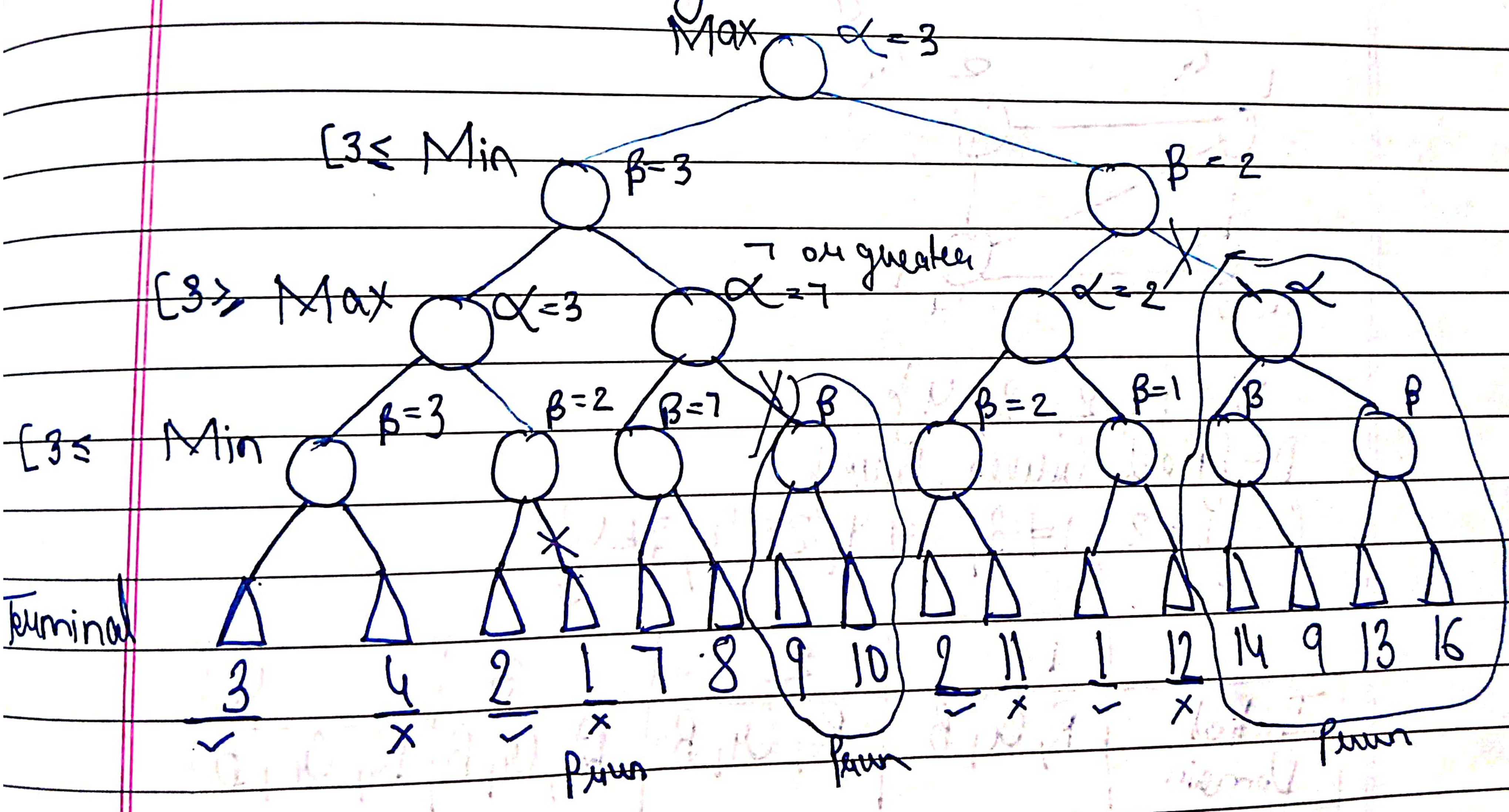
Advantages :- Save times, Allows deeper search

Disadvantages :- Slow in very large games

Ex :- Tic-Tac-Toe.

PAGE NO.:
DATE: / /

ii] Alpha- Beta Pruning (α-β)



Constraint Satisfaction Problem [CSP]

i] CSP consists of three components V, D, C

ii] V is set of variables $\{V_1, V_2, \ldots, V_n\}$

iii] D is set of Domains $\{D_1, D_2, D_3, \ldots D_n\}$ one for each variable.

iv] C is set of constraints that specify allowable combination of values.

$$C_i = (\text{Scope}, \text{relation})$$

Where scope is set of variables that participate in constraint

Rel is relation that define the values that variables can take

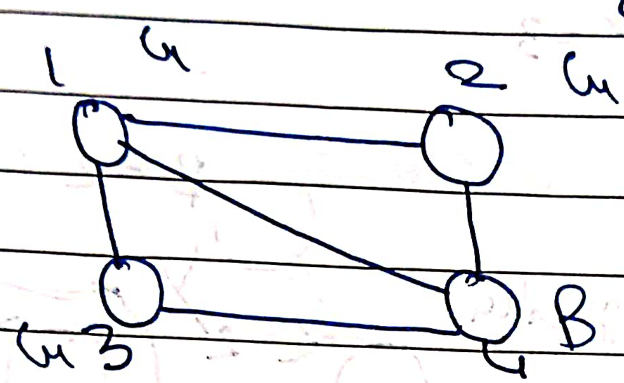
eg:- $C_1 = ((V_1, V_2), (V_1 \neq V_2))$

$C_1 = (V_1, V_2)\ (A, B)$

A	B
(1, 2)	(2, 4)

$C_1 = (V_1, V_2)\ (1, 2)\ (1, 4)\ (2, 4))$

CSP + Backtracking Algorithm



$V = \{1, 2, 3, 4\}$

$D = \{Red, Green, Blue\}$

$C = \{1 \neq 2, 1 \neq 3, 1 \neq 4, 2 \neq 4, 3 \neq 4\}$

	1	2	3	4
Initial Domain	R, G, B	R, G, B	R, G, B	R, G, B
1 = R	R	G B	G B	G B
2 = G	R	G	G B	B
3 = B	R	G	B	—
~~G~~	R	G	G	B

→ Backtrack