

3076. Shortest Uncommon Substring in an Array

Description

You are given an array `arr` of size `n` consisting of **non-empty** strings.

Find a string array `answer` of size `n` such that:

- `answer[i]` is the **shortest** substring of `arr[i]` that does **not** occur as a substring in any other string in `arr`. If multiple such substrings exist, `answer[i]` should be the lexicographically smallest. And if no such substring exists, `answer[i]` should be an empty string.

Return *the array* `answer`.

Example 1:

Input: `arr = ["cab","ad","bad","c"]`

Output: `["ab","","ba",""]`

Explanation: We have the following:

- For the string "cab", the shortest substring that does not occur in any other string is either "ca" or "ab", we choose the lexicographically smaller substring, which is "ab".
- For the string "ad", there is no substring that does not occur in any other string.
- For the string "bad", the shortest substring that does not occur in any other string is "ba".
- For the string "c", there is no substring that does not occur in any other string.

Example 2:

Input: `arr = ["abc","bcd","abcd"]`

Output: `["","","abcd"]`

Explanation: We have the following:

- For the string "abc", there is no substring that does not occur in any other string.
- For the string "bcd", there is no substring that does not occur in any other string.
- For the string "abcd", the shortest substring that does not occur in any other string is "abcd".

Constraints:

- `n == arr.length`
- `2 <= n <= 100`
- `1 <= arr[i].length <= 20`
- `arr[i]` consists only of lowercase English letters.

