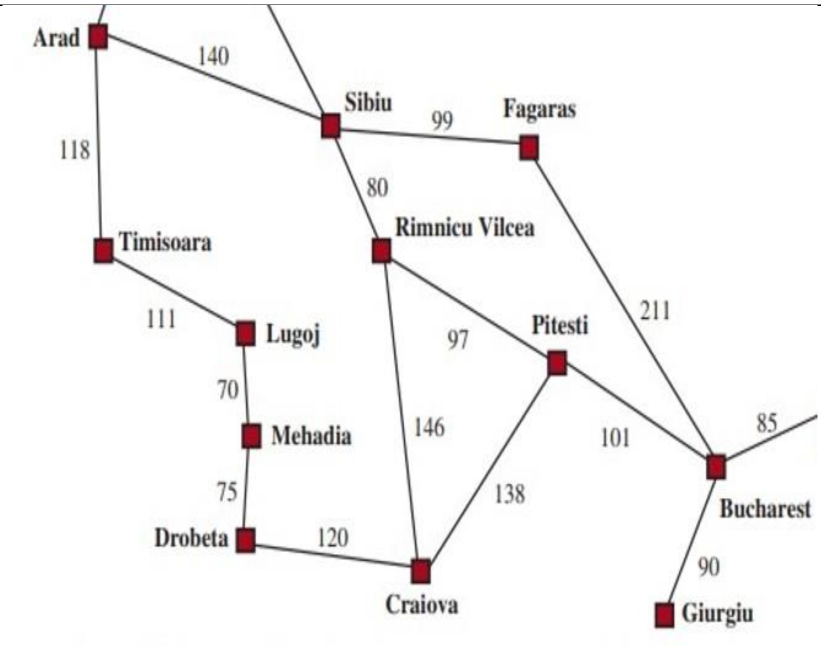
USN | | | | | | | | | | |

Internal Assessment Test 1 – March 2025

	Sub:	Artificial Intelligence and Machine Learning	Sub Code:	BDS602	Branch:	CSDS	

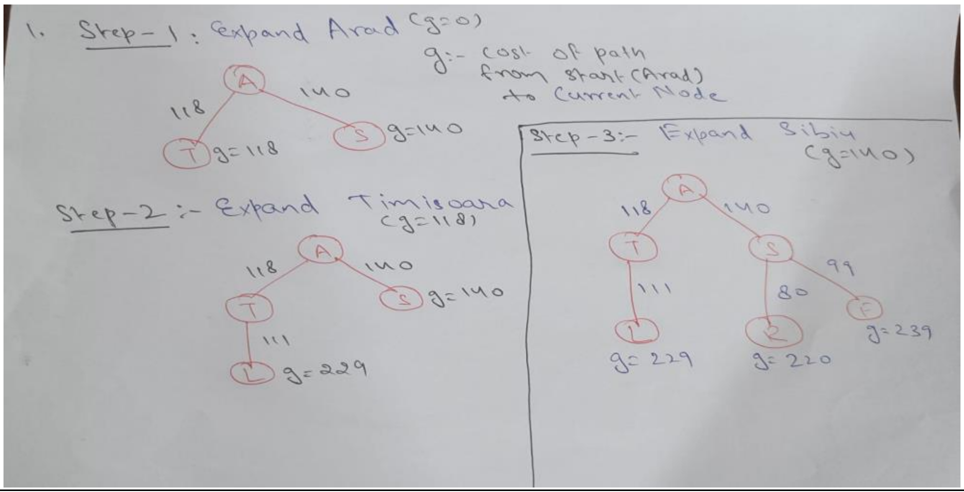
	Date:	25/03/2025	Duration:	90 mins	Max Marks:	50	Sem	VI	

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	Above is the simplified Romania map. Apply UCS Algorithm showing all steps of exploration of Nodes in search space, determining least cost path from Arad to Bucharest	10	CO2	L3



