Artificial Intelligence

Unit – 1

Basic Concept of Artificial Intelligence

In today's world, technology is growing very fast, and we are getting in touch with different new technologies day by day.

Here, one of the booming technologies of computer science is Artificial Intelligence which is ready to create a new revolution in the world by making intelligent machines. The Artificial Intelligence is now all around us. It is currently working with a variety of subfields, ranging from general to specific, such as self-driving cars, playing chess, proving theorems, playing music, Painting, etc.

Artificial Intelligence is composed of two words **Artificial** and **Intelligence**, where Artificial defines "*man-made,*" and intelligence defines "*thinking power*", hence AI means "*a man-made thinking power.*"

So, we can define AI as:

"It is a branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and able to make decisions."

<u>History of Artificial Intelligence</u>

Artificial Intelligence is not a new word and not a new technology for researchers. This technology is much older than you would imagine. Following are some milestones in the history of AI which defines the journey from the AI generation to till date development.

The birth of Artificial Intelligence (1952-1956)

- **Year 1955:** An Allen Newell and Herbert A. Simon created the "first artificial intelligence program “Which was named as **"Logic Theorist"**. This program had proved 38 of 52 Mathematics theorems, and find new and more elegant proofs for some theorems.
- **Year 1956:** The word "Artificial Intelligence" first adopted by American Computer scientist John McCarthy at the Dartmouth Conference. For the first time, AI coined as an academic field.

At that time high-level computer languages such as FORTRAN, LISP, or COBOL were invented. And the enthusiasm for AI was very high at that time.

The golden years-Early enthusiasm (1956-1974)

- **Year 1966:** The researchers emphasized developing algorithms which can solve mathematical problems. Joseph Weizenbaum created the first chatbot in 1966, which was named as ELIZA.
- **Year 1972:** The first intelligent humanoid robot was built in Japan which was named as WABOT-1.

The first AI winter (1974-1980)

- The duration between years 1974 to 1980 was the first AI winter duration. AI winter refers to the time period where computer scientist dealt with a severe shortage of funding from government for AI researches.
- During AI winters, an interest of publicity on artificial intelligence was decreased.

A boom of AI (1980-1987)

- **Year 1980:** After AI winter duration, AI came back with "Expert System". Expert systems were programmed that emulate the decision-making ability of a human expert.
- In the Year 1980, the first national conference of the American Association of Artificial Intelligence **was held at Stanford University**.

The second AI winter (1987-1993)

- The duration between the years 1987 to 1993 was the second AI Winter duration.
- Again Investors and government stopped in funding for AI research as due to high cost but not efficient result. The expert system such as XCON was very cost effective.

The emergence of intelligent agents (1993-2011)

- **Year 1997:** In the year 1997, IBM Deep Blue beats world chess champion, Gary Kasparov, and became the first computer to beat a world chess champion.
- **Year 2002:** for the first time, AI entered the home in the form of Roomba, a vacuum cleaner.
- **Year 2006:** AI came in the Business world till the year 2006. Companies like Facebook, Twitter, and Netflix also started using AI.

Deep learning, big data and artificial general intelligence (2011-present)

- **Year 2011:** In the year 2011, IBM's Watson won jeopardy, a quiz show, where it had to solve the complex questions as well as riddles. Watson had proved that it could understand natural language and can solve tricky questions quickly.

- **Year 2012:** Google has launched an Android app feature "Google now", which was able to provide information to the user as a prediction.
- **Year 2014:** In the year 2014, Chatbot "Eugene Goostman" won a competition in the infamous "Turing test."
- **Year 2018:** The "Project Debater" from IBM debated on complex topics with two master debaters and also performed extremely well.
- Google has demonstrated an AI program "Duplex" which was a virtual assistant and which had taken hairdresser appointment on call, and lady on other side didn't notice that she was talking with the machine.

Now AI has developed to a remarkable level. The concept of Deep learning, big data, and data science are now trending like a boom. Nowadays companies like Google, Facebook, IBM, and Amazon are working with AI and creating amazing devices. The future of Artificial Intelligence is inspiring and will come with high intelligence.

