Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
First / 2024-2025

Comprehensive Test
(EC-3 Makeup)

Course No.	:	AIMLCZG557
Course Title	:	Artificial and Computational Intelligence
Nature of Exam	:	Open Book
Weightage	:	40%
Duration	:	
Date of Exam	:	

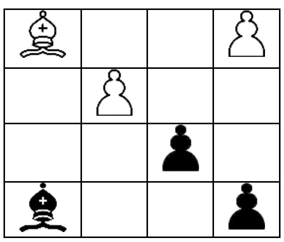
No. of Pages = 4
No. of Questions = 5

Note to Students:

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. Assumptions made if any, should be stated clearly at the beginning of your answer.

Q1 Consider the below chess game problem and answer the following questions:

[3+5+2 = 10 Marks]



- Bishops in chess can move only diagonally. This action is allowed only when making an attacking move and each successful attack earns a reward of 4 points.
- A Soldier moves only one square forward on each move. Only if opponent coins are available, it can move in diagonally one-step forward to attack the opponent coin. Every non-attack move adds a penalty of 3 and every successful attack gains a reward of 5.

a) Assume that player with White colored pieces moves first. Construct the game tree (with neat diagram) from the given current state (0th level) as Start Node up to exactly 2 levels only (1^{st} & 2nd Level - ie., one round for each of the player.)

b) Calculate the utility of the leaves of the tree with below static evaluation function.

Board value = Utility of MAX player – Utility of MIN player

Utility of a player = (Sum of Reward w.r.t all pieces – Sum of Penalty w.r.t all Pieces) + Number of possible Moves w.r.t all pieces

Note: A sample utility calculation for the Max player is provided based on the initial configuration,

White Pieces	Possible Moves	Reward	Penalty

Bishop	0	-	-
Soldier1	2 (1 Attack & 1 Non Attack)	5	3
Soldier2	1 Non Attack	-	3
Total	***3***	***5***	***6***

Utility of a MAX (White) player = (Sum of Reward w.r.t all pieces – Sum of Penalty w.r.t all pieces) + Number of possible Moves w.r.t all pieces.

