**Introduction to AI, Intelligent Agents, problem-Solving Agents**

**What is Artificial Intelligence (AI)**

**Artificial Intelligence (AI) refers to the technology that allows machines and computers to replicate human intelligence.**

* **Enables systems to perform tasks that require human-like decision-making, such as learning from data, identifying patterns, making informed choices and solving complex problems.**
* **Improves continuously by utilizing methods like machine learning and deep learning.**
* **Used in healthcare for diagnosing diseases, finance for fraud detection, e-commerce for personalized recommendations and transportation for self-driving cars. It also powers virtual assistants like Siri and Alexa, chatbots for customer support and manufacturing robots that automate production processes.**

**How Does AI Work?**

**AI works by simulating human intelligence in machines through algorithms, data and models that enable them to perform tasks that would typically require human intervention. Here's a simplified breakdown:**

1. **Data Collection: AI systems rely on vast amounts of data. This data can come from various sources, like images, texts or sensor readings. For example, if we're building an AI that recognizes cats in images, we'd need a large dataset of labeled images of cats.**
2. **Processing and Learning: Machine learning (ML), a subset of AI, uses algorithms to analyze the data. The system learns patterns from the data by training a model. For instance, an AI system might learn the features of a cat, like its shape, ears and whiskers, by being exposed to thousands of labeled images of cats and non-cats.**
3. **Model Training: The AI model undergoes training using the data. In this process, the model adjusts its parameters based on the input data and the desired output. The more data and training time, the more accurate the model becomes.**
4. **Decision Making: After training, the AI can make decisions or predictions based on new, unseen data. For example, it might predict whether an image contains a cat, based on the patterns it learned from previous training data.**
5. **Feedback and Improvement: In many AI systems, particularly in reinforcement learning, feedback is used to improve performance over time. The system's actions are continuously evaluated and adjustments are made to improve future performance.**

**Types of AI (Artificial Intelligence)**

**AI can be classified into two main categories based on its capabilities and functionalities.**

**Based on Capabilities:**

* [**Narrow AI**](https://www.geeksforgeeks.org/artificial-intelligence/what-is-narrow-ai/)**(Weak AI): This type of AI is designed to perform a specific task or a narrow set of tasks, such as voice assistants or recommendation systems. It excels in one area but lacks general intelligence.**
* [**General AI**](https://www.geeksforgeeks.org/artificial-intelligence/what-is-artificial-general-intelligence-agi/)**(Strong AI): General AI is a theoretical concept where AI can perform any intellectual task that a human can do. It demonstrates human-like reasoning and understanding across multiple domains, making it capable of tackling a wide variety of tasks.**
* [**Superintelligent AI**](https://www.geeksforgeeks.org/blogs/what-is-artificial-super-intelligence-asi/)**: Superintelligent AI is a hypothetical form of AI that would surpass human intelligence in all areas. It would be capable of performing tasks more efficiently and effectively than humans.**

**Based on Functionalities:**

* **Reactive Machines: Reactive machines are AI systems that respond to specific tasks or situations but do not store memories or improve over time. They are programmed to react in a fixed way without learning from past experiences.**
* **Limited Memory: Limited memory AI can store and learn from past experiences to make better decisions in the future. Self-driving cars are an example, as they use historical data to navigate and adapt to changing environments.**
* **Theory of Mind: The theory of mind is a theoretical type of AI that would be able to understand emotions, beliefs, intentions and other mental states. This would allow the AI to interact with humans in a more natural and empathetic manner.**
* **Self-Aware AI: Self-aware AI is a hypothetical form of AI that possesses consciousness and self-awareness. It would have an understanding of its own existence and could make decisions based on that awareness.**

**AI Models**

[**AI models**](https://www.geeksforgeeks.org/artificial-intelligence/common-ai-models-and-when-to-use-them/)**are computer programs that learn to perform tasks by recognizing patterns in data, similar to how our brains learn from experience. They are trained on large datasets and then use what they’ve learned to make decisions, whether it’s identifying faces in a photo, translating languages or generating text.**

**There are different kinds of AI models based on how they learn:**

**1. Supervised Learning Models**

**In**[**Supervised learning**](https://www.geeksforgeeks.org/machine-learning/supervised-machine-learning/)**, the AI is provided with a set of examples where both the input and the desired output are known. For example, to teach an AI to recognize handwritten numbers, we would show it many images of handwritten digits, each labeled with the correct number (0-9). Over time, the model adjusts its internal settings (called weights) to minimize the difference between its predictions and the correct labels given by the "teacher." This method works well when you have large amounts of high-quality, labeled data and is commonly used for tasks like image classification, speech recognition and spam detection.**

**2. Unsupervised Learning Models**

**In**[**Unsupervised Learning**](https://www.geeksforgeeks.org/machine-learning/unsupervised-learning/)**models, the AI is given input data without labels or explicit instructions on what to look for. Its task is to find hidden patterns, clusters, or structures on its own. For instance, if you give an unsupervised model a collection of news articles, it might automatically group them into categories like sports, politics, or entertainment, without anyone telling it those categories. This type of learning is helpful for uncovering new insights in data, reducing dimensions for visualization, and spotting unusual patterns, such as fraud or other anomalies.**

