1830. Minimum Number of Operations to Make String Sorted

Description

You are given a string `s` (**0-indexed**). You are asked to perform the following operation on `s` until you get a sorted string:

1. Find **the largest index** `i` such that `1 <= i < s.length` and `s[i] < s[i - 1]`.
2. Find **the largest index** `j` such that `i <= j < s.length` and `s[k] < s[i - 1]` for all the possible values of `k` in the range `[i, j]` inclusive.
3. Swap the two characters at indices `i - 1` and `j`.
4. Reverse the suffix starting at index `i`.

Return *the number of operations needed to make the string sorted.* Since the answer can be too large, return it **modulo** $10^9 + 7$.

Example 1:

```
Input: s = "cba"
Output: 5
Explanation: The simulation goes as follows:
Operation 1: i=2, j=2. Swap s[1] and s[2] to get s="cab", then reverse the suffix starting at 2. Now, s="cab".
Operation 2: i=1, j=2. Swap s[0] and s[2] to get s="bac", then reverse the suffix starting at 1. Now, s="bca".
Operation 3: i=2, j=2. Swap s[1] and s[2] to get s="bac", then reverse the suffix starting at 2. Now, s="bac".
Operation 4: i=1, j=1. Swap s[0] and s[1] to get s="abc", then reverse the suffix starting at 1. Now, s="acb".
Operation 5: i=2, j=2. Swap s[1] and s[2] to get s="abc", then reverse the suffix starting at 2. Now, s="abc".
```

Example 2:

```
Input: s = "aabaa"
Output: 2
Explanation: The simulation goes as follows:
Operation 1: i=3, j=4. Swap s[2] and s[4] to get s="aaaab", then reverse the substring starting at 3. Now, s="aaaba".
Operation 2: i=4, j=4. Swap s[3] and s[4] to get s="aaaab", then reverse the substring starting at 4. Now, s="aaaab".
```

Constraints:

- `1 <= s.length <= 3000`
- `s` consists only of lowercase English letters.