Above is the simplified Romania map. Apply UCS Algorithm showing all steps of exploration of Nodes in search space, determining least cost path from Arad to Bucharest

ANSWER :

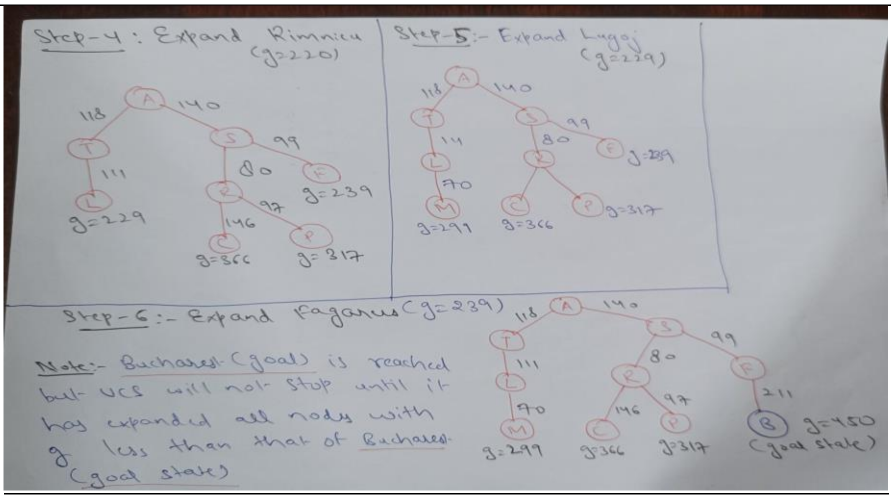
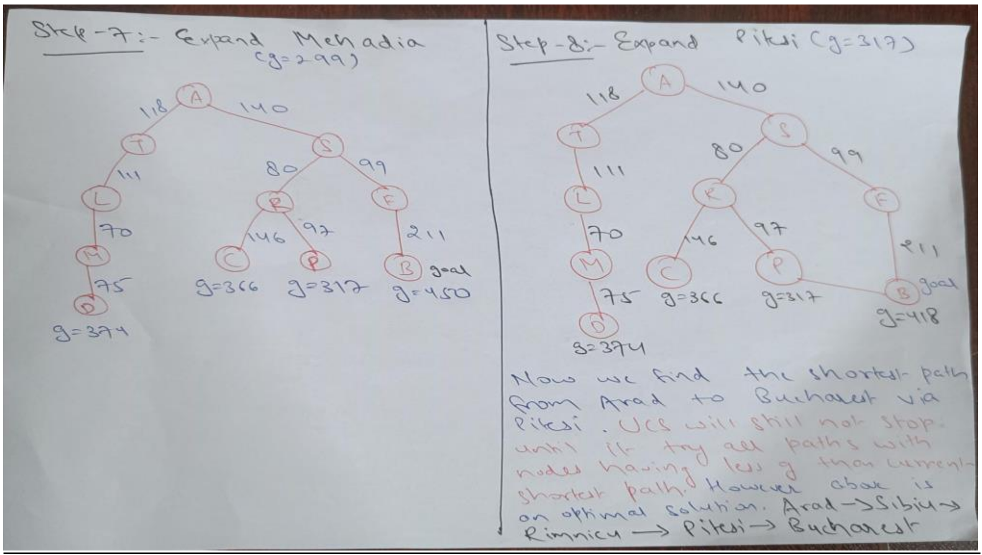


1. Step-1 : Expand Arad (g=0)

g:- cost of path from start (Arad) to current Node

Step-2 :- Expand Timisoara (g=118)

Step-3:- Expand Sibiu (g=140)

