**Justin\_Array\_Backtracking\_0079.**  **Word Search**

**Concept:**

把每個字都走訪一次，

如果走訪到外面或 value 不等於正在找的值，

則直接return False

**Code:**

class Solution:

def exist(self, board: List[List[str]], word: str) -> bool:

self.ROWS = len(board)

self.COLS = len(board[0])

self.board = board

for row in range(self.ROWS):

for col in range(self.COLS):

if self.backtrack(row, col, word):

return True

return False

def backtrack(self, row, col, suffix):

if len(suffix) == 0:

return True

if row < 0 or row == self.ROWS or col < 0 or col == self.COLS \

or self.board[row][col] != suffix[0]:

return False

ret = False

self.board[row][col] = '#'

for rowOffset, colOffset in [(0, 1), (1, 0), (0, -1), (-1, 0)]:

ret = self.backtrack(row + rowOffset, col + colOffset, suffix[1:])

if ret: break

self.board[row][col] = suffix[0]

return ret