Problem 2018: Check if Word Can Be Placed In Crossword

Problem Information

Difficulty: Medium
Acceptance Rate: 50.46%
Paid Only: No
Tags: Array, Matrix, Enumeration

Problem Description

You are given an `m x n` matrix `board`, representing the**current** state of a crossword puzzle. The crossword contains lowercase English letters (from solved words), `' '` to represent any **empty** cells, and `'#'` to represent any **blocked** cells.

A word can be placed**horizontally** (left to right **or** right to left) or **vertically** (top to bottom **or** bottom to top) in the board if:

* It does not occupy a cell containing the character `'#'`. * The cell each letter is placed in must either be `' '` (empty) or **match** the letter already on the `board`. * There must not be any empty cells `' '` or other lowercase letters **directly left or right****** of the word if the word was placed **horizontally**. * There must not be any empty cells `' '` or other lowercase letters **directly above or below** the word if the word was placed **vertically**.

Given a string `word`, return `true` _if_`word` _can be placed in_`board` _, or_`false` _**otherwise**_.

Example 1:

![](https://assets.leetcode.com/uploads/2021/10/04/crossword-ex1-1.png)

Input: board = [["#", " ", "#"], [" ", " ", "#"], ["#", "c", " "]], word = "abc" **Output:** true **Explanation:** The word "abc" can be placed as shown above (top to bottom).

Example 2:

![](https://assets.leetcode.com/uploads/2021/10/04/crossword-ex2-1.png)

Input: board = [[" ", "#", "a"], [" ", "#", "c"], [" ", "#", "a"]], word = "ac" **Output:** false **Explanation:** It is impossible to place the word because there will always be a space/letter above or below it.

Example 3:

![](https://assets.leetcode.com/uploads/2021/10/04/crossword-ex3-1.png)

Input: board = [["#", " ", "#"], [" ", " ", "#"], ["#", " ", "c"]], word = "ca" **Output:** true **Explanation:** The word "ca" can be placed as shown above (right to left).

Constraints:

* `m == board.length` * `n == board[i].length` * `1 <= m * n <= 2 * 105` * `board[i][j]` will be `' '`, `'#'`, or a lowercase English letter. * `1 <= word.length <= max(m, n)` * `word` will contain only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
bool placeWordInCrossword(vector<vector<char>>& board, string word) {

}
};
```

Java:

```
class Solution {
public boolean placeWordInCrossword(char[][] board, String word) {

}
}
```

Python3:

```
class Solution:
def placeWordInCrossword(self, board: List[List[str]], word: str) -> bool:
```