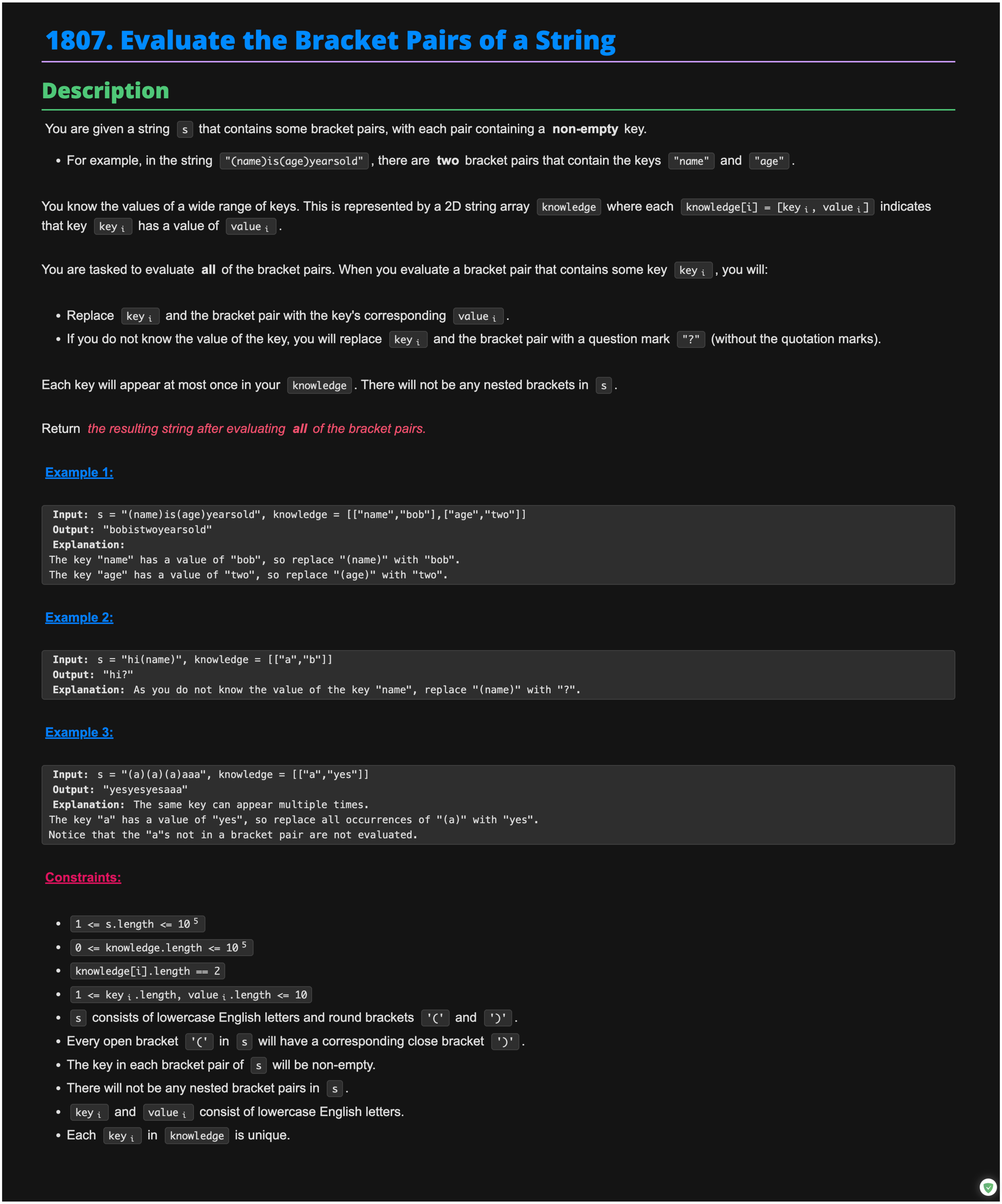

1807. Evaluate the Bracket Pairs of a String

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 $\texttt{knowledge[i] = [key}_i\texttt{, value}_i\texttt{]}$ indicates that key $\texttt{key}_i$ has a value of $\texttt{value}_i$.

You are tasked to evaluate **all** of the bracket pairs. When you evaluate a bracket pair that contains some key $\texttt{key}_i$, you will:

- Replace $\texttt{key}_i$ and the bracket pair with the key's corresponding $\texttt{value}_i$.
- If you do not know the value of the key, you will replace $\texttt{key}_i$ 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:

- $\texttt{1 <= s.length <= 10}^5$
- $\texttt{0 <= knowledge.length <= 10}^5$
- `knowledge[i].length == 2`
- $\texttt{1 <= key}_i\texttt{.length, value}_i\texttt{.length <= 10}$
- `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`.
- $\texttt{key}_i$ and $\texttt{value}_i$ consist of lowercase English letters.
- Each $\texttt{key}_i$ in `knowledge` is unique.