NAME: FARDOUSE LOMAT JAHAN RUMPA

Dept: BSc in CSE

Batch: 17th

ID NO: 22191170041

Course Title: Artificial Intelligence

Course code: CSE-341

①

Ans to the Qus No: 01(a)

(a) Ans: Blind search Algorithm:

These algorithms are brute force operations and they don't have additional information about the search space. The only information they have to is on how to traverse or visit the nodes in the tree. Thus uninformed search algoriths are also called blind search algorithms.

Types of Uninformed Search Algorithms →

The different types of uninformed search Algorithms used in AI are as follows:

→ Depth first Search
→ Breadth-first Search
→ Depth Limited Search.
→ Uniform Cost Search
→ Iterative Deepening Depth first Search
→ Bidirectional Search (if applicable)

But before we go into these search types and you go a step further wandering into any artificial Intelligence course. Let's get to know the few terms which will be frequently used in the upcoming sections.

Here are some common blind search algorithms along their properties.

① Breadth-first search (BFS)

→ **Property:** BFS expands the shallowest unexpanded node first, i.e. It explores all the neighboring nodes before moving to the next level of the search tree.

②

→ completeness: BFS is complete if the branching factor is finite and there is a solution.

→ optimality: BFS guarantees an optimal solution when the cost of each step is uniform.

② Depth-first search (DFS):

→ property: DFS expands the deepest unexpanded node first. i.e. It explores as far as possible along each branch before backtracking.

→ completeness: DFS is not complete if the search space contains loops or infinite paths.

③ Iterative Deepening Depth-first search (IDDFS):

→ property: IDDFS is a combination of depth-first seach and breadth-first seach. It proform depth-limited seach iteratively gradually increasi the depth limit until a solution is found.

⑤ Depth-limited search (DLS):

property: DLS limits the depth of the search to a pre-defined level, beyond which nodes are not expanded.

⑥ Biodirectional search:

property: Biodirectional search explores the search space form both the initial and goal states, meeti in the middle.

These are just a few examples of blind search algorithms.

Ans to the Qus NO: 01(b)

(b) Ans: The four general steps of problem solving are a systematic approach to finding solutions to problems. They provide a framework for tackling problems in a structured manner. Here is a brief description of each step:

① Define the problem:

The first step in problem solving is to clearly define and understand the problem at hand. This involves indentifying the specific issue or challenge determining its scope and boundaries, and gathering all relevant information. By clearly defining the problem. you set the foundation for finding an effective solution.

② Generate possible solutions;

Once the problem is defined, the next step is to generate a range of possible solutions. This involves brainstorming and considering different ideas, approaches, and perspective. The goal is to explore various options and be open to creative solutions without judgement or evaluation at this stage.

③ Evaluate and Select a Solution:

After generating a list of possible solutions, the next step is to evalute each option and select

(2)

the most appropriate one. This involves analyzing the potential pros and cons of each solutions. Considering factors such as feasiblity, effectiveness, resources required, and potential outcomes. It's important to critically asses the the solutions and choose the one that best aligns with the defined problem and desired outcome.

(4) Implement and Review:

Once a Solution is chosen, the next step is to implement it. This involves putting the Selected solution into action and executing the necessary steps. It's important to monitor the progress, gather feed back, and assess the effectiveness of the implemented Solution.

These four steps provides a structured approach to problem solving, guidling Individuals through the process of defining, generating evaluating and implemnting solutions. By following these steps, problem solvers can increase their chances of finding effective and efficiant solutions to a wide range of problems.

Ans to the Qus NO: 04 (a)

Ans: A problem-solving agent is an entity typically an intelligent system or program, that is designed to analyze problems, generate solutions and take actions to acheive desired goal or outcomes. It is an autonomous agent that uses its knowledge and reasoning capabilities to navigate through problem spaces and find optimal or satisfactory solutions.

