Lab 1 – State Space Search

Aim – Study state space representation for AI problem solving.

Theory -

1. **State space search**

State space search is a problem-solving technique used in Artificial Intelligence (AI) to find the solution path from the initial state to the goal state by exploring the various states. The state space search approach searches through all possible states of a problem to find a solution. It is an essential part of Artificial Intelligence and is used in various applications, from game-playing algorithms to natural language processing.

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**.

2. **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.

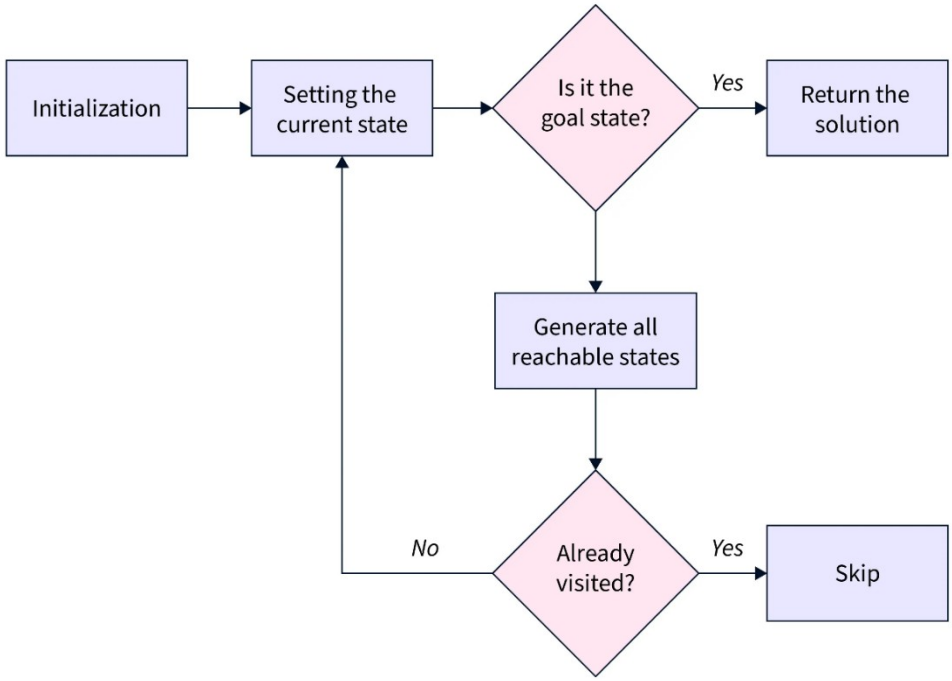
3. Steps in State Space Search

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.

4. 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.

Conclusion - State Space Search is a problem-solving technique used in AI to find a solution to a problem by exploring all possible states of the problem.

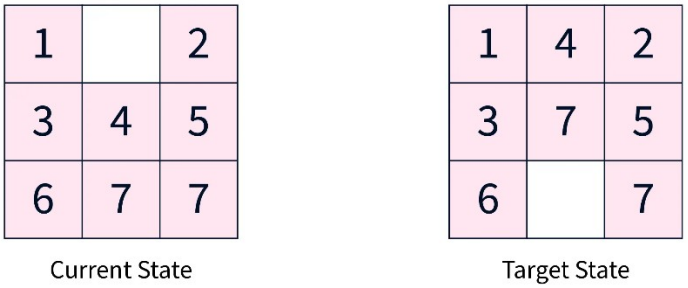
FAQs

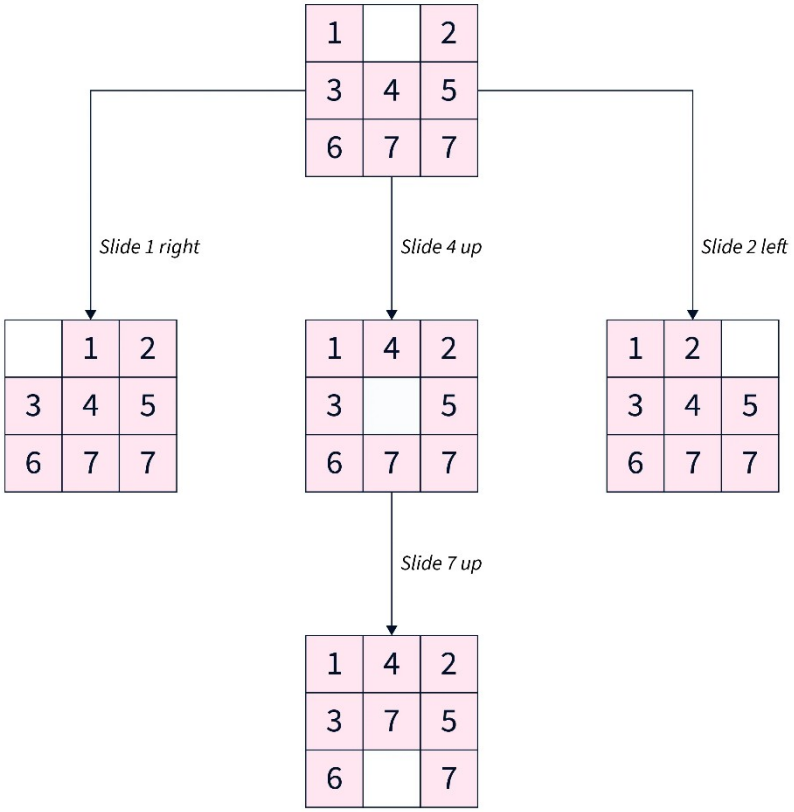
1. **State space representation of 8 Puzzle Problem**

   The **8-puzzle** problem is a commonly used example of a state space search. It is a sliding puzzle game consisting of 8 numbered tiles arranged in a 3x3 grid and one blank space. The game aims to rearrange the tiles from their initial state to a final goal state by sliding them into the blank space.

   To represent the state space in this problem, we use the nine tiles in the puzzle and their respective positions in the grid. Each state in the state space is represented by a 3x3 array with values ranging from 1 to 8, and the blank space is represented as an empty tile.

   The initial state of the puzzle represents the starting configuration of the tiles, while the goal state represents the desired configuration. **Search algorithms** utilize the state space to find a sequence of moves that will transform the initial state into the goal state.





2. Application of Stata Space Search

State space search algorithms are used in various fields, such as robotics, game playing, computer networks, operations research, bioinformatics, cryptography, and supply chain management. In artificial intelligence, state space search algorithms can solve problems like **pathfinding**, **planning**, and **scheduling**.

They are also useful in planning robot motion and finding the best sequence of actions to achieve a goal. In games, state space search algorithms can help determine the best move for a player given a particular game state.

State space search algorithms can optimize routing and resource allocation in computer networks and operations research.

In **Bioinformatics**, state space search algorithms can help find patterns in biological data and predict protein structures.

In **Cryptography**, state space search algorithms are used to break codes and find cryptographic keys.