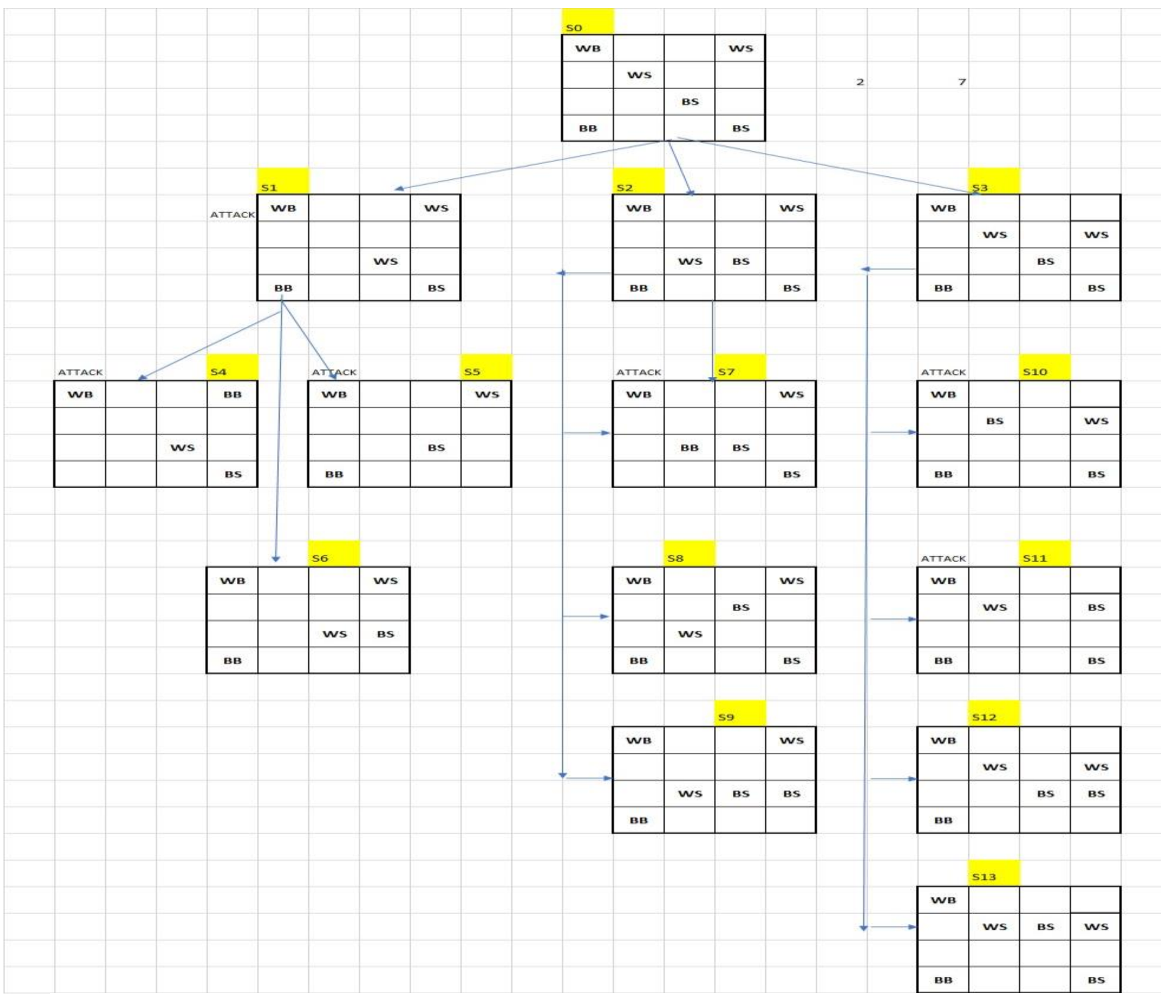
= *(5-6)* + *3* = -1+3 ➔ 2

c) Apply the MIN MAX algorithm on the game tree constructed in part a) using static evaluation values calculated in the part b) and highlight the best path chosen by players in the game given.

Sample solution and marking scheme

a. Constructing the complete game tree – 3 Marks

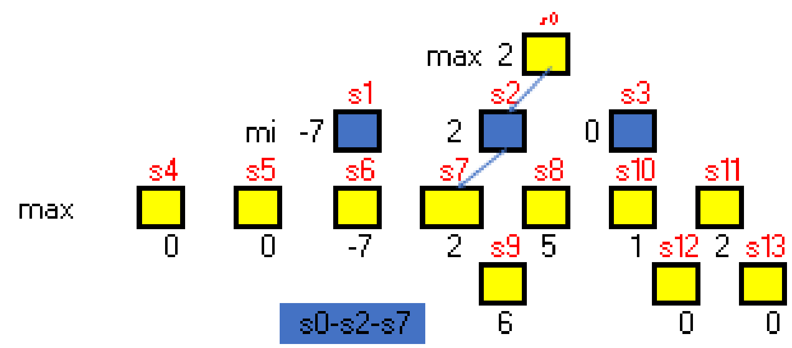
- Search Tree for level 1 expansion – 1 mark
- Search Tree for level 2 expansion – 2 mark



b. Calculating static evaluation functions – 5 Marks

	WH				BL				
	PM	RE	PE	MAX	PM	RE	PE	MIN	BORAD V.
S4	2	5	3	4	2	5	3	4	0
S5	2	4	3	3	2	4	3	3	0
S6	2	0	6	-4	2	4	3	3	-7
S7	2	4	3	3	3	4	6	1	2
S8	5	13	6	12	4	9	6	7	5
S9	4	9	6	7	3	4	6	1	6
S10	2	4	3	3	3	5	6	2	1
S11	1	0	3	-2	2	0	6	-4	2
S12	4	15	3	16	4	15	6	13	3
S13	2	0	6	-4	2	0	6	-4	0

c. MINMAX ALOGRITHM GAME TREE – 1 Mark
Best path chosen – 1 Mark



Q2 In a smart home system, devices interact based on automation rules. Smart lights turn on automatically when motion is detected in a room. Not all motion detectors trigger security alarms, but all security alarms are triggered by motion detectors. Smart devices that are connected to a central hub can be controlled remotely. Any device that can be controlled remotely receives firmware updates automatically.

