Name	ID	Department
مصطفي محمد محمود	20225236	SWE
محمود جنيدي محمد	20225071	SWE
محمد أحمد عبدالحميد	20220374	General
سيف أشرف	20225025	SWE
هاني محمد بكري	202001022	General

Project Report: N-Queens Solver

Introduction and Overview:

➢ Project Idea and Overview:

- To solve the N-Queens problem using a combination of algorithms (Backtracking, Best-First Search, Hill-Climbing, and Genetic Algorithm) to compare their efficiency, effectiveness, and applicability in finding valid solutions for various board sizes.

Similar Applications Links:

- ✓ https://www.researchgate.net/publication/349525086_Parallel_Implementations_of_Candidate_Solution_Evaluation_Algorithm_for_N-Queens_Problem
- ✓ https://www.researchgate.net/publication/220838483_Complete_and_Incomplete_Algorithms_for_the_Queen_Graph_Coloring_Problem

Literature Review Links:

- ✓ https://www.researchgate.net/publication/240320772_The_n-Queens_Problem
- ✓ https://www.sciencedirect.com/science/article/pii/S0012365X07010394#sec1

Proposed Solution, Dataset & Features:

- **Backtracking:** Systematically explore all possible placements of queens, ensuring that no two queens threaten each other. This approach guarantees finding all possible solutions but may be time-consuming for larger boards.
- **Best-First Search:** Implement a heuristic-based search to prioritize placing queens in positions with the least potential conflicts, aiming to find a solution more quickly than backtracking.
- **Hill-Climbing:** Start with an initial configuration and iteratively move to a neighboring configuration that improves the placement of queens (fewer conflicts), potentially finding a solution faster but with a risk of getting stuck in local optima.
- **Genetic Algorithm:** Use a population of potential solutions (board configurations) and evolve them over generations using selection, crossover, and mutation to find a solution. This method is robust in avoiding local optima and can handle larger board sizes effectively.

Dataset:

➢ Each configuration represents a potential solution where N queens are placed on an NxN chessboard.

Components:

- ✓ **Initial Configurations:** Randomly generated or systematically created board setups for different values of N (e.g., 4x4, 8x8).
- ✓ **Evaluation Metrics:** Number of conflicts in each configuration, where a conflict is defined as two queens threatening each other.

