Reverse Shuffle Merge ☆

Problem Submissions Leaderboard Editorial Topics

Given a string, A, we define some operations on the string as follows:

> a. $reverse(A)$ denotes the string obtained by reversing string A. Example: `reverse("abc") = "cba"`
>
> b. $shuffle(A)$ denotes any string that's a permutation of string A. Example:
> `shuffle("god") ∈ ['god', 'gdo', 'ogd', 'odg', 'dgo', 'dog']`
>
> c. $merge(A1, A2)$ denotes any string that's obtained by interspersing the two strings $A1$ & $A2$, maintaining the order of characters in both. For example, `A1 = "abc"` & `A2 = "def"`, one possible result of $merge(A1, A2)$ could be `"abcdef"`, another could be `"abdecf"`, another could be `"adbecf"` and so on.

Given a string s such that `s ∈ merge(reverse(A), shuffle(A))` for some string A, find the lexicographically smallest A.

For example, $s = abab$. We can split it into two strings of ab. The reverse is ba and we need to find a string to shuffle in to get $abab$. The middle two characters match our reverse string, leaving the a and b at the ends. Our shuffle string needs to be ab. Lexicographically $ab < ba$, so our answer is ab.

Function Description

Complete the *reverseShuffleMerge* function in the editor below. It must return the lexicographically smallest string fitting the criteria.

reverseShuffleMerge has the following parameter(s):

- *s*: a string

Input Format

A single line containing the string s.

Constraints

- s contains only lower-case English letters, *ascii[a-z]*
- $1 \leq |s| \leq 10000$

Output Format

Find and return the string which is the lexicographically smallest valid A.

Sample Input 0

```
eggegg
```

Sample Output 0

```
egg
```

Explanation 0

Split "eggegg" into strings of like character counts: "egg", "egg"
reverse("egg") = "gge"
shuffle("egg") can be "egg"
"eggegg" belongs to the merge of ("gge", "egg")

The merge is: ***eggegg***.

'egg' < 'gge'

Sample Input 1

```
abcdefgabcdefg
```

Sample Output 1

```
agfedcb
```

Explanation 1

Split the string into two strings with like characters: $abcdefg$ and $abcdefg$.
Reverse $abcdefg$ = $gfedcba$
Shuffle $agfedcb$ can be $bcdefga$
Merge to **a**bcdefg**a**$bcdefg$

Sample Input 2

```
aeiouuoiea
```

Sample Output 2

```
aeiou
```

Explanation 2

Split the string into groups of like characters: $aeiou$
Reverse $aeiou$ = $uoiea$
These merge to $aeiou$**uoiea**

Change Theme Python 3

```
#!/bin/python3

import math
import os
import random
import re
import sys
from queue import PriorityQueue

def findLettersDistribution(s):
    letterCounts = [0]*30
    for letter in s:
        letterCounts[ord(letter)-ord('a')] += 1

    return letterCounts

def isLettersDistributionCorrect(letterCounts):
    print(letterCounts)
    for letterCount in letterCounts:
        if letterCount % 2:
            return False

    return True
```

```
24
25  def constructStringFromLetterDistribution(s, letterCounts):
26      a = ''
27      q = PriorityQueue()
28      letters = {}
29
30      for c in s:
31          if (not c in letters) or (letters[c] == 0):
32              letters[c] = 1
33              q.put(c)
34          elif letters[c] == 1:
```

Line: 47 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Runtime Error :(

Sample Test case 0

Sample Test case 1

Sample Test case 2

Compiler Message

Your code did not execute within the time limits

Input (stdin) Download

```
1  eggegg
```

Your Output (stdout)

```
~ no response on stdout ~
```

Expected Output Download

```
1  egg
```