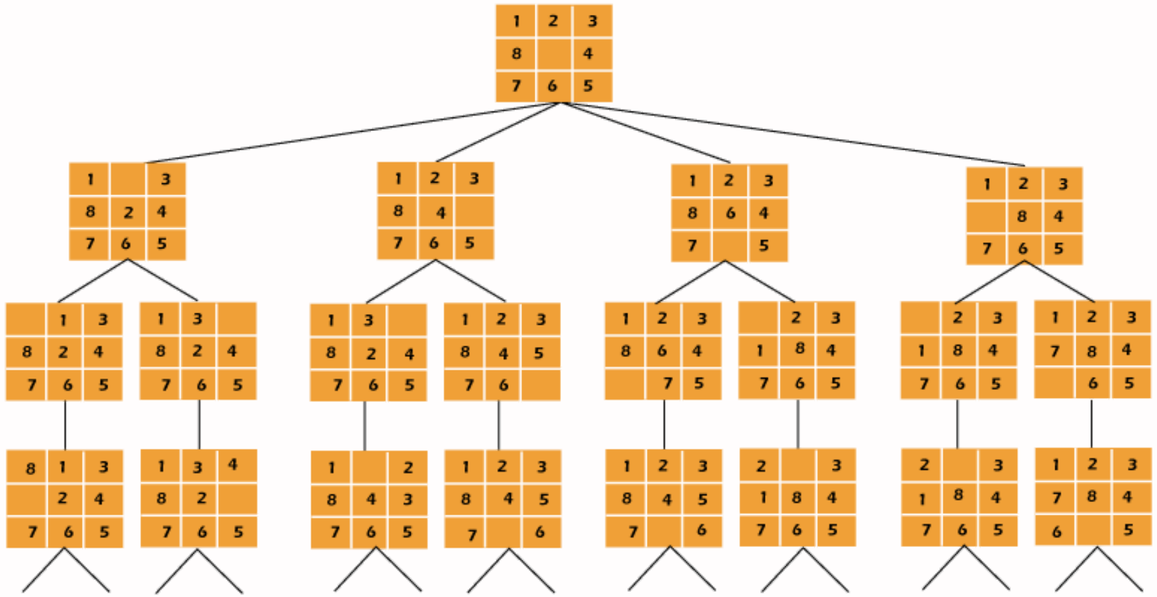
Implementing DFS & BFS for 8-Puzzle Problem

Roll no.	Name
381029	Jatin Sathe
381036	Om Lahore



Course - Artificial Intelligence

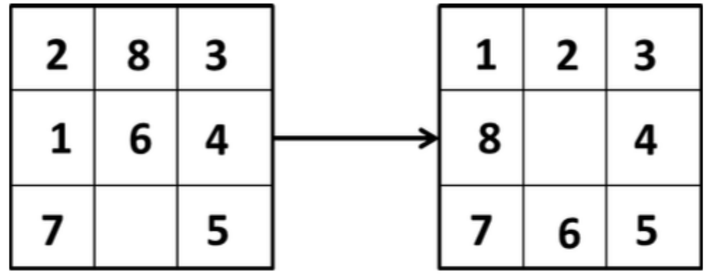
Introduction

What is the 8 Puzzle Problem?

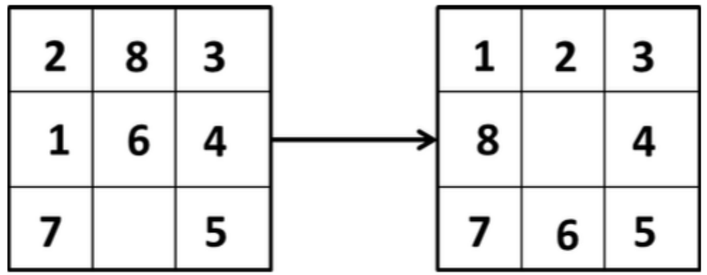
- A classic problem involving an 3x3 grid with 8 numbered tiles and one empty space.
- Goal: Arrange tiles from any starting state into a specific goal state using valid moves (left, right, up, down).
- Widely used in artificial intelligence to demonstrate search algorithms.

Puzzle Mechanics:

- The puzzle is played on a 3x3 board with tiles numbered 1 to 8 and one empty space.
- Players can move tiles into the empty space by sliding them up, down, left, or right.
- The puzzle's goal is to reach a specific configuration from any starting arrangement.



Problem setup



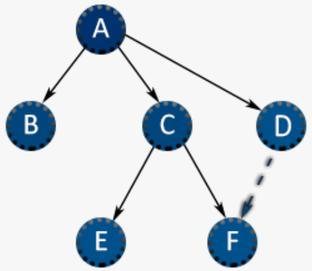
Operations: Move the blank tile (_).

- Move Left
- Move Right
- Move Up
- Move Down

Challenge: Finding the shortest/most efficient sequence of moves to solve the puzzle.

Introduction to Search Algorithms

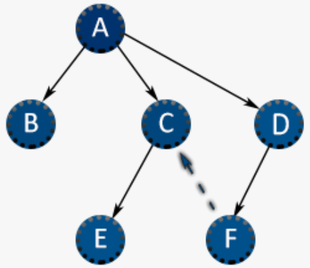
Search Algorithms: Used to explore possible states and find the solution.



ABCDEF

Breadth-First Search (BFS):

Explores neighbors level by level, ensuring shortest path is found.



ADFCEB

Depth-First Search (DFS):

Explores as far as possible down a branch before backtracking.

Key difference: DFS can get lost in deep branches, while BFS is guaranteed to find the shortest solution.

Depth-First Search (DFS)

- DFS explores one possible sequence of moves as deep as possible.
- Backtracks when no further moves are possible.
- Suitable for puzzles with fewer possible states but might be inefficient for large search spaces.

Implementation

- Stack-based approach (LIFO).
- Steps:
  - Push initial state to stack.
  - Explore possible moves, pushing new states to the stack.
  - Backtrack when no further moves possible.
  - Continue until goal state is found or stack is empty.

BFS (Breadth-First Search)

- BFS explores all possible moves at the current depth level before moving deeper.
- **Guaranteed to find the shortest solution.**
- Memory-intensive as it needs to store all nodes at the current level.

Implementation

- Queue-based approach (FIFO).
- Steps:
  - Enqueue initial state.
  - Explore all neighbors at current depth.
  - Enqueue new states at the next depth level.
  - Continue until goal state is found.

Conclusion

- DFS and BFS are fundamental search algorithms for solving the 8-puzzle.
- BFS is better for finding optimal solutions, but DFS can be more memory-efficient in smaller problem spaces.
- These algorithms provide a strong foundation for more complex AI-based search strategies.

live mini-project link

https://8-puzzle-three.vercel.app