# 题目

给你一个下标从 0 开始的数组 words ，数组中包含 互不相同 的字符串。

如果字符串 words[i] 与字符串 words[j] 满足以下条件，我们称它们可以匹配：

* 字符串 words[i] 等于 words[j] 的反转字符串。
* 0 <= i < j < words.length

请你返回数组 words 中的 最大 匹配数目。

注意，每个字符串最多匹配一次。

示例 1：

输入：words = ["cd","ac","dc","ca","zz"]

输出：2

解释：在此示例中，我们可以通过以下方式匹配 2 对字符串：

- 我们将第 0 个字符串与第 2 个字符串匹配，因为 word[0] 的反转字符串是 "dc" 并且等于 words[2]。

- 我们将第 1 个字符串与第 3 个字符串匹配，因为 word[1] 的反转字符串是 "ca" 并且等于 words[3]。

可以证明最多匹配数目是 2 。

示例 2：

输入：words = ["ab","ba","cc"]

输出：1

解释：在此示例中，我们可以通过以下方式匹配 1 对字符串：

- 我们将第 0 个字符串与第 1 个字符串匹配，因为 words[1] 的反转字符串 "ab" 与 words[0] 相等。

可以证明最多匹配数目是 1 。

示例 3：

输入：words = ["aa","ab"]

输出：0

解释：这个例子中，无法匹配任何字符串。

提示：

1 <= words.length <= 50

words[i].length == 2

words 包含的字符串互不相同。

words[i] 只包含小写英文字母。

# 分析

## 方法一：模拟/枚举

思路：

我们可以直接使用二重循环，枚举给定的数组words中的words[i]和words[j]是否可以匹配。

由于题目规定了数组words中包含的字符串互不相同，因此在枚举时，只要保证i<j，那么每个字符串最多匹配一次。

代码：

class Solution {

public:

int maximumNumberOfStringPairs(vector<string>& words) {

int n = words.size();

int ans = 0;

for (int i = 0; i < n; ++i) {

for (int j = i + 1; j < n; ++j) {

if (words[i][0] == words[j][1] && words[i][1] == words[j][0]) {

++ans;

}

}

}

return ans;

}

};

## 方法二：哈希