A* algotithm

1. Start and Goal Nodes: A* operates on a graph with nodes connected by edges. It requires a start node (initial state) and a goal node (target state).
2. Heuristic Function (h): A* uses a heuristic function to estimate the cost from any given node to the goal node. This function provides a rough estimate of how far away a node is from the goal.
3. Cost Function (g): A* also keeps track of the cost incurred from the start node to the current node. This is the actual cost or distance traveled along the path.
4. Open and Closed Lists: A* maintains two lists of nodes: the open list and the closed list. The open list contains nodes that have been discovered but not yet explored, while the closed list contains nodes that have already been visited.
5. F-score (f): For each node, A* calculates an f-score, which is the sum of the cost function (g) and the heuristic function (h). It represents the total estimated cost of the cheapest path from the start node to the goal node passing through that node.

Generic algorithm steps

1. Initialization: Create an initial population of candidate solutions randomly.
2. Evaluation: Evaluate the fitness of each candidate solution in the population using a fitness function.
3. Selection: Select individuals from the current population to become parents based on their fitness. Individuals with higher fitness have a higher chance of being selected.
4. Recombination (Crossover): Create offspring by combining genetic material from selected parents. This is typically done through techniques like single-point crossover, multi-point crossover, or uniform crossover.
5. Mutation: Introduce random changes to offspring's genetic material to maintain diversity in population.
6. Replacement: Replace the least fit individuals in the current population with the newly generated offspring to form the next generation.
7. Termination: Repeat steps 2-6 for a certain number of generations or until a termination criterion is met (e.g., reaching a satisfactory fitness level or running out of computational resources).

Resolving steps

1. Convert all the statements of F to clause form.
2. Negate P and convert the result to clause form. Add it to the set of clauses obtained in 1..
3. Repeat until either a contradiction is found, no progress can be made, or a predetermined amount of effort has been expended.
(i) Select two clauses: Call these the parent clause.
(ii) Resolve them together: The resolvent will be the disjunction of all literal of both parent clauses with appropriate substitution performed and with the following exception. If there is one pair of literals T1 and - T2 such that one of the parent clauses contains T1 and the other contains T2 and if T1 and T2 are unifiable, then neither T1 nor T2 should appear in the resolvent. We call T1 and T2 complementary literals.Use the substitution produced by the unification to create the resolvent. If there is more than one, pair of complementary literals, only one pair should be omitted from the resolvent.
iii) If the resolvent is the empty clause, then a contradiction has been found. If it is not, then add it to the set of clauses available to the procedure.

NLP steps

1. Lexical analysis: It involves identifying and analyzing the structure of words. Lexicon of a languag means the collection of words and phrases in a language.
2. Syntactic analysis (Parsing) : It involves analysis of words in the sentence for grammar and artanging words in a manner that shows the relationship among the words.
3. Semantic analysis: It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. .
4. Discourse integration: The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.
5. Pragmatic analysis: During this, what was said is reinterpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

types of planning explain any one

1. **Reactive Planning:**

Definition: Reactive planning focuses on generating plans or actions in real-time based on the current state of the environment, without explicitly considering future states or goals.**Architecture:** Reactive planners often employ reactive architectures, where the system reacts to sensory inputs and generates responses using predefined rules or policies.
Examples: Reactive planning is commonly used in robotics, where robots respond to sensor readings and perform actions to navigate or manipulate objects in the environment.

2. **Hierarchical Planning:**
Definition: Hierarchical planning organizes the planning process into a hierarchy of abstraction levels, where each level focuses on different aspects of the problem.
Structure: The hierarchy typically consists of high-level goals at the top, decomposed into subgoals at lower levels, and eventually into primitive actions or steps at the lowest level.**Advantages:** Provides a structured and modular approach to planning, allowing complex tasks to be broken down into simpler subtasks. Helps manage the complexity of planning problems by focusing on different levels of abstraction.**Examples:** Task planning in robotics, where high-level goals (e.g., "clean the room") are decomposed into subgoals (e.g., "move to the kitchen", "pick up the broom").

3.**Sequential Planning**
Definition: Sequential planning involves generating a sequence of actions to achieve a goal state from an initial state, without considering alternative action orders.
Process: The planner explores the state space by considering one action at a time, applying each action to the current state to generate successor states, until a goal state is reached.**Search Algorithms:** Sequential planning often employs search algorithms such as depth-first search, breadth-first search, or A* search to explore the state space efficiently.**Examples:** Path planning in robotics, where a robot navigates through a series of waypoints to reach a destination

4.**Partial order planning**
Definition: Partial order planning allows actions to be executed in a flexible order, rather than strictly following a predetermined sequence of steps.**Representation:** Plans are represented as partially ordered sets of actions, where some actions are ordered with respect to each other, while others can be executed concurrently.**Advantages:** Offers more flexibility than sequential planning, allowing for parallel execution of actions and handling of temporal constraints.
Examples: Planning in domains where the order of actions is not strictly defined, such as scheduling tasks in a manufacturing process or coordinating multiple agents in a multi-agent system.