Here is a breif description and a Simplified diagram of a problem-solving agent :

① Sensors:

Sensors are responsible for perceiving the environment and gathering relevant information about the current state of the agent, allowing it to observe and understand the problem at hand. Sensors can include various types of input devices, such as cameras, microphones, or other sensors specific to the problem domain.

② Knowledge Base:

The knowledge base repprents the agent's internal repository of information, facts, rules, and heuristics about the problem domain. It contains pre-existing knowledge and learned information that the

the agent can use to reason, make decision and generate solutions.

③ Problem formalation:

Problem Formulation involves converting the perceived problem into a well-denied representation that the agent can work with. It includes deffining the initial state, the goal state the available actions or operators, and any constraints or limitions within the problem space.

④ Search and planning.

The search and planning component is responsi for exploring the problem space, searching for possible solutions, and planning a sequences of actions to reach the desired goal state. It uses various search algorithms, heuristic methods, and planning teachques to navigat through the problem domain efficiently.

⑤ Decision Making:

The decision-making component analyzes the available information, evalubates potential soulotions, and selects the mosts appropriate action or plan based on the state.

Problem-Solving Agent

Perception Module

Problem-Analysis Module

Solution Generation Module

These modules work togethere in a coordinated manner to enable the problem-solving agent to understand, analyze and solve problems efficiently.

Ans to the Ques No: 04(b)

Ans: The Depth-first search (DFS) strategy has several limitations including:

① Completeness: DFS does not guarantee finding a solution if one exists. It may get stuck in an infinite loop if the search space has cycles. To overcome this techniques like cycle detection or iterative deepening can be used.

② Optimal Solution:

DFS does not guarantee finding the optimal solution.

It may find a suboptimal solution before exploring the entire search space. To find the optimal soluti[on] additional techniques like iterative deepening depth-first search or using cost functions are required.

③ Memory Usage: DFS can consume a large amoun[t] of memory when traversing deep paths. It stores all visited nodes on the stack, which can lead to stack overflow if the especially problematic in graphs with high branching factors or infin[ite] -depth search spaces.

④ Time Complexity: The time complexity of DFS c[an] be high in the worst-cas scenario, particularly when the search space is large and unbounded. It may continue exploring deep paths before finding a solution, resulting in inefficient sear[ch]

⑤ Lack of Breadth: DFS explorees a single path i[n] depth before moving to the next branch, potentially neglecting other branches and missing potentially bethe bether solutions. It may overlook shorter or more efficient paths that exist in other branc[hes]

⑥ Lack of information: DFS does not have inform[ation] about the entire search space or its structure at the beginning of the search. This lack of information makes it difficult to make infor[med] decisions about which paths to explore or prioritize.

⑦ No Backtracking: DFS does not backtrack automatically. if a path leads to a dead end or an invaild solution, DFS will continue exploring other paths wasting time and resourceing.

It's important to note that some of these limitations to note that some of these limitations can be addressed by combing DFS with other search strategies to the problem domain.

Ans to the Qus NO: 05 (a)

Ans: To avoid repeated states in search algorithms like Depth-First Search (DFS) Breath-first Search (BFS) or any other search strategy, you can employ the following solution:

① Visited set / Hash table: Maintain a set or hash table to keep track of visited states. whenever you visit a new state, check it is already exists in the visited set. If it does, skip that state and move to the ~~x~~ next one. This prevents revisiting previously explored states.

② State Representation: Ensure that the representation of each state is unique. If two different states have the same representation, they will be treated as the same state potentially leading to repeated states. Make sure the state representation captures all necessary information to differentiate between states accurately.

③ Graph search: Modify the search algorithm to treat the problem as a graph search problem rather than blindly searching through the state space. This involves representing the states and transitions as nodes and edges in a graph. Use techniques like graph traversal algorithms (BFS, DFS) with cycle detection to ensure that cycles are not revisited.

④ Memoization:- If the problem involves evaluating subproblems repeatedly, use memoization to store the results of these subproblems. This technique allows you to avoid re-computing the same subproblem multiple times, reducing the changes of revisiting states.

