Artificial Intelligence Lab (BTCOL 706)

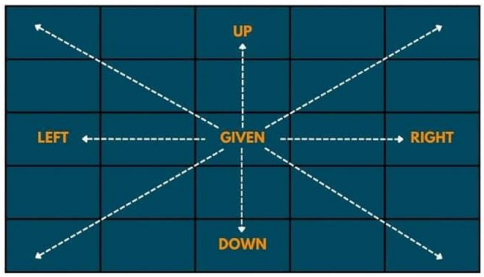
Experiment No – 02

Aim: Write a Program to solve N X N Queen Problem.

Theory:

N Queen problem demands us to place N queens on a N x N chessboard so that no queen can attack any other queen directly.

Problem Statement: We need to find out all the possible arrangements in which N queens can be seated in each row and each column so that all queens are safe. The queen moves in 8 directions and can directly attack in these 8 directions only.



The N-Queens problem is a classic combinatorial problem that requires placing N chess queens on an N×N chessboard in such a way that no two queens threaten each other. Here's a high-level algorithm for solving the N-Queens problem:

N X N Queen Problem Algorithms:

1. **Initialize the Chessboard:**

- Create an empty N×N chessboard to represent the placement of queens. This can be represented as a 2D array or a list of lists.

2. **Place Queens Recursively:**
   - Start with the first row (or any row you prefer).
   - For each cell in the row, try to place a queen and check if it's safe. If it's safe, mark the cell as occupied by a queen.
   - Move to the next row and repeat the process recursively.
   - If you reach a row where you can't place a queen without violating the rules (no two queens in the same row, column, or diagonal), backtrack to the previous row and explore other options.
3. **Base Case:**
   - When you've successfully placed N queens on the board without any conflicts, you've found a solution. Store or display this solution.
4. **Backtracking:**
   - If you encounter a situation where you cannot place a queen in any cell of the current row without conflicts, backtrack to the previous row and explore other options. This is done by undoing the placement of the last queen and trying the next available cell in the current row.
5. **Repeat:**
   - Continue this process, moving forward and backward between rows, until you've found all possible solutions or determined that no more solutions exist.
6. **Output Solutions:**
   - Keep track of all valid solutions found during the search. Once the search is complete, you can display or use these solutions as needed.

Here's a more detailed description of the steps within the recursive placement process:

- In each row, iterate through the columns to find a safe spot for a queen.
- To check if it's safe to place a queen in a cell, you need to verify that no other queens are in the same column, same row, or diagonals (both left and right diagonals).
- If a safe spot is found in the current row, place a queen, mark the cell as occupied, and proceed to the next row.

- If no safe spot is found in the current row, backtrack to the previous row, undo the placement of the queen, and explore other options.

This algorithm is often implemented using recursion and backtracking techniques. It systematically explores the solution space, trying different combinations of queen placements until all possible solutions have been found or it determines that no solution exists for the given N.

Program: Write a following program and take print with output and attached

1. Write a Program in Prolog to solve N X N Queen Problem.
2. Write a Program in Python to solve N X N Queen Problem.

Questions:

1. Explain Step performed by Problem Solving Agent
2. Explain Constraint Satisfaction Problem in details with example.
3. Explain Constraint Propagation and local consistency with type.

.

(Subject In-charge)

(Prof.S.B.Mehta)