5.Conditional planning
Definition: Conditional planning deals with planning under uncertainty by incorporating conditional branches and probabilistic outcomes.**Branching Decisions:** The planner considers multiple possible outcomes of actions and makes decisions based on conditional probabilities or logical conditions.**Branch-and-Bound:** Conditional planners often use techniques like branch-and-bound search to explore different branches of the plan space and prune unpromising paths. **Uncertainty Representation:** Uncertainty may arise from factors such as stochastic actions, noisy sensors, or incomplete knowledge about the environment.**Applications:** Planning in probabilistic domains, such as robot navigation in uncertain environments, diagnosis and treatment planning in healthcare, and decision-making in autonomous systems.

cTypes of learning explain any one

1. **Supervised Learning:**

- **Definition**: Supervised learning involves learning a mapping from input data to output labels based on a labeled dataset.
- **Training Data**: The training dataset consists of input-output pairs, where the correct output (label) is provided for each input.
- **Objective**: The goal is to learn a function that can accurately predict the output for new, unseen inputs.
- **Examples**: Classification (e.g., spam detection, image recognition) and regression (e.g., predicting house prices, stock prices).

2.Unsupervised Learning:

- Definition: Unsupervised learning involves learning patterns or structures from input data without explicit supervision.
- Training Data: The training dataset consists of input data without corresponding output labels.
- Objective: The goal is to discover hidden patterns, groupings, or relationships within the data.
- Examples: Clustering (e.g., customer segmentation, document clustering) and dimensionality reduction (e.g., principal component analysis, t-distributed stochastic neighbor embedding).

3.Semi-Supervised Learning:

- Definition: Semi-supervised learning is a combination of supervised and unsupervised learning, where the model learns from a small amount of labeled data and a large amount of unlabeled data.
- Training Data: The training dataset includes both labeled and unlabeled data.
- Objective: The goal is to leverage the unlabeled data to improve the performance of the model trained on limited labeled data.
- Examples: Text classification with a small labeled dataset and a large corpus of unlabeled text, image recognition with few labeled images and many unlabeled images

Types of sensors
Camera Sensors:Applications: Object recognition, image classification, facial recognition, gesture recognition, surveillance, autonomous vehicles, robotics, augmented reality (AR), and virtual reality (VR).
Lidar (Light Detection and Ranging) Sensors:
Applications: Distance measurement, 3D mapping, obstacle detection and avoidance (e.g., in autonomous vehicles and drones), environment modeling, and terrain analysis.
Radar (Radio Detection and Ranging) Sensors:
Applications: Object detection and tracking, speed measurement, collision avoidance (e.g., in
autonomous vehicles and drones), weather monitoring, and target recognition in military applications.
Inertial Measurement Unit (IMU):
Applications: Motion sensing, orientation tracking, gesture recognition, activity monitoring (e.g., in fitness trackers), inertial navigation (e.g., in drones and robots), and virtual reality systems.
Microphones and Audio Sensors:
Applications: Speech recognition, voice command systems, noise cancellation, acoustic monitoring (e.g., for security purposes), sound classification, and audio-based user interaction (e.g., in smart home devices).
Pressure Sensors:
Applications: Barometric pressure measurement (for weather forecasting and altitude
estimation), flow rate measurement, altitude tracking (e.g., in drones and aircraft), liquid level
sensing, and touch sensitivity in touchscreens.

Characteristics of environment and explain any one

(1) Fully observable versus partially observable
If the agent sensors describes the environment at each point in detail, then the environment is fully observable. Fully observable environments are convenient because agent need not maintain track of the world. An environment may be partially observable because an automated taxi cannot see what other drivers are thinking. A fully observable environment is also termed as accessible environment,
while partially observable environment is termed as maccessible environment.Playing card-games is a perfect example of partially observable environment where a player is not aware of the card in the opponent's hand.

(2) Static Vs-Dynamic
If the environment can change itself while an agent is deliberating then such environment is called a
dynamic environment, else it is called a static environment. Static environment are easy to deal because an agent does not need to continue looking at the world while deciding for an action.But for dynamic environment, agents need to keep looking at the world at each action.Taxi driving is an example of a dynamic environment whereas playing on Piano is an example of a static environment. If the environment does not change with the passage of time, but the agent
performance changes by time, then it is semi-dynamic.

