Problem 1452: People Whose List of Favorite Companies Is Not a Subset of Another List

Problem Information

Difficulty: Medium
Acceptance Rate: 60.29%
Paid Only: No
Tags: Array, Hash Table, String

Problem Description

Given the array `favoriteCcompanies` where `favoriteCompanies[i]` is the list of favorites companies for the `ith` person (**indexed from 0**).

Return the indices of people whose list of favorite companies is not a**subset** of any other list of favorites companies. You must return the indices in increasing order.

Example 1:

Input: favoriteCompanies = [["leetcode","google","facebook"],["google","microsoft"],["google","facebook"],["google"],["amazon"]] **Output:** [0,1,4] **Explanation:** Person with index=2 has favoriteCompanies[2]=["google","facebook"] which is a subset of favoriteCompanies[0]=["leetcode","google","facebook"] corresponding to the person with index 0. Person with index=3 has favoriteCompanies[3]=["google"] which is a subset of favoriteCompanies[0]=["leetcode","google","facebook"] and favoriteCompanies[1]=["google","microsoft"]. Other lists of favorite companies are not a subset of another list, therefore, the answer is [0,1,4].

Example 2:

Input: favoriteCompanies = [["leetcode","google","facebook"],["leetcode","amazon"],["facebook","google"]] **Output:** [0,1] **Explanation:** In this case favoriteCompanies[2]=["facebook","google"] is a subset of favoriteCompanies[0]=["leetcode","google","facebook"], therefore, the answer is [0,1].

Example 3:

Input: favoriteCompanies = [["leetcode"],["google"],["facebook"],["amazon"]] **Output:** [0,1,2,3]

Constraints:

* `1 <= favoriteCompanies.length <= 100` * `1 <= favoriteCompanies[i].length <= 500` * `1 <= favoriteCompanies[i][j].length <= 20` * All strings in `favoriteCompanies[i]` are **distinct**. * All lists of favorite companies are **distinct** , that is, If we sort alphabetically each list then `favoriteCompanies[i] != favoriteCompanies[j].` * All strings consist of lowercase English letters only.

Code Snippets

C++:

```
class Solution {
public:
vector<int> peopleIndexes(vector<vector<string>>& favoriteCompanies) {

}
};
```

Java:

```
class Solution {
public List<Integer> peopleIndexes(List<List<String>> favoriteCompanies) {

}
}
```

Python3:

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