Problem 648: Replace Words

Problem Information

Difficulty: Medium
Acceptance Rate: 68.55%
Paid Only: No
Tags: Array, Hash Table, String, Trie

Problem Description

In English, we have a concept called **root** , which can be followed by some other word to form another longer word - let's call this word **derivative**. For example, when the **root** `"help"` is followed by the word `"ful"`, we can form a derivative `"helpful"`.

Given a `dictionary` consisting of many **roots** and a `sentence` consisting of words separated by spaces, replace all the derivatives in the sentence with the **root** forming it. If a derivative can be replaced by more than one **root** , replace it with the **root** that has **the shortest length**.

Return _the`sentence`_ after the replacement.

Example 1:

Input: dictionary = ["cat","bat","rat"], sentence = "the cattle was rattled by the battery" **Output:** "the cat was rat by the bat"

Example 2:

Input: dictionary = ["a","b","c"], sentence = "aadsfasf absbs bbab cadsfafs" **Output:** "a a b c"

Constraints:

* `1 <= dictionary.length <= 1000` * `1 <= dictionary[i].length <= 100` * `dictionary[i]` consists of only lower-case letters. * `1 <= sentence.length <= 106` * `sentence` consists of only lower-case letters and spaces. * The number of words in `sentence` is in the range `[1, 1000]`

* The length of each word in `sentence` is in the range `[1, 1000]` * Every two consecutive words in `sentence` will be separated by exactly one space. * `sentence` does not have leading or trailing spaces.

Code Snippets

C++:

```
class Solution {
public:
string replaceWords(vector<string>& dictionary, string sentence) {

}
};
```

Java:

```
class Solution {
public String replaceWords(List<String> dictionary, String sentence) {

}
}
```

Python3:

```
class Solution:
def replaceWords(self, dictionary: List[str], sentence: str) -> str:
```