5980. Divide a String Into Groups of Size k

My Submissions (/contest/weekly-contest-276/problems/divide-a-string-into-groups-of-size-k/submissions/)

Back to Contest (/contest/weekly-contest-276/)

User Accepted:	33
User Tried:	43
Total Accepted:	33
Total Submissions:	42
Difficulty:	Easy

A string `s` can be partitioned into groups of size `k` using the following procedure:

- The first group consists of the first `k` characters of the string, the second group consists of the next `k` characters of the string, and so on. Each character can be a part of **exactly one** group.
- For the last group, if the string **does not** have `k` characters remaining, a character `fill` is used to complete the group.

Note that the partition is done so that after removing the `fill` character from the last group (if it exists) and concatenating all the groups in order, the resultant string should be `s`.

Given the string `s`, the size of each group `k` and the character `fill`, return *a string array denoting the **composition of every group*** `s` *has been divided into, using the above procedure*.

Example 1:

```
Input: s = "abcdefghi", k = 3, fill = "x"
Output: ["abc","def","ghi"]
Explanation:
The first 3 characters "abc" form the first group.
The next 3 characters "def" form the second group.
The last 3 characters "ghi" form the third group.
Since all groups can be completely filled by characters from the string, we do not need to use fill.
Thus, the groups formed are "abc", "def", and "ghi".
```

Example 2:

```
Input: s = "abcdefghij", k = 3, fill = "x"
Output: ["abc","def","ghi","jxx"]
Explanation:
Similar to the previous example, we are forming the first three groups "abc", "def", and "ghi".

For the last group, we can only use the character 'j' from the string. To complete this group, we add 'x' twice.
Thus, the 4 groups formed are "abc", "def", "ghi", and "jxx".
```

Constraints:

- `1 <= s.length <= 100`
- `s` consists of lowercase English letters only.
- `1 <= k <= 100`
- `fill` is a lowercase English letter.

JavaScript

```javascript
const divideString = (s, k, fill) => {
    let n = s.length, i, res = [];
    for (i = 0; i + k < n; i+= k) {
        let t = s.slice(i, i + k);
        res.push(t);
    }
    let last = s.slice(i);
    if (last.length < k) last = last.padEnd(k, fill);
    res.push(last);
    return res;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/620652112/) More Details (/submissions/detail/620652112/)

Share your acceptance!