Problem 3435: Frequencies of Shortest Supersequences

Problem Information

Difficulty: Hard
Acceptance Rate: 19.48%
Paid Only: No
Tags: Array, String, Bit Manipulation, Graph, Topological Sort, Enumeration

Problem Description

You are given an array of strings `words`. Find all **shortest common supersequences (SCS)** of `words` that are not permutations of each other.

A **shortest common supersequence** is a string of **minimum** length that contains each string in `words` as a subsequence.

Return a 2D array of integers `freqs` that represent all the SCSs. Each `freqs[i]` is an array of size 26, representing the frequency of each letter in the lowercase English alphabet for a single SCS. You may return the frequency arrays in any order.

Example 1:

Input: words = ["ab","ba"]

Output:[[1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]]

Explanation:

The two SCSs are `"aba"` and `"bab"`. The output is the letter frequencies for each one.

Example 2:

Input: words = ["aa","ac"]

Output:[[2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]]

Explanation:

The two SCSs are `"aac"` and `"aca"`. Since they are permutations of each other, keep only `"aac"`.

Example 3:

Input: words = ["aa","bb","cc"]

Output:[[2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]]

Explanation:

`"aabbcc"` and all its permutations are SCSs.

Constraints:

* `1 <= words.length <= 256` * `words[i].length == 2` * All strings in `words` will altogether be composed of no more than 16 unique lowercase letters. * All strings in `words` are unique.

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public List<List<Integer>> supersequences(String[] words) {

}
```

```
}
```

Python3:

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