You can view this report online at : https://www.hackerrank.com/x/tests/1527029/candidates/49987369/report

Full Name:	Vaishali Kapse
Email:	vaishali.kapse@wipro.com
Test Name:	**Front-End Developer Test 2**
Taken On:	7 Feb 2023 22:22:49 IST
Time Taken:	44 min 59 sec/ 60 min
Work Experience:	< 1 years
Linkedin:	https://www.linkedin.com/in/vaishali-kapse-a87b4a1b6
Invited by:	Kavitha
Invited on:	7 Feb 2023 19:32:22 IST
Skills Score:	Problem Solving (Basic) 35/50
Tags Score:	Algorithms 35/50 Easy 35/50 Interviewer Guidelines 35/50 Problem Solving 35/50 Strings 35/50

83.3%

125/150

scored in **Front-End Developer Test 2** in 44 min 59 sec on 7 Feb 2023 22:22:49 IST

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review.

	Question Description	Time Taken	Score	Status
Q1	No Pairs Allowed > Coding	13 min 58 sec	35/ 50	
Q2	Star based Rating > Front-end	23 min 20 sec	50/ 50	
Q3	Print typed text > Front-end	7 min 27 sec	40/ 50	

QUESTION 1

Needs Review

No Pairs Allowed > Coding

Strings | Easy | Algorithms | Problem Solving | Interviewer Guidelines

QUESTION DESCRIPTION

For each word in a list of words, if any two adjacent characters are equal, change one of them. Determine

Score 35

For each word in a list of words, if any two adjacent characters are equal, change one of them. Determine the minimum number of substitutions so the final string contains no adjacent equal characters.

Example

words = ['add', 'boook', 'break']

1. *'add':* change one *d* (1 change)
2. *'boook':* change the middle *o* (1 change)
3. *'break':* no changes are necessary (0 changes)

The return array is *[1,1,0]*.

Function Description

Complete the function *minimalOperations* in the editor below.

minimalOperations has the following parameter(s):

string words[n]: an array of strings

Returns:

int[n]: each element *i* is the minimum substitutions for *words[i]*

Constraints

- $1 \le n \le 100$
- $2 \le length\ of\ words[i] \le 10^5$
- Each character of *words[i]* is in the range *ascii[a-z]*.

▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*, the size of the array *words*.
Each of the next *n* lines contains a string *words[i]*.

▼ Sample Case 0

Sample Input 0

```
STDIN          Function Parameters
-----          -------------------
5          →   words[] Size = 5
ab         →   words[] = [ 'ab','aab','abb', 'abab','abaaaba' ]
aab
abb
abab
abaaaba
```

Sample Output 0

```
0
1
1
0
1
```

Explanation 0

- *words = 'ab'* is already acceptable, so *0* replacements are needed.
- *words = 'aab'* Replace an *'a'* with an appropriate character so *1* replacement.

- *words = 'abb'* is not acceptable. Replace a *'b'* with an appropriate character, again *1* replacement.
- *words = 'abab'* is already acceptable so *0* replacements are needed.
- *words = 'abaaaba'* is not acceptable. Replace the middle *'a'* in *'aaa'*, *1* replacement.

The return array is *[0, 1, 1, 0, 1]*.

INTERVIEWER GUIDELINES

▼ Hint 1

As you iterate through the string, which character(s) need to be tested for equivalence? For each character check only characters adjacent to it on the left.

▼ Hint 2

If you replace a character, can you always assume the replacement differs from the character to its right as well?
Why, and how can you use this fact?

The characters left and right can either be the same or different. There are 25 or 24 letters available in all cases.
This allows you to skip over the next character after a replacement.

▼ Solution

Concepts covered: This problem covers the concepts of strings and arrays.

Optimal Solution:
For each string, start with the character at index 1. Compare each character to the one to its left, with one exception. If the two letters are equal, assume the character to its left remains the same and the current character is replaced. It can always be replaced with a character different from both adjacent characters, left and right. The next character after a replacement can be skipped.

```
def minimalOperations(words):
    ans = []
    for w in words:
        count = 0
        i = 1
        while i < len(w):
            # test for match
            if w[i] == w[i-1]:
                # yes: increment counter and skip the next character
                count += 1
                i += 2
            else:
                # no: move to the next character
                i += 1
        ans.append(count)
    return ans
```

Sub-optimal approach: For each string, iterate its characters, checking if they are equal to the one to their left. If the characters match, replace the current character with '#'. For example: string "abbca". We check pairs one by one, 'ab', 'bb', here characters are the same, so we replace the second character with '#'. Continue checking symbols one by one, '#c', 'ca'. This finishes the process.

```
def minimalOperations(words):

    ans = []

    for i in range(len(words)):
        # replacement counter
        cur_ans = 0
        # convert the string to a list so it is mutable
```

```
        cur_word = list(words[i])

        for j in range(1, len(words[i])):
            # if characters match, replace the current character
            if cur_word[j-1] == cur_word[j]:
                # replace with a character guaranteed to be different
from the next character
                cur_word[j] = "#"
                cur_ans += 1

        ans.append(cur_ans)

    return ans
```

Error Handling:

1. The case of a zero length string must be handled separately.

▼ Complexity Analysis

Time Complexity - O(N) where N is the total number of characters in all words.
Accessing all characters in all words requires O(N) time

Space Complexity - O(1) - For the optimal solution only two integer variables are required.