Current Status of Artificial Intelligence

The current status of AI is in the Theory of Mind State.

In Theory of Mind AI a machine is able to decide like any human. It seems that the machine have a mind in it. It can think emotionally. Many data from the surrounding is saved in the machine. Machine analyzes these data and show a output in a certain condition based on those data.

Summary of the Current Status of AI

- Technologies are more enhanced than previous time. Any technological solution is available to the industries.
- Voice assistants helps in many work such as in IT, automotive and retail industries.
- Chatbots are part of Current Status of AI. Which have reduced the hassle of replying huge common messages manually.
- AI technology has made to deploy any mobile devices and apps in one of the easiest ways across industries.
- Cloud Service has been more easier in the current status of AI.
- Emerging AI technologies include augmented intelligence such as some forms of face recognition, which seeks to enhance human intelligence and edge AI, where AI algorithms are processed locally without the need for an internet connection.

Future of AI

The Future state of AI is Self Aware AI. Where the machine will be aware before any incident happens. If any incident is seems to be occur AI will provide a signal from the previous experience, knowledge and saved data.

Self Aware AI is not established yet. So, this will be the future status of AI.

Types of Artificial Intelligence:

1. Reactive AI

The most basic type of artificial intelligence is reactive AI, which is programmed to provide a predictable output based on the input it receives. Reactive machines always respond to identical situations in the exact same way every time, and they are not able to learn actions or conceive of past or future.

Examples of reactive AI include:

- Deep Blue, the chess-playing IBM supercomputer that bested world champion Garry Kasparov
- Spam filters for our email that keeps promotions and phishing attempts out of our inboxes
- The Netflix recommendation engine.

2. Limited Memory

Limited memory AI learns from the past and builds experiential knowledge by observing actions or data. This type of AI uses historical, observational data in combination with pre-programmed information to make predictions and perform complex classification tasks. It is the most widely-used kind of AI today.

For example, autonomous vehicles use limited memory AI to observe other cars' speed and direction, helping them "read the road" and adjust as needed. This process for understanding and interpreting incoming data makes them safer on the roads.

3. Theory of Mind AI

Want to hold a meaningful conversation with an emotionally intelligent robot that looks and sounds like a real human being? That's on the horizon with theory of mind AI.

With this type of AI, machines will acquire true decision-making capabilities that are similar to humans. Machines with theory of mind AI will be able to understand and remember emotions, then adjust behavior based on those emotions as they interact with people.

There are still a number of hurdles to achieving theory of mind AI, because the process of shifting behavior based on rapidly shifting emotions is so fluid in human communication. It is difficult to mimic as we try to create more and more emotionally intelligent machines.

That said, we are making progress. The **Kismet** robot head, developed by Professor Cynthia Breazeal, could recognize emotional signals on human faces and replicate those emotions on its own face. Humanoid robot Sophia, developed by Hanson Robotics in Hong Kong, can recognize faces and respond to interactions with her own facial expressions.

4. Self-aware AI

The most advanced type of artificial intelligence is self-aware AI. When machines can be aware of their own emotions, as well as the emotions of others around them, they will have a level of consciousness and intelligence similar to human beings. This type of AI will have desires, needs, and emotions as well.

Machines with this type of AI will be self-aware of their internal emotions and mental states. They will be able to make inferences (such as “I’m feeling angry because someone cut me off in traffic”) that are not possible with other types of AI.

We haven’t developed this type of sophisticated AI yet and don’t have the hardware or algorithms to support it.

What is an Agent?

An agent can be anything that perceive its environment through sensors and act upon that environment through actuators. An Agent runs in the cycle of **perceiving**, **thinking**, and **acting**. An agent can be:

- **Human-Agent:** A human agent has eyes, ears, and other organs which work for sensors and hand, legs, vocal tract work for actuators.
- **Robotic Agent:** A robotic agent can have cameras, infrared range finder, NLP for sensors and various motors for actuators.
- **Software Agent:** Software agent can have keystrokes, file contents as sensory input and act on those inputs and display output on the screen.

