6177. Optimal Partition of String

My Submissions (/contest/weekly-contest-310/problems/optimal-partition-of-string/submissions/) | Back to Contest (/contest/weekly-contest-310/)

Given a string `s`, partition the string into one or more **substrings** such that the characters in each substring are **unique**. That is, no letter appears in a single substring more than **once**.

Return *the **minimum** number of substrings in such a partition*.

Note that each character should belong to exactly one substring in a partition.

User Accepted:	831
User Tried:	954
Total Accepted:	831
Total Submissions:	977
Difficulty:	Medium

Example 1:

```
Input: s = "abacaba"
Output: 4
Explanation:
Two possible partitions are ("a","ba","cab","a") and ("ab","a","ca","ba").
It can be shown that 4 is the minimum number of substrings needed.
```

Example 2:

```
Input: s = "ssssss"
Output: 6
Explanation:
The only valid partition is ("s","s","s","s","s","s").
```

Constraints:

- $1 <= s.length <= 10^5$
- `s` consists of only English lowercase letters.

JavaScript ▾   [</>] [↻] [⚙]

```javascript
const partitionString = (s) => {
    let se = new Set(), d = [], cur = '';
    for (const c of s) {
        if (se.has(c)) {
            d.push(cur);
            cur = '';
            se.clear();
        }
        cur += c;
        se.add(c);
    }
    return d.length + (cur.length ? 1 : 0);
};
```

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

▶ Run   ☁ Submit

Submission Result: Accepted (/submissions/detail/796743107/) ❓   (More Details ❯ (/submissions/detail/796743107/))

Share your acceptance!

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)