		 Step-4 : Expand Rimnicu (g=220) Step-5:- Expand Lugoj (g=229) Step-6:- Expand Fagarus (g=239) Note:- Bucharest (goal) is reached but UCS will not stop until it has expanded all nodes with g less than that of Bucharest (goal state) A T S L R F C P M B 118 140 111 80 99 97 146 70 211 g=229 g=239 g=366 g=317 g=299 g=450 (goal state)  Step-7:- Expand Mehadia (g=299) Step-8:- Expand Pitesi (g=317) Now we find the shortest path from Arad to Bucharest via Pitesi. UCS will still not stop. until it try all paths with nodes having less g than current shortest path. However above is an optimal solution. Arad->Sibiu-> Rimnicu -> Pitesi-> Bucharest A T S L R F M C P B D 118 140 111 80 99 70 146 97 211 75 g=366 g=317 g=450 goal g=374 g=418			
2	A	What do you mean by a Heuristic to be admissible, Explain with an example. What happens if you apply A* algorithm and the heuristic is not admissible **ANSWER :** In A*, we use a function: f(n)=g(n)+h(n) **g(n)**: Cost from the start node to current node `n` (known cost) **h(n)**: Heuristic estimate of cost from `n` to the goal (unknown/future cost, estimated) **f(n)**: Estimated total cost from start to goal through node `n`	5	CO2	L2

A heuristic **h(n)** is **admissible** if it **never overestimates** the actual cost to reach the goal from node `n`.

$$h(n) \leq h*(n)$$

Where:

- **h(n)** is your heuristic estimate
- **h*(n)** is the actual (real) cost from `n` to the goal

Guarantee: An admissible heuristic ensures that **A*** finds the **shortest (optimal) path**.

If we apply A* when heuristic is not admissible, it_**mislead A*** into:

- Thinking that a node is **more expensive than it really is**
- Skipping paths that are actually shorter
- Ultimately finding a **non-optimal path**

B

Why is the **Rational Agent** approach more widely used in modern AI? Discuss the **Turing Test** as a measure of AI. What are its limitations?

ANSWER :A **rational agent** is an entity that **acts to achieve the best possible outcome**, based on:

- Its **perception** of the environment
- Its **knowledge** and
- Its **goals**

In modern AI, we often define AI systems as **agents** that **perceive** and **act** in an environment to **maximize performance measures**.

Why It's More Popular Today ?

Reason	Explanation
🎯 **Goal-Driven Behavior**	Rational agents explicitly aim to maximize performance, which aligns well with real-world tasks like optimizing routes, diagnosing diseases, recommending products, etc.
🔄 **Real-Time Decision Making**	It fits environments that are **dynamic** and require **autonomous decisions**, like robotics, self-driving cars, game-playing agents, etc.
□ **Modular Design**	Rational agent models separate **perception, reasoning, and action**, making them easier to design, test, and upgrade.
📈 **Supports Learning**	Rational agents can incorporate **machine learning** to improve decisions over time (e.g., reinforcement learning agents).

5

CO1

L2

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr>
<td>📚
Theoretical Backing</td>
<td>Based on formal logic, probability, and utility theory—well understood and mathematically sound.

What Is the Turing Test?

Proposed by **Alan Turing in 1950**, the **Turing Test** is a way to measure a machine's **ability to exhibit intelligent behavior** equivalent to—or indistinguishable from—that of a human.

In the test, a human interacts with a **machine and a human**, both hidden behind a screen. If the human **cannot reliably tell** which is the machine, the machine is said to have passed the test.