**3. Reinforcement Learning Models**

[**Reinforcement learning**](https://www.geeksforgeeks.org/machine-learning/what-is-reinforcement-learning/)**works differently from the other two methods. In this case, there isn’t a teacher providing the “correct” answer. Instead, the AI learns through a system of rewards and penalties. For example, in a video game, an agent might start by making random movements and gradually learn which actions lead to winning by receiving points or rewards. Over time, the model develops a strategy (or policy) to maximize its rewards. This type of learning is used in fields like robotics, game-playing (such as AlphaGo), and even automated trading systems.**

**Benefits of AI**

**The widespread use of Artificial Intelligence (AI) has brought numerous advantages across various sectors and aspects of our daily lives. Here are some of the primary benefits of AI:**

1. **Efficiency and Automation: AI can automate repetitive tasks, reducing human error and saving time. This leads to increased productivity and allows humans to focus on more complex tasks.**
2. **Improved Decision Making: AI can analyze vast amounts of data quickly and provide insights, helping businesses and organizations make better, data-driven decisions.**
3. **Personalization: AI can be used to offer personalized experiences in areas like retail, entertainment and online services, improving user satisfaction. For example, recommendation systems on platforms like Netflix or Amazon suggest products or content based on individual preferences.**
4. **24/7 Availability: Unlike humans, AI systems can operate around the clock without breaks. This is particularly useful in customer support, monitoring and other services that require constant attention.**
5. **Data Analysis and Pattern Recognition: AI excels at processing large datasets and recognizing patterns that may be difficult for humans to identify. This is especially beneficial in fields like healthcare, finance and marketing.**

**Agents in AI**

**An AI agent is a *software program that can interact with its surroundings, gather information, and use that information to complete tasks on its own to achieve goals set by humans.***

* **For instance, an**[**AI agent**](https://www.geeksforgeeks.org/artificial-intelligence/agents-artificial-intelligence/)**on an online shopping platform can recommend products, answer customer questions, and process orders. If agent needs more information, it can ask users for additional details.**
* **AI agents employ advanced**[**natural language processing**](https://www.geeksforgeeks.org/nlp/introduction-to-natural-language-processing/)**and**[**machine learning**](https://www.geeksforgeeks.org/machine-learning/ml-machine-learning/)**techniques to understand user input, interact step-by-step, and use external tools when needed for accurate responses.**
* **Common AI Agent Applications are *software development and IT automation, coding tools, chat assistants, and online shopping platforms*.**

**How do AI Agents Work?**

**AI agents follow a structured process to perceive, analyze, decide, and act within their environment. Here’s an overview of how AI agents operate:**

**1. Collecting Information (Perceiving the Environment)**

**AI agents gather information from their surroundings through various means:**

* **Sensors: For example, a self-driving car uses cameras and radar to detect objects.**
* **User Input: Chatbots read text or listen to voice commands.**
* **Databases & Documents: Virtual assistants search records or knowledge bases for relevant data.**

**2. Processing Information & Making Decisions**

**After gathering data, AI agents analyze it and decide what to do next. Some agents rely on pre-set rules, while others utilize machine learning to predict the best course of action. Advanced agents may also use**[**retrieval-augmented generation (RAG)**](https://www.geeksforgeeks.org/nlp/what-is-retrieval-augmented-generation-rag/)**to access external databases for more accurate responses.**

**3. Taking Action (Performing Tasks)**

**Once an agent makes a decision, it performs the required task, such as:**

* **Answering a customer query in a chatbot.**
* **Controlling a device, like a smart assistant turning off lights.**
* **Running automated tasks, such as processing orders on an online store.**

**4. Learning & Improving Over Time**

**Some AI agents can learn from past experiences to improve their responses. This self-learning process, often referred to as**[**reinforcement learning**](https://www.geeksforgeeks.org/machine-learning/what-is-reinforcement-learning/)**, allows agents to refine their behavior over time.**

**Architecture of AI Agents**

**The architecture of AI agents serves as the blueprint for how they function.**

**There are four main components in an AI agent’s architecture:**

* **Profiling Module: This module helps the agent understand its role and purpose. It gathers information from the environment to form perceptions.  
  *Example:* A self-driving car uses sensors and cameras to detect obstacles.**
* **Memory Module: The memory module enables the agent to store and retrieve past experiences. This helps the agent learn from prior actions and improve over time.  
  *Example:* A chatbot remembers past conversations to give better responses.**
* **Planning Module: This module is responsible for decision-making. It evaluates situations, weighs alternatives, and selects the most effective course of action.  
  *Example:*A chess-playing AI plans its moves based on future possibilities.**
* **Action Module: The action module executes the decisions made by the planning module in the real world. It translates decisions into real-world actions.  
  *Example:*A robot vacuum moves to clean a designated area after detecting dirt.**

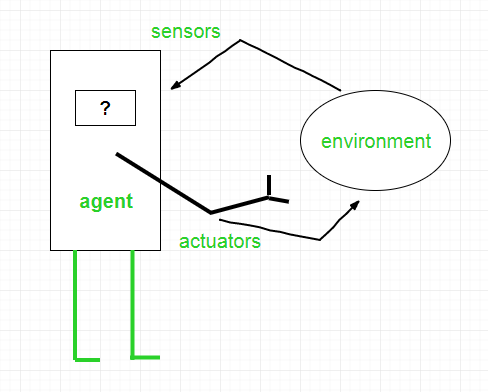
**AI Agent Classification**

**An agent is a system designed to perceive its environment, make decisions, and take actions to achieve specific goals. Agents operate autonomously, without direct human control, and can be classified based on their behavior, environment, and number of interacting agents.**