(**3).Deterministic Vs. Stochastic:**
In a deterministic environment, the outcome of actions taken by the agent is fully determined by the current state of the environment and the action itself.There is no randomness involved in the transition between states or the consequences of actions.Each action taken by the agent leads to a predictable and consistent outcome.In a stochastic environment, there is some degree of randomness or uncertainty in the outcome of actions.The transition between states and the consequences of actions may be probabilistic, meaning that the outcome is not fully determined by the current state and action.Multiple outcomes are possible for the same action taken in the same state, each with a certain probability.

Breadth-First Search (BFS):
BFS explores all nodes at a given depth level before moving to the next depth level.It uses a queue to maintain the order of nodes to be visited (FIFO).BFS guarantees finding the shortest path in unweighted graphs.Suitable for finding shortest paths and exploring neighbor nodes uniformly.Requires more memory due to the need to store nodes at each depth level.
Depth-First Search (DFS):
DFS explores as deeply as possible along each branch before backtracking. It uses a stack (or recursion) to maintain the order of nodes to be visited (LIFO).DFS may not necessarily find the shortest path.Suitable for tasks like topological sorting and maze generation.Typically requires less memory compared to BFS, as it only needs to store the path from the root to the current node

PEAS for automatic taxi driver
1. **Performance measure :** it is the measure of success for an automatic taxi driver it should include factor such as:**(a)**Safety and timely arrival of passengers**(b)** Efficient root planning for minimising travel time**(c)** Customer satisfaction and positive feedback
2. **Environment:** the environment for an automatic taxi driver includes road traffic conditions pedestrians other vehicles City infrastructure. The dynamic and unpredictable nature of traffic environment makes it Complex for taxi to navigate.
3. **Actuators:** these are mechanism through which the agent interacts with environment. actuators for taxi driver include:
(a) Accelerator and brake pedals for controlling speed**(b)** Tearing mechanism for navigation**(c)** Turn Signals and brake lights for communication with others.
4. **Sensors:** these are means by which agent purses and gathers information about environment. for automatic taxi driver sensors include:
(a) GPS for Location tracking and navigation.**(b)** Reader for measuring distance to other objects.**(c)**Microphones and speakers for communication with passengers.

A* algotithm

1. Start and Goal Nodes: A* operates on a graph with nodes connected by edges. It requires a start node (initial state) and a goal node (target state).
2. Heuristic Function (h): A* uses a heuristic function to estimate the cost from any given node to the goal node. This function provides a rough estimate of how far away a node is from the goal.
3. Cost Function (g): A* also keeps track of the cost incurred from the start node to the current node. This is the actual cost or distance traveled along the path.
4. Open and Closed Lists: A* maintains two lists of nodes: the open list and the closed list. The open list contains nodes that have been discovered but not yet explored, while the closed list contains nodes that have already been visited.
5. F-score (f): For each node, A* calculates an f-score, which is the sum of the cost function (g) and the heuristic function (h). It represents the total estimated cost of the cheapest path from the start node to the goal node passing through that node.

Generic algorithm steps

1. Initialization: Create an initial population of candidate solutions randomly.
2. Evaluation: Evaluate the fitness of each candidate solution in the population using a fitness function.
3. Selection: Select individuals from the current population to become parents based on their fitness. Individuals with higher fitness have a higher chance of being selected.
4. Recombination (Crossover): Create offspring by combining genetic material from selected parents. This is typically done through techniques like single-point crossover, multi-point crossover, or uniform crossover.
5. Mutation: Introduce random changes to offspring's genetic material to maintain diversity in population.
6. Replacement: Replace the least fit individuals in the current population with the newly generated offspring to form the next generation.
7. Termination: Repeat steps 2-6 for a certain number of generations or until a termination criterion is met (e.g., reaching a satisfactory fitness level or running out of computational resources).

Resolving steps

1. Convert all the statements of F to clause form.
2. Negate P and convert the result to clause form. Add it to the set of clauses obtained in 1..
3. Repeat until either a contradiction is found, no progress can be made, or a predetermined amount of effort has been expended.
(i) Select two clauses: Call these the parent clause.
(ii) Resolve them together: The resolvent will be the disjunction of all literal of both parent clauses with appropriate substitution performed and with the following exception. If there is one pair of literals T1 and - T2 such that one of the parent clauses contains T1 and the other contains T2 and if T1 and T2 are unifiable, then neither T1 nor T2 should appear in the resolvent. We call T1 and T2 complementary literals.Use the substitution produced by the unification to create the resolvent. If there is more than one, pair of complementary literals, only one pair should be omitted from the resolvent.
iii) If the resolvent is the empty clause, then a contradiction has been found. If it is not, then add it to the set of clauses available to the procedure.

