<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

Minimum characters to delete to make string anagram

Posted on January 26, 2019 | by Prashant Yadav

Posted in Algorithms, String | Tagged Easy

An algorithm to find the minimum characters to be removed to make two <u>strings</u> anagram.

We will implement a simple algorithm in javascript to find the minimum characters to be removed to make two string anagram. Everything will be written in <u>ES6</u>.

Note: Here we will be using lowercase alphabets in the string.

Anagram: An anagram of a string is another string that contains same characters, only the order of characters can be different.

Example

```
Input:
'abc'
'cde'

'ghj'
'jhk'

Output:
// delete a and b from abc
// delete d and e from cde
4

//delete g from ghj
//delete k from jhk
2
```

Implementation

• We will keep track of the characters occurrences in both the string.

Practically prepare for your JavaScript interview

JavaScript
Revision

JavaScriptConcept Based
Problems

Data Structures

Algorithms

Machine
Coding

Web
Fundamentals

Advertisements

• If the a character is present in both the string then we will reduce its count from both the tracker else we will return the combined count of the both trackers.

```
Сору
let makeAnagram = (a, b) => {
   //keep track of the characters count
    let countA = 0;
   let countB = 0;
    //Create an array of all the alphabets to monitor the occurrences
   let arr = new Array(26);
   arr.fill(0, 0, 26);
    //count the characters in the first string
    for (let i = 0; i < a.length; i++){</pre>
        arr[a[i].charCodeAt() - 'a'.charCodeAt()]++;
        countA++;
    }
    //count the characters in the second string
    for (let i = 0; i < b.length; i++){</pre>
        //if character is present in both the string then reduce it counts
        if (arr[b[i].charCodeAt() - 'a'.charCodeAt()]) {
            arr[b[i].charCodeAt() - 'a'.charCodeAt()]--;
            countA--;
        } else {
            countB++;
        }
   }
    //return the combined count of the characters
   return countA + countB;
}
```

```
Input:
console.log(makeAnagram('abc', 'cde'));
console.log(makeAnagram('ghj', 'jhk'));

Output:
4
2
```

The charCodeAt() function returns the ASCII value of the character. ASCII value of a is 97 and ASCII value of b is 98 so 98 - 97 = 1. Then array at index 1 will be incremented or decremented.

Time complexity: O(m + n). Space complexity: O(1).

Time and space complexity

We are iterating through both given strings and their lengths can vary, so Time
 complexity is (m + n) where m is the length of first string and n is the length of the

second string.

• We are using an array of 26 lengths and it will not grow no matter what is the length of the strings i.e we are using constant space, so Space complexity is O(1).

Advertisements

- - -

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

BEGIN LEARNING

Recommended Posts:

How to reverse a string in javascript

Learn how to reverse a linked list recursively

Find all unique paths in a grid

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

Check if two string are anagram of each other

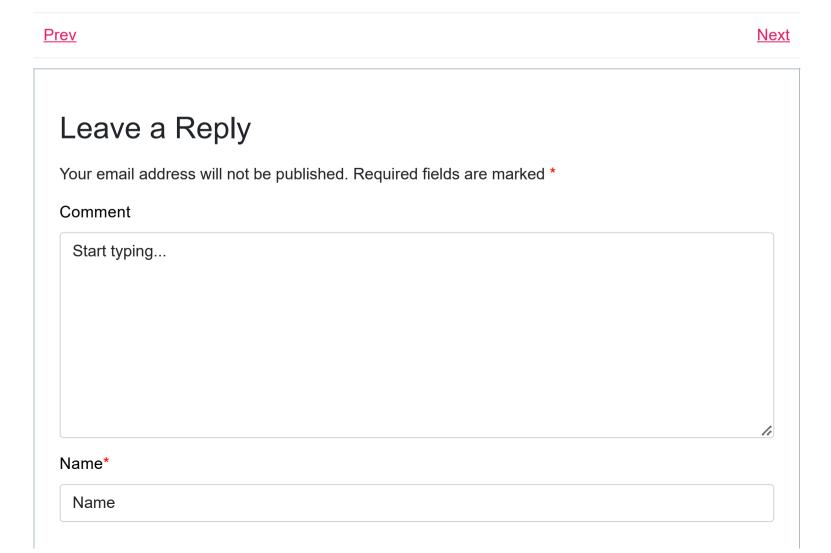
Program to find the GCD of two numbers in javascript.

Program to print the pascal triangle patterns

Count all substrings having character k.

javascript program to find largest of 2 numbers

N meetings in one room



Email*						
Email						
POST COMMENT						

Advertisements

. . .













Handcrafted with ♥somewhere in Mumbai

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