/\*

Given a 2D board and a word, find if the word exists in the grid.

The word can be constructed from letters of sequentially adjacent cell, where "adjacent" cells are those horizontally or vertically neighboring. The same letter cell may not be used more than once.

For example,

Given board =

[

['A','B','C','E'],

['S','F','C','S'],

['A','D','E','E']

]

word = "ABCCED", -> returns true,

word = "SEE", -> returns true,

word = "ABCB", -> returns false.

way-1:递归回溯,有点DFS的思想

\*/

class Solution {

public:

bool exist(vector<vector<char>>& board,int x,int y, string word,int index)

{

if(index==word.size())//都匹配完了

return true;

if(x<0 || y<0 || x>=board.size() || y>=board[0].size() || board[x][y]!=word[index])

return false;

char cun=board[x][y];

board[x][y]='0';

if(exist(board,x-1,y,word,index+1) || exist(board,x+1,y,word,index+1) || exist(board,x,y-1,word,index+1) || exist(board,x,y+1,word,index+1))

return true;

board[x][y]=cun;

return false;

}

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

{

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

{

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

{

if(exist(board,i,j,word,0))

return true;

}

}

return false;

}

};