# Maximize partitions in a String

Given a string **s** of lowercase English alphabets, your task is to return the **maximum** number of substrings formed, after possible **partitions** (probably zero) of **s** such that **no two** substrings have a **common character**.

**Examples:**

**Input: s =** "acbbcc"
**Output:** 2
**Explanation:** "a" and "cbbcc" are two substrings that do not share any characters between them.

**Input**: s = "ababcbacadefegdehijhklij"
**Output**: 3
**Explanation**: Partitioning at the index 8 and at 15 produces three substrings: “ababcbaca”, “defegde”, and “hijhklij” such that none of them have a common character. So, the maximum number of substrings formed is 3.

**Input**: s = "aaa"
**Output**: 1
**Explanation**: Since the string consists of same characters, no further partition can be performed. Hence, the number of substring (here the whole string is considered as the substring) is 1.

**Constraints:**
$1 \le s.size() \le 10^5$
$'a' \le s[i] \le 'z'$