6009. Minimum Number of Steps to Make Two Strings Anagram II

My Submissions (/contest/weekly-contest-282/problems/minimum-number-of-steps-to-make-two-strings-anagram-ii/submissions/)

Back to Contest (/contest/weekly-contest-282/)

You are given two strings s and t. In one step, you can append **any character** to either s or t.

Return *the minimum number of steps to make* s *and* t **anagrams** *of each other*.

An **anagram** of a string is a string that contains the same characters with a different (or the same) ordering.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

```
Input: s = "leetcode", t = "coats"
Output: 7
Explanation:
- In 2 steps, we can append the letters in "as" onto s = "leetcode", forming s = "leetcodeas
- In 5 steps, we can append the letters in "leede" onto t = "coats", forming t = "coatsleede
"leetcodeas" and "coatsleede" are now anagrams of each other.
We used a total of 2 + 5 = 7 steps.
It can be shown that there is no way to make them anagrams of each other with less than 7 st
```

Example 2:

```
Input: s = "night", t = "thing"
Output: 0
Explanation: The given strings are already anagrams of each other. Thus, we do not need any further steps.
```

Constraints:

- $1 <= s.length, t.length <= 2 * 10^5$
- s and t consist of lowercase English letters.

JavaScript ▾

```javascript
1    const counter = (a_or_s) => { let m = new Map(); for (const x of a_or_s) m.set(x, m.get(x) + 1 || 1); return m; };
2
3 ▾  const minSteps = (s, t) => {
4        let ms = counter(s), mt = counter(t), res = 0;
5 ▾      for (const [c, occ] of ms) {
6 ▾          if (mt.has(c)) {
7                let tocc = mt.get(c);
8                res += Math.abs(occ - tocc);
9 ▾          } else {
10               res += occ;
11           }
12       }
13 ▾     for (const [c, occ] of mt) {
14 ▾         if (ms.has(c)) {
15               // let socc = ms.get(c);
16               // res += Math.abs(occ - socc);
17 ▾         } else {
18               res += occ;
19           }
20       }
21       return res;
22   };
```

☐ **Custom Testcase**   Use Example Testcases

Run   Submit

Submission Result: Accepted (/submissions/detail/649573144/) ❓   More Details ❯ (/submissions/detail/649573144/)

Share your acceptance!