* **Reactive Agents respond to immediate stimuli in their environment, making decisions based on current conditions without planning ahead.**
* **Proactive Agents take initiative, planning actions to achieve long-term goals by anticipating future conditions.**
* **Fixed Environments have stable rules and conditions, allowing agents to act based on static knowledge.**
* **Dynamic Environments are constantly changing, requiring agents to adapt and respond to new situations in real-time.**
* **Single-Agent Systems involve one agent working independently to solve a problem or achieve a goal.**
* **Multi-Agent Systems involve multiple agents that collaborate, communicate, and coordinate to achieve a shared objective.**
* **Rational agent is one that chooses actions based on the goal of achieving the best possible outcome, considering both past and present information.**

**Key Components of an AI System**

* **An AI system includes the agent, which perceives the environment through sensors and acts using actuators, and the environment, in which it operates.**

****

**Interaction of Agents with the Environment**

**Structure of an AI Agent**

**The structure of an AI agent is composed of two key components: Architecture and Agent Program. Understanding these components is essential to grasp how intelligent agents function.**

**1. Architecture**

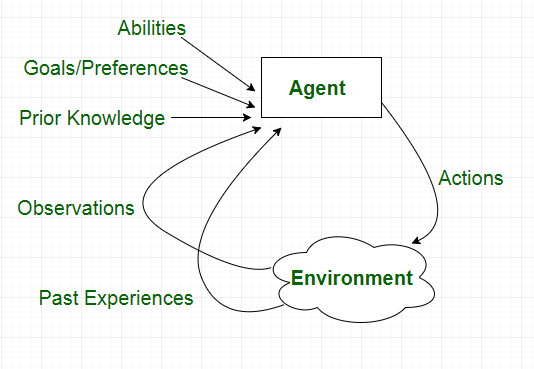
**Architecture refers to the underlying hardware or system on which the agent operates. It is the "machinery" that enables the agent to perceive and act within its environment. Examples of architecture include devices equipped with sensors and actuators, such as a robotic car, camera, or a PC. These physical components enable the agent to gather sensory input and execute actions in the world.**

**2. Agent Program**

**Agent Program is the software component that defines the agent's behavior. It implements the agent function, which is a mapping from the agent's percept sequence (the history of all perceptions it has gathered so far) to its actions. The agent function determines how the agent will respond to different inputs it receives from its environment.**

***Agent = Architecture + Agent Program***

***The overall structure of an AI agent can be understood as a combination of both the architecture and the agent program. The architecture provides the physical infrastructure, while the agent program dictates the decision-making and actions of the agent based on its perceptual inputs.***

****

**Characteristics of an Agent**

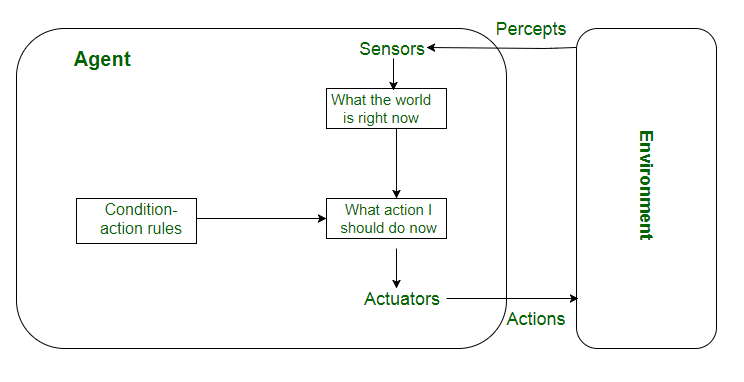
**Types of Agents**

**1. Simple Reflex Agents**

**Simple reflex agents act solely based on the current percept and, percept history (record of past perceptions) is ignored by these agents. Agent function is defined by condition-action rules.**

**A condition-action rule maps a state (condition) to an action.**

* **If the condition is true, the associated action is performed.**
* **If the condition is false, no action is taken.**

**Simple Reflex Agents**

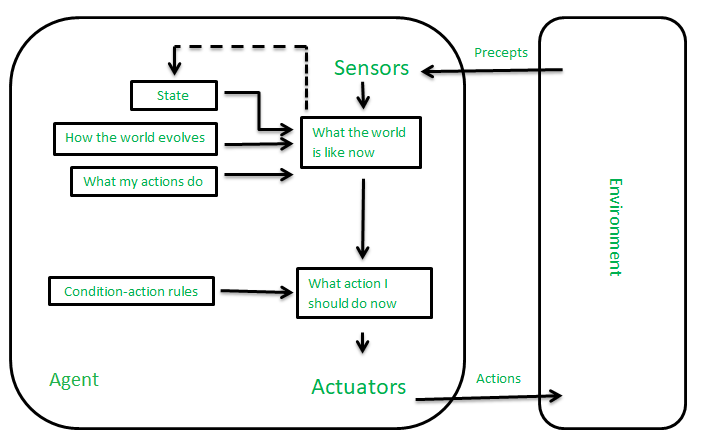
**2. Model-Based Reflex Agents**

**Model-based reflex agents finds a rule whose condition matches the current situation or percept. It uses a model of the world to handle situations where the environment is only partially observable.**