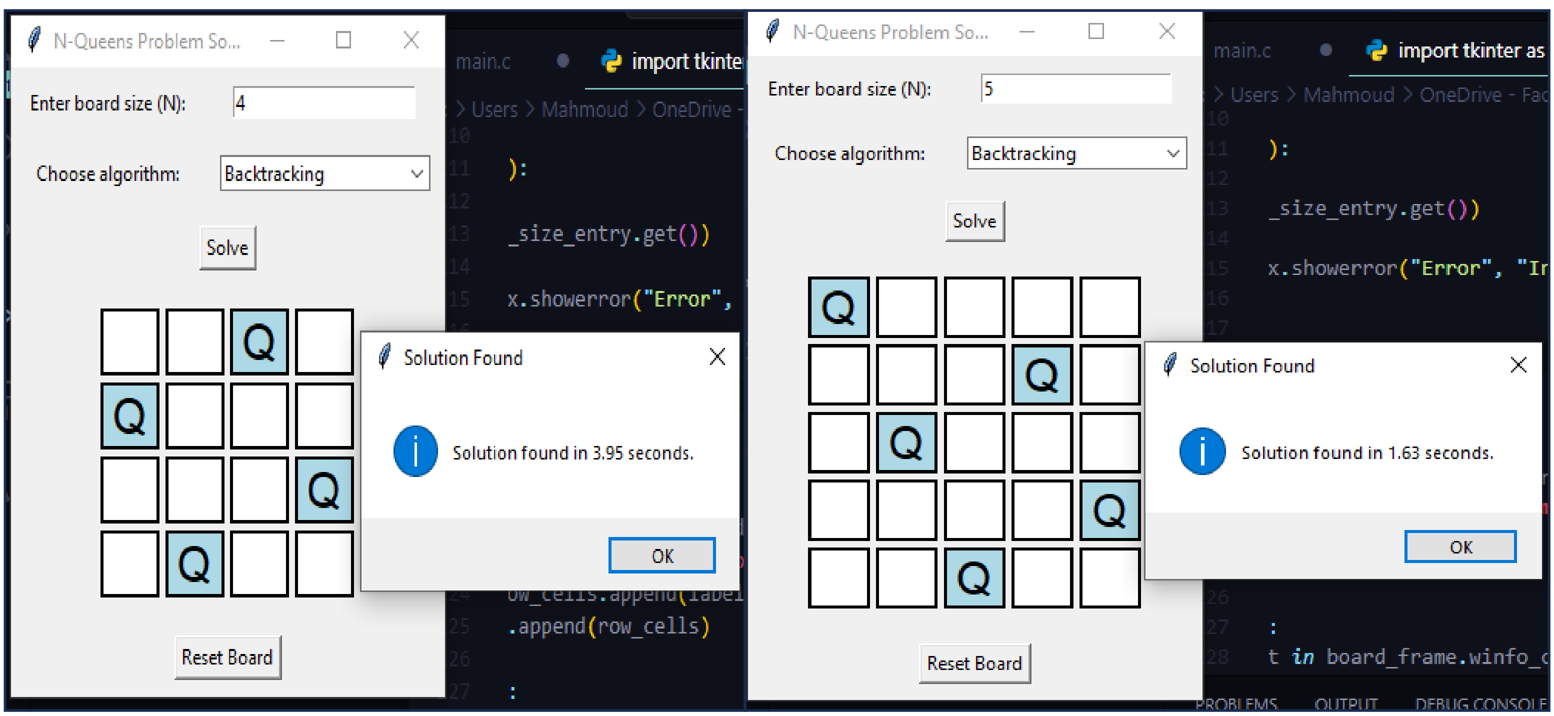
- ✓ **Heuristic Scores:** For Best-First Search and Hill-Climbing, store heuristic scores indicating the quality of each configuration.
- ✓ **Genetic Algorithm Data:** Track generations, mutation rates, crossover points, and fitness scores for each population member.

Experiments & Results:

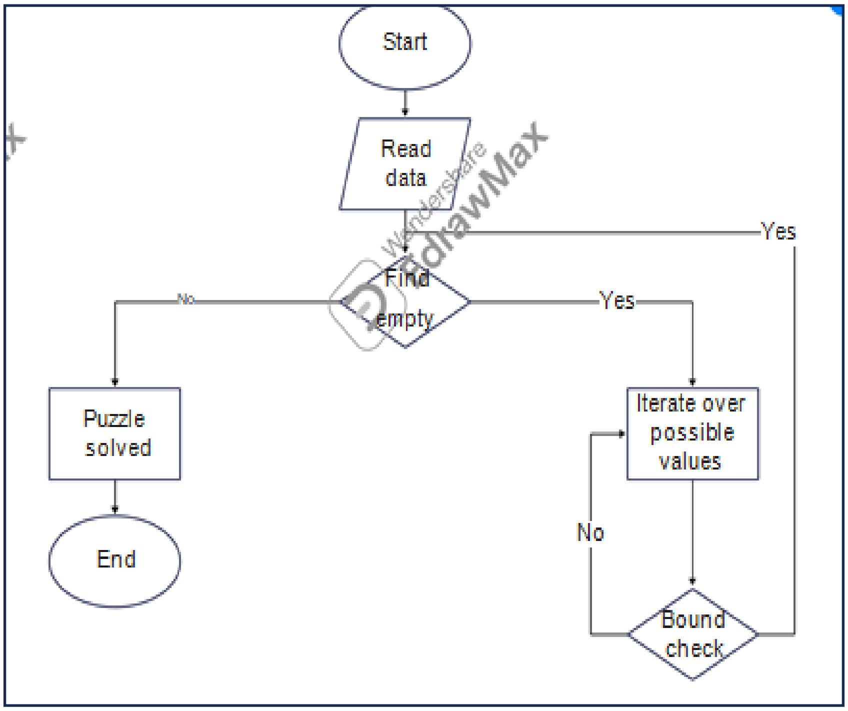
- ➢ examples of output:

Backtracking: A function to recursively place queens on the board, backtracking when a conflict is detected.

