5541. Count Substrings That Differ by One Character

My Submissions (/contest/biweekly-contest-38/problems/count-substrings-that-differ-by-one-character/submissions/)

Back to Contest (/contest/biweekly-contest-38/)

Given two strings `s` and `t` , find the number of ways you can choose a non-empty substring of `s` and replace a **single character** by a different character such that the resulting substring is a substring of `t` . In other words, find the number of substrings in `s` that differ from some substring in `t` by **exactly** one character.

For example, the underlined substrings in `"computer"` and `"computation"` only differ by the `'e' / 'a'` , so this is a valid way.

Return *the number of substrings that satisfy the condition above*.

A **substring** is a contiguous sequence of characters within a string.

User Accepted:	87
User Tried:	124
Total Accepted:	87
Total Submissions:	131
Difficulty:	Medium

Example 1:

```
Input: s = "aba", t = "baba"
Output: 6
Explanation: The following are the pairs of substrings from s and t that differ by exactly 1 character:
("aba", "baba")
("aba", "baba")
("aba", "baba")
("aba", "baba")
("aba", "baba")
("aba", "baba")
The underlined portions are the substrings that are chosen from s and t.
```

Example 2:

```
Input: s = "ab", t = "bb"
Output: 3
Explanation: The following are the pairs of substrings from s and t that differ by 1 character:
("ab", "bb")
("ab", "bb")
("ab", "bb")
The underlined portions are the substrings that are chosen from s and t.
```

Example 3:

```
Input: s = "a", t = "a"
Output: 0
```

Example 4:

```
Input: s = "abe", t = "bbc"
Output: 10
```

Constraints:

- `1 <= s.length, t.length <= 100`
- `s` and `t` consist of lowercase English letters only.

JavaScript ▼

```
1  /**
2   * @param {string} s
3   * @param {string} t
4   * @return {number}
5   */
6  const countSubstrings = (s, t) => {
```

```
 7        let ns = s.length;
 8        let nt = t.length;
 9        let setS = [];
10        let setT = [];
11 ▾      for (let i = 0; i < ns; i++) {
12 ▾          for (let j = i; j < ns; j++) {
13                let subss = s.slice(i, j + 1);
14                if (subss.length != 0) setS.push(subss);
15            }
16        }
17 ▾      for (let i = 0; i < nt; i++) {
18 ▾          for (let j = i; j < nt; j++) {
19                let subst = t.slice(i, j + 1);
20                if (subst.length != 0) setT.push(subst);
21            }
22        }
23        console.log(setS, setT);
24        let cnt = 0;
25 ▾      setS.forEach(s => {
26 ▾          setT.forEach(t => {
27 ▾              if (ok(s, t, 1) && s != t && s.length == t.length) {
28                    console.log(s, t, ok(s, t, 1));
29                    cnt++;
30                }
31            });
32        });
33        return cnt;
34    };
35
36 ▾ const ok = (short, long) => {
37        let diff = long.length - short.length;
38        let cnt = 0;
39 ▾      for (let i = 0; i < short.length; i++) {
40            if (cnt > 1) return false;
41 ▾          if (short[i] != long[i]) {
42                cnt++;
43            }
44        }
45        if ((diff + cnt) > 1) return false;
46        return true;
47    }
48
```

☐ **Custom Testcase**   (Use Example Testcases)

     ⏵ Run   ☁ Submit

Submission Result: Accepted (/submissions/detail/415254610/) ❓   (More Details ❯ (/submissions/detail/415254610/))

Share your acceptance!

◁ 1