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

- - -

Check if two strings are equal with # backspace characters

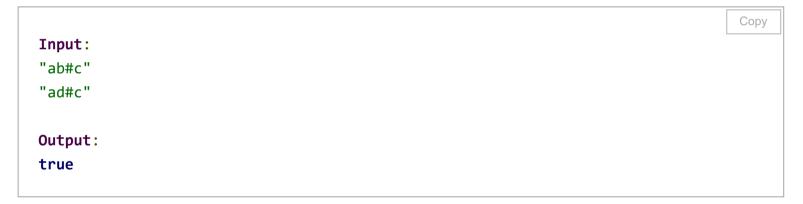
Posted on April 24, 2020 | by Prashant Yadav

Posted in Algorithms, String

Given two <u>strings</u> S and N in a text editor, check if they are equal. Each string contains # which represents backspace characters.

All the characters before backspace character will be deleted.

Example



Both first and second strings becomes ac and thus equal after characters before backspace # are removed.

Check if two strings with backspace are equal or not

Following are the list of steps to solve this problem.

As the backspace removes the character before it.

Advertisements

. . .

Practically prepare for your JavaScript interview

JavaScript
Revision

JavaScriptConcept Based
Problems

Data Structures

Algorithms

Machine
Coding

Web
Fundamentals

Advertisements

- Iterate the string in the reverse order that is from the end.
- If backspace character # is encountered then keep its count.
- Remove all the characters till the count of backspace characters.

As there can be multiple backspaces and/or they can be present in any order. We just have to count them and remove the count of character before them.

Function to remove the characters from the string.

```
Сору
const getString = (S) => {
   //Split the string into array of characters
   let sArr = S.split('');
   let del = 0;
   for(let i = sArr.length - 1; i >= 0; i--){
        //If backspace increase the count and mark the current as empty
        if(sArr[i] === '#'){
          sArr[i] = '';
          del++;
        }else{
          //Remove the characters by marking them empty
          if(del){
             sArr[i] = '';
             del--;
          }
        }
    }
   //Join to form the string again.
    return sArr.join('');
}
```

Function to compare both the strings.

```
const backspaceCompare = (S, N) => {
    return getString(S) === getString(N);
};

Input:
console.log(backspaceCompare('ab##', 'c#d#'));
console.log(backspaceCompare('a##c', '##a#c'));
console.log(backspaceCompare('a#c', 'b'));

Output:
true
true
false
```

In the first example ab## and c#d# both becomes '''

In the second example a##c and #a#c both becomes c.

Time complexity: O(M + N). Space complexity: O(M + N).

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

BEGIN LEARNING

Recommended Posts:

Binary search in javascript

Program to print the next greater element in the array

Implement a Stack using Queue

Program to print the pyramid pattern

Program to sort only positive numbers of the array

Print the last k nodes of the linked list in reverse.

<u>Trapping rain water in javascript</u>

Sort string based frequency of characters.

Convert decimal to binary, octal or hexa in javascript

Implement Iterative binary search

<u>Prev</u> <u>Next</u>

Advertisements

- - -

About Us Contact Us Privacy Policy Advertise