Sol1: **Sol2:**

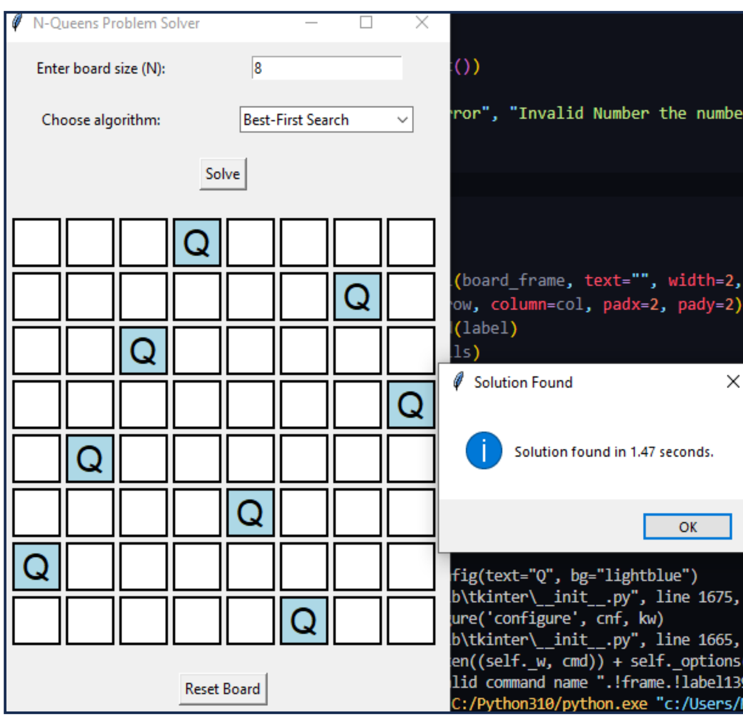


Flowchart:

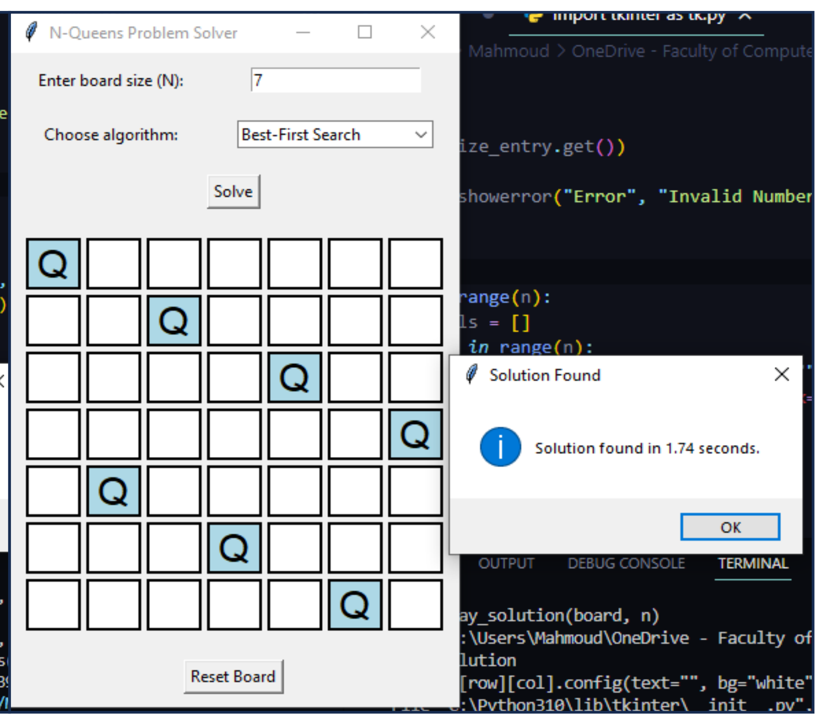


Best-First: A heuristic-driven search function that prioritizes placements leading to fewer conflicts.

