<u>LearnersBucket</u> <u>Practice</u> <u>Blog</u> <u>Youtube</u> <u>Butler-Al</u> <u>Book</u>

Ace your JavaScript Interview. <u>Get my ebook</u>. 100 solved Javascript, 20 solved React, & 2 frontend system design questions (1160+ copies sold). Get a <u>Free preview</u>.

×

Advertisements

. . .

Find missing alphabets to make a string panagram

Posted on December 29, 2018 | by Prashant Yadav

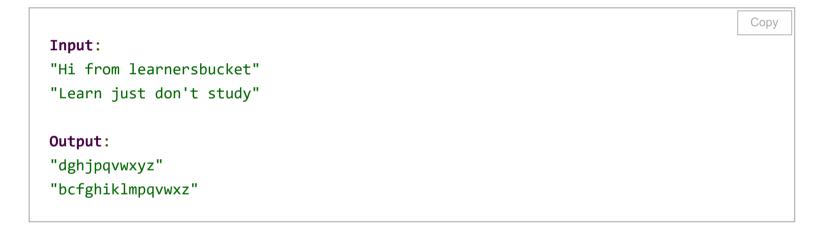
Posted in Algorithms, String | Tagged Easy

An algorithm to find the missing alphabets to make a <u>string</u> Panagram.

We will find all the missing alphabets that will make the string panagram and return the missing characters in alphabetical order.

Panagram: A sentence containing every letter in the English alphabet.

Example



A simple Solution O(n) to find the missing alphabets to make string panagram.

Implementation

- We will create an <u>array</u> of 26 numbers and keep the track of present alphabets by marking them true.
- To convert the alphabets to numbers will calculate the <u>ASCII</u> difference of the number from the small 'a'.
- Once we have marked the present alphabets we can easily print the missing alphabets in order.
- Note we are going to ignore the case here.
- Everything will be written in ES6.

Practically prepare for your JavaScript interview

JavaScript
Revision
JavaScriptConcept Based
Problems
Data Structures
Algorithms
Machine
Coding

<u>Fundamentals</u>

Web

0

```
Сору
let missingAlphabets = (str) => {
 //create a new array and initilize it with 0
  let arr = new Array(26);
  arr.fill(0, 0, 25);
 //convert the string to lowercase
  str.toLowerCase();
 //Mark the present string as true
  for(let i = 0; i < str.length; i++){</pre>
     if (str[i] >= 'a' && str[i] <= 'z') {</pre>
          arr[str[i].charCodeAt(0) - 'a'.charCodeAt(0)] = true;
     }
  }
//Create the string of the missing alphabets
 let missing = '';
 for(let i = 0; i < arr.length; i++){</pre>
     if(!arr[i]){
       missing += String.fromCharCode(97 + i);
     }
  }
return missing;
}
```

```
Input:
console.log(missingAlphabets("Learn just don't study"));
console.log(missingAlphabets("Hi from learnersbucket"));

Output:
"bcfghiklmpqvwxz"
"dghjpqvwxyz"
```

Time Complexity: O(n).

Space Complexity: O(1).

Time and Space complexity

- We are iterating through each character in the given string and marking its
 corresponding number as true in the array that will take O(n), Then we are looping
 through the array and converting the false value to form a string which will take O(26),
 so Time complexity O(n + 26) = O(n).
- As we are using constant space (An array of 26 length no matter how big is the given string), so Space complexity is O(1).

Prepare for your JavaScript Interview
practically on each Interview rounds and grab
that job.

BEGIN LEARNING

Recommended Posts:

Chop string into chunks of given length

Longest Common Subsequence

Find first or last occurrence of a given number in a sorted array

Reverse a stack using recursion.

6 ways to convert string to a number in javascript

Print all the unique 2 digit combinations of given numbers

Program to check if two stacks are equal

Longest Consecutive Sequence

<u>Prev</u>

Leave a F Your email addres	ned. Required fields	are marked *	
Comment			
Start typing			
Name*			
Name* Name			

Advertisements

About Us Contact Us Privacy Policy Advertise











Handcrafted with ♥somewhere in Mumbai

© 2023 <u>LearnersBucket</u> | <u>Prashant Yadav</u>