Problem 1702: Maximum Binary String After Change

Problem Information

Difficulty: Medium
Acceptance Rate: 47.76%
Paid Only: No
Tags: String, Greedy

Problem Description

You are given a binary string `binary` consisting of only `0`'s or `1`'s. You can apply each of the following operations any number of times:

* Operation 1: If the number contains the substring `"00"`, you can replace it with `"10"`. * For example, `"_00_ 010" -> "_10_ 010`" * Operation 2: If the number contains the substring `"10"`, you can replace it with `"01"`. * For example, `"000 _10_ " -> "000 _01_ "`

Return the**maximum binary string** you can obtain after any number of operations. Binary string `x` is greater than binary string `y` if `x`'s decimal representation is greater than `y`'s decimal representation.

Example 1:

Input: binary = "000110" **Output:** "111011" **Explanation:** A valid transformation sequence can be: "0001 _10_ " -> "0001 _01_ " "_00_ 0101" -> "_10_ 0101" "1 _00_ 101" -> "1 _10_ 101" "110 _10_ 1" -> "110 _01_ 1" "11 _00_ 11" -> "11 _10_ 11"

Example 2:

Input: binary = "01" **Output:** "01" **Explanation:** "01" cannot be transformed any further.

Constraints:

* `1 <= binary.length <= 105` * `binary` consist of `'0'` and `'1'`.

Code Snippets

C++:

```
class Solution {
public:
string maximumBinaryString(string binary) {

}
};
```

Java:

```
class Solution {
public String maximumBinaryString(String binary) {

}
}
```

Python3:

```
class Solution:
def maximumBinaryString(self, binary: str) -> str:
```