5466. Maximum Number of Non-Overlapping Substrings

Given a string `s` of lowercase letters, you need to find the maximum number of **non-empty** substrings of `s` that meet the following conditions:

1. The substrings do not overlap, that is for any two substrings `s[i..j]` and `s[k..l]`, either `j < k` or `i > l` is true.
2. A substring that contains a certain character `c` must also contain all occurrences of `c`.

Find *the maximum number of substrings that meet the above conditions*. If there are multiple solutions with the same number of substrings, *return the one with minimum total length.* It can be shown that there exists a unique solution of minimum total length.

Notice that you can return the substrings in **any** order.

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

Example 1:

```
Input: s = "adefaddaccc"
Output: ["e","f","ccc"]
Explanation: The following are all the possible substrings that meet the conditions:
[
  "adefaddaccc"
  "adefadda",
  "ef",
  "e",
  "f",
  "ccc",
]
If we choose the first string, we cannot choose anything else and we'd get only 1. If we c
```
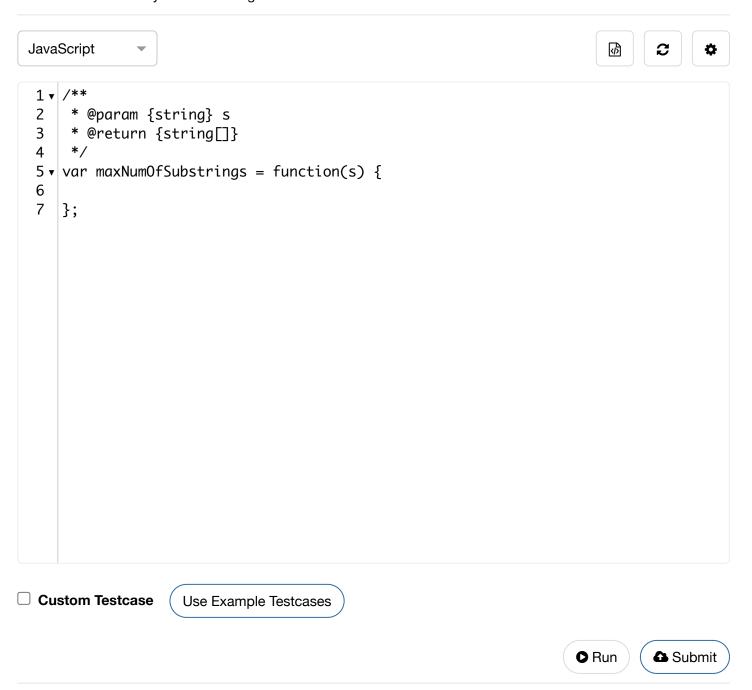
Example 2:

```
Input: s = "abbaccd"
Output: ["d","bb","cc"]
Explanation: Notice that while the set of substrings ["d","abba","cc"] also has length 3,
```

Constraints:

- `1 <= s.length <= 10^5`

- s contains only lowercase English letters.

JavaScript ▾

```
1 ▾ /**
2    * @param {string} s
3    * @return {string[]}
4    */
5 ▾ var maxNumOfSubstrings = function(s) {
6
7  };
```

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

● Run   ☁ Submit