Limitations of the Turing Test
<table>
<tr><th>Limitation</th><th>Explanation</th></tr>
<tr><td>🗣️ **Focuses Only on Imitation**</td><td>The test checks if an AI **acts like a human**, not whether it is **truly intelligent or rational**.</td></tr>
<tr><td>🎭 **Deceptive Behavior is Rewarded**</td><td>AI could pass by using **tricks, humor, or ambiguity**—not necessarily by understanding.</td></tr>
<tr><td>□ **Ignores Internal Process**</td><td>It doesn't care **how** the AI solves problems—whether it's reasoning, memorizing, or pattern-matching.</td></tr>
<tr><td>🚫 **Fails in Non-Chat Scenarios**</td><td>It’s hard to apply the Turing Test to **vision systems, robotics, planning agents, or medical AIs**.</td></tr>
<tr><td>📚 **Cultural/Language Bias**</td><td>Human judges may favor certain speech patterns, slang, or humor that vary by region or age.</td></tr>
</table>
</td>
</tr>
</table>
</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>Explain 8-puzzle problem. Explain how Manhattan distance is a better heuristic than Misplaced tiles.

ANSWER : The **8-puzzle problem** is a classic in AI and search algorithms, especially when demonstrating **heuristics** in **A***</td>
<td>5</td>
<td>CO1</td>
<td>L2</td>
</tr>
</table>

What Is the 8-Puzzle Problem?

- A **3×3 grid** with **8 numbered tiles** and **1 blank space** (represented as 0 or empty).
- The goal is to **rearrange the tiles** from a **given initial configuration** into the **goal state** by sliding tiles into the empty space.

☒ The picture can't be displayed.

Legal Moves

You can move the blank tile **up, down, left, or right**, as long as it's within the grid bounds. Each move has a **uniform cost** (usually 1).

How Manhattan distance is a better heuristic than Misplaced tiles ?

The **Manhattan distance heuristic** is generally considered better than the **Misplaced tiles heuristic** in solving problems like the 8-puzzle because it provides a more informed estimate of how far a given state is from the goal. While the Misplaced tiles heuristic simply counts the number of tiles that are not in their correct position, it treats all misplacements equally, offering no sense of how far those tiles actually are from where they need to be. In contrast, Manhattan distance calculates the total number of moves required to get each tile to its correct location by summing the vertical and horizontal distances, thus reflecting the actual effort needed to solve the puzzle more accurately. This added precision helps algorithms like A* make more intelligent decisions about which paths to explore, often resulting in fewer node expansions and faster convergence to the optimal solution. Since both heuristics are admissible (they never overestimate the true cost), Manhattan distance remains the superior choice due to its finer granularity and stronger guidance toward the goal.

	B	A vacuum cleaner operates in a **four-room environment** with locations **A, B, C, and D**. Each room can either be **clean or dirty**, and the vacuum can be in any one of the four rooms at a time 1. Each room can be in one of two states: **Clean (C) or Dirty (D).** 2. The vacuum can be in one of the four locations: **A, B, C, or D.**	5	CO1	L3

(a) Determine the total number of possible states in this environment.
(b) How many states will there be if the vacuum cleaner can also be **turned on or off**?

ANSWER : Given:

- **4 rooms**: A, B, C, D
- Each room can be **Clean (C)** or **Dirty (D)** → 2 possible states per room
- Vacuum cleaner can be in **one** of the 4 rooms at a time → 4 positions

a.

Step 1: Room states

Each room has 2 states → For 4 rooms:

2^4=16 combinations of room

Step 2: Vacuum location

Vacuum can be in **one** of 4 rooms:

