2785. Sort Vowels in a String

Solved

Medium

Topics

Companies

Hint

Given a 0-indexed string s, permute s to get a new string t such that:

All consonants remain in their original places. More formally, if there is an index i with 0 <= i < s.length such that s[i] is a consonant, then t[i] = s[i].

The vowels must be sorted in the nondecreasing order of their ASCII values. More formally, for pairs of indices i, j with 0 <= i < j < s.length such that s[i] and s[j] are vowels, then t[i] must not have a higher ASCII value than t[j].

Return the resulting string.

The vowels are 'a', 'e', 'i', 'o', and 'u', and they can appear in lowercase or uppercase. Consonants comprise all letters that are not vowels.

Example 1:

Input: s = "lEetcOde"

Output: "lEOtcede"

Explanation: 'E', 'O', and 'e' are the vowels in s; 'l', 't', 'c', and 'd' are all consonants. The vowels are sorted according to their ASCII values, and the consonants remain in the same places.

Example 2:

Input: s = "lYmpH"

Output: "lYmpH"

Explanation: There are no vowels in s (all characters in s are consonants), so we return "lYmpH".

class Solution {

public String sortVowels(String s) {

int[] vowels = new int[10]; // Array to count vowels

for (char c : s.toCharArray()) {

int idx = getVowelIndex(c);

if (idx != -1) {

vowels[idx]++;

}

}

StringBuilder sb = new StringBuilder();

for (char c : s.toCharArray()) {

if (isVowel(c)) {

for (int i = 0; i < vowels.length; i++) {

if (vowels[i] > 0) {

sb.append(getVowelFromIndex(i));

vowels[i]--;

break;

}

}

} else {

sb.append(c);

}

}

return sb.toString();

}

private boolean isVowel(char c) {

return "AEIOUaeiou".indexOf(c) >= 0;

}

private int getVowelIndex(char c) {

String vowels = "AEIOUaeiou";

return vowels.indexOf(c);

}

private char getVowelFromIndex(int idx) {

String vowels = "AEIOUaeiou";

return vowels.charAt(idx);

}

}