[2+4 = 6 Marks]

a) List all potential propositional variables and use propositional logic (without quantifiers) to efficiently represent the knowledge base given above. (Represent any 4 premises)

b) Based on the knowledge base from part (a), convert it into **Conjunctive Normal Form (CNF)** and find one sample complete **Binary Satisfiability (BSAT)** solution to the variables using the **DPLL algorithm**.

Marking Scheme

knowledge base (KB) **[2 Marks]**

R1: M → L

R2: D1 ∧ ¬A

R3: A → D1

R4: C → R

R5: R → F

Conjunctive Normal Form **[2 Marks]**

Rule name should be specified

R1: ¬M ∨ L

R2: D1

R3: ¬A

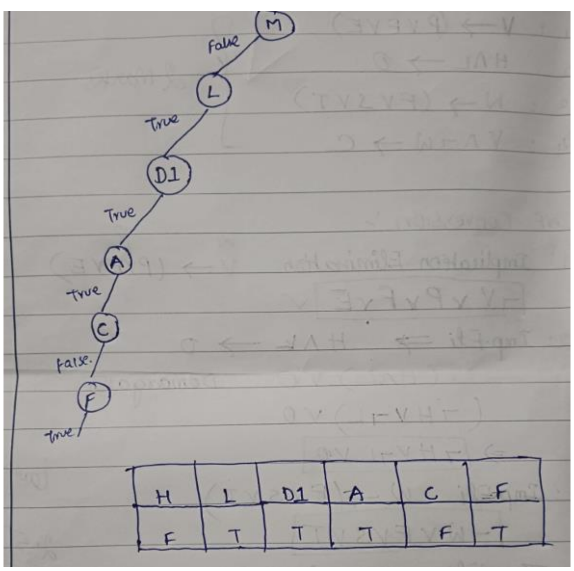
R4: ¬A ∨ D1

R5: ¬C ∨ R

R6: ¬R ∨ F

DPLL Algorithm for BSAT Solution **[2 Marks]**

Pure symbols are: M,L,D1,A,C,F (Need to be specified)

