2262. Total Appeal of A String

Description

The **appeal** of a string is the number of **distinct** characters found in the string.

- For example, the appeal of `"abbca"` is `3` because it has `3` distinct characters: `'a'`, `'b'`, and `'c'`.

Given a string `s`, return *the **total appeal of all of its substrings.***

A **substring** is a contiguous sequence of characters within a string.

Example 1:

```
Input: s = "abbca"
Output: 28
Explanation: The following are the substrings of "abbca":
- Substrings of length 1: "a", "b", "b", "c", "a" have an appeal of 1, 1, 1, 1, and 1 respectively. The sum is 5.
- Substrings of length 2: "ab", "bb", "bc", "ca" have an appeal of 2, 1, 2, and 2 respectively. The sum is 7.
- Substrings of length 3: "abb", "bbc", "bca" have an appeal of 2, 2, and 3 respectively. The sum is 7.
- Substrings of length 4: "abbc", "bbca" have an appeal of 3 and 3 respectively. The sum is 6.
- Substrings of length 5: "abbca" has an appeal of 3. The sum is 3.
The total sum is 5 + 7 + 7 + 6 + 3 = 28.
```

Example 2:

```
Input: s = "code"
Output: 20
Explanation: The following are the substrings of "code":
- Substrings of length 1: "c", "o", "d", "e" have an appeal of 1, 1, 1, and 1 respectively. The sum is 4.
- Substrings of length 2: "co", "od", "de" have an appeal of 2, 2, and 2 respectively. The sum is 6.
- Substrings of length 3: "cod", "ode" have an appeal of 3 and 3 respectively. The sum is 6.
- Substrings of length 4: "code" has an appeal of 4. The sum is 4.
The total sum is 4 + 6 + 6 + 4 = 20.
```

Constraints:

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