345. Reverse Vowels of a String

Solved ⊘

Easy   ◇ Topics   🔒 Companies

Given a string `s`, reverse only all the vowels in the string and return it.

The vowels are `'a'`, `'e'`, `'i'`, `'o'`, and `'u'`, and they can appear in both lower and upper cases, more than once.

Example 1:

> **Input:** s = `"IceCreAm"`
>
> **Output:** `"AceCreIm"`
>
> **Explanation:**
>
> The vowels in `s` are `['I', 'e', 'e', 'A']`. On reversing the vowels, s becomes `"AceCreIm"`.

Example 2:

> **Input:** s = `"leetcode"`
>
> **Output:** `"leotcede"`

```python
class Solution:
    def reverseVowels(self, s: str) -> str:
        vowels = set('aeiouAEIOU')
        s = list(s)
        left, right = 0, len(s) - 1
        while left < right:
            if s[left] not in vowels:
                left += 1
            elif s[right] not in vowels:
                right -= 1
            else:
                s[left], s[right] = s[right], s[left]
                left += 1
                right -= 1
        return ''.join(s)
```

20. Valid Parentheses

Easy   ◇ Topics   🔒 Companies   ♀ Hint

Given a string `s` containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.

2. Open brackets must be closed in the correct order.

3. Every close bracket has a corresponding open bracket of the same type.

Example 1:

> **Input:** s = "()"
>
> **Output:** true

Example 2:

> **Input:** s = "()[]{}"
>
> **Output:** true

```python
class Solution:
    def isValid(self, s: str) -> bool:
        hmap={')':'(','}':'{',']':'['}
        stk=[]
        for c in s:
            if c not in hmap:
                stk.append(c)
            else:
                if not stk:
                    return False
                else:
                    popp=stk.pop()
                    if popp!=hmap[c]:
                        return False
        return not stk
```

151. Reverse Words in a String

Solved ✔

Medium | 🏷 Topics | 🔒 Companies

Given an input string s, reverse the order of the **words**.

A **word** is defined as a sequence of non-space characters. The **words** in s will be separated by at least one space.

Return *a string of the words in reverse order concatenated by a single space*.

Note that s may contain leading or trailing spaces or multiple spaces between two words. The returned string should only have a single space separating the words. Do not include any extra spaces.

Example 1:

```
Input: s = "the sky is blue"
Output: "blue is sky the"
```

Example 2:

```
Input: s = "  hello world  "
Output: "world hello"
Explanation: Your reversed string should not contain leading or trailing spaces.
```

```python
class Solution:
    def reverseWords(self, s: str) -> str:
        s = s.strip()
        words = s.split()
        return " ".join(reversed(words))
```