* **The agent tracks its internal state, which is adjusted based on each new percept.**
* **The internal state depends on the percept history (the history of what the agent has perceived so far).**

**The agent stores the current state internally, maintaining a structure that represents the parts of the world that cannot be directly seen or perceived. The process of updating the agent’s state requires information about:**

* **How the world evolves independently from the agent?**
* **How the agent's actions affect the world?**

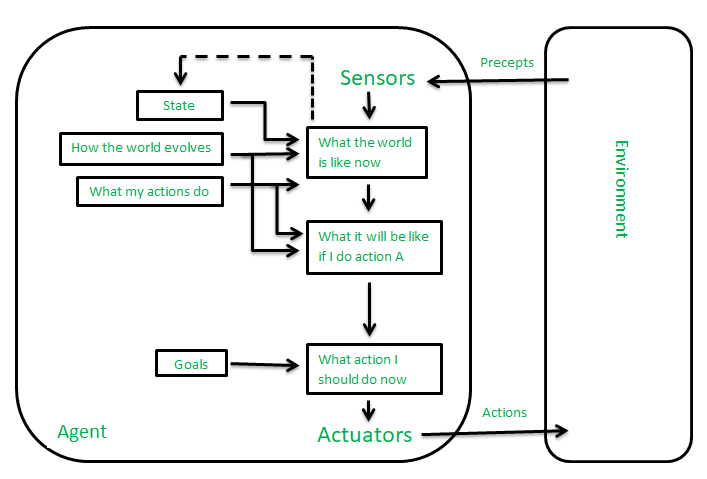
**Model-Based Reflex Agents**

**3. Goal-Based Agents**

**Goal-based agents make decisions based on their current distance from the goal and every action the agent aims to reduce the distance from goal. They can choose from multiple possibilities, selecting the one that best leads to the goal state.**

* **Knowledge that supports the agent's decisions is represented explicitly, meaning it's clear and structured. It can also be modified, allowing for adaptability.**
* **The ability to modify the knowledge makes these agents more flexible in different environments or situations.**

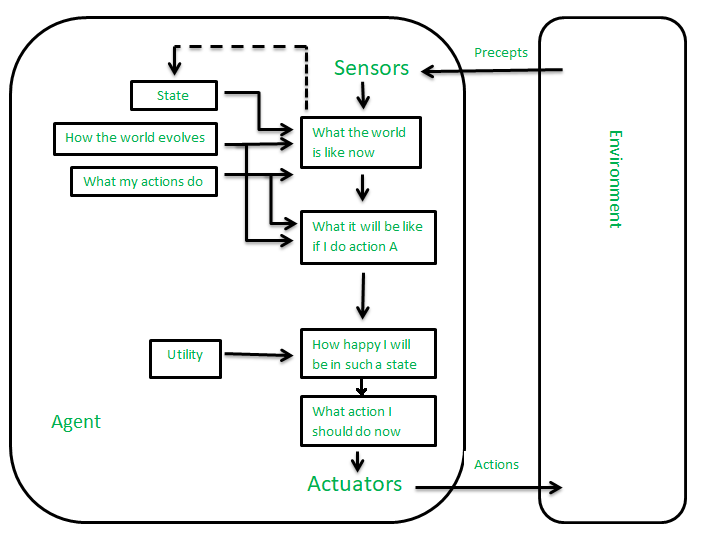
**Goal-based agents typically require search and planning to determine the best course of action.**

**Goal-Based Agents**

**4. Utility-Based Agents**

**Utility-based agents are designed to make decisions that optimize their performance by evaluating the preferences (or utilities) for each possible state. These agents assess multiple alternatives and choose the one that maximizes their utility, which is a measure of how desirable or "happy" a state is for the agent.**

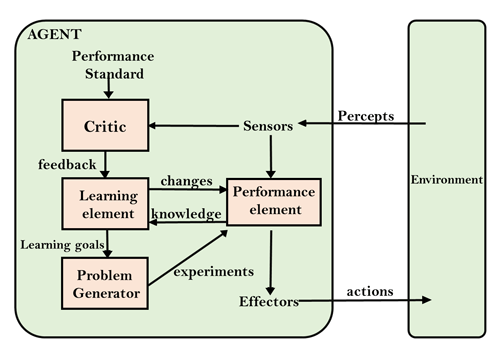
* **Achieving the goal is not always sufficient; for example, the agent might prefer a quicker, safer, or cheaper way to reach a destination.**
* **The utility function is essential for capturing this concept, mapping each state to a real number that reflects the agent’s happiness or satisfaction with that state.**

**Utility-Based Agents**

**5. Learning Agent**

**A learning agent in AI is the type of agent that can learn from its past experiences or it has learning capabilities. It starts to act with basic knowledge and then is able to act and adapt automatically through learning. A learning agent has mainly four conceptual components, which are:**

1. **Learning element: It is responsible for making improvements by learning from the environment.**
2. **Critic: The learning element takes feedback from critics which describes how well the agent is doing with respect to a fixed performance standard.**
3. **Performance element: It is responsible for selecting external action.**
4. **Problem Generator: This component is responsible for suggesting actions that will lead to new and informative experiences.**

