Problem 2157: Groups of Strings

Problem Information

Difficulty: Hard
Acceptance Rate: 27.13%
Paid Only: No
Tags: String, Bit Manipulation, Union Find

Problem Description

You are given a **0-indexed** array of strings `words`. Each string consists of **lowercase English letters** only. No letter occurs more than once in any string of `words`.

Two strings `s1` and `s2` are said to be **connected** if the set of letters of `s2` can be obtained from the set of letters of `s1` by any **one** of the following operations:

* Adding exactly one letter to the set of the letters of `s1`. * Deleting exactly one letter from the set of the letters of `s1`. * Replacing exactly one letter from the set of the letters of `s1` with any letter, **including** itself.

The array `words` can be divided into one or more non-intersecting **groups**. A string belongs to a group if any **one** of the following is true:

* It is connected to **at least one** other string of the group. * It is the **only** string present in the group.

Note that the strings in `words` should be grouped in such a manner that a string belonging to a group cannot be connected to a string present in any other group. It can be proved that such an arrangement is always unique.

Return _an array_ `ans` _of size_ `2` _where:_

* `ans[0]` _is the**maximum number** of groups_ `words` _can be divided into, and_ * `ans[1]` _is the**size of the largest** group_.

Example 1:

Input: words = ["a","b","ab","cde"] **Output:** [2,3] **Explanation:** - words[0] can be used to obtain words[1] (by replacing 'a' with 'b'), and words[2] (by adding 'b'). So words[0] is connected to words[1] and words[2]. - words[1] can be used to obtain words[0] (by replacing 'b' with 'a'), and words[2] (by adding 'a'). So words[1] is connected to words[0] and words[2]. - words[2] can be used to obtain words[0] (by deleting 'b'), and words[1] (by deleting 'a'). So words[2] is connected to words[0] and words[1]. - words[3] is not connected to any string in words. Thus, words can be divided into 2 groups ["a","b","ab"] and ["cde"]. The size of the largest group is 3.

Example 2:

Input: words = ["a","ab","abc"] **Output:** [1,3] **Explanation:** - words[0] is connected to words[1]. - words[1] is connected to words[0] and words[2]. - words[2] is connected to words[1]. Since all strings are connected to each other, they should be grouped together. Thus, the size of the largest group is 3.

Constraints:

* `1 <= words.length <= 2 * 104` * `1 <= words[i].length <= 26` * `words[i]` consists of lowercase English letters only. * No letter occurs more than once in `words[i]`.

Code Snippets

C++:

```
class Solution {
public:
vector<int> groupStrings(vector<string>& words) {

}
};
```

Java:

```
class Solution {
public int[] groupStrings(String[] words) {

}
}
```

Python3:

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