Name: Rushiraj Suwarnkar  
Roll no.: 381018  
PRN: 22310315  
———————————————————————————————

**Assignment 6: Implement Basic Search Strategies – 8-Queens Problem**  
———————————————————————————————

**Problem Statement:**  
Implement basic search strategies (like BFS, DFS, or backtracking) to solve the 8-Queens problem, which requires placing 8 queens on a chessboard such that no two queens attack each other.  
———————————————————————————————

**Objectives:**

* Understand the functioning of basic search strategies.
* Implement and apply BFS, DFS, or backtracking for constraint satisfaction problems.
* Solve the 8-Queens problem efficiently using search techniques.  
  ———————————————————————————————

**Theory:**  
———————————————————————————————

**Methodology:**  
The 8-Queens problem is a classic combinatorial problem. Each queen must be placed on a unique row and column without threatening another queen diagonally. Search strategies explore different arrangements:

* **DFS/Backtracking:** Places queens row by row and backtracks when a conflict occurs.
* **BFS:** Explores all possible positions level by level.  
  The solution space is explored until all 8 queens are placed safely on the board.  
  ———————————————————————————————

**Working Principle / Algorithm:**  
———————————————————————————————

**DFS / Backtracking Algorithm:**

1. Start with the first row.
2. Place a queen in the first safe column.
3. Move to the next row and repeat step 2.
4. If no safe column exists in a row, backtrack to the previous row and try a different column.
5. Repeat until all 8 queens are placed safely.  
   ———————————————————————————————

**BFS Algorithm:**

1. Start with an empty board.
2. Generate all possible positions for the first queen.
3. For each partial solution, add a queen to the next row in a safe column.
4. Continue expanding partial solutions until a complete solution is found.  
   ———————————————————————————————

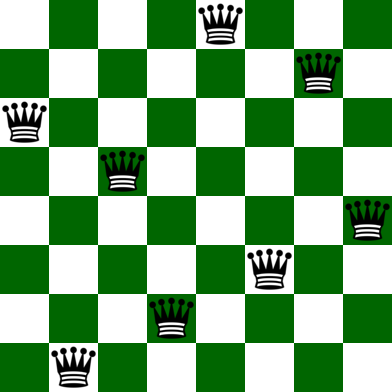
**Advantages:**

* DFS/Backtracking: Efficient memory usage, simple implementation.
* BFS: Guarantees finding the shortest sequence of placements if needed.
* Provides all possible solutions to the problem.  
  ———————————————————————————————

**Disadvantages / Limitations:**

* DFS/Backtracking: May explore many unnecessary states before finding a solution.
* BFS: Memory-intensive for larger search spaces.
* Computational cost increases rapidly with board size.  
  ———————————————————————————————

**Diagram:**

  
———————————————————————————————

**Conclusion:**  
Basic search strategies like DFS, BFS, and backtracking provide systematic approaches to solving the 8-Queens problem. Backtracking is particularly efficient, allowing the algorithm to prune invalid arrangements and find all valid solutions effectively.  
———————————————————————————————