4 positions

Final: Total states = Room combinations × Vacuum positions

16×4= **64 possible states**

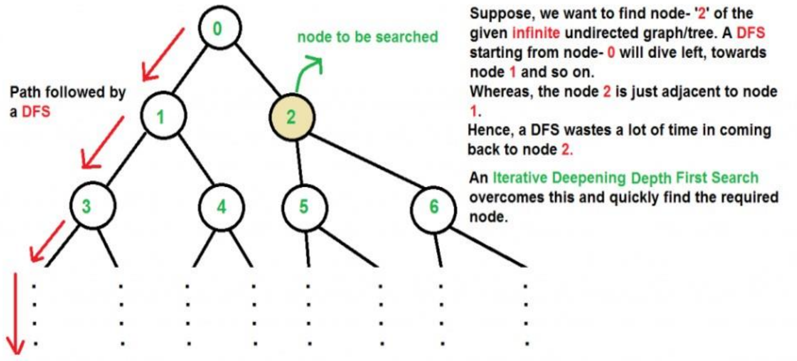
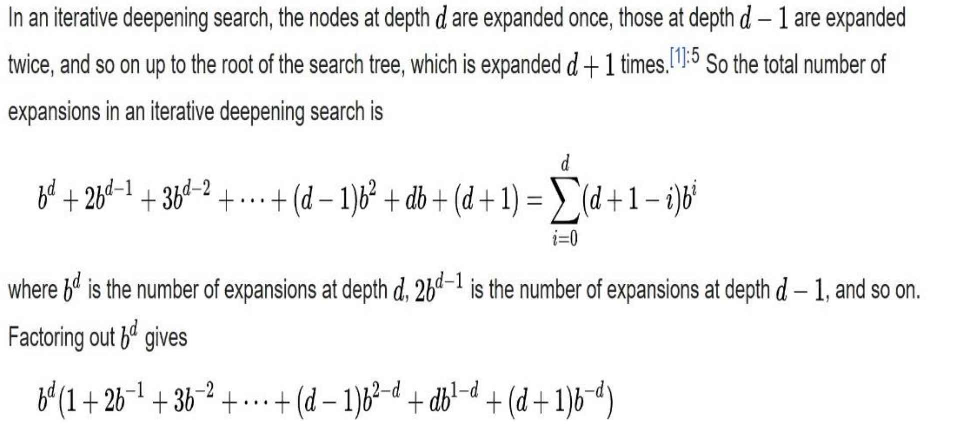
(b) Now add vacuum cleaner on/off state

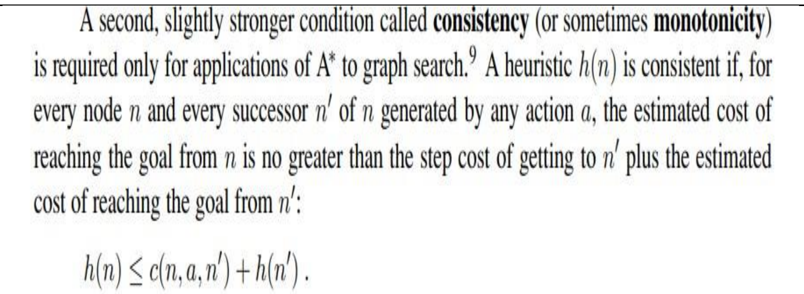
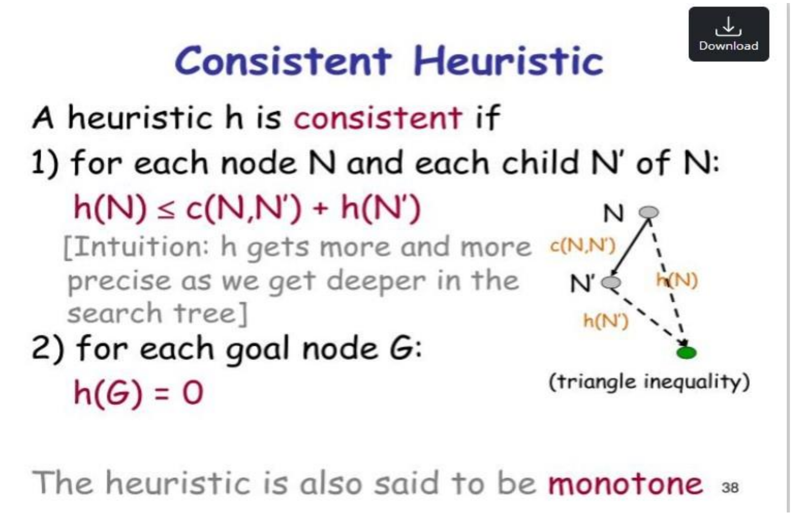
That's **2 more states** (on or off)

