2405. Optimal Partition of String

Description

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.

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.