****

**Learning Agent**

**6. Multi-Agent Systems**

**Multi-Agent Systems (MAS) consists of multiple interacting agents working together to achieve a common goal. These agents can be autonomous or semi-autonomous, capable of perceiving their environment, making decisions, and taking action.**

**7. Hierarchical Agents**

**Hierarchical Agents are organized into a hierarchy, with high-level agents overseeing the behavior of lower-level agents. The high-level agents provide goals and constraints, while the low-level agents carry out specific tasks. They are useful in complex environments with many tasks and sub-tasks.**

**Uses of Agents**

**Agents are used in a wide range of applications in artificial intelligence, including:**

* **Robotics: Agents can be used to control robots and automate tasks in manufacturing, transportation, and other industries.**
* **Smart homes and buildings: Agents can be used to control heating, lighting, and other systems in smart homes and buildings, optimizing energy use and improving comfort.**
* **Transportation systems: Agents can be used to manage traffic flow, optimize routes for autonomous vehicles, and improve logistics and supply chain management.**
* **Healthcare: Agents can be used to monitor patients, provide personalized treatment plans, and optimize healthcare resource allocation.**
* **Finance: Agents can be used for automated trading, fraud detection, and risk management in the financial industry.**
* **Games: Agents can be used to create intelligent opponents in games and simulations, providing a more challenging and realistic experience for players.**

**Overall, agents are a versatile and powerful tool in artificial intelligence that can help solve a wide range of problems in different fields.**

**Problem Solving in Artificial Intelligence**

**Problem solving is a fundamental concept in**[**artificial intelligence (AI)**](https://www.geeksforgeeks.org/artificial-intelligence/what-is-artificial-intelligence-ai/)**where systems are designed to identify challenges, make decisions and find efficient solutions. AI uses agents which are systems that perceive their environment and take actions to achieve specific goals. They go beyond simple reflex agents which react to stimuli based on pre-defined rules. Instead, problem-solving agents actively analyze situations, evaluate different options and choose the best action to reach their goal.**

**These agents work by:**

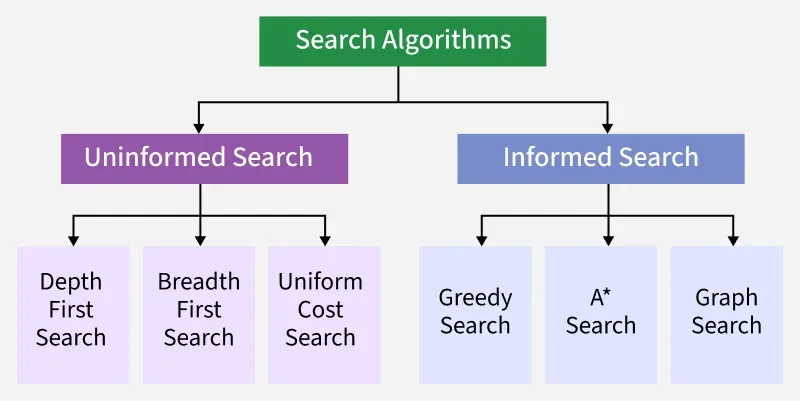
* **Perceiving the environment: They collect data about their surroundings such as sensor inputs or observations.**
* **Defining the problem: They clearly understand the problem including the starting point, the available actions and the desired goal.**
* **Exploring different possibilities: They consider various ways to solve the problem and evaluate which approach is likely to succeed.**
* **Evaluating and deciding: Once they explore options, they assess the outcomes and pick the best course of action based on factors like time, resources and success likelihood.**
* **Learning and adapting: Many problem-solving agents can learn from past experiences, improving their decision-making abilities over time.**

**Search Algorithms in AI**

**Search algorithms in AI help find solutions by exploring possible paths or options in a problem space. AI uses them in tasks like pathfinding, decision making and game playing. These algorithms work by searching through a set of possibilities to reach a goal, either blindly without extra information or with guidance using heuristics.**

**There are mainly 2 types of search algorithms i.e Uninformed Search Algorithms and Informed Search Algorithms.**

**Types of search algorithms**

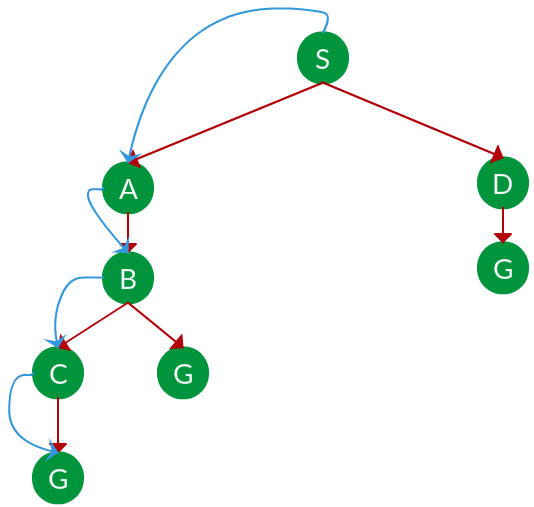
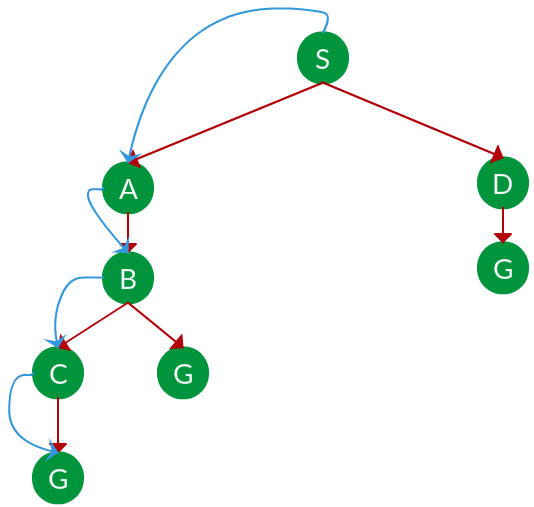
**Uninformed Search Algorithms**

