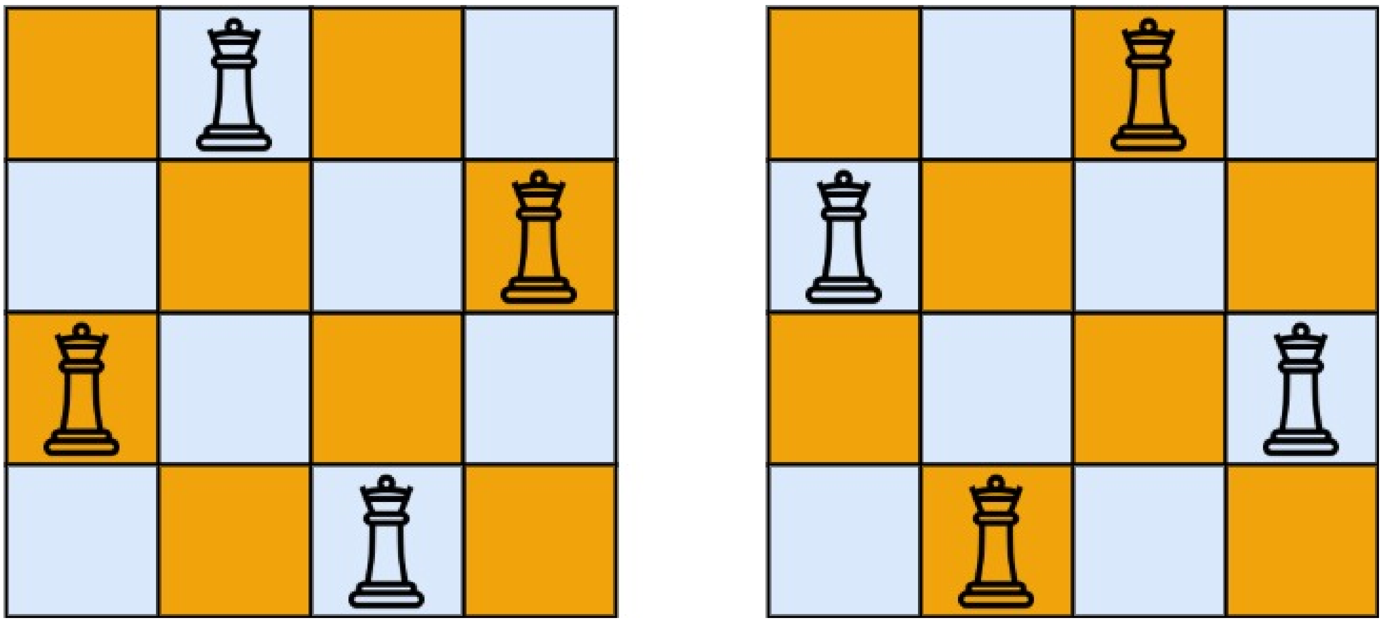
51. N-Queens

The **n-queens** puzzle is the problem of placing `n` queens on an `n x n` chessboard such that no two queens attack each other.

Given an integer `n`, return *all distinct solutions to the **n-queens puzzle***. You may return the answer in **any order**.

Each solution contains a distinct board configuration of the n-queens' placement, where `'Q'` and `'.'` both indicate a queen and an empty space, respectively.

Example 1:



```
Input: n = 4
Output: [[".Q..","...Q","Q...","..Q."],["..Q.","Q...","...Q",".Q.."]]
Explanation: There exist two distinct solutions to the 4-queens puzzle as
shown above
```

Example 2:

```
Input: n = 1
Output: [["Q"]]
```

```
class Solution:
    def solveNQueens(self, n: int) -> List[List[str]]:
        chess = [[0]*n for i in range(n)]
        res = []
        self.placeQueen(chess,res,0)
        return res



    def placeQueen(self,chess,res,row):
        if row==len(chess):
            res2 = []
            for i in range(len(chess)):
                temp = ''
                for j in range(len(chess)):
                    if chess[i][j]==1:
                        temp = temp+'Q'
                    else:
                        temp = temp+'.'
                res2.append(temp)
            res.append(res2)
            return

        for col in range(len(chess)):
            if self.isSafe(chess,row,col) is True:
                chess[row][col]=1
                self.placeQueen(chess,res,row+1)
                chess[row][col]=0

    def isSafe(self,chess,row,col):

        #Vertically Upward
        i = row-1
        while i>=0:
            if chess[i][col]==1:
                return False
            i = i-1

        j = col-1
        i = row-1
        #Diagonal Left
        while i>=0 and j>=0:
            if chess[i][j]==1:
                return False
            i = i-1
            j = j-1
        #Diagonal Right
        i = row-1
        j = col+1
        while i>=0 and j<len(chess):
            if chess[i][j]==1:
                return False
            i = i-1
            j = j+1
        return True
```