# 题目

给你一个字符串数组words（下标从0开始计数）。

在一步操作中，需先选出两个不同下标i和j，其中words[i]是一个非空字符串，接着将words[i]中的任一字符移动到words[j]中的任一位置上。

如果执行任意步操作可以使words中的每个字符串都相等，返回true；否则，返回false。

示例 1：

输入：words = ["abc","aabc","bc"]

输出：true

解释：将 words[1] 中的第一个 'a' 移动到 words[2] 的最前面。

使 words[1] = "abc" 且 words[2] = "abc" 。

所有字符串都等于 "abc" ，所以返回 true 。

示例 2：

输入：words = ["ab","a"]

输出：false

解释：执行操作无法使所有字符串都相等。

提示：

1 <= words.length <= 100

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

words[i] 由小写英文字母组成

# 分析

## 方法一：统计词频

思路：

我们需要统计每个字符在所有字符串中出现的次数，并检查是否每个字符的出现次数都是偶数。如果是，则可以通过移动字符使所有字符串相等，返回true；否则，返回false。

代码：

class Solution {

public:

bool makeEqual(vector<string>& words) {

unordered\_map<char, int> charCount;

// 统计每个字符的出现次数

for (string& word : words) {

for (char c : word) {

charCount[c]++;

}

}

// 检查每个字符的出现次数是否是字符串数目的倍数。如果有任何一个字符的出现次数不是字符串数目的倍数，则返回 false；否则，返回 true

for (auto& pair : charCount) {

if (pair.second % words.size() != 0) {

return false;

}

}

return true;

}

};