Problem 3481: Apply Substitutions

Problem Information

Difficulty: Medium
Acceptance Rate: 77.54%
Paid Only: Yes
Tags: Array, Hash Table, String, Depth-First Search, Breadth-First Search, Graph, Topological Sort

Problem Description

You are given a `replacements` mapping and a `text` string that may contain **placeholders** formatted as `%var%`, where each `var` corresponds to a key in the `replacements` mapping. Each replacement value may itself contain **one or more** such **placeholders**. Each **placeholder** is replaced by the value associated with its corresponding replacement key.

Return the fully substituted `text` string which **does not** contain any **placeholders**.

Example 1:

Input: replacements = [["A","abc"],["B","def"]], text = "%A%_%B%"

Output: "abc_def"

Explanation:

* The mapping associates `"A"` with `"abc"` and `"B"` with `"def"`. * Replace `%A%` with `"abc"` and `%B%` with `"def"` in the text. * The final text becomes `"abc_def"`.

Example 2:

Input: replacements = [["A","bce"],["B","ace"],["C","abc%B%"]], text = "%A%_%B%_%C%"

Output: "bce_ace_abcace"

Explanation:

* The mapping associates `"A"` with `"bce"`, `"B"` with `"ace"`, and `"C"` with `"abc%B%"`. * Replace `%A%` with `"bce"` and `%B%` with `"ace"` in the text. * Then, for `%C%`, substitute `%B%` in `"abc%B%"` with `"ace"` to obtain `"abcace"`. * The final text becomes `"bce_ace_abcace"`.

Constraints:

* `1 <= replacements.length <= 10` * Each element of `replacements` is a two-element list `[key, value]`, where: * `key` is a single uppercase English letter. * `value` is a non-empty string of at most 8 characters that may contain zero or more placeholders formatted as `%<key>%`. * All replacement keys are unique. * The `text` string is formed by concatenating all key placeholders (formatted as `%<key>%`) randomly from the replacements mapping, separated by underscores. * `text.length == 4 * replacements.length - 1` * Every placeholder in the `text` or in any replacement value corresponds to a key in the `replacements` mapping. * There are no cyclic dependencies between replacement keys.

Code Snippets

C++:

```
class Solution {
public:
string applySubstitutions(vector<vector<string>>& replacements, string text)
{

}
};
```

Java:

```
class Solution {
public String applySubstitutions(List<List<String>> replacements, String
text) {

}
}
```

Python3:

```
class Solution:
def applySubstitutions(self, replacements: List[List[str]], text: str) ->
str:
```