



Description

Solution

Discuss (999+)

Submissions

C#

## 79. Word Search

Medium

8110

304

Add to List

Share

Given an  $m \times n$  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.

## Example 1:

A	B	C	E
S	F	C	S
A	D	E	E

Input: `board = [["A","B","C","E"],["S","F","C","S"],["A","D","E","E"]]`, `word = "ABCCED"`

Output: `true`

## Example 2:

A	B	C	E
S	F	C	S
A	D	E	E

```

1 public class Solution {
2
3     static int Rows;
4     static int Cols;
5     static char[][]
inputBoard;
6
7     public bool Exist(c
board, string word) {
8         inputBoard = bo
9         Rows = boar
10        Cols =
board[0].Length;
11
12        for (var i
Rows; i++)
13        {
14            for (va
j < Cols; j++)
15            {
16                if(
j, word, 0))
17                {
18                    true;
19                }
20            }
21        }
22        return fals
23    }
24
25    static bool Search(
int col, string word, i
26    {
27        /* Step 1).
28        the bottom case */

```

Testcase

Run Code Result

Accepted

Runtime: 179 ms

Your input

```

[["A","B","C","E"],
["S","F","C","S"],
["A","D","E","E"]]

```

Output

true

Expected

true

Console

Use Example Testcase

Problems

Pick One

&lt; Prev

4/30

Next &gt;

Run Code ^

Submit