[**Uninformed search**](https://www.geeksforgeeks.org/artificial-intelligence/uniformed-search-algorithms-in-ai/)**also called blind search explores the search space without any domain specific knowledge or heuristics. It treats all nodes equally and chooses which path to explore next based solely on general rules like node depth or path cost.**

**1. Depth First Search**

* [**Depth First Search**](https://www.geeksforgeeks.org/dsa/depth-first-search-or-dfs-for-a-graph/)**explores paths by going as deep as possible along one direction before backtracking. It uses a stack or recursion to keep track of the path.**
* **DFS is memory efficient compared to BFS since it doesn’t need to store all siblings at each level.**
* **However it is not guaranteed to find the shortest path and may get stuck in an infinite loop if the search tree is deep or contains cycles unless depth limits or visited checks are applied.**

**For Example: Which solution would DFS find to move from node S to node G if run on the graph below.**

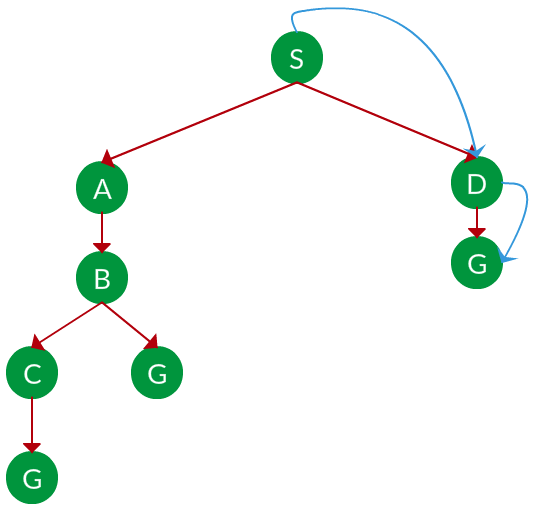
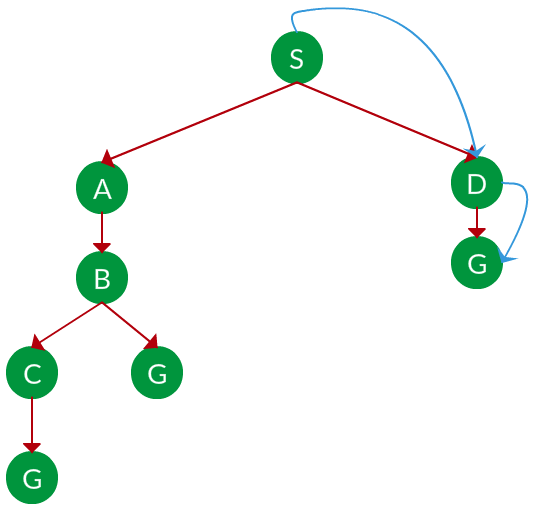


* As DFS traverses the tree deepest node first it would always pick the deeper branch until it reaches the solution or it runs out of nodes and goes to the next branch.
* **Path**: S->A->B->C->G

**2. Breadth First Search**

* [**Breadth First Search**](https://www.geeksforgeeks.org/artificial-intelligence/breadth-first-search-bfs-for-artificial-intelligence/)**is a fundamental search algorithm that explores all possible paths level by level. It begins from the root node and explores all neighboring nodes before moving to the next level of nodes.**
* **BFS is complete and guarantees finding the shortest path if each move has the same cost.**
* **However its main drawback is high memory usage as it stores all nodes at the current level before moving deeper which can grow rapidly for large or complex problems.**

**For Example: Which solution would BFS find to move from node S to node G if run on the graph below.**

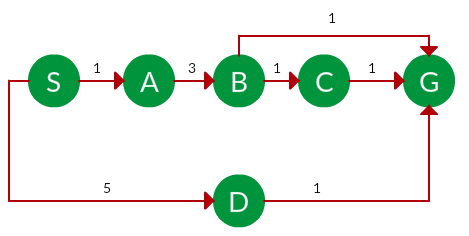
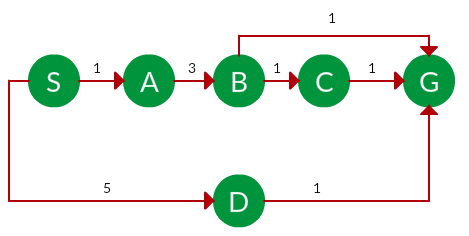


* As BFS traverses the tree shallowest node first it would always pick the shallower branch until it reaches the solution or it runs out of nodes and goes to the next branch.
* **Path**: S->D->G

