

522. Longest Uncommon Subsequence II

Add to List

Description	Hints Submissions Solutions
-----------------------------	---

- Total Accepted: 3145
- Total Submissions: 10588
- Difficulty: Medium
- Contributors: [love_Fawn](#)

Given a list of strings, you need to find the longest uncommon subsequence among them.

The longest uncommon subsequence is defined as the longest subsequence of one of these strings and this subsequence should not be **any** subsequence of the other strings.

A **subsequence** is a sequence that can be derived from one sequence by deleting some characters without changing the order of the remaining elements. Trivially, any string is a subsequence of itself and an empty string is a subsequence of any string.

The input will be a list of strings, and the output needs to be the length of the longest uncommon subsequence. If the longest uncommon subsequence doesn't exist, return -1.

Example 1:

Input: "aba", "cdc", "eae"

Output: 3

Note:

1. All the given strings' lengths will not exceed 10.
2. The length of the given list will be in the range of [2, 50].

[Subscribe](#) to see which companies asked this question.

```
class Solution {
public:
    bool isSubSequence(string s,int si,string t, int ti)
```

```

{
    if(s.size()==si) return true;
    if(t.size()==ti) return false;
    return ((s[si]==t[ti]) && isSubSequence(s,si+1,t,ti+1) ||
isSubSequence(s,si,t,ti+1));
}

int findLUSlength(vector<string>& strs) {
    int n = strs.size();
    int maxlen = -1;
    for (int i = 0; i < n; i++) {
        bool uniq = true;
        for (int j = 0; j < n; j++) {
            if (i != j && isSubSequence(strs[i], 0,strs[j], 0)) {
                uniq = false;
                break;
            }
        }
        if (uniq) {
            maxlen = max(maxlen, (int)strs[i].length());
        }
    }

    return maxlen;
}

};

```