Types of Agents in AI

In artificial intelligence, an agent is a computer program or system that is designed to perceive its environment, make decisions and take actions to achieve a specific goal or set of goals. The agent operates autonomously, meaning it is not directly controlled by a human operator.

1. **Simple Reflex agent:**

- The Simple reflex agents are the simplest agents. These agents take decisions on the basis of the current percepts and ignore the rest of the percept history.
- These agents only succeed in the fully observable environment.
- The Simple reflex agent does not consider any part of percepts history during their decision and action process.
- The Simple reflex agent works on Condition-action rule, which means it maps the current state to action. Such as a Room Cleaner agent, it works only if there is dirt in the room.
- Problems for the simple reflex agent design approach:
  - They have very limited intelligence
  - They do not have knowledge of non-perceptual parts of the current state
  - Mostly too big to generate and to store.
  - Not adaptive to changes in the environment.

2. **Model-based reflex agent**

- The Model-based agent can work in a partially observable environment, and track the situation.
- A model-based agent has two important factors:
  - **Model:** It is knowledge about "how things happen in the world," so it is called a Model-based agent.

- **Internal State:** It is a representation of the current state based on percept history.

- These agents have the model, "which is knowledge of the world" and based on the model they perform actions.
- Updating the agent state requires information about:

   a. How the world evolves
   b. How the agent's action affects the world.

3. **Goal-based agents**

- The knowledge of the current state environment is not always sufficient to decide for an agent to what to do.
- The agent needs to know its goal which describes desirable situations.
- Goal-based agents expand the capabilities of the model-based agent by having the "goal" information.
- They choose an action, so that they can achieve the goal.
- These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not. Such considerations of different scenario are called searching and planning, which makes an agent proactive.

4**. Utility-based agents**

- These agents are similar to the goal-based agent but provide an extra component of utility measurement which makes them different by providing a measure of success at a given state.
- Utility-based agent act based not only goals but also the best way to achieve the goal.
- The Utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action.
- The utility function maps each state to a real number to check how efficiently each action achieves the goals.

5. Learning Agents

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

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

Hence, learning agents are able to learn, analyze performance, and look for new ways to improve the performance.

The Nature of Environments

Some programs operate in the entirely **artificial environment** confined to keyboard input, database, computer file systems and character output on a screen.

In contrast, some software agents (software robots or softbots) exist in rich, unlimited softbots domains. The simulator has a **very detailed, complex environment**. The software agent needs to choose from a long array of actions in real time. A softbot designed to scan the online preferences of the customer and show interesting items to the customer works in the **real** as well as an **artificial** environment.

The most famous **artificial environment** is the **Turing Test environment**, in which one real and other artificial agent are tested on equal ground. This is a very challenging environment as it is highly difficult for a software agent to perform as well as a human.

Turing Test

The success of an intelligent behavior of a system can be measured with Turing Test.

Two persons and a machine to be evaluated participate in the test. Out of the two persons, one plays the role of the tester. Each of them sits in different rooms. The tester is unaware of who is machine and who is a human. He interrogates the questions by typing and sending them to both intelligences, to which he receives typed responses.

This test aims at fooling the tester. If the tester fails to determine machine's response from the human response, then the machine is said to be intelligent.

Properties of Environment

The environment has multifold properties −

- **Discrete / Continuous** − If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete (For example, chess); otherwise it is continuous (For example, driving).
- **Observable / Partially Observable** − If it is possible to determine the complete state of the environment at each time point from the percepts it is observable; otherwise it is only partially observable.
- **Static / Dynamic** − If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.
- **Single agent / Multiple agents** − The environment may contain other agents which may be of the same or different kind as that of the agent.
- **Accessible / Inaccessible** − If the agent's sensory apparatus can have access to the complete state of the environment, then the environment is accessible to that agent.
- **Deterministic / Non-deterministic** − If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is deterministic; otherwise it is non-deterministic.
- **Episodic / Non-episodic** − In an episodic environment, each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions in the

previous episodes. Episodic environments are much simpler because the agent does not need to think ahead.

Review of tree and graph structures

Both **Trees** and **Graphs** are types of non−linear data structures. They are different from each other in the context of their types of connections and loop formation. That means, a tree structure is connected such that it can never have loops, whereas a graph structure follows a network model and may have loops.

