Description | Accepted | Editorial | Submissions | Solutions

2950. Number of Divisible Substrings

Premium

Solved

Medium | Topics | Companies | Hint

Each character of the English alphabet has been mapped to a digit as shown below.

1 ab	2 cde	3 fgh
4 ijk	5 lmn	6 opq
7 rst	8 uvw	9 xyz

A string is **divisible** if the sum of the mapped values of its characters is divisible by its length.

Given a string `s`, return *the number of **divisible substrings** of* `s`.

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

Example 1:

Substring	Mapped	Sum	Length	Divisible?
a	1	1	1	Yes
s	7	7	1	Yes
d	2	2	1	Yes
f	3	3	1	Yes
as	1, 7	8	2	Yes
sd	7, 2	9	2	No
df	2, 3	5	2	No
asd	1, 7, 2	10	3	No
sdf	7, 2, 3	12	3	Yes
asdf	1, 7, 2, 3	13	4	No

```
Input: word = "asdf"
Output: 6
Explanation: The table above contains the details about every substring of word, and we can see that 6 of them are divisible.
```

Example 2:

```
Input: word = "bdh"
Output: 4
Explanation: The 4 divisible substrings are: "b", "d", "h", "bdh".
It can be shown that there are no other substrings of word that are divisible.
```

Example 3:

```
Input: word = "abcd"
Output: 6
Explanation: The 6 divisible substrings are: "a", "b", "c", "d", "ab", "cd".
It can be shown that there are no other substrings of word that are divisible.
```

Constraints:

- `1 <= word.length <= 2000`
- `word` consists only of lowercase English letters.

Seen this question in a real interview before? 1/5

Yes | No

Accepted 1.1K | Submissions 1.4K | Acceptance Rate 76.5%

Topics

Hash Table | String | Counting | Prefix Sum

Companies

1 year - 2 years

Amdocs

Hint 1

Hint 2

Hint 3

Discussion (1)

21 | 1

Code

Python3 | Auto

```python
class Solution:
    def countDivisibleSubstrings(self, word: str) -> int:

        dictx = {"a": 1, "b": 1, "c": 2, "d": 2, "e": 2, "f": 3, "g": 3, "h": 3, "i": 4, "j": 4, "k": 4, "l": 5, "m": 5, "n": 5,
        "o": 6, "p": 6, "q": 6, "r": 7, "s": 7, "t": 7, "u": 8, "v": 8, "w": 8, "x": 9, "y": 9, "z": 9}

        length = len(word)

        res = 0

        for i in range(0, length):
            sumx = 0
            count = 0
            for j in range(i, length):
                sumx += dictx[word[j]]
                count += 1

                if sumx % count == 0:
                    res += 1

        return res
```

Saved

Ln 8, Col 1

Testcase | Test Result

Accepted Runtime: 47 ms

Case 1 | Case 2 | Case 3

Input

word =

"asdf"

Output

6

Expected