**3. Uniform Cost Search**

* [**Uniform Cost Search**](https://www.geeksforgeeks.org/artificial-intelligence/uniform-cost-search-ucs-in-ai/)**is similar to BFS but takes the cost of each move into account. It always expands the node with the lowest cumulative path cost from the start.**
* **This makes UCS optimal and complete and useful when actions have different costs such as in navigation systems.**
* **It uses a priority queue to manage the frontier and ensures the cheapest path is always chosen next.**

**For Example:**Which solution would UCS find to move from node S to node G if run on the graph below.



* **The cost of each node is the cumulative cost of reaching that node from the root and based on the UCS strategy the path with the least cumulative cost is chosen.**
* **Path:** S->A->B->G

**Iterative Deepening Search:**

[**Iterative Deepening Search (IDS)**](https://www.geeksforgeeks.org/dsa/iterative-deepening-searchids-iterative-deepening-depth-first-searchiddfs/)**is a search algorithm used in AI that blends the completeness of**[**Breadth-First Search (BFS)**](https://www.geeksforgeeks.org/artificial-intelligence/breadth-first-search-bfs-for-artificial-intelligence/)**with the space efficiency of**[**Depth-First Search (DFS)**](https://www.geeksforgeeks.org/artificial-intelligence/depth-first-search-dfs-for-artificial-intelligence/)**.**

**IDS explores a graph or a tree by progressively increasing the depth limit with each iteration, effectively performing a series of DFS operations until the goal node is found.**

**This approach is particularly advantageous when the depth of the solution is unknown, and we aim to achieve both optimality and completeness without excessive memory usage.**

**How Iterative Deepening Search Works**

**The core concept of IDS revolves around repeatedly running a depth-limited DFS up to increasing depth levels. It starts with a depth limit of zero, then increments this limit iteratively. Each iteration performs a DFS search up to the current depth limit.**

**Here’s a step-by-step breakdown of the algorithm:**

1. **Start at the Root Node: Begin the search from the root node (or starting point).**
2. **Perform DFS with Depth Limit (L): In each iteration, perform a DFS with a depth limit L.**
3. **Increment Depth: After each iteration, increment L by 1.**
4. **Repeat: Continue this process until the goal node is found or the search space is exhausted.**

**Bidirectional Search in AI**

**Bidirectional search in AI is an algorithm that searches simultaneously from both the initial state and the goal state, meeting in the middle to reduce search time. The aim of the article is to provide an in-depth understanding of the bidirectional search algorithm, its working mechanism, benefits and practical applications in artificial intelligence.**

**How Bidirectional Search Works?**

**Bidirectional search uses two simultaneous searches to potentially reduce the total search time. Here’s a step-by-step breakdown of how it typically works:**

1. **Initial Setup: Initialize two searches. One starts from the initial state and expands forward. The other starts from the goal state and expands backward.**
2. **Node Expansion: Both searches alternately expand the nearest unexplored node. For each node, all possible successors (in the forward direction) or predecessors (in the backward direction) are generated.**
3. **Checking Intersections: After each expansion, check if any of the newly generated nodes are present in the frontier of the opposite search.**
4. **Meeting Point: Once a common node is discovered, this node acts as the meeting point and the optimal path can be constructed by joining the paths from the initial state to the meeting point and from the meeting point to the goal state.**

**Informed Search Algorithms**

[**Informed search**](https://www.geeksforgeeks.org/artificial-intelligence/informed-search-algorithms-in-artificial-intelligence/)**uses domain knowledge in the form of heuristics to make smarter decisions during the search process. These heuristics estimate how close a state is to the goal guiding the search more efficiently.**

**Heuristic Function**

**Heuristic functions are essential in AI search algorithms, helping estimate the cost from a current state to the goal. Instead of exhaustively exploring all possibilities, heuristics guide the search by narrowing down the most promising paths. This makes problem-solving more efficient, especially in complex scenarios where exact solutions are too costly.**

**By providing informed estimates, heuristic functions break down large problems into manageable subproblems which is widely used in AI planning and decision-making.**

**Heuristic Search Algorithms**

**Heuristic search algorithms uses heuristic functions to make more intelligent decisions during the search process. Some common heuristic search algorithms include:**

**1. A\* Algorithm**

**The**[**A\* algorithm**](https://www.geeksforgeeks.org/dsa/a-search-algorithm/)**is one of the most widely used heuristic search algorithms. It uses both the actual cost from the start node to the current node (g(n)) and the estimated cost from the current node to the goal (h(n)). The total estimated cost (f(n)) is the sum of these two values:**

***f(n)=g(n)+h(n)***

**Where:**

* **f(n) is total estimated cost of the cheapest solution through n.**
* **g(n) is actual cost from the start node to the node n.**
* **h(n) is estimated cost from node n to goal.**

**Note: This guarantees optimal path if h(n) never overestimates.**

**2. Greedy Best-First Search**

**The**[**Greedy Best-First Search**](https://www.geeksforgeeks.org/dsa/greedy-best-first-search-algorithm/)**algorithm selects the path that appears to be the most promising based on the heuristic function alone. It prioritizes nodes with the lowest heuristic cost (h(n)) but it does not necessarily guarantee the shortest path to the goal.**

***f(n)=h(n)***

* **This only considers the heuristic estimate to the goal.**
* **Ignores the actual cost already traveled (g(n)).**
* **Fast but may take suboptimal paths.**