NLP steps

1. Lexical analysis: It involves identifying and analyzing the structure of words. Lexicon of a languag means the collection of words and phrases in a language.
2. Syntactic analysis (Parsing) : It involves analysis of words in the sentence for grammar and artanging words in a manner that shows the relationship among the words.
3. Semantic analysis: It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. .
4. Discourse integration: The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.
5. Pragmatic analysis: During this, what was said is reinterpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

types of planning explain any one

1. **Reactive Planning:**

Definition: Reactive planning focuses on generating plans or actions in real-time based on the current state of the environment, without explicitly considering future states or goals.**Architecture:** Reactive planners often employ reactive architectures, where the system reacts to sensory inputs and generates responses using predefined rules or policies.
Examples: Reactive planning is commonly used in robotics, where robots respond to sensor readings and perform actions to navigate or manipulate objects in the environment.

2. **Hierarchical Planning:**
Definition: Hierarchical planning organizes the planning process into a hierarchy of abstraction levels, where each level focuses on different aspects of the problem.
Structure: The hierarchy typically consists of high-level goals at the top, decomposed into subgoals at lower levels, and eventually into primitive actions or steps at the lowest level.**Advantages:** Provides a structured and modular approach to planning, allowing complex tasks to be broken down into simpler subtasks. Helps manage the complexity of planning problems by focusing on different levels of abstraction.**Examples:** Task planning in robotics, where high-level goals (e.g., "clean the room") are decomposed into subgoals (e.g., "move to the kitchen", "pick up the broom").

3.**Sequential Planning**
Definition: Sequential planning involves generating a sequence of actions to achieve a goal state from an initial state, without considering alternative action orders.
Process: The planner explores the state space by considering one action at a time, applying each action to the current state to generate successor states, until a goal state is reached.**Search Algorithms:** Sequential planning often employs search algorithms such as depth-first search, breadth-first search, or A* search to explore the state space efficiently.**Examples:** Path planning in robotics, where a robot navigates through a series of waypoints to reach a destination

4.**Partial order planning**
Definition: Partial order planning allows actions to be executed in a flexible order, rather than strictly following a predetermined sequence of steps.**Representation:** Plans are represented as partially ordered sets of actions, where some actions are ordered with respect to each other, while others can be executed concurrently.**Advantages:** Offers more flexibility than sequential planning, allowing for parallel execution of actions and handling of temporal constraints.
Examples: Planning in domains where the order of actions is not strictly defined, such as scheduling tasks in a manufacturing process or coordinating multiple agents in a multi-agent system.

5.Conditional planning
Definition: Conditional planning deals with planning under uncertainty by incorporating conditional branches and probabilistic outcomes.**Branching Decisions:** The planner considers multiple possible outcomes of actions and makes decisions based on conditional probabilities or logical conditions.**Branch-and-Bound:** Conditional planners often use techniques like branch-and-bound search to explore different branches of the plan space and prune unpromising paths. **Uncertainty Representation:** Uncertainty may arise from factors such as stochastic actions, noisy sensors, or incomplete knowledge about the environment.**Applications:** Planning in probabilistic domains, such as robot navigation in uncertain environments, diagnosis and treatment planning in healthcare, and decision-making in autonomous systems.

cTypes of learning explain any one

1. **Supervised Learning:**

- **Definition**: Supervised learning involves learning a mapping from input data to output labels based on a labeled dataset.
- **Training Data**: The training dataset consists of input-output pairs, where the correct output (label) is provided for each input.
- **Objective**: The goal is to learn a function that can accurately predict the output for new, unseen inputs.
- **Examples**: Classification (e.g., spam detection, image recognition) and regression (e.g., predicting house prices, stock prices).

2.Unsupervised Learning:

- Definition: Unsupervised learning involves learning patterns or structures from input data without explicit supervision.
- Training Data: The training dataset consists of input data without corresponding output labels.
- Objective: The goal is to discover hidden patterns, groupings, or relationships within the data.
- Examples: Clustering (e.g., customer segmentation, document clustering) and dimensionality reduction (e.g., principal component analysis, t-distributed stochastic neighbor embedding).

