2024-02-03 - Handout – DFS

Q1: Battleships in a Board - LeetCode

Given an m x n matrix board where each cell is a battleship 'X' or empty '.', return the number of the battleships on board.

Battleships can only be placed horizontally or vertically on board. In other words, they can only be made of the shape 1 x k (1 row, k columns) or k x 1 (k rows, 1 column), where k can be of any size. At least one horizontal or vertical cell separates between two battleships (i.e., there are no adjacent battleships).

Example 1:

X			X
			X
			X

Input: board = [["X",".",".","X"],[".",".",".","X"],[".",".",".","X"]]

Example 2:

Input: board = [["."]] Output: 0 Constraints:

m == board.length

n == board[i].length

1 <= m, n <= 200 board[i][j] is either '.' or 'X'.

Q2: Detect Cycles in 2D Grid

Given a 2D array of characters grid of size m x n, you need to find if there exists any cycle consisting of the same value in grid.

A cycle is a path of length 4 or more in the grid that starts and ends at the same cell. From a given cell, you can move to one of the cells adjacent to it - in one of the four directions (up, down, left, or right), if it has the same value of the current cell.

Also, you cannot move to the cell that you visited in your last move. For example, the cycle (1, 1) -> (1, 2) -> (1, 1) is invalid because from (1, 2) we visited (1, 1) which was the last visited cell.

Return true if any cycle of the same value exists in grid, otherwise, return false.

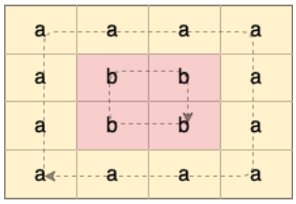
Example 1:

a	a	a	a
a	b	b	a
a	b	b	a
a	a	a	a

Input: grid = [["a","a","a","a"],["a","b","b","a"],["a","b","b","a"],["a","a","a","a"]]

Output: true

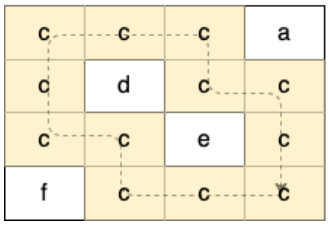
Explanation: There are two valid cycles shown in different colors in the image below:



Example 2:

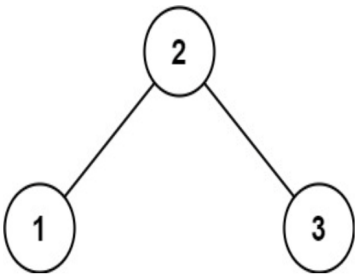
c	c	c	a
c	d	c	c
c	c	e	c
f	c	c	c

Input: grid = [["c","c","c","a"],["c","d","c","c"],["c","c","e","c"],["f","c","c","c"]]
Output: true
Explanation: There is only one valid cycle highlighted in the image below:



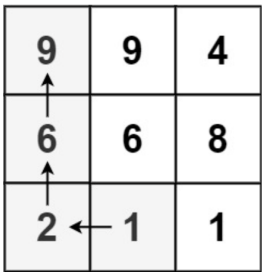
Q3. Validate Binary Search Tree
Link: https://leetcode.com/problems/validate-binary-search-tree/description/
Given the root of a binary tree, determine if it is a valid binary search tree (BST).
A valid BST is defined as follows:
The left subtree of a node contains only nodes with keys less than the node's key.
The right subtree of a node contains only nodes with keys greater than the node's key.
Both the left and right subtrees must also be binary search trees.

Example 1:




```
Input: root = [2,1,3]
Output: true
```

Q4: Longest Increasing Path in a Matrix
Link: https://leetcode.com/problems/longest-increasing-path-in-a-matrix/description/
Given an m x n integers matrix, return the length of the longest increasing path in matrix.
From each cell, you can either move in four directions: left, right, up, or down. You may not move diagonally or move outside the boundary (i.e., wrap-around is not allowed).




```
Input: matrix = [[9,9,4],[6,6,8],[2,1,1]]
Output: 4
Explanation: The longest increasing path is [1, 2, 6, 9].
```