Problem 1897: Redistribute Characters to Make All Strings Equal

Problem Information

Difficulty: Easy
Acceptance Rate: 66.78%
Paid Only: No
Tags: Hash Table, String, Counting

Problem Description

You are given an array of strings `words` (**0-indexed**).

In one operation, pick two **distinct** indices `i` and `j`, where `words[i]` is a non-empty string, and move **any** character from `words[i]` to **any** position in `words[j]`.

Return `true` _if you can make**every** string in _`words` _**equal** using **any** number of operations_,_and_`false` _otherwise_.

Example 1:

Input: words = ["abc","aabc","bc"] **Output:** true **Explanation:** Move the first 'a' in words[1] to the front of words[2], to make words[1] = "abc" and words[2] = "abc". All the strings are now equal to "abc", so return true.

Example 2:

Input: words = ["ab","a"] **Output:** false **Explanation:** It is impossible to make all the strings equal using the operation.

Constraints:

* `1 <= words.length <= 100` * `1 <= words[i].length <= 100` * `words[i]` consists of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
bool makeEqual(vector<string>& words) {

}
};
```

Java:

```
class Solution {
public boolean makeEqual(String[] words) {

}
}
```

Python3:

```
class Solution:
def makeEqual(self, words: List[str]) -> bool:
```