⑤ Heuristic Information: If you have access to heuristic information or an evaluation function incorporate it into your search algorithm. Use this information to prioritize the exploration of US states that are more likely to lead to a solution. This can help in avoiding unnecessary exploration of paths that are unlikely to yield a desired outcome.

Ans to the Qus No: 05 (b)

Ans: Certainly! Here are some additional real-world problems where searching algorithms can be applied:

① E-commerce: Searching algorithms are used to power search functionalities on e-commerce platforms, enabling users to find products based on keywords categories, filters and other criteria.

② Supply chain management: searching algorithms can help optimize supply chain processes by efficiently searching for the best routers, transportation modes, or warehouses to minimize costs and maximize efficiency.

③ Job search:

Searching algorithms can be utilized in job search platforms to match job seekers with relevant job openings based on their skills, qualifications, and preferences.

④ Natural language processing:

Searching algorithms are employed in natural language processing applications to perform tasks like sentiment analysis, information retrieval, question answering, and text summarizing.

⑤ Healthcare: Searching algorithms can assist in medical diagnosis by searching through vast medical databases to find relevent options for specific symptoms or conditions.

Ans to the QUS NO: 02 (a)

Ans: States:

The states of the robot assembley can be represent by the configuration and states of the robot at any given time. This includes the position and orientation of the robot. the status of its various components, and any relevant environmental conditions.

Initial state:

The initial state reprensents the starting configurati and status of the robot assembly. It could be a specifi position and orientation of the robot. with all it's components properly assembled and functional.

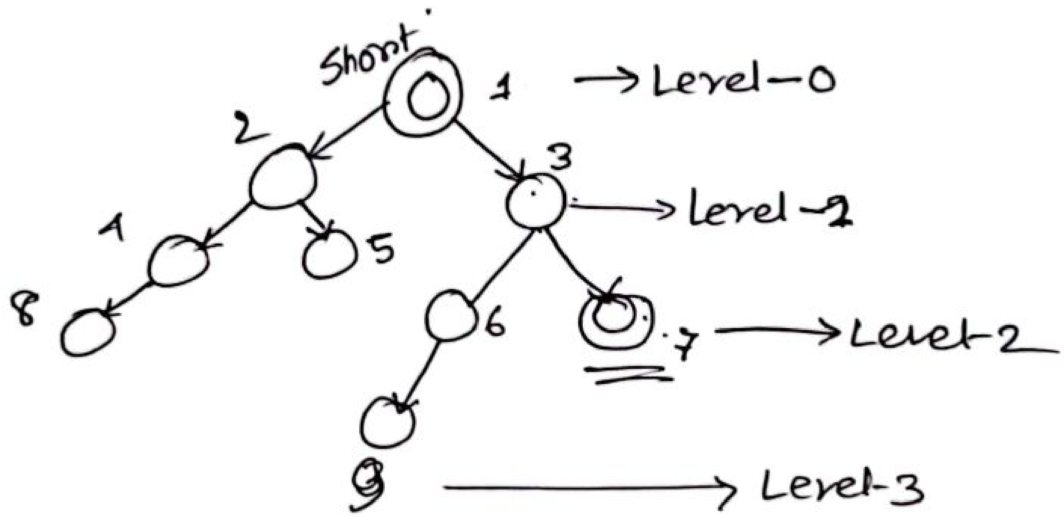
Actions: Actions are the possible moves or operati that the robot can perform to change its states. In the context of a robot assembly, actions could include picking up a component, moving to a specific location.

Goal test: The goal test determines wheather th current state of the robot assembly statisfies th desired goal or objective.

path cost: The path cost represents the cost or effort associated with reaching a particular state from the initial state. The cost can be defined based on factors such as the distance traveled. The number of actions performed, the time taken, and any resource consumption.

Ans to the Qus No: 02(b)

Ans:



BFS : 1, 2, 3, 4, 5, 6, 7, 8, 9