

953. Verifying an Alien Dictionary

Easy

2623

890

Add to List

Share

In an alien language, surprisingly, they also use English lowercase letters, but possibly in a different `order`. The `order` of the alphabet is some permutation of lowercase letters.

Given a sequence of `words` written in the alien language, and the `order` of the alphabet, return `true` if and only if the given `words` are sorted lexicographically in this alien language.

Example 1:

Input: words = ["hello","leetcode"], order = "hlabcdefgijklmnopqrstuvwxyz"
Output: true
Explanation: As 'h' comes before 'l' in this language, then the sequence is sorted.

Example 2:

Input: words = ["word","world","row"], order = "worldabcefg hijkmnpqstuvwxyz"
Output: false
Explanation: As 'd' comes after 'l' in this language, then words[0] > words[1], hence the sequence is unsorted.

Example 3:

Input: words = ["apple","app"], order = "abcdefghijklmnopqrstuvwxyz"
Output: false
Explanation: The first three characters "app" match, and the second string is shorter (in size.) According to lexicographical rules "apple" > "app", because 'l' > 'Ø', where 'Ø' is defined as the blank character which is less than any other character (More info).

Constraints:

- 1 <= words.length <= 100
- 1 <= words[i].length <= 20
- order.length == 26
- All characters in words[i] and order are English lowercase letters.

Accepted 334,790

Submissions 638,594

```
1  class Solution {
2
3      int[] mapping = new int[26];
4
5      public boolean isAlienSorted(String[] words, String order) {
6
7          for (int i = 0; i < order.length(); i++)
8              mapping[order.charAt(i) - 'a'] = i;
9
10         for (int i = 1; i < words.length; i++)
11             if (bigger(words[i - 1], words[i]))
12                 return false;
13
14         return true;
15     }
16
17     private boolean bigger(String s1, String s2) {
18
19         int n = s1.length(), m = s2.length();
20
21         for (int i = 0; i < n && i < m; ++i)
22             if (s1.charAt(i) != s2.charAt(i))
23                 return mapping[s1.charAt(i) - 'a'] > mapping[s2.charAt(i) -
24
25         return n > m;
26     }
27 }
```

Your previous code was restored from your local storage. [Reset to default](#)