



Full Name:

Email: contato@guifr.com.br Test Name: **Mock Test** 26 May 2022 16:40:59 IST Taken On: Time Taken: 20 min 48 sec/ 30 min Invited by: Ankush Invited on: 26 May 2022 16:39:58 IST Skills Score:

GUILHERME FERNANDES

Algorithms 70/70 Core CS 70/70 Easy 70/70 Strings 70/70 problem-solving

70/70

100% 70/70

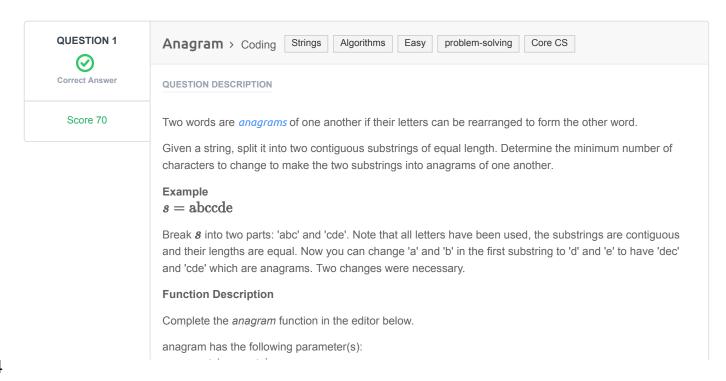
scored in Mock Test in 20 min 48 sec on 26 May 2022 16:40:59 IST

Recruiter/Team Comments:

No Comments.

Tags Score:





string s: a string

Returns

int: the minimum number of characters to change or -1.

Input Format

The first line will contain an integer, \emph{q} , the number of test cases.

Each test case will contain a string s.

Constraints

- $1 \le q \le 100$
- $1 \le |s| \le 10^4$
- s consists only of characters in the range ascii[a-z].

Sample Input

```
6
aaabbb
ab
abc
mnop
xyyx
xaxbbbxx
```

Sample Output

```
3
1
-1
2
0
1
```

Explanation

Test Case #01: We split s into two strings S1='aaa' and S2='bbb'. We have to replace all three characters from the first string with 'b' to make the strings anagrams.

Test Case #02: You have to replace 'a' with 'b', which will generate "bb".

Test Case #03: It is not possible for two strings of unequal length to be anagrams of one another.

Test Case #04: We have to replace both the characters of first string ("mn") to make it an anagram of the other one.

Test Case #05: S1 and S2 are already anagrams of one another.

Test Case #06: Here S1 = "xaxb" and S2 = "bbxx". You must replace 'a' from S1 with 'b' so that S1 = "xbxb".

CANDIDATE ANSWER

Language used: Java 8

```
class Result {

/*

* Complete the 'anagram' function below.

* *

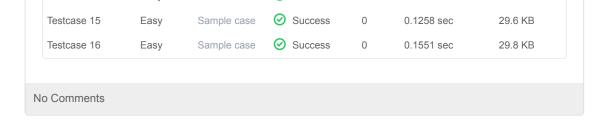
* The function is expected to return an INTEGER.

* The function accepts STRING s as parameter.

*/
```

```
9
       public static int anagram(String s) {
          if(s.length() % 2 == 1){
               return -1;
14
          Map<String, Queue<String>> count = new HashMap<>();
16
          int mid = s.length() / 2;
          String s1 = s.substring(0,mid);
          String s2 = s.substring(mid);
          for(String letter : s2.split("")){
              if(!count.containsKey(letter)){
                  count.put(letter, new ArrayDeque<>());
               count.get(letter).add(letter);
          }
           for(String letter : s1.split("")){
              if(!count.containsKey(letter)){
                  continue;
               count.get(letter).poll();
          }
          int result = 0;
          for(String letter : count.keySet()){
             if(count.get(letter).size() > 0){
                  result += count.get(letter).size();
           }
          return result;
44
45 }
46
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Hidden case	Success	5	0.1865 sec	29.8 KB
Testcase 2	Easy	Hidden case	Success	5	0.1459 sec	29.3 KB
Testcase 3	Easy	Hidden case	Success	5	0.1178 sec	29.4 KB
Testcase 4	Easy	Hidden case	Success	5	0.1492 sec	29.9 KB
Testcase 5	Easy	Hidden case	⊘ Success	5	0.1771 sec	30.3 KB
Testcase 6	Easy	Hidden case	Success	5	0.6516 sec	59.4 KB
Testcase 7	Easy	Hidden case	Success	5	0.3701 sec	43 KB
Testcase 8	Easy	Hidden case	Success	5	0.7074 sec	61.7 KB
Testcase 9	Easy	Hidden case	Success	5	0.2517 sec	42.4 KB
Testcase 10	Easy	Hidden case	⊘ Success	5	0.612 sec	54.8 KB
Testcase 11	Easy	Hidden case	Success	5	0.4355 sec	42.7 KB
Testcase 12	Easy	Hidden case	Success	5	0.4136 sec	54.8 KB
Testcase 13	Easy	Hidden case	Success	5	0.5624 sec	56.4 KB
Testcase 14	Easy	Hidden case	Success	5	0.7363 sec	60.1 KB



PDF generated at: 26 May 2022 11:33:36 UTC