

269. Alien Dictionary

Hard 2587 504 Add to List Share

There is a new alien language that uses the English alphabet. However, the order among the letters is unknown to you.

You are given a list of strings `words` from the alien language's dictionary, where the strings in `words` are **sorted lexicographically** by the rules of this new language.

Return *a string of the unique letters in the new alien language sorted in **lexicographically increasing order** by the new language's rules. If there is no solution, return `""`. If there are multiple solutions, return **any of them**.*

A string `s` is **lexicographically smaller** than a string `t` if at the first letter where they differ, the letter in `s` comes before the letter in `t` in the alien language. If the first `min(s.length, t.length)` letters are the same, then `s` is smaller if and only if `s.length < t.length`.

Example 1:

Input: words = ["wrt","wrf","er","ett","rftt"]
Output: "wertf"

Example 2:

Input: words = ["z","x"]
Output: "zx"

Example 3:

Input: words = ["z","x","z"]
Output: ""
Explanation: The order is invalid, so return "".

Constraints:

- `1 <= words.length <= 100`
- `1 <= words[i].length <= 100`
- `words[i]` consists of only lowercase English letters.

Accepted 208,878 Submissions 615,859

Seen this question in a real interview before?

Yes

No

Companies 🔒 i

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Facebook | 26

Amazon | 11

Airbnb | 7

Bloomberg | 4

Rubrik | 4

Google | 3

Microsoft | 2

Apple | 2

ByteDance | 2

- Related Topics
- Similar Questions

```
51         #
           initialization
52     for
   word in words:
53         for
   c in word:
54             indegree[c] =
55                 0
           adjacency[c] =
56                 []
57
   num_words =
   len(words)
58     for
   index in
   range(num_words
-1):
59         word1 =
   words[index]
60
           word2 =
   words[index+1]
61
           found = False
62
           min_len =
   min(len(word1),
   len(word2))
63     for
   it in
   range(min_len):
64         char1 =
   word1[it]
65
           char2 =
   word2[it]
66     if char1 !=
   char2:
67
           # word1 is
   before word2
   and they differ
   for the first
   time at char1/2
```

Your previous code was restored from local storage

Test...

Run Code ...

Deb... 🔒

Accepted Runtime: 24 ms

Your input ["wrt","wr","z","x"]

Output

Diff

Expected "wertf","zx"