



链滴

矩阵中的路径

作者: [yudake](#)

原文链接: <https://ld246.com/article/1519090809333>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

题目描述

请设计一个函数，用来判断在一个矩阵中是否存在一条包含某字符串所有字符的路径。路径可以从矩阵中的任意一个格子开始，每一步可以在矩阵中向左，向右，向上，向下移动一个格子。如果一条路径经过了矩阵中的某一个格子，则该路径不能再进入该格子。例如 a b c e s f c s a d e e 矩阵中包含一字符串"bcced"的路径，但是矩阵中不包含"abcb"路径，因为字符串的第一个字符b占据了矩阵中的一行第二个格子之后，路径不能再次进入该格子。

解题思路

找到矩阵中任意一个满足条件的点，可能有多个，取或。

需要利用一个boolean型数组，标记相应点是否走过，并且利用回溯法，在进行下一个情况时，需要复标记。

代码

```
public class Solution {
    public boolean hasPath(char[] matrix, int rows, int cols, char[] str) {
        boolean[] visited = new boolean[matrix.length];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                if (searchFromHere(matrix, rows, cols, i, j, 0, str, visited))
                    return true;
            }
        }
        return false;
    }

    public boolean searchFromHere(char[] matrix, int rows, int cols, int r, int c, int index, char[] str, boolean[] visited) {
        if (r < 0 || r >= rows || c < 0 || c >= cols || matrix[r * cols + c] != str[index] || visited[r * cols + c])
            return false;
        if (index == str.length - 1)
            return true;
        visited[r * cols + c] = true;
        if (searchFromHere(matrix, rows, cols, r - 1, c, index + 1, str, visited) ||
            searchFromHere(matrix, rows, cols, r, c - 1, index + 1, str, visited) ||
            searchFromHere(matrix, rows, cols, r + 1, c, index + 1, str, visited) ||
            searchFromHere(matrix, rows, cols, r, c + 1, index + 1, str, visited))
            return true;
        visited[r * cols + c] = false;
        return false;
    }
}
```