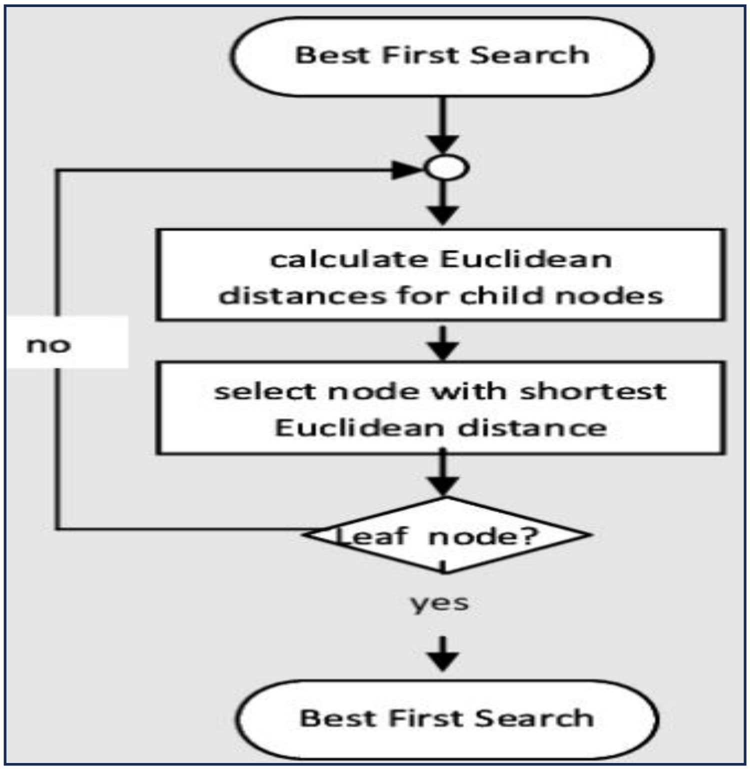
Sol1:



Sol2:

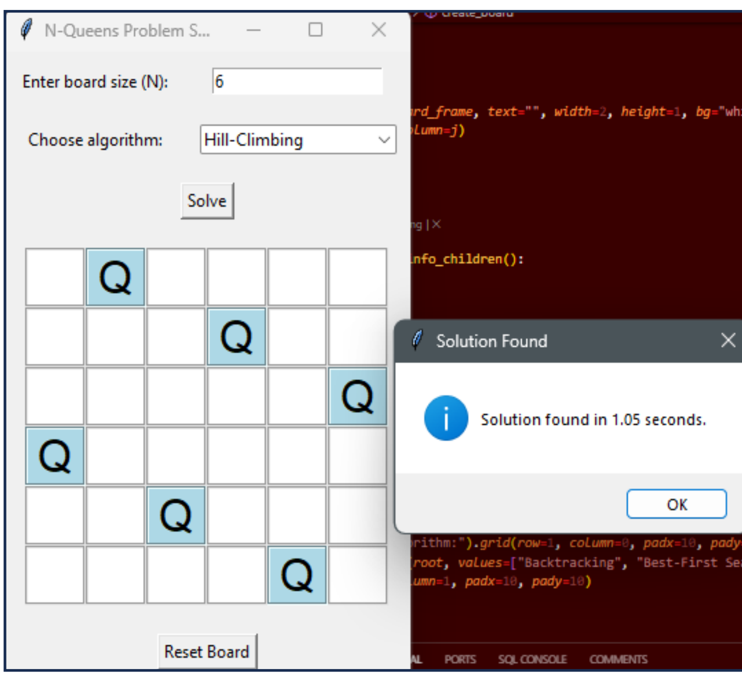


Flowchart:

