**Assignment 6**

**Problem Statement:** Implement basic search strategies – 8-Queens Problem

**Library**

For implementing the 8-Queens problem, the following libraries may be used:

* **NumPy**: For handling arrays and matrices, particularly for board representation.
* **Matplotlib**: For visualizing the board and the placement of queens.
* **Time**: To measure the performance and execution time of the search strategies.

**Theory**

The primary theory behind solving the 8-Queens problem involves search algorithms, particularly:

* **Backtracking**: A depth-first search technique that incrementally builds candidates for solutions and abandons candidates ("backtracks") as soon as it determines that they cannot be extended to a valid solution.
* **Constraint Satisfaction**: The problem can be seen as a constraint satisfaction problem (CSP), where the placement of queens must satisfy specific constraints (no two queens can attack each other).

**Methodology**

1. **Initial Setup**:
   * Create an 8x8 board representation, typically as a 2D array or list.
2. **Backtracking Algorithm**:
   * Start placing queens row by row.
   * For each row, try placing a queen in each column.
   * After placing a queen, check for conflicts (using row and diagonal constraints).
   * If a conflict arises, backtrack by removing the queen and trying the next column.
   * If all queens are placed without conflicts, a solution is found.
3. **Implementation Steps**:
   * Initialize the board.
   * Recursively call the function to place queens in each row.
   * Maintain a list or set to track columns and diagonals that are under attack.

**Advantages**

* **Simplicity**: The backtracking approach is straightforward and easy to understand.
* **Complete Search**: The algorithm guarantees finding all possible solutions, if required.
* **Efficiency**: While backtracking can be time-consuming, it can be optimized with techniques like constraint propagation to reduce the search space.

**Disadvantages**

* **Time Complexity**: The worst-case time complexity is O(N!), where N is the number of queens, due to the factorial growth of permutations.
* **Space Complexity**: The recursive stack may consume significant memory, particularly for larger N.
* **Inefficiency for Larger Boards**: The approach does not scale well beyond 8x8 due to exponential growth in possibilities.

**Conclusion**

The 8-Queens problem exemplifies fundamental concepts in search algorithms and constraint satisfaction. The backtracking method is effective for finding solutions in a manageable problem size like 8x8. However, as the size of the board increases, the limitations of this approach become more apparent, necessitating more sophisticated techniques or heuristics. Overall, the 8-Queens problem remains a valuable educational tool in computer science, particularly in the study of algorithms and problem-solving strategies.