2430. Maximum Deletions on a String

Description

You are given a string `s` consisting of only lowercase English letters. In one operation, you can:

- Delete **the entire string** `s`, or
- Delete the **first** `i` letters of `s` if the first `i` letters of `s` are **equal** to the following `i` letters in `s`, for any `i` in the range `1 <= i <= s.length / 2`.

For example, if `s = "ababc"`, then in one operation, you could delete the first two letters of `s` to get `"abc"`, since the first two letters of `s` and the following two letters of `s` are both equal to `"ab"`.

Return *the **maximum** number of operations needed to delete all of* `s`.

Example 1:

```
Input: s = "abcabcdabc"
Output: 2
Explanation:
- Delete the first 3 letters ("abc") since the next 3 letters are equal. Now, s = "abcdabc".
- Delete all the letters.
We used 2 operations so return 2. It can be proven that 2 is the maximum number of operations needed.
Note that in the second operation we cannot delete "abc" again because the next occurrence of "abc" does not happen in the next 3 letters.
```

Example 2:

```
Input: s = "aaabaab"
Output: 4
Explanation:
- Delete the first letter ("a") since the next letter is equal. Now, s = "aabaab".
- Delete the first 3 letters ("aab") since the next 3 letters are equal. Now, s = "aab".
- Delete the first letter ("a") since the next letter is equal. Now, s = "ab".
- Delete all the letters.
We used 4 operations so return 4. It can be proven that 4 is the maximum number of operations needed.
```

Example 3:

```
Input: s = "aaaaa"
Output: 5
Explanation: In each operation, we can delete the first letter of s.
```

Constraints:

- `1 <= s.length <= 4000`
- `s` consists only of lowercase English letters.