New total states=16 (room states)×4 (vacuum positions)×2 (on/off)
16×4×2=**128 possible states**

Final Answers:

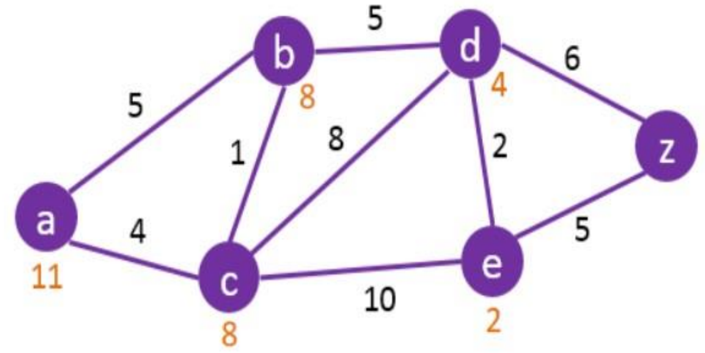
- (a) **64** states
- (b) **128** states (with on/off)

4	A	Give a scenario where DFS is not complete but Iterative Deepening Depth First Search (IDS) is ? Give expression of time and space complexity of BFS in terms of branching factor b and Depth d. **ANSWER :**  0 1 2 3 4 5 6 Path followed by a DFS node to be searched Suppose, we want to find node- '2' of the given infinite undirected graph/tree. A DFS starting from node- 0 will dive left, towards node 1 and so on. Whereas, the node 2 is just adjacent to node 1. Hence, a DFS wastes a lot of time in coming back to node 2. An Iterative Deepening Depth First Search overcomes this and quickly find the required node. Give expression of time and space complexity of BFS in terms of branching factor b and Depth d. Time complexity : b^d Space complexity : b^d	5	CO2	L2
	B	Derive time Complexity Formula for Iterative Deepening Depth First Search (IDS) in terms of branching factor b and Depth d. **ANSWER :**  In an iterative deepening search, the nodes at depth d are expanded once, those at depth $d-1$ are expanded twice, and so on up to the root of the search tree, which is expanded $d+1$ times.[1]:5 So the total number of expansions in an iterative deepening search is $b^d + 2b^{d-1} + 3b^{d-2} + \cdots + (d-1)b^2 + db + (d+1) = \sum_{i=0}^{d}(d+1-i)b^i$ where b^d is the number of expansions at depth d, $2b^{d-1}$ is the number of expansions at depth $d-1$, and so on. Factoring out b^d gives $b^d(1 + 2b^{-1} + 3b^{-2} + \cdots + (d-1)b^{2-d} + db^{1-d} + (d+1)b^{-d})$ For larger values of d, b^d is the most dominating hence time complexity is b^d	5	CO2	L3
5	A	What do you mean by Consistency of a heuristic ? How does this property make a heuristic better for a solution ? **ANSWER :**	5	CO2	L1