CANDIDATE ANSWER

Language used: **JavaScript (Node.js)**

```
/*
 * Complete the 'minimalOperations' function below.
 *
 * The function is expected to return an INTEGER_ARRAY.
 * The function accepts STRING_ARRAY words as parameter.
 */

function minimalOperations(words) {
    // Write your code here
    let ans=[];
     for(let j=0;j<words.length;j++){
       let count=0;
       for(let k=0;k<words[j].length-1;k++){
        if(words[j][k]===words[j][k+1]){
            count++;
            k++;
        }
    }
    ans.push(count);
    count=0;
}
     return ans
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	Sample case	Success	1	0.0731 sec	38.2 KB
TestCase 1	Easy	Sample case	Success	1	0.0506 sec	38.4 KB
TestCase 2	Easy	Sample case	Success	1	0.0506 sec	38.4 KB
TestCase 3	Easy	Sample case	Success	6	0.0728 sec	38.2 KB

TestCase 4	Easy	Hidden case	Success	6	0.0583 sec	38.4 KB
TestCase 5	Easy	Sample case	Success	6	0.0563 sec	42.5 KB
TestCase 6	Easy	Hidden case	Success	6	0.0709 sec	42 KB
TestCase 7	Easy	Hidden case	Success	5	0.0928 sec	54.1 KB
TestCase 8	Easy	Hidden case	Success	5	0.1539 sec	54.9 KB
TestCase 9	Easy	Hidden case	Success	5	0.1566 sec	53.5 KB
TestCase 10	Easy	Hidden case	Success	2	0.253 sec	66.7 KB
TestCase 11	Easy	Hidden case	Success	2	0.1753 sec	68.8 KB
TestCase 12	Easy	Hidden case	Success	2	0.1692 sec	68.9 KB
TestCase 13	Easy	Hidden case	Success	2	0.1354 sec	68.6 KB

App detects copy from Ishvinder
- Kavitha Kumarasamy (7 Feb 2023 23:24:41 IST)

QUESTION 2

Correct Answer

Score 50

Star based Rating › Front-end

QUESTION DESCRIPTION

Write an interactive star-based rating component

Specification

Write a functional component that allows a user to enter ratings on the five-star system. The component should initially display five empty stars in a row. When the user hovers over a star, that star as well as all the stars that come before it in the row should become filled (both the empty star and filled star image URLs are provided in the started code below). When the user stops hovering over the star, the component should return to its starting state. Further, if a user clicks on a star, then the component should continue to display that rating even after the user stops hovering; the component should reset only after the user hovers on a star that comes before the clicked star. If the user hovers over later stars, the component should fill those stars but should still display the clicked rating after the hover ends.

CANDIDATE ANSWER

Please open the report on HackerRank for Work to view the candidate's submission

https://www.hackerrank.com/x/tests/1527029/candidates/49987369/report

No Comments

QUESTION 3

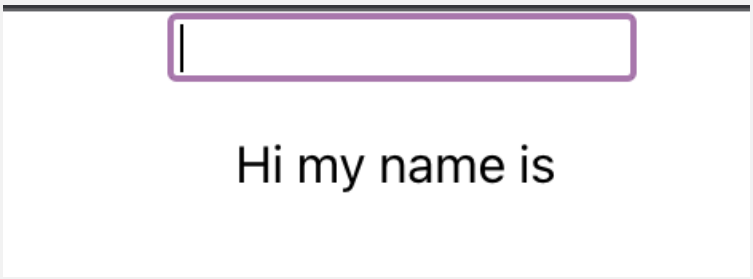
Correct Answer

Score 40

Print typed text › Front-end

QUESTION DESCRIPTION

Specification
Write a functional component that accepts an extended piece of text from the user and prints the text to the screen, beginning with the first word and appending the next word every half-second until the entire text is displayed on the screen. For example, if the user submits “Hi my name is Bob”, the screen should read “Hi”, then “Hi my”, then “Hi my name”, and so on. If the user submits another piece of text reset the display and begin printing the new text. An image of the component taken as it was printing the input “Hi my name is Bob” is shown below:



The finished product

CANDIDATE ANSWER

Please open the report on HackerRank for Work to view the candidate's submission

https://www.hackerrank.com/x/tests/1527029/candidates/49987369/report

No Comments