3.Semi-Supervised Learning:

- Definition: Semi-supervised learning is a combination of supervised and unsupervised learning, where the model learns from a small amount of labeled data and a large amount of unlabeled data.
- Training Data: The training dataset includes both labeled and unlabeled data.
- Objective: The goal is to leverage the unlabeled data to improve the performance of the model trained on limited labeled data.
- Examples: Text classification with a small labeled dataset and a large corpus of unlabeled text, image recognition with few labeled images and many unlabeled images

Types of sensors
Camera Sensors:Applications: Object recognition, image classification, facial recognition, gesture recognition, surveillance, autonomous vehicles, robotics, augmented reality (AR), and virtual reality (VR).
Lidar (Light Detection and Ranging) Sensors:
Applications: Distance measurement, 3D mapping, obstacle detection and avoidance (e.g., in autonomous vehicles and drones), environment modeling, and terrain analysis.
Radar (Radio Detection and Ranging) Sensors:
Applications: Object detection and tracking, speed measurement, collision avoidance (e.g., in
autonomous vehicles and drones), weather monitoring, and target recognition in military applications.
Inertial Measurement Unit (IMU):
Applications: Motion sensing, orientation tracking, gesture recognition, activity monitoring (e.g., in fitness trackers), inertial navigation (e.g., in drones and robots), and virtual reality systems.
Microphones and Audio Sensors:
Applications: Speech recognition, voice command systems, noise cancellation, acoustic monitoring (e.g., for security purposes), sound classification, and audio-based user interaction (e.g., in smart home devices).
Pressure Sensors:
Applications: Barometric pressure measurement (for weather forecasting and altitude
estimation), flow rate measurement, altitude tracking (e.g., in drones and aircraft), liquid level
sensing, and touch sensitivity in touchscreens.

Characteristics of environment and explain any one

(1) Fully observable versus partially observable
If the agent sensors describes the environment at each point in detail, then the environment is fully observable. Fully observable environments are convenient because agent need not maintain track of the world. An environment may be partially observable because an automated taxi cannot see what other drivers are thinking. A fully observable environment is also termed as accessible environment,
while partially observable environment is termed as maccessible environment.Playing card-games is a perfect example of partially observable environment where a player is not aware of the card in the opponent's hand.

(2) Static Vs-Dynamic
If the environment can change itself while an agent is deliberating then such environment is called a
dynamic environment, else it is called a static environment. Static environment are easy to deal because an agent does not need to continue looking at the world while deciding for an action.But for dynamic environment, agents need to keep looking at the world at each action.Taxi driving is an example of a dynamic environment whereas playing on Piano is an example of a static environment. If the environment does not change with the passage of time, but the agent
performance changes by time, then it is semi-dynamic.

(**3).Deterministic Vs. Stochastic:**
In a deterministic environment, the outcome of actions taken by the agent is fully determined by the current state of the environment and the action itself.There is no randomness involved in the transition between states or the consequences of actions.Each action taken by the agent leads to a predictable and consistent outcome.In a stochastic environment, there is some degree of randomness or uncertainty in the outcome of actions.The transition between states and the consequences of actions may be probabilistic, meaning that the outcome is not fully determined by the current state and action.Multiple outcomes are possible for the same action taken in the same state, each with a certain probability.

Breadth-First Search (BFS):
BFS explores all nodes at a given depth level before moving to the next depth level.It uses a queue to maintain the order of nodes to be visited (FIFO).BFS guarantees finding the shortest path in unweighted graphs.Suitable for finding shortest paths and exploring neighbor nodes uniformly.Requires more memory due to the need to store nodes at each depth level.
Depth-First Search (DFS):
DFS explores as deeply as possible along each branch before backtracking. It uses a stack (or recursion) to maintain the order of nodes to be visited (LIFO).DFS may not necessarily find the shortest path.Suitable for tasks like topological sorting and maze generation.Typically requires less memory compared to BFS, as it only needs to store the path from the root to the current node

PEAS for automatic taxi driver
1. **Performance measure :** it is the measure of success for an automatic taxi driver it should include factor such as:**(a)**Safety and timely arrival of passengers**(b)** Efficient root planning for minimising travel time**(c)** Customer satisfaction and positive feedback
2. **Environment:** the environment for an automatic taxi driver includes road traffic conditions pedestrians other vehicles City infrastructure. The dynamic and unpredictable nature of traffic environment makes it Complex for taxi to navigate.
3. **Actuators:** these are mechanism through which the agent interacts with environment. actuators for taxi driver include:
(a) Accelerator and brake pedals for controlling speed**(b)** Tearing mechanism for navigation**(c)** Turn Signals and brake lights for communication with others.
4. **Sensors:** these are means by which agent purses and gathers information about environment. for automatic taxi driver sensors include:
(a) GPS for Location tracking and navigation.**(b)** Reader for measuring distance to other objects.**(c)**Microphones and speakers for communication with passengers.