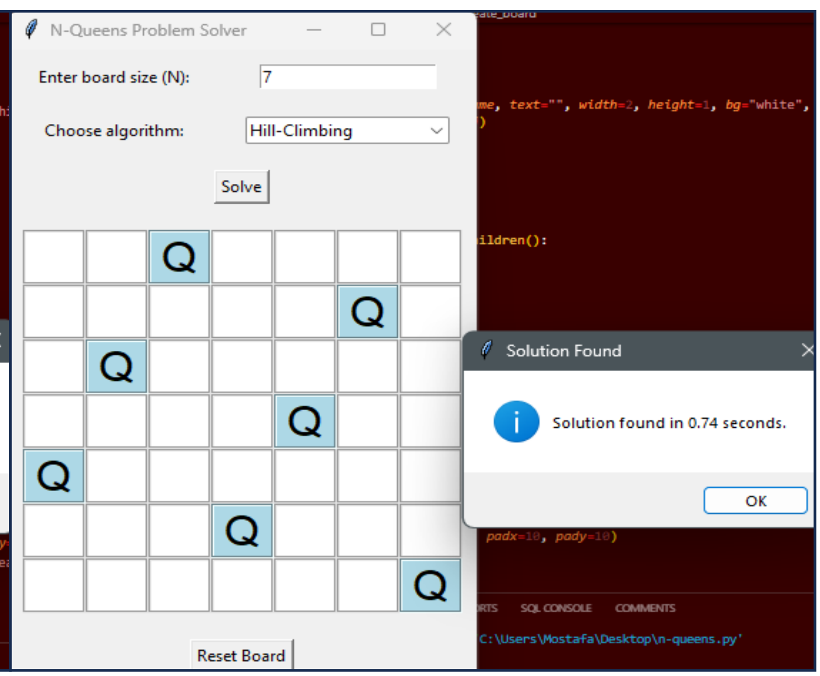


Hill-Climbing: An iterative improvement function that seeks to minimize conflicts, with an option to restart if stuck in a local optimum.

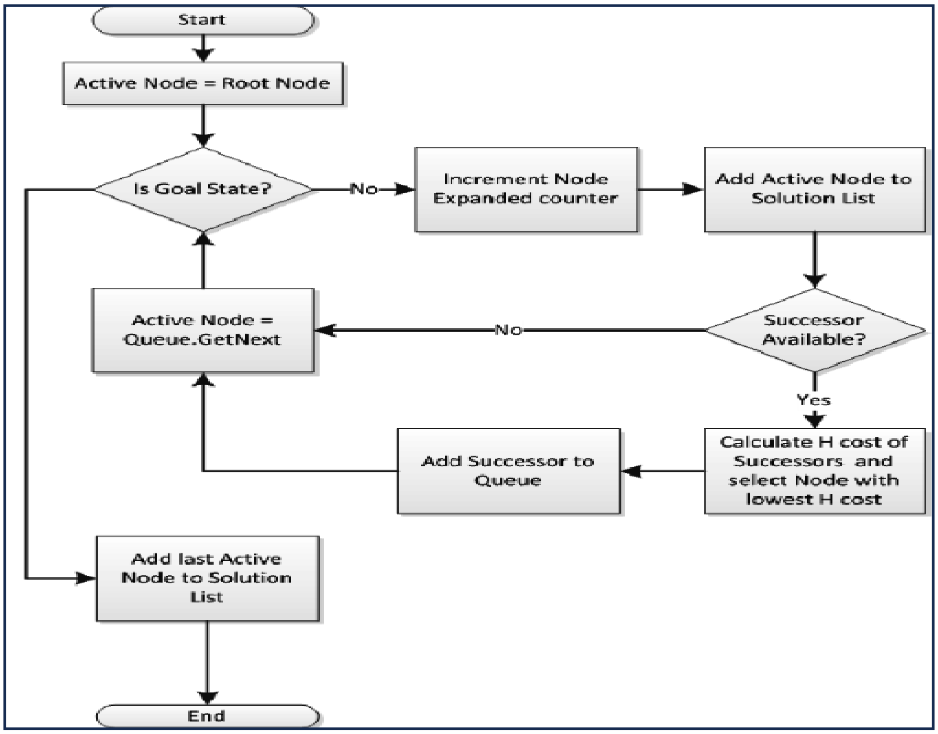
Sol1:



Sol2:

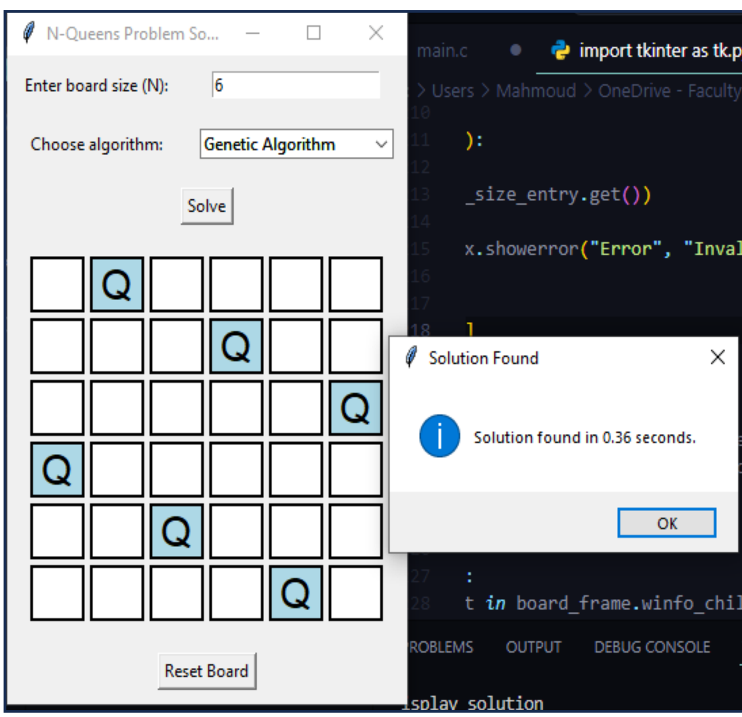


Flowchart:

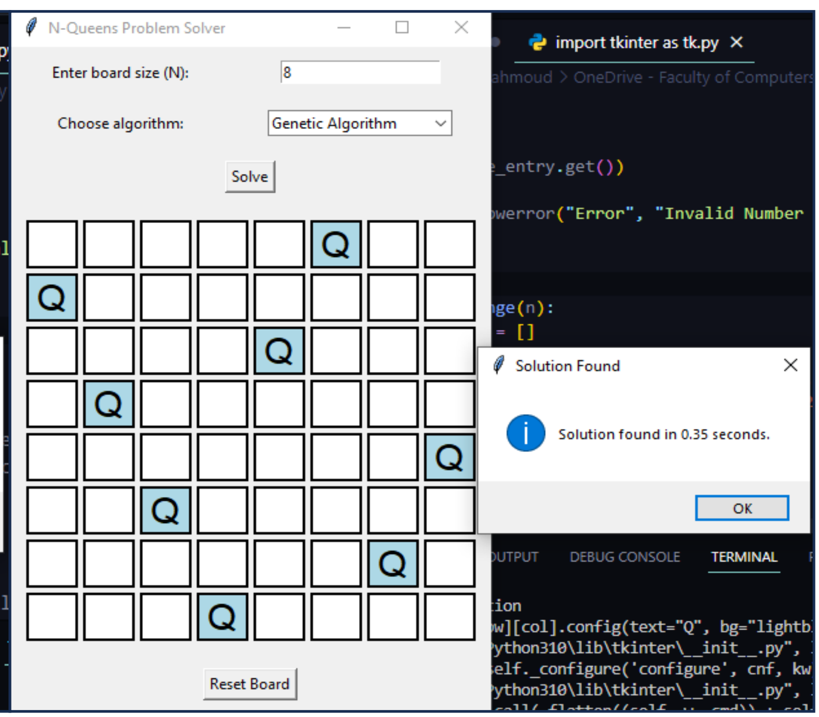


Genetic: A function to evolve a population of board configurations, using selection, crossover, and mutation to optimize queen placement.