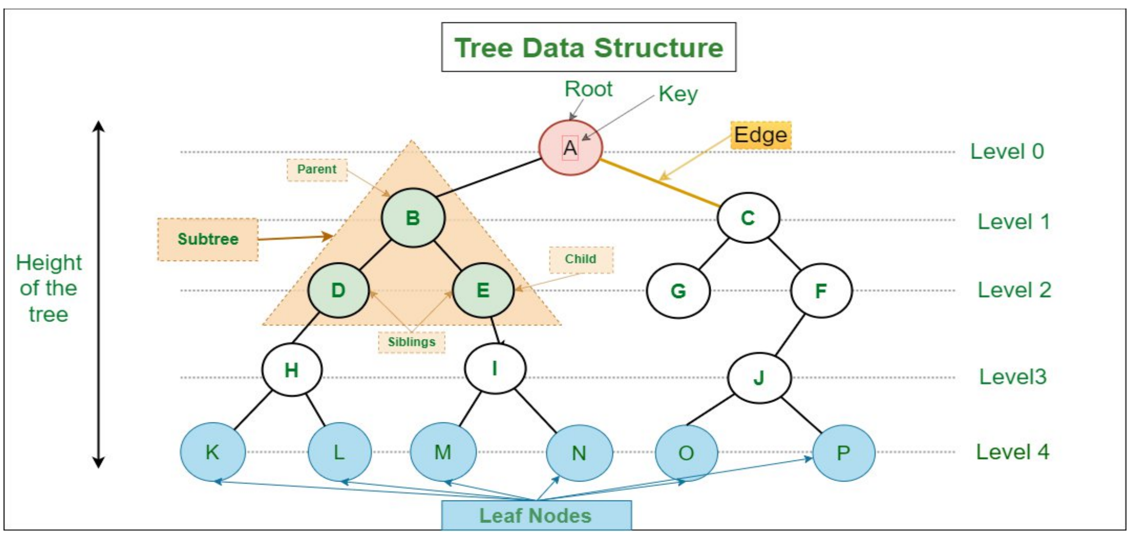
Read this article to find out more about Tress and Graphs and how they are different from each other.

What is Tree?

A **Tree** is a non−linear data structure that is used to represent hierarchy. It is a set of nodes that are joined together to form a hierarchy. In a tree structure, only one path is allowed between two nodes or vertices. The tree structure has an exactly one root node, where the root node is the topmost node in the structure and it does not have any parent node.

Loops are not allowed in a tree structure, therefore it has (n−1) edges, where is the number of nodes. Since a tree structure does not form any loops, it is a hierarchical type model.

There are three traversal techniques used in the tree structure which are pre−order, in−order, and post−order. The tree structure is comparatively a less complex type of non−linear data structure.



Basic Terminologies in Tree Data Structure:

- **Parent Node:** The node which is a predecessor of a node is called the parent node of that node. **{B}** is the parent node of **{D, E}**.

- **Child Node:** The node which is the immediate successor of a node is called the child node of that node. Examples: **{D, E}** are the child nodes of **{B}.**

- **Root Node:** The topmost node of a tree or the node which does not have any parent node is called the root node. {A**}** is the root node of the tree. A nonempty tree must contain exactly one root node and exactly one path from the root to all other nodes of the tree.

- **Leaf Node or External Node:** The nodes which do not have any child nodes are called leaf nodes. **{K, L, M, N, O, P}** are the leaf nodes of the tree.

- **Ancestor of a Node:** Any predecessor nodes on the path of the root to that node are called Ancestors of that node. **{A,B}** are the ancestor nodes of the node **{E}**

- **Descendant:** Any successor node on the path from the leaf node to that node. **{E,I}** are the descendants of the node **{B}.**

- **Sibling:** Children of the same parent node are called siblings. **{D,E}** are called siblings.

- **Level of a node:** The count of edges on the path from the root node to that node. The root node has level **0**.

- **Internal node:** A node with at least one child is called Internal Node.

- **Neighbour of a Node:** Parent or child nodes of that node are called neighbors of that node.