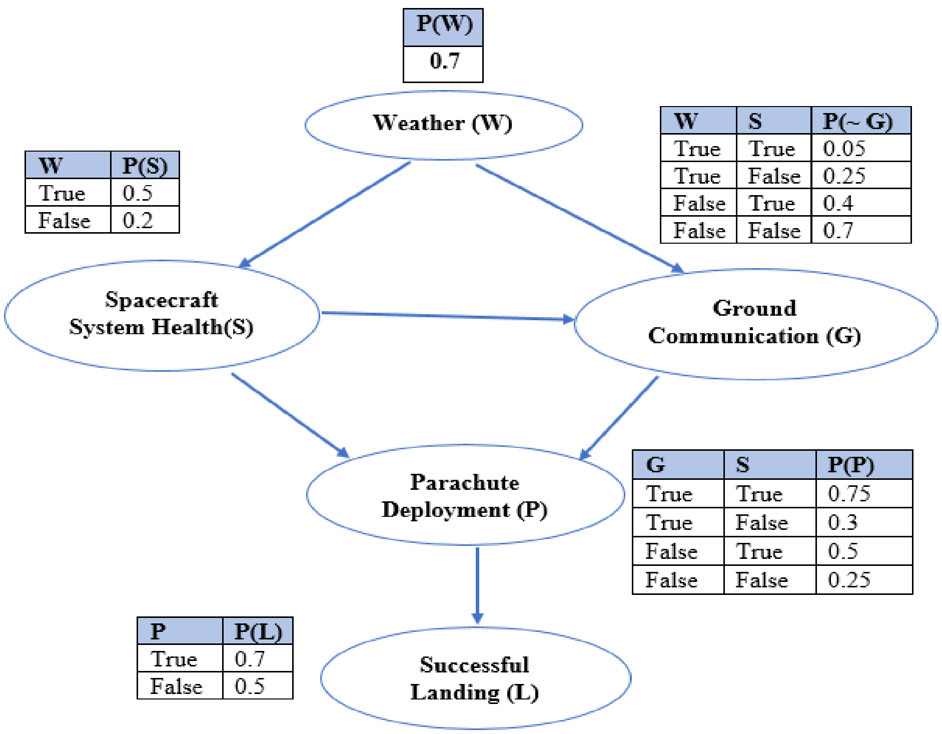


Q3 **Consider the below Bayesian Network and answer the following questions:**

[5+ 4 = 9 Marks]

Consider the random variables embedded & highlighted in below diagram as significant events, takes only binary values “true” or “false”

- **W**: Weather Conditions (Favorable/Unfavorable)
- **S**: Spacecraft System Health (Functional/Malfunctioning)
- **G**: Ground Communication (Established/Lost)
- **P**: Parachute Deployment (Successful/Failed)
- **L**: Successful Landing (Yes/No)



P(W)
0.7

W	P(S)
True	0.5
False	0.2

W	S	P(~ G)
True	True	0.05
True	False	0.25
False	True	0.4
False	False	0.7

G	S	P(P)
True	True	0.75
True	False	0.3
False	True	0.5
False	False	0.25

P	P(L)
True	0.7
False	0.5

a) What is the probability that **Sunita Williams’ spacecraft successfully landed** given the evidence that the **parachute deployed successfully** but the **ground communication was lost**

b) Apply **approximate inference** for the part (a) query using Use likelihood weighing. Generate only **5 samples** from the provided random numbers:

Random numbers:

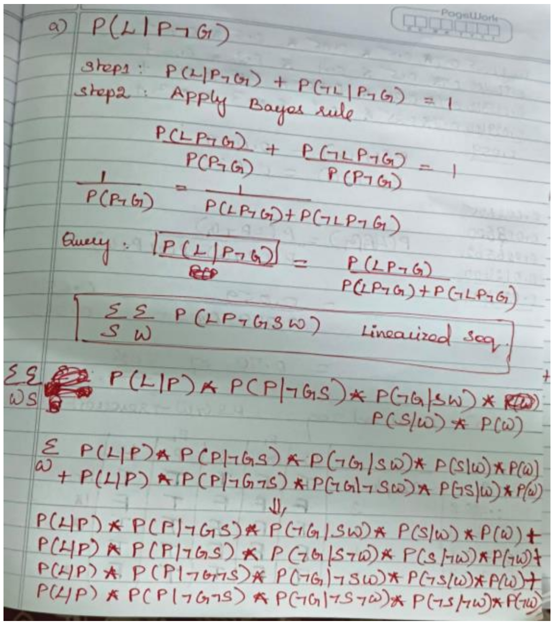
0.7, 0.3, 0.5, 0.9, 0.45, 0.8, 0.12, 0.6, 0.4, 0.75, 0.25, 0.5, 0.36, 0.3, 0.8

Marking Scheme :

Marking Scheme and Sample Solution:

Part a)

- Correct identification of query – 0.5M.
- Applying Bayes Rule – 0.5M.
- Identification of hidden variable and linearization – 0.5
- Calculation of P(L P ¬G) – 1.5M.
- Calculation of P(¬L P ¬G) – 1 M.
- Final calculation and answer – 1 M.



a) $P(L \mid P \neg G)$

step1 : $P(L \mid P \neg G) + P(\neg L \mid P \neg G) = 1$

step2 : Apply Bayes rule

$$\frac{P(L P \neg G)}{P(P \neg G)} + \frac{P(\neg L P \neg G)}{P(P \neg G)} = 1$$

$$\frac{1}{P(P \neg G)} = \frac{1}{P(L P \neg G) + P(\neg L P \neg G)}$$

Query : $\boxed{P(L \mid P \neg G)} = \frac{P(L P \neg G)}{P(L P \neg G) + P(\neg L P \neg G)}$