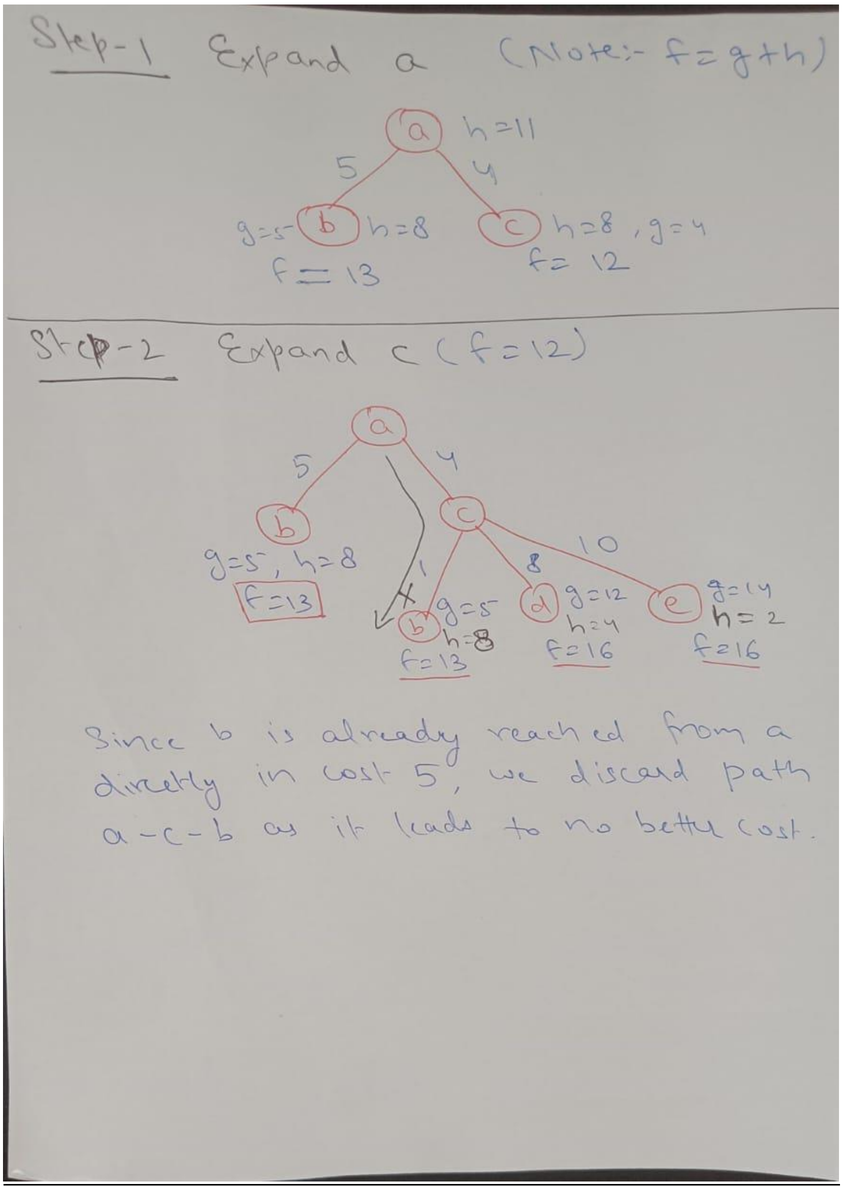
		 A second, slightly stronger condition called **consistency** (or sometimes **monotonicity**) is required only for applications of A* to graph search.[9] A heuristic $h(n)$ is consistent if, for every node n and every successor n' of n generated by any action a, the estimated cost of reaching the goal from n is no greater than the step cost of getting to n' plus the estimated cost of reaching the goal from n': $h(n) \leq c(n,a,n') + h(n')$.  Consistent Heuristic A heuristic h is consistent if 1) for each node N and each child N' of N: h(N) ≤ c(N,N') + h(N') [Intuition: h gets more and more precise as we get deeper in the search tree] 2) for each goal node G: h(G) = 0 N c(N,N') N' h(N) h(N') (triangle inequality) The heuristic is also said to be monotone 38 How does this property make a heuristic better for a solution ? • A* with a consistent heuristic doesn't need to revisit nodes already explored — it avoids unnecessary computation and re-evaluation. • This makes the search **more efficient**			
	B	**ANSWER :** The **PEAS framework** is used to define the **task environment** for an intelligent agent. It stands for: **P**erformance measure, **E**nvironment, **A**ctuators, and **S**ensors. PEAS for a Hospital Surgery Robot ◈ P – Performance Measures These define how we evaluate the robot's success: • **Surgical accuracy** (precision of incisions, sutures, etc.) • **Patient safety** (minimal damage to surrounding tissue) • **Procedure completion time** • **Post-surgery recovery rate**	5	CO1	L2

