Course: Introduction to Artificial Intelligence

Instructor: Dr. Syed Ali Raza

Spring 2025

Institute of Business Administration, Karachi

Assignment #1

Modified N-Queens Problem

Overview

In this assignment, you will implement a modified version of the classic N-Queens problem using BFS, UCS, and A* search techniques. Unlike the traditional N-Queens puzzle—where the task is to place one-by-one N queens on an N×N chessboard so that no two queens threaten each other—this version starts with a fixed initial state and allows only restricted moves.

Initial State:
All N queens are placed in the top row (one per column). For example, when N = 4, the initial state is represented as:

```
((0, 0), (0, 1), (0, 2), (0, 3))
```

Q	Q	Q	Q

`Example of the initial state for a 4x4 board`

Moves (Actions):
At each step, a queen can be moved one cell in any of the four cardinal directions: up (`U`), down (`D`), left (`L`), or right (`R`). However, there are two restrictions:

1. **Blocked Moves in the Actions Function:**
   - A queen cannot move in a direction if the adjacent cell in that direction is already occupied by another queen.
2. **Blocked Moves in the Result Function:**
   - When executing a move, if the adjacent cell in the given direction is occupied or the move would take the queen off the board, then the move is disallowed and the state remains unchanged.

Goal State:
A state is considered a goal if:

1. Every row on the board contains exactly one queen.
2. No two queens conflict with each other. Two queens are said to conflict if they are in the same row, same column, or on the same diagonal (in any of the eight directions).

Heuristic Function:
For the heuristic-based search, you will also implement a heuristic function. This function computes the total number of conflicts by counting conflicts for every queen (i.e., if queen A conflicts with queen B, the conflict is counted twice). For example, after taking the move `(0, 'D')` in the initial state for N = 4, the conflict count should be eight (value = 8).

Objectives

By the end of this assignment, you should be able to:

- Understand and modify a search problem formulation.
- Design and implement a goal test and heuristic function that account for conflicts on rows, columns, and diagonals.
- Integrate your implementation with common search strategies to solve the puzzle.

Deliverables

1. Source Code:
   - A Python file (or a set of files) that contains the implementation of `ModifiedNQueensProblem` with the modifications outlined above, and the implementation of BFS, UCS, and A*.
   - Ensure that your code is well-documented with comments explaining key parts of your logic.

2. In a text file, report the following:
   - The solution found by each algorithm.
   - The number of nodes expanded, and the number of nodes remained in the frontier (fringe) at the end.
   - The above two for N = 4, 5, 6, 7 and 8.

     Note: For N>=7, you may need to use any efficient approach to solve the problem. In these cases, a simple heuristic may not work or take too long. For example, from this webpage (https://en.wikipedia.org/wiki/Eight_queens_puzzle) you can get some ideas of how the state space grows and how to tackle it. For this, you can try different options; for example, change how the actions are taken, how a heuristic is computed, how a goal state is defined, how collisions are counted, etc. However, prioritize solving the original problem. If necessary, consider making minimal changes to the game to solve for higher values of N.

   - Mention any modifications used to efficiently find the solutions for higher values of N.