$\boxed{\sum_S \sum_W P(L P \neg G S W) \quad \text{Linearized seq.}}$

$$\sum_W \sum_S P(L \mid P) * P(P \mid \neg G S) * P(\neg G \mid S W) * P(S \mid W) * P(W)$$

$$\sum_W P(L \mid P) * P(P \mid \neg G S) * P(\neg G \mid S W) * P(S \mid W) * P(W) + P(L \mid P) * P(P \mid \neg G \neg S) * P(\neg G \mid \neg S W) * P(\neg S \mid W) * P(W)$$

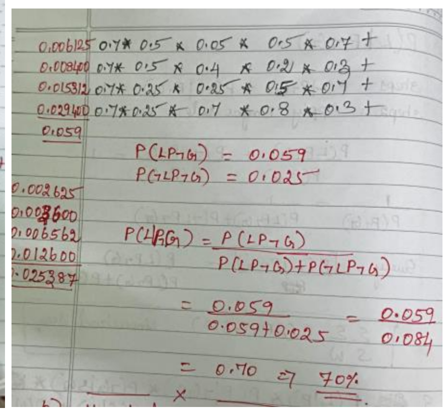
⇓

$P(L \mid P) * P(P \mid \neg G S) * P(\neg G \mid S W) * P(S \mid W) * P(W) +$

$P(L \mid P) * P(P \mid \neg G S) * P(\neg G \mid S \neg W) * P(S \mid \neg W) * P(\neg W) +$

$P(L \mid P) * P(P \mid \neg G \neg S) * P(\neg G \mid \neg S W) * P(\neg S \mid W) * P(W) +$

$P(L \mid P) * P(P \mid \neg G \neg S) * P(\neg G \mid \neg S \neg W) * P(\neg S \mid \neg W) * P(\neg W)$



0.006125 $0.7 * 0.5 * 0.05 * 0.5 * 0.7 +$

0.008400 $0.7 * 0.5 * 0.4 * 0.2 * 0.3 +$

0.015312 $0.7 * 0.25 * 0.25 * 0.5 * 0.7 +$

0.029400 $0.7 * 0.25 * 0.7 * 0.8 * 0.3 +$

0.059

$P(L P \neg G) = 0.059$

$P(\neg L P \neg G) = 0.025$

0.002625

0.003600

0.006562

0.012600

0.025387

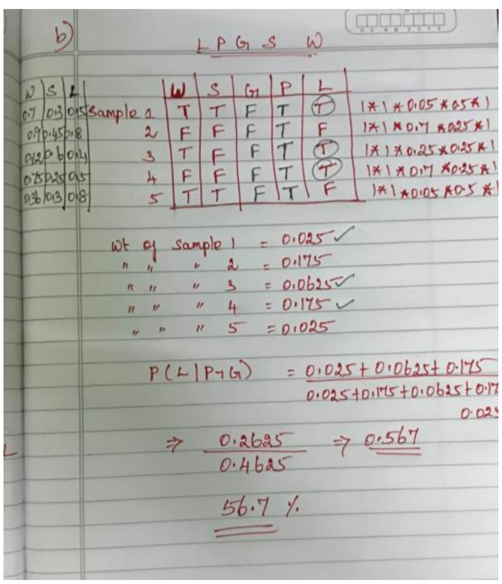
$$P(L \mid P \neg G) = \frac{P(L P \neg G)}{P(L P \neg G) + P(\neg L P \neg G)}$$

$$= \frac{0.059}{0.059 + 0.025} = \frac{0.059}{0.084}$$

$= 0.70 \Rightarrow 70\%$

Part b)

