# 题目

给你一个字符串数组 words ，数组中的每个字符串都可以看作是一个单词。请你按 任意 顺序返回 words 中是其他单词的子字符串的所有单词。

如果你可以删除 words[j] 最左侧和/或最右侧的若干字符得到 words[i] ，那么字符串 words[i] 就是 words[j] 的一个子字符串。

示例 1：

输入：words = ["mass","as","hero","superhero"]

输出：["as","hero"]

解释："as" 是 "mass" 的子字符串，"hero" 是 "superhero" 的子字符串。

["hero","as"] 也是有效的答案。

示例 2：

输入：words = ["leetcode","et","code"]

输出：["et","code"]

解释："et" 和 "code" 都是 "leetcode" 的子字符串。

示例 3：

输入：words = ["blue","green","bu"]

输出：[]

提示：

1 <= words.length <= 100

1 <= words[i].length <= 30

words[i] 仅包含小写英文字母。

题目数据 保证 每个 words[i] 都是独一无二的。

# 分析

## 方法一：暴力枚举

思路：

代码：

class Solution {

public:

vector<string> stringMatching(vector<string>& words) {

vector<string> ret;

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

// 从头开始遍历一遍数组查找

for (int j = 0; j < words.size(); j++) {

if (i != j && words[j].find(words[i]) != string::npos) {

ret.push\_back(words[i]);

break;

}

}

}

return ret;

}

};