79 Word Search

Created	@March 7, 2022 3:35 PM
Tags	Medium
link	https://leetcode.com/problems/word-search/
Problem Number	79
companies	amazon bloomberg facebook karat microsoft twitter

Description

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 neighboring. The same letter cell may not be used more than once.

```
# iterate from each cell in bfs fashion
class Solution:
    def valid_row_col(self, board, row: int, col: int, visited: Dict) -> bool:
        if row >= len(board) or row < 0:
            return False

        if col >= len(board[0]) or col < 0:
            return False

        if (row, col) in visited and visited[(row, col)]:
            return False

        return True

    def find_word(self, board: List[List[str]], row: int, col: int, word: str, idx: int, visited: Dict):
        if idx >= len(word):
            return True

        for (r, c) in [(row+1,col), (row-1,col), (row,col-1), (row,col+1)]:
            if self.valid_row_col(board, r, c, visited) and board[r][c] == word[idx]:
                visited[(r,c)] = True
                if self.find_word(board, r, c, word, idx+1, visited):
                    return True
                visited[r,c] = False
        return False


    def exist(self, board: List[List[str]], word: str) -> bool:
        row = len(board)
        col = len(board[0])
        for i in range(0, row):
            for j in range(0, col):
                if board[i][j] == word[0]:
                    if self.find_word(board, i, j, word, 1, {(i,j): True}):
                        return True

        return False
```