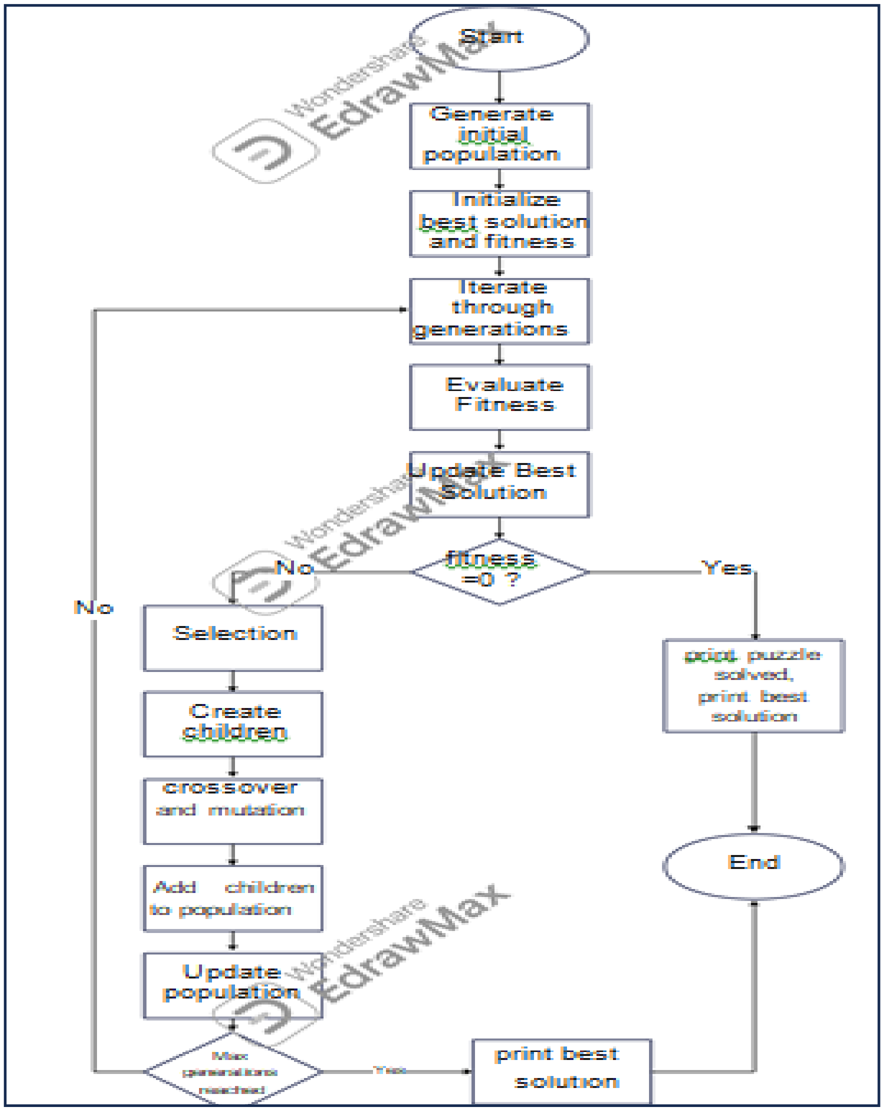
Sol1:



Sol2:



Flowchart:



Discussion and Future Work:

➢ Advantages & Disadvantages:

- **Backtracking** is comprehensive but slow.
- **Best-First Search** is efficient with good heuristics but may miss solutions.
- **Hill-Climbing** is fast but prone to getting stuck in suboptimal solutions.
- **Genetic Algorithms** offer global search and flexibility but require more computational resources and careful tuning.

Behavior Analysis and Future Modifications:

1. Backtracking

- **Behavior Analysis:**
  - Explores all possible solutions.
  - Guarantees finding all solutions but can be slow for larger boards.
- **Future Modifications:**
  - Implement pruning techniques to reduce unnecessary exploration and speed up the process.

2. Best-First Search

- **Behavior Analysis:**
  - Uses heuristics to focus on promising paths.
  - Faster than backtracking but might miss optimal solutions.
- **Future Modifications:**
  - Enhance the heuristic function to better guide the search towards optimal solutions.

3. Hill-Climbing

- **Behavior Analysis:**
  - Quickly find a solution by improving the current state.
  - Prone to getting stuck in local optima.
- **Future Modifications:**
  - Implement random restarts or simulated annealing to help escape local optima.

4. Genetic Algorithm

- **Behavior Analysis:**
  - Uses evolution-inspired techniques to explore the solution space.
  - Avoids local optima but may require many generations to converge.

GitHub Repository Link:

✓ Mostafa-moha/N-Queens_Problem_Project: Using 4 Algorithms with Simple Gui (github.com)