Problem 3561: Resulting String After Adjacent Removals

Problem Information

Difficulty: Medium
Acceptance Rate: 56.06%
Paid Only: No
Tags: String, Stack, Simulation

Problem Description

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

You **must** repeatedly perform the following operation while the string `s` has **at least** two **consecutive** characters:

* Remove the **leftmost** 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 resulting string after no more operations can be performed.

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

Example 1:

Input: s = "abc"

Output: "c"

Explanation:

* Remove `"ab"` from the string, leaving `"c"` as the remaining string. * No further operations are possible. Thus, the resulting string after all possible removals is `"c"`.

Example 2:

Input: s = "adcb"

Output: ""

Explanation:

* Remove `"dc"` from the string, leaving `"ab"` as the remaining string. * Remove `"ab"` from the string, leaving `""` as the remaining string. * No further operations are possible. Thus, the resulting string after all possible removals is `""`.

Example 3:

Input: s = "zadb"

Output: "db"

Explanation:

* Remove `"za"` from the string, leaving `"db"` as the remaining string. * No further operations are possible. Thus, the resulting string after all possible removals is `"db"`.

Constraints:

* `1 <= s.length <= 105` * `s` consists only of lowercase English letters.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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