Problem 761: Special Binary String

Problem Information

Difficulty: Hard
Acceptance Rate: 65.43%
Paid Only: No
Tags: String, Recursion

Problem Description

Special binary strings are binary strings with the following two properties:

* The number of `0`'s is equal to the number of `1`'s. * Every prefix of the binary string has at least as many `1`'s as `0`'s.

You are given a **special binary** string `s`.

A move consists of choosing two consecutive, non-empty, special substrings of `s`, and swapping them. Two strings are consecutive if the last character of the first string is exactly one index before the first character of the second string.

Return _the lexicographically largest resulting string possible after applying the mentioned operations on the string_.

Example 1:

Input: s = "11011000" **Output:** "11100100" **Explanation:** The strings "10" [occuring at s[1]] and "1100" [at s[3]] are swapped. This is the lexicographically largest string possible after some number of swaps.

Example 2:

Input: s = "10" **Output:** "10"

Constraints:

* `1 <= s.length <= 50` * `s[i]` is either `'0'` or `'1'`. * `s` is a special binary string.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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