## 1. Introduction

The 8-puzzle problem is a classical problem in Artificial Intelligence that is commonly used to test and evaluate search algorithms.
In this report, the Depth First Search (DFS) algorithm is used to solve the 8-puzzle problem by exploring the state space until the goal state is reached.

---

## 2. Problem Description

The 8-puzzle consists of a 3×3 grid containing eight numbered tiles and one empty space.
The objective of the problem is to reach the goal configuration by sliding tiles into the empty space without removing any tile from the grid.

Goal State:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 0 |

---

## 6. How DFS Solves the 8-Puzzle Step by Step

1. The initial state is pushed into the stack.
2. The algorithm pops the top state from the stack.
3. If the popped state matches the goal state, the solution is found.
4. Otherwise, the state is added to the visited set.
5. All valid neighboring states are generated using the successor function.
6. These neighboring states are pushed onto the stack.

7. The process continues until the goal state is reached or the stack becomes empty.

DFS continues exploring deeper states until a solution is found.

---

## 7. Solution Characteristics

. DFS is capable of finding a solution if one exists.

. The solution found by DFS is **not guaranteed to be optimal**.

. The number of moves may be larger than the minimum possible.

Despite this, DFS is memory-efficient compared to other uninformed search algorithms like BFS.

## 9. Conclusion

The Depth First Search algorithm successfully solves the 8-puzzle problem by exploring the state space deeply until the goal state is reached.

While DFS does not guarantee the shortest solution, it provides a simple and memory-efficient approach for solving the problem.

This implementation demonstrates how uninformed search algorithms can be applied to classical AI problems such as the 8-puzzle.