

N-Queens

For this problem we need to find all possible placements of queens on an $n \times n$ chessboard where no two queens attack each other.

Key Idea:

- Queens attack:
 - Same row
 - Same column
 - Same diagonals ($\text{row} - \text{col}$ and $\text{row} + \text{col}$ are constant for diagonals)
- Use backtracking:
 - Place queens row by row.
 - Maintain sets (or boolean arrays) for:
 - Columns used
 - Main diagonals ($\text{row} - \text{col}$)
 - Anti diagonals ($\text{row} + \text{col}$)
 - If valid placement found, recurse to next row.

Algorithm:

1. Initialize:
 - Board of size n filled with '.'
 - Sets for columns, diagonals.
2. Recursive function `backtrack(row)`:
 - If $\text{row} == n$, add board to result
 - Else iterate through columns:
 - Skip if column or diagonal occupied
 - Place queen
 - Recurse to next row
 - Remove queen (backtrack)

Complexity:

- Time: $O(n!)$ (permutation-like)
- Space: $O(n)$ for recursion + $O(n)$ for sets