- **Subtree**: Any node of the tree along with its descendant.

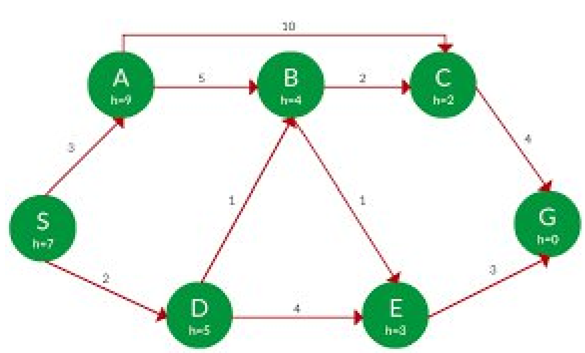
What is Graph?

A **Graph** is also a non−linear data structure used in software engineering. Graphs are used to represent various types of physical structures.

A graph consists of a group of nodes (or vertices) and set of edges. Each edge connects the two nodes. On the graph, the nodes are represented by a point or a circle, and the edges are represented by line segments or arcs.

In a graph structure, more than one path is allowed between vertices. Graphs can also have loops; therefore, they do not have a root node. Graphs follow the network model.

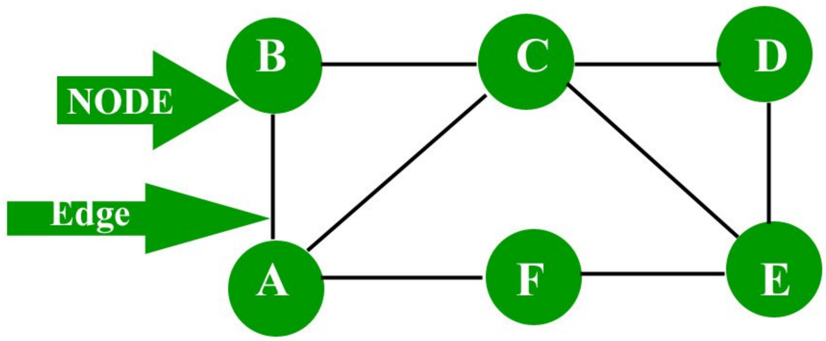
There are two traversal techniques used in the graph that are breadth−first search and depth−first search. Another important point about graphs is that we can defined number of edges in the graph. Graph structure has relatively more complex structure.



A graph is a data structure that is defined by two components:

1. A **node** or a vertex.
2. An edge E or **ordered pair is a connection between two nodes u, v** that is identified by unique pair (u,v). The pair (u,v) is ordered because (u,v) is not same as (v,u) in case of directed graph. The edge may have a weight or is set to one in case of unweighted graph.

Consider the given below graph



Applications: Graph is a data structure which is used extensively in our real-life.

1. Social Network: Each user is represented as a node and all their activities, suggestion and friend list are represented as an edge between the nodes.

2. Google Maps: Various locations are represented as vertices or nodes and the roads are represented as edges and graph theory is used to find shortest path between two nodes.

3. Recommendations on e-commerce websites: The "Recommendations for you" section on various e-commerce websites uses graph theory to recommend items of similar type to user's choice.

4. Graph theory is also used to study molecules in chemistry and physics.

State Space Representation

A **state space** is a way to mathematically represent a problem by defining all the possible states in which the problem can be. This is used in search algorithms to represent the initial state, goal state, and current state of the problem. Each state in the state space is represented using a set of variables.

The **efficiency** of the search algorithm greatly depends on the size of the state space, and it is important to choose an appropriate representation and search strategy to search the state space efficiently.

One of the most well-known **state space search algorithms** is the A algorithm. Other commonly used state space search algorithms include **breadth-first search (BFS)**, **depth-first search (DFS)**, **hill climbing**, **simulated annealing**, and **genetic algorithms**.

Features of State Space Search

State space search has several features that make it an effective problem-solving technique in Artificial Intelligence. These features include:

- **Exhaustiveness:**
  State space search explores all possible states of a problem to find a solution.
- **Completeness:**
  If a solution exists, state space search will find it.
- **Optimality:**
  Searching through a state space results in an optimal solution.
- **Uninformed and Informed Search:**
  State space search in artificial intelligence can be classified as uninformed if it provides additional information about the problem.

