class Solution:

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

rows, cols = len(board), len(board[0])

visited = set()

def dfs(r, c, k):

if k == len(word):

return True

if not (0 <= r < rows) or not (0 <= c < cols) or (r,c) in visited or board[r][c] != word[k]:

return False

visited.add((r,c))

res = dfs(r+1, c, k+1) or dfs(r-1, c, k+1) or dfs(r, c+1, k+1) or dfs(r, c-1, k+1)

visited.remove((r,c))

return res

count = {}

for c in word:

count[c] = 1 + count.get(c, 0)

if count[word[0]] > count[word[-1]]:

word = word[::-1]

for r in range(rows):

for c in range(cols):

if dfs(r, c, 0): return True

return False