

Problem 899: Orderly Queue

Problem Information

Difficulty: Hard

Acceptance Rate: 66.54%

Paid Only: No

Tags: Math, String, Sorting

Problem Description

You are given a string `s` and an integer `k`. You can choose one of the first `k` letters of `s` and append it at the end of the string.

Return _the lexicographically smallest string you could have after applying the mentioned step any number of moves_.

Example 1:

Input: s = "cba", k = 1 **Output:** "acb" **Explanation:** In the first move, we move the 1st character 'c' to the end, obtaining the string "bac". In the second move, we move the 1st character 'b' to the end, obtaining the final result "acb".

Example 2:

Input: s = "baaca", k = 3 **Output:** "aaabc" **Explanation:** In the first move, we move the 1st character 'b' to the end, obtaining the string "aacab". In the second move, we move the 3rd character 'c' to the end, obtaining the final result "aaabc".

Constraints:

* `1 <= k <= s.length <= 1000` * `s` consist of lowercase English letters.

Code Snippets

C++:

```
class Solution {  
public:  
    string orderlyQueue(string s, int k) {  
  
    }  
};
```

Java:

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

Python3:

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