# 题目

设计一个使用单词列表进行初始化的数据结构，单词列表中的单词互不相同。 如果给出一个单词，请判定能否只将这个单词中一个字母换成另一个字母，使得所形成的新单词存在于已构建的神奇字典中。

实现 MagicDictionary 类：

MagicDictionary() 初始化对象

void buildDict(String[] dictionary) 使用字符串数组 dictionary 设定该数据结构，dictionary 中的字符串互不相同

bool search(String searchWord) 给定一个字符串 searchWord ，判定能否只将字符串中 一个 字母换成另一个字母，使得所形成的新字符串能够与字典中的任一字符串匹配。如果可以，返回 true ；否则，返回 false 。

示例：

输入

inputs = ["MagicDictionary", "buildDict", "search", "search", "search", "search"]

inputs = [[], [["hello", "leetcode"]], ["hello"], ["hhllo"], ["hell"], ["leetcoded"]]

输出

[null, null, false, true, false, false]

解释

MagicDictionary magicDictionary = new MagicDictionary();

magicDictionary.buildDict(["hello", "leetcode"]);

magicDictionary.search("hello"); // 返回 False

magicDictionary.search("hhllo"); // 将第二个 'h' 替换为 'e' 可以匹配 "hello" ，所以返回 True

magicDictionary.search("hell"); // 返回 False

magicDictionary.search("leetcoded"); // 返回 False

提示：

1 <= dictionary.length <= 100

1 <= dictionary[i].length <= 100

dictionary[i] 仅由小写英文字母组成

dictionary 中的所有字符串 互不相同

1 <= searchWord.length <= 100

searchWord 仅由小写英文字母组成

buildDict 仅在 search 之前调用一次

最多调用 100 次 search

注意：本题与主站 676 题相同： <https://leetcode-cn.com/problems/implement-magic-dictionary/>

# 分析