**AGENDA (Backtracking Contd.)**

- Generate Parentheses
- All Palindromic Partitions
- N-Queens Problem

## Generate Parentheses

Given a number **n**, return all the combinations of balanced parentheses of length **n**.
**Note:** A sequence of parentheses is **balanced** if every opening bracket has a corresponding closing bracket in the **correct order**. For example, "(())", "()()", and "(()())" are balanced, whereas ")()(", "))((", and "()))" are not.

**Examples:**

**Input**: n = 6
**Output:** ["((()))", "(()())", "(())()", "()(())", "()()()"]
**Explanation:** These are the only possible valid balanced parentheses.

**Input**: n = 4
**Output:** ["(())", "()()"]
**Explanation:** These are the only possible valid balanced parentheses.

## All Palindromic Partitions

Given a string **s**, find all possible ways to partition it such that every substring in the partition is a palindrome.

**Examples:**

**Input:** s = "geeks"
**Output:** [[g, e, e, k, s], [g, ee, k, s]]
**Explanation:** [g, e, e, k, s] and [g, ee, k, s] are the only partitions of "geeks" where each substring is a palindrome.

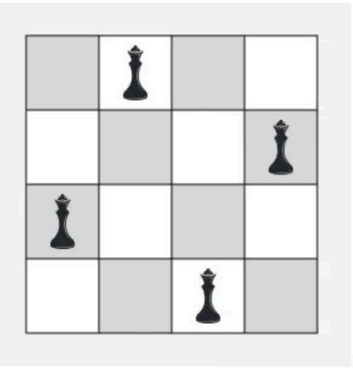
**Input:** s = "abcba"
**Output:** [[a, b, c, b, a], [a, bcb, a], [abcba]]
**Explanation:** [a, b, c, b, a], [a, bcb, a] and [abcba] are the only partitions of "abcba" where each substring is a palindrome.

# N-Queens Problem

The **n**-queens puzzle is the problem of placing n queens on a **(n × n)** chessboard such that no two queens can attack each other. Note that two queens attack each other if they are placed on the same row, the same column, or the same diagonal.

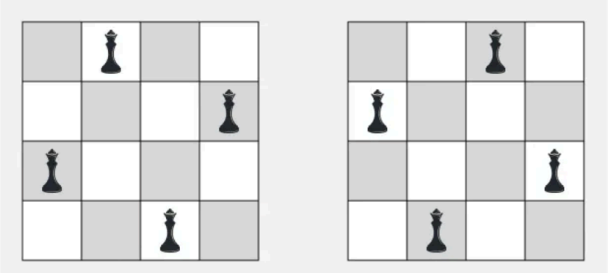
Given an integer **n**, find all distinct solutions to the **n-queens puzzle**.
You can return your answer in **any** order but each solution should represent a distinct board configuration of the queen placements, where the solutions are represented as permutations of [1, 2, 3, ..., n].

In this representation, the number in the **ith** position denotes the column in which the queen is placed in the **ith** row.
For eg. below figure represents a chessboard **[2, 4, 1, 3]**.



**Input:** n = 1
**Output:** [1]
**Explanation:** Only one queen can be placed in the single cell available.

**Input:** n = 4
**Output:** [[2, 4, 1, 3], [3, 1, 4, 2]]
**Explanation:** There are 2 possible solutions for n = 4.



**Input:** n = 3
**Output:** []
**Explanation:** There are no possible solutions for n = 3.