Problem 522: Longest Uncommon Subsequence II

Problem Information

Difficulty: Medium
Acceptance Rate: 44.02%
Paid Only: No
Tags: Array, Hash Table, Two Pointers, String, Sorting

Problem Description

Given an array of strings `strs`, return _the length of the**longest uncommon subsequence** between them_. If the longest uncommon subsequence does not exist, return `-1`.

An **uncommon subsequence** between an array of strings is a string that is a **subsequence of one string but not the others**.

A **subsequence** of a string `s` is a string that can be obtained after deleting any number of characters from `s`.

* For example, `"abc"` is a subsequence of `"aebdc"` because you can delete the underlined characters in `"a _e_ b _d_ c"` to get `"abc"`. Other subsequences of `"aebdc"` include `"aebdc"`, `"aeb"`, and `""` (empty string).

Example 1:

Input: strs = ["aba","cdc","eae"] **Output:** 3

Example 2:

Input: strs = ["aaa","aaa","aa"] **Output:** -1

Constraints:

* `2 <= strs.length <= 50` * `1 <= strs[i].length <= 10` * `strs[i]` consists of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int findLUSlength(vector<string>& strs) {

}
};
```

Java:

```
class Solution {
public int findLUSlength(String[] strs) {

}
}
```

Python3:

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