

Problem 2115: Find All Possible Recipes from Given Supplies

Problem Information

Difficulty: Medium

Acceptance Rate: 56.58%

Paid Only: No

Tags: Array, Hash Table, String, Graph, Topological Sort

Problem Description

You have information about `n` different recipes. You are given a string array `recipes` and a 2D string array `ingredients`. The `ith` recipe has the name `recipes[i]`, and you can **create** it if you have **all** the needed ingredients from `ingredients[i]`. A recipe can also be an ingredient for **other** recipes, i.e., `ingredients[i]` may contain a string that is in `recipes`.

You are also given a string array `supplies` containing all the ingredients that you initially have, and you have an infinite supply of all of them.

Return _a list of all the recipes that you can create._ You may return the answer in **any order**.

Note that two recipes may contain each other in their ingredients.

Example 1:

Input: recipes = ["bread"], ingredients = [["yeast", "flour"]], supplies = ["yeast", "flour", "corn"]
Output: ["bread"]
Explanation: We can create "bread" since we have the ingredients "yeast" and "flour".

Example 2:

Input: recipes = ["bread", "sandwich"], ingredients = [["yeast", "flour"], ["bread", "meat"]], supplies = ["yeast", "flour", "meat"]
Output: ["bread", "sandwich"]
Explanation: We can create "bread" since we have the ingredients "yeast" and "flour". We can create "sandwich" since we have the ingredient "meat" and can create the ingredient "bread".

****Example 3:****

****Input:**** recipes = ["bread", "sandwich", "burger"], ingredients = [[["yeast", "flour"], ["bread", "meat"], ["sandwich", "meat", "bread"]]], supplies = ["yeast", "flour", "meat"] ****Output:**** ["bread", "sandwich", "burger"] ****Explanation:**** We can create "bread" since we have the ingredients "yeast" and "flour". We can create "sandwich" since we have the ingredient "meat" and can create the ingredient "bread". We can create "burger" since we have the ingredient "meat" and can create the ingredients "bread" and "sandwich".

****Constraints:****

* `n == recipes.length == ingredients.length` * `1 <= n <= 100` * `1 <= ingredients[i].length, supplies.length <= 100` * `1 <= recipes[i].length, ingredients[i][j].length, supplies[k].length <= 10` * `recipes[i], ingredients[i][j]` , and `supplies[k]` consist only of lowercase English letters. * All the values of `recipes` and `supplies` combined are unique. * Each `ingredients[i]` does not contain any duplicate values.

Code Snippets

C++:

```
class Solution {
public:
    vector<string> findAllRecipes(vector<string>& recipes,
    vector<vector<string>>& ingredients, vector<string>& supplies) {
        }
    };
```

Java:

```
class Solution {
public List<String> findAllRecipes(String[] recipes, List<List<String>>
    ingredients, String[] supplies) {
        }
    }
```

Python3:

```
class Solution:

def findAllRecipes(self, recipes: List[str], ingredients: List[List[str]], supplies: List[str]) -> List[str]:
```