**Hill Climbing**

**Hill climbing is a**[**heuristic search algorithm**](https://www.geeksforgeeks.org/artificial-intelligence/heuristic-search-techniques-in-ai/)**that belongs to the family of local search methods. It is designed to solve problems where the goal is to find an optimal (or near-optimal) solution by iteratively moving from the current state to a better neighboring state, according to a heuristic or evaluation function.**

* **It is an optimisation technique used in artificial intelligence (AI) to find solutions for a wide variety of problems.**
* **It operates on the principle of incrementally improving a solution by making local changes and evaluating their merit.**
* **Its simplicity, intuitive logic and adaptability to different problems make it a go-to method.**

**Hill Climbing Algorithms**

**Hill climbing follows these steps:**

1. **Initial State: Start with an arbitrary or random solution (initial state).**
2. **Neighboring States: Identify neighboring states of the current solution by making small adjustments (mutations or tweaks).**
3. **Move to Neighbor: If one of the neighboring states offers a better solution (according to some evaluation function), move to this new state.**
4. **Termination: Repeat this process until no neighboring state is better than the current one. At this point, we have reached a local maximum or minimum.**

**Features of Hill Climbing Algorithm**

**1. Variant of Generating and Testing Algorithm: Hill Climbing is a specific variant of the**[**generating and testing algorithms**](https://www.geeksforgeeks.org/machine-learning/generate-and-test-search/)**. The process involves: This iterative feedback mechanism allows Hill Climbing to refine its search by using information from previous evaluations to inform future moves in the search space.**

* **Generating possible solutions: The algorithm creates potential solutions within the search space.**
* **Testing solutions: Each generated solution is evaluated to determine if it meets the desired criteria.**
* **Iteration: If a satisfactory solution is found, the algorithm terminates; otherwise, it returns to the generation step.**

**Types of Hill Climbing**

**1. Simple Hill Climbing Algorithm: Simple Hill Climbing is a straightforward variant of hill climbing where the algorithm evaluates each neighbouring node one by one and selects the first node that offers an improvement over the current one.**

**2. Steepest-Ascent Hill Climbing: Steepest-Ascent Hill Climbing is an enhanced version of simple hill climbing. Instead of moving to the first neighboring node that improves the state, it evaluates all neighbors and moves to the one offering the highest improvement (steepest ascent).**

**3. Stochastic Hill Climbing: Stochastic Hill Climbing introduces randomness into the search process. Instead of evaluating all neighbors or selecting the first improvement, it selects a random neighboring node and decides whether to move based on its improvement over the current state.**

**What is Simulated Annealing?**

**Simulated Annealing is an optimization algorithm designed to search for an optimal or near-optimal solution in a large solution space. The name and concept are derived from the process of annealing in metallurgy, where a material is heated and then slowly cooled to remove defects and achieve a stable crystalline structure. In Simulated Annealing, the "heat" corresponds to the degree of randomness in the search process, which decreases over time (cooling schedule) to refine the solution. The method is widely used in combinatorial optimization, where problems often have numerous local optima that standard techniques like gradient descent might get stuck in. Simulated Annealing excels in escaping these local minima by introducing controlled randomness in its search, allowing for a more thorough exploration of the solution space.**

**How Simulated Annealing Works**

**The algorithm starts with an initial solution and a high "temperature," which gradually decreases over time. Here’s a step-by-step breakdown of how the algorithm works:**

* **Initialization: Begin with an initial solution Sο, and an initial temperature *Tο*. The temperature controls how likely the algorithm is to accept worse solutions as it explores the search space.**
* **Neighborhood Search: At each step, a new solution *S*′ is generated by making a small change (or perturbation) to the current solution S.**
* **Objective Function Evaluation: The new solution S' is evaluated using the objective function. If S' provides a better solution than S, it is accepted as the new solution.**
* **Acceptance Probability: If S' is worse than S, it may still be accepted with a probability based on the temperature and the difference in objective function values.**

**The acceptance probability is given by:**

***P(accept)=e−ΔETP(accept)=e−ΔE/T​***

* **Cooling Schedule: After each iteration, the temperature is decreased according to a predefined cooling schedule, which determines how quickly the algorithm converges. Common cooling schedules include linear, exponential, or logarithmic cooling.**
* **Termination: The algorithm continues until the system reaches a low temperature (i.e., no more significant improvements are found), or a predetermined number of iterations is reached.**

**Local Search Algorithm**

**Local search algorithms are important in artificial intelligence as they can quickly find good answers, especially when finding the perfect solution would take too long or too much effort. They are useful for big or complex problems where checking every possible option isn't practical.**

* **It focus only on the current solution and the ones directly related to it rather than looking everywhere.**
* **Ideal for real-world tasks like puzzles, timetables or route finding.**

**Working of Local Search Algorithms**

**Step 1: Pick a starting point: Start with a possible solution which is often random but sometimes based on rule.**

**Step 2: Find the neighbors:**

* **Neighbors are similar solutions we can get by making small, simple changes to the current one.**
* **For example, in a puzzle, swapping two pieces creates a neighbor.**

**Step 3: Compare: Look around at all neighbors to see if any are better.**

**Step 4: Move: If a better neighbor exists, move to it, making it our new “current” solution.**

**Step 5: Repeat: Keep searching from the new point, following the same steps.**

**Step 6: Stop: When none of the neighbors are better or after enough tries.**