Name: Aziz Sayyad
Roll No: 381069
Batch: A3

Practical 1

Implement DFS, BFS for 8-Puzzle Problem

Problem Statement:
Implement Depth First Search (DFS) and Breadth First Search (BFS) algorithms to solve the 8-puzzle problem.

Objectives:

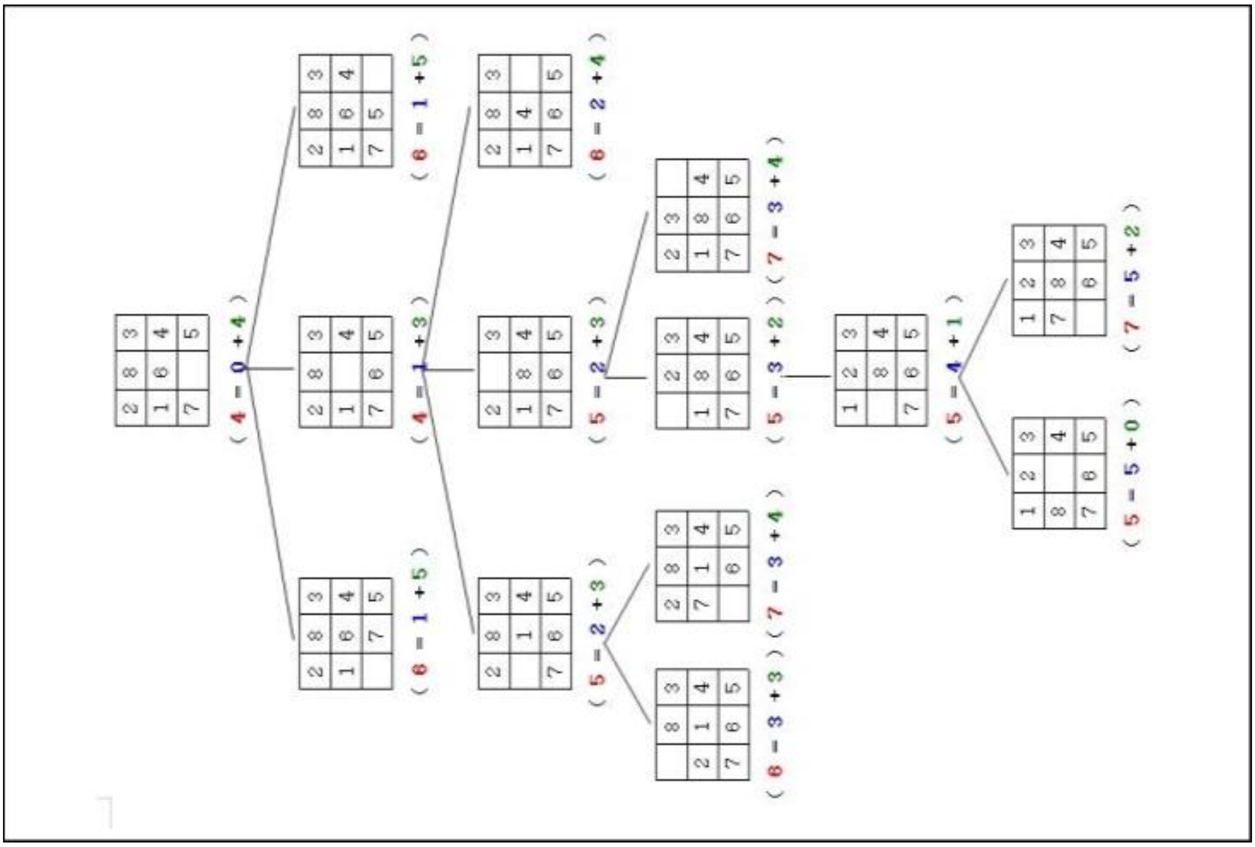
- Understand the functioning of DFS and BFS.
- Implement and compare these algorithms for solving the 8-puzzle problem.

Theory:

- **Methodology:**
  DFS explores as far as possible along each branch before backtracking, whereas BFS explores all neighbors at the present depth before moving on to nodes at the next depth level. For the 8-puzzle problem, DFS and BFS will explore the possible moves of tiles from the initial configuration to the goal state.

- **Working Principle / Algorithm:**
  - **DFS Algorithm:**
    1. Start with the root (initial configuration).
    2. Explore each branch as deep as possible.
    3. Use backtracking when no further moves are possible.
    4. Repeat until the goal configuration is reached or all configurations are explored.
  - **BFS Algorithm:**
    1. Start with the root (initial configuration).
    2. Explore all nodes at the present depth.
    3. Move to the next depth level and repeat until the goal configuration is reached.
- **Advantages:**
  - DFS: Memory efficient for deep searches.
  - BFS: Guarantees the shortest path in unweighted graphs.
- **Disadvantages / Limitations:**
  - DFS: May not find the shortest path and could get stuck in infinite loops.
  - BFS: Memory intensive for larger search spaces.

Diagram:



Conclusion:
DFS and BFS are effective search algorithms for the 8-puzzle problem. However, BFS ensures the shortest path, while DFS may consume less memory for deep searches.