

Full Name: Remi Chartier

Email: remipr.chartier@gmail.com

Test Name: Mock Test

Taken On: 27 Dec 2021 21:34:00 IST

Time Taken: 13 min 21 sec/ 30 min

Contact Number: +14084751573

Linkedin: http://www.linkedin.com/in/remichartier

Invited by: Ankush

Invited on: 27 Dec 2021 21:33:51 IST

Skills Score:

Tags Score: Algorithms 85/105

Core CS 85/105

Easy 85/105

Problem Solving 85/105

Strings 85/105

problem-solving 85/105

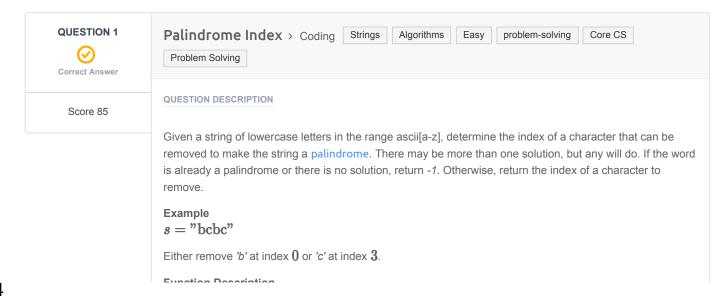
81% 85/105

scored in **Mock Test** in 13 min 21 sec on 27 Dec 2021 21:34:00 IST

Recruiter/Team Comments:

No Comments.





Function Description

Complete the *palindromeIndex* function in the editor below.

palindromeIndex has the following parameter(s):

• string s: a string to analyze

Returns

ullet int: the index of the character to remove or -1

Input Format

The first line contains an integer ${\it q}$, the number of queries.

Each of the next q lines contains a query string s.

Constraints

- $1 \le q \le 20$
- $1 \le \text{length of } s \le 10^5 + 5$
- All characters are in the range ascii[a-z].

Sample Input

```
STDIN Function

-----

3     q = 3

aaab     s = 'aaab' (first query)

baa     s = 'baa' (second query)

aaa     s = 'aaa' (third query)
```

Sample Output

```
3
0
-1
```

Explanation

Query 1: "aaab"

Removing 'b' at index 3 results in a palindrome, so return 3.

Query 2: "baa"

Removing 'b' at index 0 results in a palindrome, so return 0.

Query 3: "aaa"

This string is already a palindrome, so return -1. Removing any one of the characters would result in a palindrome, but this test comes first.

Note: The custom checker logic for this challenge is available here.

CANDIDATE ANSWER

Language used: Python 3

```
if length odd, check until (length/2)
           length = 5, 4/2 = 2 : 0 to 3, will check 0 to 1
          if length even, check until 0 to (length/2 -1)
       1.1.1
       l = len(s)
      mx = int(1/2) if len(s) % 2 == 0 else (int(1/2) -1)
      i = 0
      while i <= mx:
          if s[i] != s[1 - 1 - i]:
             return False
          i += 1
       return True
25 def palindromeIndex(s):
     # Write your code here
      print(f's = {s}')
       # if the world is already a palindrome, or there is no solution, return
29 -1
       if isPalindrome(s):
          return -1
      # check for the word, if any char different from a normal palindrome
33 word,
      # check if removing the char makes it a palindrome.
      # if yes, return the index.
      # if not, return -1
      1 = len(s)
      mx = int(1/2) if len(s) % 2 == 0 else (int(1/2) -1)
      i = 0
      while i <= mx:
41
          if s[i] != s[1 - 1 - i]:
              if isPalindrome(s[0:i] + s[i+1:]):
                  return i
               if isPalindrome(s[0:l-1-i] + s[l-1-i+1:]):
                  return 1 - 1 -i
47
           i += 1
      return -1
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.0505 sec	9.34 KB
Testcase 2	Medium	Hidden case	Success	5	0.0515 sec	9.55 KB
Testcase 3	Medium	Hidden case		5	0.0538 sec	9.5 KB
Testcase 4	Medium	Hidden case		5	0.0489 sec	9.34 KB
Testcase 5	Medium	Hidden case		5	0.0534 sec	9.43 KB
Testcase 6	Medium	Hidden case	Success	5	0.1169 sec	9.78 KB
Testcase 7	Medium	Hidden case	Success	5	0.0729 sec	9.74 KB
Testcase 8	Medium	Hidden case	Success	5	0.1361 sec	9.68 KB
Testcase 9	Hard	Hidden case	Success	10	0.0801 sec	9.78 KB
Testcase 10	Hard	Hidden case		10	0.093 sec	9.76 KB
Testcase 11	Hard	Hidden case	Success	10	0.1338 sec	9.75 KB
Testcase 12	Hard	Hidden case	Success	10	0.0518 sec	9.43 KB
Testcase 13	Hard	Hidden case	Wrong Answer	0	0.0825 sec	9.75 KB
Testcase 14	Hard	Hidden case	Wrong Answer	0	0.0719 sec	9.68 KB



PDF generated at: 27 Dec 2021 16:19:09 UTC