# Artificial Intelligence (AI101B)

**Even Semester**

## Session 2024-25

## ‘N’ QUEENS PROBLEM SOLVING

## Subhash Kumar 202410116100211

**Shubham Singh 202410116100204**

**Project Supervisor:** Ms. Komal Salgotra

(Assistant Professor)

**INTRODUCTION**

* The N-Queens Problem is a combinatorial

puzzle that requires placing N queens on an N×N

chessboard so that no two queens attack each other.

* Constraints: No two queens can be in the same row,

column, or diagonal.

* Used in AI, optimization, and algorithm research.
* Backtracking Algorithms:

Demonstrates recursion and constraint satisfaction.

* AI & Constraint Satisfaction:

Common in AI research.

* Optimization Problems:

Used in parallel computing and scheduling.

* Game Theory:

Helps in chess AI development.

**IMPORTANCE AND APPLICATION**

# METHODOLOGY

1. Problem Formulation: Place N queens without conflicts.
2. Approach: Backtracking algorithm with pruning.
3. Representation: 1D array for board state (O(N) space complexity).
4. Visualization: Matplotlib and chessboard rendering.

5. Time Complexity: O(N!) in worst case, but backtracking reduces steps.

# CODE IMPLEMENTATION

* Algorithm Steps:
* 1. Check if placing a queen at a position is safe.
* 2. Place queens recursively and backtrack when needed.
* 3. Store valid solutions.
* Python Code Snippet:
* def is\_safe(board, row, col):
* for i in range(row):
* if board[i] == col or abs(board[i] - col) == abs(i - row):
* return False
* return True

# VISUALIZATION

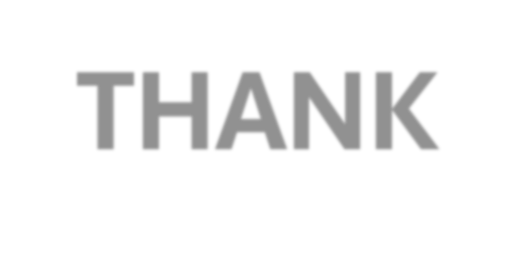
* Matplotlib Heatmap Representation: Displays the chessboard.
* Unicode Chess Symbols: Shows queens on board.
* Static Chessboard Image: Eliminates scrolling issue in Google Colab.
* Example Output (8-Queens Solution): Include Image

**OUTCOMES**

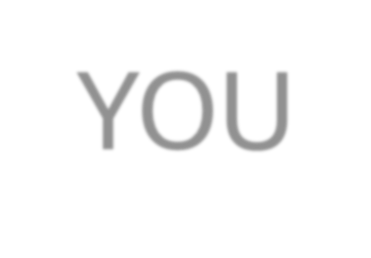
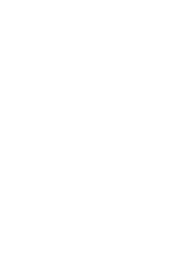
* Successfully implemented backtracking to solve N-Queens.
* Visualized solutions using Matplotlib and chessboard.
* Optimized code for performance.
* Learned about AI algorithms and problem-solving techniques.

# CONCLUSION

* The N-Queens problem is a classic AI problem demonstrating recursion and constraint satisfaction.
* - Visualization helps in understanding the solution better.
* - The project enhanced problem-solving and coding skills.
* - Future improvements could involve optimizing for large N values.



**THANK**



YOU

