Problem 3474: Lexicographically Smallest Generated String

Problem Information

Difficulty: Hard
Acceptance Rate: 30.71%
Paid Only: No
Tags: String, Greedy, String Matching

Problem Description

You are given two strings, `str1` and `str2`, of lengths `n` and `m`, respectively.

A string `word` of length `n + m - 1` is defined to be **generated** by `str1` and `str2` if it satisfies the following conditions for **each** index `0 <= i <= n - 1`:

* If `str1[i] == 'T'`, the **substring** of `word` with size `m` starting at index `i` is **equal** to `str2`, i.e., `word[i..(i + m - 1)] == str2`. * If `str1[i] == 'F'`, the **substring** of `word` with size `m` starting at index `i` is **not equal** to `str2`, i.e., `word[i..(i + m - 1)] != str2`.

Return the **lexicographically smallest** possible string that can be **generated** by `str1` and `str2`. If no string can be generated, return an empty string `""`.

Example 1:

Input: str1 = "TFTF", str2 = "ab"

Output: "ababa"

Explanation:

The table below represents the string `"ababa"`

Index | T/F | Substring of length `m` ---|---|--- 0 | `'T'` | "ab" 1 | `'F'` | "ba" 2 | `'T'` | "ab" 3 | `'F'` | "ba" The strings `"ababa"` and `"ababb"` can be generated by `str1` and `str2`.

Return `"ababa"` since it is the lexicographically smaller string.

Example 2:

Input: str1 = "TFTF", str2 = "abc"

Output: ""

Explanation:

No string that satisfies the conditions can be generated.

Example 3:

Input: str1 = "F", str2 = "d"

Output: "a"

Constraints:

* `1 <= n == str1.length <= 104` * `1 <= m == str2.length <= 500` * `str1` consists only of `'T'` or `'F'`. * `str2` consists only of lowercase English characters.

Code Snippets

C++:

```
class Solution {
public:
string generateString(string str1, string str2) {

}
};
```

Java:

```
class Solution {
public String generateString(String str1, String str2) {
```

```
}
}
```

Python3:

```
class Solution:
def generateString(self, str1: str, str2: str) -> str:
```