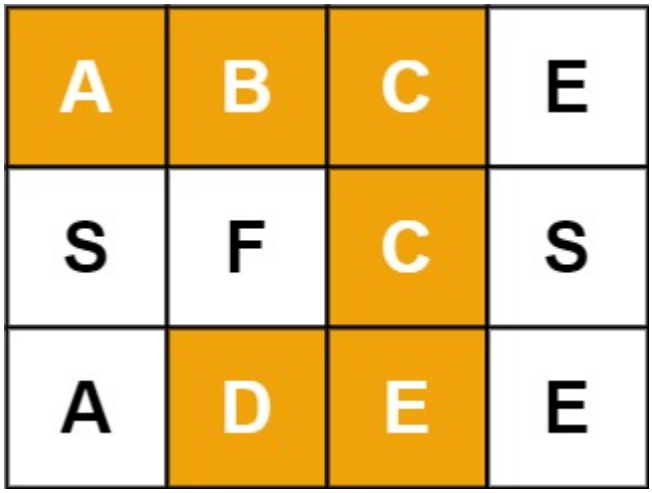
2023-07-29- Handout – Backtracking

Q1. Word Search In Grid

Given an m x n grid of characters board and a string word, return true if word exists in the grid

The word can be constructed from letters of sequentially adjacent cells, where adjacent cells are horizontally or vertically neighbouring. The same letter cell may not be used more than once



Input: board = [["A","B","C","E"],["S","F","C","S"],["A","D","E","E"]], word = "ABCCED"
Output: true

Q2. Palindrome Partitioning

Given a string s, partition s such that every substring of the partition is a palindrome. Return all possible palindrome partitioning of s.

Example 1

Input: s = "aab"
Output: [["a","a","b"],["aa","b"]]

Example 2

Input: s = "a"
Output: [["a"]]

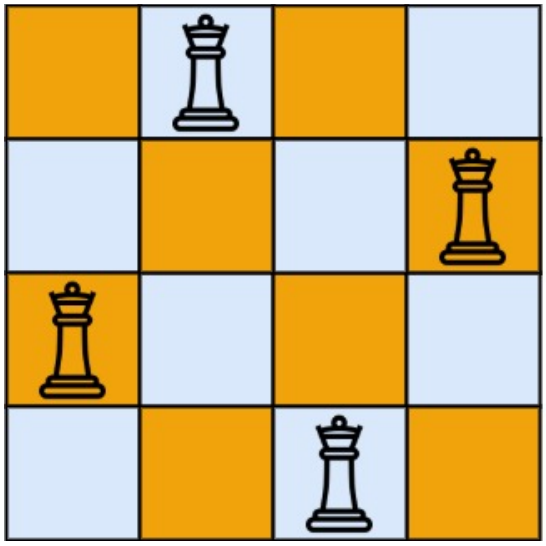
Constraints:
1 <= s.length <= 16, string s contains only lowercase English letters.

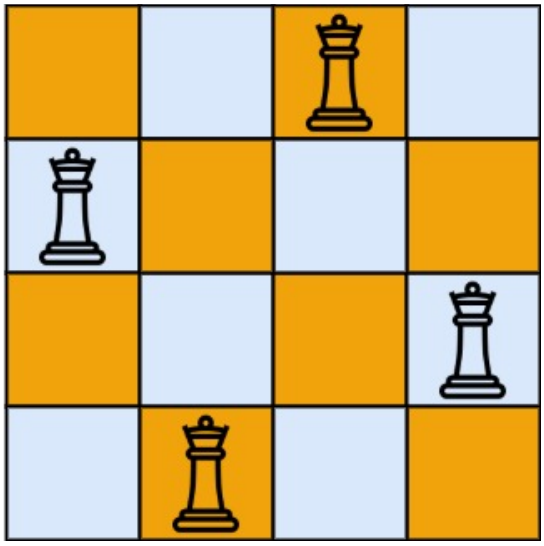
Q3. N-Queens

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.





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

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

Constraints: 1 <= n <= 9