- Sample table with proper calculations – 1.5M
- Final formula & calculations – 1.5M
- Final answer = 1Mark



b) L P G S W

W	S	L
0.7	0.3	0.5
0.9	0.45	0.8
0.2	0.6	0.4
0.75	0.25	0.5
0.36	0.3	0.8

	W	S	G	P	L	
Sample 1	T	T	F	T	T	1 * 1 * 0.05 * 0.5 * 1
2	F	F	F	T	F	1 * 1 * 0.7 * 0.25 * 1
3	T	F	F	T	T	1 * 1 * 0.25 * 0.25 * 1
4	F	F	F	T	T	1 * 1 * 0.7 * 0.25 * 1
5	T	T	F	T	F	1 * 1 * 0.05 * 0.5 * 1

wt of sample 1 = 0.025 ✓

" " " 2 = 0.175

" " " 3 = 0.0625 ✓

" " " 4 = 0.175 ✓

" " " 5 = 0.025

$$P(L \mid P \neg G) = \frac{0.025 + 0.0625 + 0.175}{0.025 + 0.175 + 0.0625 + 0.175 + 0.025}$$

$$\Rightarrow \frac{0.2625}{0.4625} \Rightarrow 0.567$$

56.7 %

Q4	

In Natural Language Processing (NLP), Part-of-Speech (POS) tagging is a crucial task where each word in a sentence is assigned its most likely grammatical category (e.g., noun, verb, adjective). Hidden Markov Models (HMMs) are widely used to perform POS tagging by modeling the sequence of words as states in a probabilistic framework. Consider a simplified English grammar with three possible POS tags:

[2+6 =8 Marks]

- •PRP (Pronoun) (e.g., She, He, It)
- •VB (Verb) (e.g., eats, runs, sings)
- •NN (Noun) (e.g., pasta, dog, book)

Given the following transition probabilities

PRP	VB	NN	(t-1)
?	?	0.5	PRP
0.7	0.1	?	VB
0.3	0.8	0.1	NN

(t)

Emission probabilities

PRP	VB	NN	
0.9	?	0.05	she
?	0.8	?	eats
0.05	0.1	0.9	pasta

Initial probabilities

PRP	VB	NN
0.6	0.3	?

a) Calculate the missing probability values and Construct the Markov Model Diagram.

b) Given the sentence "**She eats pasta**", apply the Viterbi algorithm to determine the most likely sequence of POS tags assigned to these words. Strictly follow the approach discussed in class.

Marking Scheme and solution:

Trans Prob.

	PRP	VB	NN
PRP	0.0	0.1	0.5
VB	0.7	0.1	0.4
NN	0.7	0.8	0.1

Emis. Prob

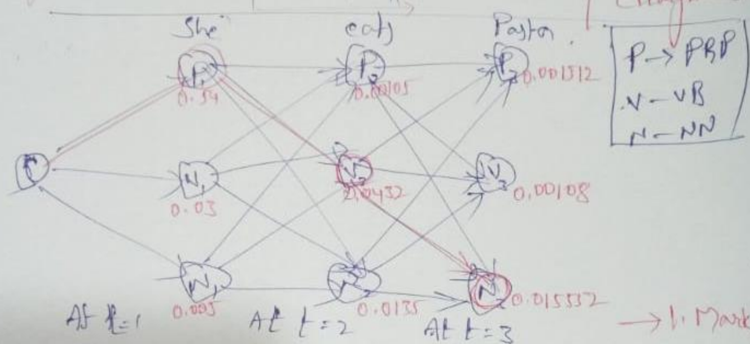
	PRP	VB	NN
She	0.9	0.1	0.05
eats	0.05	0.8	0.05
pasta	0.05	0.1	0.9

Init Prob

PRP	VB	NN
0.6	0.3	0.1

1.5 Marks

0.5 for Markov Model diagram.



→ 1 Mark

$P(P_1) = P(P) \times P(She/P) = 0.6 \times 0.9 = 0.54$

$P(V_1) = P(V) \times P(She/V) = 0.3 \times 0.1 = 0.03$

$P(N_1) = P(N) \times P(She/N) = 0.1 \times 0.05 = 0.005$

1 Mark

$P(P_2) = \max([0.54 \times 0, \underline{0.03 \times 0.7}, 0.005 \times 0.3] \times 0.05)$

$= \max[0, \underline{0.00105}, 0.000075) = 0.00105$

$P(V_2) = \max([\underline{0.54 \times 0.1}, 0.03 \times 0.1, 0.005 \times 0.8] \times 0.8)$

