

Problem 2030: Smallest K-Length Subsequence With Occurrences of a Letter

Problem Information

Difficulty: Hard

Acceptance Rate: 39.39%

Paid Only: No

Tags: String, Stack, Greedy, Monotonic Stack

Problem Description

You are given a string `s`, an integer `k`, a letter `letter`, and an integer `repetition`.

Return _the**lexicographically smallest** subsequence of_ `s` _of length_ `k` _that has the letter_ `letter` _appear**at least**_ `repetition` _times_. The test cases are generated so that the `letter` appears in `s` **at least** `repetition` times.

A **subsequence** is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters.

A string `a` is **lexicographically smaller** than a string `b` if in the first position where `a` and `b` differ, string `a` has a letter that appears earlier in the alphabet than the corresponding letter in `b`.

Example 1:

Input: s = "leet", k = 3, letter = "e", repetition = 1 **Output:** "eet" **Explanation:** There are four subsequences of length 3 that have the letter 'e' appear at least 1 time: - "lee" (from "***_lee_** t") - "let" (from "***_le_** e _**t**_ ") - "let" (from "_**l**_ e _**et**_ ") - "eet" (from "I _**eet**_ ") The lexicographically smallest subsequence among them is "eet".

Example 2:

!example-2](<https://assets.leetcode.com/uploads/2021/09/13/smallest-k-length-subsequence.png>)

Input: s = "leetcode", k = 4, letter = "e", repetition = 2
Output: "ecde"
Explanation:
"ecde" is the lexicographically smallest subsequence of length 4 that has the letter "e" appear at least 2 times.

Example 3:

Input: s = "bb", k = 2, letter = "b", repetition = 2
Output: "bb"
Explanation: "bb" is the only subsequence of length 2 that has the letter "b" appear at least 2 times.

Constraints:

* `1 <= repetition <= k <= s.length <= 5 * 104` * `s` consists of lowercase English letters.
* `letter` is a lowercase English letter, and appears in `s` at least `repetition` times.

Code Snippets

C++:

```
class Solution {
public:
    string smallestSubsequence(string s, int k, char letter, int repetition) {
        }
    };
}
```

Java:

```
class Solution {
public String smallestSubsequence(String s, int k, char letter, int repetition) {
        }
    }
}
```

Python3:

```
class Solution:
    def smallestSubsequence(self, s: str, k: int, letter: str, repetition: int)
        -> str:
```