Java → Collections → Map

Java → Sherlock Holmes - 2

131 users solved this problem. Latest completion was **1 day ago**.

Hard 15 minutes Share:

Code Challenge — Write a program

For this problem, imagine that you are Sherlock Holmes. You've deduced that the clues are somehow hidden within the pairs of words that contain only the same letters with the same frequencies. To crack the case, you now need to find out how many characters must be deleted to get such words (character sequences) from the given ones.

For example: for two words "case" and "seal" you'll need to remove characters "c" and "l" respectively to get "ase" and "sea". In this case, the answer is 2 ("c" and "l").

Remember: these "words" are case-insensitive

Sample Input 1:

```
case
seal
```

Sample Output 1:

```
2
```

Code Editor IDE

Solve in IDE

✓ IDE is opened If you don't see your IDE opened, switch to it manually

✓ **Correct, but can be improved**

28 users liked this problem. **0** didn't like it. **What about you?**

Continue

Time limit: 8 seconds Memory limit: 256 MB

Comments (4) Hints (1) Useful links (0) Solutions (0)

Share something, Sergey Kubatko

Post Please do not post solutions here

Sort by: **Last posted**

MK **Maciej Kuchta** 3 months ago Report

Well, I used maps like this: letters were keys and their values were increased each time a letter showed up in a loop. Then comparisons were made. Probably not the most efficient way of doing things but it worked.

♡ 0 Reply

SC **Shadow Crane** 5 months ago Report

This task is easy. But you need to use code from exercise Map->Multiset.

♡ 1 Reply

U2 **User 283193** 7 months ago Report

how do you use maps with this

♡ 0 Reply

SR **Shubham Rajvanshi** 7 months ago Report

for input nnnnlll, ghhnnl
do we have to handle the case of all anagrams like in this case we can have below anagrams if I am not wrong nn,,nl,nnl.
Can somebody elaborate the question.

♡ 0 Reply