

Problem 2663: Lexicographically Smallest Beautiful String

Problem Information

Difficulty: **Hard**

Acceptance Rate: 37.72%

Paid Only: No

Tags: String, Greedy

Problem Description

A string is **beautiful** if:

- * It consists of the first k letters of the English lowercase alphabet.
- * It does not contain any substring of length 2 or more which is a palindrome.

You are given a beautiful string s of length n and a positive integer k .

Return the lexicographically smallest string of length n , which is larger than s and is **beautiful**. If there is no such string, return an empty string.

A string a is lexicographically larger than a string b (of the same length) if in the first position where a and b differ, a has a character strictly larger than the corresponding character in b .

- * For example, `"abcd"` is lexicographically larger than `"abcc"` because the first position they differ is at the fourth character, and `d` is greater than `c`.

Example 1:

Input: $s = \text{"abcz"}, k = 26$ **Output:** `"abda"` **Explanation:** The string `"abda"` is beautiful and lexicographically larger than the string `"abcz"`. It can be proven that there is no string that is lexicographically larger than the string `"abcz"`, beautiful, and lexicographically smaller than the string `"abda"`.

Example 2:

****Input:**** s = "dc", k = 4 ****Output:**** "" ****Explanation:**** It can be proven that there is no string that is lexicographically larger than the string "dc" and is beautiful.

****Constraints:****

*`1 <= n == s.length <= 105` *`4 <= k <= 26` *`s` is a beautiful string.

Code Snippets

C++:

```
class Solution {
public:
    string smallestBeautifulString(string s, int k) {

    }
};
```

Java:

```
class Solution {
    public String smallestBeautifulString(String s, int k) {

    }
}
```

Python3:

```
class Solution:
    def smallestBeautifulString(self, s: str, k: int) -> str:
```