Problem 1807: Evaluate the Bracket Pairs of a String

Problem Information

Difficulty: Medium
Acceptance Rate: 68.83%
Paid Only: No
Tags: Array, Hash Table, String

Problem Description

You are given a string `s` that contains some bracket pairs, with each pair containing a **non-empty** key.

* For example, in the string `"(name)is(age)yearsold"`, there are **two** bracket pairs that contain the keys `"name"` and `"age"`.

You know the values of a wide range of keys. This is represented by a 2D string array `knowledge` where each `knowledge[i] = [keyi, valuei]` indicates that key `keyi` has a value of `valuei`.

You are tasked to evaluate **all** of the bracket pairs. When you evaluate a bracket pair that contains some key `keyi`, you will:

* Replace `keyi` and the bracket pair with the key's corresponding `valuei`. * If you do not know the value of the key, you will replace `keyi` and the bracket pair with a question mark `"?"` (without the quotation marks).

Each key will appear at most once in your `knowledge`. There will not be any nested brackets in `s`.

Return _the resulting string after evaluating**all** of the bracket pairs._

Example 1:

Input: s = "(name)is(age)yearsold", knowledge = [["name","bob"],["age","two"]] **Output:** "bobistwoyearsold" **Explanation:** The key "name" has a value of "bob", so replace "(name)" with "bob". The key "age" has a value of "two", so replace "(age)" with "two".

Example 2:

Input: s = "hi(name)", knowledge = [["a","b"]] **Output:** "hi?" **Explanation:** As you do not know the value of the key "name", replace "(name)" with "?".

Example 3:

Input: s = "(a)(a)(a)aaa", knowledge = [["a","yes"]] **Output:** "yesyesyesaaa" **Explanation:** The same key can appear multiple times. The key "a" has a value of "yes", so replace all occurrences of "(a)" with "yes". Notice that the "a"s not in a bracket pair are not evaluated.

Constraints:

* `1 <= s.length <= 105` * `0 <= knowledge.length <= 105` * `knowledge[i].length == 2` * `1 <= keyi.length, valuei.length <= 10` * `s` consists of lowercase English letters and round brackets `'('` and `')'`. * Every open bracket `'('` in `s` will have a corresponding close bracket `')'`. * The key in each bracket pair of `s` will be non-empty. * There will not be any nested bracket pairs in `s`. * `keyi` and `valuei` consist of lowercase English letters. * Each `keyi` in `knowledge` is unique.

Code Snippets

C++:

```
class Solution {
public:
string evaluate(string s, vector<vector<string>>& knowledge) {


}
};
```

Java:

```
class Solution {
public String evaluate(String s, List<List<String>> knowledge) {
```

```
    }
}
```

Python3:

```python
class Solution:
    def evaluate(self, s: str, knowledge: List[List[str]]) -> str:
```