

N-Queens Problem

The N-Queens problem is to place N queens on an $N \times N$ chessboard so that no two queens attack each other — meaning no two queens share the same row, column, or diagonal.

Backtracking Strategy:

1. Start with the first Queen placed in a fixed position (say, first row).
2. Place the next queen in the next row where it is safe (not attacked by any previous queen).
3. If no safe position is found in the current row, backtrack — move the previous queen to a new column and try again.
4. Continue until all queens are placed safely.

Concept of “Safe Position”:

A position (row, col) is safe if:

- No other queen is in the same column.
- No other queen is in the same diagonal.

Complexity:

- Time Complexity: $O(N!)$
- Space Complexity: $O(N^2)$

code

N-Queens problem using Backtracking

```
def print_board(board, n):
```

```
    for i in range(n):
```

```
        for j in range(n):
```

```
            print(board[i][j], end=" ")
```

```
        print()
```

```
    print()
```

```
# Check if it's safe to place a queen at board[row][col]
```

```

def is_safe(board, row, col, n):

    # Check column
    for i in range(row):
        if board[i][col] == 1:
            return False

    # Check upper left diagonal
    i, j = row, col
    while i >= 0 and j >= 0:
        if board[i][j] == 1:
            return False
        i -= 1
        j -= 1

    # Check upper right diagonal
    i, j = row, col
    while i >= 0 and j < n:
        if board[i][j] == 1:
            return False
        i -= 1
        j += 1

    return True

# Solve N-Queens using backtracking
def solve_n_queens(board, row, n):
    if row == n:
        print("Solution:")
        print_board(board, n)
        return True

    res = False

```

```
for col in range(n):
```

```
    if is_safe(board, row, col, n):
```

```
        board[row][col] = 1
```

```
        res = solve_n_queens(board, row + 1, n) or res
```

```
        board[row][col] = 0 # Backtrack
```

```
    return res
```

```
# ----- MAIN PROGRAM -----
```

```
n = int(input("Enter the number of Queens (N): "))
```

```
# Create NxN chessboard initialized with 0
```

```
board = [[0 for _ in range(n)] for _ in range(n)]
```

```
# Place the first queen manually
```

```
first_col = int(input(f"Enter column (0 to {n-1}) to place the first Queen in row 0: "))
```

```
board[0][first_col] = 1
```

```
if not solve_n_queens(board, 1, n):
```

```
    print("No solution exists.")
```