$= \max(\underline{0.0432}, 0.0024, 0.0032) = 0.0432$

$P(N_2) = \max([\underline{0.54 \times 0.5}, 0.03 \times 0.4, 0.005 \times 0.1] \times 0.05)$

$= \max(\underline{0.0135}, 0.0006, 0.000025)$

1.5 Marks

$P(P_3) = \max[[0.00105 \times 0, \underline{0.0432 \times 0.7}, 0.0135 \times 0.3] \times 0.05$

$= \max(0, \underline{0.001512}, 0.0002025) = 0.001512$

1.5 Mark

$P(V_3) = \max([0.00105 \times 0.1, 0.0432 \times 0.1, 0.0135 \times 0.8] \times 0.1)$

$= \max(0.0000105, 0.000432, \underline{0.00108}) = 0.00108$

$P(N_3) = \max([0.00105 \times 0.5, 0.0432 \times 0.4, 0.0135 \times 0.1] \times 0.9)$

$= \max(0.0004725, \underline{0.015552}, 0.001215) = 0.015552$

∴ Most likely sequence is:

PRP → VB → NN — 1 Mark

Q5	**Answer all the following questions. Vague answers will not be awarded marks.**

[2+2+3=7 Marks]

a. Using a numerical example explain A* algorithm guarantees the optimal solution.

b. Genetic algorithm is not the most suitable choice of search algorithms in few use cases". Justify the statement.

c. Explain how Bayesian Networks utilize **conditional probability** and **independence assumptions** to efficiently represent complex probability distributions. Provide an example illustrating its application in real-world scenarios.

a. Using a numerical example explain A* algorithm guarantees the optimal solution.

Marking Scheme and Sample Solution:
i) Numerical example for admissible and consistent property – 1.5 Mark
ii) Justification of Optimal Solution – 0.5 Mark

b. Genetic algorithm is not the most suitable choice of search algorithms in few use cases". Justify the statement.

Marking Scheme and Sample Solution: 2 Mark

1. **Well-Defined Gradient Problems**:
2. **High-Dimensional Problems**:
3. **Precise Solution Requirements**:
4. **Time-Critical Applications**
5. **Lack of Problem-Specific Knowledge Utilization**:

C. Explain how Bayesian Networks utilize conditional probability and independence assumptions to efficiently represent complex probability distributions. Provide an example illustrating its application in real-world scenarios.

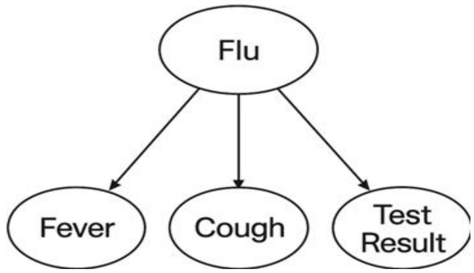
Marking Scheme and Sample Solution:
i) Conditional probability and independence assumptions(1.5 Marks)
ii) Example with explanation (1.5 Mark)

A **Bayesian Network** is a type of diagram that helps us understand how different things (called variables) are related to each other using **probabilities**. It shows which things directly affect others and helps us calculate the chances of something happening, even if we only have partial information.

1. **Conditional Probability**
   - This means we look at the chance of something happening **based on something else**.
   - Example: What's the chance someone has a fever **if** they have the flu?

2. **Independence**
   - A Bayesian network assumes that some things are **not related** once we know certain facts.
   - This helps simplify calculations because we don't have to consider every possible combination of variables.

Example: Medical Diagnosis



How It Helps:

If a patient comes in with a **fever and cough**, but no test result yet, the network helps you calculate: "What's the chance they actually have the flu?"
It combines the known probabilities in a smart way, without needing to check every possible situation.

Why It's Useful

- Saves time and effort by **reducing the number of calculations**.
- Helps in making **informed decisions** under uncertainty.
- Used in real-world problems like **medical diagnosis**, **spam filters**, **weather prediction**, and **fraud detection**.