		- **Minimization of blood loss** - **Success rate of surgeries** - **Error rate (should be extremely low or zero)** ◈ E – Environment What surrounds and affects the robot: - Operating room - Patient's body and anatomy - Human surgeons and assistants - Surgical tools and equipment - Medical staff coordination - Emergency medical protocols (e.g., heart failure or bleeding) ◈ A – Actuators What the robot uses to act upon the environment: - Robotic arms (for cutting, suturing, holding instruments) - Surgical tools (scalpel, scissors, clamps, etc.) - Laser or cauterization tools - Display/monitor interface (for communicating with human surgeons) - Automated drug injectors (e.g., anesthesia delivery) - Movement mechanisms (fine-tuned motion for delicate work) ◈ S – Sensors What the robot uses to perceive the environment: - High-definition cameras (2D/3D vision for precision) - Depth sensors (for understanding organ/tissue positioning) - Force/pressure sensors (for delicate touch and feedback) - Vital sign monitors (heart rate, blood pressure, oxygen) - Audio input (for voice commands or alerts) - Motion/position sensors (for arm calibration and alignment) - Infrared or ultrasound imaging sensors (to see beneath tissue)			
		Apply the steps of the A* Search algorithm to find the shortest path from A to Z using the following graph:	10	CO2	L3

6



ANSWER :



Step-1 Expand a (Note:- f=g+h)

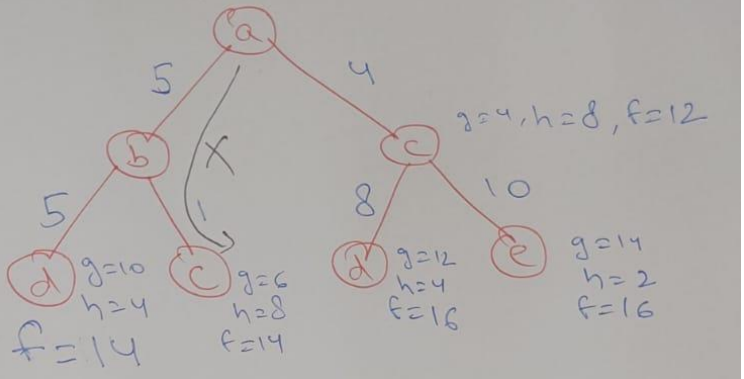
a h=11
5
4
g=5 b h=8
f=13
c h=8 , g=4
f= 12

Step-2 Expand c (f=12)

a
5
4
b
g=5, h=8
f=13
c
1
8
10
b g=5
h=8
f=13
d g=12
h=4
f=16
e g=14
h= 2
f=16

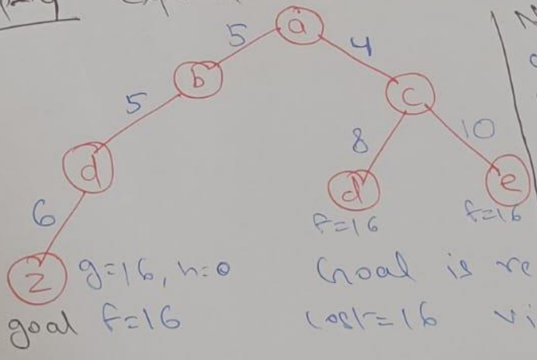
Since b is already reached from a directly in cost 5, we discard path a-c-b as it leads to no better cost.

Step-3 :- Expand b (f= 13) via a-b



Since we already have a-c path directly with g=4, we discard path a-b-c, with g=6. However, we update a-b-d as shorter path to d, than a-c-d

Step-4 Expand d (f=14) via a-b-d.



Note:- There are other paths with same cost (16), like a-c-d-z, but A* doesn't find that

Goal is reached with cost=16 via

a-b-d-z
(path followed by A*)

CCI CI HOD