Steps in State Space Search
The steps involved in state space search are as follows:

- To begin the search process, we set the current state to the initial state.
- We then check if the current state is the goal state. If it is, we terminate the algorithm and return the result.
- If the current state is not the goal state, we generate the set of possible successor states that can be reached from the current state.
- For each successor state, we check if it has already been visited. If it has, we skip it, else we add it to the queue of states to be visited.
- Next, we set the next state in the queue as the current state and check if it's the goal state. If it is, we return the result. If not, we repeat the previous step until we find the goal state or explore all the states.
- If all possible states have been explored and the goal state still needs to be found, we return with no solution.

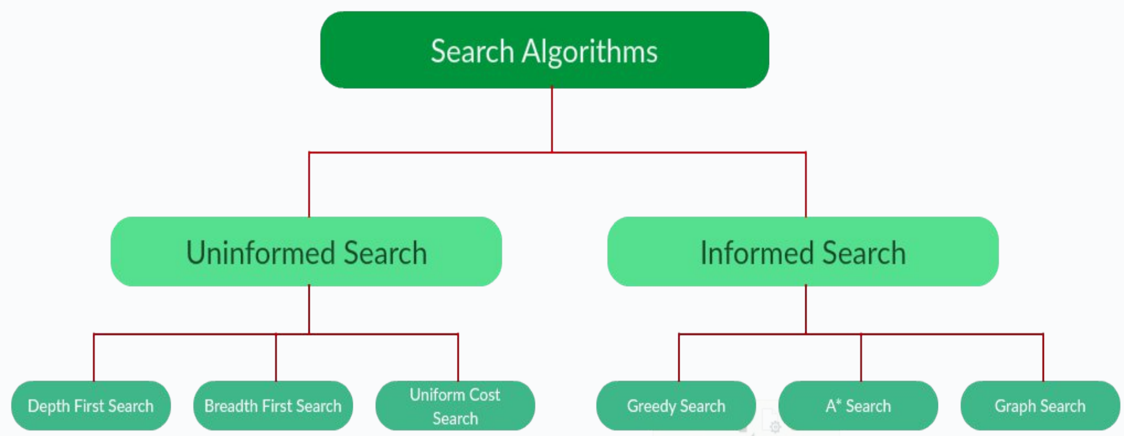
State Space Representation

State space Representation involves defining an INITIAL STATE and a GOAL STATE and then determining a sequence of actions, called states, to follow.

- **State:**
  A state can be an Initial State, a Goal State, or any other possible state that can be generated by applying rules between them.
- **Space:**
  In an AI problem, space refers to the exhaustive collection of all conceivable states.
- **Search:**
  This technique moves from the beginning state to the desired state by applying good rules while traversing the space of all possible states.
- **Search Tree:**
  To visualize the search issue, a search tree is used, which is a tree-like structure that represents the problem. The initial state is represented by the root node of the search tree, which is the starting point of the tree.
- **Transition Model:**
  This describes what each action does, while Path Cost assigns a cost value to each path, an activity sequence that connects the beginning node to the end node. The optimal option has the lowest cost among all alternatives.

Search graph & Search Tree

Types of search algorithms:
There are far too many powerful search algorithms out there to fit in a single article. Instead, this article will discuss *six* of the fundamental search algorithms, divided into *two* categories, as shown below.



Uninformed Search Algorithms:

The search algorithms in this section have no additional information on the goal node other than the one provided in the problem definition. The plans to reach the goal state from the start state differ only by the order and/or length of actions. Uninformed search is also called **Blind search**. These algorithms can only generate the successors and differentiate between the goal state and non goal state.
The uninformed search algorithms are -

1. Depth First Search
2. Breadth First Search
3. Uniform Cost Search

Informed Search Algorithms:

Here, the algorithms have information on the goal state, which helps in more efficient searching. This information is obtained by something called a *heuristic.*
The Informed search algorithms are -

1. Greedy Search
2. A* Tree Search
3. A* Graph Search