Problem 3563: Lexicographically Smallest String After Adjacent Removals

Problem Information

Difficulty: Hard
Acceptance Rate: 16.18%
Paid Only: No
Tags: String, Dynamic Programming

Problem Description

You are given a string `s` consisting of lowercase English letters.

You can perform the following operation any number of times (including zero):

* Remove **any** pair of **adjacent** characters in the string that are **consecutive** in the alphabet, in either order (e.g., `'a'` and `'b'`, or `'b'` and `'a'`). * Shift the remaining characters to the left to fill the gap.

Return the **lexicographically smallest** string that can be obtained after performing the operations optimally.

Note: Consider the alphabet as circular, thus `'a'` and `'z'` are consecutive.

Example 1:

Input: s = "abc"

Output: "a"

Explanation:

* Remove `"bc"` from the string, leaving `"a"` as the remaining string. * No further operations are possible. Thus, the lexicographically smallest string after all possible removals is `"a"`.

Example 2:

Input: s = "bcda"

Output: ""

Explanation:

* **■■■■■■■** Remove `"cd"` from the string, leaving `"ba"` as the remaining string. * Remove `"ba"` from the string, leaving `""` as the remaining string. * No further operations are possible. Thus, the lexicographically smallest string after all possible removals is `""`.

Example 3:

Input: s = "zdce"

Output: "zdce"

Explanation:

* Remove `"dc"` from the string, leaving `"ze"` as the remaining string. * No further operations are possible on `"ze"`. * However, since `"zdce"` is lexicographically smaller than `"ze"`, the smallest string after all possible removals is `"zdce"`.

Constraints:

* `1 <= s.length <= 250` * `s` consists only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
string lexicographicallySmallestString(string s) {

}
};
```

Java:

```
class Solution {
public String lexicographicallySmallestString(String s) {

}
}
```

Python3:

```
class Solution:
def lexicographicallySmallestString(self, s: str) -> str:
```