**Assignment 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.

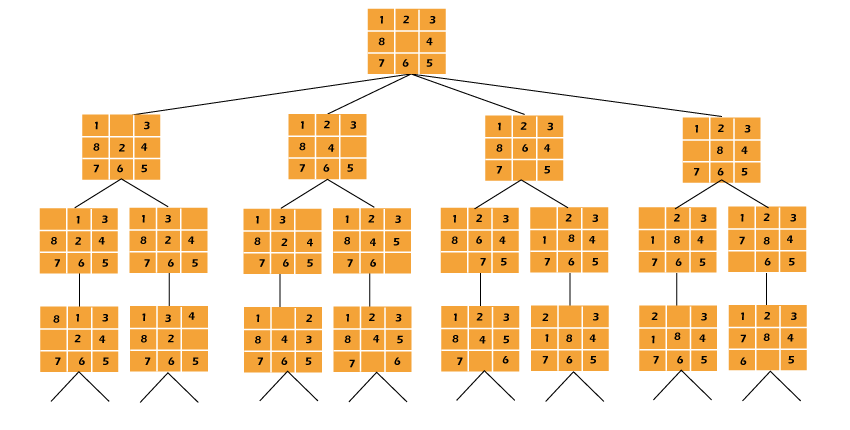
**Objectives:**

* Gain a deeper understanding of Depth First Search (DFS) and Breadth First Search (BFS) algorithms.
* Implement and compare the performance of these algorithms in solving the 8-puzzle problem.

**Theory:**

* **Methodology:**  
  DFS delves into each branch as deeply as possible before backtracking, while BFS systematically explores all nodes at the current depth level before moving to the next. In the context of the 8-puzzle problem, both algorithms will traverse possible tile moves from the initial state to the goal configuration.
* **Working Principle / Algorithm:**
  + **DFS Algorithm:**
    1. Begin from the root node (initial puzzle configuration).
    2. Traverse each branch as deeply as possible.
    3. Backtrack when no more moves are available.
    4. Continue until either the goal configuration is found or all possible configurations are explored.
  + **BFS Algorithm:**
    1. Start from the root node (initial puzzle configuration).
    2. Explore all nodes at the current depth level.
    3. Progress to the next depth level and repeat the process until the goal configuration is found.
* **Advantages:**
  + **DFS:** Efficient in terms of memory for deep searches.
  + **BFS:** Guarantees finding the shortest path in unweighted graphs.
* **Disadvantages / Limitations:**
  + **DFS:** May not always find the shortest path and can fall into infinite loops in certain cases.
  + **BFS:** Consumes more memory, especially in larger search spaces.

**Diagram:**



**Conclusion:**  
DFS and BFS are useful algorithms for solving the 8-puzzle problem. BFS has the advantage of guaranteeing the shortest path, whereas DFS may be more memory efficient for deeper searches.