5299. Find the K-Beauty of a Number

My Submissions (/contest/biweekly-contest-78/problems/find-the-k-beauty-of-a-number/submissions/) Back to Contest (/contest/biweekly-contest-78/)

User Accepted:	39
User Tried:	47
Total Accepted:	39
Total Submissions:	47
Difficulty:	Easy

The **k-beauty** of an integer `num` is defined as the number of **substrings** of `num` when it is read as a string that meet the following conditions:

- It has a length of `k`.
- It is a divisor of `num`.

Given integers `num` and `k`, return *the k-beauty of* `num`.

Note:

- **Leading zeros** are allowed.
- `0` is not a divisor of any value.

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

Example 1:

```
Input: num = 240, k = 2
Output: 2
Explanation: The following are the substrings of num of length k:
- "24" from "240": 24 is a divisor of 240.
- "40" from "240": 40 is a divisor of 240.
Therefore, the k-beauty is 2.
```

Example 2:

```
Input: num = 430043, k = 2
Output: 2
Explanation: The following are the substrings of num of length k:
- "43" from "430043": 43 is a divisor of 430043.
- "30" from "430043": 30 is not a divisor of 430043.
- "00" from "430043": 0 is not a divisor of 430043.
- "04" from "430043": 4 is not a divisor of 430043.
- "43" from "430043": 43 is a divisor of 430043.
Therefore, the k-beauty is 2.
```

Constraints:

- $1 <= num <= 10^9$
- `1 <= k <= num.length` (taking `num` as a string)

JavaScript

```
const divisorSubstrings = (x, k) => {
    let s = x + '', n = s.length, res = 0;
    for (let i = 0; i < n; i++) {
        for (let j = i; j < n; j++) {
            let sub = s.slice(i, j + 1), len = j - i + 1;
            if (len == k && x % (sub - '0') == 0) res++;
        }
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/699308443/) | More Details > (/submissions/detail/699308443/)

Share your acceptance!