class Solution {

public:

bool exist(vector<vector<char>>& board, string word) {

if(board.size() == 0|| word.size() == 0) return true;

bool find = false;

vector<vector<int>>visited(board.size(), vector<int>(board[0].size(),0));

for(int i = 0 ; i < board.size(); i++)

for(int j = 0; j < board[0].size(); j++){

if(board[i][j] == word[0]){

find = backtrack(i, j, board, word, visited);

if(find) return find;

}

}

return find;

}

bool backtrack(int i, int j, vector<vector<char>>& board, string word, vector<vector<int>>& visited){

if(word.size() == 0) return true;

if(i < 0 || i > board.size() - 1 || j < 0 || j > board[0].size() - 1) return false;

if(visited[i][j]) return false;

bool find = false;

if(board[i][j] == word[0]){

word.erase(word.begin());

visited[i][j] = 1;

find = backtrack(i-1,j,board,word,visited) || backtrack(i,j-1,board,word,visited)

|| backtrack(i+1,j,board,word,visited) || backtrack(i,j+1,board,word,visited);

visited[i][j] = 0;

}

return find;

}

};